International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Robust and Accurate Human Tracking Algorithm for Handling Occlusion and Out of Plane Rotation

Anshul Pareek, Nidhi Arora

Abstract: ***Robust and accurate human tracking using computer vision is acquiring more and more attention, seeking to meet the demands of the increasing number of applications. This paper presents a novel approach to handle occlusion and out of plane rotation. Both these issues are crucial and continue to be a challenge in all state of art algorithms of computer vision-based human tracking. In this paper a SURF oriented scheme based human tracker is proposed, which searches for the target human in expanded rectangular region surrounding the previous target location. There is an online update of object model by selecting fresh templates every time. Here, superimposition of keypoints obtained from previous templates is done on fresh template using Affine transformation. Whether it's a pose change or not, is affirmed by affine transformation, by calculating aspect ratio of target enclosed region. An Autotuned classifier discriminates the case of occlusion and pose change and confirms tracking failure. The success rate and computational time proves the accomplishment of the proposed algorithm***

Keywords: Human Tracking, SURF, Grab-Cut Algorithm, UKF(Unscented Kalman Filter) and Autotuned Classifier.

I. INTRODUCTION

Real time based visual tracking is a crucial task in the vision-based system like augmented reality, motion-based detection, assistive robotics, video indexing and human-computer interaction[1], [2]. Computer vision systems are attaining new dimensions in video and image processing owing to low computing power and inexpensive high-quality cameras. There major challenges in tracking are occlusion, pose change, random object motion, light intensity variation, and non-static camera motion. To overcome the above-mentioned challenges tracking should be robust, adaptive and implementable in real time[3],[4].

This paper deals with all the above issues but mainly addresses the pose change and occlusion challenges. This algorithm uses tracking by detection scheme[5] based on SURF tracking[6]. SURF, a rotation-invariant interest point descriptor[7] locates target in the consecutive frames. SURF is preferred over other descriptor algorithms due to its robust nature towards distortions like light variations, abrupt motion and lower rate of frames in a video[7].

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Anshul Pareek*, ECE Deptt, Maharaja Surajmal Institute Of Technology, Delhi India. Email: er.anshulpareek@gmail.com
Dr. Nidhi Arora, CSE Department, G.D. Goenka University, Gurugram, India. Email: nidhi.arora1@gdgoenka.ac.in

Considering only interest point-based tracking[8][9],it is not sufficient when a real-time scenario is taken into consideration as the number of matching points varies randomly from one frame to another, resulting in failed tracking.

Also while tracking a human, pose change becomes a critical task as it is a non-rigid structure [10]. In order to overcome these challenges along with SURF, a motion model is proposed in this paper. This motion model is updated online over time by templates stuffed with previously projected stable points, by stable we mean the points which regularly appear on frames, and this is done with the help of affine transformation [11]. SURF tracker detects the target in all the frames. Every time when the target is successfully detected a UKF motion predictor[12][13] is updated accordingly. UKF is an extension of Kalman Filter. Kalman filter [14] can manage only linear equations whereas Extended Kalman filter (EKF) [15] and UKF can deal with non-linear equations. The difference between the two is EKF, to linearize the non-linear equation uses Jacobian Matrix whereas, UKF does not linearize such equations, instead, considers points from Gaussian Distribution [16]. Once the UKF is updated it helps the tracker in future whenever the tracking fails by predicting the location of target based on its previous location. Further, the Autotuned classifier will confirm whether it was a pose change or an occlusion occurred. Autotuned classifier[17] works as a selector, it selects the index type (linear, kmeans, kd-tree) to provide optimal performance. If a pose change has occurred then scaling and repositioning is done, else matching is conducted in an expanded region for occlusion

The next section briefs the problem statement. Section III explains the tracking algorithm. Section IV comes up with the experimental Test Bed used. Experimental results are discussed in Section V. A rigorous comparative analysis of proposed algorithm with previous works is stated in Section VI. Section VII reveals the conclusion part.

