

# Online learning for video probabilistic appearance manifolds recognition algorithm and its application in coal mines

*To monitor the operation of coal mine safety production, an online learning method of fault identification for coal mine safety production through video probabilistic appearance manifolds is proposed in this paper. For a category of the coal mine equipment safety state, a common representation of the normal appearances of this category would usually be learned off-line. From video monitoring of this category, an appearance model can be learned online through a prior generic model and successive video. The further details, as well as both the normal and abnormal appearances, can be expressed as an appearance manifold. In our algorithm, an appearance manifold would be approximately estimated by a series of sub-manifolds, and each sub-manifold is further refined into a low-dimensional linear sub-space. Thus, the time required for image recognition is reduced to meet the demands of real-time image processing. Through experimental analysis, we can demonstrate that our online learning algorithm method is an efficient method for video-based image recognition, and its application in coal mine safety production has proven to be very effective.*

**Keywords:** Probabilistic appearance manifolds, online learning, image recognition, coal mine.

## 1. Introduction

Because of the rapid improvement of video cameras and computer networks, it is a very definite possibility that online learning will be directly used to recognize appearance manifolds by video streams, and some useful real-time applications such as coal mine safety production have been constructed. Existing related algorithms can only perform the recognition process in an online environment, but the training process is always implemented in an off-line environment. This means that all training video data must be captured prior, and the changes in the video cannot be reflected in real time. So, off-line training algorithms are not suitable for real-time tasks. However, online learning algorithms are practical for processing and training real-time video streams.

Coal mine production is a high-risk production process, as there are many safety risks. There are many video monitoring systems of coal mine safety production being used in underground work situations. The captured video sequences can be used to discover hidden dangers in a timely manner.

In this paper, an online learning method based on probabilistic appearance manifold for coal mine monitoring videos is proposed. Probabilistic appearance manifold [1], which is shown in Fig.1, can be modelled as a series of sub-manifolds in image space and the connection relation between these manifolds. Probabilistic appearance manifold was learned through a training process from video frames. In the training process, the training video was firstly classified into several clusters by the K-means algorithm[2]. Image frames allocated to the same cluster generally come from analogous poses. Principal component analysis (PCA)[3] is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables. In our algorithm, we use PCA to divide each cluster into a series of low dimensional linear components named sub-spaces. The connection relation among these components can be expressed by a matrix, in which adjacent elements represent successive frames.

The basic process of probabilistic appearance manifold is evolving a common appearance manifold  $M$  to a concrete manifold  $M_k$  from a series of video frames. The non-linear appearance manifold  $M$  can be approximately equivalent to several simpler linear components and their connection relations. In Fig.1, each  $C^i$  stands for a principal component analysis component, and the connection relations between

---

Messrs. Deyong Wang, School of Resource and Environmental Engineering, Wuhan University of Technology and Deyong Wang and Wei Wang, School of Information Engineering, PingDingShan University. E-mail: compclub@163.com

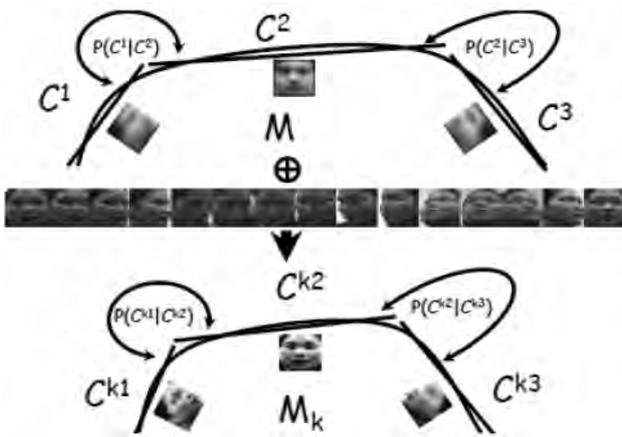


Fig.1 Probabilistic appearance manifold

these components are described by the probability  $P$ .

Our online learning algorithm is very different from conventional appearance manifold. Appearance manifold in our algorithm is derived from a multiple training video frames dataset which includes various instances of that category. In each time step, only one frame which is used to update appearance manifold is available in the frame sequence.

