

Color Particle Filter Tracking using Frame Segmentation based on JND Color and Spatial Connectivity 4-D Histogram

Aniket V Gokhale¹, Kishor K Bhoyar², Kishor M Bhurchandi³

¹. Research Scholar, Department of Electronics Engineering, Y.C.C.E., Nagpur,

². Professor, Department of Information Technology, Y.C.C.E., Nagpur

³. Professor, Department of Electronics and Communication Engineering, V.N.I.T., Nagpur

Abstract: Object tracking is one of the most important components in numerous applications of computer vision. Color can provide an efficient visual feature for tracking non-rigid objects in real-time. The color is chosen as tracking feature to make the process scale and rotation invariant. The color of an object can vary over time due to variations in the illumination conditions, the visual angle and the camera parameters. This paper presents the integration of color distributions into particle filtering. The color feature is extracted using our novel 4D color histogram of the image, which is determined using JND color similarity threshold and connectivity of the neighboring pixels. Particle filter tracks several hypotheses simultaneously and weighs them according to their similarity to the target model. The popular Bhattacharyya coefficient is used as similarity measure between two color distributions. The tracking results are compared on the basis of precision over the data set of video sequences from the website <http://visualtracking.net> of CVPR13 bench marking paper. The proposed tracker yields better precision values as compared to previous reported results.

Keywords: Frame segmentation, object detection, object tracking, particle filter, 4-d histogram, spatial connectivity

I. Introduction

Object tracking is an important task in many application areas like automated surveillance, human computer interfacing, vehicle navigation, traffic control, etc. [1, 2,3,5,6]. There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. [1,2,3,5,6,8]. Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. A tracker assigns consistent labels to the tracked objects in different frames of a video. A tracker can also provide object-centric information, such as orientation, area, or shape of an object. Two basic approaches for object tracking are deterministic method and stochastic method. The deterministic methods are based on target representation and localization. The stochastic methods are based on filtering and data association techniques. Various features like color, edges, optical flow, texture, etc. can be considered for object tracking. [1,2,11,12].

In probabilistic approach state space model is used and the tracking process is considered as state estimation problem. Basically, Kalman filter is used as a traditional approach of tracking. It provides nice tracking results for linear systems with Gaussian noise. In nonlinear systems, the nonlinear function is linearized over small region around predicted value. This approach is known as extended Kalman filter and utilized to solve the problem of nonlinear tracking. The unscented Kalman filter can approximate second order Taylor series expansion whereas extended Kalman filter can achieve only first order Taylor series expansion. Thus unscented Kalman filter have better performance in estimation using Gaussian distribution but still it cannot deal with multimodal noise distribution [3,4,5,6,8,11]. Particle filter [1,4,5,28,29] is a probabilistic method for object tracking based on posterior probability density. Particle filter is a point tracking method that has been proven successful for nonlinear and non-Gaussian estimation [4,5]. It approximates the posterior probability density of the state such as the object position by using samples which are called particles. Particle filtering uses sequential importance sampling. It estimates the posterior distribution of target state using set of weighted particles. The weight of the particle represents the likelihood of particle properties with the target properties [12]. A single color or local color distribution can be used as an observable feature for deciding weight of particle [13]. In conventional particle filters like condensation algorithm [4,5], the latest observation from the current frame in the image sequence is only used in the weighting step and not in the sampling step. As a result, a large number of particles are often required to approximate the posterior probability density properly. This limitation is overcome in particle filter algorithm which requires less number of particles and thus computations get reduced [14]. The process of object tracking involves object detection and tracking the detected object. Section II describes object detection in brief. Section III describes the process of particle filtering. The experimental results and conclusions are given in sections IV and V respectively.

II. Object Detection

Object detection can be done in every frame or when the object first appears in the video. Detecting object in every frame leads to increased accuracy but it also increases the computational cost as well as time consumed for tracking process. Thus, here object is detected only once in the video frame. Detection of object can be performed by using various methods, which include point detectors, background subtraction, segmentation techniques, supervised learning, etc. [1,2,4,5,11,15]. Segmentation techniques are widely used for the purpose of object detection. Segmentation of image refers to identification of homogeneous regions in the image. Pixel based segmentation techniques include thresholding, clustering and fuzzy clustering. Region based segmentation techniques include region growing, region merging, etc. while the edge based techniques can also be used for segmentation. Hybrid methods combine these techniques for more realistic results.

The algorithm presented here is an extension of the algorithm presented in [23]. This extended algorithm considers spatial connectivity with neighboring pixels to produce homogeneous regions of similar colors. The 4D color histogram of the image is determined using JND (Just Noticeable Difference) color similarity threshold and connectivity of the neighboring pixels, by comparing current pixel with the previously encountered immediate 8-neighbor pixels. Initial segments are then merged using a slightly higher JND threshold by applying concept of agglomeration. In JND color histogram, color corresponding to each bin is visually dissimilar from that of any other bin; whereas each bin contains visually similar colors [21,22,23,24,25,30]. The color similarity mechanism is based on the threshold of similarity which uses Euclidean distance between two colors being compared for similarity. A color spatial segmentation scheme based on spatial color JND histogram using color similarity and spatial connectivity uses additional spatial information while forming the histogram based on comparison with previously encountered neighboring pixels [22,23,30].

