Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Foreground Segmentation via Sparse Representation*

Bin Shen, Yu-Jin Zhang
Department of Electronic Engineering, Tsinghua University
Beijing 100084, China
chenbin03@mails.tsinghua.edu.cn, zhang-yj@tsinghua.edu.cn

***Abstract* — In this paper, the problem of foreground segmentation in videos is considered. The *bag of superpixels* is proposed to simultaneously model both the foreground and the background. Then it is demonstrated that an image has a hierarchical structure. Based on this observation, the discriminative nature of sparse representations is exploited to segment the foreground in each frame. Experimental results show that the proposed algorithm can accurately locate the object's position and segment its image support.**

***Keywords*—Foreground segmentation, sparse representation, L_1 norm minimization**

I. INTRODUCTION

Foreground tracking is an important issue in computer vision, and accurate object segmentation is even more useful. This can be helpful for area surveillance, user action analysis, intelligent video editing, etc.

Many researchers [2, 3, 8] have proposed various methods to conduct object tracking. They try to model the foreground objects, and then find their locations in every frame of the video. However, they usually do not model the background information. Some others [6, 7, 10] focus on background modeling, and try to segment the objects in video sequences. Meanwhile, they may not pay adequate attention for foreground information.

Recently, sparse representation has been adopted in many computer vision areas, such as face recognition [13], image inpainting [12], denoising [9], etc. Many exciting works result from that the optimal representation is sufficiently sparse. Also, Donoho shows that these corresponding optimization problems could be efficiently solved by using the penalty of L_1 norm [4].

In this paper, we propose the *bag of superpixels* to simultaneously model both the foreground and background. Then, based on this model we are able to compute sparse representations in every frame. Finally, the discriminative nature of sparseness helps to accurately find the foreground which we are interested in.

The rest of this paper is organized as follows: section 2 describes our key observation, which explains why a sparse representation of the target object exists. Section 3 describes our proposed algorithm, and experimental results are shown in section 4. We discuss our algorithm and conclude this paper in section 5.

Figure 1: Images and segmentations

II. KEY OBSERVATION

We observe that an image has a hierarchical structure: an image is made up of several segments, and each segment is made up of several superpixels (see below). More importantly, the foreground and background are made up by different sets of segments.

First we segment an image into several parts. We observe that the object which we are interested in (i.e. the foreground) and the background in an image are made up of several segments. Actually, this always holds or at least approximately holds. See Figure 1 for example, the left column shows the original images, and the right shows the segmentation results by [5]. We can easily see that the foreground and background are made up of different sets of segments respectively. This is the first level of the hierarchical structure. So, foreground segmentation can be formulated as classifying all segments in an image into two classes: foreground and background. Then, we can easily combine all the segments belonging to foreground to gain the foreground image support, and the rest for the background.

For the second level of the hierarchical structure, we introduce the concept of "superpixel" [11]. We over segment the image, and get many small regions, which we call superpixels. The left image in Figure 2 shows the segmentation result, and the right image shows the over-segment result,

* Project 60872084 supported by NSFC

Figure 2: Over-segment results (superpixels)

i.e., the superpixels. Approximately, each superpixel in right image can be assigned to a segment in the left image. So, each segment is made up of superpixels.

Now, for the first level of this hierarchical structure, our foreground segmentation problem turns into how to classify the segment. An important following step is to extract the features of the segment for classification. Luckily, the second level of the hierarchical structure enables us to find a sparse representation for each segment, which will be shown in the next section.

III. Sparse Representation for Foreground Segmentation

Now we exploit the property of the observed structure above, propose a sparse representation for each segment, and then exploit the discriminative nature of sparse representation to classify each segment as foreground or background.

