Video Background Segmentation Using Adaptive Background Models

Xiaoyu Wu, Yangsheng Wang, and Jituo Li

Institute of Automation, Chinese Academy of Sciences
{xiaoyu.wu}@ia.ac.cn

Abstract. This paper proposes an adaptive background model which combines the advantages of both Eigenbackground and pixel-based gaussian models. This method exploits the illumination changes by Eigenbackground. Moreover, it can detect the chroma changes and remove shadow pixels using gaussian models. An adaptively strategy is used to integrate two models. A binary graph cut is used to implement the foreground/background segmentation by developing our data term and smooth term. We validate our method on indoor videos and test it on the benchmark video. Experiments demonstrate our method's efficiency.

Keywords: Background modeling, Background segmentation, Adaptive graph cut.

1 Introduction

Background segmentation seeks to automatically extract the foreground object from the background. As a fundamental part of high-level task such as object tracking and recognition, the technology of background segmentation is widely used in the multimedia, video surveillance, virtual reality, etc.

In this paper, we focus our research on monocular video background segmentation. In contrast to stereo depth information, monocular video background segmentation is much less constraints. Therefore, additional assumptions are often made to limit the solution space. The proposed assumptions are different dependent on the applications. In our case, the background scene is static and known, where the background modeling plays an important role in segmenting the foreground object.

Most background modeling methods are pixel-based which make use of statistical characteristic to model each pixel independently. Many representative pixel-level methods have been presented including unimodal Gaussian model in PFinder system [5], GMMs [1], W4 system, [2], non-parameter background modeling [3], mean-shift based modeling [4], etc. These methods are based on pixel-wise density estimation, which estimates the probability of an intensity or color value for each pixel. The density-based models have the advantage of extracting detailed shapes of moving objects and updating the models online for each pixel. However, their drawbacks are that the segmentation results are sensitive to noise, illumination change, and small movement of the background.

On the other hand, eigenbackground modeling method [6] is also based on appropriate statistics of gray values over time at each pixel location. But its estimation technique is based on the principal component analysis that is different from pixel-wise probability density estimation. In Eigenbackground framework, the whole 2D images as vectors are collected to obtain principal components during the training phase. In the test phase, a new vectorized image is first projected onto trained eigenvector subspace and is reconstructed using the projected image and the principal components. The pixels will be considered as foreground points when their Euclidean distance difference between the input image and the the reconstructed image are greater than a given threshold. This eigenspace model can well describe the range of intensity that have been observed (e.g., lighting variations over the trainging.), which will be explained in section 2.1. The advantage of this method lies in the simplicity and fastness comparing with pixel-wise density-based methods. However, it is not straightforward to integrate multi-channel data such as RGB color images without significant increase of dimensionality. Moreover, it is difficult to update eigen-background online during the detection procedure because the computational cost of recalculating eigenvectors is huge.

As mentioned above, the modeling method based on eigenspace analysis and the density-based methods have their pros and cons. Therefore, the combination of two types of methods should be considered to be complementary with each other. In this paper, we adopt the eigenspace model to build the intensity information for each pixel due to its satisfied description during the light variation. Unimodal gaussian density methods with less computational cost are used to describe color information for each pixel. The reason of using unimodal gaussian models is that the color information change little in most cases even if light varies much. In addition, it is easy to update unimodal gaussian modal online. An adaptive method is proposed to combine the unimodal gaussian model with the eigenspace model. The multi-channel data is used with less computation, and the mixture model can be updated partially .

The rest of the paper is organized as follows: Section 2 proposes how to build background model fusing eigenbackground with single gaussian models. Section 3 describes the segmentation method based on graph cut algorithm. Section 4 demonstrates experimental results. Conclusion is drawn in section 5.

2 Mixture Background Model

2.1 Eigenbackground

We build an eigenspace background model using the intensity information in a static background. The eigenspace model is formed by taking a set of N sample images, $\{I_t\}_{t=1,2,\dots,N}$. From these images the mean background image μ_b is subtracted to get zero mean vector $\{X_i\}_{i=1,2,\dots,N}$ where $X_t = I_t - \mu_b$. Then the covariance matrix C_b is computed by X_t and is diagonalized as: $L_b = \Phi_b C_b \Phi_b^T$, where L_b is the diagonal matrix with the corresponding eigenvalues. Φ_b is the

eigenvector matrix that is calculated through the singular value decomposition(SVD). In order to reduce the dimensionality of the eigenvector space, only M largest eigenvalues are kept, resulting in Φ_{Mb} .