For preparing the 4D color histogram, the histogram is defined in two data structures $N \times 4$ and $(m \times n) \times 3$. The first data structure Table 1 consists of N rows corresponding to the color shades detected and first three columns correspond to R, G and B as color tri-stimulus corresponding to pixel color and 4th column represents the population of that color. Second data structure Table 2 has $(m \times n)$ rows corresponding to the total pixels of the image and first two columns corresponding to spatial location of the pixels in terms of row and column and third column represents the color index corresponding to the row number in Table 1. The pixels in the color image are scanned from top pixel to bottom last pixel row and column wise. The color of pixels are compared on the basis of the Euclidean distance between two pixel colors within the threshold of 3-JND (JND_{eye}) for similarity for merging while formation of the histogram tables.

Table 1 Cumulative color population

Color Index(k)	Color			Population
	R	G	B	H
1	5	15	20	335
2	10	100	20	450
3	20	50	10	470
.
K	25	72	90	200
.

Table 2 Color index-pixel location

Color index For(x_i, y_i)	Spatial Coordinates	
	X	Y
1	x_1	y_1
1	x_1	y_2
.	.	.
K	x_i	y_i
.	.	.
.	.	.

Algorithm for Computing the Spatial Color JND Histogram By comparing only previously encountered immediate neighbors [30]

- i) Initialize two data structures Table 1 and Table 2. Initialize the first entry in Table 1 by the first color vector in the image i.e. top left pixel color vector [R,G,B] and the frequency(population) by one. Initialize the first entry in Table 2 by the current row index value of Table 1 i.e. 1, and the top left pixel position row and column i.e. $y(\text{column})=1$ and $x(\text{row})=1$. Also initialize a (row, column) pointer to top left corner of the image. Select a similarity threshold JND_{eye} approximated to 300 (285.27) depending on the precision of vision from fine to broad as required by the application.
- ii) Read the next pixel color vector in scan line order.
- iii) Compare the next pixel in scan line with previously encountered pixel using a neighbourhood tile of 3×3 as shown in figure 1 below i.e. pixel C is compared with only pixel P.

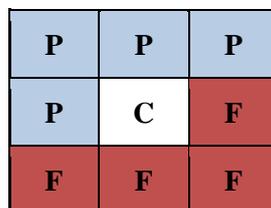


Figure 1. The 3×3 tile for color comparison of current & immediate previous neighbours

The center Pixel ‘C’ is the current pixel in the scan line, pixels ‘P’ are the previously encountered pixels in the scan line and pixel ‘F’ are future non encountered pixels in the scan line. The current pixel C is compared with the pixels P for similarity based on JND threshold using Euclidian distance between the tri-stimulus color vectors of the pixels. If color found similar include the pixel in respective bin. Update the entry in Table 1 by increasing population of respective color bin and update entry in Table 2 by entering row number of the Table 1 updated bin as color index and row and column numbers of C for spatial location of the pixel and go to step v.

- iv) If the new color vector is not equal to any of the previously recorded color vectors in Table 1, increment the row index of Table 1, enter the new color vector in it, set the population to 1, make the index, row and column entry in Table 2 and go to step ii.
- v) Repeat step ii) to iv) for all the pixels in the image in scan line order.
- vi) Sort Table 2 in the increasing order of the color index..
- vii) Find the similar colors in the table 1 using similarity threshold Θ more than JND_{eye} threshold, we have choose it to be of human perception i.e. JND_h and mark them with additional entry in column 5 of table 1 as similarity color index (S). The similarity threshold Θ used here is in squared form to eliminate the square root calculations in Euclidian distance for color similarity calculations i.e. $\Theta = JND_h^2 = 2400 (2567)$.
- viii) This will give us the maximum dissimilar colors available in the image as perceived by humans.
- ix) Perform merging of the color bins with same similarity color index (S) in Table 1.

This algorithm ensures formation of the spatial 4D color histogram having color bins produced using JND color similarity as well as the spatial connectivity. Segmentation procedure is straightforward with the data structures given in Table 1 and Table 2. In Table 2, the pixel entries are sorted spatially from left to right and from top to bottom. Segmented image can simply be formed by assigning to each pixel position a JND color from Table 1 as pointed to by the respective index in Table 2. One of the clusters from the clusters formed by segmentation by above algorithm is selected as object region. The parameters of the selected cluster like minimum and maximum X and Y coordinates, coordinates of centroid and mean color component values for the cluster are used as input parameters for tracking process.

III. Particle Filter Algorithm

Tracking process is performed by using color based particle filter algorithm which is a probabilistic approach used for state estimation. Tracking process is generally affected by rotation and scale variance. Here color is chosen as tracking feature to make the process scale and rotation invariant [15,17,18,19,27,29]. We want to apply such a particle filter in a color-based context. To achieve robustness against non-rigidity, rotation and partial occlusion we focus on color distributions as target models. These are represented by histograms which are typically calculated in the RGB space using our novel 4-d spatial color histogram based segmentation [30].

The color distribution $p_y = \{p_y^{(u)}\}_{u=1\dots m}$ of a region R at location y is calculated as

$$p_y^{(u)} = f \sum_{xi \in R} k \left(\frac{\|y - xi\|}{a} \right) \delta[h(xi) - u] \tag{1}$$

where δ is the Kronecker delta function and $h(xi)$ assigns one of the m -bins of the histogram to a given color at location xi . The variable a provides invariance against scaling of the region and the normalization factor f ensures that $\sum_{u=1}^m p_y^{(u)} = 1$.

