Dynamic Context for Tracking behind Occlusions

Fei Xiong, Octavia I. Camps, and Mario Sznaier*

Dept. of Electrical and Computer Engineering,
Northeastern University, Boston, MA 02115
{xiong.f,o.camps,msznaier}@neu.edu
http://robustsystems.ece.neu.edu

Abstract. Tracking objects in the presence of clutter and occlusion remains a challenging problem. Current approaches often rely on *a priori* target dynamics and/or use nearly rigid image context to determine the target position. In this paper, a novel algorithm is proposed to estimate the location of a target while it is hidden due to occlusion. The main idea behind the algorithm is to use contextual dynamical cues from multiple supporter features which may move with the target, move independently of the target, or remain stationary. These dynamical cues are learned directly from the data without making prior assumptions about the motions of the target and/or the support features. As illustrated through several experiments, the proposed algorithm outperforms state of the art approaches under long occlusions and severe camera motion.

Keywords: dynamics-based tracking, occlusion, context.

1 Introduction

The focus of this paper is to provide accurate estimates of the location of a target while it is not visible due to long occlusions. Persistent tracking is fundamental to many computer vision applications including, surveillance, structure from motion, activity recognition and human computer interfaces, just to mention a few. However, the problem of keeping track of a target in the face of temporary occlusions remains challenging in spite of the very extensive body of work on this topic.

Recent tracking approaches [4,5,6] incorporate concepts from object detection and recognition to recapture the target after it was occluded. However, these approaches cannot provide estimates of the location of an occluded target and hence must rely entirely on its appearance which, in turn, can change significantly and lead to tracking failures. More traditional approaches to tracking seek to improve robustness to occlusion by estimating the position of the target through Kalman, Extended-Kalman or Particle filtering [7,8,9,10]. While successful in many scenarios, these approaches often suffer from assuming too simplistic dynamical models (i.e. brownian motion, constant velocity, etc)

* This work was supported in part by NSF grants IIS–0713003 and ECCS–0901433, AFOSR grant FA9550–12–1–0271, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 580–593, 2012.

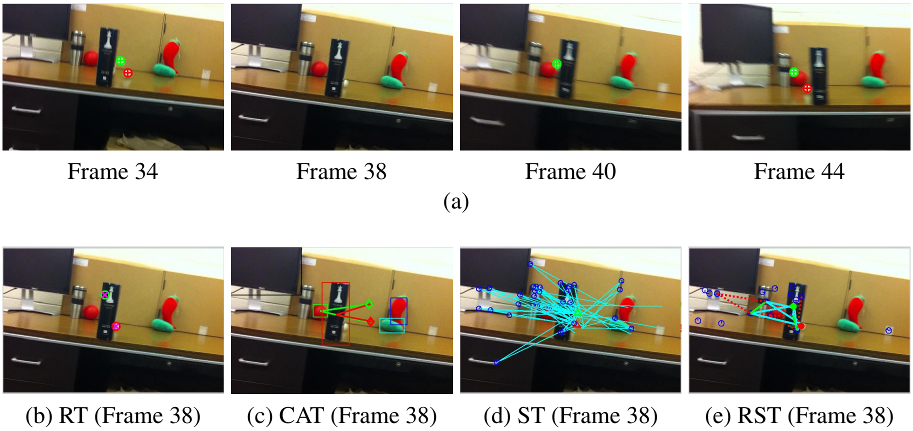
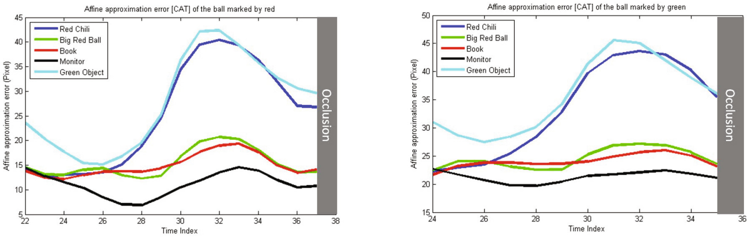


Fig. 1. Tracking with occlusion and severe camera motion. (a) Sample frames of a bouncing (marked in green) and a rolling (marked in red) ping-pong balls. (b-e) Estimated locations by using (b) a piece-wise linear motion model [1]; (c) context regions and an affine relative motion model [2]; (d) support features and an approximately constant distance model [3]; (e) dynamic support features as proposed in this paper.

and can easily drift in the presence of prolonged occlusions and clutter similar to the target. This problem can be avoided in part by assuming a piecewise linear dynamical model that is fitted as the data becomes available [1,11]. However, these techniques can also fail if occlusions are very long or if the camera undergoes severe motion. This problem is illustrated in Figure 1b where a tracker using this approach fails to estimate the correct location of a bouncing ping-pong ball while occluded by a black book in a video captured by a camera undergoing severe motion.

