An adaptive coupled-layer visual model for robust visual tracking

Luka Čehovin, Matej Kristan, and Aleš Leonardis
Faculty of Computer and Information Science, University of Ljubljana, Slovenia
Tržaška 25, Ljubljana, Slovenia
{luka.cehovin,matej.kristan,ales.leonardis}@fri.uni-lj.si

Abstract

This paper addresses the problem of tracking objects which undergo rapid and significant appearance changes. We propose a novel coupled-layer visual model that combines the target's global and local appearance. The local layer in this model is a set of local patches that geometrically constrain the changes in the target's appearance. This layer probabilistically adapts to the target's geometric deformation, while its structure is updated by removing and adding the local patches. The addition of the patches is constrained by the global layer that probabilistically models target's global visual properties such as color, shape and apparent local motion. The global visual properties are updated during tracking using the stable patches from the local layer. By this coupled constraint paradigm between the adaptation of the global and the local layer, we achieve a more robust tracking through significant appearance changes. Indeed, the experimental results on challenging sequences confirm that our tracker outperforms the related state-of-the-art trackers by having smaller failure rate as well as better accuracy.

1. Introduction

Visual tracking is an important research area in computer vision. In practice, the holistic approaches [6, 11, 5, 16], that globally model the target's appearance, have proven to be very successful. However, scenarios that contain rapid structural appearance changes present such models with serious difficulties. The reason is that such visual changes lead to reduced matches and drifting which eventually result in the trackers' failure.

To address these problems, approaches to tracking using sets of simple local parts have been proposed [15, 1, 3, 12, 9, 18]. Flock-of-features, proposed by Kölsch and Turk [9] and later extended by Hoey [7], was one of the early attempts. In flock-of-features a set of simple features (e.g. optical flow features) are used to independently track individual parts of the object. If a feature violates simple flocking rules based on a distance to other features, it is replaced by a new feature using a predefined fixed color distribution. Since the set of features is geometrically unconstrained, the tracker is likely to get stuck on the background and tracking fails. Yin and Collins [18] use the Harris corner detector to determine only the stable regions for tracking and enforce a single global affine transformation constraint to avoid drifting. However, the number of stable regions is highly dependent on the object texture. If the object's color is homogeneous, no stable regions will be found and the tracking will fail.

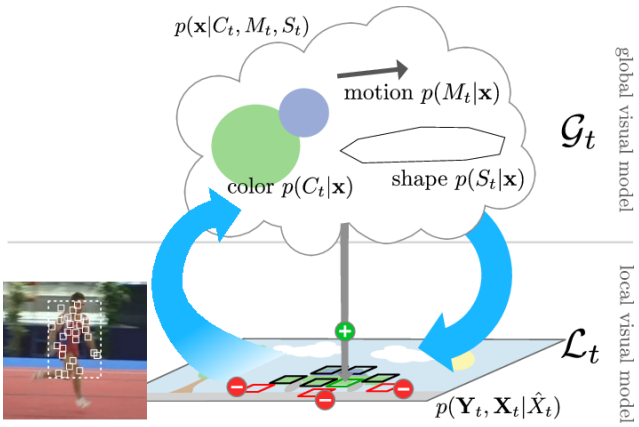

Figure 1. Illustration of the proposed coupled-layer visual model. The local layer is a geometrical constellation of visual parts that describe the target's local visual properties. The global layer encodes the target's global visual features in a probabilistic model.

To avoid such problems, Fan et al. [4] proposed to track a target with a set of kernels which are connected by a global affine transformation constraint. To enable handling slightly more involved changes in the appearance, Martinez and Binefa [15] connected multiple kernels together in triplets and constrained them with a local affine transformation. However, each kernel and the connections have to be carefully manually initialized based on the target's structural properties. This is undesirable in many tracking scenarios. Furthermore, the set of kernels is fixed and the tracker therefore cannot adapt to the target's larger appear-

ance changes.

A four-part fully-connected structure has been proposed by Badrinarayanan et al. [1] for face tracking. The visual model is composed of four patches, constrained by a flexible fully-connected graph. Because the number of parts is low, the problem can still be solved efficiently using a particle filter, however, this approach is not suitable for a larger sets of parts. Another drawback is that it requires manual initialization of positions for each patch. Chang et al. [3] used Markov random fields to encode the spatial constraints between parts. Only subsets of parts are connected in this case, making larger sets of parts easier to process. However, this approach still assumes that individual parts are manually initialized and cannot update the part set.

More flexible geometrical constraints that allow removing and adding parts during tracking have been presented by Kwon and Lee [12]. A star model connects all the parts to the center of the object. This model is simple enough that individual parts can be removed or added. The authors propose a likelihood function landscape analysis and part proximity to detect bad parts and remove them. New parts are added to the visual model using corner-like stable regions in the estimated object area. We consider this recent work to be the closest to our own research. While this approach provides a good mechanism for gradually adapting the visual model in a controlled manner, the mechanism of introducing new patches is rather nonrobust. The patch initialization fails for objects that lack textured surface and is not directly constrained to the object. On the other hand a rapid part removal can lead to false structural changes in the geometrical model and possible tracking failure.

