Visual Face Tracking: a Coarse-to-Fine Target State Estimation

Wassim Bouachir and Guillaume-Alexandre Bilodeau
LITIV Lab.
École Polytechnique de Montréal
Montréal (Québec), Canada
{*wassim.bouachir, gabilodeau*}*@polymtl.ca*

***Abstract*—Keypoint-based methods are used in visual tracking applications. These methods often model the target as a collection of keypoint descriptors. Target localization on subsequent frames is thus a complex task that involves detecting keypoints, computing descriptors, matching features, and checking match consistency to update the reference model adequately and avoid tracker drifts. This work aims to boost keypoint tracking efficiency while reducing complexity by a coarse-to-fine state estimation to track human faces. In this context, we present a novel face tracking algorithm combining color distribution and keypoints to model the target. Our tracking strategy is based on a color model to predict a coarse state where the target search should be performed using keypoints. The fine estimation of the target state is then made by matching candidate keypoints with those of a reference appearance model that evolves during the tracking procedure. Qualitative and quantitative evaluations conducted on a number of challenging video clips demonstrate the validity of the proposed method and its competitiveness with state of the art trackers.**

***Keywords*-Face Tracking; Object Tracking; SIFT Keypoints; Particle Filtering;**

I. Introduction

Detecting, tracking and recognizing individuals are key components in automated video surveillance systems. Despite great progress in automated visual tracking, person tracking remains a challenging problem due to numerous real life difficult situations, such as sophisticated object shape, complex motion, and appearance changes caused by pose, illumination, occlusion, etc. Finding the appropriate appearance model is a key problem that attracted much attention in recent years. In this work, we focus on the problem of tracking a human face with no prior knowledge other than its state in the first video frame. This tracker will be used in a face recognition application were a PTZ camera follows a face until enough information are extracted from it to allow person identification. The designed system should be able to track arbitrary movements of a human face, under different scales, with a variable background (due to camera motion), and under changing illumination conditions. In addition to these requirements, the proposed algorithm should robustly handle partial occlusions. We address the problem of finding an appearance model robust to partial occlusion and finding an efficient target search strategy. Since our algorithm is designed to track human faces, the target is represented by a region using color features as a global descriptor for coarse localization of the target position, in addition to keypoint descriptors for fine localization and occlusion handling. Both appearance models reinforce each other for more robust and more accurate state estimation. It has been shown that an adaptive appearance model, evolving during the tracking procedure is the key to good performance [1], [2]. To ensure model adaptation to the target appearance changes, our appearance model is updated during tracking, under certain conditions to avoid a too large drift.

The contributions of this paper are: 1) the appearance model of the target, and 2) the search strategy for predicting the target location in the current frame. In our search strategy, kernel tracking and point tracking are used in conjunction in order to perform a robust prediction. Note that in our work, and in accordance with the definition in [3], kernel tracking refers to the target representation (not to the iterative localization procedure *mean-shift*). For example, the kernel can be a rectangular or a circular shape with the associated color distribution. In the first step, our algorithm uses a particle filter to track a kernel for finding image region candidates where the keypoint target search should be performed. The fine estimation of the target state is based on keypoint descriptors computed in the region candidates and matched with those of the target model. The advantage of reducing the keypoint search space with a coarse state estimation is twofold: i) by reducing the search space, we reduce the number of possible false keypoint matches and simplify the keypoint matching process, and ii) considering a smaller search space in the image reduces the number of keypoints to compute. Indeed, in our method we limit the search area on the current frame to the overlapping region defined by the best particles as selected from the kernel representation.

II. Related Work

Object tracking methods can be divided into three categories according to their appearance model [3]: point tracking, kernel tracking, and silhouette tracking. Silhouette tracking methods use the information encoded inside the object region which is estimated in each frame by either shape matching or contour evolution [4], [5]. The most significant advantage of silhouette tracking methods is their

flexibility to handle various object shapes by extracting the complete object region. Among their important issues, is their capability to address the occlusion problem explicitly [3]. Moreover, contour tracking algorithms require that a part of the object in the current frame overlaps with the object region in the previous frame.

In kernel tracking methods, a kernel of different shape is used depending on the target (e.g. a rectangular template to track the complete human body, circular shape for face tracking, etc.) [6]–[8]. Targets are tracked by computing the motion of the kernel in subsequent frames in the form of a parametric transformation, such as translation and rotation. The use of geometric shapes to represent objects is very common due to computational efficiency. One of the limitations of kernel methods is that parts of the objects may lie outside the kernel, while parts of the background may lie inside it. This makes model updating more challenging as including background pixels in the model results in tracker drift.