More recently, [2,12,3] proposed to use context relationships that exploit strong motion correlations between the target and near-by regions or features. Both [2,12] propose to use "companion regions" close to the target. The approach used in [12] only works for a stationary camera and the companion region, which remains unchanged as the target moves, must be manually selected. On the other hand, in [2] context regions are found using a color-based split-and-merge segmentation algorithm and selected through a mining process that looks for large regions that co-occur with the target and whose locations are related to the target position through an affine model. Thus, in practice, the target and the context objects are roughly modeled as moving approximately as a rigid in 3D and in close proximity to each other. As a result, the number of suitable context objects is rather limited since they have to be large regions and neither regions with complex motions nor stationary background regions perform well as context features. This is illustrated in Figure 1c where the only regions available to be used as context belong to the background. In this case, these regions appear to be moving due to the camera motion. However, the relative motion between the targets (rolling and bouncing ping-pong balls) and the candidate context regions cannot be modeled as affine (see Figure 2) due to the severe camera motion. Indeed, the estimates of the locations of the targets provided by the two best context regions are very far from the true position.



(a) Rolling ball (b) Bouncing Ball

Fig. 2. Affine based localization errors using different context objects for two targets in Figure 1a using the method proposed in [2]

Furthermore, it should be noted that even when good context regions that move with the target are available, the context objects are likely to be occluded along with the target due to their proximity, leaving the tracker vulnerable.

In [3], the authors propose to use a Generalized Hough Transform approach with a more dense set of "supporter features" that are learned and updated over time which allows the tracker to handle moderate camera motion. However, the method uses a simple motion support model that assumes that the relative position between the supporter features and the target is more or less constant and uses a manually selected forgetting factor to weight previous estimates. As a result, their predictions can be incorrect due to the mismatch between the true and the assumed motion model[1] as illustrated in Figure 1d where the tracker incorrectly estimates the locations of the bouncing and the rolling ping-pong balls.

1.1 Paper Contributions

In this paper, we propose a robust context based tracking algorithm – the Robust Supporter Tracking (RST) algorithm – that uses the relative dynamics between the target and contextual features to estimate the location of the target during occlusion and in the presence of severe camera motion. The main idea is to be able to exploit dense context features that exhibit different levels of motion correlation with the target, such as the green, yellow and red trajectories shown in Figure 3, as they become available. In this way, the proposed tracker can fuse information supported by the available features, but it is not restricted to only using highly correlated or physically proximate ones. To accomplish this, the tracker estimates the motion correlation between candidate supporter features and the target, and weights the target location estimates accordingly. It should be noted that the estimates are done using Hankel matrices of sequences of measurements, *without making a priori assumptions about their dynamics.* The proposed method is inspired by the ideas proposed by [2,3] and the Hankel based trackers proposed in [1,11]. Yet, as illustrated in Figure 1e and in the experiments in section 5 , the

[1] This problem is also exacerbated by the fact that predictions are made using a polar coordinate system based on the dominant orientation of a histogram of gradients which is sensitive to noise and illumination variations.

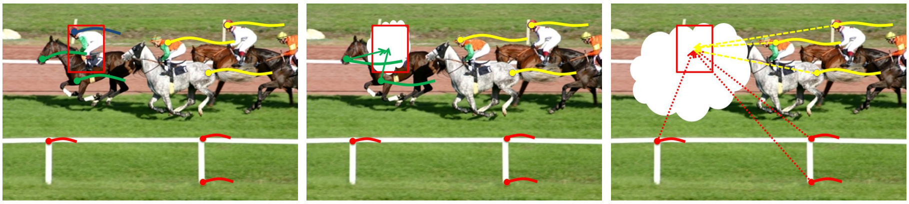


Fig. 3. 3a A frame with the target marked by a red rectangle and feature trajectories. 3b The rigidly coupled supporters are helpful to estimate the position of the occluded target. When there are no rigidly correlated features, such as in 3c, the target motion can still be modeled and estimated.

proposed tracker makes more accurate estimations of the target location and is more robust to severe camera motion than these approaches. In particular, the RST tracker has the following advantages:

1. Compared to [2,3], it can use *arbitrarily* complex relative dynamics between the context features and the target, and supporter features may move or not with the target. Thus, the number of context features available is larger than in these previous approaches. Furthermore, the context features proposed here tend to be better spatially distributed in the images as they do not need to be in close proximity with the target. Finally, it weights the reliability of the predictions according to the complexity of the dynamics. As a result, the estimates of the target position are more robust to noise, especially during large and long occlusions, even in the presence of severe camera motion.
2. Compared to [1,11], it uses multiple cues to estimate the target location. As a result, it is more robust to noise, long occlusions and changes in dynamics.

This paper is organized as follows. Section 2 summarizes affine invariant properties used in our approach. Section 3 explains the details of the proposed algorithm. Sections 4 and 5 describe implementation details and experiments comparing the proposed algorithm against state of the art approaches, respectively. Finally, section 6 gives final remarks.

2 Background: Affine Invariants

In this section, for completeness sake, we briefly review a few basic concepts used in our approach.

