

COMBINATION OF MEAN SHIFT OF COLOUR SIGNATURE AND OPTICAL FLOW FOR TRACKING DURING FOREGROUND AND BACKGROUND OCCLUSION

M. Hedayati, M.J. Cree, J. Scott

University of Waikato, School of Engineering, Hamilton 3240, New Zealand
`mh267@student.waikato.ac.nz`
`{cree,scottj}@waikato.ac.nz`

Abstract. This paper proposes a multiple hypothesis tracking for multiple object tracking with moving camera. The proposed model makes use of the stability of sparse optical flow along with the invariant colour property under size and pose variation, by merging the colour property of objects into optical flow tracking. To evaluate the algorithm five different videos are selected from broadcast horse races where each video represents different challenges that present in object tracking literature. A comparison study of the proposed method, with a colour based mean shift tracking proves the significant improvement in accuracy and stability of object tracking.

Keywords: Object Tracking, MeanShift, Optical Flow, LK, Occlusion

1 Introduction

Object tracking is a common computer vision task and it is the backbone of many applications like video surveillance, robotics, human computer interaction (HCI) and video analysis. The primary task of video tracking is to localize the object of interest in frame $t + 1$ by having its location at time t .

Although for the past decades enormous study was dedicated to object tracking, due to various problems encountered in this area, object tracking still is a challenging topic in computer vision. These challenges are mainly caused by change in size or pose of the object, noise produced at the image acquisition level, variation of light, and partial or full occlusion of the object with the background or foreground [1]. Depending on the characteristics of the scene these challenges need to be considered in building any tracking system.

Colour-based models [2–6] have achieved considerable success in video tracking applications. This success is due to the invariant property of colour under size and pose changes. However using the colour property has two main drawbacks, first the colour is sensitive to illumination changes [7], and second the occlusion of objects with similar colour may lead to incorrect tracking [8]. These shortcomings can be compensated by merging extra information such as edge

features [9, 10] or spatiotemporal motion [11, 13–16] of the object with the colour property.

This paper introduces a model for long-term object tracking by building a probability distribution from corners, motion and colour. The proposed model uses stability of corner features in light variation to extract spatio-temporal motion of the object and stability of colour under pose and size changes to handle various challenges like light variation, occlusion and object pose changes. To have fair evaluation under a highly dynamic environment, the five different videos are selected from broadcast horse races. The selected videos cover almost all the challenges in the object tracking literatures.

The rest of this article is organized as follows: Section 2 is a review of significant literature on object tracking; Section 3 describes our approaches for multiple object tracking; Section 4 compares Meanshift tracking with our proposed model and discusses the final result; and, finally, Section 5 concludes and provides recommendations for further work.

2 Literature Study

Among various tracking systems, Meanshift [2, 3, 7, 10] and optical-flow based [12, 11, 14–17, 19] tracking are two well-known tracking systems that have proven their reliability. The mean shift [18] is a statical method to find local maxima of any kind of probability distribution. The mean shift algorithm was not originally intended for tracking purposes. It was first applied to object tracking by Bradski et al. [2]. Their model calculates the centroid of the 2D colour probability distribution within its 2D window, then moves the window centre to the centroid of distribution. Thus it is called Continuously Adaptive Mean Shift (CAMSHIFT).

Over time various adaptations have been made on CAMSHIFT to improve its accuracy. Comaniciu et al. [6] used the weighted probability distribution in order to assign higher weighting to pixels nearer to the centre of window, based on the assumption that the foreground pixel is more likely selected near to the centre of the tracking window rather than its border. Allen et al. [19] introduced a background-weighted histogram by assigning lower weight to colour features that belong to the background. The background weighted-histogram weights the probability distribution by considering the distribution ratio between background colour (pixels outside tracking window) and the foreground colour (pixels inside tracking window). To handle occlusion Kai She et al. [9] build statistical models based on a hue-saturation-value (HSV) colour map and Haar-like features [20] and apply the Meanshift algorithm on each of these features to localize the object.

