Visual Tracking via Subspace Motion Model

Jun Wang[1]
jwang1004@gmail.com
Fan Zhong[1]
zhongfan@sdu.edu.cn
Guofeng Wang[1]
wangguofeng13@163.com
Qunsheng Peng[2]
peng@cad.zju.edu.cn
Xueying Qin[1]
qxy@sdu.edu.cn(Correspondence)

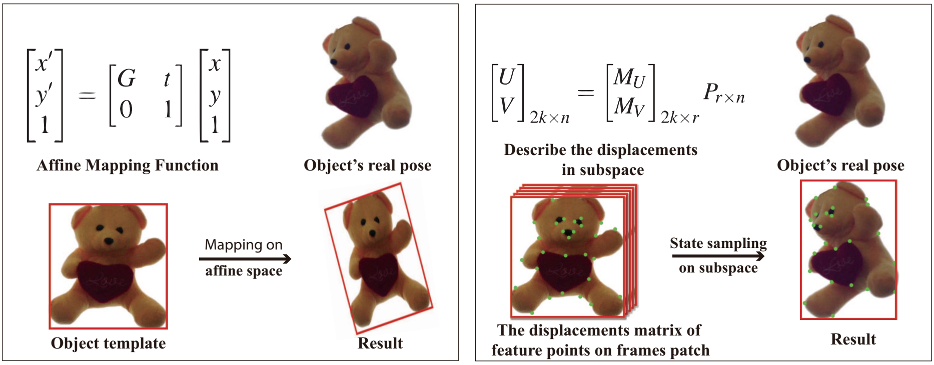
[1] School of Computer Science and Technology, Shandong University, Jinan, China
[2] The State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

Abstract

In this paper we propose a novel motion model for visual tracking, which can be used to parameterize instantaneous image motion caused by both object and camera movements. Our approach is inspired by the subspace theory of image motion, that is, for a rigid object imaged by a projective camera, the displacements matrix of its trajectories over a short period of time should approximately lie in a low-dimensional subspace with a certain rank upper bound. We adopt this subspace as the state transition space in particle filtering, by which we can obtain a state vector with alterable number of dimensions. The dimension number as well as the sampling weight for each dimension at each moment can be determined by the rank of the subspace. In this way the particle distribution will be more coincide with the probability of the object state. Based on the subspace motion model, we derive a new visual tracking approach that can handle more complicated cases. The subspace is also used to discriminate new feature points of the object for adaptively updating the motion model. Experimental results and comparisons demonstrate the effectiveness of the proposed method.

1 Introduction

Visual tracking, in general, is a very important and challenging task within the field of computer vision. The state of this art has been widely studied in the past decades [21, 22]. While most of researches focus on exploring new methods to represent object appearance [20, 23, 24], little attention has been paid on the description of object motion. In this paper, we propose a novel motion model that can better parameterize object motion between adjacent frames. Our method represents object motion in the subspace of the displacements matrix of object appearance trajectories, which is shown to be more consistent with the situation in the real world. Based on the proposed motion model, the methods based on particle filtering can


$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} G & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
Affine Mapping Function
Object's real pose
Mapping on
affine space
Object template
Result
$\begin{bmatrix} U \\ V \end{bmatrix}_{2k\times n} = \begin{bmatrix} M_U \\ M_V \end{bmatrix}_{2k\times r} P_{r\times n}$
Describe the displacements in subspace
Object's real pose
State sampling
on subspace
The displacements matrix of feature points on frames patch
Result

(a) affine model

(b) subspace model

Figure 1: (a) The affine model fails when image deformation happens brought by motion of a non-plane-like object. (b) The subspace model sample particles on the subspace with alterable dimensionality, and can build a point-to-point relation between images.

be improved by sampling in the subspace of object motion, which produces great tracking results on several tough sequences.