Once a new image, I_t , is available, it is first projected to the lower dimensional space expanded by the eigenbackgrounds images. In this reduction, only the static parts of the scene will be accurately described as a sum of the various eigenbackgrounds. The projected image, $I_t^B = \Phi_{Mb}(I_t - \mu_b)$, is now reconstructed as the background model for the new image. Finally we can determine the foreground points whose differences between the input image and the background image are greater than a given threshold θ :

$$D_t = |I_t - I_t^B| > \theta \quad (1)$$

The reason why this method works is that only the static parts of the scene will be accurately described as a sum of the various eigenbasis vectors. Moving objects will not appear in the same location of the sample images, so they contribute much less to the model than the static parts of the scene.

2.2 Single Gaussian Model

Similar to Pfunder[5], we learn the scene for N frames without the foreground object to build the density probability model for each pixel. Background representation is to use the normalized (r,g) space, where $r = R/(R + G + B)$ and $g = G/(R + G + B)$. The (r, g) color space we choose reflects the color information and is invariant to changes in brightness. Here, every channel of the color space is treated as independent to reduce the computation. By learning the scenes, we can obtain the parameter of model composed of the mean μ_i and covariance matrix σ_i for certain pixel i in the (r,g) color space.

If a new pixel value I_t belongs to the gaussian distribution at this point, the parameters are updated using the infinite impulse response filter as follows:

$$\begin{aligned} \mu_{i,t} &= (1 - \alpha)\mu_{i,t-1} + \alpha I_{i,t} \\ \sigma_{i,t}^2 &= (1 - \alpha)\sigma_{i,t-1}^2 + \alpha(I_{i,t} - \mu_{i,t})^T(I_{i,t} - \mu_{i,t}) \end{aligned} \quad (2)$$

Here, α is the learning rate, which determines the speed of the distribution's parameters change. Note that I_t denotes (r_t^i, g_t^i) different from the intensity value in Equation 1.

If the problem of shadow removal is considered, (r, g) color space is helpful to detect shadow-pixels who are wrongly classified as object-pixels by Eigenbackground. As the literature [7] pointed out, the change in intensity by a shadow is expected to stay within a certain range. In other words, the ratio between the intensity value and the expectation value varies at some range. Under this circumstance, the (r, g) color value change little. So we can use this characteristic to distinguish the shadow pixels.

2.3 Combination of Two Background Models

Although Eigenbackground is more robust to the light variant, it doesn't make use of the color information and online-update is difficult. On the other hand,

unimodal Gaussian model develop the color (r,g) space model without the intensity information because density-based unimodal Gaussian is sensitive to the light variant. Therefore, we combine two models to compensate for each other in the following way:

$$\begin{cases} D_i(B) = \lambda D_i(E) + (1 - \lambda) D_i(G) \\ D_i(E) = |I_i - B_i| \\ D_i(G) = \left(\frac{(I_{r,i} - \mu_r)^2}{\sigma_r^2} + \frac{(I_{g,i} - \mu_g)^2}{\sigma_g^2} \right) \\ \lambda = 1 - \exp(-(I_i - \mu_i)^2 / \eta^2) \end{cases} \quad (3)$$

Where $D_i(E)$ is the calculated background cost based on Eigenbackground method for the pixel i in Equation 1. $D_i(G)$ denotes the background cost based on Gaussian model in section 2.2. $D_i(B)$ is the combining cost of the pixel belonging to the background. I_i , $I_{r,i}$ and $I_{g,i}$ correspond to the illumination, normalized r and g value, respectively. λ is mixing factor that reflects the confidence in eigenspace model. η models the noise and is set as a priori. To maximize robustness, λ should be adaptively adjusted: if the intensity varies much, we should rely more on the eigenbackground model so that the whole system is robust to illumination changes of background; otherwise, we should rely more on per-pixel gaussian color models to detect R, G and B multi-channel color changes. Note that when the intensity I_i value is small and the input image is dark, it is better to only use $D_i(E)$ because the $(R + G + B)$ value is small and the $D_i(G)$ is not precise under this circumstance.

Figure 1 shows the experimental results using eigenbackground, single gaussian and mixture model, respectively. The first column is the original test images. The second shows the results obtained by eigenbackground. The third column corresponds to the results using the single gaussian model. The last column gives the results by mixture background model. From Figure 1, we can find that the mixture models have the advantages of better segmentation results over the eigenbackground, and of less noise over the single gaussian model.

