Discriminative Metric Preservation for Tracking Low Resolution Targets

Nan Jiang, Heng Su, Wenyu Liu, *Member, IEEE,* Ying Wu, *Senior Member, IEEE,*

Abstract

Tracking low resolution (LR) targets is a practical yet quite challenging problem in real video analysis applications. Lacking of discriminative details in the visual appearance of the LR target leads to the matching ambiguity, which confronts most existing tracking methods. Though artificially enhancing the video resolution by super resolution (SR) techniques before analyzing might be an option, the high demanding of computational cost can hardly meet the requirements of tracking scenario. This paper presents a novel solution to track LR targets without explicitly performing SR. This new approach is based on discriminative metric preservation that preserves the data affinity structure in the high resolution (HR) feature space for effective and efficient matching of LR images. Besides, we substantialize this new approach in a solid case study of differential tracking under metric preservation, and derive a closed-form solution to motion estimation for LR video. In addition, this paper extends the basic linear metric preservation method to a more powerful nonlinear kernel metric preservation method. Such a solution to LR target tracking is discriminative, robust and efficient. Extensive experiments validate the entrustments and effectiveness of the proposed approach, and demonstrate the improved performance of the proposed method in tracking LR targets.

Index Terms

Visual tracking, metric preservation, low resolution, discriminative.

N. Jiang and W. Liu are with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, P. R. China. (e-mail: qiningonline@smail.hust.edu.cn; liuwy@mail.hust.edu.cn).

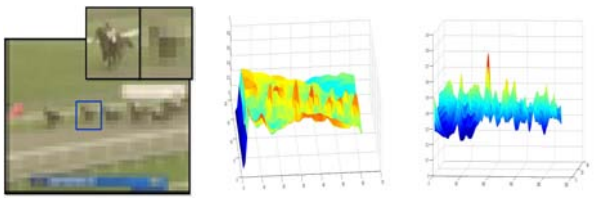
H. Su is with the Department of Automation, Tsinghua University, Beijing, 100084 China (e-mail: suh02@mails.tsinghua.edu.cn)

Y. Wu is with the Electrical Engineering and Computer Science Department, Northwestern University, Evanston, IL 60208-3118 USA (e-mail: yingwu@eecs.northwestern.edu)

I. INTRODUCTION

Many real video analysis applications have to use low resolution videos, mainly because of the practical limitations from video storage and transmission. In such videos, as shown in Fig. 1(a), the object of interest is generally imaged in tens of pixels, and the fine details of its visual appearances are lost. Since most visual features or visual primitives cannot be reliably extracted from these LR videos, it is very difficult, if not impossible, to analyze such LR targets.

This has largely confronted one of the key components in visual tracking, i.e., matching the target's visual appearances over time. For the example (i.e., the horseman) as shown Fig. 1(a), most conventional tracking approaches cannot be applied here, because it is almost impossible to extract the shape [1], contour [2], color [3] features to represent the LR target. In addition, due to the loss of visual details, the discrimination power is also lost so that the matching of LR appearances becomes quite inaccurate and sensitive to noise. For the example in Fig. 1, we directly match the LR target (depicted in the bounding box) in the LR image frames, and show the sum-of-squared-differences (SSD) matching score surface in Fig. 1(b). It is evident that this matching surface is flat and has many similar modes, implying that there are many false positive matches. Consequently, it largely plagues the matching and tracking process.



(a) (b) (c)

Fig. 1. An illustration of matching difference in LR and HR. (a)The illustration of the target of interest in both high and low resolution images. The top-left sub-image indicates the target in HR, the top-right sub-image represents the target in LR, and the bottom sub-image shows the whole image frame in LR, (b) the matching score in LR, (c) the matching score in HR.

An intuitive way to cope with this challenge is to artificially increase the resolution of such LR videos first through image super resolution (SR) techniques as a preprocessing, and then perform visual tracking over the reconstructed HR videos. However, converting LR videos to HR is a quite difficult under-constrained problem, because the SR process needs to fill in the lost information by using prior knowledge as the regularization. In general, the prior knowledge can be explicitly predefined regularization [4], or can be implicitly stipulated by training examples and obtained through learning [5], [6], [7]. Although great progress has been made, SR still remains a challenging problem itself, and most existing SR methods

are quite computationally prohibitive and time consuming.

Rather than performing SR as a preprocessing step before video analysis, it is natural and interesting to ask: ***can we track the low resolution targets without explicit super resolution?*** To the best of our knowledge, there has been very limited study on this issue. This paper presents a new attempt to answer this question, and provides a simple yet elegant and efficient solution to tracking LR targets.

We call the new idea behind the proposed solution ***metric preservation*** that preserves the data affinity structure in the HR feature space for matching LR targets. Like most learning-based SR methods, the prior knowledge we use here is implicitly conveyed through a set of training examples of LR-HR pairs. Denote a LR data vector by $\mathbf{x}_i \in \mathbb{R}^L$, and its corresponding HR feature vector by $\mathbf{y}_i \in \mathbb{R}^H$. To make the matching discriminative, the training data are supervised, i.e., we have both positive and negative examples for the target and non-target, respectively. There is a good faith assumption that the quality of matching in HR is good. This is a reasonable assumption, because the HR data have the complete information for matching. If good matching cannot be obtained even in the HR data, we do not expect it can be achieved at LR. As shown in Fig. 1(c), the HR matching surface is more discriminative than the LR one, and the matching is much less ambiguous. Then we have a good metric $\mathcal{D}_H(\mathbf{y}_i, \mathbf{y}_j)$ to measure the distances among HR feature vectors, at least we can easily specify it based on the given labeled HR training data (this is our knowledge). Going back to the LR space $\mathbb{R}^L$, any predefined metric in $\mathbb{R}^L$ in general is not likely to give good matching performance. But, if we are able to adjust the metric in $\mathbb{R}^L$, such that we can preserve the HR metric, i.e.,

$$\mathcal{D}_L^{'}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{D}_H(\mathbf{y}_i, \mathbf{y}_j),$$

where $\mathcal{D}_L^{'}(\mathbf{x}_i, \mathbf{x}_j)$ is the adjusted metric in $\mathbb{R}^L$, then we can directly perform high-quality matching in the LR space $\mathbb{R}^L$, without explicitly examining the HR space through SR.

In this paper, we propose a specific method for metric preservation by learning an optimal Mahanalobis metric which can largely preserve the affinity of the HR feature space for matching LR image patches. In addition, we integrate metric preservation with differential tracking to derive a closed-form solution to motion estimation for LR video. Due to the supervised metric preservation, the proposed method allows accurate and discriminative matching in LR data. As this method does not need explicit SR, it is computationally very efficient. Because the training data can be collected on-the-fly, the learning of metric preservation can naturally be adaptive over the video.

The novelties of the this work include the following aspects: (1) it proposes a new learning method, called metric preservation, to preserve the data affinity structure in two related feature spaces. (2) It

integrates the metric preservation mechanism into LR target tracking. This method is substantialized in a solid case study of differential tracking, and leads to a closed-form solution to motion estimation for LR target tracking. (3) It extends the basic linear metric preservation method to a more powerful nonlinear kernel metric preservation method. (4) It provides a new attempt to solve the SR problem by metric preservation. This paper is concentrated on the study of LR motion analysis that bypasses the explicit SR process. Although we include some results of SR, they just serve as the validation of our metric preservation approach, and SR is not a major focus of this paper.

The organization of this paper is as follows. After a brief introduction to the related works in Sec. II, the main methodology of metric preservation in both linear and nonlinear forms is presented in Sec III. The integration of metric preservation and differential tracking is described in Sec IV. The kernel metric preservation method and its integration to the differential tracking is introduced in Sec V. The experimental results are reported in Sec. VI followed by the conclusion.

II. Background

In LR videos, detailed visual information is missing. Normally, the target is imaged in a limited number of pixels. In addition, the target and the nearby background pixels might be mingled together. Consequently, most visual features and primitives cannot be reliably extracted to represent the objects of interest, and fail most visual tracking methods. For example, point tracking methods [8], [9] that detect and match salient points usually require high quality video frames when computing salient local features. Unfortunately, this is not viable in LR videos. Shape [10], [11] and contour [12], [13] based tracking methods basically depend on accurate localization of image edges. However, due to the mixing of the target and the background, clear image edges cannot always be expected. For segmentation-based tracking methods [14], [15], LR videos can hardly provide sufficient information for good object segmentation.

