6

Switching Local and Covariance Matching for Efficient Object Tracking

Junqiu Wang and Yasushi Yagi
Osaka University
Japan

1. Introduction

Object tracking in video sequences is challenging under uncontrolled conditions. Tracking algorithms have to estimate the states of the targets when variations of background and foreground exist, occlusions happen, or appearance contrast becomes low. Trackers need to be efficient and can track variant targets. Target representation, similarity measure and localization strategy are essential components of most trackers. The selection of components leads to different tracking performance.

The mean-shift algorithm Comaniciu et al. (2003) is a non-parametric density gradient estimator which finds local maxima of a similarity measure between the color histograms (or kernel density estimations) of the model and the candidates in the image. The mean-shift algorithm is very fast due to its searching strategy. However, it is prone to failure in detecting the target when the motion of the target is large or when occlusions exist since only local searching is carried out.

The covariance tracker Porikli et al. (2006) represents targets using covariance matrices. The covariance matrices fuse multiple features in a natural way. They capture both spatial and statistical properties of objects using a low dimensional representation. To localize targets, the covariance tracker searches all the regions; and the region with the highest similarity to the target model is taken as the estimation result. The covariance tracker does not make any assumption on the motion. It can compare any regions without being restricted to a constant window size. Unfortunately, the Riemannian metrics adopted in Porikli et al. (2006) are complicated and expensive. Since it uses a global searching strategy, it has to compute distances between the covariance matrices of the model and all candidate regions. Although an integral image based algorithm that requires constant time is proposed to improve the speed, it is still not quick enough for real time tracking. It is difficult for the covariance tracker to track articulated objects since computing covariance matrices for articulated objects is very expensive.

In this work, we propose a tracking strategy that switches between local tracking and global covariance tracking. The switching criteria are determined by the tracking condition. Local tracking is carried out when the target does not have large motion. When large motion or occlusions happen, covariance tracking is adopted to deal with the issue. The switching between local and covariance matching makes the tracking efficient. Moreover, it can deal with sudden motions, distractions, and occlusions in an elegant way. We compute covariance

matrices only on those pixels that are classified as foreground. Therefore we can track articulated objects.
To speed up the global searching process, we use Log-Euclidean metrics Arsigny et al. (2005) instead of the Riemannian invariant metrics Pennec et al. (2006); Porikli et al. (2006) to measure the similarity between covariance matrices. The model update in covariance tracking Porikli et al. (2006) is also expensive. We update the model by computing the geometric mean of covariance matrices based on Log-Euclidean metrics. The computation is simply Euclidean in the logarithmic domain, which reduces the computational costs. The final geometric mean is computed by mapping back to the Riemannian domain with the exponential. Log-Euclidean metrics provide results similar to their Riemannian affine invariant equivalent but takes much less time.
We arrange this chapter as follows. After a brief review of previous works in Section 2, we introduce the local tracking method based on foreground likelihood computation in Section 3. In specific, we discuss target representation for local tracking using color and shape texture information in Section 3.1; we describe our feature selection for local tracking in Section 3.2, and our target localization strategy for local tracking in Section 3.3. In Section 4, we apply Log-Euclidean metric in covariance tracking. We introduce a few basic concepts that are important for our covariance matching in Section 4.1. The extended covariance matching method using Log-Euclidean metric is described in Section 4.2. In Section 5, we give the switching criteria for the local and global tracking. Experimental results are given in Section 6. Section 7 concludes the paper.

2. Related work

Many tracking algorithms assume that target motion is continuous. Given this assumption, we can apply local tracking algorithms Comaniciu et al. (2003); Isard & Blake (1998); Wang & Yagi (2008b). In the local tracking algorithms, the mean-shift algorithm Comaniciu et al. (2003) aims at searching for a peak position using density gradient estimation, whereas particle filtering techniques Isard & Blake (1998); Rathi et al. (2005); Wang & Yagi (2009); Zhao et al. (2008); Zhou et al. (2006) use a dynamic model to guide the particle propagation within a limited sub-space of target state. Particle filtering tracking algorithms have certain robustness against sudden motions. The mean-shift algorithm can deal with partial occlusions.
Tracking can be formulated as template matching Hager & Belhumeur (1998). A target is characterized by a template that can be parametric or non-parametric. The task of a template matching tracking is to find the region that is the most similar to the template. Template matching techniques do not require the continuous motion assumption. Therefore, it is possible to handle occlusions and sudden motions. We will introduce local tracking and global matching techniques. The objective of our algorithm in this chapter it to combine the advantages of the local and global matching techniques.

2.1 Local tracking

There are many local tracking methods. Tracking was treated as a binary classification problem in previous works. An adaptive discriminative generative model was suggested in Lin et al. (2004) by evaluating the discriminative ability of the object from the foreground using a Fisher Linear Discriminant function. Fisher Linear Discriminant function was also using in Nguyen & Smeulders (2006) to provide good discrimination. Comaniciu et al. Comaniciu et al. (2003) take of the advantage of this method to their mean-shift algorithm, where colors that appear on the object are down weighted by colors that appear in the background. Collins et al. Collins & Liu (2005) explicitly treat tracking as a binary classification problem. They apply mean-shift algorithm using discriminative features selected by two online discriminative evaluation methods (Variance ration and peak difference). Avidan Avidan (2007) proposes ensemble tracking that updates a collection of weak classifiers. The Collection of weak classifiers are assembled to make a strong classifier, which separates the foreground object from the background. The weak classifiers are maintained by adding or removing at any time to deal with appearance variations.

Temporal integration methods include particle filtering to properly integrate measurements over time. The WSL tracking that maintains short-term and long term

2.2 Exhaustive matching

Describing a target by one or many templates, tracking can be formulated as exhaustive searching. A target represented by its whole appearance can be matched with each region in the input image by comparing the Sum of Squared Distances (SSD). Template using SSD matching is not flexible because it is sensitive to viewpoint, illumination changes. To deal with these problems, histograms are employed for characterizing targets. Histogram representation is extended to a spatiogram-based tracking algorithm Birchfield & Rangarajan (2005), which makes use of spatial information in addition to color information. A histogram contains many bins which are spatially weighted by the mean and covariance of the location of the pixels that contribute to that bin. Since the target is presented by one histogram, the tracking is not reliable when occlusion exist. The computational cost is also high due to the exhaustive matching. Tuzel et al. Tuzel et al. (2006) introduce covariance matrix to describe the target. This descriptor contains appearance and spatial information. The target localization process is formulated as an expensive exhaustive searching. Moreover, the similarity measure in Tuzel et al. (2006) is adopted from Pennec et al. (2006), which is an affine invariant metric. The affine invariant metric used in Tuzel et al. (2006) is computationally expensive.