2.1 Notation

$\mathbf{x}_{it}$: image coordinates of feature i at time t; when the feature label is clear from context, we drop the index i and simply use $\mathbf{x}_t$
$\mathbf{x}_{it}^{(pqr)}$: affine coordinates of feature i wrt affine coordinate frame defined by features p, q, r at time t.
$|| \cdot ||_*$: the nuclear norm.

2.2 Autoregressive Models and Hankel Matrices

Consider a vector dynamical process described by an n^{th} order autoregressive model of the form:

$$\mathbf{x}_k = a_1\mathbf{x}_{k-1} + a_2\mathbf{x}_{k-2} + \ldots + a_n\mathbf{x}_{k-n} \tag{1}$$

where $\mathbf{x}_k \in \mathbf{R^d}$. To every trajectory $\mathbf{x}_k$ of this model, one can associate a Hankel matrix defined as:

$$H_x^{s,r} \doteq \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_r \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_s & \mathbf{x}_{s+1} & \cdots & \mathbf{x}_{r+s-1} \end{bmatrix} \tag{2}$$

Note that the columns of the Hankel matrix correspond to overlapping subsequences of the trajectory, shifted by one, and that the block anti-diagonals of the matrix are constant. This special structure encapsulates the dynamic information of the data. In particular, a well known result from realization theory [13,14] is that, under mild conditions, the rank of the Hankel matrix is the order n of the model – i.e. rank$(H_{\mathbf{x}}^{s,r}) = n$ provided that $r, s \geq n$. Thus, the rank of this matrix measures the complexity of the underlying dynamics of a given sequence of measurements. It is easy to see that the coefficients a_i, $i = 1, \ldots, n$ of the autoregressive model (1) satisfy

$$H_x^{s,n+1} \begin{bmatrix} a_n \ldots a_1 \ -1 \end{bmatrix}^T = 0 \tag{3}$$

Moreover, it was recently shown [11] that the coefficients a_i, $i = 1, \ldots, n$ of the autoregressive model (1) are invariant under affine transformations.

2.3 Affine Coordinates

Three noncollinear points $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$, define an affine coordinate frame in which any point $\mathbf{P}$ coplanar with $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ can be expressed as

$$\mathbf{P} = \mathbf{M}_1 + \alpha(\mathbf{M}_2 - \mathbf{M}_1) + \beta(\mathbf{M}_3 - \mathbf{M}_1) \tag{4}$$

where $\mathbf{P}^{(\mathbf{M}_1,\mathbf{M}_2,\mathbf{M}_3)} \doteq (\alpha, \beta)$ are the affine coordinates of $\mathbf{P}$ with respect to the reference frame $(\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3)$. It is well known, that affine coordinates are invariant to affine transformations, i.e.

$$\mathbf{p}^{(\mathbf{m}_1,\mathbf{m}_2,\mathbf{m}_3)} = \mathbf{P}^{(\mathbf{M}_1,\mathbf{M}_2,\mathbf{M}_3)}$$

where $\mathbf{m}_i$ is the affine transformation of $\mathbf{M}_i$, $i = 1, 2, 3$, respectively:

$$\begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} A & \mathbf{t} \\ \hline 0\ 0 & 1 \end{array}\right] \begin{bmatrix} \mathbf{M}_i \\ 1 \end{bmatrix},$$

A is a 2×2 non-singular matrix and $\mathbf{t}$ is a translation vector.

3 A Robust Supporter Tracking Algorithm

In this section, we introduce a novel Robust Supporter Tracking (RST) algorithm. The main idea is to use dynamics based support from context features to accurately estimate the location of the target while it is occluded. We do this by relating the location of the target to multiple affine reference frames defined by context features as explained below. Then, multiple estimates consistent with the relative dynamics between the target and each of the reference frames are combined through a voting scheme, where estimations generated using more dynamically correlated features are trusted more and hence given more weight. The complete procedure is given in Algorithm 1 and the details are explained below.

Algorithm 1. Robust Supporter Tracking Algorithm

```
while run do
    detect and track features by matching
    if target is invisible then
        For each feature, build the feature triplets that satisfy (Eq. 13) and (Eq. 14)
        Compute $||H_{\Delta\mathbf{x}^{(ijk)}}||_*$ for each triplet
        Sort the triplets by ascending $||H_{\Delta\mathbf{x}^{(ijk)}}||_*$
        Select the first $N$ feature triplets with smallest $||H_{\Delta\mathbf{x}^{(ijk)}}||_*$ to build the voting triplet supporter set $S$.
        for each triplet in $S$ do
            Estimate $\Delta\hat{\mathbf{x}}_t^{(ijk)}$ (Eq. 8) or (Eq. 9)
            Compute $\hat{\mathbf{x}}_t^{(ijk)}$ (Eq. 5)
        end for
        Compute $p(\mathbf{x}_t|I_t)$ (Eq. 11)
        Predict the target position (Eq. 12)
    end if
end while
```

3.1 Local Autoregressive Dynamic Models

The position of the target is estimated based on its relative motion with respect to affine reference frames which are defined by triplets of context features as described next.

