HAL open science

CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking

Priyanto Hidayatullah, Hubert Konik

To cite this version:

Priyanto Hidayatullah, Hubert Konik. CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking. Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, Jul 2011, Bandung, Indonesia. pp.1-6, 10.1109/ICEEI.2011.6021825 . ujm-00634639

HAL Id: ujm-00634639

https://ujm.hal.science/ujm-00634639v1

Submitted on 21 Oct 2011

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.

2011 International Conference on Electrical Engineering and Informatics
17-19 July 2011, Bandung, Indonesia

CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking

P. Hidayatullah*, H. Konik**

* *Computer Engineering Department, Bandung State Polytechnic, Bandung, Indonesia*

** *Laboratoire Hubert Curien UMR 5516, Université Jean Monnet, 42000 Saint-Etienne, France*

* priyanto@polban.ac.id

** hubert.konik@univ-st-etienne.fr

Abstract— **CAMSHIFT (Continuously Adaptive Mean-Shift) has been well accepted as one of the prominent methods in object tracking. CAMSHIFT is good for single hue object tracking and in the condition where object's color is different with background's color. In this paper, we enhance CAMSHIFT so it can be used for multi-object tracking and improve the robustness of CAMSHIFT for multi-hue object tracking especially in the situation where object's colors are similar with background's colors. We propose a more precise object localization by selecting each dominant color object part using a combination of Mean-Shift segmentation and region growing. Hue-distance, saturation and value color histogram are used to describe the object. We also track the dominant color object parts separately and combine them together to improve robustness of the tracking on multi-hue object. For multi-object tracking, we use a separate tracker for each object. Our experiments showed that those methods improved CAMSHIFT robustness significantly and enable CAMSHIFT for multi-object tracking.**

Keywords— **Object tracking, CAMSHIFT, Segmentation, Mean-Shift**

I. INTRODUCTION

Object tracking has been one of the most emerging areas in computer vision. There are a lot of object tracking methods available which has advantages along with their disadvantages. One of which is Continuously Adaptive Mean-Shift (CAMSHIFT).

CAMSHIFT is an object tracking method which is a modification of Mean-Shift tracking method. Mean-Shift itself is a robust nonparametric technique for finding the mode (peak) in a probability distribution[1],[12]. In CAMSHIFT, Mean-Shift algorithm is modified so that it can deal with dynamically changing color probability distribution which is taken from the video frames[1].

CAMSHIFT is very good for single hue object tracking and in the condition where object's color is different with background's color. In this paper, our main aims are to improve the robustness of CAMSHIFT for multi-hue object tracking and the situation where object's colors are similar with background's colors . Our objective also to enhance CAMSHIFT so that it can be used for multi-object tracking.

II. THE CAMSHIFT ALGORITHM

The CAMSHIFT algorithm can be summarize with these steps [1]:

1. Choose the initial region of interest, which contains the object we want to track.
2. Make a color histogram of that region as the object model.
3. Make a probability distribution of the frame using the color histogram. As a remark, in the implementation, they use the histogram back projection method.
4. Based on the probability distribution image, find the center mass of the search window using mean-shift method.
5. Center the search window to the point taken from step 4 and iterate step 4 until convergence.
6. Process the next frame with the search window position from the step 5.

A. Color probability distribution and histogram back projection

In order CAMSHIFT can track colored object, it needs a probability distribution image. They use the HSV color system and using only hue component to make the object's color 1D histogram[1]. This histogram is stored to convert next frames into corresponding probability of the object. The probability distribution image itself is made by back projecting the 1D hue histogram to the hue image of the frame. The result called backproject image. CAMSHIFT is then used to track the object based on this backproject image.

Regarding histogram back projection, it is a technique to find the probability of a histogram in an image. It means each pixel of the image is evaluated on how much probability it has to the histogram.

B. Mass center, orientation and scaling calculation

The mean location of the probability image inside search window is computed using zeroth and first image moments. The orientation and scale are calculated using second image moments[6].

