Multimed Tools Appl (2014) 71:555–574
DOI 10.1007/s11042-013-1524-7

Real-time estimation of hand gestures based on manifold learning from monocular videos

Yi Wang · ZhongXuan Luo · JunCheng Liu · Xin Fan · HaoJie Li · Yunzhen Wu

Published online: 11 July 2013

Abstract Object pose estimation by manifold learning has become a hot research area recently. In this paper, we propose an efficient method that can recover pose and viewpoints for numerous hand gestures from monocular videos based on Locality Preserving Projections. We first select some hand dynamic gestures as primitive hand motions and set a 3D-2D mapping table to relate 3D joint angles of sampling static pose with their projective silhouettes from arbitrary viewpoints. Then the embedding space and explicit mapping function are learnt for every primitive motion. In order to make classification and prediction among those embedding spaces, a Subspace Filtering Algorithm is also proposed which can recognize and recover numerous hand dynamic gestures by the combination of primitive gestures. At last, by using skin color cues and oriented k-Dops, multi-hands can be labeled and tracked separately and accurately. Extensive experimental results demonstrate qualitatively and quantitatively that 3D pose recovery of hands can be achieved by our method robustly and efficiently.

Keywords Manifold learning · Locality preserving projections · Gesture recognition

Y. Wang · Z. Luo · X. Fan (✉) · H. Li · Y. Wu
School of Software, Dalian University of Technology, Dalian 116620, China
e-mail: xin.fan@ieee.org

Y. Wang
e-mail: dlutwangyi@dlut.edu.cn

Z. Luo
e-mail: zxluo@dlut.edu.cn

H. Li
e-mail: lihaojieyt@gmail.com

Y. Wu
e-mail: yunzhen.eric@gmail.com

J. Liu
School of Electronics Engineering and Computer Science, Peking University, Peking 100000, China
e-mail: ljc91122@yahoo.cn

1 Introduction

Gesture which is vivid, imaginable, directly and contains a wealth of information is an important medium for interpersonal communication. And gesture-based interaction also has become a hot research topic in the field of human-computer interface. By using indicative gestures, we can control the computer systems at distance, let intelligent robot to understand and communicate with us and by the 3D reconstruction of hand gestures, and accomplish various operations on the virtual objects in virtual reality or augmented reality system [17, 33]. Gesture is very promising in practical applications. However, because of the diversity, complexity, ambiguity, self-occlusion and high degree of freedom, as well as the appearance variations in time, space and individuals, the real-time estimation of human gestures is still a challenging multi-disciplinary problem. Especially, hand gesture estimation based on monocular camera is even more difficult, because of depth ambiguities. In the past few years, although hardware devices (e.g., Kinect) that combined with infrared sensors, are capable of adding depth information to 2D images and the cost of such type of equipments is slowly decreasing, it is still worth searching for alternative solutions that implemented with simple video cameras [27], which are widely used in personal computers, laptops and mobile phones and can cut the budget required for the deployment of multiple entertainment systems, so scholars from all walks of life still put their efforts to accomplish this task.

A variety of methods have been proposed during the past few years, which can be categorized into motion recognition methods and shape reconstruction methods [11, 13, 24]. For motion recognition, special recognizers have been constructed to keep track of temporal modeling like Hidden Markov Model (HMM) [10, 36], Neural Network (NN) [18, 22], rule based and finite state machine and so on [2, 19, 31]. For shape reconstruction, methods further can be categorized into model-based or appearance-based methods. Model-based approaches are performed by formulating an optimization problem whose objective function measures the discrepancy between the visual cues that are expected due to a model hypothesis and the actual ones. The employed optimization method must be able to evaluate the objective function at arbitrary points in the multidimensional model parameters space. Model-based approaches provide a continuum of solutions but are computationally costly and depend on the availability of a wealth of visual information. Appearance based models is also referred as 2D models or view based models [26, 28, 29, 32], which typically estimate hand configuration from images by learning a mapping from the image feature space to hand configuration space. The mapping is usually highly nonlinear due to the variation of hand appearances under different viewpoints. So their recognition abilities confined in the training set of the known hand configuration and the accuracy of collecting. The appearance based methods are usually very fast, and can be employed in monocular camera system.

In recent years, computer vision research has witnessed a growing interest in subspace analysis and manifold learning techniques [39]. Given a set of high-dimensional data points, manifold learning techniques aim at discovering the geometric properties of the data space, such as its Euclidean embedding, intrinsic dimensionality, connected components, homology and etc. Manifold learning techniques can be classified into linear (LDA [5], PCA [23], MDS [25]) and non-linear(ISOMap [34], LLE [30], LE [6], etc.) techniques, which have been applied to face, gait recognition with impressive results. Despite the high dimensionality of the configuration space, many human motion activities lie intrinsically on low dimensional manifolds. Intuitively, the gait or gesture is a 1-dimensional manifold embedded in a high dimensional visual

space [1, 39], if we consider the body kinematics as well as the observed motion through image sequences.

In this paper, we are especially interested in using manifold learning techniques to establish a low-dimensional structure to organize the visual training data of dynamic gestures from multi-viewpoints. At the same time, we try to learn an explicit mapping from feature space to subspace to recover intrinsic 3D hand configurations and viewpoints for numerous hand gestures from monocular image sequences. To be specific:

(1) Some hand dynamic gestures are selected as primitive training gestures. And sample the static pose of these gestures at certain steps. A 3D-2D mapping table is then set to relate 3D joint angle data of every pose with their projective silhouettes from arbitrary viewpoints.
(2) Subspaces for primitive training gestures from arbitrary viewpoints are learned by Locality Preserving Projections (LPP) [7, 14, 15]. Each pose in the feature space could be explicitly mapped to the low-dimensional embedding space which preserves local structure and has discriminating power to make classification.
(3) A Subspace Filtering Algorithm (SFA) is proposed, which can recognize and recover numerous hand dynamic gestures from the combinations of primitive training gestures. SFA mainly converts the multiple-motion recognition and reconstruction problems to classification and prediction process among embedding spaces.
(4) A multiple-hand estimation and tracking framework is also proposed. By combining skin color cues and oriented *k-Dops*(Discrete Orientation Polytopes) [38], several hands can be labeled and tracked separately and accurately. With the SFA, the estimation of configurations and viewpoints can be achieved robustly in real-time.

2 LPP based techniques

LPP (Locality Preserving Projections), a linear dimensionality reduction algorithm, is obtained by finding the optimal linear approximations to the Eigen-functions of the Laplace Beltrami operator on the manifold. It seeks to preserve the intrinsic geometry of the data and computes the explicitly the manifold structure of the feature space [15]. Different from ISOMap and LLE defined on the training data, LPP is defined everywhere and can be simply applied to any new data [14]. And LPP may be conducted in the original space or in the reproducing kernel Hilbert space (RKHS) into which data points are mapped. Besides, OLPP (The orthogonal locality preserving projection) method produces orthogonal basis functions and can have more locality preserving power [7, 37]. Take into account, the projective images of continuous hand motion with different viewpoints are in one manifold (will be proven by experiments in Section 6), we use the basic version of LPP. The following is a simple description, for more details, please refers to the article [15].