The motion model of visual tracking is used to predict the likely state of an object in successive frames. Due to the great success of Particle Filtering (PF) [1, 10, 13], which is also known as sequential Monte Carlo methods, visual tracking has been formulated as a problem of Bayesian inference in the state space. The performance of this technique greatly depends on the choice of state transition space. The common approach is to sample in the vector space of the affine transformation under a set of local image coordinates. As shown in Figure 1:(a), the model uses an affine matrix to map every pixel within the object area of a frame to a new position in the next frame. A recently developed motion model [14] regards the set of affine transformation as a curved space possessing the structure of a Lie group. The affine model works well when dealing with the projection of plane-like objects (e.g. book cover, license plate). However, when objects are not plane-like, as the most probable situations in the real world, object motion will naturally bring image deformation, which may violate the relative relationship of projected pixels in affine space. In this case, applying the same transition vector to all pixels as affine model implies is obviously improper. Another problem of affine model is that, it is hard to determine the weight of each dimension in the affine state vector during the tracking process, so the prediction stage of PF is guideless and may need much more particles to cover the possible object motion state.

An ideal motion model should have the ability to describe the instantaneous motion of a rigid object caused by both object and camera movements, and meantime should be able to estimate the sampling weight of each dimension of the state vector. Our key insight is that we can build such a model by leveraging the well-known subspace theory of motion, which tells us that a matrix of motion trajectories' displacements of a rigid object imaged by a moving camera over a short period of time should approximately lie in a low-dimensional subspace [12, 18]. A displacements matrix is composed of feature points displacements of the object relative to a reference frame in a short period of time. In particular, for instantaneous motions the subspace should be ranked at most 9, while in real situations the rank is usually smaller [12]. This important theory has been used to help solve a number of problems in computer vision, including structure from motion [7, 18], correspondence estimation [12], motion segmentation [19], and video stabilization [15]. Under this subspace constraint, the displacements matrix can be factored into a product of a frame-dependent matrix and a point-dependent matrix. If all trajectories are complete, the displacements matrix is complete and the factorization process is linear, fast, and robust. For an incomplete matrix, factorization is a nonlinear problem, which is well-studied [4, 5, 6] and also can be done stably.

We introduce the subspace motion model for visual tracking by adopting the subspace of object motion as the state transition space in PF. The brief idea is illustrated in Figure 1:(b). By sampling in the motion subspace, the particle distribution will be more coincide with the probability of potential object state, because the dimensionality of PF state vector and the weights for each dimension are calculated dynamically by the object motion complexity at each moment. We use the rank of the subspace to measure the complexity. The rank can be computed automatically by the singular values of the displacements matrix. Besides, the subspace motion model has the capability to naturally represent the disparity brought by object or camera rotation via the correspondence of feature points between tracked frames, as each particle in the subspace corresponds to a group of points in the current frame. The newly detected feature points in each frame can be easily identified to belong to the object or not, this can achieved by supervising the variation of the subspace rank, and a new point should belong to the object if it lies on the same subspace with existing object points. Based on the subspace motion model, we introduce a complete tracking method with a feature-points-based appearance description, while theoretically the proposed motion model is compatible with any appearance model.

2 Subspace Theory

The most common situation in visual tracking is to work with a perspective camera. In general, the motion trajectories from a perspective camera will lie on a nonlinear manifold instead of a linear subspace [8]. However, it is possible to approximate the manifold locally (over a short period of time) with a linear subspace.

We first compute the displacements matrix of the object over the frames patch at time t. The set of 2D feature trajectories is tracked by the standard KLT approach [17]. Let I_j denote a frame in the sequence, a frames patch includes k successive frames contiguous before I_t, i.e. frame I_{t-k} to I_{t-1}. The patch will update with time by adding the newly tracked frame and deleting the oldest frame. All frames contain n continuous feature points within the object area. Let (u_{ij}, v_{ij}) denote the displacement of point (x_i, y_i) from the reference frame I (e.g. the first frame in the frames patch) to frame I_j $(i = 1 \ldots n, j = 1 \ldots k)$. Let U and V denote two $k \times n$ matrices constructed from the displacements of all the points across the patch, the displacements matrix is built by:

$$U = \begin{bmatrix} u_{11} & u_{22} & \ldots & u_{n1} \\ u_{12} & u_{22} & \ldots & u_{n2} \\ & & \vdots & \\ u_{1k} & u_{2k} & \ldots & u_{nk} \end{bmatrix}, V = \begin{bmatrix} v_{11} & v_{22} & \ldots & v_{n1} \\ v_{12} & v_{22} & \ldots & v_{n2} \\ & & \vdots & \\ v_{1k} & v_{2k} & \ldots & v_{nk} \end{bmatrix} \tag{1}$$

Each row in the matrices of Equation (1) corresponds to a single frame, and each column corresponds to a single point.

