SURF-Based Algorithm to Deal with Pose Change Challenge in Human Tracking

Anshul Pareek, Poonam Dahiya, Shaifali M.Arora

ECE Department of Maharaja Surajmal Institue of Technology, New Delhi, India

Corresponding author: Anshul Pareek, Email: er.anshulpareek@gmail.com

Most of the existing interest point-based methods do not deal with tracking drift owing to out-of-plane rotations based pose change. It continues to remain a big challenge for researchers. To address the issue, a SURF-based algorithm is developed in which the object model is upgraded during the course of tracking, for this new templates are selected whenever pose change is encountered. In this process, the fresh projecting points are added to the template pool extracted from previously generated templates employing affine transformation by calculating its aspect ratio. These works propose a novel implementation of the GRABCUT algorithm on interest point-based methods. This region growing algorithm eliminates the background descriptors from the object model and this information is used by a SURF-based tracker. Later to differentiate between pose change and occlusion situation an Autotuned classifier is implemented. The human tracking algorithm developed in this paper are computable in real-time and real-time experiments are conducted in indoor as well as in outdoor environments.

Keywords: Human Tracking, SURF, GRAB-CUT, Out-of-plane rotation, Pose change, Autotuned classifier.

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 241–251. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-26

1 Introduction

Computer vision-based algorithms help us attain robust and accurate human tracking that combat the growing need in wide application [1-4]. SURF is the most robust of all the available image matching algorithms [21]. This research proposes an inventive approach to steer then crucial real-time challenges. The two aforesaid issues are still a challenge to be conquered [22-25]. Here a SURF-based [5-7] human tracker is proposed that hunts the target object in an expanded rectangular area around the target location in the last frame. Fresh templates are selected for online updation of the object model timely resulting in robust tracking. Descriptor points from the last frames are superimposed to the current frame is done using affine transformation. Affine transformation is used to confirm pose change by computing the aspect ratio of the target enclosed region. An Autotuned classifier is used to discriminate pose change from occlusion and affirm tracking abortion. The success rate and computational time prove the accomplishment of the proposed algorithm.

One of the major challenges faced while tracking non-rigid objects is the pose change. It is expected from any human to turn when and where required, tracking failure is confronted in such condition by most of the pre-existing algorithms. To overrule this challenge while tracking SURF based dynamic object model is proposed in this paper. The steps involved are a selection of new frames whenever pose change is affirmed, besides this a motion predictor the Unscented Kalman Filter (UKF). Whenever the target undergoes occlusion, predictor searches for the new location and re-initiate the tracking. Each time the predictor is updated in accordance to overcome a similar situation in the future [8-9]. After predictor, a classifier is also used that is useful in discriminating the pose change from occlusion. In this study, an auto-tuned classifier is used in our research [10]. Autotuned classifier [11] automatically works as a selector that selects the needful index type (linear, kmeans, kd-tree) to provide optimal performance. Once pose change is confirmed scaling and repositioning are implemented. Also, an expanded rectangular region search is conducted when the target is missing during or after occlusion.

2 Problem Statement

Assume the set of frames from a recorded/ live stream video. In the first frame of the video, a rectangular polygon is encased over the target. SURF descriptors are computed over the target in the box for frame sequence The SURF descriptors for the target window and SURF descriptors for the source window are computed. The tracking window B has parameters center (c), width(w), and height(h) and the aim is now to find the parameters of tracking window for the rest of the frames in the video.

3 Performance Parameters

- **Overlap Percentage:** Overlapping area of tracker window on ground truth window [12].
  - Overlap % = [area ($W_g \cap W_t$) / area (W_g U W_t)] X 100
  - $W_{t =}$ Tracker window
  - W_g = Ground truth window
  - It should be high, above 50% for successful tracking.
  - Overlap percentage is a measure of the accuracy of tracking which in turn affects the success rate.

- **Success Rate:** The ratio of successful tracking frames (n) to the total number of frames N[12].
  - Success Rate = (n/N) X 100
  - It should be high during tracking.
  - The success rate is an integral measure of the robustness of the entire tracking process.
- **Average Time:** Overall average time required for computation while tracking in msec. For efficient tracking, it should be low.

