

Efficient Silhouette Extraction with Dynamic Viewpoint

Yueting Zhuang

Cheng Chen

Institute of Artificial Intelligence, Zhejiang University, Hangzhou, China
{yzhuang, happyrobbie}@cs.zju.edu.cn

Abstract

A novel approach is proposed that extends the classical background subtraction method to extract silhouettes from videos in real time with dynamic viewpoint variation caused by camera movement. First, manifold learning is used to model the background under viewpoint variations. Then, for each new frame, the background image corresponding to the same viewpoint is synthesized on the fly by examining the local neighborhood on the manifold, and the silhouette is extracted via background subtraction. An extension is also presented to generate stabilized silhouettes at any fixed viewpoint within the training range. Experiments show that our approach can efficiently extract accurate silhouettes in complex situations while maintaining a low noise level.

1. Introduction

Silhouettes in images and videos encode important information about foreground objects that can be exploited to conduct high level analysis. Therefore, silhouettes are often crucial clues in image and video understanding. Silhouette extraction is the first and very important step in many vision systems such as surveillance [11], human computer interface [14], 3D motion recovery [2], and so on. The accuracy and robustness of silhouette extraction has a direct impact on performances of those systems.

Background subtraction is a widely used method to extract silhouette. In its simplest form [13], the background is expected to be static and the silhouettes are extracted by pixel wise comparisons. However, the assumption of static background brings severe limitation, as it requires that both the scene and the viewpoint be static.

Different approaches have been proposed to extent the original background subtraction. Horprasert *et al.* [7] used a color model decoupling brightness distortion and chromaticity distortion to eliminate effects of illumination changes such as shadows. Wren *et al.* [26] and Koller *et al.* [10] recursively updated the background model using some adaptive filters. The methods can only accommodate

gradual variations and often fail when the background changes rapidly. In addition, slowly moving foregrounds can also be absorbed into backgrounds. [9] and [21] used HMM to model large variations in the background, but the efficiency of HMM is low. The incorporation of Gaussian Mixture Model (GMM) is also popular, where GMM is used to model the intensity (color) at each pixel location [3] [5] [20]. However, GMM is mostly suitable for local background changes (e.g. swaying branches) and does not perform well with viewpoint changes which cause the intensity (color) of all pixels change drastically.

On the other hand, some researchers also proposed methods that inherently deal with varying viewpoints. In [17], GMM is used to filter out foreground pixels when building panoramic background representation. It assumes that the transition from foreground to background (e.g. the boundary) is reasonably sharp. Otherwise, foreground pixels might be absorbed into the background Gaussians, driving away the model. Moreover, for any pixel, its occurrence as background must be more than that as foreground. Thus, if a foreground object stays motionless for a long time, it will be incorporated into background. Wada and Matsuyama [24] used an omnidirectional background model called appearance sphere and parallax free sensing to synthesize images at different camera parameters. Wixson [25] used optical flow to detect foregrounds as salient motion. It assumes the foreground object moves in a consistent direction over time and has notable saliency against the background.

Global motion estimation [6] [19] can also be used for camera motion compensation, where several (typically 4, 6 or 8) affine parameters encoding the camera's global motion are estimated from pixel correspondence. However, because the scene is 3D, objects at different depth undergo different 2D transformations under the same camera movement. Thus, unless the scene's detailed 3D configuration is known or the scene is reasonably "flat", the accurate pixel correspondence for new frames cannot be estimated accurately.

In this paper, we propose a novel approach that extends the classical background subtraction to accommodate dynamic viewpoint variation. In the training stage, Isomap [22] is employed to find the intrinsic low dimensional

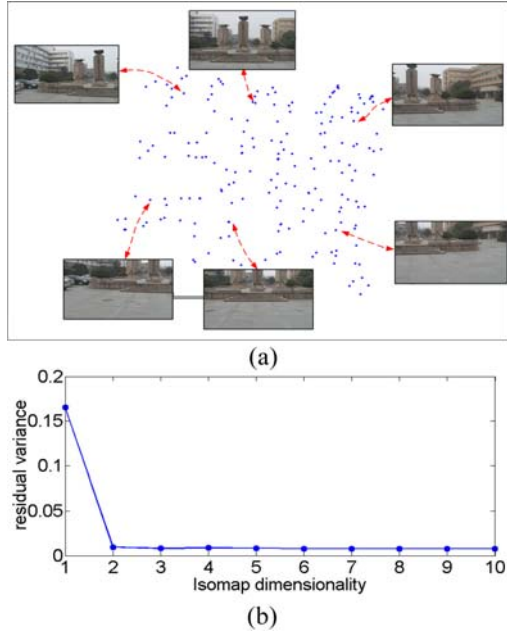


Figure 1: Isomap training. (a) The embedding of training images on a 2 dimensional manifold. (b) The dimensionality residual of dimension from 1 to 10.

