Clustering of Static-Adaptive Correspondences for Deformable Object Tracking

Georg Nebehay[1,2]
[1]Institute for Computer Graphics and Vision
Graz University of Technology
gnebehay@gmail.com

Roman Pflugfelder[2]
[2]Digital Safety and Security Department
Austrian Institute of Technology
roman.pflugfelder@ait.ac.at

Abstract

We propose a novel method for establishing correspondences on deformable objects for single-target object tracking. The key ingredient is a dissimilarity measure between correspondences that takes into account their geometric compatibility, allowing us to separate inlier correspondences from outliers. We employ both static correspondences from the initial appearance of the object as well as adaptive correspondences from the previous frame to address the stability-plasticity dilemma. The geometric dissimilarity measure enables us to also disambiguate keypoints that are difficult to match. Based on these ideas we build a keypoint-based tracker that outputs rotated bounding boxes. We demonstrate in a rigorous empirical analysis that this tracker outperforms the state of the art on a dataset of 77 sequences.

1. Introduction

Keypoints are one of the most widely used representations for objects in computer vision [12, 15, 19]. The main idea of keypoints is to break down the object into individual parts that are easier to match to a descriptor database than a holistic representation of the object. While matching is error-prone due to similar descriptors on the object and background clutter, robust methods such as RANSAC [4] are often used to evaluate the fitness of the matches to a motion model. Typically, strong assumptions are made in this motion model, of which the rigidity assumption probably is the most common one. However, one of the biggest challenges in object tracking is the deformation of the object of interest, often invalidating this assumption.

In object recognition, a strand of research has emerged recently studying how to incorporate spatial constraints into the matching of keypoints in addition to photometric constraints [23, 3, 9], aiming at replacing the strong assumptions by more flexible ones, allowing for deformations to be handled. For instance, Cho et al. [3] cluster correspondences by means of a dissimilarity measure that incorporates geometric constraints. Specifically, the clustering of correspondences has the appealing property that connected correspondences act as mediators.

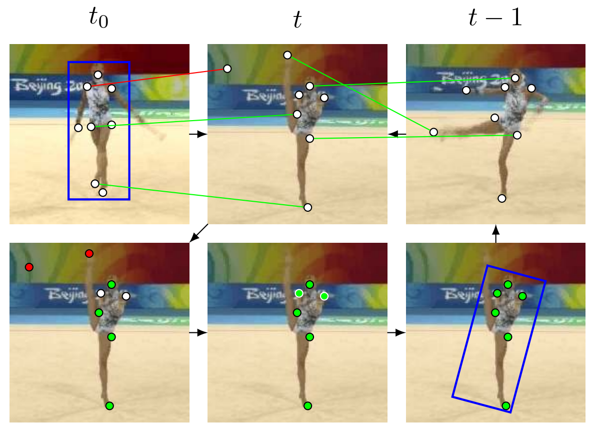

Figure 1. Top row: From the initial bounding box in frame t_0 static keypoints are extracted. Both the static keypoints and adaptive keypoints from frame $t-1$ are matched to the current frame t. Bottom row: Based on a geometric dissimilarity measure, the correspondences are clustered into inliers and outliers (left) and ambiguous matches are resolved (middle). A rotated bounding box (right) is computed as the algorithmic output.

The primary contribution of our work is the formulation of a novel dissimilarity measure for the clustering of correspondences. By applying standard hierarchical clustering techniques [22], we achieve a partitioning of the correspondences into inliers and outliers. The second contribution is a novel method for establishing correspondences based on the insight that static and adaptive correspondences complement each other as they stem from opposite ends of the adaptivity spectrum. The third contribution is a novel method for disambiguating matches that is based on our proposed dissimilarity measure.

The combination of our individual contributions forms a simple tracking algorithm that stands in stark contrast to the state of the art, as newly discovered appearance information is not incorporated permanently into the object model. The outline of our approach is shown in Figure 1. We show in an extensive evaluation that this tracker is able to outperform state-of-the-art methods by a considerable margin on a dataset of 77 sequences. Another advantage of our approach is its independence on the actual keypoint detector and descriptor, making it possible to employ a wide variety of existing methods. In our implementation we employ binary descriptors, making our tracker competitive to the state of the art with respect to computational demands. In this work, we use the terms match and correspondence interchangeably.