  Less average time signifies the compatibility of real-time tracking.

4 Tracking Algorithm

The tracking algorithm is shown in Fig 1. The steps of execution are explained point-wise and stated below: Initialization of algorithm is done by the selection of the first frame of the video, which consists of target present in it.

- Our region of interest is one that has the target in it. A rectangular box is manually drawn over the target for its selection.
- To track this target on all the coming frames, SURF descriptors are extracted on the target. But in the rectangular region, both FG and BG are present. To avoid any drifting of the model it's important to rectify background descriptors as much as possible, for this ellipse is fitted in this rectangular region. This discards more portion of background from the desired region but not all background is removed. Further, the GRAB-CUT algorithm[13,24] is implemented on the first frame which removes the entire background leaving behind only SURF descriptors from the foreground. GRAB-CUT is a region growing algorithm based on graph cuts and uses a Gaussian mixture model to predict the target object's color distribution and that of background also. SURF correspondences are obtained between two frames using image matching. Due to the displacement of the target in two frames, scaling is observed. If the limit of scaling exceeds 10% RANSAC [21] is used to remove the outliers.
- To calculate the success of tracking, the tracked window is compared with the ground truth image and their overlap is computed [14][24-25]. When the overlap of the tracking window on ground truth is greater than 50%, the target is successfully tracked on the frame processed.
- After this image segmentation, the object model is initialized and its template is generated for tracking. During tracking, there's the tendency of fading of the number of matching points over time. To make them stay for few more frames weights are assigned to them, initially, the value is 20 and increases every time a new template is allotted. If the descriptor is a matching point the weight is increased by 2, if not it is reduced by 1.
- By using tracking by detection technique on object template selected, target's location on each frame is estimated
- As we proceed with tracking it is always seen that the number of background descriptors keeps on rising, because it's not possible to eliminate each background descriptor. Even if a single point gets trespassed, it will tend to multiply for each tracking frame. This problem is handled by scaling.
- Not the entire human structure gives the same amount of descriptors. It is always observed that a good number of points are obtained on the torso of a human and a very few on its head and legs. So, the scaling is computed for the human torso instead of on the entire human structure. The

center of the human body is allotted to the torso only. This center point is moved as per the body ratio and repositioning is attained.

- Since the target undergoes a lot of transitions during tracking, it's important to update the object model in online mode from time to time. So, whenever during tracking, more than 20% limit is observed, that particular frame is selected as template and stored in the pool for future references.

- Scaling is not the sole reason for failed tracking, there are other reasons also behind it for example stability v/s plasticity dilemma. The human torso is considered to be rigid object as we cannot implement affine transformation to non rigid ones.

