

Hailong ZHU, Peng LIU, Jiafeng LIU, Xianglong TANG

A primary-secondary background model with sliding window PCA algorithm

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2011

Abstract Rain and snow seriously degrade outdoor video quality. In this work, a primary-secondary background model for removal of rain and snow is built. First, we analyze video noise and use a sliding window sequence principal component analysis de-noising algorithm to reduce white noise in the video. Next, we apply the Gaussian mixture model (GMM) to model the video and segment all foreground objects primarily. After that, we calculate von Mises distribution of the velocity vectors and ratio of the overlapped region with referring to the result of the primary segmentation and extract the interesting object. Finally, rain and snow streaks are inpainted using the background to improve the quality of the video data. Experiments show that the proposed method can effectively suppress noise and extract interesting targets.

Keywords sliding window sequence principal component analysis, primary-secondary background model, removal of rain and snow, Gaussian mixture model (GMM)

1 Introduction

Complicated climate conditions, such as rain, snow, fog, and illumination variation, cause great interference to video analysis, especially for small object detection, recognition, and tracking. Noises produced by camera equipment make it worse. Therefore, many studies involving noise suppression, eliminating weather influence to improve visual effect, and extracting interesting objects have been developed for outdoor video. For example, the methods of object detection for illumination changing [1–3] and fog processing in a space-time field on the pixel level [4] were proposed recently. Also, video processing

now focuses on how to erase the scintillation effect caused by physical particles, such as raindrops and snowflakes. Generally, there are two methods to do this: one is that non-interesting objects, such as raindrops and snowflakes described above, are directly viewed as background [5,6]; the other one is that raindrops and snowflakes in the near range are viewed as special foreground objects and filtered. Finally, the corresponding areas are inpainted using the background [7–11].

Stauffer et al. [5] built a Gaussian mixture model (GMM) for whole video, which can train and update its parameters online. The GMM processes raindrops and snowflakes as background, so it does not need any other work. This is because those raindrops and snowflakes dropping quickly close to the camera lens will form rain and snow streaks in an image since the exposure time of a camera is too long compared to the dropping speed of raindrops and snowflakes. GMM cannot distinguish rain and snow streaks from common moving objects, which is the so-called “myopia effect”. Greenspan et al. [6] proposed a piecewise Gaussian mixture model (PGMM) method which clusters active targets in the space-time field of the video feature space. PGMM can detect video events and edit images. It performs well when using PGMM to detect big objects moving slowly, but not so good with rain and snow streaks with high speed. Because it cannot split video to slices perfectly in such a scene, PGMM cannot build an expected model.

Garg et al. [7–9] studied the physical characteristics of raindrops and proposed a rain removal algorithm that has the ability of detecting rain streaks and rendering them as well. Barnum et al. [10,11] transformed an image to the frequency domain through Fourier transformation and simulated rain and snow streaks employing blurred Gaussian streaks in a spatio-temporal streak model. After that, the authors found the frequency range of rain and snow streaks and filtered them finally. Barnum’s model works well despite the sensitivity to the size and orientation of rain and snow streaks. In Ref. [11] the raindrop’s visual features were analyzed thoroughly and a

Received February 21, 2011; accepted April 7, 2011

Hailong ZHU (✉), Peng LIU, Jiafeng LIU, Xianglong TANG
School of Computer Science and Technology, Harbin Institute of
Technology, Harbin 150001, China
E-mail: zh104512004@yahoo.com.cn

rain removal method by changing the working parameters of a camera was proposed, which can avoid thinking of a complex rain and snow removal algorithm. A slight drawback of that model is that it is difficult to master all the model's adjusting parameters. Starik et al. [12] tackled the rain and snow streaks as a kind of particular foreground object and removed them using the median filter algorithm whose aliasing effect makes activity targets blurred. Zhang et al. [13] solved the problem in the way of k -means algorithm after an exact image registration and histogram distribution analysis.

The above methods merely focus on rain and snow removal and neglect the fact that video may be polluted by noises. In this paper, a primary-secondary background model is proposed to extract interesting objects from video with noise. In Fig. 1, a sliding window principal component analysis (SWPCA) algorithm is derived, the result of which is a quantity of principal components of image sequences. The current image is projected to the subspace comprised of those principal components to get the feature image. The outputs of SWPCA are input to GMM to perform the primary foreground segmentation and the binary image sequence is obtained containing all moving objects. Finally, the secondary foreground segmentation algorithm is performed according to different motion characteristics of objects to classify them. Thus, we know the space information of rain and snow streaks and use it to remove the streaks to get the purpose of video enhancement.

