Adaptive and Discriminative Metric Differential Tracking

Nan Jiang [1], Wenyu Liu [1], Ying Wu [2]

[1] Huazhong University of Science and Technology
Wuhan, Hubei, P. R. China.
{qiningonline,liuwy}@mail.hust.edu.cn

[2] Northwestern University
Evanston, IL, USA.
yingwu@eecs.northwestern.edu

Abstract

Matching the visual appearances of the target over consecutive image frames is the most critical issue in video-based object tracking. Choosing an appropriate distance metric for matching determines its accuracy and robustness, and significantly influences the tracking performance. This paper presents a new tracking approach that incorporates adaptive metric into differential tracking method. This new approach automatically learns an optimal distance metric for more accurate matching, and obtains a closed-form analytical solution to motion estimation and differential tracking. Extensive experiments validate the effectiveness of adaptive metric, and demonstrate the improved performance of the proposed new tracking method.

1. Introduction

One of the most critical issues in visual target tracking is to match the visual appearances of the target over consecutive image frames. This process can be generally formulated as:

$$\min_{\Delta x} \mathcal{D}\left(\mathbf{f}(\mathbf{x}, t), \mathbf{f}(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)\right), \qquad (1)$$

where $\mathbf{f}(\mathbf{x}, t)$ is the visual feature of the target at location $\mathbf{x}$ at time t, and $\mathcal{D}(\cdot, \cdot)$ is the distance metric in the feature space. Similar to many other computer vision problems such as object recognition, this process is largely stipulated by the interplay of two factors: visual features that characterize the target in a feature space, and the distance metric that is used to determine the closest match in such a feature space. Different from those recognition tasks, a uniqueness in this tracking task is that the matching is constructed between two consecutive frames over Δt, and thus it requires a computationally more efficient solution.

If we can identify strong features, e.g., those that are invariant to lighting changes and local deformation, and those that are discriminative from the distracters (or false positive matches), even we only use the Euclidean distance metric for simple nearest-neighbor matching, good results can be expected. But when strong features cannot be easily specified, the choices of the distance metric largely influence the matching performance. Thus, finding an appropriate metric for visual tracking becomes critical. This paper does not intend to find new image features. Instead, assuming a given set of specified image features, this paper concentrates on the selecting the adaptive distance metric for visual tracking.

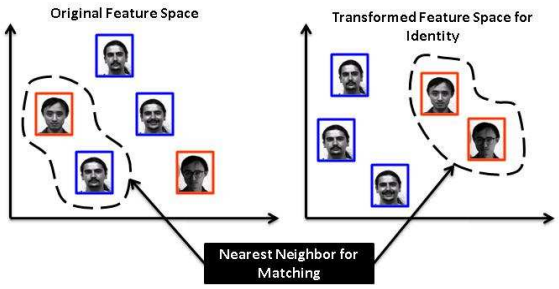

Figure 1. Illustration of distance metrics in matching. The closest match can be different due to the metric we choose.

Most existing tracking methods employ a fixed pre-specified metric. For example, besides the Euclidean metric, many other choices have been explored, including the Matusita metric [12], the Bhattacharyya coefficient [8], the Kullback-Leibler divergence [9], the information-theoretic similarity measures [15], and a combination of those [19].

However, simply using a pre-specified metric is problematic and limited in practice. One phenomenon we often observed is that the closest match under a predefined metric in a given feature space may not be the true target of interest, because the target is represented by its visual features, some of them may be more discriminative than others in telling the target apart from the background and distracters. For example, as illustrated in Fig. 1, when we want to track a particular face, the closest neighbor in a certain given feature space can be another person, if the target face is subject to some significant variations, e.g., lighting or pose changes.

Unless these uncertainties can be modeled or learned in advance, it is risky to use a pre-specified metric regardless of the appearance uncertainties. Moreover, in the tracking scenario, as the background can be dynamic and the distracters may keep changing, using a predefined metric regardless is not plausible. Therefore, a good metric in tracking must be learned, be discriminative, and be adaptive.

Different from the methods based on pre-specified metrics mentioned above, our work is targeted on a new learned and adaptive metric for differential tracking. It achieves a better separability of the target from the background and the distracters nearby. Our method is not limited to color features, but is more generally applicable.

