



**HAL**  
open science

# Traitement d'images pour la ségrégation en transport de sédiments par charriage : Morphologie et suivi d'objets

Hugo Lafaye de Micheaux

## ► To cite this version:

Hugo Lafaye de Micheaux. Traitement d'images pour la ségrégation en transport de sédiments par charriage : Morphologie et suivi d'objets. Traitement des images [eess.IV]. Université de Lyon, 2017. Français. NNT: . tel-01609142

**HAL Id: tel-01609142**

**<https://ujm.hal.science/tel-01609142>**

Submitted on 3 Oct 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2017LYSES008

**THESE de DOCTORAT DE L'UNIVERSITE DE LYON**  
opérée au sein de  
**Laboratoire Hubert Curien et IRSTEA Grenoble**

**Ecole Doctorale N° 488**  
**Sciences Ingénierie et Santé (SIS)**

**Spécialité de doctorat** : Sciences et technologie de l'information et de la communication  
**Discipline** : Image, Vision

Soutenue publiquement le 04/05/2017, par :

**Hugo LAFAYE DE MICHEAUX**

---

**Traitement d'images pour la ségrégation  
en transport de sédiments par charriage :  
Morphologie et suivi d'objets**

---

Devant le jury composé de :

CHATEAU Thierry  
CONTE Donatello  
BAILLY Yannick  
BARAT Cécile  
DUCOTTET Christophe  
FREY Philippe  
SIDIBÉ Dro Désiré

PR, Université Blaise Pascal, Institut Pascal  
MCF, HDR, Université François Rabelais, LI  
PR, Université de Franche-Comté, FEMTO-ST  
MCF, VESICON  
PR, Université Jean Monnet, LaHC  
ICPEF, HDR, Université Grenoble Alpes, IRSTEA  
MCF, Université de Bourgogne, LE2I

Rapporteur  
Rapporteur  
Examineur  
Examinatrice  
Directeur de thèse  
Directeur de thèse  
Invité

---

# Résumé

**Titre :** Traitement d'images pour la ségrégation en transport de sédiments par charriage : morphologie et suivi d'objets

Le transport de sédiments en rivières et torrents reste un phénomène mal compris en raison de la polydispersité des particules et de la ségrégation résultante. Il a été mené une étude expérimentale sur un canal permettant d'étudier la ségrégation en charriage d'un mélange de deux classes de billes. Le déplacement collectif des billes est enregistré sous la forme de séquences vidéos. Cette thèse traite des méthodes de traitement d'images développées pour analyser les données obtenues. Premièrement, nous avons développé une méthode de segmentation d'images pour étudier l'influence de l'infiltration de particules fines sur l'évolution d'un lit mobile. Avec cette méthode d'analyse, une étude expérimentale a permis de montrer que l'évolution de la pente du lit présente une décroissance exponentielle. Deuxièmement, nous avons optimisé les algorithmes déterministes de suivi de particules pour permettre l'étude des trajectoires sur l'intégralité du phénomène de ségrégation, ce qui n'était pas possible dans les travaux précédemment effectués à Irstea. Nous avons de plus mis en place des mesures d'évaluation et conçu des vérités terrains afin d'apprécier la qualité des résultats. Des gains de temps, cohérence, précision et mémoire ont été quantifiés. Troisièmement, nous avons développé un nouvel algorithme basé sur le filtrage particulaire à modèles multiples pour mieux gérer les dynamiques complexes des particules et gagner en robustesse. Cette approche permet de prendre en compte les erreurs du détecteur, les corriger et ainsi éviter des difficultés lors du suivi de trajectoires que nous rencontrons notamment avec l'algorithme déterministe.

**Mots-clés :** Imagerie en mécanique des fluides, traitement d'images, détection et suivi d'objets, évaluation d'algorithme, transport de sédiments, charriage.



# Abstract

**Title:** Image processing for segregation in bedload sediment transport: morphology and tracking

Sediment transport in rivers and mountain streams remains poorly understood partly due to the polydispersity of particles and resulting segregation. Experiments in a channel were carried out to study bedload transport of bimodal bead mixtures. The behavior of the beads is recorded through video sequences. This work is about the development of image processing methods to analyse the obtained data. Firstly, we developed a method of image segmentation to study the infiltration of fine particles and its influence on the evolution of bed mobility. Thanks to this method, an experimental study shows that the bed slope evolution follows an exponential decay. Secondly, we optimised deterministic tracking algorithms to enable the study of trajectories on long-duration phenomena of segregation, which was not possible with previous work done at Irstea. Moreover we set up relevant evaluation measures and elaborated ground truth sequences to quantify the results. We observed benefits in execution time, consistency, precision and memory. Thirdly, we developed a new algorithm based on multiple model particle filtering to better deal with complex dynamics of particles and to gain robustness. This approach allows taking unreliable detections into account, correcting them and thus avoiding difficulties in the target tracking as encountered with the deterministic algorithm.

**Keywords:** Fluid mechanics imaging, image processing, detection and tracking, algorithm evaluation, sediment transport, bedload transport.



# Remerciements

Mes premiers mots de remerciement sont à l'attention des membres du jury, Thierry Chateau, Donatello Conte, Yannick Bailly, Cécile Barat et Dro Désiré Sidibé, qui ont accepté de juger mon travail de thèse en venant souvent de loin pour assister à ma soutenance de thèse. Toutes leurs remarques ont influencé mon manuscrit ainsi que contribué au perfectionnement des éventuelles perspectives de travail. Merci pour votre intérêt.

Ensuite, j'aimerais remercier mes deux directeurs de thèse, Philippe Frey et Christophe Ducottet, qui m'ont encadré et guidé dans mon travail de manière incroyable. Chacun à sa façon, ils ont contribué au bon déroulement de ma thèse, en m'apportant leurs conseils de chercheurs avisés et en attisant ma curiosité sur leur domaine respectif.

D'une part, Philippe m'a accueilli à Irstea de la meilleure des manières, avec chaleur et convivialité. Il m'a initié à son domaine de passion, la mécanique des fluides, avec beaucoup d'enthousiasme et de pédagogie. Il a fait part d'une disponibilité à défier toute épreuve en étant présent toujours très rapidement pour éclaircir mes doutes sur le pourquoi du comment des images expérimentales et pour m'accompagner lors des périodes d'écriture. Merci pour tout Philippe.

D'autre part, Christophe m'a apporté toute son expertise et compétence en traitement d'images. Par son implication et ses conseils précieux, il m'a grandement aidé dans le développement des nouveaux algorithmes. Sa patience envers moi et ses mots toujours rassurants ont souvent ravivé ma motivation et eu un impact certain sur ma confiance en moi. Avec constamment des nouvelles idées et suggestions toujours pertinentes, il a bien stimulé mon travail et m'a permis d'en apprendre plus sur le monde dynamique de la recherche. J'admire son optimisme et son modèle de rigueur. Et j'ai aussi beaucoup apprécié la conférence et visite express de l'Arizona qu'on a vécu ensemble. Merci Christophe pour cette collaboration enrichissante.

Je tiens à remercier particulièrement Ashley Dudill pour notre coopération fructueuse sur le traitement de données du canal élargi. Elle m'a apporté un regard plus expérimental à ma vision d'ingénieur et m'a bien sûr grandement aidé lors de l'écriture en anglais. Thank you very much Ashley.

Je veux remercier tout le cadre à Irstea et notamment l'équipe ETNA qui a fait de ces trois années de thèse un réel bonheur dans une très bonne ambiance, notamment grâce à Didier Richard et Mohamed Naaïm. J'aimerais remercier spécifiquement mes deux collègues de bureau Perrine Freydier et Simon Carlados pour leur soutien, qui était réciproque je l'espère. J'aimerais également remercier les collègues des nombreuses discussions aux pauses cafés et repas de midi Gaëtan Pulfer, Raphaël Maurin, Coraline Bel, Felix Philippe, Alexandra Carrière et Gilles Nicolet ; les amateurs voire professionnels de mots-croisés Catherine Lukie et Guillaume Dupouy ; toute l'équipe des grimpeurs du midi Hervé Bellot, Guillaume Piton, Ségolène Mejean, Antoine Wautier, Mylène Bonnefoy, Caroline Le Bouteiller, Renaud

## REMERCIEMENTS

---

Jaunatre et Stéphane Lambert ; et Christian Eymond-Gris pour sa gentillesse et implication dans la réparation des vélos. Merci à tous et pour tout, c'était essentiel.

Je remercie également toutes les personnes qui m'ont chaleureusement accueilli et intégré au Laboratoire Hubert Curien et qui ont fait de mes courts séjours à St-Étienne de bons moments sociaux en plus d'être professionnels. Merci notamment à l'équipe Image avec Christophe Ducottet et Cécile Barat, et à l'équipe Crypto avec Brice Colombier, Cédric Marchand, Lilian Bossuet, Alain Aubert, Florent Bernard et Pierre-Louis Cayrel pour ces sympathiques pauses cafés riches en discussions et anecdotes et pour les midis foot favorisant cohésion et convivialité.

J'aimerais remercier de manière assez spéciale mes kinés Guillaume Provost, Véronique Homans, Johan Roche, Mickael Corvi, Manuel François et Mona-Lisa Nechita, avec qui j'ai passé énormément de temps, et qui m'ont soutenu et soulagé de mes maux de doigt (tout en m'en infligeant évidemment...) pendant près d'un an et demi. Merci beaucoup pour votre sympathie et soutien, cela a été primordial dans mon équilibre psychologique.

Un grand merci à mes colocataires Firmin Fontaine, Myrtille Reinhard, Corentin Moitié et Lucile Boust qui, bien plus qu'une maison partagée, furent et sont d'un support quotidien indispensable. Je remercie aussi tous mes amis Thibaut Capblancq, Olivier De Smet, Adrien Dulac, Thibaud Michel, Andjali Cochet, Camille Posocco et tant d'autres pour leur soutien inconditionnel. Enfin, je veux remercier mes parents, ma soeur, mon beau-frère et mes grands-parents pour être la base solide de ma vie et pour avoir confectionné et géré le pot de thèse à merveille. Merci à tous mes proches.

Finalement et non des moindres, je remercie grandement Marion pour tous les jours passés ensemble dans cette sacrée aventure, et pour bien plus encore !

# Table des matières

Résumé	iii
Abstract	v
Remerciements	vii
Table des matières	ix
Liste des figures	xiii
Liste des tableaux	xv
Liste des notations	xvii
Introduction	1
<b>PARTIE 1 : CONTEXTE SCIENTIFIQUE ET ÉTAT DE L'ART</b>	<b>5</b>
<b>I Contexte et objectifs des travaux</b>	<b>7</b>
I.1 Ségrégation en charriage : un phénomène granulaire . . . . .	7
I.1.1 Transport solide par charriage . . . . .	7
I.1.2 Phénomènes et enjeux liés au tri granulométrique . . . . .	8
I.1.3 Ségrégation par taille . . . . .	10
I.1.4 Approche à l'échelle de la particule . . . . .	10
I.2 Approche expérimentale menée à Irstea . . . . .	10
I.2.1 Objectifs scientifiques . . . . .	11
I.2.2 Dispositif expérimental . . . . .	12
I.2.3 Algorithmes de traitement d'images . . . . .	18
I.2.4 Limitations actuelles . . . . .	18
I.3 Objectifs de la thèse . . . . .	19
I.4 Conclusion . . . . .	19
<b>II Revue des méthodes de suivi d'objets</b>	<b>21</b>
II.1 Suivi de particules en mécanique des fluides . . . . .	21
II.1.1 Vélocimétrie par images de particules (PIV) . . . . .	22
II.1.2 Vélocimétrie par suivi de particules (PTV) . . . . .	23
II.1.3 Approches utilisées en suivi de sédiments . . . . .	23

## TABLE DES MATIÈRES

---

II.2	Suivi d'objets en vision par ordinateur . . . . .	24
II.2.1	Méthodes déterministes . . . . .	25
II.2.2	Méthodes probabilistes . . . . .	29
II.2.3	Métriques d'évaluation . . . . .	32
II.3	Synthèse et rapprochement à notre problématique . . . . .	35
II.4	Conclusion . . . . .	36
 <b>PARTIE 2 : CONTRIBUTIONS</b>		<b>37</b>
 <b>III Traitement d'images pour l'étude de la mobilité du lit lors de l'infiltration de particules fines</b>		<b>39</b>
III.1	Contexte, problématique et objectifs . . . . .	40
III.2	Algorithme de détection des lignes du lit et de l'eau . . . . .	41
III.2.1	Séquences d'images . . . . .	41
III.2.2	Problème de segmentation . . . . .	41
III.2.3	Description de l'algorithme de détection des lignes . . . . .	42
III.2.4	Validation des résultats . . . . .	45
III.3	Résultats et discussions . . . . .	45
III.3.1	Évolution de la pente du lit . . . . .	47
III.3.2	Évolution de la hauteur d'eau . . . . .	48
III.3.3	Éléments de discussion complémentaires . . . . .	48
III.4	Conclusion . . . . .	50
 <b>IV Algorithme de suivi déterministe : optimisation et évaluation</b>		<b>51</b>
IV.1	Algorithme déterministe de suivi de billes . . . . .	52
IV.1.1	Détection des billes . . . . .	52
IV.1.2	Suivi des billes . . . . .	55
IV.1.3	Estimation des états de mouvement . . . . .	56
IV.1.4	Limitations de cet algorithme . . . . .	59
IV.2	Améliorations de la détection . . . . .	60
IV.2.1	Pré-traitement : suppression de la base du canal . . . . .	60
IV.2.2	Nouveau détecteur de billes noires . . . . .	62
IV.2.3	Nouveau détecteur de ligne d'eau . . . . .	63
IV.2.4	Nouveau détecteur de billes transparentes . . . . .	66
IV.2.5	Synthèse des paramètres de la détection . . . . .	68
IV.3	Optimisation du code . . . . .	69
IV.3.1	Réécriture du code en Matlab . . . . .	70
IV.3.2	Restructuration des données . . . . .	71
IV.3.3	Parallélisation du code . . . . .	73
IV.4	Bases de données d'évaluation . . . . .	75
IV.4.1	Élaboration de vérités terrains . . . . .	76
IV.4.2	Choix des métriques d'évaluation . . . . .	77
IV.5	Résultats de l'algorithme déterministe de suivi et discussion . . . . .	78
IV.5.1	Évaluation des performances sur les vérités terrains . . . . .	78
IV.5.2	Quelques résultats sur une longue séquence . . . . .	79
IV.5.3	Limitations de l'algorithme déterministe de suivi . . . . .	80

---

IV.6 Conclusion . . . . .	81
<b>V Algorithme de suivi par filtrage particulière à modèles multiples</b>	<b>83</b>
V.1 Contexte, problématique et objectifs . . . . .	83
V.2 Algorithme de suivi multi-objets et multi-modèles . . . . .	85
V.2.1 Estimation de la confiance du détecteur . . . . .	85
V.2.2 Filtrage particulière à modèle multiples . . . . .	86
V.2.3 Détails d'implémentation . . . . .	89
V.3 Résultats et discussions . . . . .	91
V.3.1 Algorithmes de comparaison . . . . .	91
V.3.2 Évaluation de l'algorithme de suivi . . . . .	91
V.4 Éléments de discussion complémentaires . . . . .	95
V.5 Conclusion . . . . .	96
<b>Conclusion générale et perspectives</b>	<b>97</b>
1 Contributions et résultats principaux . . . . .	97
2 Perspectives . . . . .	99
<b>Références</b>	<b>103</b>
<b>Annexes</b>	<b>113</b>
A Rappels d'opérateurs de traitement d'images . . . . .	113
A.1 Transformations morphologiques élémentaires . . . . .	113
A.2 Corrélation croisée normalisée . . . . .	117
B Article <i>PSFVIP</i> 2015 . . . . .	118
C Article <i>JGR : Earth Surface</i> (en révision) . . . . .	129
D Article <i>ICIP</i> 2016 . . . . .	155
E Article <i>MVA</i> (en examen) . . . . .	161



# Liste des figures

1	Photo de la partie amont d'un torrent . . . . .	1
2	Exemple d'une image de billes issue de l'expérimentation . . . . .	2
I.1	Schéma des différents modes de transport de sédiments . . . . .	8
I.2	Photos de dommages à l'issue d'un épisode de charriage . . . . .	9
I.3	Photo d'un pavage par tri granulométrique en rivière à graviers . . . . .	9
I.4	Photo du dispositif expérimental à Irstea . . . . .	14
I.5	Schéma du dispositif expérimental à Irstea . . . . .	15
I.6	Images de charriage d'un mélange bimodal de particules en canal étroit . . . . .	17
I.7	Images de charriage d'un mélange bimodal de particules en canal élargi . . . . .	17
II.1	Schéma du dispositif expérimental pour PIV . . . . .	22
II.2	Illustration de l'utilité de la confiance du détecteur . . . . .	30
II.3	Illustrations des faux positifs et faux négatifs sur un détecteur de personnes . . . . .	33
II.4	Illustration d'échanges d'identités de traqueurs . . . . .	34
III.1	Image typique d'un mélange de billes dans le canal élargi . . . . .	41
III.2	Position des quatre régions distinctes dans les images . . . . .	42
III.3	Procédure de détection de la base du canal . . . . .	43
III.4	Procédure de détection des lignes du lit et ligne d'eau . . . . .	44
III.5	Exemples de lignes détectées superposées à l'image . . . . .	46
III.6	Hauteur moyenne de la ligne du lit au cours du temps . . . . .	47
III.7	Évolution de la pente du lit en fonction de la taille des particules fines . . . . .	48
III.8	Évolution de la hauteur d'eau en fonction de la taille des particules fines . . . . .	49
III.9	Évolution des pentes du lit avec les courbes d'ajustement . . . . .	49
IV.1	Image typique d'un mélange de billes avec les éléments à détecter . . . . .	53
IV.2	Procédure de détection des billes noires de <a href="#">Böhm et al. [2006]</a> . . . . .	53
IV.3	Procédure de détection de la ligne d'eau de <a href="#">Hergault et al. [2010]</a> . . . . .	54
IV.4	Procédure de détection des billes transparentes de <a href="#">Hergault et al. [2010]</a> . . . . .	55
IV.5	Visualisation de trajectoires de billes sur une séquence de 50 images . . . . .	57
IV.6	Procédure de création du masque binaire de la base du canal . . . . .	61
IV.7	Procédure de pré-traitement d'une image . . . . .	61
IV.8	Procédure de détection des billes noires . . . . .	63
IV.9	Temps d'exécution du détecteur de billes noires en fonction du nombre de billes . . . . .	64
IV.10	Procédure de détection de la ligne d'eau . . . . .	65
IV.11	Procédure de détection des billes transparentes . . . . .	68

## LISTE DES FIGURES

---

IV.12	Modèle d'anneau pour le détecteur de billes transparentes . . . . .	69
IV.13	Temps d'exécution de la détection des billes en fonction du nombre d'images	75
IV.14	Illustration de faux positif et faux négatif sur une image avec des billes . . .	78
IV.15	Visualisation des trajectoires de billes transparentes sur 15 min . . . . .	80
IV.16	Résultats de l'algorithme déterministe de suivi sur une longue séquence . . .	81
V.1	Vue d'ensemble des différentes parties de l'algorithme de suivi multi-objets .	85
V.2	Algorithme de suivi basé sur un filtrage particulaire à modèles multiples . .	86
V.3	Évaluation des performances du suivi des billes transparentes . . . . .	93
V.4	Évaluation de la fiabilité de l'état de mouvement des traqueurs . . . . .	94
A.1	Segmentation par ligne de partage des eaux contrôlée par des marqueurs . .	116

# Liste des tableaux

I.1	Caractéristiques des deux configurations du canal expérimental . . . . .	13
IV.1	Paramètres d'entrée/sortie de la création du masque de la base du canal . . .	60
IV.2	Paramètres d'entrée/sortie du pré-traitement d'une image . . . . .	61
IV.3	Paramètres d'entrée/sortie du détecteur de billes noires . . . . .	62
IV.4	Paramètres d'entrée/sortie du détecteur de ligne d'eau . . . . .	66
IV.5	Paramètres d'entrée/sortie du détecteur de billes transparentes . . . . .	68
IV.6	Résumé des paramètres d'entrée/sortie de la détection . . . . .	69
IV.7	Évaluation des performances de l'algorithme déterministe de suivi de billes .	79
V.1	Évaluation des performances des quatre algorithmes de suivi pour comparaison	92



# Liste des notations

*Symbole*   *Unité*   *Description*

## Dispositif expérimental

$FPS$	Hz	Fréquence d'images
$H_b$	mm	Hauteur de la butée
$r_1$	mm	Rayon des billes grossières
$r_2$	mm	Rayon des billes fines
$S$	%	Pente du canal
$u_{max}$	m/s	Vitesse maximale qu'une bille peut avoir
$W$	mm	Largeur du canal
$\Delta_T$	s	Intervalle inter-image
$\Delta_X$	mm	Taille du pixel

## Algorithme de détection de lignes

$l$	px	Longueur de la ligne élément structurant supprimant la ligne d'eau
$n$	images	Nombre d'images consécutives pour l'extraction du fond
$\sigma$	px	Écart-type du filtre gaussien supprimant détails fins et bruit

## Algorithmes de détection de billes

$d$	-	Identifie une détection
$D$	-	Ensemble des coordonnées $(x, y)$ de toutes les billes détectées
$D_{bn}$	-	Ensemble des coordonnées $(x, y)$ des billes noires détectées
$D_{te}$	-	Ensemble des coordonnées $(x, y)$ de la ligne d'eau détectée
$D_{bt}$	-	Ensemble des coordonnées $(x, y)$ des billes transparentes détectées
$e$	mm	Épaisseur de la ligne d'eau
$f$	-	Identifie une image d'une séquence d'images
$f_\emptyset$	-	Image du canal vide
$f_p$	-	Image issue du pré-traitement de l'image $f$
$g_{bc}$	-	Masque binaire de la base du canal
$h$	niveau de gris	Valeur du paramètre de l'opérateur $h_{concave}$
$K_{bt}$	-	Modèle de bille transparente
$x$	px	Coordonnée horizontale dans une image
$y$	px	Coordonnée verticale dans une image
$\tau_n$	-	Seuil d'intensité des pixels noirs
$\tau_t$	-	Seuil de corrélation pour la détection des billes transparentes

**Algorithme d'association**

$g(d, tr)$	-	Fonction du coût d'association entre une détection $d$ et un traqueur $tr$
$r_s$	mm	Rayon de la région de recherche des associations
$tr$	-	Identifie un traqueur
$u$	m/s	Composante horizontale de la vitesse dans une image
$ u _{max}$	m/s	Plus grande norme de vitesse de l'instant précédent
$u^{tr}$	-	Vecteur vitesse $(u, v)$ d'un traqueur $tr$
$v$	m/s	Composante verticale de la vitesse dans une image
$x^d$	-	Coordonnées $(x, y)$ d'une détection $d$
$\hat{x}^{tr}$	-	Coordonnées $(x, y)$ prédites d'un traqueur $tr$
$\alpha$	-	Proportion du terme distance dans la fonction $g$
$\beta$	-	Proportion du terme vitesse dans la fonction $g$

**Algorithme d'estimation des états de mouvement**

$a_n$	px	Distance au plus proche voisin
$a^*$	px	Distance de référence entre deux voisins
$n_i$	images	Nombre d'images pour calculer la vitesse moyenne locale et longueur de la fenêtre glissante pour le filtrage des états de mouvement
$u_b$	m/s	Vecteur vitesse $(u, v)$ d'une bille
$u_{rest}$	m/s	Vitesse seuil d'une bille en repos
$u_{salt}$	m/s	Vitesse seuil d'une bille en saltation
$\varepsilon$	-	Paramètre seuil pour la condition de distance entre voisins différenciant roulement et saltation

**Algorithme de suivi par filtrage particulaire à modèles multiples**

$c_t^i$	-	Vecteur d'état de la particule $i$ à l'instant $t$
$D^*$	-	Ensemble des coordonnées $(x, y)$ et corrélation associée des billes transparentes détectées avec $\tau_t^*$
$d_c(x^{obs})$	-	Estimation de la confiance du détecteur à la position $x^{obs}$
$i$	-	Identifie une particule
$\mathcal{I}(tr)$	-	Fonction indicatrice d'une association faite avec un traqueur $tr$
$k_{D^*}$	-	Coefficient de normalisation dans le calcul de $d_c$
$N$	particules	Nombre de particules du filtrage particulaire
$nbN^{obs}$	billes	Nombre de billes voisines à la détection observée
$nbN_{rest}$	billes	Nombre seuil de billes voisines pour l'état de mouvement repos
$nbN_{salt}$	billes	Nombre seuil de billes voisines pour l'état de mouvement saltation
$s_t^i$	-	État de mouvement d'une particule $i$ à l'instant $t$
$s^{obs}$	-	État de mouvement d'une détection observée
$u_t^i$	-	Vecteur vitesse $(u, v)$ d'une particule $i$ à l'instant $t$
$u^{obs}$	-	Vecteur vitesse $(u, v)$ d'une détection observée
$u_t^{obs}$	-	Vecteur vitesse $(u, v)$ d'une détection observée à l'instant $t$
$xcorr(d)$	-	Coefficient de corrélation croisée normalisée d'une détection $d$
$x_t^i$	-	Coordonnées $(x, y)$ d'une particule $i$ à l'instant $t$
$x^{obs}$	-	Coordonnées $(x, y)$ d'une détection observée
$x_t^{obs}$	-	Coordonnées $(x, y)$ d'une détection observée à l'instant $t$
$w_t^i$	-	Poids d'importance d'une particule $i$ à l'instant $t$

---

$\mathbf{z}_t$	-	Vecteur d'état de la variable d'observation à l'instant $t$
$\varepsilon_{rest}^{(u,v)}$	-	Variance du bruit sur la vitesse du modèle de repos
$\varepsilon_{roll}^{(u,v)}$	-	Variance du bruit sur la vitesse du modèle de roulement
$\varepsilon_{salt}^{(u,v)}$	-	Variance du bruit sur la vitesse du modèle de saltation
$\varepsilon_{rest}^{(x,y)}$	-	Variance du bruit sur la position du modèle de repos
$\varepsilon_{roll}^{(x,y)}$	-	Variance du bruit sur la position du modèle de roulement
$\varepsilon_{salt}^{(x,y)}$	-	Variance du bruit sur la position du modèle de saltation
$\gamma$	-	Constante de récompense dans le calcul des poids d'importance
$\lambda$	-	Constante de décroissance exponentielle dans le calcul de $d_c$
$\pi$	-	Table de probabilité de transition pour le changement de modèle
$\tau_t^*$	-	Seuil de corrélation très bas pour la détection des billes transparentes pour l'estimation de la confiance du détecteur

## LISTE DES NOTATIONS

---

# Introduction

Le transport de sédiments, ou transport solide, dans les rivières a des conséquences majeures sur la sécurité publique, l'évolution du paysage et la gestion des ressources en eau. Il se présente sous la forme d'un transport de particules fines par suspension et de particules plus grossières par charriage. Le charriage est défini par le transport des matériaux les plus grossiers en contact permanent avec le lit par glissement, roulement ou saltation (succession de petits sauts). Le transport solide par charriage est responsable de la formation de structures morphologiques complexes dans les écoulements à forte pente comme les torrents de montagne (voir [Figure 1](#)). Notamment, lors de crues torrentielles, il peut modifier drastiquement la stabilité et la morphologie du cours d'eau et ainsi provoquer des inondations, des dommages sur les infrastructures et sur l'environnement, et dégrader l'habitat aquatique. Il y a donc un enjeu très concret derrière les problèmes scientifiques complexes soulevés par le charriage.

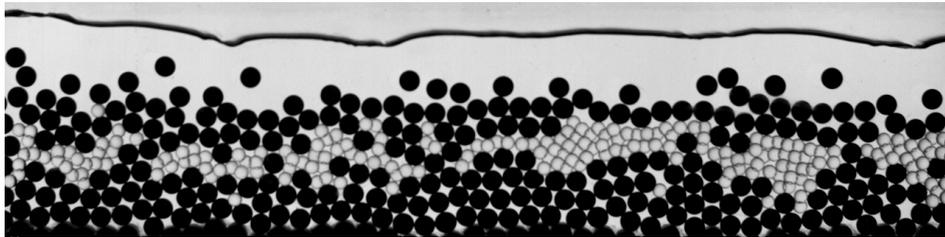


**Figure 1** Photo de la partie amont du torrent du Bens (Isère/Savoie, Auvergne-Rhône-Alpes, France). Ici, le courant s'écoule du premier plan vers l'arrière plan. On observe la formation de structures morphologiques constituées de sédiments de différentes tailles (cailloux). Photo Danieli.

Dans les torrents, une large étendue de tailles de grains constitue le mélange sédimentaire. La distribution des différentes tailles est appelée granulométrie. Un phénomène de tri granulométrique, aussi appelé ségrégation par taille, se produit pendant le transport solide par charriage. Il résulte de diverses tailles de particules et surtout de leurs interactions. C'est un des phénomènes les plus observés en charriage, il peut former un pavage constitué de strates verticales de différentes tailles dans le lit, et bien d'autres structures morphologiques.

Il est cependant difficile à modéliser et il n'existe pas encore de descriptions complètes du processus. L'analyser pour mieux le comprendre est alors essentiel pour l'ingénierie des rivières et torrents, et pour l'amélioration des écosystèmes aquatiques.

Une approche menée à Irstea se concentre sur les processus physiques de transport de sédiments et l'analyse de la ségrégation à l'échelle de la particule pour observer les interactions granulaires. Fondamentalement, il a été mis au point une expérience modèle constituée d'un canal expérimental et de sédiments modélisés par des billes de verre sphériques. Ce dispositif a pour but d'idéaliser le transport sédimentaire dans les cours d'eau à forte pente. Une expérimentation est enregistrée sous la forme de longues séquences d'images (voir un exemple d'image en [Figure 2](#)). Des algorithmes de traitement ont été développés pour la détermination des trajectoires de billes et de leur état de mouvement.



**Figure 2** Exemple d'une image de billes issue de l'expérimentation.

Cette approche pose des problèmes de traitement d'images qui ne sont pas résolus de manière satisfaisante. Ces problèmes relèvent principalement des problématiques de segmentation et de suivi (plus communément appelé *tracking*) dans un contexte applicatif particulier. Nous ne pouvons pas simplement appliquer des algorithmes standards du domaine, mais il faut les adapter et les perfectionner.

Jusqu'à maintenant, le *tracking* de sédiments se fonde sur une approche déterministe qui ne prend pas en compte d'une part la nature particulière des mouvements d'objets considérés (roulement, saltation) et d'autre part les erreurs de détection. La taille des trajectoires est aussi limitée et la performance de cet algorithme n'a pas été évaluée objectivement.

La première contribution de cette thèse est une optimisation de l'algorithme déterministe et son évaluation par rapport à des vérités terrains. Les modifications apportées visent à améliorer les méthodes de traitement d'images et gagner en temps de calcul. L'algorithme optimisé est alors rapide et adapté à de longues séquences, mais il ne permet pas d'étudier des phénomènes transitoires fins par manque de fiabilité des détections.

La deuxième contribution est la proposition d'un nouvel algorithme basé sur le filtrage particulaire qui permet de prendre en compte de manière fine la nature des mouvements considérés. L'originalité de cet algorithme est qu'il est à l'intersection entre les domaines de la mécanique des fluides et du *tracking* en vision par ordinateur. En effet, une connaissance approfondie des déplacements mécaniques de sédiments est apportée pour guider un algorithme de *tracking* probabiliste plus classiquement utilisé en vidéo surveillance et robotique.

Lorsque les objets utilisés sont trop petits pour pouvoir être suivis (particules trop fines), il n'est plus possible de les suivre individuellement. Dans ce cas, on peut s'intéresser à la structure verticale de l'écoulement : forme de la couche de sédiments et hauteur d'eau. Ce problème revient à déterminer les lignes d'eau et du lit, c'est-à-dire segmenter les images de la séquence. La troisième contribution est la proposition d'un algorithme de segmentation spécifique. Il permet de détecter les lignes avec une bonne fiabilité sur toute la séquence.

**Organisation du manuscrit**

Ce manuscrit de thèse s'articule en deux grandes parties et présente au total cinq chapitres, deux chapitres dans la première partie et trois dans la deuxième.

Dans la première partie, nous présentons le contexte général et scientifique, ainsi que l'état de l'art. Le Chapitre **I** introduit le contexte et les objectifs de la thèse. Il définit les termes techniques liés à la mécanique des fluides et décrit le dispositif expérimental mis en place à Irstea pour l'étude de l'interaction entre les particules et le fluide. Il donne les limitations des approches précédentes de traitement d'images et fixe les objectifs pour y remédier. Le Chapitre **II** fait la revue des méthodes de suivi d'objets. Il fait d'abord état des méthodes qui sont utilisées en mécanique des fluides pour le suivi de particules. Il présente ensuite l'état des connaissances sur le suivi d'objets en vision par ordinateur, tout en faisant le rapprochement et le lien avec la problématique qui est la nôtre.

Dans la deuxième partie, nous exposons les différentes contributions que nous avons apportées lors de ce travail de thèse. Le Chapitre **III** détaille les méthodes de traitement d'images qui ont été développées pour permettre l'analyse de l'infiltration de particules fines dans un lit mobile sur le canal élargi. Il résume l'article dont il a fait l'objet et qui a été publié et présenté lors de la conférence *PSFVIP*. Quelques éléments et résultats complémentaires sont ajoutés. Le Chapitre **IV** donne le détail complet de l'algorithme de suivi de billes déterministe en canal étroit. Il décrit chaque optimisation apportée aux précédentes versions de l'algorithme (nouveaux détecteurs, restructuration du code, parallélisation, etc.) et comment évaluer les résultats et performances des algorithmes de suivi de billes. Il montre également quelques résultats sur de longues séquences. Le Chapitre **V** présente un nouvel algorithme de suivi de billes basé sur un filtrage particulaire à modèles dynamiques multiples. Il résume l'article dont il a fait l'objet et qui a été soumis au journal *Machine Vision and Applications*. Quelques éléments de discussion complémentaires sont ajoutés sur les points à améliorer de ce nouvel algorithme.



Première partie

**CONTEXTE SCIENTIFIQUE ET  
ÉTAT DE L'ART**



# Chapitre I

## Contexte et objectifs des travaux

### Sommaire

---

<b>I.1</b>	<b>Ségrégation en charriage : un phénomène granulaire . . . . .</b>	<b>7</b>
I.1.1	Transport solide par charriage . . . . .	7
I.1.2	Phénomènes et enjeux liés au tri granulométrique . . . . .	8
I.1.3	Ségrégation par taille . . . . .	10
I.1.4	Approche à l'échelle de la particule . . . . .	10
<b>I.2</b>	<b>Approche expérimentale menée à Irstea . . . . .</b>	<b>10</b>
I.2.1	Objectifs scientifiques . . . . .	11
I.2.2	Dispositif expérimental . . . . .	12
I.2.3	Algorithmes de traitement d'images . . . . .	18
I.2.4	Limitations actuelles . . . . .	18
<b>I.3</b>	<b>Objectifs de la thèse . . . . .</b>	<b>19</b>
<b>I.4</b>	<b>Conclusion . . . . .</b>	<b>19</b>

---

Dans ce chapitre, nous présentons le contexte général, expérimental et scientifique de la thèse. Nous définissons les spécificités du transport solide par charriage en torrent à forte pente ainsi que les enjeux liés à la ségrégation des sédiments qui se produit lors de ce phénomène. Ensuite, nous décrivons l'approche expérimentale qui est menée à Irstea avec ses objectifs scientifiques et la mise en place d'un canal expérimental au centre de Grenoble pour l'étude de l'interaction entre les particules et le fluide. Enfin, nous donnons les objectifs de la thèse.

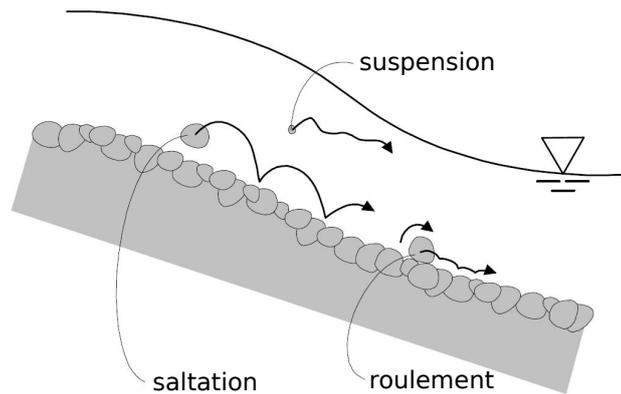
### I.1 Ségrégation en charriage : un phénomène granulaire

La ségrégation, ou tri granulométrique, en transport solide par charriage est un phénomène granulaire qui a un impact certain sur la morphologie des torrents et présente donc des risques pour l'environnement et les personnes. Nous définissons dans cette partie chacun des termes techniques liés au phénomène.

#### I.1.1 Transport solide par charriage

Dans la nature, en rivière et surtout en torrent, des éléments appartenant au milieu environnant (lits, berges, versants) tels que les sédiments majoritairement, peuvent être en-

traînés par l'écoulement d'eau. Ce transport de sédiments dans les cours d'eau, aussi appelé transport solide ou transit sédimentaire, a de fortes implications dans les domaines de la lutte contre les inondations, la gestion des ressources en eau et plus généralement du développement durable. Le transport solide se présente sous deux formes distinctes : la suspension qui concerne le transport des particules légères dans toute la colonne d'eau et le charriage constitué des sédiments plus grossiers qui restent en contact avec le lit (voir [Figure I.1](#)). Du fait de multiples forces hydrodynamiques exercées par l'écoulement d'eau, les sédiments au sein de la couche du charriage se déplacent selon différents modes de mouvement, comme le roulement, le glissement ou bien la saltation (rebonds successifs sur le dessus du lit).

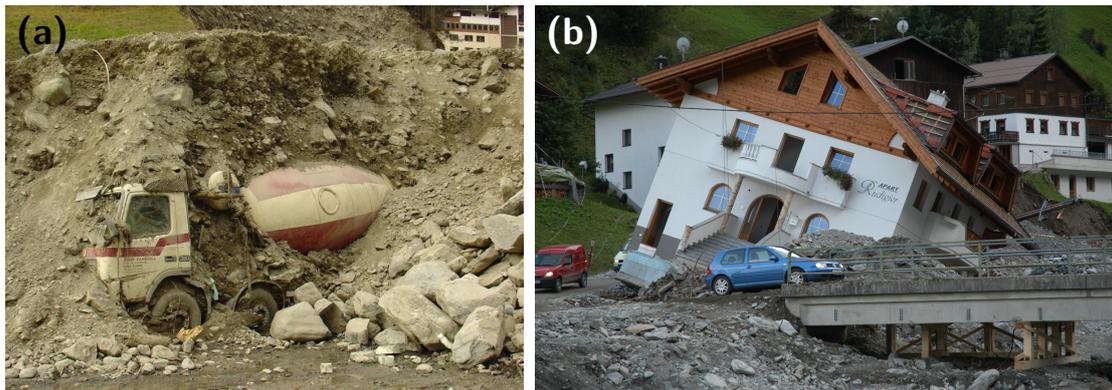


**Figure I.1** Schéma des différents modes de transport de sédiments. Le roulement et la saltation composent le charriage, et la suspension le mode éponyme. Schéma modifié de [Böhm \[2005\]](#).

On s'intéresse plus particulièrement au transport solide par charriage car il est responsable de la formation de structures morphologiques complexes dans les torrents de montagne à forte pente. En effet, ce processus provoquant érosion et dépôts a un rôle critique pour la stabilité et la morphologie des cours d'eau. Une modification profonde de la morphologie des torrents (exhaussements, incisions, divagations) par des flux intenses de charriage peut causer et aggraver des inondations dommageables pour les personnes, les biens et les infrastructures (voir des exemples de dégâts occasionnés en [Figure I.2a](#) et [Figure I.2b](#)), et a des conséquences directes sur l'environnement (qualité des eaux, vie aquatique, paysage, etc.). Il apparaît donc nécessaire de pouvoir prévenir et gérer ce type d'événements et les risques qui y sont liés. Pour cela, il est important de savoir comment les divers paramètres (granulométrie, pente, débit liquide, etc.) influent sur le transport de sédiments. C'est pour étudier les processus les plus complexes du charriage, notamment au niveau particulaire, que l'on effectue à Irstea des études expérimentales comme par exemples [Hergault et al. \[2010\]](#) et [Dudill et al. \[2016a\]](#), ou bien numériques comme celle de [Maurin et al. \[2015, 2016\]](#).

### I.1.2 Phénomènes et enjeux liés au tri granulométrique

Dans un écoulement torrentiel, il y a un mélange de sédiments de différentes tailles, caractérisé par une distribution appelée granulométrie. Plus la granulométrie est étendue, plus le phénomène de tri granulométrique est important. Ce tri ne dépend pas seulement du comportement de chacune des tailles de particules, mais aussi de l'interaction entre particules de tailles diverses. Le tri granulométrique peut s'effectuer dans la direction du courant,



**Figure I.2** Photos de dommages à l'issue d'un épisode de charriage. (a) Photo d'un dépôt solide conséquent illustrant les ravages provoqués par un exhaussement du lit suite à un flux intense de charriage (Sesslabach, Tirol, Août 2005). (b) Photo de l'effet désastreux d'un affouillement engendré par un manque d'apport solide pendant un épisode de crue en torrent (Paznauntal, Tirol, Autriche, Août 2005). Photos Chiari.

latéralement et verticalement, résultant en des morphologies caractéristiques [Powell, 1998].

Un des phénomènes de tri granulométrique les plus observés dans la nature est la stratification verticale du lit, encore appelé pavage (voir Figure I.3). Le tri est responsable de l'amincissement granulométrique de l'amont vers l'aval ainsi que de la transition brutale d'un lit de gravier vers un lit de sable (ce dernier phénomène n'étant pas encore complètement expliqué). Le tri granulométrique a également un impact certain sur la vie aquatique lorsqu'il provoque une arrivée massive de particules fines qui colmatent le lit (modification de l'habitat aquatique, difficulté de reproduction chez les salmonidés).



**Figure I.3** Photo d'un pavage (ou stratification verticale) par tri granulométrique dans un banc de graviers en rivière (Vedder River, British Columbia, Canada). Photo Frey.

Le tri granulométrique est un phénomène important pour l'ingénierie des rivières et torrents, et pour l'entretien et la restauration de leur écosystème. C'est donc pour toutes ces raisons qu'il est essentiel de bien l'analyser pour mieux le comprendre.

### I.1.3 Ségrégation par taille

Terme plus général que “tri granulométrique” et plus souvent utilisé en physique des milieux granulaires, la ségrégation fait référence à tous les phénomènes de séparation dus à la taille, la masse volumique, la forme ou encore la rugosité. On s’intéresse plus particulièrement à la ségrégation par taille car c’est le phénomène le plus important lié au charriage et également un des plus difficiles à modéliser.

Il existe deux types de ségrégation liés à la taille lors du charriage : la percolation spontanée et le tamisage cinétique. La percolation spontanée ou infiltration (*spontaneous percolation* en anglais) intervient lorsque des particules de petite taille s’infiltrent par le seul effet de la gravité dans un lit immobile de particules plus grosses. Des recherches sur la percolation spontanée en rivière et torrent ont déjà été largement menées en laboratoire [Einstein, 1968; Dermis and Papanicolaou, 2014], sur le terrain [Frostick et al., 1984; Lisle, 1989] et par la modélisation [Cui et al., 2008; Wooster et al., 2008]. Le tamisage cinétique (*kinetic sieving* en anglais) intervient quand un mélange de particules de différentes tailles est en mouvement et que les petites particules s’infiltrent dans les espaces créés par le mouvement des particules plus grosses. Globalement, il est observé un flux vers le bas des particules les plus petites et un flux vers le haut des particules les plus grosses, d’où la ségrégation observée dans les dépôts de rivières. Avec seulement quelques recherches sur le tamisage cinétique [Savage and Lun, 1988; Bacchi et al., 2014], il n’existe pas encore une description complète du processus dans le contexte fluvial.

### I.1.4 Approche à l’échelle de la particule

Le charriage, phénomène biphasique, a classiquement été étudié du point de vue du fluide. On cherche à connaître globalement quel débit solide peut être transporté par un débit liquide dans un tronçon de cours d’eau [Recking et al., 2013]. Il a été identifié que des progrès significatifs dans la compréhension des écoulements par charriage et leur modélisation passent par une meilleure prise en compte des interactions granulaires [Frey and Church, 2009, 2011]. Ceci nécessite de travailler à l’échelle de la particule et permet de tenir compte de tout le corpus de connaissances en physique des milieux granulaires. Ce qui est d’autant plus vrai que l’on s’intéresse à la ségrégation. Pour pouvoir travailler à l’échelle de la particule, il est nécessaire de détecter l’ensemble des particules en mouvement et à l’arrêt, et ensuite de calculer leur trajectoire. Pour ce faire, un canal expérimental étroit est utilisé (décrit par ailleurs en Section I.2.2) et un traitement d’images spécifique est appliqué sur les images obtenues (voir Chapitres IV et V).

## I.2 Approche expérimentale menée à Irstea

L’étude de la ségrégation en transport de sédiments par charriage est abordée de manière expérimentale à Irstea, qui a mis en place un canal spécialement conçu à cet effet. Dans cette section, nous définissons le contexte de travail dans lequel a été réalisé ce travail de thèse, les objectifs scientifiques qui sont visés, le dispositif expérimental utilisé avec ses analyses par traitement d’images et les limitations des systèmes actuels.

### **I.2.1 Objectifs scientifiques**

Les objectifs scientifiques globaux de cette thèse se placent au centre d'un organisme de recherche vaste et pluridisciplinaire, Irstea, qui organise ses recherches par centres, départements et unités.

#### **Irstea**

L'Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (Irstea) est un centre de recherche qui mène des études environnementales à l'échelle du territoire. Ses recherches ont pour but de relever les grands défis posés par le changement global, en particulier depuis le protocole de Kyoto, qui a été le commencement de la mise en avant de l'environnement comme préoccupation mondiale. Pour cela, il s'intéresse à la fois au contrôle des risques environnementaux pour l'Homme (crues, inondations, avalanches, feux de forêts, etc.) et à celui des risques humains pour l'environnement (technologies propres, aménagements du territoire, surveillance des milieux aquatiques, etc.). Afin de mieux atteindre ses objectifs, tant au niveau recherche que des applications, Irstea travaille en collaboration avec d'autres centres de recherche, des entreprises, et aide aux expertises pour les décisions de politiques publiques.

#### **Centre de Grenoble**

Le centre Irstea de Grenoble se tourne vers les enjeux liés à la montagne avec trois unités de recherches : Écosystèmes montagnards (EM), Développement des territoires montagnards (DTM), et Érosion torrentielle neige et avalanches (ETNA). Le but de ces équipes est de mener des recherches et de trouver des solutions pratiques aux difficultés et risques liés à un environnement montagneux et à son aménagement par l'Homme. Le centre de Grenoble est intégré dans le milieu scientifique et universitaire de Grenoble par de nombreuses collaborations, notamment avec l'Université Grenoble-Alpes (UGA) et l'Institut Polytechnique de Grenoble (Grenoble INP). Il est également membre de la COMUE Université Grenoble-Alpes.

#### **Unité ETNA**

L'unité de recherche ETNA s'intéresse à l'étude des risques naturels en montagne (avalanches, transport de neige par le vent, crues et laves torrentielles, chutes de blocs, risques d'origine glaciaire, etc.). Elle conduit des travaux fondamentaux dans les domaines de la formation, la rhéologie et la dynamique des écoulements. L'activité de modélisation physique et numérique y est soutenue, accompagnant des expérimentations de laboratoire et de terrain. Les autres domaines d'activité concernent l'étude des techniques de protection, la cartographie et le développement de systèmes d'information pour la gestion des risques naturels.

#### **Étude du charriage à forte pente**

Lors de fortes pluies, le charriage de sédiments devient un important facteur de risque naturel. Dans les torrents, cours d'eau à forte pente, l'avènement d'une crue torrentielle engendre une modification de la morphologie du lit, entraînant parfois un dépôt massif de sédiments sur les terres environnantes. La compréhension réelle du transport solide par charriage au-delà de la simple description pose de sérieuses difficultés. Pour progresser dans la compréhension de ces processus, un canal expérimental a été spécifiquement conçu pour

l'étude à l'échelle de la particule des interactions complexes entre particules en mouvement, lit mobile, et écoulement d'eau, et plus généralement du couplage entre phases solide et fluide. L'objectif est en particulier d'avancer dans la compréhension du tri granulométrique (ou ségrégation), véritable verrou scientifique en transport de sédiments.

### I.2.2 Dispositif expérimental

Le dispositif expérimental de Irstea Grenoble a été mis en place afin d'étudier le charriage et la ségrégation à forte pente. Deux configurations sont considérées dans cette étude : un canal très étroit et un canal un peu plus élargi. Le canal étroit permet d'accéder à des mesures très fines, nécessaires pour une analyse à l'échelle de la particule. Il a pour particularité d'être assez étroit pour pouvoir observer le mouvement bidimensionnel des particules dans un écoulement. En effet, la largeur du canal est du même ordre de grandeur que le diamètre des particules de façon à contraindre leur mouvement dans un plan vertical et ainsi accéder à toutes les trajectoires grâce à une caméra qui filme l'écoulement sur le côté du canal. Le canal élargi permet quant à lui d'étudier l'infiltration de particules fines dans un lit de particules grossières et d'observer leur influence sur l'évolution du lit mobile. Pour ce faire, différents rapports de taille entre les particules grossières et les fines sont utilisés.

Dans une première thèse, [Bigillon \[2001\]](#) s'est intéressée au mouvement d'une particule unique dans un écoulement [[Ancey et al., 2002, 2003](#)]. Ensuite, dans un deuxième travail de thèse, [Böhm \[2005\]](#) a étudié les interactions entre des particules sphériques de même taille (6 mm) et le lit mobile [[Böhm et al., 2004](#); [Ancey et al., 2005, 2006](#)]. Dans une troisième thèse, [Hergault \[2011\]](#) s'est penchée sur l'étude de la ségrégation et son influence sur le charriage en utilisant deux tailles de particules différentes (4 mm et 6 mm) dans le canal [[Hergault et al., 2010](#)]. Dans ces trois thèses, le canal expérimental était utilisé dans sa configuration étroite. Les séries d'images non publiées concernant le canal étroit qui seront exploitées dans cette thèse proviennent d'expérimentations de tamisage cinétique effectuées par [Rorsman \[2012\]](#) (stage de fin d'étude d'ingénieur encadré par P. Frey). Dans une autre thèse, [Dudill \[2016\]](#) s'est intéressée à l'étude de la percolation spontanée et du tamisage cinétique par des expériences menées sur le canal en configuration élargie [[Dudill et al., 2016a](#)] avec des méthodes d'analyse d'images adaptées [[Lafaye de Micheaux et al., 2015](#)].

Dans ce travail de thèse, le canal est utilisé dans ses deux configurations, à savoir étroite et élargie. Chaque configuration possède ses propres méthodes de traitement des données, le développement de ces méthodes étant le coeur de cette thèse. La suite de cette section donne les conditions expérimentales, ainsi que les caractéristiques des différents montages expérimentaux. Le canal dans sa forme étroite est décrit ci-après, les différences avec le canal élargi sont données dans le [Tableau I.1](#).

#### I.2.2.1 Le canal

Le dispositif expérimental est composé d'un canal inclinable d'une longueur de 2 m avec des parois en verre de 20 cm de hauteur. Dans nos expériences, la pente du canal est fixée à  $S = 10\%$ . La photo en [Figure I.4](#) et le schéma en [Figure I.5](#) permettent de comprendre l'agencement des différents outils expérimentaux du canal étroit. Les deux parois de verre sont fixées verticalement et maintenues parallèles grâce à une structure en aluminium. Le canal repose sur un banc en acier pour limiter les vibrations causées par l'écoulement et pour lui maintenir une bonne verticalité. Le fond du canal est constitué de demi-cylindres

Caractéristiques	Notations	Unités	Canal étroit	Canal élargi
Largeur canal	$W$	mm	6.5	10.3
Pente canal	$S$	%	10	10.1
∅ Billes grossières	$2r_1$	mm	6	5
∅ Billes fines	$2r_2$	mm	4	0.7 à 4
Couleur billes grossières	-	-	noir	transparent
Couleur billes fines	-	-	transparent	transparent*
Hauteur butée	$H_b$	mm	40	64
Taille du pixel	$\Delta_X$	mm	0.2056	0.2351
Caméra rapide	-	-	Baumer	Photon Focus
Dimension image	-	pixel	1280 × 320	1024 × 500
Fréquence image	$FPS$	Hz ou fps	130	130
Intervalle inter-image	$\Delta_T$	sec	0.0077	0.0077

**Tableau I.1** Caractéristiques des deux configurations, étroite et élargie, du canal expérimental.

\* Les billes fines du canal élargi sont transparentes mais apparaissent grises dans l'image.

en acier inoxydable du même diamètre que les plus grosses particules (6 mm). Ils sont disposés de manière aléatoire sur différents niveaux allant de 0 à 5.5 mm par incrément de 0.5 mm. L'étanchéité du canal est assurée par des joints placés sous le fond en acier et au niveau des jonctions entre le fond et les parois. La largeur du canal  $W$ , modifiable de 3 à 25 mm, est fixée à  $W = 6.5$  mm, ce qui est légèrement plus grand que le diamètre des plus grandes particules (6 mm). Dans cette configuration, le mouvement des particules est quasi bidimensionnel (peu voire pas de déplacement dans la largeur), ce qui est indispensable pour la détection et le suivi de toutes les particules avec une seule caméra. Pour former un lit de particules et empêcher l'érosion totale, une butée de  $H_b = 40$  mm est disposée à la sortie du canal. La procédure utilisée pour la construction du lit est donnée en Section I.2.2.4.

Malgré certains désavantages comme les effets de parois (frottement), ce canal permet une représentation très simple du charriage avec des conditions aux limites bien définies et des possibilités de mesures très fines et exhaustives par analyse d'images. En particulier, toutes les billes de verre peuvent être détectées à haute fréquence temporelle (130 Hz pour les deux configurations).

### I.2.2.2 Flux liquides et solides

L'écoulement dans le canal se présente sous la forme d'un mélange bimodal de billes de verre sphériques, composant la phase solide, entraîné par une phase liquide, soit ici de l'eau. L'approvisionnement de ces deux parties est décrit dans cette section.

#### I.2.2.2.1 Phase liquide

Le liquide utilisé dans l'écoulement est de l'eau. L'eau s'écoule dans un circuit fermé. Elle est aspirée par une pompe centrifuge placée dans un réservoir à la sortie du canal. Un autre réservoir en hauteur muni d'un trop-plein assure une charge constante. De ce réservoir, l'eau est acheminée à l'entrée du canal. Le flux transite préalablement par un débitmètre électromagnétique et une vanne de réglage avant d'être injecté dans le canal. Le débit d'eau est réglable de 0.025 à 0.14 l/s (soit un débit par unité de largeur de 3.8 à



**Figure I.4** Photo du dispositif expérimental à Irstea (Grenoble, France). Écoulement de l'arrière-plan vers le premier plan. Photo Hergault.

$21.5 \times 10^{-3} \text{ m}^2/\text{s}$ ). Il est maintenu constant pour une étude du transport de particules sous des conditions d'équilibre. Pour nos fortes pentes, le régime hydraulique de l'écoulement est dit supercritique, ce qui signifie que le comportement amont de l'écoulement n'est pas influencé par l'aval. Un filtre maillé est utilisé à l'entrée du canal pour retenir les impuretés et tranquilliser l'écoulement, et un tamis est placé à la sortie pour séparer l'eau des billes.

### I.2.2.2 Phase solide

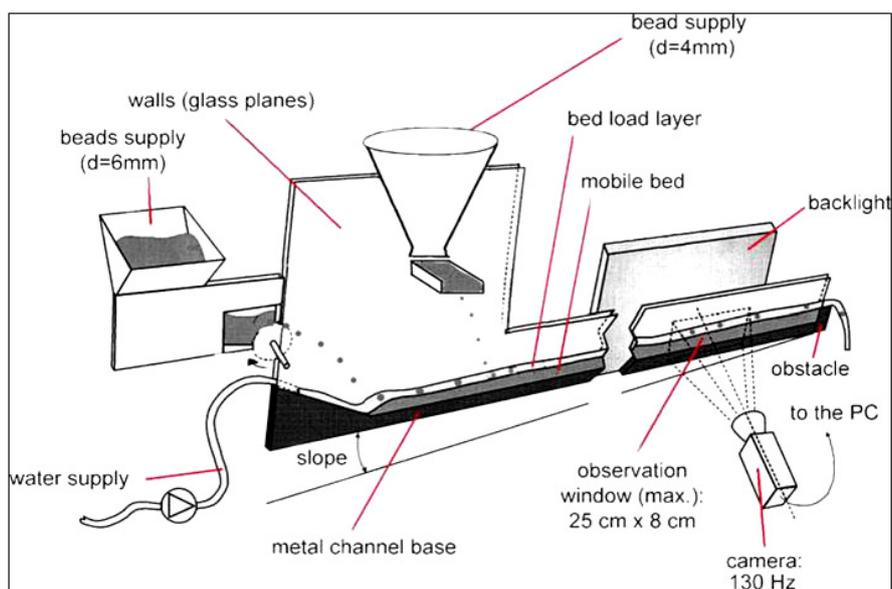
La phase solide dans la configuration étroite du canal est constituée de deux types de billes de verre sphériques : des billes "grossières" noires de diamètre 6 mm et des billes "fines" transparentes de 4 mm. La masse volumique de toutes les billes considérées dans cette thèse est de  $2500 \text{ kg}/\text{m}^3$ .

#### Billes grossières

Les billes de verre sphériques grossières et noires, fournies par Sigmund Lindner GmbH, Allemagne, ont un diamètre de  $2r_1 = 6 \text{ mm}$  avec une tolérance de  $\pm 0.3 \text{ mm}$ . Stockées dans un réservoir placé au tout début du canal, les billes noires sont injectées dans le canal au moyen d'une roue dentée motorisée. Le débit d'injection peut varier d'environ 6 à 20 billes par seconde (soit un débit par unité de largeur de  $8.7$  à  $36.5 \times 10^{-5} \text{ m}^2/\text{s}$ ) avec une incertitude de  $\pm 5 \%$ .

#### Billes fines

Les billes de verre sphériques fines et transparentes, fournies par Cimap, France, ont un diamètre de  $2r_2 = 4 \text{ mm}$  avec une tolérance de  $\pm 0.4 \text{ mm}$ . Si on avait choisi un diamètre de



**Figure I.5** Schéma du dispositif expérimental à Irstea (Grenoble, France). L'eau coule de gauche à droite entraînant un mélange de billes de verre sphériques de deux tailles différentes. Les billes noires de diamètre 6 mm sont introduites en amont et les billes transparentes de diamètre 4 mm plus en aval. Le lit mobile est observé à travers une fenêtre rétro-éclairée et filmé par une caméra rapide à 130 Hz. Schéma modifié de Böhm et al. [2004] et Hergault et al. [2010].

3 mm, une bille aurait pu être complètement cachée derrière une autre bille. Le rapport des tailles est important en ce qui concerne le tri granulométrique. Stockées dans un réservoir placé quelques dizaines de centimètres plus en aval que le réservoir de billes noires, les billes transparentes sont injectées dans le canal au moyen d'un doseur pondéral BSP (Bulk Solids Pump) (fourni par la société K-TRON) présentant un disque rotatif vertical. Le débit d'injection peut varier de 6 à 37 billes par seconde avec une très bonne précision.

### Récupération et tamisage

Les billes noires et transparentes sortant du canal sont récupérées puis séchées dans une étuve (ce qui est nécessaire car les distributeurs n'admettent pas de billes humides). Elles sont ensuite tamisées pour être séparées par taille.

#### I.2.2.3 Système d'acquisition d'images

Pendant chaque expérience, l'écoulement est filmé à l'aide d'une caméra rapide à une fréquence de  $FPS = 130$  Hz (caméra de marque Baumer avec un objectif Myutron de focale 25 mm et d'ouverture  $f/1.4$ ). Cette caméra est placée sur un trépied fixé au sol à 115 cm du canal et à environ 80 cm à l'amont de la sortie du canal. Afin d'obtenir des images horizontales sans voir la pente du canal, la caméra est placée perpendiculairement aux plaques de verre et est inclinée du même angle que le canal, soit une pente de  $S = 10\%$ . Le placement et l'inclinaison de la caméra sont vérifiés à l'aide d'un procédé à base de mires. La zone filmée mesure 25 cm de long pour 8 cm de haut. Un dispositif de rétroéclairage composé de LED blanches fortement lumineuses (fourni par Phlox, France) est placé derrière cette zone, et assure un éclairage stable et uniforme. Il permet de bien faire ressortir le contour des

billes transparentes de 4 mm, ce qui est indispensable pour les détecter. Le facteur d'échelle des images est calculé avec une mire. Il correspond en réalité à la taille d'un pixel  $\Delta_X$  et change peu d'une expérience à l'autre (autour de  $\Delta_X = 0.2056$  mm). Cette valeur d'échelle est fixe pour toutes nos expériences, la distance entre le canal et la caméra ne changeant pas. La dimension des images est de  $1280 \times 320$  pixels.

#### I.2.2.4 Procédure expérimentale

La procédure expérimentale consiste à former un lit mobile. Un lit mobile signifie que les billes qui le constituent sont mobiles et peuvent être entraînées par l'écoulement. Cette mobilité permet des échanges entre les particules solides en mouvement et le lit (érosion et dépôt). Avant toute expérience, le canal est vide avec seulement la base en acier constituée de demi-cylindres de 6 mm placés aléatoirement. Cette base est la même dans toutes nos expériences.

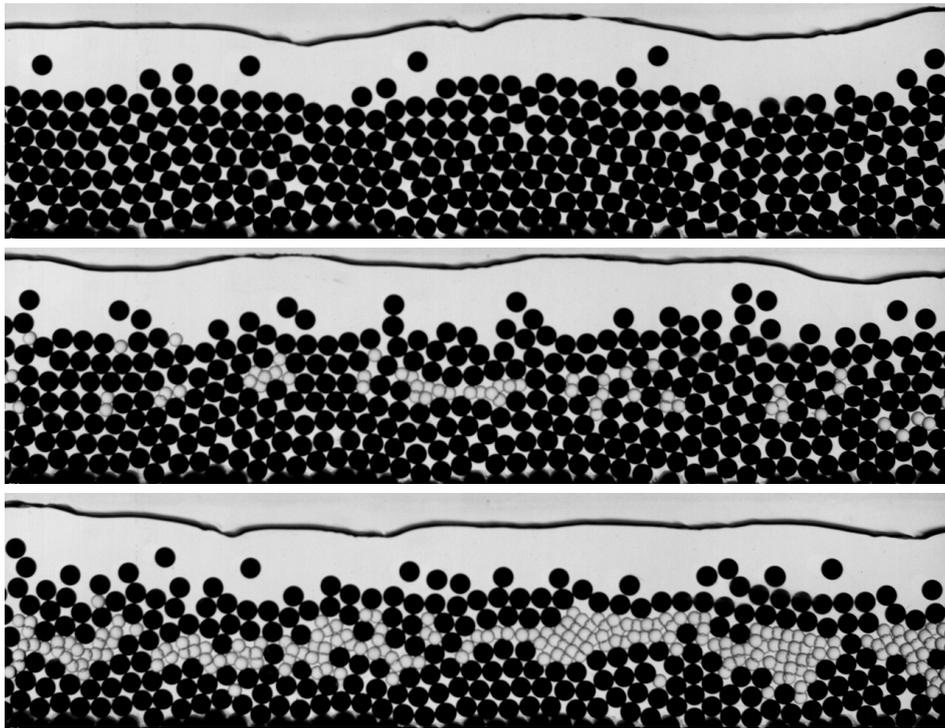
Tout d'abord, il faut créer le lit avec des billes noires. Un débit d'eau est fixé à une valeur constante et une butée est placée à la sortie du canal. L'alimentation en billes noires est alors lancée. Les premières billes sont arrêtées par la butée et commencent à former le lit qui se construit progressivement. Lorsque la ligne du lit atteint le niveau de la butée  $H_b$ , les billes commencent à quitter le canal. L'équilibre est atteint lorsque le débit sortant des billes noires est identique au débit entrant (équilibre atteint en environ 10 minutes). Le débit liquide est ajusté pour avoir une ligne du lit et une ligne d'eau parallèle à la base du canal, soit une pente de lit et une pente d'eau correspondant à l'inclinaison du canal.

Par la suite, une fois cet équilibre atteint, l'approvisionnement à débit constant en billes transparentes commence. Les premières billes rebondissent puis se déplacent sur le lit de billes noires. Ensuite, par tamisage cinétique (voir Section I.1.3), certaines billes transparentes commencent à s'infiltrer dans le lit mobile formant ainsi un mélange bimodal de billes noires et transparentes. Il y a alors trois différents types de trajectoires possibles pour les billes : saltation (bille qui rebondit sur les autres), roulement (bille qui glisse/roule sur les autres), et arrêt (bille qui ne bouge quasiment pas). Un nouvel équilibre est alors atteint lorsque le débit sortant des billes transparentes est identique au débit entrant (équilibre atteint en environ 50 minutes). Pour les expériences menées sur le canal élargi, cet équilibre bimodal est caractérisé par une diminution (*degradation* en anglais) ou augmentation (*aggradation* en anglais) de la pente du lit et de la surface libre (les deux étant toujours parallèles en moyenne), et cette pente n'est donc plus égale à l'inclinaison du canal.

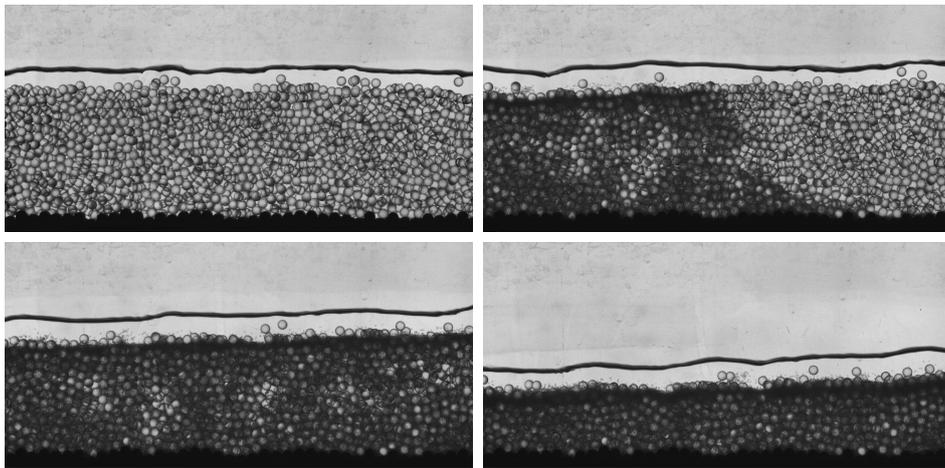
Tout ce processus de ségrégation des billes transparentes est filmé à une fréquence  $FPS = 130$  Hz dans son intégralité, du début de l'approvisionnement en billes transparentes jusqu'à l'obtention de l'équilibre du mélange bimodal. La Figure I.6 montre une partie d'une séquence d'images typique obtenue à l'issue de cette procédure expérimentale pour le canal étroit [Rorsman, 2012], et la Figure I.7 pour le canal élargi.

#### I.2.2.5 Grandeurs caractéristiques

Pour chaque expérimentation et en fonction des conditions d'acquisition, des grandeurs caractéristiques servent à définir certains paramètres des algorithmes de traitement des trajectoires. Notamment, la fréquence d'images  $FPS$  est choisie telle que le déplacement d'une bille d'une image à l'autre ne puisse jamais être supérieur au diamètre  $2r_1$  d'une bille noire, l'écart temporel entre deux images étant  $\Delta_T = 1/FPS$ . Cette condition est essentielle



**Figure I.6** Images de transport par charriage d'un mélange bimodal de particules en canal étroit (billes noires de 6 mm et billes transparentes de 4 mm). Écoulement de gauche à droite. La première image (en haut) correspond à l'équilibre du mélange unimodal. Les billes transparentes s'infiltrent dans le lit (image du milieu à 16 minutes) pour former une couche quasi-continue en sub-surface (image du bas à 48 minutes).



**Figure I.7** Images de transport par charriage d'un mélange bimodal de particules en canal élargi (billes transparentes de 5 mm et billes fines de 0.7 mm). Écoulement de gauche à droite. Dans l'ordre temporel, les images apparaissent respectivement (du haut gauche au bas droit) à : 0 min, 5 min, 7 min et 20 min. L'expérience montre ici une incision nette du lit mobile. La première et la dernière image correspondent respectivement à l'équilibre unimodal et bimodal.

non seulement pour avoir une haute résolution des trajectoires mais aussi principalement pour être capable de distinguer les billes et les suivre au cours du temps. La vitesse maximale  $u_{max}$  qu'une bille peut atteindre lorsqu'elle est emportée par le fluide est alors définie par  $u_{max} = 2r_1FPS$ .

Cette vitesse  $u_{max}$  est ensuite utilisée pour définir les vitesses seuils qui permettent de déterminer les états de mouvement des billes. Nous nous intéressons principalement aux états de mouvement liés au transport solide par charriage (voir Section I.1.1), à savoir le roulement, la saltation et aussi l'arrêt. Ainsi, les billes sont considérées à l'arrêt lorsque leur vitesse est en-dessous d'une vitesse seuil  $u_{rest}$  fixée à 1% de  $u_{max}$ , et en saltation au-dessus d'une vitesse seuil  $u_{salt}$  fixée à 40% de  $u_{max}$ . Ces valeurs de proportion ont été ajustées expérimentalement pour minimiser les différences entre les états déterminés par l'algorithme d'estimation des états (voir Section IV.1.3) et les états déterminés à l'oeil nu.

### I.2.3 Algorithmes de traitement d'images

Les images obtenues par l'expérience de charriage décrite ci-avant présentent toutes les informations nécessaires pour étudier le phénomène de ségrégation. La manière la plus simple, rapide et efficace d'extraire ces informations est d'utiliser des méthodes de traitement d'images. Pour rendre l'extraction automatique et avec l'utilisation de méthodes de morphologies mathématiques, des algorithmes ont été développés au sein d'Irstea depuis la création du canal expérimental.

Ces algorithmes cherchent à récupérer les trajectoires des billes afin d'en étudier le déplacement et l'évolution au cours du temps. Ils se basent sur une approche en deux étapes : détection et suivi. La première étape consiste à détecter les billes dans chacune des images de la séquence avec l'aide des caractéristiques des différents types de billes (couleur, taille, forme). La deuxième étape cherche à suivre les billes détectées d'une image à l'autre par association. Une fois la séquence traitée, pour chaque bille, la trajectoire est alors obtenue, avec la position dans les images et la vitesse au cours du temps.

### I.2.4 Limitations actuelles

Dans les précédents travaux sur le canal étroit, chaque séquence d'images est limitée à 8000 images pour des raisons de mémoire vive lors de l'acquisition (acquisition en RAM et sauvegarde sur disque dur différée). De plus, pendant le traitement des images, la structure utilisée pour stocker les informations de détection et de suivi des billes ne permet pas de traiter des séquences de plus de 8000 images. Or, une séquence de 8000 images ne correspond qu'à une minute d'enregistrement et donc qu'à une petite séquence du phénomène de ségrégation. L'étude de la ségrégation en charriage sur une durée longue est un élément clé qui a motivé cette nouvelle thèse.

D'autre part, les algorithmes de suivi de billes développés dans les travaux précédents présentent des limitations en terme de précision et de robustesse, ou plus précisément, ils manquent de cohérence dans les trajectoires de billes. En effet, ils se basent sur les billes détectées lors de l'étape de détection. Or, la détection des billes n'est pas toujours fiable, elle peut manquer de précision et il peut y avoir des détections manquantes et des fausses détections. Le suivi des détections peut alors être faussé et ainsi créer des trajectoires fractionnées, mélangées, perdues, imprécises. Avoir un algorithme de suivi de billes plus robuste est aussi un des éléments qui a motivé cette thèse.

Ces algorithmes de suivi sont adaptés à l'étude exhaustive des trajectoires des billes dans le canal étroit. Cependant, dans la configuration élargie du canal, cette étude exhaustive est impossible avec une seule caméra à cause de la petitesse des particules fines et des occlusions multiples occasionnées. Les algorithmes développés précédemment sont donc inappropriés pour le canal élargi. Pouvoir analyser des images issues du canal élargi est également un aspect qui a justifié ce travail de thèse.

### I.3 Objectifs de la thèse

L'objectif global des recherches menées dans cette thèse est de développer des méthodes de traitement d'images afin d'étudier, par une approche expérimentale, le phénomène de ségrégation en transport de sédiments par charriage. Déjà étudié par le passé à Irstea, le système permettant l'analyse de ce phénomène se base sur des algorithmes de traitement d'images applicables aux données expérimentales. Face aux limitations de ces algorithmes, les travaux de cette thèse sont abordés en trois axes.