Another common issue in LR videos is the loss of fine details on the targets. Only a rough impression of the target can be obtained in LR videos. As a result, the discrimination power to separate the target from the clutter background or distracters is crippled. In the literature, there are some learning-based tracking methods that attempt to find the best decision boundary between the target and non-target [16], [17], [18], [19]. These methods works fine when the training samples have enough resolution. However, when the discrimination power of the training data is reduced due to the loss of information in LR, the effectiveness of such learning-based methods will be significantly degraded.

SR aims to produce HR images from LR inputs. A straightforward way to tackle the tracking problems in LR videos mentioned above is to enhance the resolution of the LR videos by SR techniques as the

preprocessing step. The goal of SR is to recover the missing HR details that do not exist in any given LR images. To achieve this, additional assumptions and priors have to be used. SR can be roughly divided into two categories: one is multi-image SR, and the other is single-image SR. While multi-image SR [20] focuses on generating the HR image based on a set of related LR images, single-image SR [5] synthesizes HR images based on the knowledge learnt from the training LR-HR image pairs. Solving the SR problem is in general computationally demanding. This does not make explicit SR an attractive choice for real-time visual tracking applications. However, inspired by the example-based SR methods [5], [6], [7] that learn the mapping from LR to HR, our proposed method implicitly incorporates SR in matching to track LR targets. To the best of our knowledge, there has been very limited research in the literature on this topic. This work is arguably the first of its kind to approach to this problem.

Globerson and Roweis propose a metric learning method in [21] based on an intuition that a good metric means to collapse the training data points in the same class to a single point, and to project those in different class infinitely away. By using this metric learning method, [22] proposes a method named TUDAMM, to integrate the appearance modeling and motion estimation together in an Expectation-Maximization(EM) like algorithm.

In our method, as well as [23], given the set of LR-HR training pairs, it attempts to preserve the metric in HR instead of separating the positive and negative training data apart as in [21]. Meanwhile, different from finding the essential differences in object appearance to separate the foreground from background in [22], our method focuses on the visual measurement and matching function, rather than the object representation.

We further clarify the difference between Globerson and Roweis' metric learning method in [21] and our proposed method of metric preservation. Globerson and Roweis' method attempts to enlarge the discrimination between the positive and negative data by mapping them to two discrete values. It neither preserves the structure of the data space, nor proves to have good convergence property. If this method is simply used for tracking LR targets, due to the lack of detailed visual information to discriminate the target from the false positives, the tracker is still unable to give good performance. On the contrary, our method not only preserves the affinity structure of the HR space that provides extra information for matching, but also enhances the discrimination between the target and the distracters through manipulating the HR data affinity based on the labeling information. This is the essential difference between [21] and our metric preservation method.

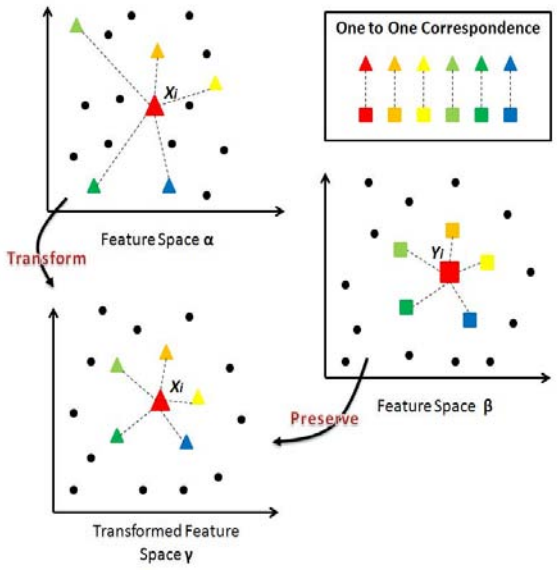


Fig. 2. The illustration of Metric Preservation.

III. Metric Preservation

A. Formulation

As a general problem, suppose we have two feature spaces α and β. $\{\mathbf{x}_i\}$ and $\{\mathbf{y_i}\}$ represent the feature vectors in feature spaces α and β, respectively. Assume we have a set of training pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, where $\mathbf{y}_i \in \beta$ is the correspondence of $\mathbf{x}_i \in \alpha$.

As illustrated in Fig. 2, for a correspondence data pair $(\mathbf{x}_i, \mathbf{y}_i)$, they may be surrounded by different data points in their own feature spaces. We seek for an optimal transformation that maps the space α to a transformed space γ, such that the metric in feature space β can be preserved in γ. This mapping can be linear or nonlinear. We call this process ***metric preservation***.

Let's start from the linear case. Without any prior knowledge, we assume the metric in feature space α to be Euclidean, i.e.,

$$\mathcal{D}_\alpha(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j).$$

When the space α is linearly mapped to γ, then the metric in γ can be represented in a Mahalanobis form, i.e.,

$$\begin{aligned}\mathcal{D}_\gamma(\mathbf{x}_i,\mathbf{x}_j) &= (\mathbf{A}\mathbf{x}_i-\mathbf{A}\mathbf{x}_j)^T(\mathbf{A}\mathbf{x}_i-\mathbf{A}\mathbf{x}_j)\\ &= (\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{S}(\mathbf{x}_i-\mathbf{x}_j),\end{aligned} \tag{1}$$

where $\mathbf{S}=\mathbf{A}^T\mathbf{A}$ has to be a positive semi-definite (PSD) matrix. We call $\mathbf{S}$ the transformation kernel.

Moreover, the transformation kernel $\mathbf{S}$ can vary for the specific feature space α and β. And the transformed feature space γ is not necessary to have the same dimension as feature space α or β. In other words, the original feature space α can be embedded into a lower dimensional feature space γ while preserving the metric in the space β.

In this paper, we treat the LR and HR data set as a specific case study of this general metric preservation problem. We let LR data space be the space α, and HR feature space be the space β. And denote a LR image patch by a vector $\mathbf{x}_i$, and its corresponding HR image patch by $\mathbf{y}_i$.

Given the data set $\{\mathbf{y}_i\}$, the structure of the space HR is largely stipulated by the affinity matrix of this data set. For example, we can use:

$$w_{ij}=\exp(-\frac{(\mathbf{y}_i-\mathbf{y}_j)^T(\mathbf{y}_i-\mathbf{y}_j)}{2\sigma_1}). \tag{2}$$

As a matter of fact, as this affinity in HR is given as the input for metric preservation, it can be arbitrarily specified by the user as long as it highlights the differences among data and exaggerates the discrimination among different classes.

To reflect the relative connectivity among data points, we perform the following normalization for the HR affinity matrix $\mathbf{P}\stackrel{\triangle}{=}[p_{ij}]$, where

$$p_{ij}=\frac{w_{ij}}{\sum_{k\neq i}w_{ik}},\qquad and\qquad p_{ii}=0, \tag{3}$$

such that p_{ij} is a distribution that represents the nearest neighbor probability from one data point to another.

Similarly, we can also construct the affinity of the LR data set $\{\mathbf{x}_i\}$, in the transformed space γ, with respect to a transformation kernel $\mathbf{S}$. This leads to a matrix $\mathbf{U}\stackrel{\triangle}{=}[u^{\mathbf{s}}_{ij}]$, where

$$u^{\mathbf{S}}_{ij}=\exp(-\frac{(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{S}(\mathbf{x}_i-\mathbf{x}_j)}{2\sigma_2}). \tag{4}$$

After normalization, we have a normalized affinity $\mathbf{Q}^{\mathbf{s}}\stackrel{\triangle}{=}[q^{\mathbf{s}}_{ij}]$, where

$$q_{ij}^{\mathbf{S}} = \frac{u_{ij}^{\mathbf{S}}}{\sum_{k \neq i} u_{ik}^{\mathbf{S}}}, \qquad and \qquad q_{ii} = 0. \tag{5}$$

Accordingly, $\mathbf{Q}^{\mathbf{S}}$ describes the affinity structure in transformed LR space γ.

Here, the affinity $\mathbf{Q}^{\mathbf{S}}$ cannot be arbitrary, as it is determined by the transformation kernel $\mathbf{S}$. Now, the metric preservation problem is formulated as the following minimization problem:

$$\mathbf{S}^* = \arg\min_{\mathbf{S}} D(\mathbf{P}, \mathbf{Q}^{\mathbf{S}}), \tag{6}$$

where $D(\cdot, \cdot)$ measures the difference between $\mathbf{P}$ and $\mathbf{Q}^{\mathbf{S}}$.

