Robust visual tracking using an adaptive coupled-layer visual model

Luka Čehovin , Matej Kristan, *Member, IEEE,* and Aleš Leonardis, *Member, IEEE*

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 by interlacing two layers. 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 these 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. We experimentally compare our tracker to eleven state-of-the-art trackers. The experimental results on challenging sequences confirm that our tracker outperforms the related trackers in many cases by having smaller failure rate as well as better accuracy. Furthermore, the parameter analysis shows that our tracker is stable over a range of parameter values.

Index Terms—Image processing and computer vision, tracking.

1 Introduction

Visual tracking is an important research area in computer vision. In practice, the holistic approaches [1], [2], [3], [4], that globally model the target's appearance, have proven to be very successful. However, scenarios that contain rapid structural and 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. Updating of the visual model [5], [6], [7], [8], [9] potentially increases the robustness of the tracker, but at the same time poses additional questions regarding when should the visual model be updated and which parts of it should be updated. For holistic approaches the entire model is updated at once, increasing the chance that the valid part of the visual information is corrupted by the new data. This can happen because the tracker fails to optimally predict the new position of the object, therefore updating the visual model with the new data, that does not belong to the object, or because the tracker simply relies on the ambiguous data (e.g. features that do not separate object from the neighborhood).

- *L. Čehovin and M. Kristan are with the Faculty of Computer and Information Science at the University of Ljubljana, Ljubljana, Slovenia. E-mail: {luka.cehovin,matej.kristan}@fri.uni-lj.si.*
- *A. Leonardis is with the School of Computer Science and Centre for Computational Neuroscience and Cognitive Robotics, University of Birmingham, United Kingdom and with the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. E-mail: a.leonardis@cs.bham.ac.uk, ales.leonardis@fri.uni-lj.si*

One way to address appearance changes is to use more different holistic trackers together [10], [11], [12]. Each of the tracker behaves well in certain situations and an increase in performance can be achieved by switching among them. This approach does not, however, suit tracking scenarios where the object is not rectangular and/or it deforms geometrically, therefore changing its shape.

In recent years new approaches to tracking using sets of simple local parts have been proposed. These parts are loosely connected [13], [14], [15], [16], [17], [18], [19] or not connected at all [20], [21], allowing some degree of spatial deformation. Besides the geometrical deformability, that is useful for short-term tracking, this approach also offers a more flexible mechanism for updating the visual model. Because each local part is a visual model on its own, the updating can be performed by removing only parts that exhibit signs of drifting and adding new parts while keeping the parts that are still performing well intact [16], [21].

In this paper we describe a novel visual model that combines a set of pars together with global appearance information of the object. Our model can be updated by removing and adding parts using a global visual model that is also updated during tracking. We have recently published some preliminary results of this model in [22]. In this paper we give a more detailed description of our visual model and the resulting tracker. We also include a more detailed experimental analysis and we have extended the comparative experiments by six additional video sequences and five additional reference trackers as well as added several experiments that investigate

the properties of the proposed visual model. We have also extended the discussion of the results and added several figures that better explain the tracker.

In the following two subsections we describe the existing methods that are most closely related to our work. We point out the main differences to our approach and state our contributions.

1.1 Related work

Part-based visual models have been investigated by many researchers in the context of visual tracking. Flock-of-features, proposed by Kölsch and Turk [21] and later extended by Hoey [20], was one of the early attempts of splitting a holistic visual model into segments. 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 the tracking fails. Yin and Collins [17] 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, they also assume that the shape of the object can be approximated with an ellipsoid and that the object does not deform. This limits the generality of the tracker. Besides that 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.

To avoid problems with locating a stable region, Fan et al. [23] 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 [13] 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 appearance changes.

A four-part fully-connected structure has been proposed by Badrinarayanan et al. [14] 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 set of parts. Another drawback is that it requires manual initialization of positions for each patch. Chang et al. [15] used Markov random fields to encode the spatial constraints between the 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.

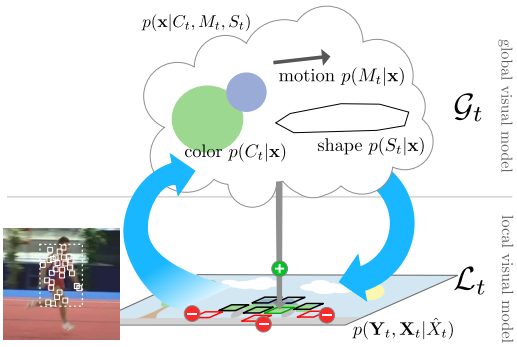

Fig. 1. Illustration of the proposed coupled-layer visual model. The local layer is a geometrical constellation of visual parts that describe the target's local visual properties. The global layer encodes the target's global visual features in a probabilistic model.

More flexible geometrical constraints that allow removing and adding parts during tracking have been presented by Kwon and Lee [16]. 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 non-robust. 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.

1.2 Our approach

In this section we describe the main idea behind our coupled-layer visual model. The coupled-layer visual model is organized in two layers. The local layer $\mathcal{L}_t$ is a geometrical constellation of visual parts (patches) that describe the target's local visual/geometrical properties (Figure 1). As the target's appearance changes or a part of the object 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.

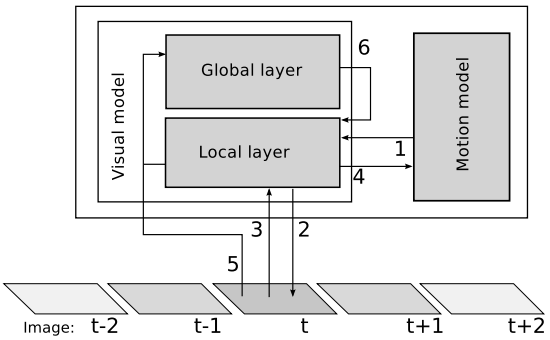

Fig. 2. A schematic overview of the main steps in processing of a single frame. 1 – prediction from the motion model, 2 – matching the local layer (Section 2.1), 3 – updating weights and removing patches (Section 2.2), 4 – updating the motion model, 5 – updating the global layer (Section 2.3), 6 – adding new patches.

The main contribution of the paper is the coupled constraint paradigm implemented within our Bayesian formulation of the two-layer model. We also integrate our visual model within a Bayesian tracker that allows tracking through significant appearance changes. We argue that the tracker's 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 our method and draw conclusions.