A. Segment and superpixel description

We use a very simple descriptor, the color histogram, to describe each segment and each superpixel. We use H^m to denote the color histogram of the m-th segment. H^m has k bins, and $H_i^m (i = 1,2, \dots k)$ denotes the number of pixels which are quantized into the i-th bin. $\widetilde{\boldsymbol{H}}^m$ denotes the normalized color histogram, *i.e.*

$$\tilde{H}_i^m = \frac{H_i^m}{\sum_j H_j^m} \tag{1}$$

Then, we use h^n to denote the color histogram of the n-th superpixel, h_j^n to denote the number of pixels which are quantized into the j-th bin in the n-th superpixel. Also, the $\tilde{h}_j^n$ (i = 1, 2, …k) is the corresponding normalized color histogram.

We have claimed in section 2 that each segment is a combination of several superpixels, so we get the following equations:

$$H_i^m = \sum_{n \in S(m)} h_i^n \tag{2}$$

$$\tilde{H}_i^m = \sum_{n \in S(m)} \alpha_n \tilde{h}_i^n \tag{3}$$

where $S(m)$ denotes the index of superpixels in the segment m, and α_n is the ratio of the number of pixels in superpixel n to the number of pixels in segment m. Note that there are only several superpixels in a segment. It is where we will exploit the sparsity.

B. Bag of superpixels for foreground and background modeling

Inspired by that the foreground and background are made up of segments, and that each segment is made up of superpixels, we here propose to use *bag of superpixels* to model the foreground and background.

Now we learn the foreground and background model in our problem. In the first one or several training frames, we manually select the segments which make up the foreground. Based on this manual initialization, we get several segments for foreground, and the rest for background. Also, using segmentation algorithm [5] to over segment the frame(s), we get many superpixels for these foreground segments, and many for these background segments, which we call training foreground superpixels and training background superpixels for convenience.

We define a matrix $\boldsymbol{M}$ for the foreground/background model as a concatenation of the color histograms of foreground superpixels and background superpixels:

$$\boldsymbol{M} \triangleq [\tilde{\boldsymbol{h}}^{f,1}, \tilde{\boldsymbol{h}}^{f,2}, \dots, \tilde{\boldsymbol{h}}^{f,p}, \tilde{\boldsymbol{h}}^{b,1}, \tilde{\boldsymbol{h}}^{b,2}, \dots, \tilde{\boldsymbol{h}}^{b,q}] \tag{4}$$

where $\tilde{\boldsymbol{h}}^{f,u}$ denotes the normalized color histogram for the u-th foreground superpixel, and $\tilde{\boldsymbol{h}}^{b,v}$ for the v-th background superpixel. Obviously, $\boldsymbol{M}$ is a $k \times (p+q)$ matrix. This means that there are totally p training foreground superpixels and q training background superpixels. Considering the formulation of $\boldsymbol{M}$, we call this model *bag of superpixels*.

C. Sparse representation for testing segment

For a testing frame, the entire image is segmented into test segments. Then for each segment, we get its color histogram description $\widetilde{\boldsymbol{H}}^m$. The linear representation in (3) can be rewritten in terms of all the training superpixels as

$$\widetilde{\boldsymbol{H}}^m = \boldsymbol{M}\boldsymbol{x}^m \tag{5}$$

where $\boldsymbol{x}^m$ is the coefficient vector over $\boldsymbol{M}$ to reconstruct the $\widetilde{\boldsymbol{H}}^m$. So the entry of the coefficient vector $\boldsymbol{x}^m$ is not zero only if the segment m contains the superpixel i. Thus, x^m reflects what superpixels are within the image support of segment.

Based on what superpixels the segment contains, we classify the segment as foreground or background. The details of the classification algorithm will be described later.

Since the entries of $\boldsymbol{x}^m$ encode the identity of the segment m, it is attractive to attempt to obtain $\boldsymbol{x}^m$ by solving the k linear equations described in (5). However, we may easily note that in our problem, there are so many training superpixels that the equations (5) are typically underdetermined, and the solution is always not unique.

In subsection *A*, we observe that there are only several superpixels contained in any segment, thus a segment can be sufficiently represented using only these superpixels which it contains, so the coefficient vector $\boldsymbol{x}^m$ is naturally sparse.