B. Linear Metric Preservation Method

To pursue a matrix $\mathbf{S}$ such that $\mathbf{Q}^{\mathbf{S}}$ is as close as possible to $\mathbf{P}$, we choose the distance function $D(\cdot, \cdot)$ in Eq. 6 to be the KL divergence:

$$\begin{aligned} \mathbf{S}^* = \arg\min_{\mathbf{S}} D(\mathbf{P}, \mathbf{Q}^{\mathbf{S}}) = \arg\min_{\mathbf{S}} \sum_{ij} \mathbf{KL}[p_{ij} | q_{ij}^{\mathbf{S}}], \\ s.t. \qquad \mathbf{S} \in PSD \end{aligned} \tag{7}$$

Denoted by $f(\mathbf{S}) \triangleq \sum_{ij} \mathbf{KL}[p_{ij} | q_{ij}^{\mathbf{S}}]$, we have:

$$f(\mathbf{S}) = \sum_{ij} p_{ij} \log p_{ij} - \sum_{ij} p_{ij} \log q_{ij}^{\mathbf{S}}. \tag{8}$$

Differentiating $f(\mathbf{S})$ with respect to the transformation kernel $\mathbf{S}$ yields a gradient rule that we can use for minimizing the objective function:

$$\nabla f(\mathbf{S}) = \frac{1}{2\sigma_2} \sum_{ij} (p_{ij} - q_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{9}$$

$$\mathbf{S}^{t+1} \leftarrow \mathbf{S}^t - \epsilon \nabla f(\mathbf{S}^t) \tag{10}$$

Since $\mathbf{S}$ has to be PSD, at each iteration, we project matrix $\mathbf{S}$ back onto the PSD cone after gradient descent. This projection is performed by ignoring the components with negative eigenvalues after the eigenvalue decomposition of matrix $\mathbf{S}$. The eigenvalue decomposition of $\mathbf{S}$ is:

$$\mathbf{S} = \sum_k \lambda_k \mathbf{u}_k \mathbf{u}_k^T, \tag{11}$$

where λ_k is the eigenvalue of $\mathbf{S}$, and $\mathbf{u}_k$ is its corresponding eigenvector. Removing the components of negative eigenvalues, we have:

$$\mathbf{S}^{'} = \sum_k \max(\lambda_k, 0) \mathbf{u}_k \mathbf{u}_k^T \tag{12}$$

$$\mathbf{S}_{t+1}^{'} \leftarrow \sum_k \max(\lambda_k, 0) \mathbf{u}_k \mathbf{u}_k^T. \tag{13}$$

We alternate the gradient descent in Eq. 10 and the PSD projection in Eq. 13 until convergence for linear metric preservation.

IV. Differential Tracking under Linear Metric Preservation

A. Motion Estimation under Linear Metric Preservation

We denote an LR video frame at time t by $I(x, y, t)$. Following the concept in [24], the velocity of a pixel is

$$\mathbf{v} = [\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}]^T = [v_x, v_y]^t. \tag{14}$$

By assuming constant brightness, we have the following optical flow constraint:

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0. \tag{15}$$

Let's consider the optical flow in a small window Ω that observes N pixels. We stack all the pixels in Ω:

$$\mathbf{I} = [I_1, I_2, \cdots, I_N]^T. \tag{16}$$

With the transformation kernel $\mathbf{S} = \mathbf{A}^T \mathbf{A}$, the objective function of solving the optical flow is:

$$\mathbf{V}^* = \arg\min_{\mathbf{v}} \|\mathbf{A}[\mathbf{I}_{t+1}(\mathbf{v}) - \mathbf{I}_t]\|^2. \tag{17}$$

We define by :

$$\mathbf{B} \triangleq \begin{bmatrix} \frac{\partial I_1}{\partial x_1} & \frac{\partial I_1}{\partial y_1} \\ \vdots & \vdots \\ \frac{\partial I_N}{\partial x_N} & \frac{\partial I_N}{\partial y_N} \end{bmatrix}_{N\times 2} \tag{18}$$

$$\mathbf{b} \triangleq -[\frac{\partial I_1}{\partial t}, \cdots, \frac{\partial I_N}{\partial t}]^T_{N\times 1}. \tag{19}$$

The necessary condition of solving Eq. 17 is given by

$$\mathbf{B}^T\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{v} = \mathbf{B}^T\mathbf{A}^T\mathbf{A}\mathbf{b}. \tag{20}$$

Then, the flow under metric preservation is obtained by:

$$\mathbf{v} = (\mathbf{B}^T\mathbf{S}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}\mathbf{b}. \tag{21}$$

The accurate matching can be found by performing a linear search along the computed motion direction. This is straightforward, because we can simply compute the matching between two LR patches $\mathbf{x}_i$ and $\mathbf{x}_j$ without explicitly performing SR:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{S}(\mathbf{x}_i - \mathbf{x}_j).$$

This is equal to measuring the distance in a transformed feature space γ that preserves the metric of the HR feature space. By doing so, a large amount of computation is saved, while keeping the same performance as if it were done in the HR space.

B. Algorithm Overview

Figure 3 gives an overview of metric preservation for tracking LR targets in our proposed approach. Our approach has three major components:

- *Training samples collection*. The off-line process collects supervised training samples in both LR and its corresponding HR video frames. We have two classes of data sets, positive and negative ones. We collect small image patches in the target region as positive training samples, and those on the background as negative ones, as shown at the top-left in Fig. 3.
- *Measure adjustment for metric preservation*. Given the training samples(top-right in Fig. 3), we compute the data affinity of the HR training samples, and learn an optimal transformation (bottom-left of Fig. 3). It projects the LR training samples to a new feature space, where the metric in the HR space is largely preserved, as described in Sec. III.

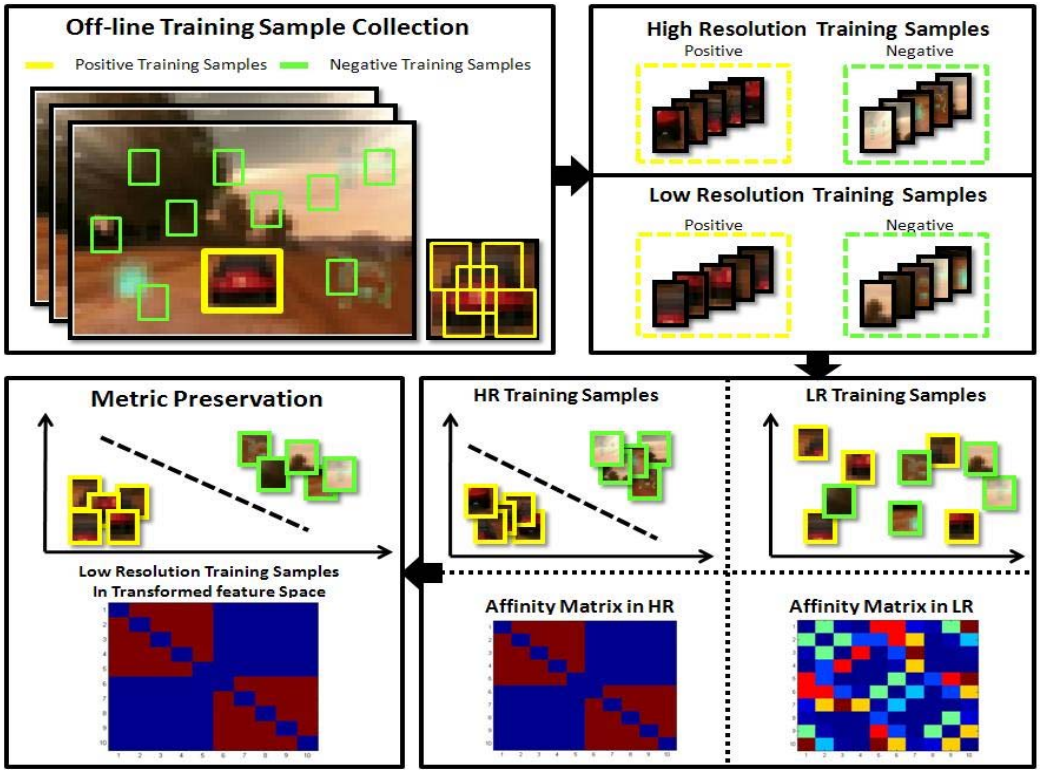


Fig. 3. The illustration of metric preservation for low resolution visual tracking.

- *Motion estimation and target localization*. Metric preservation helps preserve the affinity in the HR feature space, and tackle the LR tracking problem without explicitly performing SR. We employ the learned transformation matrix to estimate the motion and localize the target, as described in Sec. IV-A.