Let the set of input instances be $X = \{x_i \in R^d, i=1,\ldots,N\}$ and their corresponding points in the embedding space be $Y = \{y_i \in R^e, i=1,\ldots,N\}$, where d is the dimensionality of the feature space and e is the dimensionality of the embedding space. The objective of LPP is to minimize the function:

$$\min_P \sum_{i,j=1}^{n} \left|\left|y_i - y_j\right|\right|^2 S_{ij} \tag{1}$$

Where S_{ij} is a similarity measurement ($S_{ij} = S_{ji}$) in adjacency graph of the input X set and can be computed by Gaussian kernel:

$$S_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}} \tag{2}$$

Let

$$y^T = W^T x \tag{3}$$

It expects each data point and its neighbors to lie on or close to a locally linear patch of the manifold. By simple algebra formulation, the above objective formula (1) can be reduced as follows:

$$\begin{aligned} &\min_P \sum_{i,j=1}^{n} \left|\left| y_i - y_j \right|\right|^2 S_{ij} \\ &= \sum_{i,j=1}^{n} \left(W^T x_i - W^T x_j \right)^2 S_{ij} \\ &= W^T X L X^T W \end{aligned} \tag{4}$$

The transformation vector W that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$XLX^T W = \lambda XDX^T W \tag{5}$$

Where D is a diagonal matrix and its entries are column sums of S. L=D-S is the laplacian matrix. Both matrices XLX^T and XDX^T are symmetric and positive semi-definite.

At last the explicit mapping is as follows:

$$x \rightarrow y = W^T x \tag{6}$$

3 Training gesture database

According to the literature, there are a few public available gesture image databases. Cambridge-Gesture database consists of 900 image sequences of nine gesture classes, which are defined by 3 primitive hand shapes and 3 primitive motions [21]. The database published by Athitsos and Sclaroff contains more than 107000 images, covering 26 hand gestures. But this database only provides edges information [4]. The Massey Gesture Database includes about 1500 images of different hand postures in different lighting conditions [9]. However, none of the above contains both 3D joint angle data of every pose and their projective silhouettes from arbitrary viewpoints and the primitive hand motions, so that we have to build our training data base in the first.

In our work, a 24° of Freedom (DoF) kinematic model of the human hand proposed in [8] is used to imitate realistic movements see Fig. 1. In order to get more accurate motion simulation, angle constraints and dynamic constraints have been added to the 3D model. Dynamic gestures or continuous motions could be simulated and discredited into a number of poses by modifying angle parameters get from a data glove. And then in the test stage, the other orientation parameters of the hands will be estimated by some geometric strategies. The training data are generated using computer graphics by rendering from 29 viewpoints roughly distributed on an eighth view sphere, see Figs. 1 and 2. Finally we make the training images binary and scaled to the resolution of 64×64, getting a discrete training database

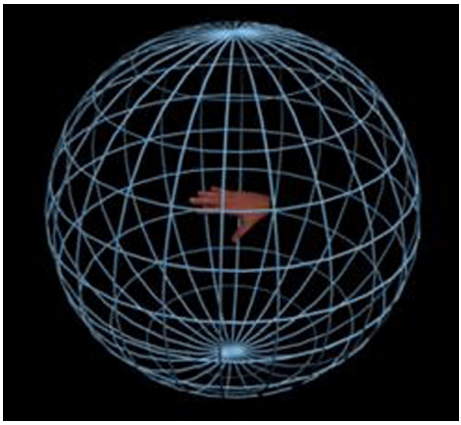


Fig. 1 Viewpoint sphere and 3d virtual hand

which consists of 1000 samples for two grasping gestures form multi-viewpoints. Some examples are shown in Fig. 3.

Meanwhile, a 2D-3D mapping table is built to connect 2D projective images with 3D pose information. Specifically, in every row of 2D-3D mapping table, the main key is the index of the training image and attributes are joint angles and orientation of the pose. The original orientation for a pose is set to the first pose in row 2 of the Fig. 3, where hand tips point to the negative direction of X axis. The orientation information of other poses is set to angle difference from the original pose in three coordinate axes. And it only needs to keep one 2D-3D mapping table for all the primitive gestures to save storage. Different from other manifold learning recognition works like [1, 12], our training data does not include rotation of Z axis.

4 Detection, segmentation and tracking

Hand detection and segmentation are critical in visual image-based gesture recognition, which directly influences the recognition accurate and rate. This section discusses a robust and automatic preprocess to solve this task.

Fig. 2 A gesture from different angel of viewpoints

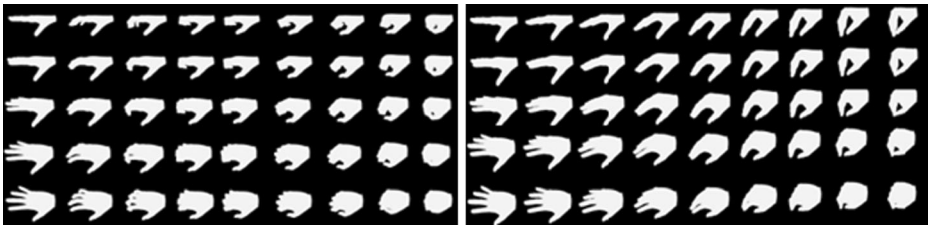


Fig. 3 Some training data of two primitive gestures

In this paper, we assume that a person is sitting in front of a fixed camera with one or two hands within the capture region. Set the normal axis of camera points to the operator *Z*-axis. Camera is in the origin of the coordinate and axis *X* and *Y* are equal to image coordinate.

For every input video image, firstly make the image smooth, noise reduction and color balanced. And then segment the skin area from background by Gaussian Mixture Model (GMM) [20, 35] and make these images binary. On the second, calculate skin regions' contour areas and remove those of small size, for they are probably the noise areas. Here there may be some complex situations in the practical applications: face, exposed arms, the intersection of hands and face or body parts, all these will become big interferences for hand segmentation and gesture recognition. Many works ignore these situations or tackle these by adding some constrains, such as the users must ware long-sleeved clothes. According to observations in the environment we set up, the first most likely interference is that part of user's forearms moving into the capture region (see Fig. 4), so a Standardized Hand Segmentation Algorithm (SHSA) is put forward to settle this issue. Subsequently an ODop(Oriented k-Dops) based Multi-objects Tracking Algorithm (MOTA) is also proposed to label and track the hand regions.