Start by considering an affine reference frame in the 3D scene defined by a triplet of non collinear points $(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k)$. Assuming that the depth of the scene is small compared to the distance between the camera and the target, express the location of the target in this reference frame at time t:

$$\mathbf{X}_t^{(ijk)} = (\alpha_t^{(ijk)}, \beta_t^{(ijk)})$$

Next, model the motion of the target with respect to the affine reference frame by using an autoregressive model of the form in (1):

$$\Delta\hat{\mathbf{X}}_t^{(ijk)} = \sum_{f=1}^{n} a_f^{(ijk)} \Delta\hat{\mathbf{X}}_{(t-f)}^{(ijk)}$$

where $\Delta\hat{\mathbf{X}}_t^{(ijk)} = \hat{\mathbf{X}}_t^{(ijk)} - \hat{\mathbf{X}}_{(t-1)}^{(ijk)}$ is the noiseless velocity of the target at time t, expressed in the reference frame $(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k)_t$. Since both affine coordinates and the regressor coefficients are invariant to affine projections, we have that the same equations are true for the projections of the target and the reference points, as illustrated in Figure 4:

$$\mathbf{x}_t^{(ijk)} = (\alpha_t^{(ijk)}, \beta_t^{(ijk)})$$

$$\Delta\hat{\mathbf{x}}_t^{(ijk)} = \sum_{f=1}^{n} a_f^{(ijk)} \Delta\hat{\mathbf{x}}_{(t-f)}^{(ijk)}$$

where $\Delta\hat{\mathbf{x}}_t^{(ijk)} = \hat{\mathbf{x}}_t^{(ijk)} - \hat{\mathbf{x}}_{(t-1)}^{(ijk)}$ and $\mathbf{x}_t^{(ijk)} = \hat{\mathbf{x}}_t^{(ijk)} + \mathbf{w}_t$ where $\mathbf{w}_t$ is measurement noise.

Then, the location of the target at time t, with respect to the affine reference frame $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$

$$\hat{\mathbf{x}}_t^{(ijk)} = \hat{\mathbf{x}}_{(t-1)}^{(ijk)} + \Delta\hat{\mathbf{x}}_t^{(ijk)} \tag{5}$$

can be estimated by enforcing that the data available so far should be explained by the lowest order autoregressive model [1]. That is, an estimate of the velocity $\Delta\hat{\mathbf{x}}_t^{ijk}$ can be found by minimizing the rank of the Hankel matrix of the available measurements with respect to the measurement noise and the future measurement:

$$\Delta\hat{\mathbf{x}}_t^{(ijk)} = \arg \min_{\Delta\mathbf{w}, \Delta\hat{\mathbf{x}}_t^{(ijk)}} \operatorname{rank}(H_{\Delta\mathbf{x}^{(ijk)}}) \tag{6}$$

where

$$H_{\Delta\mathbf{x}^{(ijk)}} \doteq \begin{bmatrix} \Delta\mathbf{x}_1^{(ijk)} & \Delta\mathbf{x}_2^{(ijk)} & \cdots & \Delta\mathbf{x}_c^{(ijk)} \\ \Delta\mathbf{x}_2^{(ijk)} & \Delta\mathbf{x}_3^{(ijk)} & \cdots & \Delta\mathbf{x}_{c+1}^{(ijk)} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta\mathbf{x}_{t-c+1}^{(ijk)} & \Delta\mathbf{x}_{t-c}^{(ijk)} & \cdots & \Delta\hat{\mathbf{x}}_t^{(ijk)} \end{bmatrix}$$

where c is chosen so that the Hankel matrix is as square as possible using all the available measurements.

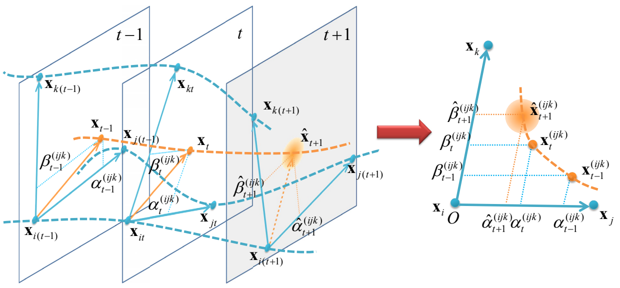


Fig. 4. The trajectory of the target in the image space is expressed relative to an affine coordinate system defined by a triplet of features (orange dash line on the right)

A problem with this approach is that rank minimization is an NP-hard problem. Fortunately, there exist good convex relaxations to this problem [15]. For example, one can solve instead

$$\begin{array}{ll} & \min \text{ trace}(P) + \text{trace}(Q) \\ s.t. & \begin{bmatrix} P & H_{\Delta\mathbf{x}} \\ H_{\Delta\mathbf{x}}^T & Q \end{bmatrix} \geq 0 \\ & \|\Delta\mathbf{w}\|_2 < T_{noise} \\ & \Delta\mathbf{x} = \Delta\hat{\mathbf{x}} + \Delta\mathbf{w} \end{array} \tag{7}$$

where the superscripts (ijk) were dropped for clarity. A further improvement on the solution to this problem can be obtained by using a method based on a re-weighted heuristic that seeks to iteratively solve the following Semi-Definite Problem [16].