The proposed method adaptively learns a good Mahalanobis metric that weights features differently, implying the identification of discriminative features. This is especially useful when tracking the target in a quite cluttered and distractive environment. Besides, our method integrates this adaptive metric into differential tracking. As the collection of the training data is performed on-the-fly, our method is naturally adaptive. In addition, our solution to motion estimation is in a closed-form, which enables a computationally efficient solution to target tracking. Moreover, our method allows dimension reduction by embedding the high dimensional feature space into a lower dimensional space, which further enhances the computational efficiency.

The organization of this paper is as follows: after a brief description of related works in Sec.2, the basic formulation of metric learning is introduced in Sec.3. The differential tracking method under metric learning is presented in Sec.4. The experiments are reported and discussed in Sec.5, and followed by the conclusion.

2. Related Work

Feature selection and distance metric are two closely related concepts, and both can contribute to the improvement of tracking performances. They can be unified in some special cases. For example, feature selection and linear distance metric adjustment can be regarded as a linear transform of the original feature space, $\mathbf{R}^* = \mathbf{A}\mathbf{R}$, where $\mathbf{R} \in \mathbb{R}^m$ is the original feature, and $\mathbf{A}$ represents the linear transform, and $\mathbf{R}^* \in \mathbb{R}^d$ is the new feature. $\mathbf{A}$ does not have to be a square matrix. To select some discriminative features, such as [13, 6], we can require that each row of $\mathbf{A}$ be an indicator row vector, i.e., all but one element in this vector are zeros. The method proposed in [21] integrates different kinds of features by distributing the weights according to their discrimination power. This can be regarded as using a diagonal matrix of $\mathbf{A}$. Then many discriminative learning methods, e.g., support vector machines [1] and boosting [2, 3], can be employed to identify useful features before combining them. The feature identification process can also be treated as learning the diagonal matrix $\mathbf{A}$ (either in batch or sequential processing). In other words, the dynamic feature selection methods, discriminative learning methods and linear distance metric learning methods share the same mathematical structure. Since dynamic feature selection and discriminative learning generally assume a sparse transformation matrix in advance, they are special cases of the general distance learning method, as proposed in this paper.

General distance metric learning methods can be divided into two categories: unsupervised and supervised approaches. In this paper, we focus on the supervised learning method. Supervised distance metric learning approaches aim to learn appropriate distance metrics for better class discrimination. Given similar and dissimilar data pairs as the supervised information, the methods in [18] and [4] explicitly learn to reduce the distances between similar data pairs, and to enlarge the distances between dissimilar data pairs. Given the training data and the known labels, the neighborhood component analysis (NCA) algorithm [11] learns a Mahalanobis distance metric to achieve the best performance for the k-NN classifier by maximizing the classification error in the leave-one-out cross validation. This is achieved by adopting the *soft-max* formulation for the nearest-neighbor classification.

A very recent piece of work related to our method was proposed in [16]. The TUDAMM method described in [16] integrates the appearance modeling and motion estimation together in an Expectation-Maximization(EM) like algorithm. In this method, its distance metric learning largely follows the concept proposed by Globerson and Roweis [10]. In Globerson and Roweis' method, it constructs a convex optimization problem that aims to find a metric by attempting to collapse all the examples in the same class to a single point and to push the examples in other classes infinitely far away. However, mapping all the examples in the same class to a single point is a much stronger constraint than the minimization task in the soft-max formulation as in NCA [11]. Comparing with NCA, [10] is much harder to be achieved.

3. Adaptive Metric Learning

3.1. Formulation

Distance metric is fundamental to the tracking problem. In this paper, we focus on adaptively adjusting the metric to improve the matching performance in differential tracking.

We use a Mahalanobis metric:

$$\begin{aligned} \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j)^T(\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j) \\ &= (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{Q}(\mathbf{x}_i - \mathbf{x}_j). \end{aligned} \quad (2)$$

The Mahalanobis distance metric is parameterized by a matrix $\mathbf{A} \in \mathbb{R}^{d\times m}$ that linearly transforms the original fea-

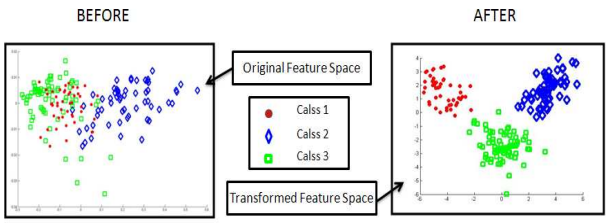

Figure 2. Adaptive distance metric learning on synthetic data. The left figure indicates the training samples in original feature space, and the right figure shows these samples in the transformed feature space through adaptive metric learning.