Fig. 1. The respective segmentation results using three methods

3 Background Segmentation Based on Graph Cut

Although we can obtain the segmentation results by setting the threshold to subtract the background in section 2, the results are not reliable because the threshold is very sensitive to noise and background illumination. Therefore, we hope to find an adaptively optimized method to determine the foreground object. The minimum graph cut algorithm based on a combinatorial optimization is an useful tool to image segmentation [8]. In graph cut framework, the observed image are divided into three labels by background model: the pixels with high background probability in the Equation 3 are the background(B). The pixels with low probability are the foreground(F). The rest are attached to the unknown label(U), which will be determined the background or foreground label by minimizing certain energy function using the min-cut algorithm. In the following, we will explain in detail how to label unclassified pixels by graph cut methods.

3.1 Graph Cut

Supposing that the input image of each frame is considered as a graph. All the image pixels are taken as the non-terminal nodes of the graph. The binary labels, namely foreground (B) and background (F), are taken as terminal nodes. Our goal is to assign the binary value to each pixel on a single frame by minimizing the following energy function:

$$E(X) = \sum_{i \in \mathcal{V}} D(x_i) + \rho \sum_{\{i,j\} \in \mathcal{E}} V(x_i, x_j) \quad (4)$$

Here, the data term D measures the cost of assigning the label x_i to the pixel i given the observed data. x_i equals either F or B . Let $\mathcal{V}$ be the set of all pixels in the image. The link term V measures the cost of assigning the labels x_i, x_j to the adjacent pixels (i, j) . The signal $\mathcal{E}$ denotes the set of all neighborhood pixels pairs (i, j) . The coefficient ρ specifies the relative importance of the link term.

3.2 Setting Data Term and Link Term

The data term includes $D_i(B)$ and $D_i(F)$ cost, which corresponds to the cost of assigning the pixel i to background and foreground, respectively. we can set $D_i(B)$ that is the cost belonging to the background according to Equation 3. By this cost, the input image are divided into the definitely foreground for which $D_i(B)$ value is high, definitely background for which $D_i(B)$ value is small, and the reminder of pixels are uncertainty region. The cost $D_i(F)$ is set to be certain constant. Note that the normalized cost are used in Equation 4 so as to set coefficient ρ in the same quantity level with the link cost:

$$\begin{aligned} D_i(B) &= D(x_i = B) = \lambda D_i(E) + (1 - \lambda) D_i(U) \\ D_i(F) &= D(x_i = F) = \varphi^2 \end{aligned} \quad (5)$$

Although the cost belonging to the foreground is estimated by a simple function, it is sufficient for the segmentation because the accurate per-pixel background

model has been used. In contrast to the simple foreground cost, we develop a global foreground model to calculate the $D(x_i = F)$. Through experimental contrast, we find that similar segmentation is obtained by using simple foreground cost and global foreground model cost. However, the former has obviously less computation cost than the latter.

The link cost in Equation 4 is designed to keep the segmentation coherent across space. Only if two nodes are assigned to opposite labels, B and F , the link cost is non-zero. Otherwise, the cost of the link (i, j) is zero because of no contribution to the cost of the cut. So the link (i, j) is defined to be:

$$V_{ij}(x_i, x_j) = \begin{cases} \exp(-\frac{d_{ij}}{2\eta^2}) & \text{if } x_i \neq x_j \\ 0, & \text{if } x_i = x_j \end{cases} \quad (6)$$

Where, η can be estimated as “camera noise”, which is set to 5 in practise. d_{ij} denotes the color different between two neighborhood pixels, which is designed referred the literatures [9] using the contrast attenuation.

$$d_{ij} = \|I_i - I_j\|^2 \cdot \frac{1}{1 + \left(\frac{\|I_i^B - I_j^B\|}{K} \right)^2 \exp(-\frac{z_{ij}^2}{\sigma_z})} \quad (7)$$

Where $z_{ij} = \max\{\|I_i - I_i^B\|, \|I_j - I_j^B\|\}$ measures the dissimilarity between pixel pair (I_i, I_j) in image I at the current frame and (I_i^B, I_j^B) in background image I^B obtained from the eigenbackground initialization. The improved link cost adaptively attenuates the contrasts in the background while preserving the contrasts across foreground/background boundaries.