In this paper we propose a coupled-layer visual model that combines the target's global and local appearance (Figure 1). The local layer $\mathcal{L}_t$ is a geometrical constellation of visual parts (patches) that describe the target's local visual/geometrical properties. As the target's appearance changes or a part of it gets occluded, some of the patches in the visual model cease to correspond to the target's visible parts. Those are identified and gradually removed from the model. The allocation of the new patches in the local layer is constrained by the global layer $\mathcal{G}_t$ that encodes the target's global visual features. The global layer maintains a probabilistic model of target's global visual features such as color, shape and apparent motion and is adapted during tracking. This adaptation is in turn constrained by focusing on the stable patches in the local layer.

The main contribution of the paper is the coupled constraint paradigm implemented within our Bayesian formulation of the two-layer model. We also integrate the proposed adaptive visual model within a Bayesian tracker that allows tracking through significant appearance changes. We argue that this robustness is achieved by the coupled-constrained updating of the visual model through the feedback loops between the global and the local layer. The experiments on the challenging sequences with significant appearance changes confirm that our tracker outperforms the state-of-the-art trackers by smaller failure rate and at greater (statistically significant) accuracy.

The rest of the paper is organized as follows: Section 2 describes the proposed visual model and the resulting tracker. In Section 3 we perform extensive experimental comparison with the state-of-the-art, and in Section 4 we discuss the method and draw the conclusions.

2. A coupled-layer visual model

During tracking, the proposed coupled-layer visual model is used as follows. Starting from an initial position (predicted by the Kalman filter in our case), the local model's geometrical structure is adapted to maximally explain the visual data – thus locating the target (Section 2.1). A mechanism is used to identify and remove the patches from the local visual model that no more correspond to the target (Section 2.2). The remaining patches are used to update the visual information of the global layer and then the global layer is used to allocate new patches in the local layer if necessary (Section 2.3).

2.1. 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:

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

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. The target's center is defined as the weighted average over the patches, i.e., $\mathbf{c}_t = \frac{1}{W_t}\sum_{i=1}^{N_t} w_t^{(i)}\mathbf{x}_t^{(i)}$, where W_t is a normalization factor $W_t = \sum_{i=1}^{N_t} w_t^{(i)}$. In the following we will denote the set of all patches at time-step t by $\mathbf{X}_t = \{\mathbf{x}_t^{(i)}\}_{i=1:N_t}$.

During tracking, we start from an initial estimate $\hat{\mathbf{X}}_t$ and the set of current image measurements $\mathbf{Y}_t$, and 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

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

where $p(\mathbf{z}^{(i)})$ is the i-th patch's prior and is approximated using the corresponding weight, i.e., $p(\mathbf{z}^{(i)}) = w_t^{(i)} / \sum_{j=1}^{N_t} w_t^{(j)}$. In our model, we assume that the position

of the i-th patch is dependent only on its direct neighbors, and we can write

$$p(\mathbf{Y}_t, \mathbf{X}_t | \hat{\mathbf{X}}_t, \mathbf{z}^{(i)}) \propto p(\mathbf{Y}_t, \mathbf{x}_t^{(i)} | \varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)}, \mathbf{z}^{(i)}), \quad (3)$$

where $\varepsilon_t^{(i)}$ and $\hat{\varepsilon}_t^{(i)}$ denote the set of the i-th patch's local neighbors' positions in the new and initial constellation, respectively. In our implementation, the local neighbors are the set of patches that are directly connected with the i-th patch in a Delaunay triangulated mesh of an entire set of patches. The conditional joint distribution can now be further decomposed in terms of visual and geometrical models as

$$\begin{aligned} &p(\mathbf{Y}_t, \mathbf{x}_t^{(i)} | \varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)}, \mathbf{z}^{(i)}) = \\ &p(\mathbf{Y}_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)}), \end{aligned} \quad (4)$$

where we have assumed that the measurement at the i-th patch is independent from the other patches. The visual model of the i-th patch is encoded by a gray-level histogram $\mathbf{h}_{\text{ref}}^{(i)}$ which is extracted when the patch is initialized in the constellation and remains unchanged during tracking. Let $\mathbf{h}_t^{(i)}$ be a histogram extracted at the current location of the patch $\mathbf{x}_t^{(i)}$. We define the visual likelihood of the i-th patch as

$$p(\mathbf{Y}_t | \mathbf{x}_t^{(i)}) \propto e^{-\lambda_{\text{v}} \rho(\mathbf{h}_{\text{ref}}^{(i)}, \mathbf{h}_t^{(i)})}, \quad (5)$$