3. Local tracking

3.1 Target representation for local tracking

The local tracking is performed based on foreground likelihood. The foreground likelihood is computed using the selected discriminative color and shape-texture features Wang & Yagi (2008a). The target is localized using mean-shift local mode seeking on the integrated foreground likelihood image.

We represent a target using color and shape-texture information. Color information is important because of its simplicity and discriminative ability. Color information only is not always sufficiently discriminative. Shape-texture information is helpful for separating a target and its background. Therefore, the target representation for our local tracking consists of color and shape-texture features.

3.1.1 Multiple Color Channels

We represent color distributions on a target and its background using color histograms. We select several color channels from different color spaces. Among them, we compute color histograms for the R, G, and B channels in the RGB space; the H, S, and V channels in the HSV space. Different from the approach in Wang & Yagi (2008a), we do not use the *r* and *g* channels in the normalized *rg* space because they are found not discriminative in many sequences. Although the *r* and *g* channels have good invariant ability to illumination changes, the gain from this advantage is not very important in our approach since we use global matching and local matching. The histograms computed in R,G, B, H, and S channels are quantized into 12 bins respectively. The color distribution in the V channel is not used here because we found that intensity is less helpful in our tracking tasks. The rg space has been shown to be reliable when the illumination changes. Thus r and g are also employed. There are 5 color features in the candidate feature set.

A color histogram is calculated using a weighting scheme. The contributions of different pixels to the object representation depend on their position with respect to the center of the target. Pixels near the region center are more reliable than those further away. Smaller weights are given to those further pixels by using Epanechnikov kernel Comaniciu et al. (2003) as a weighting function:

$$k(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2), \text{ if } \|\mathbf{x}\|^2 \leq 1; \\ \\ 0, \text{ otherwise}, \end{cases} \tag{1}$$

where c_d is the volume of the unit d-dimensional sphere; $\mathbf{x}$ the local coordinates with respect to the center of the target. Thus, we increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded.

The color distribution $h_f = \{p_f^{(b_{in})}\}_{b_{in}=1\ldots m}$ of the target is given by

$$p_f^{(b_{in})} = C_f \sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}], \tag{2}$$

where δ is the Kronecker delta function and $h(\mathbf{x}_i)$ assigns one of the m-bins ($m = 12$) of the histogram to a given color at location $\mathbf{x}_i$. C_f is a normalization constant. It is calculated as

$$C_f = \frac{1}{\sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|^2)}. \tag{3}$$

The tracking algorithm searches for the target in a new frame from the target candidates. The target candidates are represented by

$$p_c^{(b_{in})}(\mathbf{y}) = C_b \sum_{\mathbf{x}_i \in R_f} k(\frac{\|\mathbf{y} - \mathbf{x}_i\|}{h})^2 \delta[h(\mathbf{x}_i) - b_{in}], \tag{4}$$

where C_b is

$$C_b = \frac{1}{\sum_{\mathbf{x}_i \in R_c} k(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|)^2}. \tag{5}$$

and R_f is the target region.

3.1.2 Shape-texture information

Shape-texture information plays an important role for describing a target. Shape-texture information has a few nice properties such as certain invariant ability to illumination changes. Shape-texture information can be characterized by various descriptors Belongie et al. (2002); Berg & Malik (2001); Lowe (1999). We describe a target's shape-texture information by orientation histograms, which is computed based on image derivatives in x and y directions. We did not use the popular Sobel masks in this calculation. Instead, the Scharr masks (S_x and S_y) are employed here because they give more accurate results than the Sobel kernel.

The gradients at the point (x, y) in the image I can be calculated by convolving the Scharr masks with the image:

$$D_x(x,y) = S_x * I(x,y),$$

and

$$D_y(x,y) = S_y * I(x,y).$$

The strength of the gradient at the point (x, y)

$$D(x,y) = \sqrt{D_x(x,y)^2 + D_y(x,y)^2}.$$

In order to ignore noise, a threshold is given

$$D^{'}(x,y) = \begin{cases} D(x,y), \text{if } D(x,y) \geq T_D, \\ 0, \text{ otherwise,} \end{cases} \tag{6}$$

where T_D is a threshold given empirically.
The orientation of the edge is

$$\theta(x,y) = \arctan(\frac{D_y(x,y)}{D_x(x,y)}). \tag{7}$$

The orientations are also quantized into 12 bins. A orientation histogram can be calculated using a approach similar to the calculation of a color histogram, as introduced in the previous subsection.

3.2 Feature selection for local tracking

We select a subset of features from the feature pool in the 5 color channels and 1 shape-texture representation. We evaluate the discriminative ability of each feature based on the histograms calculated on the target and its background. The discriminative ability of each feature is dependent on the separability between the target and its background. The weighted histograms introduced in the last section do not directly reflect the descriptive ability of the features. A log-likelihood ratio histogram can be helpful for solving this problem Collins (2003); Swain & Ballard (1991); Wang & Yagi (2006). We calculate likelihood images for each feature. Then, we compute likelihood ratio images of the target and its background. Finally, we select good features by ranking the discriminative ability of different features.

3.2.1 Likelihood images

Given target representation using a specific feature, we want to evaluate the probability on an input image. The probability indicates the likelihood of appearance of the target. we We compute foreground likelihood based on the histograms of the foreground and background with respect to a given feature. The frequency of the pixels that appear in a histogram bin is calculated as $\zeta_f^{(b_{in})} = p_f^{(b_{in})}/n_{fg}$ and $\zeta_b^{(b_{in})} = p_b^{(b_{in})}/n_{bg}$, where n_{fg} is the pixel number of the target region and n_{bg} the pixel number of the background.
The log-likelihood ratio of a feature value is given by

$$L^{(b_{in})} = \max(-1, \min(1, \log\frac{\max(\zeta_f^{(b_{in})}, \delta_L)}{\max(\zeta_b^{(b_{in})}, \delta_L)})), \tag{8}$$

where δ_L is a very small number. The likelihood image for each feature is created by back-projecting the ratio into each pixel in the image Swain & Ballard (1991); Wang & Yagi (2008a).