II. PROBLEM STATEMENT

Let there be a set of frames F_i, where i = 0, 1, 2….N of the video recorded or live video streamed. For frame F_0 a rectangular box B_0 is drawn over the target region to be detected. To compute SURF descriptors in box B_i, set of SURF descriptors of frame F_i is stated as

$$L(B_i) = \{(k_1, l_1, w_1), (k_2, l_2, w_2) \ldots\ldots (k_n, l_n, w_n)\} \quad (1)$$

where x_i is the feature point location in 2D of 64-dimensional SURF features L_i. w_i are the weights allocated to SURF descriptors L_i .l_k is a set of feature point locations in a given frame within a window and l_y is the corresponding set of descriptors to l_k. B_s and B_t are source and target window. Now the set of good matching points between B_s and B_t are computed and they are

$$L(B_s \sim B_t) = \{(k_1, l_1, w_1)^s, (k_2, l_2, w_2)^s \ldots\ldots (k_m, l_m, w_m)^s, (k_1, l_1, w_1)^t, (k_2, l_2, w_2)^t \ldots\ldots (k_m, l_m, w_m)^t\} \quad (2)$$

where $(k_m, l_m, w_m)^t$ are SURF descriptors for target window and $(k_m, l_m, w_m)^s$ are SURF descriptors for source window . The tracking window B has parameters centre (c), width(w) and height(h) and is represented as

$$B=(c, w, h) \quad (3)$$

Now our target is to find tracking window $B_i = (c_i, w_i, h_i)$ for all the given frames.

In this rectangular window, both foreground and background SURF descriptors are present which may lead to false tracking. In order to avoid it an elliptical region [E_0] is drawn which fits inside rectangular window removing background descriptors. The set of such descriptors in elliptical region in the first frame is given by

$$L(E_0) = \{(k_i, l_i, w_i) \mid (k_i, l_i, w_i) \in L(B_n) \wedge k_i \in E_0\} \quad (4)$$

where w_i=20 and i = 1,2,3...m. This is how the segmentation of the foreground from background is done. From here object template can be quoted as

$$O_t = \{(k_i, l_i, w_i) \mid (k_i, l_i, w_i) \in L(B_n) \wedge k_i \in E_0 \wedge k_n \notin BG_R\} \quad (5)$$

BG_R is the segmented background region. O_t initializes the object model O_m. The initial weight assigned to Om here is 20. The ultimate goal is to make object model which is explained in detail in tracking algorithm section.

III. TRACKING ALGORITHM

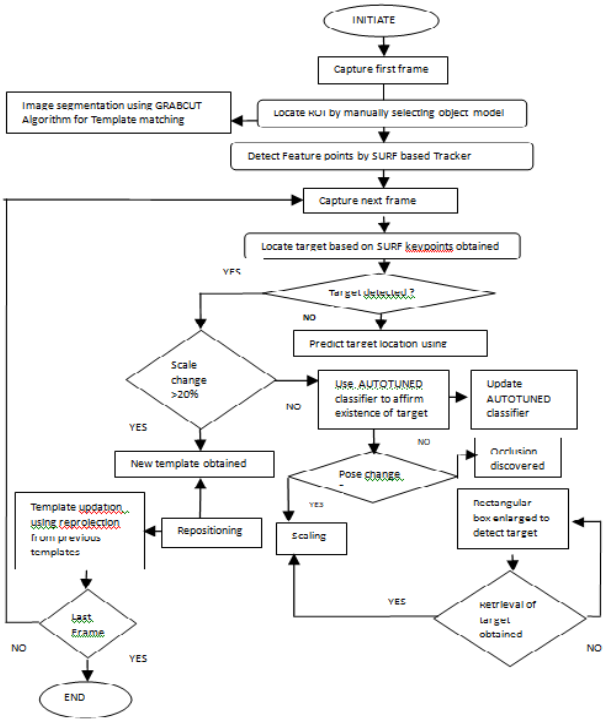


Fig 1: Tracking algorithm

The entire tracking algorithm is explained stepwise. Refer Fig 1 flowchart of algorithm.