Optical flow basically refers to the displacement of intensity patterns and is widely used in computer vision from medical image registration to automated video surveillance [21]. The fundamental optical flow equation is based on the assumption that the pixel intensity does not change due to a small displacement, and is given by,

$$f(x + \Delta x; y + \Delta y; t + \Delta t) \approx f(x; y; t), \quad (1)$$

where $f(x; y; t)$ is the intensity of the image at position (x, y) at time t , $(\Delta x, \Delta y)$ is the change in position, and Δt is the change in time.

The optical flow itself can be divided into a dense and a sparse model. In the dense model the optical field is built based on the motion of all pixels in the image. Horn and Schunk's model [13] is classic dense optical flow which estimate the motion by imposing additional constraints, such as smoothness, to the system. The smoothness constraint is an assumption that the optical flow field should vary smoothly and have few discontinuities. The dense optical flow algorithm has difficulty in calculating flow in homogeneous regions or in edges where only orthogonal displacement can be found. Sparse optical flow solves these issues by only considering points that have strong gradients in both x and y direction. In the literature these points are called *corners*.

The corner detection methods themselves are a broad topic but useful reviews are given by Tuytelaars et al. [22] and Kerr et al. [23].

The Lucas-Kanade (LK) technique [11] is a well-known sparse optical flow method due to its reliable and robust performance. This method solved the optical flow equation (1) by assuming that pixels in the small neighbourhood (patch) have the same displacement. The flow vector at pixel (x, y) is approximated by

$$E_v = \sum_{P \in N} W^2(p) [\nabla I(p)[u, v] + I_t(p)], \quad (2)$$

where $\nabla I(p)$ and $I_t(p)$ represent the spatial and temporal gradient at neighbouring pixels, N is the number of pixels inside the patch, u and v are x and y displacement respectively, and $W(p)$ is the weight parameter associated with neighbouring pixels. If in equation (2) N is bigger than two then u and v can be approximated by least squares solution as:

$$A^T W^2 A V = A^T W^2 b \Rightarrow V = (A^T W^2 A)^{-1} A^T W^2 b, \quad (3)$$

where $A = [\nabla I(p_1), \dots]^T$, $V = [u, v]^T$, $b = -[I_t(p_1), \dots]^T$ and W is the weight matrix.

LK optical flow is widely used in tracking applications. Yin et al. [15] used the LK model to suppress the object tracking problem under camouflage by modelling the motion pattern of the object and the background then object detection is achieved by cluster motion pattern using flow magnitude.

MedainFlow [14] tracking is another LK-based tracker. This model starts by initialising a set of points in the rectangular grid within the object bounding box. These points are then tracked by Lucas-Kanade tracker and the quality of the points is estimated by forward-backward error. The object displacement is approximated by calculating median displacement over remaining points. The forward-backward error first finds the forward trajectory of the object from the first to the last frame, then the backward trajectory is obtained by backward tracking from the last frame to the first one. Finally the two trajectories are

compared and if they differ significantly, the forward trajectory is considered incorrect.