2. Related Work

Part-based tracking In order to address the deformation of a-priori unknown objects, a number of part-based tracking approaches have been proposed, allowing individual parts different degrees of freedom. In what can be considered a very basic form of a part-based model, Adam et al. [2] propose an approach where each cell of a pre-defined grid represents a part. Each part votes independently for the object position in a sliding-window manner by comparing its histogram to the corresponding image patch histogram. While the approach is straightforward, it was one of the first methods to demonstrate the effectiveness of part-based approaches. In [7] an approach is proposed where points are sampled on a regular grid on the object of interest in frame $t-1$. Each point of a cell is then tracked independently to frame t by estimating its optic flow. An appearance-based error measure of the immediate surrounding of the points and a forward-backward-measure is employed to identify erroneous points. The remaining points are then used to compute the transformation from frame $t-1$ to frame t. Vojir et al. [20] modify this approach in order to allow the points to move within a predefined area. Only when they leave their cell, they are re-initialized to their original position. Pernici and Del Bimbo [17] combine matched keypoints in a RANSAC-like voting scheme. In summary, there seems to be a tendency that approaches allowing the individual parts as much freedom as possible perform better than approaches where the parts are fixed, but at the same time error-correcting measures need to be taken in order to identify erroneous parts. Our approach differs from the aforementioned methods in that we see the problem of finding correct matches mainly as one of clustering geometrically similar correspondences, a technique that has seen application in object recognition.

Keypoint matching The straightforward way of matching keypoints between two images is to compare the keypoint based on an appearance-based distance metric. [19]. While this method is simple and effective for many use-cases, it contains the inherent problem that photometric matching alone is unable to resolve ambiguities, for instance when there are multiple similar or even identical descriptors. As a remedy, researchers have come up with ways of incorporating geometrical constraints into keypoint matching, where one of the most widely used methods is RANSAC [4]. In RANSAC, the central idea is to repeatedly fit a minimal solution using a random subset of the data to a motion model and identify inliers based on whether they agree to this solution based on an error measure. Depending on the problem, motion models of varying complexity ranging from simple translational models to projective transformations are used. However, usually a global motion model is used, making it impossible to model deformations. In graph matching [9, 23], complex geometric relations among multiple features are being modeled. While an improvement over appearance-based keypoint matching was shown, its biggest drawback is the high computational cost. The approach that was most influential for ours is the method of Cho et al. [3], who cluster correspondences by employing a geometric dissimilarity measure that takes into account the reprojection error between correspondences based on the information reported from affine region detectors. Our dissimilarity measure differs considerably from their approach, as we measure the compatibility to a global similarity transformation. One of the major problems reported by clustering in object recognition is the correct setting of the number of clusters, as it directly corresponds to the number of objects found in an image. As in single-target object tracking there is exactly one object, this allows us to instead focus on the major cluster only.

Model update A recurring question in object tracking is how to update the model so that it remains a good representation of the object of interest [14], a question closely related to the stability-plasticity dilemma [1]. Different paradigms have been explored in the tracking literature to cope with this issue. Santner et al. [18] propose to employ multiple tracking methods from the whole adaptivity spectrum and overrule the more adaptive trackers by more static ones after failure. Kalal et al. [8] present a successful tracker that employs an adaptive component in order to mine positive and negative training examples for an object detector, combined with a conservative update scheme. In [17] multiple descriptors of weakly aligned keypoints are collected over time and features that match to clutter are removed from the model. We follow the principal idea of [18] and employ one completely static and one completely adaptive method for establishing an initial set of correspondences.

3. Approach

The tracking problem is defined by a bounding box b_0 in the first frame of a video sequence. In this region, we detect a set of initial keypoints $P_0 = \{x_1^0, \dots, x_m^0\}$. Without loss of generality, we mean-normalize the keypoint coordinates in P_0. A match m_i is a feature correspondence $m_i = (x_i^0, x_i^t)$, with x_i^t denoting the position of x_i^0 in frame

t. In each frame t, our aim is to identify the set of matches $\mathcal{L}_t = \{m_1, \dots, m_n\}$ that represents the object of interest as accurately as possible.

3.1. Static-Adaptive Correspondences

We employ a static appearance model that is based solely on the initial appearance of the object, composed of the descriptors around all $x_i^0 \in P_0$. We refer to matches deduced from this model as static correspondences. As the time between the initial frame and the current frame can become arbitrarily large, purely appearance-based methods have to be used to establish correspondences. We employ a global search in order to establish matches between keypoints x_i^0 from the initial frame and candidate interest points x_j^t in the current frame by enforcing a threshold as well as the second nearest neighbor distance criterion [12] on the distance $d(.,.)$ between their descriptors:

$$d(x_i^0, x_j^t) < \theta \wedge \frac{d(x_i^0, x_j^t)}{d(x_i^0, x_k^t)} < \gamma, j \neq k. \qquad (1)$$

Additionally, we exclude candidate keypoints that match to a background descriptor in the first frame. The static model is robust and handles for instance the re-detection of keypoints after occlusions. However, it does not adapt to new object appearances.

