Universal Hough dictionaries for object tracking

Fausto Milletari[1]
fausto.milletari@tum.de
Wadim Kehl[1]
kehl@in.tum.de
Federico Tombari[12]
tombari@in.tum.de
Slobodan Ilic[14]
slobodan.ilic@in.tum.de
Seyed-Ahmad Ahmadi[3]
ahmadi@cs.tum.edu
Nassir Navab[1]
navab@cs.tum.edu

[1] Computer Aided Medical Procedures
Technische Universität München
München, Germany

[2] Computer Science Department (DISI)
University of Bologna
Bologna, Italy

[3] Department of Neurology
Ludwig-Maximilians-Universität
München, Germany

[4] Siemens AG
München, Germany

Abstract

We propose a novel approach to online visual tracking that combines the robustness of sparse coding with the flexibility of voting-based methods. Our algorithm relies on a dictionary that is learned once and for all from a large set of training patches extracted from images unrelated to the test sequences. In this way we obtain basis functions, also known as atoms, that can be sparsely combined to reconstruct local image content. In order to adapt the generic knowledge encoded in the dictionary to the specific object being tracked, we associate a set of votes and local object appearances to each atom: this is the only information being updated during online tracking. In each frame of the sequence the object's bounding box position is retrieved through a voting strategy. Our method exhibits robustness towards occlusions, sudden local and global illumination changes as well as shape changes. We test our method on 50 standard sequences obtaining results comparable or superior to the state of the art.

1 Introduction

Tracking arbitrary objects in video sequences is an unsolved problem in computer vision. Current methods are required to be flexible to handle abrupt appearance changes while exhibiting robustness to external factors such as varying lighting conditions, viewpoint changes, occlusions and background clutter which can otherwise trigger algorithm failures. Since object tracking is central to many computer vision tasks, research efforts have been focused on proposing new algorithms that are capable of dealing with real-world challenges.

Current approaches can be grouped into two classes: discriminative trackers, which make use of a classifier to distinguish the object of interest from the background, and generative

trackers, which rely on appearance models to capture and match the visual characteristics of the object of interest across the frames. Recent examples of discriminative methods [5, 7] made use of a voting strategy to track deformable objects while ensuring robustness towards occlusion. Generative approaches such as [12, 15, 16] rely on a set of object templates stored in a dictionary to maintain the appearances of the object of interest and on sparse coding to robustly recognise it.

Our approach combines the advantages of both sparse coding and Hough voting-based strategies to reliably track unconstrained objects. Instead of using dictionaries containing object templates and relying on the reconstruction error to score candidate object positions as in [2, 15, 16, 17], we learn a generic, fixed, over-complete dictionary from small patches collected from images unrelated to the test sequences. Such resulting *universal* dictionary, which is estimated once and for all, is capable of reconstructing portions of the target object using a sparse combination of visual words selected with the awareness of the large range of appearances that can be found in real-world situations, as supported by the findings of [13].

At initialisation, the content of the manually placed bounding box is reconstructed patch-wise through the atoms of the dictionary and the notion of target shape and appearance is acquired by storing votes associated to each atom. The votes are stored as displacement vectors between the patches sampling positions and the center of the bounding box, while the appearances are represented by the reconstructions obtained through the dictionary. Due to the fact that the object is modelled locally by means of small patches, our approach is able to cope well with the presence of occlusion, noise, blur, sudden local and global illumination changes and background clutter. Furthermore, our approach can adapt to appearance changes of the tracked object by means of a specific update procedure of the votes and appearances associated to the dictionary atoms.