Our online learning algorithm is divided into two steps. The first step is appearance estimation; in this step our aim is to discover the best sub-manifold. The second step is gradually updating the final appearance manifold. The aim of the second step is to update all sub-spaces in the appearance manifold to minimize distortions.

The rest of this paper is organized as follows. In section 2, we introduce the relevant work. In section 3, a mathematical framework for the algorithm is introduced. An image motion tracking algorithm is proposed in section 4. In section 5, experiments and a results analysis are given. Section 6 is the conclusion.

## 2. Related work

Although there are a large number of existing appearance of recognised algorithms, studies of online learning algorithms are few [4-6].

Brand et al.[4] proposed an incremental algorithm to track and identify appearance through sub-space. Since the appearance manifold is nearly non-linear, only one sub-space may be inadequate. Therefore, it is difficult to track great changes in appearance.

According to the neighborhood sub-space of appearance manifold, Ho et al.[1] proposed an online learning algorithm by using the most recent video frames. The method can resolve the problem of tracking great changes in appearance, but the algorithm does not recognize the whole appearance manifold, so its application is restricted.

Jepson et al. [5] proposed a mixed model for online learning

algorithms for video appearance recognition. The model depends on different pixels' distribution to capture an image's appearance.

Morency et al. [7] introduced a view-based algorithm for appearance recognition by eigenspaces. A separate PCA model for image frame sets is given in this paper.

Cootes et al. [8] proposed a multiple-view appearance algorithm which could capture an object's appearance through views. The algorithm can infer an object's appearance by matching approximate poses and views.

## 3. Mathematical foundation of our algorithm

The appearance manifold is represented by  $M$ , which is made of  $m$  non-intersecting sub-manifolds  $= C^1 \cup C^2 \cup \dots \cup C^m$ , where  $C^i$  stands for  $i$ th sub-manifold. Each  $C^i$  can be obtained by linear approximation through PCA component analysis. During the whole derivation process, because of the appearance of the object, each  $C^i$  acts as a pose sub-space.

In our algorithm,  $M$  can be obtained through a simple training process. First we assign a series of video frames with analogous appearance to  $m$  clusters, and then PCA is applied to each cluster to identify the pose sub-space  $C^i$ . In addition, for each video training data set, the average image in each pose is computed.

For each sub-space  $C^i$ , a series of training samples  $\{t_1^i, t_2^i, \dots, t_w^i\}$  from  $W$  object, where  $t_j^i$  represents the average image of object  $j$  shown in pose  $i$ .

Let  $\{F_1, F_2, \dots, F_n\}$  represent a video frame sequence of object  $o$ , and let  $R_t$  be a scope containing an object clipped from  $F_t$ . Each image  $I_t$  in the training video frame sequence is a sample drawn from the appearance manifold  $M_k$ . The aim of our algorithm is updating the appearance manifold  $M$  through identifying image  $I_t$  at time  $t$ .

According to evolving  $M$  to  $M_k$ , our algorithm can be divided into two steps. The first step is estimating the pose. The probabilistic estimation of the  $C_t^{i*}$  given the current image  $I_t$  and the previous estimation  $C_{t-1}^j$ , can be expressed as:

$$C_t^{i*} = \arg \max_i p(C_t^i | I_t, C_{t-1}^j) \quad \dots \quad (1)$$

The second step is updating the  $M$  in order to minimize the appearance recognition error, whose formal definition can be written as follows:

$$E_{or}^2(M, \{I_1, I_2, \dots, I_{t-1}, I_t\}) \quad \dots \quad (2)$$

where  $\{I_1, I_2, \dots, I_{t-1}, I_t\}$  represents the previous images in the video. But  $\{I_1, I_2, \dots, I_{t-1}, I_t\}$  do not need to be retained in our online learning algorithm because the useful data on  $M$  have been characterized in each  $C^i$ .

### 3.1 APPEARANCE ESTIMATION

In our algorithm, the  $C^i$  which maximizes the probability  $p(C^i | I_t, C_{t-1}^j)$  in Eq. (1) is chosen. We can suppose that for a given  $C_t^i$ ,  $I_t$  and  $C_{t-1}^j$  are independent. The probability  $p(C^i | C^j)$  is time invariant. According to the above assumptions, we can draw the following equation:

$$p(C_t^i | I_t, C_{t-1}^j) = \frac{p(I_t | C_t^i, C_{t-1}^j) p(C_t^i | C_{t-1}^j)}{p(I_t | C_t^i) p(C^i | C^j)} \quad \dots \quad (3)$$

where  $b$  is a constant to ensure a correct probability distribution.

