Superpixel Segmentation for Robust Visual Tracking

Luka Čehovin[1] **Matej Kristan**[1] **Aleš Leonardis**[1,2]

[1] *University of Ljubljana, Faculty of Computer and Information Science*
[2] *CN-CR Centre, School of Computer Science, University of Birmingham*
e-mail: {luka.cehovin,matej.kristan,ales.leonardis}@fri.uni-lj.si

Abstract

In this paper we address visual tracking of articulated objects. We present a new visual tracker, inspired by the Local-Global tracking methodology, that uses superpixels to reduce the sample set that the model has to consider when determining which areas in the image belong to the target. This reduction enables effective use of classical machine-learning techniques to model the global appearance of the object as well as a more rational organization of the local object description. We present experimental results that demonstrate the feasibility and computational performance improvements of the proposed approach as well as point out some directions of our future work.

1 Introduction

Visual tracking is an important research area in computer vision with applications in surveillance, activity recognition, sport analysis, human-computer interaction and many other fields. In practice, the holistic approaches [5, 2], that globally model the target's appearance, have proven to be very successful. However, scenarios that contain rapid structural and other kinds of appearance changes lead to reduced matches and drifting which eventually result in the trackers' failure.

To address the appearance changes robustly, approaches using sets of simple local parts have been proposed. These local parts describe only parts of the object and are therefore more adaptable to geometry changes. Finding a good match for a geometrically constrained constellation of parts is a high-dimensional optimization problem with many local maxima that is hard to solve even for a manually initialized set of parts [10, 3]. However, in a more complex tracking scenario some parts become outdated after a while and have to be replaced. Because of the rapidly changing structure the geometrical model is usually simplified and an *add-remove* scheme (which parts to replace) is introduced to the algorithm. Several such trackers have been presented over the past years [6, 14, 8].

Recently, the idea of local geometrically constrained parts and add-remove scheme was formalized as a Local-Global tracking hierarchy [12]. In the context of this methodology the parts represent a local (bottom) layer of the hierarchy that is responsible for accurate short-term tracking. As some parts become outdated and are removed from the visual model, new parts have to be added. This is the main task of the global (top) layer that forms the global appearance model of the object which is responsible for robust long-term tracking. The global appearance model is not suitable for direct tracking, however, it can use various appearance cues to infer possible regions of the target that are not yet covered by any of the active parts. The global appearance model is updated with time as well with the reliable information from the local layer. Such tracker can be robust and accurate if both layers of the visual model are implemented well. The authors of [12] have shown that above state-of-the-art tracking results can be achieved even by organizing relatively simple computer vision concepts in the proposed way. In this paper explore an extension to the original LGT tracker [12] that combines superpixel segmentation [11] with classification methods to model the global appearance of an object.

The rest of the paper is organized as follows: Section 2 presents the related work and highlights our contributions. Section 3 briefly describes the basic LGT visual model and presents the proposed improvements to the model. In Section 4 we perform preliminary experimental evaluations, and in Section 5 we discuss the improvements and draw conclusions.

2 Related work

Superpixel algorithms group pixels into perceptually meaningful regions. The image, partitioned in superpixels is usually over-segmented, however, superpixels capture image redundancy and provide a convenient primitive concept from which to compute image features, that greatly reduces the complexity of subsequent image processing tasks. Superpixels have become popular in many computer vision tasks, most notably as object segmentation [4]. Recently, superpixels have also been used for visual tracking. In [13] superpixels are used directly to infer the position of the object using mean-shift clustering of superpixels. The visual size of the object must stay the same during tracking as this tracker relies on this information for robust tracking. Superpixels have also been used to vehicles [9], however, the visual model is even more limited in this case as it contains prior knowledge suited only for vehicle tracking.

On the other hand, machine learning approaches have become popular in visual tracking under the term "tracking by detection". On-line boosting [5] is one of the best known approaches, followed by the multiple-instance-learning [2]. This kind of tracker is able to track the object by discriminating between the object and its immediate neighborhood as well as adapt to changes in the environment. A clear limitation of these approaches, which stems from a fixed set of spatial features used, is the assumption about non-articulated objects that do not change their (visual) size.