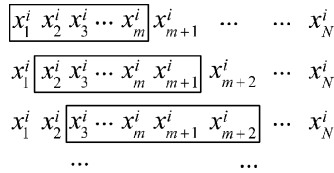


Fig. 1 Sliding window

2 Principal component analysis (PCA) method for noise suppression

2.1 Noises in outdoor video

Cameras working outdoor mostly are made of CCD/CMOS; some noises interfuse with the image sequence while imaging and transmitting, such as photon noise, readout noise, and fixed pattern noise. In addition, the environmental factors, higher or lower temperature, lack of illumination intensity, etc., lead to the image signal-to-noise ratio (SNR) decline. The power spectrum density of noises follows overall a Gaussian distribution, thus they can be reduced as white noise [14].

2.2 Image reconstruction by PCA

Let $W \times H$ denote image size, CH_1, CH_2, \dots, CH_c denote c

channels. Pixel values at the same position and the same channel in N frames compose the input vector $(\mathbf{x}^i)^T = (x_1^i, x_2^i, \dots, x_N^i)^T$, $i = 1, 2, \dots, W \times H$. For convenience, we omit the superscript i and denote $\mathbf{x} = \mathbf{x}^i - \bar{\mathbf{x}} = (x_1, x_2, \dots, x_N)^T$ as zero mean vector and its mathematical expectation $\bar{\mathbf{x}} = E\{\mathbf{x}^i\}$. The covariance matrix of \mathbf{x} is

$$\mathbf{C}_x = E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \quad (1)$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N) \in \mathbb{R}^{N \times N}$. From Eq. (1) N eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ and corresponding orthonormal eigenvector matrix $\mathbf{V} = (\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N)$ of the real symmetric matrix \mathbf{C}_x are calculated, where $\mathbf{V}_j = (v_{j1}, v_{j2}, \dots, v_{jN})^T$, $j = 1, 2, \dots, N$. Let w_T denote the weight of image information. If $\sum_{j=1}^l \lambda_j \geq w_T$ ($l \leq N$) that lets the first l eigenvectors $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_l$ as the basis of the feature subspace \mathcal{S}_l . The data of the same channel of the current image are projected to \mathcal{S}_l through the Hotelling transformation result in the point \mathbf{y} in subspace is

$$\mathbf{y} = \mathbf{S}_l^T \mathbf{x}, \quad (2)$$

and the current image is reconstructed as

$$\hat{\mathbf{x}} = \mathbf{S}_l^T \mathbf{y}. \quad (3)$$

Using Eq. (3) to all c channels, the whole feature image will then be reconstructed. Because $\mathbf{V}_{l+1}, \mathbf{V}_{l+2}, \dots, \mathbf{V}_N$ contain mostly noise and are cast away, it is obviously an offline image reconstruction method.

3 Images reconstruction by SWPCA

It is known that the calculating cost of offline method is not endured owing to mass data of surveillance video. In this paper, an SWPCA de-noising algorithm is proposed for image reconstruction, which is an online one and based on time-space correlation of video. In SWPCA, outputs of the algorithm are direct inputted to GMM to build a background model. Starting from the first element of the input vector, getting m elements x_1, x_2, \dots, x_m successively to construct a window whose width is m , the window's principal components are estimated in the window, and the feature image is reconstructed subsequently; when the $(m + 1)$ th frame comes, the window slides one frame forward and the window's width stays invariant, x_2, x_3, \dots, x_{m+1} is a new window with width m . The above steps are repeated and the principal components are updated while the window slides, so that the principal components always follow the dynamic scene.

Now we formulate the SWPCA as follows.