Point trackers represent targets by points and the association of the points across consecutive frames is based on previous objects states that can include point descriptors and locations [9]–[11]. They are naturally suited to handle occlusions as partial matches between points are sufficient for most tracking scenarios. Recent point tracking methods model an object as a set of keypoints detected by an external mechanism (i.e. a keypoints detector) [10], [11]. Once the keypoints are detected in a video frame, and their descriptors are computed, the object localisation can be achieved according to two possible approaches: classification in the case of a discriminative algorithm, and matching in the case of a generative tracker. Matching approaches store keypoint descriptors in a database. The descriptors are designed to be invariant to various perturbation factors (noise, scale, rotation, illumination, etc.) and can be matched with those of the target model in a nearest-neighbour fashion. Classification approaches are used in discriminative algorithms and consider matching as a binary classification problem: each keypoint is classified as a keypoint from the background, or a keypoint from the target model. The initial classifier needs to be learned offline, considering the background and the object observed under various transformations.

Representing a target by a set of keypoints enforces invariance against rotation, scale changes, changes in viewpoint and robustness to partial occlusions [12]. However, detecting keypoints on large image regions, computing the descriptors and matching them is quite costly. On the other hand, the keypoint classification approach requires a prior knowledge of the object appearance and a training stage.

III. Tracking method

We propose a novel generative tracking algorithm based on a combination of global and local features of the target, where the search strategy for finding the target combines kernel tracking and point tracking techniques.

A. Motivation

By using a geometric shape to contain the target, and global features for modeling, kernel trackers perform well while maintaining a low complexity. Nevertheless, this approach is not designed to handle occlusions, unless representing the target by multiple fragments to be matched. Keypoint methods can handle this problem by establishing partial correspondences, but may corrupt the target model in case of mismatches or tracker drift. In this context, the proposed method includes two steps that take advantage of both approaches, while mutually reducing their drawbacks. In more concrete terms, kernel tracking is firstly applied to provide a coarse localisation of the target. Keypoints are then used to improve the prediction by finding a final more accurate position, and by adding distinctiveness and robustness against occlusions. Moreover, we consider that their reliability is confirmed by the kernel tracker since they are located on candidate regions determined in the first step. This assumption offers several advantages:

- keypoints model adaptation becomes easier;
- drift and mismatches are considerably reduced;
- matching keypoints can be performed in a simple way, as the matched keypoints should be consistent with the target model, and no further spatial consistency should be verified.

Details of the different system components and algorithmic steps are presented in the following sections.

B. Appearance model

We delimit the target using a circular area that contains the tracked face in every video frame. As we will show in the experiments, the method presented here is not limited to faces and can be adapted, or even directly applied to track other types of arbitrary moving objects. The proposed target model describes the image region delimited by the circle that circumscribes the face. This model includes two types of features: 1) the RGB color probability distribution represented by a quantized 3D histogram, and 2) a set of keypoint descriptors computed within the face region. By constructing an m-bin histogram $\hat{\mathbf{q}} = \{\hat{q}\}_{u=1\ldots m}$, with $\sum_{u=1}^{m} \hat{q}_u = 1$, some parts of the background may lie inside the circular kernel. As discussed in [13], these pixels will affect the color distribution and cause tracking drift. To reduce the effect of these pixels in the distribution calculation, we use a kernel function $k(x)$ that assigns smaller weights to pixels farther from the kernel center. On the other hand, $\hat{\mathbf{q}}$ is normalized to ensure scale invariance. More formally, the RGB histogram is computed for the n pixels inside the circular region according to the equation:

$$\hat{q}_u = \frac{1}{\sum_{i=1}^{n} k(d_i)} \sum_{i=1}^{n} k(d_i)\delta[c_i - u] \quad (1)$$

where $d_i \in [0,1]$ is the normalized distance from the pixel x_i to the kernel center, c_i is the bin index for x_i in the quantized space, δ is the Kronecker delta function, and $k(d_i)$ is the tricube kernel profile defined by:

$$k(d_i) = \frac{70}{81}(1 - d_i^3)^3. \quad (2)$$

In this way, the proposed color model is suited to our application requirements. Indeed, the color histogram is: i) quantized to be quite general, reduce noise and light sensitivity and reduce the computation complexity, ii) normalized to enforce scale invariance, and iii) weighted to reduce the effect of the background pixels in the target model, and therefore reduce model drift.

