Multiple Collaborative Kernel Tracking

Zhimin Fan, Ming Yang, and Ying Wu

Abstract—Those motion parameters that cannot be recovered from image measurements are *unobservable* in the visual dynamic system. This paper studies this important issue of singularity in the context of kernel-based tracking and presents a novel approach that is based on a motion field representation which employs redundant but sparsely correlated local motion parameters instead of compact but uncorrelated global ones. This approach makes it easy to design fully observable kernel-based motion estimators. This paper shows that these high-dimensional motion fields can be estimated efficiently by the *collaboration* among a set of simpler local kernel-based motion estimators, which makes the new approach very practical.

Index Terms—Kernel-based tracking, multiple kernel, visual tracking.

1 Introduction

Target motion can be estimated through its visual observations (or measurements). But, there exist singular cases where not all motion parameters can be recovered through this image evidence. Those motion parameters that cannot be recovered through their image observations are *unobservable* in the visual dynamic system. Such singularities are more pronounced for motions beyond the 2D translation. Even in the simplest case of optical flow, knowing the visual measurements (i.e., the image gradients and frame differences) at one pixel is not sufficient to determine the flow at this particular pixel.

This singularity issue has been widely studied in the context of optical flow. One solution [10] is to include multiple measurements from nearby pixels, assuming they are independent pieces of evidence of the same motion so that a unique flow for the pixel of interest can be obtained by the least squares estimate. Another solution considers the spatial coherence constraints, e.g., reinforcing the smoothness among the flows of multiple nearby pixels so as to solve a global smooth flow field [9] or imposing a parametric model of the flows to estimate a parametric flow field.

Most flow computation methods assume that the corresponding pixels have the same intensity (i.e., the brightness constancy constraint). This is limited when applied to the real world. Many generalizations have been investigated, among which kernel-based methods [4], [6], [12], [14] have been very successful in visual tracking. The visual observations are the distributions of certain visual features (e.g., color) within a region and are expressed in an analytical form by kernel density estimators [2], [11].

Differently from traditional optical flow methods, these kernel-based methods employ more tolerant visual measurements for motion estimation. But, the singularity issue may still exist. The question is how do these kernel-based measurements influence the motion observability. Specifically, three critical issues need to be investigated: 1) Is there a criterion that evaluates the motion observability? 2) Is there a generalizable approach for designing fully observable kernel-based trackers? 3) Can we cope with such singularities efficiently for more complex motions (e.g., articulation)?

There have been some initial studies on multiple kernels which attempted to address these issues. For example, an outstanding investigation was presented in [8], where an unconstrained least squares formulation is given and the motion singularity is related to the rank deficiency, based on which the singularity is alleviated by concatenating multiple kernels [7]. In [3], to deal with the difficulty of estimating scales in kernel-based methods [5], multiple kernels of different resolutions are combined.

Having multiple kernels for collecting more independent measurements of the same motion (up to a Jacobian), the method in [8] extends the idea of [10] for a more general motion estimation. This method does not consider the constraints in the motion parameters, but treats them independently. This is suitable for the case where the motion is represented by a minimum number of independent motion parameters (equal to the number of degrees of freedom of the motion, e.g., six for a 2D affine motion). But, the complexity of the motion estimation is generally exponential to the number of motion parameters, which is the dimensionality of the search space.

Inspired by [8], [13], we present a new study that can be thought of as a generalization of the idea of considering the spatial coherence [9] in kernel-based motion estimation. Here, as a relaxed representation, a complex motion is represented by a set of spatially distributed but correlated simpler subpart motions (i.e., a motion field), each of which is associated with a local kernel-based motion estimator for the subpart. This redundant model brings the motion to a high-dimensional space, but the dependencies among these simpler motions (not limited to the smoothness) constrain the motion in a lower dimensional manifold. Even if some local motion parameters may not be directly observable from image observations locally associated with them, their correlation with others may still make them observable. More interestingly, this paper shows that this motion field can be estimated efficiently through collaboration among the set of simpler kernel-based motion estimators.

2 Kernel-Based Motion Estimation

To make this paper self-contained, we briefly review the kernel-based tracking methods by following the notations in [8]. A color histogram, $\mathbf{q} = [q_1, q_2, \ldots, q_m]^T \in \mathbb{R}^m$, is often used to represent the region of interest with $q_u = \frac{1}{C}\sum_{i=1}^{n} K(\mathbf{x}_i - \mathbf{c})\delta(b(\mathbf{x}_i), u)$, where $\{\mathbf{x}_i\}_{i=1\ldots n}$ are the pixel locations of the target, $b(\mathbf{x}_i)$ is a binning function that maps the color of $\mathbf{x}_i$ onto a histogram bin u, with $u \in \{1 \ldots m\}$. K is a spatially weighting kernel centered at $\mathbf{c}$ and δ is the Kronecker delta function, C is a normalization term. Its matrix form [8] is

$$\mathbf{q}(\mathbf{c}) = \mathbf{U}^T\mathbf{K}(\mathbf{c}), \tag{1}$$

where

$$\mathbf{U} = \begin{bmatrix} \delta(b(\mathbf{x}_1), u_1) & \ldots & \delta(b(\mathbf{x}_1), u_m) \\ \vdots & \vdots & \vdots \\ \delta(b(\mathbf{x}_n), u_1) & \ldots & \delta(b(\mathbf{x}_n), u_m) \end{bmatrix} \in \mathbb{R}^{n\times m}, \quad \text{and}$$

$$\mathbf{K} = \frac{1}{C}\begin{bmatrix} K(\mathbf{x}_1 - \mathbf{c}) \\ \vdots \\ K(\mathbf{x}_n - \mathbf{c}) \end{bmatrix} \in \mathbb{R}^n.$$