4.1 Standardized hand segmentation algorithm (SHSA)

Firstly, compute *OBBs* [39] for every hand skin area and them into four regions $\{R_l, R_{ml}, R_{mr}, R_r\}$ evenly along the longest axis of its *OBB* (see Fig. 4) respectively. Find the minimum span of skin region orthogonal to the longest spindle of the *OBB* in R_{ml} and R_{mr} by hill-climbing method. Then cut *OBB* into two parts OBB_l and OBB_r where the minimum span is. Then *OBBs* are computed again for skin regions in OBB_l and OBB_r. The length of the longest sideline of *OBB* is used as side to build a square image. In order to avoid the boundary error, expand 15 % percent of black background outward and resize the area to 64*64 binary images evenly. All these images will be put to SFA in section5 and only recognized hand area will be taken as *KB*s and tracked afterwards. The SHSA algorithm only suit to the cases that forearms are in the video without sleeves, for those with sleeves, SHSA is not always valid. However, SHSA is a optional supplementary in some situations.

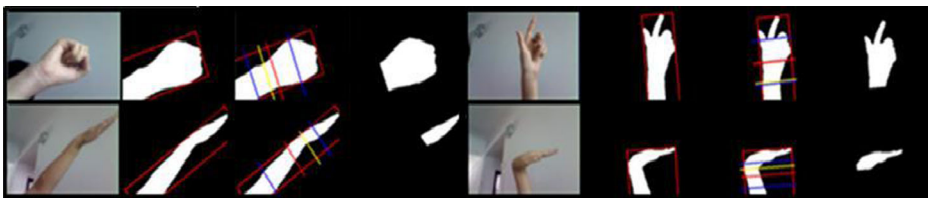


Fig. 4 Some examples for algorithm SHSA

4.2 Multi-objects tracking algorithm (MOTA)

After detecting *KB*s and estimating gestures, it is need to track the identified *KBs* in the following frames. However, in the real applications, there will probably be more than one *KB* at the same time, or hands may undergo a rotation around *X-Axis*, *Y-Axis* or an arbitrary angle with their *BKs*' forms changing so much. So the tracking algorithm should have the ability to handle all these cases. Inspired by the work [3], it'd better design a simple geometry to replace *KB*s to mark the locations, to confirm identities, to have intersection tests and so on, so we proposed a fast and accurate method to label and track multi-objects by oriented *k-Dop* (*ODop*).

K-Dops (*k* Discrete Orientation Polytopes), or fixed-direction hulls as they are sometimes called, made of *k* pairs of parallel hyperplanes in high dimensional space. Particularly, in two-dimensional space, *OBB* is a 2-*Dops*, has its axes aligned to the two principal component vectors of object. *OBB* is not the tightest, however, so we add other two pairs of parallel lines to *OBB*, forming an orientated 4-*Dops* (*ODop*) for a *BK*. Compared with an orientated ellipse used in [3], *ODop* is tighter and "large" enough to contain all pixels in *a BK*, while the inner ellipse cannot include all the pixels in *BK* and the circumscribed ellipse is not tight enough, so they are not accurate, especially in the case to distinguish closer *BKs*, see Fig. 5. Besides, it is easy to determine a point's position about an *ODop* (which is a convex polygon) by the Ray Casting or Winding methods. So our improved algorithm for tracking multiple objects operates in the following steps:

In the initial frame t=0, *ODop* is computed and labeled for every *BKs* from 1 to *N* by pairs, and stored in a tracking object set *TOS* (see formula (7)).

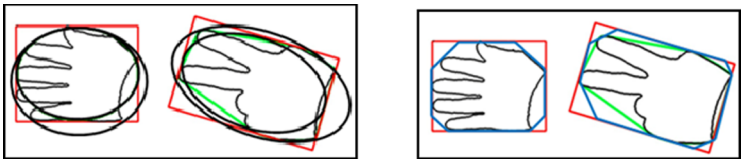
$$TOS^0 = \left\{ODop_1^0, BK_1^0\right), \left(ODop_2^0, BK_2^0\right), \ldots, \left(ODop_n^0, BK_n^0\right)\} \tag{7}$$

Next is a cyclic process until the end of tracking.

In the frame *t*, *BK*s are segmented and their $ODops^{t}$ are computed at first. Then, in order to track and label *BKs*, the relationships between $ODops^{t}$ and $ODops^{t-1}$ are reviewed in the following cases:

(a) If an $ODop^{t}$ only relates to one $ODop^{t-1}$, then update TOS^{t} by label $\{ODop^{t}, BK^{t}\}$ the same number as $\{ODop^{t-1}, BK^{t-1}\}$. That means an identified *BK* is tracked (see Fig. 6a). Then compare the similarity between BK^{t-1} and BK^{t}. If they only have the difference in the direction Z-axis, then relate 3D information of BK^{t-1} to BK^{t} without entering into the recognition process in section5. Otherwise, get 64*64 test sample and put it into recognition process.

(b) If an $ODop^{t}$ relates more than one $ODop^{t-1}$, then label $\{ODop^{t}, BK^{t}\}$ by number of the $ODop^{t-1}$ with which the intersection area is the largest (see Fig. 6b).



(a) An example for ellipsein paper [38]. (b) An example for ODops.

Fig. 5 The comparison of the ellipses and *ODops* for two *BKs*. **a** An example for ellipse in paper [3]. **b** An example for ODops

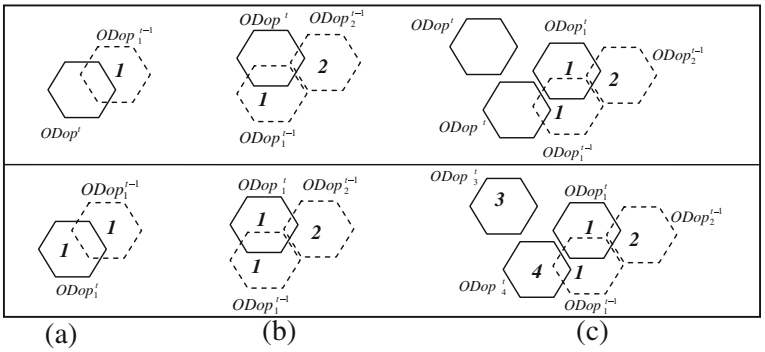


Fig. 6 The relationships between $ODops^t$ and $ODops^{t-1}$ and the tracking results in t and t-1 for three cases (**a**–**c**) described in the algorithm

(c) If an $ODop^t$ relates to no $ODop^{t-1}$ or the related $ODop^{t-1}$ has been labeled, then add this pair $\{ODop^t, BK^t\}$ to TOS^t with a new number. That means a new target is found (see Fig. 6c).
(d) If an $ODop^{t-1}$ has no *BKs* which means its *BKs* has disappeared, so remove this $ODop^{t-1}$ from TOS^t.

As to case (c) and (d), it'd better delay the addition and remove operations when the test results are the same after at least five frames to maintain the stability of the system.