$$\begin{array}{ll} & \min \text{ trace}((P_k + \delta I)^{-1} P_{k+1}) + \text{trace}((Q_k + \delta I)^{-1} Q_{k+1}) \\ s.t. & \begin{bmatrix} P_{k+1} & H_{\Delta\mathbf{x}} \\ H_{\Delta\mathbf{x}}^T & Q_{k+1} \end{bmatrix} \geq 0 \\ & \|\Delta\mathbf{w}\|_2 < T_{noise} \\ & \Delta\mathbf{x} = \Delta\hat{\mathbf{x}} + \Delta\mathbf{w} \end{array} \tag{8}$$

Alternatively, one can estimate $\Delta\hat{\mathbf{x}}_t^{ijk}$ using the regressor

$$\Delta\hat{\mathbf{x}}_t = a_1 \Delta\mathbf{x}_{t-1} + a_2 \Delta\mathbf{x}_{t-2} + \ldots + a_n \Delta\mathbf{x}_{t-n} \tag{9}$$

where the coefficients $\mathbf{a} = \begin{bmatrix} a_n \ldots a_1 \end{bmatrix}$ can be estimated from (3) using the previous measurements and total least square error minimization

$$\hat{\mathbf{a}} = -V_{12} V_{22}^{-1}$$

where $H_{\Delta\mathbf{x}}^{t-n-1,n+1} = U \Sigma V^T$ is the singular value decomposition of the Hankel matrix $H_{\Delta x}^{t-n-1,n+1}$ formed with the last $t-1$ measurements,

$$V := \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix},$$

V_{11} is $n \times n$, V_{22} is a scalar, and n is the order of the regressor[2] which can be easily estimated by computing an SVD of the Hankel matrix. Thus, the computational cost of estimating the relative velocity of the target is reduced to the cost of computing two small SVDs and a matrix multiplication.

3.2 Vote by Rank Minimization Estimates

Estimates of the target location from different feature triplets are combined through a Generalized Hough Transform (GHT) voting scheme. Naturally, we prefer to trust estimates computed from a triplet of features whose motion is most correlated to the motion of the target itself. Thus, the stronger the correlation between their motions, the higher voting weight these supporters are assigned. Motion correlation is measured in

[2] As long as there are at least $2n$ measurements available.

terms of the order of the auto regressive model that is needed to explain the motion of the target with respect to the triplet, which, in turn, is estimated by the nuclear norm of the corresponding Hankel matrix, $||H_{\Delta \mathbf{x}^{ijk}}||_*$ as a surrogate of rank. Then, we can formulate a voting scheme with a single Gaussian Model as follows.

$$p(\mathbf{x}_t|(\mathbf{x}_{it}, \mathbf{x}_{jt}, \mathbf{x}_{kt})) \sim \frac{1}{||H_{\Delta \mathbf{x}^{(ijk)}}||_*} N(\mathbf{x}_t|\hat{\mathbf{x}}_t^{(ijk)}, \Sigma) \tag{10}$$

where $\Sigma = \sigma I$, and σ is a constant. Then, combining the prediction from all supporter triplets, we have the probability density function of the target position

$$p(\mathbf{x}_t|I_t) = \sum_{i \in S} p(\mathbf{x}_t|(\mathbf{x}_{it}, \mathbf{x}_{jt}, \mathbf{x}_{kt}))p((\mathbf{x}_{it}, \mathbf{x}_{jt}, \mathbf{x}_{kt})|I_t) \tag{11}$$

where I_t is the t^{th} frame image and S is the voting triplet feature set. Finally, the prediction result is given by

$$\hat{\mathbf{x}}_t = \arg \max_{\mathbf{x}_t} p(\mathbf{x}_t|I_t) \tag{12}$$

4 Implementation Details

Any tracker can be used while the target is visible. For simplicity, we used a KLT tracker to track the target. Then, occlusion is detected when the KLT template matching fails and the RST algorithm begins to estimate the position of the occluded target.

Any reliable feature such as Scale Invariant Feature Transform (SIFT) [17], Harris corner, etc can be used as context features. In our implementation, we use SIFT features. First, SIFT features are extracted from the first frame and are used to initialize a feature set. Then, the features in the present frame are matched against the ones in the feature set by the function provided in [18]. A matching score lower than $T_{match} = 10^4$ indicates that the corresponding feature has been tracked in the present frame, otherwise the feature is considered lost. After matching, the present positions of the tracked features are added to their trajectories and the unmatched SIFT features in the present frame are added to the feature set as new detected features. We used 20 frames long supporter trajectories and hence we can handle relative dynamics of up to order 10. Typically, there are between 20 to 30 such features per frame and $N \leq 5$ triplets are used.