Premièrement, les algorithmes précédemment développés ne sont pas adaptés à l'étude des images issues du canal élargi. Les séquences d'images qui y sont obtenues illustrent pourtant un phénomène de ségrégation. L'objectif ici est donc de développer des méthodes de traitement d'images, et notamment de segmentation, pour extraire des lignes caractéristiques (ligne d'eau, ligne du lit) dans les images. Ces lignes permettent ensuite d'analyser la hauteur d'eau et de la pente du lit, ce qui rend alors possible l'étude de l'influence de l'infiltration de particules fines sur l'évolution d'un lit mobile.

Deuxièmement, les algorithmes de suivi de billes dit "déterministes" élaborés lors des travaux antérieurs ne permettent pas d'étudier les trajectoires sur de longues séquences d'images. Or, le phénomène de ségrégation en charriage peut être de longue durée, comme le montrent nos expérimentations. L'objectif ici est alors de raffiner et optimiser ces algorithmes de suivi déterministes afin de permettre l'analyse des trajectoires complètes sur l'intégralité du phénomène de ségrégation.

Troisièmement, avec l'approche déterministe de ces algorithmes, nous avons constaté des problèmes lors de la détection des billes (détections manquantes et fausses détections), ce qui engendre ensuite des erreurs dans le suivi des trajectoires. Le régime de l'écoulement étant transitoire, il est important de pouvoir avoir des résultats fiables sur toute la durée de la séquence, en particulier au début lorsque le phénomène transitoire est fort. L'objectif ici est donc de développer un nouvel algorithme de traitement d'images basé sur un filtrage particulière à modèles multiples pour gagner en robustesse et en précision, et aussi pour mieux gérer les dynamiques complexes des billes.

Ces trois aspects sont abordés et détaillés respectivement dans les Chapitres [III](#), [IV](#) et [V](#) de cette thèse.

### I.4 Conclusion

Dans ce premier chapitre, nous avons dans un premier temps fourni les éléments fondamentaux sur lesquels se fonde cette thèse. Nous avons décrit la phénoménologie du transport sédimentaire. Nous avons ensuite mis en évidence l'importance du tri granulométrique, ou ségrégation par taille, un phénomène qui est encore mal compris dans le domaine du charriage. Puis, nous avons montré en quoi aborder l'étude de la ségrégation par une approche à

l'échelle de la particule était indispensable pour observer les interactions entre les particules.

Dans un second temps, nous avons décrit l'approche expérimentale menée à Irstea avec ses objectifs et le dispositif expérimental permettant d'étudier la dynamique du charriage à l'échelle de la particule. Ce dispositif, déjà exploité dans de précédents travaux de recherche, est un canal incliné alimenté en eau et en particules dont nous avons donné toutes les caractéristiques. L'écoulement produit dans ce canal a pour but d'idéaliser le transport de sédiment dans les rivières à forte pente (ou torrents) à lit de graviers. Nous avons précisé toutes les particularités de notre canal, à savoir : sa faible largeur pour pouvoir observer l'intégralité du mouvement particules, l'utilisation de billes sphériques bimodales en guise de particules et son système d'acquisition d'images bien adapté. Nous avons également donné le protocole expérimental complet pour produire la ségrégation en charriage recherchée et les algorithmes utilisés pour traiter les données obtenues.

Dans un dernier temps, après avoir énoncé les différentes limitations rencontrées durant les précédents travaux réalisés sur le canal, nous avons donné les trois objectifs principaux qui ont motivé ce travail de thèse, à savoir :

1. Développement d'un algorithme de segmentation pour l'étude de l'infiltration et de la pente en canal large,
2. Optimisation du suivi déterministe pour des séquences de longue durée,
3. Développement d'un suivi probabiliste à base de filtrage particulaire.

## Chapitre II

# Revue des méthodes de suivi d'objets

### Sommaire

---

<b>II.1 Suivi de particules en mécanique des fluides . . . . .</b>	<b>21</b>
II.1.1 Véllocimétrie par images de particules (PIV) . . . . .	22
II.1.2 Véllocimétrie par suivi de particules (PTV) . . . . .	23
II.1.3 Approches utilisées en suivi de sédiments . . . . .	23
<b>II.2 Suivi d'objets en vision par ordinateur . . . . .</b>	<b>24</b>
II.2.1 Méthodes déterministes . . . . .	25
II.2.2 Méthodes probabilistes . . . . .	29
II.2.3 Métriques d'évaluation . . . . .	32
<b>II.3 Synthèse et rapprochement à notre problématique . . . . .</b>	<b>35</b>
<b>II.4 Conclusion . . . . .</b>	<b>36</b>

---

Le suivi d'objets est un problème récurrent en vision par ordinateur, il est utilisé dans de nombreuses applications, par exemples la vidéo surveillance, l'analyse sportive ou encore la sécurité du trafic routier. En particulier, le suivi d'objets est utilisé en mécanique des fluides pour l'analyse de mouvement de particules ou encore de sédiments.

Dans ce chapitre, nous présentons l'état de l'art concernant le suivi d'objets. Dans un premier temps, nous rattachons le suivi d'objets au domaine de la mécanique des fluides avec notamment les méthodes de suivi de particules telles que la PIV et la PTV, et les approches classiquement utilisées en suivi de sédiments. Dans un second temps, nous faisons état du suivi d'objets en vision par ordinateur en donnant quelques unes des méthodes les plus utilisées et comment les évaluer.

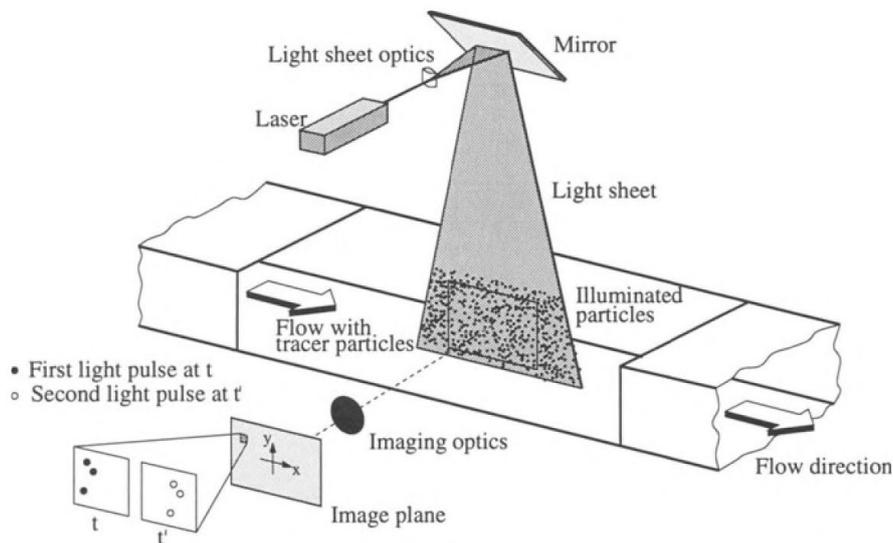
## II.1 Suivi de particules en mécanique des fluides

Il existe plusieurs techniques permettant d'étudier le mouvement de particules dans un fluide. Dans cette section, nous introduisons une des méthodes les plus anciennes et répandues, la PIV. Ensuite, nous voyons la PTV, un dérivé de la PIV pour le suivi individuel des particules. Enfin, nous présentons les approches de suivi de sédiments qui sont les plus classiquement utilisées.

### II.1.1 Vélométrie par images de particules (PIV)

La vélocimétrie par images de particules (PIV pour *particle image velocimetry*), reconnue sous ce terme en 1984 par [Pickering and Halliwell \[1984\]](#) et [Adrian \[1984\]](#), est une méthode d'imagerie optique pour la visualisation de fluides. Elle permet d'obtenir des mesures de vitesses instantanées dans un écoulement et de faire ressortir certaines propriétés du fluide. Elle est une des méthodes les plus importantes dans la recherche en dynamique des fluides pour déterminer le champ de vitesse d'un fluide. La PIV est maintenant bien maîtrisée, elle a notamment été révisée plusieurs fois dans la littérature [[Adrian, 1991, 2005](#); [Wereley and Meinhart, 2010](#)] et est aussi le sujet de plusieurs livres [[Adrian and Westerweel, 2011](#); [Raffel et al., 2013](#)]. Elle est utilisée dans une large gamme de problèmes liés aux fluides, comme l'étude de la turbulence [[Westerweel et al., 2013](#)], l'aérodynamique [[Taylor et al., 2010](#)] ou encore la recherche biomédicale [[Lima et al., 2008](#)].

Pour appliquer un traitement par PIV, un montage et protocole particuliers sont nécessaires (voir une installation typique en [Figure II.1](#)). Le principe se base sur l'introduction de petites particules, aussi appelées traceurs, dans le fluide. Ces particules sont suffisamment petites pour pouvoir suivre assez fidèlement la dynamique du fluide. Pour avoir une bonne résolution de mesure, une forte concentration de traceurs est essentielle. Le fluide, qui entraîne ces traceurs, est illuminé par un plan laser (ou une nappe), ce qui permet de rendre les traceurs visibles sur une coupe de l'écoulement. Avec une caméra orientée à  $90^\circ$  du plan et synchronisée sur la fréquence du laser, il est alors possible d'enregistrer des images des traceurs à des instants rapprochés.



**Figure II.1** Schéma du dispositif expérimental pour faire de la vélocimétrie par images de particules (PIV) dans un écoulement. Schéma tiré de [Raffel et al. \[2013\]](#).

Le traitement PIV consiste à diviser les images en petites fenêtres de recherche. Ensuite, par des techniques de corrélation d'images appliquées sur une même fenêtre de recherche de deux images consécutives, il est possible d'obtenir la vitesse moyenne dans cette région. Le champ de vitesse sur toute l'image est obtenu en répétant ce processus sur toutes les régions de recherche. La PIV est donc une méthode continue basée sur une spécification Eulérienne du champ de l'écoulement.

Avec la PIV, la concentration en particules est telle qu'il est possible d'identifier indivi-

duellement chacune des particules d'une image, mais il est plus difficile de les suivre d'une image à l'autre et d'autant plus en présence de forts gradients de vitesse. Lorsque la concentration en particules est plus faible, il est possible de suivre les particules individuellement, on parle alors de vélocimétrie par suivi de particules (PTV).

### II.1.2 Vélocimétrie par suivi de particules (PTV)

La vélocimétrie par suivi de particules (PTV pour *particle tracking velocimetry*), aussi appelée PIV à image à faible densité [Adrian, 1991], est une méthode d'imagerie pour le suivi individuel des trajectoires de particules dans un fluide. Elle est applicable quand la concentration en particules est assez faible. Elle permet d'associer les particules d'une image à l'autre et donc d'en suivre leurs trajectoires sur des longues périodes, même lorsque les vitesses présentent de forts gradients. Tout comme la PIV, la PTV est une technique de mesure qui a été largement expérimentée dans des écoulements de fluide bi-dimensionnels [Ohmi and Li, 2000], tri-dimensionnels [Maas et al., 1993; Malik et al., 1993], ou bien détaillée dans un chapitre de livre [Dracos, 1996]. Contrairement à l'approche Eulérienne de la PIV, la PTV est une technique de mesure discrète basée sur une approche Lagrangienne. Dans la littérature, elle est parfois appelée suivi Lagrangien de particules (LPT pour *lagrangian particle tracking*) [Ouellette et al., 2006]. Il existe également des méthodes hybrides qui appliquent la PIV en prétraitement de la PTV pour estimer les vitesses locales [Cowen and Monismith, 1997].

L'identification et la détection de la position des particules dans l'espace est une étape primordiale en PTV. En effet, le suivi des particules n'est possible et efficace que si la détection des particules est fiable, c'est-à-dire lorsqu'elle présente un nombre de fausses détections (*i.e.* particule détectée alors qu'il n'y en a pas réellement à cette position) et de détections manquantes (*i.e.* particule non détectée alors qu'elle est bien réellement présente) le plus petit possible. Ensuite, le suivi des particules est assuré par une résolution du problème d'association de données, aussi appelé problème d'affectation. Le principe est d'associer chaque particule d'une image à une particule de l'image précédente pour ainsi créer des trajectoires (voir Section II.2.1.2 pour plus de détails).

La limitation de la PTV réside dans le manque de robustesse de l'algorithme pour suivre les particules sur de longues séquences avec une bonne précision. En effet, l'étape d'association de la PTV accorde une confiance totale en la sortie du détecteur, et peut donc rencontrer des difficultés en cas d'erreurs de détection (faux positifs et faux négatifs).

### II.1.3 Approches utilisées en suivi de sédiments

Comme énoncé dans la Section I.1, l'étude du déplacement des sédiments a déjà été abordé en laboratoire, en terrain et par la modélisation. Lorsque l'étude est faite sur des sédiments naturels (en terrain voire en laboratoire), il est très difficile de les suivre individuellement à cause de leur grande diversité de formes et tailles. Pour limiter ces situations complexes, des expériences simplifiées utilisent des matériaux idéalisés, en général des billes de verre ou de plastique, sur des canaux d'épaisseurs relativement faibles. Ensuite, les suivis de billes sont habituellement faits par le biais d'algorithmes de PTV.

Dans la littérature, le suivi de sédiments à l'échelle de la particule en canal expérimental est très peu étudié, et les méthodes utilisées sont assez rarement détaillées. Parmi les quelques recherches dans ce domaine, on retrouve notamment le suivi de billes de taille unique par

PTV en canal dit “étroit” (*i.e.* canal à peine plus large que le diamètre des billes) [Böhm et al., 2006; Tuyen and Cheng, 2012] et en canal dit large (canal beaucoup plus large que le diamètre des billes) [Houssais et al., 2015; Ni and Capart, 2015]. En canal étroit, on retrouve également le suivi de billes de deux tailles différentes [Hergault et al., 2010]. Le canal de ce dernier travail présente quasiment les mêmes caractéristiques qu’un canal utilisé dans ce travail de thèse. Celui-ci est décrit en Section I.2.2.

Les algorithmes de suivi traitant d’un seul type de billes sont adaptés à détecter et suivre uniquement ce type de billes. Bien que Hergault et al. [2010] aient adapté un algorithme de PTV pour deux types de billes, ils ne se consacrent qu’à l’étude de courtes séquences d’images et ne garantissent pas un suivi optimal en présence de possibles erreurs de détection. Pour faire face à ces difficultés et pour pallier le peu d’algorithmes existants dans le domaine du suivi de sédiments, nous nous sommes alors tournés vers un domaine plus large et plus complet : le suivi en vision par ordinateur.

## II.2 Suivi d’objets en vision par ordinateur

La vision par ordinateur fait partie intégrante de l’intelligence artificielle. Son but est de permettre à un ordinateur ou une machine de traiter automatiquement des images obtenues par un système d’acquisition (comme une caméra par exemple) pour en tirer des informations et propriétés. L’idée est d’essayer d’imiter l’interprétation de l’image qu’aurait pu faire la vision humaine. Une des branches importantes de la vision par ordinateur est le suivi d’objets (*object tracking*). Le suivi d’objets est l’analyse de séquences vidéos ayant pour objectif d’établir la position d’une cible sur la séquence pour en suivre son évolution. Une difficulté de cette analyse est de la rendre automatique. Pour cela, il faut pouvoir initialiser les cibles à leur arrivée dans l’image (champ de vision de la caméra), les terminer à leur sortie, les suivre durant leurs déplacements, gérer leurs arrêts et ne pas les confondre avec d’autres cibles proches ou similaires.

Le nombre de recherches sur le suivi d’objets a une étroite liaison avec l’évolution de la puissance de calcul des ordinateurs. Le suivi d’objets est alors devenu un sujet très actif depuis l’arrivée dans les années 1990 du traitement en temps réel des données. Les applications du suivi d’objets sont maintenant multiples et diverses : l’indexation automatique de vidéos, la vidéo surveillance, la reconnaissance de mouvement, la navigation autonome de véhicule, l’imagerie médicale, etc. Et il existe de nombreuses méthodes pour aborder le suivi d’objets, tout comme il existe de nombreux articles de synthèse pour énumérer et classer ces méthodes [Yilmaz et al., 2006; Cannons, 2008; Wu et al., 2013; Smeulders et al., 2014]. Yilmaz et al. [2006] les classe notamment en trois catégories : suivi de points (chaque objet est résumé par un point, par exemple son centre), suivi de fenêtres englobantes (fenêtre qui contient seulement l’objet) et suivi de silhouettes.

Dans notre étude expérimentale (voir Section I.2.2), nous connaissons à l’avance les formes et tailles des objets, et celles-ci ne varient pas dans le temps. Nous nous focalisons donc sur la catégorie suivi de points qui comporte les cas où les cibles peuvent être détectées à chaque instant de manière rapide et fiable. Ce problème de suivi d’objets multiples (MOT pour *multiple objects tracking*) devient alors un problème d’association des détections d’une image à l’autre. Dans la littérature, on parle de méthodes *detect-before-track* (DBT) ou plus récemment de *tracking-by-detection* [Andriluka et al., 2008; Breitenstein et al., 2011; Henriques et al., 2012]. Ces méthodes peuvent être classées en déterministes ou probabilistes.

Dans cette section, nous discutons des méthodes déterministes et probabilistes de suivi d'objets multiples en vision par ordinateur et nous présentons les métriques qui permettent de les évaluer.

### II.2.1 Méthodes déterministes

Le principe des algorithmes de suivi d'objets basés sur une méthode déterministe repose sur une approche standard en deux étapes : (1) la détection des objets dans une image, (2) le suivi des objets en résolvant le problème d'association des détections aux cibles de l'image précédente. La première étape de détection des objets est cruciale dans les méthodes déterministes. En effet, la deuxième étape d'association se base uniquement sur ces détections, aussi appelées observations. Un objet non détecté ou un objet détecté par erreur rend le suivi plus difficile et peut provoquer des erreurs importantes.

Les algorithmes de résolution du problème d'association de données utilisent généralement des informations sur le passé des trajectoires (positions, vitesses, accélérations) et des lois de la physique pour estimer les positions courantes et futures. Si ces informations sont exhaustives et qu'il n'y a pas d'erreurs de détections, l'association des particules et la prédiction de leur mouvement sont assez fidèles avec la réalité. Chaque événement est donc déterminé en vertu du principe de causalité, d'où le déterminisme.

Nous faisons ici un bref état des méthodes qui existent pour la détection et l'association dans des approches déterministes du suivi d'objets.

#### II.2.1.1 Détection d'objets

La détection d'objets dans une image digitale ne doit pas être confondue avec la reconnaissance d'objets. La reconnaissance permet d'identifier la classe d'un objet détecté parmi une base de classes d'objets connues. La détection d'objets, aussi appelée localisation d'objets, consiste, comme son nom l'indique, à détecter un objet particulier dans l'image et à en donner sa position précise. La reconnaissance s'apparentant plus à de la labellisation et catégorisation d'objets, d'humains ou de parties de corps, nous nous focalisons sur les méthodes de localisation plus adaptées à nos cas d'étude.

Il existe une multitude de méthodes pour détecter et localiser des objets. Les plus populaires sont la segmentation, la détection de points d'intérêt, la modélisation de fond, la classification supervisée et la comparaison à un modèle d'objet. Elles sont brièvement exposées ci-après.

#### Segmentation

La segmentation consiste à partitionner l'image en régions aux propriétés locales similaires. Méthode assez ancienne mais encore très utilisée dans divers domaines comme l'imagerie médicale ou la vidéo surveillance, on trouve un grand nombre d'algorithmes pour faire de la segmentation. Les différentes approches peuvent être répertoriées en trois grandes catégories : l'approche région qui consiste à regrouper les pixels ayant des propriétés locales similaires, l'approche contours qui consiste à détecter des frontières entre régions ayant des propriétés différentes et les méthodes mixtes qui combinent les deux approches précédentes. Les méthodes s'appuient également sur des modèles de natures différentes. On distingue notamment les modèles géométriques (morphologie mathématique), statistiques (classification, estimation statistique) ou fonctionnels (gradients, équations aux dérivées partielles).

On retrouve aussi des méthodes de segmentation spécifiques au suivi d'objets comme la segmentation par le mouvement. Cette dernière méthode cherche à différencier et extraire de l'arrière plan des objets en mouvement à l'aide par exemple de recherches de similitudes des flots optiques [Zhang et al., 2007] ou de l'estimation conjointe du mouvement d'une segmentation des objets et d'une image en super-résolution [Shen et al., 2007]. Ces méthodes peuvent également intégrer une étape d'estimation du mouvement de la caméra, qui peut ensuite être compensé [Bewley et al., 2014]. La segmentation est très largement abordée dans la littérature, elle présente des avantages pour certaines applications et des inconvénients pour d'autres, il n'existe pas de méthode unique et efficace en toutes situations.

### Points d'intérêt

Le principe de ces méthodes est de modéliser l'objet à segmenter par un ensemble de points caractéristiques (points d'intérêt) associés à un descripteur local et de rechercher dans l'image les sous-ensemble de points d'intérêt qui sont en bonne correspondance avec ce modèle. Un point d'intérêt est un point de l'image possédant des propriétés locales particulières qui permettent de le détecter de manière robuste et précise à l'aide d'opérateurs mathématiques particuliers. Il doit correspondre à une région ayant des propriétés locales représentatives des objets ou de la texture de l'image et présentant une bonne stabilité vis-à-vis des variations liées à l'éclairage et aux changements de point de vue ou d'échelle de la scène. Les points d'intérêt sont généralement issus des contours, des coins, des lignes de crêtes ou des régions présentant un maximum ou un minimum local d'intensité. Dans la littérature, on trouve le plus ancien détecteur de Moravec [Moravec, 1979], le plus répandu détecteur de Harris [Harris and Stephens, 1988], ou encore le détecteur multi-échelle intégré dans la méthode SIFT (pour *scale-invariant feature transform*) [Lowe, 2004] très utilisé dans les problèmes de mise en correspondance d'images. Ces méthodes de détection sus-citées ainsi que de nombreuses autres sont évaluées et comparées dans le travail de Mikolajczyk and Schmid [2005]. En le plaçant dans un contexte de mise en correspondance d'images ou de segmentation par le mouvement, le détecteur de points d'intérêt peut aussi permettre le suivi d'objets en mouvement [Ochs et al., 2014]. Plus spécifiquement, Gauglitz et al. [2011] ont évalué les différents détecteurs de points d'intérêt utilisés dans le domaine du suivi d'objets. La détection par points d'intérêt est donc une méthode bien adaptée lorsqu'on ne dispose pas de modèle précis de l'objet à détecter ou lorsque ce modèle possède de fortes variations de forme apparente.

### Modélisation de fond

La modélisation de fond (*background modelling*) consiste à construire une représentation de la scène, appelée modèle de fond, pour pouvoir détecter les objets en mouvement par différence avec ce modèle de fond. Ainsi, chaque point du fond peut être modélisé par une distribution gaussienne [Wren et al., 1997], un mélange de gaussiennes (GMM pour *gaussian mixture model*) [Stauffer and Grimson, 2000], une distribution non paramétrique estimée par un noyau (KDE pour *kernel density estimation*) [Elgammal et al., 2000], un modèle statistique mis à jour par filtre temporel médian [Cucchiara et al., 2003], une décomposition sur une base de vecteurs propres issus d'un échantillon d'images (*eigenbackgrounds*) [Oliver et al., 2000], et encore d'autres méthodes, tant le sujet est abordé dans de nombreuses applications (vidéo surveillance, capture de mouvement optique, interaction homme-machine, etc.). Les méthodes de modélisation de fond ont fait l'objet de plusieurs articles de syn-

thèse [Benezeth et al., 2008; Sobral and Vacavant, 2014] et sont aussi le sujet de livres comme celui de Bouwmans et al. [2014]. Le but de la modélisation de fond est de trouver les régions en mouvement par différences avec un modèle de fond. Les régions connectées de chaque image constituent des objets en mouvement, alors facilement localisables par traitement d'images. Cependant, les changements de fond, mouvements de caméras ou objets longtemps à l'arrêt peuvent engendrer des problèmes avec une telle approche.

### Classification supervisée

La classification supervisée est une méthode d'apprentissage statistique permettant de construire un modèle pour séparer des populations d'individus à partir de leurs attributs. Elle peut notamment être utilisée pour détecter des objets. Pour ce faire, il faut disposer d'une représentation d'une région de l'image par exemple sous la forme d'un vecteur, puis il faut apprendre à reconnaître les deux classes de région correspondant aux objets et au fond. En balayant toutes les régions de l'image, on peut ainsi détecter celles qui correspondent à l'objet recherché. On peut par exemple utiliser un classifieur basé sur une machine à vecteurs de support (SVM pour *support vector machine*) [Papageorgiou et al., 1998] ou un réseau de neurones [Rowley et al., 1998]. Dans le domaine du suivi d'objets, la classification supervisée a été beaucoup utilisée et notamment en *tracking-by-detection* où elle présente de très bons résultats [Andriluka et al., 2008; Breitenstein et al., 2011; Henriques et al., 2012]. Il existe aussi des méthodes dans lesquelles le détecteur est appris au fur et à mesure que le suivi des objets progresse dans la séquence, ce qui permet de prendre en compte les variations au cours du temps et d'éviter de reproduire des erreurs de détection. Cette approche est appelée *tracking-learning-detection* (TLD) [Kalal et al., 2012]. Pour résumer, les méthodes utilisant de la classification supervisée consistent à trouver la classe d'un objet en se basant sur une base de données apprise au préalable avec des exemples connus. Le principal inconvénient est qu'il faut disposer d'une large collection d'exemples pour établir la base d'apprentissage.

### Comparaison à un modèle d'objet

La comparaison à un modèle d'objet (*template matching*) est une technique de traitement d'images pour trouver les parties d'une image qui correspondent le mieux à un modèle donné. Le modèle de référence de l'objet ou partie de l'image que l'on recherche est généralement une représentation de l'apparence de l'objet dans l'image (aussi appelée *template*). Ce modèle peut être une liste des valeurs brutes d'intensité de l'objet [Alt et al., 2010], un sous-espace prenant en compte les variations d'apparence de l'objet [Ross et al., 2008], une représentation parcimonieuse [Mei et al., 2011], ou encore un histogramme d'orientation du gradient (HOG) [Dalal and Triggs, 2005]. Ensuite, l'idée est de chercher ce modèle dans l'image et de calculer en chaque pixel une valeur de similarité entre le voisinage du pixel et le modèle. Plusieurs méthodes de comparaison peuvent être utilisées comme la corrélation croisée, la corrélation croisée normalisée, le coefficient de corrélation, la distance euclidienne, etc. Brunelli [2009] fait la liste de toutes les méthodes existantes à ce sujet. Ces méthodes peuvent également être utilisées conjointement avec la classification supervisée [Henriques et al., 2012]. Les méthodes de comparaison à un modèle de l'objet donnent généralement de très bons résultats si l'objet n'a pas trop de changement d'apparence mais elles peuvent être assez lourdes en temps de calcul car le modèle doit être balayé sur toute l'image et l'opération de comparaison peut être coûteuse (similarité à l'échelle du pixel).

### II.2.1.2 Problème d'association

Une fois les objets détectés dans l'image, il faut associer (on dit aussi assigner ou affecter) chacune des détections aux détections de l'image précédente. En fait, les détections sont plutôt assignées aux traqueurs, aussi appelés cibles ou *trackers*, déjà créées au cours des pas de temps précédents. Cette étape est appelée résolution du problème d'association de données. Le problème réside dans l'assignation d'au maximum une détection à au maximum un traqueur. Ce problème n'est pas trivial lorsque qu'il y a plusieurs objets dans l'image, et d'autant plus quand ces objets sont très nombreux, proches, se déplacent en même temps ou s'occulent entre eux. Pour le résoudre, il faut d'abord choisir une fonction de dissimilarité (ou de similarité) entre les objets qui permet ensuite de définir un coût (ou un score) pour chaque association possible. Ces fonctions, aussi appelées critères d'association, peuvent être une combinaison de plusieurs contraintes comme la proximité, la vitesse maximale, le plus petit changement de vitesse, le même déplacement et la rigidité. Le choix du ou des critères d'association est ce qui fait entre autres l'originalité de chacune des méthodes de résolution du problème d'association de données.

L'étape suivante est un problème d'optimisation combinatoire visant à minimiser les coûts (ou maximiser les scores) d'associations. Il existe plusieurs stratégies pour aborder la résolution de ce problème : l'optimisation locale classique des associations de deux images consécutives [Sethi and Jain, 1987], cette même stratégie mais appliquée dans les deux sens temporels (une version de la première vers la deuxième image et une version de la deuxième vers la première image, chaque version prenant en compte les résultats de l'autre) [Salari and Sethi, 1990], l'optimisation locale élargie avec la prise en compte des associations sur plusieurs images consécutives pour préserver une cohérence temporelle des trajectoires [Shafique and Shah, 2005], l'optimisation globale en considérant la totalité de la séquence d'images avant de valider la solution [Kuhn, 1955]. La solution obtenue présente alors la meilleure (globalement ou localement selon les méthodes) combinaison des associations possibles qui fait correspondre les détections des images aux traqueurs.

Les méthodes classiques de résolution du problème d'association dans les suivis déterministes incluent l'algorithme du plus proche voisin, l'algorithme hongrois et l'algorithme glouton. Ils sont exposés brièvement ci-après.

#### Algorithme du plus proche voisin

Une des premières méthodes de résolution du problème d'association est une approche évidente qui consiste à associer à un traqueur la détection la plus proche de sa position estimée. Cette approche est connue comme l'algorithme du plus proche voisin [Crowley et al., 1988]. C'est une technique très populaire de par sa simplicité et son efficacité. Elle peut être particulièrement adaptée dans certaines situations (par exemple quand les objets sont suffisamment éloignés). Cependant, elle est considérée comme sous-optimale car lorsque des traqueurs proches se déplacent ensemble, il y a des chances de mauvaises associations. Les choix d'association ne sont pas remis en cause au cours des pas de temps suivants, ce qui est une limitation importante de cette approche.

#### Algorithme hongrois

L'algorithme hongrois [Kuhn, 1955], aussi appelé algorithme de Kuhn-Munkres [Munkres, 1957], résout le problème d'association de manière optimale sur sa globalité. Il permet de trouver le couplage parfait qui minimise le coût des associations sur la totalité de la séquence

à associer. Pour ce faire, l'algorithme regarde toutes les combinaisons possibles d'associations et garde la solution globale la meilleure. En terme de résultats, il est le meilleur des algorithmes d'optimisation combinatoire. Mais le principal problème de cette méthode réside dans sa complexité. En effet, son temps d'exécution est polynomial, cubique en fonction du nombre de trajectoires, ce qui peut être un facteur limitant.

### Algorithme glouton

L'algorithme glouton est une méthode qui choisit, étape par étape, la combinaison d'associations qui présente localement la solution optimale. Cet algorithme est souvent utilisé notamment en vertu de son faible coût computationnel par rapport à l'algorithme hongrois [Sethi and Jain, 1987; Rangarajan and Shah, 1991; Veenman et al., 2001; Shafique and Shah, 2005]. La principale limitation de l'algorithme glouton est qu'il peut tomber dans une solution qui n'est optimale que localement. Cependant, en pratique, cela peut être suffisant, comme énoncé par Wu and Nevatia [2007].

## II.2.2 Méthodes probabilistes

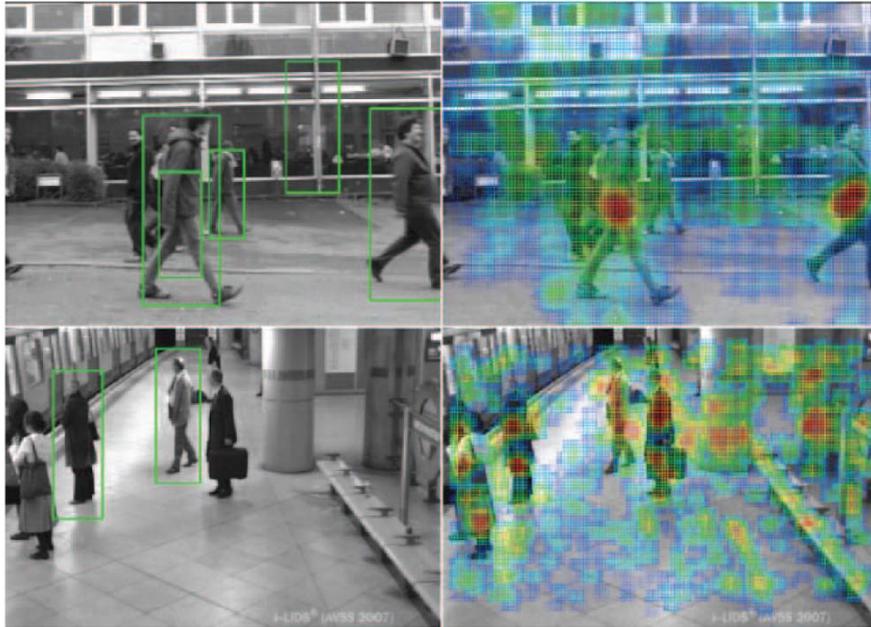
Les observations obtenues avec un algorithme de détection peuvent présenter des erreurs dues au bruit ou à des configurations complexes. Ces configurations peuvent être provoquées par des mouvements aléatoires ou des changements d'apparence d'un objet d'une image à l'autre. Pour faire face à ce type de problème, des méthodes probabilistes de suivi exploitant les imprécisions, incertitudes et variations de mesures ont été développées. Ces méthodes probabilistes utilisent, tout comme les méthodes déterministes, une première étape de détection des objets et une étape d'association. Mais elles présentent une étape supplémentaire de filtrage. Nous décrivons ci-après ces trois étapes.

### II.2.2.1 Détection des objets

La détection d'objets se fait de la même manière que pour les méthodes déterministes (voir Section II.2.1.1). La différence est que, dans une approche probabiliste, les détections obtenues ne sont plus considérées comme étant 100% fiables. En plus de son état (position, apparence, orientation, etc.), la détection d'un objet est donnée avec une probabilité, plus ou moins forte en fonction de la fiabilité de sa présence. Breitenstein et al. [2011] ont introduit le terme de confiance du détecteur (*detector confidence*). La confiance du détecteur est la sortie brute du détecteur d'objets, comme la valeur de corrélation par exemple, ensuite utilisée pour le seuillage final. Elle permet de former une densité de confiance, soit une carte de probabilité de présence de détection. Pour récupérer les détections individuelles, il est habituellement appliqué une recherche des maximums locaux dans cette carte pour ne garder que les détections ayant une confiance élevée. Mais cette étape peut également supprimer des informations potentiellement utiles (voir Figure II.2). Garder toute la densité de confiance permet alors de pallier cette perte d'informations, et ainsi aide à guider l'algorithme de suivi quand des détections sont manquantes. C'est une technique qui est notamment utilisée dans un algorithme développé pendant cette thèse (voir Chapitre V).

### II.2.2.2 Problème d'association

Les méthodes de résolution du problème d'association dans un suivi probabiliste sont soit celles utilisées classiquement dans les suivis déterministes (voir Section II.2.1.2), soient



**Figure II.2** Illustration de deux exemples (un en haut et un en bas) montrant deux types de sortie d'un détecteur de personnes : à gauche les détections à haute valeur de confiance (rectangle vert), à droite la confiance du détecteur (surimpression nuancée). La densité de confiance contient souvent des informations utiles à la position des détections manquantes. Illustration tirée de Breitenstein et al. [2011].

des méthodes statistiques telles que le filtre de probabilité conjointe et le suivi par hypothèses multiples. Nous en discutons très brièvement ci-après.

### Filtre de probabilité conjointe

Le filtre de probabilité conjointe (JPDAF pour *joint probabilistic data association filter*) est une méthode bayésienne de Fortmann et al. [1983] qui cherche à faire la meilleure combinaison d'associations possible à chaque pas de temps. Elle considère pour cela toutes les combinaisons d'associations entre les détections et les traqueurs de manière conjointe. Le problème principal de cette méthode est que le nombre de traqueurs doit être connu et resté constant entre deux images.

### Suivi par hypothèses multiples

Le suivi par hypothèses multiples (MHT pour *multiple hypothesis tracking*) de Reid [1979] considère plusieurs combinaisons d'associations possibles sur plusieurs pas de temps consécutifs, appelées hypothèses. Le nombre de pas de temps est généralement fixé à seulement quelques uns en raison de la complexité qui augmente rapidement lorsqu'il est grand. La complexité augmente de manière exponentielle avec le nombre de traqueurs.

#### II.2.2.3 Méthodes de filtrage

A l'inverse des méthodes déterministes de suivi, les approches probabilistes cherchent à utiliser les incertitudes de mesure pour guider le suivi. Pour ce faire, ces méthodes utilisent un ou des modèles de filtrage pour corriger les positions erronées des détections et mieux

prédire les positions futures. Les méthodes les plus classiques sont le filtre de Kalman, le filtrage particulaire et le filtrage à modèles multiples. Elles sont brièvement décrites ci-après.

### Filtre de Kalman

Le filtre de Kalman [Kalman, 1960] est un filtre à réponse impulsionnelle infinie qui permet d'estimer l'état d'un système linéaire lorsque que celui-ci est supposé avoir une distribution gaussienne. Lorsque le système est non-linéaire, on parle de filtre de Kalman étendu [Bar-Shalom, 1987] ou de filtre de Kalman "unscented" [Julier and Uhlmann, 1997].

Le filtre de Kalman s'appuie sur les mesures bruitées et incomplètes pour prédire le nouvel état puis le corriger. Dans la première étape de prédiction, un modèle d'état est appliqué sur l'état estimé à l'instant précédent afin d'estimer l'état courant. La deuxième étape de correction utilise alors les observations de l'instant courant pour corriger cet état estimé. Le filtre de Kalman est donc une méthode d'estimation et de prédiction puissante qui prend en compte un modèle dynamique du système et un modèle de mesures bruitées.

Le principal inconvénient de cette méthode est que le développeur doit modéliser le système assez précisément pour concevoir un filtre efficace. De plus, si le bruit n'est pas gaussien, la méthode n'est pas adaptée.

### Filtrage particulaire

Le filtrage particulaire, ou filtre à particules, est une alternative au filtre de Kalman, il peut s'adapter à des systèmes non-linéaires et à des distributions non-gaussiennes. Dans la littérature, le filtrage particulaire de Del Moral [1996] est aussi trouvé sous d'autres noms tels que le filtre *bootstrap* [Gordon et al., 1993], la méthode séquentielle de Monte-Carlo [Liu and Chen, 1998; Doucet et al., 2001] ou l'algorithme de condensation [Isard and Blake, 1998b]. Le filtrage particulaire est populaire dans des applications comme la robotique [Choset and Nagatani, 2001], l'analyse sportive [Lu et al., 2009], la vidéo surveillance [Rathi et al., 2007], etc.

Le principe est de modéliser l'état de chaque traqueur en incluant des éléments non-linéaires et non-gaussiens, et de procéder à la volée (ou en ligne, *i.e.* traiter les données au fur et à mesure qu'elles arrivent). Pour cela, le filtrage particulaire implémente un filtre bayésien récursif par la méthode de Monte Carlo. Il utilise pour ce faire un modèle dynamique pour la prédiction de l'état d'un traqueur et un modèle d'observation pour établir la vraisemblance de l'état prédit. L'état d'un traqueur est représenté par un échantillon d'états potentiels, aussi appelés particules. Par exemple, pour un traqueur avec ses particules à un instant donné, les étapes principales de l'estimation de l'état à l'instant suivant sont : (1) propager les particules à l'aide d'un modèle de mouvement, (2) donner un poids de vraisemblance à chaque particule à l'aide d'un modèle d'observation, (3) rééchantillonner la distribution des particules pour ne garder que les plus vraisemblables. Ces particules rééchantillonnées définissent alors au mieux le traqueur à l'instant suivant. L'application d'une opération comme la moyenne ou la variance sur ces particules permet ensuite de récupérer l'état final estimé du traqueur.

Le filtrage particulaire présente cependant une complexité assez élevée, et d'autant plus quand le nombre de particules est élevé. De plus, la plupart des algorithmes de filtrage particulaire dépendent d'un seul modèle de mouvement, ce qui peut être problématique dans un environnement avec des dynamiques complexes.

### Filtrage à modèles multiples

Pour faire face à la limitation du filtrage particulaire dans des conditions de dynamiques complexes, le filtrage à modèles dynamiques multiples a été introduit [Arulampalam et al., 2004; Li and Jilkov, 2005]. Dans une approche de suivi stochastique basé sur la dynamique des objets [Odobez et al., 2006], l'idée principale est d'utiliser les mesures explicites de mouvement pour guider les prédictions. Cette approche est aussi connue sous le nom d'approche à modèle dynamique changeant (*switching dynamical model*). Elle est notamment utilisée dans des applications comme la robotique et la navigation [Boers and Driessen, 2003] ou la vidéo surveillance [Zhai et al., 2009].

Elle suppose que, étant donné un nombre fini de modèles dynamiques (chacun correspondant à un comportement particulier), un traqueur peut passer d'un modèle à un autre selon une table de probabilité de transition. L'algorithme à modèle multiple interactif (IMM pour *interacting multiple model*) [Blom and Bar-Shalom, 1988; Mazor et al., 1998] était le premier à proposer cette approche en se basant sur un processus de Markov à sauts (*jump Markov process*). Puis, le filtrage particulaire à modèle multiple (MMPF pour *multiple model particle filter*) [Isard and Blake, 1998a; McGinnity and Irwin, 2000] est arrivé comme une alternative prometteuse qui présente des meilleurs résultats que l'algorithme IMM dans certaines situations [Arulampalam et al., 2004]. L'idée de la méthode MMPF, comme son nom l'indique, est de se placer dans un cadre de filtrage particulaire et d'utiliser des modèles dynamiques multiples (*i.e.* les mouvements d'objets peuvent être classifiés en différents états de mouvement selon la situation). Chaque état de mouvement décrit un modèle spécifique d'évolution d'état pour l'étape de prédiction.

Cette méthode requiert de connaître tous les états de mouvement au préalable avec une description mathématique pour chacun. Ce point peut être une limitation si le développeur n'a pas une connaissance approfondie sur le comportement des objets. Étant donné que nous avons une bonne connaissance du comportement des billes dans notre écoulement, un algorithme MMPF est utilisé dans cette thèse (voir Chapitre V).

### II.2.3 Métriques d'évaluation

Le suivi d'objets multiples (MOT pour *multiple object tracking*) est un sujet très actif et difficile pour la recherche académique et dans de nombreuses applications. Comme énoncé précédemment, il existe de nombreuses méthodes MOT. Il est donc nécessaire d'évaluer ces méthodes et de les comparer. Pour ce faire, dans un premier temps, il faut définir quelles sont les qualités d'un bon algorithme MOT. En général, et à la manière de Smith et al. [2005], on dit qu'un bon algorithme MOT doit : (1) bien suivre les objets - placer le bon nombre de traqueurs aux bonnes positions, (2) bien identifier les objets - suivre les objets individuels sans arrêt sur de longues séquences, (3) estimer précisément les paramètres des objets (comme la vitesse par exemple), (4) être rapide - avoir un coût computationnel faible. Dans un second temps, il faut utiliser des métriques permettant d'évaluer ces qualités par rapport à une vérité terrain (*ground truth*, *i.e.* version de la séquence de suivi étiquetée par un expert). Il existe pour cela les mesures classiques des algorithmes à décision, comme le nombre de fausses détections (ou faux positifs), le nombre de détections manquantes (ou faux négatifs), la précision, le rappel, etc. ; et les mesures plus adaptées à l'étude des algorithmes MOT comme les métriques CLEAR MOT de Bernardin and Stiefelhagen [2008]. À l'aide de ces mesures, il est alors possible d'étudier les performances d'un algorithme MOT et de le comparer à d'autres algorithmes. Nous décrivons les métriques ci-après. Beaucoup d'entre

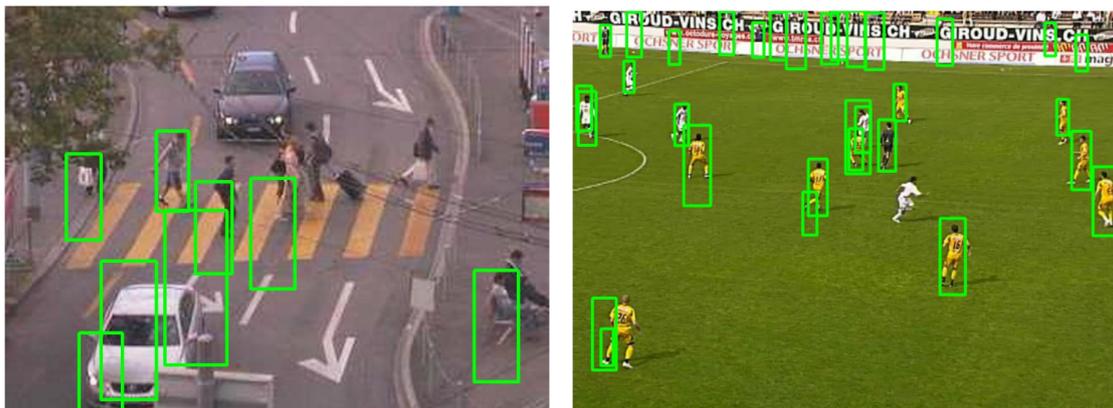
elles sont utilisées dans notre travail.

### II.2.3.1 Métriques classiques

Les métriques, ou mesures, classiques pour l'évaluation d'un algorithme MOT se basent quasiment toutes sur les deux grandeurs que sont les faux positifs et les faux négatifs. On retrouve notamment la précision et le rappel qui sont souvent interprétés conjointement dans un même graphique appelé courbe précision-rappel, la sensibilité et la spécificité également interprétés conjointement dans un même graphique appelé courbe ROC, l'exactitude (*accuracy*), le F-score, etc. Dans un autre registre, il y a aussi une mesure qui calcule le nombre d'échanges d'identités de traqueur. Dans cette section, nous ne détaillons que les faux positifs, faux négatifs et les échanges d'identités de traqueur.

#### Faux positifs

Un faux positif est un résultat d'une prise de décision donné positif alors qu'il devrait en réalité être négatif. Dans un cadre MOT, un faux positif, aussi appelé fausse détection, se produit lorsqu'un objet est détecté à une position alors qu'il ne devrait pas, c'est-à-dire qu'il n'y a pas réellement d'objet à cette position d'après la vérité terrain (Figure II.3). La mesure utilisée est le nombre de faux positifs FP, par image ou dans toute la séquence.



**Figure II.3** Illustrations de multiples faux positifs et faux négatifs typiques d'un détecteur de personnes. Illustrations tirées de Breitenstein et al. [2011].

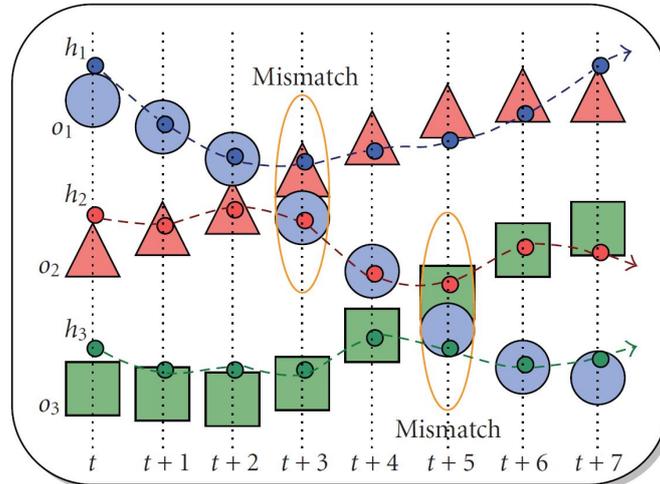
#### Faux négatifs

Un faux négatif est un résultat d'une prise de décision donné négatif alors qu'il devrait en réalité être positif. Dans un cadre MOT, un faux négatif, aussi appelé détection manquante, se produit lorsqu'un objet n'est pas détecté à une position alors qu'il le devrait, c'est-à-dire qu'il y a réellement un objet à cette position d'après la vérité terrain (Figure II.3). La mesure utilisée est le nombre de faux négatifs FN, par image ou dans toute la séquence.

#### Échanges d'identités de traqueurs

Un échange d'identité (ID) de traqueur (*ID switch*) intervient lorsque l'étape d'association a confondu deux traqueurs et échangé leur ID. Ce cas peut arriver quand deux trajectoires se croisent (un objet est occulté par l'autre), quand un nouvel objet d'apparence similaire apparaît proche d'un autre ou quand une trajectoire perdue est réinitialisée avec

un ID différent. Lorsqu'une trajectoire est coupée en plusieurs morceaux d'ID différents, un échange d'ID est comptabilisé à chaque changement d'ID (et non pas à chaque ID différent du morceau d'ID le plus long) (Figure II.4). Rattacher les morceaux d'ID pour retrouver la trajectoire réelle de l'objet est une étape difficile et qui ne peut être réalisée qu'en post-traitement de l'algorithme de suivi. La mesure utilisée est le nombre d'échanges d'ID. Une valeur faible est une des propriétés les plus importantes d'un bon algorithme MOT.



**Figure II.4** Illustration d'échanges d'identités de traqueurs. Sur 8 images consécutives (de  $t$  à  $t + 7$ ), les identités de traqueur, ou hypothèses d'objets, issues du suivi d'objets sont notées  $\{h_1, h_2, h_3\}$  et les objets réels sont notés  $\{o_1, o_2, o_3\}$ . Ici,  $h_2$  est d'abord rattachée à  $o_2$ . Après quelques images,  $o_1$  et  $o_2$  se croisent et  $h_2$  suit le mauvais objet. Plus tard,  $h_2$  échange encore avec  $o_3$ . Dans ce cas, on dénombre donc 2 échanges d'identité. Illustration tirée de [Bernardin and Stiefelwagen \[2008\]](#).

### II.2.3.2 Métriques CLEAR MOT

Dans leur article, [Bernardin and Stiefelwagen \[2008\]](#) proposent deux métriques intuitives, indépendantes et assez générales pour mesurer et comparer la performance et les caractéristiques d'algorithmes MOT. La mesure MOTP se concentre sur la précision et la mesure MOTA sur l'exactitude. Nous les introduisons ici.

#### MOTP

La mesure MOTP (*multiple object tracking precision*) est une estimation de la précision des positions des objets. Elle calcule l'erreur totale des distances euclidiennes de l'estimation des positions (par rapport aux positions d'une vérité terrain) sur toute la séquence d'images et en fait une moyenne. Seuls les objets ayant été associés correctement sont ici considérés, les faux positifs et faux négatifs n'en font pas partie. Cette mesure ne prend donc pas en compte les erreurs comme les faux positifs, faux négatifs et échanges d'ID. Dans la littérature, l'erreur existe sous la forme d'une distance par rapport à la vérité terrain (exprimée en mètres ou pixels, plus cette valeur est petite, meilleure est la précision) ou sous la forme d'une superposition de la fenêtre englobante de l'objet par rapport à la vérité terrain (exprimée en pourcentage, plus cette valeur est élevée, meilleure est la précision).

**MOTA**

La mesure MOTA (*multiple object tracking accuracy*) est une estimation de l'exactitude, c'est-à-dire la capacité à détecter les objets et suivre leur trajectoire de manière cohérente sur toute la durée de la séquence. Cette mesure est indépendante de la précision des positions des objets. Elle prend en compte toutes les erreurs faites par l'algorithme dues à des configurations d'objets, à savoir les faux positifs, les faux négatifs et les échanges d'ID, et ce sur toute la séquence.

**II.3 Synthèse et rapprochement à notre problématique**

Les méthodes citées dans ce chapitre ne sont pas toutes adaptées à la problématique qui est la nôtre, à savoir la détection et le suivi de multiples objets similaires.

Dans un premier temps, les objets que l'on cherche à détecter, des billes en l'occurrence, ont une apparence (forme, taille, couleur) qui ne change pas au cours du temps. Donc les méthodes de détection du type détecteur de points d'intérêt et classification supervisée ne semblent pas pertinentes ici car nous possédons déjà un modèle précis des objets et n'avons alors pas besoin d'une base d'apprentissage pour les reconnaître. Les méthodes de détection basées sur la segmentation, la modélisation de fond et la comparaison à un modèle d'objet apparaissent plus adaptées et ont ainsi particulièrement attiré notre attention.

Dans un deuxième temps, le nombre de billes par image étant assez élevé (env. 400 billes), l'utilisation d'un algorithme hongrois pour la résolution du problème d'association des billes serait beaucoup trop coûteux en raison de sa complexité polynomiale. D'autre part, les collisions étant nombreuses lors du phénomène de ségrégation en charriage, la méthode du plus proche voisin provoquerait assurément des erreurs dans les associations. Pour ce qui est des méthodes probabilistes, les méthodes JPDAF et MHT ne s'avèrent pas adaptées à cause du grand nombre d'objets à suivre rendant leur complexité importante et du fait que ce nombre varie constamment d'une image à l'autre (les billes entrent et sortent du champ de vision de la caméra tout au long de la séquence). Nous nous sommes alors plus intéressés à l'algorithme glouton qui retourne une combinaison d'associations avec une optimisation locale à l'image. Il apparaît être un très bon compromis entre un faible temps d'exécution et une bonne précision des résultats.

Un des objectifs de cette thèse est de permettre l'étude des trajectoires sur des très longues séquences d'images (env. 500000 images). Par définition, la PIV n'est donc pas adaptée car elle ne permet pas le suivi individuel des objets. L'approche déterministe par contre, et notamment la PTV, semble bien appropriée à ce type d'étude du fait de sa faible complexité et donc de son exécution en un temps raisonnable pour un résultat plus qu'honnête (voir Chapitre IV).

Cependant, pour l'étude des moments clés du phénomène de ségrégation (*i.e.* lors des forts régimes transitoires), il faut être capable de résoudre un problème important des algorithmes déterministes. Ce problème est celui des erreurs liées aux faux positifs, aux faux négatifs et aux changements d'identité des trajectoires. Ces erreurs ne permettent pas de caractériser avec une fiabilité suffisante des événements rares liés par exemple au phénomène d'infiltration des petits objets. Nous avons donc étudié un deuxième algorithme basé sur une approche statistique par filtrage particulière, et plus particulièrement, un filtrage particulière à modèles dynamiques multiples (MMPF) pour pouvoir suivre les différents comportements des billes de manière adaptée (voir Chapitre V).

### II.4 Conclusion

Ce chapitre était consacré à l'état de l'art des méthodes de suivi d'objets. Nous avons d'abord vu les deux méthodes d'imagerie classiquement utilisées en mécanique des fluides et notamment pour le suivi de sédiments, à savoir la PIV permettant d'étudier les vitesses instantanées dans un écoulement et la PTV permettant d'étudier les trajectoires individuelles de particules dans un fluide.

Ensuite, nous avons fait une revue du suivi d'objets en vision par ordinateur. Nous avons constaté que les méthodes de *tracking-by-detection* se basent la plupart du temps sur une approche en deux étapes : la détection des objets et leur association entre les images. Chacune de ces étapes présente différentes méthodes en fonction du domaine d'application, toutes ayant des avantages et inconvénients selon la situation. A la différence des méthodes déterministes, les méthodes probabilistes apportent en plus une étape de filtrage qui permet, grâce à un modèle dynamique et un modèle d'observation, de corriger les erreurs de détections et mieux prédire les positions. Face à l'inconvénient d'un modèle unique, nous avons vu que les filtrages à modèles multiples sont une bonne alternative aux célèbres filtres de Kalman et filtres particulaires.

**Deuxième partie**

**CONTRIBUTIONS**



## Chapitre III

# Traitement d'images pour l'étude de la mobilité du lit lors de l'infiltration de particules fines

### Sommaire

---

<b>III.1 Contexte, problématique et objectifs . . . . .</b>	<b>40</b>
<b>III.2 Algorithme de détection des lignes du lit et de l'eau . . . . .</b>	<b>41</b>
III.2.1 Séquences d'images . . . . .	41
III.2.2 Problème de segmentation . . . . .	41
III.2.3 Description de l'algorithme de détection des lignes . . . . .	42
III.2.4 Validation des résultats . . . . .	45
<b>III.3 Résultats et discussions . . . . .</b>	<b>45</b>
III.3.1 Évolution de la pente du lit . . . . .	47
III.3.2 Évolution de la hauteur d'eau . . . . .	48
III.3.3 Éléments de discussion complémentaires . . . . .	48
<b>III.4 Conclusion . . . . .</b>	<b>50</b>

---

Dans ce chapitre, nous détaillons les méthodes de traitement d'images qui ont été développées pour permettre l'analyse de l'infiltration de particules fines dans un lit mobile sur le canal expérimental en version élargie. Ces méthodes ont notamment permis d'étudier l'évolution de la hauteur d'eau et pente du lit au cours du temps, ce qui permet d'avancer dans la compréhension de processus fondamentaux en mécanique du charriage. Ce travail de recherche appliquée est utilisé pour l'analyse d'écoulements en mécanique des fluides, ce qui explique les contributions modestes pour le domaine de traitement d'images. Il est notamment au coeur d'une collaboration internationale avec Ashley Dudill et l'Université de Colombie Britannique de Vancouver, Canada.

Notre algorithme de segmentation combinant des méthodes de traitement d'images a fait l'objet d'une présentation et d'un article publié dans les actes de la conférence internationale *PSFVIP* de Naples, Italie en 2015. L'article de conférence est exposé tel qu'il a été publié en Annexe B, un résumé est donné dans ce chapitre. Par la suite, l'algorithme a été utilisé pour analyser plus largement et en détails l'introduction de particules fines en mode charriage, ce qui a donné lieu à une publication dans le *Journal of Geophysical Research : Earth Surface* (en

cours de révision) donné en Annexe C, et dont certains résultats obtenus sont discutés en fin de chapitre.

Nous présentons d'abord le contexte, la problématique et les objectifs de cette méthode de traitement d'images. Ensuite, nous exposons les différentes étapes de l'algorithme, suivi de l'évaluation des résultats. Enfin, nous discutons de résultats complémentaires obtenus avec notre algorithme.

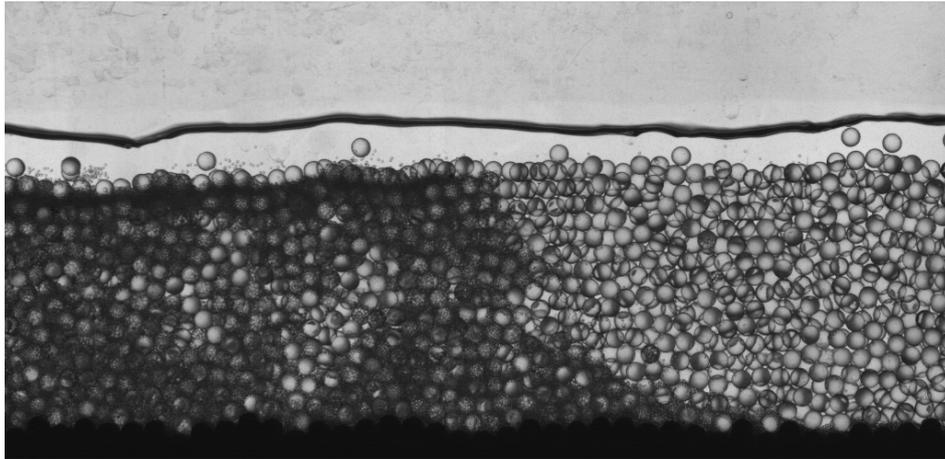
### III.1 Contexte, problématique et objectifs

Comme nous l'avons présenté dans le Chapitre I de cette thèse, différentes versions d'un canal expérimental à Irstea permettent d'étudier la dynamique du phénomène de ségrégation en transport solide par charriage. Une des deux versions du canal, le canal dit "élargi", a été spécifiquement conçu pour réaliser des expériences sur deux types de ségrégation liés à la taille lors du charriage : la ségrégation par percolation spontanée et la ségrégation par tamisage cinétique (voir Section I.1.3). Les expériences consistent à introduire des particules fines dans un lit mobile formé de particules plus grosses et d'observer l'évolution du lit avec l'infiltration qui se produit alors. En fonction du rapport entre la taille des particules fines (taille unique pour une même expérience et ayant pour valeur 0.7 mm, 0.9 mm, 1.5 mm, 2 mm, 3 mm ou 4 mm) et celle des particules grosses (5 mm pour toutes les expériences), la réponse du lit n'est pas la même. Comprendre comment ce rapport de taille influence la mobilité du lit suite à l'introduction de particules fines est une problématique de recherche importante. L'analyse de cette influence peut notamment être faite par l'étude de l'évolution de la pente du lit et de la hauteur d'eau au cours du temps après l'introduction des particules fines.

Les données récoltées à l'issue des expériences sont des séquences d'images (voir un exemple d'image en Figure III.1). Pour obtenir des informations de ces images, il est nécessaire de les traiter de manière spécifique. L'objectif ici est donc de réaliser un traitement automatique des images afin d'en faire ressortir des éléments qui permettent l'analyse de l'évolution du lit mobile, et plus particulièrement, l'évolution de la pente du lit et de la hauteur d'eau. La pente et hauteur d'eau peuvent être calculées à partir de la position de la ligne du lit et de la ligne d'eau. Ces lignes sont des caractéristiques des deux régions impliquées dans le transport de sédiment, à savoir le lit stable et le lit mobile.

Dans cette optique, nous avons développé un algorithme de traitement d'images pour réaliser une segmentation des images. Il est principalement basé sur des opérations de morphologie mathématique (voir Annexe A.1 pour un rappel des transformations morphologiques élémentaires) et de calculs de gradients. Il a été présenté lors de la 10<sup>ème</sup> conférence *Pacific Symposium on Flow Visualization and Image Processing (PSFVIP)* pour laquelle un article a aussi été écrit [Lafaye de Micheaux et al., 2015]. Cet article est résumé dans ce chapitre (les figures du chapitre sont tirées de l'article). L'article complet est exposé en Annexe B, il apporte notamment quelques détails supplémentaires sur le canal élargi et sa procédure expérimentale.

Dans un travail qui fait suite à l'élaboration de cet algorithme, nous avons appliqué l'algorithme sur un plus grand nombre de données et avons analysé les résultats [Dudill et al., 2016b]. Ces derniers travaux sont en cours d'examen dans le *Journal of Geophysical Research : Earth Surface* et l'état courant de l'article est donné en Annexe C. Quelques uns des résultats importants sont présentés en Section III.3.3.



**Figure III.1** Image typique d'un mélange de billes dans le canal élargi. Les anneaux gris foncé sont des billes grosses de 5 mm, les zones gris foncé sont des amas de billes fines (ici de taille 0.7 mm) et la ligne noire dans le haut de l'image correspond à la ligne d'eau.

### III.2 Algorithme de détection des lignes du lit et de l'eau

Pour étudier l'évolution de la pente du lit et de la hauteur d'eau, il est nécessaire d'avoir la position des lignes du lit et lignes d'eau. Pour ce faire, nous avons développé un algorithme de segmentation spécifique. Nous le détaillons dans cette section après avoir rappelé le type de séquences d'images traitées et exposé notre problème de segmentation.

#### III.2.1 Séquences d'images

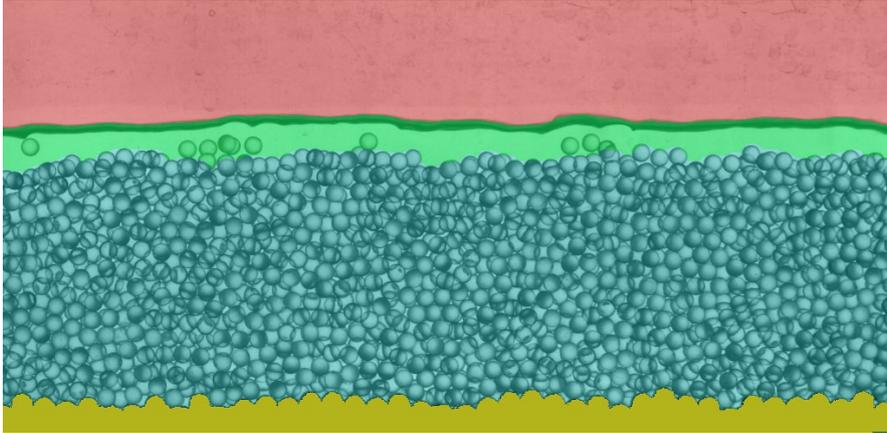
Les données obtenues pendant l'enregistrement de la période d'infiltration forment une séquence d'images en noir&blanc d'environ 20 minutes à une fréquence  $FPS = 130$  Hz (voir une séquence type en [Figure I.7](#)). Une fois que l'infiltration des particules fines dans le lit de grosses particules est presque terminée, les particules fines forment une couche mobile (comme un "tapis") d'épaisseur variable au-dessus de laquelle les grosses particules sont essentiellement en roulement (sauf pour le cas des particules fines de 4 mm). Pour notre étude de l'infiltration de particules fines, nous n'avons pas besoin de toutes les images de la séquence. Nous n'avons donc gardé qu'une seule image par seconde. Ainsi, une séquence est composée d'environ 1200 images de dimension  $1024 \times 500$  pixels. Sur les images, les grosses particules sont transparentes et les fines apparaissent gris foncé (sauf pour les billes fines de 3 mm et 4 mm qui apparaissent respectivement noires et transparentes). La ligne noire en haut des images correspond à la surface de l'eau, ou ligne d'eau.

#### III.2.2 Problème de segmentation

Un algorithme de traitement d'images a été développé pour évaluer l'évolution de la pente du lit au cours du temps. Le but est de segmenter les images pour extraire les limites des principales régions qui apparaissent au cours du transport de sédiments ([Figure III.2](#)), à savoir :

- La ligne de la base du canal, frontière entre la base fixe du canal et le lit mobile (frontière jaune-bleu),

- La ligne du lit, frontière entre le lit mobile et l'écoulement d'eau (frontière bleu-vert),
- La ligne d'eau, frontière entre l'écoulement d'eau et l'air (frontière vert-rouge).



**Figure III.2** Position des quatre régions distinctes dans les images : la base fixe du canal (jaune) correspond aux demi-cylindres posés aléatoirement ; le lit mobile (bleu) correspond aux billes quasi-immobiles composant le lit ; l'écoulement d'eau (vert) contient le fluide et les billes se déplaçant vite ; la partie supérieure est l'air (rouge).

Traditionnellement, ce type de problème de segmentation peut être résolu par une détection de contours ou une segmentation en régions. Beaucoup de solutions peuvent être envisagées, nous avons opté pour une approche morphologique basée sur un calcul de ligne de partage des eaux [Beucher and Lantuejoul, 1979]. Cette approche consiste à calculer le module du gradient, puis à détecter la ligne de crête séparant deux bassins positionnés de chaque côté de la ligne. Le principal avantage de cette méthode est qu'elle fournit la détection d'une seule ligne grâce à la définition de marqueurs pertinents utilisés comme sources d'inondation.

La combinaison de la ligne de partage des eaux et du gradient est une méthode de segmentation puissante introduite par Beucher and Lantuejoul [1979] et raffinée dans différentes travaux [Vincent and Soille, 1991; Beucher and Meyer, 1992]. Elle repose sur l'utilisation d'un calcul de ligne de partage des eaux qui détermine les frontières des objets adjacents comme des crêtes. Pour éviter la sur-segmentation, la ligne de partage des eaux est basée sur un ensemble de marqueurs. Cette méthode a été utilisée dans de nombreux problèmes de segmentation comme la segmentation de route [Beucher and Bilodeau, 1994], de vidéos [Patras et al., 2001] et d'images médicales [Grau et al., 2004]. Plus de détails sur cette méthode sont donnés en Annexe A.1.

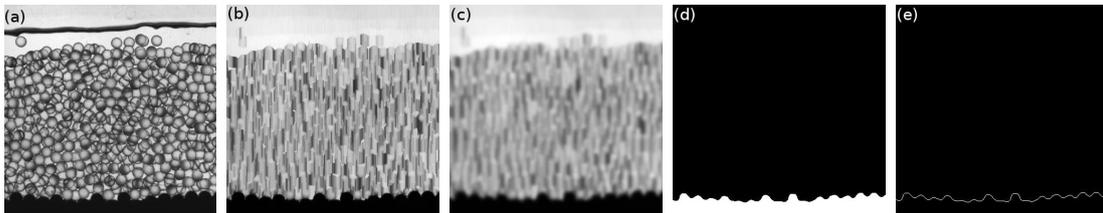
### III.2.3 Description de l'algorithme de détection des lignes

Pour détecter la surface du lit et celle de l'eau sur les images, nous détectons d'abord la position de la base du canal sur la première image de la séquence. Ensuite, sur chaque image, nous supprimons les billes en mouvement avec un algorithme d'extraction du fond. Et enfin, nous estimons la position des lignes du lit et de l'eau.

### III.2.3.1 Détection de la base du canal

La première étape de l'algorithme est de détecter la base du canal. Ceci est réalisé une seule fois, sur la première image, et est ensuite utilisé pour toutes les images de la séquence. Le principe de cette détection est d'appliquer un seuillage binaire pour détecter la région de la base fixe du canal. Ensuite, la limite supérieure de cette région est obtenue par des opérations morphologiques binaires simples. Bien que la base du canal soit la région la plus noire de l'image, il faut supprimer la contribution de la ligne d'eau et du lit mobile qui peuvent être noirs également. La détection de la base du canal procède donc par les étapes illustrées en [Figure III.3](#) et décrites ci-après :

1. Appliquer une fermeture de la première image de la séquence ([Figure III.3b](#)). Une ligne verticale est utilisée comme élément structurant, elle doit être assez longue pour supprimer la ligne d'eau. La longueur de cette ligne verticale est dénotée  $l$ . Dans nos images, la valeur optimale pour supprimer la ligne d'eau est  $l = 18$  pixels.
2. Filtrer l'image obtenue avec un filtrage gaussien d'écart-type  $\sigma$  pour éliminer les détails fins et le bruit ([Figure III.3c](#)). Le bruit est principalement causé par les billes transparentes dans le lit, il est supprimé en prenant  $\sigma = 3$  pixels.
3. Appliquer un seuillage de l'image filtrée ([Figure III.3d](#)). Le seuil est automatiquement fixé au premier pic de l'histogramme des intensités de pixels pour supprimer les parties grises de l'image.
4. Appliquer une dilatation de l'image binaire avec une ligne verticale de 2 pixels comme élément structurant.
5. Soustraire l'image binaire issue du point 3 de l'image dilatée pour faire ressortir la limite entre le noir et le blanc ([Figure III.3e](#)). L'image résultante donne la position de la ligne supérieure de la base du canal.



**Figure III.3** Procédure de détection de la base du canal. (a) image originale, (b) fermeture de l'image originale, (c) image filtrée, (d) image seuillée, (e) soustraction de l'image seuillée de sa dilatation donnant la ligne supérieure de la base du canal.

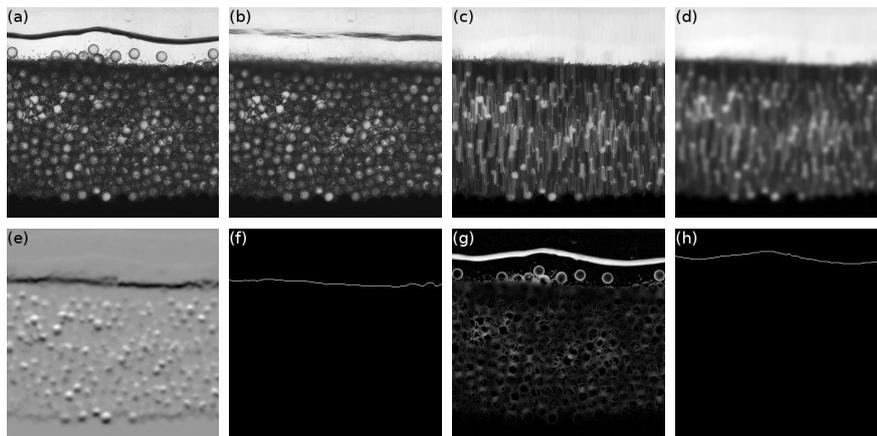
### III.2.3.2 Extraction du fond

La deuxième étape est de calculer la partie immobile d'une image en moyennant sur  $n$  images consécutives (en commençant  $\lceil (n-1)/2 \rceil$  images avant et en finissant  $\lfloor (n-1)/2 \rfloor$  images après l'image courante). Ce moyennage a pour effet de supprimer les billes qui sont en mouvement pendant l'intervalle de temps autour de l'image courante ([Figure III.4b](#)). Le nombre  $n$  d'images est choisi pour être un bon compromis entre avoir assez d'images pour avoir une bonne carte des particules en mouvement, mais pas trop non plus qui résulterait en une perte d'information à la surface du lit. Dans nos expériences, la valeur  $n = 30$  permet ce compromis, elle correspond à 30 secondes d'expérience.

