PEOPLE DETECTION USING FEATURE VECTOR MATCHING

Hwa-Young Kim[1] *and Rae-Hong Park*[1, 2]

[1] Dept. of Electronic Engineering, Sogang University
[2] Interdisciplinary Program of Integrated Biotechnology, Sogang University
C. P. O. Box 1142, Seoul 100-611, Korea
{lily, rhpark}@sogang.ac.kr

ABSTRACT

This paper presents a novel method to detect people in real time from a video sequence taken from a single fixed camera. Foreground images are obtained from a sequence of operations: background differencing, thresholding, and morphological operation. We make use of quantized edge orientation to represent shape of human because human has particular edge orientation distribution on each of body parts. Then, a feature vector derived from the edge orientation map is computed in each of body parts of foreground images. The dataset, composed of sets of feature vectors, is used for matching with feature vectors computed from the current image of the input sequence. We use the K-nearest neighbors to match the feature vectors. The proposed people detection method uses a proper feature vector that represents human edge orientation distribution, and employs simple matching steps for real-time processing. Experimental results with a number of test sequences demonstrate the effectiveness of the proposed method.

Index Terms— crowd segmentation, people detection, edge orientation map, surveillance

1. INTRODUCTION

Many computer vision applications such as security, safety, smart interfaces, and video retrieval require detection and tracking of people. These applications usually use fixed cameras, background modeling, and change detection.

Although a number of people detection and tracking methods, which use various image processing and computer vision techniques, have been developed, they still have problems in localizing individual humans in a crowd.

For splitting crowd to individual humans, many methods with human appearance modeling and composing features have been proposed. Among them, Rittscher *et al.* proposed an expectation maximization (EM) formulation which extracts shape parameters for all potential individuals and makes feature assignments [1]. The approach achieved global optimization without reliance on random search and strong initialization.

Wang and Suter presented a sample consensus based method, which utilizes both color and spatial information of human bodies, to model the appearance of people [2].

Kong *et al*. developed a viewpoint invariant learning based method using simple features and the feature normalization to deal with perspective projection and camera orientation [3].

In order to detect pedestrians in videos acquired, Ran *et al*. described two approaches based on gait [4]. The first approach estimates a periodic motion frequency using two hypothesis testing steps to filter out non-cyclic pixels. In the second approach, they converted the cyclic pattern into a binary sequence by maximal principal gait angle fitting.

We propose a simple feature vector matching method for human detection and segmentation. The approach takes advantage of K-nearest neighbors for a real-time surveillance system. Features of human are derived from the edge orientation map for feature vector matching.

2. FOREGROUND REGION DETECTION

For foreground extraction we modify Haritaoglu *et al*.'s method [5] which is verified and works quite well. A single fixed camera allows us to use all the background scene components. For background scene modeling, we compute three values (minimum and maximum intensity values and the largest interframe absolute difference) at each pixel over several seconds of video. These values can be updated periodically with some parts of a background scene containing no foreground objects.

Foreground regions are extracted from the background in each frame of the video sequence in three steps: background differencing, thresholding, morphological operation.

At each pixel (x, y), a foreground pixel is classified using the background model. Given the minimum (N), maximum (M), and the largest interframe absolute difference (D) images, pixel (x,y) with intensity $I(x,y)$ is determined as foreground pixel if

$$|M(x,y)-I(x,y)|>D(x,y) \text{ or } |N(x,y)-I(x,y)|>D(x,y) \tag{1}$$

is satisfied. After this thresholding step, there are still many clutters, for example, due to illumination changes. So we apply the morphological operation to the thresholded region. First, erosion is applied to foreground pixels to eliminate small isolated noises. Then, we detect the foreground object if its area is larger than an arbitrary threshold, which is followed by a single iteration of dilation and erosion operations. Fig. 1(a) shows a test image (384×288) of CAVIAR dataset [6] and Fig. 1(b) is the result of foreground extraction. Because of shadows, there are long bounding boxes. So we should reduce unnecessary foreground area at the next stage.

3. PEOPLE DETECTION

We use a feature vector to distinguish people in the extracted foreground. Feature vector matching with dataset can detect people. In this section we describe features that are used to decide whether the detected foreground is people or not.

3.1. Feature extraction

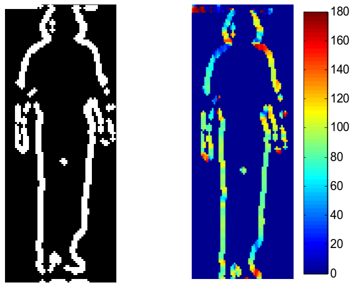
Before feature extraction, to reduce the influence of illumination, we use image normalization, in which square root operation of each color channel is employed as power-law compression.