manifold for the training background images at different viewpoints, and training images at nearby viewpoints are registered using the optical flows calculated rapidly and robustly. In the online stage, silhouettes are extracted independently for new frames. For each new frame, we examine its relation with regard to the background manifold using the out-of-sample extension of Isomap [1] and synthesize a background image at the new frame’s viewpoint on the fly. Finally, the silhouette is extracted by classical background subtraction. We also propose an extension that exploits the neighborhood graph computed by Isomap to synthesize stabilized silhouettes at any training viewpoint. Experiments show that our method is accurate and efficient. In addition, as the silhouette extraction is operated independently for each frame, it is free of common problems in tracking mechanisms such as error propagation and difficulty in recovering from errors.

The paper is organized as follows. Sections 2 and 3 describe the offline training and online silhouette extraction, respectively. Section 4 gives the extension of generating stabilized silhouettes. Experiments are presented in Section 5, followed by the conclusion in Section 6.

2. Offline background modeling

A set of background images $\{B_n\} = \{B_1, B_2, \dots, B_N\}$ taken on the same scene at some different viewpoints is used as training images. Many videos contain “empty” frames with no foregrounds and those frames can be used,

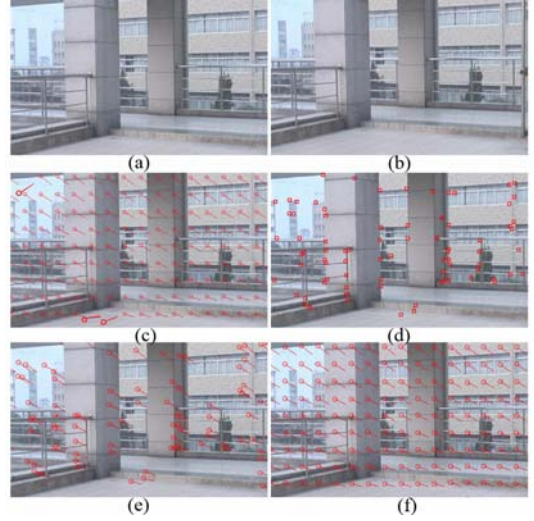


Figure 2: Fast optical calculation. (a)(b) Two background images. (c) Optical flows calculated explicitly for each pixel (uniformly down sampled for visualization). (d) Corner points. (e) The optical flows calculated at corners. (f) Interpolated optical flows (uniformly down sampled for visualization).

or we can also explicitly take some training images at different viewpoints (especially in the case of surveillance camera).

The background modeling consists of two steps: Isomap training and images registration.

2.1. Isomap training

The relations of the training images $\{B_n\}$ in the high dimensional Euclidean space are very complex and fail to convey useful cues to linear analysis (e.g. PCA). However, the seemingly complex relations are actually controlled by a subset of very few parameters, such as camera’s yaw, pitch, zoom, rolling angle, or a few others. Therefore, $\{B_n\}$ lie on a manifold whose dimensionality is significantly smaller than the original Euclidean space. Isomap [22] is a method that finds the low manifold embedding of high dimensional data. Formally, suppose G is the neighborhood graph over $\{B_n\}$ where B_i and B_j are connected if they are close under some distance metric d_X . The edge length between B_i and B_j is set to $d_X(B_i, B_j)$. Isomap first computes d_G , a matrix encoding the shortest path distances between all pairs in G . Let c_i denote B_i ’s embedding on the manifold. c_i is calculated by:

$$L = [c_1^T, c_2^T, \dots, c_N^T] = \begin{bmatrix} \sqrt{\lambda_1} \cdot e_1 \\ \sqrt{\lambda_2} \cdot e_2 \\ \dots \\ \sqrt{\lambda_d} \cdot e_d \end{bmatrix}, \quad (1)$$

where λ_i and e_i are the positive eigenvalues (in descending order) and corresponding eigenvectors of matrix $Hd_GH/2$, with the centering matrix H defined by $H_{ij} = \delta_{ij} - 1/N$. The dimension of c_i is d and is controlled by the number of reserved eigenvectors on the right side of equation (1).

We tried two different distance metrics d_X in constructing G : the $L2$ distance and the Hausdorff distance [8]. The Hausdorff distance between B_i and B_j is calculated by:

$$HD(B_i, B_j) = \max(h(\mathbf{E}_i, \mathbf{E}_j), h(\mathbf{E}_j, \mathbf{E}_i)), \quad (2)$$

where \mathbf{E}_i is the set of edge points in B_i , and h is defined by:

$$h(\mathbf{E}_i, \mathbf{E}_j) = (1/m_i) \sum_{p \in \mathbf{E}_i} \min_{q \in \mathbf{E}_j} (\|p - q\|), \quad (3)$$

where m_i is the number of points in \mathbf{E}_i and $\|p - q\|$ is the Euclidean distance between points p and q on the image.