5 Identification and estimation

5.1 Feature extraction

The key step of manifold learning algorithm is feature selection which determines whether the mode of images and the relationship with other modes can be presented correctly. Instead of taking all pixels as feature vector directly with consideration of making XDX^T nonsingular, noise reduction, we use Hu's seven moment invariants [16] as feature vector for *BKs*, noted by *Fv*. *Fv* is invariant under changes in translation, scale and rotation [23]. So the influence of hand shape to the accurate of gesture estimation could be minimized. The invariant quality in translation and scale are welcome for recognition, however, if the hand only rotated around *Z* axis in *XOY* plane, the rotation invariant is not useful to figure out the difference between *BK*s of the same pose. Especially in low-dimensional space, the samples of the same pose with different *Z* angles will be clustered closely, or overlapped. Therefore, in our method, *AngleZ* is calculated and stored in preprocess, and then combined with recognition result to recover 3D configuration. By this way, a lot of samples and training time have been saved for LPP.

5.2 Learning embedding spaces

Actually for a continuous motion (for instance, a hand's grasping action), its 2D projective images reside on a low-dimensional sub-manifold [34]. So by using nonlinear dimensionality reduction methods LPP, we can get a corresponding low-dimensional linear embedding

space. If we have N_m continuous hand movement units, we can get N_m multi-view embedding spaces denoted by S_i with transfer matrix W_i respectively. The estimation of a gesture in time t could be converted to a classification problem among those embedding spaces. And as a result, a continuous gesture can be reconstructed by the combination of some primitive embedding spaces, see Fig. 7.

The posteriori probability of each class S_i is in formula (8). The priori probability of every class could be set to a constant, so that only class-conditional probability of each class is unknown.

$$P\left(S_i \middle/ Fv_t\right) = p\left(Fv_t \middle/ S_i\right)P(S_i) \middle/ \sum_{j=1}^{N_m} p\left(Fv_t \middle/ S_j\right)P(S_j) \tag{8}$$

Given learned mapping function $y=W^{\mathrm{T}}x$, we consider that the class conditional probability of an embedding space S_i is determined by the number and distance of neighbors that fall in the near region of y in the embedding space. So by adopting Gaussian kernel function in formula (9), where d represents Euclidean distance between y and its neighbors, we finally could get the posteriori probability of each embedding space.

$$P(Fv^t|S_i) = \sum_{j=1}^{N_m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{d_j - \mu_i}{\sigma_i}\right)^2\right] \tag{9}$$

Based on the above analysis, in the following, we propose a Subspace Filtering Algorithm (SFA) (see Fig. 8) to find potential movement manifold spaces to guide tracking and estimation.

At initialization time step $t=1$, the position Y_i^t of the input Fv^t is located in every embedding space S_i ($i=1,\ldots,N_m$) by transfer matrix W_i in parallel. And all the neighbors (denoted by NB_i^t) of Y_i^t are found within domain radius. Then put those most similar to Y_i^t into the set $Ebest_i^t$ respectively in S_i. Then compute the posteriori probability for each embedding space S_i. The result of this step is the bilinear interpolation of 3D information of $Ebest_i$ ($i=1,\ldots,N_m$).

In the second stage, put the potential spaces into set PS^t (the number is N_{ps}), whose posterior probability is larger than the threshold, and put the left into set LS^t (the number is N_m-N_{ps}). These two sets will be kept and updated in the whole process of recognition by *Update* (*PS*, *LS*). At last Kalman filter is added in each potential movement manifold space to predict the position of motion in $t+1$ denoted by $\hat{Y}_i^{t+1}$. And in the following time steps, SFA will first consider those spaces in PS^t, whose predicted $\hat{Y}_i^{t+1}$ is near Y_i^{t+1}. At last, if there are more than one candidates found in

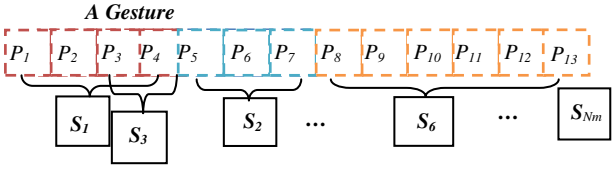


Fig. 7 A continuous gesture is segmented discretely to 13 poses (P_1-P_{13}) and reconstructed by some primitive embedding spaces (S_1-S_{Nm}): (1) $P1$ and P_2 by S_1; (2) P_3 and P_4 by the combination S_1 *and* S_3; (3) P_5 -P_7 by S_2; (4) P_8 -P_{13} by S_6

Algorithm SFA (Subspace Filtering Algorithm)

Notation: W_i *represents the transfer matrix*;
N_m *represents the number of 2D embedding space*;
N_{ps} *represents the number of 2D embedding space in PS*;

Initialization step, at time *t*=1:

For ($i=1,\ldots,N_m$) **parallel do**

$Y_i^t = W_i\ FV^t$; //Locate position of the input in embedding space Si.

$NB_i^t = GetNeighbor(Y_i^t)$; // $\| Y_i^t - nb_j \| < \varepsilon$, *for every* $nb_j \in NB_i$

$Ebest_i^t = Getbest(Y_i^t, NB_i^t)$;

$Posterior(Y_i^t, NB_i^t)$;

End for

$Update\ (PS^t, LS^t)$;

For ($i=1,\ldots, N_{ps}$) **parallel do** // N_{ps} is the number of S_i in PS^t.

$\hat{Y}_i^{t+1} = KalmanFilter(Y_i^t \in PS^t)$;

End for

Return $\mathrm{Re}\,sult^t = BI(Get3D(Ebest_i^t))$; //BI( ) means function of bilinear interpolation

At time *t*+1:

For ($i=1,\ldots,N_m$) **do**

$Ebest_i^{t+1} = NULL$;

End for

If PS^t=*NULL* **then**

PS^{t+1}=LS^t;

End if

For *every* S_i *in* PS^t **parallel do**

$Y_i^{t+1} = W_i\ FV^{t+1}$;

If $\| Y_i^{t+1} - \hat{Y}_i^{t+1} \| < \varepsilon$

$NB_i^{t+1} = GetNeighbor(Y_i^{t+1})$;

$Ebset_i^t = GetBest(Y_i^{t+1}, NB_i^{t+1})$;

$Posterior(Y_i^{t+1}, NB_i^{t+1})$;

End if

$Update(PS^{t+1}, LS^{t+1})$;

For ($i=1,\ldots,N_{ps}$) **parallel do**

$\hat{Y}_i^{t+2} = KalmanFilter(Y_i^{t+1})$;

End for

Return $Resultt^{+1}$=$BI\ (Get3D\ (PSet_{best}))$;

Fig. 8 The description for SFA

embedding spaces, then bilinear interpolate the 3D joint angles of those candidates as the estimation result. At last, compute the posteriori probability for each S_i by function *Posterior* ( ) based on the formula (8) and (9).

For an instance, in Fig. 9, there are two primitive gestures with their embedding spaces S_i and S_j. The input video is recognized and tracked in both spaces in time *t*-2, *t*-1 and *t*. And the estimation of 3D configuration is the bilinear interpolation of the results from two spaces.