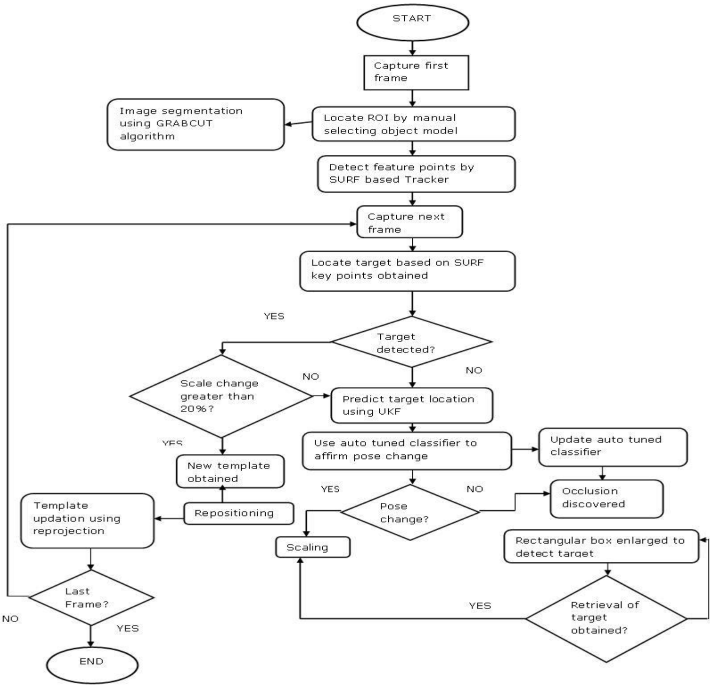

START
Capture first frame
Image segmentation using GRABCUT algorithm
Locate ROI by manual selecting object model
Detect feature points by SURF based Tracker
Capture next frame
Locate target based on SURF key points obtained
YES
Target detected?
NO
Scale change greater than 20%?
NO
Predict target location using UKF
YES
New template obtained
Use auto tuned classifier to affirm pose change
Update auto tuned classifier
Template updation using reprojection
Repositioning
YES
NO
Pose change?
Occlusion discovered
Scaling
Rectangular box enlarged to detect target
Last Frame?
NO
YES
YES
Retrieval of target obtained?
END

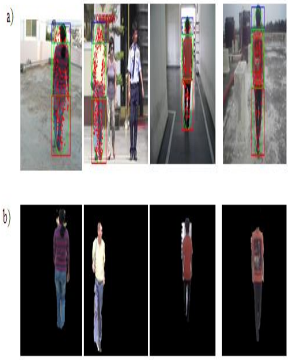
Fig. 1. Tracking Algorithm

- All the safety measures are taken to avoid any kind of tracking failure. But whenever the target undergoes occlusion and poses change, there’s a chance of tracking failure. To avoid such circumstances, it is important to first find out whether it is pose change or occlusion. For this auto-tuned classifier is implemented. This is a classifier that generates an index and automatically channels to tune between randomized kd-tree, hierarchical K-means, or linear fast search. Kd-tree searches in parallel, k means searches hierarchically and linear will do a brute-force search. In any of the three selected, the descriptors from the first template are used for creating kd-tree, k-means, or linear classifier data. These descriptors are loaded in the template pool. All the descriptors of the new template chosen are pushed in this pool. This step is done for the reconstruction and updation of the classifier.

- If failed tracking is encountered the new location of the target is predicted by UKF. This uses the nearest neighbour search to detect the closest match. Here the calculation of FG descriptor (FGD) and BG descriptor (BGD) is done. If the FGD number is 1.5 less than the BGD count, this is a case of occlusion encountered else the pose change is affirmed. For occlusion, UKF does the recovery of target by window expansion till the target is recovered. If not tracking is declared to be failed.

5 Results

The performance evaluation of the algorithm by implementing it on a number of data sets. The targets in the datasets face various real-time challenges. Six sets of videos are used, 1 set is from the pedestrian data-set of Pedestrian, 1 set is from the IIT-k dataset, and the 2 self-shot sets. In the entire datasets target undergoes various real-time challenges .The results of the experiment are discussed in this section for the above-mentioned datasets. The evaluation of the efficiency of the algorithm is done by implementing it on the datasets. As Fig 2 shows how the target is selected and the application of region growing algorithm i.e. grab cut on the target. In Fig 2(a) the manual selection of target enclosed in a rectangular box is shown. Later ellipse is fitted in a box; the entire human body is divided into head, torso, and legs. SURF descriptors extracted on the target are shown in red color blobs for all four datasets. Later implementation of grab cut on the first frame is shown in Fig 2(b). Projection points on target are shown in Fig 4(a,b,c) for various poses. Projection points that are bounded inside the ellipse are shown in Fig 4(d,e,f) for all four data. These images show how a number of projection points vary with the pose. Whenever there is a pose change the number of points lies in a line. How pose change is confirmed and a new template is selected is shown in Fig 3. As can be seen in Fig 3(a) the distribution of projection points is distributed over the entire human structure. In the next frame, the target is subjected to pose change. In this case, all the projection points lie in a line due to affine transformation as shown in Fig 3(b). This case can be compared to a transparent ball where the entire points lie on the ball's periphery. If the ball is observed from a distance and the ball is rotated continuously, the points appear to come close.