$C^i$  which can be expressed by an affine sub-space,  $S = (c, \mathfrak{G}, \Lambda, P)$  where  $c$  is the center of the sub-space,  $\mathfrak{G}$  is the eigenvector matrix,  $\Lambda$  is the diagonal matrix of eigenvalues, and  $P$  is the quantity of video samples which were used to identify the sub-space.

The linear transformation from  $I_t$  to  $S$  can be defined as follows:

$$y = (y_1, y_2, \dots, y_M) = (\mathfrak{G})^T (I_t - c) \quad \dots \quad (4)$$

So the similarity probability can be written as:

$$p(I_t | C^i) = p(I_t | S) = \frac{\exp(-\frac{1}{2} (\sum_{r=1}^M \frac{y_r^2}{\lambda_r}))}{(2\pi)^{\frac{M}{2}} \prod_{r=1}^M \frac{1}{\lambda_r}} \left[ \frac{\exp(-\frac{1}{2} d^2(I_t, S))}{(2\pi\rho)^{\frac{N-M}{2}}} \right] \quad \dots \quad (5)$$

where  $N$  represents the space of the image,  $M$  represents the sub-space of a pose, and  $S$  and  $d^2(I_t, S)$  represent the distance between a video image  $I_t$  and sub-space  $S$ . We select the parameter  $\rho$  as  $\frac{1}{2} \lambda_{M+1}$ .

The geometric interpretations of Eq.(5) are shown in Fig.2.

The probability  $p(C^i | C^j)$  denotes the transition

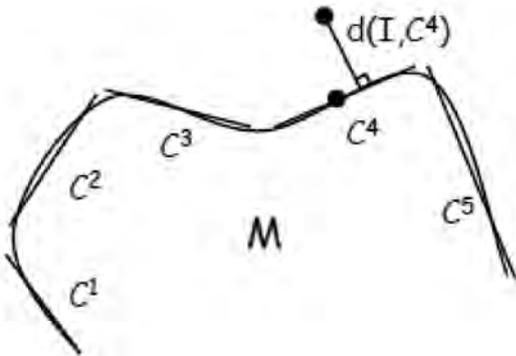


Fig.2 Geometric interpretation of pose sub-space

probability between sub-spaces. It can capture the motion of an object in the training image frame. The  $p(C^i | C^j)$  represents the motion possibilities from  $C^i$  to  $C^j$  if the two poses are discrete.

### 3.2 VIDEO IMAGE APPROXIMATION

Before updating the appearance manifold  $M$ , we should determine which sub-space  $C_t^{i*}$  the  $I_t$  belongs to. The specific steps are as follows:

If the current machine state in the training video looks like a combination of some states from the pre-training data set, then the machine state probably is the same combination of other states in all other appearances as well.

More specifically, we would discover a series of  $K$  nearest neighbours  $\{z_1^{i*}, z_2^{i*}, \dots, z_K^{i*}\}$  of  $I_t$  from the pre-training sample video images  $\{x_1^{i*}, x_2^{i*}, \dots, x_Q^{i*}\}$  for the sub-space  $C_t^{i*}$ .

The  $K$  nearest neighbours can be used to approximate  $I_t$  through a set of coefficients  $w_r$  which would be used to construct the objective function:

$$\min \left\| I_t - \sum_{r=1}^K w_r z_r^{i*} \right\|_{L^2}^2 \quad \dots \quad (6)$$

Let  $\{z_1^{j*}, z_2^{j*}, \dots, z_K^{j*}\}$  represent the average image frames of another sub-space  $C^j$ , where  $z_r^j$  and  $z_r^{i*}$  contain the same object appearance.

The coefficients  $w_r$  in Eq.(6) can be used to compute  $I_t^j$  for pose  $j$  through the image frame set  $\{z_1^j, z_2^j, \dots, z_K^j\}$  through the following formula:

$$I_t^j = \sum_{r=1}^K w_r z_r^j \quad \dots \quad (7)$$

The final results would be a series of machine images for all other appearances.