Noise on the position of the features used as reference triplets affects the values of the affine coordinates of the target with respect to the affine coordinate system. Thus, to reduce the effect of noise, features for a triplet are selected according to the following three rules:

1. The distance between pairs of features must be greater than a threshold $T_{dist} = 20$.
2. The angle between the vectors $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_k - \mathbf{x}_i$ should be as close to 90 degrees as possible.
3. Triplets that move as a rigid should be preferred. Thus, following [19], the order of the dynamics of $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_k - \mathbf{x}_i$ - i.e. the rank of the corresponding Hankel matrices or its convex envelope, the nuclear norm, should be as low as possible.

In summary, given a set of candidate features $\mathcal{F}$ and a feature $\mathbf{x}_i \in \mathcal{F}$, the other two points in the reference triplet $\mathbf{x}_j$ and $\mathbf{x}_k$ are selected from $\mathcal{F}$ by comparing the nuclear norm of the Hankel matrices of their differences:

$$||H_{\mathbf{x}_j - \mathbf{x}_i}||_* \leq ||H_{\mathbf{x}-\mathbf{x_i}}||_* \forall \mathbf{x} \in \mathcal{F} \text{ such that } ||\mathbf{x} - \mathbf{x}_i||_2 > T_{\text{dist}} \tag{13}$$

and

$$||H_{\mathbf{x}_k - \mathbf{x}_i}||_* \leq ||H_{\mathbf{x}-\mathbf{x}_i}||_* \forall \mathbf{x} \in \mathcal{F} \text{ such that } ||\mathbf{x} - \mathbf{x}_i||_2 > T_{\text{dist}} \text{ and } |\cos(\phi^{(ijk)})| < T_\phi \tag{14}$$

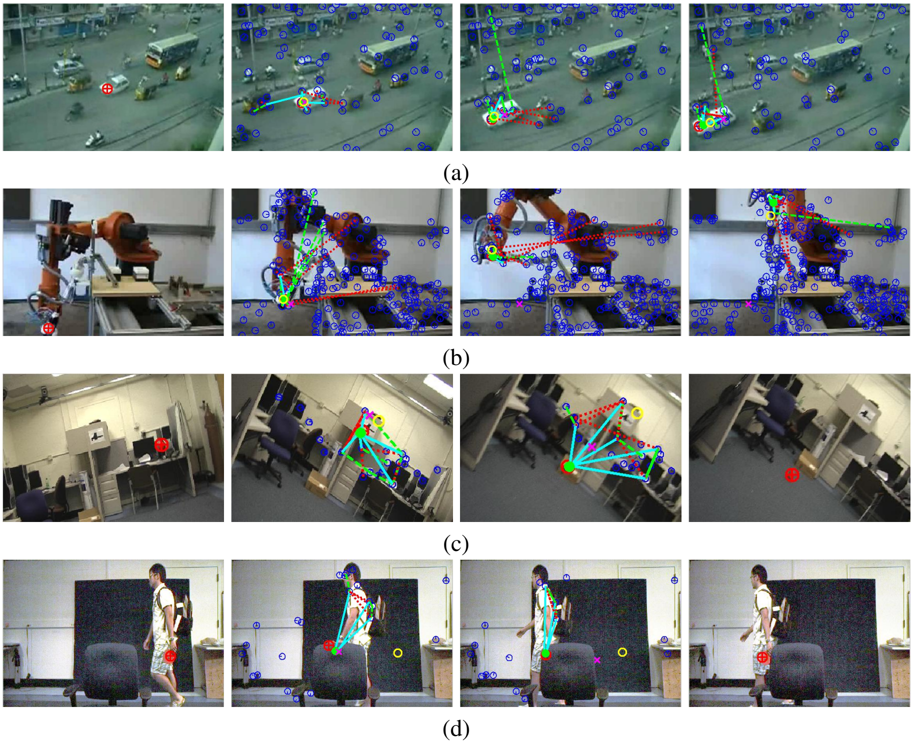
where $\cos \phi^{(ijk)} = \frac{(\mathbf{x}_j - \mathbf{x}_i)^T . (\mathbf{x}_k - \mathbf{x}_i)}{||\mathbf{x}_j - \mathbf{x}_i|| ||\mathbf{x}_k - \mathbf{x}_i||}$ and T_{dist} and T_ϕ are thresholds.

5 Experiments

In this section, we demonstrate the advantage of the proposed algorithm. First, we show that our algorithm improves the accuracy of the estimated target location remarkably. Second, the advantage of using adaptive dynamics model is presented. Finally, it is shown that by exploiting the supporter triplet coordinate system the proposed algorithm is robust to severe camera motion and long occlusions. As illustrated in Figure 1, using context features [3] performs better than using context regions [2]. Hence, in our experiments we compare the performance with the Supporter Tracker (ST) [3] and the Hankel-based Robust Tracker (RT) [1]. In addition to the video shown in Figure 1, we show results for seven videos. Four of these videos have ground-truth on where the target is located and were used to establish quantitative measures of the performance of the three algorithms being compared. The remaining three videos are increasingly challenging and show that the proposed algorithm performs similarly or better than ST when rigidity between the target and supporters exists and that it is more robust and accurate than the state of art approaches in the presence of severe camera motion and prolonged occlusions.