### III.2.3.3 Détection de la ligne du lit

La troisième étape est la détection de la ligne supérieure du lit sur l'image du fond. Le principe est de calculer un gradient de Sobel [Sobel and Feldman, 1968] et de détecter la ligne de crête de ce gradient grâce à un calcul de ligne de partage des eaux contrôlé par des marqueurs. La détection de la ligne du lit procède donc par les étapes illustrées en Figure III.4 et décrites ci-après :

1. Appliquer une fermeture de l'image de fond avec une ligne verticale de longueur  $l = 18$  pixels comme élément structurant pour supprimer la ligne d'eau (Figure III.4c).
2. Filtrer l'image obtenue avec un filtre gaussien d'écart-type  $\sigma = 3$  pixels pour éliminer les détails fins et le bruit (Figure III.4d).
3. Calculer le gradient vertical sur l'image filtrée (Figure III.4e). Une opération de gradient vertical de Sobel est utilisée ici pour faire ressortir les variations verticales dans l'intensité des pixels.
4. Appliquer une segmentation par ligne de partage des eaux sur le complémentaire de l'image issue du gradient avec pour marqueurs le haut de l'image et la ligne supérieure de la base du canal (Figure III.4f). L'image résultante donne la position de la ligne du lit.



**Figure III.4** Procédure de détection des lignes du lit et ligne d'eau. (a) image originale, (b) image du fond, (c) fermeture de l'image du fond, (d) image filtrée, (e) gradient de l'image filtrée, (f) segmentation par ligne de partage des eaux sur le complémentaire de l'image gradient donnant la ligne du lit, (g) image obtenue en soustrayant l'image originale de l'image filtrée, (h) segmentation par ligne de partage des eaux sur la dernière image obtenue donnant la ligne d'eau.

### III.2.3.4 Détection de la ligne d'eau

La dernière étape est la détection de la ligne d'eau. Comme pour la ligne du lit, l'idée est de segmenter l'image par ligne de partage des eaux pour détecter la ligne de crête de la surface de l'eau. À partir des résultats obtenus après la détection de la ligne du lit, la procédure est la suivante :

5. Soustraire l'image originale de l'image filtrée issue du point 2 de la détection de la ligne du lit (Figure III.4g). L'image obtenue montre des fortes intensités aux pixels proche de et sur la surface de l'eau.
6. Appliquer une segmentation par ligne de partage des eaux avec pour marqueurs le haut de l'image et la ligne du lit (Figure III.4h). L'image résultante donne la position de la ligne d'eau.

### III.2.4 Validation des résultats

Dans cette section, nous présentons deux manières de valider les résultats des algorithmes de détection de lignes : une manière est basée sur une vérification visuelle des lignes détectées, une autre sur une évaluation statistique.

#### III.2.4.1 Vérification visuelle

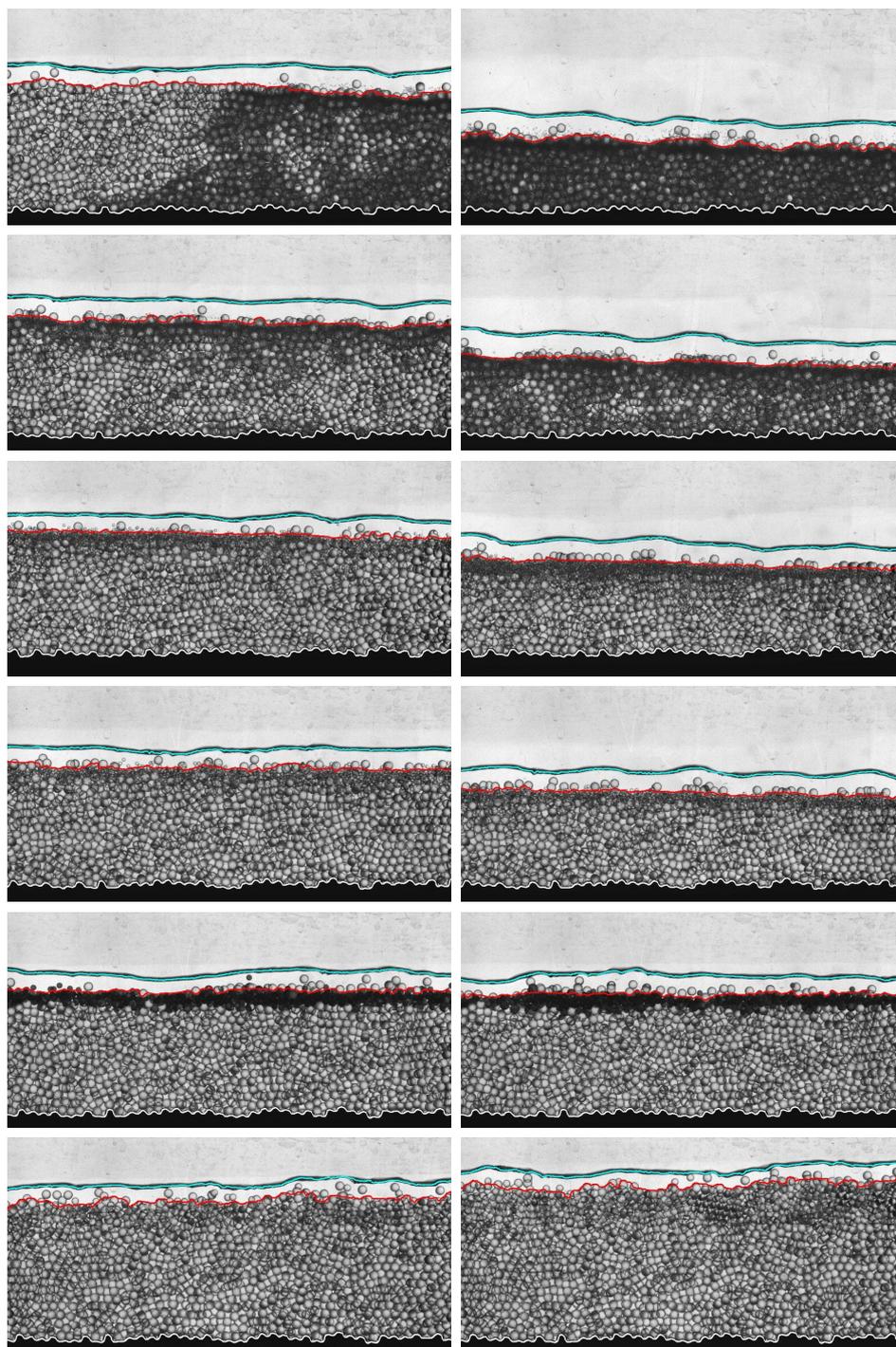
Une première méthode pour valider les résultats est de vérifier les lignes détectées visuellement. Pour cela, nous créons une séquence d'images où les trois lignes détectées (ligne supérieure de la base du canal, ligne du lit et ligne d'eau) sont superposées sur l'image correspondante. Cela nous permet de contrôler la précision de la position des lignes à l'oeil nu. La Figure III.5 montre quelques exemples de détection de lignes où les trois lignes semblent être correctement détectées. Cependant, avec cette méthode de validation, il n'est pas possible d'examiner la totalité de séquence car celle-ci est longue avec plus de 1000 images.

#### III.2.4.2 Évaluation statistique

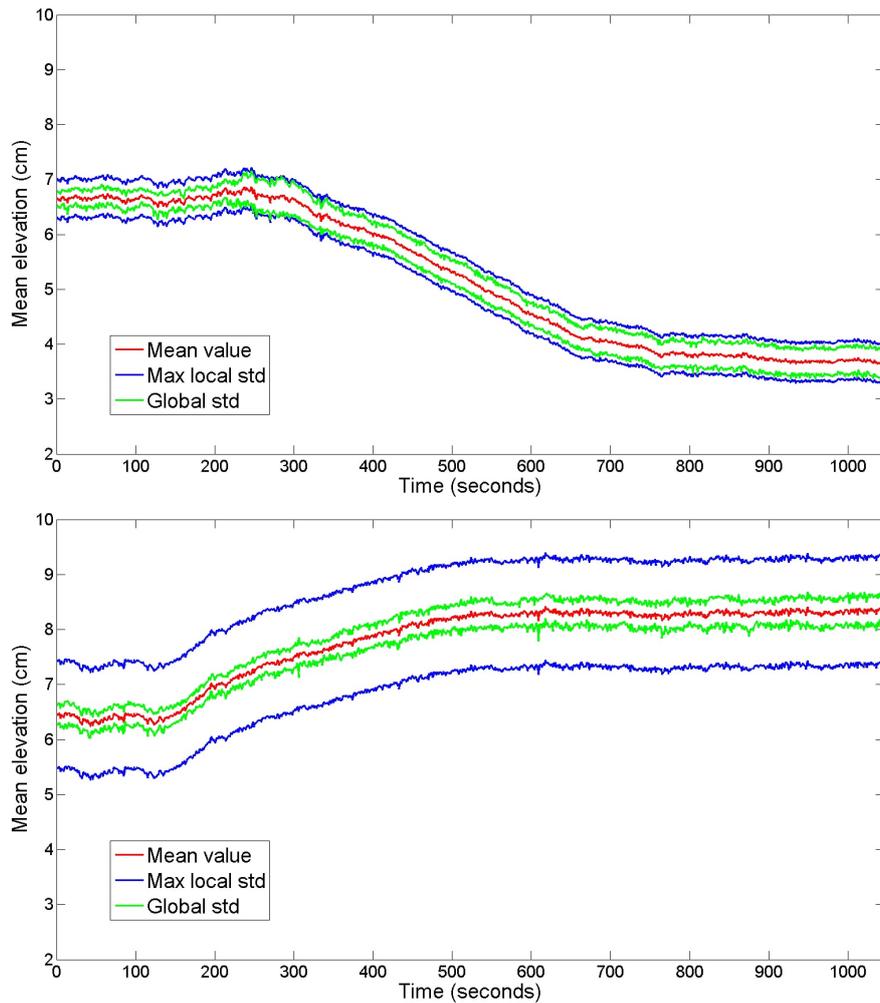
Une seconde méthode pour vérifier l'intégrité des résultats est d'utiliser des statistiques descriptives sur la séquence complète. Ici, nous calculons la hauteur moyenne et l'écart-type de la ligne du lit détectée pour chaque image de la séquence ainsi que l'écart-type local. En regardant la valeur maximale des écarts-types locaux, nous pouvons déterminer si des mauvaises détections se produisent (une valeur élevée correspondrait à un saut dans la ligne et révélerait probablement une erreur). La Figure III.6 montre deux graphiques avec ces valeurs appliquées sur les séquences où des particules fines de 0.7 mm et 4 mm ont été introduites. Nous avons réalisé ce même graphique sur les autres séquences pour conclure qu'il n'y a apparemment pas d'erreurs dans la détection des lignes du lit (pas de pics isolés de valeurs élevées dans les graphiques). Dans le cas 4 mm, nous pouvons constater des valeurs élevées dans les écarts-types locaux, elles sont dues à la nature de la couche de particules fines en surface de lit. Dans ce cas, la surface du lit est plus irrégulière (à cause du faible rapport de taille entre particules fines et grosses) et donc le détecteur est moins précis.

## III.3 Résultats et discussions

Dans cette section, nous présentons les résultats obtenus et les analyses réalisées à partir de notre algorithme de détection de lignes. Pour comprendre pourquoi nous avons concentré notre étude sur la pente du lit et la hauteur d'eau, commençons par considérer un cas réel de transport de sédiments. Dans les montagnes, les torrents adaptent leur lit (pente, profondeur et largeur) pour maintenir un équilibre entre l'eau, les sédiments apportés par l'amont et ceux transportés vers l'aval. Quand une crue torrentielle se produit, une importante quantité de



**Figure III.5** Exemples de détection des lignes de la base du canal (blanc), du lit (red) et de l'eau (cyan) superposées à l'image correspondante à 5 min (colonne à gauche) et 15 min (colonne de droite) pendant l'infiltration de particules fines de 0.7 mm, 0.9 mm, 1.5 mm, 2 mm, 3 mm et 4 mm (respectivement dans cet ordre de haut en bas).



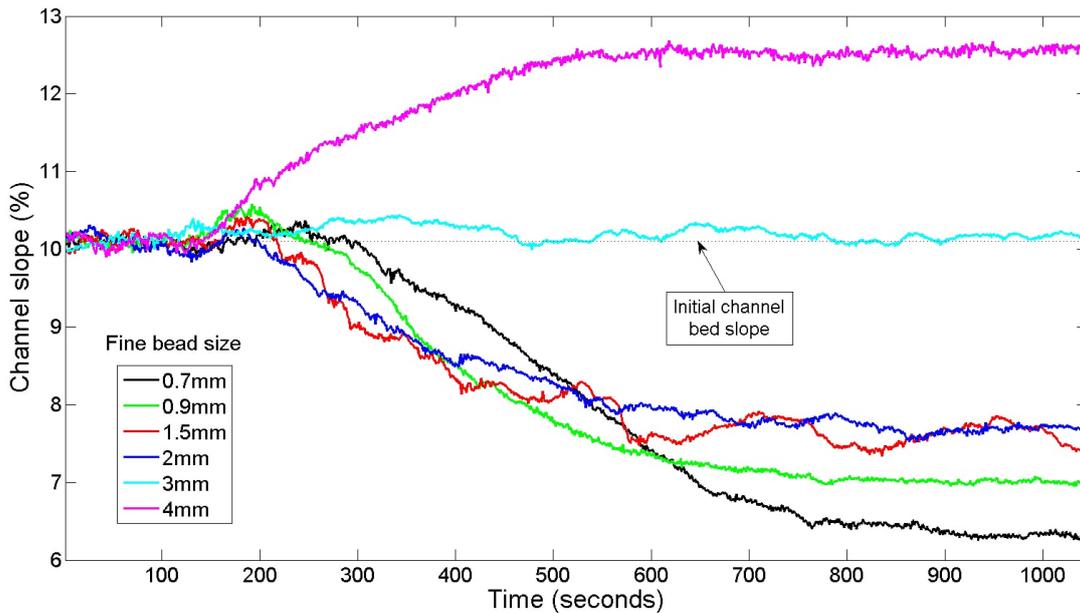
**Figure III.6** Hauteur moyenne (*mean elevation*) de la ligne du lit au cours du temps avec l'écart-type local maximal et l'écart-type global pendant l'infiltration de particules fines de 0.7 mm (haut) et 4 mm (bas).

sédiments est transportée dans les écoulements de montagne ce qui impacte leur stabilité. En fonction de la taille et de la distribution des tailles de ces sédiments, l'impact sur les écoulements peut résulter en des modifications de lit comme la pente, la profondeur et la largeur. Ces modifications sont ce que nous essayons de mieux comprendre. Dans notre étude expérimentale, la largeur ne peut pas varier (fixée par les parois du canal), donc nous nous concentrons uniquement sur l'étude de la pente du lit et de la hauteur d'eau.

### III.3.1 Évolution de la pente du lit

Une des principales raisons qui a motivé le développement de l'algorithme de traitement d'images décrit ci-avant était d'évaluer l'influence de la taille des particules fines sur la mobilité des sédiments dans le lit. Étudier la pente du lit au cours du temps est une manière de le faire. Le calcul de la pente du lit est basé sur les résultats de détection de la ligne du lit. La ligne qui passe par l'extrémité supérieure de la butée en sortie du canal et par la hauteur moyenne du lit en milieu d'image est considérée. La pente de cette ligne droite définit la

pente du lit. Elle est calculée sur chaque image de la séquence afin de pouvoir les comparer. La Figure III.7 montre l'évolution de la pente du lit sur toutes les séquences, chacune utilisant différentes tailles de particules fines. La pente du lit décroît pour les séquences à particules fines de 0.7 mm, 0.9 mm, 1.5 mm et 2 mm, ce qui indique que le lit s'incise (*i.e.* s'érode). Cette incision peut être expliquée par le fait que les particules fines s'infiltrent dans les interstices du lit, ayant pour effet d'accentuer l'exposition des grosses particules à l'écoulement et donc d'augmenter leur mobilité. Inversement, quand les particules fines sont de 4 mm, la pente du lit a tendance à s'exhausser (*i.e.* à augmenter).



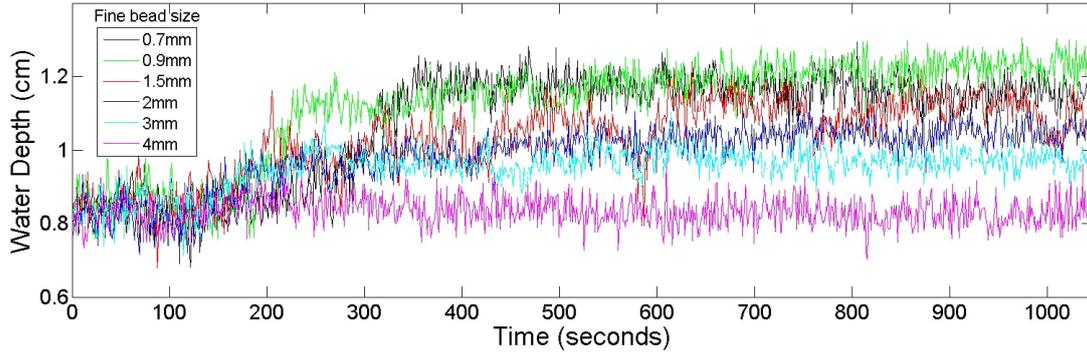
**Figure III.7** Évolution de la pente du lit (*channel slope*) en fonction de la taille des particules fines. La ligne en pointillé représente la pente initiale du lit à l'équilibre, soit 10.1%.

### III.3.2 Évolution de la hauteur d'eau

Un autre objectif de cette étude était d'examiner l'évolution de la hauteur d'eau pendant la période d'introduction de particules fines et voir l'influence de la taille des particules fines. La Figure III.8 montre l'évolution de la hauteur d'eau au cours du temps pour toutes les séquences. Il apparaît que plus les particules sont fines, plus la hauteur d'eau est élevée. Un lit plus fin offre moins de résistance à l'écoulement et devrait entraîner de plus faibles hauteurs. Mais la réduction de pente, d'autant plus marquée avec les particules les plus fines, signifie aussi une augmentation de hauteur d'eau étant donné que le débit liquide est constant. Ce dernier effet s'exprime plus dans les hauteurs d'eau de la figure.

### III.3.3 Éléments de discussion complémentaires

Les résultats de l'algorithme qui sont présentés ci-avant sont préliminaires. Plus de résultats ont été obtenus sur de nouvelles expériences (Dudill et al. [2016b] et Annexe C). L'analyse de ces nouvelles mesures permet d'élaborer un modèle d'évolution de la pente du lit (calculée avec notre algorithme). En effet, pendant l'infiltration de particules fines, nous avons remarqué que l'évolution de la pente du lit suit une courbe de décroissance



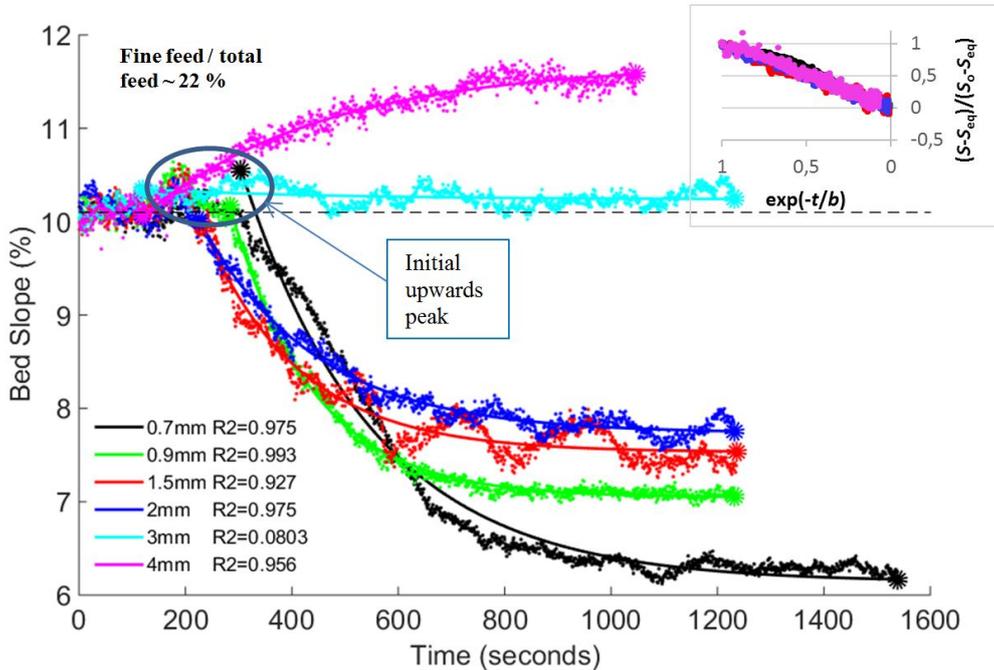
**Figure III.8** Évolution de la hauteur d'eau (*water depth*) en fonction de la taille des particules fines.

exponentielle. Le modèle s'écrit alors sous la forme :

$$S(t) = S_{eq} + (S_0 - S_{eq}).exp(-t/b) \quad (III.1)$$

où  $S$  est la pente du lit,  $S_{eq}$  est la pente finale du lit à l'équilibre bimodal,  $S_0$  est la pente initiale (10.1%),  $b$  est la constante de décroissance et  $t$  est le temps.

La [Figure III.9](#) montre la superposition des courbes de décroissance exponentielle sur les données de pente du lit. Visuellement, nous pouvons constater que les courbes semblent suivre assez fidèlement les évolutions de pente. De plus, les valeurs de coefficients de détermination  $R^2$  sont élevées, ce qui prouve que le modèle décrit bien les données.



**Figure III.9** Évolution de la pente du lit en fonction du temps et de la taille des particules fines. Une courbe de décroissance exponentielle a été ajustée à chaque courbe d'évolution de pente, avec les coefficients de détermination  $R^2$  associés donnés en légende. Graphique tiré de [Dudill et al. \[2016b\]](#).

### III.4 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle approche pour analyser expérimentalement la mobilité du transport de sédiments par charriage avec le développement d'un algorithme de traitement d'images adapté. Cet algorithme est basé sur des opérations morphologiques et des calculs de gradients. Il permet de segmenter les images issues du canal élargi en différentes régions distinctes et de détecter des lignes caractéristiques, comme la ligne d'eau et la ligne du lit. Ces lignes permettent ensuite de calculer la hauteur d'eau et la pente du lit, et ainsi de pouvoir mettre en évidence leur évolution au cours du temps. En faisant varier la taille des particules fines de 0.7 mm à 4 mm, la taille des grosses particules étant fixée à 5 mm, nous avons ainsi pu observer que le rapport de taille entre les particules fines et les plus grosses a une influence importante sur la mobilité du lit.

Cet algorithme est donc un outil très utile pour pouvoir analyser l'évolution globale d'un lit mobile bimodal. Cette méthode a été utilisée sur des expériences issues du canal élargi mais pourrait l'être tout autant sur des expériences issues du canal étroit. Cependant, la limitation du canal élargi est que, contrairement au canal étroit, il ne permet pas d'étudier le phénomène de ségrégation en charriage à l'échelle de la particule (ou bille), ou du moins pas de façon exhaustive (impossibilité de suivre les billes fines). Pouvoir distinguer chaque trajectoire de billes est une manière plus raffinée d'étudier la ségrégation en charriage qui doit donner plus d'informations quant à l'évolution du lit mobile, comme le suivi des positions des billes au cours du temps, leur vitesse et leurs différents régimes de déplacement. Le chapitre suivant traite de ce travail de suivi de billes individuelles au cours du temps.

## Chapitre IV

# Algorithme de suivi déterministe : optimisation et évaluation

### Sommaire

---

<b>IV.1 Algorithme déterministe de suivi de billes</b> . . . . .	<b>52</b>
IV.1.1 Détection des billes . . . . .	52
IV.1.2 Suivi des billes . . . . .	55
IV.1.3 Estimation des états de mouvement . . . . .	56
IV.1.4 Limitations de cet algorithme . . . . .	59
<b>IV.2 Améliorations de la détection</b> . . . . .	<b>60</b>
IV.2.1 Pré-traitement : suppression de la base du canal . . . . .	60
IV.2.2 Nouveau détecteur de billes noires . . . . .	62
IV.2.3 Nouveau détecteur de ligne d'eau . . . . .	63
IV.2.4 Nouveau détecteur de billes transparentes . . . . .	66
IV.2.5 Synthèse des paramètres de la détection . . . . .	68
<b>IV.3 Optimisation du code</b> . . . . .	<b>69</b>
IV.3.1 Réécriture du code en Matlab . . . . .	70
IV.3.2 Restructuration des données . . . . .	71
IV.3.3 Parallélisation du code . . . . .	73
<b>IV.4 Bases de données d'évaluation</b> . . . . .	<b>75</b>
IV.4.1 Élaboration de vérités terrains . . . . .	76
IV.4.2 Choix des métriques d'évaluation . . . . .	77
<b>IV.5 Résultats de l'algorithme déterministe de suivi et discussion</b> . .	<b>78</b>
IV.5.1 Évaluation des performances sur les vérités terrains . . . . .	78
IV.5.2 Quelques résultats sur une longue séquence . . . . .	79
IV.5.3 Limitations de l'algorithme déterministe de suivi . . . . .	80
<b>IV.6 Conclusion</b> . . . . .	<b>81</b>

---

L'algorithme déterministe de suivi de billes est un algorithme de traitement d'images qui permet de calculer et d'étudier la trajectoire des billes dans des images de ségrégation en charriage issues de notre canal (voir Section I.2.2.1). Étant donné qu'un des objectifs de cette thèse est de traiter de très longues séquences d'images, synonymes de phénomènes complets de ségrégation, il est important de pouvoir gérer la quantité importante de données. Cela passe par le développement de méthodes efficaces et adaptées, une bonne gestion de la mémoire de l'ordinateur de calcul et une appropriation adéquate de ses ressources informatiques.

C'est dans une optique d'optimisation de coût sans perte de précision que les méthodes de traitement des versions de l'algorithme développées lors de travaux précédents [Böhm et al., 2006; Hergault et al., 2010] ont été raffinées, améliorées et modifiées. Pour ce faire, nous utilisons des opérations de traitement d'images adaptées comme les outils de la morphologie mathématique. La définition plus détaillée des opérateurs de traitement d'images utilisés est donnée dans l'Annexe A.1. D'autre part, nous avons aussi travaillé sur d'autres optimisations qui se basent sur une réécriture du code en Matlab, une restructuration du stockage des trajectoires et une implémentation d'un traitement parallèle.

Dans ce chapitre, nous présentons cet algorithme déterministe de suivi de billes. Dans un premier temps, nous détaillons l'algorithme séquentiel étape par étape avec d'abord la détection des billes, puis leur suivi entre images consécutives, puis comment leur état de mouvement est défini et enfin les limitations de ces méthodes. Dans un deuxième temps, nous décrivons les améliorations que nous avons développées pour remédier à des problèmes de détection. Dans un troisième temps, nous donnons les différentes optimisations apportées par rapport aux autres versions de l'algorithme développées dans notre laboratoire. Dans un quatrième temps, nous présentons comment évaluer nos algorithmes de suivi de billes. Enfin, nous montrons quelques résultats obtenus et discutons des limitations de l'approche déterministe de cet algorithme.

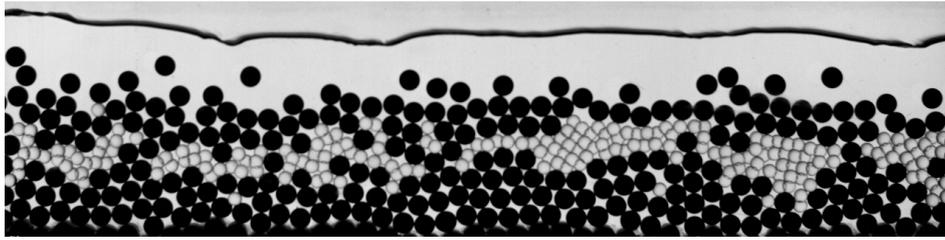
### IV.1 Algorithme déterministe de suivi de billes

L'algorithme déterministe de suivi de billes s'appuie sur une procédure en trois étapes séquentielles : (1) détection des billes dans toutes les images, (2) suivi des billes entre les images, (3) élaboration des états de mouvement des trajectoires en chaque instant. Les états de mouvement sont utilisés par la suite dans l'analyse des séquences. Les principales hypothèses sur lesquelles s'appuie cet algorithme de suivi déterministe sont le faible déplacement des objets, l'absence de recouvrement entre objets (ou très faible recouvrement), la forme simple des objets et leur faible variation d'apparence. Les trois étapes sont détaillées dans cette section.

#### IV.1.1 Détection des billes

La détection des billes est une étape primordiale dans un algorithme déterministe de suivi de billes. En effet, le suivi des billes dans une séquence d'images ne peut être fiable que si l'ensemble des billes retourné par le détecteur est le plus exhaustif possible dans chaque image de la séquence. Le but est donc de détecter les billes de manière efficace avec la plus grande précision possible pour être fidèle à la réalité. Les algorithmes de détection présentés dans cette section sont ceux développés dans les travaux précédents de Böhm et al. [2006] et Hergault et al. [2010]. Ils sont tous issus de méthodes de traitements d'images et se basent plus particulièrement sur des chaînes d'opérations morphologiques. En effet, les opérations morphologiques sont très bien adaptées pour le traitement d'images en niveaux de gris comme les nôtres.

En prenant l'image de la Figure IV.1 comme référence, dans un premier temps, les billes noires sont détectées (apparaissant comme des disques noirs sur l'image). Dans un deuxième temps, la ligne d'eau est localisée (apparaissant comme une ligne noire dans le haut de l'image). Enfin, avec l'aide des détections des billes noires et de la ligne d'eau, les



**Figure IV.1** Image typique d'un mélange de billes avec les éléments à détecter. Les disques noirs sont des billes noires, les anneaux gris foncé sont des billes transparentes et la ligne noire dans le haut de l'image correspond à la ligne d'eau.

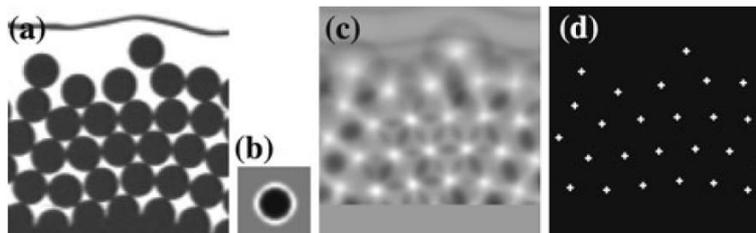
billes transparentes sont détectées (apparaissant comme des anneaux gris foncés sur l'image). Cette suite de détection est appliquée sur chaque image de la séquence.

Ces méthodes de détection présentant des limitations (voir Section IV.1.4), des améliorations ont été faites pendant cette thèse, elles sont données en Section IV.2.

#### IV.1.1.1 Détecteur de billes noires

Les billes noires apparaissent comme des cercles pleins de diamètre  $2r_1 = 6$  mm et de couleur noire. Étant donné que la largeur du canal  $W = 6.5$  mm est à peine plus grande que  $2r_1$ , l'effet de la position transversale (*i.e.* la position dans la 3<sup>ème</sup> dimension qu'on ne peut distinguer sur une image en 2D) des billes noires est quasiment inexistant sur leur rendu visuel dans l'image. Dans ces conditions, elles ont toujours une taille égale à  $2r_1$  (convertie en pixels) dans toutes les configurations possibles. Le détecteur de billes noires est donc très efficace en terme de présence/absence, c'est-à-dire qu'il ne retourne presque jamais, voire jamais, d'erreurs comme les faux positifs et les faux négatifs.

Dans la version de Böhm *et al.* [2006], le détecteur de billes noires se base sur une méthode de *template matching*. Pour ce faire, il utilise un modèle de bille (Figure IV.2b) combiné à une méthode de corrélation de ce modèle dans l'image (Figure IV.2c) et récupère la position des maximums locaux (Figure IV.2d). L'ensemble  $D_{bn}$  des coordonnées des centres des billes est obtenu en détectant les maximums locaux de l'image de corrélation. La sélection des maximums significatifs est faite à l'aide d'un seuil.

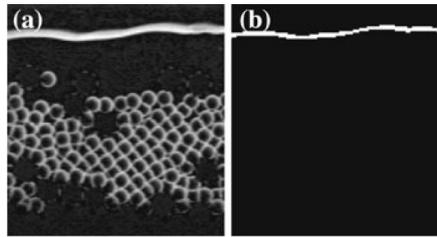


**Figure IV.2** Procédure de détection des billes noires de Böhm *et al.* [2006]. (a) image originale, (b) modèle de bille, (c) résultat de la corrélation, (d) positions des billes noires. Images issues de Böhm *et al.* [2006].

### IV.1.1.2 Détecteur de ligne d'eau

La ligne d'eau apparaît comme une ligne noire dans la partie supérieure de l'image. Il est important de la détecter pour trois raisons : (1) pouvoir supprimer sa contribution sur l'image initiale, (2) ajouter un critère permettant d'éliminer les détections situées au-dessus de la ligne d'eau, (3) étudier l'évolution de la mobilité du lit liée à la ligne d'eau. La première raison est que dans certaines configurations, la ligne d'eau peut entraîner des détections de fausses billes transparentes. Détecter la ligne d'eau et la supprimer des images permet alors d'éviter ces cas problématiques. La deuxième raison est qu'il peut arriver que des fausses billes transparentes soient détectées au-dessus de la ligne d'eau (qui sont des cas impossibles car le centre des billes est toujours dans l'eau). Ces situations peuvent se produire lorsqu'il y a des impuretés sur les parois en verre ou, comme la raison précédente, dans des configurations particulières de la ligne d'eau. Avec la position de la ligne d'eau, il est alors possible de supprimer toutes les éventuelles détections qui sont au-dessus. La troisième raison est pour étudier l'évolution de la mobilité du lit en regardant les variations de la hauteur d'eau et de la pente de la ligne d'eau.

Dans la version de [Hergault et al. \[2010\]](#), le détecteur de ligne d'eau consiste à appliquer une transformation *top-hat* par fermeture de l'image originale pour faire ressortir les éléments qui ont la même épaisseur que la ligne d'eau ([Figure IV.3a](#)). L'image obtenue est ensuite seuillée pour ne garder que la ligne d'eau, puis amincie pour que l'épaisseur de la ligne ne soit que d'un pixel ([Figure IV.3b](#)). Le chaînage des pixels de la ligne détectée permet d'extraire l'ensemble  $D_{le}$  des pixels de la ligne d'eau.



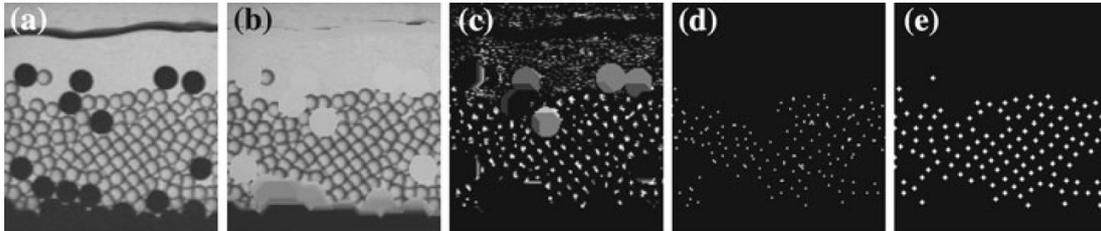
**Figure IV.3** Procédure de détection de la ligne d'eau de [Hergault et al. \[2010\]](#). (a) résultat du *top-hat* appliqué sur l'image originale ([Figure IV.4a](#)), (b) la ligne d'eau obtenue après seuillage et amincissement. Images issues de [Hergault et al. \[2010\]](#).

### IV.1.1.3 Détecteur de billes transparentes

Les billes transparentes apparaissent sur les images comme des anneaux gris foncés légèrement floutés en bordure avec des disques centraux de la même couleur que l'arrière plan de l'image (gris clair). Bien que la taille des billes transparentes de  $2r_2 = 4$  mm soit invariable, les anneaux peuvent présenter des formes irrégulières en fonction de leur position transversale dans le canal et de la configuration du voisinage. En effet, la largeur du canal étant de  $W = 6.5$  mm, des billes transparentes peuvent être occultées par d'autres billes, noires ou transparentes. Ainsi, la détection des billes transparentes ne peut être faite aussi facilement que pour les billes noires.

Dans les travaux de [Hergault et al. \[2010\]](#), le principe du détecteur de billes transparentes repose sur l'utilisation de l'opérateur morphologique *hconvexe* [[Soille, 1999](#)] pour détecter des maximums locaux. L'algorithme remplace d'abord les billes noires et la ligne d'eau par du

fond (Figure IV.4b), puis applique l'opérateur *hconvexe* (Figure IV.4c). Ensuite, il détecte les maximums locaux (Figure IV.4d), et enfin il ne garde que les maximums locaux les plus élevés par seuillage (Figure IV.4e). Une dernière étape de post-traitement est réalisée afin de supprimer les fausses détections qui sont positionnées au-dessus de la ligne d'eau. Au final, l'ensemble  $D_{bt}$  des détections alors restantes donne la position du centre des billes transparentes détections.



**Figure IV.4** Procédure de détection des billes transparentes de Hergault et al. [2010]. (a) image originale, (b) remplacement des billes noires et de la ligne d'eau par du fond, (c) résultat de l'opérateur *hconvexe*, (d) seuillage de l'image *hconvexe*, (e) résultat final après suppression des faux maximums. Images issues de Hergault et al. [2010].

## IV.1.2 Suivi des billes

Une fois les billes noires et transparentes détections, un algorithme de suivi des billes permet d'obtenir les trajectoires des billes. Dans le domaine expérimental de la mécanique des fluides, la PIV et la PTV sont des techniques très populaires. La PTV est plus adaptée à l'étude des trajectoires de particules individuelles, comme c'est le cas pour les billes. La PTV est une méthode déterministe qui utilise le principe de prédiction pour sélectionner les candidats d'une association et une fonction coût pour en choisir le meilleur. Ce principe est un problème d'association de données. Ici, le problème consiste à associer les billes d'une image à l'autre pour ainsi créer des trajectoires. Bien que l'utilisation de billes sphériques de même taille facilite la détection, le calcul des trajectoires est plus difficile du fait qu'elles aient toutes la même forme. Néanmoins, grâce à la haute fréquence d'acquisition des images ( $FPS = 130$  Hz), le déplacement d'une bille entre deux images est généralement plus petit que le diamètre d'une bille noire. Cela est essentiel pour avoir une haute résolution des trajectoires mais surtout pour pouvoir distinguer les billes dans notre algorithme d'association.

La méthode utilisée pour résoudre ce problème d'association a été développée pour être adaptée aux images issues du canal étroit et aux mouvements spécifiques des billes. Nous la décrivons ci-après.

### IV.1.2.1 Problème d'association par un algorithme glouton

La résolution du problème d'association des billes se base sur un algorithme glouton. À cause du grand nombre de détections et de la longueur des séquences à traiter, un algorithme glouton est un bon compromis entre une solution efficace en terme de temps de calcul et suffisante en terme de résultat. En effet, la solution optimale obtenue par l'algorithme hongrois apparaît ici trop coûteuse.

Étant donnés les coûts d'associations  $g(tr, d)$  entre toutes les paires traqueur-détection calculés comme décrit ci-après, l'algorithme glouton sélectionne itérativement le meilleur candidat et supprime les autres associations concurrentes. Pour limiter le nombre de paires traqueur-détection possibles, les détections possiblement associées à un traqueur  $tr$  sont limitées aux détections qui sont localisées dans une région de recherche. Cette région de recherche est centrée sur la position prédite du traqueur  $\hat{x}^{tr}$ . La position prédite se base sur un modèle à vitesse constante, c'est-à-dire qu'elle est calculée en supposant que le traqueur a une vitesse constante. Cet algorithme glouton d'association est également utilisé dans l'algorithme basé sur le filtrage particulaire développé pendant cette thèse (voir Chapitre V).

#### IV.1.2.2 Coût d'association

Le coût d'association  $g(tr, d)$  entre une détection  $d$  et un traqueur  $tr$  consiste en une combinaison de deux facteurs : le premier basé sur la distance de la position de la détection  $x^d$  à la prédiction de la position du traqueur  $\hat{x}^{tr}$ , et l'autre basé sur la vitesse du traqueur à l'instant précédent  $u^{tr}$ . Il est défini par :

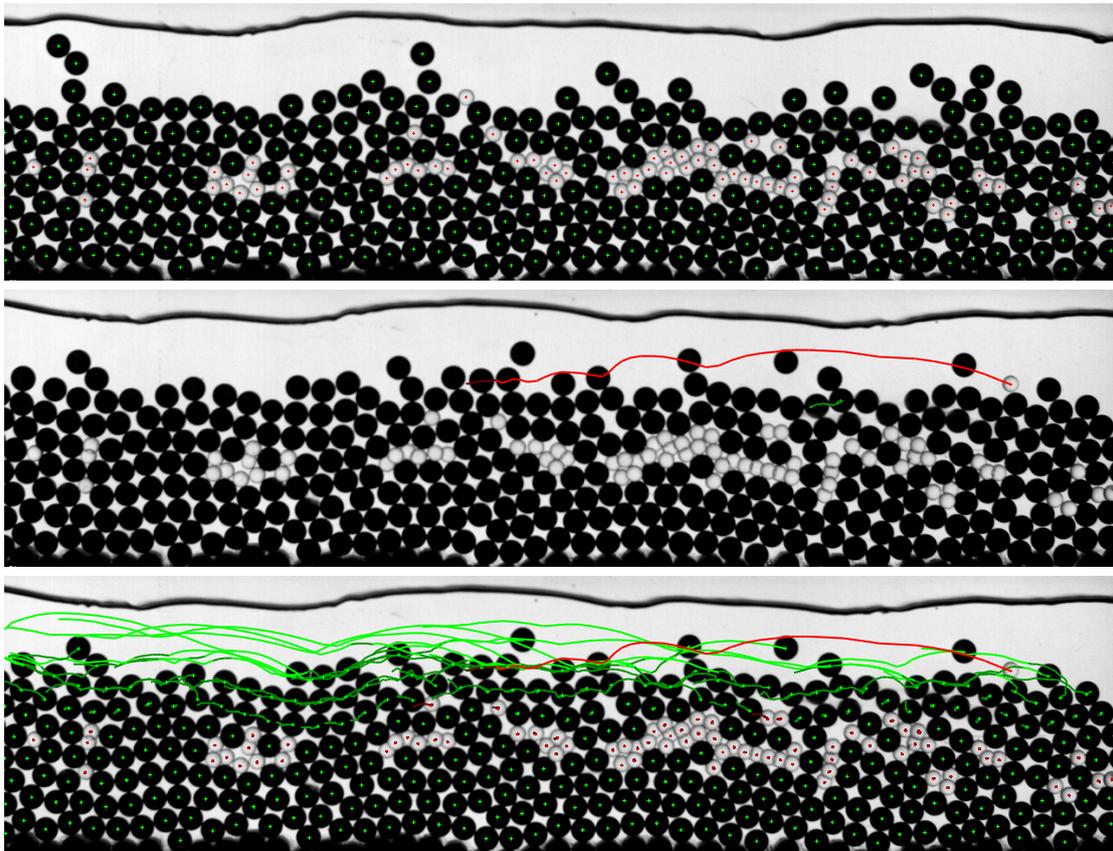
$$g(tr, d) = \alpha \underbrace{\frac{|x^d - \hat{x}^{tr}|}{r_s}}_{\text{terme distance}} + \beta \underbrace{\frac{|u^{tr}|}{|u|_{max}}}_{\text{terme vitesse}} \quad (\text{IV.1})$$

où  $|\cdot|$  est la norme euclidienne d'un vecteur. Le terme de distance permet de favoriser les détections les plus proches de la prédiction. Le terme de vitesse permet de favoriser les faibles vitesses pour mieux traiter les cas de collisions. Lors de l'association des deux premières images de la séquence, le terme de vitesse est fixé à 0 car il n'y a pas encore de valeur de vitesse. La valeur  $|u|_{max}$  est un coefficient de normalisation égal à la plus grande norme de vitesse mesurée à l'instant précédent. Les paramètres  $\alpha$  et  $\beta$  représentent la proportion de chacun des termes. Ils sont fixes pour toutes les séquences et sont réglés respectivement à 0.75 et 0.25 pour donner plus d'importance au terme de distance (le terme de vitesse n'étant utile que dans de rares cas). La valeur  $r_s$  correspond au rayon de la région de recherche, elle est fixe sur toute une séquence. Elle est utilisée comme un coefficient de normalisation et est égale au déplacement maximal d'une bille entre deux images. La fréquence d'acquisition de la caméra a été choisie de sorte que le déplacement maximal d'une bille entre deux images soit égal au diamètre d'une bille noire. Nous avons donc  $r_s = 2r_1$ .

L'application de l'algorithme d'association des billes sur la séquence d'images complète permet d'obtenir les trajectoires de toutes les billes observées. La [Figure IV.5](#) montre des exemples de trajectoires de billes sur une séquence de 50 images.

#### IV.1.3 Estimation des états de mouvement

La dernière étape de l'algorithme de suivi déterministe consiste à estimer un état de mouvement pour chacune des trajectoires en chaque instant. Cette étape ne peut être faite qu'une fois le suivi complet des billes réalisé car des informations sur les conditions futures de la trajectoire d'une bille sont nécessaires pour estimer son état de mouvement courant. Ces états sont ensuite utilisés pour analyser les séquences, les mouvements et les vitesses des billes. L'algorithme d'estimation se base sur les idées initiales de [Böhm et al. \[2006\]](#) et [Hergault et al. \[2010\]](#) avec quelques modifications et optimisations pour limiter son temps d'exécution. Les optimisations sont traitées plus en détails dans la [Section IV.3](#).



**Figure IV.5** Visualisation de trajectoires de billes sur une séquence de 50 images. La première image (en haut) montre les résultats de la détection des billes noires (croix vertes) et transparentes (croix rouges) sur la première image de la séquence. La deuxième image (au milieu) montre les trajectoires de deux billes, une bille noire (courte trajectoire verte) et une bille transparente (longue trajectoire rouge), sur toute la séquence de 50 images. La troisième image (en bas) montre les trajectoires de toutes les billes sur les 50 images. Les états de mouvement des billes (voir Section IV.1.3) sont également représentés ici : les billes noires (resp. transparentes) en saltation apparaissent en vert clair (resp. rouge clair), celles en roulement apparaissent en vert foncé (resp. rouge foncé).

Comme énoncé dans la Section I.1.1, lors d'un transport solide par charriage, il existe deux états de mouvement pour les particules : le roulement et la saltation. Dans nos expériences, un grand nombre de billes se déplacent très lentement dans le canal. Cet effet de lenteur est d'autant plus marqué quand les billes sont positionnées profondément dans le lit. En effet, les vitesses dans le haut du lit sont de l'ordre de  $10^{-1}$  m/s alors qu'en bas du lit elles ne sont guère plus élevées que  $10^{-5}$  m/s. En réalité, nous avons constaté que ces billes ne bougent que par à-coups, elles n'ont pas une vitesse continue. Elles sont donc à l'arrêt pendant un certain temps, ou autrement dit elles sont en "repos". En chaque instant d'une trajectoire, une bille peut alors être dans un des trois états de mouvement : repos, roulement ou saltation.

### IV.1.3.1 Définition des états de mouvement

#### Billes en repos

Les billes en repos (*resting beads*) forment le lit, elles sont en contact continu avec leurs voisins. Elles ne sont pas censées bouger, ou plus précisément, leur possible vitesse de déplacement (en prenant aussi en compte les vitesses fluctuantes) ne dépasse pas un seuil de vitesse  $u_{rest}$  :  $|\mathbf{u}_b| \leq u_{rest}$ , où  $|\mathbf{u}_b|$  est la vitesse moyenne de la bille sur  $n_i$  images consécutives, soit  $n_i \Delta_T$ .

#### Billes en roulement

Les billes en roulement (*rolling beads*) sont positionnées au-dessus des billes en repos. Elles restent toujours en contact avec le lit et se déplacent avec une certaine vitesse. L'algorithme utilise deux critères pour distinguer une bille en roulement : la bille doit être en mouvement mais ne pas avoir une vitesse trop rapide, et elle doit avoir au moins un voisin. Le premier critère correspond à vérifier la condition  $u_{rest} < |\mathbf{u}_b| \leq u_{salt}$ , où  $u_{salt}$  est la vitesse à ne pas dépasser (la dépasser serait mécaniquement impossible pour une bille considérée en roulement). Pour que le deuxième critère soit validé, il doit vérifier la condition  $a_n/a^* \leq \varepsilon$ , où  $a_n$  représente la distance au plus proche voisin,  $a^*$  représente la distance de référence entre les deux voisins concernés et  $\varepsilon$  est la valeur seuil.  $a^*$  est fixée à la plus petite distance possible entre deux billes et dépend donc du diamètre des billes.  $a^*$  peut alors prendre trois valeurs différentes en fonction du type des billes voisines en question :  $a^* = 2r_1$  mm entre deux billes noires,  $a^* = r_1 + r_2$  mm entre une bille noire et une transparente, et  $a^* = 2r_2$  mm entre deux billes transparentes.

#### Billes en saltation

Les billes en saltation (*saltating beads*) font des rebonds sur le lit et ont généralement une vitesse assez élevée. Elles n'ont pas de voisin hormis quand elles entrent en collision avec les autres billes. Pour distinguer une bille en saltation, l'algorithme vérifie d'abord la condition  $|\mathbf{u}_b| > u_{salt}$ . Si la bille a une vitesse rapide, alors elle est en saltation. Si cette condition n'est pas vérifiée, alors l'algorithme vérifie les deux conditions :  $|\mathbf{u}_b| > u_{rest}$  et  $a_n/a^* > \varepsilon$ .

### IV.1.3.2 Algorithme d'estimation des états de mouvement

L'algorithme d'estimation des états de mouvement se déroule en deux étapes. La première étape consiste, pour chaque image, à affecter les états de mouvement dit "locaux" aux billes en appliquant les critères définis ci-avant. Une fois les états locaux estimés sur chaque image, la deuxième étape consiste à appliquer un filtrage par moyenne glissante sur les états d'une même trajectoire. La fenêtre glissante est d'une longueur de  $n_i$  pas de temps. Cette étape permet d'avoir une meilleure continuité dans les états et d'éviter des cas d'états isolés pouvant être causés par des erreurs d'estimation locales.

### IV.1.3.3 Synthèse des paramètres de l'estimation des états de mouvement

Pour quasiment toutes nos expériences, la fréquence d'acquisition des images est de  $FPS = 130$  Hz, soit un écart temporel entre deux images de  $\Delta_T = 8$  ms. Le nombre d'images consécutives  $n_i$  à étudier pour mesurer la vitesse moyenne d'une bille et appliquer le filtrage

par moyenne glissante sur les états (deuxième étape de l'algorithme d'estimation ds états de mouvement) est fixé à  $n_i = 5$ , ce qui correspond à une période de  $5\Delta_T = 40$  ms. Cette valeur représente un bon compromis pour une analyse lissée de l'état de la bille sans trop prendre en compte les éventuelles imprécisions locales de la détection et les erreurs d'estimation d'état.

Comme énoncé dans la Section I.2.2.5, les seuils de vitesse  $u_{rest}$  et  $u_{salt}$  dépendent de la vitesse maximale  $u_{max}$  qu'une bille peut atteindre dans l'écoulement. Or nous savons que  $u_{max} = 2r_1 FPS = 0.6$  m/s, avec classiquement  $2r_1 = 6$  mm. Nous pouvons alors en déduire les valeurs des seuils de vitesse suivantes :  $u_{rest} = 0.01u_{max} = 0.006$  m/s et  $u_{salt} = 0.4u_{max} = 0.24$  m/s. Les trois valeurs possibles de  $a^*$  dépendent des rayons des billes. En général, dans nos expériences, les billes noires ont un rayon de  $r_1 = 3$  mm et les billes transparentes un rayon de  $r_2 = 2$  mm. Les valeurs de  $a^*$  sont donc :  $a^* = 2r_1 = 6$  mm entre deux billes noires,  $a^* = r_1 + r_2 = 5$  mm entre une bille noire et une transparente, et  $a^* = 2r_2 = 4$  mm entre deux billes transparentes. La valeur du paramètre de seuil  $\varepsilon$  a été ajustée expérimentalement pour minimiser les différences entre les états (roulement et saltation) déterminés par l'algorithme et les états déterminés à l'oeil nu. Un bon compromis a été trouvé pour  $\varepsilon = 1.07$ .

#### IV.1.4 Limitations de cet algorithme

Une première limitation de cet algorithme déterministe de suivi de billes est qu'il ne prend pas en compte la base du canal. La base du canal, apparaissant de la même couleur que les billes noires, peut fausser la détection des billes noires. En effet, elle peut engendrer des erreurs comme des fausses détections (une partie de la base détectée comme étant une bille noire) ou des détections manquantes (une bille noire non-détectée car prise comme faisant partie de la base). Une étape de pré-traitement pour supprimer la base permettrait de gagner en robustesse de détection.

Une deuxième limitation intervient avec le détecteur de billes noires concernant son temps d'exécution. La solution utilisée est en effet assez coûteuse car elle nécessite de balayer chaque pixel de l'image avec un modèle de bille. Dans le but de traiter de très longues séquences d'images, la moindre étape coûteuse peut rendre le temps de calcul de tout l'algorithme de suivi irraisonnable. Pour y remédier, une méthode évitant l'utilisation d'un modèle de bille permettrait de gagner un temps précieux sans perdre en précision.

Une troisième limitation concerne le détecteur de ligne d'eau. Celui-ci renvoie régulièrement une ligne incohérente lorsque le contour des billes transparentes présente une intensité forte après l'opération de *top-hat*. Utiliser un simple seuillage sur l'image *top-hat* ne retourne pas toujours une ligne continue. Une méthode de segmentation plus adaptée à la détection de ligne serait alors mieux appropriée pour obtenir une ligne d'eau précise et cohérente.

Une dernière limitation concerne la robustesse du détecteur de billes transparentes. En effet, il retourne très fréquemment des faux positifs et faux négatifs, ces erreurs étant commises à cause de la configuration des billes (une légère occlusion de billes peut entraîner un faux négatif, ou un interstice entre des billes peut être pris pour une bille et donc entraîner un faux positif). Utiliser une corrélation adaptée pourrait apporter plus de précision et de fiabilité dans les détections.

Les limitations énoncées ici concernent exclusivement la partie détection de l'algorithme de suivi. Ces limitations sont traitées dans la section suivante. D'autres améliorations et optimisations sont données dans la Section IV.3.

## IV.2 Améliorations de la détection

Face aux limitations des algorithmes de détection décrits ci-avant, des nouveaux détecteurs ont été développés. Ceux-ci permettent de gagner en précision, en rapidité d'exécution, et en robustesse grâce à des méthodes de traitement d'images mieux adaptées. Ils sont appliqués l'un à la suite de l'autre, dans l'ordre ici énoncé : (1) pré-traitement, (2) détection des billes noires, (3) détection de la ligne d'eau, (4) détection des billes transparentes. Ils se basent sur des outils issus de la *Image processing toolbox* de Matlab. Les fonctions Matlab utilisées sont données pour les opérations clés.

### IV.2.1 Pré-traitement : suppression de la base du canal

Pour faciliter les futures étapes de détection, nous avons ajouté une étape de pré-traitement pour chaque image. Le but est d'éliminer la base du canal pouvant générer des erreurs lors des détections de billes. Cette étape ne peut pas s'appuyer sur la détection de la base du canal décrite en Section III.2.3.1 car la base est noire et donc de la même couleur que les billes noires.

#### Création du masque binaire de la base du canal

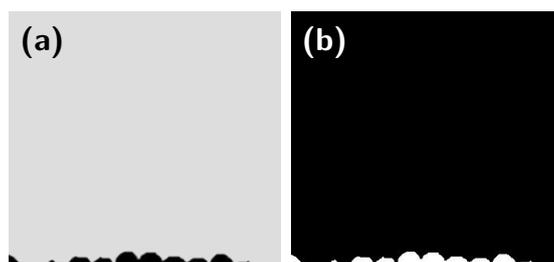
La base du canal apparaît en noir sur chaque image et est donc de la même couleur et forme que les billes noires (voir l'explication en Section I.2.2.1 pour comprendre ce qu'est la base du canal). Une image  $f_{\emptyset}$  (Figure IV.6a) du canal vide est prise avant le début de l'expérience.  $f_{\emptyset}$  présente donc seulement la base du canal dans l'image. Pour éliminer la base d'une image, il est nécessaire de récupérer la position des pixels qui en font partie. Le masque binaire  $g_{bc}$  de la base du canal correspond à l'ensemble des pixels apparaissant noirs dans  $f_{\emptyset}$ .  $g_{bc}$  est obtenue avec une opération de seuillage  $T$  par un seuil  $\tau_n$ . Ce seuil  $\tau_n$  peut varier selon la séquence d'images. En effet, les conditions d'acquisition peuvent être légèrement différentes d'une séquence à l'autre (par exemple la luminosité, la propreté des parois en verre, etc.) voire même enregistrées avec des caméras différentes (anciennes et futures caméras), ce qui peut modifier l'intensité des pixels des images. Nous cherchons alors à estimer automatiquement  $\tau_n$  en début de chaque séquence. Pour ce faire, l'histogramme de l'intensité des pixels de  $f_{\emptyset}$  est étudié. La valeur du premier maximum local de cet histogramme correspond à la valeur d'intensité des pixels noirs. On donne alors à  $\tau_n$  le double de cette valeur d'intensité pour être sûr d'avoir tous les pixels de la base. Cette estimation de  $\tau_n$  est donc faite une seule fois au début de la séquence à traiter, puis elle est constante pour tous les pré-traitements d'une même séquence ( $\tau_n$  est d'ailleurs aussi utilisé comme seuil pour la détection des billes noires). Le masque binaire  $g_{bc}$  (Figure IV.6b) est alors défini comme :

$$g_{bc} = T_{\tau_n}(f_{\emptyset}) \quad (\text{IV.2})$$

Les paramètres d'entrée et de sortie pour la création du masque binaire de la base du canal sont donnés dans le tableau suivant :

Entrées	Sorties
$r_1, f_{\emptyset}$	$\tau_n, g_{bc}$

**Tableau IV.1** Paramètres d'entrée/sortie de la création du masque de la base du canal.



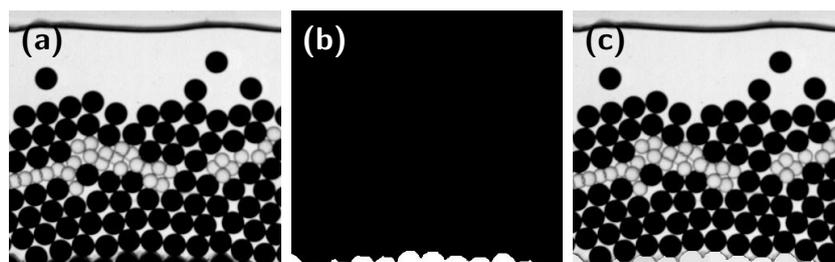
**Figure IV.6** Procédure de création du masque binaire de la base du canal. (a) image du canal vide, (b) résultat du seuillage de l'image montrant le masque binaire avec la position de la base du canal.

### Algorithme de pré-traitement

A partir de  $g_{bc}$ , il est alors maintenant possible de supprimer la base du canal dans chaque image. Pour ce faire, pour une image  $f$ , la procédure consiste à éliminer la base par reconstruction de l'arrière plan (ou fond) aux pixels non nuls de  $g_{bc}$ , et de garder les valeurs d'intensité de  $f$  aux pixels nuls de  $g_{bc}$  (équivalent aux pixels non nuls de son complémentaire  $\overline{g_{bc}}$ ). L'image pré-traitée sans base  $f_p$  (Figure IV.7c) est définie par :

$$f_p = g_{bc} \cdot \phi_{B_{r_1 \cdot (3/2)}}(f) + \overline{g_{bc}} \cdot f \quad (\text{IV.3})$$

où  $\phi_{B_{r_1}}$  est l'opérateur de fermeture et  $B_{r_1 \cdot (3/2)}$  son élément structurant (un disque ayant pour rayon 1.5 fois celui d'une bille noire, soit  $r_1 \cdot (3/2)$ , de manière à bien supprimer toutes les billes noires de l'image). Ici, la fermeture de  $f$  par  $B_{r_1 \cdot (3/2)}$ , notée  $\phi_{B_{r_1 \cdot (3/2)}}(f)$  (fonction `imclose` dans Matlab), a pour effet de remplacer tous les pixels de  $f$  par du fond reconstruit. Ainsi, dans  $f_p$ , seule la base du canal est remplacée par du fond, le reste de l'image demeure identique à  $f$ .



**Figure IV.7** Procédure de pré-traitement d'une image. (a) image originale, (b) masque binaire de la base du canal, (c) résultat de la suppression de la base du canal dans l'image originale.

Les paramètres d'entrée et de sortie de cet algorithme de pré-traitement sont donnés dans le tableau suivant :

Entrées	Sorties
$r_1, g_{bc}, f$	$f_p$

**Tableau IV.2** Paramètres d'entrée/sortie du pré-traitement d'une image  $f$ .

### IV.2.2 Nouveau détecteur de billes noires

Face à la complexité importante du détecteur de billes noires et non souhaitée pour un traitement de longues séquences d'images, un nouveau détecteur de billes noires a été développé. La nouvelle méthode se base sur une extraction de composantes connexes.

#### Algorithme

La base du canal étant supprimée par le pré-traitement décrit ci-avant, seules les billes noires et la ligne d'eau sont de couleur noire dans les images pré-traitées. Il est alors possible d'isoler ces éléments noirs avec une opération de seuillage  $T$  par  $\tau_n$ . Pour une image  $f_p$  issue du pré-traitement d'une image  $f$  et avec le seuil  $\tau_n$  fixé, le nouveau détecteur de billes noires, illustré en Figure IV.8, fonctionne de la façon suivante :

1. Appliquer un seuillage de  $f_p$  par  $\tau_n$ . L'image  $g_n$  (Figure IV.8b), qui est alors un masque binaire des pixels noirs de  $f_p$ , est obtenue par :

$$g_n = T_{\tau_n}(f_p) \quad (IV.4)$$

2. Appliquer une érosion de  $g_n$  par un disque  $B_{r_1.(2/3)}$  (disque ayant pour rayon  $2/3$  du rayon  $r_1$  d'une bille noire). Cette érosion, notée  $\varepsilon_{B_{r_1.(2/3)}}(g_n)$  (fonction `imerode` dans Matlab), a pour but de détacher les billes les unes des autres. Elle permet de supprimer la ligne d'eau et de faire apparaître les billes noires comme des petites tâches distinctes. L'image érodée, un masque binaire des tâches noté  $g_{tn}$  (Figure IV.8c), est définie par :

$$g_{tn} = \varepsilon_{B_{r_1.(2/3)}}(g_n) \quad (IV.5)$$

3. Détecter automatiquement les tâches dans  $g_{tn}$  par une méthode d'extraction de composantes connexes (*connected component labelling* dans Soille [1999] ou encore *region labelling*). Cette méthode, notée RP (fonction `regionprops` dans Matlab), est basée sur la notion de connectivité, elle permet d'identifier chacune des régions connectées d'une image. Le but est d'affecter un label à chaque région connectée et de retourner la liste des labels avec les propriétés associées. Les propriétés concernent l'aire et le centre de gravité des régions par exemple. Le centre de gravité (*centroid*) de ces régions connectées constitue alors le centre des billes noires détectées  $D_{bn}$  (Figure IV.8d). Ils sont obtenus par :

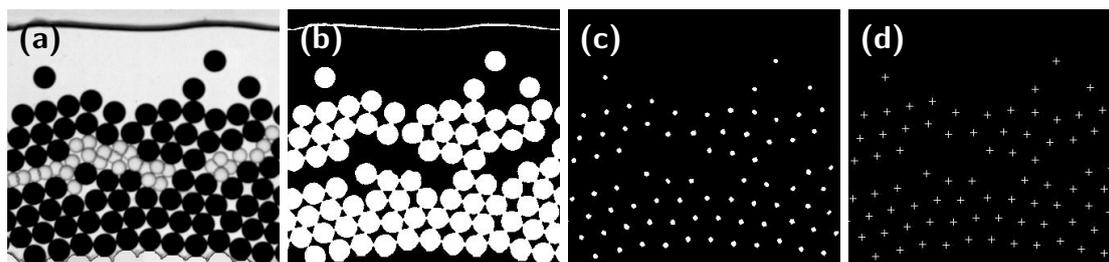
$$D_{bn} = RP(g_{tn}, \text{"centroids"}) \quad (IV.6)$$

La sortie du détecteur de billes noires  $D_{bn}$  est un ensemble de coordonnées  $(x, y)$  donnant la position du centre des billes noires détectées. Cette nouvelle version du détecteur de billes noires est aussi performante en terme de résultat que la précédente version mais s'exécute beaucoup plus rapidement (voir ci-après).

Les paramètres d'entrée et de sortie du détecteur de billes noires sont donnés dans le tableau suivant :

Entrées	Sorties
$r_1, \tau_n, f_p$	$D_{bn}$

**Tableau IV.3** Paramètres d'entrée/sortie du détecteur de billes noires appliqué sur l'image  $f_p$  issue du pré-traitement de l'image  $f$ .



**Figure IV.8** Procédure de détection des billes noires. (a) image pré-traitée, (b) seuillage de l'image pré-traitée, (c) érosion de l'image seuillée pour faire apparaître des tâches distinctes, (d) résultat final de la détection des centres des tâches par extraction de composantes connexes. Ces centres correspondent aux centres des billes noires.

### Analyse des performances du nouveau détecteur de billes noires

Comme détaillé ci-avant, le détecteur de billes noires a été complètement modifié par rapport à la version de Böhm et al. [2006] afin d'éviter l'application d'une coûteuse opération de corrélation (calcul de corrélation avec un modèle en chaque pixel l'image). Avec l'utilisation d'une méthode d'extraction de composantes connexes, le temps d'exécution du détecteur de billes noires a pu être réduit considérablement. La Figure IV.9 montre une comparaison des temps d'exécution<sup>1</sup> des deux détecteurs (ancien et nouveau) en fonction du nombre de billes dans l'image. On constate que les deux détecteurs ne dépendent que peu voire pas du nombre de billes dans l'image (très faibles pentes des deux droites de régression linéaire), ce qui est d'autant plus vrai pour le nouveau détecteur. De plus, le nouveau détecteur est beaucoup plus rapide que le détecteur de Böhm et al. [2006]. Il traite 100 images en environ 2.8 sec de moyenne contre environ 9.7 sec de moyenne pour l'autre version, soit un gain de temps de plus de 70% sur la détection des billes noires.

### IV.2.3 Nouveau détecteur de ligne d'eau

Pour pallier les problèmes de lignes incohérentes retournées par le détecteur de ligne d'eau, nous avons développé un nouveau détecteur. La nouvelle méthode se base une segmentation par ligne de partage des eaux contrôlée par des marqueurs.

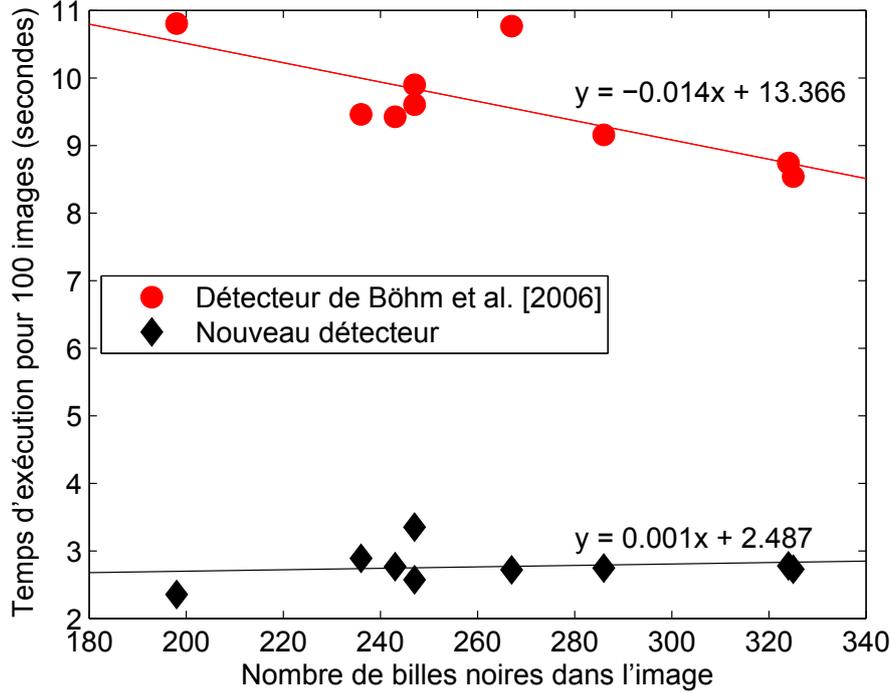
#### Algorithme

Pour une image  $f_p$  issue du pré-traitement de  $f$  et avec l'ensemble des billes noires détectées  $D_{bn}$ , le nouveau détecteur de ligne d'eau, illustré en Figure IV.10, procède de la manière suivante :

1. Créer un masque binaire de billes noires à partir de  $D_{bn}$ . Pour ce faire, sur une image vide remplie de 0, mettre un 1 à chaque position de  $D_{bn}$ , cette image est un masque binaire noté  $g_{bnp}$ . Appliquer ensuite une dilatation par un disque  $B_{r_1}$ , notée  $\delta_{B_{r_1}}(g_{bnp})$  (fonction `imdilate` dans Matlab), pour créer le masque binaire des billes noires (Figure IV.10b), noté  $g_{bn}$  et défini par :

$$g_{bn} = \delta_{B_{r_1}}(g_{bnp}) \quad (\text{IV.7})$$

1. Calculés sur un ordinateur avec un Intel(R) Xeon(R) 2.40 GHz et une mémoire RAM de 32 GB.



**Figure IV.9** Temps d'exécution des détecteurs de billes noires en fonction du nombre de billes dans l'image. Comparaison du détecteur de Böhm et al. [2006] (en rouge) et du nouveau détecteur développé dans cette thèse (en noir). Le temps d'exécution est calculé sur un traitement de 100 images. Les droites de régression linéaire respectives sont ajoutées avec les équations associées.

- Supprimer les billes noires dans  $f_p$  par reconstruction du fond, *i.e.* remplacer les billes noires par les valeurs de fermeture de  $f_p$  par  $B_{r_1}$ . Ceci permet de ne laisser noir que la ligne d'eau. La nouvelle image  $f_{le}$  (Figure IV.10c) sans billes noires et n'ayant que la ligne d'eau en noir, est obtenue par :

$$f_{le} = g_{bn} \cdot \phi_{B_{r_1}}(f_p) + \overline{g_{bn}} \cdot f_p \quad (\text{IV.8})$$

- Amplifier la ligne d'eau par transformation *top-hat* par fermeture, notée  $BTH(f)$  (fonction `imbothat` dans Matlab), *i.e.* en soustrayant  $f_{le}$  de sa fermeture. La fermeture a pour élément structurant  $L_{e,(3/2)}$ , une ligne verticale d'une longueur égale à  $3/2$  de l'épaisseur  $e$  de la ligne d'eau, ce qui a pour effet de la supprimer sur toute son épaisseur. Par cette opération *top-hat*, la nouvelle image  $f_{leth}$  (Figure IV.10d) présente une ligne d'eau fortement contrastée avec le reste de l'image. Elle est définie par :

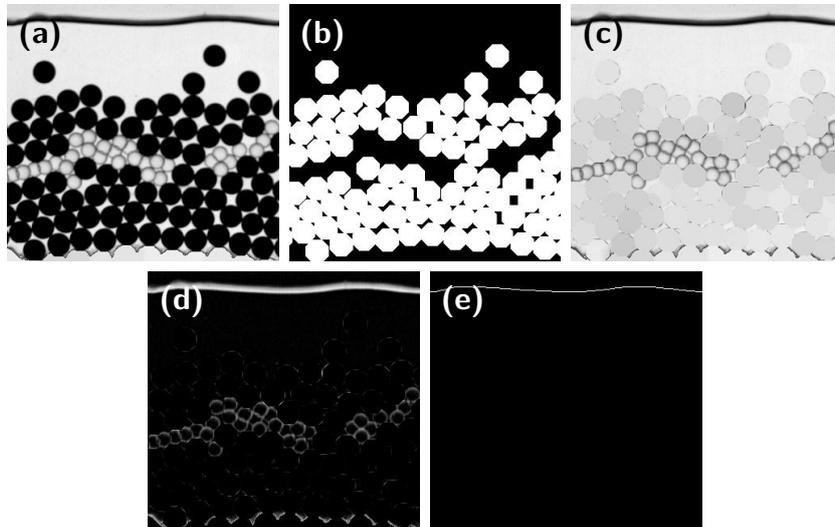
$$f_{leth} = BTH(f_{le}) = \phi_{L_{e,(3/2)}}(f_{le}) - f_{le} \quad (\text{IV.9})$$

- Appliquer une segmentation par ligne de partage des eaux (*watershed transform* dans [Beucher and Lantuejoul, 1979]) sur  $f_{leth}$  pour détecter la ligne d'eau. Cette opération, notée WS (fonction `watershed` dans Matlab), consiste à considérer l'image comme un relief topographique et à simuler une inondation pour faire ressortir les digues, aussi appelées lignes de partage des eaux. Dans le monde image, cela revient à trouver

les lignes de pixels à hautes intensités. Dans notre cas, on utilise une segmentation par ligne de partage des eaux avec des marqueurs sur la première et dernière ligne de l'image. Pour combiner les marqueurs à l'image, nous utilisons une imposition de minimums sur  $f_{leth}$  aux positions des marqueurs. Cette reconstruction de l'image, notée  $R_{(f+1)\wedge g_m}^\varepsilon(g_m)$  (fonction `imimposemin` dans Matlab), a pour objectif d'imposer aux pixels des marqueurs (répertoriés dans un masque binaire  $g_m$ ) d'avoir une valeur d'intensité égale à la plus faible valeur d'intensité de l'image. Ces marqueurs sont donc le départ de la simulation d'inondation. Comme dans l'image  $f_{leth}$  la ligne d'eau apparaît avec des valeurs d'intensité fortes, la segmentation par inondation détecte deux régions dans l'image (une en haut et l'autre en bas) et la frontière entre les deux se trouve exactement sur cette ligne d'eau pour départager les deux zones inondées. Ainsi, sur l'image obtenue par la segmentation  $f_{lews}$  (Figure IV.10e), la ligne d'eau est détectée et les pixels nuls constituent la position de la ligne d'eau  $D_{le}$  (les pixels égaux à la valeur 1 correspondent à une région inondée et ceux égaux à la valeur 2 à l'autre région, ceux égaux à 0 correspondent à la frontière entre les deux, soit la ligne détectée). Soit, pour  $g_m$  le masque binaire des marqueurs et  $\mathbf{x}$  un pixel de coordonnées  $(x, y)$  :

$$f_{lews} = \text{WS}\left(R_{(f_{leth}+1)\wedge g_m}^\varepsilon(g_m)\right) \quad (\text{IV.10})$$

$$D_{le} = \{\mathbf{x} \mid f_{lews}(\mathbf{x}) = 0\}$$



**Figure IV.10** Procédure de détection de la ligne d'eau. (a) image pré-traitée, (b) masque binaire des billes noires créé à partir d'une dilatation aux positions de leur centre, (c) remplacement des billes noires par du fond, (d) ligne d'eau amplifiée grâce à une transformation *top-hat* par fermeture, (e) résultat final de la détection de la ligne d'eau grâce à une segmentation par ligne de partage des eaux.