2 Previous Work

Discriminative approaches to visual object tracking make use of classifiers to produce binary [1, 9] or structured predictions [5, 6, 7, 8] and therefore distinguish the object of interest from the background. The most relevant factors influencing the performance of the trackers are the discriminative capabilities of the features, the choice of the learning algorithm, and the online update strategy. In [1] a classifier based on boosting was updated online using multiple instance learning, while [9] proposed to integrate structural constraints in order to limit the impact of data-samples that are unlikely to be related to the target during update. More recently [8] employed a kernelized structured output Support Vector Machine (SVM) to regress the transformation of the bounding box between subsequent frames. In [6], Gaussian Process Regression (GPR) was employed to discriminate the target position from background points, by means of a semi-supervised approach that learns the discriminative model from both previously seen samples as well as unseen candidates directly extracted from the current frame. Hough forests have been used [5, 7] to jointly perform classification and localization of the target bounding box. Data-points are classified and, depending on their label, they are enabled to cast votes which localize the target. Recently, Convolutional Neural Networks (CNN) have also been employed to classify between target and background [10].

Generative approaches such as [3, 4] model the appearances of the object of interest using histograms and ensure robustness using a fast segmentation strategy which prevents the appearances of portions of the background from triggering failures. In [3] a Graph Cut-based segmentation method was employed in each frame to obtain reliable histograms. In

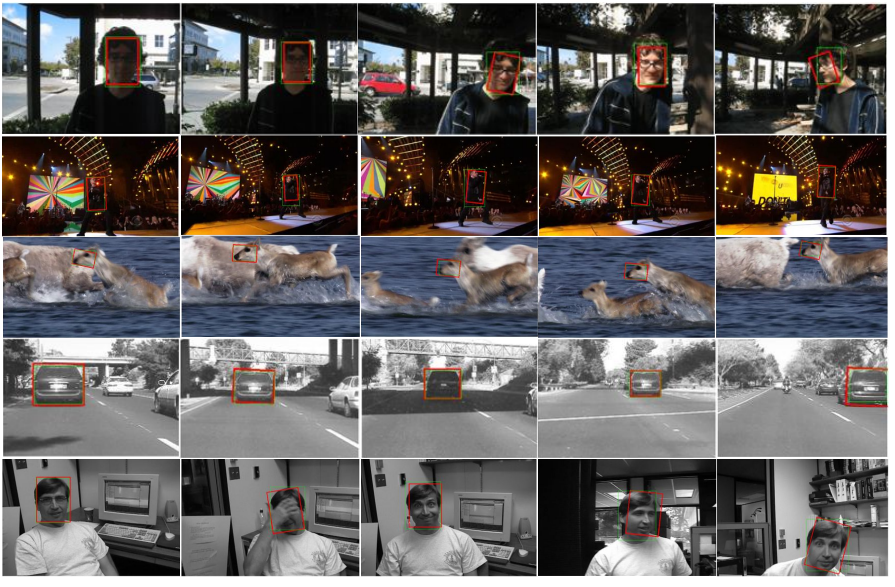

Figure 1: Qualitative results on the sequences 'Trellis', 'Singer2', 'Deer', 'Car4' and 'Dudek'. Our results are highlighted in red, manual annotation from the benchmark sequence is depicted in green.

[4] a probabilistic framework was developed to obtain Maximum-A-Posteriori estimations (MAP) of both a level set-based segmentation contour wrapping the object of interest, as well as an affine transformation accounting for rigid object motion. Other generative methods [2, 15, 16, 17] make use of a dictionary of target object templates and sparse coding to score candidate bounding boxes positions. In each frame, patches are collected from the image and sparsely reconstructed through the dictionary. The reconstruction fidelity serves as a likelihood of candidate patches to depict the object of interest.

3 Method

We propose to carry out visual object tracking by means of an universal dictionary, learned offline, together with a specific voting strategy. The algorithm comprises three steps:
Offline Dictionary learning — Where we learn a dictionary of visual words from a large set of randomly sampled image patches, with the goal of obtaining a set of basis functions (i.e. atoms) capable of reconstructing a large variety of local image appearances.
Tracker Initialisation — Aiming at adapting the generic knowledge captured in the dictionary to the target object. This is achieved by storing votes to the bounding box centroid and associated local object appearances in correspondence to each dictionary atom.
Online Tracking — Whose purpose is to track the object across the sequence using a generalised Hough voting strategy. We reconstruct image patches through the dictionary and we cast the votes associated to each atom employed for the reconstructions in order to obtain the updated bounding box centroid position.