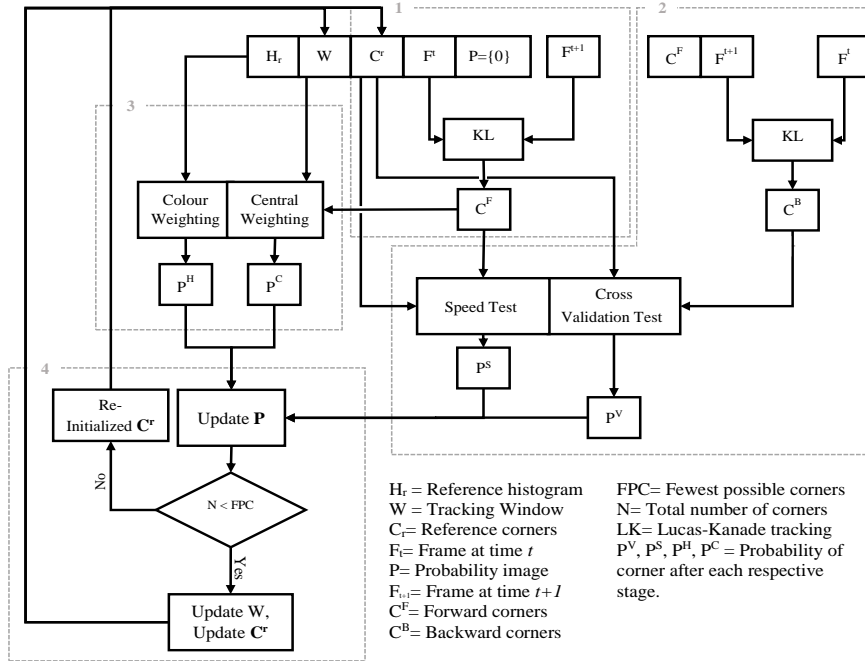
The tracking system proposed by Oshima et al. [17] is the closest approach to our proposed model. They designed a model to track a single object with a static near-infrared camera. They build three different histograms, based on flow magnitude, flow direction and colour. Values of magnitude and direction of each flow vector are estimated on based dense optical flow [13] and then fitted into two separate histograms. Final object localization is achieved by applying Meanshift on a combination of magnitude, flow and colour histograms.

3 The Proposed Model

This paper proposes a multiple hypothesis model for multiple object tracking in video captured with a single moving camera. The proposed model takes advantage of the invariance of object colour under size and pose changes as well as the stability of sparse optical flow to build the probability distribution of the object. As this model used corner, motion, and colour features to build an object model, the proposed model name is CMC tracker.

CMC tracker can be separated into four blocks, namely, pre-localization, filtering, weighing and updating, (See Fig. 1).

Fig. 1. Flow diagram of proposed model. The blocks are identified with numbered grey boxes.



The pre-localization, block 1, roughly estimates the location of the object in the next frame by applying the LK tracker [11] using the set of corner points.

The filtering, block 2, deals with removing the corner points that do not satisfy two constraints. The first constraint is based on the assumption that the speed of corner points are almost constant therefore the corners should not accelerate much. Any corner violating this constraint is removed. The second constraint is based on the forward-backward error [14]. Forward-backward error is used to remove the corner if its forward trajectory significantly differs from its backward trajectory.

The weighting, block 3, adjusts the significance of corners based on colour similarity with the reference object colour and by their distance from the centre of the tracking window. The corners tracked by the LK tracker are not necessarily reliable because they only consider motion of the object, hence these corners can not handle occlusion when two or more objects are moving in the same direction and with similar speed. To handle this situation the colour property of the object merges into the tracked corners, therefore the colour similarity of corners with the reference target colour is measured, and then they are given weights according to their distance to the centre of the tracking window.

The Updating, block 4, applies Meanshift on the probability distribution of image to find the new location of the object and finally update related parameters. The probability distribution image here refers to the corners which passed filtering block and weighted in weighting block. The effect of each of these blocks can be seen in Fig. 2.

To initialize the tracking at the start frame, the rectangular box is placed manually around each object (jockey's upper body). Three sets of features are extracted from these reference targets, as follows:

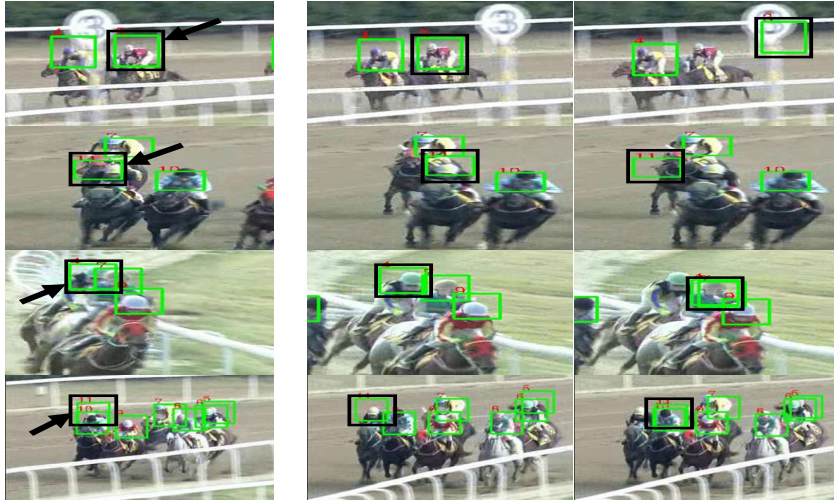
- I. The first feature is the coordinate of the reference target, window (W), in the 2D image plane.
- II. The second feature is the colour distribution of the reference targets or reference histogram. A reference histogram is calculated from the colour distributions of objects in the hue and saturation channel in an HSV colour map and it is represented by $H_r = \{h_n^r\}_{(n=1\dots b)}$ where b is the number of bins.
- III. The LK gives better tracking performance if it operates on stable points like corners, therefore as last feature the well-known Shi and Tomasi [16] method is used to extract corner points from reference target. These points called reference corners are represented by $C^r = \{c_n^r\}_{(n=1\dots N)}$ where N is total number of reference corners.

In brief, the CMC tracker aims to utilize the above information and build a probability image P to feed to the Meanshift tracking. The probability image is a 2D matrix with the same size as the input frame with all of its elements set to zero at the beginning of tracking. The rest of this section gives further detail on each block.

3.1 Pre-Localization

The pre-localization block roughly estimates the location of the object in the next frame by applying the LK tracker [12] on the set of corners points. LK estimates the locations of reference corners in the consecutive frame by calculating their sparse motion field. In the literature these tracked points are called forward corners (C^F). The forward corners alone are not reliable in a highly dynamic environment with various occlusion and pixel intensity changes, therefore a series of assumptions are exploited in the filtering and weighting stage to refine these points as much as possible.

Fig. 2. The effect of filtering and weighting. The arrows in the first column show the notable contenders, the sample result using the proposed model for the subsequent frame is shown in the second column, and third column shows how tracking is lost without the speed test, the cross validation test, and the central and colour weighting from top to bottom.



3.2 Filtering

The filtering block deals with removing the corner points that do not satisfy two assumptions. These two assumptions are called “speed filtering” and “cross validation” testing.

Speed Filtering: After tracking reference corners using the LK model the speed of the object can be estimated by taking the median value from the displacement vector. The displacement vector itself is a vector that represents the distance value between each pairs of points (4).

$$S^t = \{ \| c_n^F - c_n^r \| \}_{(n=1 \dots N)} \quad (4)$$

where S^t is the displacement vector at time t , and c_n^F and c_n^r are forward and reference corners respectively .

Having estimated the speed of the object at time $t - 1$ and displacement vector at time t the speed errors E^s can be calculated by

$$E^s = \{ \| S_n^t - \text{Md}(S^{t-1}) \| \}_{(n=1\dots N)}, \quad (5)$$

and a the new probability is assigned to each corner by

$$P^S(C^F) = \begin{cases} 0 & \text{for } E_n^s > \text{Md}(E^s) + \alpha \\ 1 & \text{for } E_n^s \leq \text{Md}(E^s) + \alpha \end{cases} \quad (6)$$

where $P^S(C^F)$ indicates the speed filtering weight function, α is the margin of error and $\text{Md}(E^s)$ is the median function.

Cross Validation Testing: To increase the accuracy of optical flow tracking the forward-backward error [14] is calculated for two consecutive frames. In this process, reference corners are tracked from frame at time t to frame at $t + 1$ (forward test). Next the tracked corners are tracked backward to the frame at time t (backward test). Then a distance error E^V can be calculated for each reference corner using

$$E^V = \{ \| c_n^r - c_n^B \| \}_{(n=1\dots N)}, \quad (7)$$

where E^V is forward-backward error vector. Eventually the forward corners are given the weights by

$$P^V(C^F) = \begin{cases} 0 & \text{for } E_n^V > \epsilon \\ 1 & \text{for } E_n^V \leq \epsilon, \end{cases} \quad (8)$$

where $P^V(C^F)$ is cross validation testing function and ϵ is the maximum distance error.