### 3.3 SUB-SPACE UPDATE

After we have processed the images for each sub-space  $C^i$ , we should update the eigenspace model of  $C^i$  with the sample.

There were a lot of available algorithms based on updating the eigenbases [9-12]. But almost all algorithms updated the eigenbases without storing the previous training samples. In this paper, we use an algorithm that gradually updates the sub-space for a fixed appearance.

Supposing that we are going to gradually update the current sub-space  $S$  specified by  $(c, \mathfrak{G}, \Lambda, P)$ , an updating

sub-space  $S'$  specified by  $(c', \vartheta', \Lambda', P+1)$  with the new observing variable can minimize the reconstruction error. So the above problem is in fact equal to solving the eigenproblem.

$$T'\vartheta' = \vartheta' \Lambda' \quad \dots \quad (8)$$

where  $T'$  is the new covariance matrix,  $\vartheta'$  is the new eigenvector matrix, and  $\Lambda'$  is the new diagonal matrix of the eigenvalues.

The projection of the current sub-space  $S$  can be defined as follows:

$$g = \vartheta'^T \bar{x} \quad \dots \quad (9)$$

The orthogonal residue vector can be concluded as follows:

$$h = \bar{x} - \vartheta' g \quad \dots \quad (10)$$

where  $\bar{x} = x - c$ .

So  $c'$  and  $S'$  can be defined individually as follows:

$$c' = \frac{1}{P+1}(Pc + x) \quad \dots \quad (11)$$

$$S' = \frac{P}{P+1}S + \frac{P}{P+1} \frac{\bar{x}\bar{x}^T}{x^T x} \quad \dots \quad (12)$$

The old sub-space can be expanded through adding the new variable  $x$  to its dimension. It can be done through adding the vector  $\hat{h}$  to construct an orthonormal basis based on  $[\vartheta', \hat{h}]$ , where  $\hat{h}$  is defined as follows:

$$\hat{h} = \begin{cases} \frac{h}{\|h\|_2} & \text{if } \|h\|_2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \dots \quad (13)$$

The rotation matrix  $R$  is the pivotal point to translate the orthonormal basis  $[\vartheta', \hat{h}]$  to the eigenbasis  $\vartheta'$  which expands the sub-space  $C'$ .

$$\vartheta' = [\vartheta', \hat{h}]R \quad \dots \quad (14)$$

According to Eqs. (8), (12), and (14), we can obtain the following equation:

$$[\vartheta', \hat{h}]^T \left( \frac{P}{P+1}S + \frac{P}{P+1} \frac{\bar{x}\bar{x}^T}{x^T x} \right) [\vartheta', \hat{h}] R = R\Lambda' \quad \dots \quad (15)$$

So  $S$  can be expressed as:

$$S \approx \vartheta' \Lambda \vartheta'^T \quad \dots \quad (16)$$

Through Eqs. (15) and (8), Eq. (14) can be further expressed as:

$$\left( \frac{P}{P+1} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} + \frac{P}{(P+1)^2} \begin{bmatrix} gg^T & \gamma g \\ \gamma g & \gamma^2 \end{bmatrix} \right) R = R\Lambda' \quad \dots \quad (17)$$

where  $\gamma$  represents  $\hat{h}^T \bar{x}$ .

Now all variables have been updated in the new pose sub-space. The new  $c'$  is updated by Eq. (10). The new eigenvalues  $\Lambda'$  can be obtained directly by the eigenproblem in Eq. (16), and the new eigenvectors  $\vartheta'$  can be obtained by Eq. (14) through  $R$ .

The rotation matrix  $R$  is  $(M+1) \times (M+1)$ , where  $M$  is the dimension of the sub-space. Because the value of  $M$  is very small, Eq.(17) can be computed effectively in real-time.