ture space $\mathbb{R}^m$ to a new space $\mathbb{R}^d$ (where $d \leq m$ in general). And $\mathbf{Q}_{m\times m} = \mathbf{A}^T\mathbf{A}$. Then, tracking can be formulated as:

$$\min_{\Delta x} \|\mathbf{A}\mathbf{f}(\mathbf{x}, t) - \mathbf{A}\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)\|^2. \qquad (3)$$

Once we have supervised training data, the transform $\mathbf{A}$ can be adaptively and discriminatively learned from the supervised training data. We can increase the discriminative power of the tracker to better separate the target from the distracters in the transformed feature space. This new space shrinks the distances among the data in the same class, and tolerates their appearance variations. When $\mathbf{A}$ is an identity matrix, the Mahalanobis distance is degenerated to the Euclidean distance.

Fig.2 illustrates the main idea behind the distance metric learning on a synthetic data set. Before applying the transform $\mathbf{A}$, the data points in the original feature space are mixed together, as shown to the left in Fig.2. The transform $\mathbf{A}$ pulls the data points of the same class together, and pushes the other data points away, as shown to the right in Fig.2. By design, this metric adjustment is expected to largely improve the matching accuracy, and thus improving the tracking accuracy. Fig.2 shows an example where the incorrect matching in the original input space can be corrected by adaptively learning an appropriate linear metric.

In this formulation, the method proposed in [6] can be treated as a special case of our method, where only three parameters in the transform $\mathbf{A}$ are adjustable, and it employs an exhaustive search to identify the best parameters. As a more general treatment, our formulation allows a much more flexible adjustment, and search for an optimal adjustment in a more efficient way. Moreover, the transform $\mathbf{A}$ in our method is not necessary to be a square matrix, which can be regarded as a low-dimensional embedding of the original space.

The general approach of our method is presented in Fig. 3. In the following sections, we introduce the training sample collection and linear adaptive metric in 3.2, and the integration of adaptive metric and differential tracking in 4.

3.2. Metric Learning Method for Tracking

Our adaptive metric learning method for tracking is supervised, and we need both positive and negative data. Once we have localized the target at a time instant, we collect the image regions of the target, extract their feature vectors, and label them as positive training samples. Image region in its very close vicinity (e.g., shifting 1 or 2 pixels) are also treated as positive samples. Nearby image regions that are not close enough (i.e., that create false positive matches) are collected and used as negative samples. This training sample collection process is applicable to every video frame. The training data collected in this manner are quite informative for discriminative learning, because they are located fairly close to the true classification boundary, and thus the learned classification is expected to be accurate. Therefore, based on the training data collected in this way, we are able to differentiate true matches of the target from false positive matches.

In the case of multiple targets, we collect a labeled data set consisting of feature vectors $\{x_1, x_2, \ldots x_n\}$ in $\mathbb{R}^m$ and their corresponding class labels $\{c_1, c_2, \ldots c_n\}$. Our method is naturally applicable for tracking multiple objects.

Our adaptive metric learning method is performed on this set of labeled training data. Following the concept in [11], we aim to find a linear transform $\mathbf{A} \in \mathbb{R}^{d\times m}$ (where $d \leq m$) that maximizes the k-NN classification accuracy in the projected space.

The k-NN classifier predicts the class label of an input data point by the consensus of its k-nearest neighbors under a given distance metric. This method is simple, but it is not analytical or differentiable with respect to $\mathbf{A}$. Fortunately, this difficulty can be alleviated by using a soft representation of the nearest neighbors. As proposed in [11], we can consider the entire transformed data set to be probabilistic ("soft") nearest neighbors. To measure the neighborhood relationship between a pair of feature vectors x_i and x_j in the training set, p_{ij} is defined to represent a *soft-max* over the Euclidean distance in the transformed space.

$$p_{ij} = \frac{\exp(-\|\mathbf{A}x_i - \mathbf{A}x_j\|^2)}{\sum_{k\neq i}\exp(-\|\mathbf{A}x_i - \mathbf{A}x_k\|^2)}. \qquad (4)$$

In other words, each feature vector x_i treats another feature vector x_j as its neighbor with probability p_{ij}, and inherits its class label from the feature vector it selects.