The histograms of the candidate region and the target are represented by $\mathbf{p}(\mathbf{c}) = \mathbf{U}^T\mathbf{K}(\mathbf{c})$ and $\mathbf{q} = \mathbf{U}^T\mathbf{K}$. Given an initial start

• *The authors are with the Electrical Engineering and Computer Science Department, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208. E-mail: {zfa825, mya671, yingwu}@ece.northwestern.edu.*

Manuscript received 28 May 2005; revised 7 Mar. 2006; accepted 3 Aug. 2006; published online 18 Jan. 2007.
Recommended for acceptance by D. Comaniciu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0272-0505.
Digital Object Identifier no. 10.1109/TPAMI.2007.1034.

location $\mathbf{c}$, the tracker needs to find the best displacement $\Delta\mathbf{c}$ such that the measurements $\mathbf{p}(\mathbf{c}+\Delta\mathbf{c})$ best match the target $\mathbf{q}$, i.e.,

$$\Delta\mathbf{c}^* = \arg\min_{\Delta\mathbf{c}} O(\mathbf{q}, \mathbf{p}(\mathbf{c}+\Delta\mathbf{c})), \tag{2}$$

where $O(\cdot,\cdot)$ is the matching objective function, e.g., the Bhattacharyya coefficient [5] or the equivalent Matusita metric [8], i.e., $O(\Delta\mathbf{c}) \triangleq \|\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c}+\Delta\mathbf{c})}\|^2$.

Various optimization techniques have been employed to solve this problem, such as the mean shift procedure [5] or a Newton-style method [8]. In practice, the optimization is sometimes plagued by a singularity situation where the optimal solution to $\Delta\mathbf{c}$ cannot be uniquely determined. Inspired by the initial analysis in [8], we present in the next sections a more general view and solution.

3 Motion Observability for Kernel-Based Methods

The issue of motion observability can be related to the "system-observability" of a more general system in (3) for a better definition and explanation.

$$\begin{cases} \Omega(\mathbf{x}) &= \mathbf{w}_1 \\ \mathbf{y} &= \mathcal{H}(\mathbf{x}) + \mathbf{w}_2, \end{cases} \tag{3}$$

where $\Omega(\mathbf{x})$ represents the inherent property of the hidden variable $\mathbf{x}$, such as the prior constraints or the dynamics, $\mathcal{H}$ denotes the observation or measurement process, and $\mathbf{w}_1$ and $\mathbf{w}_2$ are noise terms. The variable $\mathbf{x}$ is hidden and can only be estimated through the measurement $\mathbf{y}$. In the tracking scenario, the hidden variable refers to the motion. A critical issue is whether or not $\mathbf{x}$ can be *uniquely determined* from $\mathbf{y}$, i.e., the *observability* of this general system.

In the context of kernel tracking, we treat $\mathbf{x} \triangleq \Delta\mathbf{c}$. Our analysis is based on the linearization of the system at a given initial start $\mathbf{c}$. Since the observation model for $\mathbf{c}+\Delta\mathbf{c}$ is the histogram difference, i.e., $\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c}+\Delta\mathbf{c})}$, we linearize it and obtain

$$\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c})} = \mathbf{M}\Delta\mathbf{c} + o(\Delta\mathbf{c}),$$

where $\sqrt{\mathbf{q}}$, $\sqrt{\mathbf{p}(\mathbf{c})} \in \mathbb{R}^m$, $\Delta\mathbf{c} \in \mathbb{R}^r$, $\mathbf{M} \in \mathbb{R}^{m\times r}$, and

$$\mathbf{M} = \frac{1}{2}\mathtt{diag}(\mathbf{p}(\mathbf{c}))^{-\frac{1}{2}}\mathbf{U}^T\mathbf{J}_\mathbf{K}(\mathbf{c}), \tag{4}$$

$$\mathbf{J}_\mathbf{K}(\mathbf{c}) = \begin{bmatrix} \nabla_c K(\mathbf{x}_1-\mathbf{c}) & \nabla_c K(\mathbf{x}_2-\mathbf{c}) & \cdots & \nabla_c K(\mathbf{x}_n-\mathbf{c}) \end{bmatrix}^T, \tag{5}$$

$\mathtt{diag}(\mathbf{p})$ denotes a matrix with $\mathbf{p}$ on its diagonal, and r is the dimension of the motion. This result was actually obtained in [8]. In view of this, treating the measurement $\mathbf{y} \triangleq \sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{c})}$ and higher order terms $o(\Delta\mathbf{c})$ as noise, we write the linearized **measurement equation** as

$$\mathbf{y} = \mathbf{M}\Delta\mathbf{c} + \mathbf{w}_2 = \mathbf{M}\mathbf{x} + \mathbf{w}_2. \tag{6}$$