#### 3.4 CONSTRUCTING MODEL AND LEARNING PROCESS

For each image from the online training video of the  $k$ -th machine, we will determine which pose sub-space  $C^{i^*}$  it belongs to by Eqs. (1), (2), and (4). Once the sub-space is detected, we will prefabricate all the possible images in other poses by Eqs.(5) and (6). Through these images, we will use all the formulas from Eqs.(7) to (17) to update the variables denoting all sub-spaces. Finally, the transition matrix can be obtained by adding up the transitions between various pose manifolds found in the video frame sequence:

$$p(C^{ki} | C^{kj}) = \frac{T_{ij} p(C^{ki} | C^{kj}) + \delta(I_t \in C^{ki}) \delta(I_{t-1} \in C^{kj})}{T_{ij} + \delta(I_t \in C^{ki}) \delta(I_{t-1} \in C^{kj})} \quad \dots \quad (18)$$

If there is the smallest probabilistic to sub-space  $C^{ki}$  for  $I_t$ ,  $\delta(I_t \in C^{ki}) = 1$  and otherwise it would be 0.  $T_{ij}$  represents the accumulation value of transitions between  $C^{ki}$  and  $C^{kj}$ .

The advantage of our online learning recognition algorithm is demonstrated by our experiments. The input video frame is shown in Fig.3.

Our algorithm is compared to another conventional online updating algorithm. The comparison results are shown in Fig.4.

From Fig. 4, we can see that our algorithm shown in Fig. 4(b) is to assess the sub-space and its updating effect is better than that of a conventional algorithm shown in Fig. 4 (a).



Fig.3 Two consecutive images of input video frame

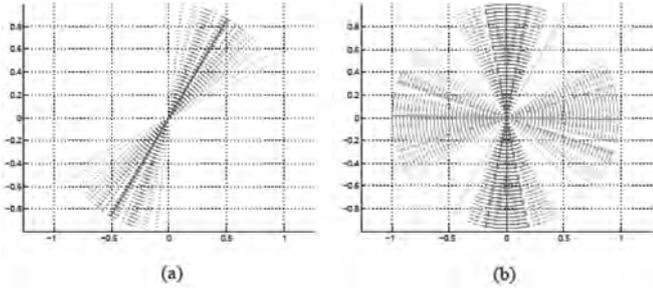


Fig.4 Comparison results of different algorithms

#### 4. Image motion tracking

Our algorithm can be further extended to an image motion tracking algorithm. Let  $M$  represent the common appearance manifold which consists of a set of sub-spaces  $C^i$ , and let  $\{F_1, F_2, \dots, F_{s-1}, F_s\}$  represent a sequence of  $s$  image frames. In the tracking algorithm, we will assess the object's location in each image frame  $F_t$ .

$GL(o)$  is a linear group which consists of  $o \times o$  non-singular matrices. A sub-group  $SO(o)$  is a sub-manifold of  $GL(o)$ ; it includes  $o^2$  elements and  $o+o(o-1)/2$  constraints.

For the  $o \times o$  matrix  $I$ , the space  $T_I(SO(o))$  of the element of  $SO(o)$  is a set of  $o \times o$  skew-symmetric matrices. For a point  $o \in SO(o)$ , the space at the point would be computed by a rotation of  $T_I(SO(o))$  as follows:

$$T_o(SO(o)) = \{OX \mid X \in T_I(SO(o))\} \quad \dots \quad (19)$$

With the above formula, the metric  $SO(n)$  becomes a Riemannian manifold. By using the Riemannian manifold, lengths of paths on a manifold can be defined.

We define  $\alpha : [0,1] \rightarrow SO(o)$  to be a path on  $SO(o)$  which is different everywhere on section  $[0,1]$ . For any two points  $P_1, P_2 \in SO(o)$ , which can define a distance between them, the distance starts at  $P_1$  and ends at  $P_2$ :

$$d(P_1, P_2) = \inf_{\{\alpha : [0,1] \rightarrow SO(o) \mid \alpha(0) = O_1, \alpha(1) = O_2\}} \left( \int_0^1 \sqrt{\left\langle \frac{d\alpha(t)}{dt}, \frac{d\alpha(t)}{dt} \right\rangle} dt \right) \quad \dots \quad (20)$$

To help track image motion, we illustrate the motions of tangent planes in Fig.5.

The motions of the exponential map are shown in Fig.6.

Let  $f(u, F_t)$  represent the mapping function which can give a sub-image  $I$  cropped from the rectangular region.