where $\rho(\cdot, \cdot)$ is the Bhattacharyya distance between the histograms [16]. We constrain the local geometry using an elastic deformation model

$$p(\mathbf{x}_t^{(i)} | \varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)}) \propto e^{-\lambda_{\text{g}} || \mathbf{x}_t^{(i)} - \mathbf{A}(\varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)}) \hat{\mathbf{x}}_t^{(i)} ||}, \quad (6)$$

where $\mathbf{A}(\varepsilon_t^{(i)}, \hat{\varepsilon}_t^{(i)})$ is an affine transformation matrix computed from correspondences between the i-th patch's initial and current neighborhoods. Note that this geometric model assumes that the deformations of the constellation are locally approximately affine. Therefore, during adaptation of the local layer to the target's current appearance, we seek an approximately affine deformation of an initial set of patches $\hat{\mathbf{X}}_t$ that maximizes the joint probability in (2).

We determine the unknown deformation by optimizing (2) for $\mathbf{X}_t$ using the standard cross-entropy method [17]. However, due to the high dimensionality of the problem at hand, (2) may contain many local maxima that may cause the method to take a long time to converge. We therefore write our deformation model as a composition of a globally affine deformation $\mathbf{A}_t^{\text{G}}$, that is equal for all patches, and of local perturbations $\Delta_t^{(i)}$ which may vary between the patches:

$$\mathbf{x}_t^{(i)} = \mathbf{A}_t^{\text{G}} \hat{\mathbf{x}}_t^{(i)} + \Delta_t^{(i)}. \quad (7)$$

In our implementation we therefore first optimize (2) w.r.t. the global affine deformation $\mathbf{A}_t^{\text{G}}$. After convergence, we fix the value of $\mathbf{A}_t^{\text{G}}$ and sequentially optimize the positions of each patch $\mathbf{x}_t^{(i)}$.

2.2. Updating the local layer

Recall from (1) that there is a weight $w_t^{(i)}$ associated with each patch that reflects the relevance of the corresponding patch in the mixture of patches. After adapting the set of patches to the target's appearance, as described in the previous section, each patch is analyzed and its weight is increased or decreased by Δ_w by applying the following two consistency rules:

- **Visual consistency:** If the Bhattacharyya distance between the patch's reference and the current histogram exceeds a threshold T_{histHi} then its weight is decreased; if the distance falls below a threshold T_{histLo} the weight is increased.
- **Drift from majority:** If the median of the distances from the patch to all other patches in the set is greater than a predefined threshold T_{major}, then the patch's weight is decreased.

The weight of a patch can be interpreted as a frequency at which each patch has been selected as belonging to the object (increasing weight) minus the frequency at which the patch was selected as a possible outlier (decreasing weight). When normalized, these weights can be regarded as a probability that a patch belongs to the object. Patches with low probability (lower than T_{R}) are considered as either outdated or mispositioned and are removed from the set. To allow a good coverage of the target in the image, new patches have to be added in the local layer. The patches are allocated by sampling their position from a probability density function (pdf) that determines locations in the image which are likely to contain the target. This pdf is constructed from the global layer and is described in the next section. The weight $w_t^{(i)}$ of the allocated patch is initialized with a value of twice the threshold for patch removal, i.e., $w_0 = 2T_{\text{R}}$. The remaining question is *how many* patches should be allocated. Let $\tilde{N}_t$ denote the number of patches in the local layer after removing the irrelevant patches. We define N_t^{cap} to be the local layer's *capacity*, i.e., the maximum number of patches allowed in the local layer at time-step t. To allow the number of allocated patches to grow with the target's size, we always try to allocate at most $N_t^{\text{all}} \leq N_t^{\text{cap}} - \tilde{N}_t + 1$ new patches. To prevent sudden significant changes in the estimated capacity, we adapt it using the autoregressive scheme:

$$N_{t+1}^{\text{cap}} = \alpha_{\text{cap}} N_t^{\text{cap}} + (1 - \alpha_{\text{cap}}) N_t, \quad (8)$$

where $N_t = N_t^{\text{all}} + \tilde{N}_t$ and α_{cap} is an exponentially forgetting factor.

2.3. The global layer

The global layer $\mathcal{G}_t$ captures the target's global visual properties, in particular color C_t, apparent motion M_t, and

shape S_t,

$$\mathcal{G}_t = \{C_t, M_t, S_t\}. \quad (9)$$

When required, this information is used to allocate new patches in the local layer. The allocation is implemented by drawing positions from the following distribution

$$p(\mathbf{x}|C_t, M_t, S_t) \propto p(C_t, M_t, S_t|\mathbf{x}). \quad (10)$$

Assuming that the visual cues are independent given a position $\mathbf{x}$, then (10) factors as

$$p(\mathbf{x}|C_t, M_t, S_t) \propto p(C_t|\mathbf{x})p(M_t|\mathbf{x})p(S_t|\mathbf{x}). \quad (11)$$