3.2.2 Color and shape-texture likelihood ratio images

Based on the multi-cue representation of the target and its background, we can compute the likelihood probability in an input image. The values in likelihood images have large variations since they are not normalized. We need good representation of different features and evaluate their discriminative ability. Log-likelihood ratios of the the target and background provide such representation. We calculate log-likelihood ratios based on the histograms of the foreground and background with respect to a given feature. The likelihood ratio produces a function that maps feature values associated with the target to positive values and the background to negative values. The frequency of the pixels that appear in a histogram bin is calculated as

$$\zeta_f^{(b_{in})} = \frac{p_f^{(b_{in})}}{n_{fg}}, \tag{9}$$

and

$$\zeta_b^{(b_{in})} = \frac{p_b^{(b_{in})}}{n_{bg}}, \tag{10}$$

where n_{fg} is the pixel number of the target region and n_{bg} the pixel number of the background. The log-likelihood ratio of a feature value is given by

$$L^{(b_{in})} = \max(-1, \min(1, \log \frac{\max(\zeta_f^{(b_{in})}, \delta_L)}{\max(\zeta_b^{(b_{in})}, \delta_L)})), \tag{11}$$

where δ_L is a very small number. The likelihood image for each feature is created by back-projecting the ratio into each pixel in the image.

We use likelihood ratio images as the foundation for evaluating the discriminative ability of the features in the candidate feature set. The discriminative ability will be evaluated using variance ratios of the likelihood ratios, which will be discussed in the next subsection.

3.2.3 Feature selection using variance ratios

Given m_d features for tracking, the purpose of the feature selection module is to find the best subset feature of size m_m, and $m_m < m_d$. Feature selection can help minimize the tracking error and maximize the descriptive ability of the feature set.

We find the features with the largest corresponding variances. Following the method in Collins (2003), based on the equality $\mathrm{var}(x) = E[x^2] - (E[x])^2$, the variance of Equation(11) is computed as

$$\mathrm{var}(L; p) = E[(L^{b_{in}})^2] - (E[L^{b_{in}}])^2.$$

The variance ratio of the likelihood function is defined as Collins (2003):

$$\mathrm{VR} = \frac{\mathrm{var}(B \cup F)}{\mathrm{var}(F) + \mathrm{var}(B)} = \frac{\mathrm{var}(L; (p_f + p_b)/2)}{\mathrm{var}(L; p_f) + \mathrm{var}(L; p_b)}. \tag{12}$$

We evaluate the discriminative ability of each feature by calculating the variance ratio. In the candidate feature set, the color feature includes 7 different features: the color histograms of R, G, B, H, S, r, and g, while the appearance feature includes a gradient orientation histogram. These features are ranked according to the discriminative ability by comparing the variance ratio. The feature with the maximum variance ratio is taken as the most discriminative feature.

3.3 Location estimation for local tracking

We select discriminative features from the color and shape-texture feature pool. These features are employed to compute likelihood images. We extend the basic mean-shift algorithm to our local tracking framework. We combine the likelihood images calculated using different discriminative features. The combined likelihood images are used for our location estimation. In this section, we will introduce the localization strategy in the basic mean-shift algorithm. Then, we discuss how many features are appropriate for the local tracking. Finally, we will describe the localization in our local tracking.

3.3.1 Localization using the standard mean-shift algorithm

The localization process for our local tracking can be described as a minimization process, which aims at searching for the position with maximum similarity with the target. The minimizing process can be formulated as a gradient descent process in the basic mean-shift algorithm. The mean-shift algorithm is a robust non-parametric probability density gradient estimation method. It is able to find the mode of the probability distributions of samples. It can estimate the density function directly from data without any assumptions about underlying distribution. This virtue avoids choosing a model and estimating its distribution parameters Comaniciu & Meer (2002). The algorithm has achieved great success in object tracking Comaniciu et al. (2003) and image segmentation Comaniciu & Meer (2002). However, the basic mean shift tracking algorithm assumes that the target representation is sufficiently discriminative against the background. This assumption is not always true especially when tracking is carried out in a dynamic background such as surveillance with a moving camera. We extend the basic mean shift algorithm to an adaptive mean shift tracking algorithm that can choose the most discriminative features for effective tracking.

The standard mean shift tracker finds the location corresponding to the target in the current frame based on the appearance of the target. Therefore, a similarity measure is needed between the color distributions of a region in the current frame and the target model. A popular measure between two distributions is the Bhattacharyya distance Comaniciu et al. (2003); Djouadi et al. (1990). Considering discrete densities such as two histograms $p = \{p^{(u)}\}_{u=1...m}$ and $q = \{q^{(u)}\}_{u=1...m}$ the coefficient is calculated by:

$$\rho[p,q] = \sum_{b_{in}=1}^{m} \sqrt{p^{(b_{in})} q^{(b_{in})}}. \tag{13}$$

The larger ρ is, the more similar the distributions are. For two identical histograms we obtain $\rho = 1$, indicating a perfect match. As the distance between two distributions, the measure can be defined as Comaniciu et al. (2003):

$$d = \sqrt{1 - \rho[p,q]}, \tag{14}$$

which d is the Bhattacharyya distance.

The tracking algorithm recursively computes an offset value from the current location $\hat{\mathbf{y}}_0$ to a new location $\hat{\mathbf{y}}_1$ according to the mean shift vector. $\hat{\mathbf{y}}_1$ is calculated by using Comaniciu &

Meer (2002); Comaniciu et al. (2003)

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\frac{y-x_i}{h})}{\sum_{i=1}^{n_h} w_i g(\frac{y-x_i}{h})}. \tag{15}$$

where $w_i = \sum_{u=1}^{m} \sqrt{\frac{q^{(u)}}{p^{(u)}(\mathbf{y}_0)}}\delta[h(\mathbf{x}_i) - b_{in}]$ and $g(x) = -k'(x)$.

3.3.2 How many features are appropriate?

We evaluate the discriminative abilities of different features in the feature pool. In the Evaluation, we rank the features according to their discriminative ability against the background. Features with good discriminative ability can be combined to represent and localize the target. The combination of features needs to be carried out carefully. Intuitively, the more features we use, the better the tracking performance; however, this is not true in practice. According to information theory, the feature added into the system can bring negative effect as well as improvement of the performance Cover & Thomas (1991). This is due to the fact that the features used are not totally independent. Instead, they are correlated. In our implementation, two kinds of features are used to represent the target, a number, which according to the experimental results, is appropriate in most cases. We have tested a system using 1 or 3 features, which gave worse performances. During the initialization of the tracker, the features ranked in the top two are selected for the tracking. The feature selection module runs every 8 to 12 frames. When the feature selection module selects features different from those in the initialization, only one feature is replaced each time. Only the second feature of the previous selection will be discarded and replaced by the best one in current selection. This strategy is very important in keeping the target from drifting.