Our image motion tracking algorithm can be defined as an optimization problem as follows:

$$u_t^* = \arg \max_u p(f(u, F_t) \mid C_{t-1}^j) \quad \dots \quad (21)$$

where  $p(I \mid C_{t-1}^j)$  had been defined in Eq. (5); it represents the similarity between the image  $I$  and sub-space  $C_{t-1}^j$ . The above

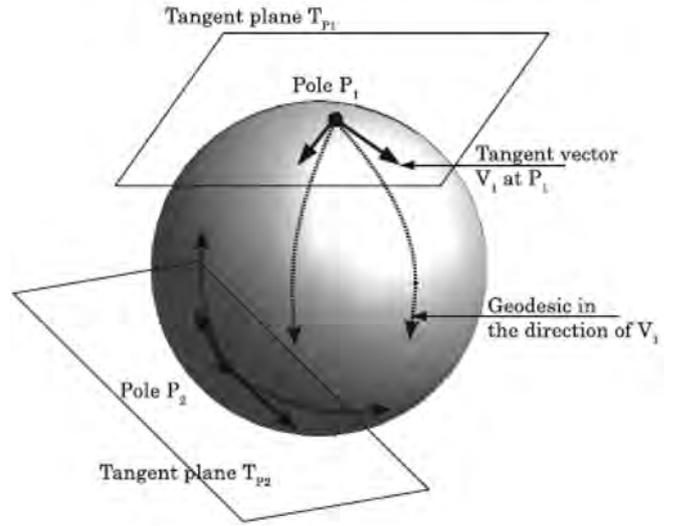


Fig.5 Illustration of tangent spaces of image motion

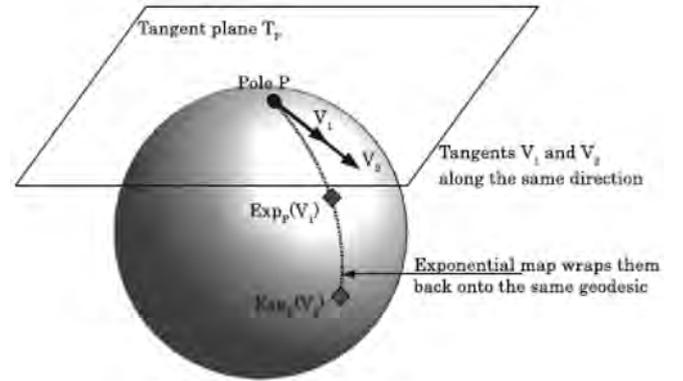


Fig.6 Exponential map of image motion

Eq. (21) can be evaluated by a set of sub-images.

Our image motion tracking algorithm can be defined in detail as follows:

Input parameters:  $(\varphi, S)$ :

$\varphi = \{\omega_x, \omega_y, \omega_w, \omega_h, \omega_\theta\}$  represents a collection of 5 parameters for sampling regions.

$S$  represents the number of regions for each frame.

Output:  $(I^*, u^*)$ :

$I^*$  denotes the image of the tracked object.

$u^*$  denotes the region position of  $I^*$  on the screen.

Model parameters:  $(m, n, C^i, T, u^*)$ :

$m$  represents the number of sub-spaces of the appearance manifold  $M$ .

$n$  represents the number of dimensions of the linear sub-spaces  $C^i$ .

$C^i$  represents the  $i$ -th pose sub-space, which is denoted by a collection of images.

$T$  represents an  $m \times m$  probability transition matrix for the tracked object where each point is the probability  $p(C^i | C^j)$ .

$u^* = (x, y, w, h, q)$  represents a rectangular region in the image centered at  $(x, y)$  and of the size  $(w, h)$  with the orientation  $q$ .

Through our tracking algorithm, we can construct a pyramid structure for the two video frame sequences  $P$  and  $Q$ , and in Fig. 7, we can plot the distribution of all of the video frames from the two frame sequences  $P$  and  $Q$ .