La sortie du détecteur de ligne d'eau  $D_{le}$  est un ensemble de coordonnées  $(x, y)$  donnant la position des pixels qui sont sur la ligne d'eau. Par rapport à la méthode précédente, l'opération de segmentation par ligne de partage des eaux semble plus adaptée et intuitive pour détecter une telle ligne caractéristique.

Les paramètres d'entrée et de sortie du détecteur de ligne d'eau sont donnés dans le tableau suivant :

Entrées	Sorties
$r_1, e, D_{bn}, f_p$	$D_{le}$

**Tableau IV.4** Paramètres d'entrée/sortie du détecteur de ligne d'eau appliqué sur l'image  $f_p$  issue du pré-traitement de l'image  $f$ .

#### IV.2.4 Nouveau détecteur de billes transparentes

Le détecteur de billes transparentes retournant trop souvent des erreurs du type faux positifs et faux négatifs, nous avons mis au point un nouveau détecteur plus fiable et robuste aux configurations complexes des billes. Son concept principal est d'utiliser une corrélation par un modèle de bille après avoir appliqué une transformation *hconcave*.

##### Algorithme

Pour une image  $f_p$  issue du pré-traitement de  $f$  et avec les résultats de la détection des billes noires  $D_{bn}$  et de la ligne d'eau  $D_{le}$ , le nouveau détecteur de billes transparentes, illustré en [Figure IV.11](#), procède de la manière suivante :

1. Créer un masque binaire de billes noires et de ligne d'eau à partir de  $D_{bn}$  et  $D_{le}$ . Placer à chaque position de  $D_{bn}$  la valeur 1 dans une image vide, ce qui donne le masque binaire  $g_{bnp}$ . Et, de la même façon, placer à chaque position de  $D_{le}$  la valeur 1 dans une image vide, ce qui donne le masque binaire  $g_{lep}$ . Appliquer ensuite une dilatation de  $g_{bnp}$  par  $B_{r_1}$  et une dilatation de  $g_{lep}$  par  $L_{e.(3/2)}$ . En sommant ces deux résultats, le masque binaire de billes noires et de ligne d'eau  $g_{bnle}$  ([Figure IV.11b](#)) est obtenu :

$$g_{bnle} = \delta_{B_{r_1}}(g_{bnp}) + \delta_{L_e}(g_{lep}) \quad (\text{IV.11})$$

2. Supprimer les billes noires et la ligne d'eau dans  $f_p$  par reconstruction du fond, *i.e.* les remplacer par les valeurs de fermeture de  $f_p$  par  $B_{r_1.(3/2)}$  (ici le disque est plus grand qu'une bille noire pour être sûr de supprimer les billes noires dans l'image de fermeture). Ceci a pour effet de ne garder que les billes transparentes dans l'image. La nouvelle image reconstruite  $f_{bt}$  ([Figure IV.11c](#)), est alors définie par :

$$f_{bt} = g_{bnle} \cdot \phi_{B_{r_1.(3/2)}}(f_p) + \overline{g_{bnle}} \cdot f_p \quad (\text{IV.12})$$

3. Amplifier la bordure des billes transparentes par transformation *top-hat* par fermeture, *i.e.* en soustrayant  $f_{bt}$  de sa fermeture par  $B_{r_1}$ . Par cette transformation, la nouvelle image  $f_{btth}$  ([Figure IV.11d](#)) présente des billes transparentes avec une bordure fortement contrastée avec le reste de l'image. Elle est définie par :

$$f_{btth} = \text{BTH}(f_{bt}) = \phi_{B_{r_1}}(f_{bt}) - f_{bt} \quad (\text{IV.13})$$

4. Appliquer l'opérateur *hconcave* sur  $f_{btth}$  pour détecter les pics d'intensité et accentuer encore davantage les bordures des billes transparentes. L'opérateur *hconcave*, noté  $\text{HCONCAVE}_h(f)$ , permet de détecter les minimums locaux dont la profondeur

est supérieure ou égale à une valeur  $h$ . Il utilise une reconstruction par érosion, notée  $R_f^\varepsilon(f + h)$  (fonction `imhmin` dans Matlab), qui permet de supprimer les minimums locaux. Ensuite, en soustrayant  $f$ , les hautes intensités de la reconstruction se retrouvent avec des valeurs proches de 0, et les basses avec des valeurs proches de  $h$ . La nouvelle image  $f_{bthc}$  (Figure IV.11e) est ainsi inversée. Cette méthode *hconcave* appliquée sur  $f_{btth}$  est définie par :

$$f_{bthc} = \text{HCONCAVE}_h(f_{btth}) = R_{f_{btth}}^\varepsilon(f_{btth} + h) - f_{btth} \quad (\text{IV.14})$$

- Appliquer une corrélation de  $f_{bthc}$  avec un modèle de bille transparente  $K_{bt}$ . L'utilisation ici d'une corrélation croisée normalisée, notée NXC (fonction `normxcorr2` dans Matlab, plus de détails en Annexe A.2), permet de trouver dans l'image les zones qui ressemblent le plus au modèle, soit les positions des billes transparentes. Le modèle  $K_{bt}$  consiste en un anneau du même rayon  $r_2$  que les billes transparentes, avec des valeurs négatives sur l'anneau et des valeurs positives à l'intérieur (pour plus de détails sur le modèle choisi, voir la Section IV.2.5). Ainsi, à une position donnée, plus la valeur de corrélation est élevée (proche de 1), plus la chance qu'il y ait une bille à cette position est élevée. L'image  $f_{btnc}$  (Figure IV.11f) présentant les valeurs de corrélation est définie par :

$$f_{btnc} = \text{NXC}(f_{bthc}, K_{bt}) \quad (\text{IV.15})$$

- Détecter les maximums locaux dans  $f_{btnc}$ . Pour ce faire, la méthode se base sur une extraction des maximums locaux, notée RMAX( $f$ ) (fonction `imregionalmax` dans Matlab), qui consiste à détecter toutes les valeurs de pixels qui sont plus fortes que les valeurs de leurs 8 voisins. Cette opération détecte donc les maximums locaux et les retourne dans un masque binaire  $g_{btrm}$ . Ensuite, il faut reconstruire l'image avec seulement les corrélations stockées aux positions non nulles de  $g_{btrm}$ . La nouvelle image  $f_{btrm}$  (Figure IV.11g) est ainsi définie par :

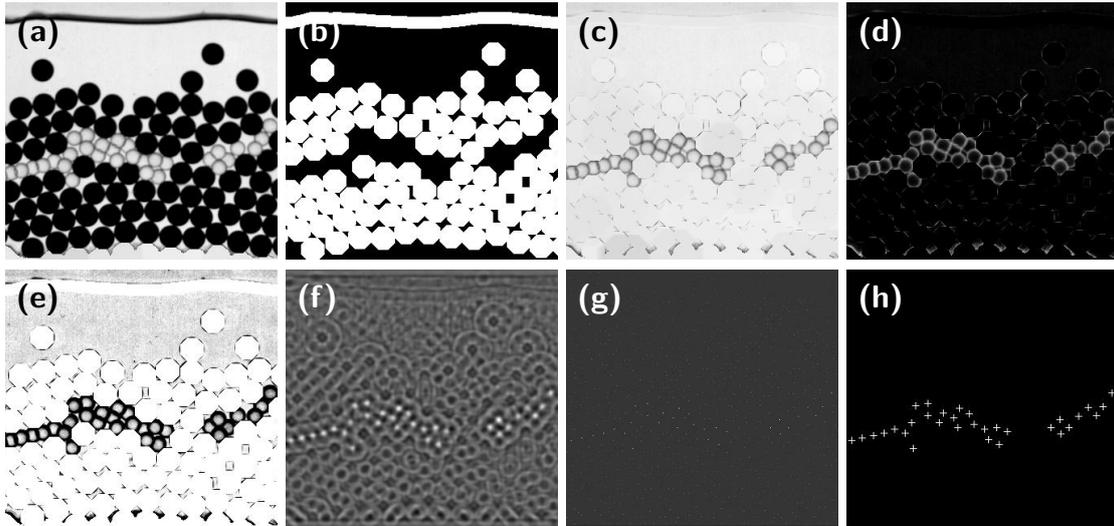
$$\begin{aligned} g_{btrm} &= \text{RMAX}(f_{btnc}) = f_{btnc} - R_{f_{btnc}}^\delta(f_{btnc} - 1) \\ f_{btrm} &= g_{btrm} \cdot f_{btnc} \end{aligned} \quad (\text{IV.16})$$

- Appliquer un seuillage de  $f_{btrm}$  par  $\tau_t$ . Les pixels non nuls du masque binaire  $g_{bt}$  alors obtenu (Figure IV.11h) correspondent à la position des billes transparentes  $D_{bt}$ . Soit, pour  $\mathbf{x}$  un pixel de coordonnées  $(x, y)$  :

$$\begin{aligned} g_{bt} &= T_{\tau_t}(f_{btrm}) \\ D_{bt} &= \{\mathbf{x} \mid g_{bt}(\mathbf{x}) = 1\} \end{aligned} \quad (\text{IV.17})$$

La sortie du détecteur de billes transparentes  $D_{bt}$  est l'ensemble des positions  $(x, y)$  des centres des billes transparentes. Cette nouvelle méthode permet de détecter avec beaucoup plus de fiabilité et robustesse les billes transparentes. Le gain apporté par cette nouvelle méthode sur la méthode précédente est évident et incontestable mais il n'a pas été quantifié. En contre partie, la méthode choisie de corrélation à un modèle d'objet étant plus coûteuse que la version précédente, ce nouveau détecteur est un peu plus long à s'exécuter.

Les paramètres d'entrée et de sortie du détecteur de billes transparentes sont donnés dans le Tableau IV.5.



**Figure IV.11** Procédure de détection des billes transparentes. (a) image pré-traitée, (b) masque binaire des billes noires et de la ligne d'eau, (c) remplacement des billes noires et de la ligne d'eau par du fond, (d) bordure des billes transparentes amplifiée grâce à une transformation *top-hat* par fermeture, (e) résultat de l'opérateur *hconcave*, (f) résultat de la corrélation avec un modèle de bille, (g) détection des maximums locaux, (h) seuillage des corrélations issues des maximums locaux et résultat final avec la position des billes transparentes détectées.

Entrées	Sorties
$r_1, e, h, K_{bt}, \tau_t, D_{le}, D_{bn}, f_p$	$D_{bt}$

**Tableau IV.5** Paramètres d'entrée/sortie du détecteur de billes transparentes appliqué sur l'image  $f_p$  issue du pré-traitement de l'image  $f$ .

### Post-traitements

Des post-traitements sont appliqués aux détections des billes transparentes pour gérer certaines erreurs. Ils consistent à supprimer les fausses détections de billes transparentes qui sont positionnées au-dessus de la ligne d'eau et sur les billes noires. Ce dernier cas, assez rare cependant, peut intervenir dans certaines configurations après le remplacement des billes noires par du fond.

### IV.2.5 Synthèse des paramètres de la détection

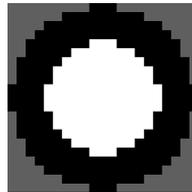
Ce nouvel algorithme de détection est donc constitué des quatre étapes principales décrites ci-avant, à savoir : (1) pré-traitement, (2) détection des billes noires, (3) détection de la ligne d'eau, (4) détection des billes transparentes. Les paramètres d'entrée de ces étapes sont dépendants des grandeurs caractéristiques de l'écoulement (voir Section I.2.2.5), de la particularité de la séquence d'images à traiter et des résultats des étapes de détection précédentes. Le [Tableau IV.6](#) résume ces paramètres.

Le paramètre  $r_1$  est le rayon des billes noires. Le paramètre  $f_0$  correspond à l'image du canal vide. Le paramètre  $e$ , qui correspond à l'épaisseur de la ligne d'eau, dépend des conditions du fluide, en particulier son débit, et de l'effet des parois. Il est mesuré expé-

Étape de détection	Entrées	Sorties
1. Pré-traitement	$r_1, f_0, f$	$\tau_n, f_p$
2. Détecteur billes noires	$r_1, \tau_n, f_p$	$D_{bn}$
3. Détecteur ligne d'eau	$r_1, e, D_{bn}, f_p$	$D_{le}$
4. Détecteur billes transparentes	$r_1, e, h, K_{bt}, \tau_t, D_{le}, D_{bn}, f_p$	$D_{bt}$

**Tableau IV.6** Résumé des paramètres d'entrée/sortie de chaque étape de la détection appliquée sur une image  $f$ .

rimentalement sur une image de la séquence, une seule fois en début de traitement. Le paramètre  $h$  est ajusté sur quelques exemples d'images ayant beaucoup de billes transparentes pour sélectionner des maximums locaux pertinents. Le paramètre  $K_{bt}$  est le modèle de bille transparente pour l'étape de corrélation du détecteur de billes transparentes. C'est un modèle d'anneau qui permet de détecter les anneaux caractéristiques des billes transparentes (voir sa représentation grossie en Figure IV.12). Il dépend du rayon  $r_2$  des billes transparentes et de leur apparence. L'apparence d'une bille transparente, et plus particulièrement l'épaisseur de son anneau gris foncé qui apparaît sur l'image, peut varier d'une séquence à l'autre en fonction des conditions d'acquisition. L'épaisseur est alors mesurée en début de traitement en regardant quelques exemples de billes (les billes transparentes d'une même séquence ont généralement la même apparence). Le paramètre  $\tau_t$  est ajusté en même temps que le paramètre  $h$ .



**Figure IV.12** Modèle d'anneau pour le détecteur de billes transparentes. Les pixels blancs sont égaux à 1, les pixels gris sont égaux à 0 et les pixels noirs sont égaux à une valeur négative qui permet de vérifier que la somme des pixels de l'image soit égal à 0. Le rayon de l'anneau est égal à  $r_2$ , soit au rayon des billes transparentes.

### IV.3 Optimisation du code

L'algorithme de suivi de billes déterministe initialement développé pendant les travaux de Böhm et al. [2006] et Hergault et al. [2010] présente certaines limitations : précision, robustesse, cohérence, quantité de données traitable, temps de calcul, etc. Ce sont ces limitations qui ont motivé une amélioration et une refonte profonde du code de traitement des données. Comme décrit dans la Section IV.1.4, leur algorithme rencontre des difficultés lors des étapes de détection. Pour faire face à ces difficultés, nous avons dans un premier temps cherché à améliorer les algorithmes de détection (voir Section IV.2). Dans un deuxième temps, nous nous sommes concentrés sur la réécriture complète du code et son optimisation, que ce soit en terme de langage, structure de données ou même dans la manière d'exécuter le code. Nous discutons des optimisations apportées dans cette section.

Tout d'abord, leur code de traitement est écrit en langage C alors que le post-traitement

est écrit en Matlab. C'est pour pallier ce manque d'harmonie entre les parties que nous avons choisi de réécrire le code en Matlab et de l'organiser sous la forme d'une bibliothèque de fonctions.

Ensuite, en plus des problèmes que rencontre leur algorithme lors des étapes de détection (en partie corrigés par nos améliorations de la section précédente), cet algorithme a l'inconvénient de ne pouvoir traiter qu'un nombre réduit d'images. En effet, il est limité à l'analyse de séquence de 8000 images (enregistrée à 130 Hz) en raison d'une structure de stockage des données assez lourde et d'un traitement d'images mal adapté à des longues séquences. Ceci est un facteur largement limitant du fait qu'une séquence de 8000 images correspond en fait à environ une minute d'enregistrement ( $8000/130 = 61.5$  sec), et donc qu'à une très courte portion du phénomène de ségrégation. Certes, l'étude d'une séquence aussi courte peut être intéressante lors des périodes de régimes transitoires forts, mais pouvoir trouver les bornes temporelles de telles périodes n'est pas évident sans avoir déjà des informations sur le phénomène complet. C'est afin de permettre le traitement de très longues séquences, jusqu'à 500000 images (environ une heure d'enregistrement), que nous avons apporté une restructuration des données de traitement. Le fait d'avoir toutes les trajectoires sur la séquence complète permet aussi plus de flexibilité quant aux choix des séquences courtes. Un post-découpage des trajectoires est ainsi possible en fonction des besoins de l'analyste.

Enfin, dans le but de réaliser ce traitement sur de longues séquences d'images en un temps de calcul raisonnable, nous avons utilisé une parallélisation de l'exécution du code. Cela permet d'exécuter certaines parties du code en parallèle (*i.e.* en même temps) sur plusieurs processeurs et/ou coeurs et ainsi d'exploiter au maximum les capacités de l'ordinateur de calcul.

Nous décrivons dans cette section les trois choix principaux que nous avons fait vis-à-vis de l'optimisation et la réécriture du code de traitement des données, à savoir la réécriture du code en Matlab, la restructuration des données et la parallélisation du code.

### IV.3.1 Réécriture du code en Matlab

L'étape initiale pour gagner en cohérence et pour avoir tous les traitements et post-traitements intégrés dans un seul et même outil a été de réécrire le code en Matlab et de le réorganiser en bibliothèque de fonctions.

#### IV.3.1.1 Choix d'une réécriture en Matlab

La première modification drastique du code de traitement a été de le réécrire en Matlab dans sa totalité. Dans les travaux précédents, différents outils étaient utilisés avec différents langages de programmation. Les algorithmes de traitement d'images étaient développés dans le langage C alors que les analyses postérieures des trajectoires, des vitesses, etc. étaient elles développées en Matlab. Cette manière dispersée de traiter les données est un inconvénient certain du fait du manque d'intégration et de la complication de communication entre les parties de traitement et post-traitement. C'est donc dans une optique de cohérence et d'intégration que nous avons choisi un seul langage, Matlab, commun pour toute la chaîne de traitement des données.

L'avantage de Matlab est de bénéficier d'une très grande bibliothèque de fonctions spécialisées et optimisées, via les boîtes à outils (*toolboxes*). Notamment, Matlab possède de nombreux outils adaptés pour le traitement d'images (avec la boîte à outils *Image processing*

*toolbox*, voir Section IV.2 pour plus de détails sur les fonctions que nous avons utilisées) et d'autres très utiles pour paralléliser l'exécution d'un code (avec la boîte à outils *Parallel computing toolbox*, voir Section IV.3.3 pour plus de détails sur la parallélisation du code). Il présente un environnement de développement très pratique basé sur l'utilisation de scripts et de fonctions de visualisation avancées. D'autre part, en utilisant à bon escient les possibilités de calcul matriciel et de calcul parallèle, on peut obtenir des temps d'exécutions compétitifs par rapport au langage C. Il est également possible de développer des fonctions coûteuses en temps de calcul directement en C et les exécuter dans Matlab par les fonctions `mex`.

De plus, il y a la possibilité de faire des calculs avec GPU (*Graphics Processing Units*) pour accélérer les opérations compliquées et longues. Notamment avec les deux boîtes à outils sus-citées, de nombreuses opérations morphologiques permettent de tirer profits des calculs sur GPU. Elles s'utilisent avec la même syntaxe que les opérations classiques et permettent alors d'éviter de se plonger dans la complexité des architectures GPU.

Finalement, nous avons pour projet de rendre disponible notre code pour la communauté scientifique car l'outil Matlab est très utilisé par les chercheurs. C'est aussi une des raisons pour lesquelles ce langage courant et étendu a été choisi.

#### IV.3.1.2 Bibliothèque de fonctions

Nous avons profité de cette réécriture du code pour réorganiser la structure globale de l'algorithme. Nous avons établi un découpage du code en fonctions et avons ainsi construit une bibliothèque de fonctions. De cette façon, les trois grandes étapes de l'algorithme déterministe de suivi de billes, à savoir la détection, le suivi et l'estimation des états de mouvement, sont réparties en trois fonctions : `Detection.m`, `Tracking.m` et `MotionStateEstimation.m`. Les quatre étapes de détection (Section IV.2) exécutées successivement dans `Detection.m` ont également leur propre fonction : `ComputePreprocessing.m`, `DetectBlackBeads.m`, `DetectWaterline.m` et `DetectTransparentBeads.m`. Dans `Tracking.m`, de la même façon, les fonctions des deux étapes du suivi (Section IV.1.2) sont `ComputeMatchingCosts.m` et `ComputeGreedyAssociation.m`. Dans la dernière fonction `MotionStateEstimation.m`, les deux étapes de l'algorithme d'estimation des états de mouvement (Section IV.1.3.2) sont réparties en deux fonctions : `ComputeLocalMotionStates.m` et `FilterMotionStates.m`.

### IV.3.2 Restructuration des données

Une autre perspective d'amélioration est de mieux gérer la mémoire utilisée et donc de modifier la manière dont sont stockées les données pendant l'exécution de l'algorithme.

#### IV.3.2.1 Typage des données

La première idée est de déterminer à l'avance le typage des données pour limiter l'espace mémoire utilisé par chaque variable. Cela revient à définir la classe (logique, entier, réel, etc.) de chaque variable et de faire les conversions nécessaires pendant les calculs pour rester dans les classes choisies. Par exemple, pourquoi réserver en mémoire la place d'une variable réelle alors que cette variable ne prendra que des valeurs entières ? C'est afin d'éviter ce genre de situation, qui arrive souvent avec les classes automatiquement prédéfinies par les langages de programmation (comme la classe réel automatiquement choisie par Matlab), que nous avons opté pour un typage des données. Cela nécessite de bien connaître la gamme de

valeurs de chaque variable. Avec un typage adapté, le gain de place en mémoire peut être très important.

### IV.3.2.2 Structure de stockage des données

La deuxième idée consiste en la modification de la manière globale de stocker les trajectoires des précédentes méthodes de Böhm et al. [2006] et Hergault et al. [2010].

#### Structure de Böhm et al. [2006] et Hergault et al. [2010]

Dans les versions de Böhm et al. [2006] et Hergault et al. [2010], les trajectoires sont stockées par traqueur, c'est-à-dire sous la forme d'un tableau de traqueur. Chaque traqueur, qui représente donc la trajectoire d'une bille, possède trois types d'informations : la catégorie de la bille en question (noire ou transparente), le numéro de l'image où la bille est apparue pour la première fois, et la liste des états de la bille en chaque instant. L'état de la bille à un instant donné est défini par sa position  $(x, y)$  dans l'image, sa vitesse  $(u, v)$  et son état de mouvement  $s$ . En Matlab, cette structure de stockage de données est appelée `structure array`, autrement dit c'est un tableau d'éléments de type `structure` (consiste en plusieurs champs spécifiés par des noms de champs, ici `cat` pour la catégorie de la bille, `numim` pour le numéro d'image de l'apparition de la bille, et `stateList` pour la liste des états de la bille).

L'inconvénient de cette structure de données est qu'il n'est pas possible d'utiliser des boucles sur les listes d'états des billes pour analyser les états de mouvement par exemple. De plus, elle peut prendre beaucoup de place en mémoire car chaque élément peut être de type différent. Cette manière de structurer les données par trajectoire est tout de même très utile pour faire des analyses de trajectoires. Cependant, il est plus difficile d'en déduire une analyse par image, c'est-à-dire une analyse qui permet d'étudier l'état des trajectoires à un instant donné. En effet, pour une image donnée, un parcours complet de la structure de stockage par traqueur est nécessaire pour obtenir les informations sur les trajectoires présentes dans l'image. Et nous avons constaté qu'une analyse par image est plus souvent utilisée qu'une analyse par trajectoire. Nous avons alors mis en place une nouvelle structure de données.

#### Nouvelle structure

Cette nouvelle structure de données ne stocke plus les données par trajectoire mais par image. Ainsi, en chaque image, un tableau stocke l'état de chaque bille présente dans l'image. Cette fois-ci, en plus des informations  $(x, y, u, v, s)$ , l'état d'une bille est défini par sa catégorie  $c$  (0 pour une bille noire et 1 pour une transparente) et son numéro de traqueur  $tr$ . Donc pour résumer, chaque ligne du tableau présente les informations  $(x, y, c, tr, u, v, s)$ . En Matlab, cette structure de stockage de données est appelée `cell array`, autrement dit c'est un tableau d'éléments de type `cell` (consiste en plusieurs cellules spécifiées par des numéros, ici 1 pour  $x$ , 2 pour  $y$ ,... 7 pour  $s$ ). L'avantage de cette structure de données est qu'elle peut être étudiée de la même façon que des tableaux car elle est indexée avec des numéros. Ainsi, il est possible d'utiliser des boucles pour rechercher des informations sur les trajectoires, les états de mouvement ou n'importe quel autre élément du tableau.

De plus, cette manière de stocker les données permet de réutiliser directement la structure de données de la sortie de l'étape de détection, soit un tableau pour chaque image avec les informations  $(x, y, c)$  pour chaque bille détectée. L'idée est alors d'agrandir cette structure en rajoutant quatre colonnes pour les informations  $(tr, u, v, s)$ . Ceci permet de gagner une

place importante en mémoire par rapport à la structure de Böhm et al. [2006] et Hergault et al. [2010] qui duplique les informations de positions stockées dans le tableau des détections et crée ainsi de la redondance. Sur des longues séquences, la redondance des informations peut être fatale à l'usage mémoire, il est donc primordial de la limiter. De surcroît, avec un typage des données adapté, simple et identique à chaque élément (voir Section IV.3.2.1), le gain mémoire est d'autant plus important.

D'autre part, le fait d'avoir un identifiant de traqueur à chaque bille permet de chaîner les billes d'une image à l'autre, et ainsi de pouvoir faire une analyse par trajectoire si besoin. Donc les deux types d'analyse, par trajectoire et par image, sont possible rapidement et facilement avec cette structure de données en `cell array`, contrairement à celle en `structure array`.

C'est cette modification de la structure de données qui nous a permis de résoudre les problèmes de mémoire et par conséquent de pouvoir lancer l'algorithme déterministe de suivi de billes sur des très longues séquences.

### IV.3.3 Parallélisation du code

Une dernière perspective d'optimisation est la parallélisation du code. Maintenant qu'il est possible de lancer l'algorithme sur des longues séquences grâce à une nouvelle structure des données (voir ci-avant), l'idée est qu'il puisse s'exécuter dans un temps convenable. Dans cette section, nous discutons des principes de la parallélisation, de son implémentation en Matlab, des étapes parallélisables de notre algorithme de suivi déterministe et du gain apporté par cette parallélisation sur les temps d'exécution. Les temps d'exécution sont calculés avec un ordinateur à 2 processeurs Intel(R) Xeon(R) 2.40 GHz, chaque processeur possède 6 coeurs, la mémoire RAM est de 32 GB. Quand nous parlons de traitement parallèle, le code est exécuté sur les 12 coeurs alors disponibles.

#### IV.3.3.1 Principe de parallélisation

Le principe du traitement parallèle (*parallel computing*) est d'utiliser simultanément des ressources informatiques multiples pour résoudre un problème calculatoire. Pour ce faire, il est nécessaire de séparer le problème en plusieurs parties pouvant être résolues en même temps. Ces parties, comportant généralement une suite d'instructions, doivent être indépendantes les unes des autres. Les instructions de chaque partie peuvent alors s'exécuter simultanément sur différents processeurs (ou coeurs). Cependant, le traitement parallèle n'est envisageable que si l'ordinateur de calcul possède plusieurs processeurs (ou un processeur avec plusieurs coeurs). Or de nos jours, tous les ordinateurs ont au moins un processeur qui a au moins deux coeurs. Le traitement parallèle permet ainsi d'exploiter au maximum les ressources de l'ordinateur, de gagner du temps, voire de l'argent (avec de multiples composants bon marché pouvant utiliser le traitement parallèle plutôt qu'un composant cher ne pouvant exécuter qu'en série).

#### IV.3.3.2 Parallélisation sous Matlab

Pour paralléliser l'exécution de notre code, nous avons utilisé les boucles parallèles mises à disposition par la boîte à outils *Parallel computing toolbox* de Matlab. Cette boîte à outils permet de choisir le nombre de processeurs et/ou coeurs, appelés *workers*, sur lesquels les calculs (qui sont indépendants les uns des autres) sont envoyés en parallèle. Ainsi, avec

l'utilisation d'une boucle nommée `parfor` à la place d'une boucle `for` dans le programme principal, Matlab gère tout seul l'envoi des calculs en parallèle sur les `workers`, il récupère leurs résultats et leur envoie ensuite les calculs suivants. Par exemple, pour traiter 100000 calculs indépendants avec 5 `workers`, le programme principal présentant la boucle `parfor` envoie automatiquement 20000 calculs sur chaque `worker` en même temps. Si le temps de communication des résultats entre le programme principal et les `workers` était nul (ce temps reste assez faible quand la quantité de données à transmettre n'est pas trop élevée) et si l'utilisation des ressources de l'ordinateur n'était dédiée qu'aux calculs (en réalité impossible car l'ordinateur doit au moins entretenir le système d'exploitation), le gain de temps optimal serait égal au nombre de `workers`, soit un facteur 5 dans l'exemple donné. En pratique, le gain de temps est plutôt légèrement inférieur à l'optimal et dépend surtout de la quantité de ressources laissées disponibles par l'ordinateur.

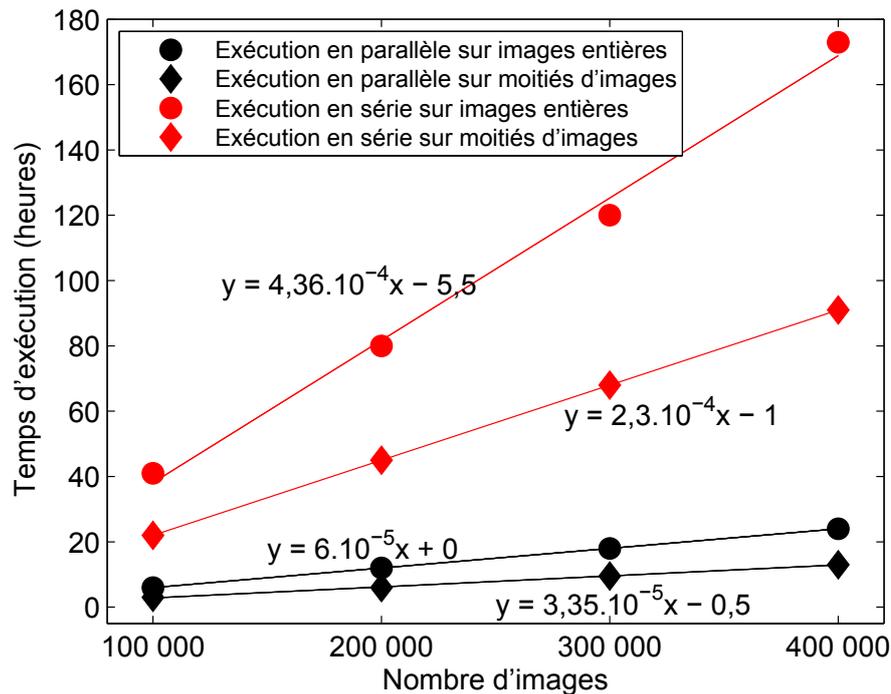
### IV.3.3.3 Parallélisation de l'algorithme de suivi déterministe

Dans notre algorithme déterministe de suivi de billes, toutes les étapes ne sont néanmoins pas parallélisables. En effet, l'étape de suivi des billes ne peut l'être car elle nécessite d'être exécutée en série, c'est-à-dire qu'il faut associer les billes entre l'image 1 et l'image 2, puis utiliser le résultat des associations obtenues pour associer les billes entre l'image 2 et l'image 3, etc. jusqu'au dernier couple d'images consécutives. L'étape d'estimation des états de mouvement ne peut l'être non plus car il faut pouvoir accéder à plusieurs images consécutives pour moyenner les vitesses des trajectoires et aussi accéder aux trajectoires complètes pour filtrer les états par trajectoire. En revanche, l'étape de détection des billes qui s'exécute image par image se prête très bien à une parallélisation. En effet, il est possible de réaliser la détection des billes sur plusieurs images simultanément du fait de l'indépendance entre les images, autrement dit la détection d'une image ne dépend que de cette image. Nous avons donc adapté l'étape de détection afin de la rendre parallèle. Cette adaptation consiste à rendre chaque itération indépendante de la boucle de détection. Pour ce faire, chaque opération dans la boucle ne doit dépendre que de l'indice de l'itération courante, de paramètres globaux et de résultats d'opérations exécutées dans cette même itération.

### IV.3.3.4 Analyse du gain d'une parallélisation

La [Figure IV.13](#) montre les temps d'exécution de la détection des billes quand elle est exécutée de manière parallèle versus en série (ou non-parallèle). Pour tester sur différents cas de figure, nous avons fait varier le nombre d'images à traiter (entre 100000 et 400000) et leur dimension. Ici, pour faire varier la dimension des images, nous avons coupé verticalement les images en deux parties égales et pris que la première partie. Nous avons donc utilisé deux tailles d'images, soit les images entières, soit les moitiés d'images. Étant donné que la répartition des billes est homogène dans l'image (*i.e.* autant de billes dans la partie gauche que dans la partie droite), le nombre de billes est alors divisé en deux dans les moitiés d'images. Mais comme l'étape de détection est indépendante du nombre de billes, l'impact du nombre de billes dans le temps d'exécution est négligeable.

Dans cette figure, nous pouvons observer que le calcul en parallèle (appliqué sur 12 coeurs cadencés à 2.40 GHz) permet de gagner un temps considérable par rapport au calcul en série, et cela peu importe le cas de figure. La valeur du gain de temps est d'un facteur d'environ 7 (obtenu en calculant le rapport entre les temps en série sur ceux en parallèle).



**Figure IV.13** Temps d'exécution de la détection des billes en fonction du nombre d'images. Comparaison d'une exécution du code en parallèle (en noir) et en série (en rouge), sur des images entières (marqueur rond) ou sur des moitiés d'images (marqueur diamant), avec les droites de régression linéaire respectives et équations associées.

De plus, nous pouvons aussi constater que les temps d'exécution sont linéaires au nombre d'images (les ordonnées à l'origine des droites de régression linéaire sont proches de zéro) et à la dimension des images (quand l'image est deux fois plus petite, le temps d'exécution est divisé par deux).

C'est donc cette parallélisation du code qui nous a permis d'exécuter la détection des billes sur des très longues séquences dans un temps raisonnable. Pour donner une idée du temps qu'il faut à l'algorithme complet de suivi de billes déterministe pour traiter une séquence de 400000 images, la détection s'exécute donc en parallèle en 24 h, le suivi s'exécute en série en 51 h et l'estimation des états de mouvement s'exécute en un peu plus de 150 h.

## IV.4 Bases de données d'évaluation

Dans notre cas d'étude très spécifique, à savoir des images avec des billes similaires et nombreuses, il est difficile de tester nos algorithmes de suivi de billes sur des bases de données d'évaluation *open source* (*i.e.* des jeux de données d'évaluation mis à la disposition des chercheurs dans le domaine du tracking). En effet, les bases de données usuelles en suivi d'objets multiples sont plutôt adaptées au suivi de personnes, véhicules ou objets en vidéo surveillance, analyse sportive, ou autres mais rarement pour des objets similaires de même taille, forme et couleur qui sont en très grand nombre. Il est donc nécessaire de créer nos propres bases de données pour permettre d'évaluer les performances, la précision et l'efficacité de nos algorithmes. Pour ce faire, nous avons élaboré des vérités terrains

qui permettent ensuite de mesurer les performances des algorithmes avec des métriques d'évaluation.

### IV.4.1 Élaboration de vérités terrains

Pour évaluer nos algorithmes de suivi, nous avons réalisé deux séquences de vérité terrain. La première séquence est issue de l'expérience décrite en Section I.2.2. La deuxième a été créée à partir d'une simulation numérique.

#### IV.4.1.1 Séquence expérimentale

La séquence expérimentale sur laquelle nous avons élaboré une vérité terrain est une séquence de 1000 images avec approximativement 400 billes par images (environ 300 billes noires et 100 transparentes). Elle est donnée par la suite sous le nom ExpSeq. La vérité terrain est obtenue en quatre étapes : (1) détection des billes, (2) correction des erreurs de détections, (3) suivi des détections, (4) estimation des états de mouvement. Cette vérité terrain est limitée à une séquence de 1000 images car elle a pris beaucoup de temps à être construite. Les quatre étapes sont décrites ici.

1. La première étape consiste à détecter les billes dans les images avec nos détecteurs (voir Section IV.2). Ces détecteurs, bien qu'optimisés, peuvent présenter quelques erreurs de détection, et particulièrement le détecteur des billes transparentes.
2. La deuxième étape consiste à corriger les éventuelles erreurs. Pour ce faire, les détections d'une image sont affichées par-dessus l'image par des croix à leur position. Visuellement, il est possible de voir les erreurs de détection sur l'image, les détections peuvent alors être éditées. Nous pouvons alors supprimer les fausses détections (faux positifs) et rajouter manuellement les détections manquantes (faux négatifs). Ces corrections sont faites sur toute la séquence, ce qui permet d'obtenir une vérité terrain des détections.
3. La troisième étape consiste à établir la vérité terrain du suivi des billes. Ceci est réalisé en appliquant l'algorithme glouton d'association de billes (voir Section IV.1.2) sur la vérité terrain des détections créée ci-avant. Quand il n'y a aucune erreur de détection, cet algorithme est très efficace, il ne retourne que très rarement des erreurs de suivi (car pas d'occlusion possible). Nous avons tout de même fait une vérification visuelle des trajectoires. Les associations obtenues forment alors la vérité terrain des trajectoires des billes.
4. La dernière étape permet de donner en chaque instant de chaque trajectoire un état de mouvement. Nous utilisons pour cela la méthode d'estimation des états de mouvement décrite en Section IV.1.3. Nous obtenons alors la vérité terrain des états de mouvement associés aux trajectoires.

La limitation de cette vérité terrain vient de la précision des positions. En effet, les centres des détections manquantes ont été sélectionnés manuellement pour être visuellement au centre des billes. Ils peuvent donc être imprécis d'un ou deux pixels. De plus, comme nous utilisons la sortie directe de son détecteur pour créer la vérité terrain, les positions détectées peuvent elles aussi être entachées d'erreurs et nous ne pourrons pas les utiliser pour évaluer la précision du détecteur.

Cependant, cette vérité terrain convient très bien pour vérifier comment les algorithmes de suivi gèrent les problèmes de détection comme les faux positifs et faux négatifs par exemple.

#### IV.4.1.2 Séquence numérique

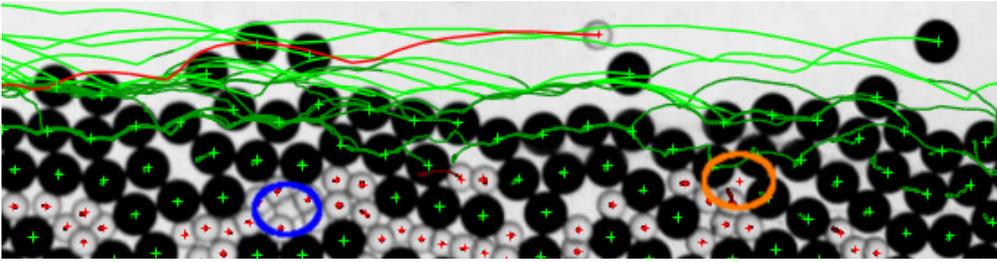
Dans le but d'étudier une séquence d'images plus longue, nous avons créé une séquence numérique de transport solide par charriage de 10000 images grâce à un modèle développé à Irstea. Ce modèle numérique est basé sur une méthode aux éléments discrets couplée à un modèle hydrodynamique adapté à notre écoulement turbulent (voir Maurin et al. [2015, 2016]). Ce modèle a été validé par rapport aux expériences faites à Irstea (voir Frey [2014]). L'idée de cette simulation est de reproduire du mieux possible les conditions de nos expériences (comme par exemples : la largeur du canal, le nombre de chaque type de billes, les débits d'alimentation des flux solides, la vitesse du fluide, la fréquence d'images, etc.). Grâce à ParaView, un logiciel *open source* utilisé pour la visualisation scientifique, nous avons pu avoir un rendu des images de la séquence numérique très proche du rendu des images expérimentales. Avec ce rendu, il est possible d'utiliser la séquence numérique de la même manière qu'une séquence expérimentale et ainsi d'appliquer les mêmes algorithmes de suivi de billes (détection, suivi et estimation des états de mouvement).

Contrairement à la vérité terrain de la séquence expérimentale, nous pouvons utiliser la séquence numérique comme une vérité terrain pour tous types de tests. En effet, elle est calculée par une autre méthode que l'algorithme déterministe et ne présente aucune erreur de détection. C'est donc bien une vérité terrain pour la précision à l'inverse de la version expérimentale. Les états de mouvement de cette vérité terrain numérique sont obtenus de la même manière que la séquence expérimentale. Cette séquence numérique est désignée NumSeq dans la suite.

#### IV.4.2 Choix des métriques d'évaluation

Les métriques d'évaluation que nous utilisons pour tester les performances des algorithmes de suivi de billes sont celles décrites en Section II.2.3. Ces métriques comprennent les faux positifs, les faux négatifs, le nombre d'échanges d'ID et les deux métriques CLEAR MOT à savoir MOTP et MOTA. Dans des images avec des billes comme les nôtres, les métriques sont adaptées. Les faux positifs correspondent à des billes faussement détectées, ce sont majoritairement des billes transparentes. Les faux négatifs correspondent à des détections de billes manquantes, et là aussi ce sont majoritairement des billes transparentes. Des exemples de faux positifs et faux négatifs sont illustrés en Figure IV.14. Les échanges d'ID sont des confusions entre traqueurs entraînant un échange de leur identifiant, cette confusion intervient lors de la résolution des associations et est principalement causée par les faux positifs et faux négatifs (les occlusions sont habituellement responsables d'échanges d'ID mais nous n'en avons pas dans nos séquences d'images). Le MOTP mesure la précision des positions estimées des billes par rapport à la vérité terrain. On l'utilise dans ses deux versions en fonction des besoins, en pourcentage (noté  $MOTP_{\%}$ ) ou en pixels (noté  $MOTP_{px}$ ). Le MOTA est une mesure qui prend en compte les FP, FN et le nombre d'échanges d'ID.

En plus de ces métriques, nous avons utilisé une métrique qui détermine le pourcentage de trajectoires correctes. Une trajectoire est considérée comme correcte sur toute sa longueur si elle ne présente ni faux négatifs ni échanges d'ID sur plus de 95% du temps.



**Figure IV.14** Illustration des erreurs de type faux positif et faux négatif sur une image de trajectoires de billes. Le rond orange montre un faux positif (ou une fausse détection), le rond bleu montre un faux négatif (ou détection manquante).

## IV.5 Résultats de l'algorithme déterministe de suivi et discussion

Dans cette section, nous discutons des résultats obtenus par l'application de l'algorithme déterministe de suivi. Tout d'abord, nous évaluons les performances de l'algorithme sur les deux séquences de vérité terrain ExpSeq et NumSeq avec les différentes métriques décrites dans la section précédente. Puis, nous donnons quelques exemples de résultats que l'on peut obtenir sur une longue séquence d'images. Enfin, nous présentons les limitations de cette approche déterministe.

### IV.5.1 Évaluation des performances sur les vérités terrains

Le [Tableau IV.7](#) montre les résultats d'évaluation des performances de l'algorithme déterministe de suivi de billes sur deux séquences de vérité terrain (la séquence expérimentale ExpSeq et la numérique NumSeq), et sur les deux catégories de billes (billes noires et transparentes). Les performances sont illustrées par les mesures d'évaluation évoquées dans la section précédente. Une observation assez flagrante à la vue des performances est que chaque mesure présente de très bons résultats, moins de 1.5% pour les taux de FP et FN, et supérieur à 95% pour les valeurs de MOTA et MOTP% (pas de valeur MOTP pour ExpSeq comme expliqué en [Section IV.4.1.1](#)). Ceci vient du fait que les seuils de détection sont tous très bien adaptés aux séquences, et donc que les détecteurs de billes ne retournent que peu d'erreurs. L'étape de suivi, se basant sur un ensemble de détections avec très peu d'erreurs, est toujours efficace. D'où les bons résultats.

Un autre résultat que l'on peut tirer de ce tableau est une estimation de la précision des détecteurs de billes avec les valeurs de MOTP<sub>px</sub>. On constate que le détecteur de billes noires présente une erreur moyenne de 0.44 px et celui de billes transparentes de 0.65 px. On peut aussi constater que les performances sont bien meilleures pour le détecteur de billes noires que pour celui des transparentes. Ceci s'explique par la fiabilité et l'efficacité du détecteur de billes, alors que les billes transparentes ne sont pas toujours si faciles à détecter à cause des légères occlusions qu'elles subissent dans les images.

Il est important de préciser ici que la plupart des billes sont immobiles, c'est-à-dire que généralement autour de 70% des billes de la séquence sont en repos, 25% sont en roulement et 5% en saut. Sur les FN, nous avons remarqué que la grande majorité des erreurs interviennent dans le lit quasi-immobile, soit sur les billes en repos. Ceci s'explique par la complexité des configurations de billes en réseau et le nombre d'occlusions partielles plus

Séquence	Categ.	Nb/im	FP	FN	Éch. ID	Tr. Correct.	MOTA	MOTP%	MOTP <sub>px</sub>
ExpSeq	Noires	312.4	0.00%	0.00%	0	100.00%	100.00%	-	-
	Trans.	77.4	1.04%	1.49%	34	97.44%	97.35%	-	-
	Total	389.8	0.21%	0.30%	34	99.57%	99.47%	-	-
NumSeq	Noires	392.6	0.58%	0.14%	76	98.58%	99.28%	98.01%	0.44 px
	Trans.	77.4	0.63%	0.32%	69	95.10%	99.03%	95.66%	0.65 px
	Total	470.0	0.59%	0.17%	145	98.05%	99.23%	97.62%	0.48 px

**Tableau IV.7** Évaluation des performances de l'algorithme déterministe de suivi de billes sur deux séquences de vérité terrain ExpSeq et NumSeq (respectivement une séquence expérimentale et une numérique). Les résultats sur les billes noires, transparentes et les deux (noté Total) donnent le nombre de billes moyen par image et montrent les valeurs des mesures d'évaluation : taux de faux positifs (FP), taux de faux négatifs (FN), nombre d'échanges d'ID de traqueur (Éch. ID), pourcentage de trajectoires correctes (Tr. Correct.) (une trajectoire est considérée correcte si sur plus de 95% du temps elle n'a ni faux négatif et échange d'ID), et les métriques CLEAR MOT [Bernardin and Stiefelhagen, 2008] d'exactitude du suivi (MOTA) et de précision des positions (MOTP). Cette dernière est donnée dans ses deux versions, en pourcentage (MOTP%) et en pixels MOTP<sub>px</sub>.

important qu'en surface du lit. De plus, les billes à l'état repos étant les plus représentées, les taux d'erreurs présentés dans le tableau sont ainsi principalement guidés par les taux d'erreur obtenus dans l'état de repos. Les billes en roulement et en saltation quant à elles ne présentent qu'un nombre limité de voisins et donc d'occlusions partielles, ce qui permet de les détecter plus facilement et de rencontrer des erreurs que très rarement.

Cependant, bien que très bonnes, les valeurs de ces mesures sont à replacer dans le contexte du grand nombre d'objets dans les images et du grand nombre d'images dans la séquence. En effet, dans l'optique d'étudier des séquences de 500000 images avec environ 400 billes par image, le nombre total de détections est de 200 millions. Un taux de FP de 0.50% par exemple, équivaldrait à 1 million de fausses détections. De plus, le nombre total de trajectoires dans une telle séquence est d'environ 160000. Si 99% des trajectoires sont correctes, cela signifie que 1600 sont incorrectes. Si l'on veut étudier le phénomène d'infiltration, il faut considérer des trajectoires de billes transparentes qui vont s'insérer progressivement dans le réseau du lit et impliquant donc des billes transparentes passant par des états immobiles. Un trop grand nombre d'erreurs sur ces trajectoires peu fréquentes pourrait fausser les mesures.

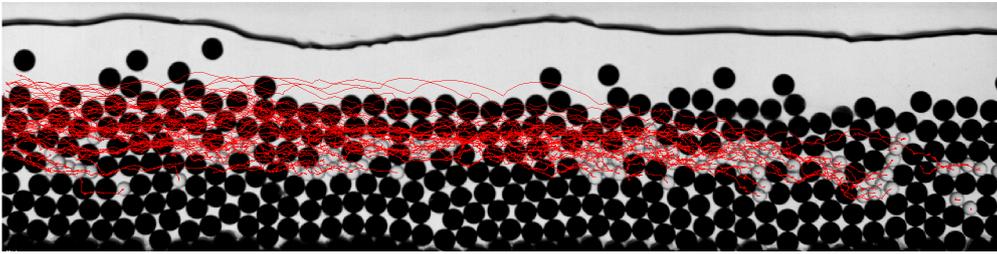
#### IV.5.2 Quelques résultats sur une longue séquence

Avec les optimisations apportées, il est maintenant possible d'utiliser l'algorithme déterministe de suivi sur des longues séquences. Il permet d'analyser le phénomène de ségrégation sur une échelle temporelle plus étendue. Nous présentons ici quelques exemples de résultats d'analyses sur une séquence de 400000 images. Ces premiers résultats devront être analysés plus finement par les chercheurs en mécanique des fluides.

##### Trajectoires des billes fines

La Figure IV.15 illustre les trajectoires de billes transparentes 15 minutes après le début de leur approvisionnement dans le canal. On peut notamment constater le phénomène d'infiltration. Cet outil de visualisation des trajectoires peut être très pratique pour suivre l'évolution des déplacements, pour constater des schémas de mouvement, ou encore tout simplement

pour vérifier visuellement si l'algorithme a correctement fonctionné et voir d'éventuelles erreurs.



**Figure IV.15** Visualisation des trajectoires de billes transparentes sur 15 minutes.

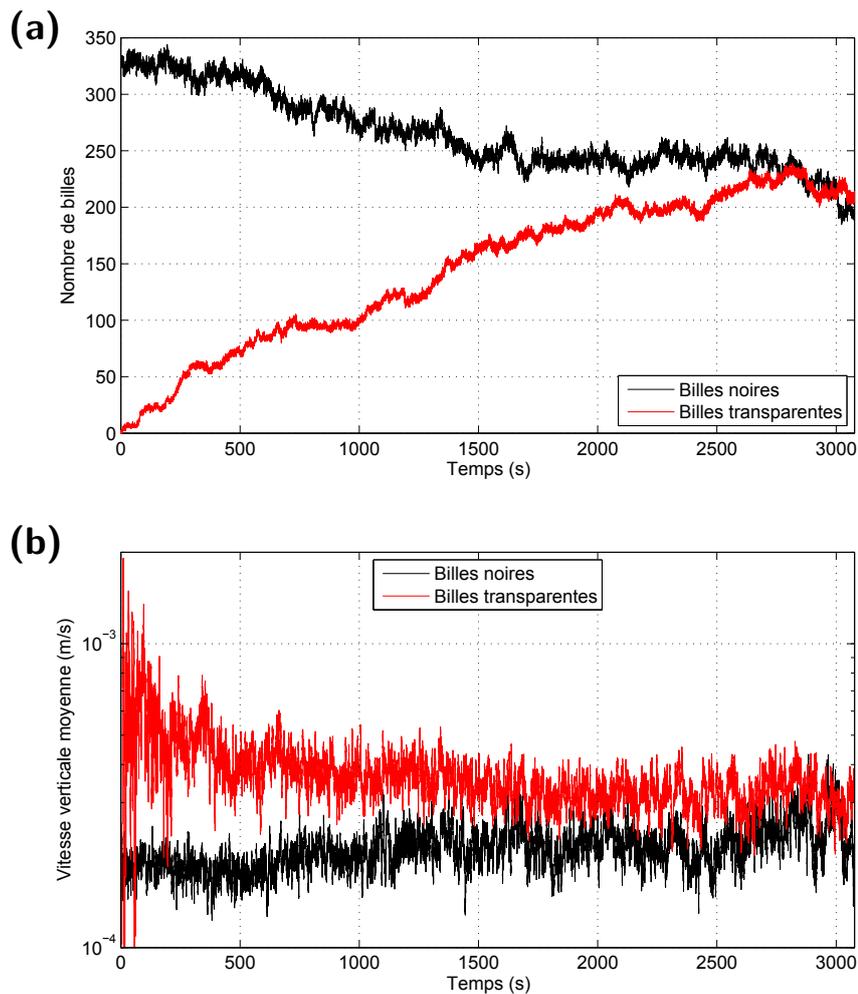
### Évolution du nombre de billes et de la vitesse

La [Figure IV.16a](#) permet de suivre l'évolution temporelle du nombre de billes dans le champ de l'image selon la taille. On peut par exemple constater qu'on arrive à un plateau de billes noires vers 1500 sec alors que le nombre de billes transparentes continue d'évoluer. La [Figure IV.16b](#) donne l'évolution de la valeur absolue de la vitesse verticale (moyennée sur le champ de l'image) des billes dans le lit. Ici, ne sont prises en compte que les billes en repos. Une vitesse verticale peut être négative ou positive car les petites billes sont typiquement cisailées par les grosses billes et suivent les creux et les bosses des grosses billes. Quel que soit le signe, on observe une décroissance rapide de la vitesse moyenne des billes transparentes qui passe de initialement  $10^{-3}$  m/s à  $2.5 \times 10^{-4}$  m/s vers 500 sec. La décroissance est ensuite beaucoup plus lente. Visuellement, cela correspond au début à un faible nombre dilué de billes transparentes dans les couches supérieures du lit alors qu'ensuite on observe beaucoup plus d'amas (ou clusters) dans des couches inférieures. Ces amas finissent par former une zone quasi-continue vers 1500 sec. La vitesse verticale des billes transparentes n'évolue alors plus et sa valeur est assez proche de celle des billes noires. Cet exemple montre que la possibilité de traiter de longues séquences permet une vue d'ensemble rapide et aide à découper de façon pertinente la séquence temporelle pour une analyse plus fine des divers régimes de ségrégation.

Les observations évoquées ci-dessus devront être étudiées plus précisément par les mécaniciens pour élaborer des conclusions sur la ségrégation.

### IV.5.3 Limitations de l'algorithme déterministe de suivi

L'algorithme déterministe de suivi de billes est certes bien utile pour l'étude de longues séquences mais il présente certaines limitations. D'une part, il ne prend pas en compte les erreurs de détection. Dans les résultats sur une séquence de 400000 images présentés ci-avant, l'algorithme fonctionne très bien car c'est une séquence que nous avons beaucoup étudiée et donc les différents paramètres de l'algorithme y sont tous ajustés. Mais ce n'est pas toujours le cas, la détection présente souvent des erreurs de faux positifs et faux négatifs qui rendent ensuite le suivi plus difficile et problématique. L'algorithme global n'en est alors que moins précis. D'autre part, l'algorithme procède en plusieurs passes et notamment les états de mouvements ne sont disponibles qu'après la dernière passe. Si l'on cumule le temps d'exécution de chacune des étapes, cela donne un temps de calcul supérieur à une semaine. Cette durée peut être un facteur limitant lorsqu'il y a plusieurs longues séquences à analyser



**Figure IV.16** Résultats de l'algorithme déterministe de suivi sur une longue séquence. (a) évolution temporelle du nombre de billes selon la taille, (b) évolution de la valeur absolue de la vitesse verticale des petites billes.

ou lorsqu'un résultat rapide est souhaité sur une séquence pour vérifier la pertinence des divers paramètres de l'expérimentation.

## IV.6 Conclusion

Dans ce chapitre, nous avons présenté chacune des étapes d'un nouvel algorithme dit déterministe de suivi de billes. Ce nouvel algorithme, basé sur les versions initiales proposées par Böhm et al. [2006] et Hergault et al. [2010], présente de nombreuses optimisations pour le rendre applicable sur des longues séquences avec un temps d'exécution raisonnable. Ces optimisations comportent notamment des améliorations sur les algorithmes de détection, des nouvelles structures de données et une parallélisation du code sous Matlab. De plus, nous avons mis au point une mesure d'évaluation des performances et conçu des séquences vérités terrains pour pouvoir juger la qualité des résultats. Notamment, nous avons pu voir que les performances en terme de précision de détection sont de l'ordre de 0.5 pixels et que les trajectoires sont correctes plus de 95% du temps.

Cette manière déterministe de traiter le suivi de billes s'appuie sur l'efficacité de la détection. Donc en présence d'un détecteur moins fiable, le suivi peut présenter des erreurs dans les trajectoires. Une perspective d'évolution serait de prendre en compte ces éventuelles erreurs de détection dans l'étape de suivi. Par ailleurs, l'algorithme ne considère pas les différents modes de déplacement des billes que l'on retrouve dans la ségrégation en charriage (repos, roulement, saltation). Fournir les connaissances mécaniques sur les comportements des billes dans l'expérience pourrait permettre de mieux prédire les positions. De plus, un traitement en ligne du suivi et de la détermination de l'état de mouvement permettrait de pouvoir vérifier en cours d'exécution de la pertinence des résultats et éviter un traitement a posteriori pour déterminer les états de mouvement. Le chapitre suivant traite de ces évolutions.

## Chapitre V

# Algorithme de suivi par filtrage particulière à modèles multiples

### Sommaire

---

<b>V.1</b>	<b>Contexte, problématique et objectifs . . . . .</b>	<b>83</b>
<b>V.2</b>	<b>Algorithme de suivi multi-objets et multi-modèles . . . . .</b>	<b>85</b>
V.2.1	Estimation de la confiance du détecteur . . . . .	85
V.2.2	Filtrage particulière à modèle multiples . . . . .	86
V.2.3	Détails d'implémentation . . . . .	89
<b>V.3</b>	<b>Résultats et discussions . . . . .</b>	<b>91</b>
V.3.1	Algorithmes de comparaison . . . . .	91
V.3.2	Évaluation de l'algorithme de suivi . . . . .	91
<b>V.4</b>	<b>Éléments de discussion complémentaires . . . . .</b>	<b>95</b>
<b>V.5</b>	<b>Conclusion . . . . .</b>	<b>96</b>

---

Dans ce chapitre, nous présentons un nouvel algorithme de suivi de billes qui s'appuie sur un filtrage particulière à modèles dynamiques multiples. Il a été développé pour faire face aux limitations de précision et de robustesse de l'algorithme de suivi déterministe détaillé dans le Chapitre IV. Il a donné lieu à un premier travail présenté à la conférence internationale *IEEE ICIP* de Phoenix, États-Unis en 2016, dont l'article [Lafaye de Micheaux et al. \[2016\]](#) donné en Annexe D a été publié dans ses actes. Dans la continuité, nous avons poursuivi ce travail et amélioré encore l'algorithme, ce qui a fait l'objet d'une publication soumise au journal *Machine Vision and Applications* (en cours d'examen). Cet article est exposé au complet en Annexe E tel qu'il a été soumis, il est résumé dans ce chapitre.

Nous présentons d'abord le contexte, la problématique et les objectifs de ce nouvel algorithme. Ensuite, nous exposons ses différentes étapes et son évaluation par rapport à des vérités terrains. Enfin, nous discutons de ces limitations et de quelques points à améliorer.

### V.1 Contexte, problématique et objectifs

L'algorithme déterministe de suivi présenté dans le chapitre précédent est utilisé pour traiter de longues séquences d'images grâce à sa bonne gestion de la mémoire et sa rapidité d'exécution. Cependant, il ne gère pas les éventuelles erreurs du détecteur (faux positifs

et faux négatifs). Ces erreurs peuvent ainsi créer des trajectoires inexistantes et en arrêter d'autres bien réelles. Or, dans notre étude de ségrégation en charriage, dans les périodes à forts régimes transitoires, il est important d'avoir des trajectoires les plus précises possibles pour pouvoir tirer des conclusions justes sur les caractéristiques du phénomène de ségrégation. D'autre part, l'état de mouvement n'est pas estimé en ligne et nécessite une deuxième passe de calcul qui peut être coûteuse en temps de calcul. Ces différentes limitations ont motivé l'élaboration d'un nouvel algorithme de suivi de billes. L'objectif ici est donc de prendre en compte les erreurs de détection dans le modèle de suivi de trajectoires et d'estimer en ligne les états de mouvement.

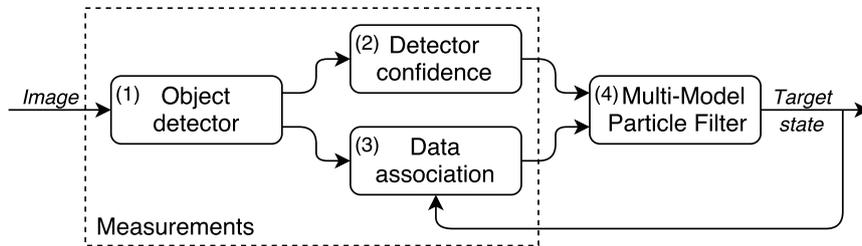
Pour traiter les erreurs de détection, certaines approches se basent sur un filtrage particulière (PF), elles sont notamment utilisées dans différentes applications [Choset and Nagatani, 2001; Rathi et al., 2007; Lu, 2007]. Le principal avantage de ces méthodes est de permettre l'estimation de la distribution de l'état de la cible en incorporant un modèle de mouvement linéaire. Dans ce contexte, Breitenstein et al. [2011] a proposé une approche de suivi multi-cible très efficace en introduisant la confiance du détecteur pour gérer les détections manquantes. Cependant, cette méthode ne prend en compte qu'un seul modèle de mouvement, ce qui peut être limitant dans un contexte à fortes interactions entre objets comme le nôtre.

Pour pallier cette limitation, l'approche à modèle multiple (MM) ([Arulampalam et al., 2004; Li and Jilkov, 2005]), aussi appelée approche à modèle dynamique changeant [Odobez et al., 2006], a été introduite et utilisée dans plusieurs applications comme la robotique et la navigation [Boers and Driessen, 2003] ou la vidéo surveillance [Zhai et al., 2009]. Elle suppose que, étant donné un nombre fini de modèles dynamiques, une cible peut sauter d'un modèle à un autre selon des probabilités de transition. Il y a également l'algorithme à modèle multiple interactif (IMM) [Blom and Bar-Shalom, 1988; Mazor et al., 1998] et celui à filtrage particulière à modèle multiple (MMPF) [Isard and Blake, 1998a; McGinnity and Irwin, 2000] dans cette même perspective. Ce dernier utilise l'idée de l'approche multi-modèle, c'est-à-dire que les mouvements d'un objet peuvent être classifiés en différents états en fonction de la situation, et ceci dans un cadre de filtrage particulière. Néanmoins, cette approche requiert de connaître tous les états de mouvement au préalable avec une description mathématique pour chacun, ce qui n'est pas toujours évident selon les applications.

Dans ce chapitre, en utilisant un filtrage particulière et l'idée de la confiance du détecteur, nous proposons d'introduire un algorithme MMPF dédié à notre expérience de transport de sédiments par charriage. Il se base sur la technique de modèle dynamique changeant avec plusieurs modèles de mouvement où chaque modèle correspond à un état de mouvement spécifique lié à la mécanique des sédiments en mode charriage : repos, roulement, saltation. L'intérêt de cette approche est : (1) de prendre en compte la mécanique de déplacement des objets grâce à des modèles mécaniques ce qui permet de suivre le comportement naturel des billes sphériques dans notre écoulement expérimental ; (2) de permettre une correction de la mesure en cas d'imprécision ou d'absence de celle-ci ; (3) d'utiliser l'état de mouvement des objets comme une information dans le calcul de vraisemblance des particules et ainsi aider à choisir le modèle de mouvement.

## V.2 Algorithme de suivi multi-objets et multi-modèles

Notre algorithme de suivi multi-objets et multi-modèles est basé sur le principe du filtrage particulière à modèles multiples (MMPF) et une estimation de la confiance du détecteur. Une vue d'ensemble de l'algorithme est présentée en [Figure V.1](#). À chaque pas de temps sont réalisés : (1) la détection des objets, (2) l'estimation de la densité de confiance du détecteur, (3) l'association de chaque détection à un traqueur, (4) la mise à jour de l'état de chaque filtre particulière en fonction du modèle de mouvement spécifique et de la nouvelle détection. Le principe de l'algorithme de suivi MMPF est de prédire l'état de la cible (*i.e.* la position, la vitesse et l'état de mouvement de la cible) en appliquant un modèle de mouvement adapté dépendant du comportement de la cible. Cet état de la cible est ensuite corrigé avec les mesures faites sur l'image observée pour estimer l'état final de cette cible. La cible estimée à l'instant  $t$  est utilisée par la suite pour l'association des données de l'instant  $t + 1$ . Chaque partie de l'algorithme est expliquée ci-après.



**Figure V.1** Vue d'ensemble des différentes parties de l'algorithme de suivi multi-objets.

Pour chaque image, la phase de détection des objets vise à détecter toutes les billes par traitement d'images. La méthode utilisée est la même que celle de l'algorithme déterministe (voir [Chapitre IV.2](#)). Elle retourne donc un ensemble  $D$  de coordonnées  $(x, y)$  du centre de toutes les billes noires et transparentes détectées. Cependant, le détecteur de billes transparentes peut manquer de fiabilité, notamment avec des faux négatifs, à cause de l'occultation partielle de billes. Nous avons alors traité ces cas problématiques à l'aide d'une estimation de la densité de confiance du détecteur (voir [Section V.2.1](#)). Concernant la résolution du problème d'association, la méthode utilisée est la même que celle de l'algorithme déterministe (voir [Section IV.1.2](#)). Nous détaillons la mise à jour des filtres particulières dans la [Section V.2.2](#).

### V.2.1 Estimation de la confiance du détecteur

Pour gérer les détections manquantes de billes transparentes, nous avons introduit une estimation de la confiance du détecteur et l'avons utilisée comme un modèle d'observation [[Breitenstein et al., 2011](#)]. En se basant sur la sortie brute du détecteur (*i.e.* le coefficient de corrélation à un modèle de bille), la confiance du détecteur doit donner une estimation de la vraisemblance d'une détection en chaque pixel. En considérant l'ensemble complet  $D^*$  des maximums locaux en sortie du détecteur (corrélations supérieures à un seuil très bas  $\tau_t^*$ ), chaque détection  $d \in D^*$  possède une position  $x^d$  et un coefficient de corrélation croisée  $\text{xcorr}(d)$  avec un modèle de bille (voir plus de détails sur ce dernier en [Annexe A.2](#)). En supposant que la position d'une détection suive une distribution exponentielle, nous pouvons estimer la vraisemblance d'observer un objet à la position  $x^{obs}$  étant donné l'ensemble des détections  $D^*$  de la manière suivante :

$$\begin{aligned}
 p(\mathbf{x}^{obs}|D^*) &= d_c(\mathbf{x}^{obs}) \\
 &= \frac{1}{k_{D^*}} \sum_{d \in D^*} \text{xcorr}(d) \exp(-\lambda |\mathbf{x}^{obs} - \mathbf{x}^d|)
 \end{aligned} \tag{V.1}$$

où  $|\cdot|$  dénote la norme euclidienne d'un vecteur,  $\lambda$  dénote la constante de l'exponentielle décroissante (voir Section V.2.3), et  $k_{D^*}$  est le coefficient de normalisation.

## V.2.2 Filtrage particulaire à modèle multiples

Chaque traqueur (ou trajectoire) est décrit par un filtre particulaire indépendant à  $N$  particules. À l'instant  $t$ , le vecteur d'état  $\mathbf{c}_t^i$  d'une particule  $i$  est défini par  $\{\mathbf{x}_t^i, \mathbf{u}_t^i, s_t^i\}$  où  $\mathbf{x}_t^i = (x, y)$  dénote la position,  $\mathbf{u}_t^i = (u, v)$  la vitesse et  $s_t^i$  l'état de mouvement. La variable d'observation à l'instant  $t$  est donnée par  $\mathbf{z}_t = \{\mathbf{x}_t^{obs}, \mathbf{u}_t^{obs}, s_t^{obs}\}$  résultant des mesures de position, vitesse et état de mouvement. Nous utilisons le filtre séquentiel à rééchantillonnage par importance (SIR pour *sequential importance resampling*), aussi connu sous le nom de filtre *bootstrap* [Gordon et al., 1993] où la densité de transition d'état est utilisée comme distribution d'importance, pour approximer la fonction de probabilité. Pour une cible, l'algorithme passe par différentes étapes décrites en Figure V.2, soit : (1) mesure de l'état de mouvement de la cible en fonction de la configuration de son voisinage et de sa vitesse, (2) sélection aléatoire des nouveaux états de mouvement selon la table de transition des probabilités conditionnelles, (3) mise à jour de la position et vitesse des particules en fonction des modèles de mouvement, (4) calcul des poids d'importance des particules, normalisation et rééchantillonnage, (5) estimation finale de l'état de mouvement, position et vitesse de la cible. La principale originalité par rapport l'algorithme préliminaire exposé dans Lafaye de Micheaux et al. [2016] est une meilleure prise en considération de la mécanique des objets notamment leur état de mouvement, ce qui est une importante amélioration du modèle d'évolution des états des objets. Pour ce faire, nous utilisons l'estimation de l'état de mouvement de la cible comme une observation dans le calcul du poids (voir Section V.2.2.2).

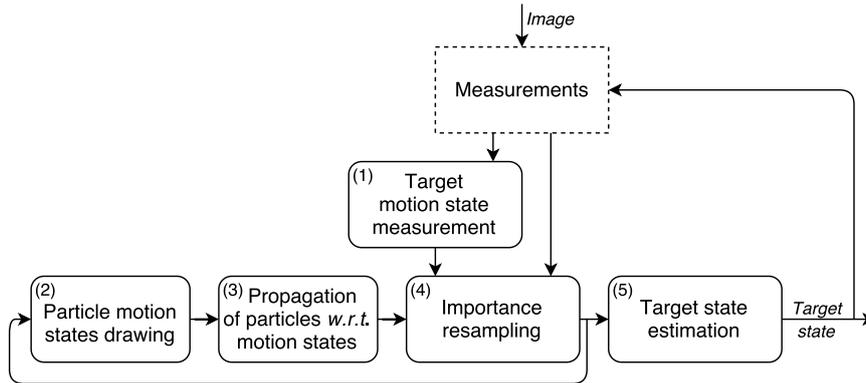


Figure V.2 Algorithme de suivi basé sur un filtrage particulaire à modèles multiples.

