VISUAL TRACKING WITH ROBUST TARGET LOCALIZATION

Ilker Ersoy, Kannappan Palaniappan

Computer Science Department
University of Missouri-Columbia
Columbia, MO 65201 USA

Guna Seetharaman

U.S. Air Force Research Laboratory
Rome, NY 13441 USA

ABSTRACT

In this paper, we present a tracking method with robust target localization for tracking of visual objects. We use an adaptive appearance model that incorporates structural information to avoid drifts and can be updated incrementally using partial models. The proposed method works especially well for aerial surveillance sequences where the objects of interest are small and detecting robust feature points in a repeatable manner is difficult due to scale, blur and changing viewpoints. We compare our method using standard sequences and show results on aerial video sequences including wide-area motion imagery (WAMI).

***Index Terms*—** Visual tracking, surveillance, adaptive appearance model.

1. INTRODUCTION

Visual object tracking for surveillance applications poses challenges due to many factors such as the distractor objects in the scene, changing imaging conditions (e.g. illumination, viewpoint), scale, blur and appearance change. Many trackers in the literature [1, 2, 3] utilize adaptive models to keep up with the dynamic appearance of objects. While some trackers utilize adaptive templates, others utilize keypoint based models (e.g. visual bags of words). Keypoint based tracking methods usually rely on a keypoint detector and descriptor (such as SIFT or SURF) in order to detect points on an object that can be robustly and repeatably detected in the subsequent frames and describe the regions around them with a robust descriptor. For objects with enough support (large scale) this approach works well, but it suffers from the lack of good feature points in aerial surveillance and WAMI where objects are blurry and have small support. Template based methods usually perform better at small scale but may have difficulties with partial occlusions of the object. In both cases, careful consideration has to be given to the appearance model update method to avoid the drift problem. Drift occurs when the tracker updates (or learns) new object appearance with a poor localization of the object. The small localization errors accumulate in time and the tracker starts drifting and adapting more to the background or to a distractor. This becomes more of a challenge with the partial occlusions. In order to address the drift problem, two problems have to be solved; the accurate localization of the target, and a robust model update method that does not degrade with partial occlusions. We present a tracking method that addresses these problems and can be applied to surveillance sequences with a wide variety of scale. We utilize an approach similar to bag of words with the exception that the model does not rely on repeatable keypoint detection. Section 2 describes the appearance model with structural constraints, the matching algorithm and the update rule. Section 3 describes the experimental results. Section 4 concludes the paper.

2. DENSE REGION DESCRIPTORS AND APPEARANCE ADAPTIVE TRACKING

Several trackers in the literature utilize keypoint detection, where a set of interest points or image patches represented by descriptors are matched against the next frame in order to locate the object [3, 4]. In applications where objects have large enough support with distinct features, this approach works well. In aerial surveillance applications where cameras are far from potential targets, this approach poses challenges due to small support. We opt for detection based tracking, but propose a different approach using a clustered set of structured uniformly dense robust features (CSURF) to describe regions rather than finding interest points. This is due to the lack of prominent features that can reliably serve as unique interest points on small objects (about 20×30 pixels) from frame to frame. We create an adaptive appearance model with these dense descriptors with overlapping support to account for the uncertainties in a robust manner. Initiated with the object in the first frame, the tracker searches for local region matches in the next frame within a search window and the structural constraints of matching descriptors are utilized in a voting scheme to accurately detect the centroid of the object. Our choice of descriptors are the Speeded-Up Robust Feature (SURF) descriptors [5] but other robust descriptors of several features [6] can be also be employed. SURF and SIFT are very popular interest point detectors and descriptors that have many applications for image matching. They have been used in many tracking approaches due to their robust nature of finding unique interest points. Here, we only use the descriptor scheme of SURF, and not the interest point detector. Finding unique, reliable and repeatable interest points is very challenging and prone to mismatches for small objects. This problem was reported by Ma and Grimson [4] for even larger vehicles in a mid-field surveillance framework. By only borrowing the descriptor of SURF, we represent local image regions by this robust descriptor that is invariant to slight changes in illumination, scale and orientation. It is worth noting that scale-invariant nature of SURF is not a result of its descriptor but of its interest point detector. We do not utilize the scale-invariant feature descriptors. Similarly we do not utilize rotation-invariant type of SURF descriptor since it would produce too many false matches within a given search window because keypoint detection is not used. Based on these considerations, our model consists of a collection of 64 dimensional SURF descriptors with structural information that represent the local image patches around regularly spaced points on the support of the object at a fixed scale.