According to this reason, we try to seek the sparsest solution to (5). This optimization problem can be described as:

$$\hat{\boldsymbol{x}}^m = \arg\min ||\boldsymbol{x}^m||_{\boldsymbol{0}} \text{ subject to } \tilde{\boldsymbol{H}}^m = \boldsymbol{M}\boldsymbol{x}^m \tag{6}$$

where $||\cdot||_0$ denotes the L_0 norm, *i.e.* the number of the nonzero entries of a vector.

However, it is NP hard to solve (6) [1]. Luckily, Donoho [4] shows that the L_0 norm constraint can be approximated by the L_1 constraint, *i.e.*, the optimization problem (6) can be well approximated by the following one:

$$\hat{\boldsymbol{x}}^m = \arg\min ||\boldsymbol{x}^m||_{\boldsymbol{1}} \text{ subject to } \tilde{\boldsymbol{H}}^m = \boldsymbol{M}\boldsymbol{x}^m \tag{7}$$

Equation (7) can be efficiently solved.

However, we should note that $\tilde{\boldsymbol{H}}^{\boldsymbol{m}}$ describes a segment in a test frame, and $\boldsymbol{M}$ is constructed from the training frame(s). Due to foreground's appearance change, such as non-rigidity, scale change, or view angel change, $\tilde{\boldsymbol{H}}^m$ may be contaminated by some noise. So the model (5) should be modified into

$$\tilde{\boldsymbol{H}}^m = \boldsymbol{M}\boldsymbol{x}^m + \boldsymbol{z} \tag{8}$$

where $\boldsymbol{z}$ is the noise.

Accordingly, (7) should be modified into

$$\hat{\boldsymbol{x}}^m = \arg\min ||\boldsymbol{x}^m||_{\boldsymbol{1}} \text{ subject to} ||\tilde{\boldsymbol{H}}^m - \boldsymbol{M}\boldsymbol{x}^m||_2 \le \boldsymbol{\varepsilon} \tag{9}$$

where $\boldsymbol{\varepsilon}$ is the bounded energy of the noise $\boldsymbol{z}$. This optimization problem can also be efficiently solved. The result $\hat{\boldsymbol{x}}^m$ can be viewed as a sparse representation of the segment m.

D. Segment classification

Now we use the computed $\hat{\boldsymbol{x}}^m$ to classify the segment m as foreground or background.

Since $\boldsymbol{M}$ is made up of p foreground superpixels and q background superpixels, the first p entries of $\hat{\boldsymbol{x}}^m$ encode the probability of segment m to be foreground, while the last q entries encode the probability to be background. $\hat{\boldsymbol{x}}^m$ can be viewed as a concatenation of $\hat{\boldsymbol{x}}^{m,f}$ and $\hat{\boldsymbol{x}}^{m,b}$, which are made up of the first p and last q entries of $\hat{\boldsymbol{x}}^m$ respectively, i.e. $\hat{\boldsymbol{x}}^m = [\hat{\boldsymbol{x}}^{m,f}, \hat{\boldsymbol{x}}^{m,b}]$.

We define the confidence of the segment m being a part of foreground as:

$$\mathrm{c}_m = \frac{||\hat{\boldsymbol{x}}^{m,f}||_0 + \xi}{||\hat{\boldsymbol{x}}^{m,b}||_0 + \xi} \tag{10}$$

where $||\cdot||_0$ denotes the L_0 norm and ξ is a small positive constant.

We classify the segment m as foreground or background by simply thresholding the confidence. The threshold t is initially learned from the segments in the training frame(s), and then updated in each frame. We firstly set the threshold as the arithmetic average of the lowest confidence of the foreground segment and the highest confidence of the background segment, and then update it as the arithmetic average of the lowest confidence of the foreground segment and the highest confidence of the background segment in the previous test frame. Of course, here the classifier can be replaced by any of other ones, such as support vector machine, AdaBoost, and the computed $\hat{\boldsymbol{x}}^m$ can be simply treated as the input feature vector. The classifier can be trained in the manual initialization. Also, if some incremental classifier, such as incremental SVM, is adopted, it may be updated in the testing frames.