The proposed system should be able to handle many difficult scenarios, such as occlusions and the presence of background regions with colors similar to those of the target. In addition, it has been shown that even for individuals of different races, the skin color distributions are very similar [14]. To ensure a more robust and distinctive feature set, the target reference model also includes SIFT keypoints [15] detected and described in the target region. Our method is not specific to SIFT. Even faster keypoint detector/descriptor combination may be used, although SIFT remains one of the most reliable method under various image transformations [16]. Including keypoints to the target model increases the distinctiveness of the tracking algorithm to distinguish the tracked person from the other individuals who may enter the field of view and ensures robustness against changes in lighting conditions and scale. Moreover, the tracking algorithm will be able to handle efficiently partial occlusions due to the locality propriety of SIFT keypoints that allows partial match.

C. Coarse target state estimation

To localize the target in the current frame, the search is guided by a probabilistic particle filtering approach [17], where each particle is a circular region characterized by its RGB color distribution as described in section III-B. More specifically, the possible target states in frame t are represented by N randomly generated particles $\{s_t^{(n)} : n = 1, ..., N\}$. Each particle is defined by:

- the current state values: position (x, y) and radius r;
- the weight $\pi_t^{(n)}$ that reflects the importance of the particle.

To reduce the computational cost, we assign a cumulative weight $c^{(n)}$ to each pair $(s^{(n)}, \pi^{(n)})$ where $c^{(N)} = 1$. When processing a frame t, the new particles are generated from the states of $t-1$ according to the following procedure:

1) generate a random number $R \in [0,1]$;
2) find the particle $s_{t-1}^{(j)}$ with the smallest value of j verifying $c_{t-1}^{(j)} \geq R$;
3) generate for the selected particle $\hat{s}_t^{(n)}$, a new particle $s_t^{(n)}$ with $s_t^{(n)} = f(\hat{s}_t^{(n)}, W_t^{(n)})$, where $W_t^{(n)}$ represents the error;
4) determine the weight $\pi_t^{(n)}$ for the particle $s_t^{(n)}$ by comparing its RGB color distribution estimated at frame t to the reference model color distribution.

To evaluate the similarity between the reference color model $\hat{\mathbf{q}}$ and the color distribution $\hat{p}_t^{(n)}$ of a generated particle $s_t^{(n)}$, we define the distance between the two distributions as:

$$d(\hat{\mathbf{q}}, \hat{p}_t^{(n)}) = \sqrt{1 - \rho[\hat{q}, \hat{p}_t^{(n)}]} \quad (3)$$

where

$$\rho[\hat{q}, \hat{p}_t^{(n)}] = \sum_{u=1}^{m} \sqrt{\hat{q}_u . \hat{p}_{u,t}^{(n)}} \quad (4)$$

is the Bhattacharyya coefficient between $\hat{\mathbf{q}}$ and $\hat{p}_t^{(n)}$. The particles weights are finally normalized and saved with the cumulative probability $c_t^{(n)}$ to form a triplet $(s_t^{(n)}, \pi_t^{(n)}, c_t^{(n)})$ for each particle. The area covered by the best particles in the image (i.e. the particles having the highest weights) represents a coarse estimation of the target state, and thus constitutes a reduced search space where keypoints will be detected and matched. The keypoints are detected and described in the overlapping region defined by the best particles as explained in the following section.

D. Target prediction and model adaptation

1) Fine state estimation: The second step of the tracking procedure relies on keypoints. These keypoints are the centers of salient patches located on the target face region. In our work, we use SIFT as keypoints detector and descriptor. SIFT features are reasonably invariant to changes in illumination, image noise, rotation, scaling, and changes in viewpoint between two consecutive frames [12], [16], [18]. SIFT keypoints are detected and described on the target region to be included to the initial appearance model. For each subsequent frame, keypoints matching will consider only the image region delimited by the best particles. SIFT keypoints are detected on the overlapping region defined by the N^* best particles. The 128 element descriptors are then computed for each keypoint to summarize the local gradient information.