3.3.3 Target localization for local tracking

The proposed tracking algorithm combines the top two features through back-projection Bradski (1998) of the joint histogram, which implicitly contains certain spatial information that is important for the target representation. Based on Equation(4), we calculate the joint histogram of the target with the top two features,

$$p_f^{(b_{in}^{(1)}, b_{in}^{(2)})} = C \sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}^{(1)}]\delta[h(\mathbf{x}_i) - b_{in}^{(2)}], \tag{16}$$

and a joint histogram of the searching region

$$p_b^{(b_{in}^{(1)}, b_{in}^{(2)})} = C \sum_{\mathbf{x}_i \in R_b} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}^{(1)}]\delta[h(\mathbf{x}_i) - b_{in}^{(2)}]. \tag{17}$$

We get a division histogram by dividing the joint histogram of the target by the joint histogram of the background,

$$p_d^{(b_{in}^{(1)}, b_{in}^{(2)})} = \frac{p_f^{(b_{in}^{(1)}, b_{in}^{(2)})}}{p_b^{(b_{in}^{(1)}, b_{in}^{(2)})}}. \tag{18}$$

The division histogram is normalized for the histogram back-projection. The pixel values in the image are associated with the value of the corresponding histogram bin by histogram back-projection. The back-projection of the target histogram with any consecutive frame generates a probability image $p = \{p_w^i\}_{i=1...n_h}$ where the value of each pixel characterizes the probability that the input pixel belongs to the histograms. The two images of the top two features have been computed for the back-projection. Note that the H, S, r, and g images are calculated by transferring the original image to the HSV and the rg spaces; the orientation image has been calculated using the approach introduced in section III(B).

Since we are using an Epanechnikov profile the derivative of the profile, $g(x)$, is constant. The target's shift vector in the current frame is computed as

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i p_w^i}{\sum_{i=1}^{n_h} p_w^i}. \tag{19}$$

The tracker assigns a new position to the target by using

$$\hat{\mathbf{y}}_1 = \frac{1}{2}(\hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1). \tag{20}$$

If $\|\hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_1\| < \varepsilon$, this position is assigned to the target. Otherwise, compute the Equation(19) again. In our algorithm, the number of the computation is set to less than 15. In most cases, the algorithm converges in 3 to 6 loops.

3.4 Target model updating for local tracking

The local tracker needs adaptivity to handle appearance changes. The model is computed by mixing the current model with the initial model which is considered as correct Wang & Yagi (2008a). The mixing weights are generated from the similarity between the current model and the initial model Wang & Yagi (2008a). The initial model works in a similar way to the stable component in Jepson et al. (2003). But the updating approach in Wang & Yagi (2008a) takes less time.

Updating the target model adaptively may lead to tracking drift because of the imperfect classification of the target and background. Collins and Liu Collins (2003) proposed that forming a pooled estimate allows the object appearance model to adapt to current conditions while keeping the overall distribution anchored to the original training appearance of the object. They assume that the initial color histogram remains representative of the object appearance throughout the entire tracking sequence. However, this is not always true in real image sequences.

To update the target model, we propose an alternative approach that is based on similarities between the initial and current appearance of the target. The similarity s is measured by a simple correlation based template matching Atallah (2001) performed between the current and the initial frames. The updating is done according to the similarity s:

$$H_m = (1 - s)H_i + sH_c, \tag{21}$$

where the H_i is the histogram computed on the initial target; the H_c the histogram of the target current appearance, the H_m the updated histogram of the target.

The template matching is performed between the initial model and the current candidates. Since we do not use the search window that is necessary in template matching-based tracking, the matching process is efficient and brings little computational cost to our algorithm. The performance of the proposed algorithm is improved by using this strategy, which will be shown in the next section.

4. Covariance matching in riemannian manifold

We describe our covariance matching in Riemmannian manifold in this section. We introduce some important theories on Riemannian manifold. Since the affine invariant metric used in Tuzel et al. (2006) is computationally expensive, we apply the efficient Log-Euclidean metric in the manifold. Finally, we give the updating strategy for the covariance matching.

4.1 Basic concepts for global matching in riemannian manifold

We will introduce some basic concepts of Riemannian geometry, which is important for our global tracking formulation. We describe differentiable manifold, Lie groups, Lie algebras, and Riemannian manifold. The details of the theories are referred to Gilmore (2006); Jost (2001).

4.1.1 Differentiable manifold

A manifold $\mathcal{M}$ is a Hausdorff topological space, such that for every point $\mathbf{x} \in \mathcal{M}$ there exists a neighborhood $\mathcal{N} \subset \mathcal{M}$ containing x and an associated homeomorphism from $\mathcal{N}$ to some Euclidean space R^m. The neighborhood $\mathcal{N}$ and its associated mapping ϕ together form a coordinate chart. A collection of chart is named as an atlas.

If a manifold is locally similar enough to Euclidean space, it is allowed to do calculus. A differentiable manifold is such kind of manifold that is also a topological manifold with globally defined differential structure. Any topological manifold can be given a differential structure locally by using the homeomorphisms in this atlas. One may apply ideas from calculus which working within the individual charts, since these lie in Euclidean spaces to which the usual rules of calculus apply.

4.1.2 Lie groups

Lie groups are finite-dimensional real smooth manifold with continuous transformation group properties Rossmann (2003). Group operations can be applied into Lie groups.

Assuming we have two groups, G_1 and G_2, we can define a homomorphism $f_A : G_1 \to G_2$ for them. The homomorphism f is required to be continuous (not necessarily to be smooth). If we have another homomorphism $f_B : G_3 \to G_4$, the two homomorphisms are combined into a new homomorphism. A category is formulated by composing all the Lie groups and morphisms. According to the type of homomorphisms, there are two kinds of Lie groups: isomorphic Lie groups with bijective homomorphisms.

Homomorphisms are useful in describing Lie groups. We can represent a Lie group on a vector space V. We chose a basis for the vector space, the Lie group representation is expressed as a homomorphisms into $GL(n, K)$, which is known as a matrix representation. If we have two vector spaces V_1 and V_2, the two representations of G on V_1 and V_2 are equivalent when they have the same matrix representations with respect to some choices of bases for V_1 and V_2.