We also update the model $\boldsymbol{M}$ in every frame. For a segment in a testing frame, if $\mathrm{c}_m > \alpha t$, add the normalized color histogram of the superpixels of the segment into $\boldsymbol{M}$ as foreground basis; if $\mathrm{c}_m < t/\alpha$, add the normalized color histogram of the superpixels of the segment into $\boldsymbol{M}$ as background basis. If $\boldsymbol{M}$ has more than s colums, resample $\boldsymbol{M}$ so that $\boldsymbol{M}$ has s colums, which is the largest allowable size of $\boldsymbol{M}$.

E. Overall algorithm

Now we summarize our algorithm in Figure 3 for convenience.

Input: One or several frames (k training frames) with the foreground manually selected.
1, Train:
1),Get superpixels by over-segmenting training frames.
2),Assign each superpixel to foreground or background with larger overlapping area.
3), Get the model $\boldsymbol{M}$ by computing the normalized color histograms of all superpixels.
2, Test:
For each test frame
1), Segment the current frame to get several segments.
For each segment:
i), Compute the normalized color histogram $\tilde{\boldsymbol{H}}$, then the $\hat{\boldsymbol{x}}$ and its confidence c.
ii), Classify this segment as foreground or background based c and threshold t.
End
2), Update the model $\boldsymbol{M}$, p, q, and the threshold t
End
Output: All the test frames with their foregrounds automatically segmented.

Figure 3: Overall algorithm

Figure 4: Initialization

IV. Experimental Results

To verify our algorithm, we have tested it on many videos.

We first test it on an image sequence. First we manually select the foreground. It is very convenient since we only need to specify the segments which belong to the foreground after segmentation. The details of initialization algorithm are shown in Figure 4. The images in Figure 4 from left to right are the original image, the segmented one for manually selection, the selected foreground mask, and the over-segmented result (superpixel).

We test our algorithm on the remaining frames, and the results are shown in Figure 5. The top row shows original images, the middle shows the segmentation result, and bottom row shows the final segmentation result. We can easily see that this algorithm can very accurately find the image support of the foreground. More importantly, note that there is global motion for the background in the test video, and our algorithm can be adapted to the moving background. Usual background modeling techniques [6, 7, 10] need more frames to construct the background model, and usually require that there be no global motion for the background. Traditional tracking algorithms [2, 3, 8] may be able to deal with moving background, but they cannot accurately segment the image support of foreground.

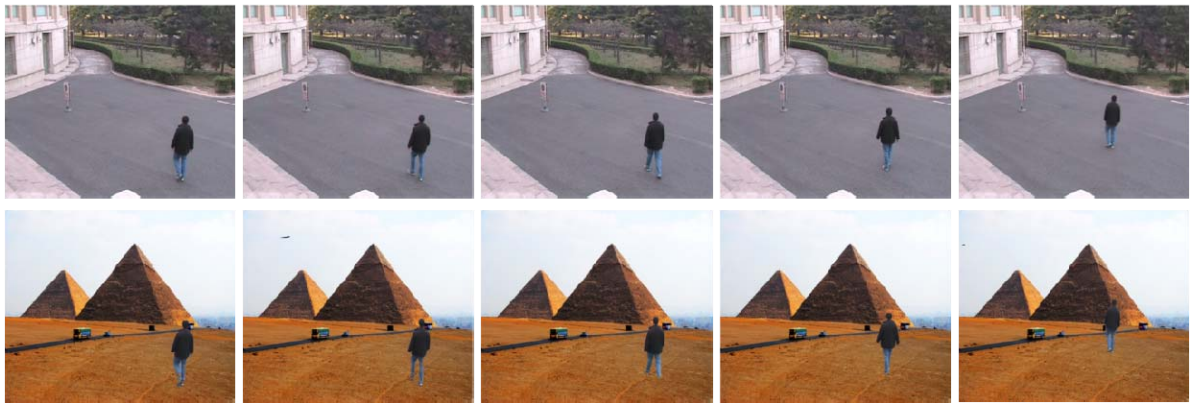
Also, we test the algorithm on other videos, including both videos containing large high resolution foregrounds and others containing relative small foregrounds. The algorithm accurately exacts the foreground of the frames, and we then replace the background in the sequences for interest. The background replacing results are shown in Figure 6.