By reducing the search region to the most important candidate particles, we avoid detecting the keypoints, computing the local descriptors and matching them on the entire image. The descriptors of the overlapping region are then matched with those of the target model based on the Euclidian distance. To do so, we construct a first set of candidate matches by selecting for each keypoint of the target model, the most similar keypoint detected in the search region in the current frame. In [15], it has been shown that the probability

that a match is correct can be determined by evaluating the ratio of distance from the closest neighbour to the distance of the second closest. For our algorithm, we keep only the matches in which the distance ratio from the closest neighbour to the distance of the second closest is less than 0.7. Given the final set of matched pairs and their locations, we use a mapping table that indicates the particles where each keypoint in the search region is located to compute N^* matching scores. Finally, the predicted target location is given by the particle having the highest matching score.

2) Appearance model adaptation: After the target model is built, tracking is made in subsequent frames using the procedure described above. To adapt the reference model to the latest appearance of the target, the color distribution and the keypoint descriptors set are updated every time a good prediction is achieved. Our definition of a good prediction is that 50% of the SIFT descriptors in the model are matched with the keypoints of a candidate particle. This avoids too large drifts of the model. Therefore, the reference color model is adapted every time we hypothesize that the prediction is relatively precise. For this, the color distribution of the predicted target $\hat{q}_{new}$ is computed, and the adaptation is made according to the equation:

$$\hat{q} = (1 - \alpha)\hat{q}_{old} + \alpha\hat{q}_{new}. \quad (5)$$

The learning factor α is determined automatically based on the quality of the new prediction and is defined as $\alpha = 0.5M$ with $M \in [0.5, 1]$ being the matching rate of the keypoint descriptors in the target model. Furthermore, we update the target keypoint model to include the newly detected features.

IV. Experiments

To evaluate the effectiveness of the proposed tracking algorithm, we performed two series of tests using two different datasets. The first set of video sequences was captured in our laboratory to evaluate the quality of the proposed algorithm when tracking faces in different scenarios. To compare with the state of the art results and evaluate quantitatively the tracking performance in a more general context with different types of objects, we also tested our method on a publicly available dataset used in the latest visual tracking works.

A. Qualitative evaluation

The first dataset includes seven scenarios. Seven video sequences were captured using one IP PTZ camera in a laboratory room. The room was cluttered with desks, chairs, and technical video equipment of various shapes and textures in the background. Lighting was uneven in the room. The Sony SNC-RZ50N camera used for capture was mounted on a tripod at a height of approximately 1.8 meters. The video frames are 320x240 pixels and were sent via IP network, at a frame rate of 15 fps, to a 3.4 GHZ Core i7-3770 CPU on which the processing is done.

Figure 1 shows tracking results on a few key frames for different scenarios. Each row in this figure corresponds to a video sequence where the selected video frames are numbered. The goal of scenario 1 (331-frame video) is to track the face of a person moving randomly in the room. The walking speed is varying abruptly and the person changes direction and orientation, with distances to the camera varying from 2 to 6m. After manually initializing the system, the face was successfully tracked in practically all the processed frames. We observed a decrease of tracking precision when the subject changes direction or turns its face quickly (frame 69). Nevertheless, the adaptive target model is robust enough to quickly recover a stable track after few frames (frame 80).

In scenario 2 (428-frame video), we test the robustness of our algorithm in the case of two moving persons. Here we track a subject that can cross in front or behind another walking or immobile person. As shown in the second row of figure 1, the algorithm can keep correct track of the target face, even though there are partial occlusions by another face. In the case where the target person's face is completely occluded by another face (no keypoint match), the system detects a total occlusion, and thus avoids tracking the occluding person's face, and therefore does not erroneously update the face model. This is mainly due to the target model keypoints that does not match with those of the occluding face.

In scenarios 3 (339-frame video) and 4 (357-frame video), we evaluate the tracking quality in case of change in pose and orientation, and severe changes of viewpoint. Although these changes can co-occur with fast lateral movements of the target, the tracking results of sequences 3 and 4 show that our tracker can handle such situations very well.

The last scenarios (5-7) test the ability of the system to handle different types of partial occlusion. In scenario 5 (321-frame video), the subject tries to hide behind a structure of the background. During the partial occlusion, the target continues moving laterally and the tracker successfully predicts its position in all the frames. In scenario 6 (304-frame video), the face is occluded by the person's hand, while in scenario 7 (532-frame video) a book is used to partially hide the target from different sides. In both cases, the tracker continues to correctly predict the target position without drifting, even when the face is severely occluded. This observation highlights the advantage of using a keypoint-based model.