Instead of using superpixel segmentation or machine learning directly to track the target, we propose to use both aforementioned concepts within the Local-Global tracking methodology to model global appearance of the target. In the original LGT tracker [12], the global visual model of the target is composed of three visual modalities (that are considered independent): color, motion and shape. Each modality is modeled in a domain-specific way, however, they all supply a mapping from a pixel in the image to the probability that this pixel belongs to the target. The contribution of this paper is the the introduction of superpixel segmentation to partition the image into homogeneous regions that can be considered as individual entities in the global appearance model instead of raw image pixels. Because of a lower number of samples and their limited dimensionality, classical machine-learning classifiers can be used to effectively learn the appearance of the object. Image-aware partitioning is also used to initialize local parts better thus reducing their number and therefore improving performance of the tracker even further.

3 Superpixel Local-Global Tracking

In this section we present a theoretical description of our work. In the beginning we briefly summarize the segment of the original LGT visual tracker that we have preserved, i.e. the local layer of the visual model. From there we move on to the presentation of our new global layer that uses superpixel segmentation and machine-learning algorithms and after that explain how superpixels can also be beneficial for object agnostic initialization of the tracker.

3.1 The Local Layer

The local layer $\mathcal{L}_t$ of the the target's visual model at time-step t is described by a geometrical constellation of weighted patches

$$\mathcal{L}_t = \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1\ldots N_t}, \tag{1}$$

where $\mathbf{x}_t^{(i)}$ represents the image coordinates of the i-th patch and the weight $w_t^{(i)}$ represents the belief that the target is well-represented by the i-th patch. We denote the set of all patches at time-step t by $\mathbf{X}_t = \{\mathbf{x}_t^{(i)}\}_{i=1:N_t}$. Each patch is very small and is compared to the image only using a simple gray-scale histogram. This extremely simple visual description provides low computational complexity and a good short-term tracking support, especially when using more such patches together, however, it is not sufficient for more than a certain period of time, usually a few frames.

During tracking, we start from an initial estimate $\hat{\mathbf{X}}_t$ and a set of current image measurements $\mathbf{Y}_t$. We seek the value of $\mathbf{X}_t$ that maximizes the joint probability $p(\mathbf{Y}_t, \mathbf{X}_t|\hat{\mathbf{X}}_t)$. By treating the local-layer visual model $\mathcal{L}_t$ as a mixture model, in which each patch competes to explain the target's appearance, we can decompose the joint distribution into

$$p(\mathbf{Y}_t, \mathbf{X}_t|\hat{\mathbf{X}}_t) = \sum_{i=1}^{N_t} p(\mathbf{z}_t^{(i)}) p(\mathbf{Y}_t, \mathbf{X}_t|\hat{\mathbf{X}}_t, \mathbf{z}_t^{(i)}), \tag{2}$$

where $\mathbf{z}_t^{(i)}$ is the appearance property of the i-th patch that determines the prior, and $p(\mathbf{z}_t^{(i)}) = w_t^{(i)} / \sum_{j=1}^{N_t} w_t^{(j)}$ quantifies the representativeness of the i-th patch for the tracked model using weights of the patches. The details about the optimization algorithm that takes into account the ad-hoc geometric properties of the patch set can be found in [12].

3.2 The Global Layer

The main function of the global layer is long-term appearance modeling with the primary goal of providing local-layer potential positions for new patches. In this section we describe how to use superpixel segmentation a binary classifier to discriminate between foreground (target) and background regions in the image. The overview of the idea is illustrated in Figure 1.

In our proposed approach, we use the popular SLIC superpixel algorithm [1] to segment an image. We have selected this specific algorithm because of three advantages: the number of superpixels can be given in advance[1], the resulting superpixels have uniform size, and computational performance. The original method operates in the *Lab* color space, however, we have modified it to work in the YCbCr color space as the conversion from the RGB color space is much faster and the two provide a nearly identical segmentation for our purpose.