When the motion constraints hold at $\mathbf{c}+\Delta\mathbf{c}$, i.e., $\Omega(\mathbf{c}+\Delta\mathbf{c}) = 0$, we can always linearize it as $\Omega(\mathbf{c}) + \Omega'(\mathbf{c})\Delta\mathbf{c} + o(\Delta\mathbf{c}) = 0$. Thus, when we define $l \triangleq -\Omega(\mathbf{c})$ and $\mathbf{G} \triangleq \Omega'(\mathbf{c})$, we have a linearized **system constraint equation**

$$l = \Omega'(\mathbf{c})\Delta\mathbf{c} = \mathbf{G}\mathbf{x} + \mathbf{w}_1, \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^r$ and $\mathbf{G} \in \mathbb{R}^{s\times r}$, s is the number of linear constraints. We have the following theorem that stipulates the kernel observability:

Theorem 1 (Motion Observability). *The system described by (6) and (7) is observable, i.e., unique recovery of* $\mathbf{x}$ *is guaranteed, iff*

$$\mathtt{rank}(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G}) = r, \quad \forall\gamma > 0, \tag{8}$$

i.e., $(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})$ *is of a full rank.*

The proof of the theorem is given in [7]. According to this theorem, for a single kernel without the system constraint, if $\mathbf{M}$ is not of a full rank, it is clear that the unique solution to the motion $\Delta\mathbf{c}$ cannot be obtained. This motivates our proposed approach of multiple collaborative kernels of a field in Section 4.

4 Multiple Collaborative Kernels

The proposed approach represents the target motion as a field of spatially distributed but correlated simpler motions, each of which is associated with a local kernel-based motion estimator. In this relaxed and redundant model, the motion lies in a high-dimensional space that is a direct product of the spaces of the motion for all kernels. In addition, constraints among these local motion parameters are imposed. Therefore, the motion actually lies in a lower dimensional manifold (linear or nonlinear) of such a high-dimensional space.

4.1 The Formulation

We consider a set of w kernels with constraints. Suppose $\mathbf{c}_i$ is the motion parameters (e.g., the location) of the ith kernel, and the constraints among them are represented by $\Omega(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_w) = 0$. Thus, we consider a new objective function

$$O(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_w) = \sum_{i=1}^{w} \|\sqrt{\mathbf{q}_i} - \sqrt{\mathbf{p}_i(\mathbf{c}_i)}\|^2 + \gamma\|\Omega(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_w)\|^2, \tag{9}$$

where $\gamma > 0$ is constant and can be the optimal Lagrange multipliers if the constraints need to hold exactly. If the constraints can tolerate some small errors, γ balances such an inaccuracy of the motion constraints and the matching of collected image evidence.

After the linearization with regard to $\Delta\mathbf{c}_1, \Delta\mathbf{c}_2, \ldots, \Delta\mathbf{c}_w$, we have the following general system equation and measurement equation:

$$\begin{cases} l &= \mathbf{G}\underline{\Delta\mathbf{c}} + \mathbf{w}_1 \\ \mathbf{y} &= \mathbf{M}\underline{\Delta\mathbf{c}} + \mathbf{w}_2, \end{cases} \tag{10}$$

where

$$\begin{aligned} \underline{\Delta\mathbf{c}} &= \begin{bmatrix} \Delta\mathbf{c}_1 \\ \Delta\mathbf{c}_2 \\ \cdots \\ \Delta\mathbf{c}_w \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \sqrt{\mathbf{q}_1} - \sqrt{\mathbf{p}(\mathbf{c}_1)} \\ \sqrt{\mathbf{q}_2} - \sqrt{\mathbf{p}(\mathbf{c}_2)} \\ \cdots \\ \sqrt{\mathbf{q}_w} - \sqrt{\mathbf{p}(\mathbf{c}_w)} \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} \mathbf{M}_1 & 0 & 0 & 0 \\ 0 & \mathbf{M}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{M}_w \end{bmatrix}, \\ \mathbf{G} &= \begin{bmatrix} \frac{\partial\Omega}{\partial\mathbf{c}_1} & \frac{\partial\Omega}{\partial\mathbf{c}_2} & \cdots & \frac{\partial\Omega}{\partial\mathbf{c}_w} \end{bmatrix}, \quad l = -\Omega(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_w), \end{aligned} \tag{11}$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are noise terms. Based on the kernel observability theorem in Section 3, the observability of this formulation is given by checking $\mathtt{rank}(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})$. This is equivalent to the column rank of

$$\begin{bmatrix} \mathbf{M} \\ \sqrt{\gamma}\mathbf{G} \end{bmatrix},$$

which will be no less than that of $\mathbf{M}$. Now, it is worth pointing out that, without the introduced constraint $\Omega(\cdot)$, i.e., $\mathbf{G} = \mathbf{0}$, the solution will be reduced to $\mathbf{y} = \mathbf{M}\underline{\Delta\mathbf{c}}$, which is equivalent to solving the w kernel tracking problems *independently*, requiring $\mathbf{M}$ to have a full column rank, i.e., every kernel needs to be observable.

The advantage of the collaborative kernels is that it does not require all of the kernels to be fully observable. Another advantage of this relaxed approach is that it enables recovery of complex motions. We can use a number of kernels with simple motions (e.g., the 2D displacements) and deploy them at different places in the image of the target. These 2D displacements are constrained. It is much easier to estimate these simple motions than complex motions directly. Thus, this approach greatly facilitates the recovery of other motion parameters, such as the affine motion, articulation, deformation, etc.