In contrast, our adaptive model is updated in every frame, comprising the image patches around all $x_i^{t-1} \in \mathcal{L}_{t-1}$. While for the static model a global search is necessary, for the adaptive matches we assume that the time between two frames is small. By estimating sparse optic flow from frame $t-1$ to frame t, we establish correspondences efficiently by means of a local optimization [13]. Additionally, we employ a forward-backward error measure [7] in order to filter out erroneous correspondences. Similar to [18], we overrule adaptive correspondences by static ones when both models yield a result as the latter are not affected by drift. In the following, the combined correspondences are referred to as $\mathcal{L}_t^*$.

3.2. Correspondence Clustering

The central idea of our approach is to employ a pairwise dissimilarity measure D between correspondences m_i and m_j based on their geometric compatibility, directly reflecting the deformation of the object of interest. As depicted in Figure 2, we define D to be

$$D(m_i, m_j) = \left\|(x_i^t - Hx_i^0) - (x_j^t - Hx_j^0)\right\|, \qquad (2)$$

where $\|.\|$ denotes the Euclidean distance and H is a similarity transform that is estimated from $\mathcal{L}_t^*$, which will be described later. Note that D is invariant to translations of x_i^t and x_j^t by a common displacement vector. It is therefore sufficient to estimate H up to scale and rotation.

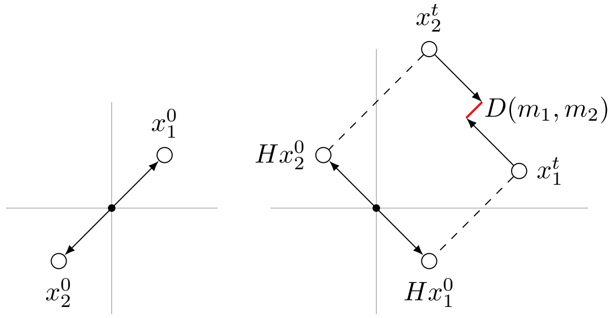

Figure 2. Based on an estimated similarity transform H, the initial keypoints are transformed into the coordinate system of the current frame and used to compute the dissimilarity measure D.

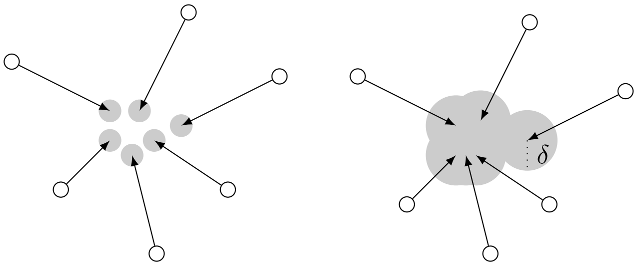

Figure 3. The degree of tolerated deformation is steered by the parameter δ. Left: δ is small, leading to all keypoints being identified as outliers. Right: δ is large enough to correctly recognizing all inliers.

D is then used to partition $\mathcal{L}_t^*$ into subsets by employing a standard agglomerative clustering algorithm [22] using single linkage, where a cutoff threshold δ is used in order to form flat clusters. We assume that the largest cluster $\mathcal{L}_t^+$ contains the correspondences relevant for the object, while correspondences of all other clusters belong to clutter. We would like the reader to appreciate that the parameter δ steers the degree of tolerated deformation, where 0 means complete rigidity, as shown in Figure 3. An appealing property of agglomerative clustering is that inliers are propagated, meaning that individual correspondences in $\mathcal{L}_t^+$ may be dissimilar, as long as there are sufficient mediating parts in-between.

We adopt existing heuristics for estimating s and α, both of which compute a robust statistic over pairwise geometric properties of pair of estimates, denoted by indices i, j with respect to their initial constellation. An estimate for the scale s as proposed by [7] is

$$s = \text{med}\left(\left\{\frac{\|x_i^t - x_j^t\|}{\|x_i^0 - x_j^0\|}, i \neq j\right\}\right), \qquad (3)$$

where med denotes the median. Section 4.1 provides evidence that this heuristic is able to correctly identify the current scale of the object of interest. As proposed by [16], an

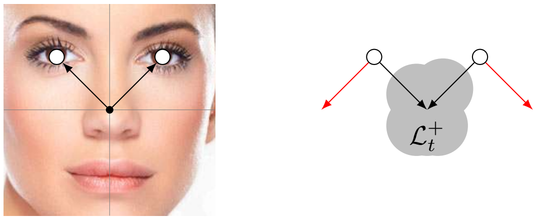

Figure 4. Left: Keypoints with similar descriptors are difficult to match based solely on their appearance. Right: We disambiguate these keypoints by excluding candidate correspondences that are geometrically dissimilar from the correspondences $\mathcal{L}_t^+$.