In a tracking approach the estimated state is updated at each time step by incorporating the new observations. Therefore, a similarity measure is needed between the color distributions of a region in the newly observed image and the target model. A popular measure between two distributions is the Bhattacharyya coefficient [1, 7,11,12,15,16,29]. Considering discrete densities such as two color histograms $p = \{p^{(u)}\}_{u=1\dots m}$ and $q = \{q^{(u)}\}_{u=1\dots m}$ the coefficient is defined as

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p^{(u)}q^{(u)}} \tag{2}$$

Larger the value of ρ is, the more similar the distributions are. For two identical histograms we obtain $\rho = 1$, indicating a perfect match. As distance between two distributions we define the measure d

$$d = \sqrt{1 - \rho[p, q]} \tag{3}$$

which is called the Bhattacharyya distance coefficient [1, 7,11,12,15,16].

The color particle filter algorithm consists of the following steps:

A. Initialization

All the pixels in the detected object region are considered as initial particles. Particle filter algorithm uses state transition model. Thus initial state S can be defined as

$$S = [X \ Z]^T \tag{4}$$

where X represents the state vector and Z represents the observation vector. State of the particle indicates the position of the particle in terms of coordinates. The propagation of particles is achieved using the state space model [1,4,5,9,10,20].

B. State update

In this step, new state is set for each particle using proposal distribution. The particle state is updated by using following equations:

$$S_k = F_{update} * S_{k-1} \tag{5}$$

$$X_k = X_k + \sigma_{pos} * randn(2, N) \tag{6}$$

$$Z_k = Z_k + \sigma_{vec} * randn(2, N) \tag{7}$$

Where, σ_{pos} and σ_{vec} are the standard deviations for noise distribution and value of F_{update} [11,28] used is :

$$F_{update} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C. Weighing

The important step in particle filter algorithm is weighing in which each particle is assigned a weight according to value of likelihood function for that particle. For color based particle filter the color similarity of the particle with the initially selected object region is used as measure to compute likelihood. The likelihood distance (π) for the updated particle is calculated using equation (8) [1,4,5,14,15,20].

$$\pi = \frac{1}{\sqrt{2 * pi * \sigma}} e^{(-\frac{d^2}{2\sigma^2})} \tag{8}$$

If the updated position of the particle lies outside the video frame boundaries, lowest possible value ($-\infty$) is assigned to it as the likelihood distance. After the likelihood computation, weights (w_t^i) are assigned to each particle. Larger weight or value is assigned to the particle having larger value of the likelihood function and vice versa. During filtering, samples with a high weight may be chosen several times, leading to identical copies, while others with relatively low weights may not be chosen at all.

D. Resampling

When the above algorithm is used for tracking, some low weight particles are also generated. These low weight particles affect the accuracy of the tracking algorithm. This is known as particle degeneracy [8,19,20]. To overcome this problem, resampling process is used for generating new set of particles while removing the low weight particles. The effective sample size (N_{eff}) is used as a measure to determine particle degeneracy.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_t^i)^2} \tag{9}$$

If the effective sample size is less than the threshold sample size (N_{th}), degeneracy is said to have occurred and resampling stage is activated. During resampling step, only the particles with lowest weights i.e. the particle with weight less than the threshold weight, are eliminated and replaced with new set of particles while other particles are stored for further processing [19,20]. This reduces the computational cost and the processing time required for resampling.

IV. Results

The experimental results are obtained on many video sequences in avi and mp4 format. The algorithm has been implemented in Matlab. The algorithm uses 4-d histogram based color spatial segmentation method for object detection and color particle filter for tracking. This work mainly focuses on the tracking of single target. The annotated dataset available on the website <http://visualtracking.net> of CVPR13 benchmarking paper [26] has been used for testing the results. The dataset consists of 50 fully annotated sequences to facilitate tracking

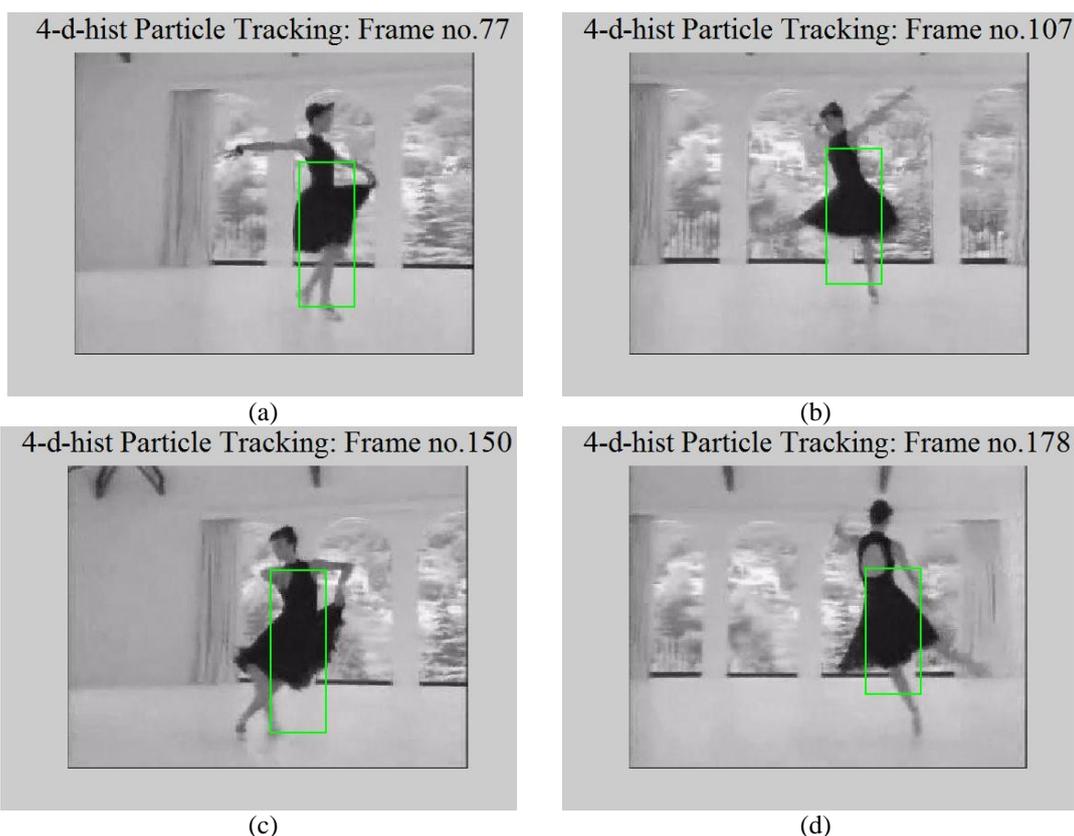
evaluation under different conditions like Illumination Variation, Scale Variation, Occlusion, Background Clutters etc. Object to be tracked is selected by the user and only this selected region of object and its surrounding is segmented to obtain the final object region. As segmentation is not performed over complete video frame, the computation time is reduced. In this work, we use the precision value as a parameter for quantitative analysis. Precision plot is a widely used evaluation metric for tracking. Precision is the center location error, which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labelled ground truths. Then the average centre location error over all the frames of one sequence is used to summarize the overall performance for that sequence. However, when the tracker loses the target, the output location can be random and the average error value may not measure the tracking performance correctly [26]. Recently the precision plot [26] has been adopted to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. As the representative precision score for each tracker we use the score for the threshold is equal to 20 pixels [26].