III. PREVIOUS WORK

Several researchers have tried to improve CAMSHIFT in various ways. Ning et. al. in [7] proposed a joint color-texture histogram to represent a target and then applying it to the mean shift framework. The idea is that every pixel in the image is not only described by color information, but also the texture value using local binary pattern (LBP)

Ganoun et. al. in [9] tried to improve CAMSHIFT by adding feature points descriptors to the object model. One other implementation of this method is by Qiu Xuena et. al. in [8] which is using SIFT as the interest point descriptor combined with spatial features to create probability distribution of the tracked object. The interest point feature is well known with its invariant to illuminant, rotation and scaling.

In [4], G. Tian et. al. propose an improved H, S, V combined one-dimensional color histogram model for CAMSHIFT object tracking by making specific bins for every color component. Each bin also has a specific range.

J. A. Corrales et. al. in [3] use different approach. They use hue and saturation color component but modify the hue component. Instead of using hue, they use hue distance**.**

The hue distance is a function which represents each hue value *H* as a distance from a reference hue value *Href*. The following distance function is used instead of the hue component:

$$d(H,H_{ref}) = \begin{cases} |H - H_{ref}| if |H - H_{ref}| \le 180^o \\ 360^o - |H - H_{ref}| if |H - H_{ref}| > 180^o \end{cases} \quad (1)$$

The hue reference Href is the hue value which has the highest frequency in the histogram *h(x)* obtained from the histogram calculation in the first step of CAMSHIFT algorithm:

$$H_{ref} = Arg(Max(h(x))) \mid x \in \{0^o..360^o\} \quad (2)$$

Some researchers try to improve CAMSHIFT by improving the object localization. There is a method called foreground extraction, which tries to extract the foreground part with a specific formula. With this method, in the object rectangle, they make a circle area centered at the center of the rectangle. They put very high positive weight to the center of the circle and giving negative weight to the parts beyond the circle [11]. The formula is

$$FEM(x,y) = h_i - \sqrt{x^2 + y^2} \quad (3)$$

where *x,y* is a pixel coordinate, h_i is any value desired by user to filter out background color clusters.

Another object localization improvement is using weighted and ratio histogram[6]. This method has, in some way, similar with foreground extraction.

There is also improvement of CAMSHIFT using adaptive background (ABCShift). It tries to improve the tracker by modeling the background based on Bayesian probability model[10].

Improvement can also be made by background subtraction. The basic principle is to model the background and subtract the frame sequences by that background. For the conclusion, background subtraction helps CAMSHIFT very much in static background videos. Unfortunately, in dynamic background videos, background subtraction is not helpful. This is because the movement in the background is detected as foreground. Some implementations are available in [14].

IV. PROPOSED METHOD

In this paper, we propose several ways to improve CAMSHIFT. The use of multi-dominant color object localization and track the dominant color object separately are the key methods to improve CAMSHIFT. This section will describe the proposed method.

A. Object Localization

Object rectangle is the most common method to do object localization. User makes a rectangle to the object. This is simple and easy to use. Nevertheless, a problem may occur because most of the time the object is not exactly in rectangle shape. This makes some background's information are included in the object model. If this happens, drifting often occurs. The tracker is not robust in tracking the object.

Instead of using the classic rectangle or classic rectangle combined with foreground extraction[11] and weighted and ratio histogram[6], we propose object localization by combining mean-shift segmentation and region growing. Mean-shift is applied to segment each part of the object and make them homogeneous enough to be chosen easily. Mean-shift segmentation smoothen the image while preserving discontinuity[2]. This is actually happened until some level. Because if the object is in front of very similar color background at the localization phase, then the object will be merged with the background.

For mean-shift segmentation, we use mean-shift segmentation implementation in OpenCV library[14]. As proposed by Bradski et. al. in [5], we use spatial range (hs) = 20 and color range (hr) = 40 and maximum level = 2, which is good for an image with dimension 640 x 480.

The next step is we transform the image into HSV color space. We choose HSV because this model is based on

the human perception of eyes, which use the Munsell three-dimensional color coordinate system to present[4].

Next step is the object selection. After mean-shift segmentation, user can choose the object by clicking each object part. The clicks' position becomes set of "seed" points that have specific properties. From these seed points, the region grows by appending their neighbors which have similar properties to the seeds [13]. We use Flood Fill method, which is also available in OpenCV 2.0 library [14]. In addition, we give some tolerance values so that the seeds can grow further more until reaching the edges of object parts. Those selected object parts are considered as containing the dominant colors of the object which will be tracked in the tracking phase. The

formula is stated in (4) where S_{all} is the whole object which contains some set of dominant color object parts (S_1,S_2, ., S_n).