After setting the data term and link term, we minimize the energy $E(X)$ in Equation 4 to achieve the segmentation using the implementation of dynamic graph cut algorithm [10]. Before getting the final binary segmentation results, we use region maker algorithm to remove the small regions.

Figure 2 shows the segmentation results on the original images using the graph cut. In contrast with the last column of the Figure 1, the results in Figure 2 demonstrate that the segmentations are improved by graph cut algorithm.

Fig. 2. The segmentation results using graph cut

4 Experimental Results

We validate our method on indoor video in our laboratory captured by a common web-camera. Meanwhile, our methods are tested on well-known benchmark videos¹. The experimental results are shown from Figure 3 to Figure 6.

Fig. 3. Some segmentation results using graph cut

Fig. 4. Segmentation results compared with Eigenbackground model

¹ <http://cvrr.ucsd.edu/aton/shadow>

Fig. 5. The contrasted segmentation results between ours and wallflow

Fig. 6. Segmentation results on “JMleft” video

Figure 4 compare our method with the Eigenbackground model on public test image and sequence images in our lab. The second column gives the segmentation results by Eigenbackground model. The rightmost column shows the segmentation results through our adaptive background model and shadow removal.

In Figure 5, the first column is the original images. The second column is the ground truth data. The third is the results from the wallflow algorithm [11]. The last is the results obtained by our method

Figure 6 and Figure 8 give some segmentation result and quantitative evaluation. Some segmentation results by our method on the benchmark “JMleft” video from [12] are given in Figure 6.

Figure 7 shows the error rate curve that is drawn based on the segmentation results in the “JMleft” sequences from [12], which demonstrates that few pixels is misclassified and low error rate is achieved by our proposed method using multiple cues. Equation 8 gives how to calculate error rate.

$$Error\ rate = 100\% \times \left((FN + FP) / Image\ area \right) \quad (8)$$

Where FP denotes the number of misclassified non-foreground points, and FN equals to the number of misclassified foreground points.

Fig. 7. Error rate curve on “JMleft” video

From Figure 3, Figure 5 and Figure 6, we can see that the proposed method can well obtain the better foreground segmentation on different videos in real time. In our experiment, the frame rate is about 16-18frames/seconds for a 320x240 video on a 3.0 GHZ laptop with 512MB RAM by using dynamic graph cut algorithm, which can meet the real-time requirement.

5 Conclusion

In this paper, we propose a mixture model that adaptively combines the eigen-background and single gaussian models. The proposed model is robust to noise and illumination change due to inheriting eigenbackground advantages. Meanwhile, it can make use of color information to detect multi-channel changes and remove the shadows by gaussian model. To avoiding the fixed threshold, we use the graph cut algorithm to obtain accurate binary label values. Experimental results on the benchmark video demonstrate that our methods are efficient.

References

1. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 246–252 (1999)
2. Haritaoglu, I., Harwood, D., Davis, L.S.: W4s: A Real-Time System for Detecting and Tracking People in 2 1/2 D. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (1998)

3. Elgammal, A., Harwood, D., Davis, L.: Non-parametric Model for Background Subtraction. In: Proceedings of International Conference on Computer Vision, pp. 751–767 (1999)
4. Han, B., Comaniciu, D., Davis, L.: Sequential kernel density approximation through mode propagation: applications to background modeling. In: Asian Conference on Computer Vision, pp. 922–929 (2005)
5. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfunder: Real-Time Tracking of the Human Body. In: W4s: A Real-Time System for Detecting and Tracking People in 2 1/2 D, pp. 780–785 (1997)
6. Oliver, N., Rosario, B., Pentland, A.: A Bayesian Computer Vision System for Modeling Human Interactions. In: Int'l Conf. on Vision Systems, Gran Canaria, Spain (1999)
7. Schindler, K., Wang, H.: Smooth foreground-background segmentation for video processing. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 581–590. Springer, Heidelberg (2006)
8. Boykov, Y., Jolly, P.M.: Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-d Images. In: ICCV, pp. 102–112 (2001)
9. Sun, J., Zhang, W., Tang, X., Shum, H.Y.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
10. Kohli, P., Torr, P.H.S.: Efficiently solving dynamic Markov Random Fields Using Graph cuts. In: IEEE ICCV, pp. 922–929 (2005)
11. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: Proceedings of International Conference on Computer Vision, pp. 255–261 (1999)
12. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer Segmentation of Live Video. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 53–60 (2006)