Denote the set of input data in the same class of i by $\mathcal{C}_i = \{j|c_i = c_j\}$. Then, our goal is to find the best transform to maximize the expected number of input data that are correctly classified:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} g(\mathbf{A}) = \arg\max_{\mathbf{A}} \sum_i \log(\sum_{j\in\mathcal{C}_i} p_{ij}). \qquad (5)$$

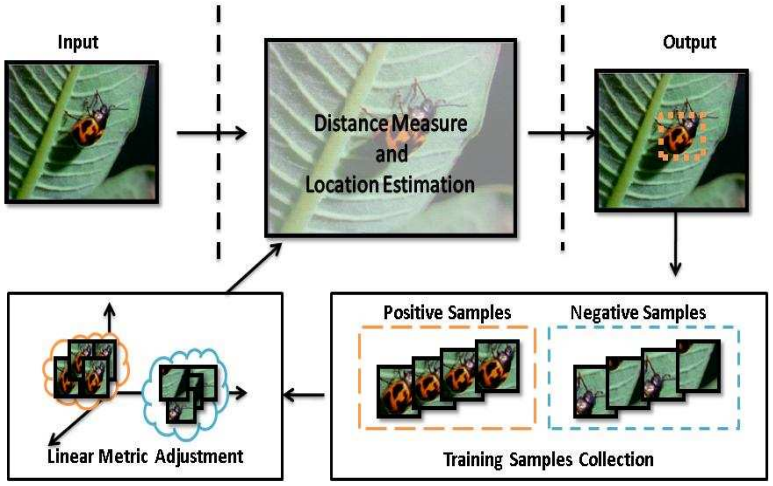

Figure 3. The approach of adaptive metric learning in differential tracking. It has three major components, including training sample collection, adaptive distance metric learning and motion analysis and target localization.

This choice of objective function is preferable as it is differentiable with respect to $\mathbf{A}$. This optimization problem can be approached by any gradient-based technique, as the gradient of the objective function can be obtained in a closed-form:

$$\frac{\partial g}{\partial \mathbf{A}} = 2\mathbf{A}\sum_i (\sum_k p_{ik} x_{ik} x_{ik}^T - \frac{\sum_{j\in\mathcal{C}_i} p_{ij} x_{ij} x_{ij}^T}{\sum_{j\in\mathcal{C}_i} p_{ij}}), \quad (6)$$

where $x_{ij} = x_i - x_j$. In our implementation, we use a conjugate gradient-descent method to iteratively maximize the objective function to obtain an optimal solution of $\mathbf{A}$.

4. Adaptive Metric Differential Tracking

In this paper, we focus on a solid case study of integrating adaptive metric and differential tracking. Differential tracking methods, such as mean-shift tracking [7] and contextual flow [17], assume small motion among two consecutive image frames. Based on a linear approximation of the two frames, differential tracking methods give a closed-form solution to the motion, and thus allowing very computational efficient algorithms for real-time tracking.

Kernel-based tracking method attracts much attention in the literature because of its computational efficiency, simplicity and robustness [5, 12, 14, 20]. In the kernel-based method, the target is represented by its feature histogram (e.g, a color histogram), $\mathbf{q} = [q_1, q_2, \cdots, q_m]^T \in \mathbb{R}^m$,

$$q_u = \frac{1}{C}\mathbf{K}(y_i - c)\delta(b(y_i), u), , \quad (7)$$

where $b(y_i)$ maps a feature at the pixel location y_i into a histogram bin u. $\mathbf{K}$ is a homogeneous spatial weighting kernel centered at c. $\delta(\cdot)$ is the delta function, and the constant C is used for normalization. Rewrite Eq.(7) in a matrix form as in [12]:

$$\mathbf{q}(\mathbf{c}) = \mathbf{U}^\mathbf{T}\mathbf{K}(\mathbf{c}), \quad (8)$$

where

$$\mathbf{U} = \begin{bmatrix} \delta(b(y_1), u_1) & \cdots & \delta(b(y_1), u_m) \\ \vdots & \vdots & \vdots \\ \delta(b(y_t), u_1) & \cdots & \delta(b(y_t), u_m) \end{bmatrix} \in \mathbb{R}^{t\times m}, \quad (9)$$

$$\mathbf{K} = \frac{1}{C}\begin{bmatrix} K(y_1 - c) \\ \vdots \\ K(y_t - c) \end{bmatrix} \in \mathbb{R}^t. \quad (10)$$