V. Differential Tracking under Kernel Metric Preservation

A. Kernel Metric Preservation Method

For those situations that cannot be handled by the linear metric preservation method, we employ the *kernel trick* to generalize the linear method to a nonlinear metric preservation method. Although in practice the linear metric preservation method generally works well in many cases, there exist cases where such a linear method is confronted. In these cases, no matter what linear projections we use, the projected data can hardly preserve the affinity of HR in transformed LR feature space. Therefore, we cannot find a plausible linear projection in such cases. We demonstrate the effectiveness for using kernel metric learning in Fig. 4, where the left column shows the data distribution, and the right column is the

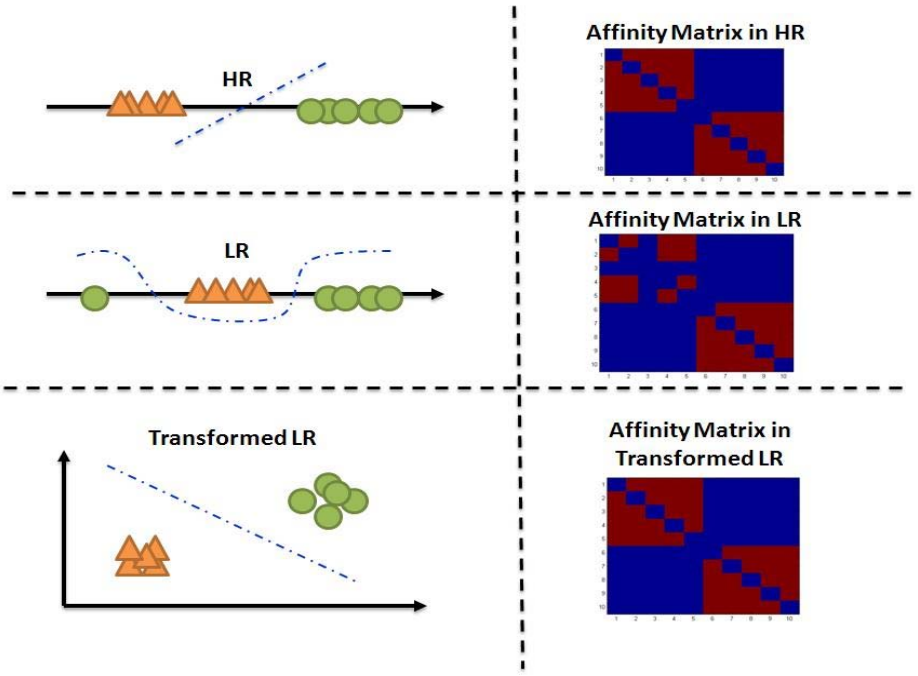


Fig. 4. The demonstration of the effectiveness of kernel metric preservation, where the left column represents the data distribution, and the right column represents the corresponding affinity matrix. The green dots represent the positive training samples, and the orange triangles represent the negative training samples.

corresponding affinity matrix. As shown in Fig. 4, the positive and negative training data in LR cannot be linearly separated as they are in HR. To preserve the HR affinity in LR, a feasible method is to project those training data into a higher dimension based on a kernel function.

To tackle these nonlinear cases in practice, we design the ***kernel metric preservation method***. Kernel metric preservation approaches the problem by mapping the original LR data $\mathbf{x}_i$ to a higher dimensional or even an infinite dimensional feature space, and aiming to find a linear metric in the new space to achieve our objective function. This learned kernel metric is more powerful to preserve the affinity. In addition, since the mapping can be quite general, e.g., not necessary to be linear, the kernel metric preservation method can be quite general as well.

In order to learn a Mahalanobis distance metric in a high, possibly infinite, dimensional feature space, by a nonlinear mapping ϕ. We restrict our analysis to nonlinear maps ϕ for which there exists a kernel function $\mathbf{k}$ that can be used to compute the inner product without carrying out the nonlinear mapping explicitly, such that $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

Let $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)]^T \in \mathbb{R}^{n \times l}$, where n is the number of data points in the training set, l

is the dimension of the projected feature space, and l might be infinite. We consider the parametrization of $\mathbf{A}$ in the form $\mathbf{A} = \Lambda\Phi$, where $\mathbf{A} \in \mathbb{R}^{d\times l}$ is a linear combination of feature vectors in high(or infinite) dimensional feature space. Define $\mathbf{k}_i = \Phi\phi(\mathbf{x}_i) = [\mathbf{k}(\mathbf{x}_1, \mathbf{x}_i), \ldots, \mathbf{k}(\mathbf{x}_n, \mathbf{x}_i)]^T$, and $\mathbf{k}_{ij} = \mathbf{k}_i - \mathbf{k}_j$. Substituting input vector $\mathbf{x}_i$ by its non-linear mapping $\phi(\mathbf{x}_i)$ in Eq.1, the Mahalanobis distance in the nonlinearly transformed space is:

$$u^{\phi}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}_{ij}^T \Lambda^T \Lambda \mathbf{k}_{ij}. \tag{22}$$

This means that we do not need to explicitly identify the non-linear transform $\phi(\mathbf{x})$ before computing u^{ϕ}. Even if $\phi(\mathbf{x})$ may project the original data into an infinite dimensional space, as $\Lambda_{d\times n}$ is finite, it is still feasible to compute u^{ϕ}.

By using the Mahalanobis distance in the nonlinear kernel space, similarly, we define $\mathbf{L} = \Lambda^T\Lambda$, and rewrite Eq.5

$$q_{ij}^{\mathbf{L}} = \frac{\exp(-\mathbf{k}_{ij}^T \Lambda^T \Lambda \mathbf{k}_{ij})}{\sum_{k\neq i} \exp(-\mathbf{k}_{ik}^T \Lambda^T \Lambda \mathbf{k}_{ik})}. \tag{23}$$

Considering the notation in Eq.8, we have:

$$\begin{aligned} f(\mathbf{L}) &= \sum_{ij} p_{ij} \log p_{ij} - \sum_{ij} p_{ij} \log q_{ij}^{\mathbf{L}} \\ &= \sum_{ij} p_{ij} \log p_{ij} \\ &- \sum_{ij} p_{ij} \log \frac{\exp(-\mathbf{k}_{ij}^T \mathbf{L} \mathbf{k}_{ij})}{\sum_{k\neq i} \exp(-\mathbf{k}_{ik}^T \mathbf{L} \mathbf{k}_{ik})}. \end{aligned} \tag{24}$$

The gradient of this objective function is:

$$\nabla f(\mathbf{L}) = \frac{1}{2\sigma} \sum_{ij} (p_{ij} - q_{ij}) \mathbf{k}_{ij} \mathbf{k}_{ij}^T. \tag{25}$$

$$\mathbf{L}^{t+1} \leftarrow \mathbf{L}^t - \epsilon \nabla f(\mathbf{L}^t). \tag{26}$$

The same as in the linear metric preservation method, PSD projection is needed in this kernel method as well. The eigenvalue decomposition of $\mathbf{L}$ is:

$$\mathbf{L} = \sum_k \delta_k \mathbf{v}_k \mathbf{v}_k^T, \tag{27}$$

where δ_k is the eigenvalue of $\mathbf{L}$, and $\mathbf{v}_k$ is the eigenvector of $\mathbf{L}$.

Discarding the components of negative eigenvalues we have:

$$\mathbf{L} = \sum_k \max(\delta_k, 0)\mathbf{v}_k\mathbf{v}_k^T. \tag{28}$$

$$\mathbf{L}_{t+1}^{'} \leftarrow \sum_k \max(\lambda_k, 0)\mathbf{v}_k\mathbf{v}_k^T. \tag{29}$$

We alternate the gradient descent in Eq. 26 and the PSD projection in Eq. 29 until convergence for kernel metric preservation.