2 A COUPLED-LAYER VISUAL MODEL

In this section we first overview our coupled-layer visual model by iterating through the steps of matching and updating the model during tracking. An overview of the steps is shown in Figure 2. Using a new image from the image sequence we start 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 do not correspond to the target (Section 2.2). The remaining patches are used to update the motion model and 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 a 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

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

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. In our model, we assume that the position of the i-th patch is dependent only on its direct neighbors, and we can write

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

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 as shown in Figure 3. The distribution in the right-hand side of (3) can now be further decomposed in terms of visual and geometrical models as

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

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

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

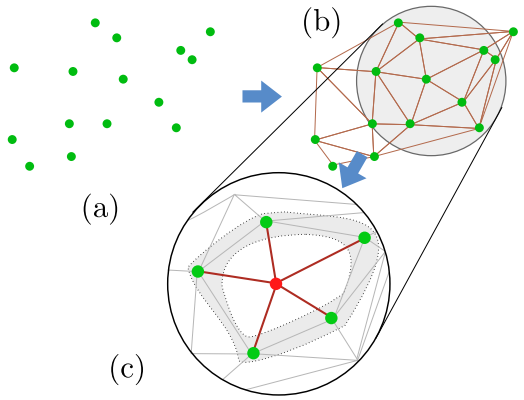

Fig. 3. Determining patch neighborhood using Delaunay triangulated mesh. The figure shows: positions of the patches (a), Delaunay mesh edges for the point set (b), neighborhood of a single patch (c).

where $\rho(\cdot,\cdot)$ is the Bhattacharyya distance between the histograms [3] and λ_{v} is a constant factor. We constrain the local geometry using an elastic deformation model

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

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 and λ_{g} is a constant factor. 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 $\tilde{\mathbf{X}}_t$ of an initial set of patches $\hat{\mathbf{X}}_t$ that maximizes the joint probability in (2)

$$\tilde{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t} p(\mathbf{Y}_t, \mathbf{X}_t|\hat{\mathbf{X}}_t). \qquad (7)$$

We determine the unknown deformation $\tilde{\mathbf{X}}_t$ by optimizing (2) using the standard cross-entropy method [24]. However, due to the high dimensionality of the problem at hand, (2) may contain many local maxima. We therefore use an approach that is based on the idea of Graduated Non-Convexity [25]. We write our deformation model as a composition of a globally affine deformation $\mathbf{A}_t^{\mathrm{G}}$, that is equal for all patches, and of local perturbations $\Delta_t^{(i)}$ which may vary between the patches:

$$\mathbf{x}_t^{(i)} = \mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)} + \Delta_t^{(i)}. \qquad (8)$$

In our implementation we therefore first optimize (2) w.r.t. the global affine deformation $\mathbf{A}_t^{\mathrm{G}}$. The problem is considered in a five-dimensional problem space of $G = [t_x, t_y, r, s_x, s_y]$, where t_x and t_y represent the target's position, r represents rotation and s_x and s_y represent scale. The five parameters define a transformation matrix $\mathbf{A}_t^{\mathrm{G}}$ as

$$\mathbf{A}_t^{\mathrm{G}} = \begin{bmatrix} s_x\cos(r) & -\sin(r) & t_x \\ \sin(r) & s_y\cos(r) & t_y \\ 0 & 0 & 1 \end{bmatrix}. \qquad (9)$$

Fig. 4. Global matching of patch set using the cross-entropy method.

- **Input:** The set of patches $\hat{\mathbf{X}}_t$.
- **Initialization:** Set the initial mean and covariance $\mu_0 = \mathbf{0}$ and $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_G$.
- For $j = 1 \ldots M_G$:
  - Sample S_G samples of parameter values from $\mathcal{N}(\mu_{j-1}, \mathbf{\Sigma}_{j-1})$.
  - For each sample construct the corresponding affine transformation and evaluate (2) for $\tilde{\mathbf{X}}_t = \{\mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)}\}_{i=1\ldots N_t}$.
  - Select E_G best samples according to (2) and use them to recalculate μ_j and $\mathbf{\Sigma}_j$.
  - Break the loop if $det(\Sigma_j) < 0.1$.
- **Output:** The optimal global affine transformation $\mathbf{A}_t^{\mathrm{G}}$, constructed from the parameters of μ_j.

The cross entropy method iteratively searches for an optimal combination of parameters according to the cost function by updating the candidate probability distribution over the parameters of G. In our case we model the probability distribution as a normal distribution, defined by a mean value $\mu_.$ and the covariance matrix $\mathbf{\Sigma}_.$ as seen in the overview of the algorithm in Figure 4. At the beginning the parameters of the distribution are initialized using constant values. The idea of the cross-entropy method is that the parameters of the distribution are iteratively updated using only the E_G best samples of the total S_G samples sampled from the distribution $\mathcal{N}(\mu_., \mathbf{\Sigma}_.)$. This can be done by computing a weighted mean and a weighted covariance of the best samples using the cost function to determine the weights. This process is then repeated for M_G iterations or until the distribution collapses, i.e. the determinant of the covariance matrix falls bellow 0.1.

Note that in the case of the global optimization, (2) can be simplified. Because a global affine transformation constraint is enforced in the problem space it is clear that $\mathbf{A}(\varepsilon_t^{(i)},\hat{\varepsilon}_t^{(i)})$ from (6) equals to $\mathbf{A}_t^{\mathrm{G}}$ and that the value of (6) is 1 for every patch. Therefore, (4) can be reduced to a weighted sum of a visual likelihood for every patch.

After convergence of the global optimization, the value of $\mathbf{A}_t^{\mathrm{G}}$ is fixed and the positions of each patch $\mathbf{x}_t^{(i)}$ are additionally optimized by using a stochastic coordinate descent in the cross-entropy framework. The search for optimal position of every patch is represented as a cross-entropy optimization problem in a two dimensional problem space using (4) as a cost function. For each iteration of the algorithm we iterate through all the patches and perform an iteration of the cross-entropy method for the patch, while fixing the positions of all other patches. The algorithm outline

Fig. 5. Local matching of patch set using the cross-entropy method.

- **Input:** The set of patches $\mathbf{x}_t^{(i)} = \mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)}$
- **Initialization:** Set the initial mean vectors and covariance matrices $\mu_0^i = \mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)}$ and $\mathbf{\Sigma}_0^i = \mathbf{\Sigma}_L$. Compute the neighborhood of each patch using Delaunay triangulation.
- For $j = 1 \dots M_L$:
  - For each patch $\mathbf{x}_t^{(i)}$:
    - Sample S_L samples from $\mathcal{N}(\mu_{j-1}^i, \mathbf{\Sigma}_{j-1}^i)$
    - For each sample $\Delta_t^{(i)}$ evaluate (4) for $\mathbf{x}_t^{(i)} = \mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)} + \Delta_t^{(i)}$.
    - Select E_L best samples according to (4) and use them to recalculate μ_j^i and $\mathbf{\Sigma}_j^i$
- **Output:** The optimized set of points $\mathbf{X}_t = \{\mathbf{x}_t^{(i)} = \mathbf{A}_t^{\mathrm{G}}\hat{\mathbf{x}}_t^{(i)} + \mu_j^i\}_{i=1\dots N_t}$

is in Figure 5.