In the following we describe the models for each of the cues. The global **color model** is encoded by two HSV histograms $\mathbf{h}_t^F$ and $\mathbf{h}_t^B$, the first corresponding to the target and the second to the background. Let $I(\mathbf{x})$ be a pixel value at the position $\mathbf{x}$ in image I. Using the histograms, the probability that a pixel corresponds to the background or foreground is $p(x|\mathrm{F}) = \mathbf{h}_t^F(I(\mathbf{x}))$ and $p(x|\mathrm{B}) = \mathbf{h}_t^B(I(\mathbf{x}))$, respectively. The likelihood that a pixel at the location $\mathbf{x}$ belongs to the target is therefore

$$p(C_t|\mathbf{x}) = \frac{p(\mathbf{x}|\mathrm{F})p(\mathrm{F})}{p(\mathrm{F})p(\mathbf{x}|\mathrm{F}) + (1 - p(\mathrm{F}))p(\mathbf{x}|\mathrm{B})}. \quad (12)$$

Both histograms are updated during tracking as follows. After the local layer is fitted to the target (Section 2.1), a histogram $\hat{\mathbf{h}}_t^F$ is extracted in the current image from the regions that correspond to the patches of the local layer. The background histogram $\hat{\mathbf{h}}_t^B$ is extracted from a ring-shaped region defined by the convex hull of the patches in the local layer. These histograms are used to update the global color model by a simple autoregressive scheme

$$\begin{aligned} \mathbf{h}_{t+1}^F &= \alpha_F \mathbf{h}_t^F + (1 - \alpha_F)\hat{\mathbf{h}}_t^F \\ \mathbf{h}_{t+1}^B &= \alpha_B \mathbf{h}_t^B + (1 - \alpha_B)\hat{\mathbf{h}}_t^B, \end{aligned} \quad (13)$$

where α_F and α_B are fixed constants that determine the rate of adaptation.

The **apparent motion model** is defined by the local motion model from [11]. Briefly, the local motion model [11] first determines salient points $\{\mathbf{x}_i\}_{i=1}^{N_\mathrm{s}}$ with sufficient texture in the image. It then computes the motion likelihood $p(\mathbf{x}_i|M_t)$ at each salient point $\mathbf{x}_i$ by comparing the local velocity of a pixel $\mathbf{v}(\mathbf{x}_i)$ (estimated by Lucas-Kanade optical flow [14]) with the global velocity $\mathbf{v}_t$ estimated by the tracker. As in [11], the motion likelihood at salient point $\mathbf{x}_i$ is defined as

$$p(\mathbf{x}_i|M_t) \propto (1 - w_\mathrm{noise})e^{-\lambda_\mathrm{M}(d(\mathbf{v}(\mathbf{x}_i), \mathbf{v}_t))} + w_\mathrm{noise}, \quad (14)$$

where $d(\mathbf{v}(\mathbf{x}_i), \mathbf{v}_t))$ is the distance between two velocities and w_noise is uniform noise. Finally, to obtain a dense estimate, the set of salient points is convolved with a smoothing kernel. We therefore define the motion likelihood as

$$p(M_t|\mathbf{x}) \propto \frac{1}{K}\sum_{i=1}^{N_\mathrm{s}} p(\mathbf{x}_i|M_t)\Phi_{\mathbf{\Sigma}}(\mathbf{x} - \mathbf{x}_i), \quad (15)$$

where K is a normalization factor, $\Phi_{\mathbf{\Sigma}}(\mathbf{x})$ is a Gaussian kernel with covariance $\mathbf{\Sigma}$ and N_s is the number of salient patches. The covariance is estimated automatically from the weighted set of salient points using the multivariate Kernel Density Estimation [10].

The **shape model** is a weighted superposition of the past Δt approximate object shapes. An approximate object shape at time-step t is defined as an object-centered region P_t, which is calculated by a convex envelope over the patches from the local layer. To maintain the growing capability we dilate each hull by the size of a local patch. We define a function $s(\mathbf{x}, P_t) \equiv 1$ if $\mathbf{x} \in P_t$ and 0 otherwise and the shape likelihood model for a pixel at $\mathbf{x}$ is thus defined as

$$p(S_t|\mathbf{x}) \propto \sum_{i=0}^{\Delta_t} {\alpha_\mathrm{S}}^i s(\mathbf{x}, P_{t-i}), \quad (16)$$

where α_S is a weighting factor which reduces the influence of the older shapes.

As mentioned above, (11) is used for allocating new patches in the local layer. We do not sample (11) directly, but rather discretize it first, by calculating its value for each pixel in the image. This discretized distribution is then used to draw positions for new patches from the potential target region. To make sure that the patches are allocated only in regions whose likelihood of containing the target is high enough, we set to zero those regions of the discretized distribution, whose value is smaller than 30% of the maximal value from $p(\mathbf{x}|C_t, M_t, S_t)$. To avoid duplicating patches in the local layer, the regions of the discretized distribution that correspond to existing patches are set to zero.