### V.2.2.1 Modèles de mouvement basés sur les dynamiques mécaniques

Dans un suivi d'objets stochastique basé sur le mouvement [Odobez et al., 2006], des mesures explicites de mouvement sont utilisées pour guider les prédictions. Dans notre application, les billes ont des comportements différents en fonction de leur position, vitesse et

voisinage. Nous avons donc introduit des modèles de mouvement spécifiques à notre application. Comme énoncé dans les chapitres précédents, nous pouvons distinguer trois états de mouvement distincts liés aux dynamiques dans un transport par charriage : repos (*resting*, pas ou peu de mouvement), roulement (*rolling*, roulement ou glissement sur les autres billes) et saltation (*saltating*, rebond sur les autres billes). Dans l'état de saltation, nous pouvons faire la distinction entre deux sous-états : un mouvement à vitesse constante et un mouvement de rebond. Pour propager une particule  $i$  à l'instant  $t$ , nous mettons d'abord à jour son ancien état de mouvement  $s_{t-1}^i$ , et ensuite nous appliquons un modèle de mouvement spécifique pour mettre à jour sa position et vitesse. Le nouvel état de mouvement  $s_t^i$  est tiré selon la table de transition des probabilités conditionnelles  $\pi$  (voir Section V.2.3). Les modèles de mouvement sont :

$$\begin{aligned}
\text{repos : } (x, y)_t &= (x, y)_{t-1} + \varepsilon_{rest}^{(x,y)} \\
\text{roulement : } (x, y)_t &= (x, y)_{t-1} + (u)_{t-1} \Delta_T + \varepsilon_{roll}^{(x,y)} \\
&(u, v)_t = (u, 0)_{t-1} + \varepsilon_{roll}^{(u)} \\
\text{saltation-constant : } (x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \Delta_T + \varepsilon_{salt}^{(x,y)} \\
&(u, v)_t = (u, v)_{t-1} + \varepsilon_{salt}^{(u,v)} \\
\text{saltation-rebond : } (x, y)_t &= (x, y)_{t-1} + (u, -v)_{t-1} \Delta_T + \varepsilon_{salt}^{(x,y)} \\
&(u, v)_t = (u, -v)_{t-1} + \varepsilon_{salt}^{(u,v)}
\end{aligned} \tag{V.2}$$

où  $\varepsilon_{rest}^{(x,y)}$ ,  $\varepsilon_{roll}^{(x,y)}$ ,  $\varepsilon_{salt}^{(x,y)}$  sont les variances du bruit sur la position et  $\varepsilon_{roll}^{(u)}$ ,  $\varepsilon_{salt}^{(u,v)}$  les variances du bruit sur la vitesse (voir Section V.2.3), elles sont toutes tirées indépendamment avec des distributions normales centrées;  $\Delta_T$  est la période d'échantillonnage temporel. Le modèle de repos n'a pas de terme de vitesse étant donné qu'il décrit les billes à l'arrêt; le modèle de roulement ne présente que la composante  $u$  (horizontale) de la vitesse étant donné que les billes qui roulent le font généralement dans le sens longitudinal de l'écoulement; les modèles de saltation présentent les deux composantes  $(u, v)$  et se différencient entre constant et rebond sur la composante  $v$  (verticale) de la vitesse dont le signe est modifié dans la saltation-rebond pour conserver la norme de vitesse mais changer la direction.

### Mesure de la vitesse et de l'état de mouvement

Avant d'appliquer le filtrage *bootstrap* sur une cible, il est possible de faire une estimation de son état de mouvement en tenant compte des observations. La mesure de cet état de mouvement est calculée en examinant la configuration du voisinage et la vitesse de la cible. La vitesse de la cible est considérée comme une observation, elle est calculée sur l'hypothèse d'association obtenue à l'issue du module "data association".

L'idée est d'utiliser notre connaissance de la mécanique des déplacements en transport par charriage pour satisfaire au mieux les comportements mécaniques de nos objets. Étant donné le nombre de billes voisines  $\text{nbN}^{obs}$  en contact avec la bille observée, nous pouvons considérer que l'état de mouvement est au repos si  $\text{nbN}^{obs}$  est significativement élevé ou si la norme de vitesse de la bille  $|\mathbf{u}^{obs}|$  est faible. De la même façon, l'état de mouvement est en saltation si  $\text{nbN}^{obs}$  est faible ou  $|\mathbf{u}^{obs}|$  est élevé. Formellement, nous avons :

$$s^{obs} = \begin{cases} \text{repos,} & \text{si } \text{nbN}^{obs} \geq \text{nbN}_{rest} \text{ ou } |\mathbf{u}^{obs}| \leq u_{rest}, \\ \text{saltation,} & \text{si } \text{nbN}^{obs} \leq \text{nbN}_{salt} \text{ ou } |\mathbf{u}^{obs}| \geq u_{salt}, \\ \text{roulement,} & \text{sinon.} \end{cases} \tag{V.3}$$

où  $\text{nbN}_{rest}$  (resp.  $\text{nbN}_{salt}$ ) est le nombre seuil de voisins pour l'état repos (resp. saltation) et  $u_{rest}$  (resp.  $u_{salt}$ ) est la norme seuil de vitesse pour l'état repos (resp. saltation). Ces valeurs sont déterminées expérimentalement sur une séquence test (voir Section V.2.3).

L'estimation de l'état de mouvement est utilisée comme une indication dans le calcul des poids d'importance (voir Section V.2.2.2) et pour fixer l'état de mouvement final de la cible (voir Section V.2.2.3).

### V.2.2.2 Poids d'importance des particules

#### Modèle d'observation

Le poids d'importance  $w_t^i$  d'une particule  $i$  à l'instant  $t$  est mis à jour à partir de sa valeur à l'instant précédent  $w_{t-1}^i$  et de la vraisemblance conditionnelle de la nouvelle observation  $z_t$  étant donné l'état précédent  $c_{t-1}^i$  de la particule :

$$w_t^i \propto w_{t-1}^i p(z_t | c_{t-1}^i) \quad (\text{V.4})$$

Grâce au rééchantillonnage réalisé à chaque pas de temps, tous les poids sont les mêmes à la fin et  $w_{t-1}^i = 1/N$ , donc ce terme peut être ignoré. De plus, en pratique, la distribution de probabilité *a priori* est généralement choisie comme étant la distribution d'importance afin d'approximer la fonction de densité de probabilité [Gordon et al., 1993]. Avec ces simplifications appliquées à Eq. (V.4), le calcul du poids d'importance d'une particule ne dépend que de l'état de la particule propagée :

$$w_t^i \propto p(z_t | c_t^i) \quad (\text{V.5})$$

#### Calcul des poids d'importance

En supposant que les variables d'observation soient indépendantes les unes des autres, nous avons :

$$\begin{aligned} p(z_t | c_t^i) &= p(\{\mathbf{x}_t^{obs}, s_t^{obs}\} | c_t^i) \\ &= p(\mathbf{x}_t^{obs} | c_t^i) p(s_t^{obs} | c_t^i) \end{aligned} \quad (\text{V.6})$$

Pour les billes transparentes, il n'y a pas toujours une détection associée à un traqueur. Dans ce cas, la vraisemblance de la particule  $i$  à la position  $\mathbf{x}_t^i$  est donnée par la confiance du détecteur. Sinon, elle est estimée par une distribution normale centrée sur la détection correspondante  $d$  (voir Section V.2.3 pour la valeur de la variance). Nous avons alors :

$$p(\mathbf{x}_t^{obs} | c_t^i) = \begin{cases} p_{\mathcal{N}}(\mathbf{x}_t^i - \mathbf{x}_t^{obs}) & \text{si } \mathcal{I}(tr) = 1, \\ d_c(\mathbf{x}_t^i) & \text{si } \mathcal{I}(tr) = 0. \end{cases} \quad (\text{V.7})$$

où  $\mathcal{I}(tr)$  est une fonction indicatrice qui retourne 1 si une détection est associée au traqueur et 0 sinon. Cette indication est donnée par l'association des données.

Pour la vraisemblance de l'état de mouvement  $p(s_t^{obs} | c_t^i)$ , nous supposons qu'elle vaut la valeur constante  $\gamma$  si  $s_t^i = s_t^{obs}$  et  $1 - \gamma$  sinon. Cette valeur est fixée expérimentalement (voir Section V.2.3). Nous avons donc :

$$p(s_t^{obs} | c_t^i) = p_\gamma(s_t^i) = \begin{cases} \gamma, & \text{si } s_t^i = s_t^{obs}, \\ 1 - \gamma, & \text{sinon.} \end{cases} \quad (\text{V.8})$$

Ensuite, la mise à jour des poids est donnée par :

$$w_t^i = \begin{cases} p_\gamma(s_t^i) p_{\mathcal{N}}(\mathbf{x}_t^i - \mathbf{x}_t^{obs}) & \text{si } \mathcal{I}(tr) = 1, \\ p_\gamma(s_t^i) d_c(\mathbf{x}_t^i) & \text{sinon (pour les billes transparentes),} \\ p_\gamma(s_t^i) & \text{sinon (pour les billes noires).} \end{cases} \quad (\text{V.9})$$

Le dernier cas de Eq. (V.9) correspond à une bille noire n'ayant pas de détection associée. Comme il n'y a pas de confiance du détecteur pour les billes noires, il n'y a pas de terme de poids basé sur la position. Cependant, ce dernier cas ne se produit que très rarement étant donné que l'association des données est très efficace quand elle est appliquée sur des détecteurs fiables comme celui des billes noires.

### Normalisation et rééchantillonnage

Après avoir pondéré toutes les particules, les poids sont normalisés tels que  $\sum_{i=1}^N w_t^i = 1$ . Ensuite, le rééchantillonnage est utilisé pour résoudre la dégénérescence des particules. Il consiste à supprimer les particules ayant un poids faible et multiplier celles ayant un poids fort. L'approche utilisée ici est un rééchantillonnage multinominal basé sur la méthode *bootstrap* [Efron and Tibshirani, 1994].

#### V.2.2.3 Estimation de l'état final

Pour finir, la position et la vitesse de la cible sont estimées en moyennant les particules rééchantillonnées. Après que toutes les positions et vitesses des cibles aient été estimées, nous estimons également leur état de mouvement de la même façon que pour obtenir l'état de mouvement observé (voir Section V.2.2.1 et Eq. (V.3)). La raison pour laquelle nous réévaluons l'état de mouvement est que nous voulons être cohérent avec l'état de mouvement mécanique. En effet, prendre l'état de mouvement le plus représenté des particules rééchantillonnées ne garantit pas un état de mouvement compatible avec le comportement dynamique de la cible.

### V.2.3 Détails d'implémentation

#### Paramètres d'équations

Le nombre de particules est fixé à  $N = 100$ . Cette valeur est un bon compromis entre temps de calcul et fiabilité des résultats du suivi. Le paramètre  $\lambda$  dans Eq. (V.1) est fixé expérimentalement<sup>1</sup> et reste le même pour les différentes séquences de test utilisées (voir Section IV.4.1), tout comme les paramètres de la fonction du coût d'association (voir leur valeur en Section IV.1.2.2). La variance de la distribution normale dans Eq. (V.7) est fixée expérimentalement à 3 pixels pour les coordonnées  $x$  et  $y$ . Le paramètre de vraisemblance de l'état de mouvement  $\gamma$  dans Eq. (V.8) est fixé à 2/3 dans le but de promouvoir les particules ayant le même état de mouvement que la mesure.

#### Paramètres de bruit et de traqueur

Chaque état de mouvement dans Eqs. (V.2) a sa propre variance de bruit sur la position et la vitesse. Elles sont fixées automatiquement en analysant une séquence de test<sup>1</sup>. Les

1. Nous avons utilisé une séquence de test extraite de la séquence expérimentale qui est distincte de la séquence sur laquelle nous avons réalisé les évaluations.

positions et vitesses initiales des particules sont tirées selon une distribution normale avec les positions centrées sur la détection correspondante et les vitesses centrées sur la vitesse nulle. Pour gérer les difficultés liées à l'arrivée de nouveaux traqueurs, nous avons augmenté les variances pour rendre les modèles de mouvement plus flexibles pendant les 3 premières images. Pour arrêter les mauvaises prédictions, un traqueur au repos (resp. roulement et saltation) ne survit que 100 images (resp. 10 images et 5 images) sans être associé à une détection et est ensuite automatiquement stoppé.

### Table de probabilité de transition

Pour une particule  $i$  donnée, la transition d'un état de mouvement  $s_{t-1}^i$  à un autre  $s_t^i$  est contrôlée par des probabilités conditionnelles. Nous avons estimé ces probabilités sur une séquence de test<sup>1</sup>. Comme les probabilités sont quasiment les mêmes pour toutes les séquences, nous ne présentons qu'un seul cas ici. La table de probabilité de transition  $\pi$  requise pour le processus de changement d'état présenté en Section V.2.2.1 est estimée par :

$$\pi = p(s_t^i | s_{t-1}^i) = \begin{bmatrix} 0.95 & 0.05 & 0.00 \\ 0.07 & 0.92 & 0.01 \\ 0.00 & 0.09 & 0.91 \end{bmatrix} \quad (\text{V.10})$$

Étant donné que  $N$  est fixé à 100 et que nous souhaitons que chaque état de mouvement ait une chance d'être représenté à chaque pas de temps, nous avons modifié la table de probabilité de transition pour être légèrement moins restrictive en imposant une probabilité minimale de 0.05. Nous avons alors :

$$\pi \approx \begin{bmatrix} 0.80 & 0.15 & 0.05 \\ 0.10 & 0.80 & 0.10 \\ 0.05 & 0.15 & 0.80 \end{bmatrix} \quad (\text{V.11})$$

L'état saltation étant divisé en deux sous-états, quand il est tiré après être passé dans  $\pi$ , il y a ensuite une probabilité de 0.5 d'être dans le sous-état saltation-constant (resp. saltation-rebond). À l'initialisation, les états de mouvement sont fixés comme des états "inconnus". Au pas de temps suivant, les particules ont une chance équivalente d'aller dans n'importe lequel des états de mouvement.

### Seuils de la mesure des états de mouvement

Pour mesurer l'état de mouvement d'une cible (Eq. (V.3)), nous avons besoin de calculer son nombre de voisins. Nous considérons une détection comme une voisine de la cible si les deux sont en contact, c'est-à-dire quand la distance entre les deux centres est égale à la somme de leur rayon. Les nombres seuils de voisins pour les états repos et saltation sont fixés expérimentalement<sup>1</sup> à  $\text{nbN}_{rest} = 5$  et  $\text{nbN}_{salt} = 0$  respectivement. Les seuils sur les normes de vitesse sont fixés expérimentalement<sup>1</sup> à  $u_{rest} = 0.015\text{m/s}$  et  $u_{salt} = 0.25\text{m/s}$  sur les séquences étudiées dans ce travail (voir Section IV.4.1).  $u_{rest}$  et  $u_{salt}$  sont spécifiques à chaque séquence. Ils diffèrent légèrement des valeurs de l'algorithme déterministe car la mesure se fait en ligne et non *a posteriori*, ils doivent donc être plus permissifs. Ces quatre seuils sont ajustés expérimentalement<sup>1</sup> sur la séquence de test pour minimiser les différences entre les états de mouvement déterminés par l'algorithme et ceux à l'oeil nu.

## V.3 Résultats et discussions

Dans cette section, nous présentons les résultats de l’algorithme de suivi proposé en le comparant à d’autres algorithmes de la littérature. Ils sont tous évalués sur nos propres vérités terrains ExpSeq et NumSeq (voir Section IV.4.1). En effet, comme notre étude de cas est très spécifique, il est impossible de tester notre algorithme de suivi de billes sur les jeux de données ouverts au public. Nous utilisons également les métriques CLEAR MOT [Bernardin and Stiefelhagen, 2008] pour évaluer les performances du suivi. De plus, nous étudions comment les algorithmes se comportent avec des détecteurs de différentes qualités et avec quelle précision ils estiment les états de mouvement.

### V.3.1 Algorithmes de comparaison

Pour étudier la fiabilité et l’efficace de notre algorithme, nous le testons face à trois autres algorithmes. Premièrement, il y a notre algorithme de suivi par filtre particulaire décrit ci-avant qui est basé sur trois modèles (ou états) de mouvement, la confiance du détecteur et une mesure de l’état de mouvement guidant la vraisemblance (MMPF+). Deuxièmement, il y a notre travail préliminaire [Lafaye de Micheaux et al., 2016] sur un filtrage particulaire basé sur trois modèles de mouvement et la confiance du détecteur (MMPF). Troisièmement, il y a une adaptation de l’algorithme de Breitenstein et al. [2011] basé sur un modèle unique à vitesse constante et un filtrage particulaire avec confiance du détecteur (PF). Et enfin, il y a notre algorithme déterministe (décrit dans le Chapitre IV) qui se fie à la sortie du détecteur et ne présente pas de traitement particulier pour les faux positifs et faux négatifs (DET).

### V.3.2 Évaluation de l’algorithme de suivi

Pour évaluer les performances du suivi, nous utilisons les mêmes métriques que celles énoncées dans le Chapitre IV, à savoir les taux de faux positifs et faux négatifs, le nombre d’échanges d’identité de traqueur, le pourcentage de trajectoires correctes, le MOTP et le MOTA. Dans l’optique d’analyser l’influence de la sortie du détecteur sur les différents algorithmes de suivi, nous faisons varier le seuil de détection des billes transparentes. Cela permet d’avoir différentes qualités de détecteur et de voir comment les algorithmes traitent plus de faux positifs et faux négatifs. Pour finir, nous étudions avec quelle précision les algorithmes basés sur un filtrage particulaire à modèles multiples estiment l’état de mouvement des billes.

#### V.3.2.1 Analyse quantitative

Le Tableau V.1 présente les résultats d’évaluation de performances des quatre algorithmes de suivi MMPF+, MMPF, PF et DET sur les deux jeux de données ExpSeq et NumSeq. Il montre que la plupart du temps notre méthode MMPF+ est plus performante que les trois autres. Particulièrement, l’algorithme MMPF+ a de meilleurs résultats que les algorithmes à filtrage particulaire MMPF et PF dans toutes les métriques. Par exemple, nos valeurs de MOTP et MOTA sont supérieures sur les deux jeux de données. Il faut tout de même noter que toutes les valeurs sont déjà assez élevées. Cela est dû à la bonne qualité des détecteurs de billes transparentes pour les seuils choisis. Nous avons choisi les seuils  $\tau_t = 0.25$  pour ExpSeq et  $\tau_t = 0.27$  pour NumSeq car ils donnent les meilleurs résultats de suivi et nous savons l’importance de bons détecteurs pour avoir un suivi fiable. Cependant,

le détecteur n'est pas toujours aussi efficace car il dépend de la qualité des images. L'analyse de l'effet de différentes valeurs de  $\tau_t$  est faite en Section V.3.2.2.

Jeu données	Algo.	Tr. Correct	MOTP	FN	FP	Éch. ID	MOTA
ExpSeq	MMPF+	<b>99.57%</b>	<b>98.40%</b>	<b>0.21%</b>	0.27%	<b>9</b>	<b>99.52%</b>
	MMPF	99.36%	97.96%	0.23%	0.34%	9	99.42%
	PF	99.15%	97.29%	0.24%	0.36%	13	99.39%
	DET	99.57%	-	0.30%	<b>0.21%</b>	34	99.47%
NumSeq	MMPF+	<b>98.50%</b>	97.38%	<b>0.14%</b>	0.61%	<b>85</b>	<b>99.24%</b>
	MMPF	98.50%	97.21%	0.15%	0.66%	111	99.18%
	PF	97.59%	96.40%	0.15%	0.66%	91	99.19%
	DET	98.05%	<b>97.62%</b>	0.17%	<b>0.59%</b>	145	99.23%

**Tableau V.1** Évaluation des performances des quatre algorithmes de suivi : notre MMPF+, le MMPF de Lafaye de Micheaux et al. [2016], le PF adapté de Breitenstein et al. [2011], le DET du Chapitre IV sur les deux jeux de données ExpSeq et NumSeq. Les résultats montrent les valeurs des mesures d'évaluation : pourcentage de trajectoires correctes (Tr. Correct.) (une trajectoire est considérée correcte si sur plus de 95% du temps elle n'a ni faux négatif et échange d'ID), taux de faux négatifs (FN), taux de faux positifs (FP), nombre d'échanges d'ID de traqueur (Éch. ID) et les métriques CLEAR MOT [Bernardin and Stiefelhagen, 2008] d'exactitude du suivi (MOTA) et de précision des positions (MOTP). Le meilleur score de chaque métrique apparaît en gras.

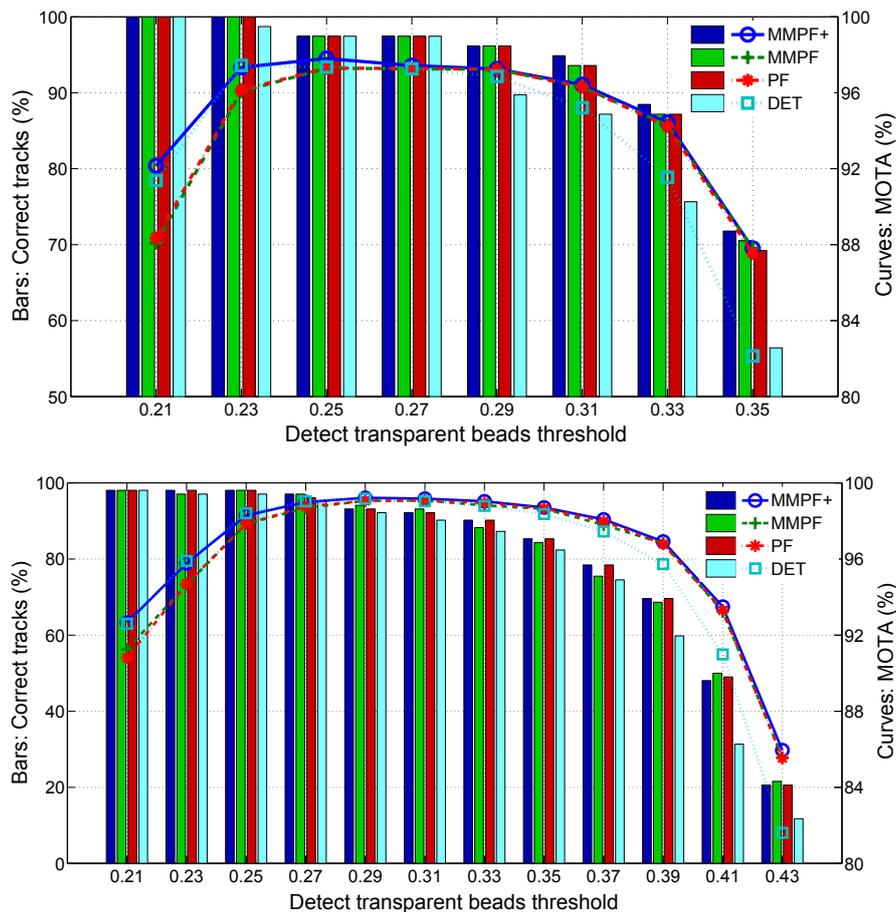
Nous pouvons observer que le nombre d'échanges d'ID de l'algorithme MMPF+ est le plus faible. Cette propriété est une des plus importantes pour un bon algorithme de suivi. Dans nos images, des échanges d'ID peuvent intervenir lorsque des trajectoires changent subitement de direction (lorsque les billes rebondissent par exemple) ou de vitesse (quand accélération  $\gg 0$ ), ou quand de nouvelles billes apparaissent très près de la position prédite d'une trajectoire selon son modèle de mouvement. L'implémentation de plusieurs modèles de mouvement dans notre méthode permet de mieux gérer ces situations, ce qui était une des motivations de l'élaboration de l'algorithme MMPF+.

En terme de taux de faux négatifs, notre algorithme surpasse les autres également. En effet, en l'absence de détection associée à un traqueur, l'algorithme DET stoppe immédiatement le traqueur correspondant, ce qui entraîne un faux négatif, contrairement aux algorithmes à filtrage particulaire qui estiment sa position à l'aide de son comportement précédent. Notre algorithme est meilleur que le PF car il utilise plusieurs modèles de mouvement adaptés pour prédire les positions. Et il est également meilleur que le MMPF car il estime les états de mouvement de manière plus conforme au comportement réel des billes.

Néanmoins, l'approche déterministe semble être la meilleure en terme de taux de faux positifs. Cela vient principalement des terminaisons de traqueurs en bord d'image (*i.e.* effets de bord). En effet, dans les algorithmes à filtrage particulaire, quand il n'y a pas de détection associée à un traqueur parce que la bille a quitté le champ de vision, le traqueur essaie d'estimer son nouvel état. Si sa position estimée est toujours dans le champ de vision, le traqueur continue l'estimation de l'état de l'objet, ce qui résulte en un faux positif. Il survit ainsi sur un nombre limité d'images sans détections associées et est ensuite automatiquement terminé (voir Section V.2.3 pour les temps de survie). Au contraire, l'algorithme DET stoppe immédiatement un traqueur quand il n'a pas de détection associée.

### V.3.2.2 Impact de la qualité du détecteur

Afin d'étudier les performances du suivi avec différents niveaux de détection, nous faisons varier le seuil  $\tau_t$  du détecteur de billes transparentes. Cela permet d'évaluer comment la qualité du détecteur influence le pourcentage de trajectoires correctes et le MOTA sur les billes transparentes. Nous avons réalisé cette analyse sur les quatre algorithmes et sur les deux jeux de données ExpSeq (Figure V.3, en haut) et NumSeq (Figure V.3, en bas).



**Figure V.3** Évaluation des performances du suivi des billes transparentes sur les quatre algorithmes MMPF+, MMPF, PF et DET, sur les deux jeux de données ExpSeq (graphique du haut) et NumSeq (graphique du bas), et pour différentes qualités du détecteur de billes transparentes lorsque l'on fait varier les seuils de détection. Le pourcentage de trajectoires correctes (barres verticales, échelles de gauche) et le MOTA (courbes, échelles de droite) sont tracés.

Les résultats montrent que notre algorithme est globalement plus performant que les trois autres en terme de MOTA pour chaque jeux de données. En effet, nous pouvons observer que pour des seuils faibles (plus de faux positifs) la valeur du MOTA est la plus haute pour l'algorithme MMPF+, tout comme pour les seuils élevés (plus de faux négatifs). Ceci vient de la bonne gestion des erreurs de détection : stopper rapidement les traqueurs faussement créés par des faux positifs et continuer la prédiction des traqueurs sans billes associées quand il y a des détections manquantes.

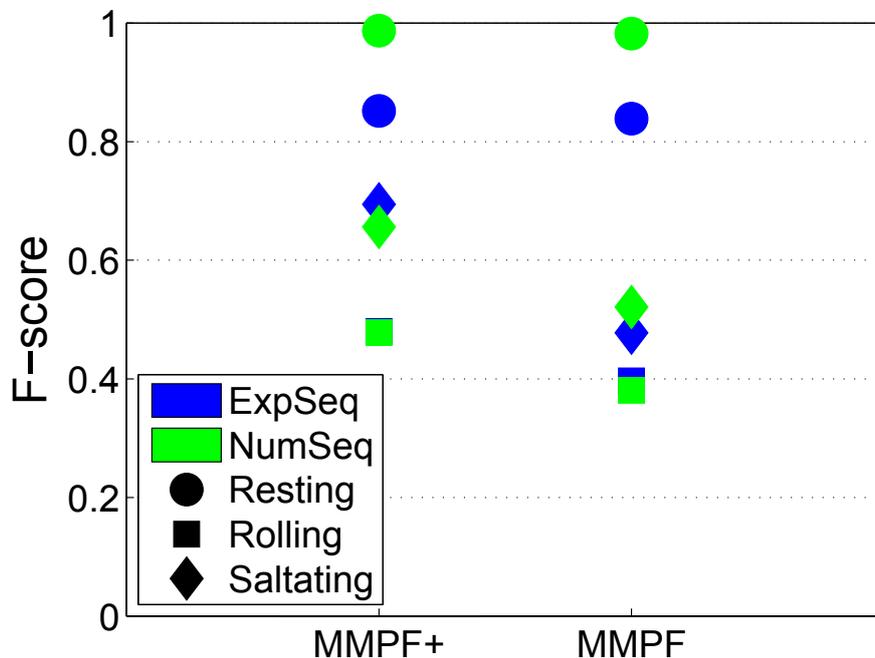
Pour chaque algorithme, les performances du suivi dépendent de la qualité du détecteur :

plus  $\tau_t$  est élevé, plus le pourcentage de trajectoires correctes est faible. En particulier, l'algorithme DET qui ne présente pas d'étape de prédiction est le plus impacté par les fortes valeurs de  $\tau_t$ , comme on pouvait s'y attendre.

### V.3.2.3 Estimation des états de mouvement

Les états de mouvement des traqueurs ne sont pas uniquement utilisés pour fixer spécifiquement les modèles de mouvement des filtres particulaires. Ils sont aussi utilisés pour étudier le comportement des traqueurs dans chaque état de mouvement. Le but est d'analyser, pour chaque état de mouvement séparément, le nombre de traqueur à chaque pas de temps, les profils de vitesse, l'infiltration et la concentration dans le lit mobile. Pour une analyse fiable, nous avons besoin d'estimer les états de mouvement au plus proche des états mécaniques. Nous les comparons donc aux états de mouvement des vérités terrains.

La Figure V.4 montre le F-score de l'identification des états obtenus avec notre algorithme par rapport à l'algorithme MMPF pour les deux jeux de données ExpSeq et NumSeq. Pour un état de mouvement, le F-score est une mesure de justesse de classification dans cet état de mouvement qui considère la précision et le rappel. La précision est la proportion d'instances retournées qui sont pertinentes et le rappel est la proportion d'instances pertinentes qui sont retournées. Nous avons choisi le F-score plutôt que l'exactitude car l'exactitude amène à des résultats qui peuvent induire en erreur quand le jeu de données n'est pas équilibré, c'est-à-dire quand le nombre d'échantillons dans les différentes classes varie beaucoup, comme c'est le cas dans nos jeux de données avec approximativement 2 fois moins de billes en roulement et 20 fois moins de billes en saltation que de billes en repos.



**Figure V.4** Évaluation de la fiabilité de l'état de mouvement des traqueurs donnée par les F-scores pour les deux jeux de données ExpSeq et NumSeq et pour les trois états de mouvement repos, roulement et saltation.

Premièrement, nous pouvons observer que les résultats sont relativement proches pour les

deux jeux de données. Cela confirme que le jeu de données numérique peut être utilisé pour évaluer les algorithmes de suivi à modèles multiples. Deuxièmement, l'algorithme MMPF+ est plus performant que le MMPF dans tous les états ce qui confirme sa meilleure capacité à prédire l'état de mouvement. En particulier, la précision des billes en saltation est supérieure de 0.2 pour le MMPF+, notamment grâce à l'introduction des deux sous-états de la saltation "constant" et "rebond" (voir Section V.2.2.1). Troisièmement, les deux algorithmes sont plus ou moins performants en fonction de l'état de mouvement. En effet, pour l'algorithme MMPF+ par exemple, les billes en repos sont très bien classifiées ( $> 0.85$ ) en comparaison à celles en saltation ( $\approx 0.70$ ) et en roulement ( $\approx 0.45$ ). Cela n'est pas surprenant étant donné que les billes en repos sont facilement classifiables grâce à leur faible vitesse et leur grand nombre de voisins. Au contraire, les billes en roulement peuvent être confondues avec des billes en repos quand elles ont une faible vitesse ou à des billes en saltation quand elles bougent vite sur le haut du lit.

## V.4 Éléments de discussion complémentaires

Bien que notre algorithme de suivi par filtrage particulière ait l'avantage de corriger les erreurs de détection, il présente quelques points qu'il faudrait améliorer pour pallier certains problèmes. Nous discutons ici de ces éléments limitants et donnons quelques pistes d'amélioration.

### Difficultés lors de la terminaison des trajectoires

Comme nous l'avons vu dans l'analyse quantitative ci-avant, le suivi par filtrage particulière présente des difficultés pour terminer les trajectoires lorsque celles-ci sont proches de sortir du champ de l'image. En effet, il peut arriver qu'un traqueur continue d'estimer son état alors que la trajectoire en question a déjà quitté l'image. Ceci provoque alors une erreur de type faux positif. Dans ce type de situation, le suivi déterministe apparaît plus efficace car il stoppe automatiquement le traqueur lorsque la trajectoire sort de l'image. Il faudrait peut-être ici gérer les bords d'images de la même manière que le suivi déterministe.

### Manque de précision dans l'estimation des positions

Le suivi par filtrage particulière a permis de corriger les erreurs de détection sans perdre significativement en précision de positions par rapport au suivi déterministe. Le fait de ne pas avoir gagné en précision de positions vient du fait que les nombreux paramètres de l'algorithme à filtrage particulière pourraient être mieux ajustés à la séquence d'images (comme les paramètres influents et difficiles à régler que sont  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\pi$  par exemples). Pour améliorer ce point, on pourrait envisager des méthodes d'estimation automatique des paramètres.

### Temps de calcul assez long

D'autre part, le temps de calcul du suivi par filtrage particulière est beaucoup plus long que le déterministe. En effet, nous avons observé que le suivi déterministe s'exécute environ 9 fois plus rapidement que le suivi par filtrage particulière (pour une séquence de 5000 images, le suivi déterministe met 1h40 contre 15h25 pour le filtrage particulière). Pour réduire le temps d'exécution du filtrage particulière, il peut être envisagé d'écrire certaines parties de l'algorithme en C et d'appeler une fonction mex depuis Matlab, voire d'utiliser un GPU.

**Estimation en ligne des états de mouvement pas optimale**

Les états de mouvement estimés en ligne présentent des erreurs, notamment pour l'état de roulement qui peut être confondu avec les états repos et saltation. L'estimation des états de mouvement se base actuellement sur deux critères : un sur la vitesse et l'autre sur la distance aux voisins. Une piste d'amélioration serait de considérer également la profondeur de la bille dans le lit ou bien l'état de mouvement de ses voisins.

**V.5 Conclusion**

Dans ce chapitre, nous avons présenté un nouvel algorithme de suivi en ligne par filtrage particulaire basé sur des modèles dynamiques multiples. Le but de cet algorithme est de calculer les trajectoires de sédiments dans un canal expérimental reproduisant du charriage avec plus de robustesse aux erreurs de détection que le suivi déterministe (Chapitre IV). Premièrement, ce nouvel algorithme utilise plusieurs modèles de mouvement pour guider les traqueurs avec des prédictions adaptées. Nous sommes alors capable d'estimer la trajectoire des objets de manière plus précise grâce à des informations *a priori* sur leur mécanique de déplacement. Ces informations sont basées sur les trois modes de mouvement mécanique que l'on observe lors d'un charriage, à savoir la saltation, le roulement et le repos. Deuxièmement, cet algorithme gère les éventuelles erreurs de détection comme les faux positifs et faux négatifs en utilisant une estimation de la confiance du détecteur. Cette estimation est utilisée pour guider les particules du filtre quand le détecteur ne retourne pas de détection avec une confiance élevée. Troisièmement, l'algorithme utilise l'état de mouvement des objets comme une information dans le calcul de vraisemblance des particules pour aider à choisir le modèle de mouvement.

Grâce aux mêmes vérités terrains que celles décrites en Section IV.4.1, nous avons montré que notre algorithme de suivi par filtrage particulaire surpasse les autres algorithmes de suivi de l'état de l'art, et notamment le suivi déterministe, en terme de précision de trajectoires. De plus, il est moins impacté par les détections manquantes et présente moins d'échanges d'identité de traqueur que les autres quand il est appliqué sur des détecteurs de moins bonne qualité. D'autre part, notre algorithme de suivi permet d'obtenir une bonne estimation des états de mouvement grâce à l'étude de la configuration du voisinage et de la vitesse des objets.

En perspective, les points qui pourraient être améliorés sont la terminaison des trajectoires, la précision de l'estimation des positions, le temps de calcul et l'estimation des états de mouvement.

# Conclusion générale et perspectives

## Sommaire

---

1	Contributions et résultats principaux . . . . .	97
2	Perspectives . . . . .	99

---

Le contexte applicatif de ces travaux est l'étude expérimentale de la dynamique du transport de particules sédimentaires dans un écoulement et plus généralement l'étude des processus physiques liés au charriage dans les torrents. Pour l'étude du charriage et l'amélioration de sa modélisation, les chercheurs d'Irstea ont réalisé un dispositif expérimental idéalisant le transport de sédiments dans un cours d'eau à forte pente. Ils utilisent pour cela un canal incliné dans lequel les débits solides et liquides sont maintenus constants à l'entrée. Les particules sédimentaires, en mouvement ou formant le lit, sont filmées et l'évolution de la mobilité du lit et des trajectoires sont calculées. L'objectif principal de notre travail est de donner des méthodes de traitement d'images pour analyser les données issues du canal expérimental.

## 1 Contributions et résultats principaux

Les contributions de ce travail ont été guidées par trois objectifs : (1) optimisation du suivi déterministe pour permettre l'étude de longues séquences d'images issues du canal étroit, constitution de vérités terrains et évaluation des algorithmes, (2) développement d'un suivi probabiliste à base de filtrage particulaire pour une étude plus robuste, (3) développement d'un algorithme de traitement d'images pour une étude de l'infiltration de particules fines dans un canal plus large.

### Optimisation d'un algorithme déterministe de suivi

La première contribution a été de rendre possible l'analyse du phénomène de ségrégation en charriage dans sa globalité. En effet, il est important de considérer ce phénomène dans toute sa durée étant donnée sa complexité. Pour ce faire, nous avons optimisé un algorithme déterministe de suivi précédemment développé dans notre laboratoire et avons évalué ses performances grâce à des vérités terrains que nous avons conçues. Cet algorithme permet de calculer les trajectoires de sédiments. Il se base sur une approche en deux étapes : la détection des objets dans les images puis leur suivi dans le temps. Les optimisations apportées s'appuient sur trois idées de raffinement et d'amélioration des algorithmes de Böhm et al. [2006] et Hergault et al. [2010] pour gagner en précision et permettre l'exécution du code

sur une grande quantité de données (jusqu'à 500000 images) :

1. Les méthodes de traitement d'images des détecteurs ont été améliorées pour gagner en précision et robustesse. Avec l'utilisation d'opérations morphologiques plus intuitives et mieux adaptées à l'apparence des billes dans les images, les erreurs de détection ont été réduites. Les détecteurs sont ainsi plus fiables et robustes face aux problèmes classiques d'un suivi d'objets comme les fausses détections et les occlusions.
2. Le code a été réécrit en bibliothèque de fonctions Matlab pour plus de cohérence entre le traitement (qui était avant en C) et l'analyse des trajectoires (qui était en Matlab). De plus, les données de calcul ont été restructurées pour gagner de l'espace mémoire et mieux gérer cet espace lors du traitement de 500000 images.
3. Le code a été organisé de manière à rendre son exécution parallèle et ainsi à réaliser le traitement des images dans un temps plus raisonnable qu'en série. Ceci a été fait en exploitant les fonctions de calcul multi-tâches intégrées à Matlab. Le gain de temps (avec un ordinateur à 2 processeurs de 6 coeurs) sur l'étape de détection est estimé à un facteur d'environ 7 par rapport à une exécution en série.
4. Des vérités terrains ont été créées pour permettre d'évaluer les performances de l'algorithme de suivi de billes et juger de la qualité des résultats. Une séquence expérimentale issue du canal a permis d'élaborer une vérité terrain de 1000 images, et une séquence simulée par un modèle numérique de transport par charriage une vérité terrain de 10000 images.

### **Développement d'un algorithme à filtrage particulière**

La deuxième contribution a été le développement d'un algorithme de suivi probabiliste plus précis que le suivi déterministe. L'algorithme de suivi déterministe, bien qu'adapté à l'étude de longues séquences, peut manquer de précision localement, et notamment lorsque le régime transitoire de la ségrégation en charriage est fort. En effet, la détection a bien été améliorée dans le point précédent mais certains objets des images, les billes transparentes par exemple, sont toujours délicats à détecter de manière exhaustive et exclusive. Nous avons alors conçu un algorithme basé sur un filtrage particulière à modèles dynamiques multiples pour gérer les éventuelles erreurs de détection. Le filtrage corrige les erreurs et les modèles dynamiques guident la correction. Ces modèles sont fondés sur les connaissances que nous avons des déplacements et mouvements mécaniques des sédiments, comme les modes saltation et roulement. Ce nouvel algorithme a été comparé à l'algorithme déterministe et à deux autres algorithmes de l'état de l'art. Les évaluations quantitatives ont montré qu'il apporte un gain de précision sur les trajectoires et une robustesse aux erreurs que le détecteur est susceptible de provoquer notamment lorsque le taux d'erreurs du détecteur est élevé. Il permet en outre une estimation en ligne de l'état de mouvement des objets, évitant d'avoir recours à un post-traitement coûteux en temps de calcul.

### **Traitement d'images pour l'étude de l'infiltration**

La troisième contribution a été le développement d'un algorithme de traitement d'images qui permette d'étudier les images de ségrégation issues du canal élargi. Le canal élargi est destiné à étudier l'infiltration de particules fines dans un lit mobile. Les méthodes que nous utilisons sont des opérations morphologiques liées à la segmentation d'images. Grâce à des transformations par ligne de partage des eaux, nous détectons des lignes caractéristiques

dans les images comme la ligne du lit et la ligne d'eau. Avec ces lignes, nous calculons des hauteurs d'eau et des pentes de lit, qui permettent de faire l'analyse temporelle de l'évolution de la mobilité du lit pendant l'infiltration de particules fines.

## 2 Perspectives

Nous nous questionnons ici sur les perspectives de notre travail en terme de traitement d'images et sur les évolutions qu'il serait utile de conduire par la suite. Comment améliorer les résultats obtenus? Comment réduire le temps d'exécution des algorithmes de traitement? Nous présentons nos suggestions point par point ci-après, dont certaines ont déjà été évoquées dans les chapitres de la thèse.

### **Détection des billes transparentes par apprentissage**

La détection des billes transparentes est l'étape de l'algorithme de suivi la plus problématique. En effet, l'apparence des billes transparentes dans les images peut être modifiée à cause des occlusions partielles qui se produisent dans le canal. Cependant, ces occlusions ont généralement des impacts similaires sur les apparences des billes transparentes, et quelques motifs reviennent régulièrement. Il pourrait être envisagé d'utiliser une classification supervisée pour les détecter en apprenant au préalable une collection de toutes les formes d'apparence que peut avoir une bille transparente, avec et sans occlusions. En comparant les images avec la base de formes alors apprises, la détection serait plus fiable.

### **Parallélisation complète de l'algorithme de suivi**

Les étapes les plus coûteuses de l'algorithme de suivi sont l'étape de résolution du problème d'association de données et l'estimation de états de mouvement. Pour limiter l'impact de leur longue exécution sur l'algorithme complet, il serait intéressant de couper la séquence en plusieurs morceaux et exécuter l'algorithme de suivi en parallèle sur plusieurs morceaux. Par exemple, avec 5 coeurs à disposition et une séquence à traiter de 10000 images, la séquence pourrait être coupée en 5 portions de 2000 images consécutives (un morceau de l'image 1 à 2000, un deuxième morceau de la 2001 à 4000, etc.). L'algorithme de suivi pourrait ensuite être appliqué sur chaque portion simultanément sur les 5 coeurs. Avec cette approche, il faudrait alors ajouter une dernière étape pour mettre en commun les résultats de chaque coeur, et ainsi mettre bout à bout les trajectoires d'une portion à l'autre.

### **Combinaison du suivi déterministe et du suivi par filtrage particulaire**

Par ailleurs, pour limiter la longueur de la séquence à étudier avec précision, il est possible d'utiliser conjointement le suivi déterministe et le suivi probabiliste. Pour ce faire, sur une longue séquence, l'algorithme déterministe est d'abord appliqué. Il permet de faire une analyse globale des trajectoires et de repérer les périodes à fort régime transitoire. Ensuite, sur ces périodes, l'algorithme à filtrage particulaire peut alors être lancé pour une étude approfondie et plus précise des trajectoires.

### **Amélioration de la terminaison des trajectoires du suivi par filtrage particulaire**

Comme nous l'avons vu dans l'analyse quantitative ci-avant, le suivi par filtrage particulaire présente des difficultés pour terminer les trajectoires lorsque celles-ci sont proches de sortir du champ de l'image. En effet, il peut arriver qu'un traqueur continue d'estimer son

état alors que la trajectoire en question a déjà quitté l'image. Ceci provoque alors une erreur de type faux positif. Dans ce type de situation, le suivi déterministe apparaît plus efficace car il stoppe automatiquement le traqueur lorsque la trajectoire sort de l'image. Il faudrait peut-être ici gérer les bords d'images de la même manière que le suivi déterministe.

### **Détermination automatique des paramètres du suivi par filtrage particulaire**

Le suivi par filtrage particulaire a permis de corriger les erreurs de détection sans perdre significativement en précision de positions par rapport au suivi déterministe. Le fait de ne pas avoir gagné en précision de positions vient du fait que les nombreux paramètres de l'algorithme à filtrage particulaire pourraient être mieux ajustés à la séquence d'images (comme les paramètres influents et difficiles à régler que sont  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\pi$  par exemples). Pour améliorer ce point, on pourrait envisager des méthodes d'estimation automatique des paramètres.

### **Amélioration de l'estimation en ligne des états de mouvement**

Les états de mouvement estimés en ligne présentent des erreurs, notamment pour l'état de roulement qui peut être confondu avec les états repos et saltation. L'estimation des états de mouvement se base actuellement sur deux critères : un sur la vitesse et l'autre sur la distance aux voisins. Une piste d'amélioration serait de considérer également la profondeur de la bille dans le lit ou bien l'état de mouvement de ses voisins.

### **Réduction du temps de calcul du suivi par filtrage particulaire**

Le temps de calcul du suivi par filtrage particulaire est beaucoup plus long que le déterministe. En effet, nous avons observé que le suivi déterministe s'exécute environ 9 fois plus rapidement que le suivi par filtrage particulaire (pour une séquence de 5000 images, le suivi déterministe met 1h40 contre 15h25 pour le filtrage particulaire). Pour réduire le temps d'exécution du filtrage particulaire, il peut être envisagé d'écrire certaines parties de l'algorithme en C et d'appeler une fonction mex depuis Matlab, voire d'utiliser un GPU.

### **Utilisation d'un cluster de calcul**

Les calculs que nous effectuons sur des longues séquences d'images sont longs (près d'une semaine pour l'algorithme déterministe) et peuvent nécessiter beaucoup de place mémoire pendant leur exécution. Dans notre cas, nous utilisons un ordinateur à 12 coeurs et 32 Go de mémoire vive. Une perspective pour gagner en temps de calcul serait d'utiliser une ferme de calcul, aussi appelée cluster de calcul. Un cluster de calcul est un regroupement d'ordinateurs permettant de dépasser les limitations d'une machine seule en terme de puissance de calcul. Avec des ressources beaucoup plus importantes qu'un simple ordinateur (serveurs, processeurs, coeurs, mémoire vive, etc.), le gain de temps serait considérable. On pourrait alors envisager d'appliquer l'algorithme basé sur le filtrage particulaire à modèles multiples sur des longues séquences et ainsi avoir des résultats précis sur la totalité du phénomène de ségrégation en charriage.

### **Création d'une longue séquence numérique**

La séquence numérique de 10000 images que nous avons réalisée et qui nous sert de vérité terrain présente des limitations. Le nombre de billes transparentes est assez faible (aucune au début, une centaine à la fin), leur infiltration est limitée et la longueur de la séquence peut paraître courte pour présenter tous les cas problématiques. Pour pallier cette difficulté, nous

pourrions envisager de créer une séquence numérique beaucoup plus longue (plus proche des 500000 images que nous analysons avec les séquences expérimentales). Avec une séquence plus longue, plus de billes transparentes apparaîtraient dans les images au bout d'un certain temps et leur infiltration dans le lit serait plus marquée car elles auraient plus de temps pour s'infiltrer.

### **Classification des trajectoires**

Plus généralement, on pourrait envisager d'étudier les différents types de trajectoires que l'on rencontre dans une expérience. Cela pourrait se faire par une méthode de classification non supervisée. Ce type de méthode est capable de former automatiquement des groupes de trajectoires similaires à condition de définir préalablement une métrique de comparaison entre trajectoires. On pourrait ainsi faire l'inventaire des types de comportements que l'on rencontre et les relier à des paramètres physiques relatifs à la ségrégation.



# Références

- Adrian, R. J. Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow : speckle velocimetry vs particle image velocimetry. *Applied Optics*, 23(11) : 1690–1691, June 1984. doi:10.1364/AO.23.001690.
- Adrian, R. J. Particle-imaging techniques for experimental fluid mechanics. *Annual Review of Fluid Mechanics*, 23(1) :261–304, January 1991. doi:10.1146/annurev.fl.23.010191.001401.
- Adrian, R. J. Twenty years of particle image velocimetry. *Experiments in Fluids*, 39(2) : 159–169, August 2005. ISSN 1432-1114. doi:10.1007/s00348-005-0991-7.
- Adrian, R. J. and Westerweel, J. *Particle image velocimetry*. Cambridge University Press, 2011. ISBN 978-0-521-44008-0.
- Alt, N., Hinterstoisser, S., and Navab, N. Rapid selection of reliable templates for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1355–1362. IEEE, IEEE, June 2010. doi:10.1109/cvpr.2010.5539812.
- Ancey, C., Bigillon, F., Frey, P., Lanier, J., and Ducret, R. Saltating motion of a bead in a rapid water stream. *Physical Review E*, 66(3) :1–16, September 2002. doi:10.1103/physreve.66.036306. ID 036306.
- Ancey, C., Bigillon, F., Frey, P., and Ducret, R. Rolling motion of a bead in a rapid water stream. *Physical Review E*, 67(1) :1–11, January 2003. doi:10.1103/physreve.67.011303. ID 011303.
- Ancey, C., Böhm, T., Frey, P., Jodeau, M., and Reboud, J.-L. Saltating or rolling stones? In *River, Coastal and Estuarine Morphodynamics (RCEM), IAHR 4th Symposium on*, volume 1, pages 3–11. CRC Press, October 2005. doi:10.1201/9781439833896.ch1.
- Ancey, C., Böhm, T., Jodeau, M., and Frey, P. Statistical description of sediment transport experiments. *Physical Review E*, 74(1) :1–14, July 2006. doi:10.1103/physreve.74.011302. ID 011302.
- Andriluka, M., Roth, S., and Schiele, B. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1–8. IEEE, June 2008. doi:10.1109/CVPR.2008.4587583.
- Arulampalam, M. S., Ristic, B., Gordon, N., and Mansell, T. Bearings-only tracking of manoeuvring targets using particle filters. *Journal on Advances in Signal Processing, EURASIP*, 2004(15) :1–15, December 2004. doi:10.1155/S1110865704405095.

## RÉFÉRENCES

---

- Bacchi, V., Recking, A., Eckert, N., Frey, P., Piton, G., and Naaim, M. The effects of kinetic sorting on sediment mobility on steep slopes. *Earth Surface Processes and Landforms*, 39 (8) :1075–1086, June 2014. doi:10.1002/esp.3564.
- Bar-Shalom, Y. *Tracking and data association*. Academic Press Professional, Inc., 1987. ISBN 0-120-79760-7.
- Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., and Rosenberger, C. Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition (ICPR), IEEE 19th International Conference on*, pages 1–4. IEEE, IEEE, December 2008. doi:10.1109/icpr.2008.4760998.
- Bernardin, K. and Stiefelhagen, R. Evaluating multiple object tracking performance : the CLEAR MOT metrics. *Journal on Image and Video Processing*, 2008(246309) :1–10, May 2008. doi:10.1155/2008/246309.
- Beucher, S. and Bilodeau, M. Road segmentation and obstacle detection by a fast watershed transformation. In *Intelligent Vehicles '94 Symposium, Proceedings of the*, pages 296–301. IEEE, October 1994. doi:10.1109/IVS.1994.639531.
- Beucher, S. and Lantuejoul, C. Use of watersheds in contour detection. In *International Workshop on Image Processing : Real-time Edge and Motion Detection/Estimation, Rennes, France*, September 1979.
- Beucher, S. and Meyer, F. *The morphological approach to segmentation : the watershed transformation*, volume 34, pages 433–433. Marcel Dekker AG, 1992.
- Bewley, A., Guizilini, V., Ramos, F., and Upcroft, B. Online self-supervised multi-instance segmentation of dynamic objects. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1296–1303. IEEE, IEEE, May 2014. doi:10.1109/icra.2014.6907020.
- Bigillon, F. *Étude du mouvement bidimensionnel d'une particule dans un courant d'eau sur forte pente*. PhD thesis, Université Grenoble 1 - Joseph Fourier, 2001.
- Blom, H. A. P. and Bar-Shalom, Y. The interacting multiple model algorithm for systems with Markovian switching coefficients. *Automatic Control, IEEE Transactions on*, 33(8) : 780–783, August 1988. doi:10.1109/9.1299.
- Boers, Y. and Driessen, J. N. Interacting multiple model particle filter. *Radar, Sonar and Navigation, IEE Proceedings*, 150(5) :344–349, October 2003. doi:10.1049/ip-rsn:20030741.
- Böhm, T. *Motion and interaction of a set of particles in a supercritical flow*. PhD thesis, Université Grenoble 1 - Joseph Fourier, 2005.
- Böhm, T., Ancey, C., Frey, P., Reboud, J.-L., and Ducottet, C. Fluctuations of the solid discharge of gravity-driven particle flows in a turbulent stream. *Physical Review E*, 69(6) : 1–13, June 2004. doi:10.1103/PhysRevE.69.061307. ID 061307.
- Böhm, T., Frey, P., Ducottet, C., Ancey, C., Jodeau, M., and Reboud, J.-L. Two-dimensional motion of a set of particles in a free surface flow with image processing. *Experiments in Fluids*, 41(1) :1–11, July 2006. doi:10.1007/s00348-006-0134-9.

- Bouwmans, T., Porikli, F., Höferlin, B., and Vacavant, A. *Background modeling and foreground detection for video surveillance*. Chapman and Hall/CRC, August 2014. doi: [10.1201/b17223](https://doi.org/10.1201/b17223).
- Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 33(9) :1820–1833, September 2011. doi:[10.1109/TPAMI.2010.232](https://doi.org/10.1109/TPAMI.2010.232).
- Brunelli, R. *Template matching techniques in computer vision : theory and practice*. John Wiley & Sons, 2009.
- Cannons, K. A review of visual tracking. Technical report, Dept. Comput. Sci. Eng., York University, Toronto, Canada, Technical Report CSE-2008-07, September 2008.
- Choset, H. and Nagatani, K. Topological simultaneous localization and mapping (SLAM) : toward exact localization without explicit localization. *Robotics and Automation, IEEE Transactions on*, 17(2) :125–137, April 2001. doi:[10.1109/70.928558](https://doi.org/10.1109/70.928558).
- Cowen, E. A. and Monismith, S. G. A hybrid digital particle tracking velocimetry technique. *Experiments in Fluids*, 22(3) :199–211, January 1997. doi:[10.1007/s003480050038](https://doi.org/10.1007/s003480050038).
- Crowley, J. L., Stelmazyk, P., and Discours, C. Measuring image flow by tracking edge-lines. In *Computer Vision (ICCV), IEEE 2nd International Conference on*, pages 658–664. IEEE, December 1988. doi:[10.1109/ccv.1988.590047](https://doi.org/10.1109/ccv.1988.590047).
- Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 25(10) :1337–1342, October 2003. ISSN 0162-8828. doi:[10.1109/TPAMI.2003.1233909](https://doi.org/10.1109/TPAMI.2003.1233909).
- Cui, Y., Wooster, J. K., Baker, P. F., Dusterhoff, S. R., Sklar, L. S., and Dietrich, W. E. Theory of fine sediment infiltration into immobile gravel bed. *Journal of Hydraulic Engineering*, 134(10) :1421–1429, October 2008. doi:[10.1061/\(asce\)0733-9429\(2008\)134:10\(1421\)](https://doi.org/10.1061/(asce)0733-9429(2008)134:10(1421)).
- Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, IEEE, June 2005. doi:[10.1109/cvpr.2005.177](https://doi.org/10.1109/cvpr.2005.177).
- Del Moral, P. Non-linear filtering : interacting particle resolution. *Markov Processes and Related Fields*, 2(4) :555–581, 1996.
- Dermisis, D. and Papanicolaou, A. N. T. The effects of protruding rock boulders in regulating sediment intrusion within the hyporheic zone of mountain streams. *Journal of Mountain Science*, 11(6) :1466–1477, November 2014. ISSN 1993-0321. doi: [10.1007/s11629-014-3054-9](https://doi.org/10.1007/s11629-014-3054-9).
- Doucet, A., de Freitas, A., and Gordon, N. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2001. doi:[10.1007/978-1-4757-3437-9](https://doi.org/10.1007/978-1-4757-3437-9).

## RÉFÉRENCES

---

- Dracos, T. *Particle tracking velocimetry (PTV)*, pages 155–160. Springer, September 1996. ISBN 978-94-015-8727-3. doi:10.1007/978-94-015-8727-3\_7.
- Dudill, A. *Experimental study of segregation mechanisms in bedload sediment transport*. PhD thesis, University of British Columbia and University of Grenoble Alpes, September 2016.
- Dudill, A., Frey, P., and Church, M. Infiltration of fine sediment into a coarse mobile bed : a phenomenological study. *Earth Surface Processes and Landforms*, pages 1–15, December 2016a. ISSN 1096-9837. doi:10.1002/esp.4080.
- Dudill, A., Lafaye de Micheaux, H., Frey, P., and Church, M. Introducing finer grains into bedload : the transition to a new equilibrium. *Journal of Geophysical Research : Earth Surface*, Submitted to, October 2016b.
- Efron, B. and Tibshirani, R. J. *An introduction to the bootstrap*. CRC press, 1994. doi:10.1007/978-1-4899-4541-9.
- Einstein, H. A. Deposition of suspended particles in a gravel bed. *Journal of Hydraulics Division*, 94(5) :1197–1205, 1968.
- Elgammal, A., Harwood, D., and Davis, L. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767. Springer, April 2000. ISBN 978-3-540-45053-5. doi:10.1007/3-540-45053-X\_48.
- Fortmann, T., Bar-Shalom, Y., and Scheffe, M. Sonar tracking of multiple targets using joint probabilistic data association. *Journal of Oceanic Engineering, IEEE*, 8(3) :173–184, July 1983. doi:10.1109/joe.1983.1145560.
- Frey, P. Particle velocity and concentration profiles in bedload experiments on a steep slope. *Earth Surface Processes and Landforms*, 39(5) :646–655, April 2014. doi:10.1002/esp.3517.
- Frey, P. and Church, M. How river beds move. *Science*, 325(5947) :1509–1510, September 2009. doi:10.1126/science.1178516.
- Frey, P. and Church, M. Bedload : A granular phenomenon. *Earth Surface Processes and Landforms*, 36(1) :58–69, January 2011. doi:10.1002/esp.2103.
- Frostick, L. E., Lucas, P. M., and Reid, I. The infiltration of fine matrices into coarse-grained alluvial sediments and its implications for stratigraphical interpretation. *Journal of the Geological Society*, 141(6) :955–965, November 1984. doi:10.1144/gsjgs.141.6.0955.
- Gauglitz, S., Höllerer, T., and Turk, M. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3) :335, March 2011. doi:10.1007/s11263-011-0431-5.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2) :107–113, April 1993. doi:10.1049/ip-f-2.1993.0015.

- Grau, V., Mewes, A. U. J., Alcaniz, M., Kikinis, R., and Warfield, S. K. Improved watershed transform for medical image segmentation using prior information. *Medical Imaging, IEEE Transactions on*, 23(4) :447–458, April 2004. doi:10.1109/tmi.2004.824224.
- Harris, C. and Stephens, M. A combined corner and edge detector. *Alvey Vision Conference, Proceedings of the*, 15(50) :10–5244, 1988. doi:10.5244/c.2.23.
- Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision (ECCV)*, volume 7575, pages 702–715. Springer, October 2012. doi:10.1007/978-3-642-33765-9\_50.
- Hergault, V. *Étude microstructurelle du transport par charriage de mélanges bidisperses à forte pente*. PhD thesis, Université de Grenoble, 2011.
- Hergault, V., Frey, P., Métivier, F., Barat, C., Ducottet, C., Böhm, T., and Ancey, C. Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow. *Experiments in Fluids*, 49(5) :1095–1107, November 2010. doi:10.1007/s00348-010-0856-6.
- Houssais, M., Ortiz, C. P., Durian, D. J., and Jerolmack, D. J. Onset of sediment transport is a continuous transition driven by fluid shear and granular creep. *Nature communications*, 6(6527) :1–8, March 2015. doi:10.1038/ncomms7527.
- Isard, M. and Blake, A. A mixed-state condensation tracker with automatic model-switching. In *Computer Vision (ICCV), IEEE 6th International Conference on*, pages 107–112. IEEE, January 1998a. doi:10.1109/ICCV.1998.710707.
- Isard, M. and Blake, A. Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1) :5–28, August 1998b. doi:10.1023/A:1008078328650.
- Julier, S. J. and Uhlmann, J. K. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, SPIE-Intl Soc Optical Eng, July 1997. doi:10.1117/12.280797.
- Kalal, Z., Mikolajczyk, K., and Matas, J. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 34(7) :1409–1422, July 2012. doi:10.1109/tpami.2011.239.
- Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1) :35–45, March 1960. doi:10.1115/1.3662552.
- Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2) :83–97, March 1955. doi:10.1007/978-3-540-68279-0\_2.
- Lafaye de Micheaux, H., Dudill, A., Frey, P., and Ducottet, C. Image processing to study the evolution of channel slope and water depth in bimodal sediment mixtures. In *10th Pacific Symposium on Flow Visualization and Image Processing (PSFVIP), Naples, Italy*, pages 1–10, June 2015. Paper ID 43.

## RÉFÉRENCES

---

- Lafaye de Micheaux, H., Ducottet, C., and Frey, P. Online multi-model particle filter-based tracking to study bedload transport. In *Image Processing (ICIP), IEEE International Conference on*, pages 3489–3493. IEEE, September 2016. doi:10.1109/ICIP.2016.7533008.
- Li, X. R. and Jilkov, V. P. Survey of maneuvering target tracking. Part V. Multiple-model methods. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(4) :1255–1321, October 2005. doi:10.1109/taes.2005.1561886.
- Lima, R., Wada, S., Tanaka, S., Takeda, M., Ishikawa, T., Tsubota, K.-I., Imai, Y., and Yamaguchi, T. In vitro blood flow in a rectangular PDMS microchannel : experimental observations using a confocal micro-PIV system. *Biomedical Microdevices*, 10(2) :153–167, September 2008. doi:10.1007/s10544-007-9121-z.
- Lisle, T. E. Sediment transport and resulting deposition in spawning gravels, north coastal California. *Water Resources Research*, 25(6) :1303–1319, June 1989. doi:10.1029/wr025i006p01303.
- Liu, L. S. and Chen, R. Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443) :1032–1044, September 1998. doi:10.1080/01621459.1998.10473765.
- Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, November 2004. doi:10.1023/b:visi.0000029664.99615.94.
- Lu, W.-L. *Tracking and recognizing actions of multiple hockey players using the boosted particle filter*. PhD thesis, Citeseer, 2007.
- Lu, W.-L., Okuma, K., and Little, J. J. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1) :189–205, January 2009. doi:10.1016/j.imavis.2008.02.008.
- Maas, H. G., Gruen, A., and Papantoniou, D. Particle tracking velocimetry in three-dimensional flows. Part 1 : Photogrammetric determination of particle coordinates. *Experiments in Fluids*, 15(2) :133–146, July 1993. ISSN 1432-1114. doi:10.1007/BF00190953.
- Malik, N. A., Dracos, T., and Papantoniou, D. A. Particle tracking velocimetry in three-dimensional flows. Part II : Particle tracking. *Experiments in Fluids*, 15(4) :279–294, September 1993. ISSN 1432-1114. doi:10.1007/BF00223406.
- Maurin, R., Chauchat, J., Chareyre, B., and Frey, P. A minimal coupled fluid-discrete element model for bedload transport. *Physics of Fluids (1994-present)*, 27(11) :113302, November 2015. doi:10.1063/1.4935703.
- Maurin, R., Chauchat, J., and Frey, P. Dense granular flow rheology in turbulent bedload transport. *Journal of Fluid Mechanics*, 804 :490–512, October 2016. doi:10.1017/jfm.2016.520.
- Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. Interacting multiple model methods in target tracking : a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1) :103–123, January 1998. doi:10.1109/7.640267.

- McGinnity, S. and Irwin, G. W. Multiple model bootstrap filter for maneuvering target tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, 36(3) :1006–1012, July 2000. doi:10.1109/7.869522.
- Mei, X., Ling, H., Wu, Y., Blasch, E., and Bai, L. Minimum error bounded efficient L1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1257–1264. IEEE, IEEE, June 2011. doi:10.1109/CVPR.2011.5995421.
- Mikolajczyk, K. and Schmid, C. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 27(10) :1615–1630, October 2005. ISSN 0162-8828. doi:10.1109/TPAMI.2005.188.
- Moravec, H. P. Visual mapping by a robot rover. In *Artificial Intelligence, Proceedings of the 6th International Joint Conference on (Volume 1)*, pages 598–600. Morgan Kaufmann Publishers Inc., 1979.
- Munkres, J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1) :32–38, March 1957. doi:10.1137/0105003.
- Ni, W.-J. and Capart, H. Cross-sectional imaging of refractive-index-matched liquid-granular flows. *Experiments in Fluids*, 56(8) :163, August 2015. doi:10.1007/s00348-015-2034-3.
- Ochs, P., Malik, J., and Brox, T. Segmentation of moving objects by long term video analysis. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 36(6) :1187–1200, June 2014. doi:10.1109/tpami.2013.242.
- Odobez, J.-M., Gatica-Perez, D., and Ba, S. O. Embedding motion in model-based stochastic tracking. *Image Processing, IEEE Transactions on*, 15(11) :3514–3530, November 2006. doi:10.1109/TIP.2006.877497.
- Ohmi, K. and Li, H.-Y. Particle-tracking velocimetry with new algorithms. *Measurement Science and Technology*, 11(6) :603–616, May 2000. doi:10.1088/0957-0233/11/6/303.
- Oliver, N. M., Rosario, B., and Pentland, A. P. A Bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 22(8) :831–843, August 2000. ISSN 0162-8828. doi:10.1109/34.868684.
- Ouellette, N. T., Xu, H., and Bodenschatz, E. A quantitative study of three-dimensional Lagrangian particle tracking algorithms. *Experiments in Fluids*, 40(2) :301–313, November 2006. doi:10.1007/s00348-005-0068-7.
- Papageorgiou, C. P., Oren, M., and Poggio, T. A general framework for object detection. In *Computer Vision (ICCV), IEEE 6th International Conference on*, pages 555–562. IEEE, January 1998. doi:10.1109/iccv.1998.710772.
- Patras, I., Hendriks, E. A., and Lagendijk, R. L. Video segmentation by MAP labeling of watershed segments. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 23(3) :326–332, March 2001. ISSN 0162-8828. doi:10.1109/34.910886.

## RÉFÉRENCES

---

- Pickering, C. J. D. and Halliwell, N. A. Speckle photography in fluid flows : signal recovery with two-step processing. *Applied Optics*, 23(8) :1128–1129, April 1984. doi:10.1364/ao.23.001128.
- Powell, D. M. Patterns and processes of sediment sorting in gravel-bed rivers. *Progress in Physical Geography*, 22(1) :1–32, March 1998. doi:10.1191/030913398666402127.
- Raffel, M., Willert, C. E., Wereley, S., and Kompenhans, J. *Particle image velocimetry : a practical guide*. Springer, 2013.
- Rangarajan, K. and Shah, M. Establishing motion correspondence. *CVGIP : Image Understanding*, 54(1) :56–73, July 1991. doi:10.1016/1049-9660(91)90075-z.
- Rathi, Y., Vaswani, N., Tannenbaum, A., and Yezzi, A. Tracking deforming objects using particle filtering for geometric active contours. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 29(8) :1470–1475, August 2007. doi:10.1109/TPAMI.2007.1081.
- Recking, A., Richard, D., and Degoutte, G. *Torrents et rivières de montagne : dynamique et aménagement*. Editions Quae, 2013.
- Reid, D. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6) :843–854, December 1979. doi:10.1109/tac.1979.1102177.
- Rorsman, K. Étude du tri granulométrique bidisperse en charriage à forte pente. mathesis, Stage fin d'étude INP Ense3, Grenoble (Erasmus MEng Imperial College, London), August 2012.
- Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3) :125–141, August 2008. doi:10.1007/s11263-007-0075-7.
- Rowley, H. A., Baluja, S., and Kanade, T. Neural network-based face detection. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 20(1) :23–38, January 1998. doi:10.1109/34.655647.
- Salari, V. and Sethi, I. K. Feature point correspondence in the presence of occlusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1) :87–91, 1990.
- Savage, S. B. and Lun, C. K. K. Particle size segregation in inclined chute flow of dry cohesionless granular solids. *Journal of Fluid Mechanics*, 189 :311–335, April 1988. doi:10.1017/s002211208800103x.
- Sethi, I. K. and Jain, R. Finding trajectories of feature points in a monocular image sequence. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 9(1) :56–73, January 1987. doi:10.1109/tpami.1987.4767872.
- Shafique, K. and Shah, M. A noniterative greedy algorithm for multiframe point correspondence. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 27(1) :51–65, January 2005. doi:10.1109/TPAMI.2005.1.

- Shen, H., Zhang, L., Huang, B., and Li, P. A MAP approach for joint motion estimation, segmentation, and super resolution. *Image Processing, IEEE Transactions on*, 16(2) : 479–490, February 2007. doi:10.1109/tip.2006.888334.
- Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. Visual tracking : An experimental survey. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 36(7) :1442–1468, July 2014. doi:10.1109/TPAMI.2013.230.
- Smith, K., Gatica-Perez, D., Odobez, J.-M., and Ba, S. Evaluating multi-object tracking. In *Computer Vision and Pattern Recognition (CVPR) Workshops, IEEE Computer Society Conference on*, pages 36–36. IEEE, IEEE, 2005. doi:10.1109/cvpr.2005.453.
- Sobel, I. and Feldman, G. A 3x3 isotropic gradient operator for image processing. *A talk at the Stanford Artificial Intelligence Project*, pages 271–272, 1968.
- Sobral, A. and Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding (CVIU)*, 122 :4–21, May 2014. doi:10.1016/j.cviu.2013.12.005.
- Soille, P. *Morphological image analysis : Principles and Applications*. Springer-Verlag Berlin Heidelberg, 1999. doi:10.1007/978-3-662-05088-0.
- Stauffer, C. and Grimson, W. E. L. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 22(8) :747–757, August 2000. ISSN 0162-8828. doi:10.1109/34.868677.
- Taylor, Z. J., Gurka, R., Kopp, G. A., and Liberzon, A. Long-duration time-resolved PIV to study unsteady aerodynamics. *Instrumentation and Measurement, IEEE Transactions on*, 59(12) :3262–3269, December 2010. ISSN 0018-9456. doi:10.1109/TIM.2010.2047149.
- Tuyen, N. B. and Cheng, N.-S. A single-camera technique for simultaneous measurement of large solid particles transported in rapid shallow channel flows. *Experiments in Fluids*, 53 (5) :1269–1287, November 2012. doi:10.1007/s00348-012-1358-5.
- Veenman, C. J., Reinders, M. J. T., and Backer, E. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 23(1) :54–72, January 2001. doi:10.1109/34.899946.
- Vincent, L. and Soille, P. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 13(6) :583–598, June 1991. doi:10.1109/34.87344.
- Wereley, S. T. and Meinhart, C. D. Recent advances in micro-particle image velocimetry. *Annual Review of Fluid Mechanics*, 42(1) :557–576, January 2010. doi:10.1146/annurev-fluid-121108-145427.
- Westerweel, J., Elsinga, G. E., and Adrian, R. J. Particle image velocimetry for complex and turbulent flows. *Annual Review of Fluid Mechanics*, 45(1) :409–436, January 2013. doi:10.1146/annurev-fluid-120710-101204.

## RÉFÉRENCES

---

- Wooster, J. K., Dusterhoff, S. R., Cui, Y., Sklar, L. S., Dietrich, W. E., and Malko, M. Sediment supply and relative size distribution effects on fine sediment infiltration into immobile gravels. *Water Resources Research*, 44(3) :1–18, March 2008. ISSN 1944-7973. doi:10.1029/2006WR005815. ID W03424.
- Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. Pfinder : real-time tracking of the human body. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 19(7) :780–785, July 1997. ISSN 0162-8828. doi:10.1109/34.598236.
- Wu, B. and Nevatia, R. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2) :247–266, January 2007. doi:10.1007/s11263-006-0027-7.
- Wu, Y., Lim, J., and Yang, M.-H. Online object tracking : A benchmark. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2411–2418. IEEE, June 2013. doi:10.1109/CVPR.2013.312.
- Yilmaz, A., Javed, O., and Shah, M. Object tracking : A survey. *ACM Computing Surveys (CSUR)*, 38(4) :13, December 2006. doi:10.1145/1177352.1177355.
- Zhai, Y., Yearly, M. B., Cheng, S., and Kehtarnavaz, N. An object-tracking algorithm based on multiple-model particle filtering with state partitioning. *Instrumentation and Measurement, IEEE Transactions on*, 58(5) :1797–1809, May 2009. doi:10.1109/tim.2009.2014511.
- Zhang, J., Shi, F., Wang, J., and Liu, Y. 3d motion segmentation from straight-line optical flow. In *Multimedia Content Analysis and Mining*, pages 85–94. Springer, 2007. doi:10.1007/978-3-540-73417-8\_15.