estimate for the rotation α is obtained by $\alpha =$

$$\text{med}\left(\left\{\text{atan2}(x_i^0 - x_j^0) - \text{atan2}(x_i^t - x_j^t), i \neq j\right\}\right), \quad (4)$$

where atan2 computes the angle in the appropriate quadrant by means of the arctangent. Depending on the number of correspondences n in $\mathcal{L}_t^*$, the algorithmic complexity is in $\mathcal{O}(n^2)$ for Eq. 3 and Eq. 4 as well as for the agglomerative clustering. Selection algorithms are able to compute the median in linear time. In our experience n typically ranges in the order of 50-250, not posing any computational problems,

3.3. Disambiguation of Correspondences

Similar descriptors appearing on multiple parts of the object or in the background pose a major problem in descriptor matching, as shown in Figure 4. We disambiguate these correspondences by excluding candidate keypoints that are geometrically dissimilar to $\mathcal{L}_t^+$ in a second matching round. The keypoint disambiguation aims at (a) improving the quality of the keypoints used for computing the algorithmic output and (b) enlarging the matching database for adaptive correspondences in the next frame.

Instead of matching the ith keypoint in P_t to the whole static model P_0, we match only to the subset

$$P_0^i = \{x_j^0 \mid \min_{m_k \in \mathcal{L}_t^+} D((x_j^0, x_i^t), m_k) < \delta\}, \quad (5)$$

which comprises all candidate correspondences exhibiting a dissimilarity to $\mathcal{L}_t^+$ of less than δ. We employ the same matching criteria as was presented in Section 3.1. $\mathcal{L}_t^+$ augmented with the disambiguated correspondences constitutes the final set of correspondences $\mathcal{L}_t$, on which the algorithmic output is based.

3.4. Algorithmic Output

Cho et al. [3] compute the convex hull of $\mathcal{L}_t$ as algorithmic output. As the de-facto standard for tracker output are (potentially) rotated bounding boxes, we instead employ the following heuristic. We compute an estimate for the object center μ by averaging

$$\mu = \frac{1}{|\mathcal{L}_t|} \sum_{m_i \in \mathcal{L}_t} \left(x_i^t - Hx_i^0\right). \quad (6)$$

A rotated bounding box is then obtained by applying (μ, s, α) as a similarity transform to the initial bounding box b_0.

4. Experiments

For our experiments, we detect and describe interest points by using BRISK [10], due to their invariance to scaling and rotation. BRISK uses binary descriptors, leading to the Hamming distance as a natural distance metric between descriptors. We employ the pyramidal variant of Lucas and Kanade [13] for estimating the optical flow. Unless noted otherwise, we employ the parameters settings $\delta = 20$, $\theta = 0.25$ and $\gamma = 0.8$. We implemented our approach[1] in C++. All of the following experiments were performed on an Intel Core i7 CPU 970 with a clock speed of 3.20GHz.

For both the quantitative and qualitative assessment of tracking performance we employ the tracking dataset[2] of Vojir et al. [20] that is composed of 77 sequences. The sequences are a compilation of datasets that have been widely used in the evaluation of various tracking approaches. The dataset is diverse with respect to different object classes, camera viewpoints, sequence lengths and challenges, such as partial and full object occlusions and disappearance of the object of interest. Most of the objects of interest in this dataset are non-rigid, thus rendering it suitable for evaluating our approach. Groundtruth data is available for each frame.

We compare tracker output b_T to ground truth data b_{GT} using the standard overlap measure

$$\phi(b_T, b_{GT}) = \frac{b_T \cap b_{GT}}{b_T \cup b_{GT}}. \quad (7)$$

When a threshold is imposed onto Equation 7, each frame in which the object is visible can be interpreted as a true positive (TP) or as a false negative (FN). By computing

$$\text{recall} = \frac{TP}{TP + FN} \quad (8)$$

an overall measure for a set of frames is given. We adopt the recent use of success plots [21, 16], where a performance metric is shown on the x axis and the rate of frames/sequences that achieve at least this value (the success rate) is plotted on the y axis.