According to the subspace theory, although the matrices U and V are large, their ranks are small. This low-rank constraint will be exploited to build the PF state transit space.

Particularly, we stack U and V in Equation (1) vertically as $[\frac{U}{V}]_{2k\times n}$. It is worth noting that, in this situation, we need to meet $k \geq 5$ and $n \geq 9$ to make sure the scale of the displacement matrix beyond the subspace rank upper bound. Fortunately, for our application that k is small, there should be a reasonable number of complete trajectories that already span all k frames and describe the subspace.

2.1 Subspace Decomposition

Under the perspective camera model, Irani [12] showed that in the instantaneous case, when the rotation angle is small and the forward translation is small, the 2D displacement (u_{ij}, v_{ij}) of (x_{ij}, y_{ij}) between frame I_j and reference frame I can be rewritten as a bilinear product:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}_{2\times 1} = \begin{bmatrix} (M_U)_j \\ (M_V)_j \end{bmatrix}_{2\times 9} P_{i(9\times 1)} \tag{2}$$

where P_i is a point-dependent column vector ($i = 1 \ldots n$) involving only points positions parameters, while $(M_U)_j$ and $(M_V)_j$ are frame-dependent row vectors ($j = 1 \ldots k$) involving only camera motion parameters.

Therefore, the displacements matrix of all points across all frames can be expressed as a bilinear product of matrices:

$$\begin{bmatrix} U \\ V \end{bmatrix}_{2k\times n} = \begin{bmatrix} M_U \\ M_V \end{bmatrix}_{2k\times r} P_{(r\times n)} \tag{3}$$

where the upper bound of rank r is 9.

By employing the theory of Equation (3), we may factorize the displacements matrix to get the frame-dependent matrix $[\frac{M_U}{M_V}]_{2k\times r}$ and the point-dependent matrix $P_{r\times n}$. We do this subspace decomposition in the displacements matrix of every frames patch. The decomposition is calculated by Singular Value Decomposition (SVD) [9]. We truncate the out put of SVD to the rows, columns, and values corresponding to the largest r singular values, and then distributing the square root of each singular value to the left (frames) and right (points) matrices.

2.2 Subspace Rank Detection

In practice, due to the degenerate camera motions or degenerate scene structures, the actual rank of the subspace matrices is often much smaller than the theoretical upper bound. We detect the actual rank of the subspace automatically during the process of subspace decomposition.

Let $[\frac{U}{V}]$ be the $2k \times n$ displacements matrix of the frames patch with the actual rank r and rank upper bound r'. The rank reduction is done with the singular values calculated by SVD, which is the same process to Section 2.1. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$ be the singular values of $[\frac{U}{V}]$, where $m = \min(2k, n)$. We check for the existence of a lower rank r'' such that:

$$\| \frac{\lambda_{r''+1}}{\lambda_1} \| < \varepsilon \tag{4}$$

where ε is the threshold set for some noise tolerance. We usually choose $\varepsilon \approx 1\%$.