The intuition is that, as long as the appearances of the target do not radically change,

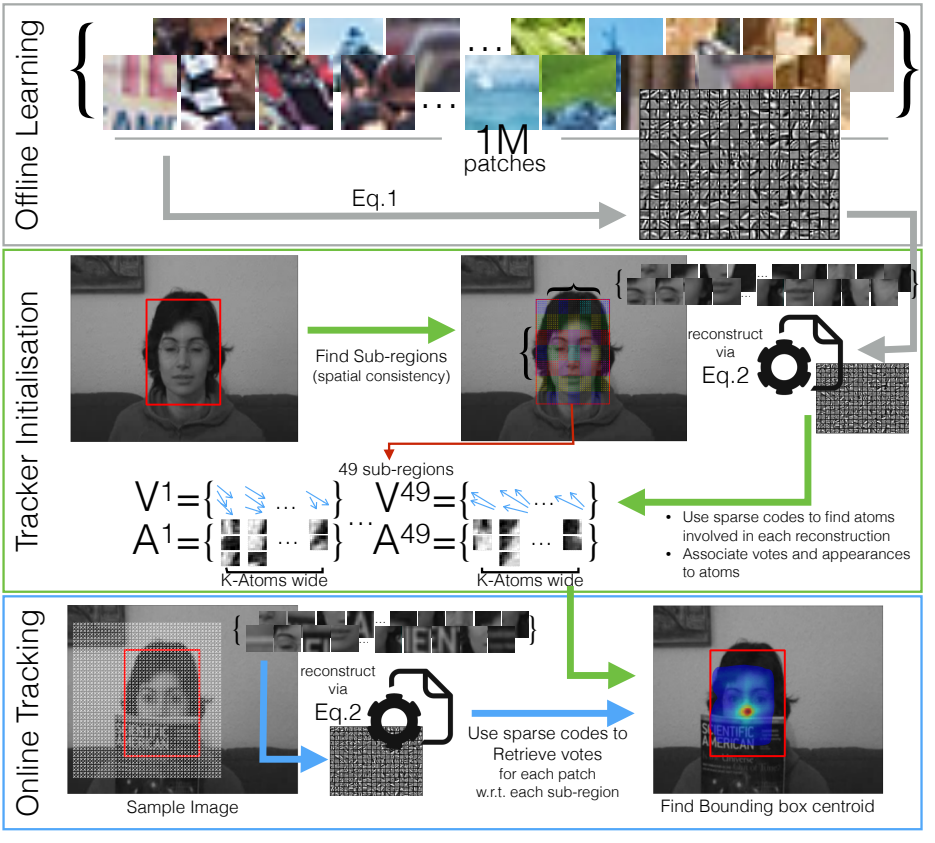

Figure 2: During offline learning we obtain a generic dictionary from image patches (Sec. 3.1). The initialisation aims at collecting object-specific information in the form of votes and local appearances (Sec. 3.2). Online tracking is implemented using a voting strategy to retrieve the centroid of the bounding box (Sec. 3.3).

its parts are always reconstructed using the same set of atoms. Therefore, the bounding box position in each frame can be retrieved using the proposed voting strategy. To cope with object appearance changes, the votes and appearances are updated in each frame to achieve robustness, while, conversely, the atoms of the dictionary are never modified.