Intuitively, Hausdorff distance which is based on the edge distance may be more sensitive in reflecting the viewpoint difference. However, we observe that the two metrics generate comparable results, and $L2$ distance is adopted because it is significantly faster to calculate.

Figure 1 shows the dimension reduction result of Isomap on a set of training images taken by a rotatable camera at 300 random viewpoints. Here the intrinsic space is two dimensional (camera's yaw and pitch).

2.2. Training images registration

Each two training images connected in Isomap's neighborhood graph have similar viewpoints, and we determine their pixel correspondence by optical flow. A lot of methods can be used to calculate optical flows [4] [16]. In this paper we use a simple algorithm that compares the cross correlation between image patches. Formally, to find the pixel location on image I_2 that corresponds to a location (x_0, y_0) on image I_1 , we define a square window and search the neighborhood of (x_0, y_0) on I_2 for a pixel (x, y) that maximizes the cross correlation r between the window centered at (x_0, y_0) on I_1 and the windows centered at (x, y) on I_2 . r is defined by:

$$r = \frac{\sum_{i,j} (W_1(i,j) - \bar{W}_1)(W_2(i,j) - \bar{W}_2)}{\sqrt{\sum_{i,j} (W_1(i,j) - \bar{W}_1)^2 \cdot \sum_{i,j} (W_2(i,j) - \bar{W}_2)^2}}, \quad (4)$$

where W_1 and W_2 are the windows on I_1 and I_2 , and \bar{W}_k is the mean intensity (color) in W_k .

Accurate optical flow calculation is notorious for its low efficiency [15]. We use the optical flow interpolation to improve both the efficiency and the accuracy. As in Figure 2, if we calculate optical flows explicitly at each pixel, the results are highly redundant, and some outliers emerge at pixels whose neighborhoods take on a uniform color (e.g.

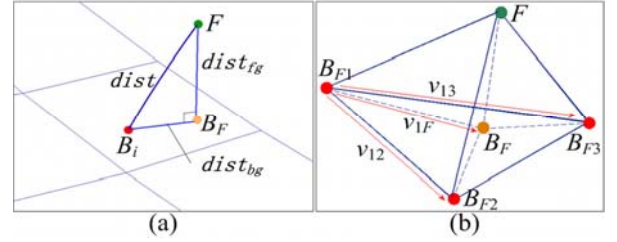


Figure 3: (a) The manifold's local neighborhood near F . (b) The calculation of optical flow from B_{F1} to B_F .

the gray ground in the bottom and the white building in the upper left in Figure 2(c)). It's well known that optical flow is most robust at corner points due to the aperture problem [23], and so we can explicitly calculate optical flows only at corners and this usually eliminates outliers as well as accelerates the computation. On the other hand, since the flows are highly redundant, it is safe to interpolate flows at non-corners from those at corners (in this paper knn interpolation is used). Regular outlier removal procedures (such as EM in our experiments) can be applied before interpolation to ensure that no outliers are present. In this way, optical flows are derived accurately and rapidly. In addition, note that our estimation of pixel correspondence is non-parametric and so does not suffer from the problem of global motion estimation as stated in Section 1.

3. Online silhouette extraction

Now stream of new frames arrives for silhouette extraction. We independently extract silhouette for each frame. Suppose the current frame image is F . First, the background image B_F at exactly the same viewpoint as F is synthesized on the fly. To synthesize B_F , two steps are taken: determining B_F 's low dimensional embedding on the manifold, and constructing its image. These two steps are described in Sections 3.2 and 3.3, preceded by Section 3.1 which describes the out-of-sample extension to Isomap.

3.1. Out-of-sample extension to Isomap

To synthesize B_F , we have to determine the relation of the new frame F to the learned background manifold. The original Isomap [22] does not provide the mapping of new data, so the out-of-sample extension [1] and incremental extension [12] are of interest here. The out-of-sample version only determines new data's embedding, while the incremental version updates the manifold according to the new data as well. The former is preferred in this paper as it is faster and we don't want the new frames containing foregrounds to interfere in the background manifold. It is shown in [1] that using the proposed kernel, the out-of-sample extension is equivalent to another extension called Landmark Isomap [18] by considering the training

data as landmark points and the new data as non-landmark points. Formally, let d_G be the matrix of shortest path distances over a set of m landmark points. First, similar to the original Isomap, the embeddings of landmark points are given by equation (1). Then, for a new point x , suppose d_x denotes the distances between x and each landmark point, then the embedding of x is given by:

$$c_x = L^* (\overline{d_G} - d_x) / 2, \quad (5)$$

where L is defined in equation (1), L^* is L 's psudoinverse transpose, and $\overline{d_G}$ is d_G 's row mean.