- The tracking algorithm starts with capturing the first frame from the video sequence obtained.
- On this frame, the object to be tracked is manually defined by selecting a region of interest in a rectangular box
- SURF descriptors are calculated for the target enclosed in the rectangular box. As explained previously, these descriptors are not only of the target but also include background descriptors. This leads to false or failed tracking. To remove them an ellipse is drawn over the target which removes the background descriptors. Still few background descriptors are present are a part of ellipse. To further filter them out from template model a region growing algorithm[18] is applied. In present case grabcut [19] segmentation is used. It is based on graph cuts. This algorithm uses the Gaussian mixture model to predict the target object's color distribution and that of background also. SURF matching is done between current template and previous template and here the correspondences are computed. Each time tracking window undergoes an expansion of 10% from the previous window. RANSAC homography [20] is used to remove the outliers.
- For tracking to be declared successful percentage overlapping of tracking window[21] in the current template on previous template is to be greater than 50%.
  $$\text{overlapping } \% = (S_m / S_c) * 100 \quad (6)$$
  where S_m is the number of SURF keypoints from last selected template and Sc are number of SURF keypoints obtained through correspondences.
- This segmentation provides us with object template O_t and marks the initialization of object model O_m. Since descriptors have the tendency of fading over the due course of time, there's a need to make them present over the tenure of few frames, so weight is assigned to them. 20 is the initial weight and assigned every time a new template is selected.
- Once the object template is declared, target is located on each frame using Tracking by detection method. If tracking is conducted to be successful, the descriptor weights are reassigned by adding two points if descriptor is a matching point else subtracting one point for all other conditions
- Over time background descriptors keep on increasing in the tracking window. The reason behind this pattern is that it is impossible to filter all the background descriptors.
  So if even a single background descriptor creeps in, it will get multiplied over each frame. To avoid this scaling is required, which yields a scaling factor that is obtained from SURF correspondences (matching points) between source and target windows.

- The entire human body doesn't provide uniform descriptors over the human structure. It is commonly seen that torso provides the maximum number of descriptor as compared to head and legs. So, to compute scaling factor torso of human body is preferred. The center point is located on torso. According to the body ratio obtained this center point is shifted. Hence repositioning is obtained.
- Online updation of the object model is required from time to time else it would lead to false or failed tracking. If a scale variation of 20% or more is obtained, a new template is selected to avoid frequent tracking failures.
- The selection of new template isn't a forever thing. Tracking still gets failed over time due to stability vs plasticity dilemma [22]. There are a few steady descriptors which are present in all previous frames. This would reduce the number of descriptors with increasing number of frames further resulting in high computational complexity and memory requirement. For this we add positive weight SURF descriptors with their keypoint locations to the newly selected template using affine transformation. To achieve this torso of human body is assumed to be rigid structure because affine transformation cannot be applied to non-rigid structures.
- After all the above measures are taken one cannot guarantee successful tracking. Pose change and occlusion still remain to be the biggest reason for tracking failure than any other challenges faced in visual tracking. When one such situation is faced, it is important to first differentiate whether it is a pose change or an occlusion has occurred. The solution to this problem is autotuned classifier[23], which will conclude the challenge faced.
- Autotuned classifier when used, creates index which automatically tunes to offer best performing fast search structure by choosing any of the randomized kd-tree, hierarchical kmeans or linear fast search[24]. Kd-tree searches in parallel, k means searches hierarchically and linear will do a brute-force search. In any of the three selected basically the descriptors from the first template are used for creating kd-tree, kmeans or linear classifier data. Further these descriptors are thrust onto template pool created. Every time a new template is chosen, the descriptors of it are pushed into the pool created. This would help in reconstruction of classifier selected. This simultaneously updates the classifier and helps when tracking fails. The location is predicted by UKF. And with the help of nearest neighbor search, the closest match is found out. From here the number of foreground and background descriptors are computed. Based on this it is concluded whether a pose change or occlusion has occurred. If the number of foreground descriptors are less then 1.5 times that of background descriptors then occlusion is confirmed else it is a case pose change.
- If it is a case of occlusion, recovery is done by UKF motion predictor by expanding the size of the tracker window to recover target in a given frame.
- If pose change is detected, it is found whether it is in-plane rotation or out-of-plane rotation. Side pose is out-of-plane rotation leading to confirmed failed tracking. All such templates are required to be discarded and not added to template pool.