Denoting by $\mathbf{p}(\mathbf{c} + \mathbf{\Delta c})$ the target candidates' color histogram, and $D(\mathbf{q}(\mathbf{c}), \mathbf{p}(\mathbf{c} + \mathbf{\Delta c}))$ the objective function for matching, visual tracking is to estimate the best displacement $\mathbf{\Delta c}$ that optimizes the following objective function.

$$\mathbf{\Delta c}^* = \arg\min_{\mathbf{\Delta c}} D(\mathbf{q}, \mathbf{p}(\mathbf{c} + \mathbf{\Delta c})). \quad (11)$$

In this paper, we start from the Matusita metric, and notice that the proposed approach can easily be extended to other forms. When we project the original data into a new feature space by linear transform $\mathbf{A}$ obtained in Eq. 5, the distance between two data points is:

$$D(\mathbf{\Delta c}) = \|\mathbf{A}\sqrt{\mathbf{q}} - \mathbf{A}\sqrt{\mathbf{p}(\mathbf{c} + \mathbf{\Delta c})}\|^2. \quad (12)$$

This is equivalent to use a linear transform $\mathbf{A}$ to adjust the metric. We approximate $\sqrt{\mathbf{p}(\mathbf{c} + \mathbf{\Delta c})}$ linearly by using its Taylor expansion and dropping the higher order terms. The new objective function for tracking is:

$$\Delta \mathbf{c}^* = \arg\min_{\Delta c} \|\mathbf{A}\sqrt{\mathbf{q}} - \mathbf{A}\sqrt{\mathbf{p}(\mathbf{c})} - \mathbf{A}\mathbf{M}\Delta\mathbf{c}\|^2. \quad (13)$$

The optimal displacement $\mathbf{\Delta c}$ in our objective function is obtained by:

$$\mathbf{\Delta c} = (\mathbf{M}^T\mathbf{A}^T\mathbf{A}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{A}^T\mathbf{A}(\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c})}), \quad (14)$$

where $\mathbf{\Delta c} \in \mathbb{R}^r$, and

$$\mathbf{M} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{diag}(\mathbf{p}(\mathbf{c}))^{(-\frac{\mathbf{1}}{\mathbf{2}})}\mathbf{U}^{\mathbf{T}}\mathbf{J}_{\mathbf{K}}(\mathbf{c}) \in \mathbb{R}^{m\times r}, \quad (15)$$

$$\mathbf{J}_{\mathbf{K}} = \begin{bmatrix} \nabla_c K(y_1 - c) \\ \vdots \\ \nabla_c K(y_t - c) \end{bmatrix}. \quad (16)$$

Let

$$\mathbf{S} = (\mathbf{M}^T\mathbf{A}^T\mathbf{A}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{A}^T\mathbf{A}.$$

Then, we rewrite Eq.(14). The optimal displacement $\mathbf{\Delta c}$ under a new metric is obtained by:

$$\mathbf{\Delta c} = \mathbf{S}(\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c})}). \quad (17)$$

5. Experiments

This section reports our experiments that validate the effectiveness of the proposed adaptive metric methods for differential tracking. These experiments include a comparison of our method and a differential tracker without metric learning, a comparison of our method and a dynamic feature selection method, an example of tracking multiple targets, and a convergence study of the proposed metric learning methods.

In all of our experiments, the tracker is initialized through a manual selection of the image region of interest on the target. The visual appearance of the target is represented by a normalized RGB color histogram that has 225 bins. For each frame, we collect 50 training examples and they are accumulated over time. In our experiments, we evaluate the performance of the tracker for every frame. When the value of the objective function $g(\mathbf{A})$ in Eq.(5) is greater than a pre-defined threshold, we re-run metric adjustment method to obtain a new metric that adapts to the new video scene. This threshold is 10^{-5} in all of our experiments.