From Eqs. (1)–(3) current image is reconstructed using the first principal component

$$\hat{\mathbf{x}}(k) = \mathbf{V}_1 \mathbf{y}_1(k) \Rightarrow \mathbf{y}_1(k) = \mathbf{V}_1^T \mathbf{x}(k) = \sum_{p=1}^m v_{1p} x_p(k), \quad (4)$$

where k is the iteration times, m is the window width. Its cost function is

$$J_1(\mathbf{V}_1) = E \left[\|\mathbf{x}(k) - \mathbf{V}_1 \mathbf{V}_1^T \mathbf{x}(k)\|_2^2 \right], \quad (5)$$

and the equivalent form is

$$\begin{cases} J_1(\mathbf{V}_1) = \frac{1}{2} \|\mathbf{e}_1(t)\|_2^2 = \frac{1}{2} \sum_{p=1}^m e_{1p}^2(t), \\ \mathbf{e}_1(t) = \mathbf{x}(t) - \mathbf{V}_1 \mathbf{y}_1(t), \\ \mathbf{y}_1(t) = \mathbf{V}_1^T \mathbf{x}(t), \end{cases}$$

where $\mathbf{e}_1(t)$ is the residual error between origin image and the reconstruction one at time t using the first principal component. Calculating the derivation for both sides of Eq. (5), we can get the result with an iteration step parameter α :

$$\frac{dv_{1p}}{dt} = -\alpha_1 \frac{\partial J_1(\mathbf{V}_1)}{\partial v_{1p}} = -\alpha_1 [\mathbf{y}_1 e_{1p}(t)],$$

$$p = 1, 2, \dots, m,$$

and then,

$$\frac{d\mathbf{V}_1}{dt} = \alpha_1 \mathbf{y}_1 \mathbf{e}_1 = \alpha_1 \mathbf{y}_1 (\mathbf{x} - \mathbf{V}_1 \mathbf{y}_1).$$

We have the iterative equation for SWPCA by the first principal component:

$$\tilde{\mathbf{V}}_1(k+1) = \mathbf{V}_1(k) + \beta_1(k) \mathbf{y}_1(k) [\mathbf{x}(k) - \mathbf{V}_1(k) \mathbf{y}_1(k)], \quad (6)$$

where $\beta_1(k)$ denotes the adjustment coefficient. The normalization of Eq. (6) is

$$\mathbf{V}_1(k+1) = \frac{\tilde{\mathbf{V}}_1(k+1)}{\|\tilde{\mathbf{V}}_1(k+1)\|_2}. \quad (7)$$

Now, we compute the second principal component using the residual $\mathbf{e}_1(k)$ as input data:

$$\begin{cases} \mathbf{e}_1(k) = \mathbf{x}_2(k) = \mathbf{x}_1(k) - \hat{\mathbf{x}}(k) \\ \quad = \mathbf{x}_1(k) - \mathbf{V}_1(k) \mathbf{y}_1(k), \\ \mathbf{y}_2(k) = \mathbf{V}_2^T \mathbf{e}_1(k). \end{cases} \quad (8)$$

Substituting Eq. (8) to Eq. (4), we can get

$$\hat{\mathbf{x}}(k) = \sum_{p=1}^m v_{2p} x_p(k).$$

Then, recalculating Eqs. (5)–(8), we have the iteration equation for the second principal component of SWPCA:

$$\begin{aligned} \tilde{\mathbf{V}}_2(k+1) &= \mathbf{V}_2(k) + \beta_2(k) \mathbf{y}_2(k) [\mathbf{x}(k) - \mathbf{V}_2(k) \mathbf{y}_2(k)] \\ &= \mathbf{V}_2(k) + \beta_2(k) \mathbf{V}_2^T(k) \mathbf{e}_1(k) [\mathbf{x}(k) - \mathbf{V}_2(k) \mathbf{V}_2^T(k) \mathbf{e}_1(k)] \\ &= \mathbf{V}_2(k) + \beta_2(k) \mathbf{V}_2^T(k) (\mathbf{x}_1(k) - \mathbf{V}_1(k) \mathbf{y}_1(k)) \\ &\quad \cdot [\mathbf{x}(k) - \mathbf{V}_2(k) \mathbf{V}_2^T(k) (\mathbf{x}_1(k) - \mathbf{V}_1(k) \mathbf{y}_1(k))]. \end{aligned} \quad (9)$$

The normalized one is

$$\mathbf{V}_2(k+1) = \frac{\tilde{\mathbf{V}}_2(k+1)}{\|\tilde{\mathbf{V}}_2(k+1)\|_2}. \quad (10)$$