We characterize each superpixel as a sample vector containing the center position ($x_t^{(i)}$ and $y_t^{(i)}$) and a mean color in the YCbCr color space ($Y_t^{(i)}$, $Cb_t^{(i)}$, and $Cr_t^{(i)}$), which can be retrieved directly from the algorithm, as well as the estimated local motion ($\dot{x}_t^{(i)}$ and $\dot{y}_t^{(i)}$)

$$\begin{aligned}\mathcal{S}_t &= \{s_t^{(i)}\}_{i=1\ldots S_t}, \\ s_t^{(i)} &= \{x_t^{(i)}, y_t^{(i)}, Y_t^{(i)}, Cb_t^{(i)}, Cr_t^{(i)}, \dot{x}_t^{(i)}, \dot{y}_t^{(i)}\},\end{aligned} \tag{3}$$

where S_t denotes the number of superpixels at frame t. Estimating local motion in the image is usually done using one of the optical-flow algorithms. This approach is adopted in [12] to compare local image motion with the motion of the object, however, it is only able to estimate motion in stable regions. We avoid the calculation of

[1] Note that the number of superpixels for a specific image can in fact be lower than specified because of merging, however, the algorithm nevertheless provides an uniform segmentation across the entire sequence.

optical-flow altogether and estimate the motion using the distance between two most similar superpixels in frames $t-1$ and t using their position and color similarity as a 5-dimensional vector

$$\{\dot{x}_t^{(i)}, \dot{y}_t^{(i)}\} = \{x_t^{(i)}, y_t^{(i)}\} - \{x_{t-1}^{(j)}, y_{t-1}^{(j)}\},$$
$$s_{t-1}^{(j)} = \underset{s_{t-1}^{(k)} \in \mathcal{S}_{t-1}}{\arg\min} \; \|d_t^{(i)} - d_{t-1}^{(k)}\|, \quad (4)$$

where $d_t^{(i)} = \{x_t^{(i)}, y_t^{(i)}, Y_t^{(i)}, Cb_t^{(i)}, Cr_t^{(i)}\}$ denotes the 5-dimensional vector. To complete the initial definitions of the model, we define two relations. Let the first one, $\mathbf{l}_t$, be a mapping from image coordinates to the appropriate superpixel, and the second one, $\mathbf{n}_t$, a neighborhood relationship between the superpixels in $\mathcal{S}_t$ (two superpixels are neighbours if they appear next to each other in image-space)

$$\mathbf{l}_t : \mathbb{R}^2 \rightarrow \mathcal{S}_t \cup \emptyset, \;\; \mathbf{n}_t : \mathcal{S}_t \rightarrow \mathcal{S}_t \quad (5)$$

To train a classifier, we have to separate our data into two sets, positive and negative samples (in our case superpixels). As the a-priori labels are not available during tracking, we have to rely on the local layer to provide the mapping. We define the positive sample subset of $\mathcal{S}_t$ as those superpixels, that are occupied by the patches with their weight higher than a threshold

$$\mathcal{F}_t = \{s_t^{(i)} \mid \exists \{\mathbf{x}_t^{(j)}, w_t^{(j)}\} \in \mathcal{L}_t : \mathbf{x}_t^{(j)} \mathbf{l}_t \, s_t^{(i)} \wedge w_t^{(j)} > \lambda_F\}, \quad (6)$$

where λ_F denotes the patch weight threshold that discards potentially misperforming patches. The negative sample subset for $\mathcal{S}_t$ are those superpixels, that are not in the $\mathcal{F}_t$, nor are they neighbors with one of the superpixels in $\mathcal{F}_t$

$$\mathcal{B}_t = \mathcal{S}_t - \{s_t^{(i)} \mid s_t^{(i)} \in \mathcal{F}_t \vee (\exists s_t^{(j)} \in \mathcal{F}_t : s_t^{(j)} \mathbf{n}_t \, s_t^{(j)})\}. \quad (7)$$

Removing the neighborhood of $\mathcal{F}_t$ is important as these regions are the most likely to belong to the target. By excluding them from the training set we essentially let the classifier decide if they belong to the target or not.