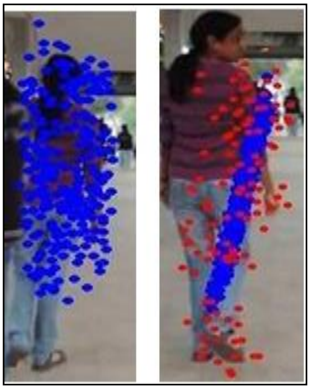


Fig 2(a). Selection of target

2(b). Grab Cut segemented image transformation

Fig 3. Pose change detection using affine

When the ball suffers a displacement of 90°, one can see that all the points lie in a line. This is how pose change is confirmed in any frame. Further, this plane is selected as a new template.

In Fig 6(a) and Fig 6 (b) statistical data of the number of descriptors and computational time is shown respectively. This graph shows the consistent values of both parameters giving consistent average data throughout the tracking process. This shows the robustness of the algorithm.

Each dataset during execution generates and selects fresh templates for online updation of the object model. These templates for each dataset are shown in Fig 5. Each template is selected when pose change is encountered and this is prominently visible in Fig 5. The original video of tracking is available online for inspection and viewing [17-19]. Tracking snapshots for various frames are shown in Fig 7 for all four data sets.

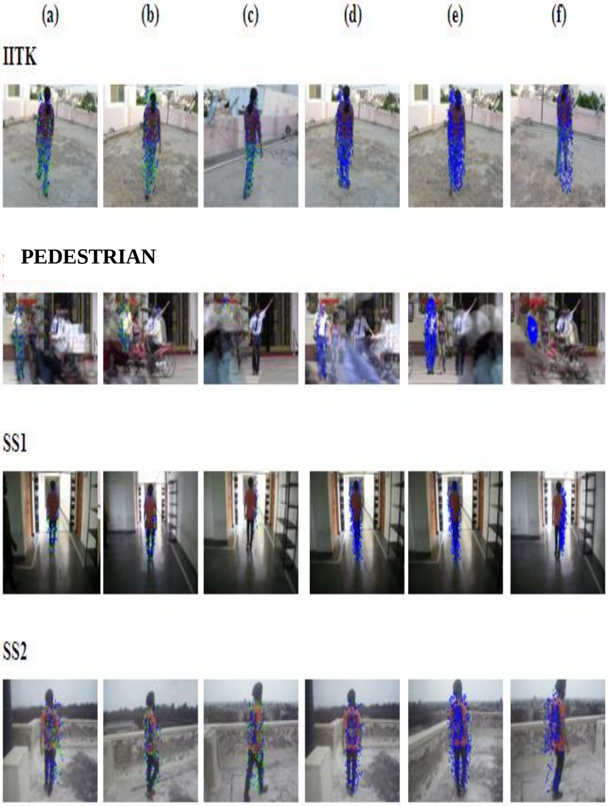


Fig. 4. Projection points under various pose

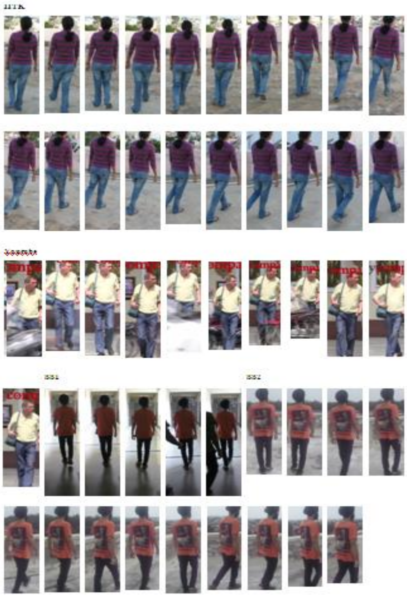


Fig. 5. Templates selected for each dataset

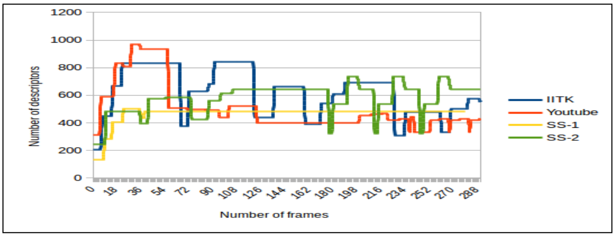


Fig. 6(a). Average number of descriptors

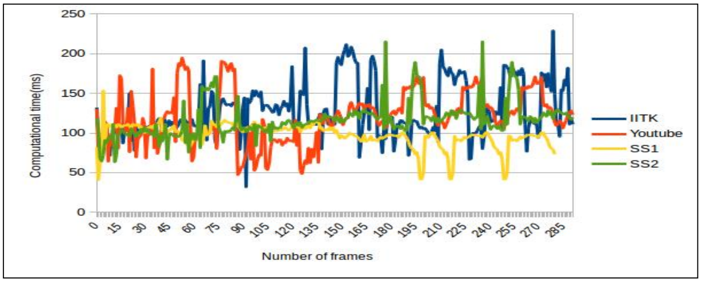


Fig. 6(b). Average computational time

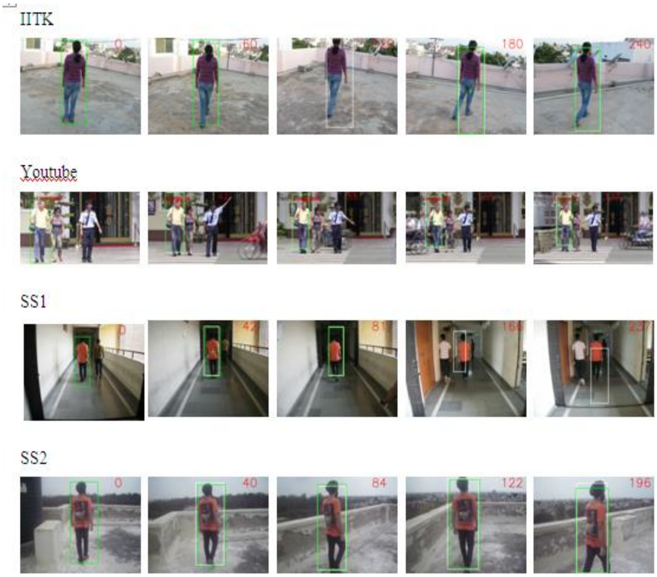


Fig. 7. Tracking results

The performance analysis of the proposed algorithm for all four datasets is shown in Table1. Three out of four datasets exhibit abrupt camera motion and pose change, and target undergoes scaling between 2-51%, still the success rate achieved is in between 82-98%. This proves the robustness and accuracy of the proposed algorithm. The maximum limit of descriptor is 1000, this limit is suitable for real time implementation as it requires low memory consumption for template storage. Also, this value is quite less than other approaches discussed in previous paper.