In Eq. (10), as input data, the first principal component of $\mathbf{e}_1(k)$ is used to calculate the second principal component and we have the second residual error $\mathbf{e}_2(k)$:

$$\begin{cases} \mathbf{e}_2(k) = \mathbf{x}_3(k) = \mathbf{x}_2(k) - \mathbf{V}_2(k) \mathbf{y}_2(k), \\ \mathbf{y}_3(k) = \mathbf{V}_3^T \mathbf{e}_2(k). \end{cases} \quad (11)$$

Repeating the above steps, the first principal component of $\mathbf{e}_2(k)$ as input data is used to compute the third principal component and so on and so forth, and we have the whole iteration equations of SWPCA for all principal components from Eqs. (4)–(11):

$$\begin{cases} \mathbf{V}_i(k+1) = \frac{\tilde{\mathbf{V}}_i(k+1)}{\|\tilde{\mathbf{V}}_i(k+1)\|_2}, \\ \tilde{\mathbf{V}}_i(k+1) = \mathbf{V}_i(k) + \beta_i(k) \mathbf{y}_i(k) [\mathbf{x}_i(k) - \mathbf{V}_i(k) \mathbf{y}_i(k)], \\ \mathbf{e}_i(k) = \mathbf{x}_{i+1}(k) = \mathbf{x}_i(k) - \mathbf{V}_i(k) \mathbf{y}_i(k), \\ \mathbf{y}_i(k) = \mathbf{V}_i^T(k) \mathbf{x}_i(k), \\ \mathbf{x}_1(k) = \mathbf{x}(k), \end{cases} \quad (12)$$

$$i = 1, 2, \dots, m; k = 1, 2, \dots$$

The traditional PCA applies K-L algorithm to compute the basis of subspace \mathcal{S}_i^T , in which eigenvalue λ_i and eigenvector \mathbf{V}_i of the covariance matrix need a large amount of calculations. The proposed SWPCA gets the first principal component and residual error by the gradient descent algorithm, and then uses the residual error as input data to get the second principal component and so on, avoiding heavy calculation. Furthermore, traditional PCA needs to store all images and SWPCA only stores current m images in the sliding window.

4 Foreground segmentation algorithms

In this section, we use the feature image sequence obtained by SWPCA to perform two-phase foreground segmentation.

4.1 Primary segmentation using GMM

After getting feature subspace $\mathbf{S}_l = (\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_l)^\top$ from Eq. (12), we can estimate the value of \mathbf{x} :

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{S}_l \mathbf{S}_l^\top \mathbf{x} = \bar{\mathbf{x}} + \mathbf{S}_l \varphi, \quad (13)$$

where $\varphi = \mathbf{S}_l^\top \mathbf{x}$ is the projection weight of \mathbf{x} , then $P(\hat{\mathbf{x}}) = P(\varphi)$. According to GMM from Ref. [5],

$$P(\mathbf{x}_t) = \sum_{i=1}^l \omega_{i,t} \eta(\mathbf{x}_t, \mu_{i,t}, \Sigma_{i,t}),$$

we have

$$P(\mathbf{x}) \propto \sum \exp\left\{-\frac{1}{2} \mathbf{x}^\top \mathbf{C}^{-1} \mathbf{x}\right\}. \quad (14)$$

From Eqs. (13) and (14),

$$P(\hat{\mathbf{x}}) = P(\varphi) = \sum \frac{\exp\left\{-\frac{1}{2} \sum_{k=1}^l \frac{\varphi_k^2}{\lambda_k}\right\}}{(2\pi)^{l/2} \prod_{k=1}^l \lambda^{1/2}}. \quad (15)$$

Thus, we have the background model [5]

$$B = \arg \min_l \left(\sum_{k=1}^l \varphi_k > T \right), \quad (16)$$

and the binary image at time t :

$$\mathbf{I}_t = \begin{cases} 255, & \text{foreground,} \\ 0, & \text{background.} \end{cases} \quad (17)$$

As described previously, the image \mathbf{I}_t contains each moving object both interesting and non-interesting.