3.3 Weighting

The cross validation and speed filtering simply eliminate the points that are likely to be in error considering some margin of error. Typically these corners do not handle occlusion well when two or more objects are moving in the same direction and with similar speed. These corners are weighted by the distance of the corners to the centre of the tracking window (central weighting), and by the colour similarity between surrounding patch around each corner and the reference histogram (colour weighting).

Central Weighting: Normally the foreground region is more likely to be at the centre of the window, therefore the corner point near the window's centre should be weighted by a higher value. The distance r of the forward corner from the window centre,

$$r = \{ \| W_{center} - c_n^F \| \}_{(n=1\dots N)} \quad (9)$$

is calculated and the corner is given the central weight,

$$P^C(C^F) = \begin{cases} 0 & \text{for } r_n > W_{radius} \\ \frac{1}{r_n} & \text{for } r_n \leq W_{radius}. \end{cases} \quad (10)$$

Colour Weighting: The final step to build the probability distribution of the object is the colour similarity estimation. This process differs from the traditional histogram matching, which compares the entire tracked window with the reference histogram. The histograms of 7 by 7 patches around forward corners are calculated and compared with the reference histogram using the Bhattacharyya distance [24].

Lower values of Bhattacharyya distance indicate higher similarity, a perfect match is indicated by zero and a total mismatch by one. Thus the inverse value of Bhattacharyya distance is used to weight the corners (11)

$$P^H(C^F) = 1 - \sqrt{1 - \sum_i \frac{\sqrt{H(c_n^F)(i) \cdot H_r(i)}}{\sqrt{\sum_i H(c_n^F)(i) \cdot H_r(i)}}} \quad (11)$$

where P^H indicate the probability of forward corners after colour matching, $H(c_n^F)$ is the histogram of patches around each corners and i is the bin number.

3.4 Updating

In the updating block all weighted corners are merged together to form a probability image by (12)

$$P = \begin{cases} 0 & \text{if } P_n^{FB} = 0 \vee P_n^S = 0 \vee P_n^C = 0 \vee P_n^H = 0 \\ P_n^{FB} \times P_n^S \times P_n^C \times P_n^H & \text{Otherwise,} \end{cases} \quad (12)$$

and the reference corners are updated by

$$C^r = \{C_n^F \mid P_n > 0, 0 < n < N\}. \quad (13)$$

Eventually, to locate the new position of window the probability image (P) is fed to a Meanshift. Note if the total number of reference corners is below some pre-defined number, the forward corners can not represent the object well and can result in mistracking. If the model reaches this critical point, the reference corners (C^r) are reinitialized and the process is repeated from the filtering stage. This number is called the fewest possible corners(FPC).

4 Evaluation Methodology

The tracking algorithm was implemented using a C++-based computer vision library (OPENCV) on an Intel (R) core (TM) i7- 4770 @ 3.4 GHz CPU with 16 GB RAM.

In order to detail assessment of the proposed model five challenging videos of horse races are selected and the tracking results are compared with the Meanshift model proposed by Allen et al. [19]. Each video highlights different challenges that are faced in object tracking literature. These challenges can be identified as light variation (C1), partial occlusion with the background (C2) or the foreground (C3), full occlusion with the background (C4) or the foreground (C5), camera zoom in/zoom out (C6) and angle changes of camera (C7). The properties of the test videos are listed in table 1. The parameters used in the proposed model are tabulated in table 2.