We set r to be $r = \min(r', r'')$ and all singular values other than the r largest ones to be zero (i.e. $\lambda_{r+1} = \lambda_{r+2} = \ldots = \lambda_m = 0$). The matrices produced from the SVD (with the new

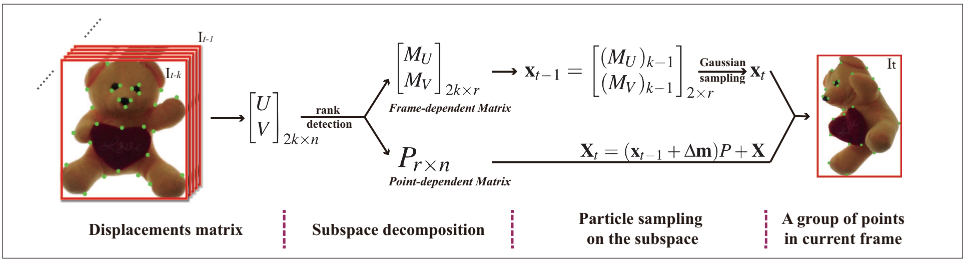

I_{t-1}
I_{t-k}
$\begin{bmatrix} U \\ V \end{bmatrix}_{2k\times n}$
rank detection
$\begin{bmatrix} M_U \\ M_V \end{bmatrix}_{2k\times r}$
Frame-dependent Matrix
$\rightarrow \mathbf{x}_{t-1} = \begin{bmatrix} (M_U)_{k-1} \\ (M_V)_{k-1} \end{bmatrix}_{2\times r}$
Gaussian sampling
$\mathbf{x}_t$
$P_{r\times n}$
Point-dependent Matrix
$\mathbf{X}_t = (\mathbf{x}_{t-1} + \Delta\mathbf{m})P + \mathbf{X}$
It
Displacements matrix
Subspace decomposition
Particle sampling on the subspace
A group of points in current frame

Figure 2: Flow chart of the subspace motion model

singular values) then can be truncated and recomposed, yielding a new displacements matrix of rank r. This new matrix is closest to $[\frac{U}{V}]$ in the Frobenius norm.

3 Tracking via Subspace Motion Model

We now explain in detail our visual tracking approach based on subspace motion model in this section. We start with a detailed description of the subspace motion model, the flow chart of which is shown in Figure 2. We then build the complete tracking process and introduce the method to update the displacements matrix during the tracking process.

3.1 Subspace Motion Model

After we have obtained the frame-dependent matrix $[\frac{M_U}{M_V}]_{2k\times r}$ and the point-dependent matrix $P_{r\times n}$ of the frames patch with a certain rank upper bound, we may now build the subspace motion model of visual tracking by sampling particles on subspace. As the motions of camera and object are relative, we may always deem a moving camera with a static object. Then the state transition may be considered only happen in $[\frac{M_U}{M_V}]_{2k\times r}$, while $P_{r\times n}$ remain the same. So the initial state of the PF can be described in subspace as:

$$\mathbf{x}_{t-1} = \begin{bmatrix} (M_U)_{k-1} \\ (M_V)_{k-1} \end{bmatrix}_{2\times r} \tag{5}$$

where $(M_U)_{k-1}$ and $(M_V)_{k-1}$ are the last rows of M_U and M_V respectively, corresponding to the frame-dependent vectors of last frame I_{t-1}.

In the prediction stage of PF, the samples in the state space are propagated through a dynamic model, which is usually an autoregressive process (AR). Here we use a first-order AR model for fair comparison and simplicity:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \Delta\mathbf{m} \tag{6}$$

where $\Delta\mathbf{m}$ is a two-row weighted Gaussian random sample drawn from the normal distribution. Each row of $\Delta\mathbf{m}$ is a multivariate vector sample correspond to the oscillation of camera parameters vertically or horizontally. And columns of $\Delta\mathbf{m}$ are Gaussian random numbers multiplied by the weight of each subspace dimension, which is measured by the singular values from SVD used in subspace decomposition and defined as:

$$\omega_i = \|\frac{\lambda_i}{\lambda_1}\| \tag{7}$$

where ω_i is the weight of each dimension (i.e. each column of $\Delta\mathbf{m}$), and λ_i is the i-th largest singular value from SVD.