# Annexes

## A Rappels d'opérateurs de traitement d'images

### A.1 Transformations morphologiques élémentaires

Cette section fait une brève présentation de certaines notions de morphologie mathématique qu'il nous paraît utile de rappeler dans cette thèse. Les lecteurs qui désirent approfondir certains points sont invités à se reporter à l'ouvrage de Soille [1999].

Par la suite, nous nous plaçons dans l'ensemble  $\mathbb{Z}^2$  qui modélise le support des images à deux dimensions. Une image 2D  $f$  (ou fonction) est alors définie par :

$$f : D_f \subset \mathbb{Z}^2 \rightarrow \{0, 1, \dots, t_{max}\} \quad (\text{A.1})$$

où  $D_f$  est le support spatial de l'image et  $t_{max}$  est le niveau de gris maximal.

Nous utilisons aussi le terme  $B$  pour désigner un sous-ensemble de  $\mathbb{Z}^2$ , appelé élément structurant. L'élément structurant joue le rôle de sonde dans l'image à traiter. Il est balayé dans l'image pour étudier sa relation avec elle. Si  $\mathbf{x}$  est un élément de  $\mathbb{Z}^2$ , alors  $B_{\mathbf{x}}$  est l'ensemble  $B$  translaté de  $\mathbf{x}$  tel que :

$$B_{\mathbf{x}} = \{\mathbf{b} + \mathbf{x} | \mathbf{b} \in B\} \quad (\text{A.2})$$

#### A.1.1 Érosion et dilatation

L'érosion  $\varepsilon_B(f)$  et la dilatation  $\delta_B(f)$  sont les opérateurs de base de la morphologie mathématique. Ils se basent sur le filtrage d'une image  $f$  en appliquant un élément structurant  $B$  (filtre). L'érosion (resp. la dilatation) consiste à chercher dans  $f$  les sous-ensembles qui incluent (resp. intersectent)  $B$ , et donc à donner à tout pixel  $\mathbf{x}$  la valeur minimale (resp. maximale) de  $f$  dans la fenêtre d'observation définie par  $B$ , lorsque  $B$  est centré en  $\mathbf{x}$  :

$$\varepsilon_B(f) = \bigwedge_{\mathbf{b} \in B} f_{-\mathbf{b}} \quad \text{ou} \quad [\varepsilon_B(f)](\mathbf{x}) = \min_{\mathbf{b} \in B} f(\mathbf{x} + \mathbf{b}) \quad (\text{A.3})$$

$$\delta_B(f) = \bigvee_{\mathbf{b} \in B} f_{-\mathbf{b}} \quad \text{ou} \quad [\delta_B(f)](\mathbf{x}) = \max_{\mathbf{b} \in B} f(\mathbf{x} + \mathbf{b}) \quad (\text{A.4})$$

où  $\wedge$  et  $\vee$  sont respectivement les opérations de minimum et de maximum point par point, et  $f_{-\mathbf{b}}$  est la translation de  $f$  par le vecteur  $-\mathbf{b}$ .

De manière intuitive, l'érosion réduit la taille des zones claires en rapport avec l'élément structurant et la dilatation les augmente.

### A.1.2 Ouverture et fermeture

L'ouverture  $\gamma_B(f)$  et la fermeture  $\phi_B(f)$  sont une combinaison des opérateurs précédents. L'ouverture consiste en une érosion d'une image  $f$  par un élément structurant  $B$  suivie d'une dilatation par  $B$ , et la fermeture en une dilatation par  $B$  suivie d'une érosion par  $B$  :

$$\gamma_B(f) = \delta_B[\varepsilon_B(f)] \quad (\text{A.5})$$

$$\phi_B(f) = \varepsilon_B[\delta_B(f)] \quad (\text{A.6})$$

D'une manière intuitive, l'ouverture fait disparaître les pics de l'image et la fermeture les vallées, et ceci selon un critère de taille et de forme déterminé par l'élément structurant.

### A.1.3 Transformation *top-hat* par ouverture et par fermeture

La transformation *top-hat* (ou chapeau haut de forme) par ouverture (ou *white top-hat*)  $WTH(f)$  est le résidu entre l'image  $f$  et son ouverture. La transformation *top-hat* par fermeture (ou *black top-hat*)  $BTH(f)$  est le résidu entre la fermeture de  $f$  et  $f$ . Ces transformations sont définies par :

$$WTH(f) = f - \gamma_B(f) \quad (\text{A.7})$$

$$BTH(f) = \phi_B(f) - f \quad (\text{A.8})$$

D'une manière intuitive, le *top-hat* par ouverture permet de détecter ce que l'ouverture a fait disparaître de l'image (*i.e.* les pics) et le *top-hat* par fermeture ce que la fermeture a fait disparaître (*i.e.* les vallées).

### A.1.4 Érosion et dilatation géodésique

L'érosion géodésique  $\varepsilon_g^{(1)}(f)$  (resp. dilatation géodésique  $\delta_g^{(1)}(f)$ ) est une érosion (resp. dilatation) unitaire (*i.e.* d'élément structurant de taille 1) de l'image  $f$  à l'intérieur d'un masque  $g$ . L'érosion géodésique  $\varepsilon_g^{(n)}(f)$  (resp. dilatation géodésique  $\delta_g^{(n)}(f)$ ) de taille  $n$  est l'application successive de  $n$  érosions (resp. dilatations) géodésiques :

$$\varepsilon_g^{(1)}(f) = \varepsilon_B(f) \vee g \quad \text{soit} \quad \varepsilon_g^{(n)}(f) = \varepsilon_g^{(1)}[\varepsilon_g^{(n-1)}(f)] \quad (\text{A.9})$$

$$\delta_g^{(1)}(f) = \delta_B(f) \wedge g \quad \text{soit} \quad \delta_g^{(n)}(f) = \delta_g^{(1)}[\delta_g^{(n-1)}(f)] \quad (\text{A.10})$$

où  $B$  est l'élément structurant de taille 1.

Ces opérations géodésiques sont principalement utilisées pour des opérations de reconstruction d'images.

### A.1.5 Reconstruction par érosion et par dilatation

La reconstruction par érosion  $R_g^\varepsilon(f)$  (resp. reconstruction par dilatation  $R_g^\delta(f)$ ) consiste en une itération d'érosions (resp. dilatations) géodésiques d'une image  $f$  par une image

masque  $g$  jusqu'à stabilité, soit l'idempotence (*i.e.* la réitération de l'opération ne change plus le résultat). Ces reconstructions sont définies par :

$$R_g^\varepsilon(f) = \varepsilon_g^{(i)}(f) \quad (\text{A.11})$$

$$R_g^\delta(f) = \delta_g^{(i)}(f) \quad (\text{A.12})$$

où  $i$  est tel que  $\varepsilon_g^{(i)}(f) = \varepsilon_g^{(i+1)}(f)$  ou  $\delta_g^{(i)}(f) = \delta_g^{(i+1)}(f)$ .

De manière intuitive, une reconstruction par érosion reconstruit les vallées de l'image qui sont marquées par les vallées du masque, les autres vallées sont comblées, ou autrement dit, les vallées du masque sont élargies jusqu'à atteindre les vallées correspondantes de l'image.

### A.1.6 Extraction des maximums et minimums locaux

Un maximum (resp. minimum) local est défini comme un pixel ayant une valeur plus grande (resp. petite) que ces pixels voisins. L'ensemble des maximums locaux extraits d'une image  $f$  est noté  $RMAX(f)$  et celui des minimums  $RMIN(f)$  :

$$RMAX(f) = f - R_f^\delta(f - 1) \quad (\text{A.13})$$

$$RMIN(f) = R_f^\varepsilon(f + 1) - f \quad (\text{A.14})$$

### A.1.7 Transformation *hmax* et *hmin*

La transformation *hmax* (resp. *hmin*) est utilisée pour supprimer dans une image  $f$  tous les maximums (resp. minimums) dont la hauteur (resp. profondeur) est inférieure ou égale à une valeur seuil  $h$  donnée. Elles sont notées respectivement  $HMAX_h(f)$  et  $HMIN_h(f)$ . Elles se basent respectivement sur une reconstruction par dilatation et par érosion :

$$HMAX_h(f) = R_f^\delta(f - h) \quad (\text{A.15})$$

$$HMIN_h(f) = R_f^\varepsilon(f + h) \quad (\text{A.16})$$

### A.1.8 Transformation *hconvexe* et *hconcave*

La transformation *hconvexe* (resp. *hconcave*) est utilisée pour détecter dans une image  $f$  les maximums (resp. minimums) locaux dont la hauteur (resp. profondeur) est supérieure ou égale à une valeur seuil  $h$  donnée. Elles sont notées respectivement  $HCONVEXE_h(f)$  et  $HCONCAVE_h(f)$ . Elles se basent respectivement sur une transformation *hmax* et *hmin* :

$$HCONVEXE_h(f) = f - HMAX_h(f) \quad (\text{A.17})$$

$$HCONCAVE_h(f) = HMIN_h(f) - f \quad (\text{A.18})$$

### A.1.9 Imposition de minimums

L'imposition de minimums consiste à imposer certains pixels de l'image à avoir la valeur d'intensité minimale de l'image et que seuls eux soient des minimums locaux. Ces pixels sont appelés des marqueurs et ils sont répertoriés dans un masque binaire  $g_m$  défini par :

$$g_m(\mathbf{x}) = \begin{cases} 0 & \text{si } \mathbf{x} \text{ appartient à un marqueur,} \\ t_{max} & \text{sinon.} \end{cases} \quad (\text{A.19})$$

L'imposition de minimums dans une image  $f$  à partir d'un masque  $g_m$  est une reconstruction par érosion notée :

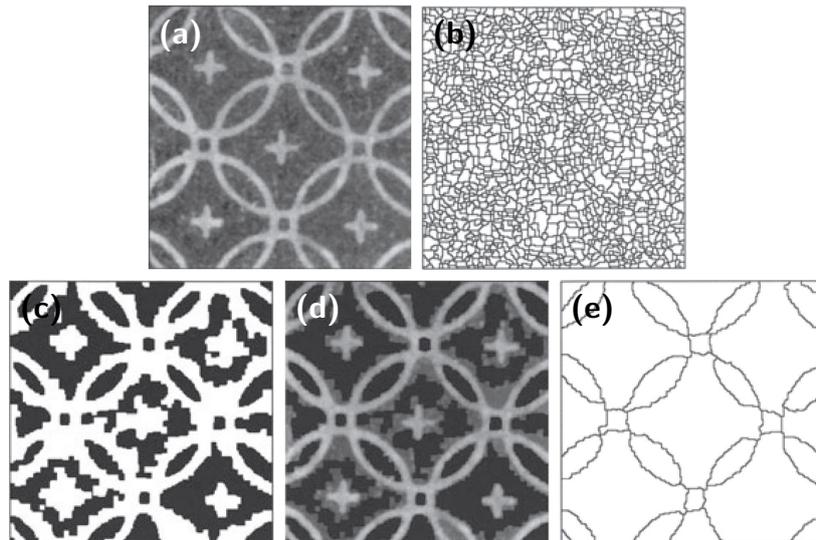
$$R_{(f+1)\wedge g_m}^{\varepsilon}(g_m) \quad (\text{A.20})$$

Une imposition de minimums est utilisée dans beaucoup d'opérations se basant sur des marqueurs, comme la segmentation par ligne de partage des eaux contrôlée par marqueurs.

### A.1.10 Segmentation par ligne de partage des eaux

En traitement d'images, une segmentation par ligne de partage des eaux (LPE) est une transformation définie dans une image à niveaux de gris. La transformation traite l'image comme un relief topographique (*i.e.* un pixel à haute intensité équivaut à un pic et un pixel à basse intensité à une vallée) dans lequel est simulée une inondation (il existe aussi d'autres méthodes que l'inondation mais nous n'en parlons pas ici). Le but de cette inondation est de calculer la ligne de partage des eaux (ou crête ou digue) qui sépare les bassins versants.

Pour segmenter des objets par LPE, un calcul de gradient est souvent appliqué au préalable à l'image. Ce calcul de gradient permet de fabriquer les bassins et de faire ressortir les lignes de crêtes en bordure des objets. Que ce soit avec ou sans calcul de gradient, le nombre de bassins et donc de crêtes dans l'image peut être assez important, c'est-à-dire présenter une sur-segmentation (Figure A.1b). Pour limiter cette sur-segmentation, certains bassins (ou minimums) non-significatifs peuvent par exemple être supprimés par l'application d'une opération *hmin*, ou alors il peut être utilisé des marqueurs indiquant quels bassins garder pour contrôler les sources d'inondation de la segmentation et donc le nombre de zones à mettre en évidence (Figure A.1c). Pour cette dernière solution, les minimums sont simplifiés à partir d'une image de marqueurs et d'une opération de reconstruction de type imposition de minimums (Figure A.1d). La segmentation par LPE est ensuite réalisée sur l'image simplifiée (Figure A.1e).



**Figure A.1** Segmentation par ligne de partage des eaux (LPE) contrôlée par des marqueurs. (a) image originale, (b) sur-segmentation quand la LPE est appliquée sur l'image originale, (c) image des marqueurs, (d) imposition des minimums localisée sur les marqueurs. (e) résultat de la LPE appliquée sur cette dernière image. Images tirées de Soille [1999].

## A.2 Corrélation croisée normalisée

La corrélation croisée normalisée est une corrélation croisée qui a été normalisée à l'image. Une corrélation croisée est une mesure de similitude entre une image et un modèle d'objet (ou motif ou *template*). Le fait de la normaliser permet de limiter les effets de modification de l'intensité lumineuse dans l'image pouvant être causée par les conditions d'éclairage et exposition. La corrélation croisée normalisée d'un *template*  $t(x, y)$  avec une sous-image  $f(x, y)$  est définie par :

$$\frac{1}{n} \sum_{x,y} \frac{[f(x, y) - \bar{f}][t(x, y) - \bar{t}]}{\sigma_f \sigma_t} \quad (\text{A.21})$$

où  $n$  est le nombre de pixels dans  $f(x, y)$  et  $t(x, y)$ , et  $\bar{f}$ ,  $\sigma_f$ ,  $\bar{t}$ ,  $\sigma_t$  sont respectivement la moyenne et l'écart-type de  $f$  et  $t$ .

C'est une méthode beaucoup utilisée en comparaison à un modèle d'objets (*template matching*) pour trouver des objets dans une image.

**B Article *PSFVIP* 2015**

## Image processing to study the evolution of channel slope and water depth in bimodal sediment mixtures

Hugo Lafaye de Micheaux<sup>1,2,4,\*</sup>, Ashley Dudill<sup>1,2,3</sup>, Philippe Frey<sup>1,2</sup>, Christophe Ducottet<sup>4</sup>

<sup>1</sup>Irstea, Unité de Recherche Erosion Torrentielle Neige et Avalanches, Grenoble, France

<sup>2</sup>Univ. Grenoble Alpes, Grenoble, France

<sup>3</sup>Department of Geography, University of British Columbia, Vancouver, Canada

<sup>4</sup>Laboratoire Hubert Curien, CNRS UMR5516, Université de Saint-Etienne, Université de Lyon, Saint-Etienne, France

\*corresponding author: hugo.lafaye-de-micheaux@irstea.fr

---

**Abstract** Bedload transport is the sediment load transported in contact with the bed of a river channel. Empirical relations are often used to predict sediment transport rates. However, these predictions usually lack accuracy. Consequently, a more detailed comprehension of the physical processes is required. For example, we require an improved understanding on how fine sediment inputs to river channels influence sediment transport, channel stability, ecology and stratigraphy. In this context, an experimental investigation was undertaken in a laboratory flume equipped with controlled water and sediment supplies to create bimodal mixtures of coarse and fine spherical glass beads. The flume experiments were recorded using a high frequency camera. We propose a new image analysis chain dedicated to automatically detect the water free surface and the bed elevation for a sequence of images. This method is based on a combination of morphological operations, gradient calculations and watershed transformations. The detections allowed the measurement of important physical quantities such as water depth, slope and their evolution over time. By making comparisons for varying sizes of fine sediment, we have highlighted several phenomena which permit a better understanding of bedload mobility, specifically related to the influence of grain size ratio.

**Keywords:** bedload, sediment transport, segregation, experimental flume, image analysis, line detection, marker-controlled watershed

---

### 1 Introduction

The transport of sediment through river channels has major consequences for public safety, management of water resources, and environmental sustainability. Most studies of sediment transport in rivers have focused on mass flux and its relation to water flow. But there is still no satisfactory theory for bedload transport, the sediment load transported in contact with the stream bed. Viewing bedload as a granular phenomenon may provide a better insight [1].

It is well known that when grains of different sizes are subjected to a shear flow, such as in a gravel-bed river, they will segregate, leading to fluvial morphological features [2]. There are different mechanisms of vertical size segregation: spontaneous percolation when fine sediment arrives in a static coarse sediment bed and falls into the voids between grains, and kinetic sieving when fine sediment arrives in a coarse sediment bed in motion and falls into voids which become dynamically available between the moving coarse grains. Both are influenced by the grain size ratio between the coarse sediment bed and the infiltrating finer sediment [3]. This paper will focus on kinetic sieving only.

We simulated a fluvial context, in a highly simplified manner through the use of an experimental laboratory, similar to the setup described in [4], using spherical beads and a narrow channel. We introduced a small concentration of fine beads into a bed formed solely of larger beads in equilibrium (input rate = output rate); this allowed us to visualise the segregation process. By varying the size of the fine beads, we were able to change the grain size ratio, and observe the influence upon segregation response.

The goal of this paper is to propose a new image analysis chain dedicated to automatically detect the water free surface and the bed elevation for a sequence of images. More precisely, this method allows to study over time the boundary of two main regions involved in sediment transport: the stable bed and

the mobile bed. Each boundary is detected in the form of horizontal lines in the observation window. The innovation here was to mix morphological operations with a combination of gradient calculations and watershed transformations [6] to detect those lines. The detected lines were then used to study the channel slope and water depth evolution over time for different grain size ratios.

The paper is structured as follows: first, we present the experimental facilities and procedure, and show what type of data is processed; then, after positioning the problem, we describe the image processing algorithm used to detect the bed and water free surface, and how we validate the results of the algorithm; finally, we study the channel slope and the water depth evolution over time for different grain size ratios.

## 2 Experimental study

The experiments were undertaken at IRSTEA (Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture), Centre de Grenoble, France. They were highly simplified and allowed clear visualisation of the phenomena and investigation at the scale of the particle.

### 2.1 Experimental setup

**Facilities.** The experimental setup consisted of a 2 m long, 10.3 mm wide, glass-sided flume set at a slope of 10.1 %. In order to cause bed formation, a 6.4 cm obstacle was placed at the channel outlet. The base of the flume is composed of different combinations of steel half cylinders, randomly arranged at different levels. The random arrangement of the bed prevents the occurrence of geometrically regular lattice (crystallographic) arrangements within the channel. This setup (Fig. 1) was also used by *Hergault et al.* [4] but had been widened to better study the infiltration of fine particles.

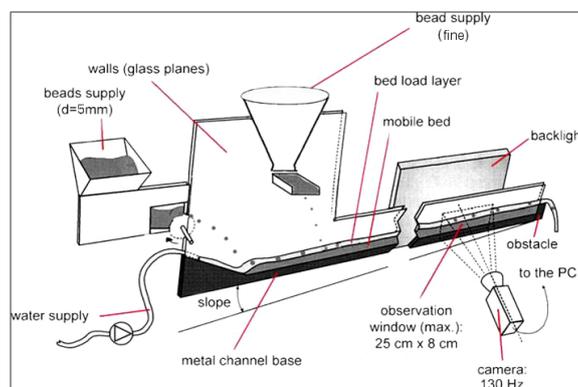


Fig. 1 A schematic diagram of the experimental flume for bimodal sediment mixture (after [4])

The bimodal sediment mixture was obtained by feeding coarse and fine spherical glass beads from two different bead distributors. The coarse bead diameter was 5 mm and the fine one was variable (0.7 mm, 0.9 mm, 1.5 mm, 2 mm, 3 mm and 4 mm) to modify the grain size ratio. All beads had a density of 2500 kg/m<sup>3</sup>.

The flume walls were made of glass in order to allow individual and bulk grain movements to be clearly visualised and tracked; this was done using a black and white partial scan motion camera. This Photonfocus motion camera was placed perpendicular to the glass planes of the flume, and inclined at the same angle. It was operated at a resolution of 1024 x 500 pixels and a frame rate of 130 frames/second. Stable and uniform backlighting for the camera was provided by a LED panel on the opposite side of the flume.

**Parameters.** The water flow was turbulent and supercritical. Flow rate and coarse particle transport rate were kept constant at respective values of 0.0495 liter/s (4.80 l/s/m) and 8 kg/hr ( $8.63 \cdot 10^{-2}$  l/s/m). For all experiments with different diameters presented in this paper, the fine particle rate was also kept constant at approximately 2.3 kg/hr, which represented 22.2% of the total feed (coarse plus fine). The high slope of the flume resulted in a low water depth, and a low submergence ratio (water depth to bead diameter). The Shields number was approximately 0.09, *i.e.* about twice the threshold for incipient motion, at the beginning of the experiment, before the fine sediment was introduced.

## 2.2 Experimental procedure

During the experiments, coarse grains were in motion, so that fine grains fell into dynamically available open spaces ('kinetic sieving' process). First, the coarse sediment and water feeds were turned on to build up the bed to the height of the obstacle at the channel outlet. Experiments were undertaken in 'equilibrium' conditions, meaning that the bed slope was constant and that the feed rate was equal to the transport rate over a long time period. Once the coarse sediment was in transport equilibrium, the fine feed was introduced. Meanwhile, a sequence of images was taken using the camera so that the interaction between the fine and coarse sediments could be visualised. The experiment ran until a new equilibrium was reached (lasted approximately 18-20 minutes). This new equilibrium was characterized either by a lower (degradation) or higher (aggradation) bed slope. The experiment was repeated for each size of fine sediment.

## 2.3 Data type

Data obtained during the recording of the infiltration period formed a  $\approx 20$ -minute sequence of black&white images (Fig. 2) at 130 frames/second. For the study reported in this paper, we did not need that many images. After the infiltration of the fine sediment into the coarse bed was nearly complete, the fine sediment formed a moving carpet-like layer of variable thickness above which coarse particles were essentially rolling (except in the 4 mm case). We kept only one image per second. So, the sequence was composed of  $\approx 1200$  images of size 1024 x 500 pixels. On images, the coarse sediment was transparent and the fine appeared in dark gray (except the 3 mm and 4 mm fine beads, which appeared as black and transparent, respectively). The upper dark line was the water free surface. The steel base of the flume can be seen with a dark color.

## 3 Detection of bed line and water line

### 3.1 Positioning of the problem

An image processing algorithm was developed to evaluate the evolution of the channel over time. Image processing allows the computation of a high number of operations efficiently on many images with high precision. In this work, the aim was to extract the boundaries of the principal regions appearing in sediment transport (Fig. 3):

- the base line which is the limit between the fixed bottom bed and the evolving bed (yellow to blue boundary on Fig. 3),
- the bed line which is the limit between the evolving bed and the water flow (blue to green boundary on Fig. 3),
- the water line which is the limit between the water flow and the upper external air (green to red boundary on Fig. 3).

Classically such segmentation problems can be solved using either contour detection or region segmentation. Many algorithms can be used but here we have chosen a morphological approach based on

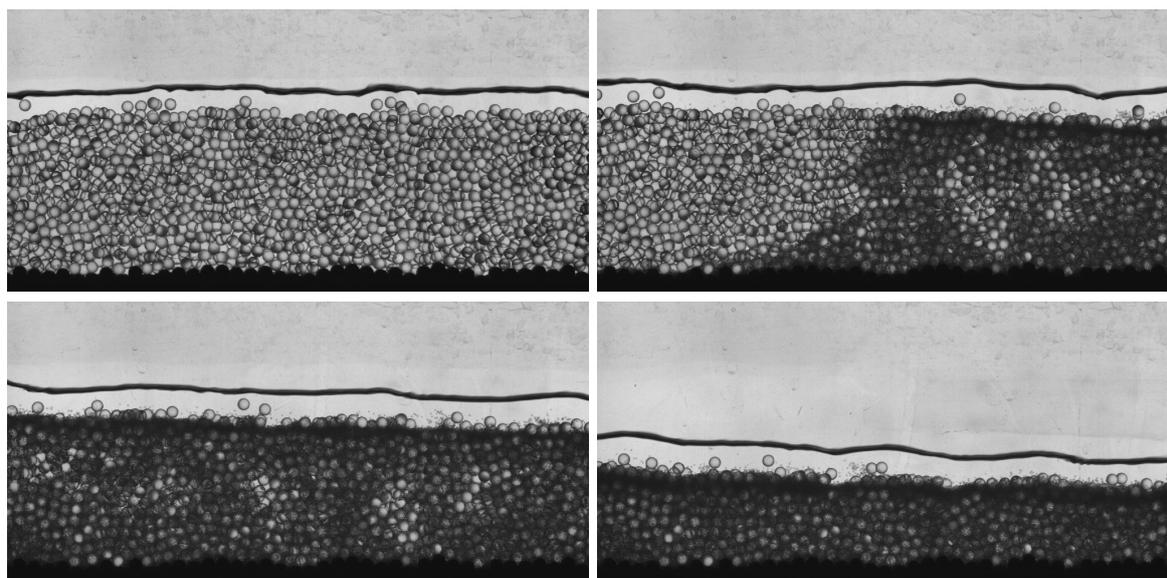


Fig. 2 Example of images from a sequence obtained with the experimental flume showing the evolution of the bed during the infiltration of fine particles of 0.7 mm. Images appeared respectively (from top left to bottom right) at: 0 min, 5 min, 7 min, 20 min. Flow from right to left.

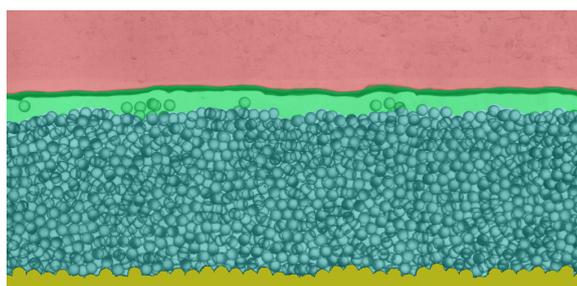


Fig. 3 Position of the 4 distinct regions in the images: the fixed base (yellow) corresponds to random half-cylinders; the mobile bed (blue) corresponds to the quasi-immobile beads composing the bed; the water flow (green) contains the fluid and the fast moving objects; the upper part is the air (red).

the watershed transformation [6]. This approach basically consists in computing the modulus of the gradient and then to detect the crest line separating two basins located on either side of the line. A great advantage of the method is that it provides the detection of a single connected line thanks to the definition of relevant markers used as flooding sources.

The combination of morphological watershed and gradient is a powerful segmentation method first introduced by *Beucher et al.* [6] and refined in different works ([7] or [8]). It relies on the use of the watershed transformation which determines the boundaries of adjacent objects as ridges. To avoid oversegmentation, the watershed is applied on the gradient of the image and is based on a set of markers. This method was successfully used in many segmentation problems such as road segmentation [10], video segmentation [11] and medical image segmentation [12].

### 3.2 Algorithm description

To detect the bed surface and the water free surface on the images, we first detected the position of the base line on the first image of the sequence. Then, on each image, we removed the moving beads with a

background extraction algorithm. Finally, we estimated the bed line and the water line positions.

**Base line detection.** The first step of the algorithm was to detect the base line. This was done only once - on the first image - and was subsequently used for all the images of the sequence. The principle of this algorithm was to apply a binary threshold to detect the fixed base region. Then, the upper boundary of this region was obtained by simple binary morphological operations. Although the fixed base was the darker region of the image, we had to remove the contribution of the water line and the mobile bed which were also dark. This was done using pre-processing. We proceeded with the following steps illustrated in Fig. 4:

1. Apply a morphological closing to the first image of the sequence (Fig. 4b). A vertical line was used as a structuring element, just long enough to remove the water free surface. The height of the line is denoted  $h$ . In our setup, the optimal value to remove the water surface was  $h = 18$  pixels.
2. Filter the obtained image with a Gaussian filter of standard deviation  $\sigma$  to get rid of fine details and noise (Fig. 4c). The noise was mainly due to the transparent beads in the bed which was successfully removed by taking  $\sigma = 3$  pixels.
3. Threshold the Gaussian filtered image (Fig. 4d). The threshold was automatically set to the first peak of the pixel intensity histogram to remove unwanted gray parts.
4. Dilate the binary image with a vertical line of 2 pixels as a structuring element.
5. Subtract the binary image formed in point 3. from the dilated image to highlight the edges between black and white (Fig. 4e). The resulting image gave the base line position.

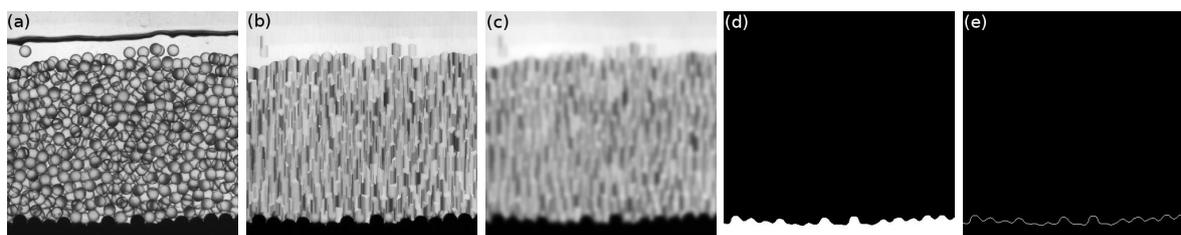


Fig. 4 Steps of the base line detection algorithm. **a** original image, **b** morphologically closed image, **c** filtered image, **d** thresholded image, **e** final result after subtracting image (d) from its dilatation.

**Background extraction.** The second step was to calculate the background of an image by averaging over  $P$  consecutive images (starting  $(P - 1)/2$  images before and ending  $(P - 1)/2$  after the current image). The effect of the averaging is to remove the moving bed which consists of all the beads moving during the time interval around the current image (see Fig. 5b). The number  $P$  of images was chosen to be a good compromise between having enough images to have a good map of moving particles, but not too many which would result in information at the bed surface being lost. In experiments, a value of  $P = 30$  was chosen because it covered 30 seconds of the experiment.

**Bed line detection.** The third step was to detect the bed line on the background image. Here, the principle was to detect the line by computing a Sobel gradient [9] and to detect the crest line of this gradient through a marker based watershed transformation. As for the base line detection, some pre-processing steps were undertaken. The procedure, illustrated in Fig. 5, was:

1. Apply a morphological closing to the background image using a vertical line of height  $h = 18$  pixels as a structuring element to remove the water free surface (Fig. 5c).

2. Filter the obtained image with a Gaussian filter of standard deviation  $\sigma = 3$  pixels to get rid of fine details and noise (Fig. 5d).
3. Calculate the vertical gradient on the Gaussian filtered image (Fig. 5e). A vertical Sobel gradient operator was used here to see vertical variations in pixel intensity.
4. Compute the watershed transformation on the complement of the gradient image with the top of the image and the base line as markers (Fig. 5f). The resulting image gave the bed surface position.

**Water line detection.** The last step was the detection of the water line. As for the bed line, the idea was to compute a watershed transformation to detect the water line as the crest line of the free surface region. For that, we need a grayscale image with high values on the free surface region. Starting from the results obtained after the bed line detection, we proposed the following process:

5. Subtract the original image from the Gaussian filtered image obtained at point 2 of the bed line detection (Fig. 5g). The resulting image showed high pixel values near and on the water free surface.
6. Compute the watershed transformation with the top of the image and the bed line as markers (Fig. 5h). The resulting image gave the water free surface position.

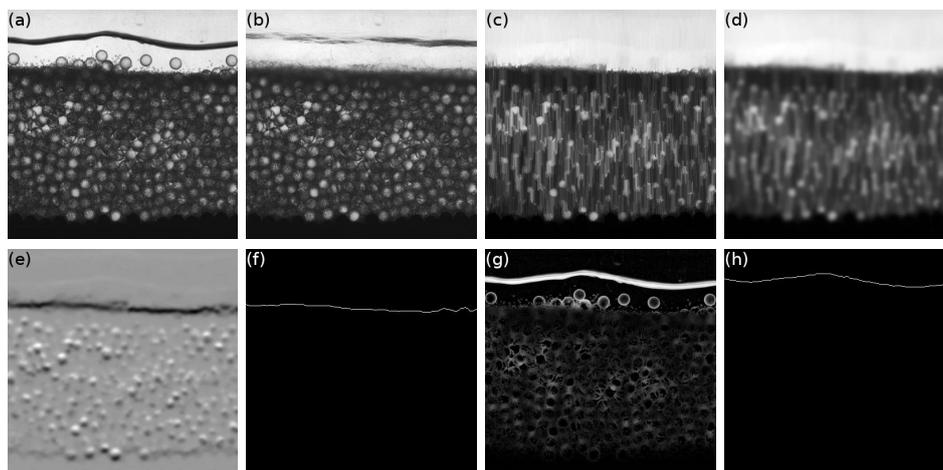


Fig. 5 Steps of the bed line and water line detection algorithms. **a** original image, **b** background image **c** morphological close of the background image, **d** filter of the closed image, **e** gradient of the filtered image, **f** result of the watershed transformation applied on the complement of the gradient image, **g** image obtained by subtracting the original image (a) from the filtered image (d), **h** result of the watershed transformation applied on the image (g).

### 3.3 Validation

In this section, we present two ways to validate the results of our algorithms: one way is based on the visual check of the detected lines, the other way is a statistical evaluation.

**Visually.** One method to validate the results was to check the detected lines visually. We created a sequence of images where the three detected lines - base, bed and water line - were all superimposed on the corresponding image. It allowed us to verify the precision of the position of the detected lines with the naked eye. In Fig. 6, we can see examples of line detections where the three lines appear to be well detected. But with this method, we could not review the whole sequence as it was more than 1000 images long.

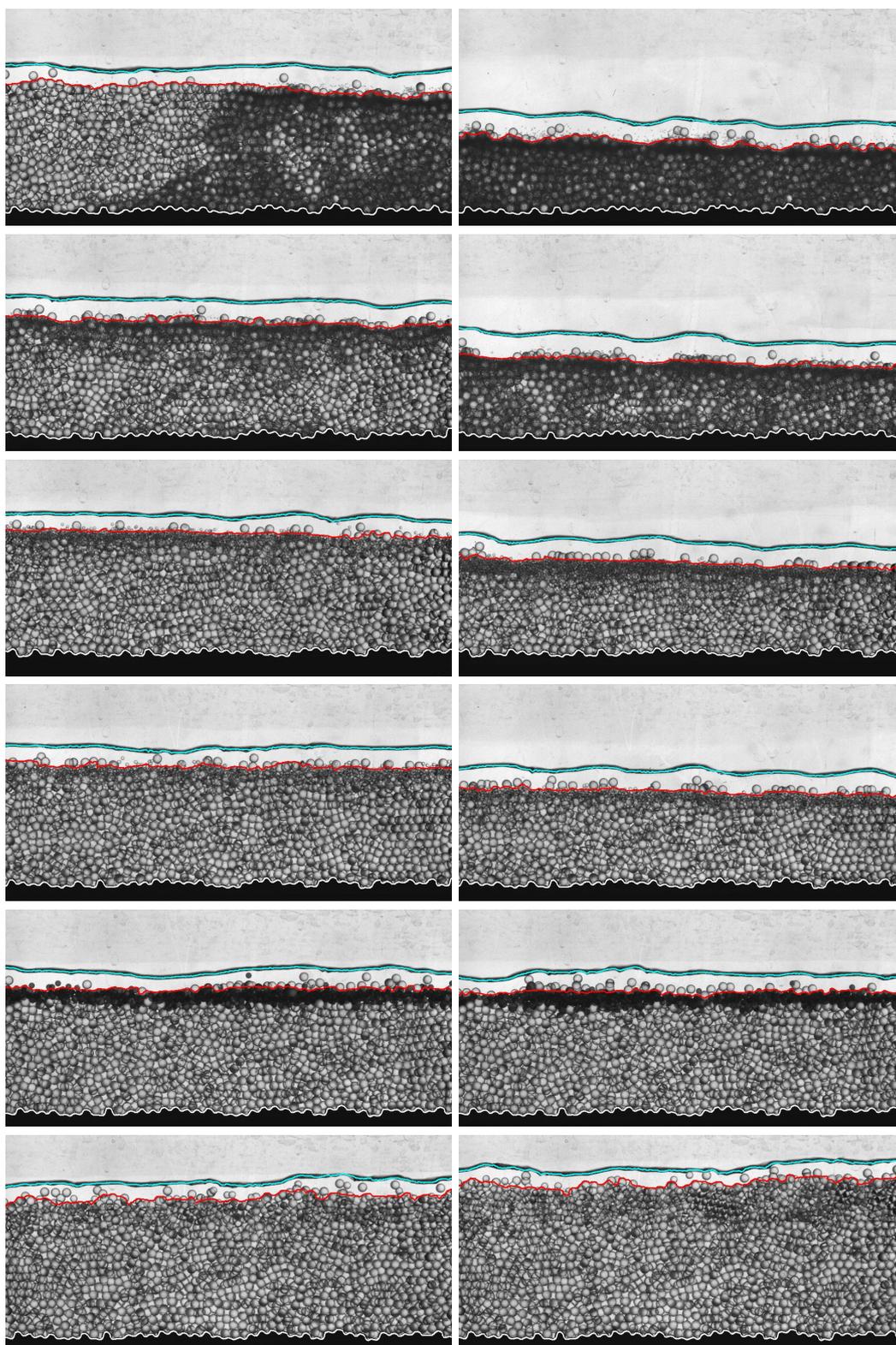


Fig. 6 Examples of detected base line (in white), bed line (in red) and water line (in cyan) superimposed on the corresponding image at 5 min (left column) and 15 min (right column) during the infiltration of fine beads of 0.7 mm, 0.9 mm, 1.5 mm, 2 mm, 3 mm and 4 mm (respectively in this order from top to down).

**Statistically.** A second method to verify the integrity of the results was to use descriptive statistics along the whole sequence. Here we calculated the mean elevation of the detected bed line for each image of the sequence together with the standard deviation and the local standard deviation. By keeping only the maximum value of the local standard deviations, we were able to detect if bad detections occurred (a high value would correspond to a jump in the line and so would reveal probably an error). In Fig. 7, we can see two graphs with these values applied on the sequences where fine beads of 0.7 mm and 4 mm were introduced. We made the same graph for all other sequences to conclude that there were apparently no errors in the bed line detections (no high isolated peaks in the graphs). In the 4 mm case, we can see higher maximum local standard deviations which was due to the nature of the fine sediment layer on the bed surface. In this case, the bed surface was more irregular (due to the small grain size ratio) and therefore the detector was less accurate.

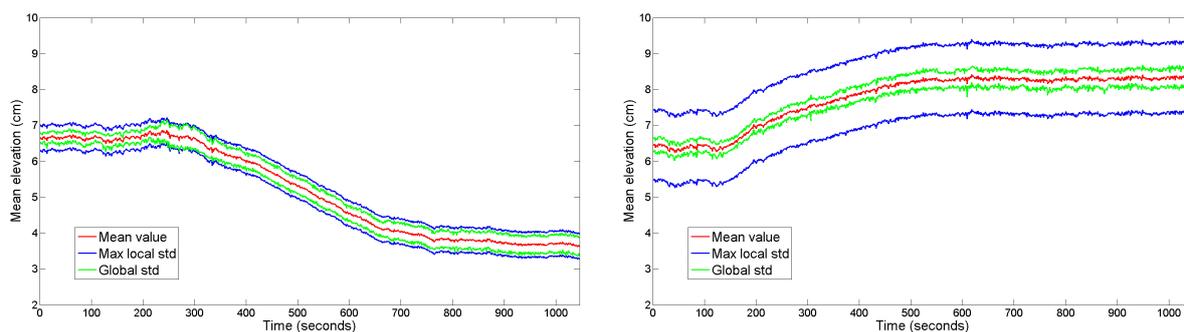


Fig. 7 Mean elevation of the detected bed lines over time with maximum local standard deviation and global standard deviation during the infiltration of fine beads of 0.7 mm (left) and 4 mm (right).

## 4 Results

In this section, we present the results obtained and the analysis performed using our algorithms. To understand why we concentrated our study on the channel slope and the water depth, let's consider the real case of sediment transport. In mountains, torrents adapt their slope, depth and width to maintain a balance between the water and sediment supplied from upstream, and transported downstream. When catastrophic floods occur, a huge quantity sediment is transported in mountain streams which may impact their stability. According to the size and the grain size distribution of these sediments, the impact on the streams may result in different modifications in slope, depth and width. These modifications are what we aim to understand. In our experimental setup, the width could not vary, so we focused our study on the channel slope and water depth.

### 4.1 Channel slope evolution

One main purpose that motivated the development of the described image processing algorithm was to evaluate the influence of fine sediment size on the sediment mobility of the bed. Studying the channel slope over time is one way of doing that. The calculation of the channel slope was based on the bed line detection results. The line between the upper extremity of the outlet stopper and the mean bed height positioned in the middle of each image was considered. The slope of this straight line was defined as the channel slope. The slope was calculated on each image of all sequences in order to compare them. In Fig. 8, we can see the evolution of the channel slope for all of the sequences, each using a different size of fine particle. The channel slope with fine particles of size 0.7 mm, 0.9 mm, 1.5 mm and 2 mm decreased, indicating that the channel bed degraded. This degradation can be explained by the fine particles filling the interstices of the bed, enhancing the exposure of the coarse particles to the flow,

therefore increasing their mobility. Conversely, when the fine component was formed of 4 mm beads, the channel aggraded (increasing the channel slope).

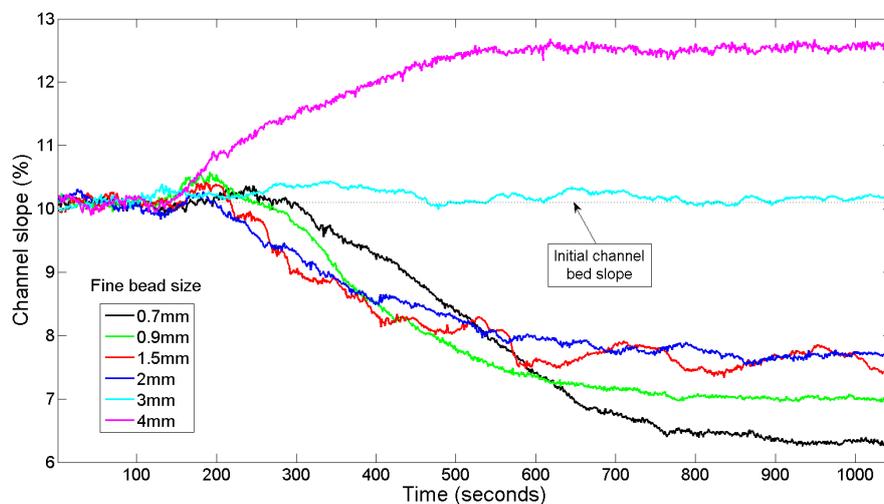


Fig. 8 Evolution of channel slope over time according to the size of fine sediment. The dotted line represents the initial channel bed slope at equilibrium, so at 10.1 %.

#### 4.2 Water depth evolution

Another objective of this study was to examine the evolution of water depth during the fine sediment introduction period and see the influence of the fine sediment size. In Fig. 9, we can see the evolution of water depth over time for all sequences. It appears clearly that the smaller the fine beads, the higher the water depth. This can partly be explained by the reduction of slope when finer sediment is introduced. However, there may also be an opposite effect of fine sediment on the flow resistance.

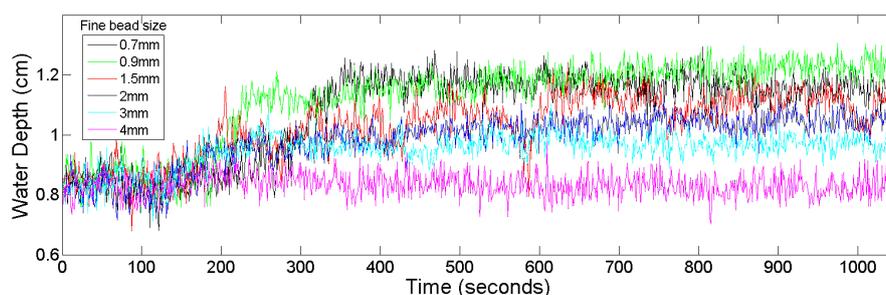


Fig. 9 Evolution of water depth over time according to fine sediment size.

## 5 Conclusion

In this paper, we have presented a novel approach for analysing bedload mobility in a laboratory experiment with image processing based algorithms. We used an inclined channel in which sediment and water supplies at the inlet were controlled accurately. A bimodal sediment mixture was obtained by feeding one-sized coarse beads of 5 mm and then one-sized fine beads of variable size across experiments (0.7 mm, 0.9 mm, 1.5 mm, 2 mm, 3 mm and 4 mm). Each experiment was filmed with a camera and gave a temporal sequence of images. Image processing was used as it enabled us to automatically

detect features on images, and to deal with a large number of images. Our approach relies on the use of simple morphological operations such as closing, dilating, filtering and more advanced ones like gradient calculation and watershed transformation. Combining them, we were able to detect lines separating distinct regions on images: the base support, the mobile bed, the water flow and the air. These lines allow us to study the evolution of channel slope and water depth over time, and to conclude that grain size ratio has an important influence upon the bedload mobility.

As perspectives, the flow resistance could be studied by exploiting water depth evolution. Also, other experiments with different fine particle rates could be done to evaluate the influence of fine sediment feed concentration on both slope and water depth.

**Acknowledgements.** This research was supported by *Irstea* (formerly *Cemagref*), by the French *Institut National des Sciences de l'Univers* programme EC2CO-BIOHEFECT, by the *Natural Sciences and Engineering Research Council of Canada* (NSERC) and by the *Rhône-Alpes* region as part of its higher education, research and innovation regional strategy (*Environment Academic Research Community*).

## References

- [1] Frey, P., Church, M. (2011). Bedload: a granular phenomenon. *Earth Surface Processes and Landforms*, 36(1), pp 58-69. doi: 10.1002/esp.2103
- [2] Parker, G., Klingeman, P. C. (1982). On why gravel bed streams are paved. *Water Resources Research*, 18(5), pp 1409-1423. doi: 10.1029/WR018i005p01409
- [3] Thomas, N. (2000). Reverse and intermediate segregation of large beads in dry granular media. *Physical Review E*, 62(1), pp 961. doi: 10.1103/PhysRevE.62.961
- [4] Hergault, V., Frey, P., Métivier, F., Barat, C., Ducottet, C., Böhm, T., Ancey, C. (2010). Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow. *Experiments in fluids*, 49(5), pp 1095-1107. doi: 10.1007/s00348-010-0856-6
- [5] Böhm, T., Frey, P., Ducottet, C., Ancey, C., Jodeau, M., Reboud, J. L. (2006). Two-dimensional motion of a set of particles in a free surface flow with image processing. *Experiments in fluids*, 41(1), pp 1-11. doi: 10.1007/s00348-006-0134-9
- [6] Beucher, S., Lantuéjoul, C. (1979). Use of watersheds in contour detection. *In International Workshop on Image Processing, Real-Time Edge and Motion Detection*, Rennes, France, Sept.17-21
- [7] Beucher, S., Meyer, F. (1992). The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated*, 34, pp 433-433
- [8] Vincent, L., Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), pp 583-598. doi: 10.1109/34.87344
- [9] Sobel, I., Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *Presented a talk at the Stanford Artificial Intelligence Project*
- [10] Beucher, S., Bilodeau, M. (1994). Road segmentation and obstacle detection by a fast watershed transformation. *In Intelligent Vehicles' 94 Symposium*, pp 296-301
- [11] Patras, I., Hendriks, E. A., Lagendijk, R. L. (2001). Video segmentation by MAP labeling of watershed segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), pp 326-332. doi: 10.1109/34.910886
- [12] Grau, V., Mewes, A. U. J., Alcaniz, M., Kikinis, R., Warfield, S. K. (2004). Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging*, 23(4), pp 447-458. doi: 10.1109/TMI.2004.824224

**C Article *JGR : Earth Surface* (en révision)**

# Introducing finer grains into bedload: the transition to a new equilibrium

Ashley Dudill<sup>1,2,\*</sup>, Hugo Lafaye de Micheaux<sup>2,3</sup>, Philippe Frey<sup>2</sup>, Michael Church<sup>1</sup>

<sup>1</sup> Department of Geography, University of British Columbia, Vancouver, Canada

<sup>2</sup> Université Grenoble Alpes, Irstea, UR ETGR, St-Martin-d'Hères, France

<sup>3</sup> Université de Lyon, UJM-Saint-Étienne, CNRS, UMR5516, Laboratoire Hubert Curien, Saint-Étienne, France

Corresponding author: Ashley Dudill ([ashley.dudill@gmail.com](mailto:ashley.dudill@gmail.com))

Submitted on October 26, 2016 to *Journal of Geophysical Research: Earth Surface*

## Key points:

- The grain size ratio between a fine input and a coarse bed dictates sediment mobility.
- The percentage of fines in the feed also determines the transition from uniform coarse equilibrium to two-size equilibrium.
- Collapse of bed slope response curves of fine grain inputs of varying diameter demonstrates similarity of bed response.

**Abstract.** Simplified experiments with fine grain inputs to a coarse bed in mobile equilibrium were undertaken in a small, narrow flume using spherical glass particles. The experiments demonstrate the influence of the size ratio between the bed ( $D_c$ ) and the input ( $D_f$ ) upon the channel response. Size ratios ( $D_c/D_f$ ) between 7.14 and 1.25 were tested, with a constant flow and coarse sediment feed rates, and a variety of fine feed rates. Transition to a new, two-size equilibrium occurs through slope adjustment (aggradation/degradation), accompanying a change in sediment mobility. Previous work has documented superior mobility as a result of a fine grain input; the experiments presented herein identify boundaries within this behaviour related to both the fine grain diameter and the proportion of fines introduced. Despite the variation in bed slope response depending upon these factors, the data collapse exponentially.

**Keywords.** Fluvial sediment, Transport experiment, Bedload, Equilibrium slope, Grain size ratio

## 1. Introduction

The influence of fine sediment inputs to gravel bed rivers has been heavily investigated due to implications for sediment transport rates [e.g. *Curran and Wilcock*, 2005], channel morphology [e.g. *Jackson and Beschta*, 1984], stratigraphy [e.g. *Frostick et al.*, 1984] and fluvial ecology [e.g. *Lisle*, 1989]. In addition to the applied motivations, the problem is scientifically intriguing. Imagine a channel with a mobile uniform bed (constant grain diameter) in equilibrium, whereby the sediment input rate is equal to the sediment output rate. An increase in sediment supply to the channel, with the additional material being finer than that forming the bed, has been shown to

cause an increase in the coarse sediment transport rate in comparison to the preceding uniform conditions. The superior mobility of grain size mixtures was first demonstrated by *Gilbert* [1914], and has since been explored experimentally [*Jackson and Beschta*, 1984; *Iseya and Ikeda*, 1987; *Ikeda and Iseya*, 1988; *Wilcock et al.*, 2001; *Cui et al.*, 2003; *Curran and Wilcock*, 2005; *Venditti et al.*, 2010a, 2010b], and also observed in the field [e.g., *Ferguson et al.*, 1989; *Montgomery et al.*, 1999; *Major et al.*, 2000].

The increase in transport rate resulting from a fine sediment addition may even be so large as to increase the sediment output rate above the sediment input rate, despite the increased supply to the system, which leads to channel degradation and a reduced equilibrium bed slope [*Iseya and Ikeda*, 1987; *Ikeda and Iseya*, 1988; *Curran and Wilcock*, 2005]. In contrast, an increase in sediment supply to the channel with the additional material being the same size as that forming the bed, would result in a steeper channel slope [*Lane*, 1955], as more energy would be required to transport the larger load.

The superior mobility of mixtures has been attributed to several causes. (1) Enhanced entrainment potential: *Houssais and Lajeunesse* [2012] demonstrated that, in a bimodal bed, an increasing fraction of fine sediment on the bed surface leads to a reduction in the critical Shields number for the coarse component. (2) Elimination of deposition locations: infilling of the bed surface with fines removes potential deposition locations [*Venditti et al.*, 2010b], consequently, once a particle is entrained, it is less likely to stop again. (3) Changes in the flow structure: infilling of fines on the bed surface results in a smoother top layer, which has been observed to cause flow acceleration close to the bed, resulting in an increased drag force on the coarse particles [*Venditti et al.*, 2010b]. (4) Pivoting angle: *Komar and Li* [1986] noted that, for a grain to be entrained, it must pivot about a contact point with an underlying grain. This ‘pivoting angle’ is a function of the ratio of the grain diameter to the diameter of the underlying grain, with a greater difference in diameter leading to a smaller pivoting angle. (5) Exposure: The projection of a larger grain above a bed formed of finer material increases the exposure of the grain to the flow [*Fenton and Abbott*, 1977], resulting in a lower critical Shields number for the larger grains when compared to uniform conditions [*Parker and Klingeman*, 1982]. The finer grains have a higher critical Shields number as they are ‘hidden’ from the flow.

The extent of the increase in sediment transport rate following the introduction of fines to a channel has been shown to depend upon the proportion of fines in the total feed [*Jackson and Beschta*, 1984; *Iseya and Ikeda*, 1987; *Ikeda and Iseya*, 1988; *Wilcock et al.*, 2001, *Curran and Wilcock*, 2005]. However, results vary. For example, *Iseya and Ikeda* [1987] held the gravel feed rate constant, and varied the sand feed rate between 0 and 58 % of the total feed. They observed a reduction in the equilibrium slope between 0 and 48 % sand content, and then a slope at 58 % no different than that at 48 %. *Curran and Wilcock* [2005], using the same method, saw a reduction in the equilibrium bed slope between 33 and 72 % sand content, and then a more gradual decrease between 72 and 86 %. We hypothesise that the results vary due to the different grain size ratios used in the experiments.

Examination of how the grain size ratio (coarse sediment forming the bed,  $D_c$ : fine sediment introduced,  $D_f$ ) influences the superior mobility of mixtures has received limited attention despite *Gilbert* demonstrating its importance in 1914 (but see *Venditti et al.*, 2010a). The influence of grain size ratio ( $D_c/D_f$ ) upon the amount of fine infiltration into a static coarse bed, however, has been heavily investigated [*Beschta and Jackson*, 1979; *Frostick et al.*, 1984; *Diplas and Parker*, 1992; *Gibson et al.*, 2009], demonstrating that a larger grain ratio results in a greater amount of infiltration.

New work by *Dudill et al.* [in review] examined fine infiltration into a mobile coarse bed at a range of grain size ratios, and hypothesised that the enhancement of mobility following a fine grain input is, in part, controlled by the infiltration behaviour, and therefore is a function of  $D_c/D_f$ . The justification for this hypothesis is as follows: within the mobile bedload transport layer, ‘kinetic sieving’ [*Middleton, 1970*] occurs. Kinetic sieving is the process of fine particles moving down from the bed surface by falling into voids which become dynamically available between the coarse moving grains. Consequently, the fine grains move down to the base of the mobile layer and the coarse grains are displaced upward. At the bottom of the bedload layer, the fine sediment reaches the interface with the underlying quasi-static coarse bed; if the geometry permits, the fines infiltrate into the bed through ‘spontaneous percolation’ [*Bridgwater et al., 1969*]. Spontaneous percolation is the process of fine particles infiltrating into an immobile coarse bed through the void spaces. If the amount of fines arriving at this interface either greatly exceeds the amount that can infiltrate into the coarse bed or, if the fines are too large in diameter to spontaneously percolate into the bed, a quasi-static fine layer is created on top of the coarse deposit. The presence or absence of the quasi-static fine layer was observed to be critical in determining the channel response (degradation or aggradation) to a fine sediment input.

Understanding how grain size ratio influences sediment mobility following a fine grain input is an important research gap. Here, we quantitatively analyse the channel response in the experiments presented in *Dudill et al.* [in review], along with additional experiments, all of which were undertaken with two-size mixtures of spherical glass beads in a specialised flume. The paper investigates the transition from initial to final transport equilibrium with a variety of fine feed proportions (fine feed/total feed), over a range of grain size ratios ( $D_c/D_f$ ). This is achieved through examination of the adjustment in bed slope following a fine grain input. Further analysis is presented in the form of sediment output rates from the channel, the characteristics of the bedload movement and the vertical bed composition.

## 2. Experimental Methods

### 2.1 Experimental Arrangement

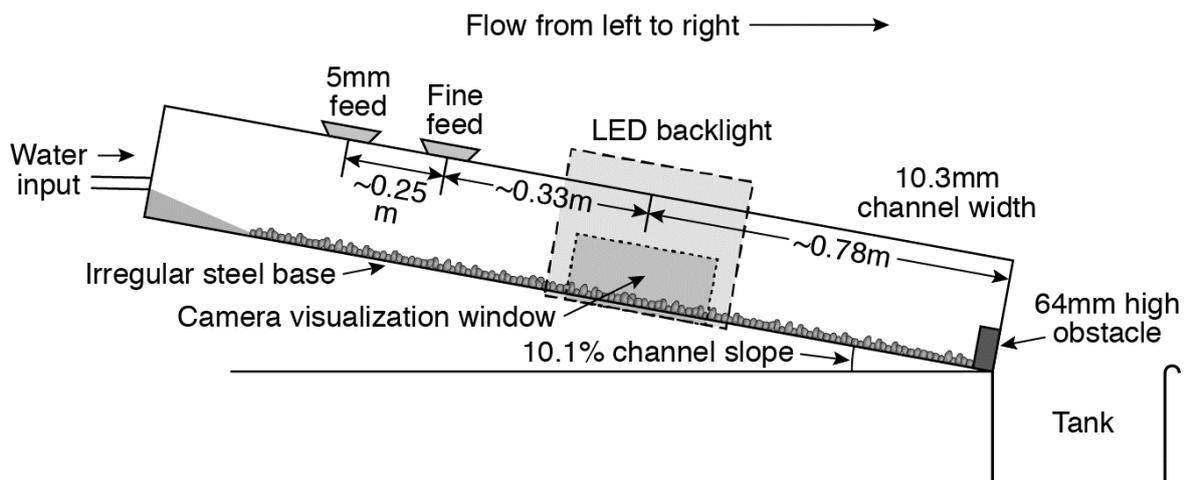
Experiments were undertaken using artificial media (spherical glass beads) in a channel (see Figure 1) located at the National Research Institute of Science and Technology for Environment and Agriculture (Irstea), Grenoble, France. The channel has a steel frame, glass sides, and an irregular steel base to prevent a regular arrangement of grains developing within the sediment bed. The width is 10.3 mm to permit clear visualisation, and the channel slope is  $10.1 \pm 0.01$  % to ensure full mobility. To cause bed formation, a 64 mm high obstacle was inserted into the end of the channel. Water is supplied from a constant head reservoir and controlled using an electromagnetic flow meter. The reductionist approach utilised in these experiments enable examination and description of the basic processes, and therefore constitute a step toward developing a deeper understanding of sediment transport.

The diameter of the beads forming the coarse component of the two-size mixture was fixed at  $5.0 \pm 0.3$  mm, therefore ensuring consistent void spacing within the bed for all experiments. The diameter of the fine component varied; diameters of  $1.5 \pm 0.2$  mm,  $2.0 \pm 0.2$  mm,  $3.0 \pm 0.3$  mm

and  $4\pm 0.3$  mm were used. Additionally, two sets of material below 1 mm in diameter were used;  $D_{50}=0.7$  mm ( $D_{75}/D_{25}=1.15$ ) and  $D_{50}=0.9$  mm ( $D_{75}/D_{25}=1.12$ ). All of the beads used in the experiments have a density of  $2500\text{ kg/m}^3$ , and all are transparent, except the 3 mm beads, which are translucent blue. The beads in all experiments moved as bedload, the weight of the grains being supported by the sediment bed.

Two feeders were used to introduce sediment into the channel. The 5 mm feeder, a K-Tron® Bulk Solids Pump™ Loss-in-Weight feeder, was located 1.36 m upstream of the channel outlet, and the fine feeder, a ‘Tinker’ distributor, was 1.11 m upstream. The ‘Tinker’ distributor consists of a rotating pipe within a reservoir filled with beads. The inclination of the reservoir determines the feed rate.

All of the experiments were recorded using a Photon Focus® high-speed camera, provided by Alliance Vision®, which was positioned perpendicular to, and inclined to the same angle as, the channel. The camera recorded at 130 frames/second, and the images from the camera visualisation window are  $1024 \times 500$  pixels. A LED backlight panel was positioned behind the channel, approximately 0.78 m upstream from the outlet, providing stable and uniform lighting for the camera.



**Figure 1: Schematic diagram of the experimental arrangement, containing no beads. This channel was shown in *Dudill et al.* [in review] to produce sediment transport parameters in good agreement with classic semi-empirical bedload formulae.**

## 2.2 Experimental Procedure

In order to explore the question of how the proportion of fines within a sediment feed influences grain mobility, two types of sediment-feed experiments have been previously undertaken. The first type, used by *Ikeda and Iseya* [1988; chapters 3 and 6], involves fixing the total feed rate, and varying both the fine and coarse feed rates. The second type involves fixing the coarse feed rate, and varying both the fine and total feed rate [e.g. *Beschta and Jackson* [1984]; *Iseya and Ikeda* [1987]; *Ikeda and Iseya* [1988; chapter 4]; *Curran and Wilcock* [2005]]. The experiments presented in this paper were undertaken using the second approach for two reasons. First, from a scientific perspective, by fixing the coarse feed rate, the flow rate required for one-size equilibrium of the coarse-grain bed, which is the initial condition in these

experiments, does not vary. Second, from an applied perspective, this approach allows us to explore how the response of a channel varies if a source of fine sediment is introduced, depending upon the rate and grain size of that input.

A consistent procedure, with two stages, was followed in each experiment. In the first stage, one-size equilibrium was achieved with the 5 mm grains only. One-size equilibrium was achieved when the sediment input rate equaled the sediment output rate and the bed and water slope matched the channel slope. An example frame from the camera visualisation window of the bed in one-size equilibrium is shown in Figure 2a. The 5 mm feed rate and the flow rate remained constant throughout the experiment. Details are given in Table 1; the uncertainty in the reported hydraulic parameters is a result of the uncertainty in the mean flow depth and diameter of the coarse grains. In the second stage, fines of a fixed diameter and a fixed feed rate were introduced into the channel. The bed response to the fine grain input was filmed using the high-speed camera.

Twenty two experiments were undertaken in two sets:

- **Set 1:** For each grain size ratio combination ( $D_c/D_f$  ranging from 7.14 to 1.25), the bed response was examined with three different fine feed proportions (fine feed/total feed): approximately 22, 41 and 68 % (when the fine feed composes 50 % of the total feed, the coarse feed rate is equal to the fine feed rate). Experiments with 3 mm and 4 mm fine components at approximately 68 % were not feasible as the flow rate was not sufficient to transport the load. A summary of these experiments is given in Table 2: Set 1.
- **Set 2:** Three additional fine feed proportions were investigated with grain size ratios of 5.56 (0.9 mm fines) and 3.33 (1.5 mm fines). A summary of these experiments is given in Table 2: Set 2.

Following the fine sediment input, the channel passed through a transient phase to ultimately reach a new, two-size equilibrium bed slope, whereby the sediment output rate was again approximately equal to the sediment input rate for each size component. In most cases the channel had aggraded or degraded to reach this new equilibrium slope, consequently the bed and water slopes were no longer parallel to channel slope. Figures 2b and 2c show the bed in two-size equilibrium in Experiment 2 ( $D_f = 0.7$  mm) and Experiment 14 ( $D_f = 3$  mm) respectively. The difference between the height when the bed is in one-size equilibrium and when it is in two-size equilibrium is illustrated by the dashed white line.

During the Set 1 experiments, samples of the sediment output were taken for the entire duration of the experiment. The first sample of the 5 mm bed in one-size equilibrium was taken for two minutes. Once the fine sediment had been introduced to the channel, the samples were initially taken for 30 seconds in order to track the first arrival of the fines at the flume outlet. As the experiment progressed, the samples were taken for longer periods of time. The final sample of the bed in two-size equilibrium was taken for five minutes. The samples were oven dried and then sieved to separate the fine and coarse components to enable comparison between the input and output rates. The difference between the input and output rate for the coarse component was consistently  $\leq 5$  %. For the fine component, the difference was always  $< 9$  %. For the Set 2 experiments, two two-minute samples of the sediment output from the channel were taken at the end of each experiment to ensure that the bed had reached two-size equilibrium. The difference between the input and output rate for the coarse component was consistently  $< 6$  %. For the fine component, the difference was always  $< 8$  %, with the exception of one of the fine material samples in Experiment 20, when the difference was 10.9 %. However, over the two minute

sample, this is a difference of just 13.2g (with a fine feed rate of 1.01g/s), which is not substantial.

Using a combination of image analysis techniques, detailed in *Lafaye de Micheaux et al.* [2015], on the images taken by the high-speed camera, the bed elevation could be detected, permitting computation of the bed slope. Consequently, it is possible to document the evolution of the bed slope over time, from the fine grain input until the bed reached a stable two-size equilibrium state. This analysis was undertaken on one frame per second.

**Table 1: Experimental conditions for 5 mm one-size equilibrium<sup>a</sup>.**

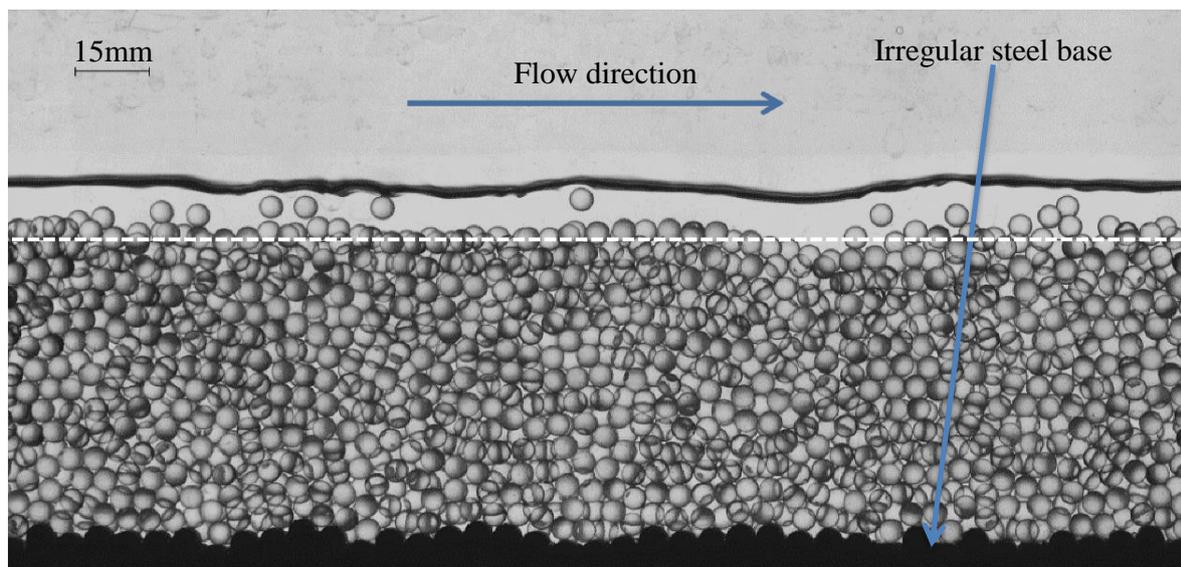
Flow rate (mL/s)	5 mm feed rate (g/s)	Mean flow depth, $h$ (mm)	Submergence ratio ( $h/D_c$ )	Side-wall corrected dimensionless shear stress <sup>b</sup>	Mean velocity, $U$ (m/s)	Froude number ( $\frac{U}{\sqrt{gh}}$ )	Reynolds Number ( $\frac{4UR_h}{\nu}$ )	Particle Reynolds Number ( $\frac{u_* D_c}{\nu}$ )	Rouse Number ( $\frac{w_s}{ku_*}$ )
49.5	2.22	8.02 ± 0.38	1.6 ± 0.18	0.076 ± 0.011	0.60 ± 0.03	2.14 ± 0.15	8420 ± 237	417 ± 25	16.3 ± 0.5

<sup>a</sup>Where  $h$  is the mean flow depth (measured between the water line and the stationary bed surface),  $D_c$  is the diameter of the coarse grains ( $5 \pm 0.3$  mm),  $R_h$  is the hydraulic radius,  $w_s$  is the fall velocity calculated using the Ferguson and Church (2004) equation,  $k$  is the von Karman constant (0.4),  $u_*$  is the shear velocity ( $u_* = \sqrt{\frac{\tau_o}{\rho}}$ ),  $\tau_o$  is the shear stress ( $\tau_o = \rho g R_h S$ ), and  $S$  is the slope.

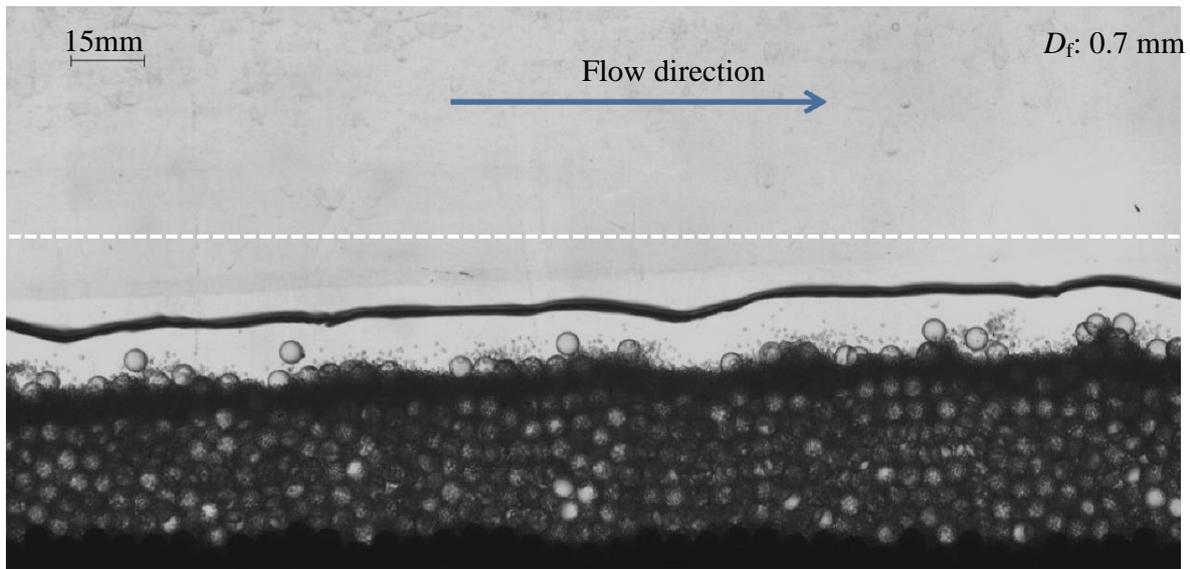
<sup>b</sup>The side-wall corrected dimensionless shear stress was calculated using a Blasius-type law for the friction factor of the smooth walls.