4.1.3 Lie algebras

We may consider Lie groups as smoothly varying families of symmetries. Small transformation is an essential property of Lie groups. In such situations, Lie algebras can be defined because Lie groups are smooth manifold with tangent spaces at each point. Lie algebra, an algebraic structure, is critical in studying differentiable manifolds such as Lie groups. Lie algebra is able to replace the global object, the group, with its local or linearized version. In practice, matrices sets with specific properties are the most useful Lie groups.

Matrix Lie groups is defined as closed subgroups of general linear groups $\mathbf{GL}(n, \mathcal{R})$, the group of $n \times n$. nonsingular matrices.

We associate a Lie algebra with every Lie group. The underlying vector space is the tangent space of Lie group at the identity element, which contains the complete local structure of the group. The elements of the Lie algebra can be thought as elements of the group that are infinitesimally close to the identity. The Lie algebra provides a commutator of two such infinitesimal elements with the Lie bracket. We can connect vector space with Lie algebra preserves the Lie bracket.

Each Lie group has a identity component, which is an open nomal subgroup. All the connected Lie groups forms the universal cover of these groups. Any Lie group G can be decomposed into discrete abelian groups.

We can not define a global structure for a Lie group using its Lie algebra. However, if the Lie group is simply connected, we can determine the global structure based on its Lie algebra.

Tensors are defined as multidimensional arrays of numbers. It is an extension of matrix, which is a 2D definition. The entries of such arrays are symbolically denoted by the name of tensor with indices giving the position in the array. Covariance

4.1.4 Exponential maps

A Lie algebra homomorphism is a mapping: every vector v in Lie algebra g is a linear map from R taking 1 to v. Because R is the Lie algebra of the simply connected Lie group R, this induces a Lie group homorphism $f : R \rightarrow G$. The operation of c is

$$c(s+t) = c(s)c(t) \tag{22}$$

for all s and t. We easily find that it is similar to exponential function

$$exp(v) = c(1). \tag{23}$$

This exponential function is name as exponential map which maps the Lie algebra g into the Lie group G. Between a neighborhood of the identity element of g, there is a diffeomorphism. The exponential map is a generalization of the exponential function for real numbers. In fact, the exponential function can be extended into complex numbers and matrices, which is important in computing Lie groups and Lie algebras.

Since we are interested in symmetric matrices, matrix operators are important for the computation on Lie algebra. The exponential map from the Lie algebra is defined by

$$\exp(A) = \sum_{i=0}^{\infty} \frac{1}{i!} A^i, \tag{24}$$

It is possible to decompose A into an orthogonal matrix U and a diagonal matrix ($A = UDU^T$, $D = DIAG(d_i)$), we compute power k of A using the same basis

$$A^k = UD^k U^T, \tag{25}$$

where the rotation matrices in the computation is factored out. The mapping of exponential to each eigenvalue:

$$\exp(A) = U\mathrm{DIAG}(\exp(d_i))U^T. \tag{26}$$

An inverse mapping is defined in the neighborhood: $\log_{\mathbf{X}} : G \rightarrow T_{\mathbf{X}}R$. This definition is unique. For certain manifolds, the neighborhood can be extended more regions in the tangent space and manifold.
This operator is able to be applied to any square matrix. The definitions above are meaningful only for matrix groups. Since we concern matrix groups in this work, the definitions are very important for understanding our algorithm.

4.2 Improving covariance tracking

The covariance tracker Porikli et al. (2006) describes objects using covariance matrices. The covariance matrix fuses different types of features and modalities with small dimensionality. Covariance tracking searches all the regions and guarantees a global optimization (Up to the descriptive ability of the covariance matrices). Despite of these advantages, covariance tracking is relatively expensive due to the distance computation and model updating in Riemannian manifold. We speed up the global searching and the model updating by introducing Log-Euclidean metrics.

4.2.1 Target representation

The target is described by covariance matrices that fuse multiple features. We adopt the features used in Porikli et al. (2006), which consist of pixel coordinates, RGB colors and gradients. The region R is described with the $d \times d$ covariance matrix of the feature points in R

$$C_R = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{z}_k - \bar{})(\mathbf{z}_k - \bar{})^T, \tag{27}$$

where $\bar{}$ is the mean of the points.
The covariance of a certain region reflects the spatial and statistical properties as well as their correlations of a region. However, the means of the features are not taken into account for tracking. We use the means by computing the foreground likelihoods and incorporate them into the covariance computation.

4.2.2 Similarity measuring for covariance matrices

The simplest way for measuring similarity between covariance matrices is to define a Euclidean metric, for instance, $d^2(C_1, C_2) = \text{Trace}((C_1 - C_2)^2)$ Arsigny et al. (2005). However, the Euclidean metric can not be applied to measure the similarity due to the fact that covariance matrices may have null or negative eigenvalues which are meaningless for the Euclidean metrics Forstner & Moonen (1999). In addition, the Euclidean metrics are not appropriate in terms of symmetry with respect to matrix inversion, e.g., the multiplication of covariance matrices with negative scalars is not closed for Euclidean space.
Since covariance matrices do not lie on Euclidean space, affine invariant Riemannian metrics Forstner & Moonen (1999); Pennec et al. (2006) have been proposed for measuring similarities between covariance matrices. To avoid the effect of negative and null eigenvalues, the distance measure is defined based on generalized eigenvalues of covariance matrices:

$$\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)}, \tag{28}$$

where $\{\lambda_i(C_1, C_2)\}_{i=1\ldots n}$ are the generalized eigenvalues of C_1 and C_2, computed from

$$\lambda_i C_1 \mathbf{x}_i - C_1 \mathbf{x}_i = 0, i = 1 \ldots d, \tag{29}$$

and $\mathbf{x}_i \neq 0$ are the generalized eigenvectors. The distance measure ρ satisfies the metric axioms for positive definite symmetric matrices C_1 and C_2. The price paid for this measure is a high computational burden, which makes the global searching expensive.

In this work, we use another Riemannian metrics – Log-Eucliean metrics proposed in Arsigny et al. (2005). When only the multiplication on the covariance space is considered, covariance matrices have Lie group structures. Thus the similarity can be measured in the domain of logarithms by Euclidean metrics:

$$\rho_{LE}(C_1, C_2) = \| \log(C_1) - \log(C_2) \|_{Id}. \tag{30}$$

This metric is different from the classical Euclidean framework in which covariance matrices with null or negative eigenvalues are at an infinite distance from covariance matrices and will not appear in the distance computations.