B. Motion Estimation under Kernel Metric Preservation

Following the notations in Eq. 17, under the kernel metric preservation, the objective function of motion estimation is:

$$\begin{aligned}
\mathbf{V}^* &= \arg\min_{\mathbf{v}} \|\mathbf{A}[\phi(\mathbf{I}_{t+1}(\mathbf{v})) - \phi(\mathbf{I}_t)]\|^2 \\
&= \arg\min_{\mathbf{v}} \|\Lambda[\mathbf{k}(\mathbf{x}_1, \mathbf{I}_{t+1}(\mathbf{v})), \cdots \mathbf{k}(\mathbf{x}_n, \mathbf{I}_{t+1}(\mathbf{v}))]^T \\
&- \Lambda[\mathbf{k}(\mathbf{x}_1, \mathbf{I}_t), \cdots \mathbf{k}(\mathbf{x}_n, \mathbf{I}_t)]^T\|^2.
\end{aligned} \tag{30}$$

According to the Taylor expansion, we have:

$$\mathbf{k}(\mathbf{x}_i, \mathbf{I}_{t+1}(\mathbf{v})) = \mathbf{k}(\mathbf{x}_i, \mathbf{I}_t) + \mathbf{k}^{'}(\mathbf{x}_i, \mathbf{I}_t)(\mathbf{B}\mathbf{v} - \mathbf{b}). \tag{31}$$

Suppose we employ the RBF kernel here,we have:

$$\mathbf{k}^{'}(\mathbf{x}_i, \mathbf{I}_t) = \frac{1}{\sqrt{2\pi}\sigma^3} exp(-\frac{\|x_i - \mathbf{I}_t\|^2}{2\sigma^2})[\mathbf{x}_i - \mathbf{I}_t]^T. \tag{32}$$

Denote by:

$$\mathbf{F} = \begin{bmatrix} \mathbf{k}^{'}(\mathbf{x}_1, \mathbf{I}_t) \\ \mathbf{k}^{'}(\mathbf{x}_2, \mathbf{I}_t) \\ \vdots \\ \mathbf{k}^{'}(\mathbf{x}_n, \mathbf{I}_t) \end{bmatrix}. \tag{33}$$

By solving Eq. 30, we have:

$$\mathbf{B}^T\mathbf{F}^T\mathbf{L}\mathbf{F}\mathbf{B}\mathbf{v} = \mathbf{B}^T\mathbf{F}^T\mathbf{L}\mathbf{F}\mathbf{b}. \tag{34}$$

Then, the motion under kernel metric preservation is obtained by:

$$\mathbf{v} = (\mathbf{B}^T\mathbf{F}^T\mathbf{L}\mathbf{F}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{F}^T\mathbf{L}\mathbf{F}\mathbf{b}. \tag{35}$$

This motion information can be easily used to located the target in the LR video.

VI. Experiments

This section reports our experiment results that validate the effectiveness of the proposed metric preservation method for tracking LR targets. The size of videos we used for experiments is 60×80 pixels on average. Normally, the target sizes are less than 50 pixels in total.

The experiments in this section are grouped into 4 parts. We explain the training process and convergence study of the training method firstly. After that, we explicitly demonstrate the effectiveness of metric preservation in both matching and tracking. The tracking results under linear metric preservation are shown in the third part. And then the tracking results of kernel method are given in the fourth part.

In all our experiments, the tracker is initialized through manually selecting of the image region of interest on the target in the first frame. The gradient descent parameter ϵ is set to be 0.1 in all the experiments.

To have a fair comparison, our experiments are focused on the evaluation of metric preservation, rather than resorting to the efforts of tracking by detection methods [25]. In addition, although template updating can be easily achieved and integrated in our method by employing the algorithm proposed in [26], it is not a major point of this paper. So we do not report the experiments of combining template updating in our tracking approach.

A. The training Process and the Convergence Study of Metric Preservation

1) Training Process: Our training data is a set of labeled LR-HR patch pairs. Some of the image patches are from the target, and the others are from the background. LR patches (3×3 pixels) are represented by LR features and HR patches (9×9 pixels) by HR features. We aim to find a transformed LR space such that its affinity structure is the closest to that in the HR space.

We have off-line collected a training video database, including 200 LR video clips, and their corresponding HR ones. Based on this database, we extracted 100,000 image patch pairs, and obtained 4,000

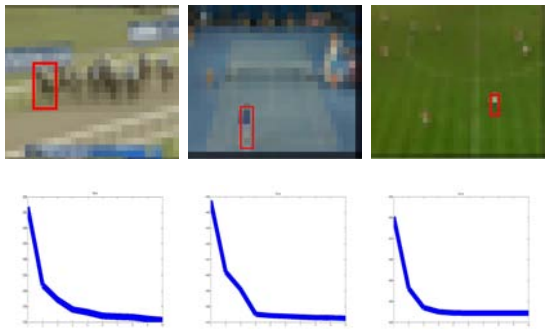


Fig. 5. A demonstration of the rate of convergence on different sequences. The first row shows the test videos, and second row represents the convergency result of metric preservation.

representative ones by K-means clustering in the HR space. Using these 4,000 training data, we perform metric preservation. This large data set is representative enough. Testing data (i.e., video to perform LR tracking) are from other 20 different video clips. It is clear that once the metric is learned, tracking is simply performed on the available LR video (and the corresponding HR video is neither available, nor needed).

Besides this off-line procedure, we also have an alternative on-line scheme. Although it is not practical to perform super-resolution on every single LR frames before tracking, for a given video, it is viable to do it for the initial several frames (e.g., 5-10 frames). In this way, we can collect both LR-HR pairs for training. Then the rest of the video frames are used for testing. Different from our off-line scheme that aims to find a generic metric, this learnt specific metric is only good for the specific video clip where the training data are collected.

2) The Convergence Study of Metric Preservation: Convergence is an important issue in the gradient descent procedure in metric preservation. Fig. 5 gives examples of convergence on different LR videos. The first row in Fig. 5 indicates the test video frames. The second row gives the convergence study of the metric preservation method, where the x-axis represents the number of iterations, and the y-axis represents the output of Eq. 8. From this study, we have two observations. First, in all the cases, our method is able to converge after a finite number of iterations. Second, our method converges rapidly. Normally, our method converges within 4-6 iterations.

(a) (b)

Fig. 6. A demonstration shows the effectiveness of metric preservation. (a)Target of interest for matching, (b)first row indicates the best four matches without metric preservation, second row shows the best four matches after metric preservation.

B. The Effectiveness of Metric Preservation

1) The Effectiveness of Metric Preservation on matching: Matching is one of the critical components for visual tracking. In order to clearly demonstrate the effectiveness of metric preservation, in Fig. 6, we present the matching results with and without metric preservation. Fig. 6(a) shows the target of interest at time t. In this video, there are several similar objects in the close vicinity of the target of interest. The first row in Fig. 6(b) shows the best four matches in this video frame without metric preservation at time $t+1$. We represent the matching without metric preservation by:

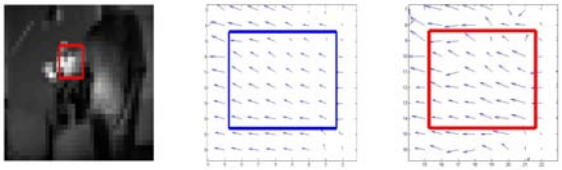
$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j).$$

By introducing transformation kernel $\mathbf{S}$, the second row in Fig. 6(b) shows the best four matches with metric preservation at time $t+1$. We represent the matching with metric preservation by:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{S} (\mathbf{x}_i - \mathbf{x}_j).$$

It's clear in Fig. 6(b) that with the help of metric preservation, the best matching are all located very close to the true location, while the matching without metric preservation are all false positives. This example well demonstrates the effectiveness of metric preservation in matching.

2) A Demonstration of Motion Field: To further justify the effectiveness of metric preservation in motion estimation, we provide a demonstration of estimating the entire motion field using the *Toy* sequence. The target of interest is shown in Fig. 7(a), and the estimated motion field with metric



(a) (b) (c)

Fig. 7. A demonstration of motion estimation under metric preservation. (a)Target of interest, (b)Motion estimation with metric preservation, (c) Motion estimation without metric preservation.

presentation is shown in Fig. 7(b), and that without metric preservation in Fig. 7(c). It is obvious that the method with metric preservation reports a much more accurate estimation of the motion field for LR video.

3) A Demonstration of Super Resolution: Although our tracking method eliminates the needs of explicit SR, the proposed metric preservation approach is able to perform SR. To validate metric preservation, it is helpful to check the synthesized HR so as to evaluate the quality of the learned metric.

Given a set of HR training vectors $\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$, and its corresponding LR training vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, where n is the total number of training samples. Based on these training data, our method learns a transformed feature space parameterized by matrix $\mathbf{A}$. In this transformed feature space, the affinity in HR have been kept for LR.