2.4. Tracking with the coupled-layer visual model

Recall that the proposed coupled-layer visual model starts from an initial estimate of the target's position and then refines its estimate by adapting to the current image as described in Section 2.1. The center of the target can then be identified as a weighted average $\mathbf{c}_t$ of the patches' positions. During tracking we require prediction of the local layer's patches to initialize the adaptation of the visual model. We also require an estimate of the target's velocity in the global layer's apparent motion model. We therefore apply a Kalman filter [8] with a nearly-constant velocity (NCV) dynamic model [13] to filter the estimates of the target's center $\mathbf{c}_t$. Thus, at time-step t, the target's velocity $\hat{\mathbf{v}}_t$ estimated by the Kalman filter is used to initialize the local layer patches $\hat{\mathbf{X}}_t = \{\hat{\mathbf{x}}_t^{(i)}\}_{i:1:N_t}$ by predicting the location

of the patches from the previous frame:

$$\hat{\mathbf{x}}_t = \mathbf{x}_{t-1}^{(i)} + \hat{\mathbf{v}}_t^{(i)}. \quad (17)$$

In the first frame, the tracker is manually initialized by placing a rectangular region over the target. We give no other a-priori structural information and the set of patches in the local layer is uniformly initialized in a grid pattern within the rectangular region. The weights of the patches are initialized to the value w_0. We summarize the relevant steps of our tracker in Algorithm 1.

Algorithm 1 The coupled-layer visual tracker.

Initialization:

i **Input:** Place a rectangular region over a target.

ii Distribute patches in a regular grid in the region and assign uniform weights.

Tracking: For time-step $t = 1, 2, 3 \ldots$

1. Predict the target's velocity $\hat{\mathbf{v}}_t$ using the Kalman filter and initialize the local-layer patches with the NCV model (17).
2. Adapt the local layer patches by maximizing $p(\mathbf{Y}_t, \mathbf{X}_t | \hat{\mathbf{X}}_t)$ (Section 2.1), recalculate the target's center $\mathbf{c}_t$ and update the Kalman filter estimate.
3. Identify/remove irrelevant patches from the local layer (Section 2.2).
4. To maintain numerical stability (e.g. Delaunay triangulation works better if the input points are not too close to each other) and decrease redundant comparisons, merge patches in the local layer that are too close to each other.
5. Using the remaining patches, update the visual cues of the global layer (Section 2.3).
6. If required, construct a discretized distribution $p(\mathbf{x}|C_t, M_t, S_t)$ and sample positions of new patches for the local layer.

3. Experimental results

We have analyzed the performance of the proposed local-global tracker (LGT) from Algorithm 1 on several examples of tracking either a nonrigid object or an object that undergoes a significant appearance change. Our tracker has been implemented in Matlab/C and runs at approximately 4 frames per second on an Intel Core 2 Duo 6600. The parameters in our tracker were set as follows. The maximum number of iterations in the cross-entropy was 10, with 50 samples per iteration. We set $\lambda_{\mathrm{v}} = 0.1$ and $\lambda_{\mathrm{g}} = 0.015$. For the adaptation of the local layer (Section 2.2) the following parameters were used: $\Delta_w = 0.1$, $T_{histLo} = 0.4$, $T_{histHi} = 0.8$, $T_{major} = 40$, $T_R = 0.1$ and $\alpha_{\mathrm{cap}} = 0.8$. To update the global layer, parameter values $\alpha_F = 0.95$, $\alpha_B = 0.5$, $\lambda_M = 1$, $w_{noise} = 0.01$, $\Delta t = 7$, and $\alpha_{\mathrm{S}} = 0.7$ were used. We would like to emphasize that all the parameters were kept constant for all the experiments.

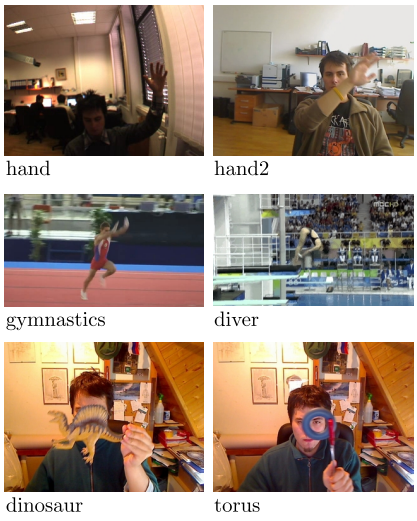

Figure 2. Samples from the experimental video sequences.

We have compared our tracker, i.e. LGT, with five related state-of-the-art reference trackers, which address the problem of object appearance changes: a color-based particle filter [16] (PF), an online boosting tracker [5] (OBT), a flock-of-features tracker [9] (FOF), a piecewise-affine kernel tracker [15] (PAKT) and the basin-hopping Monte Carlo tracker [12] (BHMC). The experiments involved tracking a hand, a human body, and objects with challenging view changes (Figure 2). The basic properties of the experimental sequences are collected in Table 1[1].