The above method is quite general. Let's have a special example, where the constraints are $\Delta\mathbf{c}_1 = \Delta\mathbf{c}_2 = \cdots = \Delta\mathbf{c}_w$ (up to a noise term). It is easy to see that

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & & \\ & \mathbf{I} & -\mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I} & -\mathbf{I} \end{bmatrix}, \quad \text{and} \quad l = 0.$$

We have $\mathtt{rank}(\mathbf{G}) = (w-1) \times \mathtt{dim}(\mathbf{c}_1)$. Supposing $w = 10$ and $\mathtt{dim}(\mathbf{c}_1) = 2$, this implies that the motion resides in a 2D manifold in $\mathbb{R}^{20}$. Thus, as long as $\mathtt{rank}(\mathbf{M})$ is not less than $\mathtt{dim}(\mathbf{c}_1)$, all of the motion parameters are observable or can be uniquely determined. This condition can be easily satisfied if any of the $\mathbf{c}_i$ is observable through its kernel or there are a number of $\mathtt{dim}(\mathbf{c}_1)$ motion parameters that are observable through multiple kernels. This is different from the method in [8] that represents the motion directly in a space with dimensionality $\mathtt{dim}(\mathbf{c}_1)$ and concatenates multiple kernel measurements [7].

4.2 The System Constraints

The constraints among multiple kernels may have different types. For example, one can use *componentwise* constraints that correlate two nearby kernels [7]. In this paper, we introduce a more general type, i.e., the *subspace* constraints that capture the global relations among multiple kernels.

We assume that the multiple kernels are placed at $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_w$. For subspace modeling, a common treatment is to eliminate the translation by subtracting the *mean* vector, for $j = 1, \ldots, w$, $\bar{\mathbf{s}}_j = \mathbf{s}_j - \mathbf{s}_{mean}$, where

$$\mathbf{s}_{mean} = \frac{\sum_{j=1}^{w} \mathbf{s}_j}{w}.$$

In the following, the script, f ($f = 1, 2, \ldots$), indexes the frame number. After obtaining a set of concatenated vectors of kernel locations collected from a series of frames,

$$\underline{\bar{\mathbf{c}}}^f = [\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \ldots, \bar{\mathbf{s}}_w]_f^T = [\mathbf{s}_1 - \mathbf{s}_{mean}, \mathbf{s}_2 - \mathbf{s}_{mean}, \ldots, \mathbf{s}_w - \mathbf{s}_{mean}]_f^T \in \mathcal{R}^{2w}, \tag{12}$$

we can apply PCA to obtain a lower d-dimensional subspace representation, $\mathcal{S}^d$. For example, the subspace of the in-plane rigid motion will be of dimension 2. An affine motion model will yield an even higher motion subspace. Then, the concatenated kernel *displacement* in each frame, $\underline{\Delta\bar{\mathbf{c}}}^f$, $f = 1, 2, \ldots$, resides in the same subspace as well, $\underline{\Delta\bar{\mathbf{c}}}^f \in \mathcal{S}^d$. Recalling (12), it should be clear that

$$\begin{aligned} \underline{\Delta\bar{\mathbf{c}}}^f &= [\Delta\bar{\mathbf{c}}_1, \Delta\bar{\mathbf{c}}_2, \ldots, \Delta\bar{\mathbf{c}}_w]_f^T = \underline{\Delta\mathbf{c}}^f - \underline{\Delta\mathbf{c}}_{mean}^f, \\ \underline{\Delta\mathbf{c}}^f &= [\Delta\mathbf{c}_1, \Delta\mathbf{c}_2, \ldots, \Delta\mathbf{c}_w]_f^T, \\ \underline{\Delta\mathbf{c}}_{mean}^f &= [\Delta\mathbf{c}_{mean}, \Delta\mathbf{c}_{mean}, \ldots, \Delta\mathbf{c}_{mean}]_f^T, \end{aligned}$$

where $\Delta\mathbf{c}_{mean} = \frac{\sum_{j=1}^{w} \Delta\mathbf{c}_j}{w}$. The notation $\underline{\Delta\bar{\mathbf{c}}}$, which has the mean vector subtracted, is used here to differentiate with respect to $\underline{\Delta\mathbf{c}}$, the vector without mean subtraction, in Section 4.1.

This brings a practical constraint on $\underline{\Delta\bar{\mathbf{c}}}^f$ when we solve for the kernel displacements in each frame f. The imposed subspace constraint, or the **system constraint equation**, is formulated as follows:

$$(\mathbf{I} - \mathbf{V}\mathbf{V}^T)(\underline{\Delta\mathbf{c}} - \underline{\Delta\mathbf{c}}_{mean}) = 0, \tag{13}$$

where $V = [v_1, v_2, \ldots, v_d]$ is the learned orthonormal basis of the model subspace $\mathcal{S}^d$. The dimension d is determined by checking the steepest drop in the sorted eigenvalues. Equation (13) actually represents that the vector $\underline{\Delta\mathbf{c}} - \underline{\Delta\mathbf{c}}_{mean}$ resides in the subspace spanned by $\mathbf{V}$. Combining (13) with (6), we have

$$\begin{cases} (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\underline{\Delta\mathbf{c}}_{mean} &= (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\underline{\Delta\mathbf{c}} + \mathbf{w}_1 \\ \mathbf{y} &= \mathbf{M}\underline{\Delta\mathbf{c}} + \mathbf{w}_2. \end{cases} \tag{14}$$

Being in the form of (10), such subspace constraints also improve the kernel-observability.