2.2 Updating the local layer

The local patches are very small and are compared only using their grayscale histograms. This simple appearance representation is robust to some deformations, e.g. rotation. It provides good short-term tracking support, especially when using more such patches together, however, it is not sufficient for more than a short period of time, usually a few frames. In the long-run some patches become outdated. Because in most cases not all of the parts of the objects change at the same time, patches can be replaced sequentially, making a gradual transition to a new patch set.

Recall from (1) that there is a weight $w_t^{(i)}$ associated with each patch and it reflects the belief 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 modified as

$$w_t^i = \lambda_W w_{t-1}^i + (1 - \lambda_W)\hat{w}_t^i, \tag{10}$$

where λ_W is defined as a persistence constant and $\hat{w}_t^i$ is defined as the estimated weight in the current time step. The weight $\hat{w}_t^i$ is equal to

$$\hat{w}_t^i = p(\mathbf{Y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{X}_t). \tag{11}$$

In (11) the first part of the product is *visual consistency* and is based directly on the visual likelihood, defined in (5). The second one is *drift from majority* and is defined by a sigmoid function:

$$p(\mathbf{x}_t^{(i)}|\mathbf{X}_t) = \frac{1}{1 + e^{\lambda_D(mdst(\mathbf{x}_t^{(i)}, \mathbf{X}_t) - T_D)}}, \tag{12}$$

where $mdst(\mathbf{x}_t^{(i)}, \mathbf{X}_t)$ stands for the median of Euclidean distances between the patch position and position of every other patch in the set. The T_D and λ_D are constants that determine the size of the object and the influence of the consistency constraint respectively.

Patches with low weight (lower than a constant threshold T_{R}) are considered as either outdated or mispositioned and are removed from the set. To maintain numerical stability [1] and avoid unnecessary computations, we also merge patches that are too close to each other. A replacement patch is initialized at the weighted average of the positions of merged patches and is given their average weight.

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 which 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., $w_0 = 2T_{\mathrm{R}}$. 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 vary with the target's size, we always try to allocate at most $N_t^{\mathrm{all}} \leq N_t^{\mathrm{cap}} - \tilde{N}_t + 1$ new patches. To prevent sudden significant changes in the estimated capacity, we adapt it using the autoregressive scheme:

$$N_{t+1}^{\mathrm{cap}} = \alpha_{\mathrm{cap}} N_t^{\mathrm{cap}} + (1 - \alpha_{\mathrm{cap}})N_t, \tag{13}$$

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

2.3 The global layer

The global layer in our coupled-layer visual model is denoted as $\mathcal{G}_t$. It captures different aspects of the target's global appearance. In this work we used the following three visual properties: color C_t, apparent motion M_t and shape S_t,

$$\mathcal{G}_t = \{C_t, M_t, S_t\}. \tag{14}$$

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

$$p(\mathbf{x}|C_t, M_t, S_t) \propto p(C_t, M_t, S_t|\mathbf{x}). \tag{15}$$

Assuming that the visual cues are independent given a position $\mathbf{x}$, then (15) factors as

$$p(\mathbf{x}|C_t, M_t, S_t) \propto p(C_t|\mathbf{x})p(M_t|\mathbf{x})p(S_t|\mathbf{x}). \tag{16}$$

In the following we describe the models for each of the cues.

Color

1. Delaunay triangulation works better if the input points are not too close to each other.

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 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

location $\mathbf{x}$ belongs to a target is therefore

$$p(C_t|\mathbf{x}) = \frac{p(\mathbf{x}|\mathrm{F})p(\mathrm{F})}{p(\mathrm{F})p(\mathbf{x}|\mathrm{F}) + (1-p(\mathrm{F}))p(\mathbf{x}|\mathrm{B})}. \quad (17)$$

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

$$\begin{aligned} \mathbf{h}_{t+1}^F &= \alpha_F \mathbf{h}_t^F + (1-\alpha_F)\hat{\mathbf{h}}_t^F \\ \mathbf{h}_{t+1}^B &= \alpha_B \mathbf{h}_t^B + (1-\alpha_B)\hat{\mathbf{h}}_t^B, \end{aligned} \quad (18)$$

where α_F and α_B determine the rate of adaptation.

Motion

The apparent motion model is defined by the local-motion model from [2]. Briefly, the local motion model [2] 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 [26]) with the global velocity $\mathbf{v}_t$ estimated by the tracker. In our implementation we apply Harris corner detection [27] to determine the salient points. As in [2], the motion likelihood at salient point $\mathbf{x}_i$ is defined as

$$p(\mathbf{x}_i|M_t) \propto (1-w_\mathrm{noise})e^{-\lambda_\mathrm{M}(d(\mathbf{v}(\mathbf{x}_i),\mathbf{v}_t))} + w_\mathrm{noise}, \quad (19)$$

where $d(\mathbf{v}(\mathbf{x}_i), \mathbf{v}_t))$ is the distance between two velocities as defined in [2] and w_noise is uniform noise. Finally, to obtain a dense estimation, the set of salient points is convolved with a smoothing kernel. We therefore define the motion likelihood as

$$p(M_t|\mathbf{x}) \propto \frac{1}{K}\sum_{i=1}^{N_\mathrm{s}} p(\mathbf{x}_i|M_t)\Phi_{\mathbf{\Sigma}}(\mathbf{x}-\mathbf{x}_i), \quad (20)$$

where $\Phi_{\mathbf{\Sigma}}(\mathbf{x})$ 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 [28].

Shape

The shape model is an estimate of the object's shape. 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 the hull by a constant amount of D_S pixels. We define a function $s(\mathbf{x}, S_t) \equiv 1$ if $\mathbf{x} \in S_t$ and 0 otherwise and the shape likelihood model for a pixel at $\mathbf{x}$ is thus defined as

$$p(S_t|\mathbf{x}) \propto s(\mathbf{x}, S_t). \quad (21)$$

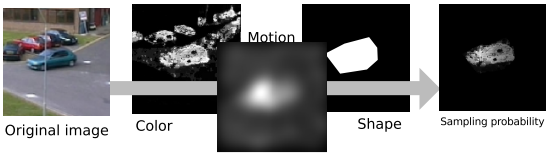

Fig. 6. Illustration of the cumulative probability map construction.

As mentioned before, (16) is used for allocating new patches in the local layer. We do not sample (16) 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. The process of probability map construction is illustrated on a real example in Figure 6. To make sure that 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 patch's 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 [29] with a nearly-constant velocity (NCV) dynamic model [30] 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 patches from the previous frame:

$$\hat{\mathbf{x}}_t = \mathbf{x}_{t-1}^{(i)} + \hat{\mathbf{v}}_t^{(i)}. \quad (22)$$

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 Figure 7.

Fig. 7. The integration of a coupled-layer visual model in a tracker.