Table 1. An overview of the video sequences.

Sequence	Type	Comments	Len.
hand	arti. body part	rapid motion	242
hand2	arti. body part	rapid motion	267
gymnast.	articulated	rapid motion	206
diver	articulated	rotation	214
dinosaur	rigid	elab. struct.	324
torus	rigid	empty center	262

The target was tracked in each sequence $R = 30$ times by each tracker. For comparison, we recorded the number

[1]The annotated sequences, as well as a reference implementation of the tracker are available at `http://vicos.fri.uni-lj.si/lukacu/research/tracking/`.

Figure 3. Results for the *hand* sequence. Results are shown for trackers *FOF* (first row), *PF* (second row) and *LGT* (last row).

of times each tracker failed and had to be reinitialized. We also recorded the tracked trajectories. The tracking failure was automatically determined by measuring the overlap between the ground-truth region Ω^t_{gt} and the region estimated by the tracker Ω^t. The overlap was measured as $F(\Omega^t_{\text{gt}}, \Omega^t) = \Omega^t_{\text{gt}} \cap \Omega^t / \Omega^t_{\text{gt}} \cup \Omega^t$. A failure was proclaimed at time-step t if $F(\Omega^t, \Omega^t_a) < 0.09$. This threshold is based on our observation of the behavior of the estimated region, produced by a tracker vs. the ground truth region.

To evaluate the tracker accuracy with respect to the other trackers, we have performed a one-sided standard hypothesis test [2] on the estimated trajectories.

3.1. Results

Table 2 shows the average failure rates for each tracker. We see that the LGT is indeed superior to the reference trackers as the average failure rate is the lowest for all the sequences. Looking at the number of failures per sequence, we also see that the sequence *hand2* was the most difficult to track for all trackers. Visual properties of the hand, such as color, are similar for the entire arm, making trackers that rely heavily on color more vulnerable to drifting. Furthermore, due to homogeneous color, skin contains only few distinct local regions, which makes it difficult to reliably estimate local motions on the object. The problem of color ambiguity and *background clutter* was also apparent in the sequence *hand* in Figure 3, where the PF tracker (second row), which relies only on color information, confused the head for the hand on the third image from the sequence. Because of the difficulty of estimating the local motion from small regions, the FOF tracker (first row), which uses a set of optical flow features for tracking, failed. On the other hand, the LGT tracker succeeds in tracking (third row) since it integrates multiple cues at a global level to handle background clutter and enforces geometrical constraints at a local level to handle local ambiguity.

The sequences *gymnastics* and *diver* are the only two sequences that include camera motion (following the target). It is worth noting that the objects do not move much spatially in these sequences, but rather significantly change their appearance. PAKT and BHMC do not explicitly assume the object's translational motion (do not estimate the object's velocity), but rather assume Brownian-like motion. For this reason their failure rate is somewhat lower for these two sequences in comparison to other sequences. Nevertheless, the LGT outperformed both trackers in these sequences. Figure 4 compares the BHMC tracker (first row) and LGT tracker (second row) on several frames of the *gymnastics* sequence, in which the target significantly changes its appearance as well as scale. We can see from the estimated bounding boxes that the size of the object is often poorly estimated by the BHMC tracker which leads to failures (Table 2). On the other hand, the LGT successfully tracks the target through the scale change.

Table 2. Average number of failures per sequence.

	PAKT [15]	FOF [9]	PF [16]	BHMC [12]	OBT [5]	**LGT**
hand	22.6	10.0	4.3	29.9	10.0	**0.2**
hand2	40.0	13.1	10.1	45.4	31.0	**1.9**
gymnast.	3.0	3.7	4.7	9.7	4.0	**0.2**
diver	2.4	2.2	4.3	3.9	7.0	**1.2**
dinosaur	8.2	2.7	2.2	15.6	7.0	**0**
torus	10.6	6.0	2.5	23.4	13.0	**0**

The advantages of the LGT tracker are also evident in the sequences *dinosaur* and *torus* for the case of rigid objects with more complex structure that undergo rapid orientation and translation changes with respect to the camera. Even though these kinds of objects are not as deformable as a human body or a hand, the changes in the appearance are still hard to describe without a predefined geometrical model for a specific object. As seen in Figure 5, when tracking a torus, the PAKT and OBT reference trackers drift from the object several times during the sequence, while the LGT tracker