2.1. Object Model

Given a bounding box of size $m \times n$ that surrounds the object of interest at frame $t = 0$, a set of descriptors $\mathbf{D} = \{d(x_{ij})\}$ are

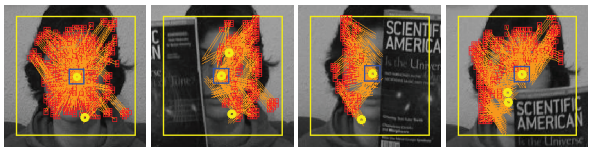

Fig. 1. Accurate object centroid localization under occlusion events through collective voting of matching descriptors. Small red squares show the locations of matching descriptors, orange lines show the tentative votes for the center of the object relative to each matching descriptor, large yellow circles show the cluster centers, large blue square is the center with the largest votes.

computed first where i and j are local coordinates with respect to the bounding box and d is the 64 dimensional SURF descriptor of the image patch around the points $\mathbf{X} = \{x_{ij}\}$ at a scale s:

$$\mathbf{X} = \{x_{ij} | i \in \mathbb{Z}_m, i \bmod \mathrm{q} = 0, j \in \mathbb{Z}_n, j \bmod \mathrm{q} = 0\} \quad (1)$$

so that $\mathbf{X}$ are sampled every q pixels in both directions in the bounding box. Given $\mathbf{X}$, $d(x_{ij})$ are computed as

$$\mathbf{D} = \{d(x_{ij}) | x_{ij} \in \mathbf{X}\} \quad (2)$$

Since computing SURF descriptors involves a $20s \times 20s$ image patch, there is a big overlap between the patches around points $\mathbf{X}$ for very small values of q. This redundancy provides the robustness in case of major occlusion. Overlapping patches produce similar descriptors especially for flat image patches or along the edges. To reduce the amount of computation as well as bias to such patches, we run a mean shift procedure on the set $\mathbf{D}$ to obtain natural clusters in the 64D descriptor space. Mean shift is used for clustering since it can detect the modes of a density function given a discrete sampling of that function. The obtained set of cluster centers $\mathbf{C} = \{c_k\}$ serve as the initial model. To impose structural constraints on this model, we also compute the relative coordinates $\mathbf{R} = \{r_{ij}\}$ of $\mathbf{X}$ with respect to the center $\mathbf{c}$ of the bounding box which is assumed to be the centroid of the object. Even though the clustering runs on the descriptor space, spatially close descriptors are assigned to similar clusters, hence for each cluster center c_k we compute the median of relative coordinates of all d_{ij} that belong to that cluster and store it. So the model becomes

$$\mathbf{V} = \{(c_k, r_k) | k \in \mathbb{Z}_K\} \quad (3)$$

where K is the number of clusters.

2.2. Matching Algorithm

Given this model and a search window SW around the current location p in the next frame $t+1$, detection process becomes finding the most likely position of the object centroid by matching the descriptors of the model to every possible search window descriptor d_{ij}^{SW} in SW with a similar scale s and grid length q. Matching is accomplished by computing the Euclidean distance between every pair of d_{ij}^{SW} and c_k, ranking the distances of all matches of c_k and computing the distance ratio of the best match to the next best match. Given $e_{ij,k}$ is the Euclidean distance between d_{ij}^{SW} and c_k, the distance ratio of the matching pair (ij, k) is

$$DR_{ij,k} = \frac{e_{ij,k}^{(1)}}{e_{ij,k}^{(2)}}. \quad (4)$$

If this ratio is smaller than a threshold, this match is considered a good match. The rationale behind this is to obtain only those matches that are substantially significant. If a c_k matches to many descriptors in SW with similar Euclidean distance, that particular c_k is not a good descriptor for the object for that frame, but it still remains in the model because it may serve as a better descriptor in a different search window (e.g. a descriptor that represents a particular edge on the object can match to many other descriptors in a given search window with similar edges, but not in an other one where there are no such edges in the background). Those matches (ij, k) that pass the ratio test are retained to compute the most likely location of the object.

$$\mathbf{M} = \{(ij, k) | DR_{ij,k} < T_{DR}\} \quad (5)$$

where the image patch around x_{ij} represented by the descriptor d_{ij}^{SW} is a good match to the model descriptor c_k. $\mathbf{M}$ is the set of matching pairs and its cardinality is expected to be high because of the redundancy in the model. Even though they may be dispersed through the search window, most of them are expected to be located on the object of interest in frame $t+1$. Instead of taking their absolute coordinates, we assign the relative coordinates r_k of each c_k to the corresponding match d_{ij}^{SW} and these vectors point to the hypothesized centroids. This, in fact, is a voting process where every significant match votes for a centroid. This way we enforce the structural constraints. This is a robust way to deal with the outliers due to sporadic matches. These votes form clusters in spatial domain which, again, can be discerned by the mean shift procedure. Practically, the mode of this distribution is taken as the most likely centroid of the object. Figure 1 shows an example of this process. The red small squares are the locations of matching d_{ij}^{SW} after the ratio test. The orange lines represent the relative coordinates r_k of the corresponding c_k that are assigned to (i, j). If indeed these are good matches, all of them should point towards the most likely location of the centroid. This also handles occlusions gracefully as shown in the same figure. As the vehicle gets occluded, the remaining matches vote towards to correct centroid, eliminating abrupt jumps or drifts in the location of the centroid. It is also likely that there may be very few good matches in a given search window. In that case, the tracker relies on a simple prediction rather than the weak matches to keep the search window on the most likely position of the car in the next frame. If there are no good matches for a predetermined number of frames (e.g. the object leaves the scene), the tracker can quit.