4.2 Secondary foreground segmentation

To remove the rain and snow streaks, namely non-interesting targets in the video, it is necessary to perform a secondary segmentation for binary image \mathbf{I}_t . Based on the observation, the streaks move at a higher speed than other objects (e.g., the pedestrian). Furthermore, raindrops and snowflakes show strong statistics property of the direction to ground horizons. We use the von Mises distribution probability [15] density function to represent it:

$$p(\theta; \mu_{\text{von}}, d_{\text{von}}) = \frac{1}{2\pi B(d_{\text{von}})} \exp\{d_{\text{von}} \cos(\theta - \mu_{\text{von}})\}, \quad (18)$$

where $B(d_{\text{von}}) = \sum_{r=0}^{\infty} \left(\frac{1}{r!}\right)^2 \left(\frac{1}{2}d_{\text{von}}\right)^{2r}$ is the Bessel function of the first kind and order 0, θ is the direction angle of the velocity vector, μ_{von} is the velocity magnitude,

$d_{\text{von}} = 1/\sigma^2$ is the dispersion parameter, and here we use the mean velocity of all pixels in an object.

After training the samples via k -means algorithm, we get c cluster centers of the object direction. Current observation is classified into rain and snow streaks or interesting objects using the Euclidean distance. However, it cannot classify all objects completely only by the direction angle feature, it must be used to form a joint classifier along with the magnitude of the mean velocity vector.

Optical measure of rain and snow streaks is determined by camera parameters [11,16] (focal length, exposure time, etc.). Speed of a raindrop with the diameter a is [17]

$$s(a) = -0.2 + 5.0a - 0.9a^2 + 0.1a^3, \quad (19)$$

where $0.1 \text{ mm} < a < 3 \text{ mm}$; the speed of a snowflake is half that of a raindrop [7,9], about $s(a)/2$. The interesting objects for us, pedestrians, move at a lower speed. Therefore, the motion property of an object will be indirectly represented through the overlapped area of the same object between two consecutive frames, denoted as $P(O_{n,t}, O_{n,t+1})$ (see Fig. 2). Garg and Nayar [8] use the photometric constraint to reduce pixels belonging to rain through intensity changing over space and time. The intensity d_t of the pixel with rain in the t th frame is replaced by $(d_{t-1} + d_{t+1})/2$. It is based on pixels and most effective

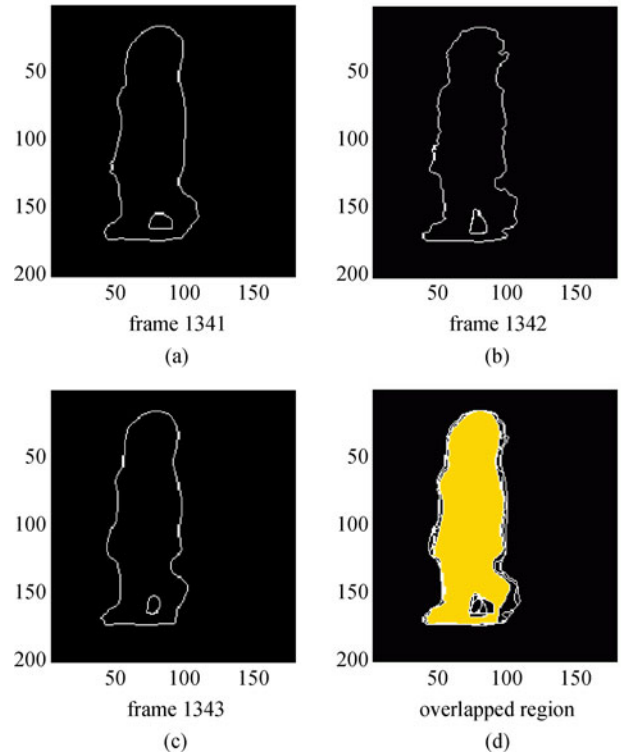


Fig. 2 Overlapped region of three consecutive frames of a child. (a) Frame 1341; (b) frame 1342; (c) frame 1343; (d) overlapped region

in a relatively textureless scenario. However, most surveillance videos have dynamic and complex background. We develop a streaks removal algorithm based on region to overcome that.

Suppose that there are M objects.