[1] Available at http://www.gnebehay.com/cmt

[2] Available at http://cmp.felk.cvut.cz/~vojirtom/dataset/

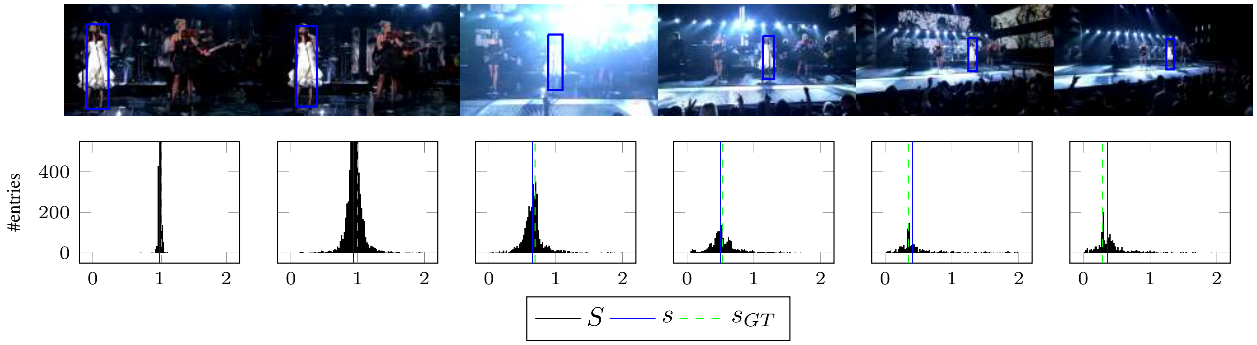

Figure 5. Distribution of the pairwise changes in scale S for 6 individual frames. The x axis denotes the scale. The y axis denotes the absolute number of entries in the respective histogram bin. s and s_{GT} denote our estimate and ground truth values for scale, respectively.

4.1. Heuristic for Scale Estimation

In the first experiment, we evaluate the suitability of the heuristic in Section 3.2 for estimating the scale s of the object. To this end, we analyze the distribution of the pairwise scale changes

$$S = \left\{ \frac{\|x_i^t - x_j^t\|}{\|x_i^0 - x_j^0\|}, i \neq j \right\} \tag{9}$$

that appears in Equation 3 on the sequence *singer* where the object of interest undergoes a considerable change in scale. S is shown as a histogram in Figure 5 for 6 individual frames together with the scale estimate s and the scale information extracted from the rectangular groundtruth information. We use

$$s_{GT} = \frac{\sqrt{w_t^2 + h_t^2}}{\sqrt{w_0^2 + h_0^2}} \tag{10}$$

as an approximation for the object scale, with w and h denoting the width and height of the bounding box, respectively. Clearly, s and s_{GT} differ only slightly, strengthening our choice of this heuristic.

4.2. Effect of Cutoff Threshold

In order to quantitatively assess the effect of the parameter δ on the performance, we vary δ from 0 to 100 while measuring average recall on the dataset. The results are shown in Figure 6 for three different thresholds $\phi > 0.25, \phi > 0.5$ and $\phi > 0.75$. This experiment shows that on average the performance monotonically increases to a broad saddle and slowly starts to decrease for larger values of δ. This experiments demonstrates that an appropriate setting of δ is important. In further tests we could not observe a correlation between the object size in pixels and the optimal parameter setting. Our explanation for this effect is that the optimal settings highly depend on the specific deformations the object of interest undergoes.

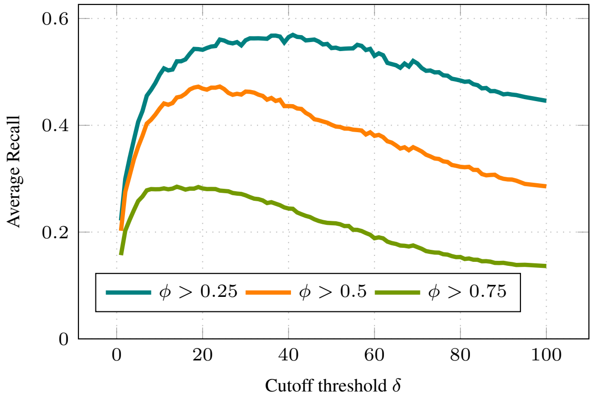

Figure 6. Effect of the cutoff threshold δ on the average recall computed on the Vojir dataset for different thresholds on the overlap measure ϕ.

4.3. Comparison to Baselines

We implemented RANSAC and a variant of the Hough transform in order to investigate the performance difference between our method and well-established methods for robustly estimating outliers and transformation parameters. For RANSAC, we compute an exact solution for a similarity transform between the correspondences. We tested a range of different parameter settings and employed those that yielded best results for our comparison. For the Hough Transform, we employ coarse bins of a tenth of the width and height of the image for the x and y dimension, respectively as well as 10 bins for the scaling dimension and 20 bins for the rotation dimension. It has to be noted that we added the comparison to the Hough transform for reasons of completeness, as it is not practical when used in more than two dimensions. The results in Figure 7 were computed on the Vojir dataset, showing the success rate with respect to recall. The results show that the restrictive baselines perform poorly compared to our approach, the main reason being their inherent incapability of handling deformations.