Figure 2 (a - f) shows different output frames of the video sequence *Dancer*. It can be seen clearly that the object to be tracked is changing shape, size and is subjected to in plane rotation then also the tracker is holding the object in the frames. Figure 2(g) shows the precision curve for the tracker output which shows the precision value of >50% for the 20 pixel location threshold.

Figure 3 (a - f) shows different output frames of the video sequence *Human2*. It can be seen clearly that the object to be tracked is changing shape, size and is subjected to in plane rotation and also illumination variation then also the tracker is holding the object in the frames. Figure 3(g) shows the precision curve for the tracker output which shows the precision value of just <50% for the 20 pixel location threshold.

Figure 4 (a - f) shows different output frames of the video sequence *Skater2*. It can be seen clearly that the object to be tracked is changing shape, size and is subjected to in plane rotation & background clutter then also the tracker is holding the object in the frames. Figure 4(g) shows the precision curve for the tracker output which shows the precision value of >75% for the 20 pixel location threshold.

The precision error is calculated for all the videos tested using the tracker and from this the average precision error is calculated. The average precision plot indicates the percentage of successfully tracked frames of all the video sequences against the location threshold in pixels between the tracker output and the ground truth. The precision plot obtained using our tracker refer Figure 5(b) is showing improvement over previous trackers [26] in terms of percentage successful frames at location threshold of 20 pixels. The value at the threshold is 66.6% which is better than earlier reported values which are given in the legend of Figure 5(a) plot [26] and the precision comparison as given in Table 3.