3.2. Determining B_F 's embedding

For clear visualization, in Sections 3.2 and 3.3 we assume the $\{B_n\}$ ' intrinsic dimension $d=2$. That is, $\{B_n\}$ lie on a 2D manifold. The scenarios where $d>2$ can be easily extended (at the end of Section 3.3).

Because B_F is a pure background image, it should lie somewhere on the same 2D manifold formed by $\{B_n\}$. To determine B_F 's embedding, the first idea might be treating F as an out-of-sample data and calculating its position on the 2D manifold using equation (5). However, this strategy is not desirable. F is not a pure background image and there is not a position on the background manifold for F . In other words, the 2D manifold is not sufficient to express F , and to remedy it, the dimension has to be increased. As illustrated in Figure 3(a), we increase the manifold's dimension by one (i.e. calculate the 3D rather than 2D embeddings for $\{B_n\}$ in equation (1)). For $\{B_n\}$, the 3D coordinates are redundant and they still lie on a 2D sub-manifold within the 3D manifold. As to F , its 3D embedding can now be calculated by equation (5) and it lies somewhere outside the 2D sub-manifold. In a word, by incrementing the manifold's dimension, we make room for F without destroying the intrinsic structure within $\{B_n\}$.

B_F 's embedding on the 2D sub-manifold is then approximated by calculating the perpendicular projection of F 's 3D coordinate onto the 2D sub-manifold (Figure 3(a)).¹ We argue that this is reasonable. The two dimensions of the 2D sub-manifold encode the two DOFs of viewpoint. F differs from $\{B_n\}$ in ways that it has not only the DOFs of viewpoint but also the presence of foreground. In addition, the foreground DOF is orthogonal to the DOFs of viewpoint because they do not have cross effects on each other. Formally, assume B_F 's embedding on the 2D sub-manifold is known and we examine the relations among F , B_F and a training image B_i (See Figure 3(a)). Let $dist$ denote the distance between F and B_i . $dist$

consists of $dist_{bg}$, which is the distance in the background region R_{bg} , and $dist_{fg}$, which is the distance in the foreground region R_{fg} . Moreover, $dist_{bg}$ and $dist_{fg}$ are mathematically orthogonal:

$$\begin{aligned} dist &= \sqrt{\sum_{(x,y)} (F(x,y) - B_i(x,y))^2} \\ &= \sqrt{\sum_{(x,y) \in R_{bg}} (F(x,y) - B_i(x,y))^2 + \sum_{(x,y) \in R_{fg}} (F(x,y) - B_i(x,y))^2} \\ &= \sqrt{dist_{bg}^2 + dist_{fg}^2} \end{aligned} \quad (6)$$

Furthermore, $dist_{fg}$ can be approximated by the difference between F and B_F since they only differ in R_{fg} . $dist_{bg}$ can be approximated by the difference between B_F and B_i because they encode the differences in background caused by viewpoint variation and R_{bg} usually occupies a large proportion of the image. Thus, the difference between F and a training image B_i is decoupled orthogonally into the viewpoint distance and the distance caused by foreground.

3.3. Synthesizing B_F 's image

The next step is constructing the image of B_F given its 2D embedding. Direct interpolation using similar images in $\{B_n\}$ with some weights is not valid, because the linear interpolation of images at different viewpoints is an image with ghosts of each component image. We synthesize B_F 's image via pixel correspondence. If the optical flows from any nearby training image B_i to B_F are known, then B_F 's image can be synthesized.

Suppose three training images that are closest to B_F on the manifold are B_{F1} , B_{F2} and B_{F3} . As shown in Figure 3(b), let v_{12} , v_{13} and v_{1F} denote the vectors from B_{F1} to B_{F2} , from B_{F1} to B_{F3} , and from B_{F1} to B_F on the 2D manifold's local neighborhood, and let O_{12} , O_{13} and O_{1F} denote the optical flows from B_{F1} to B_{F2} , from B_{F1} to B_{F3} , from B_{F1} to B_F , respectively. v_{1F} can be written as a combination of v_{12} and v_{13} : $v_{1F} = av_{12} + bv_{13}$, with weights a and b . Then, the unknown optical flow O_{1F} can be interpolated using the same weights: $O_{1F} = aO_{12} + bO_{13}$, where O_{12} and O_{13} are pre calculated during training. Because B_{F1} , B_{F2} , B_{F3} and B_F are close on the manifold, this interpolation is safe. For better accuracy, more than three close training images can be used, where the determination of O_{1F} becomes an optimization.