5.1. Effectiveness of Adaptive Distance Metric

5.1.1 Comparing with Pre-defined Metric

To validate and demonstrate the effectiveness of adaptive metric methods for differential tracking in real video, we compare our proposed method and the basic differential tracking method with pre-defined metric proposed in [12]. We select the *Sailing* sequence for this investigation for the following two reasons. First, in this testing video, the target (i.e., the sailing boat) is subject to significant illumination changes due to the sunset. Second, the target itself has a low contrast to the background and thus its color appearance is quite similar to the background. Because of the above two issues, the target can hardly be well separated from the background by pre-specified color features.

Fig. 4 shows our comparison over this testing video. The yellow rectangle in Fig.4 shows that the closest matches by the baseline method [12]. False positives are closest matches that are not the target we want to track. Fortunately, by having a supervised metric learning in tracking, our method is able to adaptively learn a better distance metric based on the supervised training data. As this step significantly enhances the discrimination power of the metric, it largely improves the tracking accuracy even only using some simple color features. As the blue rectangle shown in Fig.4, the proposed tracker smoothly tracks the sailing boat through out the low contrast and the illumination changes, while the method without adaptive metric learning loses track of the target from the beginning. This example well demonstrates the effectiveness of using adaptive metric learning in differential tracking.

By using the same *Sailing* sequence, we also compare our method with the dynamic feature selection method proposed in [6]. The candidates pool [6] is predefined as well. As the green rectangle shown in of Fig.4, this method cannot give satisfactory results for this testing sequence. As a more general case of [6], our method is more flexible, since more parameters in transformation matrix $\mathbf{A}$ are adjustable. In addition, the flexibility in our method does not incur a high computational cost, because our method achieves the optimal solution by using a more efficient gradient-based method than the exhaustive search as done in [6]. As the green and blue rectangles shown in Fig.4, the method proposed in [6] fails to track the target in *Sailing* sequence, while our method accurately tracks the target through out the entire sequence, which clearly shows the superiority of our proposed method.

To have a quantitative comparison, we obtain the ground truth of the testing sequences by manually labeling all the frames. For the *Sailing* sequence, we compare our method with [6] and [12], and show the comparison of the tracking error (in pixel) in Fig.5. This quantitative evaluation is consistent with the subjective evaluation shown in Fig.4. In other words, the method we proposed outperforms [12] and [6] in both objective and subjective evaluations.

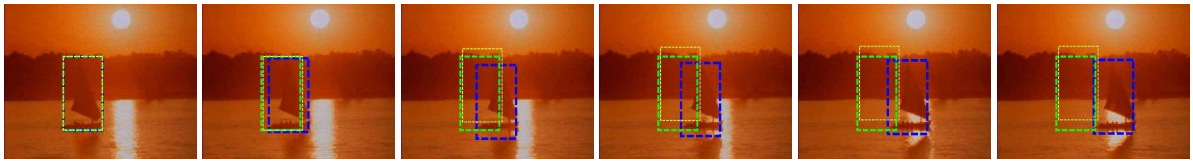

Figure 4. A comparison among the baseline method without adaptive metric learning [12](*Yellow*), dynamic feature selection method [6](*Green*) and our method(*Blue*) on the *Sailing* sequence.

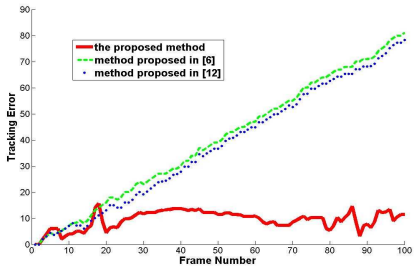

Figure 5. A comparison of tracking error of the proposed method with the methods in [6] and [12]

5.1.2 Comparing with TUDAMM

We compare our method with a very recent method called TUDAMM [16] that use a probabilistic metric learning technique in tracking. To compare the performances, we created a testing video sequence, in which the target is a paper segment and the background is a large paper with the identical texture. Tracking such a target is extremely difficult because the target is almost indistinguishable without motion, even for human eyes. We name this test sequence *Texture*, and we use a white bounding box to indicate the location of the target. The tracking results of TUDAMM are shown in red bounding boxes of Fig. 6, and our results are in blue. For this testing sequence, the key to a successful tracking of target is the quality of the metric that differentiates the subtle differences between the target and the background. TUDAMM is unable to handle this kind of sequences, because it loses track of the target right in the beginning of the sequence, and is completely off and out of the image boundary at the 19th frame. On the contrary, our method is able follow the target quite well. This is not surprising because our method can find to a more discriminative and powerful metric that separates the target from the background.