In SR reconstruction, for each given LR image patch $\mathbf{p}$, we find its k nearest neighbors in the transformed feature space (determined by the transform matrix $\mathbf{A}$). We denote them by $\mathcal{N} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, and collect them into a data matrix $\mathbf{N} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_k]$. Now, we want to find the best set of weights that give the best local linear reconstruction of $\mathbf{p}$ based on these nearest neighbors. Then the objective function for local linear reconstruction is:

$$\begin{aligned} \mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{p} - \sum_{j \in N} w_j \mathbf{x}_j\|^2 \\ s.t. \qquad w_{ij} = 0 \qquad if\ \mathbf{x}_j \notin \mathcal{N} \\ and\ \sum_j w_j = 1, \end{aligned} \tag{36}$$

where $\mathbf{w} = [w_1, \ldots, w_n]^T$.

To solve this constrained least-squares fitting problem, for all $\mathbf{x}_t \in \mathcal{N}$, we introduce a local covariance matrix $\mathbf{C}$,

$$\mathbf{C}_{ik} = (\mathbf{p} - \mathbf{x}_i)^T(\mathbf{p} - \mathbf{x}_k). \tag{37}$$

We can rewrite the reconstruction error for $\mathbf{p}$:

$$e(\mathbf{w}) = \|\mathbf{p} - \mathbf{N}\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{C}\mathbf{w}. \tag{38}$$

By constructing the Lagrangian, we have:

$$\boldsymbol{L}(\mathbf{w}, \lambda) = \mathbf{w}^T\mathbf{C}\mathbf{w} + \lambda(\mathbf{1}^T\mathbf{w} - 1), \tag{39}$$

where $\mathbf{1}$ is an all-1 column vector. It is easy to derive that:

$$\mathbf{w} = \frac{\mathbf{C}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}. \tag{40}$$

Once this set of LR reconstruction weights have been obtained, we use them for SR. First we find the corresponding HR patches $\{\mathbf{y}'_1, \mathbf{y}'_2, \ldots, \mathbf{y}'_k\}$ of these nearest neighbor LR patches identified before. Then the recovery of the HR patch is the linear interpolation of these HR patches on the same weights obtained in Eq. 40.

The following experiment demonstrates the local linear reconstruction results for SR under metric preservation. In this experiment, for an LR input image (Fig. 8(a)), we want to increase the resolution only for the foreground (i.e., the head of the moose), while blurring the background. This is feasible as we can manipulate the training data. After collecting a set of supervised LR-HR training pairs (positive pairs for the foreground and negative ones for the background), we replace the HR image patches in the background class by homogeneous patches at the average intensity of these HR patches. This in turn changes the local neighborhood relationship and the affinity in the HR space. When we have learnt the transformation kernel to preserve this HR affinity, for any LR patch, we find its local neighbors in the transformed space. Then a linear combination of their corresponding HR patches is the output of this LR patch. Fig. 8(c) shows the output of the entire image. Comparing with the HR image in Fig. 8(b), it is evident that the resolution of the head of the moose has been significantly increased, while the rest of the image is largely blurred, as expected.

C. Linear Metric Preservation for Tracking

1) Comparing with Method without Metric Preservation: To validate and demonstrate the effectiveness of metric preservation for tracking LR targets, we compare our proposed method with a differential

(a) (b) (c)

Fig. 8. The super resolution result of the foreground part. (a) the low resolution image, (b) the high resolution image, (c) the super resolution result of foreground part only.

tracker without metric preservation. We select the *Car* sequence for this comparison for the following three reasons. First, the target is quite small in the whole video. There are merely 35 pixels to describe the target of interest. Shape features and corners can hardly work well on this video. Second, the target shares the similar appearance as the background, and it has a low contrast to the gray road. Third, those cars that are passing by the target are strong distracters to the tracker. Due to the above three reasons, the target cannot be easily separated from the background and be tracked continuously.

Fig. 9 shows our comparison on this testing video. The first row shows the results of the proposed method, and the second row shows the results of the baseline method (SSD template matching). It's clear that, by having the metric preservation in LR tracking, our method can adaptively select a better metric to separate the target from the background. As the metric adjustment significantly enhances the discrimination power, it largely improves the tracking accuracy. Even when there are distracters, as shown on the left in Fig. 9, the proposed method can smoothly track the target without jittering. This example well demonstrates the effectiveness of using metric preservation in LR visual tracking.

We also give a quantitative comparison. For the *Car* sequence, we obtain the ground truth of this sequence by manually labeling all the video frames. The comparison on tracking error (in pixels) is shown in Fig. 10. It is consistent with the subjective evaluation as shown in Fig. 9. It is clear that the proposed method outperforms the baseline.

2) Comparison between Linear Metric Preservation and Discriminative-based Feature Selection Method: To better demonstrate the effectiveness of metric preservation for tracking, we compare our algorithm with discriminative-based feature selection [27] on tracking LR object. The subjective tracking results are shown in Fig.11. In the *Parade* sequence, the target of interest shares the same appearance as the background objects which makes it very challenging for tacking, especially in LR video. It's evident

Fig. 9. A comparison of the tracking result with metric preservation(top) and without metric preservation(bottom) on the *Car* sequence.

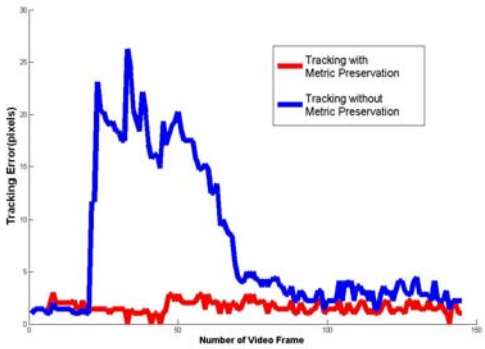


Fig. 10. A quantitative comparison between methods with and without metric preservation on the *Car* sequence.

that the discriminative-based feature selection method [27] fails to locate the target while our method can keep outputting precise results as shown in Fig.11. The quantitative comparison is shown in Fig.12, which is consistent with the subjective results. Both results show the superiority of our method over the discriminative-based feature selection method [27].

3) Comparison between Linear Metric Preservation and Discriminative Metric Learning: One of the recent publications most related to our work is TUDAMM [22]. It is largely based on Globerson and Roweis' method [21]. The comparisons between our method with TUDAMM are shown in Fig. 13 and Fig. 14. Globerson and Roweis' method aims to enlarge the discrimination between positive and negative data by mapping them to two discrete values. This method neither preserves the structure of the data space, nor proves to have good convergence property. The results of directly employing TUDAMM for LR tracking are shown in the bottom row in Fig. 13. Due to the lack of detailed visual information to

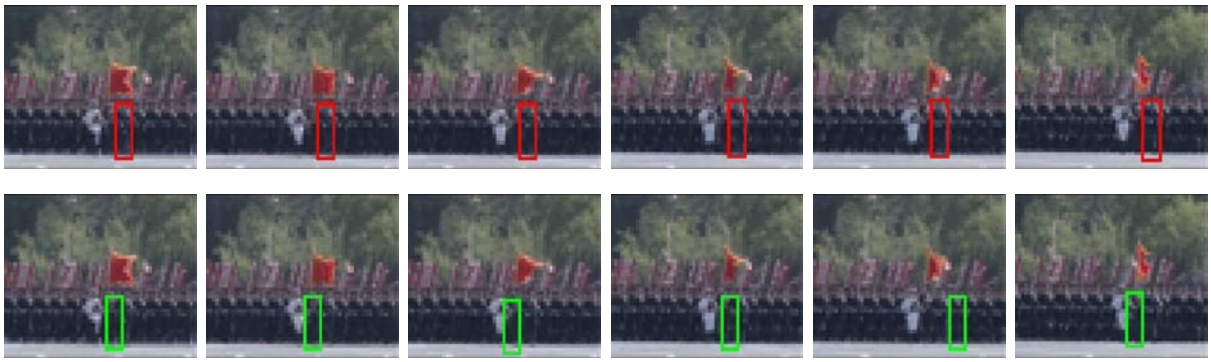


Fig. 11. Comparison between metric preservation and discriminative-based feature selection [27] on the *Parade* sequence.