After O_{1F} is determined, the image B_F can be constructed. Specifically, B_F is initialized to be empty and each pixel on B_{F1} is shifted to a target location on B_F according to the value of O_{1F} at that source location. If a target location on B_F has multiple sources then its color is the average value. If a target location on B_F has no source pixel, then its color is interpolated locally on B_F .

As soon as the image B_F is constructed, silhouette for frame F can be extracted by classical background subtraction using F and B_F . Before background subtraction, F and B_F are convolved by a small Gaussian kernel to

¹ It's a fundamental property that a manifold is a topological space that is locally Euclidean (i.e., around every point, there is a local neighborhood that can be "Euclideanly" approximated). The local neighborhood near B_F can be treated as Euclidean and F is projected onto this neighborhood



Figure 4: Results of the rotatable surveillance camera experiment. Top row: original video frames. Middle row: constructed backgrounds. Bottom row: extracted silhouettes.

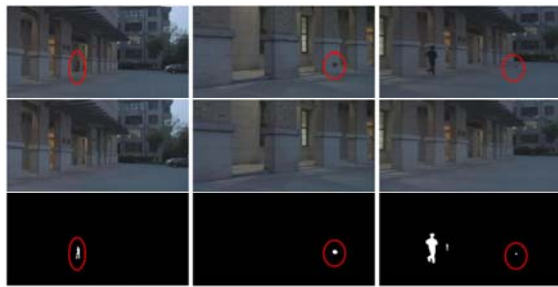


Figure 5: Results of rotatable camera in poor illumination. Top row: original video frames. Middle row: constructed backgrounds. Bottom row: extracted silhouettes.

compensate for noises and subtle deviation introduced in the preceding processes.

The above discussion assumes that the training images' intrinsic dimension $d=2$. The scenarios where $d>2$ can be readily extended, where B_F 's d dimensional embedding is determined by incrementing manifold's dimension by one and projecting F 's $d+1$ dimensional embedding onto the d dimensional sub-manifold. The determination of O_{1F} is also similar, with at least $d+1$ neighbors on manifold used.

4. Generating stabilized silhouettes

The silhouette sequence produced in the last section has the same viewpoint variation as the original video. In some cases, it is more desirable to generate silhouettes with viewpoints stabilized either for better visualization or for easier analysis. Suppose we'd like the silhouettes to be fixed at the same viewpoint as a training image B_v (or, any location on the manifold). For each new frame F , as soon as we find the low dimensional embedding of B_F , we get the shortest path from B_F to B_v on the manifold, using the neighborhood graph computed by Isomap. Suppose the shortest path is $\langle B_F, B_1, B_2, \dots, B_v \rangle$. Since the optical flows of each consecutive pair $\langle B_F, B_1 \rangle, \langle B_1, B_2 \rangle, \dots$ have been computed during training, the optical flows O_{Fv} from B_F to B_v can be quickly generated. Then, silhouettes at the same



Figure 6: Results of the camera shaking experiment. Top row: original video frames. Bottom row: extracted silhouettes.

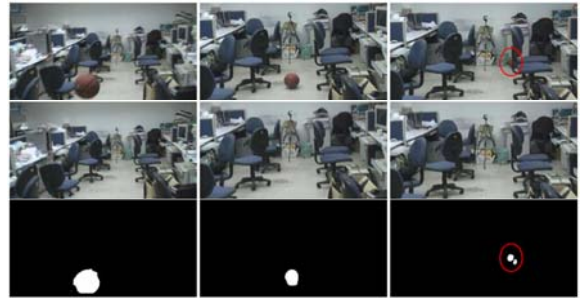


Figure 7: Results of the shaking camera with zoom change. Top row: original video frames. Middle row: constructed backgrounds. Bottom row: extracted silhouettes.



Figure 8: Results of stabilized silhouette generation. Top row: original video frames. Middle row: silhouettes before stabilizing. Bottom row: stabilized silhouettes.

viewpoint as B_v can be constructed from the unstabilized silhouettes extracted as in Section 3 and O_{Fv} .

5. Experimental results

5.1. Results of real videos

We conduct some experiments to test our approach in different scenarios, including rotating camera, shaking camera with or without zoom changes, and the case of generating stabilized silhouettes.