Another advantage is that the subspace constraints can tolerate some scale changes. Seen from (12), if the object exhibits scale changes, i.e., the object shrinks or expands, so do the kernels' relative positions, $\bar{\mathbf{c}}^f$. However, only scale changes, i.e., changes in the kernels' relative positions, will not affect the formation of the subspace. In other words, the subspace model, built upon the relative position of the kernels, still governs the kernel positions when scale changes are present. Thus, it tolerates scale changes. Experiments are given in Section 5.3.

4.3 The Collaboration

The solution to the linear system (10) for multiple collaborative kernel tracking is given by:

$$\underline{\Delta\mathbf{c}} = (\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})^{-1}(\mathbf{M}^T\mathbf{y} + \gamma\mathbf{G}^T l). \tag{15}$$

Specifically, for the componentwise description in Section 4.1, $\mathbf{G} = \begin{bmatrix} \frac{\partial\Omega}{\partial\mathbf{c}_1} & \frac{\partial\Omega}{\partial\mathbf{c}_2} & \cdots & \frac{\partial\Omega}{\partial\mathbf{c}_w} \end{bmatrix}$ and $l = -\Omega(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_w)$. For the subspace constraint model, $\mathbf{G} = \mathbf{I} - \mathbf{V}\mathbf{V}^T$, $l = (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\underline{\Delta\mathbf{c}}_{mean}$.

Due to the relaxation of the system states, the dimension of the matrix $(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})$ can be quite large (the sum of motion parameters of all individual kernels). Thus, it is computationally demanding to calculate its inverse. Considering the special structure of $\mathbf{M}$, we obtain a much more efficient method, which precisely reveals the collaboration among multiple kernels.

By applying matrix inversion lemma,[1] we can obtain

$$\underline{\Delta\mathbf{c}} = (\mathbf{I} - \mathbf{D})(\mathbf{M}^T\mathbf{M})^{-1}(\mathbf{M}^T\mathbf{y} + \gamma\mathbf{G}^T l), \tag{16}$$

where $\mathbf{D} = \gamma(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T(\gamma\mathbf{G}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T + \mathbf{I})^{-1}\mathbf{G}$.

Provided that $\mathbf{M}^T\mathbf{M}$ is nonsingular, this equation means that we can save the computational cost on $(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})^{-1}$ by computing $(\gamma\mathbf{G}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T + \mathbf{I})^{-1}$ and $(\mathbf{M}^T\mathbf{M})^{-1}$ instead. Generally, the dimensionality of $(\gamma\mathbf{G}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T + \mathbf{I})$, which is equal to the number of constraints, is smaller than the number of parameters to be estimated, i.e., the dimensionality of $(\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})$. Moreover, the calculation of $(\mathbf{M}^T\mathbf{M})^{-1}$ is easy since it has a block-diagonal structure form (recalling the structure of $\mathbf{M}$ in (11)). In practice, the complexity is even lower since $\mathbf{G}$ is generally sparse. All of these count toward a potential reduction in the computational cost.

Note that the solution to the unconstrained problem, (i.e., independent kernels) is given by:

1. $(\mathbf{A} + \mathbf{B}\mathbf{D})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{I})^{-1}\mathbf{D}\mathbf{A}^{-1}$, where $\mathbf{A}$ is an n by n matrix, $\mathbf{B}$ is an n by m matrix, and $\mathbf{D}$ is an m by n matrix.

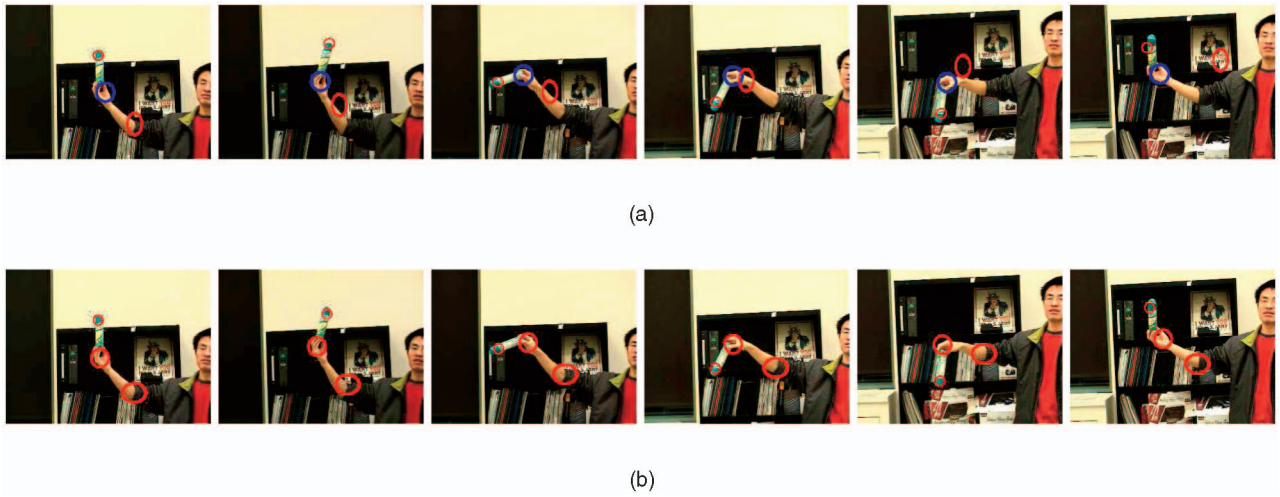

Fig. 1. Tracking an articulated structure. (a) Tracking without collaboration. (b) Tracking with collaboration by using componentwise constraint.