As already mentioned, the proposed global layer $\mathcal{G}_t$ is represented as a binary classifier. Any binary classifier could be used, however, in our current implementation we have tested a decision tree classifier. The classifier is trained using the data from several past frames. When a new patch is required (the criterion for that is described in [12]), the classifier is used to predict which superpixels from the current frame belong to the object. The superpixels that are already covered with an existing patch ($s_t^{(i)} \in \mathcal{S}_t : \exists \{\mathbf{x}_t^{(j)}, w_t^{(j)}\} \in \mathcal{L}_t : \mathbf{x}_t^{(j)} \; \mathbf{l}_t \; s_t^{(i)}$) are filtered out and a new patch is initialized in the center of one of the remaining randomly chosen superpixels using its size to optimally cover the described region.

When and how the global layer $\mathcal{G}_t$ is updated is very important because a stable global appearance model is crucial for long-term tracking. In a preliminary experiment we have tested two simple update strategies: (a) only train the model in the beginning of the sequence and (b) re-train the model every several frames, when enough new training samples are available. From the results of the experiment te have selected the second strategy as the better one and set the update interval to 10 frames, which means that every 10 frames a new decision tree is used to classify superpixels.

Table 1: Results for tracking accuracy (higher is better), failure rate (lower is better) and processing speed (FPS, higher is better).

Tracker →	**LGT**			**SLGT**		
Sequence ↓	**acc.**	**fail.**	**speed**	**acc.**	**fail.**	**speed**
bicycle	**0.524**	1.000	1.459	0.501	1.000	**4.603**
bolt	**0.444**	**0.067**	0.859	0.418	2.667	**4.047**
car	**0.492**	**0.000**	1.839	0.465	0.067	**4.910**
cup	0.627	0.000	2.352	**0.746**	0.000	**4.713**
david	**0.596**	**0.000**	2.804	0.474	0.667	**5.010**
diving	**0.445**	**1.133**	2.586	0.424	1.467	**4.066**
face	**0.596**	0.000	2.584	0.430	0.000	**5.048**
gymnastics	**0.483**	1.333	2.695	0.476	**0.867**	**3.970**
hand	**0.552**	**0.133**	2.103	0.529	0.267	**4.131**
iceskater	**0.548**	**0.000**	3.362	0.438	1.867	**4.728**
juice	**0.714**	0.000	2.773	0.572	0.000	**5.072**
jump	**0.566**	0.000	1.417	0.553	0.000	**4.189**
singer	**0.232**	0.000	2.995	0.223	0.000	**4.922**
sunshade	**0.551**	**0.267**	1.884	0.480	1.533	**3.573**
torus	0.672	**0.000**	2.236	**0.691**	0.067	**4.421**
woman	0.348	**1.133**	3.035	**0.396**	2.267	**4.971**
Average	**0.524**	**0.317**	2.312	0.488	0.796	**4.523**

3.3 Tracker Initialization

The original LGT tracker is initialized using only a bounding box of the object and expects no other a-priori structural information. The initial set of patches is therefore uniformly initialized in a grid pattern within the rectangular region. Besides that all have the same (predefined) size. In the proposed tracker, superpixel segmentation is used to select the number of initial patches as well as their position and size. Using the input bounding box we select superpixels with more than 90% of their pixels falling within the region. We assume that these superpixels belong to the object and initialize patches using their center and size (one patch per superpixel). This approach implicitly provides a number of initial patches and positions them in a more object-specific, but nevertheless homogeneous, constellation.

4 Results

To evaluate the performance of the proposed tracker we have performed a comparative experiment with the original LGT tracker using the VOT2013 [7] sequence dataset that contains 16 annotated video sequences. We have also used the official VOT2013 testing methodology. A comparison of results with the original LGT tracker for tracking accuracy, failure rate and tracking speed are shown in Table 1.