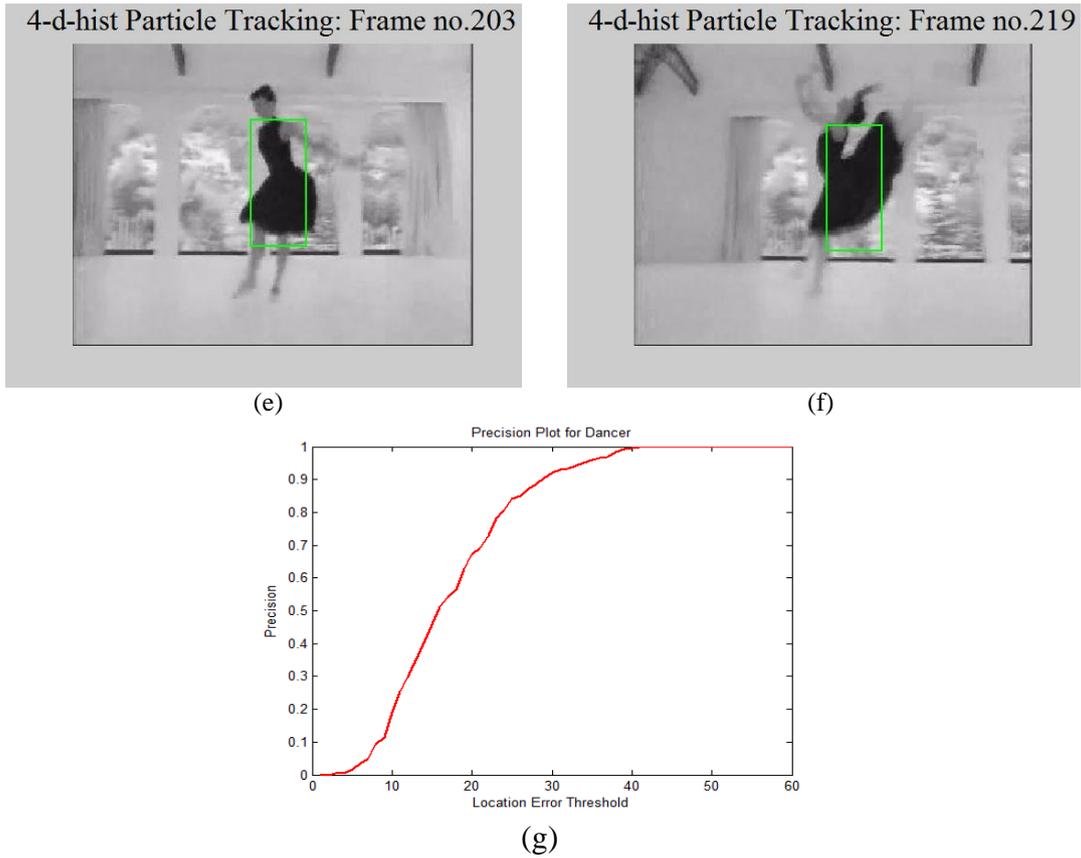
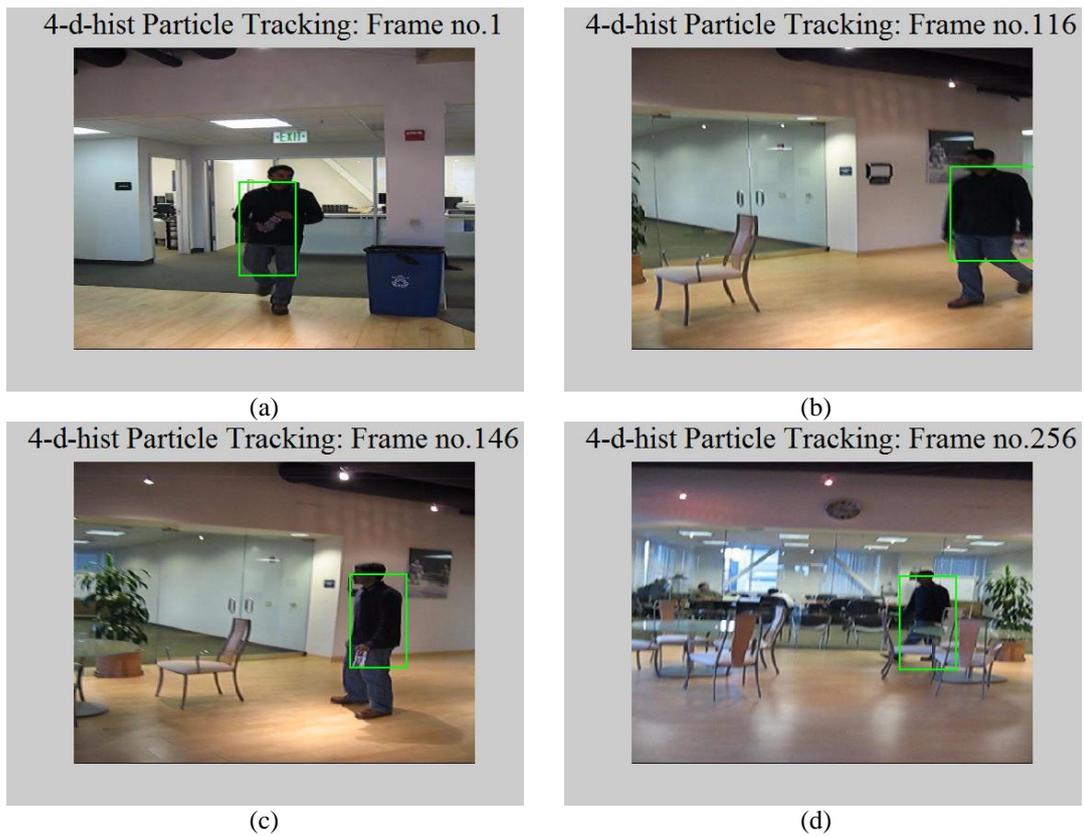


Figure 2 (a-f) The output frames of video sequence Dancer showing tracked object, (g) Precision curve