Each time an object part is selected, we store the minimum and maximum value of the object parts color component. This means we store the hmin, hmax, smin, smax, vmin, vmax. These values are needed to make color masking which will be describe in the next part.

One advantage with the proposed method is avoiding background's taken into the object model. This makes the tracker more robust and avoid drifting problem.

Another advantage is for some less object part case, this method is more practical. For example, for the yellow trunk which has only one hue, the selection is just clicking any part of it, and it will be chosen entirely (Figure 4b). Even though the shape of the object is complex and irregular, there will be no problem as long as it has homogeneous color.

$$S_{all} = S_1 \cup S_2 \cup ... \cup S_n \quad (4)$$

B. Object Modelling

The next step is object modeling. We use color histogram to model the object in HSV color space. In this paper, we adopt the idea of using the hue distance histogram[3]. However, we put a modification by adding value color component into the model instead of only using hue-distance and saturation component. This gives better discrimination.

We do quantization to each color histogram by using fewer numbers of bins. We use 30 bins for hue-distance component, nine bins for saturation and six bins for value. After histogram calculation which takes the hue-distance, saturation and value color component, we make a threshold to the hue histogram. This threshold is important to retain close hue pixels and remove unwanted far hue pixels in the back projection.

First we make a threshold of 255 for the histogram. In our case, which mostly uses video with 640x480 in size, the value of 255 is considered to be sufficient for a hue to be stated as dominant color (peak). Any bin that has value above the threshold will be stated as a peak.

Number of close hue is 70% of the number of peaks. In our experiments, 70% is our optimal value. If we use less than that, the tracker will lose the object and if we use more than that the tracker will drift. We round it after because we need an integer number. For example, if we have 10 peaks, then the number of close hue will be 70% x 10 = 7. This means only hue that has distance below 7 to the hue reference that will be taken into the back projection. All other hues will be discarded (Figure 1). If no histogram bin exceeds 255, then the hue-distance threshold will be set as 1.

C. Color Masking

Color mask is made for each object part. It is made based on the minimum and maximum values taken from the object localization step. Each pixel in every frame will be evaluated according to those minimum and maximum values. If a pixel values lies within the minimum and maximum values range, then it will be given a value 1. Otherwise, it will be given value 0.

Figure 1 Hue histogram of air plane body (grayish white). Number of peaks = 10. Hue-distance threshold= 7

D. Segmentation

For next frames, mean-shift segmentation is carried out. This will ease the differentiation of the object from the background. Mean-shift will smoothen the image and removes noise while preserving the discontinuity.

We smoothen the frame by mean-shift segmentation starting from second frame. We use spatial range hs=5, color range hr = 20 and maximum pyramid level = 2. With these parameters, noise is reduced significantly while the discontinuity (e.g. edge) is still preserved.

E. Histogram Back Projection

Histogram back projection means evaluating each pixel in the frame sequence based on the histogram model that we have made in the object modeling phase. Before we do back projection, we put the color mask to the frame to pass only pixels that satisfy the object color ranges. We then do histogram back projection to the hue-distance histogram, saturation histogram and value histogram. Each histogram back projection will give a back projection image as the result which contains the probability of each pixel in the frame according to the histograms. We then combine all back projection images into a single back projection image using AND operator. This single back projection image is the input for the tracking.

F. Tracking

Good localization is a first step towards good tracking. But that does not mean we will have 100% accurate tracking. If we choose all the object parts as one part and take the HSV color histogram on it, it will be difficult to track the object if it passes through a background which has color in the range of the object color range.

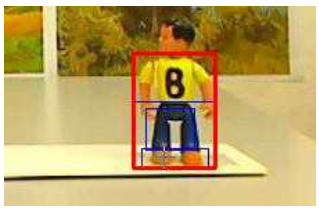


Figure 2 Maximum rectangle (thick red) of body, trouser and shoes' rectangle (thin blue).