In fact, each particle $\mathbf{x}$ in subspace is corresponding to a group of feature points positions in current frame I_t. Let $\mathbf{X}$ be the set of feature points positions in the reference frame I, a prediction of points positions $\mathbf{X}_t$ in current frame can be computed by:

$$\mathbf{X}_t = (\mathbf{x}_{t-1} + \Delta\mathbf{m})P + \mathbf{X} \tag{8}$$

Since the sampling is on the subspace of the displacements matrix over a short period of k frames, the distribution of particles will be constrained by the complexity of object motion in this period of time. This complexity is represented by the rank r in our subspace motion model and is dynamically detected for every displacements matrix in each frames patch. Therefore the dimensionality of the PF state vector is alterable during the whole tracking process which is very different from the traditional PF tracking model. Meanwhile, the sampling weight of each vector dimension is also calculated automatically by the model. This will lead to a flexible and adaptable states prediction stage of PF. As a result, particles will lie in a distribution coincide with the object history moving area in the period of time, which will naturally be considered a more likely potential object area. What's more, the subspace motion model could reach to a point-to-point mapping of the object image. This mapping solved the disparity problem, as it could be used to guide the object template warping or be used directly to match the object, both of which could surely reach to a more precise appearance matching result.

3.2 Tracking Approach

Based on the subspace motion model we proposed in previous sections, we can now exploit a complete visual tracking algorithm. We firstly initialise the object feature points trajectories in the beginning k frames, which are used to locate the object in these frames and build the first displacements matrix. We usually set $k = 6$ to 10 to maintain the reliability of points tracking, surpass the subspace rank and keep enough complete trajectories (to ensure that $n \geq 9$). However, there is no need to set n a very large number. For most cases, we let $n \leq 30$ to reduce the computation cost.

We represent the state vector of PF as the positions of all feature points spanning the frames patch, and take the KLT points tracking results $\mathbf{Y}_t$ in current frame I_t as the observation state. The posterior is approximated by a finite set of N samples $\{\mathbf{x}_t^i\}_{i=1,\dots,N}$ on subspace, corresponding to N groups of predicted feature points positions $\{\mathbf{X}_t^i\}_{i=1,\dots,N}$ in current frame. In our approach, N is usually set 400 to 600. The importance weights w_t^i of each particle is estimated by the exponent projection of the Euclidean distance between the predicted points positions and the observation, which could be represented by:

$$w_t^i = \frac{1}{W_t}\exp(-\frac{1}{\eta}\|\mathbf{X}_t^i - \mathbf{Y}_t\|_F^2) \tag{9}$$

where $W_t = \sum_{i=1}^{N}\exp(-\frac{1}{\eta}\|\mathbf{X}_t^i - \mathbf{Y}_t\|_F^2)$ is the normalization term, and η the smooth term usually set to be 0.2. Note that the SVD used in subspace decomposition won't break the order of trajectories in matrix, so the points correspondence between $\mathbf{Y}_t$ and $\mathbf{X}_t$ can be ensured.

Then the tracking result $\mathbf{Z}_t$ is represented by:

$$\mathbf{Z}_t = \sum_{i=1}^{N} w_t^i \mathbf{X}_t^i \tag{10}$$

with the tracking error measured by:

$$err_t = \frac{1}{N}\|\mathbf{Z}_t - \mathbf{Y}_t\|_F^2 \tag{11}$$

We use an iterative strategy to control the tracking error into a level of under 0.5 pixel, while it could be more than 1.0 pixels before iteration. We take the $\mathbf{x}_t^i$ corresponding to the $\mathbf{X}_t^i$ with the smallest distance to $\mathbf{Y}_t$ as the initial state of next iteration, and sample particles with a smaller Gaussian variance (usually set to be half of that in last iteration). The iterations may be expected to be less than 3 according to our experiments. After we got $\mathbf{Z}_t$, we may locate the object in current frame I_t using the points correspondence between $\mathbf{Z}_{t-1}$ and $\mathbf{Z}_t$. The typical method is to take the displacement of the object as the average displacements of the points in $\mathbf{Z}_t$, and the scale changing as the average of all the ratios between current point distance and previous point distance for each pair of points.