Step 1 Locate the centroid of the n th object in binary image I_t , $1 \leq n \leq M$;

Step 2 Search object pixel from the centroid until we find a pixel whose pixel value is 255 and then use it as seed to perform region growth algorithm to segment the whole object. Thus, we get this object's pixel value set;

Step 3 Calculate the mean velocity \bar{v}_n and its direction angle θ_n of an object O_n ; compute the Euclidean distance $D(\theta_n, c)$ between θ_n and each of the c cluster centers; using the nearest neighbor algorithm to classify O_n into raindrop/snowflake candidate or interesting objects;

Step 4 Calculate $P(O_{n,t}, O_{n,t+1})$: if $P(O_{n,t}, O_{n,t+1}) < T_p$ (the threshold value) then O_n is a raindrop/snowflake and can be filled by background; else, an interesting object;

Step 5 Repeat Step 1 to Step 4 M times.

5 Experimental results

Next, we show a set of experiments in which we extract interesting objects within the proposed primary-secondary background model after SWPCA de-noising and Nayar's method [9]. We start with an experiment for SWPCA and

traditional PCA, choose the number of principal components depending on scene complexity, and topically is 5–50.

As described in Fig. 3, the error of SWPCA at the beginning is higher than traditional PCA and declines sharply while iteration goes on. All the principal components must be initialized with arbitrary real numbers, and we use 1.0. Finally, SWPCA and traditional PCA show the same performance. In addition, SWPCA delay m frames to form a sliding window.

The second example is presented in Fig. 4. In this scene, some people are walking in a snowy scene. Figure 4(a) is the video and a moving area of an interesting object, the velocity field of the moving block and the corresponding orientation by the way of von Mises distribution; Fig. 4(b) shows the result of primary segmentation; Fig. 4(c) describes that snow streaks and pedestrian have different motion property; Fig. 4(d) shows the snow streaks removal effect using the proposed method; and Fig. 4(e) gives the comparison with Nayar's method described in Ref. [7]. Compared to the proposed method, the person contour is blurred showing that Nayar's method cannot classify objects with different speeds.