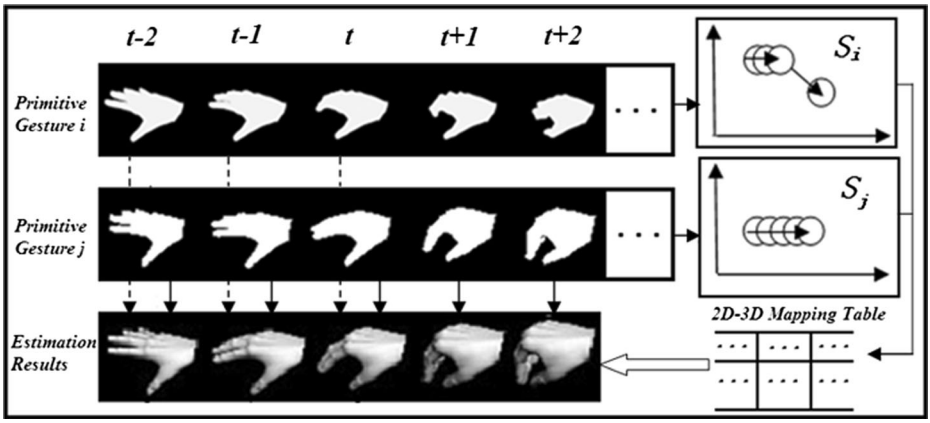


Fig. 9 Two motions with their embedding spaces and the input video is recognized and tracked by SFA

However, from the time $t+1$, the mapping of the input is far from the prediction in S_i, so S_i is removed from *PS*, the algorithm will only consider S_j until it does not meet the criteria or end of estimation.

6 Experimental results

The gesture estimation system developed by VS2010, OpenCV and OpenGL library, was executed on a laptop running with Core 2 Duo CPU at 2.40 GHz and 3G RAM. The camera's resolution was 640×480, but we reduced the images to 320×240 as raw data.

6.1 Learning multi-view manifold

Some specific settings were taken according to our method in LPP Code shared by [15] to learn multi-view manifolds for primitive gestures in matlab7.0. Firstly, we used all pixels of a training image as its feature vector to construct neighbor graph. Then *Fvs* of training images are put into dimensionality reduction learning process, getting multi-view motion embedding spaces. The training datasets were created as explained in section 3.

Figure 10 shows the 3D embedding space for two primitive gestures in section3. Specifically Fig. 10a–c are three views of the embedding space for primitive gesture1 and Fig. 10c–e are three views of the embedding space for primitive gesture2. The transfer matrixes are in formula (10) and (11) respectively. From the 3D point cloud, we can see there are still some intrinsic geometric structures embedded, so we tried to reduce them to 2D spaces. The two transfer matrixes are in formula (12) and (13). The results are promising, from Fig. 11, we can easily find out the structure of the training data: (1) Data in the horizontal direction have the tendency of hand's grasping motion from open to close; (2) Data in the vertical direction have the tendency of rotation of the hands. These results proved that our *Fv* is valid for manifold learning of multi-view continues gesture and the best dimension of embedding space is 2D.

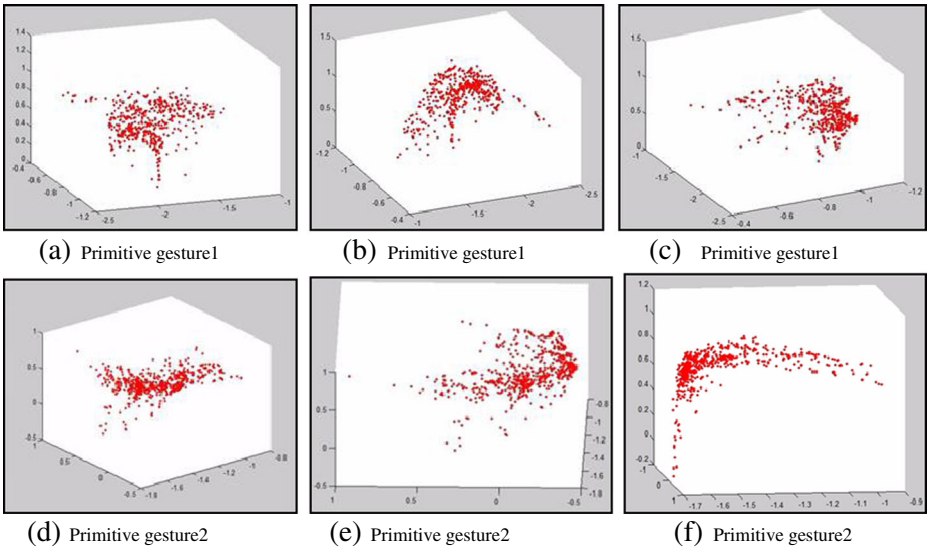


Fig. 10 Three views of three-dimensional manifolds learned by LPP for primitive gesture1 (**a**–**c**) and gesture2 (**d**–**f**). There are still some intrinsic geometric structures embedded

For recognition part, we quantified and rounded these two 2D primitive embedding spaces by linear transformation: T_1 (u=[(x+1.7)×500], v=[(−(y−1))×500]) and T_2 (u=[(x+2.5)×500], v=[(−y-0.4)×500]), then stored them in two images.

$$W_1^T = \begin{Bmatrix} -0.9995 & -0.0087 & 0.0043 & 0.0287 & -0.0055 & -0.0028 & -0.0014 \\ -0.9878 & 0.0335 & 0.0036 & 0.1512 & -0.0092 & -0.0073 & -0.0053 \\ 0.9773 & -0.2105 & 0.0052 & 0.0149 & -0.0029 & 0.0143 & -0.0040 \end{Bmatrix} \quad (10)$$

$$W_2^T = \begin{Bmatrix} -0.2619 & 0.1688 & 0.2007 & 0.4640 & 0.4687 & 0.3967 & 0.5197 \\ 0.9177 & -0.0768 & 0.2025 & -0.2405 & -0.0262 & -0.1379 & 0.1823 \\ 0.9882 & -0.1165 & -0.0548 & -0.0729 & 0.0097 & 0.0195 & 0.0306 \end{Bmatrix} \quad (11)$$

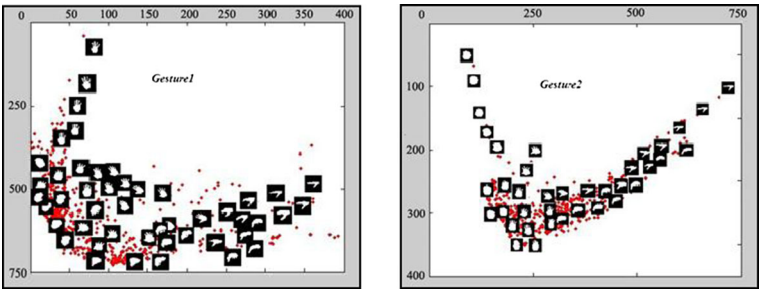


Fig. 11 Two-dimensional manifolds images learned by LPP for primitive gesture1 (a) and gesture2 (b). Points in the horizontal direction have the tendency of hand's grasping motion from open to close and points in the vertical direction have the tendency of rotation of the hands