3.1 Offline Dictionary Learning

Recent approaches demonstrated the capabilities of sparse coding to perform tasks such as denoising, texture synthesis, compression and audio processing [□]. In these approaches, a dictionary of non-orthogonal basis functions is employed to obtain sparse reconstructions of the input signals. We propose to reconstruct parts of the image using a limited number of basis patches, the atoms of the dictionary, which capture phenomena underlying real-world appearances. In our intuition we can retrieve sparse codes that are discriminative of the object of interest by deploying a dictionary capable of reconstructing a large range of different image patches. In contrast to previous methods based on l_1-sparse coding, we do not try to

explain parts of the image using templates depicting the object of interest [2, 12, 16, 17], neither we employ the reconstructions fidelities, which are potentially misleading, to score candidate object positions. In our approach, we employ a dictionary that can approximately reconstruct every possible image patch and which possesses knowledge about recurrent intensity patterns as seen in the training set. As a result, we encode the object of interest through combinations of dictionary atoms, each of which encodes the causes underlying intensity patterns occurring in real scenes [13]. This is possible because our dictionary is trained with an amount of data that goes well beyond that which is available in the first frame of the sequence. During the first step of our algorithm (Fig. 2, top), we collect a large set $\mathbf{T} = \{\mathbf{t}_1, ..., \mathbf{t}_n\}$ of grayscale image patches from generic images downloaded from the Internet and we learn a dictionary $\mathbf{D} = \{\mathbf{d}_1, ..., \mathbf{d}_k\}$ containing k atoms by optimising the following problem with respect to $\mathbf{D}$:

$$\arg\min_{\mathbf{D}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} ||\mathbf{t}_i - \mathbf{D}\alpha_i||_2^2 + \lambda ||\alpha_i||_1. \tag{1}$$

We aim to minimise the sum of squared differences (SSD) between the patches contained in the dataset $\mathbf{T}$ and their sparse reconstructions obtained as a linear combination of the columns of $\mathbf{D}$ through the coefficients $\alpha_\mathbf{i} \in \mathbb{R}^k$. The strength of the sparsity constraint can be controlled through the parameter λ.

3.2 Tracker Initialization

Using the object bounding box provided in the first frame of every sequence, we initialise the method by capturing the shape and appearance of the object of interest: we rely, as previously stated, on a set of votes pointing to the bounding box centroid $\mathbf{c} = (c_x, c_y)$ together with a representation of the appearances of the region where each vote originated from.

Specifically, the initial bounding box is subdivided into $M \times N$ sub-regions $R_1, ..., R_{M \times N}$ which are linked with dedicated data structures storing votes and appearances. In this way, we ensure that even if patches from different sub-regions are reconstructed using the same set of dictionary atoms, the resulting votes never induce a violation of the initial spatial configuration of the object's sub-regions during tracking. That is, the absolute ordering of the sub-regions cannot be changed. The subdivision of each template into sub-regions is graphically depicted in Fig. 2, middle.

For each of the sub-regions we densely extract image patches $\mathbf{p}_1, ..., \mathbf{p}_s$ at locations $\mathbf{x}_1, ..., \mathbf{x}_s$ having the same dimensionality as the atoms in $\mathbf{D}$. Each patch $\mathbf{p}_i$ is reconstructed through $\mathbf{D}$ by solving the l_1-sparse optimization problem

$$\arg\min_{\alpha_i} \frac{1}{2} ||\mathbf{p}_i - \mathbf{D}\alpha_i||_2^2 + \lambda ||\alpha_i||_1 \tag{2}$$

yielding the sparse coefficients α_i, and the reconstructions $\hat{\mathbf{p}}_i = \mathbf{D}\alpha_i$. Using these sparse coefficients α_i we identify the indices of the atoms that contributed to the reconstruction of $\mathbf{p}_i$. Supposing that the i-th patch $\mathbf{p}_i$ belongs to the j-th sub-region and that it required the contribution of the k-th atom during its reconstruction, the vote $\mathbf{v}_i = \mathbf{c} - \mathbf{x}_i$ and the appearance $\hat{\mathbf{p}}_i$ are respectively added to the sets $\mathbf{V}_k^j$ and $\mathbf{A}_k^j$ (Fig. 2, middle). Importantly, storing sparse reconstructions $\hat{\mathbf{p}}_i$ as robust representations of region appearances is advantageous since it allows to reduce the effects of noise, and it implicitly encodes the configuration of the sparse coefficient vector α characteristic of the patches used for initialisation.