IV. TEST BED

Here all the implementation is done in C++ on opencv 3.4.0 version with linux based operating system configuration on an apple MAC book air with intel i5 processor. Videos used for the compilation of results are one from IIT Kanpur(IITk) dataset, one from Youtube (Youtube) and two are our datasets(SS1, SS2).SS1 is indoor tracking and SS2 is outdoor. Videos show several challenges like out-of-plane rotation, occlusion, scaling, abrupt camera motion, and light variation. All the videos used are of 640X480 resolutions.

V. EXPERIMENTAL RESULTS

The experimental results are discussed hereby. As discussed in the previous section four datasets are used to check the efficiency of the algorithm. Fig 1 shows the result of manual selection of target in first frame. Target is captured in a rectangular box containing background descriptors, to eradicate them an ellipse is drawn over the target that fits in this rectangle. The human body is divided into three parts head, torso and legs as visible in Fig 2(a) for all the four datasets. Later to remove the remaining background descriptors, grabcut is applied and results are shown in Fig 2(b).

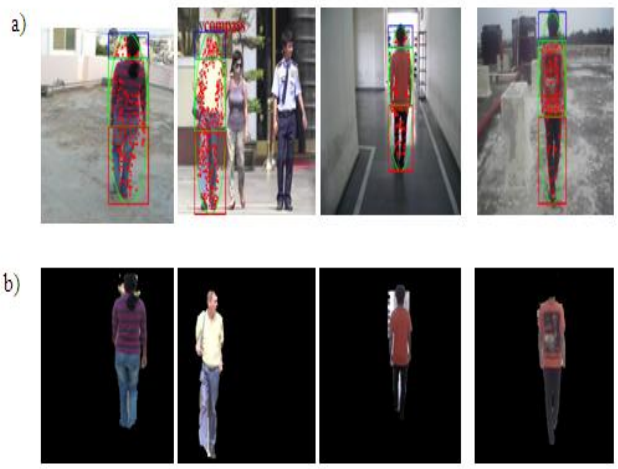


Fig 2: (a) Target manually selected and bounded within ellipse with head, torso, and legs division enclosed in seperate boxes.
(b) image segmentation results using Grabcut

Fig 3(a,b,c) shows the projection points and projection points bounded inside the ellipse are shown in Fig 3(c,d,e) for all four data .

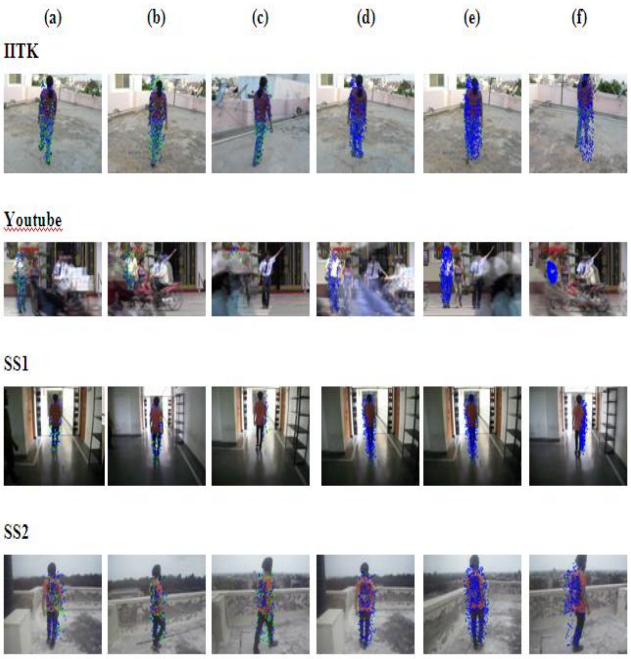
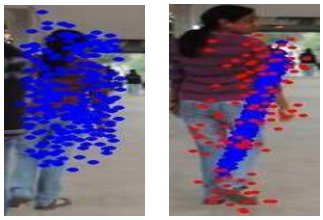