- **Input:** Place a region over a target.
- **Initialization:** Distribute patches in a regular grid in the region and assign uniform weights.
- For $t = 1, 2, 3 \ldots$
  - Predict the target's velocity $\hat{\mathbf{v}}_t$ using the Kalman filter and initialize the local-layer patches with the NCV model (22).
  - Adapt the local layer patches by maximizing $p(\mathbf{Y}_t, \mathbf{X}_t | \hat{\mathbf{X}}_t)$ as described in algorithms in Figure 4 and Figure 5.
  - Recalculate the target's center $\mathbf{c}_t$ and update the Kalman filter estimate.
  - Identify/remove irrelevant patches from the local layer (Section 2.2).
  - Using the remaining patches, update the visual cues of the global layer (Section 2.3).
  - 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.

TABLE 1
List of the parameters, used in the experiments for our method.

Section	Parameters
patches	size: 6×6 pixels, bins: 16
optimization	$\lambda_{\mathrm{v}} = 0.1$, $\lambda_{\mathrm{g}} = 0.015$ $M_G = 10$, $S_G = 300$, $E_G = 10$, $\mathbf{\Sigma}_0^G = 20I$ $M_L = 5$, $S_L = 50$, $E_L = 5$, $\mathbf{\Sigma}_0^L = 5I$
updating	$\lambda_D = 3$, $T_D = 40$, $T_R = 0.1$, $\alpha_{\mathrm{cap}} = 0.8$
modalities	bins (HSV): $16 \times 16 \times 4$, $\alpha_F = 0.95$, $\alpha_B = 0.5$ $w_{noise} = 0.01$, $\lambda_M = 1$, $D_S = 10$
Kalman filter	$\mathbf{R} = I$, $\mathbf{Q} = I$

3 Experimental Results

We have analyzed the performance of our tracker on several challenging video sequences. The sequences include either a nonrigid object or an object that undergoes significant appearance changes. Our tracker was implemented in Matlab/C and runs at approximately 4 frames per second on an Intel Core 2 Duo 6600. The parameters of our tracker were set as shown in Table 1. We would like to emphasize that *all the parameters were kept constant* for all experiments. For brevity, we will from now on refer to our tracker as the LGT [2] (as in Local-Global Tracker).

We have compared the LGT with eleven related state-of-the-art reference trackers, which are related to our tracker in a way that they all address the problem of object appearance changes:

1) A color-based particle filter (PF) [3].
2) The online boosting tracker (OBT) [7].
3) The beyond online boosting tracker (BOBT) [9].
4) The flock-of-features tracker (FOF) [21].
5) The piecewise-affine kernel tracker (PAKT) [13].
6) The basin-hopping Monte Carlo tracker (BHMC) [16].
7) The incremental visual tracker (IVT) [5].
8) The histograms-of-blocks tracker (BH) [18].
9) The multiple instance tracker (MIL) [6].
10) The fragment tracker (FRT) [31].
11) The P-N tracker (TLD) [8].

The experiments involved tracking a hand, a human body and objects with challenging view changes

Fig. 8. Samples from the experimental video sequences.

TABLE 2
An overview of the video sequences.

Sequence	Type	Comments	Frames
hand [22]	articulated body part	rapid motion	242
hand2 [22]	articulated body part	rapid motion	267
gymnastics	articulated	rapid motion	206
diver [16]	articulated	rotation	214
bicycle	articulated	occlusion, scale change	271
dinosaur [22]	rigid	elaborated structure	324
torus [22]	rigid	empty center	262
can	rigid	rolling, blurring	212
david_indoor [5]	face	illumination changes, scale change, appearance change	770
trellis [5]	face	illumination changes	569
car [32]	car	blur, distortion, rapid motion	267
david_outdoor [32]	body	occlusion	186

(Figure 8). The basic properties of the experimental sequences are collected in Table 2[3]. The target was tracked in each sequence $R = 30$ times by each tracker, except for the PAKT tracker, where only $R = 10$ tries were done for each sequence due to time consuming manual initialization.

For comparison, we recorded the number of times each tracker failed and had to be reinitialized. We also recorded the tracked trajectories. The tracking

2. To enable the comparison with our tracker, we have made our code publicly available at http://go.vicos.si/lgt.

3. The annotated sequences are available at http://go.vicos.si/lgt.

Fig. 9. Results for the *hand* sequence. Results are shown for trackers *FOF* (first row), *PF* (second row) and *LGT* (last row). The best run is used for all trackers. On the right side the top graph shows the center error for the trackers and the bottom graph shows the likelihood of failure.

Fig. 10. Results for the *bicycle* sequence. Results are shown for trackers *BOBT* (first row), *TLD* (second row) and *LGT* (last row). The best run is used for all trackers. On the right side the top graph shows the center error for the trackers and the bottom graph shows the likelihood of failure.

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

$$F(\Omega^t_{\text{gt}}, \Omega^t) = \frac{\Omega^t_{\text{gt}} \cap \Omega^t}{\Omega^t_{\text{gt}} \cup \Omega^t}. \tag{23}$$

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's accuracy with respect to the other trackers, we computed a root-mean-squared-error (RMS) between the target's trajectory estimated by a tracker and the target's ground truth trajectory.

Note that some methodologies for tracker comparison (e.g. [16], [5], [6]) disregard failures and measure the RMS error throughout the entire video. In that case the tracker might drift to the background after the beginning of the sequence and just by accident drift back to the target at some later point in the video. The RMS decreases in this case, but we are no longer measuring the trackers true expected accuracy as the accidental drift-back frames cannot be fairly used without reinitialization. We believe that apart from the tracking accuracy, it is also very important to measure the trackers ability to follow the target without drifting away.

When comparing the results we have also performed a one-sided standard hypothesis test [33] to ensure that the differences in performance are statistically significant. Let E^r_1 and E^r_2 be the number of failures for the supposedly better fist tracker and the second tracker tracker, respectively, at the r-th tracking repetition. To test the hypothesis that the first tracker really performs better than the second tracker on a specific sequence, we have to compute the mean difference μ_T and the standard deviation σ^2_T

$$\mu_T = \frac{1}{R} \sum\nolimits_{r=1}^{R} E^r_1 - E^r_2 \tag{24}$$

$$\sigma^2_T = \frac{1}{R^2} \sum_{r=1}^{R} (E^r_1 - E^r_2 - \mu_T)^2. \tag{25}$$

A null hypothesis, that the first tracker does not perform better that the second tracker, can be rejected at significance level α if $\frac{\mu_t}{\sigma_t} > L_\alpha$. In our experiments we have used a common significance level $\alpha = 0.05$ ($L_\alpha = 1.564$).

The results for the comparison experiment are summarized in Table 3. Besides that we also visualize selected tracking results in Figure 9 to Figure 13. Due to space limitations only two reference trackers are compared in each figure. For each sequence we selected two trackers that are related to the problematic aspect of the sequence.