5.1.3 Other Examples

We have tested our method extensively on many other challenging sequences. This section gives the results on the testing sequences to show the effectiveness of our method on handling two major difficulties: heavy background clutters and spatial distracters. One example of cluttered background is depicted in Fig. 7. The object being tracked in this *Running* sequence is a man in blue, which is subject to a large scale change. This sequence is quite challenging, because of the heavily cluttered background. Our tracking algorithm can correctly estimate the positions and scales of the target throughout the entire sequence.

In Fig. 8, we present the tracking result on the *Zebra* sequence. This example shows how our method is able to handle spatial distracters. In this sequences, the two zebras are almost identical in their visual appearances. Thus, when we want to track one zebra, the other zebra generates a very faithful false positive match, and then distracting the tracker easily. This situation largely confronts most of the existing tracking methods. Fortunately, by obtaining to a more discriminative metric automatically, our tracking method is able to identify the nuance of the two similar objects and distinguish the target by the subtle difference from the distracter. Our method can comfortably give a promising result.

5.2. Tracking Multiple Targets

The adaptive metric learning method proposed in this paper is not limited to the 2-class case. Based on K-NN classifier, one of the main advantages of the proposed tracking method is that it can naturally handle multiple targets. We validate this benefit of our tracking method in the example of the *Racing* sequence, as shown in Fig.9.

In this testing video, we want to track three targets. These target appear to be distracters to one another. In addition, this test video has a clustered background, and the appearances of the targets have large variations due to camera-view changes. Unlike the 2-class classifier proposed in [3, 1], the k-NN classifier employed in our method can be easily extended to the multiple classes scenario. By collecting and labeling the training samples for each target, the proposed adaptive metric learning method is able to pull together the samples in the same class, and to push away all the other samples in different classes. This leads to a good separation of the classes. As shown in Fig. 9, our tracking algorithm can smoothly track all the targets. It avoids the tracker from drifting due to the distracters or appearance changes. This example clearly shows that our method is able to handle multiple targets quite well.

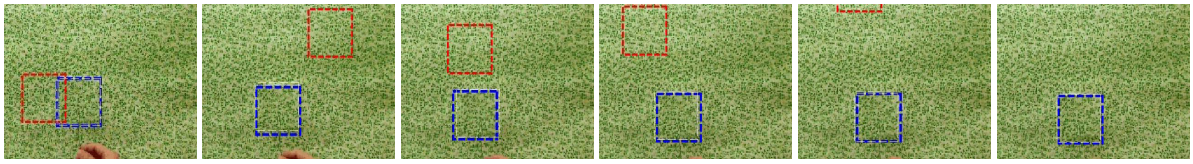

Figure 6. The comparison result of our method and TUDAMM on the *Texture* sequence. Reg rectangle represents the tracking result of TUDAMM, the blue rectangle shows the tracking result of our method, and the white rectangle indicates the ground truth.

Figure 7. The results of our method on the *Running* sequence that presents cluttered background and large scaling changes

5.3. Convergence Study of our Metric Learning

The issue of convergence is important for the gradient descent method in the proposed metric adjustment methods. Meanwhile, the convergence of the proposed adaptive metric learning method is tightly related to dimension reduction in our method. Thus, we analyze them together here.

Fig.10 gives an example of convergence and dimension reduction. The first row in Fig.10 shows the test video frames. The second row in Fig.10 presents the convergence result of the adaptive metric learning method. In the second row, the x-axis indicates the iteration number, and the y-axis represents the value of the objective function $g(\mathbf{A})$. From this study, we have three observations. First, in all the cases, our method is able to converge after a finite number of iterations. Second, our algorithm converges rapidly. In most cases, our method converges within 4 iterations. Third, the convergence is not influenced much by the optimization techniques we used.

In addition, the different color lines in Fig.10 indicate the different numbers of dimension we chose for metric learning. In our experiment, we have tested the dimensions from 3 to 30. This is indeed a large range for this parameter. From these results, we can see that the reduction of dimensionality does not actually have much influence on the speed of convergence. Thus, our proposed metric adjustment method has a nice convergence property, as it is not affected much by the dimensionality of the embedded space.