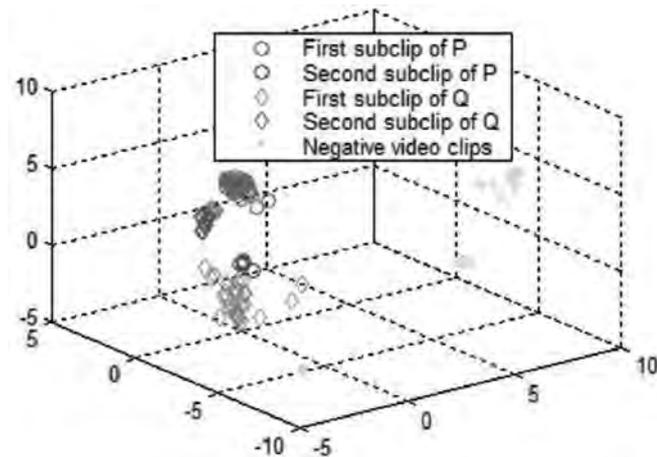


Fig.7 Projection of image frames of videos P and Q

## 5. Experiments

We have conducted experiments with large video data sets to compare our algorithm with the SLEMD method[13]. In the experiment process, we called our algorithm and the SLEMD method level-1 and level-2, respectively.

The average precision of the two algorithms for image frame sequences under No Align is shown in Fig.8.

All of the motion tracking of the video frame sequences was also compared to other recognition algorithms. The comparison results are shown in Table 1.

The comparison of the probabilistic manifold recognition using our online algorithm, off-line algorithm [1], as well as three other conventional recognised algorithms is shown in Table 1. The comparison results show that our online learning recognised algorithms are much better than other conventional recognised algorithms.

TABLE 1: COMPARISON OF MOTION IMAGE RECOGNISED ALGORITHMS

Algorithm	Accuracy	
	Videos w/o occlusion	Videos with occlusion
Online learning	93.7	91.2
Off-line learning	95.2	88.7
Eigen-recognition	68.2	57.2
Fisher-recognition	70.2	63.4
Nearest neighbour	81.6	77.3

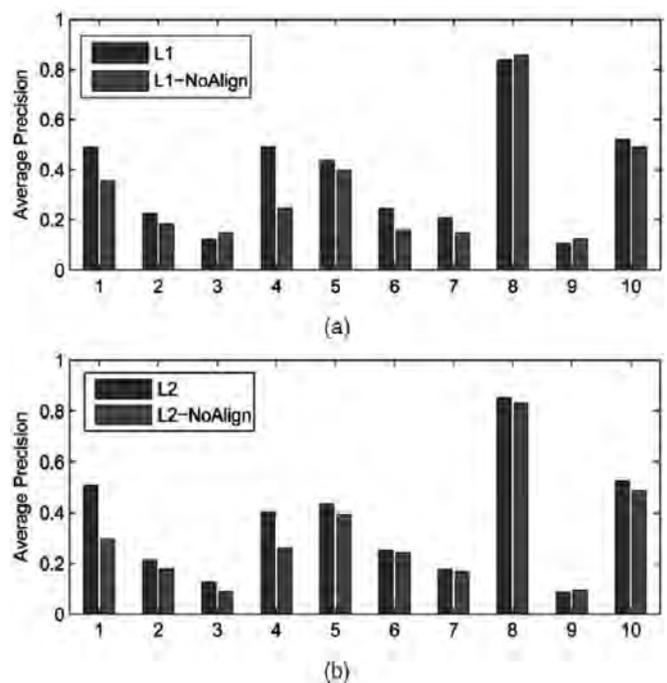


Fig.8 Comparison with different temporal boundaries for the two algorithms

## 6. Conclusions

In this paper, a novel online learning recognised algorithm through constructing an appearance manifold from a video frame sequence is proposed. Our algorithm's effectiveness for video-based machine recognition and tracking in coal mines has been demonstrated through a considerable amount of experimental work. But our algorithm for tracking image motions can not track those objects without a known class. More research needs to be done in the future on how to track unknown objects.

## References

1. Lee, K. C., Ho, J., Yang, M. H. and Kriegman, D. (2003): "Video-based face recognition using probabilistic appearance manifolds," in Proc. IEEE Conf. on Comp. Vision and Patt. Recog, 2003, pp.313–320.
2. Krishna, K. and Narasimha, M. (1999): "Genetic K-means algorithm," *IEEE Transactions on Systems Man & Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999. DOI: 10.1109/3477.764879.
3. Pentland, A., Moghaddam, B. and Starner, T. (1994): "View-based and modular eigenspaces for face recognition," in Proc. IEEE Conf. on Comp. Vision and Patt. Recog, 1994, pp.84–91.
4. Brand, M. (2002): "Incremental singular value decomposition of uncertain data with missing values," in Proc. European Conf. on Computer Vision, 2002, pp.707–720.