3.3 Updating Displacements Matrix

A tough problem in visual tracking is to deal with the changing of object appearance with time. As object moves, there is always part of the object appearance disappearing and new object area arising, which in our approach means the lost of existing object feature points and the addition of new object feature points. The key is to find a method to discriminate whether a newly detected point belongs to the object, or otherwise belongs to the background. Fortunately, this challenging problem can be easily solved with our subspace motion model by adopting the idea that, all feature trajectories of the object within the current frames patch should lie on the same subspace. This means the rank of the displacements matrix will remain unchanged when a new point added into the matrix belongs to the object, although the specific singular values from SVD will be affected generally.

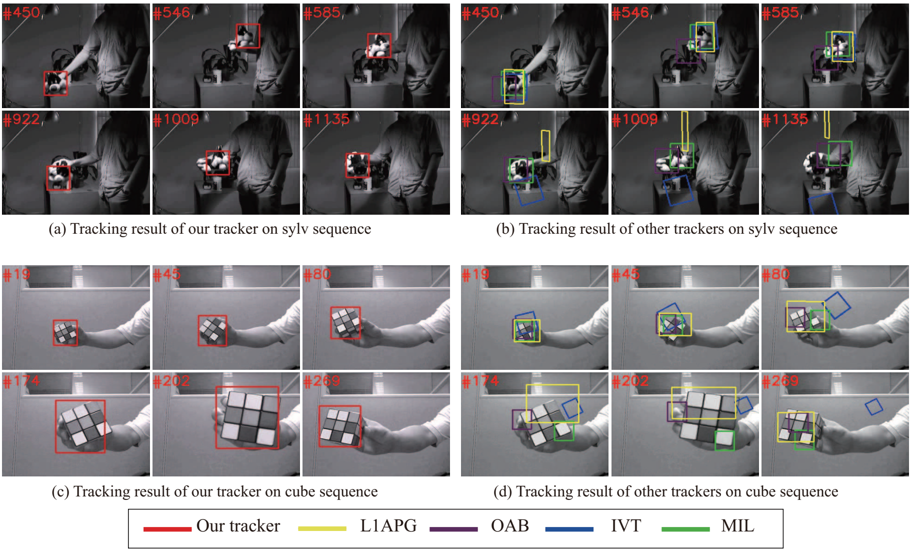
Assume that we have n' newly detected feature points spanning the current frames patch. Let the displacements of one of these points be the column vector of $(u'_j, v'_j)^T, j = 1, \ldots, n'$. We add each of these column vectors into the existing $2k \times n$ displacements matrix $[\frac{U}{V}]$ as a new column to obtain:

$$\left[\begin{array}{c|c} U & u'_j \\ V & v'_j \end{array}\right]_{2k\times(n+1)} = \left(\begin{array}{ccc|c} u_{11} & \ldots & u_{n1} & u'_{j1} \\ & \vdots & & \vdots \\ u_{1k} & \ldots & u_{nk} & u'_{jk} \\ v_{11} & \ldots & v_{n1} & v'_{j1} \\ & \vdots & & \vdots \\ v_{1k} & \ldots & v_{nk} & u'_{jk} \end{array}\right) \tag{12}$$

Then the point will be considered as belongs to the object when the rank of the left matrix in Equation (12) is the same to the rank of $[\frac{U}{V}]$. So that the displacements matrix can be replenished and updated. The rank number is detected by singular values from SVD via the same method introduced in Section 2.2.

4 Experiments

To examine the effectiveness of our proposed tracking method, we tested it on several challenge videos and do comparison with four other tracking algorithms that represent the state of



(a) Tracking result of our tracker on sylv sequence

(b) Tracking result of other trackers on sylv sequence

(c) Tracking result of our tracker on cube sequence

(d) Tracking result of other trackers on cube sequence

Figure 3: Experimental results comparison (1)

the art, including Real-time L1 Tracker (L1APG) [3], Multiple Instance Learning (MIL) [2], Online AdaBoost (OAB) [11] and Incremental Visual Tracking (IVT) [16]. We get the source code or the binaries from the authors with the same initial positions at the first frame in order to be more comparative. The videos *cube* and *sylv* are collected from prior work [2, 14]. We also made our own sequences of *toybear*, *penholder* and *vehicle*.