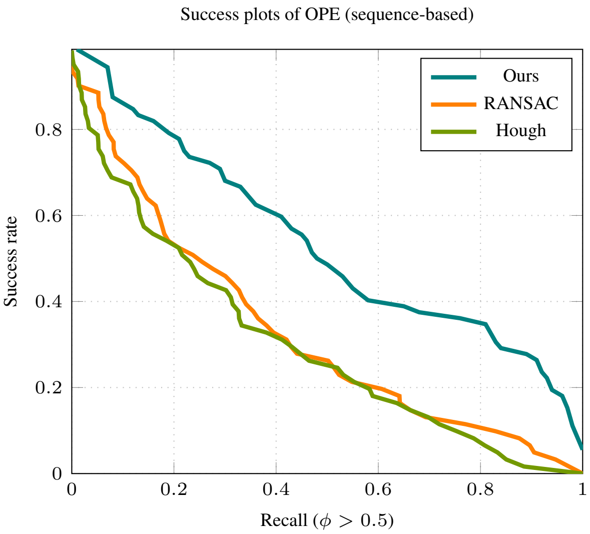

Figure 7. Comparison of RANSAC, Hough transform and our method on the Vojir dataset.

Abbrev.	Method
STR	Structured Output Tracking [5]
TLD	Tracking-Learning-Detection [8]
SCM	Sparsity-based Collaborative Model [25]
FT	Fragments-based Tracking [2]
CT	Compressive Tracking [24]

Table 1. Tracking algorithms used for comparison.

4.4. Comparison to State of the Art

For providing a quantitative comparison to the state of the art in tracking, we obtained the source code of the trackers shown in Table 1 and ran them on the Vojir dataset. The selected trackers contain the top 3 ranking trackers [25, 5, 8] from a recent tracking performance evaluation [21], a basic parts-based tracker [2] and a tracker that operates at high processing speeds [24].

Wu et al. [21] set a standard for evaluating tracking approaches that has seen broad adoption. We perform their one-pass evaluation (OPE), where each tracker is initialized using the bounding box of the first ground truth entry. As suggested by Wu et al., we show a success plot based on the overlap measure in the left plot of Figure 9. The plot depicts the distribution of the overlap measures of all individual frames in the dataset. Our algorithm outperforms all other trackers for an overlap $\phi < 0.8$ and is equal to SCM for $\phi \geq 0.8$. However, as the Vojir dataset contains sequences of varying length, the long-term sequences are overrepresented in the evaluation. Also, while the overlap measure has desirable theoretical properties [6], it is biased by the subjective ground truth annotation [11]. In order to overcome these limitations, we employ another evaluation method proposed by [16], where instead the per-sequence recall is reported in a success plot. This way, each sequence contributes equally to the plot, regardless of its length and errors in the human annotations are compensated for up to a certain extent. The success plot is shown in the right plot of Figure 9, where it can be seen that our algorithm performs especially well in the segment of excellent tracking results (recall > 0.8). The overall evaluation demonstrates that our algorithm is applicable to a wide variety of object classes and scenarios and outperforms the state of the art on the Vojir dataset. Qualitative results of selected sequences are shown in Figure 10.

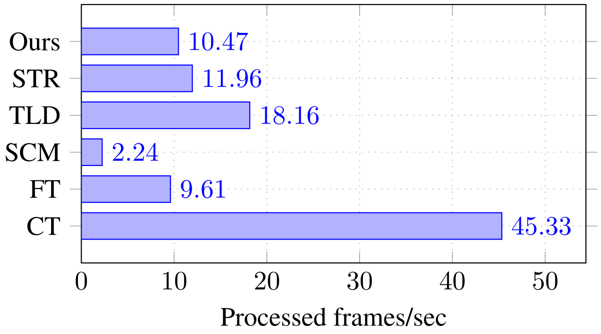

Figure 8. Average number of frames processed per second on the Vojir dataset. Our algorithm compares favorably to the state of the art.

To assess the speed of the considered algorithms, we measured the time spent on computing the output of each frame and report the the average frame rate over the whole dataset in Figure 8. While the real-time tracker CT achieves superior results with respect to computational demands, our method ranks close to STR, demonstrating an excellent performance-speed ratio. The keypoint matching and the clustering are responsible for the majority of the computational demand in our method.