5.1 Estimation Accuracy

To measure the estimation accuracy, the algorithms are tested with the videos where the ground truth is available. In three of the test videos (India, Beer Robot, Bouncing Ball), there is no occlusion and the ground truth of the target position is obtained by KLT and occlusion is simulated. In the swinging racket video, the ground truth during the occlusion was obtained by using a VICON system to track a label attached on the table tennis racket. In the short-term estimation case, the estimation is always made upon the past ground truth. In the long-term estimation, the estimation is made upon the past estimations and the ground truth before estimations. Sample frames for the long term occlusion cases are shown in Figure 5. The means of the estimation errors for short and long term occlusion are listed in Table 1a and 1b and visualized in Figures 6 and 7, respectively. As seen there, the proposed algorithm gives consistently more accurate estimations, and it is significantly better when there are long occlusions.



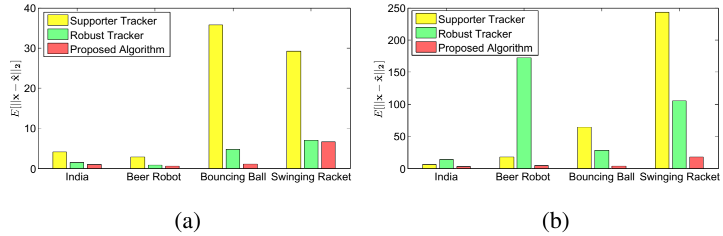
(a)

(b)

(c)

(d)

Fig. 5. Sample frames with known ground truth data from sequences: (a) India traffic (occlusion = 21frs.); (b) Beer Robot (occlusion = 125frs.); (c) Bouncing ball (occlusion = 12frs.); and (d) Swinging racket (occlusion = 13frs.). We mark the tracked target with a red crossed circle, the RST estimation with a green dot, the ST estimation with a yellow circle, and the RT estimation with a magenta cross. The cyan lines point to where the RST features vote and the green and red dash lines are the affine coordinate basis of each supporter. Also, the sequentially tracked features are marked as blue circles.



(a)

(b)

Fig. 6. Visualization of the estimation error. ST is short for Supporter Tracker [3]; RT is short for Robust Tracker(RT) [1]; Robust Supporter Tracker (RST) is proposed algorithm. (a) Short term occlusion. (b) Long term occlusion.

Table 1. Short-term and long-term estimation. The mean of $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2$ in the occlusion. Length of occlusion in frames, given in parenthesis. (SR is short for Swing Racket; BB is short for Bouncing Ball).

(a) Short-term estimation. The mean of $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2$ in the first frame of occlusion.

	India (21 frs)	Beer Robot (125 frs)	BB (12 frs)	SR (13 frs)
ST	4.1384	2.8408	35.7987	29.1990
RT	1.5077	0.7924	4.7730	6.9537
RST	0.9781	0.5645	1.0438	6.6842

(b) Long-term estimation.The mean of $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2$ during entire occlusion.

	India (21 frs)	Beer Robot (125 frs)	BB (12 frs)	SR (13 frs)
ST	6.1147	17.4931	63.9240	243.0031
RT	13.5030	171.9341	27.6450	105.2612
RST	2.4778	3.9857	3.9303	17.9683

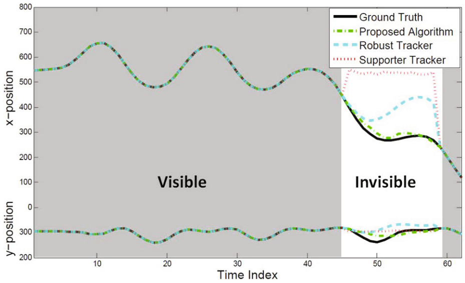


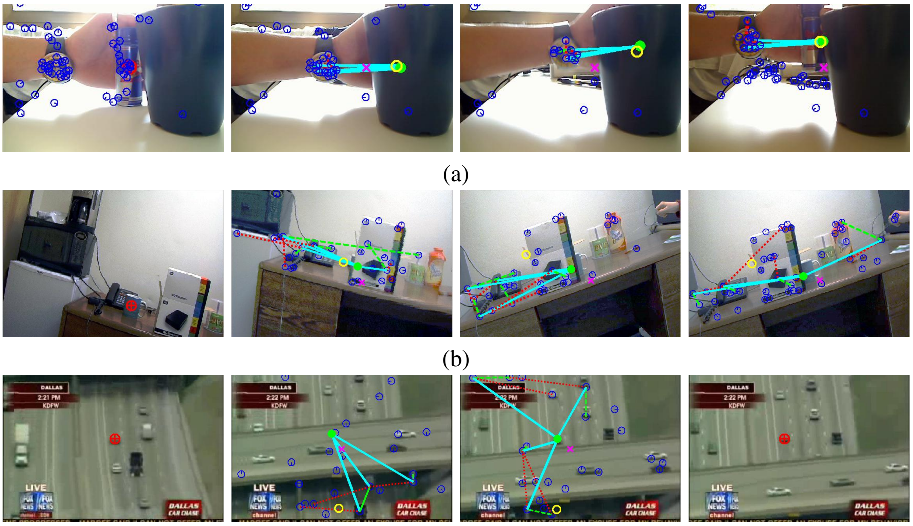
Fig. 7. True position of the pingpong racket and long term estimates.