Figure 4 shows an experiment on a surveillance camera which scans a large lot by rotating its viewpoint both vertically and horizontally. Training background images

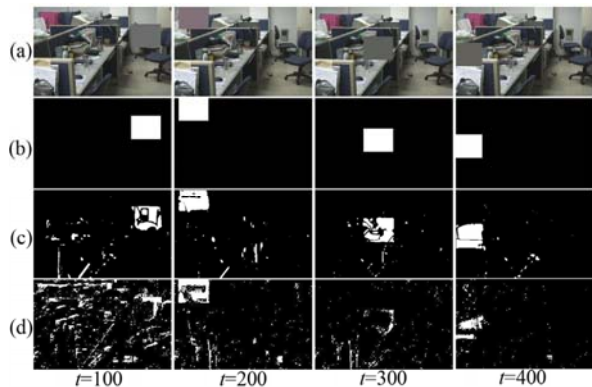


Figure 9: Experiment of our method compared to GMM based method in gentle camera shaking scenario. (a) The video frame with foregrounds. (b) The ground-truth silhouette. (c) Result of our method. (d) Result of GMM method with component number k adaptively selected.

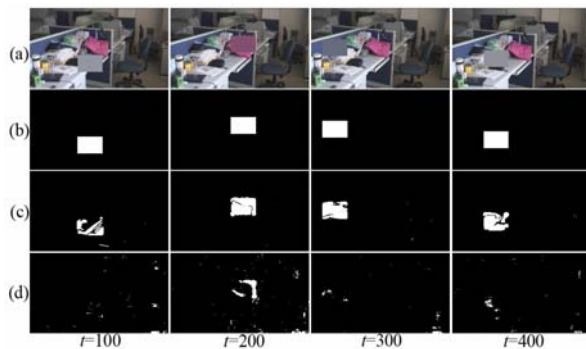


Figure 10: Experiment of our method compared to GMM based method in severe camera shaking scenario. (a) The video frame with foregrounds. (b) The ground-truth silhouette. (c) Result of our method. (d) Result of GMM method with component number k adaptively selected.

are acquired by taking 300 pictures at random viewpoints (these training images are used in Figure 1). In this case the background manifold's dimension is two, corresponding to the camera's pan and tilt angle. In the online stage, a person walks into the lot, leaves a bag on the ground and walks out, while in the distance there is another person wandering back and forth. The online clip has 500 frames and Figure 4 shows several results. For each frame, its original video frame image, the constructed background image and the extracted silhouette are presented. Our approach successfully extracts the two persons and the bag, while maintaining a very low noise rate. At several frames around $t=236$ the viewpoints are rotated to an angle far from any training images so that the extracted silhouettes are incorrect (see the accompanying video), but the error is immediately recovered as soon as the viewpoint returns to

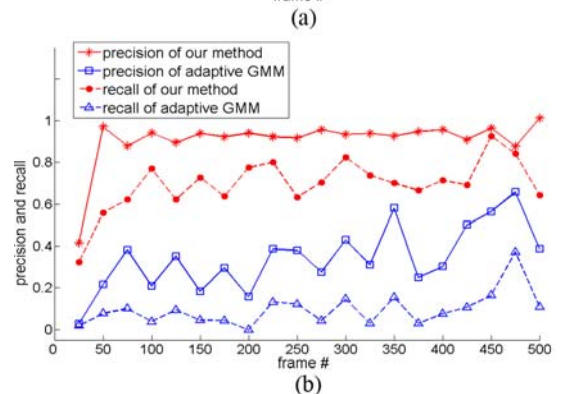
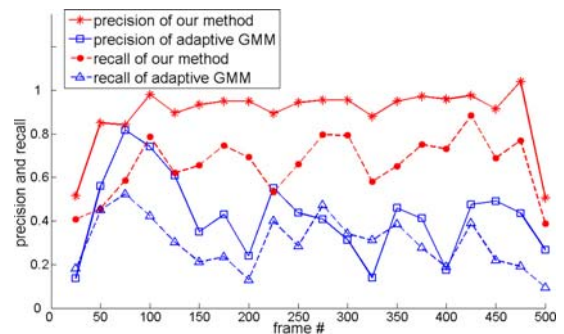


Figure 11: Quantitative comparison. (a): the gentle shaking scenario. (b): the severe shaking scenario. The data points are smoothing averages of each segment.

the training range.

Figure 5 is also an experiment of rotating camera. The video is taken in the dusk with poor illumination.

Figure 6 shows an experiment on shaking camera held by hands. The first 120 frames in the clip containing no foregrounds are used for training and the remaining segment for online silhouette extraction has 610 frames. Note that camera shaking is not equivalent to small-range rotation which has only pan and tilt. In the camera shaking case, the hand cannot keep the camera's rolling angle stable and the camera's motion is the composite of pan, tilt and roll. Therefore, the background manifold's dimension is 3.