Fig3: The projection points and projection points bounded inside the ellipse

Fig 4(a) shows the projection points of the last frame after which the pose change occurred and confirmed by the affine transformation that is shown in Fig 4(b). It is clearly visible in Fig 4(a) that all the points are spread over the target as there is a pose change, these points start concentrating with the increase in rotation. This is similar to the case where let's assume points on the periphery of a transparent ball, as the ball rotates these points come closer and closer. At rotation of 90^0 all the points come in straight line. Fig 4(b) detects out-of-plane rotation by using Affine transformation. There's a straight line showing concentration of points hence confirming out-of-plane-rotation.



Last Frame New template using AT

(a) (b)

Fig4 :Affine transformation on pose change

Fig 5 and Fig 6 shows the average number of descriptors present in object model per frame and average computational time per frame for the data sets used.

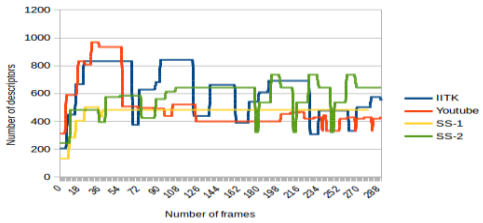


Fig 5:Average number of descriptors

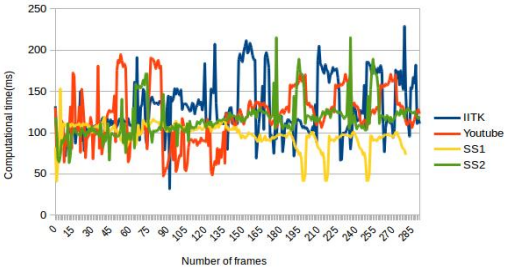


Fig 6:Average computational time

Fig 7: Templates generated for each dataset

Fig 7 shows templates generated for each dataset. It is seen that each template shows a different pose during motion. Tracking results are shown in Fig 8 for all datasets. The results of tracking videos are available online for verification [25]-[27].

Fig 8: Tracking results

Table I shows the algorithm performance for all datasets. It shows the robustness against the abrupt camera motion and pose change. Except Youtube dataset all other datasets have abrupt camera motion and still good success rate is obtained. The descriptor range is maximum 1000. This is way less than obtained in other methods, this leads to less memory required to store templates.

Table I: Performance sheet of the algorithm for all datasets

Dataset	Total no. Of frames	Camera motion and Pose change	Scaling upto	No. of descriptors	No. of templates generated	Average time (ms)	Success rate
IITK	290	Abrupt, Yes	18%	200-900	20	129	96.78
Youtube	290	Smooth, Yes	51%	300-1000	11	120	82.56
SS1	280	Abrupt, No	2%	100-500	5	97	86.23
SS2	290	Abrupt, Yes	20%	200-800	13	117	98.74

In order to state the efficiency of the proposed algorithm, there should be a transparent comparison with other prevalent algorithms. So, in this paper we compare our proposed algorithm like reprojection based Mean-Shift-SURF algorithm[28] and SURF Mean-Shift based object model[29]. To compare them three parameters are selected Success Rate, Computational time and percentage of overlap. Table II shows this comparative analysis. Computational time is calculated for each frame to process and an overall average is taken. This average is compared for all the three algorithms when run on all four datasets. Next important parameter that decides the performance of any tracking algorithm is its overlap percentage. To calculate this the ground truth of each frame is manually computed. This selection is compared and the percentage region that is common with tracker window is found out, this is overlap percentage. If the common area is more than 50 %, it is considered to be successful tracking else the tracking considered to be failed. Once this overlap percentage is in hand it is simple find the success rate[]. The success rate is simply the ratio of the total number of successfully tracked frames(n) to that of the total number of frame(N). Mathematically it is presented as

$$SR = (n/N) \times 100 \quad (7)$$

From the above-mentioned parameters it is clear that for an algorithm to have a good performance it should have a high overlap percentage and success rate with low computational time.