2.3. Robust Model Adaptation

It is clear that this model has to be adapted to changing appearance. To facilitate that, we pick the top best matches in every frame and replace c_k with their corresponding d_{ij}^{SW}. This ensures that the model is only partly adapted, retaining some of the old descriptors, but also quickly adapting to the best matches. For that we compute the average Euclidean distance of matches $\overline{e}_{ij,k}$, if it is less than a threshold, those matches that are better than this average replace the corresponding c_k. This process may still cause a drift in the tracker if there are very few matches in a search window which happen to be sporadic matches due to similar image patches in the background. This is specifically the case when the object is totally occluded yet there are a few good matches in the scene. To avoid that, we also check the number of matches to make sure there are enough good matches to update the model.

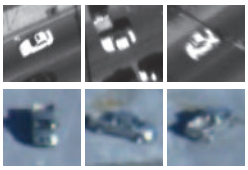

Fig. 2. Small targets undergoing appearance changes.

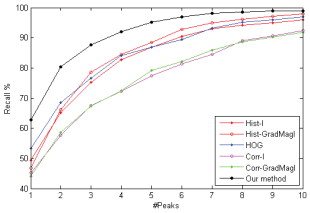

Fig. 3. Average recall versus number of peaks detected for each feature descriptor.

3. EXPERIMENTAL RESULTS

In this section, we describe the results of the tracker on aerial and WAMI sets as well as standard videos. After extensive experimentation, the following model parameters were selected: scale $s = 2$, grid length $q = 2$ for aerial images, $s = 5$, $q = 3$ for standard data sets, DR threshold $T_{DR} = 0.75$ for all sequences. The thresholds to update the model are set based on the distance of the camera to the objects, this is basically a function of image resolution, the same tracker can be used for different resolutions ranging from WAMI data [7] to webcams as in standard benchmark sets. We use the same set of thresholds for all standard data sets, and similarly another set of thresholds for all aerial data sets since the scale difference between these two sets are drastic. Persistent wide-area motion imagery as described in [7] contain vehicles with small support (20×30) undergoing drastic appearance changes due to the low video frame rate (1 fps) and moving camera platform (see Figure 2 top). Our tracker can track these vehicles and update appearance model in order to avoid drift. For this type of videos, a simple Kalman prediction model is used for search window localization. Figure 4 shows other aerial surveillance videos with similar size of support. In these sequences, camera is mounted on a low-flying craft and no stabilization or Kalman prediction is necessary since the the camera follows the scene of the vehicles and the frame rate is high (20-30 fps). Our tracker can keep up with tracked objects and updates the appearance until there is a sharp change in the appearance that the update scheme cannot cope with. In that case the tracker starts tracking without updates until to the point of no more significant matches. Unlike many other tracking approaches [8, 9, 10] in the visual surveillance literature, our approach does not require stabilized background or moving object detection which makes it suitable for challenging WAMI data. A detailed analysis for WAMI sequences is reported in [11].

3.1. Target Detection Performance in WAMI

In this experiment we evaluate the target detection performance of the proposed tracker independent of the influence of other steps in a similar way as reported by Palaniappan et al. [6]. We compute likelihood maps for a given a search window centered on the target, and compare the detection performance to a set of two block correlation-based features and three local histogram-based features. Intensity and gradient magnitude normalized cross correlation (Corr-I and Corr-GradMagI) are chosen for block correlation-based features. Local histogram based features are local intensity histogram (Hist-I), local intensity gradient magnitude histogram (Hist-GradMagI) and histogram of oriented gradients (HOG). For this experiment, target likelihood maps in a WAMI surveillance sequence [7] are computed by using a sliding window histogram differencing scheme. Local maxima in likelihood maps are considered as possible detected target locations. These peaks are ranked with respect to their likelihood and recall is computed by evaluating their distance to the actual object centroid location. Figure 3 shows the aggregate detection performance. As shown in the figure, the proposed scheme of dense descriptors outperforms other five features.