Figure 7 is an experiment on shaking camera with zoom change. Training background images are acquired by taking 576 pictures at different viewpoints (including 9 different zoom levels). A basketball is thrown into the scene in the online stage. In this case, the camera's motion is the composite of pan, tilt, roll and zoom. Therefore, the background manifold's dimension is 4.

Figure 8 shows the experiment of generating stabilized silhouettes. 120 frames are used for training and the online clip contains 718 frames. The camera is shaking by hand and the dimension of background manifold is 3. Figure 8 shows the original video frame along with the silhouettes both before and after stabilizing for three frames. Please

refer to the accompanying video for a more clear perception of silhouettes stabilizing.

5.2. Quantitative comparison

We test our method against GMM based methods [3] [5] [20]. Specifically, we take two video clips, each containing 700 frames on a scene without foregrounds, when the camera is shaking. In one clip the shaking is gentle and in the other one the shaking is severer. For each clip, 150 frames are randomly selected for training and the remaining 550 frames are used for testing. To conveniently get the ground-truth silhouettes for quantitative evaluation, foregrounds are synthesized by programmatically adding a flying box to the testing frames. To “confuse” the algorithms, the color of the foreground box at any frame is set to the mean color of the area masked by the box on that frame. Note that this is a very difficult situation, as the foreground box behaves like a chameleon that dynamically resembles its local neighborhood.

We test GMM methods and our method on these two clips. For GMM methods, Gaussian component numbers both fixed (from 1 to 6) and adaptively selected by minimum description length (MDL) are tested. Figures 9 and Figure 10 show some image results in the gentle and severe shaking scenario.

Two metrics are used to quantitatively evaluate the performance: *precision* and *recall*:

$$precision = \frac{\sum_{x,y} S(x,y) \cdot S_g(x,y)}{\sum_{x,y} S(x,y)}, \quad (7)$$

$$recall = \frac{\sum_{x,y} S(x,y) \cdot S_g(x,y)}{\sum_{x,y} S_g(x,y)}. \quad (8)$$

where S is the extracted silhouette and S_g is the ground-truth silhouette.

	gentle shaking		severe shaking	
	mean <i>precision</i> (%)	mean <i>recall</i> (%)	mean <i>precision</i> (%)	mean <i>recall</i> (%)
our method	93.22	68.73	92.49	70.03
GMM, $k=1$	51.44	21.68	49.57	7.38
GMM, $k=2$	52.20	22.40	65.43	7.21
GMM, $k=3$	52.45	25.98	47.63	8.30
GMM, $k=4$	48.08	29.31	39.19	10.25
GMM, $k=5$	41.38	35.05	37.45	13.24
GMM, $k=6$	37.83	37.66	36.18	16.20
GMM, k =adaptive	44.18	32.48	34.48	13.57

Table 1: Detailed comparison. Note that the adaptive- k GMM tends to get a balance between *precision* and *recall* compared to fixed- k GMMs.

Figure 11 and Table 1 show the graphical and numerical comparison. Clearly, our approach notably outperforms the GMM based methods. In addition, as the shaking gets heavier, the performance of GMM methods degrades rapidly, while our method does not show deterioration.

5.3. System efficiency

We develop a system implementing the presented approach. The codes are not optimized and the test bed is a PC with 3.2 GHz CPU and 2GB RAM. The training stage typically takes from several minutes to over thirty minutes, depending on the amount of training images. As to the online extraction speed, for videos of pixel dimension 320*240, the system can achieve 11~12 fps with 120 training images and 5~6 fps with 300 training images. Most of the running time is occupied by the online calculation of d_x , i.e. the distances between the new frame and each training image. As long as d_x is known, the embedding of the new frame can be immediately determined by equation (5) and the optical flows can be quickly interpolated for background synthesis. Note that the calculation of d_x can be easily parallelized for higher efficiency.

6. Conclusion and future work

We have presented a new approach that extends classical background subtraction to accommodate dynamic viewpoint. The Low dimensional manifold of the background under varying viewpoints is exploited to synthesize background images at the same viewpoints as new frames. We also present an extension to generate stabilized silhouettes. Our method gets rid of problems of error accumulation and difficulty in recovering from errors. Experiments show that our method is accurate and efficient.

The efficiency of our approach comes from several innovations. Pixel correspondence is only explicitly calculated in training and in the online stage they are interpolated rapidly on the manifold. New frames’ low dimensional embeddings are determined by out-of-sample extension of Isomap, which is significantly faster than the original Isomap. Besides, we use a novel method to calculate optical flows which improves both efficiency and accuracy during training.

In Section 2.2, the optical flows are calculated by a simple algorithm that compares image patches. Because the explicit optical flow calculation only occurs in the training, any of the more accurate (yet potentially slower) optical flow calculation algorithms can be employed without affecting the efficiency of online silhouette extraction.