DataSet	Parameters	Algorithms Comparative Analysis		
		Reprojection based MeanShift	URF-Mean-Shift based object model	Proposed algorithm
IITK	SR	88.29	46.56	96.78
	AOL	68.9%	60.32%	65.34 %
	AT	126 ms	582ms	129 ms
Youtube	SR	0	6.25	82.56
	AOL	8.34%	24.24%	63.23 %
	AT	102 ms	431 ms	120 ms
SS1	SR	14.67	38.45	86.23
	AOL	29.64%	40.23%	47.24%
	AT	88 ms	320ms	97 ms
SS2	SR	92.45	62.47	98.74
	AOL	71.9%	64.34%	76.2%
	AT	110 ms	576 ms	117 ms

Going through the results in Table II it can be concluded that our proposed algorithm has a highest success rate for all the datasets. Reprojection based Mean-Shift-SURF algorithm shows good results for IITK and SS2 datasets because the target faces no occlusion and also the color of foreground is quite different from the background. But at the same time it is quite unsuccessful while tracking Youtube and SS2 dataset because there are number of occlusion and color intensity is dark in respective datasets. But it takes less time than the proposed algorithm in all the cases. The proposed algorithm surpasses performances of SURF Mean-Shift based object model in all the datasets and parameters.

VI. CONCLUSION

In the proposed algorithm the object model is updated online by projecting the most stable descriptors on to the latest template. This makes the visual tracking robust for the tracking challenges like pose change, scale change, illumination variation, all types of occlusion and abrupt camera motion. Also the Autotuned classifier used select the high-performance classifiers to distinguish between pose change and occlusion cases. This makes it unique for point-based methods.

To conclude, this algorithm is successful in dealing with major visual tracking challenges providing high success rate at low computational time.