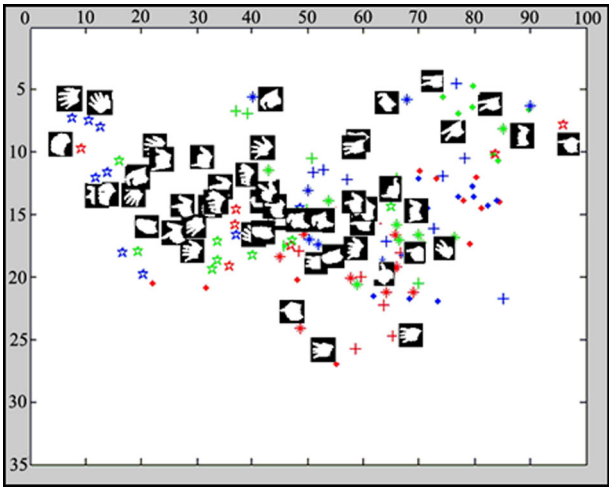
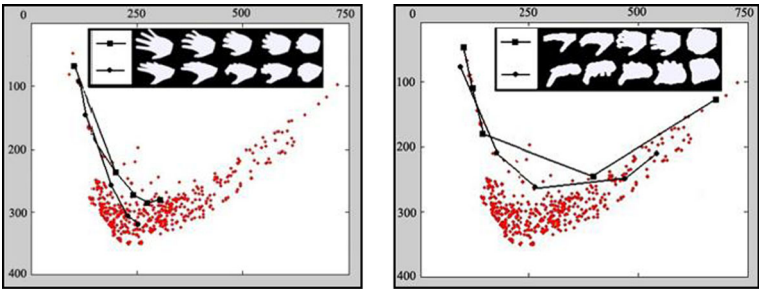


Fig. 12 A two-dimensional manifold learned by LPP for ten classes of static poses from Chinese Sign Language

$$W_1^T = \begin{Bmatrix} -0.9995 & -0.0087 & 0.0043 & 0.0287 & -0.0055 & -0.0028 & -0.0014 \\ -0.9878 & 0.0335 & 0.0036 & 0.1512 & -0.009 & -0.007 & -0.005 \end{Bmatrix} \quad (12)$$

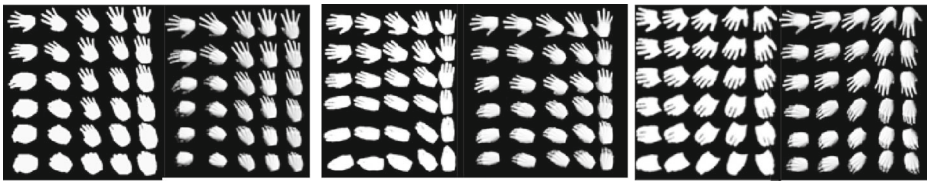
$$W_2^T = \begin{Bmatrix} -0.2619 & 0.1688 & 0.2007 & 0.4640 & 0.4687 & 0.3967 & 0.5197 \\ 0.9177 & -0.0768 & 0.2025 & -0.2405 & -0.0262 & -0.1379 & 0.1823 \end{Bmatrix} \quad (13)$$

In addition, we built a primitive gesture set3 including ten classes of static poses from Chinese Sign Language database. Each class represents an Arabic number and has 10 samples from arbitrary views including rotation around Z axis. There are total 100 samples. Same to the method mentioned above, this set was reduced to 2D embedding space by transfer matrix W^T (formula (14)) and stored in an image, see Fig. 12. From this manifold, we can see, the data are arranged mainly according to the number and direction of figure tips, and most classes are clustered in near region. But there still has some singular points which may have interference in classification, and it will become worse for recognition for



(a) Fixed viewpoint (b) Rotation viewpoints

Fig. 13 Recognition and tracking paths for (**a**) test data of gesture2 from two fixed viewpoint and (**b**) test data of gesture2 from two rotation viewpoints, all of them can achieve 100 % recognition rate



(a) Set1 and the estimation result; (b) Set2 and the estimation result; (c) Set3 and the estimation result;

Fig. 14 Three test data sets with their estimation results

continuous gestures which has a large amount of training samples. So it is wise to compute the rotation parameter of Z axis and translational motion along the X axis and the Y beforehand and train each primitive continuous gesture separately.

$$W_3^T = \begin{Bmatrix} -0.7248 & 0.0092 & 0.1307 & 0.5591 & 0.1959 & 0.2425 & 0.2184 \\ 0.8813 & -0.2347 & 0.1094 & 0.2088 & -0.1225 & -0.2625 & 0.1691 \end{Bmatrix} \quad (14)$$

6.2 Recognition

Our SFA can estimate any combination of primitive gestures. If we have N primitive gestures, and each gesture has M static poses, then the total number of gestures that can be recognized is in formula (15), where S is the number of duplicate poses.

$$\sum_{i=1}^{MN-S} C_{NM-S}^{i} \quad (15)$$

So if given enough primitive gestures, we can estimate numerous gesture at last. The worst case for SFA is that a continuous gesture does not absolutely identical to any primitive gesture, but its static poses can be found in some primitive spaces. That is such gesture cannot be tracked only within one embedding space and it need to be examined in every spaces in every time step, so it will cost much time for recognition, but the accuracy can be achieved.

In Fig. 13, two groups of test data were selected from training data of primitive gesture2 to illustrate the recognition and tracking paths for continuous gestures in embedding spaces. In Fig. 13a test data from two fixed viewpoint and in Fig. 13b test data from two rotation viewpoints, and all of them can achieve 100 % recognition rate.