Our l_1-sparse optimisation of Eq.1 is almost instantaneous due to the fact that the dictionary atoms $\mathbf{d}_i$, as well as the signals $\mathbf{p}_i$, consist of very small patches with low dimensionality, thus yielding an average computational cost for the initialisation step of typically just a few milliseconds.

3.3 Online Tracking

We track the object across the sequence by retrieving the position of the bounding box centroid in each frame through the voting strategy. We extract image patches $\mathbf{p}_1^j, ..., \mathbf{p}_N^j$ from the area surrounding the last known position of each sub-region R_j ($50px^2$ in our experiments) and we reconstruct them using the dictionary $\mathbf{D}$ solving the l_1-sparse optimisation as stated in Eq. 2. The obtained sparse codes α_i^j and the reconstructions $\hat{\mathbf{p}}_i^j = \mathbf{D}\alpha_i^j$ are respectively employed to identify the atoms involved in each reconstruction and to obtain weights for the votes $\mathbf{V}_k^j$ by comparison with the learned appearances stored in $\mathbf{A}_i^j$ (Fig. 2, bottom). Let us suppose the i-th image patch belongs to the search area of the j-th subregion and that is reconstructed through the k-th dictionary atom: we cast all the votes $\mathbf{v}_l^{(k,j)}$ stored in $\mathbf{V}_j^k$ after weighting their contributions with the weights $w_l^{(k,j)}$ obtained as the reciprocal of the SSD between the appearances $\mathbf{a}_l^{(k,j)}$ and the reconstruction $\hat{\mathbf{p}}_i^j$:

$$w_l^{(k,j)} = \frac{1}{(\mathbf{a}_l^{(k,j)} - \hat{\mathbf{p}}_l^j)^\top (\mathbf{a}_l^{(k,j)} - \hat{\mathbf{p}}_l^j)}. \tag{3}$$

The weighted votes contribute to a vote map. The bounding box centre is found by identifying the location of the highest peak in the smoothed vote map.

Since the different search areas often overlap, an efficient implementation of the reconstruction can be achieved by solving Eq. 2 only once for all the patches in the global search area, regardless of the sub-regions they belong to. After the sparse codes are retrieved, they are interpreted using the information stored in the data structures of the specific sub-regions.

3.3.1 Update strategy

Once the bounding box is estimated, we select the atom of the dictionary that was employed the most for reconstruction of the background area and we prune its votes and appearances from the data structures of every sub-region. On the other hand, all the samples contained inside the estimated bounding box serve to update the voting structures through a procedure similar to the one used during initialisations. In this way, we aim to keep information about the object until the moment it becomes misleading. This happens when votes and appearances get coincidentally associated to background structures.

3.3.2 Handling scale changes and rotations

The votes and appearances employed in our method are not invariant to rotation and scale changes. When the object changes orientation or size, the votes do not accumulate in clear peaks anymore. To handle scale changes and rotations of the target object, we create different versions of the input frame which are rotated and rescaled by fixed quantities. We decide for the rotation and scale for which we obtain the vote map yielding the maximum peak. The inverse of the estimated parameters are then added to the current state of the tracker.