V. Discusiion and Conclusion

In this paper, we propose the *bag of superpixels* for both foreground and background modeling. Base on this model, we find the image support of the foreground. This algorithm computes any sparse representation for each segment by L_1 minimization in each frame, and classifies the segment as foreground or background based on the coefficient vector. Experimental results show our algorithm can effectively conduct foreground segmentation. Since our algorithm does not rely on the spatial information among different superpixels, it is robust to camera motion. Moreover, we only make use of the color histogram as descriptors of superpixel and segment, thus it should be robust to rigid and non-rigid transformation of the foreground and background. Of course, other descriptors, such as HoG, LBP histogram, can replace color histogram in our algorithm for certain special purposes, such as robustness to illumination change. The main framework needs not be

Figure 5. Segmentation result on testing sequence

(a) Replace background for relative large foreground video



(b) Replace background for relative small foreground video

Figure 6: More experimental results

modified except for the feature exaction module.

Our algorithm employs the *bag of superpixels* to model the foreground and background, and gains exciting results in foreground segmentation. Actually, we find that this *bag of superpixels* model can be used in other problems, such as pedestrian identification in relative low resolution video where face recognition is not valid, or video retrieval for object of interest, etc.

ACKNOWLEDGMENT

Here we would like to thank Professor Yung-Yu Chuang for providing us valuable testing data.

REFERENCES

[1] E. Amaldi and V. Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,” Theoretical Computer Science, vol. 209, pp. 237–260, 1998.

[2] S. Avidan. “Support vector tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(8):1064–1072, Aug. 2004.

[3] R. T. Collins. “Mean-shift blob tracking through scale space,” Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2003.

[4] D. Donoho, “For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution,” Communications on Pure and Applied Mathematics, 59 (2006), pp. 907-934.

[5] P. F. Felzenszwalb, D. P. Huttenlocher, “Efficient graph-based image segmentation,” International Journal of Computer Vision. 59(2) pp. 167-181. 2004.

[6] A. R. François and G. G. Medioni, "Adaptive color background modeling for real-time segmentation of video streams," Proceedings of the International Conference on Imaging Science, Systems, and Technology, pp. 227–232, (Las Vegas, NA), 1999.

[7] D. Haritaoglu, I. Harwood and L. Davis, "W4: Who? when? where? what? a real time system for detecting and tracking people," International Conference on Face and Gesture Recognition, 3, ed., April 1998.

[8] M. Isard and A. Blake, “CONDENSATION -- conditional density propagation for visual tracking,” Int. J. Computer Vision, 29, 1, 5--28, (1998).

[9] J. Mairal, M. Elad,and G. Sapiro, "Sparse representation for color image restoration," IEEE Transaction on image processing,vol.17, No. 1, Jan. 2008.

[10] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transaction on Systems, Man,and Cybernetics, pp. 62–66, 1979.

[11] X. Ren and J. Malik, “Learning a classification model for segmentation,” Proceedings of International Conference on Computer Vision, 2003.

[12] B. Shen, W. Hu, Y. Zhang, Y.-J. Zhang, “Image inpainting via sparse representation,” Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 697-700, 2009.

[13] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Transaction on Pattern Analysis and Machine Intelligence, 31(2): 210-227, Feb. 2009.