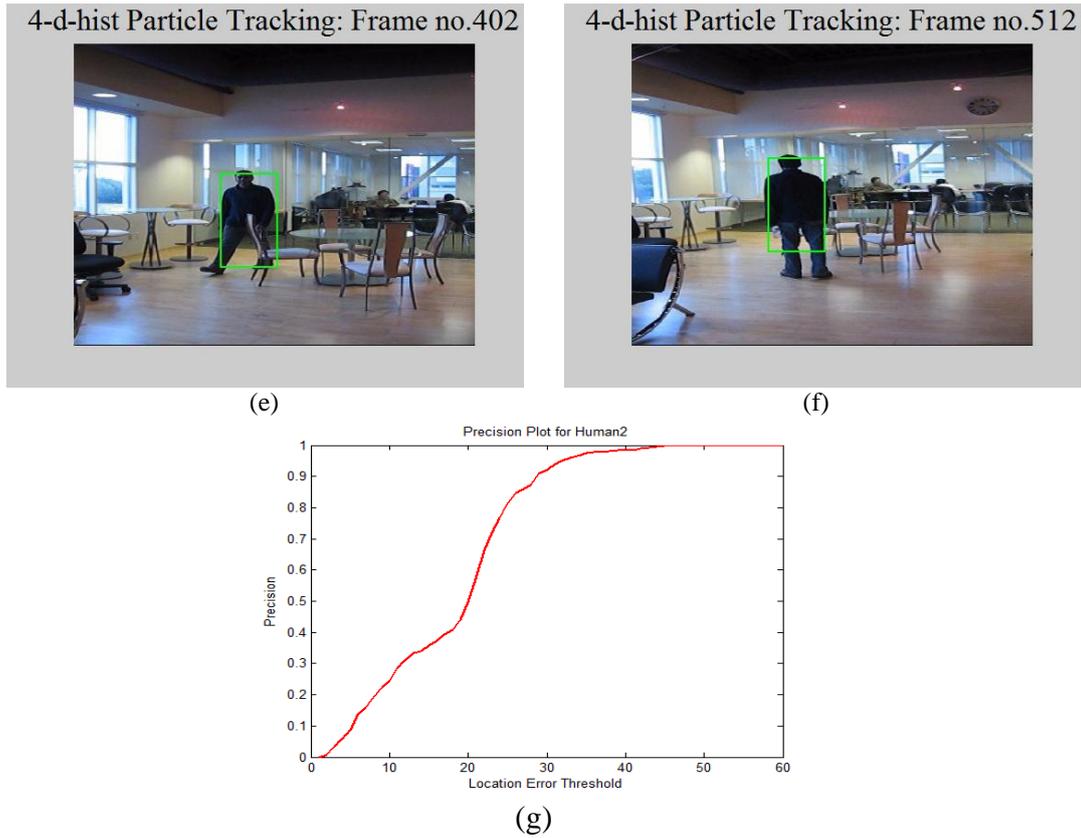
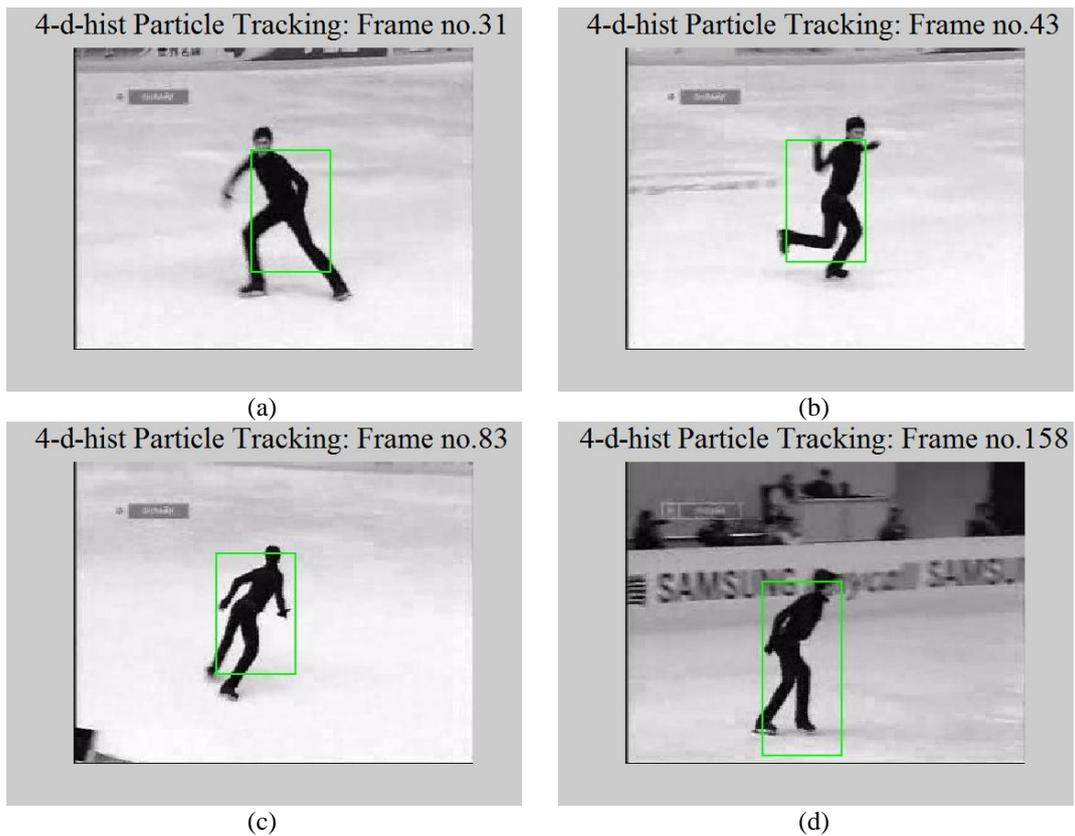


Figure 3 (a-f) The output frames of video sequence Human2 showing tracked object, (g) Precision curve