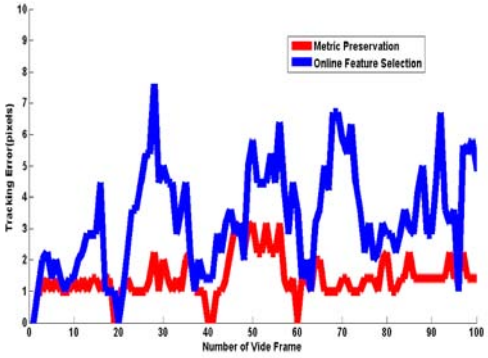


Fig. 12. A quantitative comparison between metric preservation and discriminative-based feature selection [27] on the *Parade* sequence.

discriminate the target and the false positives, TUDAMM is still unable to give good performance. On the contrary, metric preservation tries to preserve the affinity structure of the HR space which provides extra information for matching, and enhances the discrimination between the target and the distracters. The tracking results under metric preservation are shown in the top row in Fig. 13. Moreover, we give the quantitative comparison between metric preservation and TUDAMM in Fig. 14. Comparing with the method that applies Globerson and Roweis' method directly on the LR videos, both objective and subjective comparisons demonstrate the superiority of our method.

4) Other Tracking Results of Linear Metric Preservation: We have extensively tested our method on many other challenging sequences. The videos we tested are all in LR. The sizes of the targets are within tens of pixels in total. Good features can hardly be extracted from these videos, which makes it difficult

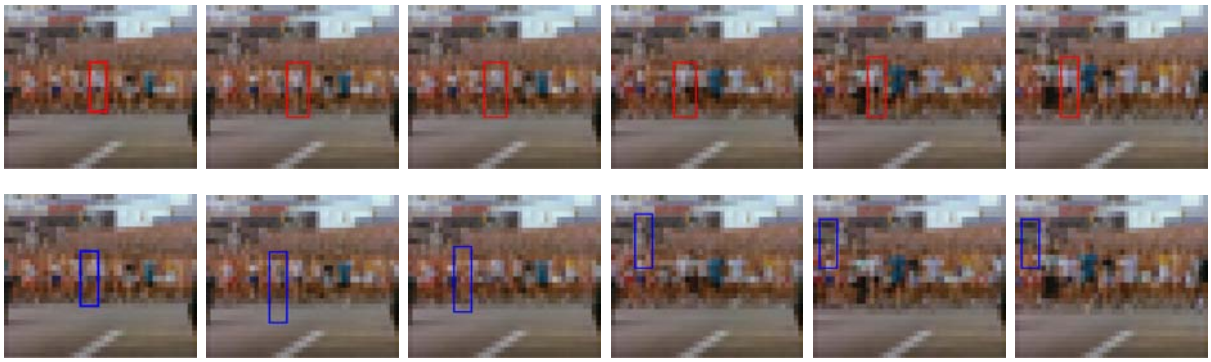


Fig. 13. A comparison of the tracking result between metric preservation(top) and TUDAMM [22](bottom) on the *Runner* sequence.

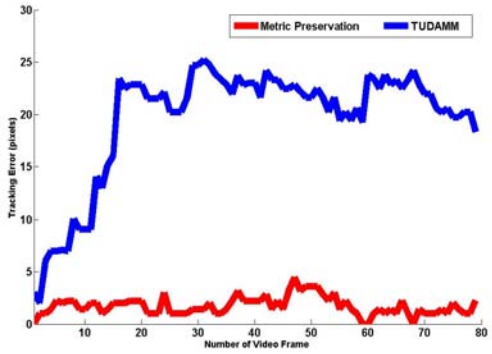


Fig. 14. A quantitative comparison between metric preservation and TUDAMM [22] on the *Runner* sequence.

to track the target accurately and consistently.

In Fig. 15, we present the tracking results on the *Tennis* sequence. This example shows how our method is able to handle the unstable observations. In this sequence, the target movement causes large appearance changes. Thanks to metric preservation, we can perform matching in the transformed feature space. As shown in Fig. 15, our method can comfortably output accurate tracking results.

An example of handling the rapid movement of target is shown in Fig. 16 on the *Badminton* sequence. The target in red uniform is moving fast in this video. Though the target only has tens of pixels, our method can correctly estimate the target position throughout the entire sequence.

Another example exhibiting spatial distracters is given in the *Horseracing* sequence in Fig. 17. As we mentioned before, in this sequence, there are several horses sharing almost identical visual appearance. Thus, when we want to track the target horse, the other horses are very likely to be the false positive

Fig. 15. An example on the *tennis* sequence using linear metric preservation.

Fig. 16. An example on the *Badminton* sequence using linear metric preservation.

matches, as illustrated in the first row in Fig. 6(b). Fortunately, preserving the metric in HR in LR feature space helps us discriminate the target from the distracters as shown in the second row in Fig. 6(b). By adjusting the metric in LR, our method is able to preserve the subtle difference in HR, and discriminate different objects in LR. As shown in Fig. 17, our method can track the target of interest robustly.

5) Extension of Basic Linear Metric Preservation for Tracking - Multiple Objects: The proposed method can easily handle the multiple objects tracking problem by sightly modifying the basic single tracking algorithm. As shown in Fig. 18, we want to track three cars, each of which is a distracter of the other two. Besides that, each car only has about 30 pixels. Moreover, the cars do not show good contrast to the background, which makes the tracking problem even harder. However, with the help of metric preservation, extra information leant from the LR-HR training data is used in LR tracking. It improves the discrimination power for matching, and enhances the tracking performance. The tracker under linear metric preservation comfortably tackles the multiple objects tracking problem as shown in Fig. 18.

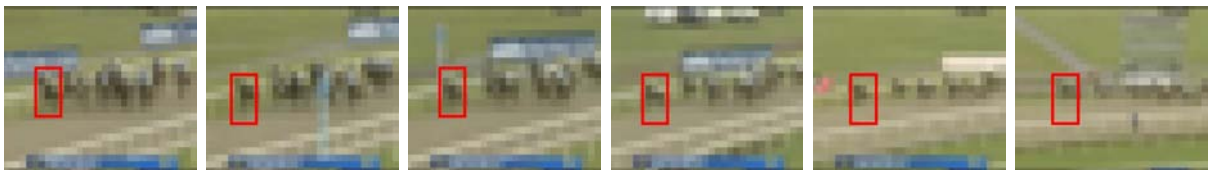


Fig. 17. An example on the *horseracing* sequence using linear metric preservation.

Fig. 18. An example on the *MultipleCar* sequence using linear metric preservation.

D. Kernel Metric Preservation for Tracking

1) Comparison between Linear and Kernel Metric Preservation: Although the proposed linear metric preservation method works well for most of our testing videos, it is confronted by the situation where the positive and negative data are not linearly separable. The proposed kernel metric preservation method aims to handle this difficult situation.

Fig. 19 gives a subjective comparison on the *Sprite* sequence. In this sequence, the target is moving in front of a clutter background. The tracker can be easily distracted by the similar appearance in the background. The top row in Fig. 19 shows the tracking results of the linear metric preservation method, and the bottom row shows the tracking results of the kernel method. We can observe that the linear metric preservation method does not work very well on this *Sprite* sequence, because the positive and negative data are heavily mingled together. In contrast, as shown in Fig.19, the proposed kernel method is able to cope with this difficulty very well. The kernel metric preservation method tracks the object successfully throughout the whole sequence.

The quantitative comparison of tracking error (in pixel) between the kernel method and linear method is given in Fig.20. Both evaluations show that the proposed kernel method works better than the linear metric preservation method in such a challenging case.

2) A Convergence Study of Kernel Metric Preservation: In Fig. 21, we give examples of convergence on different LR videos under kernel metric preservation. The first row in Fig. 21 indicates the test video frames. The second row gives the convergence study of the metric preservation method, where the x-axis represents the number of iterations, and the y-axis represents the output of Eq. 24. For kernel metric preservation, the testing samples need to perform convolution. Considering the large amount of training samples, without losing generality, we further choose 20 representative clustering centers in kernel metric preservation. From Fig. 21, it's obvious that, in all the cases, our kernel method is able to converge after 8-10 iterations.

3) Other Tracking Results of Kernel Metric Preservation: To demonstrate the tracking effectiveness using kernel metric preservation, we test the kernel method on various videos, as shown in Fig. 22, 23, and 24. As we mentioned earlier, in LR videos, the target does not have sharp edges. Pixels on the target boundary are usually mingled with the nearby background pixels, which makes the LR tracking problem challenging. Besides that, the unstable visual appearance makes tracking even harder.

The gray car in Fig. 22 shares the similar appearance with the background. By employing the kernel metric preservation method, the discriminative information is integrated into the motion estimation as described in V-B. Moreover, although the target undergoes large appearance changes over time, as shown in Fig. 22, the kernel method can track the target persistently. Another example is shown in Fig. 23.

Distracters nearby are likely to fail the tracker. As shown in Fig. 24, there is a distracter that produces a false positive match around the target. The tracker may fail because of the inaccurate matching in the linear method. Yet the kernel method, giving more accurate matching, reliably estimates target location throughout the entire sequence.