Although Log-Eucliean metrics are not affine-invariant Arsigny et al. (2005), some of them are invariant by similarity (orthogonal transformation and scaling). It means that the Log-Euclidean metrics are invariant to changes of coordinates obtained by a similarity Arsigny et al. (2005). The properties of Log-Euclidean make them appropriate for similarity measuring of covariance matrices.

4.3 Model updating

Covariance tracking has to deal with appearance variations. Porikli et al. Porikli et al. (2006) construct and update a temporal kernel of covariance matrices corresponding to the previously estimated object regions. They keep a set of previous covariance matrices $[C_1 \ldots C_T]$. From this set, they compute a sample mean covariance matrix that blends all the previous matrices. The sample mean is an intrinsic mean Porikli et al. (2006) because covariance matrices do not lie on Euclidean spaces. Since covariance matrices are symmetric positive definite matrices, they can be formulated as a connected Riemannian manifold. The structure of the manifold is specified by a Riemannian metric defined by collection of inner products. The model updating is computationally expensive due to the heavy burden of computation in Riemannian space.

In this work, we use the Log-Euclidean mean of T covariance matrices with arbitrary positive weights $(w_i)_{i=1}^{T}$ such that $\sum_{i=1}^{T} w_i = 1$ is a direct generalization of the geometric mean of the matrices. It is computed as

$$C_m = \exp(\sum_{i=1}^{T} \log(C_i)). \tag{31}$$

This updating method need much less computational costs than the method used in Porikli et al. (2006).

5. Switching criteria

The local tracking strategy is adopted when the tracker runs in steady states. When sudden motion, distractions or occlusions happen, local tracking strategy tends to fail due to its limited searching region. We switch to the global searching strategy based on the improved covariance tracker described in the previous section. Motion prediction techniques such the Kalman filter have been used to deal with occlusions. However, when the prediction is far away from the true location, a global searching is preferred to recover from tracking failure.

Algorithm	Seq1	Seq2	Seq3
Meanshift	72.6	78.5	35.8
Covariance	89.7	90.4	78.8
TheProposed	91.3	88.1	83.1

Table 1. Tracking percentages of the proposed and other trackers.

The detection of sudden motion and distraction is performed using the effective methods proposed in Wang & Yagi (2007). Occlusions are announced when the objective function value of the local tracking is lower than some threshold t_l. The threshold for switching between local and covariance tracking is computed by fitting a Gaussian distribution based on the similarity scores (Bhattacharyya distances) of the frames labeled as occlusion. The threshold is set to $3\sigma_t$ from the mean of the Gaussian. The covariance tracking is applied when the above threats are detected.

6. Experiments

We verify our approach by tracking different objects in some challenging video sequences.
We compare the performance of the mean-shift algorithm and the proposed method in Figure. 1. The face in the sequence moves very fast. Therefore, the mean-shift tracker fails to capture the face. The proposed method combines multiple features for local tracking. It is possible to track the target thorough the sequence. The example in Figure. 1 demonstrate the power of the local tracking part in our approach.
In Figure. 2, we show the tracking results on the street sequence Leibe et al. (2007). Pedestrians are articulated objects which are difficult to track. The occlusions in frame 7574 brings more difficulty to the tracking. The proposed tracker successfully tracks through the whole sequence.
We compare the proposed tracker with the mean-shift and covariance trackers. Different objects in the three sequences Leibe et al. (2007) are tracked and the tracking percentages are given in Table. 1. The proposed tracker provides higher or similar correct ratio.

6.1 Computation complexity

The tracking is faster when the local tracking method is applied since the searching of local tracking is only performed on certain the regions. It takes less than 0.02 seconds to process one frame.
The covariance tracking is also sped up thanks to the efficiency of Log-Euclidean distance computation adopted in this work. The iterative computation of the affine invariant mean leads to heavy computational cost. In contrast, the Log-Euclidean metrics are computed in a closed form. The computation of mean based on Log-Euclidean distances takes less than 0.02 seconds, whereas the computation based on Riemannian invariant metrics takes 0.4 seconds.

7. Conclusions

We propose a novel tracking framework taking the advantages of local and global tracking strategies. The local and global tracking are performed by using the mean-shift and covariance matching. The proposed tracking algorithm is efficient because local searching strategy is adopted for most of the frames. It can deal with occlusions and large motions for the switching