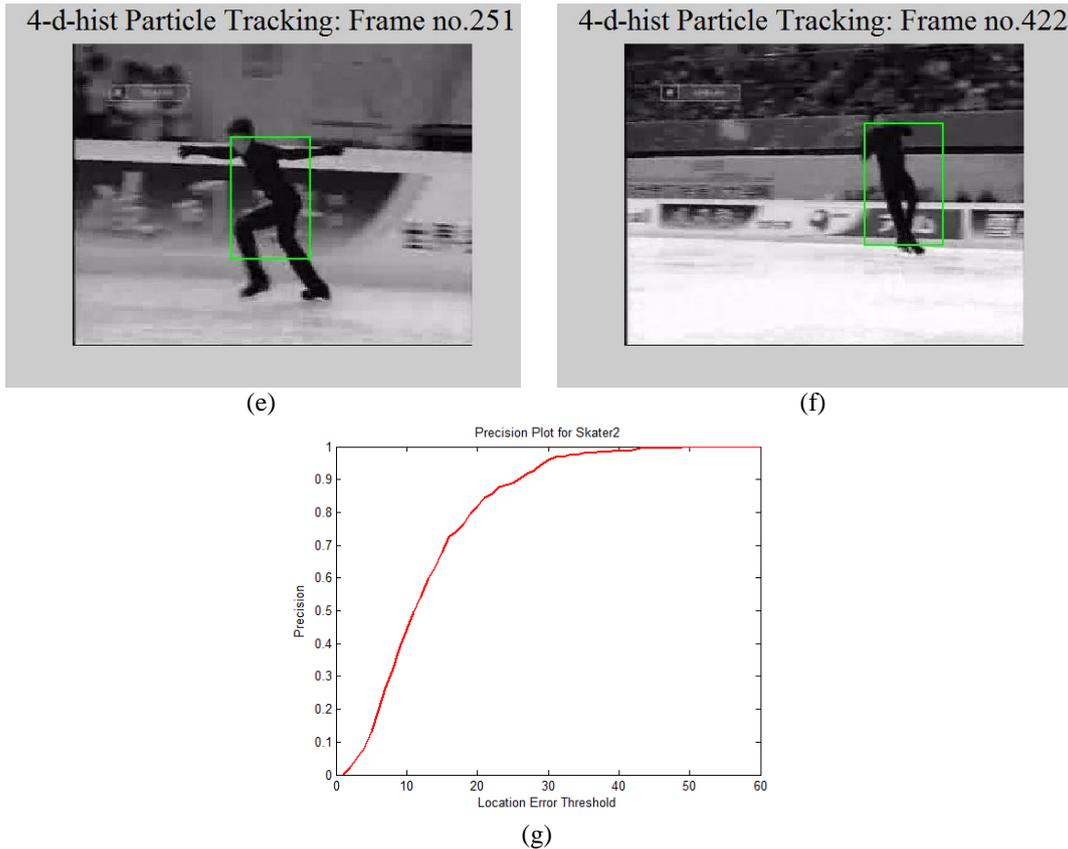
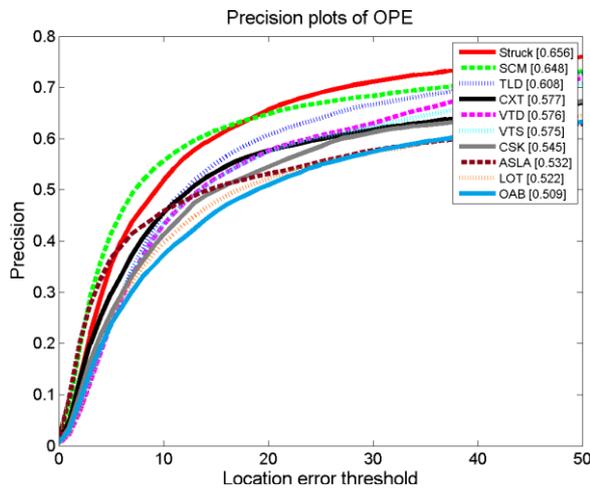


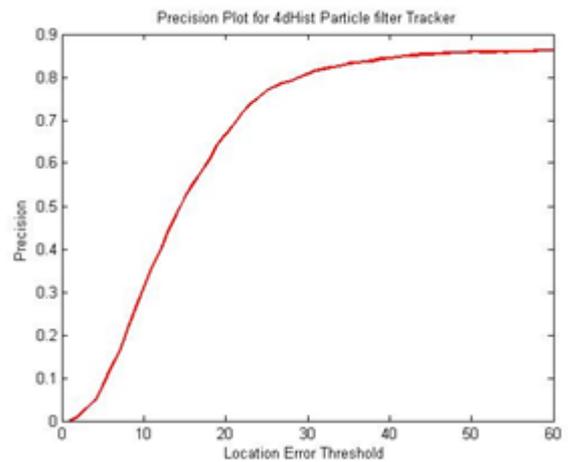
Figure 4 (a-f) The output frames of video sequence Skater2 showing tracked object, (g) Precision curve

Table 3 Comparison of Precision with previous trackers for location threshold of 20 [26]

Tracker	Stuck	SCM	TLD	CXT	VTD	CSK	LOT	OAB	4-d hist based proposed tracker
Precision in % successful frames	65.6	64.5	60.8	57.7	57.6	54.5	52.2	50.9	66.6



(a) From cvpr13_benchmark[26]



(b) 20 pixel location threshold value = 66.6%

Figure 5 Precision plots for the dataset trackers (a) Top most tracker responses from benchmark [26] and (b) Our proposed 4-d Histogram based colour particle filter tracker

V. Conclusion

Comparison of results of proposed tracking algorithm with the previous tracking algorithms, on cvpr13_benchmark datasets is given in Figure 5 and Table 3. It can be observed that the proposed tracker approach performs better than the previous tracking algorithms reported in cvpr13 benchmark [26] in terms of precision value. Also the algorithm performs quite well under condition of the tracked object changing shape, size and orientation. The effect of illumination variation and background clutter is also overcome by the proposed tracker. The bounding box around the tracked object is always going to be smaller as tracker is using colour features of only largest segment. Due to this the success plot will never be appreciable. This can be overcome by using additional location features of the object like shape and boundary.