Silhouette extraction in complex situations with cluttered background and complicated camera movements is a challenging work, and our method still has some limitations. Our method requires a set of appropriate

training images, which might be difficult to acquire in some situations. Besides, the background modeling will be interfered if the scene itself is changing simultaneously with dynamic viewpoint (e.g. moving clouds or swaying trees with dynamic viewpoint).

Some further improvement can enable the proposed method to tackle more complex situations. For example, our method does not exploit the statistical properties of the pixels, and statistical approaches (such as GMM) can be incorporated. Those statistical methods model the color value of corresponding pixels over frames. With dynamic viewpoints, the corresponding pixels are no longer at the same image location, but in our approach the pixel correspondences are kept track of during training and so statistical modeling is still viable. Therefore, our method has the potential to deal with simultaneously dynamic scene (e.g. swaying branches) with dynamic viewpoint. On the other hand, some further steps could be taken to accommodate gradual background changes. The key idea is that for each new frame, after the silhouette is extracted, the background region can serve as new samples containing the latest background information and the manifold can be updated accordingly. Another interesting future work is to study how to synthesize the training images when no pure background images are directly available from the video or the available background images are not sufficient to do manifold learning.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.60525108, No.60533090), Program for Changjiang Scholars and Innovative Research Team in University (IRT0652), 973 Program (No.2002CB312101), Science and Technology Project of Zhejiang Province (2005C13032, 2006C13097).

References

- [1] Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *NIPS*, 2004.
- [2] C. Chen, Y. Zhuang, and J. Xiao. Towards robust 3D reconstruction of human motion from monocular video. *Lecture Notes on Computer Science*, vol. 4282, pp. 594-603, 2006.
- [3] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *PAMI*, 24(3):381-396, 2002.
- [4] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *BMVC*, vol. 1, pp. 195-204, 1998.
- [5] W. E. L. Grimson, C. Stauffer, and R. Romano. Using adaptive tracking to classify and monitor activities in a site. In *CVPR*, pp. 22-29, 1998.
- [6] J. Heuer and A. Kaup. Global motion estimation algorithm for video segmentation. In *ACM Multimedia*, pp. 261-264, 1999.
- [7] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV Frame-rate Workshop*, 1999.
- [8] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *PAMI*, 15(9):850-863, 1993.
- [9] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake. An HMM based segmentation method for traffic monitoring movies. *PAMI*, 24 (9):1291-1296, 2002.
- [10] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Toward robust automatic traffic scene analysis in real-time. In *ICPR*, pp. 126-131, 1994.
- [11] Y. Kuno, T. Watanabe, Y. Shimosakoda, and S. Nakagawa. Automated detection of human for visual surveillance system. In *ICPR*, pp. 865-869, 1996.
- [12] M. H. C. Law and A. K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *PAMI*, 28(3):377-391, 2006.
- [13] M. K. Leung and Y. H. Yang. Human body motion segmentation in a complex scene. *Pattern Recognition*, 20(1):55-64, 1987.
- [14] H. Li and M. Greenspan. Multi-scale gesture recognition from time-varying contours. In *ICCV*, pp.236-243, 2005.
- [15] H. Liu, T. Hong, M. Herman, T. Camus, and R. Chellappa. Accuracy vs. efficiency trade-offs in optical flow algorithms. *CVIU*, 72(3):271-286, 1998.
- [16] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: a synopsis of current problems and methods. *IJCV*, 19(1):29-55, 1996.
- [17] A. Mittal and D. Huttenlocher. Scene modeling for wide area surveillance and image synthesis. In *CVPR*, vol. 2, pp.160-167, 2000.
- [18] V. de Silva and J.B. Tenenbaum. Global versus local approaches to nonlinear dimensionality reduction. In *NIPS*, vol. 15, pp. 721-728, 2003.
- [19] A. Smolic, M. Hoeynck, and J.-R. Ohm. Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications. In *ICIP*, vol. 2, pp. 271-274, 2000.
- [20] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, vol. 2, pp.246-252, 1999.
- [21] B. Stenger, V. Ramesh, N. Paragios, F. Coetsee, and J.M. Buhmann. Topology free hidden markov models: application to background modeling. In *ICCV*, pp. 294-301, 2001.
- [22] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323, 2000.
- [23] A. Verri and T. Poggio. Against quantitative optical flow. In *ICCV*, pp. 171-180, 1987.
- [24] T. Wada and T. Matsuyama. Appearance sphere: Background model for pan-tilt-zoom camera. In *ICPR*, pp. 718-722, 1996.
- [25] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *PAMI*, 22(8):774-780, 2000.
- [26] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfunder: Real-time tracking of human body. *PAMI*, 19(7):780-785, 1997.