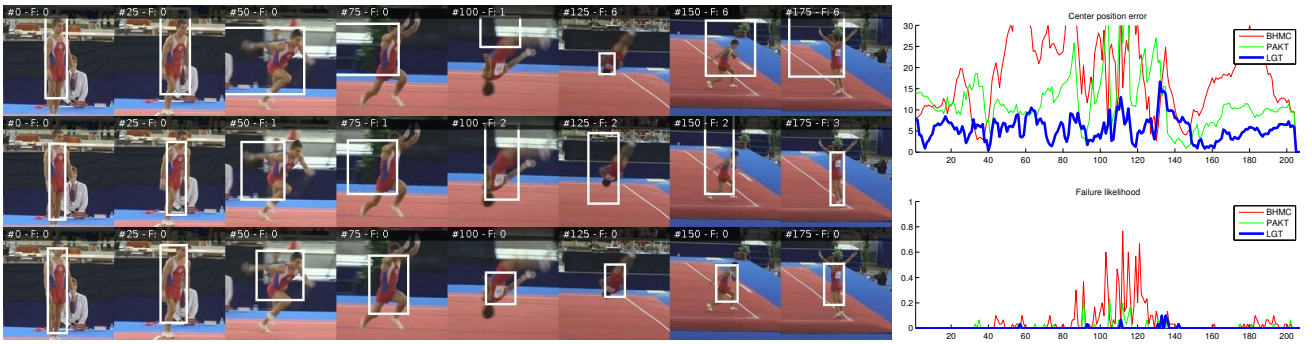

Fig. 11. Results for the *gymnastics* sequence. Results are shown for trackers *BHMC* (first row), *PAKT* (second row) and *LGT* (last row). The best run is used for all trackers. On the right side the top graph shows the center error for the trackers and the bottom graph shows the likelihood of failure.

3.1 Failure rate

The first number for each tracker-sequence combination in Table 3 shows the average failure rates for each tracker. We see that the LGT achieves the lowest failure rate for eight of the sequences while being close to the top for the remaining four. Looking at the number of failures per sequence, we also see that the sequences *hand2*, *bicycle*, and *david_outdoor* were the most difficult to track, not just for the proposed tracker, but also for many of the reference trackers.

In the *hand2* sequence, 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 9, 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 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 *bicycle* sequence is another notably difficult sequence, mainly because of the complete occlusion of the target that occurs in the second half of the sequence and because of the size change that can be seen because the target first moves closer to the camera and then back away from it. Because of the occlusion every evaluated tracker had to be reinitialized at least once using our failure criteria. The IVT tracker performs best in this case, the LGT comes third (Table 3). Frames from the sequence can be seen in Figure 10 as a comparison of BOBT, TLD, and LGT. The average of 2.43 failures per trial for the LGT can be mainly attributed to the full occlusion of the target which occurs around frame 170 and causes problems to the reference trackers as well. Likewise, the *david_outdoor* sequence also contains several occlusions.

We would like to point out that occlusion handing was not the focus of this work, however, we do further analyze its effects in subsection 3.4. The other notable problem was the small size of the object towards the end of the sequence. In our current visual model very small objects are difficult to describe using a set of patches if the size of a patch is almost the same as the size of the object. The problem of the small size of the object can also be noticed for some reference trackers, e.g. the BHMC tracker, which consistently failed to initialize properly due to the small target size at the end of the *bicycle* sequence, resulting in a high number of failures.

Six of the sequences (*gymnastics*, *diver*, *bicycle*, *david_indoor*, *trellis*, and *car*) include camera motion. In cases of *gymnastics* and *diver* the camera is following the object very strictly while in the other three cases the camera moves more freely. It is therefore worth noting that in *gymnastics* and *diver* the objects do not move much spatially, but rather significantly modify 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 11 compares the BHMC tracker (first row), PATK tracker (second row) and LGT (last 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 3). On the other hand, the LGT successfully tracks through the scale change. The tracking results for the all 30 tries, summarized in center error and failure graphs at the right side of Figure 11 show that the LGT only fails in some tries at the second full turn of the body and that it follows the center of the object better that the other two trackers.

The advantages of the LGT-based tracker are also

TABLE 3
The number of failures and RMS errors with respect to the ground truth. The bold text marks the best tracker for each sequence. The $\cdot^*$ denotes that the improved performance was statistically significant.

	PAKT [13]	FOF [21]	PF [3]	BHMC [16]	OBT [7]	IVT [5]	FRT [31]	BH [18]	BOBT [9]	MIL [6]	TLD [8]	LGT
hand [22]	23.50 (17.81)	11.03 (18.56)	4.07 (14.17)	31.57 (26.89)	11.00 (17.75)	12.00 (18.11)	8.00 (19.93)	20.00 (26.00)	31.87 (11.28)	6.33 (16.73)	19.40 (53.14)	**0.57*** (9.41)
hand2 [22]	43.00 (18.45)	13.47 (17.64)	9.83 (16.36)	47.43 (26.01)	31.00 (22.37)	18.00 (18.08)	17.00 (19.56)	30.00 (48.07)	48.07 (14.89)	12.83 (16.59)	26.03 (60.10)	**1.23*** (12.12)
gymnastics [22]	5.80 (18.20)	4.33 (23.31)	5.10 (21.45)	10.50 (26.93)	5.00 (18.93)	7.00 (20.11)	6.00 (14.05)	7.00 (17.01)	34.97 (10.39)	6.00 (14.69)	21.30 (57.41)	**0.47*** (9.84)
diver [16]	3.90 (17.53)	2.07 (13.72)	5.33 (16.09)	3.87 (20.96)	7.00 (17.52)	9.00 (19.90)	9.00 (18.48)	17.00 (26.36)	35.57 (13.79)	5.27 (16.49)	7.53 (20.97)	**0.50*** (12.57)
bicycle	9.10 (10.04)	5.27 (10.18)	8.87 (12.30)	74.80 (20.10)	4.00 (6.60)	**1.00*** (5.75)	4.00 (7.15)	18.00 (19.78)	9.60 (4.06)	1.27 (4.75)	2.77 (16.69)	2.43 (10.93)
dinosaur [22]	8.00 (24.86)	2.27 (17.11)	2.23 (21.82)	16.37 (34.41)	5.00 (26.43)	8.00 (31.28)	5.00 (23.81)	13.00 (36.45)	29.00 (18.80)	4.20 (29.48)	5.60 (27.70)	**0.00*** (10.84)
torus [22]	11.30 (14.29)	5.63 (14.04)	2.30 (16.46)	25.07 (21.04)	14.00 (14.28)	2.00 (10.26)	3.00 (13.51)	29.00 (39.59)	27.10 (6.38)	3.53 (11.20)	3.20 (14.12)	**0.00*** (7.54)
can	9.70 (16.12)	**0.00** (10.20)	0.03 (9.66)	11.73 (22.81)	**0.00** (9.02)	1.00 (6.93)	2.00 (19.68)	7.00 (27.89)	10.13 (6.69)	**0.00** (11.24)	3.63 (28.24)	0.03 (19.15)
david_indoor [5]	6.70 (21.95)	2.00 (15.56)	2.23 (21.71)	23.93 (30.39)	2.00 (17.24)	**0.00*** (14.47)	6.00 (25.42)	30.00 (37.52)	14.80 (8.50)	1.87 (24.07)	0.40 (10.38)	0.67 (15.65)
trellis [5]	12.40 (16.80)	2.90 (14.34)	1.13 (18.92)	16.70 (24.19)	9.00 (17.42)	**0.00** (22.97)	11.00 (19.38)	69.00 (50.76)	41.70 (7.09)	4.97 (22.53)	24.07 (53.11)	**0.00** (14.79)
car [32]	6.20 (9.48)	3.87 (10.90)	7.63 (15.96)	23.03 (20.04)	1.00 (5.25)	3.00 (4.05)	3.00 (9.11)	6.00 (12.06)	5.67 (3.52)	0.60 (7.49)	2.43 (19.81)	**0.53** (11.66)
david_outdoor [32]	5.20 (12.34)	2.90 (14.03)	2.23 (14.78)	5.00 (19.24)	3.00 (6.98)	2.00 (4.52)	4.00 (9.06)	12.00 (22.64)	13.00 (4.50)	**0.00*** (8.82)	5.03 (16.37)	1.70 (13.37)