B. Quantitative comparison with state of the art methods

While our proposed tracker was designed specifically to track human faces, a quantitative comparison with several state of the art trackers is presented in this section. We show that it can achieve very competitive results when tracking different types of objects. We tested our tracking system on challenging video sequences used in the latest works in

Figure 1: Tracking results of the proposed method for different scenarios.

visual tracking [19]–[21]. We also used publicly available ground truth data and experimental results provided by [19]. The performance comparison is made with two versions of the online AdaBoost algorithm (OAB) presented in [22]. The first version (OAB-1) generates one positive example per frame, while the second version (OAB-45) generates 45 image patches comprising one positive bag. To further evaluate our tracker performance, we also compare our results with three other algorithms: the SemiBoost tracker [23], FragTrack [24], and MILTrack [19]. Note that the experimental results for the compared methods are obtained by using the default parameters provided by the authors. In order to quantify several results, we consider the two metrics used in [19]:

- the precision at a fixed distance threshold, which is the percentage of frames where the tracker is within a certain distance of the ground truth;
- the average center location error of the tracker.

Since the proposed tracker is non-deterministic, the results presented in tables I and II are the averages over 5 runs. Also note that these video sequences are available in gray scale only. Thus, we adapted our algorithm to use intensity distributions instead of RGB color distribution.

The sequence *Sylvester* shows a moving stuffed animal undergoing pose variations, lighting changes, and scale variations. For this sequence, the OAB-45 tracker fails with only 11% of precision. The other trackers, including our, are able to track the target with a high accuracy. MILTrack has the best results while our tracker achieved the second best performance in terms of precision and average error.

The sequence *David indoor* shows the robustness of our algorithm when tracking a human face under severe camera motion, background and illumination changes, and large scale variations. Our tracker achieved the best precision of 80% and an average error of 26 pixels.

The results of *Occluded face* sequence show that FragTrack outperforms all the other methods because it is specifically designed to handle occlusions via a part-based model. Our tracker is also designed to handle occlusion, but we did not use yet the position of the keypoints in a particle to improve object localization. Our tracker achieved a good result, outperforming MILTrack, OAB-1, and OAB-45, with a 76% precision.

The sequence *tiger 1* exhibit many challenges, showing a stuffed tiger in many different poses, with frequent occlusion level, fast motion and rotations causing motion blur. With this sequence, our algorithm outperforms significantly the others in precision, having also the best average error. Figure 2 presents a few screenshots of tracking results. In general, the proposed tracker performed well for all the sequences. The results of tables I and II show that every tracker fails in at least one sequence. Nevertheless, our method outperformed all the other algorithms when averaging the precision and error results over all the experiments.

V. Conclusion

We developed a novel face tracking method that learns and updates the target model during the tracking procedure. It is based on coarse-to-fine state estimation that combines kernel and keypoint tracking. Our experiments show the

Video Sequence	OAB-1	OAB-45	SemiBoost	FragTrack	MILTrack	Ours
Sylvester	0.71	0.11	0.81	0.87	**0.98**	*0.90*
David indoor	0.36	0.18	0.51	0.50	*0.70*	**0.80**
Occluded face	0.32	0.04	*0.97*	**1**	0.71	0.76
Tiger 1	0.61	0.38	0.46	0.36	*0.89*	**0.93**
Average	0.50	0.18	0.69	0.68	*0.82*	**0.85**

Table I: Tracker precisions at a fixed threshold of 30: percentage of frames where the center of the predicted location is within 30 pixels of the ground truth. **Bold red** font indicates best results, *blue italics* font indicates second best.

Video Sequence	OAB-1	OAB-45	SemiBoost	FragTrack	MILTrack	Ours
Sylvester	25	79	16	**11**	**11**	*14*
David indoor	49	72	39	46	**23**	*26*
Occluded face	43	105	*7*	**6**	27	20
Tiger 1	*35*	57	42	39	**16**	**16**
Average	38	78.25	26	25.50	*19.25*	**19**

Table II: The average tracking errors: the error is measured using the Euclidian distance from the center of the predicted location to the center of ground truth. **Bold red** font indicates best results, *blue italics* font indicates second best.