One good way to improve the robustness of the tracking is localize the problem of the tracking itself so then it will be

easier to solve. To localize the problem, we propose to track the object parts separately. Object parts represent the dominant color parts of the whole object. Each object part is modeled using hue-distance, saturation and value histogram then we track it. The whole object rectangle is the maximum rectangle of each object rectangle (Figure 2). The whole object center position is the center of the maximum rectangle. Maximum rectangle is defined as the smallest possible rectangle that covers all rectangles inside it[5].

We also propose a mechanism to detect if the tracker lost the object parts and how it deals with the maximum rectangle. If the next tracking rectangle area of an object part is equal to 0, that particular object tracker is stated as lost. If an object tracker is lost, then the rectangle will not be taken into account in the maximum rectangle. The maximum rectangle will only consider the "surviving" tracking rectangles. CAMSHIFT tracking and maximum rectangle calculation are taken from the available functions in OpenCV 2.0 library [14].

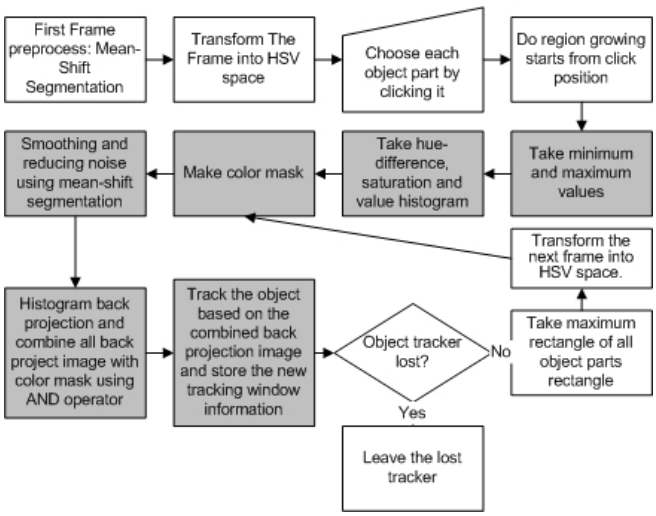


Figure 3 The proposed method schema.

The proposed method can be summarized in the following steps (Figure 3):

1) Do mean-shift segmentation to the image so the object parts will be easier to choose.
2) Transform the frame into HSV space
3) Choose each object by clicking it and do region growing starts from the click position.
4) For the tracking, state each selected object part as separate part.
5) For each dominant color object part (gray boxes in Figure 3):
   a. Take minimum and maximum value of hue, saturation and value.
   b. Calculate hue-difference, saturation and value histogram.
   c. Make color mask by evaluating each pixel in the frame based on the minimum and maximum values taken from step 5.
   d. Do mean-shift segmentation to the image for smoothing the image and reducing noise.
   e. Do histogram back projection from the histogram in step 6 and combine all back projection images.
   f. Track the object based on the combined back projection image and stored the new tracking window information.
6) Find maximum rectangle and leave lost trackers.
7) Transform the next frame into HSV space.
8) Do step 5-7 using new tracking window from step 5f until the end of frame sequence.

For multi-object tracking, we use separate tracker for each object.

V. Results and Discussions

We compare face to face between classic CAMSHIFT results and the proposed method results.

In classic CAMSHIFT, we have to set the parameters manually by manipulating the sliding bar. Those parameters are minimum value (vmin), maximum value (vmax) and minimum saturation (smin). We try to find the best parameter values for each test video and compare with the proposed method's results.

For object localization, the proposed method is simply better than classic CAMSHIFT (Figure 4).

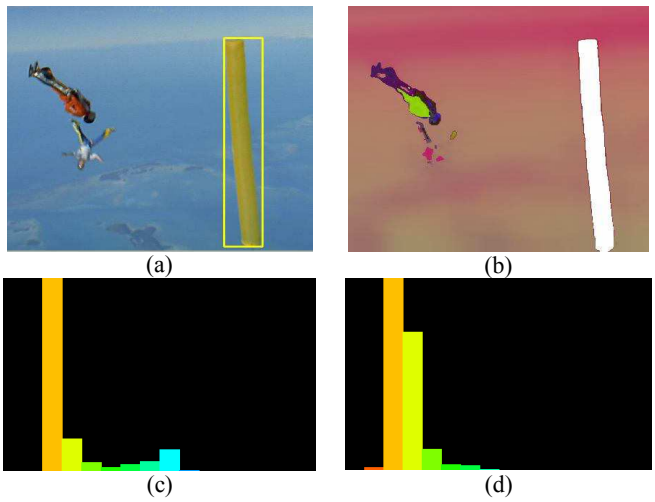


Figure 4 (a) Classic object localization using rectangle (b) Proposed object localization result (c) Hue histogram using rectangle object localization (d) Better Hue histogram using proposed object localization