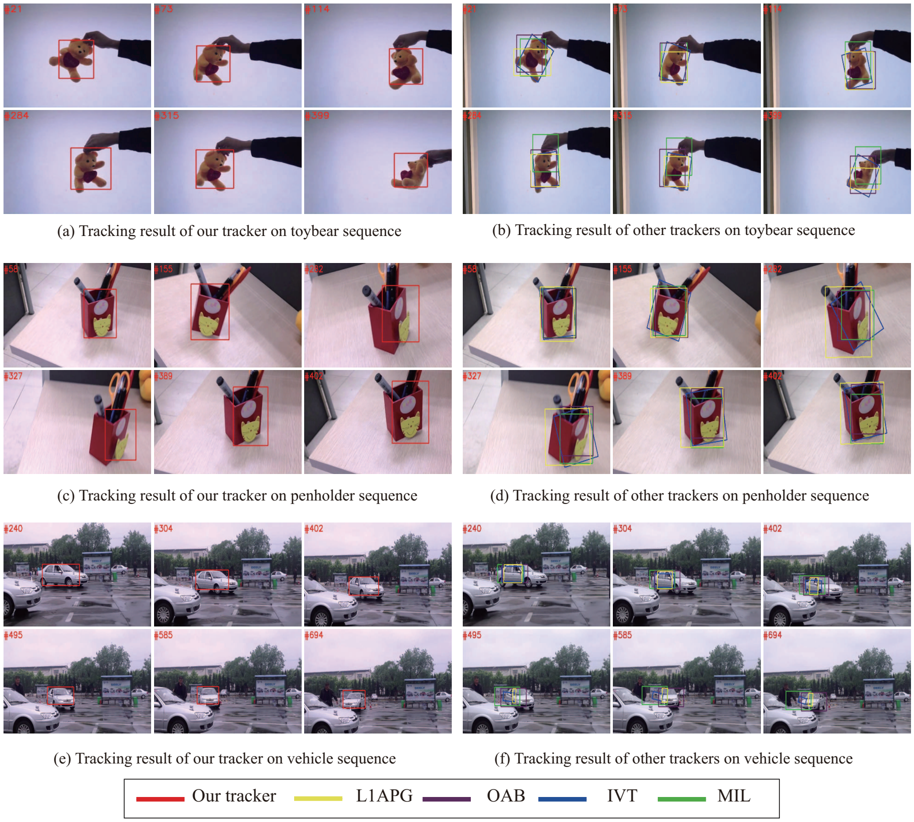
4.1 Qualitative Comparison

We compare our algorithm with other four trackers in five video sequences. The first video contains a toy undergoing lot of pose changes, as well as some illuminance variation. Some selected frames of our tracker and other trackers are shown in Figure 3 (a) and (b). The OAB tracker first lost the target in frame 450. Our tracking approach can track through the whole sequence well.

In the second video, a cube held by a hand is doing all sorts of rotation and the scale of the object changed a lot too. In this sequence, all other four tracking approach failed very quickly, the IVT tracker can not even work for 20 frames. But our tracker based on the subspace motion model can handle this sequence very well, including dealing with the rotation and focal changing. The result images are selected and shown in Figure 3 (c) and (d).

In the third video, a toy bear holding by a hand doing all kinds of poses. The bear's appearance is irregular, and the projection area of the object in the image is always in deformation. There are many drifts happens for the other four tracker, especially the MIL tracker shown as the green color rectangle. Our tracking approach is very stable. Some selected frames of our tracker and other trackers are shown in Figure 4 (a) and (b).

In the fourth video, a static pen holder shot by a moving camera. There is a bulging area in the front of the pen holder. Therefore, when the camera moves rotatingly, our subspace tracker has the advantage to deal with this situation. Some selected frames of our tracker and other trackers are shown in Figure 4 (c) and (d).