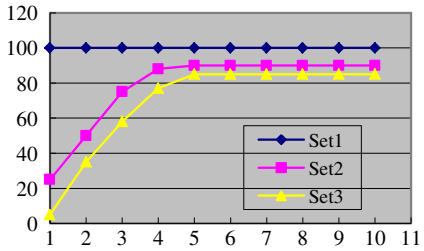


Fig. 15 The recognition rate versus the value of radius of neighborhood in SFA for three testing sets in Fig. 14

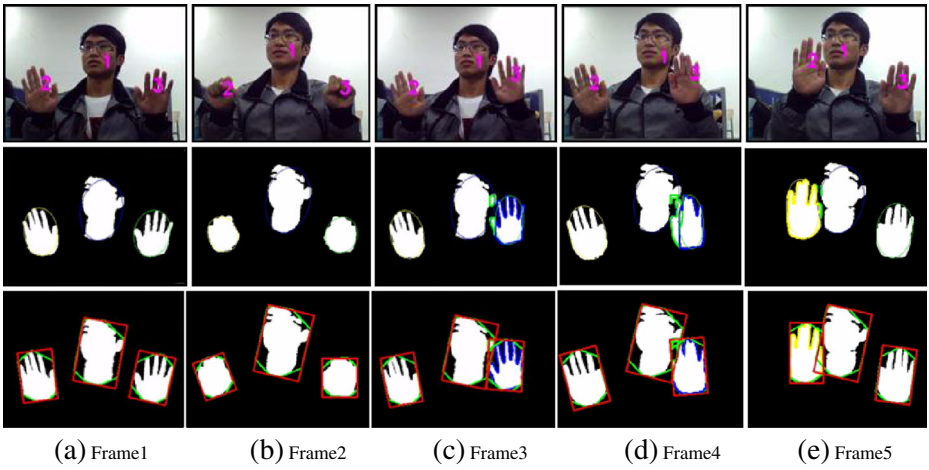


Fig. 16 Comparison for our *ODop* based tracking method (MOTA) and ellipse based method in [3]

By being fully aware of the hand shape and personalized action might have some the influence on the accurate of recognition, we have test three kinds of gesture sets to show generality of our approach. Specifically, Set1 was selected from training data of gesture1, Set2 and Set3 were taken from monocular video of two real hands' personalized grasping. Figure 14 shows the training data of primitive gesture1 and their estimation results. Figure 15 illustrates the recognition rate versus the value of radius of neighborhood in SFA. The radius is adjusted from 1 to 10. It is clear that with the increase of radius, the recognition rate will rise until the peak is reached and then it keeps stable. For Set2 and Set3, some singular points located in far away from training data in embedding spaces, so the recognition rate cannot achieve 100 %. But in the real application, we can program to fit it between two identified poses.

6.3 Label and tracking

ODop based tracking method compared with ellipse based method in [3] is shown in Fig. 16. For visualization purposes, *ODop* is colored by green and *OBB* is colored by red. We can see when the distance between objects is relatively far, both tracking algorithms could get satisfied results (Fig. 16a and b), however, in the case objects are closer (Fig. 16c–e), the contour of hand is colored by blue, misclassification by *ODop* is less than that ellipse. On the other hand, both algorithms can operate in real-time. Figure 17 is our gesture estimation system.

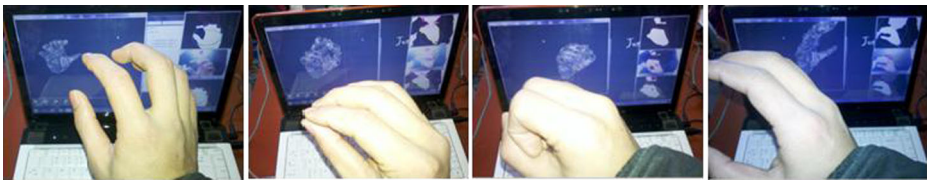


Fig. 17 The proposed gesture estimation system based on manifold learning from Monocular videos

7 Conclusions

In this paper, we proposed a generative framework that efficiently recovers intrinsic 3D hand configurations and viewpoints from monocular image sequences. Firstly, a LPP-based filtering algorithm converts the multiple-motion recognition and reconstruction problems to a classification among embedding spaces, and proximity query and prediction process within embedding spaces. Then a multiple-hand tracking method is presented which works well when hands move in complex trajectories and occlude each other. In the future, we will investigate how to improve accuracy and speed of segmentation algorithm, and how to identify effective gesture effectively.

Acknowledgments The research activities as described in this paper were funded by Doctor Startup Fund of Liaoning Province, China (20111023), the National Natural Science Funds of China (61033012, 61003177, 61272371, 11171052 and 61173104), and the program for New Century Excellent Talents (NCET-11-0048) and Specialized Research Fund for the Doctoral Program of Higher Education (20120041120050).

References

1. Abdelkader MF, Abd-Almageed W, Srivastava A, Chellapp R (2011) Silhouette-based gesture and action recognition via modeling trajectories on Riemannian shape manifolds[J]. Comp Vision Image Underst 115(3):439–455
2. Alvarez-Alvarez A, Cordon O (2012) Human gait modeling using a genetic fuzzy finite state machine [J]. IEEE Trans Fuzzy Syst 20(2):205–223
3. Argyros AA; Lourakis MIA (2004) Real-time tracking of multiple skin-colored objects with a possibly moving camera[C], European Conference on Computer Vision, Springer Berlin Heidelberg, ECCV2004, LNCS 3023:368–379
4. Athitsos V, Sclaroff S (2003) Estimating 3D hand pose from a cluttered image[C]. In proceeding of IEEE Conference on Computer Vision and Pattern Recognition, CVPR2003, Vol.2(2) 432–439
5. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection [J]. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
6. Belkin M, Niyogi P (2003) Laplacian Eigenmaps for dimensionality reduction and data representation [J]. Neural Comput 15(6):1373–1396
7. Cai D, He X, Han J, Zhang H-J (2006) Orthogonal laplacianfaces for face recognition [J]. IEEE Trans Image Process 15(11):3608–3614
8. Cobes S, Ferre M, Uran MA (2008) Efficient human hand kinematics for manipulation tasks[C], International conference on Intelligence Robots and Systems, 2246–2251
9. Dadgostar F, Barczak ALC, Sarrafzadeh A (2005) A color hand gesture database for evaluating and improving algorithms on hand gesture and posture recognition [J]. Res Lett Inf Math Sci 7:127–134
10. Elmezain M, Al-Hamadi A, Appenrodt J et al (2008) A hidden markov model-based continuous gesture recognition system for hand motion trajectory[C], 19th International Conference on Pattern Recognition, ICPR 2008, 1–4
11. Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: a review. Computer Vision and Image Understanding[J]. In Special Issue on Vision for Human-Computer Interaction Vol. 108(1–2):52–73
12. Ge SS, Yang Y, Lee TH (2008) Hand gesture recognition and tracking based on distributed locally linear embedding[J]. Image Vis Comput 26(12):1607–1620