3.2. Tracking Performance in Standard Videos

To evaluate our tracker with standard benchmark sets and compare to other trackers, we use the tables reported in [2] and [3]. The scores are computed as the mean Euclidean distance between ground truth centroids and tracker-produced centroids, and the best results of five runs are given as in [2]. Table 1 shows the performance of our tracker compared to the results in [3]. Our tracker can handle a small degree of rotation and scale change even though we use a single scale for upright SURF descriptors. In general, it performs well in these sequences due to the accurate localization of object centroids as in Figure 1. It temporarily loses the object in 'david' sequence due to large rotations, scale and illumination changes, but on average it still scores competitively with other trackers, since accurate centroids on most frames compensate for loss of target in a few frames. To the disadvantage of our tracker, we do not use the whole image as the search window unlike other approaches. When the target moves out of the search window, the tracker cannot reacquire it until it reappears in the search window.

4. CONCLUSION

We propose a tracker to address the drift problem by incorporating the structural constraints in a robust appearance update scheme. The proposed CSURF tracker works well with different scales of aerial and WAMI surveillance sequences as well as standard videos. It shows competitive performance in comparison to the state of the art techniques reported in the literature. Using a dense set of region descriptors adds robustness to the object model since there are no discernible interest points that an interest point detector can reliably and repeatably find at such small scales. By using a model that can be partially updated, CSURF tracker can still update the appearance in occlusions without degrading the model. The structural constraints

Table 1. Comparison using standard data sets.

Sequence	OAB[12]	FragT[13]	MILT[1]	PROST[2]	NN[3]	CSURF
Girl	43.3	26.5	31.6	19.0	18.0	**13.1**
David	51.0	46.0	15.6	**15.3**	15.6	15.4
Faceocc1	49.0	6.5	18.4	7.0	10.0	**6.3**
Faceocc2	19.6	45.1	14.3	17.2	**12.9**	16.5

Fig. 4. Samples of tracking in aerial sequences. Top: balloon3 sequence, frames 17, 200, 400, 500. Middle: egtest01 sequence, frames 1, 150, 250, 350. Bottom: hollywood sequence, frames 100, 450, 900, 1400.

accurately localize the target in order to avoid the drift problem. The detection scheme, by utilizing the number and match similarity of the best matching descriptors, evaluates the tracking quality and decides for partial updates of the model. Large number of overlapping region descriptors lead to many matches that makes the target localization more robust by incorporating large number of votes. CSURF tracker can run at 2-5 FPS on a quad-core processor including all SURF computations, matching, update and disk IO for all the reported sequences. One significant failure scenario is a sharp turn of an object in a low frame rate sequence where the appearance model can not adapt fast enough. Since we chose not to use rotationally invariant descriptors to avoid high number of sporadic matches, we will address this by incorporating motion models that explicitly account for rotations. Another direction is to employ heterogeneous descriptors to utilize other features for added robustness.

5. ACKNOWLEDGMENTS

This research was partially supported by U.S. Air Force Research Laboratory (AFRL) under agreement AFRL FA8750-11-C-0091. Approved for public release (case 88ABW-2012-0926). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of AFRL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

6. REFERENCES

[1] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE CVPR*, 2009, pp. 983–990.

[2] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *Proc. IEEE CVPR*, 2010, pp. 723–730.

[3] S. Gu, Y. Zheng, and Carlo Tomasi, "Efficient visual object tracking with online nearest neighbor classifier," in *10th Asian Conf. on Computer Vision*, New Zealand, Nov. 2010.

[4] Xiaoxu Ma and W. Eric L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. IEEE ICCV*, 2005, vol. 2, pp. 1185–1192.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.

[6] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. Rao, and G. Seetharaman, "Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video," in *13th Int. Conf. Information Fusion*, 2010.

[7] K. Palaniappan, R. Rao, and G. Seetharaman, "Wide-area persistent airborne video: Architecture and challenges," in *Distributed Video Sensor Networks: Research Challenges and Future Directions*, B. Banhu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds., pp. 349–371. Springer, 2011.

[8] N. P. Cuntoor, A. Basharat, A. G. A. Perera, and A. Hoogs, "Track initialization in low frame rate and low resolution videos," in *Proc. IEEE ICPR*, 2010, pp. 3640–3644.

[9] F. Bunyak, K. Palaniappan, S.K. Nath, and G. Seetharaman, "Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking," *Journal of Multimedia*, vol. 2, no. 4, pp. 20–33, 2007.

[10] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *Proc. 11th European Conf. Computer Vision*, 2010, pp. 186–199.

[11] Guna S. Seetharaman Ilker Ersoy, Kannappan Palaniappan and Raghuveer M. Rao, "Interactive target tracking for persistent wide-area surveillance," in *Proc. of SPIE Volume 8396: Geospatial Infofusion II*, 2012.

[12] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE CVPR*, 2006, vol. 1, pp. 260–267.

[13] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE CVPR*, 2006, vol. 1, pp. 798–805.