5.2 Estimation by Nonrigidly Coupled Features

With enough sequential measurements, the dynamical model that we use can adapt to slowly changing linear dynamics. Thus, our algorithm can use supporting features to estimate the position of the target, even if they are not rigidly correlated with the target. This feature is illustrated with four examples. In Figures 1e and 8, it is shown that our algorithm correctly estimates the position of the ping-pong balls and the cup using stationary background supporters, even in the presence of severe camera rotation. In Figure 5d, we show a few frames of the swinging racket video, where our algorithm can model and estimate the swinging table tennis racket, using features on different body parts of the person holding the racket, while the other trackers fail. Finally, Figure 8c shows frames from a video of a police car chasing another vehicle, where our tracker uses supporters from the background and other independent moving objects to accurately estimate the location of the occluded car being chased by the police vehicle.

5.3 Robustness to Camera Motion

By combining the triplet coordinate system and the adaptive dynamic model, our algorithm is more robust to severe camera motion. This is clearly seen in the examples shown in Figures 1 and 8b where the camera rotates wildly. Another example is shown in Figure 5c, where our tracker follows a bouncing ball through long occlusions under



(a)

(b)

(c)

Fig. 8. Sample frames of increasingly challenging videos. (a) The ETH moving cup and supporters move almost rigidly. (b) A moving cup observed by a swinging camera and (c) Chasing car with moving camera and very long occlusion. We mark the tracked target with a red crossed circle, the RST estimation with a green dot, the ST estimation with a yellow circle, and the RT estimation with a magenta cross. The cyan lines point to where the RST features vote and the green and red dash lines are the affine coordinate basis of each supporter. Also, the sequentially tracked features are marked as blue circles.

irregular camera motion while the other trackers fail due to the complex motions of the target and the camera.

6 Conclusion

In this paper we presented a novel tracking algorithm that uses context features and systems dynamics to estimate the location of a target in the presence of long occlusions and camera motion. The algorithm does not assume a priori information about the motion of the target or the supporter features. Supporter features can move with the target, independently of the target, or not at all. The algorithm includes a mechanism to weight the reliability of the estimations by quantifying the extent of motion correlation between the target and the supporter features from the available measurements. The algorithm was tested and compared against other context based trackers and a dynamics based tracker using several challenging videos. In all cases, the performance of the proposed tracker was superior.

References

1. Ding, T., Sznaier, M., Camps, O.: Receding horizon rank minimization based estimation with applications to visual tracking. In: CDC, pp. 3446–3451 (2008) 2, 4, 7, 10, 12

2. Yang, M., Wu, Y., Hua, G.: Context-aware visual tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 1195–1209 (2009) 2, 3, 4, 10
3. Grabner, H., Matas, J., Van Gool, L., Cattin, P.: Tracking the invisible: Learning where the object might be. In: CVPR, pp. 1285–1292 (2010) 2, 3, 4, 10, 12
4. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR (2008) 1
5. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009) 1
6. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR (2011) 1
7. Beymer, D., Konolige, K.: Real-time tracking of multiple people using continuous detection. In: ICCV (1999) 1
8. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. IJCV 29, 5–28 (1998) 1
9. Julier, S., Uhlmann, J., Durrant-Whyte, H.: A new approach for filtering nonlinear systems. In: ACC, vol. 3, pp. 1628–1632 (1995) 1
10. North, B., Blake, A., Isard, M., Rittscher, J.: Learning and classification of complex dynamics. IEEE Trans. PAMI 22 (2000) 1
11. Ayazoglu, M., Li, B., Dicle, C., Sznaier, M., Camps, O.: Dynamic subspace-based coordinated multicamera tracking. In: ICCV (2011) 2, 4, 5
12. Cerman, L., Matas, J., Hlaváč, V.: Sputnik Tracker: Having a Companion Improves Robustness of the Tracker. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 291–300. Springer, Heidelberg (2009) 2
13. Ho, B., Kalman, R.: Effective construction of linear, state-variable models from input/output functions. Regelungstechnik 14, 545–548 (1966) 5
14. Moonen, M., Moor, B.D., Vandenberghe, L., Vandewalle, J.: On- and off-line identification of linear state space models. Int. J. of Control 49, 219–232 (1989) 5
15. Fazel, M., Hindi, H., Boyd, S.: A rank minimization heuristic with application to minimum order system approximation. In: ACC, vol. 6, pp. 4734–4739 (2001) 8
16. Fazel, M., Hindi, H., Boyd, S.: Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In: ACC, vol. 3, pp. 2156–2162 (2003) 8
17. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV (1999) 9
18. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008) 9, `http://www.vlfeat.org/`
19. Lublinerman, R., Sznaier, M., Camps, O.: Dynamics based robust motion segmentation. In: CVPR, vol. 1, pp. 1176–1184 (2006) 10