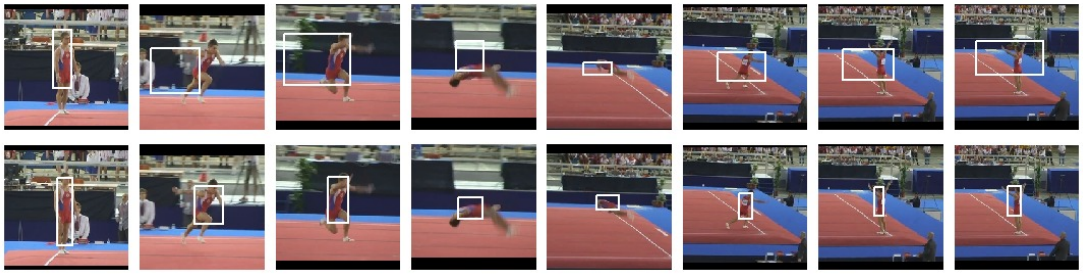

Figure 4. Results for the *gymnastics* sequence. Results are shown for trackers *BHMC* (first row) and *LGT* (second row).

successfully accomplishes the task. In the case of the PAKT tracker (first row) the problem lies in its inability to follow fast movements because of the locality of the optimization and the limited adaptation capabilities due to a fixed parts set. The OBT tracker (second row) on the other hand fails many times because it focuses on the more visually interesting central region, which, however, belongs to the background. The LGT tracker (third row) does not have these problems and can successfully track the object throughout the sequence.

Table 3. RMS errors with respect to the ground truth. In all cases the LGT produced smaller RMSE and $(\cdot)^*$ denotes that the difference was statistically significant.

	PAKT [15]	FOF [9]	PF [16]	BHMC [12]	OBT [5]	**LGT**
hand	18.5^*	19.7^*	14.4^*	27.4^*	17.5^*	**9.1**
hand2	18.7^*	17.4^*	16.6^*	26.1^*	22.5^*	**10.3**
gymnast.	17.1^*	23.1^*	22.8^*	27.6^*	21.3^*	**11.3**
diver	18.1^*	14.6	16.5^*	21.1^*	17.1^*	**13.7**
dinosaur	23.6^*	19.2^*	23.3^*	35.1^*	30.3^*	**11.5**
torus	15.2^*	14.8^*	16.4^*	21.5^*	14.8^*	**5.1**

Table 3 shows the tracking accuracy in terms of the average RMSE. From the comparison of the RMSEs of the reference trackers and the proposed tracker we can conclude that the proposed tracker, LGT, outperforms all the reference trackers in accuracy at a standard significance level $\alpha = 0.05$ ($L_\alpha = 1.564$) except in one case (tracker FOF on the sequence *diver*) where the difference is not *statistically significant*. The better accuracy can be largely attributed to the two-stage optimization of the local layer, that first finds a good globally affine match for the entire set of the patches and then fine-tunes positions of individual patches to better match the target's new appearance. The difference is less significant in the cases of the *gymnastics* and *diver* sequences because the targets do not move very much spatially, which makes the drawbacks of some of the related trackers less apparent.

4. Discussion and conclusion

We have proposed a coupled two-layer visual model for efficient tracking of targets that undergo significant appearance changes. The proposed model is a coupled combination of a local and global layer. The local layer is a set of local patches that geometrically constrain the changes in the target's appearance. The set probabilistically adapts to the target's appearance by maximizing the joint distribution over the model's geometrical constraints and visual observations. As the target's appearance significantly changes, some of the patches in the visual model cease to correspond to the target's visible parts. Those patches are identified by the local layer and gradually removed from the model. The allocation of the new patches in the local layer is constrained by the global layer that encodes the target's global visual features. The global layer maintains a probabilistic model of the target's global visual features such as color, shape, and the apparent motion and is adapted during tracking. This adaptation is in turn constrained by focusing on the stable patches in the local layer. We believe that it is exactly this constrained coupled updating between the layers that results in the robust tracking.

We have incorporated the proposed visual model in a tracker and compared the tracker to the state-of-the-art on several challenging sequences. The results show that our tracker outperforms the related trackers by smaller failure rate and at a greater accuracy. The experiments have shown that even in the cases when the background's color is similar to the target's, tracking will not fail. The reason is that the global layer uses many more features, such as foreground-background similarity, shape, local motion, and temporal proximity from the Kalman filter to determine which regions in the image potentially contain the target. Therefore new patches are more likely initialized on the target. Only after these patches have been validated by the local layer over several frames, they start to play a stronger role in the model. Similarly, the global layer is updated only by using the stable patches from the local layer. These constrained feedbacks between the two layers, allow the tracker to track the target through scale and appearance changes as shown in the experiments. In the same respect, the tracker is ex-