5. Jepson, A., Fleet, D. and Maraghi, T. (2003): "Robust online appearance models for visual tracking," in Proc. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, pp.1296–1311.
6. Ross, D., Lim, J. and Yang, M. H. (2004): "Adaptive probabilistic visual tracking with incremental subspace update," in Proc. European Conf. on Computer Vision, 2004, pp.470–482.
7. Morency, L., Rahimi, A. and Darrell, T. (2003): "Adaptive viewbased appearance model," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2003, pp.803–810.
8. Cootes, T. F., Edwards, G. J. and Taylor, C. J. (2001): "Active appearance models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–684, 2001. DOI: 10.1109/34.927467.
9. Levy, A. and Lindenbaum, M. (2003): "Sequential karhunen-loeve basis extraction and its application to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371–1374, 2003. DOI: 10.1109/83.855432.
10. Pham, V. T. and Tran, T. T. (2014): "Multiphase B-spline level set and incremental shape priors with applications to segmentation and tracking of left ventricle in cardiac MR images," *Machine Vision and Applications*, vol. 25, no. 8, pp. 1967–1987, 2014. DOI: 10.1007/s00138-014-0626-1.
11. Li, J. and Han, G. (2011): "Robust tensor subspace learning for anomaly detection," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 89–98, 2011. DOI: 10.1007/s13042-011-0017-0.
12. Tian, Q. and Chen, S. (2016): "Ordinal margin metric learning and its extension for cross-distribution image data," *Information Sciences*, vol. 345, no. 7, pp. 50–64, 2016.
13. Naphade, N. and Smith, J. (2006): "Large-Scale Concept Ontology for Multimedia," *IEEE Multimedia*, vol. 13, no. 3, pp. 86–91, 2006.

## INVERSION OF RHEOLOGICAL PARAMETERS OF SURROUNDING ROCKS IN A MINE ROADWAY BASED ON BP NEURAL NETWORK

*Continued from page 114*

9. Qiao, J. F. and Han, H. G. (2012): "Modeling and identification of nonlinear dynamic systems using a novel self-organizing RBF-based approach," *Automatica*, vol. 48, no. 8, pp. 1729-1734, Aug. 2012. DOI: 10.1016/j.automatica.2012.05.034.
10. Singh, K. P., Basant, A., Malik, A. and Jain, G. (2009): "Artificial neural network modeling of the river water quality—a case study," *Ecological Modelling*, vol. 220, no. 6, pp. 888-895, Mar. 2009. DOI: 10.1016/j.ecolmodel.2009.01.004.
11. Wei, R. L. (1993): "Predication of Coefficient of Consolidation from Actually-measured Subsidence," *Chinese Journal of Geotechnical Engineering*, vol. 15, no. 2, pp. 12-19, Mar. 1993.
12. Gong, X. N. (1991): "Back Analysis Method for Determinating Material Parameters in Foundation Consolidation Process," *Chinese Journal of Applied Mechanics*, no. 2, pp. 131-133, 1991.
13. Yang, Z. F. (2002): "Back Analysis Principles and Application in Geotechnical Engineering," Beijing: Seismological Press, pp. 63-65, 2002.
14. Shang, Y. Q. (1999): "Method for Determining Boundary Conditions in Analogy Model for Rock Mass Stabilization and Region Stability Values," *Rock Mechanics and Engineering*, pp. 201-204, 1999.
15. Liu, W. Q. (2005): "Experimental Design," Beijing: Tsinghua University Press, pp. 64-65, 2005.
16. Hagan, M. T. (2014): "Neural network design," Wseas International Conference on Circuits, Martin Hagan, 2014.
17. Gregor, K., Danihelka, I., Graves, A. and Wierstra, D. (2015): "DRAW: A Recurrent Neural Network For Image Generation," *Computer Science*, 2015.
18. Yu, F. and Xu, X. Z. (2014): "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Applied Energy*, vol. 134, no. 134, pp. 102-113, Dec. 2014. DOI: 10.1016/j.apenergy.2014.07.104
19. Huo, L., Jiang, B., Ning, T. and Yin, B. (2014): "A BP Neural Network Predictor Model for Stock Price," *Intelligent Computing Methodologies*. pp. 362-368, 2014. DOI: 10.1007/978-3-319-09339-0\_37.