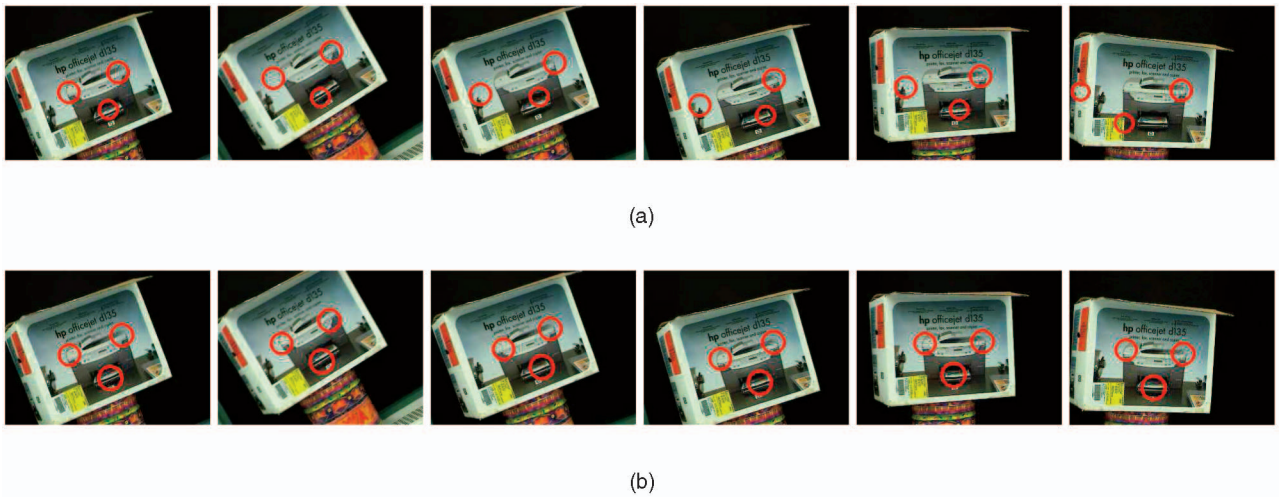

Fig. 2. Tracking three parts of interest on a box. (a) Tracking without collaboration. (b) Tracking with collaboration by using subspace constraint.

$$\underline{\Delta \mathbf{c}}_u = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{y} = \mathbf{M}^\dagger\mathbf{y}, \tag{17}$$

where $\mathbf{M}^\dagger$ is the pseudoinverse of $\mathbf{M}$. This unconstrained solution can be easily calculated with linear cost with regard to the number of kernels. Using the unconstrained solution $\underline{\Delta \mathbf{c}}_u$, we can write the solution to the problem, (16), as

$$\underline{\Delta \mathbf{c}} = (\mathbf{I} - \mathbf{D})\underline{\Delta \mathbf{c}}_u + \mathbf{z}(\mathbf{c}), \tag{18}$$

where $\mathbf{z}(\mathbf{c}) = \gamma(\mathbf{I} - \mathbf{D})(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T l$. The scheme of the collaboration among multiple kernels is pronounced: 1) Each individual single-kernel tracker follows its designated target and 2) exchanges "messages" to other single-kernel tracker. Such a collaboration ends up with an equilibrium where the entire target is tracked and the structural constraints of multiple kernels are satisfied. The collaboration actually suggests a very efficient recursive method of calculating the constrained solution by

$$\underline{\Delta \mathbf{c}}^{k+1} \longleftarrow (\mathbf{I} - \mathbf{D}^k)\left[\mathbf{M}(\underline{\Delta \mathbf{c}}^k)\right]^\dagger \mathbf{y}^k + \mathbf{z}^k, \tag{19}$$

which is very similar to the fixed point iteration and converges very quickly.

The most noticeable features of our work are the general description (in (3)) and the fixed-point collaboration scheme (in (19)). The general description sets a paradigm to accommodate various forms of constraints, such as the local componentwise model, the global subspace model, or the more complicated motion dynamics and the learned motion priors. The collaboration scheme breaks down the difficulty of searching a high-dimensional space into several simpler search tasks in lower dimensional spaces, thus achieving computational efficiency.

5 Experiments

In this section, we show the experiments of the proposed method to track articulated objects, deformable objects, objects with scale changes, and objects undergoing rotation.

5.1 Tracking Articulated Objects

To extend kernel-based tracking to an articulated object, we can put a set of collaborative kernels on various parts of the target. Each kernel only collect local evidence for a subpart motion. Once these subpart motions are estimated through the decentralized collaboration, it is easy to calculate the articulation (e.g., the joint angles). This is different from [1], which is a direct method to estimate the joint angles of the articulated structure by considering all the image measurements jointly.