Table 1. Property of sample videos






Video ID	Duration (Secound)	Frame Size(Pixel)	Frame Rate(fps)	Challenges	Sample Frame
V1	8s	800 × 450	28	C1, C3	
V2	10s	800 × 450	28	C1, C3	
V3	13s	800 × 450	28	C1, C2 C3, C6	
V4	31s	800 × 450	28	C1, C3 C6, C7	
V5	50s	640 × 480	25	C1, C2, C3 C4, C5, C6 C7	

Table 2. The parameter used for the implementation

No.Hue Bin	No.Sturation Bin	Patch Size	ϵ	α	W	FPC
128	128	7 × 7	5	5	40 × 40	15

Table 3. Number of total contenders actually present in the frame at the test point.

Video ID	T1	T2	T3	T4
V1	12	12	12	9
V2	12	12	12	12
V3	12	12	12	5
V4	12	12	10	9
V5	9	9	9	9

To estimate the accuracy of the tracking model, the contenders (i.e. jockeys) at the first quarter (T1), half way (T2), third-quarter (T3) and at the end (T4) of each video were manually selected as ground truth. The contender tracking is considered successful if the centre of the tracking window lays inside the ground truth window. The performance of a tracking algorithm is then measured by calculating ratio of successful tracked contenders to the total number of contenders, and is called the percentage of correct tracking (PCT). The number of total contenders actually present in the frame at the test point for five samples shows in table (3).

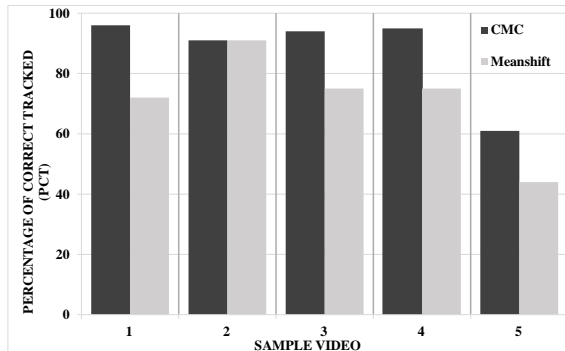
5 Result and Discussion

Figure 3 shows the mean PCT for five sample videos and table 4 tabulates the tracking accuracy based on PCT at the test points. The overall PCT (Fig.3) clearly indicate that the proposed model improves the tracking accuracy compared to the colour-based Meanshift tracker.

By observing the PCT at each interval (table 4) it can be observed that Meanshift accuracy significantly drops with respect to time. One main reason behind this drop is change in illumination which cause a shift in significant colours of the object.

The light variation is unpredictable, therefore under light changes the behaviour of Meanshift tracking is also unpredictable. This instability can be clearly seen in V1 and V2 where both cases suffer from same challenges but one result (V2) is comparable to CMC tracker while the other one (V1) is notably lower than CMC.

Fig. 3. Mean percentage correct tracking of contenders in the horse races for five sample videos



The partial occlusion is one common problem in all of the video samples. Meanshift tracking has difficulty in handling partial occlusion especially where occlusion happens between two objects with similar colour. The CMC is more robust to occlusion because, first, CMC only uses the colour property around

stable corners for the colour matching, therefore it makes objects more recognisable under foreground occlusion, and second CMC estimates the object speed using optical flow, hence it can overcome background occlusion by estimating relative speed of the object with respect to its background (sample V3).

As it can be observed from sample V5 (table 4) the PCTs of the proposed model is better than Meanshift at the first three intervals, however the Meanshift gives better tracking at the last interval. The reason behind this problem lays in the characteristic of CMC and the challenges in video V5.

Video V5 suffers from one main problem which is full occlusion with other contenders, therefore as contenders fully cover one another, all reference corners are swept from the target and stick to the obstacle. As a result the reference corners are no longer there to represent the desired target and cause false tracking.

Table 4. Results of percentage of correct tracking for both Meanshift and CMC at the test points.