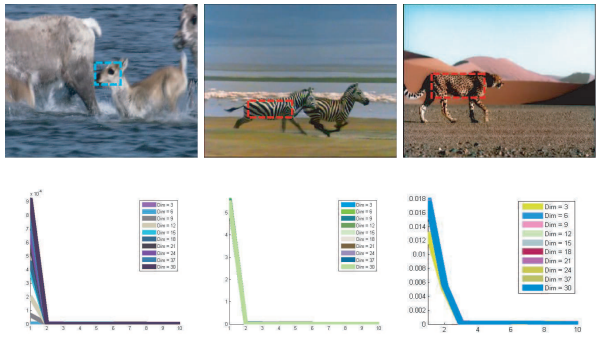

Figure 10. A study of convergence on different sequences and the analysis of dimension reduction. The first row shows the test videos, the second row presents the convergence result of the proposed metric learning method.

6. Conclusion

Matching the visual appearances of the target is the most important issue in visual tracking. Finding an optimal distance metric is critical in solving this matching problem. In this paper, we propose a new method for matching based on adaptive metric learning, and integrate this metric adjustment and differential tracking. The proposed method describes a discriminative and adaptive distance metric learning method that identifies a Mahalanobis distance metric to maximize the performance of the k-NN classifier in the projected feature space. Comparing with tracking methods using pre-defined metric, our tracking method can achieve more robust and promising tracking performance in our extensive experiments.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China (grant No. 60873127,60903096), and in part by National Science Foundation grant IIS-0347877 and IIS-0916607, and US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504.

Figure 8. The results of our method on the *Zebra* sequence that presents distracters

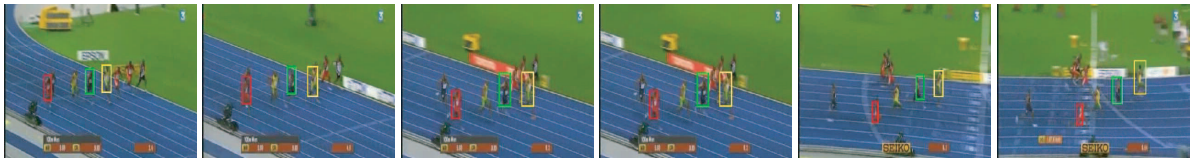

Figure 9. The results of our method on tracking multiple targets on the *Racing* sequence

References

[1] S. Avidan. Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1064 – 1072, aug. 2004.

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

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 983 – 990, 2009.

[4] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning*, pages 11–18, 2003.

[5] R. T. Collins. Mean-shift blob tracking through scale space. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Computer Society Conference on*, volume 2, pages II – 234–40 vol.2, 18-20 2003.

[6] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631 – 1643, oct. 2005.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Conference on*, volume 2, pages 142 –149 vol.2, 2000.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564 – 577, may 2003.

[9] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Computer Society Conference on*, volume 1, pages I–781 – I–788 vol.1, 18-20 2003.

[10] A. Globerson and S. Roweis. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*, 18:451–458, 2006.

[11] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighborhood component analysis. In *Advances in Neural Information Processing Systems*, 2005.

[12] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Conference on*, volume 1, pages I–790 – I–797 Vol.1, 2004.

[13] B. Han and L. Davis. Object tracking by adaptive feature extraction. In *Image Processing, International Conference on*, volume 3, pages 1501 – 1504 Vol. 3, 24-27 2004.

[14] C. Shen, J. Kim, and H. Wang. Generalized kernel-based visual tracking. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(1):119 –130, jan. 2010.

[15] P. Viola and W. M. Wells. Alignment by maximization of mutual information. *Computer Vision, IEEE International Conference on*, 0:16, 1995.

[16] X. Wang, G. Hua, and T. X. Han. Discriminative tracking by metric learning. In *European Conference on Computer Vision, 11th Conference on*, 2010.

[17] Y. Wu and J. Fan. Contextual flow. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 33 –40, 20-25 2009.

[18] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, 2002.

[19] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Conference on*, volume 1, pages 176 – 183 vol. 1, 20-25 2005.

[20] A. Yilmaz. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In *Computer Vision and Pattern Recognition. IEEE Conference on*, pages 1 –6, 17-22 2007.

[21] Z. Yin, F. Porikli, and R. T. Collins. Likelihood map fusion for visual object tracking. In *Applications of Computer Vision, IEEE Workshop on*, pages 1 –7, 7-9 2008.