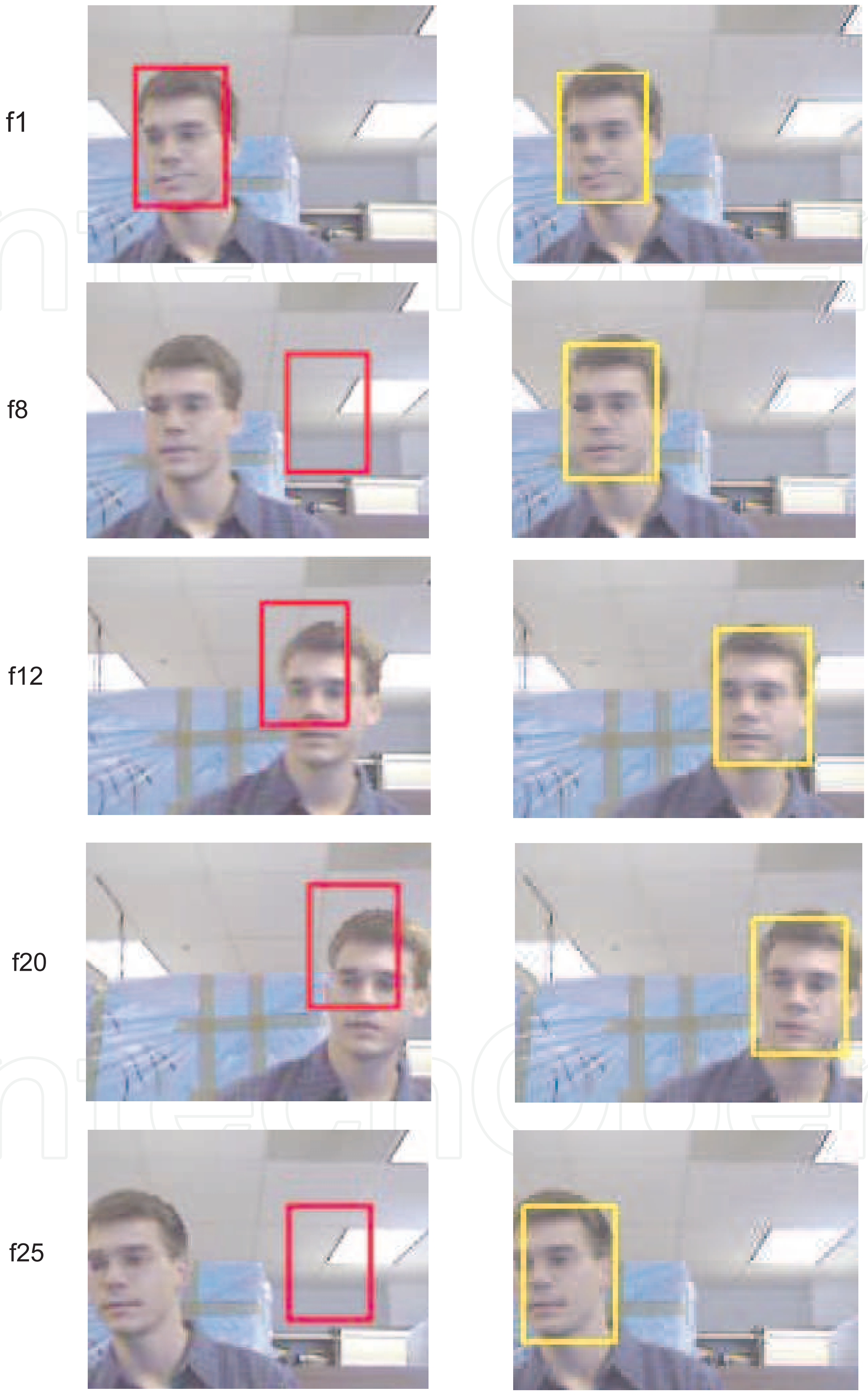


Fig. 1. Face tracking results using the basic mean shift algorithm (in the first row) and the proposed method (in the second row). The face in the sequence moves quickly.

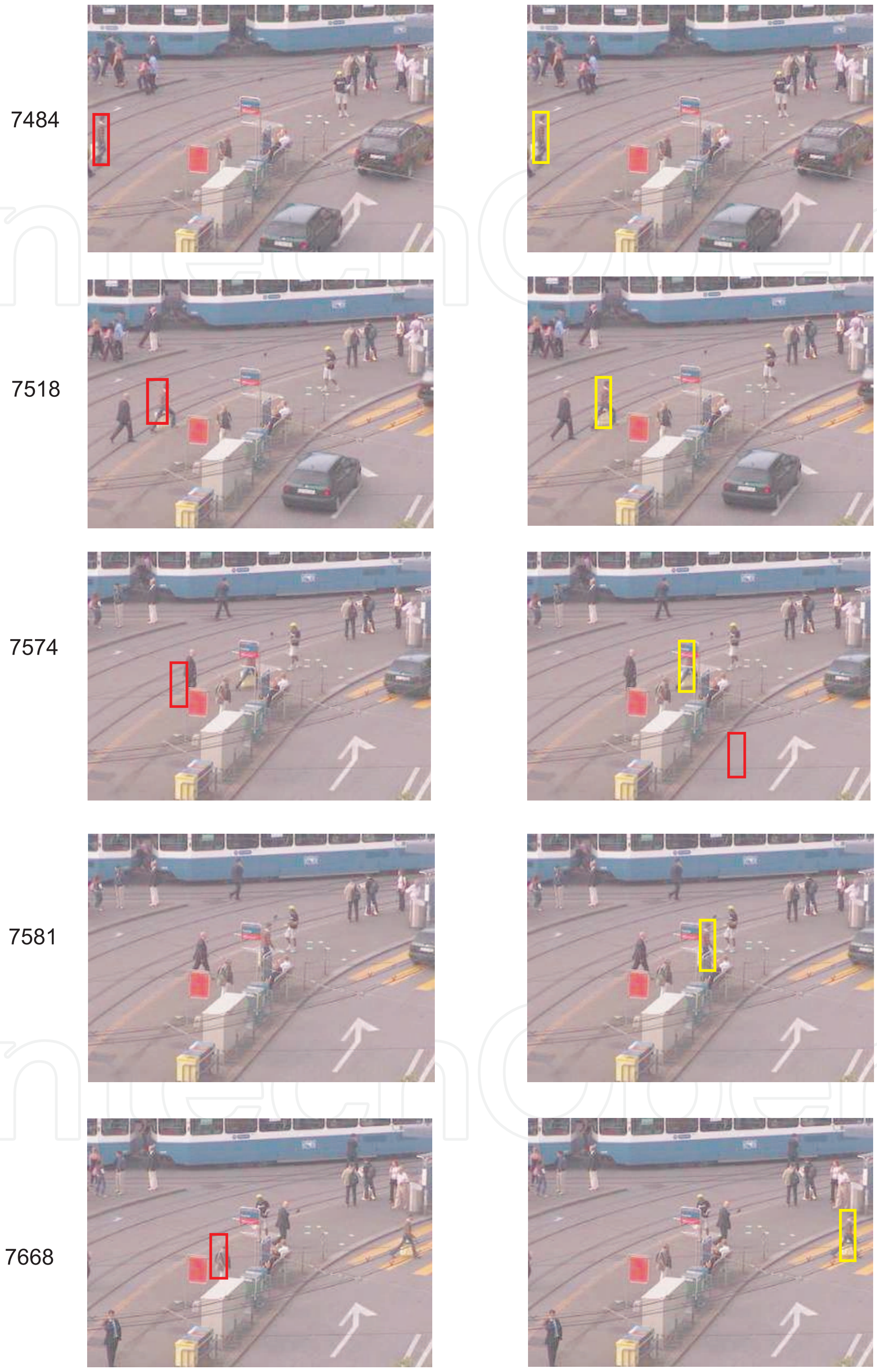


Fig. 2. Tracking pedestrian in the complex background. No background subtraction is applied in the tracking.

from local to global matching. We adopt Log-Euclidean metrics in the improved covariance tracking, which makes the global matching and model updating fast.

8. References

Arsigny, V., Fillard, P., Pennec, X. & Ayache, N. (2005). Fast and simple calculus on tensors in the log-euclidean framework, *Proc. MICCAI'05*, pp. 115–122.

Atallah, M. J. (2001). Faster image template matching in the sum of the absolute value of differences measure, *IEEE Transactions on Image Processing* 10(4): 659–663.

Avidan, S. (2007). Ensemble tracking, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(2): 261–271.