REFERENCES

1. Zhigang Bing, Yongxia Wang, Jinsheng Hou, Hailong Lu, and Hongda Chen " Research of tracking robot based on surf features". International Conference on Natural Computation (ICNC) IEEE Yantai Shangdong(2010) 3523–3527.
2. W. Hu, X. Zhou, W. Li, W. Luo, X. Zhang, S. Maybank, "Active contour-based visual tracking by integrating colors shapes and motions",(2013) IEEE Trans. Image Process., vol. 22, no. 5, pp. 1778-1792.
3. K. Rasool Reddy, K. Hari Priya, N. Neelima "Object detection and tracking: A survey". International Conference on Computational Intelligence and Communication Networks (CICN)(2015).
4. Alper Yilmaz, Omar Javed, and Mubarak Shah "Object tracking: A survey," ACM Computing Surveys (CSUR)(2006) 38(4): Article 13.
5. [5]M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. InCVPR, 2008.
6. H. Bay et al(2008) "Speeded-up robust features (SURF) ." Computer Vision and Image Understanding, 110(3): 346-359, 2008.
7. Pareek Anshul and Arora Nidhi (2018), "Evaluation of Feature Detector-Descriptor Using Ransac for Visual Tracking " International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM-2019). Available at SSRN: https://ssrn.com/abstract=3354470.
8. T. Lindeberg "Scale selection properties of generalized scale-space interest point detectors", Journal of Mathematical Imaging and Vision, Volume 46, Issue 2, pages 177-210, 2013.
9. [9] Schmid, Cordelia; Mohr, Roger; Bauckhage, Christian (1 January 2000). "Evaluation of Interest Point Detectors" (PDF). International Journal of Computer Vision. **37** (2): 151–172. doi:10.1023/A:1008199403446
10. D. A. Klein, D. Schulz, S. Frintrop, A. B. Cremers, "Adaptive real-time video-tracking for arbitrary objects", Proc. IEEE IROS, pp. 772-777, 2010.
11. V. Ferrari, T. Tuytelaars, and L. van Gool. Real-time affine region tracking and coplanar grouping. In IEEE Conf. CVPR, volume 2, pages 226-233, 2001.
12. Han, S. et al. Design and capability analyze of highdynamic carrier tracking loop based on UKF.Proc. of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation(ION GNSS 2010), 21-24 September 2010, pp. 1960–1966.
13. Chen, X. et al.A novel UKF based scheme for GPS signal tracking in high dynamic environment.Proc. of 3rd International Symposium on Systems and Control in Aeronautics and Astronautics(ISSCAA), Harbin, 2010, pp. 202-206.
14. A. Kiruluta, M. Eizenman, S. Pasupathy, "Predictive head movement tracking using a Kalman filter", Systems Man and Cybernetics Part B: Cybernetics IEEE Transactions on, vol. 27, no. 2, pp. 326-331, 1997.
15. T. Song, J. Speyer, "The modified gain extended Kalman filter and parameter identification in linear systems", Automatica, vol. 22, pp. 1, 1986.
16. WAN Li, LIU Yan-chun, PI Yi-ming, "Comparing of Target-Tracking Performances of EKF,UKF and PF" RADAR Science and Technology,2007,vol 1
17. Z. Kalal, J. Matas, K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints", Proc. IEEE CVPR, 2010.
18. Asano, T. and N. Yokoya (1981). Image segmentation schema for low-level computer vision. Pattern Recognition 14 (1-6), 267-273.
19. C. Rother, V. Kolmogorov, and A. Blake, GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph., vol. 23, pp. 309–314, 2004.
20. Martin A. Fischler & Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" (PDF). Comm. ACM. **24** (6): 381–395. dio:10.1145/358669.358692.
21. F. Bashir, F. Porikli. "Performance evaluation of object detection and tracking systems", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), June 2006.
22. Steve Gu, Ying Zheng, and Carlo Tomasi, "Efficient visual object tracking with online nearest neighbor classifier," in 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand, 2010, pp. 271–282, Springer Berlin Heidelberg.
23. https://docs.opencv.org/2.4/modules/flann/doc/flann_fast_approximate_nearest_neighbor_search.html
24. D. Comanicui, V. Ramesh, and P. Peer. Real-time tracking of non-rigid objects using mean shift. In IEEE Conf. CVPR, volume 2, pages 142-149, 2000.
25. AnshulPareek-Human Tracking https://youtu.be/T9mTClv4RZA
26. AnshulPareek-Human Tracking https://youtu.be/gdLrWOIgFzA
27. AnshulPareek-Human Tracking https://youtu.be/mMhuW0697yI
28. Sourav Garg and Swagat Kumar, "Mean-shift based object tracking algorithm using surf features," in Recent Advances in Circuits, Communications and Signal Processing. 2013, pp. 187–194, WSEAS.
29. Meenakshi Gupta, Sourav Garg, Swagat Kumar, and Laxmidhar Behera, "An online visual human tracking algorithm using surf-based dynamic object model," in International Conference on Image Processing (ICIP), Australia, 2013, IEEE.

AUTHORS PROFILE

Ms. Anshul Pareek, is B.E. in Electronics and Communication from University of Rajasthan, and M.Tech. in Digital Communication from Rajasthan Technical university. Currently working as an Assistant Professor in Maharaja Surajmal Institute of Technology, New Delhi. Her research interests in fields of Artificial intelligence, Machine learning are Human Computer interaction, mainly Human motion tracking.

Dr Nidhi R. Arora, holds a PhD in the field of Information Retrieval from INHA University South Korea. Her dissertation work focused on designing a ranking algorithm to produce top-k search results for keyword query on data graphs. She is currently working as Associate Professor in GD Goenka University. Her research interests are in the field of Deep Learning, Natural Language Processing and Machine Learning. She has publications in top conferences and journals such as DEXA, DASFAA, Expert Systems With Applications (ESWA) and New Generation Computing.