5. Conclusion

We proposed a keypoint-based tracking algorithm that employs the clustering of correspondences as the central idea of distinguishing between inlier and outlier keypoints. The reason why our approach improves on state-of-the-art tracking results lies in the the flexible nature of the hierarchical clustering algorithm, allowing for the propagation of inliers, even when correspondences are located on deformed parts of the object. The evaluation demonstrated clearly that our algorithm is highly successful on a diverse dataset, strongly suggesting an application to real-world scenarios. An interesting future research direction lies in finding the optimal value for the cutoff threshold in an automated fashion during processing. We also plan to exploit the parallelizability of the clustering and the keypoint matching to improve computation time.

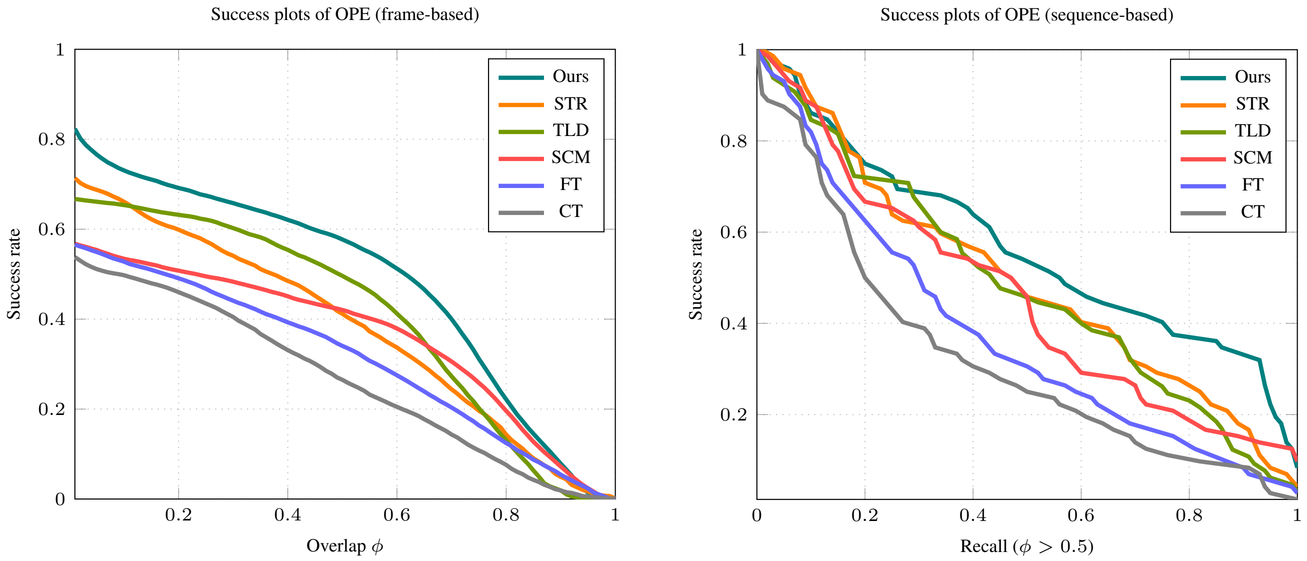

Figure 9. Results of the OPE experiment on the Vojir dataset. Left: Success plot of the overlap measure, computed over all frames. Right: Success plot of recall, computed over all sequences. Our method dominates both evaluations.

Acknowledgments

This work was supported by the research initiative 'Mobile Vision' with funding from the AIT Austrian Institute of Technology and the Austrian Federal Ministry of Science, Research and Economy HRSM programme (BGBl. II Nr. 292/2012) and from the Austrian Research Promotion Agency (FFG) project PAMON no 835916.

References

[1] W. C. Abraham and A. Robins. Memory retention–the synaptic stability versus plasticity dilemma. *TINS*, 28(2), 2005.

[2] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006.

[3] M. Cho, J. Lee, and J. Lee. Feature correspondence and deformable object matching via agglomerative correspondence clustering. In *ICCV*, 2009.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus. *CACM*, 24(6), 1981.

[5] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[6] B. Hemery, H. Laurent, and C. Rosenberger. Comparative study of metrics for evaluation of object localisation by bounding boxes. In *ICIG*, 2007.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-Backward Error: Automatic Detection of Tracking Failures. In *ICPR*, 2010.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-detection. *TPAMI*, 34(7), 2012.

[9] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, 2011.

[11] T. List, J. Bins, J. Vazquez, and R. B. Fisher. Performance evaluating the evaluator. In *PETS*, 2005.

[12] D. G. Lowe. Distinctive image features from Scale-Invariant keypoints. *IJCV*, 60(2), 2004.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[14] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *TPAMI*, 26(6), 2004.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10), 2005.

[16] G. Nebehay and R. Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *WACV*, 2014.

[17] F. Pernici and A. Del Bimbo. Object tracking by oversampling local features. *TPAMI*, 36(12), 2014.