VII. Conclusion

In this paper, we propose a visual tracking method for LR video based on metric preservation. Because of metric preservation, we no longer need to explicitly perform SR as a preprocessing step before analyzing the input LR video. Thus our tracking method is computationally efficient. Moreover, we obtain a closed form solution to motion estimation under both linear and kernel metric preservation. The theoretical analysis and experimental results show that the proposed method is robust and accurate.

VIII. 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.

References

[1] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1262 –1273, Aug. 2006.

[2] Y. Chen, Y. Rui, and T. Huang, "Multicue hmm-ukf for real-time contour tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1525 –1529, Sept. 2006.

[3] Y. Wu and T. Huang, "Color tracking by transductive learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 133 –138 vol.1, 2000.

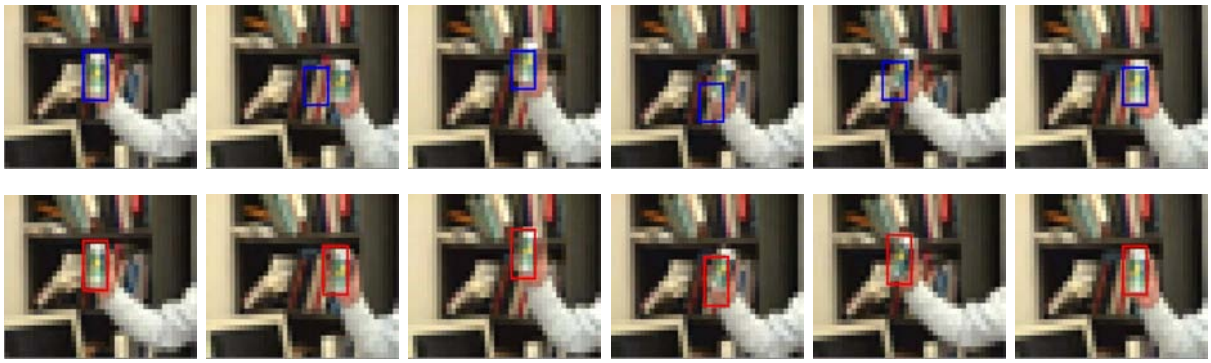


Fig. 19. A comparison of the tracking results between linear (top) and kernel(bottom) metric preservation on the *Sprite* sequence.

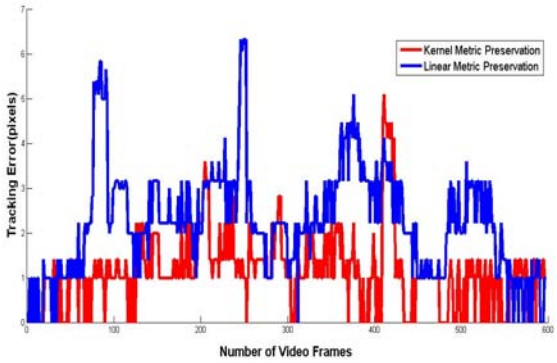


Fig. 20. A quantitative comparison between linear and kernel metric preservation on the *Sprite* sequence.

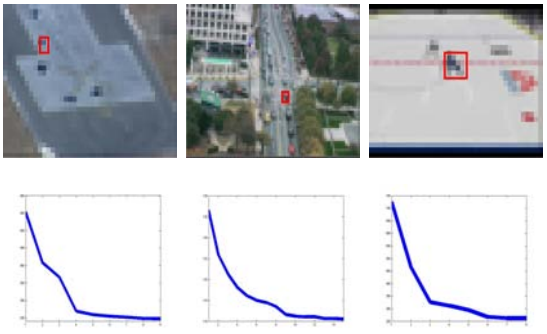


Fig. 21. A demonstration of the rate of convergence on different sequences. The first row shows the test videos, and second row represents the convergency result of kernel metric preservation.

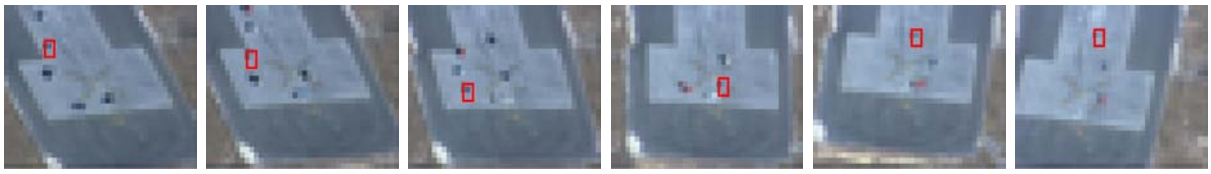


Fig. 22. An example on the *Gray Car* sequence using kernel metric preservation.

Fig. 23. An example on the *Red Car* sequence using kernel metric preservation.

[4] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2007.

[5] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56 –65, Mar./Apr. 2002.

[6] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–275 – I–282 Vol.1, Jun. 2004.

[7] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, pp. 2861 –2873, Nov. 2010.

[8] S. Fazli, H. Pour, and H. Bouzari, "Particle filter based object tracking with sift and color feature," in *International Conference on Machine Vision*, pp. 89 –93, Dec. 2009.

[9] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in *European Conference on Computer Vision*, Sept. 2010.

[10] X. S. Zhou, A. Gupta, and D. Comaniciu, "An information fusion framework for robust shape tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 115 –129, Jan. 2005.

[11] J. Nascimento and J. Marques, "Robust shape tracking in the presence of cluttered background," in *International Conference on Image Processing*, vol. 3, pp. 82 –85 vol.3, 2000.

Fig. 24. An example on the *Ice Hockey* sequence using kernel metric preservation.

[12] W. Li, X. Zhang, J. Gao, W. Hu, H. Ling, and X. Zhou, "Discriminative level set for contour tracking," in *International Conference on Pattern Recognition*, pp. 1735 –1738, Aug. 2010.

[13] W. Li, X. Zhang, and W. Hu, "Contour tracking with abrupt motion," in *International Conference on Image Processing*, pp. 3593 –3596, Nov. 2009.

[14] C. Gentile, O. Camps, and M. Sznaier, "Segmentation for robust tracking in the presence of severe occlusion," *IEEE Transactions on Image Processing*, vol. 13, pp. 166 –178, Feb. 2004.

[15] C. Aeschliman, J. Park, and A. Kak, "A probabilistic framework for joint segmentation and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1371 –1378, Jun. 2010.

[16] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 261 –271, Feb. 2007.

[17] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via On-line Boosting," in *British Machine Vision Conference*, 2006.

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

[19] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *European Conference on Computer Vision*, ECCV '08, pp. 234–247, 2008.

[20] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, pp. 1327 –1344, Oct. 2004.

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

[22] X. Wang, G. Hua, and T. X. Han, "Discriminative tracking by metric learning," in *European Conference on Computer Vision*, 2010.

[23] N. Jiang, H. Su, W. Liu, and Y. Wu, "Tracking low resolution objects by metric preservation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1329–1336, 2011.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pp. 674–679, 1981.

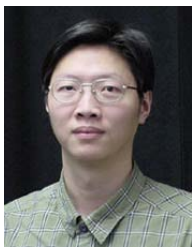
[25] S. Stalder, H. Grabner, and L. van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *International Conference on Computer Vision Workshops*, pp. 1409 –1416, Oct. 2009.

[26] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 810 –815, Jun. 2004.

[27] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1631 –1643, Oct. 2005.

Nan Jiang received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006. She is currently pursuing the Ph.D degree in the Department of Electronics and Information Engineering, HUST. Her research interests include computer vision and pattern recognition.

Wenyu Liu (M08) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a Professor and Associate Dean of the Department of Electronics and Information Engineering, HUST. His current research areas include multimedia information processing, and computer vision.



Ying Wu (SM06) received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 1994, the M.S. degree from Tsinghua University, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 2001. From 1997 to 2001, he was a Research Assistant at the Beckman Institute for Advanced Science and Technology, UIUC. During summer 1999 and 2000, he was a research intern with Microsoft Research, Redmond, WA. In 2001, he joined the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, as an Assistant Professor. He is currently an Associate Professor of electrical engineering and computer science at Northwestern University. His current research interests include computer vision, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction. Dr. Wu serves as an associate editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, the SPIE Journal of Electronic Imaging, and the IAPR Journal of Machine Vision and Applications. He received the Robert T. Chien Award at UIUC in 2001 and the National Science Foundation CAREER Award in 2003.