Table 1 Face to face result comparison between CAMSHIFT (C) and the proposed method (PM). Bd=body, Sh=shoes, Tr=trouser

	1st video		2nd video		3rd video				4th video	
Method	**C**	**PM**	**C**	**PM**	**C**	**PM**			**C**	**PM**
						Bd	**Sh**	**Tr**		
Lost at	145	-	-	-	300	-	855	105	62	-
Total frames	169		401		1061				134	

In the first video, classic CAMSHIFT failed at frame 145 (Figure 5a) because some background information are taken

In the first video, classic CAMSHIFT failed at frame 145 (Figure 5a) because some background information are taken into the object model (Figure 5c). In the second video, both methods can track the object until the end of video but our object marker is more precise (Figure 5b). In the third video, classic CAMSHIFT starts to drift at frame 300 when the object passes the small tree (Figure 5e), while with the proposed method, we can track the body of the doll until the end of video without drifting (Figure 5f). We can track the trouser until frame 105 and shoes until frame 855. This is because trouser is very dark and the shoes are too small when they are too far from the camera. In the forth video, with classic CAMSHIFT, even for single object, it fails to track. In Figure 6a, we try to track a player in orange but then it starts to drift when a player in blue passing through at frame 61 (Figure 6b). In Figure 6c the tracker tracks the player in blue at frame 62. While with the proposed method, we can track several players successfully until the end of the video and does not drift even though the tracked players are almost fully covered by other players with blue or orange jersey. This multi-object tracking capability is for single hue object. The illustration is in Figure 7 (a) and Figure 7 (b). The numerical result comparison is in Table 1.

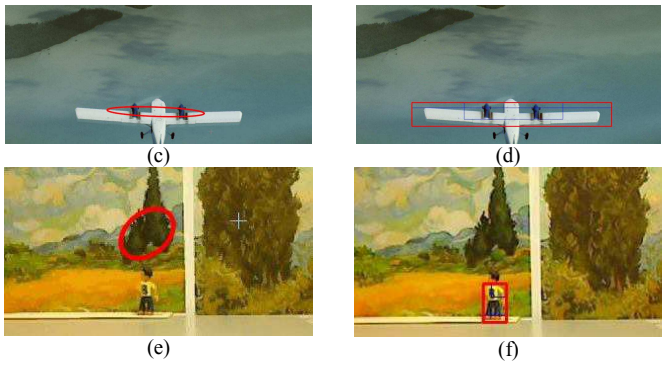
CAMSHIFT achieves real-time performance in every video (25-30 fps) while in the proposed method, we achieve 1-12 fps mainly depending on the resolution of the image on AMD Turion X2 1,6 Ghz and 2GB RAM laptop. This is not a problem yet because our main aim now is to improve the robustness, not the speed.

VI. Conclusions and Future Works

We have developed different ways to improve CAMSHIFT robustness. Firstly, the proposed object localization method improves the robustness of the object model. With it, we can avoid background's information to be taken into the object model so drifting and lost tracker can be avoided. Secondly, tracking dominant color object parts also increases the robustness of CAMSHIFT so it can track multi-hue object in similar color background without drifting. Thirdly, the proposed can improve CAMSHIFT so that it can be used for multi-object tracking.

Our future works will be improving the methods using Graphical Processing Unit (GPU) or parallel programming in multi-core processor to increase the speed so it can achieve real time speed. Beside that, we propose to improve the proposed method so it can re-track the lost object.