Belongie, S., Malik, J. & Puzicha, J. (2002). Shape matching and object recognition using shape contexts, *IEEE Trans. Pattern Anal. Mach. Intell.* 24(4): 509–522.

Berg, A. C. & Malik, J. (2001). Geometric blur for template matching, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 607–614.

Birchfield, S. & Rangarajan, S. (2005). Spatiograms versus histograms for region-based tracking, *Proc. of Conf. Computer Vision and Pattern Recognition*, pp. 1158–1163.

Bradski, G. (1998). Computer vision face tracking as a component of a perceptural user interface, *Proc. of the IEEE Workshop Applications of Computer Vision*, pp. 214–219.

Collins, R. T. (2003). Mean-shift blob tracking through scale space, *Proc. CVPR*, pp. 234–240.

Collins, R. T. & Liu, Y. (2005). On-line selection of discriminative tracking features, *IEEE Trans. Pattern Anal. Mach. Intell.* 27(10): 1631–1643.

Comaniciu, D. & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(5): 603–619.

Comaniciu, D., Ramesh, V. & Meer, P. (2003). Kernel-based object tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* 25(5): 564–577.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, John Wiley and Sons Press.

Djouadi, A., Snorrason, O. & Garber, F. D. (1990). The quality of training sample estimates of the bhattacharyya coefficient, *IEEE Trans. Pattern Anal. Mach. Intell.* 12(1): 92–97.

Forstner, W. & Moonen, B. (1999). A metric for covariance matrices, *Technical report, Dept. of Geodesy and Geoinformatics* .

Gilmore, R. (2006). *Lie Groups, Lie Algebras, and Some of Their Applications*, Dover Publications.

Hager, G. D. & Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination, *IEEE Trans. Pattern Anal. Mach. Intell.* 20(10): 1025–1039.

Isard, M. & Blake, A. (1998). Condensation - conditional density propagation for tracking, *Intl. Journal of Computer Vision* 29(1): 2–28.

Jepson, A. D., Fleet, D. J. & EI-Maraghi, T. (2003). Robust online appearance models for visual tracking, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(10): 1296–1311.

Jost, J. (2001). *Riemannian Geometry and Geometric Analysis*, Springer.

Leibe, B., Schindler, K. & Gool, L. V. (2007). Coupled detection and trajectory estimation for multi-object tracking, *Proc. of Int'l Conf. on Computer Vision*, pp. 115–122.

Lin, R.-S., Ross, D. A., Lim, J. & Yang, M.-H. (2004). Adaptive discriminative generative model and its applications, *Proc. Conf. Neural Information Processing System*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features, *Proc. ICCV'99*, pp. 1150–1157.

Nguyen, H. T. & Smeulders, A. W. M. (2006). Robust tracking using foreground-background texture discrimination, *International Journal of Computer Vision* 69(3): 277–293.

Pennec, X., Fillard, P. & Ayache, N. (2006). A riemannian framework for tensor computing, *Intl. Journal of Computer Vision* 66: 41–66.

Porikli, F., Tuzel, O. & Meer, P. (2006). Covariance tracking using model update based on lie algebra, *Proc. of Intl Conf. on Computer Vision and Pattern Recognition*, pp. 728–735.

Rathi, Y., Vaswani, N., Tannenbaum, A. & Yezzi, A. (2005). Particle filtering for geometric active contours with application to tracking moving and deforming objects, *Proc. of Conf. Computer Vision and Pattern Recognition*, pp. 2–9.

Rossmann, W. (2003). *Lie groups: an introduction through linear groups*, London: Oxford University Press.

Swain, M. & Ballard, D. (1991). Color indexing, *Intl. Journal of Computer Vision* 7(1): 11–32.

Tuzel, O., Porikli, F. & Meer, P. (2006). Region covariance: A fast descriptor for detection and classification, *ECCV*, pp. 589–600.

Wang, J. & Yagi, Y. (2006). Integrating shape and color features for adaptive real-time object tracking, *Proc. of Conf. on Robotics and Biomimetrics*, pp. 1–6.

Wang, J. & Yagi, Y. (2007). Discriminative mean shift tracking with auxiliary particles, *Proc. 8th Asian Conference on Computer Vision*, pp. 576–585.

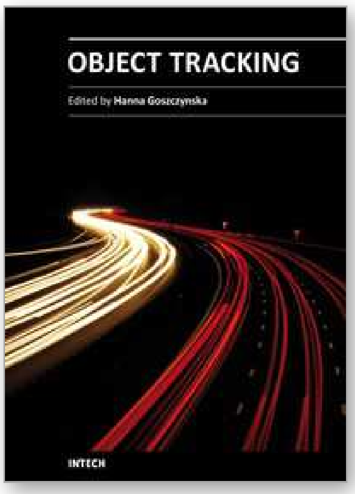
Wang, J. & Yagi, Y. (2008a). Integrating color and shape-texture features for adaptive real-time tracking, *IEEE Trans. on Image Processing* 17(2): 235–240.

Wang, J. & Yagi, Y. (2008b). Patch-based adaptive tracking using spatial and appearance information, *Proc. International Conference on Image Processing*, pp. 1564–1567.

Wang, J. & Yagi, Y. (2009). Adaptive mean-shift tracking with auxiliary particles, *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* 39(6): 1578–1589.

Zhao, T., Nevatia, R. & Wu, B. (2008). Segmentation and tracking of multiple humans in crowded environments, *IEEE Trans. Pattern Anal. Mach.* 30(7): 1198–1211.

Zhou, S. K., Georgescu, B., Comaniciu, D. & Shao, J. (2006). Boostmotion: boosting a discriminative similarity function for motion estimation, *Proc. of CVPR*, pp. 1761–1768.



Object Tracking
Edited by Dr. Hanna Goszczynska

ISBN 978-953-307-360-6
Hard cover, 284 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

Object tracking consists in estimation of trajectory of moving objects in the sequence of images. Automation of the computer object tracking is a difficult task. Dynamics of multiple parameters changes representing features and motion of the objects, and temporary partial or full occlusion of the tracked objects have to be considered. This monograph presents the development of object tracking algorithms, methods and systems. Both, state of the art of object tracking methods and also the new trends in research are described in this book. Fourteen chapters are split into two sections. Section 1 presents new theoretical ideas whereas Section 2 presents real-life applications. Despite the variety of topics contained in this monograph it constitutes a consisted knowledge in the field of computer object tracking. The intention of editor was to follow up the very quick progress in the developing of methods as well as extension of the application.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821