seen 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 hand, the changes in the appearance are still hard to describe without a predefined geometrical model for a specific object. As seen in the Figure 12, when tracking a torus, the OBT and FRT reference trackers drift from the object several times during the sequence, while the LGT successfully accomplishes the task. The OBT tracker (first 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 FRT tracker (second row) does partition the appearance, but the partitioning does not change and does not take into the account the difficult structure of the object. The LGT (last row) does not have these problems and can successfully track the object throughout the sequence.

In case of the *can* sequence the main challenge is the fast out-of-plane rotation of the object. Because of fast changes in appearance, this presents problems for trackers that maintain a set of local parts. To compensate for these changes parts have to be removed and new parts have to be added at a fast rate. Because no part exists for more than a few frames this increases the chance of failure. On the other hand, trackers that rely only on a high-level appearance abstraction, like color, e.g. the PR tracker, handle this specific type of scenario easily because the color of the object can be easily separated from the background and the size of the object does not change. Same reasoning can be applied to grayscale images in case of the OBT tracker.

The main challenge of the *david_indoor* and *trellis* sequences is are the illumination changes that occur as the person moves around. Illumination changes influence the appearance of the object in a way that cannot be described using geometric transformations. For a tracker that uses local parts for tracking this means that it should replace several of the parts instead of trying to adapt the geometrical structure. Figure 13 shows selected frames of the *trellis* sequence for the IVT, MIL, and LGT. We can see that a good strategy for feature replacement is needed for the tracker to stay on the face of the person. The face does not change a lot geometrically, however parts of the object are covered by the shadows of the trees above the scene. The LGT can handle these changes as it replaces the local parts using a multi-modal global visual model of the object. While the other two trackers also track the object quite well, we can see from the center error vs. frame graph that the LGT does have some problems with the estimation of the object size in the beginning of the sequence, but improves in time.

3.2 Tracking accuracy

The second number for each tracker-sequence combination in Table 3 shows the tracking accuracy in terms of the average RMSE. From comparison of the RMSEs of reference trackers and the proposed tracker we can conclude that the proposed tracker, LGT, in most cases outperforms the reference trackers in accuracy. The LGT outperforms all the reference trackers for five sequences and produces comparable accuracy to the best tracker for another two sequences, *david_indoor* and *trellis*. We have to point out here

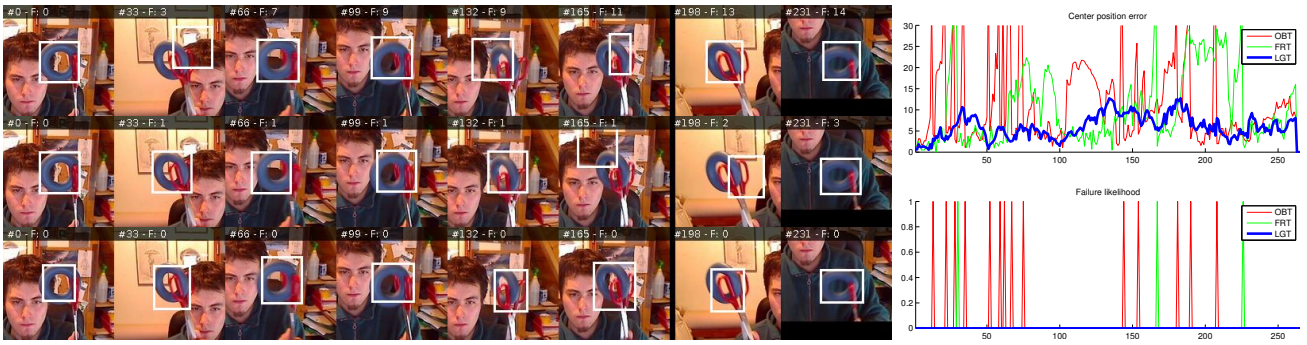

Fig. 12. Results for the *torus* sequence. Results are shown for trackers *OBT* (first row), *FTR* (second row) and *LGT* (last row). The best run is used for all trackers. On the right side the top graph shows the center error for the trackers and the bottom graph shows the likelihood of failure.

that the tracker that achieves the lowest RMS error in seven sequences, the BOBT tracker, does this well mainly because of the large amount of manual initializations (see Table 3) that reset the position to the correct one according to the groundtruth data. This tracker is not really well suited for the context of our experiments because it is designed to report when it does not see the object, which it does quite often and we count that as a failure as we require a constant tracking trajectory. From this example we can also see that the two results (failure rate and RMS error) have to be observed together as a tracker that automatically fails in every frame would achieve 0 RMS error in our experimental setup, however the failure rate would be very high.

For the *bicycle* and *can* sequences the accuracy of the LGT is notably worse than the accuracy of several reference trackers. The main causes for the lower accuracy in the *bicycle* sequence were the full occlusion and the small size of the object in the image towards the end of the sequence as already discussed in Section 3.1. For the *can* sequence the main reason for the lower accuracy is the lack of a strong shape model. Because the estimated region of the object, that is reported by the LGT is dependent on the positions of the patches, it can change very rapidly when a lot of patches are removed or added to the set at once. To stabilize the estimated region of the object a stronger shape constraint should be introduced into the visual model. This would, however, reduce the flexibility of the tracker to adapt to the shape change in other scenarios.