Video ID	T1		T2		T3		T4	
	CMC	Mean shift	CMC	Mean shift	CMC	Mean shift	CMC	Mean shift
V1	100	91	91	75	91	66	100	55
V2	91	91	91	91	91	91	91	91
V3	100	83	91	91	83	66	100	60
V4	100	91	100	81	90	70	88	55
V4	77	55	77	44	55	33	33	44

6 Conclusion

This paper proposed a novel model for multiple object tracking under highly dynamic environment. The tracking accuracy of the model is evaluated by five broadcast videos in four different intervals. The statistical result shows this model improves the accuracy of tracking under occlusion, light variation, size and pose changes by overall PCT average of 27% respect to colour based Mean-shift tracking.

References

1. Maggio,E., Cavallaro,A.: Video Tracking: Theory and Practice. Wiley Publishing (2011)
2. Bradski, G. R.: Real time face and object tracking as a component of a perceptual user interface. In 4th IEEE Workshop on Applications of Computer Vision, pp. 214–219. IEEE Computer Society, Washington (1998)
3. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In the 7th European Conference on Computer Vision-Part I. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)

4. Nummiaro, K., Koller-Meier, E., Van Gool, L.: Object Tracking with an Adaptive Color-Based Particle Filter. In *The 24th DAGM Symposium on Pattern Recognition*, pp. 353–360. Springer-Verlag, London (2002)
5. Isard, M., and Blake, A.: Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*. 29(1), 528 (1998)
6. Comaniciu, D., Visvanathan R., and Meer P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 564-577 (2003)
7. Hayashi, Y., Fujiyoshi, H.: Mean-shift-based color tracking in illuminance change. In *Robot Soccer World Cup XI*, pp 302–311. Springer, Heidelberg (2008)
8. Chandel H., Vatta S.: Occlusion Detection and Handling: A Review. *International Journal of computer application*. 120(10), 33–38 (2015)
9. She, K., Bebis, G., Gu, H., Miller, R.: Vehicle tracking using on-line fusion of color and shape features. In *7th International Conference on Intelligent Transportation Systems*, pp. 731–736. IEEE (2004)
10. Zhou, H., Yuan, Y., Shi, C.: Object tracking using SIFT features and mean shift. *Computer vision and image understanding*, 113(3), 345–352 (2009)
11. Lucas B.D., Kanade T.: An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence*, pp. 674–679, Vancouver (1981)
12. J. Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation, (5), 1–10 (2001)
13. Horn B., Schunck B.: Determining optical flow. *Artificial Intelligence*, 17, 185203 (1981)
14. Kalal Z., Mikolajczyk K., Jiri Matas J.: Forward-Backward Error: Automatic Detection of Tracking Failures. In *20th International Conference on Pattern Recognition*, pp. 2756–2759. IEEE Computer Society, Washington (2010)
15. Hou j. Y. Y. H. W., Li J.: Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*. 15, pp. 2201-2205 (2011)
16. Shi and Carlo Tomasi C.: Good Features to Track. Technical Report. Cornell University, Ithaca (1993)
17. Oshima N., Saitoh T., Konishi R.: Real time mean shift tracking using optical flow distribution. In *Joint Conference on SICE-ICASE*. pp. 4316-4320. IEEE (2006)
18. Cheng Y.: Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17(8), pp. 790-799 (1995)
19. Allen J. G., Xu R. Y., Jin, J. S.: Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*. pp. 3–7. Australian Computer Society, Inc., Darlinghurst (2004)
20. Viola P., Jones M.: Robust real-time object detection. *International Journal of Computer Vision*, 4, pp. 51-52 (2001)
21. Fortun D., Bouthemy, P. Kervrann, C.: Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*. 134, pp. 1-21 (2015)
22. Tuytelaars T., Mikolajczyk K.: Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*. 3(3), 177-280 (2008)
23. Kerr D., Coleman S., Scotney B.: Comparing Cornerness Measures for Interest Point Detection. In *8th Machine International conference on Vision and Image Processing*. pp.105–110, IEEE, Portrush (2008)
24. Kailath T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Comm. Technology*. 15, pp. 52–60 (1967)