(a) (b)



(c) (d)

(e) (f)

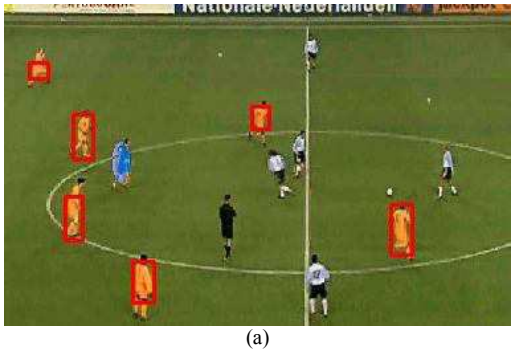
Figure 5 Left: Classic CAMSHIFT result. Right: Proposed method result

(a)

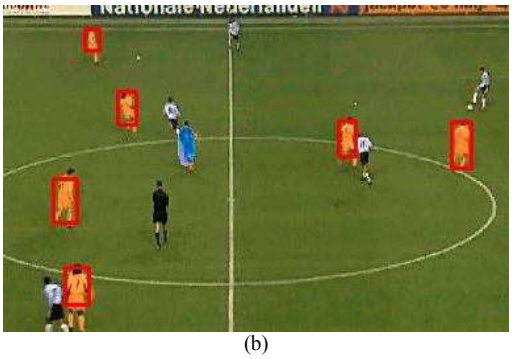
(b)

(c)

Figure 6 (a) The object is tracked at frame 54 (b) The object is almost fully occluded at frame 57 (c)Track wrong object at frame 62



(a)



(b)

Figure 7 Multi-object tracking using the proposed method. (a) Frame 2 (b) Frame 45

REFERENCES

[1] Bradski, G. R. "Computer Vision Face Tracking for Use in a Perceptual User Interface". *Intel Technology Journal,* 2(2), 13-27, 1998

[2] Comaniciu, D. and P. Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(5), 603-619, 2002

[3] J. A. Corrales, P. Gil, F. A. Candelas, F. Torres. "Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System". In *Proceedings Book of 40th International Symposium on Robotics,* Asociación Española de Robótica y Automatización Tecnologías de la Producción – AER-ATP, Barcelona, Spain. pp.107, 2009

[4] Tian, G., Hu, R., Wang, Z., and Fu, Y. "Improved Object Tracking Algorithm Based on New HSV Color Probability Model". In *Proceedings of the 6th international Symposium on Neural Networks: Advances in Neural Networks - Part II,* Wuhan, China, 2009

[5] Bradski, G., and Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008

[6] J. G. Allen, R. Y. D. Xu, and J. S. Jin. "Object tracking using camshift algorithm and multiple quantized feature spaces", in *Proceedings of the Pan-Sydney area workshop on Visual information processing*, ser. ACM International Conference Proceeding Series, vol. 100. Darlinghurst, Australia: Australian Computer Society, Inc., pp. 3–7, 2004

[7] J. Ning, L. Zhang, David Zhang and C. Wu. "Robust Object Tracking using Joint Color-Texture Histogram". *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, No. 7 (2009), World Scientific Publishing Company 1245–1263, 2009

[8] Qiu, X., Liu, S., Liu, F. Kernel-based Target Tracking with Multiple Features Fusion. *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference,* Shanghai, P.R. China, 2009

[9] Ganoun, A., Ould-Dris, N., and Canals, R. "Tracking System Using CAMSHIFT and Feature Points". *14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, 2006

[10] Stolkin, R., I. Florescu, M. Baron, C. Harrier and B. Kocherov. Efficient Visual Servoing with the ABCshift Tracking Algorithm. In: *IEEE International Conference on Robotics and Automation*, pp. 3219-3224, Pasadena, California, USA, 2008

[11] Xu, R Y D; Allen, J & Jin, J S. Robust real-time tracking of non-rigid objects, *Conferences in Research and Practice in Information Technology, VIP'03*, Sydney, Australia, 2003

[12] K. Fukunaga and L.D. Hostetler. "The estimation of the gradient of a density function, with applications in pattern recognition", *IEEE Trans. Inf0rmation Theory,* vol. 21, pp. 32-40, 1975

[13] R. C. Gonzalez, R.E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB 1st Edition*, Dorsing Kindersley, USA, 2004

[14] (2010) OpenCV 2.0 library. [Online]. http://sourceforge.net/projects/opencvlibrary/

[15] S. Stalder, H. Grabner, and L. Van Gool. (2010). [Online].. http://www.vision.ee.ethz.ch/boostingTrackers/contactBoosting.html

[16] R Valenti, F Hageloh. (2010) [Online]. Available: http://student.science.uva.nl/~rvalenti/uva/MIR/movies/soccer.avi