6 Conclusions

It is hard to build a background model in a dynamic

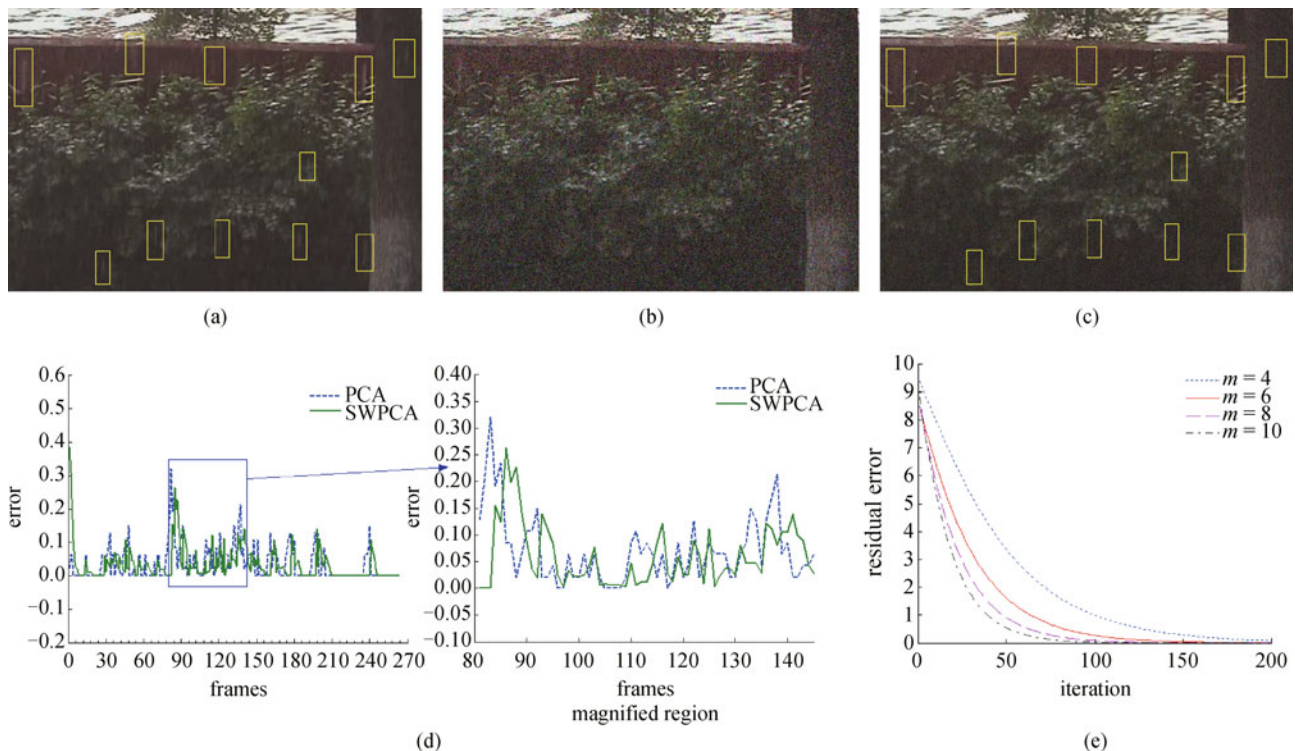


Fig. 3 Results of SWPCA and traditional PCA. (a) Video; (b) add $\sigma = 0.1$ Gaussian noise; (c) rain removal after SWPCA de-noising; (d) errors of SWPCA and PCA; (e) residual errors convergence curves. ($\sigma = 0.1$, $m = 6$, $\beta(k) \in [0.0001, 0.001]$, $T_{\max} = 0.5$, $T_{\min} = 0.1$)