An articulated structure consisting of an arm and a bottle in hand is shown in Fig. 1. We apply three kernels to the elbow, the hand, and the one end of the bottle, respectively. For each pair of kernels, we use a componentwise length constraint, $\|\mathbf{c}_1 - \mathbf{c}_2\|^2 = L^2$, with L given by the initialization. Compared with the result yielded by

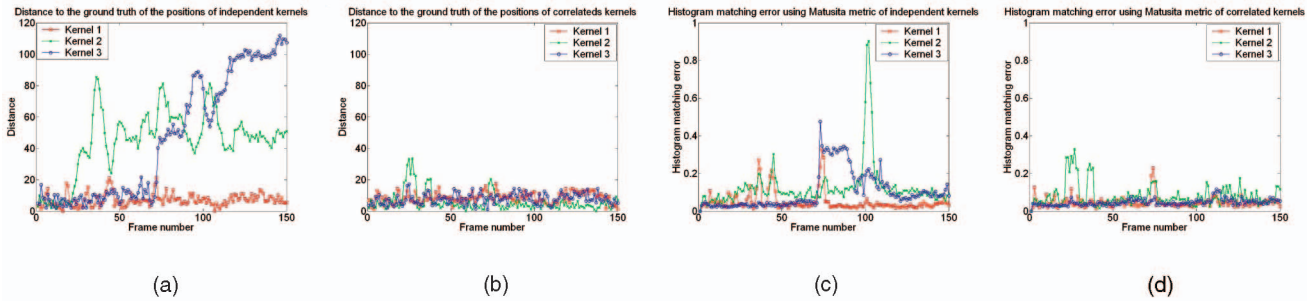

Fig. 3. Errors in kernel position compared with ground truth: (a) independent kernels and (b) collaborative kernels. Errors in histogram matching: (c) independent kernels and (d) collaborative kernels.

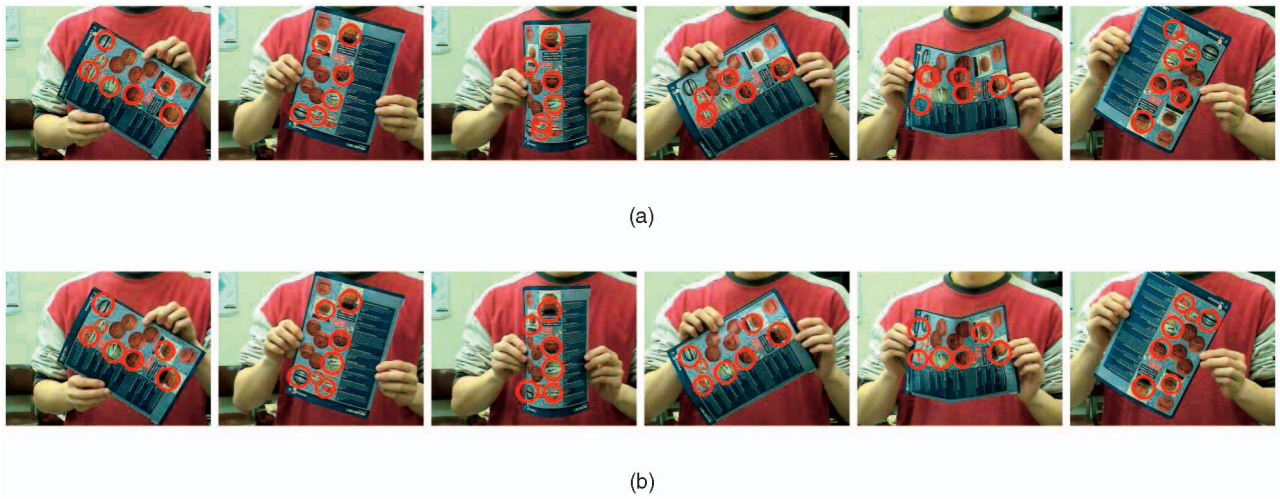

Fig. 4. Tracking a deformable target. (a) Tracking without collaboration. (b) Tracking with collaboration by using subspace constraint.