3.3 Tracker analysis

We have analyzed how the parameters affect the trackers performance on three sequences: *dinosaur*, *gymnastics* and *hand2*. For this analysis we have selected six parameters that control the behavior of the proposed visual model. Parameter T_D controls the second weight updating rule, described in Section 2.1. Parameter T_R determines which patches should be removed from the set. Parameter λ_{g}, defined in (6), controls the deformation of the constellation of parts. Parameter D_S, described in Section 2.3, controls the dilation of the shape modality shape estimation. Parameters α_B, and α_F control the adaptation persistence of the color histograms in color modality. The results of the experiment are summarized in Figure 14.

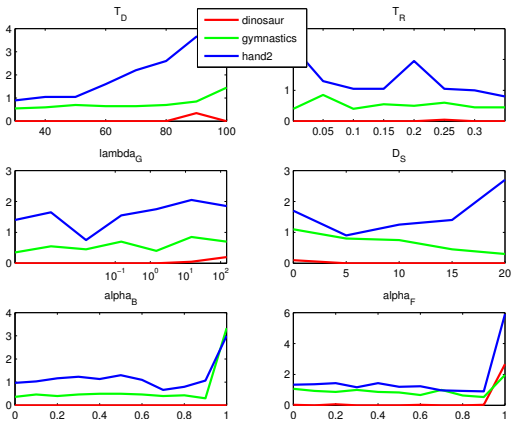

Fig. 14. Results for parameter value analysis for parameters T_D, T_R, λ_{g}, D_S, α_B, and α_F on sequences *dinosaur*, *gymnastics* and *hand2*. The y-axis shows the average number of failures. The x-axis shows the values of the evaluated interval.

We have tested the T_D parameter in the interval between 30 and 100. From the graphs we can observe that if the value of T_D is higher than 50 the weights of the parts that move from the majority (because they do not belong to the object) do not get decreased and the tracker fails more frequently. The parameter is related to the estimated size of the object in the image, however, it is important to note that a wider range of values produces similar results (values 30 to 50), therefore the value does not have to be set for each sequence separately. Parameter T_R was tested on the interval 0.001 to 0.35. The performance decreases in the bottom 20% of the range, because in this case the outdated patches would not get removed in time and is stable in the remaining part of the interval. Parameter λ_{g} was tested in the interval 0.00014 to 148 and the results are shown with a logarithmic scale. When increasing the value, the performance

Fig. 13. Results for the *trellis* sequence. Results are shown for trackers *MIL* (first row), *IVT* (second row) and *LGT* (last row). The best run is used for all trackers. On the right side the top graph shows the center error for the trackers and the bottom graph shows the likelihood of failure.

gradually drops. High values of this parameter result in rigid geometrical constraints between parts. This tells us that deformation of the patch constellation is required to robustly track deformable objects. The parameter D_S was tested on interval 0 to 20. We can see two opposing trends on the graph. In all three sequences value 0 decreases the performance because the dilation influences the ability of the patch set to grow in size. The increase of the value first improves the performance (first 20% of the interval), however, the performance then drops for the *hand2* sequence. We explain this by the fact that in that sequence the color modality, that usually constraints the object area, is not as reliable because of other similarly-colored areas in the scene (e.g. face). The parameters α_B and α_F were tested on their entire valid range. The value of these parameters does not influence the performance significantly unless their value is set to the top 10% of the interval, which means that we do not update the color modality at all. The experiments show that the proposed tracker is not sensitive to the parameter values even for different types of objects as long as they stay within reasonable intervals.

To gain a better insight into how different existing modalities affect the tracking, we have performed several additional experiments. We have analyzed the contribution of each modality separately by running the tracker without that modality on tracking sequences and compared the performance to the tracker that used all modalities. The results are summarized in Table 4.

The results show the tracking performance for the three selected sequences. We compare the original tracker configuration to the one with either color, motion or shape removed. From the results we can see that color plays most important role since the drop in performance is the largest in this modality is left out. From the results for the hand sequence we can see that the removal of either motion or shape if these cases even slightly improves the trackers performance. On the other hand, these modalities improve tracking in the bicycle sequence. This suggests that not all modalities are equally important in every situation

TABLE 4
Influence of individual modalities in the global layer. We report the avg. number of failures per experiment.

Sequence	Original	No color	No motion	No shape
bicycle	2.43	3.87	4.13	3.83
hand [22]	0.57	5.57	0.10	0.23
torus [22]	0.00	1.83	0.00	0.00

during the tracking. An application of a technique for automatic modality selection [4], [34] would be an interesting venue of research to further strengthen the tracker.

3.4 Occlusion

Despite the fact that the LGT was not designed with special attention to object occlusions we have performed an additional experiment that focuses exactly on that. We use two sequences presented from [31] that we will call *face* and *woman*. The results of the experiment are presented in Figure 15.

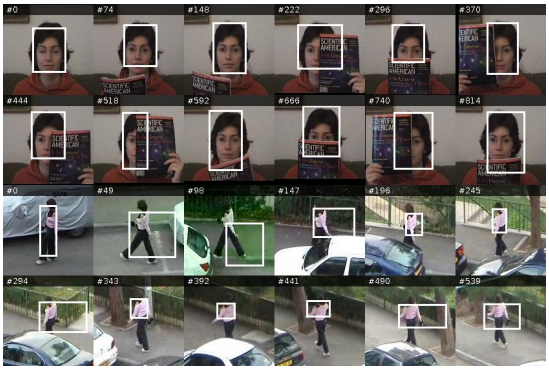

Fig. 15. Results for *face* and *woman* sequences.

We can see that the LGT successfully tracks the face in the first sequence despite the numerous partial occlusions. There are, however, problems with the estimation of the size of the object due to removal of patches on occluded regions in the second sequence where the tracker recovers very slowly after half of the woman is occluded by a car because the global layer focuses only on the visible part of the object and gradually forgets about the occluded one. In 30 trials the LGT did not have to be manually repositioned

for the *face* sequence, however the average of 1.9 manual re-initializations were required for the *woman* sequence using the same methodology as in the previous experiments. This shows us that our tracker is robust even to some types of occlusions. Other types of occlusions, where the occluded part of the object is visually different from the visible one, still need additional constraints in the model. This is a part of our ongoing research.

4 Discussion and conclusion

We have proposed a coupled-layer visual model for efficient tracking of targets that undergo significant appearance changes. The model is a 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 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 implemented the proposed visual model in a tracker and compared it to the state-of-the-art on several challenging sequences. Results show that our tracker on average outperforms the related trackers by smaller failure rate and at a greater accuracy. Experiments have shown that even in 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 focusing on 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 experiments. In the same respect, the tracker is expected to handle some types of 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, because we do not impose a strong prior model on targets appearance and shape [17] 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. As seen in Section 3, the current update mechanism also does not work well for complete occlusions of the object where the entire set of parts has to be replaced in a short interval. In these cases a more restrictive update mechanism will be required, which is a topic of our ongoing research.