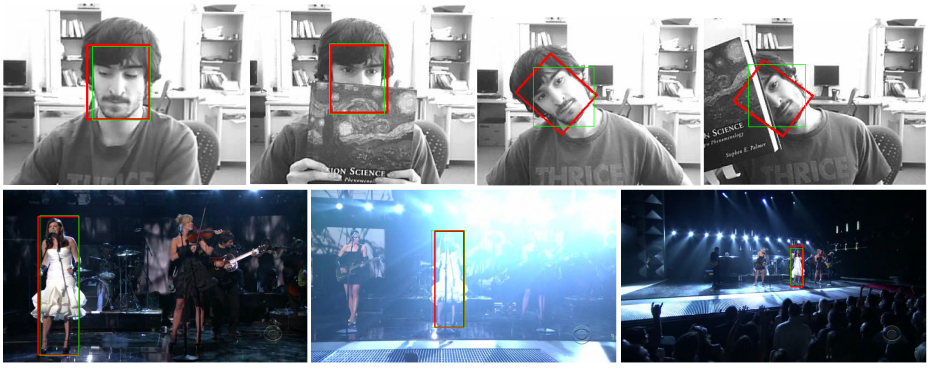

Figure 3: Our method, whose output is depicted using a red bounding box, is able to cope with large rotations and scale changes. Note that the manual annotation provided in the benchmark dataset [14], depicted in green, does not take into account rotations.

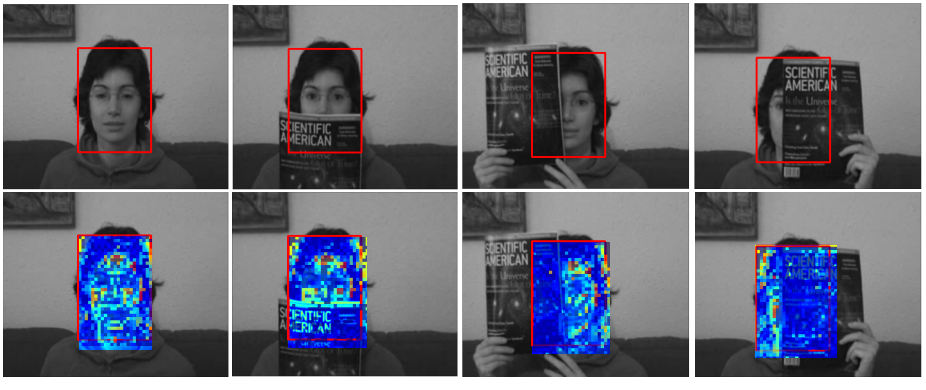

Figure 4: Backprojection of the Hough votes. Upper row: Output of our algorithm. Lower row: Votes having high weights (jet colormap) were generated only by patches belonging to the visible region of the object: the occlusion has a negligible effect on the vote map.

Since performing an exhaustive search by considering a large range of rotations and scale changes is a computationally intensive task, we rely on the assumption, commonly used in tracking, that the position, scale and rotation of the object changes smoothly from one frame to the other. In this way, as shown in Fig. 3, we can deal with those changes by only considering a small range of rotations and scale factors.

3.3.3 Robustness against occlusions

As previously stated, our method exhibits robustness to large amounts of occlusion. Since the reconstruction of the object is performed patch-wise and a few patches are already sufficient to cast a high number of votes with high confidence, we are able to localise the bounding box even when large portions of the target are not visible. In Fig. 4 we show the behaviour of our approach when the object undergoes occlusions. We re-project the votes that contributed to the estimation of the bounding box in each frame to the position of the patches that generated

Figure 5: Robustness towards illumination changes is achieved by normalising the patches extracted from the image. Even in sequences like 'David', where extreme illumination changes are present, our method performs correctly.

them and we observe that only visible parts of the object are able to effectively contribute to the estimation of the bounding box centroid.

3.3.4 Robustness against illumination changes

The patches extracted from the images both during initialisation and tracking are normalised to zero mean and unit standard deviation. The same applies to the appearances stored in correspondence of the dictionary atoms. As briefly shown in Fig. 5, our method exhibits robustness against extreme illumination changes.