**Table 2: Experimental conditions for the fine grain inputs.**

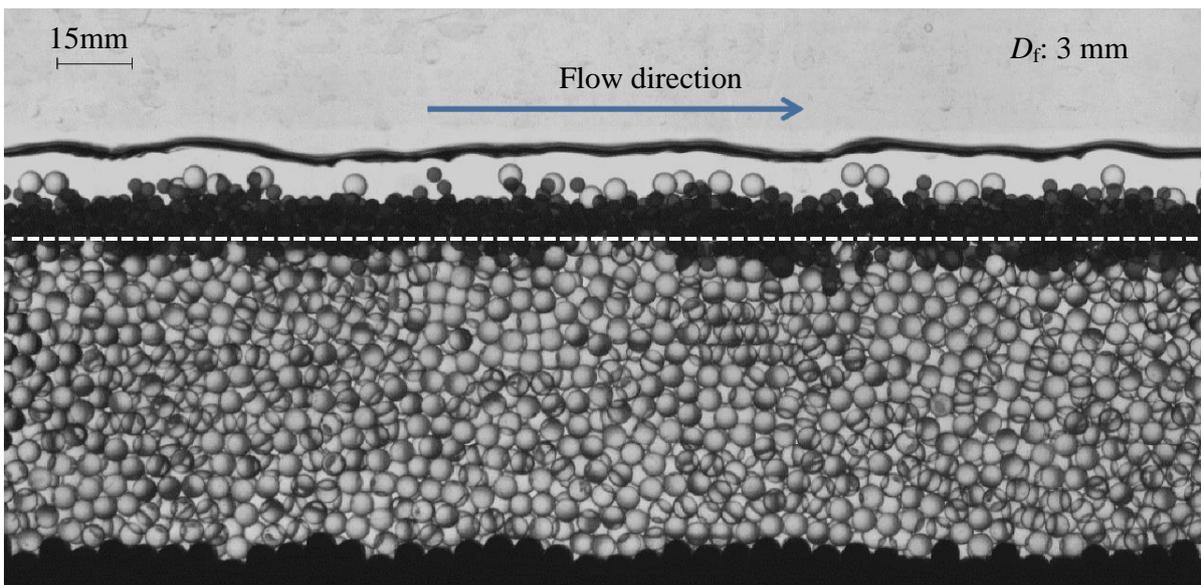
Experiment Set	Experiment number	Fine grain diameter ( $D_f$ ) (mm)	Grain size ratio ( $D_c / D_f$ )	Fine grain feed rate (g/s)	Total feed rate (g/s)	Fine feed rate / Total feed rate (%)
1	1	0.7	7.14	0.63	2.85	22
	2			1.54	3.76	41
	3			4.82	7.04	68
	4	0.9	5.56	0.62	2.84	22
	5			1.54	3.76	41
	6			4.92	7.14	69
	7	1.5	3.33	0.65	2.87	23
	8			1.60	3.82	42
	9			4.74	6.96	68
	10	2	2.5	0.62	2.84	22
	11			1.57	3.79	41
	12			4.72	6.94	68
	13	3	1.67	0.62	2.84	22
	14			1.59	3.81	42
	15	4	1.25	0.65	2.87	23
	16			1.55	3.77	41
2	17	0.9	5.56	1.04	3.26	32
	18			2.77	4.99	56
	19			3.69	5.91	62
	20	1.5	3.33	1.01	3.23	31
	21			2.70	4.92	55
	22			3.74	5.96	63



**Figure 2a**



**Figure 2b**



**Figure 2c**

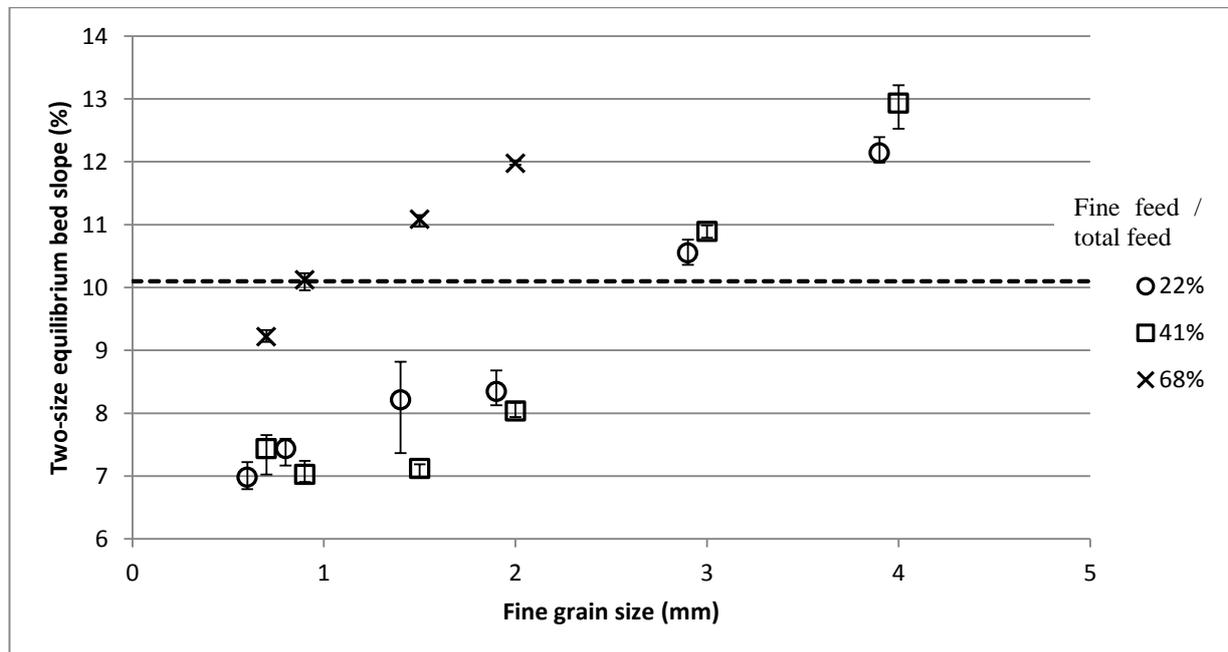
**Figure 2:** Example frame from the camera visualisation window of the bed: (a) in one-size equilibrium, and in two-size equilibrium during (b) Experiment 2 ( $D_f = 0.7$  mm, fine feed rate/total feed rate = 41 %) and (c) Experiment 14 ( $D_f = 3$  mm, fine feed rate/total feed rate = 42 %). The white dashed line in Figures 2a, 2b and 2c shows the bed height in Figure 2a, when the bed is in one-size equilibrium.

### 3. Results

#### 3.1 Set 1

Following the addition of the fines to the channel, the newly introduced fines with diameter  $\leq 2$  mm infiltrated into the bed, with greater amounts of infiltration occurring for smaller grain diameters. In the experiments with the 3 mm and 4 mm inputs, the fines were unable to infiltrate into the bed (see Figure 2c).

Figure 3 presents the two-size equilibrium bed slope measured manually for each of the experiments in Set 1, permitting comparison under various conditions. Experiments with final two-size equilibrium slopes less than 10.1 % underwent degradation, while those with slopes greater than 10.1 % aggraded. Figure 3 shows influences of both the grain size ratio ( $D_c/D_f$ ) and the proportion of fines within the feed upon the two-size equilibrium bed slope. Particular note should be made of the transition in behaviour between fine diameter  $\leq 2$  mm and  $\geq 3$  mm. For comparison, degradation of the bed can be seen in Figure 2b ( $D_f = 0.7$  mm), and aggradation in Figure 2c ( $D_f = 3$  mm). Near agreement can be seen between the two-size equilibrium bed slopes when the fine size composes 22 % and 41% of the total feed for all of the grain size ratios tested. However, when the fine size composes 68 % of the total feed, the two-size equilibrium slope is consistently substantially higher, with the channel response passing from degradation to aggradation in certain cases.



**Figure 3: Manual measurements of the two-size equilibrium bed slope for the experiments in Set 1\*.**

**\*For each experiment the slope was measured five times. The initial bed slope (10.1 %) before the fine sediment introduction is shown using a dashed black line. The 22 % data have been offset by 0.1 mm (decreased) on the x-axis to permit comparison of the data. Error bars show absolute range of measurements.**

The evolution of the bed slope over the duration of each experiment is shown in Figure 4, illustrating how the bed slope transitions from one-size equilibrium to two-size equilibrium due to the addition of fines to the bed. To ensure that the full bed slope evolution was captured, the fine sediment was introduced two minutes after the start of the recording. Consequently, the first two minutes of the slope evolution curves in Figure 4 are of the bed in one-size equilibrium. Following the fine sediment introduction, an initial small increase in the bed slope before an eventual fall can be seen in several degrading experiments (see annotation in Figure 4a). This initial upwards peak in the bed slope is due to the wave of fine sediment which moves downstream on the bed surface following the fine sediment introduction, causing the bed to temporarily thicken before it starts to degrade.

Fluctuations in the bed slope can be observed in some of the experiments presented in Figure 4. These fluctuations are particularly prominent and regular in the experiments utilising 1.5 mm fines with a content of 22 % and 41 %. Possible causes of these fluctuations are addressed later in the paper.

For each of the experiments presented in Figure 4, exponential decay curves were fitted to the data:

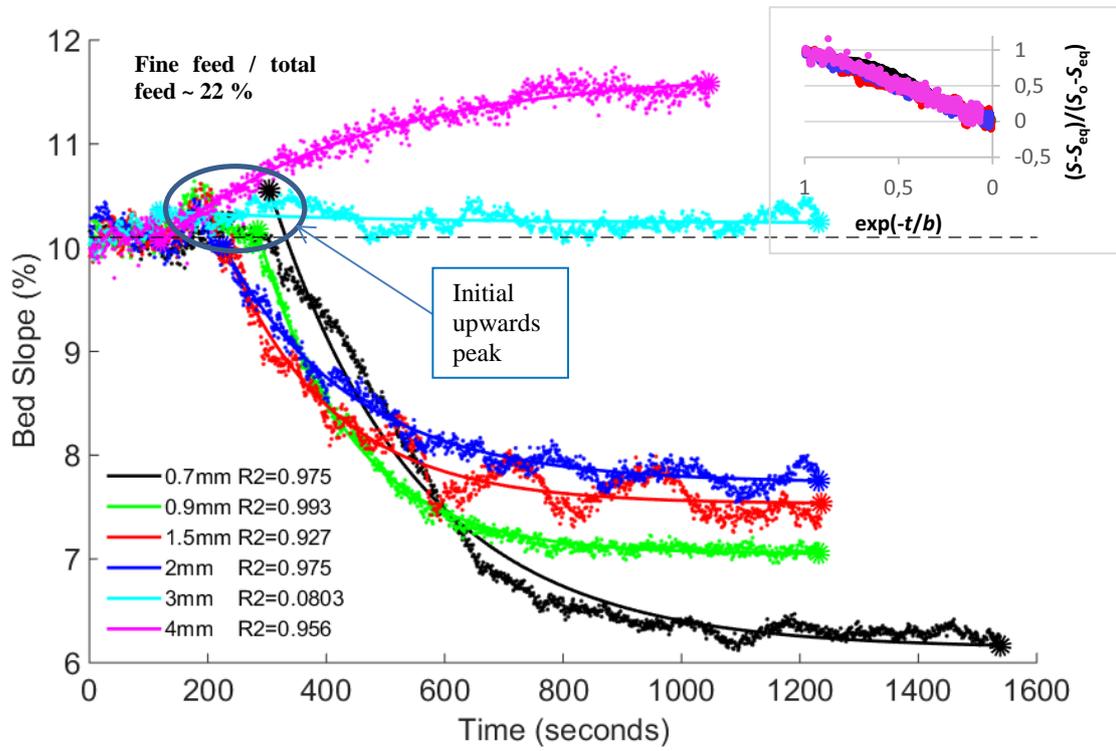
$$S(t) = S_{eq} + (S_o - S_{eq}) \cdot \exp(-t/b)$$

wherein  $S$  is the bed slope,  $S_{eq}$  is the two-size equilibrium bed slope,  $S_o$  is the initial slope (10.1 %),  $b$  is the decay timescale, and  $t$  is time from the commencement of the adjustment ( $t_o$  defined below). The values of  $(S_o - S_{eq})$ ,  $b$  and  $S_{eq}$  are presented in Table 3. The exponential decay curve was fitted between the final data point for the experiment (when the bed is in two-size equilibrium) and a pre-determined start time,  $t_o$  (presented in Table 3). The start time for the aggradation cases is the time when the fine sediment is introduced. Due to the initial upward peak in the bed slope following the fine grain input for the degradation cases, the start time is when the bed slope passes through 10.1 % again after this initial upward peak. The start time for the degradation cases varied as, depending upon the fine grain size and the fine feed proportion, it required different lengths of time for the fine sediment to reach the camera visualisation window from the feeder. The  $R^2$  value for each of the curves is presented in the legend of the figure. With the exception of three experiments, for which there is little difference between  $S_o$  and  $S_{eq}$ , the  $R^2$  values are greater than 0.9.

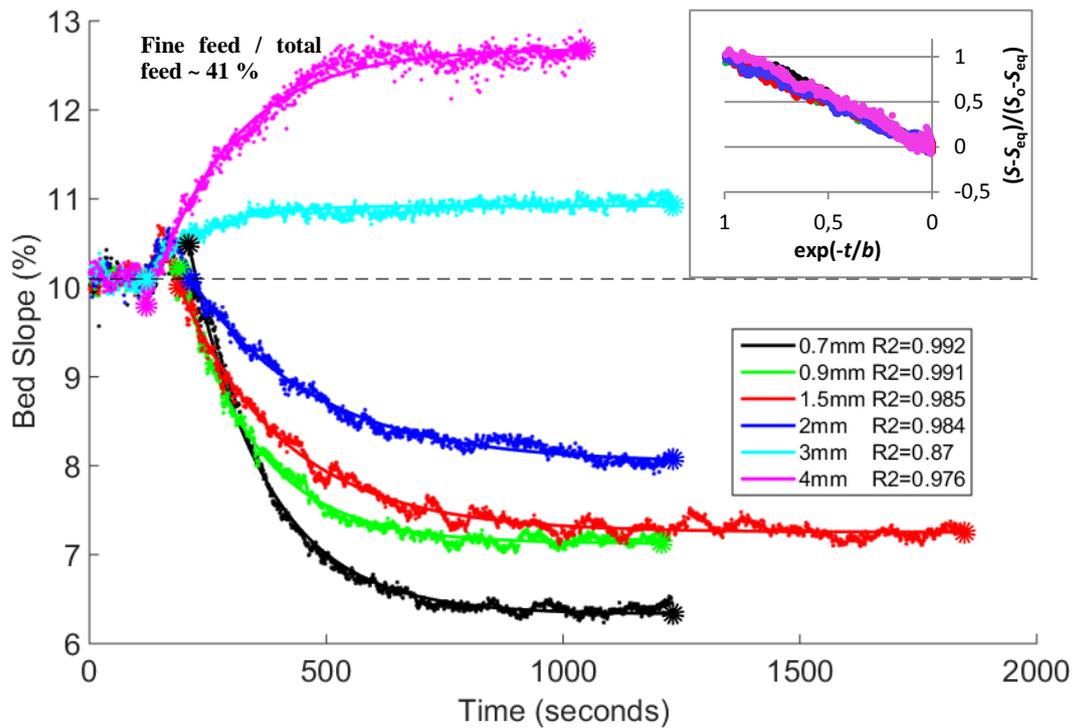
Insets in Figure 4 show the collapse of the slope evolution curves. The data used for the collapse in each case lie between the initial and final data points used for the exponential fitting. Cases for which little slope adjustment occurred (the  $R^2$  values are less than 0.9), have been discounted from the collapse plots as the fluctuations obscure the minor net change.

The sediment output was measured over the duration of the experiments. Figure 5 shows the difference between the cumulative input and the cumulative output for the 5 mm grains in each of the experiments, and Figure 6 does the same for the fine sediment. The difference between the cumulative sediment input to the channel and the cumulative sediment output over a period of time measures either how much additional sediment has been stored in the channel (if input > output), or how much has been extracted from the bed (if output > input). Figure 5, similar to Figure 4, shows an influence of both the grain size ratio and the proportion of fines within the feed upon the channel response to the fine grain input; under certain conditions, 5 mm beads are stored within the channel and under other conditions 5 mm beads are extracted from the bed. Figure 6 shows fines being retained in the channel in every experiment.

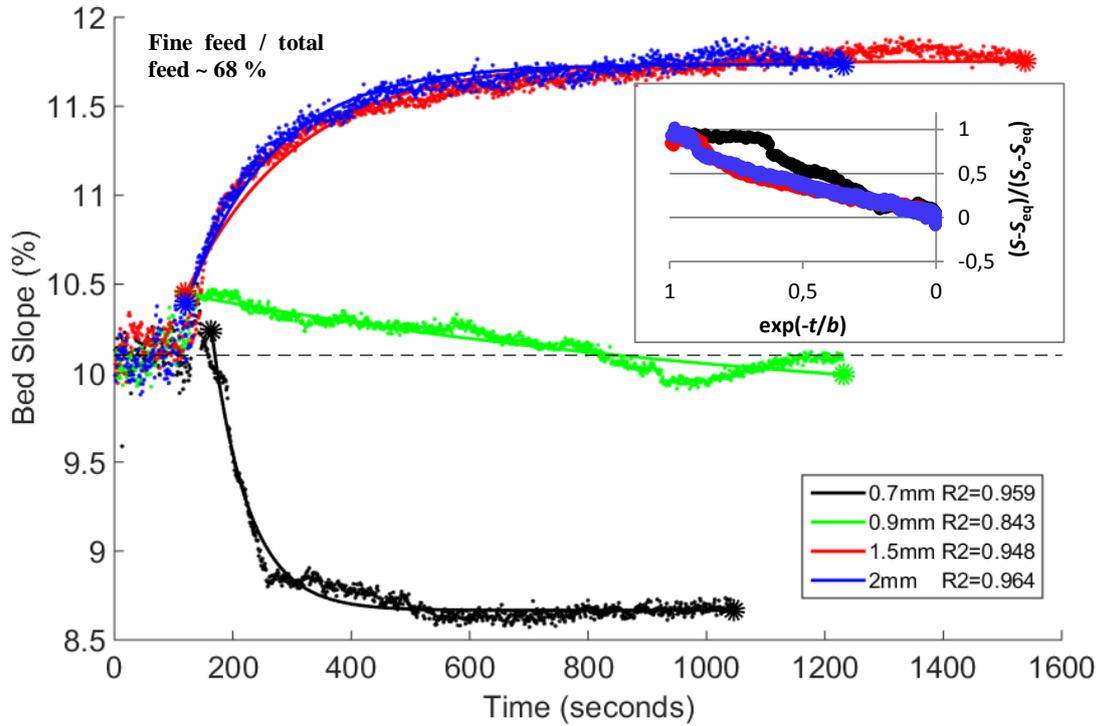
A summary of the relevant hydraulic parameters when the bed reached two-size equilibrium ( $S_{eq}$ ) is given in Table 4.



**Figure 4a**



**Figure 4b**

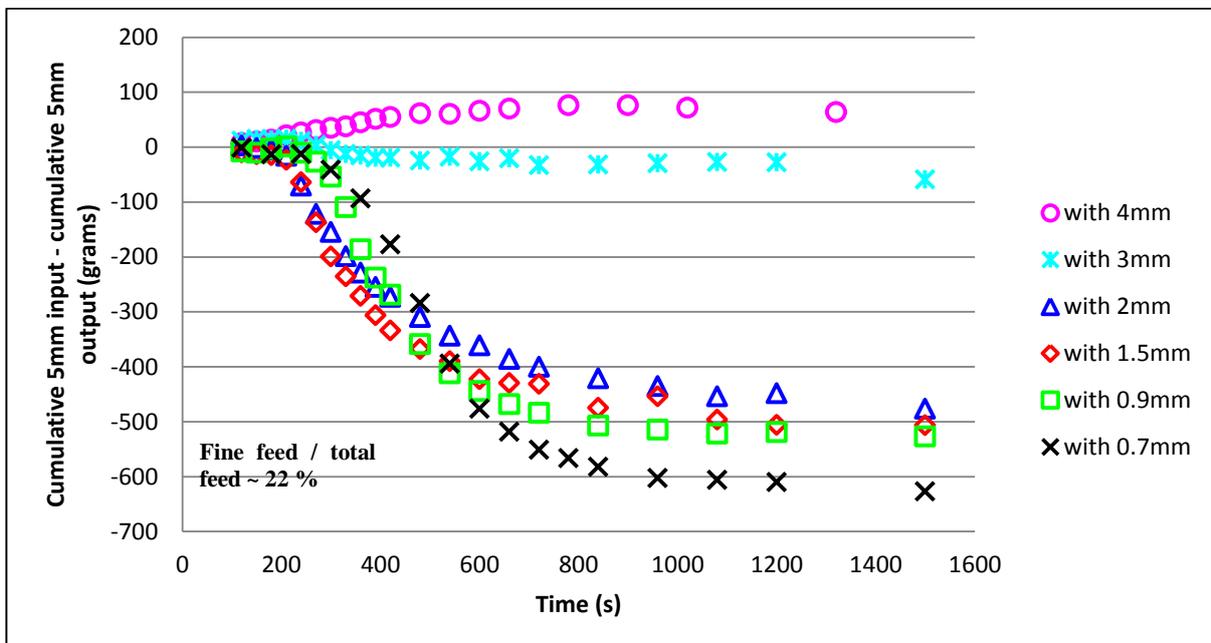


**Figure 4c**

**Figure 4: Main figures: Bed slope evolution over time. Figure insets: Collapse of the bed slope evolution profiles. The initial bed slope (10.1 %) before the fine sediment was introduced is shown using a dashed black line. The fine sediment was introduced to the channel after 120 seconds. An exponential decay curve (start and end points indicated with spots) has been fitted to each of the bed slope evolution curves, with the  $R^2$  value included in the legend, and the parameter values shown in Table 3.**

**Table 3: The data for the exponential decay curves shown in Figure 4.**

Approximate fine feed/total feed	Start time for curve fitting, $t_0$ (s)	Fine sediment size (mm)	$R^2$ value	$S_o - S_{eq}$ (%)	$b$	$S_{eq}$ (%)
22 %	300	0.7	0.98	4.4	250	6.1
	280	0.9	0.99	3.1	160	7.0
	230	1.5	0.93	2.4	180	7.5
	230	2	0.98	2.3	220	7.7
	120	3	0.080	0.11	360	10.2
	120	4	0.96	-1.6	340	11.7
41 %	210	0.7	0.99	4.2	160	6.3
	190	0.9	0.99	3.1	160	7.1
	180	1.5	0.99	2.8	220	7.3
	220	2	0.98	2.1	260	8.0
	120	3	0.87	-0.83	110	10.9
	120	4	0.98	-2.9	180	12.7
68 %	160	0.7	0.96	1.6	62	8.7
	120	0.9	0.84	0.73	1100	9.7
	120	1.5	0.95	-1.3	190	11.8
	120	2	0.96	-1.4	150	11.7



**Figure 5a**

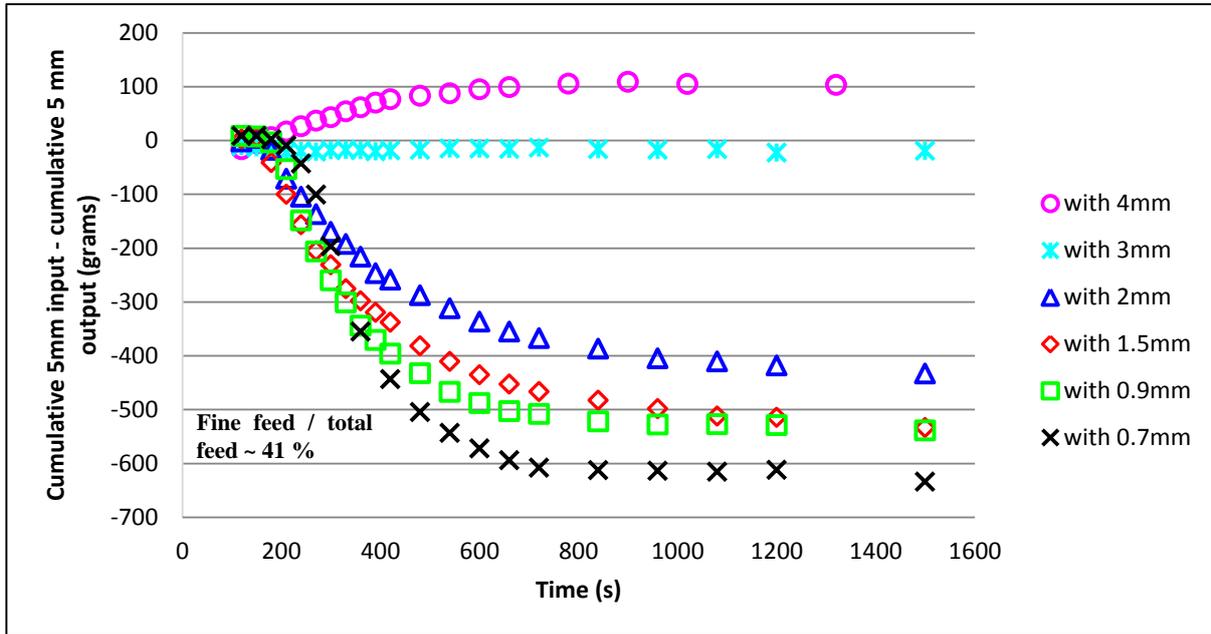


Figure 5b

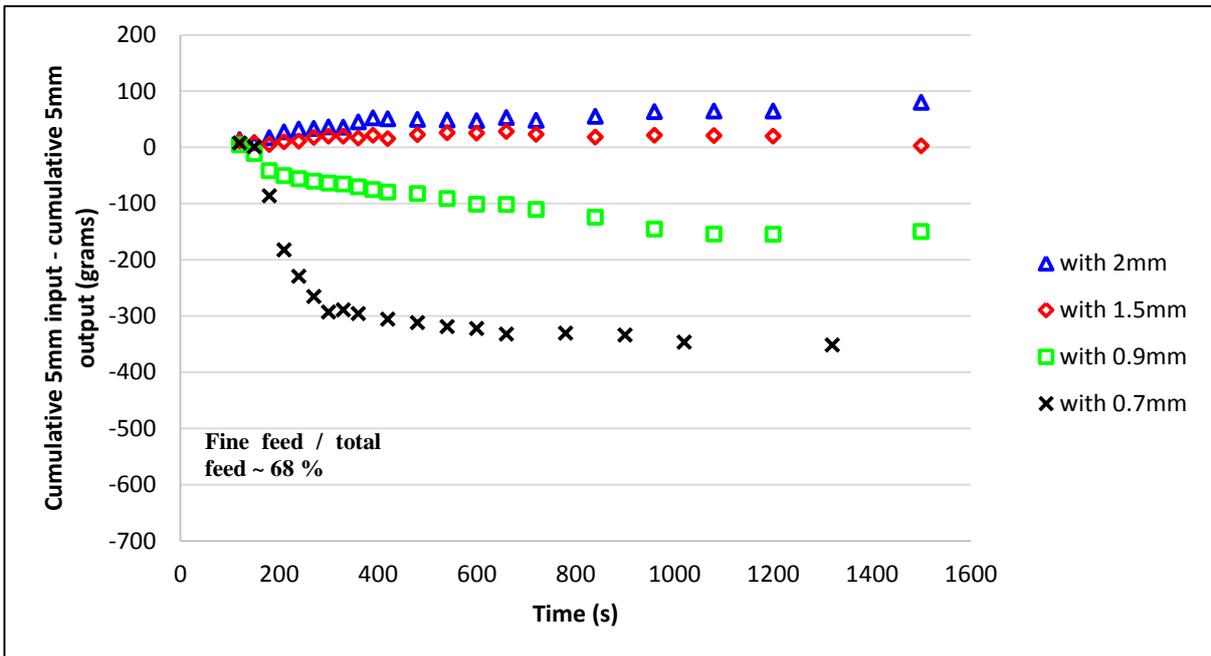


Figure 5c

Figure 5: Cumulative 5 mm input minus cumulative 5 mm output over time. Fine feed/total feed is: 22 % (Figure 5a), 41 % (Figure 5b) and 68 % (Figure 5c).

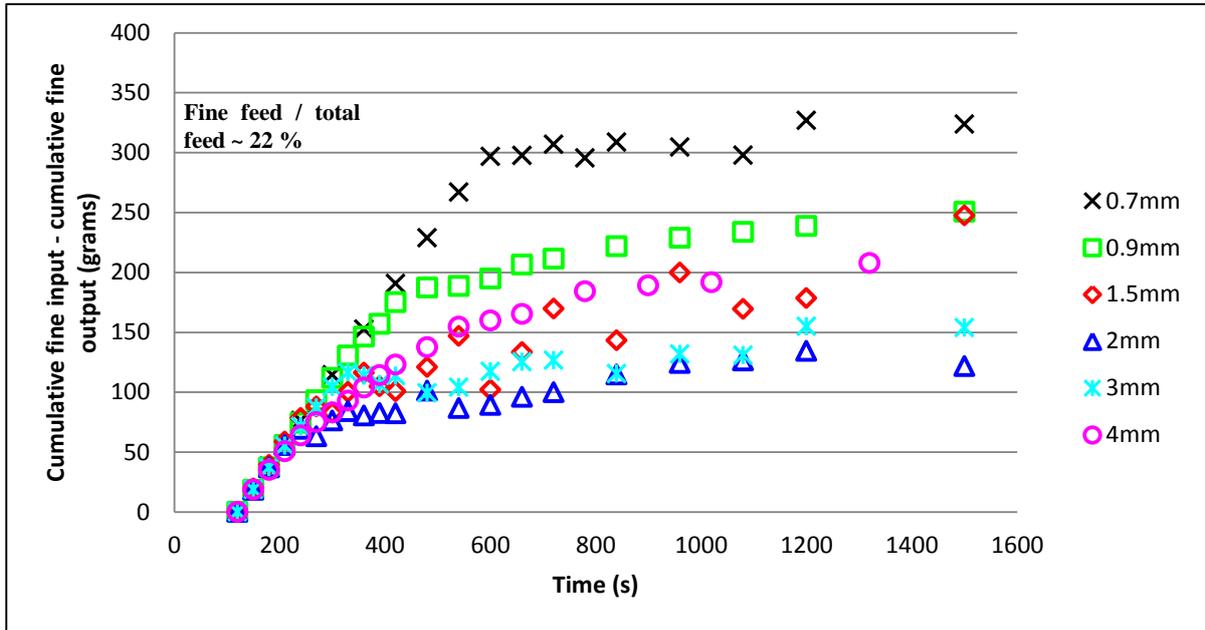


Figure 6a

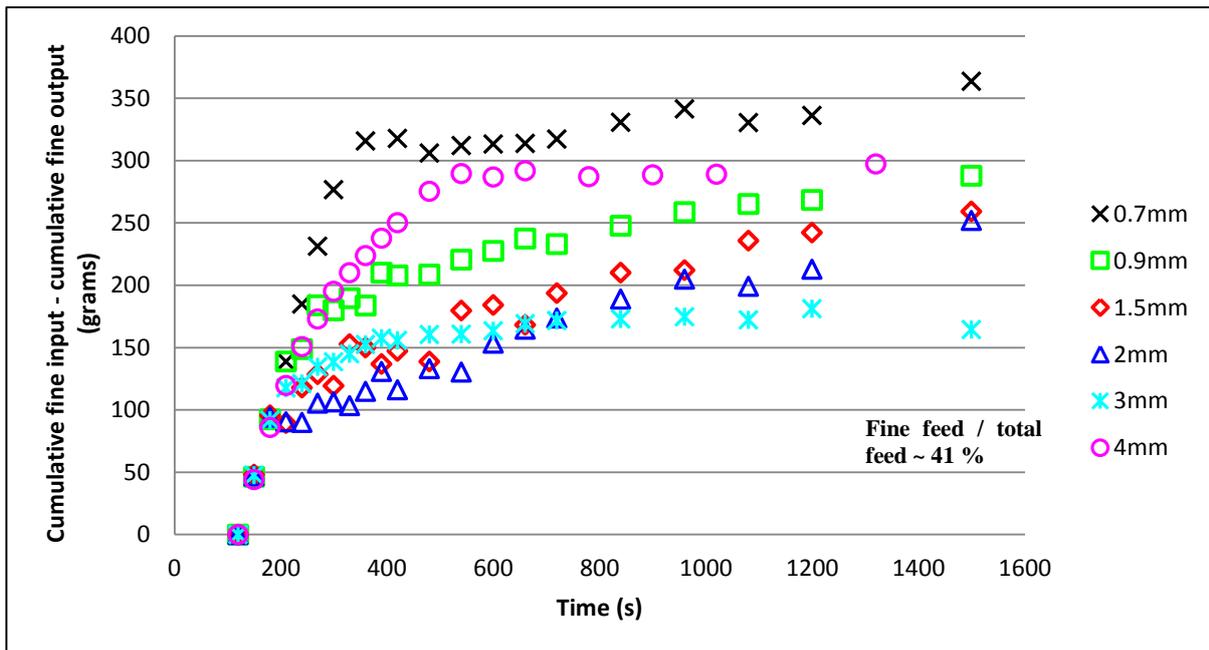
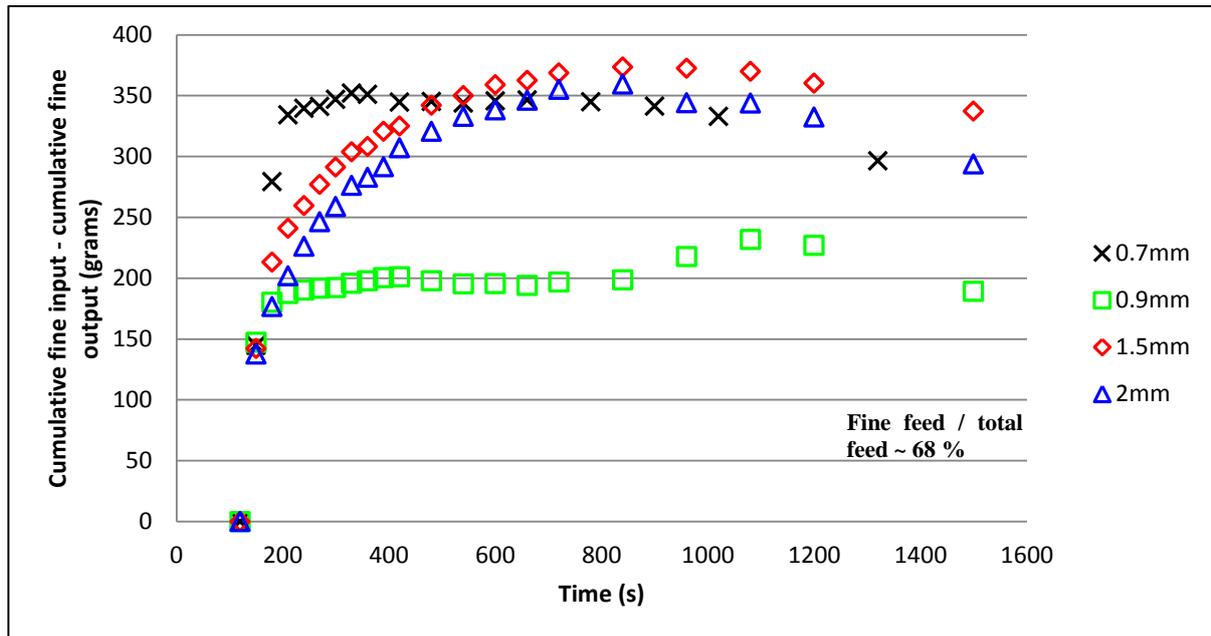


Figure 6b



**Figure 6c**

**Figure 6: Cumulative fine input minus cumulative fine output over time. Fine feed/total feed is: 22 % (Figure 6a), 41 % (Figure 6b) and 68 % (Figure 6c).**

**Table 4: Hydraulic parameters when the bed is in two-size equilibrium.**

Experiment Number	Mean flow depth, h (mm)	Mean velocity, U (m/s)	Froude number $r \left( \frac{U}{\sqrt{gh}} \right)$	Reynolds Number <sup>1</sup> $\left( \frac{4UR_h}{\nu} \right)$	Side-wall corrected dimensionless shear stress – with $D_c = 5 \text{ mm}^2$	Side-wall corrected dimensionless shear stress – with $D_f^2$
1	11.0	0.44	1.33	6868	0.064	0.459
2	10.3	0.47	1.48	7195	0.06	0.428
3	6.7	0.72	2.80	9367	0.043	0.309
4	11.3	0.42	1.27	6726	0.080	0.446
5	10.2	0.47	1.49	7222	0.070	0.387
6	6.0	0.80	3.29	9938	0.041	0.227
7	10.3	0.47	1.47	7184	0.076	0.252
8	9.6	0.50	1.64	7530	0.065	0.218
9	6.1	0.79	3.22	9861	0.058	0.192
10	9.8	0.49	1.59	7439	0.072	0.181
11	9.1	0.53	1.77	7783	0.068	0.170
12	6.1	0.78	3.20	9839	0.057	0.144
13	9.1	0.53	1.76	7766	0.094	0.156
14	8.0	0.60	2.15	8445	0.083	0.138
15	7.8	0.62	2.23	8567	0.088	0.110
16	7.7	0.63	2.29	8664	0.095	0.118

<sup>1</sup>The value used for the kinematic viscosity assumes a water temperature of 25C.

<sup>2</sup>The  $S_{eq}$  values in Table 3 were used to calculate the side-wall corrected dimensionless shear stresses.

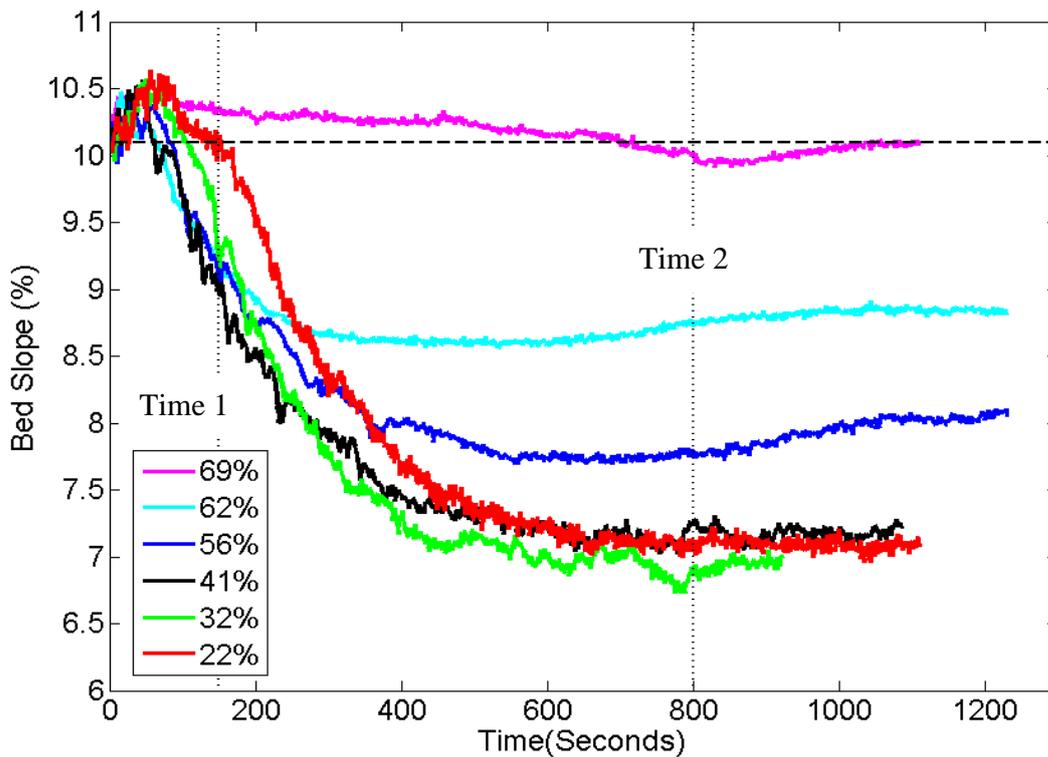
### 3.2 Set 2

During the Set 2 experiments, the fine sediment was introduced as soon as the camera recording started. Figure 7a shows the bed slope evolution for the 0.9 mm experiments, and Figure 7b for the 1.5 mm experiments. In the 0.9 mm input experiments, the bed degraded in all cases, except for the 68 % fine content experiment, when the two-size equilibrium bed slope was very similar to the one-size equilibrium bed slope. The 1.5 mm input experiments in which the fines composed approximately 63 and 68 % of the total feed aggraded following the fine grain input whereas, in the other experiments, the bed degraded in response to the input.

In Figure 7, ‘time 1’ is when the bed slope is still responding to the fine grain input, and at ‘time 2’ the bed appears to be close to, or in, two-size equilibrium in all of the experiments. However, as can be seen in Figure 7b, in the 23 %, 31 % and 42 % 1.5 mm experiments, the bed slope oscillated whilst in two-size equilibrium. The amplitude of the oscillations appears to grow as the slope declines. During these experiments it could be seen that the fine grain layer upon the bed surface oscillated in thickness, causing these bed slope oscillations.

The number of the 5 mm bedload grains present within the camera visualisation window at ‘time 1’ and ‘time 2’ for both the 0.9 mm and 1.5 mm experiments are presented in Table 5. For both ‘time 1’ and ‘time 2’, a decrease in the number of the 5 mm bedload grains can be seen as the proportion of fines in the total feed increases. Additionally, the number of bedload grains is lower at ‘time 2’ compared to ‘time 1’ for the experiments with fine feed contents less than or equal to 42 %. In general, the number density is higher in the experiments with the 0.9 mm fines compared to those with the 1.5 mm fines. Further, the range in number density over the measurements is larger for the 1.5 mm experiments, potentially due to the oscillating fine layer thickness.

Over the range of fine sediment feed rates explored, the 1.5 mm experiments transitioned from degradation to aggradation, therefore analysis of the quasi-static fine layer thickness was undertaken for this case. Figure 8 shows the thickness of the quasi-static fine layer at ‘time 1’ and ‘time 2’ for the 1.5 mm experiments. In all of the experiments, the quasi-static fine layer was thicker at ‘time 2’ than at ‘time 1’. At ‘time 1’ there is a positive relation between the quasi-static fine layer thickness and the proportion of fines in the total feed. For all of the experiments, excluding the 68 % fine content, the thickness of the quasi-static fine layer at ‘time 2’ is similar. The thickness of the quasi-static layer at ‘time 2’ in the experiment with a 68 % fine content is higher. Figures 9a and 9b illustrate the difference in quasi-static fine layer thickness between two contrasting cases: Experiment 8 at ‘time 1’ and Experiment 9 at ‘time 2’.



**Figure 7a**

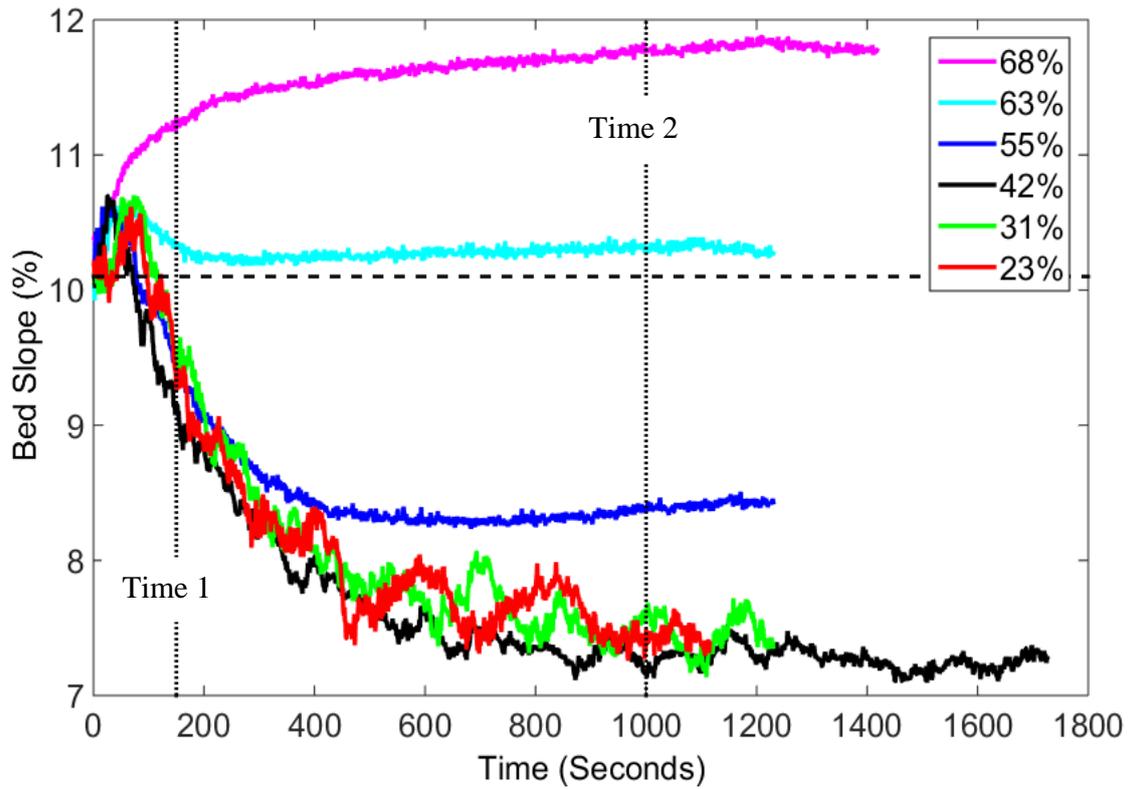


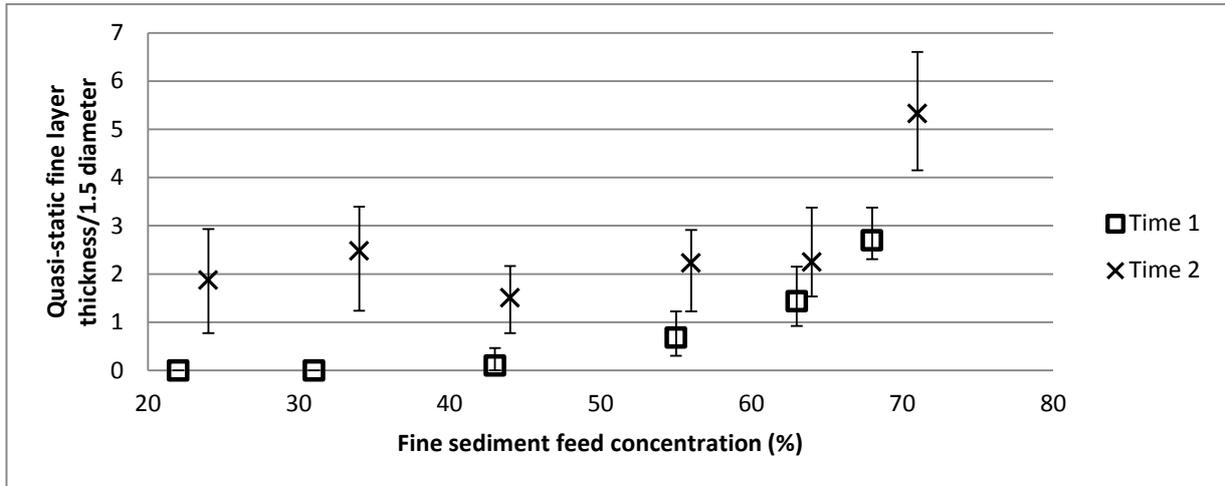
Figure 7b

Figure 7: Bed slope evolution over time for the experiments with the 0.9 mm fines (Figure 7a) and 1.5 mm fines (Figure 7b). The initial bed slope (10.1 %) before the fine sediment was introduced is shown using a dashed black line. The fine sediment was introduced to the channel at 0 seconds. ‘Time 1’ and ‘time 2’ are explained in the text.

Table 5: Number of 5 mm bedload particles within the camera visualisation window at ‘time 1’ and ‘time 2’\*.

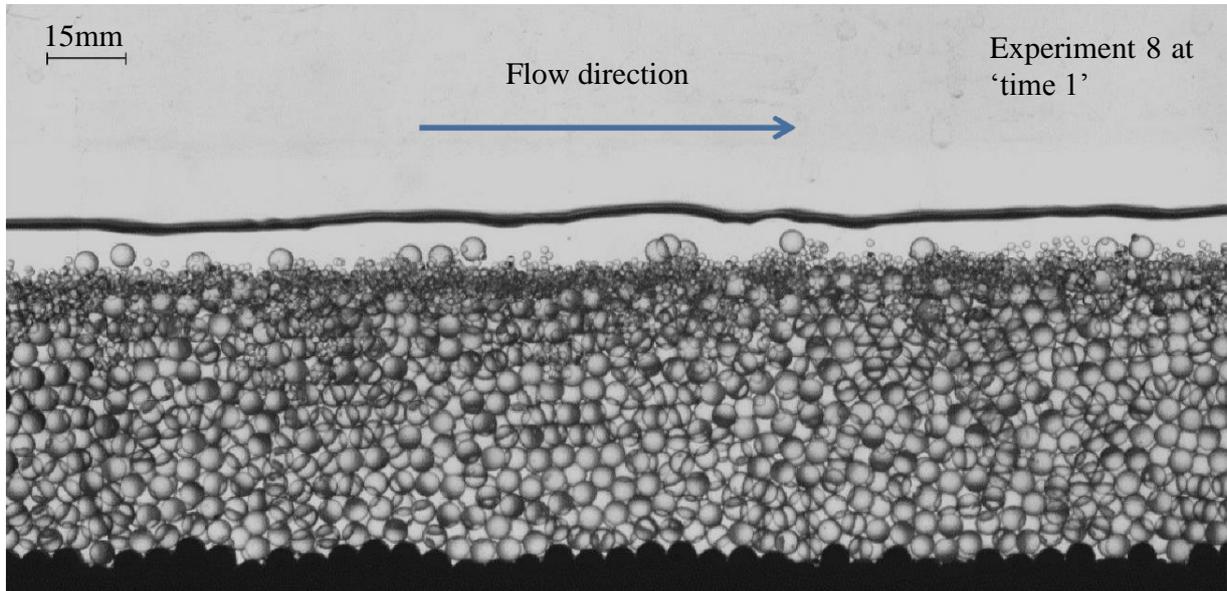
	Time 1		Time 2	
	0.9 mm	1.5 mm	0.9 mm	1.5 mm
23 %	59 ± 25	57 ± 33	57 ± 13	41 ± 18
31 %	50 ± 18	44 ± 34	41 ± 16	26 ± 18
42 %	35 ± 27	26 ± 29	30 ± 10	20 ± 16
55 %	16 ± 22	14 ± 6	12 ± 8	11 ± 7
63 %	10 ± 5	7 ± 6	8 ± 5	9 ± 6
68 %	8 ± 5	8 ± 5	8 ± 4	9 ± 6

\*Measurements were taken over two minutes, with the number of bedload grains within the camera visualisation window counted every 5 seconds starting at ‘time 1’ and ‘time 2’. The errors show the range in number density over the twenty four measurements.

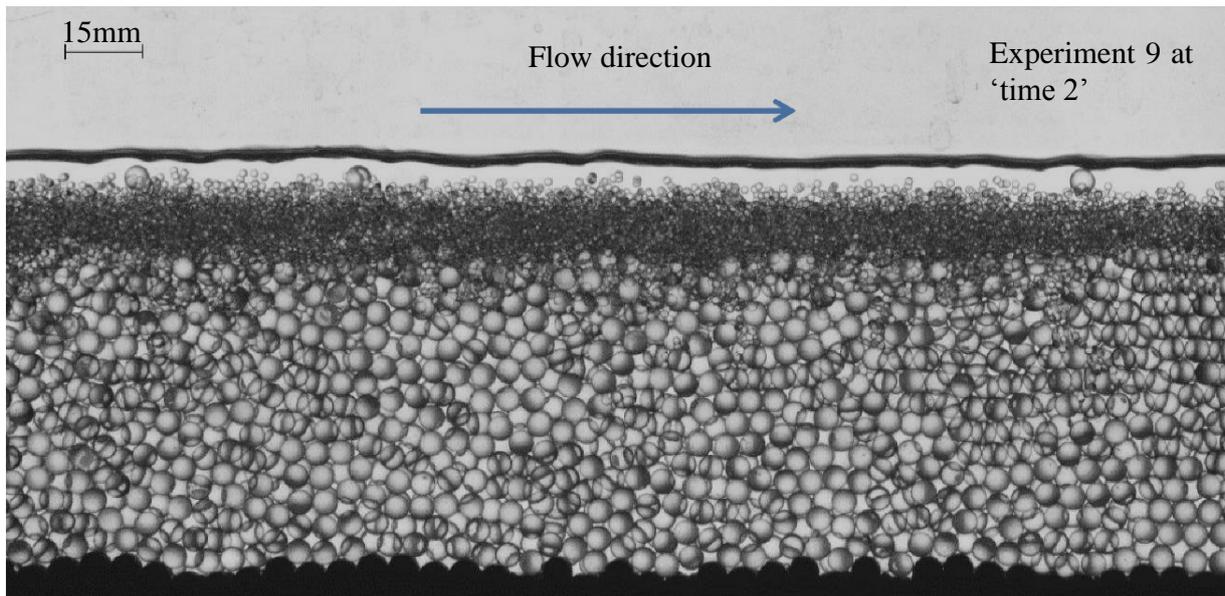


**Figure 8:** Quasi-static fine layer thickness for the 1.5 mm experiments at ‘time 1’ and ‘time 2’\*.

\*Measurements were taken over 10 seconds, every second starting at ‘time 1’, the error bars show the range in quasi-static fine layer thickness over these ten measurements. The ‘time 2’ data have been offset by 1 % on the x-axis to permit comparison of the data. ‘Time 1’ and ‘time 2’ are explained in the text.



**Figure 9a**



**Figure 9b**

**Figure 9:** Example frame from the camera visualisation window of the bed: at ‘time 1’ in Experiment 8 (42 % fine feed content) (Figure 9a), and at ‘time 2’ in Experiment 9 (68 % fine feed content) (Figure 9b).

#### 4. Discussion

Since *Gilbert's* [1914] work, over 100 years ago, no further work, involving a wide range of grain size ratios in the context of fine grain inputs, has been undertaken. The results of the present experiments confirm *Gilbert's* [1914] observations of a relation between the grain size ratio and the capacity transported. Further, this work demonstrates an inverse relation between two-size equilibrium bed slope values and grain size ratios when the proportion of fines in the total feed is constant. This explains why *Curran and Wilcock* [2005] observed a reduction in the equilibrium slope over a wider range of sand contents than *Iseya and Ikeda* [1987]: the *Curran and Wilcock* [2005] experiments used a larger grain size ratio.

This inverse relation may be attributed to a number of factors; first, to the negative relation between grain size ratio and pivoting angle. According to *Komar and Li* [1986], the pivoting angle required for entrainment decreases as the difference between the diameter of a grain and the diameter of the underlying grain around which it needs to pivot increases, potentially explaining why experiments with larger grain size ratios degraded farther. Additionally, the finer the grains introduced, the smoother the bed surface. A smoother bed surface due to a finer gravel pulse was shown by *Venditti et al.* [2010b] to cause flow acceleration close to the bed and therefore to increase drag upon the coarse particles.

The bed slope response profiles following the fine grain inputs are found to be well-described by exponential decay curves. The coefficients of these curves are reported in Table 3, including the parameter  $b$ , which is the decay timescale. Using these coefficients, the data collapse,

showing that there is a timescale and unique exponential behaviour to pass from one equilibrium state to another following a fine grain input. The timescale is dependent upon the grain size ratio. Discounting the 0.7 mm results, a positive relation is present between the fine grain diameter and the decay timescale for the degradational cases. The results for the 0.7 mm experiments do not conform to this behaviour. This may be a result of the exponential curve not being a good fit to the data in the 22 % fine content case (see Figure 4). Despite this, excluding the 68 % fine content case, the 0.7 mm data collapse in line with the other experiments (Figure 4 insets), suggesting that the behaviour controlling the channel response is similar.

Presentation of the 5 mm sediment output over the duration of the experiments is given in Figure 5. The data are organized with respect to the fine grain input diameter. This pattern is a result of the bed slope response to the fine grain input. The larger the grain size ratio, the greater the amount of degradation (see Figure 4), hence the larger the amount of 5 mm material evacuated from the flume. As the grain size ratio decreases, the bed response changes from degradation to aggradation. In the experiments with aggradation, additional 5 mm material was often retained in the flume.

Figure 6 presents the sediment output for the fine material. The fine material output is more complex than the 5 mm behaviour and the plots, considering all of the data, are not organized according to the fine grain input diameter. This outcome is due to competing factors. In the experiments with the 0.7 mm, 0.9 mm, 1.5 mm and 2 mm beads, the fine material was able to infiltrate into the coarse bed, and was therefore retained in the channel. Additionally, the finer the input diameter, the more infiltration took place, and therefore the greater the amount of fine material retained in the channel. In the 22 % (Figure 6a) and 41 % (Figure 6b) fine feed content experiments, the 0.7 to 2 mm fine inputs all resulted in degradation of the bed. It is seen in Figures 6a and 6b, when only the degradation experiments are considered, that the data are arranged according to the fine grain diameter; with the most material retained for the smallest fine grain diameter. In the 3 mm and 4 mm experiments, the fines were not able to infiltrate into the bed, but the channel aggraded in response to the fine grain inputs due to a deposited layer of fines upon the bed surface, therefore causing the data to overlap with the 0.7 to 2 mm cases. Considering just the 3 mm and 4 mm cases, in Figures 6a and 6b, it can be seen that the greater the fine grain diameter, the larger the amount of fines retained in the channel, so that the bed aggraded more at lower grain size ratios.

In the 68 % fine feed content experiments, the fines were able to infiltrate into the bed in all investigated cases, again with more fines infiltrating at smaller fine grain diameters. However, in the experiments with the 1.5 mm and 2 mm inputs, the bed responded by aggrading, with a layer of deposited fines upon the bed surface, therefore increasing the amount of fines retained. Consequently, the 1.5 mm and 2 mm data in Figure 6c plot higher than expected.

The second set of experiments was designed to gain further insight into the influence of fine feed content at a given grain size ratio. Following a fine grain input to a coarse bed in motion, the two-size equilibrium bed slope has previously been found to depend upon the proportion of fines in the total feed [Jackson and Beschta, 1984; Iseya and Ikeda, 1987; Ikeda and Iseya, 1988; Wilcock *et al.*, 2001; Curran and Wilcock, 2005b]. This was confirmed during the present experiments. Figure 7 shows that there is a positive relation between the fine feed content and the two-size equilibrium bed slope. In the 1.5 mm experiments there is a transition from degradation to aggradation between 55 % and 63 % fine feed content.

In order to understand why the 63 % and 68 % cases lead to aggradation, but the other cases lead to degradation, it is important to examine the quasi-static fine layer thickness at ‘time 1’,

presented in Figure 8. For both the 63 % and the 68 % cases at ‘time 1’, there is a quasi-static fine layer at least one grain diameter thick covering the underlying quasi-static coarse bed surface, preventing interaction between the bedload transport layer and the underlying bed, therefore inhibiting degradation. This is the first revelation, to the authors’ knowledge, why the equilibrium bed slope changes at a given fine feed content. At ‘time 2’, when the bed has reached two-size equilibrium, it can be seen in Figure 8 that the quasi-static fine layer in all of the experiments is thicker than at ‘time 1’, and always greater than one fine grain diameter in thickness. Further degradation is not possible under these conditions as the thick quasi-static fine layer isolates the underlying bed.

This thickening of the layer at ‘time 2’ compared to ‘time 1’ has a substantial impact upon the number of mobile coarse grains (see Table 5) in the experiments with a quasi-static fine layer thickness of approximately zero at ‘time 1’. In these experiments, the number density is reduced at ‘time 2’, compared to ‘time 1’ as the thicker quasi-static fine layer results in a smoother bed. Thickening of the quasi-static fine layer from ‘time 1’ to ‘time 2’ did not impact the number density of bedload grains in the experiments wherein the quasi-static fine layer thickness at ‘time 1’ was greater than zero.

*Iseya and Ikeda* [1987] and *Ikeda and Iseya* [1988] identified ‘congested’ and ‘smooth’ bed states occurring as a function of the proportion of fines present within a mixture. When ‘congested’, the bed surface is densely packed with coarse particles moving slowly; this occurs when there is a lack of fines on the surface of the bed. An abundance of fines, however, leads to a ‘smooth’ bed state, which is sparsely populated with coarse particles moving at higher velocities. *Ikeda and Iseya* [1988] found that the transition between the ‘congested’ and ‘smooth’ bed states occurred around a 50:50 fine to coarse ratio within the feed. . A similar transition can be seen in Table 5 during the present experiments, with a much higher number density occurring when the fine content is less than 50 %, and vice versa. Although the number density appears not to reach a stable value until the fine content is greater than or equal to 63 %. These experiments indicate a link between the quasi-static fine layer thickness and bedload number density, hence the character of the transport mechanics for the larger grains.

## 5. Conclusions and Future Work

Experiments exploring the superior mobility of sediment mixtures demonstrate that the size ratio ( $D_c/D_f$ ), in addition to the fine feed content, has an important influence upon sediment transport rates following a fine grain input to a coarse mobile bed.

The first set of experiments explored the role of grain size ratio. The transition from the one-size equilibrium bed slope, to the two-size equilibrium bed slope depends upon the fine grain diameter, a previously undocumented finding. For a given fine sediment feed rate, the largest amount of degradation occurs at the greatest grain size ratio. Progressively less degradation occurs as the grain size ratio decreases, until a transition from degradation to aggradation occurs. Despite the influence of the size ratio upon the channel response, exponential collapse of the bed slope evolution curves is feasible. The slope collapse reveals similarity in the characteristics determining the channel response, irrespective of the grain size ratio. Further, the evolution of the slope over time is controlled by the initial and final equilibrium bed slopes. If these values are known, it would be possible to predict slope evolution over time for any size of fine grain input. This finding is potentially important for river engineering and fluvial geomorphology.

The second set of experiments explored the role of fine content in the total feed. The transition from one-size equilibrium bed slope to two-size equilibrium bed slope depends upon the fine feed content. The greatest amount of degradation occurs at the smallest fine feed content. As the fine feed content increases, the amount of degradation is reduced, until there is a transition from degradation to aggradation. An increase in the sediment transport rate for the coarse component, which results in bed degradation, is feasible only when the bedload layer is able to interact with the underlying coarse bed. The fine content at which the bed response transitions from degradation to aggradation depends upon the infiltration characteristics, hence the grain size ratio.

These experiments, using spherical glass beads, have enabled investigation of the phenomena at the grain scale, permitting important observations of the dominant processes. Thanks to the reductionist approach taken, the experiments clearly demonstrate the importance of the finer sediment upon mobility and sediment transport rates. In order to develop our understanding towards natural systems, a parallel set of the experiments need be undertaken using the same experimental procedure, but with natural materials. These experiments will introduce the element of particle shape, but we hypothesise that the processes revealed in this paper will still control the channel response. The successive outcomes will move us steadily closer to addressing the ultimate question of what controls channel response in more or less widely graded natural sediments.

**Acknowledgements.** This research was partly funded by a Discovery grant to M. Church from the Natural Sciences and Engineering Research Council of Canada (NSERC). The work was also partly funded by the French “Institut National des Sciences de l’Univers” programme EC2CO-BIOHEFECT and DRIL, and ANR (the French national research agency) project SegSed. A. Dudill, H. Lafaye de Micheaux and P. Frey acknowledge the support of Irstea (formerly Cemagref). A. Dudill and H. Lafaye de Micheaux also acknowledge funding provided by grants from la Région Rhône-Alpes. This work would not have been possible without the technical support of F. Ousset in the laboratory at Irstea.

## References

- Beschta, R. L., and W. L. Jackson (1979), The intrusion of fine sediments into a stable gravel bed, *Journal of the Fisheries Research Board of Canada*, 36, 204–210.
- Bridgwater, J., N. W. Sharpe, and D. C. Stocker (1969), Particle mixing by percolation, *Transactions of the Institute of Chemical Engineers*, 47, T114-T119.
- Cui, Y., G. Parker, T. E. Lisle, J. Gott, M. E. Hansler-Ball, J. E. Pizzuto, N. E. Allmendinger, and J. M. Reed (2003), Sediment pulses in mountain rivers: 1. Experiments, *Water Resources Research*, 39, 1239.
- Curran J. C., and P. R. Wilcock (2005), Effect of sand supply on transport rates in a gravel-bed channel, *Journal of Hydraulic Engineering*, 131, 961–967.
- Diplas P., and G. Parker (1992), Deposition and removal of fines in gravel-bed streams, in *Dynamics of Gravel-Bed Rivers*, edited by P. Billi, R. D. Hey, C. R. Thorne, and P. Tacconi, pp. 313–329, John Wiley & Sons Ltd, New York.
- Dudill A., P. Frey, and M. Church (in review), Infiltration of fine sediment into a coarse mobile bed: a phenomenological study.
- Fenton JD., and J. E. Abbott (1977), Initial movement of grains on a stream bed: The effect of relative protrusion, *Proceedings of the Royal Society Series A*, 352, 523-537.

- Ferguson R. I., K. L. Prestegard, and P. J. Ashworth (1989), Influence of sand on hydraulics and gravel transport in a braided gravel bed river, *Water Resources Research*, 25, 635–643.
- Frostick L. E., P. M. Lucas, and I. Reid (1984), The infiltration of fine matrices into coarse-grained alluvial sediments and its implications for stratigraphical interpretation, *Journal of the Geological Society London*, 141, 955–965.
- Gibson S., D. Abraham, R. Heath, and D. Schoellhamer (2009), Vertical gradational variability of fines deposited in a gravel framework, *Sedimentology*, 56, 661–676.
- Gilbert G. K. (1914), The transportation of debris by running water, *Professional paper 86*, US Geological Survey, Washington DC.
- Houssais M., and E. Lajeunesse (2012), Bedload transport of a bimodal sediment bed, *Journal of Geophysical Research*, 117, F04015.
- Ikeda H., and F. Iseya (1988), Experimental study of heterogeneous sediment transport, *Environmental Research Center Paper No. 12, Univ. of Tsukuba, Tsukuba, Japan*.
- Iseya F., and H. Ikeda (1987), Pulsations in bedload transport rates induced by longitudinal sediment sorting: a flume study using sand and gravel mixtures, *Geografiska Annaler, Series A*, 69, 15–27.
- Jackson W. L., and R. L. Beschta (1984), Influences of increased sand delivery on the morphology of sand and gravel channels, *Water Resources Bulletin American Water Resources Association*, 20, 527-533.
- Komar P. D., and Z. Li (1986), Pivoting analyses of the selective entrainment of sediments by shape and size with application to gravel threshold, *Sedimentology*, 33, 425-436.
- Lafaye de Micheaux H., A. Dudill, P. Frey, and C. Ducottet (2015), Image processing to study the evolution of channel slope and water depth in bimodal sediment mixtures, in *10<sup>th</sup> Pacific Symposium on Flow Visualisation and Image Processing*, edited by G. Cardone.
- Lane E. W. (1955), The importance of fluvial morphology in hydraulic engineering. *Proceedings of the American Society of Civil Engineers*, 81 (745), 17.
- Lisle T. E. (1989), Sediment transport and resulting deposition in spawning gravels, North Coastal California, *Water Resources Research*, 25, 1303–1319.
- Major J. J., T. C. Pierson, R. L. Dinehard, and J. E. Costa (2000), Sediment yield following severe volcanic disturbance – A two-decade perspective from Mount St. Helens, *Geology*, 28 (9), 819-822.
- Merion D. I., G. R. Baker, and S. A. Orszag (1982), Analytic structure of vortex sheet dynamics. Part 1. Kelvin-Helmholtz instability, *Journal of Fluid Mechanics*, 114, 283-298.
- Middleton G. V. (1970), Experimental studies related to problems of flysch sedimentation, in *Flysch Sedimentology in North America*, edited by J. Lajoie, pp. 253-272, Geological Association of Canada, St John's Canada.
- Montgomery D. R., M. S. Panfil, and S. K. Hayes (1999), Channel-bed mobility response to extreme sediment loading at Mount Pinatubo, *Geology*, 27, 271-274.
- Parker G. and P. C. Klingeman (1982), On why gravel bed streams are paved, *Water Resources Research*, 18, 1409-1423.
- Recking A., P. Frey, A. Paquier, and P. Belleudy (2009), An experimental investigation of mechanisms involved in bed load sheet production and migration, *Journal of Geophysical Research*, 114, F03010.
- Venditti J. G., W. E. Dietrich, P. A. Nelson, M. A. Wydzga, J. Fadde, and L. Sklar (2010a), Effect of sediment pulse grain size on sediment transport rates and bed mobility in gravel bed rivers, *Journal of Geophysical Research: Earth Surface*, 115, F03039.
- Venditti J. G., W. E. Dietrich, P. A. Nelson, M. A. Wydzga, J. Fadde, and L. Sklar (2010b), Mobilization of coarse surface layers in gravel-bedded rivers by finer gravel bed load, *Water Resources Research*, 46, W07506.
- Wilcock P. R., S. T. Kenworthy, and J. C. Crowe (2001) Experimental study of the transport of mixed sand and gravel, *Water Resources Research*, 37, 3349–3358.

**D Article *ICIP* 2016**

# ONLINE MULTI-MODEL PARTICLE FILTER-BASED TRACKING TO STUDY BEDLOAD TRANSPORT

Hugo Lafaye de Micheaux<sup>\*†</sup>    Christophe Ducottet<sup>\*</sup>    Philippe Frey<sup>†</sup>

<sup>\*</sup> Université de Lyon, UJM-Saint-Étienne, CNRS, UMR5516, Laboratoire Hubert Curien  
F-42023 Saint-Étienne, France

<sup>†</sup> Université Grenoble Alpes, Irstea, UR ETGR, F-38402 St-Martin-d’Hères, France

## ABSTRACT

Multi-object tracking is a difficult problem underlying many computer vision applications. In this work, we focus on sediment transport experiments in a flow where sediments are represented by spherical calibrated beads. The aim is to track all beads over long time sequences to obtain sediment velocities and concentration. Classical algorithms used in fluid mechanics fail to track the beads over long sequences with a high precision because they incorrectly handle both miss-detections and detector imprecision. Our contribution is to propose a particle filter-based algorithm including an adapted multiple motion model. Additionally, this algorithm integrates several improvements to account for the lack of precision of the detector. The evaluation was made using a test sequence with a dedicated ground-truth. The results show that the method outperforms state-of-the-art concurrent algorithms.

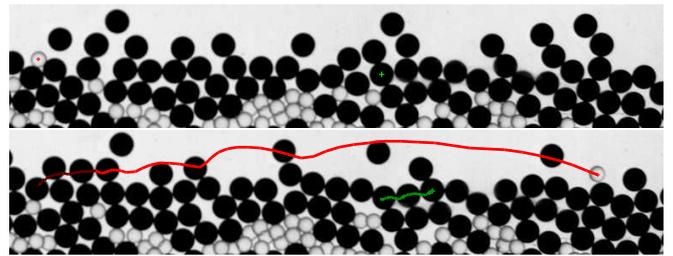
**Index Terms**— Particle filtering, multi-model tracking, multi-object tracking, detector confidence, bedload transport.

## 1. INTRODUCTION

Visual target tracking is a recurrent problem in computer vision, it has been used in many applications, *e.g.* video surveillance, sports analysis, or traffic safety. In particular, object tracking is used in fluid mechanics to track particles with techniques as PIV [1] and PTV [2]. We focus here on bedload sediment transport experiments. The aim is to track all spherical beads (see Fig. 1) over a long time to obtain particle velocities and concentrations [3], for studying bedload granular rheology [4], size segregation [5] and associated morphology [6].

In this context, most of the proposed algorithms are deterministic and rely on a standard two step approach: (1) object detection, (2) object tracking by solving a data association problem [7]. However, these are not robust enough to track objects over very long sequences with a good precision for object positions. This is mainly because they incorrectly handle miss-detections and do not take into account the lack of precision of the detector.

This research is funded by Irstea, labex OSUG@2020 and by the Rhône-Alpes region as part of its Environment Academic Research Community.



**Fig. 1.** Example of trajectories for two tracked beads (cropped images). Green color is for a coarse black bead and red is for a smaller transparent bead.

In recent years, particle filter-based approaches have been successfully used in many applications [8, 9, 10]. The main benefit of these methods is to enable the estimation of the distribution of the target’s system state incorporating a non linear motion model. In this context, Breitenstein *et al.* [11] proposed a very efficient online multi-target tracking approach accounting for the detector confidence to handle miss-detections. However, this method is not directly adapted to our problem mainly because it uses a single motion model unable to describe all occurring object interactions.

In this paper, using the same framework, we propose to introduce a multiple motion model dedicated to our simulation of bedload transport. Each motion model corresponds to a specific state of motion of our objects: resting, rolling or bouncing. We also introduce improvements to correctly handle the lack of precision of the detector providing tracking over long sequences with a high precision for objects position.

The paper is structured as follows. After discussing the related work in next section, Sec. 3 describes our algorithm and design choices. Sec. 4 presents the evaluation results of tracking performances and a comparison with concurrent algorithms. Finally, Sec. 5 concludes the paper.

## 2. RELATED WORK

Various visual tracking approaches have been reviewed [12, 13] and they can be separated in two categories: detection-

based methods and Bayesian filtering ones. In fluid mechanics, researchers usually employ the first one, *i.e.* taking the object detector output as a straight truth and then solving the association problem on pairwise frames [7, 14]. This deterministic approach lacks precision since it strongly relies on detectors that can be unreliable and imprecise.

Sequential monte carlo methods or particle filters were introduced to provide better estimations and predictions [15, 16, 17]. Basically, they consist of a dynamic model for prediction and an observation model to evaluate the likelihood of a predicted state. The main steps are: random generation of a sample of potential states called particles, propagation of these particles through the motion model and resampling of the distribution according to the observation model. To handle difficult tracking situations in a multi-object context, Breitenstein *et al.* [11] proposed an online and automatic particle filter-based algorithm that exploits the intermediate output of the object detector as a graded observation model. As most particle filter-based algorithms, it relies on a single motion model which can be problematic in an environment with complex dynamics.

To face this limitation, multiple dynamics models such as switching dynamical model approaches [18], have been recently developed under particle filters for robotics and navigation [19] or video surveillance [20]. These methods use the fact that object motions can be classified in different motion states according to the situation. Each motion state is dealt with specific state evolution model for prediction. However, it requires to know all motion states beforehand with a mathematical description for each. In contrast, our method exploits the mechanistic knowledge we have in our application.