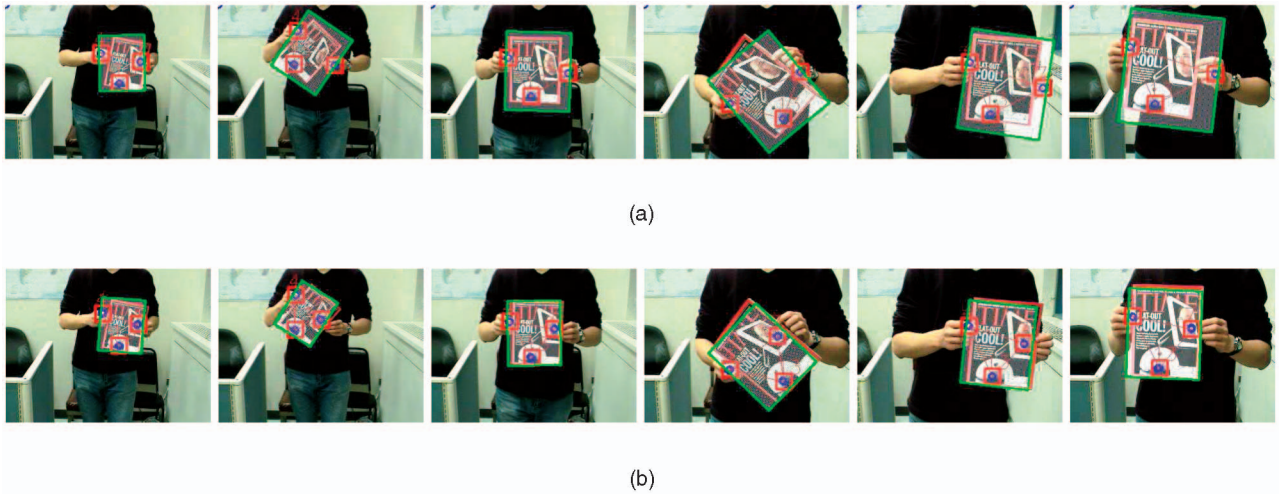

Fig. 5. Tracking a target with scale changes and rotation. (a) Tracking without collaboration. (b) Tracking with collaboration by using subspace constraint.

independent kernels (in Fig. 1a), two pairs of collaborative kernels (elbow and hand, hand and bottle tip) provide a much more robust performance as shown in Fig. 1b.

5.2 Using the Subspace Model

In this section, we demonstrate the advantages of using the subspace constraint model. In the first experiment, we manually labled the positions of three corners of a box, see Fig. 2, from 50 frames for training. A subspace model is then obtained by applying PCA on the vectors of kernel displacements $\underline{\Delta\bar{\mathbf{c}}} = \underline{\Delta\mathbf{c}} - \underline{\Delta\mathbf{c}}_{mean}$. Then, the solution (18) is used to guide the collaborative kernels for tracking.

We also test the single independent kernel method as the comparison in Fig. 2. Both methods are applied on every other frame of a testing video to simulate the presence of fast scene changes. Our results show that single kernels are more vulnerable to fast motion and distractions, as in Fig. 2a. In contrast, the enforced subspace constraint is capable of stabilizing the kernels, thus having a much better tracking performance, as shown in Fig. 2b.

We have also compared the results of single and collaborative kernels against the ground truth provided by manual labeling. Figs. 3a and 3b show the errors of kernel positions (measured by

pixels) obtained through 150 frames. Figs. 3c and 3d show the error of histogram matching.

In our second experiment, a more challenging task of tracking a deformable target is shown in Fig. 4. Fig. 4a shows the tracking result of six independent kernels. Their ability to tolerate distortion is poor. Several kernels easily drift away. The collaborative kernel tracker, as shown in Fig. 4b, achieves more robust performance. By framing in the subspace model, both the histogram matching and the subspace model contribute to the robustness against large shape distortions.

5.3 Tracking Object with Scale Changes and Rotation

Fig. 5 shows the result of tracking an object with scale changes and rotation. We initialize three kernels and train a subspace model of them for the collaborative tracker. The planar motion of the bounding box in Fig. 5 is easily estimated based on these three kernels. As shown in Fig. 5a, the performance of independent kernels is not warranted and the estimation of the planar motion of the bound box is not accurate. Since the subspace model is invariant to scale changes (as explained in Section 4.2), our method has better performance, as shown in Fig. 5b.

6 Conclusions

This paper studies the motion observability issue in the context of kernel-based tracking and presents a generalizable approach that prevents the singularity where part of the motion cannot be recovered from visual measurements. This new approach employs a relaxed representation for motion, i.e., by a network of subpart motions, each of which is only associated with its local image measurements. In addition, a flexible tracking scheme is proposed based on the fixed-point *collaboration* among the set of simpler kernel-based motion estimators for the individual subparts. This new multiple-kernel approach is naturally extended to track articulated objects.

Acknowledgments

This work was supported in part by US National Science Foundation (NSF) Grant IIS-0308222, NSF IIS-0347877 (CAREER), Northwestern startup funds, and the Murphy Fellowships.

References

[1] C. Bregler and J. Malik, "Tracking People with Twists and Exponential," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 8-15, 1998.
[2] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 790-799, Aug. 1995.
[3] R.T. Collins, "Mean-Shift Blob Tracking through Scale Space," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 234-240, 2003.
[4] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.
[5] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-575, May 2003.
[6] D. Comaniciu, V. Ramesh, and P. Meer, "The Variable Bandwidth Mean Shift and Data-Driven Scale Selection," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 438-445, 2001.
[7] Z. Fan, Y. Wu, and M. Yang, "Multiple Collaborative Kernel Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 502-59, 2005.
[8] G.D. Hager, M. Dewan, and C.V. Stewart, "Multiple Kernel Tracking with SSD," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 790-797, 2004.
[9] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.
[10] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.
[11] M.P. Wand and M.C. Jones, *Kernel Smoothing,* first ed. Chapman and Hall, 1995.
[12] J. Wang, B. Thiesson, Y. Xu, and M.F. Cohen, "Image and Video Segmentation by Anisotropic Kernel Mean Shift," *Proc. European Conf. Computer Vision,* 2004.
[13] Y. Wu, G. Hua, and T. Yu, "Tracking Articulated Body by Dynamic Markov Network," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1094-1101, 2003.
[14] Z. Zivkovic and B. Krose, "An EM-Like Algorithm for Color-Histogram-Based Object Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 798-803, 2004.