[18] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Parallel robust online simple tracking. In *CVPR*, 2010.

[19] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *FTCGV*, 3, 2008.

[20] T. Vojir and J. Matas. The enhanced flock of trackers. In *RRIV*. 2014.

[21] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[22] R. Xu and D. Wunsch. Survey of clustering algorithms. *TNN*, 16(3), 2005.

[23] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.

[24] K. Zhang, L. Zhang, and M.-H. Yang. Real-Time compressive tracking. In *ECCV*, 2012.

[25] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012.

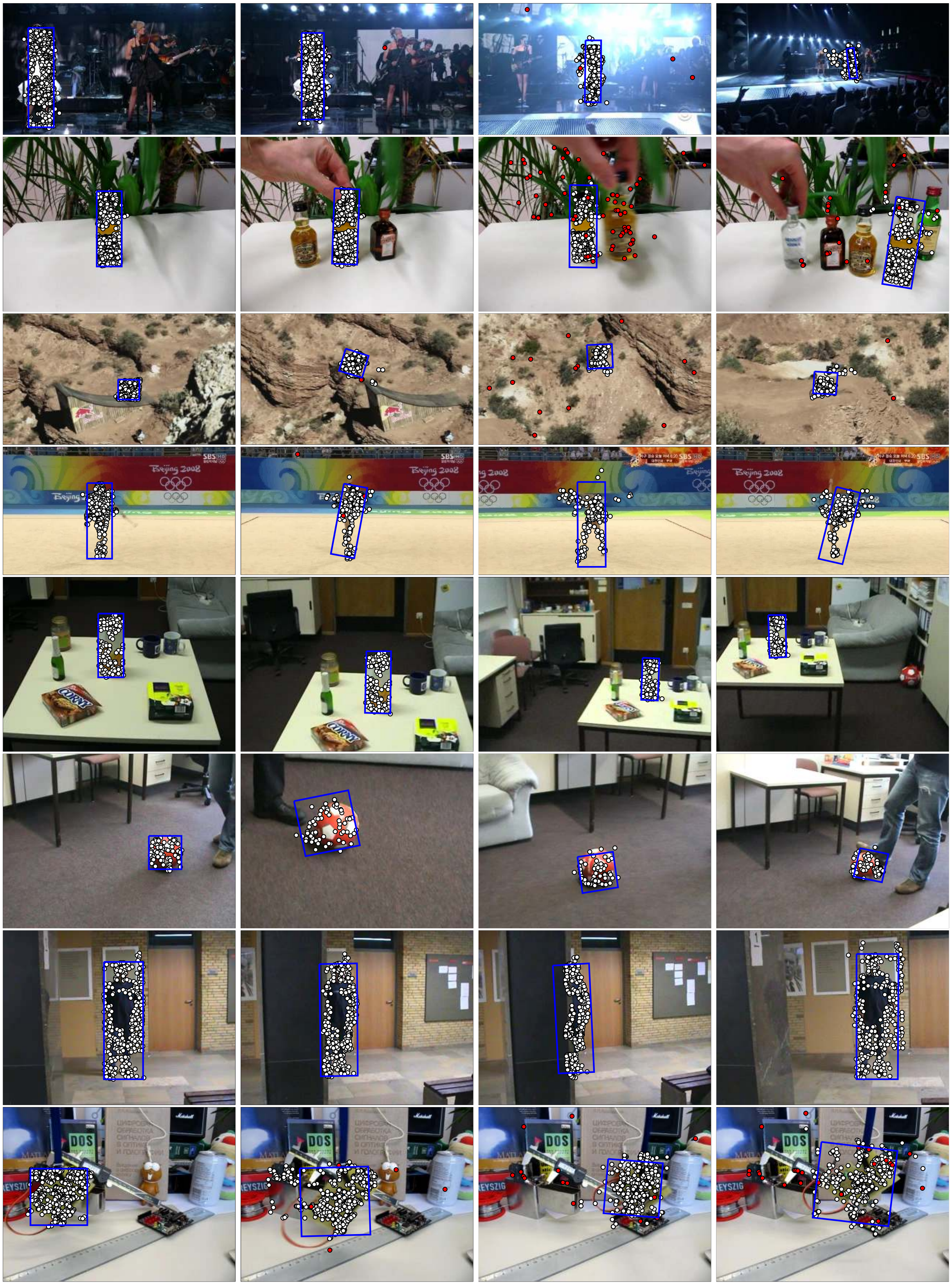

Figure 10. Qualitative results on *singer*, *liquor*, *mountain-bike*, *gym*, *juice*, *ball*, *person occ*, *board*, showing the correspondences (white), detected outliers (red) and the output bounding box (blue).