The results indicate that the proposed tracker does not outperform the original tracker in terms of accuracy or

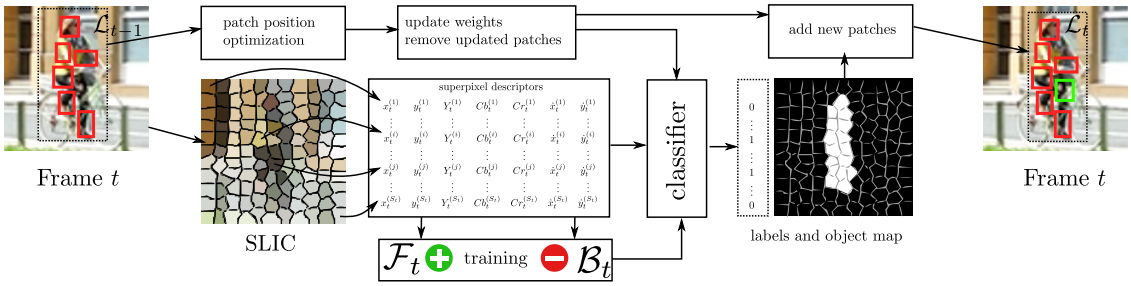

Figure 1: Overview of the single frame processing for SLGT tracker. The local layer is the same as in the original tracker and is only mentioned as a reference, while the global layer is presented in detail.

failure rate *on average* even though these scores are in many specific cases comparable or do even surpass the reference ones. This is mainly due to several sequences that are specifically challenging for the current implementation, such as *bolt*, *sunshade* (hard to separate background from foreground) and *iceskater* (articulated with large scale changes). Small objects are also hard to track, because the size of a superpixel is fixed. Another important problem, that is responsible for a lot of failures is the case of new samples that are not similar to the training samples. In this specific case it would be better to classify those samples as background because misclassified foreground samples negatively influence the performance of the local visual model. Such strategy has not been implemented so far, however, we plan to investigate it together with introduction of prior knowledge into the model as such knowledge exists for many real-world tracking scenarios.

In terms of processing speed, however, the improvement is clear. We have conducted both evaluations on the same machine in the same circumstances using the same reference code-base (written mostly in Matlab). The $\sim 100\%$ improvement can therefore only be explained with the new superpixel segmentation of image space and smarter local patch management.

5 Conclusion

In this paper we have presented a visual tracker, based on the Local-Global tracking methodology, that uses superpixel segmentation in order to reduce the operational domain for the global layer of the visual model. This reduction has enabled us to model the global appearance of the target using classical machine-learning algorithms. We have tested our approach using decision trees, however, other classifiers can be used instead. Many of these algorithms implicitly support feature selection, therefore the proposed tracker is able to choose between position, color, and motion cues depending on the current context.

Preliminary experiments have shown that the proposed tracker is behaving in a predictable way and is already performing well on given benchmark sequences. We have identified important issues that will have to be addressed to develop the tracker to its full potential. As our future work we will also investigate how could an analysis of classification results be used for occlusion and failure detection and how multiple classifiers can be used in the same visual model to form a better multi-view appearance representation of the object.

Acknowledgments: This research was in part supported by: ARRS projects J2-3607, J2-2221 and J2-4284.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–82, November 2012.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *TPAMI*, 33(8):1619–1632, August 2011.

[3] W.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Tracking by Parts: A Bayesian Approach With Component Collaboration. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):375–388, 2009.

[4] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, pages 670–677. IEEE, September 2009.

[5] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, pages 47–56, 2006.

[6] M. Kölsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *CVPR Workshops*, volume 10, page 158, Washington, DC, USA, 2004. IEEE Computer Society.

[7] M. Kristan and L. Čehovin. Visual Object Tracking Challenge (VOT2013) Evaluation Kit. 2013.

[8] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *CVPR*, pages 1208–1215, 2009.

[9] L. Liu, J. Xing, H. Ai, and S. Lao. Semantic superpixel based vehicle tracking. In *ICPR*, pages 2222–2225, 2012.

[10] B. Martinez and X. Binefa. Piecewise affine kernel tracking for non-planar targets. *Pattern Recognition*, 41(12):3682–3691, 2008.

[11] X Ren and J Malik. Learning a classification model for segmentation. In *ICCV*, pages 10–17. IEEE, 2003.

[12] L. Čehovin, M. Kristan, and A. Leonardis. Robust Visual Tracking using an Adaptive Coupled-layer Visual Model. *TPAMI*, 35(4):941–953, April 2013.

[13] M.-H. Yang, H. Lu, and F. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330. IEEE, November 2011.

[14] Z. Yin and R. Collins. On-the-fly object modeling while tracking. In *CVPR*, 2007.