13. Hasan MM, Mishra PK (2012) Hand gesture modeling and recognition using geometric features: a review[J]. Can J Image Process Comput Vision 3(1):12–26
14. He X, Niyogi P (2002) Locality preserving projection, technical report, TR-2002-09, Department of Computer Science, the University of Chicago
15. He X, Yan S, Hu Y, Zhang H (2003) Learning a Locality Preserving Subspace for Visual Recognition[C]. In Proceedings of IEEE International Conference on Computer Vision Vol.1:385–392
16. Hu MK (1962) Visual pattern recognition by moment invariants[J]. IRE Trans Inf Theory 8(2):179–187
17. Hurst W, Wezel C (2013) Gesture-based interaction via finger tracking for mobile augmented reality[J]. Multimed Tools Appl 62:233–258
18. Ibraheem NA, Khan RZ (2012) Vision based gesture recognition using neural networks approaches: a review[J]. Int J Hum Comput Interact IJHCI 3(1):1–12
19. Junejo IN, Dexter E, Laptev I, Pérez P (2011) View-independent action recognition from temporal self-similarities[J]. IEEE Trans Pattern Anal Mach Intell 33(1):172–185
20. Khan R, Hanbury A, Stöttinger J, Bais A (2012) Color based skin classification[J]. Pattern Recognit Lett 33(2):157–163
21. Kim T-K, Wong S-F, Cipolla R (2007) Tensor canonical correlation analysis for action classification[C], In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 1–8
22. Li W, Deng C (2012) Fast and robust method for dynamic gesture recognition using hermite neural network[J]. J Comput 7(5):1163–1168
23. Martinez AM, Kak AC (2001) PCA versus LDA[J]. IEEE Trans Pattern Anal Mach Intell 23(2):228–233
24. Moeslund TB, Hilton A, Krüger V (2006) A survey of advances in vision-based human motion capture and analysis[J]. Comput Vision Image Underst 104(2–3):90–126
25. Mugavin ME (2008) Multidimensional scaling: a brief overview [J]. Nurs Res 57(1):64–68
26. Oikonomidis I, Kyriazis N, Argyros A (2011) Efficient model-based 3d tracking of hand articulations using kinect [C]. Br Mach Vis Conf 101.1–101.11
27. Roccetti M, Marfia G, Semeraro A (2012) Playing into the wild: a gesture-based interface for gaming in public spaces[J]. J Vis Commun Image Represent 23(3):426–440
28. Romero J, Kjellstrom H, Kragic D (2009) Monocular real-time 3D articulated hand pose estimation[C]. IEEE-RAS Int'l Conf Humanoid Robot :87–92
29. Rosales R, Athitsos V, Sigal L, Sclaroff S (2001) 3d hand pose reconstruction using specialized mappings[C]. IEEE Int Conf Comput Vis ICCV 1(1):378–385
30. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding[J]. Science 290(5500):2323–2326
31. Song Y, Tang S, Zheng YT et al (2012) Exploring probabilistic localized video representation for human action recognition [J]. Multimed Tools Appl 58(3):663–685
32. Stenger B, Mendonça PRS, Cipolla R (2001) Model-based 3D tracking of an articulated hand[C], Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Vol. 2:990–976. doi:10.1109/CVPR.2001.990976
33. Takahashi M, Fujii M, Naemura M et al (2013) Human gesture recognition system for TV viewing using time-of-flight camera[J]. Multimed Tools Appl 62(3):761–783
34. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction[J]. Science 290(5500):2319–2323
35. Vezhnevets V, Sazonov V, Andreeva A (2007) A survey on pixel-based skin color detection techniques[J]. Pattern Recog 40(3):1106–1122
36. Wang X, Xia M, Cai H, Gao Y, Cattani C (2012) Hidden-Markov-Models-Based Dynamic Hand Gesture Recognition[J]. Math Probl Eng, Vol 2012, Article ID 986134.11
37. Yen S-H, Wu C-M, Wang H-Z (2012) A block-based orthogonal locality preserving projection method for face super-resolution[J]. Intell Inf Database Syst Lect Notes Comput Sci 7197:253–262
38. Zachmann G (1998) Rapid Collision Detection by Dynamically Aligned DOP-trees[C]. In Proc. IEEE Virtual Reality Annual International Symposium, 90–97
39. Zhang Z, Wang J, Zha H (2012) Adaptive manifold learning[J]. IEEE Trans Pattern Anal Mach Intell 34(2):253–265

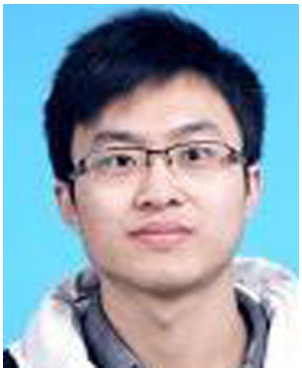
Yi Wang received the B.E. and Ph.D. degree, both in computer science from Jilin University, Changchun, China, in 2002 and 2009 respectively, Since July 2009, she has been a lecturer in the School of Software, Dalian University of Technology, Dalian, China. Her research interests include Computer Vision, Pattern Recognition, and Virtual Reality.

Zhongxuan Luo received Ph.D. in Computational Math in 1991 Dalian Univ. of Tech., China. From 1991 to 1993, he worked as a postdoctoral research fellow in Institute of Engineering Mechanics in Dalian Univ. From 1993 to 1997, he worked as Associate Professor Institute of Mathematical Sciences, Dalian Univ. of Tech., From 1997 to 2000, He worked as a research Fellow in Institute of Textiles & Clothing, in the Hong Kong Polytechnic University. From 2006 to 2007, he was a Visiting Professor in Michigan State Uni., Delaware State Uni. Since 1993, he has been full Professor Institute of Mathematical Sciences, Dalian Univ. of Tech. Since 1997, he has been Doctorial Supervisor, Department of Applied Mathematics, Dalian Univ. of Tech. His research interests include Computational geometry and graphics image processing, computer-aided geometric design, scientific computing

Juncheng Liu received the B.E. degree in software engineering from Dalian University of Technology, school of software, Dalian, China, in 2013, Since July 2013, he has been a postgraduate in the school of electronics engineering and computer science, Peking University, Beijing, China. His research interests include computer graphics, computer vision and virtual reality.

Xin Fan He received the B.E. and Ph.D. degree, both in Information and Communication Engineering, from Xi'an Jiaotong University in 1998 and 2004 respectively. From September to December 2000, he was an software engineer at Dalian Everspry CO., Ltd. Dr. Fan had been a visiting student at Microsoft Research Asia (MSRA) for 6 months since February 2002. He is an assistant professor (lecturer) in the Institute of Signal and Image Processing, Dalian Maritime University China since July 2005. He worked as a postdoctoral research fellow at the Visual Computing and Image Processing Laboratory (VCIPL) of Oklahoma State University from May 2006 to December 2007 and the University of Texas Southwestern Medical Center at Dallas from January 2008 to June 2009. Since November 2009, Dr. Fan was appointed as an associate professor at Dalian University of Technology (DUT) China. His research interests include digital multimedia technology, computer vision and image processing.

Haojie Li received the B.E. degree in computer science from Nankai University, Tianjin, China, in1996, and the Ph.D degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007. From1996 to 2001,he was with Yantai Post, Shan-dong, China, asa software engineer. From 2007 to 2009,he was a research fellow with the School of Computing, National University of Singapore, Singapore. Since December 2009, he has been an associate professor with the School of Software, Dalian University of Technology, Dalian, China. His research interests include computer vision, pattern recognition, and image/video retrieval.



Wu Yunzhen has received the B.E. degree in software engineering from Dalian University of Technology, school of software, Dalian, China, in 2011. He is now studying for a master's degree in engineering at Dalian University of Technology. His research interests include software engineering and computer vision.