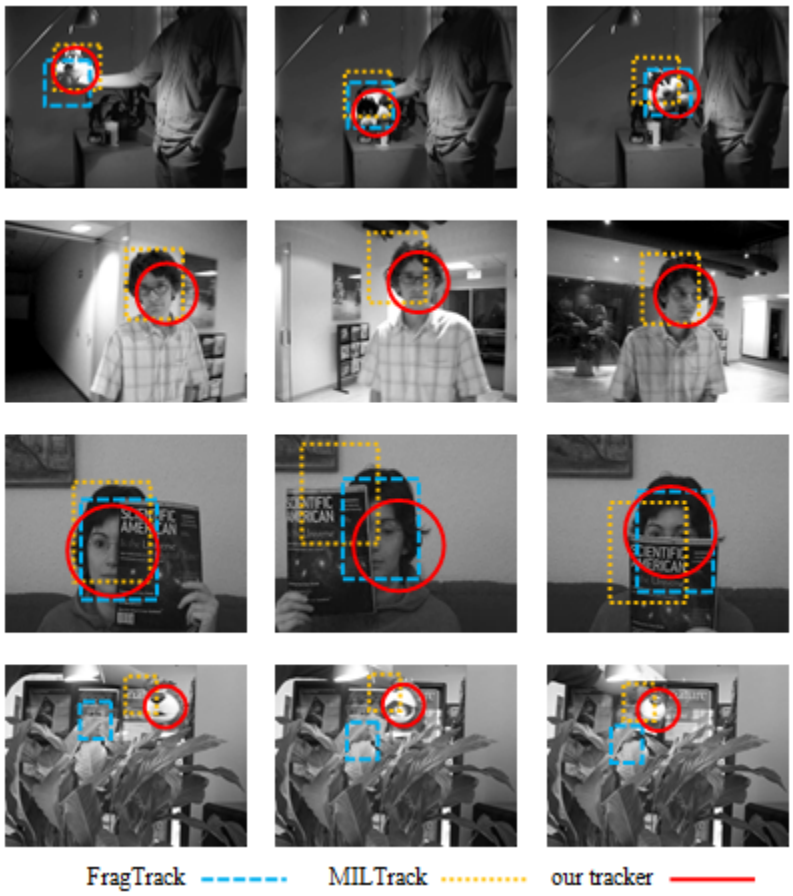

Figure 2: Screenshots of tracking results for FragTrack, MILTrack, and the proposed tracker. The rows correspond respectively to the video sequences *Sylvester*, *David indoor*, *Occluded face*, and *tiger 1*.

robustness of our algorithm and its competitiveness with the state of the art trackers when tracking human faces, or even other types of targets. As a future work, we aim to apply our tracking algorithm to online person tracking by active IP PTZ camera. In such a system, the camera should be controlled after each target prediction to keep the subject in the field of view. The camera control is closely related to the tracking algorithm. It is essential to reduce the complexity of the tracking algorithm to process the images and control the camera quickly, without losing the target. In our algorithm, the complexity can be controlled by three parameters: 1) the number of particles generated at each iteration, 2) the size of the subset of particles used for keypoint matching, and 3) the keypoint detector/descriptor used.

ACKNOWLEDGMENT

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 810–815, 2004.

[2] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[4] J. Kang, I. Cohen, and G. Medioni, "Object reacquisition using invariant appearance model," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 759–762.

[5] L. Guan, J.-S. Franco, and M. Pollefeys, "3d occlusion inference from silhouette cues," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[6] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with ssd," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–790.

[7] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1815–1821.

[8] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 263–270.

[9] M. Gouiffes, C. Collewet, C. Fernandez-Maloigne, and A. Trémeau, "Feature points tracking: robustness to specular highlights and lighting changes," *Computer Vision–ECCV 2006*, pp. 82–93, 2006.

[10] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.

[11] S. Hare, A. Saffari, and P. H. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1894–1901.

[12] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[13] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic, "Segmentation based particle filtering for real-time 2d object tracking," *Computer Vision–ECCV 2012*, pp. 842–855, 2012.

[14] H.-M. Sun, "Skin detection for single images using dynamic skin color modeling," *Pattern recognition*, vol. 43, no. 4, pp. 1413–1420, 2010.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," *Computer Vision–ECCV 2012*, pp. 759–773, 2012.

[17] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.

[18] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[19] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[20] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, 2012, vol. 7575, pp. 702–715. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33765-9_50

[21] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1830–1837.

[22] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. BMVC*, vol. 1, 2006, pp. 47–56.

[23] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," *Computer Vision–ECCV 2008*, pp. 234–247, 2008.

[24] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 798–805.