TABLE-1 Result analysis

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

The analysis states the efficiency of the algorithm. Now it is to be proved how this algorithm is better than other pre-existing algorithms. To attain this, a transparent comparison is presented in TABLE-2. The proposed algorithm is compared with the reprojection based Mean-Shift-SURF algorithm (22) and SURF Mean-Shift based object model(24). The evaluation parameters for comparison are the standard open CV parameters and they are Success Rate(SR), Average computational Time (AT), and Percentage of Overlap(AOL).

AT is computed in msec or sec and this the time taken by each frame to get processed and in the end average is taken for all frames. In the last, this average time is compared for all the three algorithms when applied to the datasets.

The next evaluation parameter is AOL. To compute it, for each frame manual ground truth is extracted. The tracking window generated by the algorithm is compared with this ground truth, the percentage overlap of two windows gives AOL. To consider tracking to be successful the standard value of AOL has to be a minimum of 50% else tracking on a particular frame is treated as a failure. The factor that computes SR is also AOL. The success rate is simply the ratio of the total number of successfully tracked frames(n) to that of the total number of frames (N). Mathematically it is presented as

$$SR = (n/N) X 100$$

To design a robust and accurate tracking algorithm the algorithm must exhibit low AT.

The comparison analysis as shown in TABLE 2 suggests that the highest success rate for all the datasets is delivered by the proposed algorithm. 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 the foreground is quite different from the background. But at the same time, it is quite unsuccessful while tracking Youtube and SS2 datasets because there are a number of occlusions, 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 the performances of the SURF Mean-Shift based object model in all the datasets and parameters.