One interesting aspect of the proposed local layer is that it can robustly track the target even though the local patches are modeled by simple gray-scale histograms. But when combined with geometrical constraint (that also adapts) and the constrained allocation/removal of the patches, the resulting visual model becomes powerful. In future work, we will analyze the tracker's performance if more complex descriptors (e.g. random forest [19], patch covariance [35]) 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 and with the type of the object (as suggested by the experiment in subsection 3.3), our visual model would benefit from a mechanisms for on-line selection of salient cues [4], [34] in the global layer. While the constrained adaptation of the model can address particular cases of occlusion (subsection 3.4), we have not explored any mechanisms for explicitly handling complete or long-lasting occlusions. In the future research we will extend our framework to address these issues as well.

Acknowledgments

The authors would like to thank the authors of [7], [13], [16], [5], [31], [9], [6], [8] for providing the reference implementations of their trackers. This research was in part supported by: ARRS projects J2-3607, J2-2221 and J2-4284 and EU project CogX (FP7-ICT215181-IP).

References

[1] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," in *ICPR*, vol. 1, 2004, pp. 790–797.

[2] M. Kristan, J. Perš, S. Kovačič, and A. Leonardis, "A local-motion-based probabilistic model for visual tracking," *Pattern Recognition*, 2008.

[3] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-Based probabilistic tracking," in *ECCV*, vol. 1. Springer-Verlag, 2002, pp. 661–675.

[4] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *TPAMI*, vol. 27, no. 10, pp. 1631–1643, 2005.

[5] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, p. 125–141, May 2008, ACM ID: 1346002.

[6] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *TPAMI*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[7] H. Grabner, M. Grabner, and H. Bischof, "Real-Time tracking via on-line boosting," in *BMVC*, 2006.

[8] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 49–56.

[9] S. Stalder, H. Grabner, and L. van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *ICCVW*. IEEE, Oct. 2009, pp. 1409–1416.

[10] B. Stenger, T. Woodley, and R. Cipolla, "Learning to track with multiple observers," in *CVPR*. IEEE, Jun. 2009, pp. 2647–2654.

[11] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR*. IEEE, Jun. 2010, pp. 1269–1276.

[12] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel, "Probabilistic color and adaptive Multi-Feature tracking with dynamically switched priority between cues," in *ICCV*, 2007, pp. 1–8.

[13] B. Martinez and X. Binefa, "Piecewise affine kernel tracking for non-planar targets," *Pattern Recognition*, vol. 41, no. 12, pp. 3682–3691, 2008.

[14] V. Badrinarayanan, F. Le Clerc, L. Oisel, and P. Perez, "Geometric layout based graphical model for Multi-Part object tracking," in *International Workshop on Visual Surveillance*, 2008.

[15] W. Chang, C. Chen, and Y. Hung, "Tracking by parts: A bayesian approach with component collaboration," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 375–388, 2009.

[16] 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 *CVPR*, 2009, pp. 1208–1215.

[17] Z. Yin and R. Collins, "On-the-fly object modeling while tracking," in *CVPR*, 2007, pp. 1–8.

[18] S. S. Nejhum, J. Ho, and M. Yang, "Online visual tracking with histograms and articulating blocks," *CVIU*, vol. 114, no. 8, pp. 901–914, Aug. 2010.

[19] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *ICCV*, Barcelona, Nov. 2011.

[20] J. Hoey, "Tracking using flocks of features, with application to assisted handwashing," in *BMVC*, 2006.

[21] M. Kölsch and M. Turk, "Fast 2D hand tracking with flocks of features and Multi-Cue integration," in *CVPRW*, vol. 10. Washington, DC, USA: IEEE Computer Society, 2004, p. 158.

[22] L. Čehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," in *ICCV*, Barcelona, 2011.

[23] Z. Fan, M. Yang, and Y. Wu, "Multiple collaborative kernel tracking," *TPAMI*, vol. 29, no. 7, pp. 1268–1273, 2007.

[24] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, May 1997.

[25] A. Blake and A. Zisserman, *Visual reconstruction*. MIT Press, 1987.

[26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (IJCAI)," in *International Joint Conference on Artificial Intelligence*, Apr. 1981, pp. 674–679.

[27] C. Harris and M. Stephens, "A combined corner and edge detection," in *Alvey Vision Conference*, 1988, pp. 151, 147.

[28] M. Kristan, A. Leonardis, and D. Skočaj, "Multivariate online kernel density estimation with gaussian kernels," *Pattern Recogn.*, vol. 44, no. 10-11, p. 2630–2642, Oct. 2011, ACM ID: 1998844.

[29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering*, vol. 82, pp. 34–45, 1960.

[30] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333– 1364, Oct. 2003.

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

[32] Q. Wang, F. Chen, W. Xu, and M. Yang, "An experimental comparison of online object-tracking algorithms," in *SPIE: Image and Signal Processing*, San Diego, 2011, pp. 81 381A–81 381A–11.

[33] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, 1st ed. Wiley-Interscience, Jun. 2001.

[34] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *ICCV*. IEEE, 2007, pp. 1–7.

[35] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on means on riemannian manifolds," Mitsubishi Electric Research Laboratories, Tech. Rep., 2006.

Luka Čehovin received his Dipl.ing. and M.Sc. degree at the Faculty of Computer Science and Informatics, University of Ljubljana, Slovenia in 2007 and 2010, respectively. Currently he is working at the Visual Cognitive Systems Laboratory, Faculty of Computer Science and Informatics, University of Ljubljana, Slovenia as an assistant and a Researcher. His research interests include computer vision, HCI, distributed intelligence and web-mobile technologies.

Matej Kristan received a Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana in 2008. He is currently an Assistant Professor at the Faculty of Electrical Engineering and the Faculty of Computer and Information Science, University of Ljubljana. His research interests include probabilistic methods for computer vision and pattern recognition with focus on tracking, probabilistic dynamic models, online learning and mobile robotics.

Aleš Leonardis is a full professor and the head of the Visual Cognitive Systems Laboratory at the Faculty of Computer and Information Science, University of Ljubljana, and a Chair of Robotics at University of Birmingham. He is also an adjunct professor at the Faculty of Computer Science, Graz University of Technology. His research interests include robust and adaptive methods for computer vision, object and scene recognition and categorization, statistical visual learning, 3D object modeling, and biologically motivated vision. He has been an associate editor of the IEEE PAMI, an editorial board member of Pattern Recognition, an editor of the Springer book series Computational Imaging and Vision. He was also a Program Co-chair of the European Conference on Computer Vision 2006. In 2002, he coauthored a paper, Multiple Eigenspaces, which won the 29th Annual Pattern Recognition Society award. He is a fellow of the IAPR and a member of the IEEE and the IEEE Computer Society.