4 Experimental evaluation

To test our approach we employ the benchmark dataset published in [14] that consists of 50 annotated sequences (51 targets) including challenging situations such as illumination changes, deformations, occlusions, background clutter and motion blur. We compared with the most recent approaches having publicly available results on this benchmark, in particular 'L1APG' [2], 'MTT' [16], 'SCM' [17], 'Struck' [8], 'TGPR' [6] and all the others which have been evaluated in [14]. We follow the experimental protocols proposed in the benchmark [14] and evaluate our approach in terms of success and precision. All the sequences were converted to grayscale. The parameters of each algorithm are fixed for all the sequences and the bounding box used for initialisation is provided in the first frame. Since the first frame of the sequence 'David' is very dark and unsuited for the initialisation of many tracking algorithms, all the methods used for comparison were initialised at frame 300 while ours was initialised at frame 1. Although our approach yields better performance when initialised at frame 300, we want to demonstrate that we are able to track the object correctly even if the initialisation frame is extremely dark as shown in Fig. 5. The average overlap and precision plots for all the experiments are depicted in Fig. 6. The performance of the trackers is expressed in terms of area under curve (AUC) and these values are enclosed in brackets in the plot of Fig.6. Qualitative results are shown in Fig. 1.

4.1 Parameters of the algorithm

The parameter λ, which controls the sparsity of both the dictionary and the sparse reconstructions is set to 0.1. The dictionary D consists of $k = 300$ 8×8 pixels atoms. During online tracking, candidate patches are collected using a regular grid which has a 4 pixels spacing and which covers a 50 pixels wide area around the last known position of the bounding box.

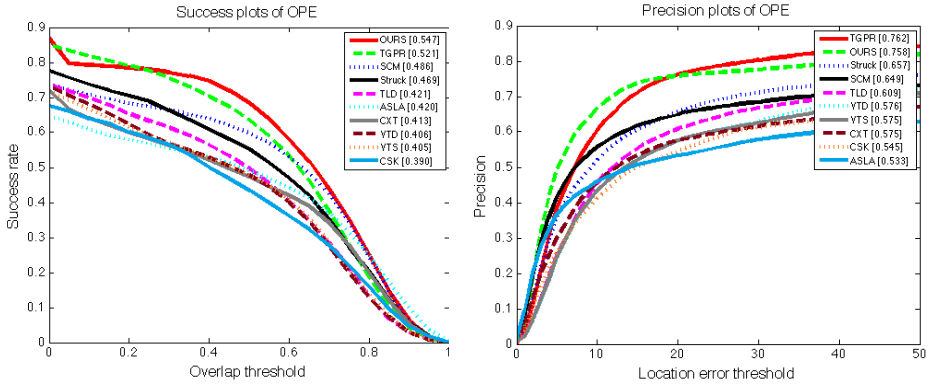

Figure 6: Results in terms of success and precision comparing our method with top performing algorithms on the 50 sequences (51 targets) of the CVPR13 Visual Tracking Benchmark[14]. Area under curve (AUC) is reported in brackets.

To handle scale changes and rotations, we transform each frame by considering each possible pair of scale and rotation from the set of possible rotation offsets, $\Delta r = \begin{bmatrix} -3 & 0 & 3 \end{bmatrix}$ degrees, and the set of possible scale offsets, $\Delta s = \begin{bmatrix} -0.03 & 0 & 0.03 \end{bmatrix}$. With these empirically selected parameters, our MATLAB implementation processes approximately 5 frames per second.

4.2 Results on selected sequences

From empirical observations we have noticed that the our tracking method tends to fail over low-resolution sequences depicting small target objects or objects that are hardly distinguishable from the surroundings (in grayscale). As a result, the algorithm performs unsatisfactorily in sequences such as 'Basketball', 'Bolt', 'Freeman3', 'Freeman 4', 'Girl' and 'CarDark'. We conclude that, failure over 'Freeman3', 'Freeman 4' and 'CarDark' sequences is due to the small size of the initial bounding box (with an area of resp. 156, 240, 667 px^2), which causes the number of votes stored during training to be low. Failure in the 'Basketball', 'Bolt', 'Girl' and 'CarDark' sequences are instead mostly determined by the additional presence of background clutter and lack of contrast between the objects and their surroundings in the grayscale images. Once these sequences are left out from the evaluation, we observe that the performance gap between our approach and the others remarkably increase to our favor, as witnessed by Fig. 7, which shows the results, in terms of success and precision, on 44 sequences (45 targets), where the 6 benchmark sequences having smallest resolution and target size were excluded.