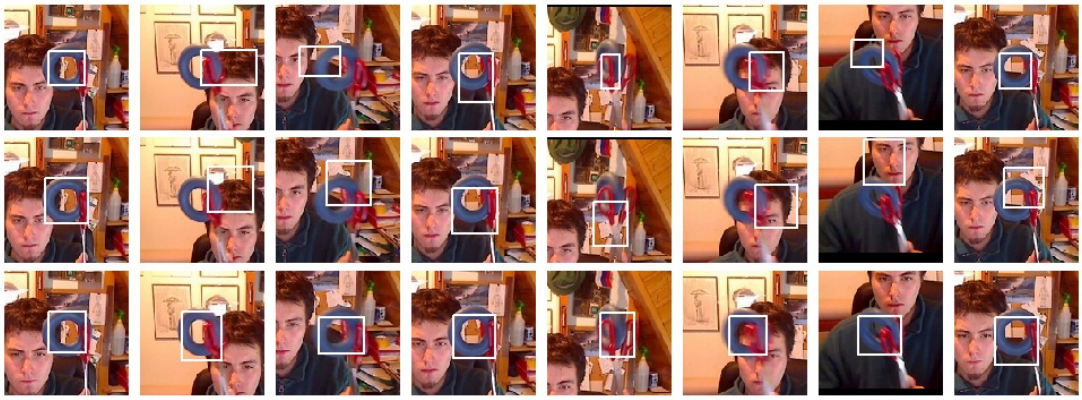

Figure 5. Results for the *torus* sequence. Results are shown for trackers *PAKT* (first row), *OBT* (second row) and *LGT* (last row).

pected to handle partial occlusions, since the occluded parts are removed from the model. As long as at least some of the occluder's visual properties are different from the target's, new patches will only be allocated on the target. On the other hand, since we do not impose a strong prior model on the targets' appearance and shape this makes it difficult to cope with situations when the objects appearance is very similar to the background, in which case more conservative update mechanism would be required.

In future work, we will analyze the tracker's performance if more complex descriptors are used at the local layer. Note that we currently employ three visual cues in the global layer (foreground/background color, apparent local motion, and shape), however, other cues can be used as well. Since the discriminative properties of different visual cues may vary with time, our tracker would benefit from a mechanisms for on-line selection of salient cues in the global layer. While the constrained adaptation of the model can address particular cases of occlusion, we have so far not explored any mechanisms for explicitly handling the complete long-lasting occlusions. In the future research we will extend our framework to address these issues as well.

Acknowledgements: We thank the authors of [5, 15, 12] for providing the reference implementations of their trackers. This research was in part supported by: ARRS projects J2-3607 and J2-2221 and EU project CogX (FP7-ICT215181-IP).

References

[1] V. Badrinarayanan, F. L. Clerc, L. Oisel, and P. Perez. Geometric layout based graphical model for Multi-Part object tracking. In *IWVS*, 2008. 1, 2

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 1 edition, 2001. 6

[3] W. Chang, C. Chen, and Y. Hung. Tracking by parts: A bayesian approach with component collaboration. *IEEE TSMC, Part B: Cybernetics*, 39(2):375–388, 2009. 1, 2

[4] Z. Fan, M. Yang, and Y. Wu. Multiple collaborative kernel tracking. *IEEE TPAMI*, 29(7):1268–1273, 2007. 1

[5] H. Grabner, M. Grabner, and H. Bischof. Real-Time tracking via on-line boosting. In *BMVC*, 2006. 1, 5, 6, 7, 8

[6] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. In *IEEE CVPR*, volume 1, pages 790–797, 2004. 1

[7] J. Hoey. Tracking using flocks of features, with application to assisted handwashing. In *BMVC*, 2006. 1

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *J. Basic. Eng.*, 82:34–45, 1960. 4

[9] M. Kölsch and M. Turk. Fast 2D hand tracking with flocks of features and Multi-Cue integration. In *CVPRW*, volume 10, page 158, Washington, DC, USA, 2004. 1, 5, 6, 7

[10] M. Kristan, A. Leonardis, and D. Skočaj. Multivariate online kernel density estimation with gaussian kernels. *Pattern Recognition*, 44:2630–2642, October 2011. 4

[11] M. Kristan, J. Perš, S. Kovačič, and A. Leonardis. A local-motion-based probabilistic model for visual tracking. *Pattern Recognition*, 2009. 1, 4

[12] J. S. 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 *IEEE CVPR*, pages 1208–1215, 2009. 1, 2, 5, 6, 7, 8

[13] X. R. Li and V. J. P. Survey of maneuvering target tracking: Dynamic models. *IEEE AES*, 39(4):1333–1363, 2003. 4

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981. 4

[15] B. Martinez and X. Binefa. Piecewise affine kernel tracking for non-planar targets. *Pattern Recognition*, 41(12):3682–3691, 2008. 1, 5, 6, 7, 8

[16] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based probabilistic tracking. In *ECCV*, volume 1, pages 661–675. Springer-Verlag, 2002. 1, 3, 5, 6, 7

[17] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *EJOR*, 99(1):89–112, 1997. 3

[18] Z. Yin and R. Collins. On-the-fly object modeling while tracking. In *IEEE CVPR*, pages 1–8, 2007. 1