(a) Tracking result of our tracker on toybear sequence

(b) Tracking result of other trackers on toybear sequence

(c) Tracking result of our tracker on penholder sequence

(d) Tracking result of other trackers on penholder sequence

(e) Tracking result of our tracker on vehicle sequence

(f) Tracking result of other trackers on vehicle sequence

Figure 4: Experimental results comparison (2)

The last sequence (Figure 4 (e) and (f)) is a vehicle car reversing back to the parking lot. The appearance of the vehicle varies large. Therefore, the other four trackers partly or fully drift away, and only our subspace tracker can through the whole sequence well.

4.2 Quantitative Comparison

For quantitative comparison, we label the ground truth position of each frame manually. The pixel-wise tracking errors, which are measured by the Euclidean distance from the center of the target to the ground truth, are then calculated. The results are shown in Table 1. The errors are ideally zero. Compared with other methods, our method achieves smaller pixel-wise tracking errors.

5 Conclusion

In this paper, we have proposed a novel motion model for visual tracking by using the subspace theory on the displacements matrix of a rigid object. This subspace motion model is capable to describe the image deformation brought by motion of non-plane-like object as it can reach to a point-to-point mapping of the object image. This model also has the capability to calculate dynamically and automatically the dimensionality of PF state vector and the

	sylv	cube	toybear	penholder	vehicle
L1APG	11.79	12.45	10.85	15.20	16.61
MIL	7.45	36.41	17.85	18.61	8.19
OAB	10.79	39.04	12.02	18.13	6.45
IVT	12.32	61.42	18.56	**14.10**	18.61
Ours	**4.55**	**4.19**	**7.48**	15.72	**5.23**

Table 1: The pixel-wise tracking errors for different sequences and tracking methods.

weights for each dimension, by taking advantage of the subspace rank of the displacements matrix and the singular value for each dimension. Therefore the particle distribution from the PF process could cover the probable object motion states more accordantly. Based on this subspace motion model, we derived a new visual tracking method that can nicely deal with the appearance changing of the object by means of supervising the variation of the subspace rank. The feasibility of our approach bas been effectively demonstrated via comparative experimental studies.

6 Acknowledgement

The authors gratefully acknowledge the anonymous reviewers for their comments to help us to improve our paper. This work is supported by 973 program of China (No.2009CB320802), NSF of China (Nos.U1035004, 61173070, 61202149), and the National Science & Technology Pillar Program (No. 2012BAF10B03-3).

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

[2] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Proc. CVPR*, pages 983–990, 2009.

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Proc. CVPR*, pages 1830–1837, 2012.

[4] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proc. ECCV*, pages 707–720, 2002.

[5] A. M. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Proc. CVPR*, pages 316–322 vol.2, 2005.

[6] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *Int. J. Computer Vision*, 80(1):125–142, 2008.

[7] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *Proc. CVPR*, pages 2018–2025, 2012.

[8] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proc. CVPR*, pages 1–6, 2007.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations 3rd Ed.* Johns Hopkins University Press, 1996.

[10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proc. F Radar and Signal Processing*, 140(2): 107–113, 1993.

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. CVPR*, pages 260–267, 2006.

[12] M. Irani. Multi-frame correspondence estimation using subspace constraints. *Int. J. Computer Vision*, 48(3):173–194, 2002.

[13] M. Isard and A. Blakes. Condensation-conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

[14] J. Kwon, K. M. Lee, and F. C. Park. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In *Proc. CVPR*, pages 991–998, 2009.

[15] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Trans. Graphics*, 30(1):Article No.4, 2011.

[16] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *Int. J. Computer Vision*, 77(1).

[17] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, 1994.

[18] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *Int. J. Computer Vision*, 9(2):137–154, 1992.

[19] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *Int. J. Computer Vision*, 79(1):85–105, 2008.

[20] G. Wang, F. Zhong, Y. Liu, Q. Peng, and X. Qin. Visual tracking in continuous appearance space via sparse coding. In *Proc. ACCV*, pages 57–70, 2012.

[21] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74:3823–3831, 2011.

[22] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38 (4):Article No.13, 2006.

[23] W. Zhong, H. Lu, and M. Yang. Robust object tracking via sparsity-based collaborative model. In *Proc. CVPR*, pages 1838–1845, 2012.

[24] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing*, 13(11): 1491–1506, 2004.