5 Conclusion and future work

We presented a novel method for robust object tracking which uses dictionaries in a new fashion: generic, non object-specific information is learned from random images and it is used to reconstruct the object of interest patch-wise. The locality of these reconstructions coupled with the robustness of Hough voting allows the algorithm to perform in presence of large occlusions, illumination changes, motion blur and background clutter. Our approach

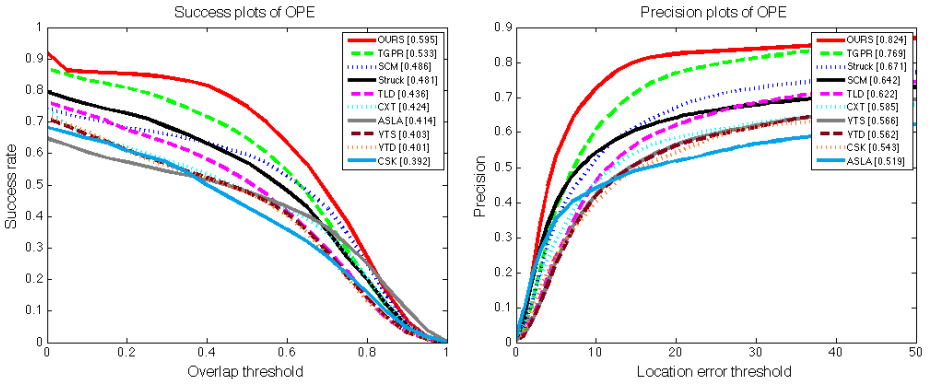

Figure 7: Results in terms of success and precision our method in comparison with top performing algorithms on 44 sequences (45 targets). Area under curve (AUC) reported in brackets.

outperforms the state of the art on every sequence apart from the ones that suffer from very low resolutions and depict very small, hard to distinguish, target objects.

As a future work we plan to investigate a similar strategy using deep sparse auto-encoders instead of dictionaries.

6 Acknowledgement

We would like to acknowledge DFG for having supported this work through the grant BO 1895/4-1.

References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

[2] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1830–1837. IEEE, 2012.

[3] Vasileios Belagiannis, Falk Schubert, Nassir Navab, and Slobodan Ilic. Segmentation based particle filtering for real-time 2d object tracking. In *Computer Vision–ECCV 2012*, pages 842–855. Springer, 2012.

[4] Charles Bibby and Ian Reid. Robust real-time visual tracking using pixel-wise posteriors. In *Computer Vision–ECCV 2008*, pages 831–844. Springer, 2008.

[5] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

[6] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *Computer Vision–ECCV 2014*, pages 188–203. Springer, 2014.

[7] Martin Godec, Peter M Roth, and Horst Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2013.

[8] Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

[9] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010.

[10] Hanxi Li, Yi Li, and Fatih Porikli. Robust online visual tracking with a single convolutional neural network. In *Computer Vision–ACCV 2014*, pages 194–209. Springer, 2015.

[11] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[12] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009.

[13] Ivana Tosic and Pascal Frossard. Dictionary learning. *Signal Processing Magazine, IEEE*, 28(2):27–38, 2011.

[14] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[15] Junliang Xing, Jin Gao, Bing Li, Weiming Hu, and Shuicheng Yan. Robust object tracking with online multi-lifespan dictionary learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 665–672. IEEE, 2013.

[16] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2042–2049. IEEE, 2012.

[17] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845. IEEE, 2012.