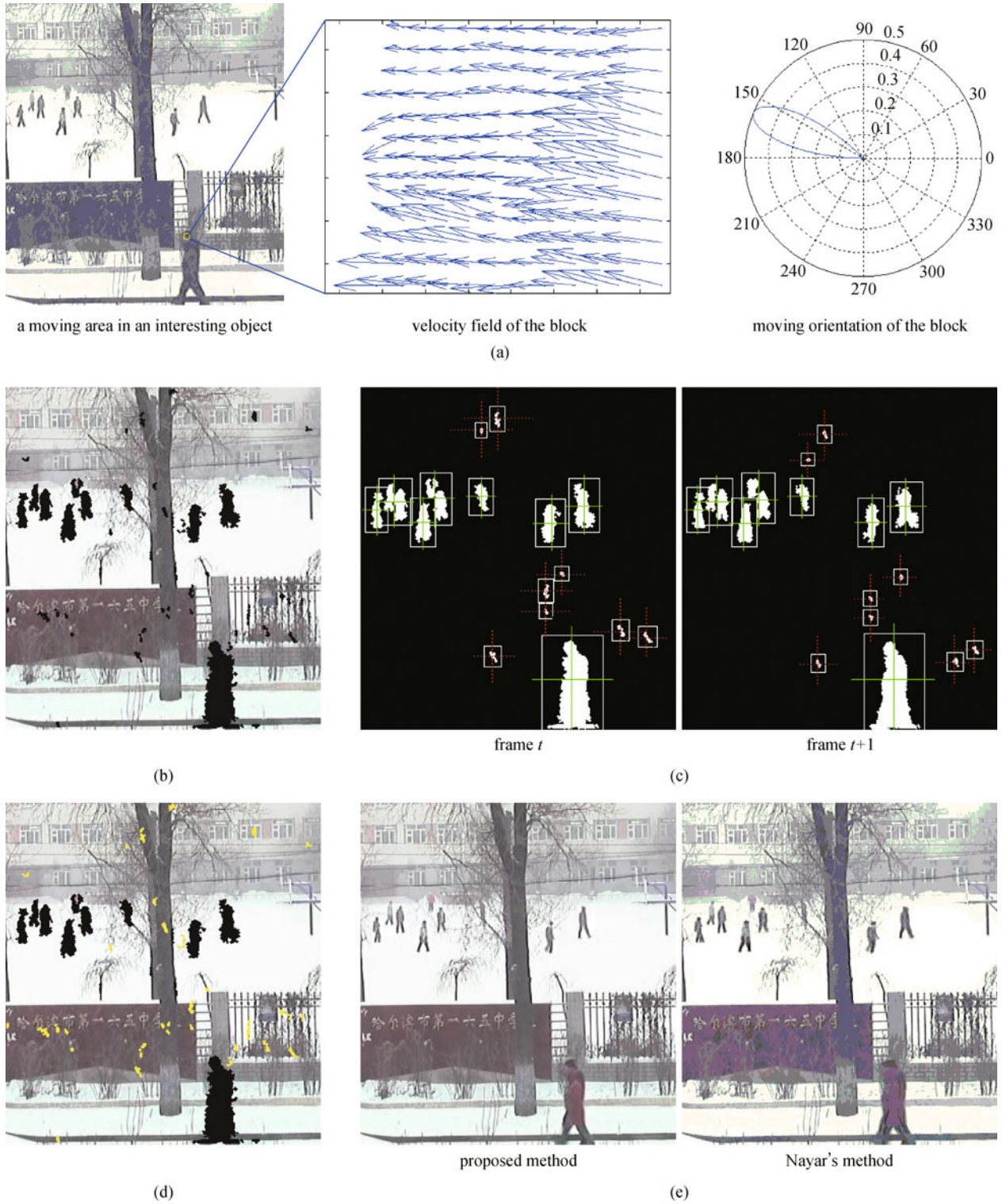


Fig. 4 Snow removal. (a) Video and orientation statistic of an image block by von Mises distribution; (b) all foreground objects by the primary segmentation; (c) different motion property between snowflakes and persons; (d) result of the secondary segmentation; (e) snow removal result

weather scene. We proposed a primary-secondary model to extract interesting objects. In the first phase, we combine SWPCA to GMM to segment all active objects in the video

with noises; in the second phase we combine von Mises distribution with the overlapped region probability algorithm to extract interesting targets, and remove the rain and

snow streaks successfully. Experiments show that the method can suppress noise and extract interesting targets effectively.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 60702032), the Natural Science Foundation of Heilongjiang Province (No. F201021), the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (No. HIT.NSRIF.2008.63).

References

1. Drew M S, Wei J, Li Z N. Illumination — Invariant image retrieval and video segmentation. *Pattern Recognition*, 1999, 32(8): 1369–1388
2. Bianco S, Cusano C. Color target localization under varying illumination conditions. *Computational Color Imaging. Lecture Notes in Computer Science*, 2011, 6626: 245–255
3. Freedman D, Turek M W. Illumination — Invariant tracking via graph cuts. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington DC: IEEE Computer Society, 2005, 2: 10–17
4. Narasimhan S G, Nayar S K. Vision and the atmosphere. *International Journal of Computer Vision*, 2002, 48(3): 233–254
5. Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking. In: *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington DC: IEEE Computer Society, 1999, 2: 2246–2252
6. Greenspan H, Goldberger J, Mayer A. Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, 26(3): 384–396
7. Garg K, Nayar S K. Detection and removal of rain from videos. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington DC: IEEE Computer Society, 2004, 1: 528–535
8. Garg K, Nayar S K. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics*, 2006, 25(3): 996–1002
9. Garg K, Nayar S K. Vision and rain. *International Journal of Computer Vision*, 2007, 75(1): 3–27
10. Barnum P C, Narasimhan S, Kanade T. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 2009, 86(2–3): 256–274
11. Barnum P, Kanade T, Narasimhan S G. Spatio-temporal frequency analysis for removing rain and snow from videos. In: *Proceedings of the First International Workshop on Photometric Analysis for Computer Vision*, in conjunction with International Conference of Computer Vision. Rio de Janeiro: INRIA, 2007, 1–8
12. Starik S, Werman M. Simulation of rain in videos. In: *Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis*. Edinburgh: IEEE Press, 2003, 95–100
13. Zhang X P, Li H, Qi Y Y, Leow W K, Ng T K. Rain removal in video by combining temporal and chromatic properties. In: *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*. Washington DC: IEEE Computer Society, 2006, 461–464
14. Faraji H, MacLean W J. CCD noise removal in digital images. *IEEE Transactions on Image Processing*, 2006, 15(9): 2676–2685
15. Khatri C G, Mardia K V. The von Mises-Fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977, 39(1): 95–106
16. Rousseau P, Jolivet V, Ghazanfarpour D. Realistic real-time rain rendering. *Computers & Graphics*, 2006, 30(4): 507–518
17. Foote G B, Du Toit P S. Terminal velocity of raindrops aloft. *Journal of Applied Meteorology*, 1969, 8(2): 249–253