### 3. MULTI-OBJECT TRACKING ALGORITHM

Our fully automatic multi-object tracking algorithm uses the same principle as [11]. Each trajectory is estimated by an individual particle filter (tracker), automatically initialized when a new object is detected and terminated when the object is leaving the field of view. At each time step we perform: (1) object detection, (2) association of each detection to a tracker, (3) update of the state of each particle filter according to the motion model and the new detection.

#### 3.1. Object detector

Black beads are detected by thresholding the image and taking the center of objects having an area close to the mean bead area. As transparent beads appear as faint dark rings of different shapes because of their neighboring beads, we use a specific chain of morphological operations to detect them. We apply a *hconvex* operator [21] on the image, then a normalized cross-correlation with a ring-shape model and finally we keep the relevant maxima using an adjusted threshold. The object detector returns a set of detections  $D$ .

To handle missing detections on transparent beads, we use the detector confidence as a graded observation model [11]. Based on the raw output of the detector, the detector confidence must give an estimation of the likelihood of a detection at each candidate pixel. In our case, for a given detection  $d \in D$ , we propose to define the confidence value as a decreasing function of the distance to the detection position  $\mathbf{x}_d$  with a maximum equal to the cross-correlation output  $\text{xcorr}(d)$  of the detector, *i.e.* the correlation value before thresholding. Formally, for each position  $\mathbf{x}$  of the image, the detector confidence density  $d_c(\mathbf{x})$  is defined as:

$$d_c(\mathbf{x}) = \frac{1}{k_D} \sum_{d \in D} \text{xcorr}(d) \exp(-\lambda \|\mathbf{x} - \mathbf{x}_d\|) \quad (1)$$

where  $\lambda$  denotes the exponential decay constant (see Sec. 3.4), and  $k_D$  is the normalization coefficient.

#### 3.2. Data association

To assign at most one detection to at most one tracker, a data association process is needed. Due to the high number of detections and the long time series, a greedy algorithm was found to be an effective solution. Given the matching cost  $c(tr, d)$  between all tracker-detection pairs, the greedy algorithm iteratively selects the best candidate and removes the corresponding concurrent associations. To limit the number of possible tracker-detection pairs, the set of possible detections associated to a given tracker is limited to the detections located inside a circular region centered at the predicted position  $\mathbf{x}_{\hat{tr}}$  given by this tracker. This predicted position is made assuming a constant velocity model.

Given a tracker  $tr$  and a detection  $d$ , the matching cost is:

$$c(tr, d) = \alpha \underbrace{\frac{\|\mathbf{x}_d - \mathbf{x}_{\hat{tr}}\|}{r_s}}_{\text{distance term}} + \beta \underbrace{\frac{|\mathbf{v}_d|}{\mathbf{v}_{max}}}_{\text{velocity term}} \quad (2)$$

where  $\mathbf{v}_d$  is the measured velocity of detection  $d$ <sup>1</sup>, the parameters  $\alpha$  and  $\beta$  represent the proportion of each term (see Sec. 3.4), and  $r_s$  is the radius of the circular searching region calculated from the fluid velocity. The distance term promotes the detections closest to the prediction. With the velocity term, we want to promote low velocities as it allows to have better association in case of beads collisions.

#### 3.3. Particle filtering

Each tracker is described by a fixed number  $N$  of particles, having a state  $\{\mathbf{x}_p, \mathbf{v}_p, s_p\}$  where  $\mathbf{x}_p = (x, y)$  denotes the position,  $\mathbf{v}_p = (u, v)$  the velocity and  $s_p$  the motion state. We employ the Sequential Importance Resampling filter, also

<sup>1</sup>The measured velocity  $\mathbf{v}_d$  corresponds to the displacement between the detection position  $\mathbf{x}_d$  and the estimated position at previous time step.

known as the bootstrap filter [16], to approximate the probability function. For one target, it works as follows: (1) Estimation of particle states, positions and velocities, (2) Particle importance weighting, (3) Normalization of weights, (4) Resampling, (5) Estimation of final state, position and velocity of the target. The main difference with [11], is the use of the velocity estimation of a particle as an observation in the importance weighting.

**State model.** In motion-based stochastic tracking [22], explicit motion measurements are used to guide predictions. In our application, beads have different behaviors according to their location, velocity and neighborhood. We can distinguish three distinct motion states: resting (not moving), rolling (rolling/sliding on others) and saltating (bouncing on others). To propagate the particles, we first update their state and then we apply a motion model to update the position and the velocity. To update the state, we draw the new state according to a conditional probability table (see Sec. 3.4). The motion models are:

$$\text{resting: } (x, y)_t = (x, y)_{t-1} + \varepsilon_{(x,y)}^{rest} \quad (3a)$$

$$\text{rolling: } (x, y)_t = (x, y)_{t-1} + (u)_{t-1} \Delta t + \varepsilon_{(x,y)}^{roll} \quad (3b)$$

$$(u)_t = (u)_{t-1} + \varepsilon_{(u)}^{roll} \quad (3c)$$

$$\text{saltating: } (x, y)_t = (x, y)_{t-1} + (u, v)_{t-1} \Delta t + \varepsilon_{(x,y)}^{salt} \quad (3d)$$

$$(u, v)_t = (u, v)_{t-1} + \varepsilon_{(u,v)}^{salt} \quad (3e)$$

where  $\varepsilon_{(x,y)}^{rest}$ ,  $\varepsilon_{(x,y)}^{roll}$ ,  $\varepsilon_{(x,y)}^{salt}$  are the process noises on the position and  $\varepsilon_{(u)}^{roll}$ ,  $\varepsilon_{(u,v)}^{salt}$  are the process noises on the velocity, they are all independently drawn from zero-mean normal distributions ;  $\Delta t$  depends on the framerate of the sequence.

**Particle weighting.** The importance weight  $w_{tr,p}$  for a particle  $p$  of tracker  $tr$  is given by the conditional likelihood of the new observation given the propagated particle. Given detection  $d^*$  associated to the tracker, we have:

$$w_{tr,p} = \mathcal{I}(tr) \left( \underbrace{\delta p_{\mathcal{N}}(\mathbf{x}_p - \mathbf{x}_{d^*})}_{\text{particle position}} + \underbrace{\gamma p_{\mathcal{N}}(\mathbf{v}_p - \mathbf{v}_{d^*})}_{\text{particle velocity}} \right) + \underbrace{\eta d_c(\mathbf{x}_p)}_{\text{det. conf}} \quad (4)$$

where  $\mathcal{I}(tr)$  is an indicator function that returns 1 if a detection is associated to the tracker and 0 otherwise.  $\delta$ ,  $\gamma$  and  $\eta$  are set experimentally (see Sec. 3.4).

The particle position (resp. velocity) term calculates the distance between particle position (resp. velocity) and the measured detection position (resp. velocity), and is evaluated under a normal distribution  $p_{\mathcal{N}}$ . The detector confidence term evaluates the detector confidence density at the particle position.

**Resampling and estimating.** After the weighting, the resampling is used to solve particle degeneracy, namely removing the particles of small weight and reproducing those of large weights [23]. Finally, the position and the velocity are estimated by averaging the resampled particles.

### 3.4. Implementation details

The number of particles  $N$  was fixed to  $N = 100$ , being a good compromise between computational cost and tracking reliability. The parameters in equations (1), (2) and (4) have been set experimentally and remained the same for the different experiments used in Sec. 4.1.  $\lambda$  was set to 1 in eq. (1). In eq. (2),  $\alpha$  and  $\beta$  were set respectively to 0.75 and 0.25 to give more importance to the distance term. The proportion parameters  $(\delta, \gamma, \eta)$  in eq. (4) were fixed at  $(20, 1, 1)$ . Therefore, when a detection is associated, the particle weight is mainly influenced by its position and also a little by its velocity, to reduce the imprecision of the detection position.

Each motion state in eqs. (3) has its own variance for the process noise on position and velocity. The initial particle states were set as rolling, and their initial positions and velocities were drawn from a Normal distribution with positions centered at the detection center and velocities centered to null velocity. To handle the difficulty of dealing with new trackers, we increased the variances to make the motion model more flexible during the first 3 frames. A tracker survives only 10 frames without associated detection and is then automatically terminated.

For a given particle, the transition from one state to another is controlled with conditional probabilities. We have estimated these probabilities on a representative training sequence. Concerning the particle velocity at the beginning of a track, the estimation was set to the measured particle velocity (*i.e.* the displacement between the particle position and the estimated position at previous time step).

## 4. RESULTS

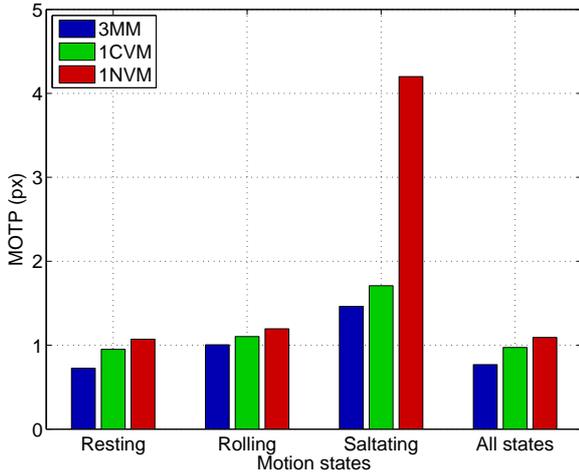
In this section, we present the results of the proposed tracking algorithm with three motion models (3MM) (see Sec. 3.3) against two simplified versions using only one motion model: one with a constant velocity model (1CVM) and the other with a null velocity model (1NVM). These simplified versions can be considered as direct adaptations of Breitenstein *et al.* [11] algorithm. The results are compared to a ground truth to have information about the false positives and false negatives. We also use the CLEAR MOT metrics [24] to evaluate the tracking performance.

### 4.1. Experiments and ground truth

We worked on a 1000-frames sequence recorded at 130 fps with approximately 400 beads per frame (about 300 coarse and 100 small beads). The sequence is split in two parts, one for optimizing the parameters, and the other part for evaluation. We created a ground truth on the sequence by selecting manually the detections on each frame. To get the ground truth of trajectories, we applied a simple data association algorithm on the ground truth of detections just created.

Algo. Models	% Correct Tracks	MOTP	FN	FP	Id Sw.	MOTA
3MM	<b>98.35%</b>	<b>0.64px</b>	<b>0.07%</b>	<b>0.16%</b>	<b>4</b>	<b>99.77%</b>
1CVM [11]	95.05%	0.77px	0.09%	0.17%	23	99.72%
1NVM [11]	89.90%	0.89px	0.14%	0.54%	84	99.28%

**Table 1.** Tracking evaluation results for the three algorithm configurations. It shows the percentage of correct tracks (*i.e.* more than 95% of the track is correct) and the CLEAR MOT metrics [24] such as precision (MOTP), false negative rate (FN), false positive rate (FP), number of mismatches (Id Sw.) and accuracy (MOTA).



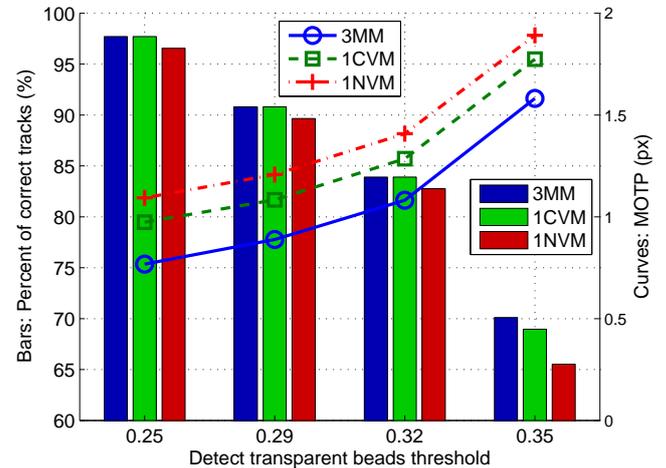
**Fig. 2.** Precision score MOTP on transparent beads tracking according to motion state (resting, rolling, saltating) for each algorithm model (3MM, 1CVM, 1NVM).

#### 4.2. Evaluation of the tracking algorithm

In Tab. 1, we present the evaluation of the three tracking algorithms 3MM, 1CVM and 1NVM. The results show that 3MM is always better than the two others. Especially for the percentage of correct tracks (a track is considered as correct on the total length if more than 95% of it has no false negatives or mismatches), the 3MM reaches 98.35%. We also computed the CLEAR MOT metrics to evaluate the precision score MOTP (average error in estimated position) and the accuracy score MOTA (accounts for all object configuration errors made by the tracker, false negatives, false positives and mismatches). Here again, the 3MM appears to be the most precise and accurate.

In Fig. 2, we plotted the precision MOTP of the tracking of transparent beads for each motion state described in Sec. 3.3. The precision depends on the motion states, the higher the velocity (saltating > rolling > resting), the worse the precision. However, our 3MM performs better on each motion state especially on saltation.

In Fig. 3, we varied the threshold of the transparent beads detector in order to see the tracking performance with different levels of detection. It illustrates the influence of this threshold on the percentage of correct tracks and the precision MOTP on transparent beads.



**Fig. 3.** Evaluation of tracking performance on transparent beads for different detection thresholds. The percentage of correct tracks (bars, left scale) and the precision score MOTP (curves, right scale) are plotted.

Again, 3MM performs better even when the miss detection rate increases due to the increasing threshold.

## 5. CONCLUSION

We have presented a new online particle filter algorithm based on multiple dynamic models for automatic multi-object tracking over long sequences. Having *a priori* information about the object mechanical dynamics, we were able to approach the real trajectories. This allows us to study bedload transport with higher confidence.

Our multiple motion model based algorithm provides a high tracking precision and accuracy when applied on detectors of different quality. Moreover, the approach has been shown to outperform state-of-the-art algorithms presenting a single motion model such as constant or null velocity model.

A possible extension would be to use the state of neighboring objects as an information to help choosing between motion states and bring more estimation precision. Finally, the tracking algorithm could be applied on very long sequences to observe lower frequency phenomena.

## 6. REFERENCES

- [1] J. Westerweel, G. E. Elsinga, and R. J. Adrian, "Particle image velocimetry for complex and turbulent flows," *Annual Review of Fluid Mechanics*, vol. 45, pp. 409–436, 2013.
- [2] K. Ohmi and H.-Y. Li, "Particle-tracking velocimetry with new algorithms," *Measurement Science and Technology*, vol. 11, no. 6, pp. 603, 2000.
- [3] P. Frey, "Particle velocity and concentration profiles in bedload experiments on a steep slope," *Earth Surface Processes and Landforms*, vol. 39, no. 5, pp. 646–655, 2014.
- [4] R. Maurin, J. Chauchat, B. Chareyre, and P. Frey, "A minimal coupled fluid-discrete element model for bedload transport," *Physics of Fluids (1994-present)*, vol. 27, no. 11, pp. 113302, 2015.
- [5] V. Bacchi, A. Recking, N. Eckert, P. Frey, G. Piton, and M. Naaim, "The effects of kinetic sorting on sediment mobility on steep slopes," *Earth Surface Processes and Landforms*, vol. 39, no. 8, pp. 1075–1086, 2014.
- [6] A. Recking, V. Bacchi, M. Naaim, and P. Frey, "Antidunes on steep slopes," *Journal of Geophysical Research: Earth Surface*, vol. 114, no. F4025, 2009.
- [7] V. Hergault, P. Frey, F. Métyvier, C. Barat, C. Ducottet, T. Böhm, and C. Ancey, "Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow," *Experiments in Fluids*, vol. 49, no. 5, pp. 1095–1107, nov 2010.
- [8] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 2, pp. 125–137, 2001.
- [9] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 8, pp. 1470–1475, 2007.
- [10] W.-L. Lu, K. Okuma, and J. J. Little, "Tracking and recognizing actions of multiple hockey players using the boosted particle filter," *Image and Vision Computing*, vol. 27, no. 1, pp. 189–205, 2009.
- [11] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 9, pp. 1820–1833, sep 2011.
- [12] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 7, pp. 1442–1468, jul 2014.
- [13] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer vision and pattern recognition (CVPR)*. jun 2013, pp. 2411–2418, IEEE.
- [14] N. T. Ouellette, H. Xu, and E. Bodenschatz, "A quantitative study of three-dimensional lagrangian particle tracking algorithms," *Experiments in Fluids*, vol. 40, no. 2, pp. 301–313, nov 2006.
- [15] A. Doucet, A. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*, Springer Science & Business Media, 2001.
- [16] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*. IET, 1993, vol. 140, pp. 107–113.
- [17] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [18] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *IEEE 6th International Conference on Computer Vision (ICCV)*. jan 1998, pp. 107–112, IEEE.
- [19] Y. Boers and J. N. Driessen, "Interacting multiple model particle filter," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 150, no. 5, pp. 344–349, 2003.
- [20] Y. Zhai, M. B. Yeary, S. Cheng, and N. Kehtarnavaz, "An object-tracking algorithm based on multiple-model particle filtering with state partitioning," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 5, pp. 1797–1809, 2009.
- [21] P. Soille, *Morphological image analysis: Principles and Applications*, Springer-Verlag Berlin Heidelberg, 1999.
- [22] J.-M. Odobez, D. Gatica-Perez, and S. O. Ba, "Embedding motion in model-based stochastic tracking," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3514–3530, nov 2006.
- [23] A. Milstein, J. N. Sánchez, and E. T. Williamson, "Robust global localization using clustered particle filtering," in *AAAI/IAAI*, 2002, pp. 581–586.
- [24] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *Journal on Image and Video Processing*, , no. 246309, 2008.

**E Article *MVA* (en examen)**

# Multi-model particle filter-based tracking with switching dynamical state to study bedload transport

Hugo Lafaye de Micheaux · Christophe Ducottet · Philippe Frey

Received: date / Accepted: date

**Abstract** Multi-object tracking is a difficult problem underlying many computer vision applications. In this work, we focus on bedload sediment transport experiments in a turbulent flow where sediments are represented by small spherical calibrated glass beads. The aim is to track all beads over long time sequences to obtain sediment velocities and concentration. Classical algorithms used in fluid mechanics fail to track the beads over long sequences with a high precision because they incorrectly handle both miss-detections and detector imprecision. Our contribution is to propose a particle filter-based algorithm including a multiple motion model adapted to our problem. Additionally, this algorithm includes several improvements such as the estimation of the detector confidence to account for the lack of precision of the detector. The evaluation was made using two test sequences - one from our experimental setup and one from a simulation created numerically - with their dedicated ground truths. The results show that this algorithm outperforms state-of-the-art concurrent algorithms.

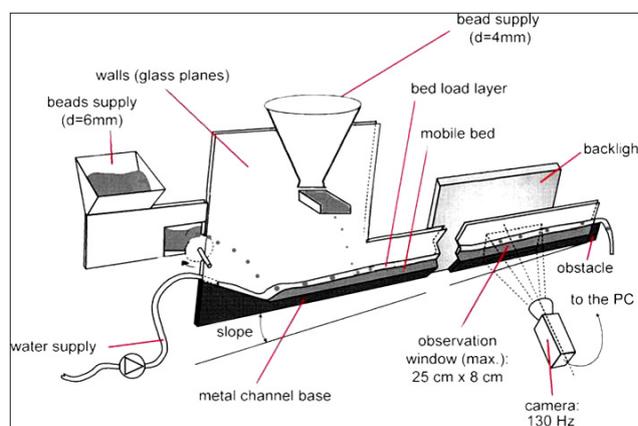
**Keywords** Visual object tracking · multiple targets tracking · particle filter · switching dynamical state · detector confidence · bedload transport

## 1 Introduction

Visual target tracking is a recurrent problem in computer vision, it has been used in many applications, *e.g.* video surveillance, sports analysis, or traffic safety. In particular, object

H. Lafaye de Micheaux · Christophe Ducottet  
Université de Lyon, UJM-Saint-Étienne, CNRS, UMR5516, Laboratoire Hubert Curien, F-42000 Saint-Étienne, France  
E-mail: hugo.lafaye-de-micheaux@irstea.fr

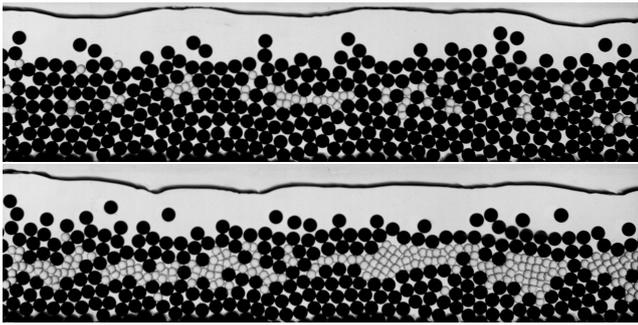
H. Lafaye de Micheaux · Philippe Frey  
Université Grenoble Alpes, Irstea, UR ETGR, 2 rue de la Papeterie-BP 76, F-38402 St-Martin-d'Hères, France



**Fig. 1** Sketch of the experimental arrangement (modified from [8, 16]). Water flows from left to right, entraining a mixture of two-size spherical glass beads. Black beads of diameter  $d=6\text{mm}$  are introduced upstream and transparent beads of diameter  $d=4\text{mm}$  further downstream. The mobile bed is observed through a backlit window and recorded by a high-speed video camera at 130Hz.

tracking is used in fluid mechanics to track particles with techniques such as Particle Image Velocimetry (PIV) [38] and Particle Tracking Velocimetry (PTV) [31]. We focus here on bedload sediment transport experiments conducted with a laboratory facility, as illustrated by the sketch in Fig. 1. Experiments consist in a two-dimensional steep channel where mixtures of two-size spherical glass beads are entrained by a turbulent supercritical water flow with a mobile bed. The aim is to track all beads (see Fig. 2) over a long time to obtain trajectories, particle velocities and concentrations [13], for studying bedload granular rheology [26], size segregation [4] and associated morphology [34].

In this context, PIV with background modeling-based trackers can be used to detect moving objects by background subtraction [17, 37]. But the main problem with these methods is that it does not resolved strong gradients (*e.g.* of ve-



**Fig. 2** Example of images from the sequence recorded during an experiment. An image with a small number of transparent beads at the beginning of the sequence (top), and an image with a much higher number of transparent beads later in the sequence (bottom). The black line on top of the images is the water free surface. The image resolution is 1280x320.

locity). Moreover, it is very sensitive to framerate, changing background, moving cameras and can remove locally non-moving objects that are not really part of the background. In body-part, person or car tracking applications, these problems can be faced through tracking-by-detection approaches to gain robustness. These approaches have become increasingly popular by global optimization based on position, size and appearance similarity [1, 15, 22]. In fluid mechanics, the particle tracking algorithms (PTV) are deterministic (or data association-based tracking) and rely on a standard two step approach: (1) object detection, (2) object tracking by solving a data association problem [16]. However, they are not robust enough to track objects over very long sequences with a good precision for object positions. This is mainly because they fully trust the object detector output, so they incorrectly handle miss-detections and do not take into account the lack of precision of the detector.

In recent years, particle filter-based approaches have been successfully used in many applications [10, 24, 33]. The main benefit of these methods is to enable the estimation of the distribution of the target’s system state incorporating a non linear motion model. In this context, Breitenstein et al. [9] proposed a very efficient online multi-target tracking approach accounting for the detector confidence to handle miss-detections. However, this method is not directly adapted to our problem mainly because it uses a single motion model unable to describe all occurring object interactions.

In this paper, using particle filters and with this idea of detector confidence, we propose to introduce a Bayesian mixed-state framework [20] dedicated to our simulation of bedload transport. It is based on multiple motion models where each motion model corresponds to a specific state of motion linked with the mechanics of our objects: resting, rolling or bouncing also called saltation. Such an approach permits to follow the natural behavior of spherical beads in a fluid flow. We also introduce several improvements to correctly handle the lack of precision of the detector providing

tracking over long sequences with a high precision for object positions.

The paper is structured as follows. After discussing the related work in the next section, Sec. 3 describes our proposed multiple model particle filter-based tracking algorithm and design choices. Sec. 4 presents the evaluation and a discussion of the results of tracking performances and a comparison with concurrent algorithms. Finally, Sec. 5 provides a conclusion of the paper.

## 2 Related work

This work stands at the intersection of computer vision, fluid mechanics and Bayesian statistical inference ; it builds on previous approaches we will discuss below.

Various visual tracking approaches have been reviewed [35, 40] and they can be separated in two categories: detection-based methods and Bayesian filtering ones. In fluid mechanics, researchers usually employ the first one, *i.e.* taking the object detector output as a straight truth and then solving the association problem on pairwise frames [8, 16]. It is also the case for the Lagrangian Particle Tracking (LPT) [18, 32] widely used in experimental fluid dynamics. This kind of approach is deterministic and lacks precision since the association problem between detections and targets strongly relies on the object detector that can be unreliable and imprecise.

To handle this problem, several recent algorithms optimize detection assignments over a large temporal window in an offline step [2, 22, 42], *i.e.* by using information from future frames and after observing the complete sequence. Nevertheless, these methods are time-critical and unsuitable for online applications<sup>1</sup>, unlike Sequential Monte Carlo (SMC) methods. SMC or particle filters (PF) were introduced to provide better estimations and predictions in an online manner [11, 14, 19]. Basically, they consist of a dynamic model for prediction and an observation model to evaluate the likelihood of a predicted state. The main steps are: random generation of a sample of potential states called particles, propagation of these particles through the motion model and resampling of the distribution according to the observation model.

In order to cope with difficult tracking situations in a multi-object context, Breitenstein et al. [9] proposed an automatic PF-based algorithm that exploits the detector confidence (*i.e.* the intermediate output of the object detector) as a graded observation model. It allows using a combination of unreliable information sources for robust multi-person tracking. As most particle filter-based algorithms, it relies

<sup>1</sup> An application is online when the inputs to the associated algorithm is provided as a stream, piece-by-piece.

on a single motion model which can be problematic in an environment with complex dynamics.

To face this limitation, multiple model (MM) paradigm [3, 23], also called switching dynamical model approach, has been introduced and is used for several applications such as robotics and navigation [7] or video surveillance [41]. It assumes that, given a finite state of dynamic models (each of them corresponding to a particular behavior), a particular target can jump from one model to the other according to a set of transition probabilities. The interacting multiple model (IMM) algorithm [6, 27] was the first to propose this approach through jump Markov process modeling. Then, the multiple model particle filter (MMPF) framework [20, 28] came as a promising alternative outperforming the IMM algorithm in a bearings-only tracking problem [3]. The MMPF method uses the idea of MM, *i.e.* object motions can be classified in different motion states depending on the situation, in a PF framework. Each motion state is dealt with a specific state evolution model for prediction. However, it requires to know all motion states beforehand with a mathematical description for each. In contrast, our method exploits the mechanistic knowledge of behaviors we have in our application.

### 3 Multi-object multi-model tracking algorithm

Our fully automatic multi-object tracking algorithm uses the same principle as [30] by embedding multiple motion models in a particle filter-based approach (MMPF). Moreover, it uses the technique of switching dynamical model introduced by [20] to increase the motion models flexibility of visual trackers. Thus, each physical trajectory is estimated by an individual particle filter (tracker), automatically initialized when a new object is detected and terminated when the object is leaving the field of view. A global overview of the algorithm is shown in Fig. 3. At each time step we perform: (1) object detection, (2) estimation of the detector confidence density, (3) association of each detection to a tracker, (4) update of the state of each particle filter according to the specific motion model and the new detection. The principle of the MMPF tracking algorithm is to predict the target state (*i.e.* target location, velocity and motion state) by applying an adapted motion model depending on the target behavior. This target state is then corrected with measurements made on the observed image to estimate the final target state. The target estimated at time  $t$  is further used for data association at time  $t + 1$ . Each part of the algorithm is explained below.

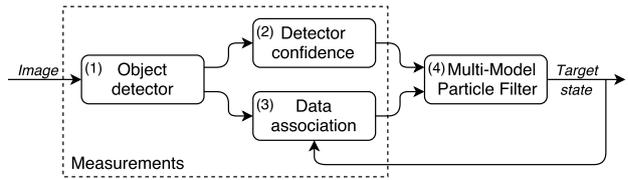


Fig. 3 Overview of the different parts of the multi-object tracking algorithm.

#### 3.1 Object detection

For each image, the object detection phase aims at detecting all beads by image processing. It consists in 3 steps: (1) detect black beads, (2) remove them from the image, (3) detect transparent beads. Facing lacks of reliability with the transparent bead detector, a special treatment is applied on it (cf. Sec. 3.1.2).

##### 3.1.1 Object detectors

First, black beads are detected by thresholding the image at level  $\tau_b$  to keep only black pixels, and by taking the center of connected components having an area close to the mean bead area. This detector is very efficient as black beads are easily distinguishable from other kind of beads and they are not overlapping (6mm diameter beads in a channel width of 6.5mm). So it always returns the entire set of black beads. Then, black beads are erased by replacing their gray levels by those of the background. The background is reconstructed through a morphological closing with a disk shaped structuring element (radius just greater than bead radius). This makes transparent beads more discernible. Secondly, since transparent beads appear as faint dark rings of different shapes because of their neighboring configurations (4mm diameter beads that can be partially overlapped by others), we use a specific chain of morphological operations to detect them (initially developed by [16]). We apply a *hconvex* operator [36] on the image obtained at the previous step, then a normalized cross-correlation with a ring-shape model, local maxima extraction and maxima selection according to the adjusted threshold  $\tau$ . It returns the set of transparent beads with some fallibility due to partially occluded beads. How this problem is dealt with is detailed in next section. Finally, the object detector returns a set of detections  $D$  containing the center of the black and transparent beads.

##### 3.1.2 Detector confidence estimation

To handle missing detections on transparent beads, we introduce a detector confidence estimation and we use it as a graded observation model [9]. The principle is: based on the raw output of the detector, the detector confidence must give an estimation of the likelihood of a detection at each candidate pixel. Considering the whole set  $D^*$  of local maxima

outputs of the detector (down to a very low threshold  $\tau^*$ ), each detection  $d \in D^*$  has a location  $\mathbf{x}^d$  and a cross-correlation coefficient  $\text{xcorr}(d)$  with the ring shape model. Assuming an exponential distribution for the position of each detection, we can estimate the likelihood of observing an object at location  $\mathbf{x}^{obs}$  given the set of detections  $D^*$  as:

$$\begin{aligned} p(\mathbf{x}^{obs} | D^*) &= d_c(\mathbf{x}^{obs}) \\ &= \frac{1}{k_{D^*}} \sum_{d \in D^*} \text{xcorr}(d) \exp(-\lambda \|\mathbf{x}^{obs} - \mathbf{x}^d\|_2) \end{aligned} \quad (1)$$

where  $\|\cdot\|_2$  denotes the Euclidean distance between two points in Euclidean space,  $\lambda$  denotes the exponential decay constant (see Sec. 3.4), and  $k_{D^*}$  is the normalization coefficient.

### 3.2 Data association

To assign at most one detection to at most one tracker, a data association process is needed. Due to the high number of detections and the long time series, a greedy algorithm was found to be an effective and sufficient solution [39] with a lower computational cost than the optimal Hungarian algorithm [21].

#### 3.2.1 Greedy data association

Given the matching cost  $g(tr, d)$  between all tracker-detection pairs computed as described below, the greedy algorithm iteratively selects the best candidate and removes the corresponding concurrent associations. To limit the number of possible tracker-detection pairs, the set of possible detections associated to a given tracker is limited to the detections located inside a circular region centered at the predicted position  $\hat{\mathbf{x}}^{tr}$  given by this tracker. This predicted position is made assuming a constant velocity model.

#### 3.2.2 Matching cost

The matching cost  $g(tr, d)$  between a detection  $d$  and a tracker  $tr$  consists of a combination of two factors: one based on the distance of the detection position  $\mathbf{x}^d$  to the predicted position of the tracker  $\hat{\mathbf{x}}^{tr}$ , the other based on the tracker velocity at the previous time step  $\mathbf{u}^{tr}$ . It is defined as:

$$g(tr, d) = \alpha \underbrace{\frac{\|\mathbf{x}^d - \hat{\mathbf{x}}^{tr}\|_2}{r_s}}_{\text{distance term}} + \beta \underbrace{\frac{|\mathbf{u}^{tr}|}{|\mathbf{u}|_{max}}}_{\text{velocity term}} \quad (2)$$

where  $|\cdot|$  denotes the Euclidean norm of a vector. The distance term promotes the detections closest to the prediction. With the velocity term, we promote low velocities as it allows

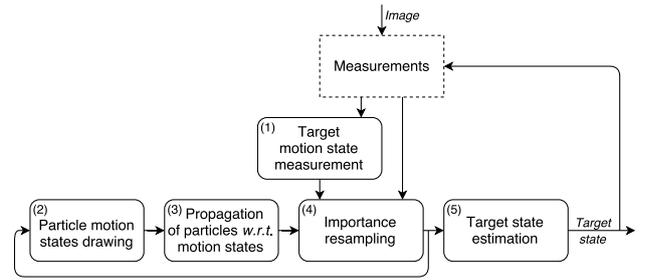


Fig. 4 The multi-model particle filter-based tracking algorithm.

better associations in case of bead collisions. When associating the first two frames, the velocity term is set to 0 since we do not have a velocity value yet. The value  $|\mathbf{u}|_{max}$  is a normalization coefficient equal to the highest velocity norm in the previous time step, the parameters  $\alpha$  and  $\beta$  represent the proportion of each term (see Sec. 3.4), and  $r_s$  is the radius of the circular searching region calculated from the fluid velocity, it is used here as a normalization coefficient. Therefore, the greedy algorithm returns the subset of detections associated to a tracker. The location  $\mathbf{x}^d$  (resp. velocity  $\mathbf{u}^d$ ) of the detection is considered as the observation variable  $\mathbf{x}^{obs}$  (resp.  $\mathbf{u}^{obs}$ ) of this tracker. Additionally, motion state  $s^{obs}$  will be associated to this tracker (see Sec. 3.3.1).

### 3.3 Multi-model particle filtering

Each tracker (or trajectory) is described by a separate particle filter of  $N$  particles. At time step  $t$  the state space  $\mathbf{c}_t^i$  of a particle  $i$  is defined as  $\{\mathbf{x}_t^i, \mathbf{u}_t^i, s_t^i\}$  where  $\mathbf{x}_t^i = (x, y)$  denotes the position,  $\mathbf{u}_t^i = (u, v)$  the velocity and  $s_t^i$  the motion state. The observation variable at time  $t$  is given by  $\mathbf{z}_t = \{\mathbf{x}_t^{obs}, \mathbf{u}_t^{obs}, s_t^{obs}\}$  resulting from measurement of location, velocity and motion state. We employ the Sequential Importance Resampling (SIR) filter, also known as the bootstrap filter [14] where the state transition density is used as importance distribution, to approximate the probability function. For one target, the algorithm goes through different steps described in Fig. 4. It works as follows: (1) Measurement of target motion state according to neighbors configuration and target velocity, (2) Random selection of new motion states according to conditional probability table, (3) Update of particle positions and velocities according to motion models, (4) Particle importance weighting, normalizing and resampling, (5) Estimation of final motion state, position and velocity of the target. The main difference with the algorithm in [29], is the use of the target motion state estimation as an observation in the importance weighting (see Sec. 3.3.2).

### 3.3.1 Motion models based on mechanical dynamics

In motion-based stochastic tracking [30], explicit motion measurements are used to guide predictions. In our application, beads have different behaviors according to their location, velocity and neighborhood. We can distinguish three distinct motion states linked to the dynamics in bedload transport: resting (not moving), rolling (rolling/sliding on others) and saltating (bouncing on others). In the saltating state, we can discriminate two sub-states: a constant motion and a rebound motion. To propagate a particle  $i$  at time step  $t$ , we first update its previous motion state  $s_{t-1}^i$ , and then we apply a specific motion model to update its position and velocity. The new updated motion state  $s_t^i$  is drawn according to a conditional probability table  $\pi$  (see Sec. 3.4). The motion models are:

$$\begin{aligned}
\text{resting: } (x, y)_t &= (x, y)_{t-1} + \varepsilon_{rest}^{(x,y)} \\
\text{rolling: } (x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \Delta t + \varepsilon_{roll}^{(x,y)} \\
&\quad (u, v)_t = (u, v)_{t-1} + \varepsilon_{roll}^{(u,v)} \\
\text{saltating-constant: } (x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \Delta t + \varepsilon_{salt}^{(x,y)} \\
&\quad (u, v)_t = (u, v)_{t-1} + \varepsilon_{salt}^{(u,v)} \\
\text{saltating-rebound: } (x, y)_t &= (x, y)_{t-1} + (u, -v)_{t-1} \Delta t + \varepsilon_{salt}^{(x,y)} \\
&\quad (u, v)_t = (u, -v)_{t-1} + \varepsilon_{salt}^{(u,v)}
\end{aligned} \tag{3}$$

where  $\varepsilon_{rest}^{(x,y)}$ ,  $\varepsilon_{roll}^{(x,y)}$ ,  $\varepsilon_{salt}^{(x,y)}$  are the process noises on the position and  $\varepsilon_{roll}^{(u)}$ ,  $\varepsilon_{salt}^{(u,v)}$  are the process noises on the velocity (see Sec. 3.4), they are all independently drawn from zero-mean normal distributions ;  $\Delta t$  depends on the framerate of the sequence.

**Motion state measurement.** Before applying the sequential importance resampling filtering on a target, it is possible to make an estimation of its motion state with the observations. The measurement of this motion state is computed by inspecting the neighbors configuration and velocity of the target with the help of the data association output. Indeed the idea is to use our mechanistic knowledge to best fit the mechanical dynamics of our objects. Given the number of neighboring beads  $\text{nbN}^{obs}$  touching the observed bead, we can consider the motion state to be "resting" if either  $\text{nbN}^{obs}$  is significantly high or the velocity norm  $|\mathbf{u}^{obs}|$  of the bead is low. Similarly, the motion state will be "saltating" if  $\text{nbN}^{obs}$  is low or  $|\mathbf{u}^{obs}|$  is high. Formally, we have:

$$s^{obs} = \begin{cases} \text{resting,} & \text{if } \text{nbN}^{obs} \geq \text{nbN}_{rest} \text{ or } |\mathbf{u}^{obs}| \leq u_{rest}, \\ \text{saltating,} & \text{if } \text{nbN}^{obs} \leq \text{nbN}_{salt} \text{ or } |\mathbf{u}^{obs}| \geq u_{salt}, \\ \text{rolling,} & \text{otherwise.} \end{cases} \tag{4}$$

where  $\text{nbN}_{rest}$  (resp.  $\text{nbN}_{salt}$ ) is the threshold number of neighbors for resting (resp. saltating) and  $u_{rest}$  (resp.  $u_{salt}$ ) is the threshold velocity norm for resting (resp. saltating).

These values are determined experimentally on the test sequence (see Sec. 3.4).

The estimation of the motion state is used as a cue in the particle weighting (in Sec. 3.3.2) or to set the final motion state of the target (in Sec. 3.3.3).

### 3.3.2 Particle importance weighting

**Observation model.** The importance weight  $w_t^i$  for a particle  $i$  at time step  $t$  is updated from its value at previous time step  $w_{t-1}^i$  and from the conditional likelihood of the new observation  $z_t$  given the previous state  $c_{t-1}^i$  of the particle:

$$w_t^i \propto w_{t-1}^i p(z_t | c_{t-1}^i) \tag{5}$$

Thanks to resampling in each time step, all weights are the same at the end and  $w_{t-1}^i = 1/N$ , thus this term can be ignored. Moreover, in practice, the prior is usually chosen as the importance distribution to approximate the probability density function [14]. With these simplifications applied to Eq. (5), the particle weighting only depends on the propagated particle state and is so quite simpler:

$$w_t^i \propto p(z_t | c_t^i) \tag{6}$$

**Importance weighting.** Assuming independence between observation variables, we have:

$$\begin{aligned}
p(z_t | c_t^i) &= p(\{\mathbf{x}_t^{obs}, s_t^{obs}\} | c_t^i) \\
&= p(\mathbf{x}_t^{obs} | c_t^i) p(s_t^{obs} | c_t^i)
\end{aligned} \tag{7}$$

For transparent beads, there is not always a detection associated to the tracker. In such cases, the likelihood of particle  $i$  at location  $\mathbf{x}_t^i$  is given by the detector confidence. Otherwise, it is estimated by a normal distribution centered on the corresponding detection  $d$  (see Sec. 3.4 for the variance value). Formally:

$$p(\mathbf{x}_t^{obs} | c_t^i) = \begin{cases} p_{\mathcal{N}}(\mathbf{x}_t^i - \mathbf{x}_t^{obs}) & \text{if } I(tr) = 1, \\ d_c(\mathbf{x}_t^i) & \text{if } I(tr) = 0. \end{cases} \tag{8}$$

where  $I(tr)$  is an indicator function that returns 1 if a detection is associated to the tracker and 0 otherwise, it is given by the data association.

For the motion state likelihood  $p(s_t^{obs} | c_t^i)$ , we assume a constant value  $\gamma$  if  $s_t^i = s_t^{obs}$  and  $1 - \gamma$  otherwise. This value is set experimentally (see Sec. 3.4). Formally:

$$p(s_t^{obs} | c_t^i) = p_{\gamma}(s_t^i) = \begin{cases} \gamma, & \text{if } s_t^i = s_t^{obs}, \\ 1 - \gamma, & \text{otherwise.} \end{cases} \tag{9}$$

Then, the update of weights is given by:

$$w_t^i = \begin{cases} p_{\gamma}(s_t^i) p_{\mathcal{N}}(\mathbf{x}_t^i - \mathbf{x}_t^{obs}) & \text{if } I(tr) = 1, \\ p_{\gamma}(s_t^i) d_c(\mathbf{x}_t^i) & \text{otherwise (trans. bead),} \\ p_{\gamma}(s_t^i) & \text{otherwise (black bead).} \end{cases} \tag{10}$$

Note that the last case of Eq. (10) corresponds to a black bead having no associated detection. Since there is no detector confidence for black beads, no weight term is based on the location. Nevertheless, this last case rarely happens since the data association is very efficient when applied on reliable detectors such as the black bead one (see Sec. 3.1.1).

**Normalizing and resampling.** After weighting all particles, the weights are normalized so that  $\sum_{i=1}^N w_t^i = 1$ . Then, the resampling is used to solve particle degeneracy, namely removing particles with low weights and multiplying those with high weights. The approach used here is the multinomial resampling based on the bootstrap method [12].

### 3.3.3 Final estimation

Finally, the position and the velocity of the target are estimated by averaging the resampled particles. After all positions and velocities of the targets are estimated, we measure again their motion state as it was done before the particle importance weighting (see Sec. 3.3.1 and Eq. (4)). The reason why we reassess the motion state is that we want to be consistent with the mechanical motion state. Indeed, taking the most represented motion state of the resampled particles does not guaranty to give a motion state relevant to the mechanical behavior of the target.

### 3.4 Implementation details

**Equation parameters.** The number of particles  $N$  was fixed to  $N = 100$ , being a good compromise between computational cost and tracking reliability. The parameters in Eqs. (1) and (2) have been set experimentally and remained the same for the different experiments used in Sec. 4.2.  $\lambda$  was set to 1 in Eq. (1). In Eq. (2),  $\alpha$  and  $\beta$  were set respectively to 0.75 and 0.25 to give more importance to the distance term. The variance of the normal distribution in Eq. (8) was set experimentally to 3 pixels for both x- and y-coordinates. The reward parameter  $\gamma$  in Eq. (9) was fixed to  $2/3$  in order to promote particles having the same motion state as the measurement.

**Process noise and tracker parameters.** Each motion state in Eqs. (3) has its own variance for the process noise on position and velocity. They were set automatically by analyzing representative training sequences corresponding to the first halves of the studied sequences given in Sec. 4.2. The initial particle positions and velocities were drawn from a Normal distribution with positions centered at the detection center and velocities centered to null velocity. To handle the difficulty of dealing with new trackers, we increased the variances to make the motion model more flexible during the first 3 frames. To stop bad predictions, a tracker at resting state (resp. rolling and saltating) survives only 100 frames

(resp. 10 frames and 5 frames) without associated detection and is then automatically terminated.

**Transition probability table.** For a given particle  $i$ , the transition from one motion state  $s_{t-1}^i$  to another  $s_t^i$  is controlled with conditional probabilities. We have estimated these probabilities on representative training sequences corresponding to the first halves of the studied sequences given in Sec. 4.2. As the probabilities were found to be nearly the same for all datasets, we present only one case here. The transition probability table  $\pi$  required for the switching state process presented in Sec. 3.3.1, was estimated as:

$$\pi = p(s_t^i | s_{t-1}^i) = \begin{bmatrix} 0.95 & 0.05 & 0.00 \\ 0.07 & 0.92 & 0.01 \\ 0.00 & 0.09 & 0.91 \end{bmatrix} \quad (11)$$

As  $N$  was fixed to 100 and we want each state to have a chance to be represented at each time step, we modified the transition probability table to be slightly less restrictive by imposing a minimal probability of 0.05, leading to:

$$\pi \approx \begin{bmatrix} 0.80 & 0.15 & 0.05 \\ 0.10 & 0.80 & 0.10 \\ 0.05 & 0.15 & 0.80 \end{bmatrix} \quad (12)$$

The saltating state being divided in two sub-states, when it is drawn after getting through  $\pi$ , there is then a probability of 0.5 to be in the saltating-constant sub-state (resp. saltating-bouncing sub-state). At initialization, the particle motion states were set as *unknown* states. At the next step, the particles had an equivalent chance to go in any of the motion states.

**Motion state measurement thresholds.** To measure the motion state of a target (Eq. (4)), we need to calculate its number of neighbors. We consider a detection as a neighbor of the target if they are in contact, *i.e.* when the distance between both centers is equal to the sum of their corresponding radius. The thresholds on the number of neighbors for the resting and saltating states were set experimentally to  $\text{nbN}_{rest} = 5$  and  $\text{nbN}_{salt} = 0$  respectively. The thresholds on velocity norms were set experimentally to  $u_{rest} = 0.015\text{m/s}$  and  $u_{salt} = 0.25\text{m/s}$  for the sequences studied in this paper (cf. Sec. 4.2).  $u_{rest}$  and  $u_{salt}$  are specific to each sequence. These four thresholds were adjusted by trial and error to minimize the differences between the motion state determined by the algorithm and the motion state determined by the naked eye.

## 4 Results and discussion

In this section, we present the results of the proposed tracking algorithm against other algorithms of the literature. They are all compared to a ground truth of a short experimental

sequence obtained with our facilities and a longer numerical one obtained with a simulation. Indeed, as our case study is very specific, it is impossible to test our tracking algorithm on open datasets. We also use the CLEAR MOT metrics [5] to evaluate the tracking performance. Moreover, we study how the algorithms deal with different detector qualities and how good they estimate the motion states.

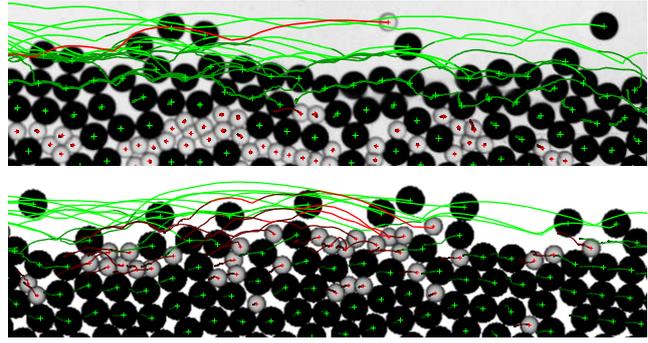
#### 4.1 Algorithms

To study the reliability and efficiency of our method, we tested it against three other algorithms of the literature. First of all, we have our particle filter-based tracking algorithm (see Sec. 3) which is based on the use of three motion models (or states), the detector confidence and a measurement of the motion state as a cue in the likelihood (MMPF+). Second, we have our preliminary work on three motion models-based particle filters with detector confidence [29] (MMPF). Third, we have an adaptation of the algorithm of Breitenstein et al. [9] based on a single constant velocity model and detector confidence particle filters (PF). Finally, we have an initial and simple deterministic tracking approach previously developed in our laboratory [16] (DET). This last algorithm blindly relies on the output of the detector, without any special treatment for the false positives and false negatives.

#### 4.2 Datasets and ground truths

To compare these algorithms, we worked on two datasets: one from the experiment described in Sec. 1 and one from a simulation created numerically. Both datasets are freely available<sup>2</sup>.

**Experimental sequence.** The experimental sequence is a 1000-frame sequence recorded at 130 fps with approximately 400 beads per frame (about 300 coarse and 100 small beads). The image resolution is 1280x320. The sequence is split in two parts, one for optimizing and estimating the parameters, and the other part for evaluation. We created a ground truth on this sequence by first detecting beads with our object detectors on each frame (Sec. 3.1.1), and then by editing visually the detections, *i.e.* by removing the false positives and adding manually the false negatives. To get the ground truth of trajectories, we applied our data association algorithm (Sec. 3.2) on the ground truth of detections just created and then computed the motion states of each target by watching neighbors configuration (as in [16]). The limitation of this ground truth is the precision of the positions. Indeed, the center of missing detections were selected manually to fit visually the center of beads as good as possible. So they



**Fig. 5** Example of trajectories for the experimental ExpSeq (top) and simulated NumSeq (bottom) datasets (cropped images). Green color is for a coarse black bead and red is for a smaller transparent bead. The green and red color gradations from darker to lighter show respectively the three motion states (resting, rolling and saltating).

can be inaccurate by one or two pixels. Moreover, as we used the direct output of the detector to create this ground truth, the deterministic algorithm cannot be part of the comparison of the algorithms to study precision as it also used the direct output of the detector. Nevertheless, this ground truth sequence is fine to study how the different algorithms deal with false positives and false negatives on a real experimental case. The experimental sequence will be denoted ExpSeq, an example of trajectories is given in Fig. 5 (top).

**Numerical sequence.** In order to study a longer sequence and lower frequency phenomena, we created a numerical sequence of bedload transport of 10,000 images thanks to a model developed at Irstea based on a coupled fluid discrete element method (cf. [25] and [26]) and validated against our experiments (cf. [13]). The idea of this simulation is to reproduce as best as possible our experiments (*e.g.* flume size, number of each type of beads, feeding rates, fluid velocity, image frequency, etc.). Moreover, thanks to ParaView, an open source software used for scientific visualization, we were able to have a rendering of images of the obtained sequence really close to our experimental images. Thanks to this rendering, we were able to use this numerical sequence in the same manner as an experimental sequence and apply all the previously presented algorithms. As for ExpSeq, half of the sequence is used to optimize and estimate the parameters, and the other for evaluation. Unlike the ground truth of ExpSeq, we can use the numerical sequence as a ground truth for all tests since it is computed from an external method and without any detection error. The numerical sequence will be denoted NumSeq, an example of trajectories is given in Fig. 5 (bottom).

**Ground truth motion states.** On the obtained ground truth sequences, we need to assign a motion state to each tracker at each time step. To do this, we follow the offline method initially developed by Böhm et al. [8] which consists in combining two criteria: first testing the tracker

<sup>2</sup> <http://webperso.univ-st-etienne.fr/~ducottet/>

Dataset	Algorithm	Correct Tracks	MOTP	FN	FP	Id Sw.	MOTA
ExpSeq	MMPF+	<b>99.57%</b>	<b>98.40%</b>	<b>0.21%</b>	0.27%	<b>9</b>	<b>99.52%</b>
	MMPF [29]	99.36%	97.96%	0.23%	0.34%	9	99.42%
	PF [9]	99.15%	97.29%	0.24%	0.36%	13	99.39%
	DET [16]	99.57%	-	0.30%	<b>0.21%</b>	34	99.47%
NumSeq	MMPF+	<b>98.50%</b>	97.38%	<b>0.14%</b>	0.61%	<b>85</b>	<b>99.24%</b>
	MMPF [29]	98.50%	97.21%	0.15%	0.66%	111	99.18%
	PF [9]	97.59%	96.40%	0.15%	0.66%	91	99.19%
	DET [16]	98.05%	<b>97.62%</b>	0.17%	<b>0.59%</b>	145	99.23%

**Table 1** Tracking evaluation results for the four algorithms (our MMPF+, the MMPF from [29], the PF from [9], the DET from [16]) on both ExpSeq and NumSeq datasets (respectively an experimental and a numerical sequence). It shows the percentage of correct tracks (*i.e.* track correct when more than 95% of its length is correct), the false negative rate (FN), the false positive rate (FP), the number of mismatches (Id Sw.) and the CLEAR MOT metrics [5] such as the precision score (MOTP) and the accuracy score (MOTA). In bold the best score for each metrics.

velocity (averaged over five consecutive frames) against a threshold velocity and second testing the distance between the tracker and its nearest neighbor against a reference distance. The main difference with the way we measure the motion states in our algorithm is that here it is done after the tracking execution in a post-processing as it uses information from future frames, while we do it during the tracking execution in an online manner. We consider that the offline method gives motion states close enough to real motion behaviors to be used as ground truth.

#### 4.3 Evaluation of the tracking algorithm

To evaluate the tracking performance, we use the CLEAR MOT metrics [5]. It gives a precision score MOTP (intersection over union of the estimated bead and the ground truth) and an accuracy score MOTA (accounts for all object configuration errors, *i.e.* false negatives, false positives and mismatches<sup>3</sup>). We also use the percentage of correct tracks, where a track is considered as correct on its total length if more than 95% of it has no false negatives or mismatches. In order to see the influence of the detector output on the different tracking algorithms, we vary the threshold for the detection of the transparent beads. It gives different detector qualities and allows seeing how the tracking algorithms deal with more false negatives or false positives. At last, we study how well the multiple motion model-based algorithms estimate the motion state of the beads.

**Quantitative analysis.** In Table 1, we present the evaluation results of the four tracking algorithms MMPF+, MMPF, PF and DET on both datasets ExpSeq and NumSeq. It shows that most of the time our MMPF+ method outperforms the three others. Particularly, MMPF+ has better results than the other particle filter-based algorithms MMPF and PF in all metrics. For example, our precision MOTP and accuracy MOTA results are higher for both datasets. Note that all precision and accuracy scores are already high. This is due to the good quality of the transparent bead detectors for the

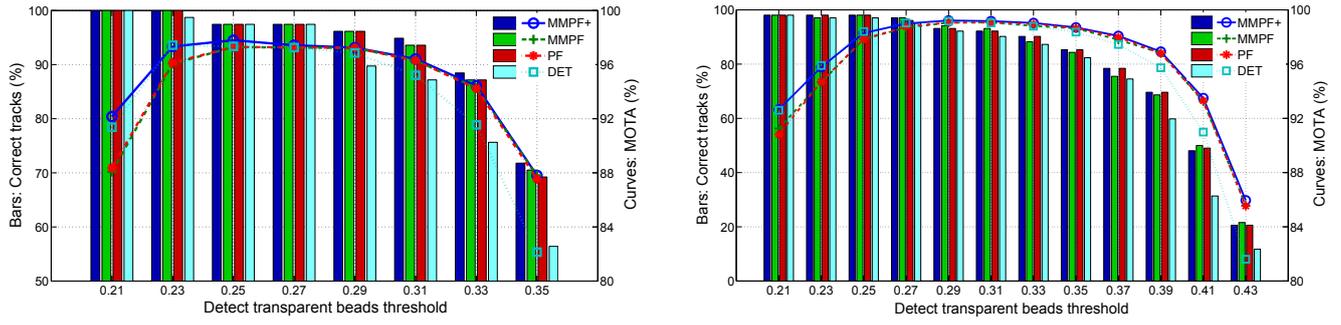
chosen thresholds. We chose the threshold of the transparent bead detector  $\tau = 0.25$  for ExpSeq and  $\tau = 0.27$  for NumSeq as they give the best tracking results and as we know the importance of good detectors for reliable tracking operations. But the detector is not always that efficient as it depends on the image quality of the sequences. The analysis of the effect of different  $\tau$  is done below.

We can also observe that the number of identity switches is the lowest for MMPF+. This property is one of the most important for a good tracking algorithm. In our images, ID switches can happen for tracks changing abruptly of direction (*e.g.* rebounding tracks) or velocity (*e.g.* acceleration  $\gg 0$ ), or for newly appearing beads located really close to where a track should be according to its motion model. The implementation of multiple motion models in our method allows better dealing with these situations: this was one of the motivations of the MMPF+ algorithm.

In terms of false negative rates, our algorithm surpasses the others too. Indeed, in absence of an associated detection to a tracker, the DET algorithm stops immediately the corresponding tracker leading to a false negative, contrary to particle filter algorithms that estimate its position with the help of its previous behavior. Our algorithm is better than the PF method since it uses multiple adapted motion models to predict positions. And it is also better than MMPF since it estimates the motion states more closely to the real behavior of the beads.

Nevertheless, the deterministic approach seems to be the best in terms of false positive rate. This is mainly because of the termination of trackers at borders (*i.e.* side effects). Indeed, in the particle filter algorithms, when there is no associated detection to a tracker because the bead left the field of view, the tracker tries to estimate its new state. If its estimated position is still inside the field of view, the tracker continues, resulting in a false positive. It survives like this a limited number of frames without associated detection and is then automatically terminated (see Sec. 3.4 for the surviving times). On the contrary, the DET algorithm immediately stops a tracker when there is no associated detection. So, in

<sup>3</sup> A mismatch is a mistaken switch of tracker identifiers.



**Fig. 6** Evaluation of performances of the tracking of transparent beads for the ExpSeq (left graph) and NumSeq (right graph) datasets, and for different qualities of the transparent bead detector that are controlled by the detection thresholds. The percentage of correct tracks (bars, left scales) and the accuracy score MOTA (curves, right scales) are plotted.

this case of side effect and in both studied datasets, it appears that the DET method is better than the particle filter methods.

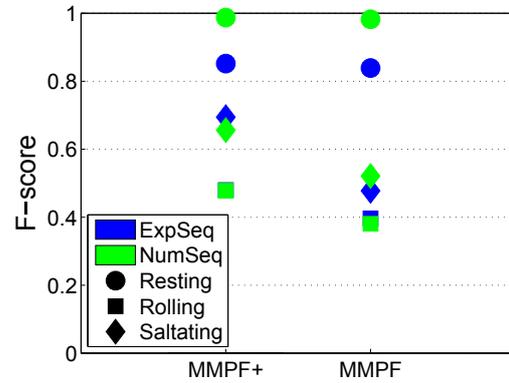
**Impact of detector quality.** In order to study the tracking performance with different levels of detection, we varied the threshold  $\tau$  of the transparent bead detector. It allows assessing how the detection quality influences the percentage of correct tracks and the accuracy MOTA on transparent beads. We made this analysis for the four algorithms and on both ExpSeq (Fig. 6, left) and NumSeq (Fig. 6, right) datasets.

The results show that our algorithm globally outperforms the three others in terms of accuracy for both datasets. Indeed, we can observe that for low thresholds (more false positives) the MOTA score is higher for MMPF+ than the others, as well as for high thresholds (more false negatives). This results from the good management of detection errors: quickly stopping trackers wrongly created by false detections and continuing predicting trackers without associated beads when missing detections.

Whatever the algorithm used, the tracking performances depend on the detector quality: the higher  $\tau$ , the smaller the percentage of correct tracks. Especially, the DET algorithm without the prediction step is the most impacted by high values of  $\tau$ , as expected.

**Motion state estimation.** The motion states of trackers are not only used to set specific motion models for the particle filters. They are also used to study the behavior of trackers in each motion state. The aim is to analyze, for each motion state separately, the number of trackers at each time step, the velocity profiles, and the infiltration and concentration inside the mobile bed. For reliable analysis, we need the estimated motion states to be as close as possible to mechanical states. Therefore, we compare them against the motion states of the ground truths.

In Fig. 7, we present the F-score of the state identification obtained by our algorithm against MMPF for both ExpSeq and NumSeq. For each motion state, the F-score is a measure of accuracy of the classification in this motion state that



**Fig. 7** Evaluation of the reliability of the estimated motion state of the trackers given by F-scores for both datasets ExpSeq and NumSeq and for the three motion states resting, rolling and saltating.

considers both the precision<sup>4</sup> and recall<sup>5</sup>. We choose F-score instead of accuracy since accuracy yields misleading results if the dataset is unbalanced, *i.e.* when the number of samples in different classes vary greatly, as it is the case for our datasets with approximately 2 times less rolling beads and 20 times less saltating beads than resting beads.

Firstly, we can observe that results are almost the same for both datasets. This confirms that the numerical dataset can be used to evaluate multiple model tracking algorithms. Secondly, the MMPF+ algorithm outperforms MMPF for all states which confirms its better ability to predict the state. In particular, the accuracy of saltating beads is increased by 0.2 for the MMPF+ algorithm especially thanks to the introduction of the two saltating sub-states 'constant' and 'rebound' (see Sec. 3.3.1). Thirdly, algorithms perform more or less well depending on the motion state. Indeed, resting beads are very well classified (0.85) compared to saltating ones (0.70) or rolling ones (0.45). This is not surprising since resting beads are easy to classify thanks to their low velocity and high number of neighbors. On the contrary, rolling beads can be confused either with resting beads when they have low

<sup>4</sup> The precision is the fraction of retrieved instances that are relevant.

<sup>5</sup> The recall is the fraction of relevant instances that are retrieved.

velocity or with saltating beads when they are moving fast on top of the bed.

## 5 Conclusions

We have presented a new online particle filter algorithm based on multiple dynamic models for automatic multi-object tracking over long sequences. Our motivation was the study of bedload transport in an experimental channel with the tracking of spherical beads entrained by a water flow. The first contribution of our work is the use of several motion models to guide bead trackers with adapted predictions. Having *a priori* information about the object mechanical dynamics, we are able to better estimate object trajectories. The second contribution concerns the handling of unreliable detections, *i.e.* false positives and false negatives (missing detections). A robust bead tracking is obtained through the estimation of the detector confidence to guide particles of the filter when no discrete high-confidence detection is issued by the detector. Furthermore, we use the state of neighboring objects as an information in the likelihood of the particles to help choosing between motion states.

To evaluate the tracking performances, we carefully prepared two ground truth datasets, one created manually on an experimental sequence of images and one created numerically through a simulation reproducing the experimental conditions. When compared to state-of-the-art algorithms, our algorithm provides the highest tracking precision and accuracy. Moreover, it appears to be the least impacted by missing detections and presents less identity switches than the others when applied on detectors of different qualities. In terms of classification of the bead behaviors in motion states, our classifier outperforms another one from a similar multiple model algorithm thanks to inspection of neighbors configurations and velocities.

In our images, the main problem is to detect efficiently the transparent beads as they appear sometimes partially occluded. A possible extension would be to train the detector on several configuration problems instead of using a single appearance model of the bead. Furthermore, particles of the filter could be used in the data association step with their specific motion model for predictions. The tracking algorithm will be applied on very long sequences to observe both high and lower frequency phenomena which appear in sediment transport.

**Acknowledgements** This research was supported by Irstea; labex OSUG@2020; INSU programme EC2CO-BIOHEFECT and DRIL Modsed; the French national research agency project SegSed ANR-16-CE01-0005; and the Rhône-Alpes region as part of its higher education, research and innovation regional Strategy (Environment Academic Research Community).

## References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1–8. IEEE (2008). doi: [10.1109/CVPR.2008.4587583](https://doi.org/10.1109/CVPR.2008.4587583)
2. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on* **50**(2), 174–188 (2002). doi: [10.1109/78.978374](https://doi.org/10.1109/78.978374)
3. Arulampalam, M.S., Ristic, B., Gordon, N., Mansell, T.: Bearings-only tracking of manoeuvring targets using particle filters. *Journal on Advances in Signal Processing, EURASIP* **2004**(15), 1–15 (2004). doi: [10.1155/S1110865704405095](https://doi.org/10.1155/S1110865704405095)
4. Bacchi, V., Recking, A., Eckert, N., Frey, P., Piton, G., Naaim, M.: The effects of kinetic sorting on sediment mobility on steep slopes. *Earth Surface Processes and Landforms* **39**(8), 1075–1086 (2014). doi: [10.1002/esp.3564](https://doi.org/10.1002/esp.3564)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal on Image and Video Processing* **2008**(246309), 1–10 (2008). doi: [10.1155/2008/246309](https://doi.org/10.1155/2008/246309)
6. Blom, H.A.P., Bar-Shalom, Y.: The interacting multiple model algorithm for systems with Markovian switching coefficients. *Automatic Control, IEEE transactions on* **33**(8), 780–783 (1988). doi: [10.1109/9.1299](https://doi.org/10.1109/9.1299)
7. Boers, Y., Driessen, J.N.: Interacting multiple model particle filter. *Radar, Sonar and Navigation, IEE Proceedings* **150**(5), 344–349 (2003). doi: [10.1049/ip-rsn:20030741](https://doi.org/10.1049/ip-rsn:20030741)
8. Böhm, T., Frey, P., Ducottet, C., Ancey, C., Jodeau, M., Reboud, J.L.: Two-dimensional motion of a set of particles in a free surface flow with image processing. *Experiments in Fluids* **41**(1), 1–11 (2006). doi: [10.1007/s00348-006-0134-9](https://doi.org/10.1007/s00348-006-0134-9)
9. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on* **33**(9), 1820–1833 (2011). doi: [10.1109/TPAMI.2010.232](https://doi.org/10.1109/TPAMI.2010.232)
10. Choset, H., Nagatani, K.: Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *Robotics and Automation, IEEE Transactions on* **17**(2), 125–137 (2001). doi: [10.1109/70.928558](https://doi.org/10.1109/70.928558)
11. Doucet, A., de Freitas, A., Gordon, N.: Sequential Monte Carlo methods in practice. Springer Science & Business Media (2001). doi: [10.1007/978-1-4757-3437-9](https://doi.org/10.1007/978-1-4757-3437-9)
12. Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. CRC press (1994). doi: [10.1007/978-1-4899-4541-9](https://doi.org/10.1007/978-1-4899-4541-9)
13. Frey, P.: Particle velocity and concentration profiles in bedload experiments on a steep slope. *Earth Surface Processes and Landforms* **39**(5), 646–655 (2014). doi: [10.1002/esp.3517](https://doi.org/10.1002/esp.3517)
14. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F* **140**(2), 107–113 (1993). doi: [10.1049/ip-f-2.1993.0015](https://doi.org/10.1049/ip-f-2.1993.0015)
15. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision (ECCV), vol. 7575, pp. 702–715. Springer (2012). doi: [10.1007/978-3-642-33765-9\\_50](https://doi.org/10.1007/978-3-642-33765-9_50)
16. Hergault, V., Frey, P., Métivier, F., Barat, C., Ducottet, C., Böhm, T., Ancey, C.: Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow. *Experiments in Fluids* **49**(5), 1095–1107 (2010). doi: [10.1007/s00348-010-0856-6](https://doi.org/10.1007/s00348-010-0856-6)
17. Honkanen, M., Nobach, H.: Background extraction from double-frame PIV images. *Experiments in Fluids* **38**(3), 348–362 (2005).

- doi: [10.1007/s00348-004-0916-x](https://doi.org/10.1007/s00348-004-0916-x)
18. Houssais, M., Ortiz, C.P., Durian, D.J., Jerolmack, D.J.: Onset of sediment transport is a continuous transition driven by fluid shear and granular creep. *Nature communications* **6**(6527), 1–8 (2015). doi: [10.1038/ncomms7527](https://doi.org/10.1038/ncomms7527)
  19. Isard, M., Blake, A.: Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision* **29**(1), 5–28 (1998). doi: [10.1023/A:1008078328650](https://doi.org/10.1023/A:1008078328650)
  20. Isard, M., Blake, A.: A mixed-state condensation tracker with automatic model-switching. In: *Computer Vision (ICCV), IEEE 6th International Conference on*, pp. 107–112. IEEE (1998). doi: [10.1109/ICCV.1998.710707](https://doi.org/10.1109/ICCV.1998.710707)
  21. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval research logistics quarterly* **2**(1-2), 83–97 (1955). doi: [10.1007/978-3-540-68279-0\\_2](https://doi.org/10.1007/978-3-540-68279-0_2)
  22. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on* **30**(10), 1683–1698 (2008). doi: [10.1109/TPAMI.2008.170](https://doi.org/10.1109/TPAMI.2008.170)
  23. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part V. Multiple-model methods. *Aerospace and Electronic Systems, IEEE Transactions on* **41**(4), 1255–1321 (2005). doi: [10.1109/taes.2005.1561886](https://doi.org/10.1109/taes.2005.1561886)
  24. Lu, W.L., Okuma, K., Little, J.J.: Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing* **27**(1), 189–205 (2009). doi: [10.1016/j.imavis.2008.02.008](https://doi.org/10.1016/j.imavis.2008.02.008)
  25. Maurin, R., Chauchat, J., Chareyre, B., Frey, P.: A minimal coupled fluid-discrete element model for bedload transport. *Physics of Fluids (1994-present)* **27**(11), 113,302 (2015). doi: [10.1063/1.4935703](https://doi.org/10.1063/1.4935703)
  26. Maurin, R., Chauchat, J., Frey, P.: Dense granular flow rheology in turbulent bedload transport. *Journal of Fluid Mechanics* **804**, 490–512 (2016). doi: [10.1017/jfm.2016.520](https://doi.org/10.1017/jfm.2016.520)
  27. Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J.: Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on* **34**(1), 103–123 (1998). doi: [10.1109/7.640267](https://doi.org/10.1109/7.640267)
  28. McGinnity, S., Irwin, G.W.: Multiple model bootstrap filter for maneuvering target tracking. *Aerospace and Electronic Systems, IEEE Transactions on* **36**(3), 1006–1012 (2000). doi: [10.1109/7.869522](https://doi.org/10.1109/7.869522)
  29. Lafaye de Micheaux, H., Ducottet, C., Frey, P.: Online multi-model particle filter-based tracking to study bedload transport. In: *Image Processing (ICIP), IEEE International Conference on*, pp. 3489–3493. IEEE (2016). doi: [10.1109/ICIP.2016.7533008](https://doi.org/10.1109/ICIP.2016.7533008)
  30. Odobez, J.M., Gatica-Perez, D., Ba, S.O.: Embedding motion in model-based stochastic tracking. *Image Processing, IEEE Transactions on* **15**(11), 3514–3530 (2006). doi: [10.1109/TIP.2006.877497](https://doi.org/10.1109/TIP.2006.877497)
  31. Ohmi, K., Li, H.Y.: Particle-tracking velocimetry with new algorithms. *Measurement Science and Technology* **11**(6), 603–616 (2000). doi: [10.1088/0957-0233/11/6/303](https://doi.org/10.1088/0957-0233/11/6/303)
  32. Ouellette, N.T., Xu, H., Bodenschatz, E.: A quantitative study of three-dimensional Lagrangian particle tracking algorithms. *Experiments in Fluids* **40**(2), 301–313 (2006). doi: [10.1007/s00348-005-0068-7](https://doi.org/10.1007/s00348-005-0068-7)
  33. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A.: Tracking deforming objects using particle filtering for geometric active contours. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on* **29**(8), 1470–1475 (2007). doi: [10.1109/TPAMI.2007.1081](https://doi.org/10.1109/TPAMI.2007.1081)
  34. Recking, A., Bacchi, V., Naaim, M., Frey, P.: Antidunes on steep slopes. *Journal of Geophysical Research: Earth Surface* **114**(F04025), 1–11 (2009). doi: [10.1029/2008JF001216](https://doi.org/10.1029/2008JF001216)
  35. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on* **36**(7), 1442–1468 (2014). doi: [10.1109/TPAMI.2013.230](https://doi.org/10.1109/TPAMI.2013.230)
  36. Soille, P.: *Morphological image analysis: Principles and Applications*. Springer-Verlag Berlin Heidelberg (1999). doi: [10.1007/978-3-662-05088-0](https://doi.org/10.1007/978-3-662-05088-0)
  37. Theunissen, R., Scarano, F., Riethmuller, M.: On improvement of PIV image interrogation near stationary interfaces. *Experiments in Fluids* **45**(4), 557–572 (2008). doi: [10.1007/s00348-008-0481-9](https://doi.org/10.1007/s00348-008-0481-9)
  38. Westerweel, J., Elsinga, G.E., Adrian, R.J.: Particle image velocimetry for complex and turbulent flows. *Annual Review of Fluid Mechanics* **45**(1), 409–436 (2013). doi: [10.1146/annurev-fluid-120710-101204](https://doi.org/10.1146/annurev-fluid-120710-101204)
  39. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision* **75**(2), 247–266 (2007). doi: [10.1007/s11263-006-0027-7](https://doi.org/10.1007/s11263-006-0027-7)
  40. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 2411–2418. IEEE (2013). doi: [10.1109/CVPR.2013.312](https://doi.org/10.1109/CVPR.2013.312)
  41. Zhai, Y., Yeary, M.B., Cheng, S., Kehtarnavaz, N.: An object-tracking algorithm based on multiple-model particle filtering with state partitioning. *Instrumentation and Measurement, IEEE Transactions on* **58**(5), 1797–1809 (2009). doi: [10.1109/tim.2009.2014511](https://doi.org/10.1109/tim.2009.2014511)
  42. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8. IEEE (2008). doi: [10.1109/CVPR.2008.4587584](https://doi.org/10.1109/CVPR.2008.4587584)