References

- [1] Alper Yilmaz, Omar Javed and Mubarak Shah, "Object Tracking: A Survey," ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006.
- [2] D. Comaniciu, V. Ramesh and P. Meer, Real-Time Tracking of Non-Rigid Objects using Mean Shift, CVPR, pp. 142-149, Vol. 2, 2000.
- [3] N. Gordon and D. Salmond, Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter, Journal of Guidance, Control and Dynamics, pp. 1434-1443, Vol. 18(6), November-December, 1995.
- [4] M. Isard and A. Blake, Contour Tracking by Stochastic Propagation of Conditional Density, ECCV, pp. 343-356, Vol. 1, 1996.
- [5] M. Isard and A. Blake, CONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework, ECCV, pp. 893-908, Vol. 1, 1998.
- [6] M. Isard and A. Blake, CONDENSATION – Conditional Density Propagation for Visual Tracking, International Journal on Computer Vision, pp. 5-28, Vol. 1(29), 1998.
- [7] T. Kailath, The Divergence and Bhattacharyya Distance Measures in Signal Selection, IEEE Transactions on Communication Technology, pp. 52-60, Vol. COM-15(1), 1967.
- [8] G. Kitagawa, Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, Journal of Computational and Graphical Statistics, pp. 1-25, Vol. 5(1), 1996.
- [9] D.Koller, J. Weber and J. Malik, Robust Multiple Car Tracking with Occlusion Reasoning, ECCV, pp. 189-196, 1994.
- [10] J. MacCormick and A. Blake, A Probabilistic Exclusion Principle for Tracking Multiple Objects, ICCV, pp. 572-587, Vol. 1, 1999.
- [11] Kun Wang, Xiaoping P. Liu, "Visual Object Tracking Based on Filtering Methods", Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE.
- [12] Jan Valiska, Stanislav Marchevsky and Rastislav Kokoska, "Object tracking by Color-Based Particle Filter Techniques in Video Sequences", Radioelektronika, 24th International Conference, IEEE, 2014
- [13] Jacob Scharcanski, Alessandro Bof de Oliveira, Pablo G. Cavalcanti and Yessenia Yari, "A Particle-Filtering Approach for Vehicular Tracking Adaptive to Occlusions", in IEEE Transactions on Vehicular Technology, Vol. 60, No. 2, February 2011.
- [14] Ahmed Abdel-Hadi, "Real-Time Object Tracking using Color-based Kalman Particle Filter", in International Conference on Computer Engineering and Systems (ICCES), 2010, pp. 337-341.
- [15] Y. Satoh, T. Okatani, and K. Deguchi, "A color-based tracking by Kalman particle filter," in Proceeding of International Conference on Pattern Recognition, 2004.
- [16] Maikokudo, Takuyaakashi, Mimorufukumi, and Yoshisukekurozumi, "Improvement of Likelihood in Particle Filter for Interactive Color Tracking", World Automation Congress © 2010 TSI Press.
- [17] Jin-Xia Yu, Yong-Li Tang, Xian-Cha Chen and Qian Zhao, "An Adaptive Resampling Strategy in Particle Filter", in Proceedings of the 2011 International Conference on Wavelet Analysis and Pattern Recognition, Guilin, 10-13 July, 2011.
- [18] Wei Leong Khong, Wei Yeang Kow, Ismail Saad, Chung Fan Liau and Kenneth Tze Kin Teo, "Particle Filter Based Vehicle Tracking Approach with Improved Resampling Stage", ICTACT Journal on Image and Video Processing, February 2014, Volume: 04, Issue: 03
- [19] W. L. Khong, W. Y. Kow, Y. K. Chin, I.Saad and K. T. K. Teo, "Enhancement of Particle Filter for Vehicle Tracking via Adaptive Resampling Algorithm", in Third International Conference on Computational Intelligence, Communication Systems and Networks, 2011, pp. 259-263.
- [20] Nummiaro K., Koller-Meier E., Van Gool L. , "Object Tracking with an Adaptive Color-Based Particle Filter", In: Van Gool L. (eds) Pattern Recognition. DAGM 2002. Lecture Notes in Computer Science, vol 2449. Springer, Berlin, Heidelberg
- [21] Gaurav Sharma, "Digital color imaging", IEEE Transactions on Image Processing, Vol. 6, No.7, , pp.901-932, July 1997.
- [22] K. M. Bhurchandi, P. M. Nawghare, A. K. Ray, "An analytical approach for sampling the RGB color space considering limitations of human vision and its application to color image analysis", Proceedings of ICVGIP 2000, Bangalore, pp.44-49.
- [23] Kishor Bhojar & Omprakash Kakde, "Color image segmentation based on jnd color histogram" International Journal of Image Processing (IJIP) Volume(3), Issue(6) pp 283-292
- [24] Megha Sahu, K.M. Bhurchandi, "Color Image Segmentation using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 140 – No.5, April 2016 15
- [25] Gargi V. Sangamnerkar, Dr. K.K.Bhojar, "Color Image Segmentation in HSI Color Space Based on Color JND Histogram", International Journal of Image Processing and Visual Communication ISSN (Online) 2319-1724 : Volume 2 , Issue 3 , April 2014 20
- [26] Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang. "Online object tracking: A benchmark." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.
- [27] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1090–1097.
- [28] Satoshi Hirose, Nozomu Hamada "A model update scheme of color-based particle filter for multi-color object tracking", IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2012) November 4-7, 2012
- [29] Jingjing Xiao, Mourad Oussalah, 'Collaborative Tracking for Multiple Objects in the Presence of Inter-Occlusions' IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 26, NO. 2, FEBRUARY 2016
- [30] Aniket V Gokhale, Kishor K Bhojar and Kishor M Bhurchandi. 'Color Image Segmentation based on 4-D Histogram using JND Color and Spatial Connectivity', International Journal of Computer Applications 167(10),28-35, June 2017