TABLE-2 Comparison analysis

DataSet	Para-meters	Algorithms Comparative Analysis		
		Reprojection based MeanShift	**SURF-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

6 Summary

The proposed algorithm puts forward the scheme of online update of an object model from time to time. This enhances the robustness of the algorithm.

The most stable matching points are superimposed on the current frame using affine transformation. The template selected from this process temporal and stable data information.

- The proposed algorithm has tremendous potential to overcome the real-time challenges. Besides pose change other challenges are also successfully handled like, considerable scaling factor, intensity variation, abrupt camera motion, and partial or full occlusion.

- Implementation of auto-tuned classifier discriminates pose change from occlusion. Making it a new contribution to tracking.

References

[1] Yilmaz, A., Javed, O. And Shah, M. (2006). Object tracking: A survey. *Acm computing surveys (CSUR),* 38(4): 13-es.

[2] Yang, H. et al.(2011). Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18): 3823-3831.

[3] Van de Weijer, J., Gevers, T. and Bagdanov, A. D. (2005). Boosting color saliency in image feature detection. *IEEE transactions on pattern analysis and machine intelligence*, 28(1): 150-156.

[4] Lukac, R. and Plataniotis, K. N. (Eds.). (2018). *Color image processing: methods and applications*. CRC press.

[5] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of computer vision*, 60(2): 91-110.

[6] Bay, H. et al. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3): 346-359.

[7] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition (CVPR'05),* 1: 886-893.

[8] Chen, X. et al. (2010). A novel UKF based scheme for GPS signal tracking in high dynamic environment. In *3rd International Symposium on Systems and Control in Aeronautics and Astronautics,* 202-206.

[9] Kiruluta, A., Eizenman, M. and Pasupathy, S. (1997). Predictive head movement tracking using a Kalman filter. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(2): 326-331.

[10] Song, T. L. and Speyer, J. L. (1986). The modified gain extended Kalman filter and parameter identification in linear systems. *Automatica*, 22(1): 59-75.

[11] Wan, L. et al. (2007). Comparing of Target-Tracking Performances of EKF, UKF and PF. *Radar science and technology*, 1: 003.

[12] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

[13] Kalal, Z., Matas, J. and Mikolajczyk, K. (2010). Pn learning: Bootstrapping binary classifiers by structural constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 49-56.

[14] Rother, C., Kolmogorov, V. and Blake, A. (2004). GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3): 309-314.

[15] Bashir, F. and Porikli, F. (2006). Performance evaluation of object detection and tracking systems. In *Proceedings 9th IEEE International Workshop on PETS,* 7-14.

[16] AnshulPareek-Human Tracking https://youtu.be/T9mTClv4RZA

[17] AnshulPareek-Human Tracking https://youtu.be/gdLrWOIgFzA

[18] AnshulPareek-Human Tracking https://youtu.be/mMhuW0697yI

[19] Garg, S. and Kumar, S. (2013). Mean-shift based object tracking algorithm using SURF features. In *Recent Advances in Circuits, Communications and Signal Processing Conference,* 187-194.

[20] Gupta, A. M. et al. (2013). An on-line visual human tracking algorithm using SURF-based dynamic object model. In *IEEE International Conference on Image Processing,* 3875-3879.

[21] Pareek, A. and Arora, N. (2019). Evaluation of Feature Detector-Descriptor Using RANSAC for Visual Tracking. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM).*

[22] Pareek, A. and Arora, N. (2020). Re-projected SURF Features Based Mean-Shift Algorithm For Visual Tracking. *Procedia Computer Science*, 167: 1553-1560.

[23] Pareek, A. and Arora, N. (2019). Robust and accurate human tracking algorithm for handling occlusion and out of plane rotation. *International Journal of Innovative Technology and Exploring Engineering*, 8(12): 3768- 3773.

[24] Pareek, A. and Arora, N. (2019). A hybrid SURF-based tracking algorithm with online template generation. *International Journal of Innovative Technology and Exploring Engineering*, 8(12): 794-798.

[25] Pareek, A., Arora, V. and Arora, N. (2021). A Robust Surf-Based Online Human Tracking Algorithm Using Adaptive Object Model. In *Proceedings of International Conference on Artificial Intelligence and Applications*, 543-551.