In this paper, we use a simple feature vector matching, in which selection of proper features is important. The proposed people detection method requires real-time processing, so we prefer simple features based on edge information. At first, we apply Canny edge detector to each frame to compute the edge and orientation maps. Then, we compute features only in the part that is obtained by AND operation between the edge map and the detected foreground region. Figs. 2(a) and 2(b) show the edge and orientation maps, respectively. Fig. 2(b) shows the different orientation features of a human body in terms of the edge orientation map. In order to select proper feature details, we assume that human body can be divided into three parts such as head, torso, and legs because each part has different edge orientation distribution. For example, legs part has many vertical or diagonal components whereas it has a few horizontal ones. Thus, to efficiently represent the features of people we quantize non-uniformly the orientation angle to r intervals.

In learning step, we use the features in each body part region (e.g., rectangular region) that contains each type of

(a) (b)

Fig. 1. Foreground extraction. (a) Input image (CAVIAR data set, 384×288). (b) Detected foreground



(a) (b)

Fig. 2. Edge features. (a) Canny edge detection map. (b) Edge orientation map around edge area.

the human body. We first segment a subimage that contains a human body upto three body part regions containing human body parts. Then the edge map and orientation map are computed in each body part region. Next, we divide the subimage into a number of $\delta\times\delta$ sized square blocks and derive the edge orientation features block by block. Contrast normalization is applied to each block for detection of robust edge features. Then, we count the number of pixels whose orientation values are within the non-uniformly quantized orientation intervals. Let $\mathbf{o}$ be an edge orientation vector defined as

$$\mathbf{o}=[o^1\ o^2\ \cdots\ o^R]^T \tag{2}$$

where o^r denotes the rth feature component of $\mathbf{o}$, representing the number of pixels in the non-uniformly quantized orientation interval r (between θ_r and θ_{r+1}), $1\le r\le R$. We divide the orientation from 0° to 180° with 44 intervals. As previously stated, to non-uniformly quantize the orientation, we divide more finely the orientation around 45°, 90°, and 135°. Fig. 3(a) shows quantization levels, in which the quantization index is represented as a function of the orientation angle. The level indices are abruptly changed around 90°. Fig. 3(b) shows examples of the orientation angle distribution of three different parts: head part (no predominant direction), leg part (dominant vertical direction), and shoulder part (dominant horizontal direction) in the first, second, and third rows, respectively. Then, we formulate a feature vector for each $\delta\times\delta$ sized square block.

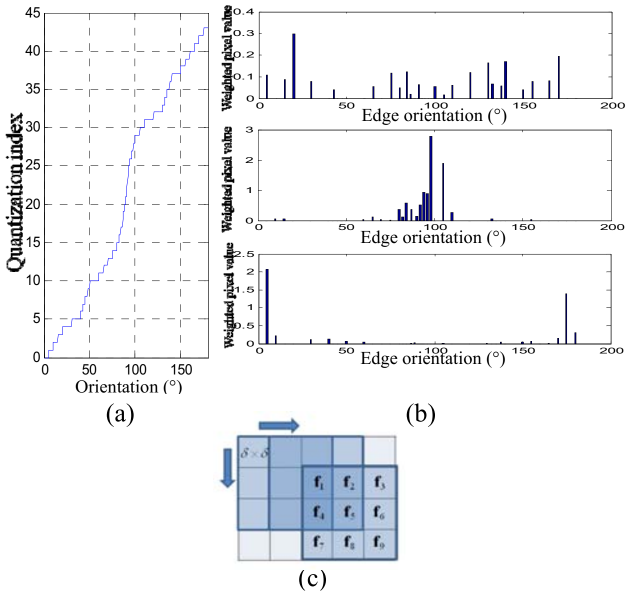


Fig. 3. Feature components. (a) Orientation quantization. (b) Examples of the edge orientation distribution of three different parts. (c) Feature components of the center block $\mathbf{f}_5$.

After computing feature components for every block, we combine nine feature components in a 3×3 block, which is shown in Fig. 3(b), defined as

$$\mathbf{f} = [\mathbf{o}_1\ \mathbf{o}_2\ \mathbf{o}_3\ \mathbf{o}_4\ \mathbf{o}_5\ \mathbf{o}_6\ \mathbf{o}_7\ \mathbf{o}_8\ \mathbf{o}_9]^T \tag{3}$$

where $\mathbf{f}$ represents a feature vector of the center block $\mathbf{o}_5$ and $\mathbf{o}_i$ $(1 \le i \le 9)$ denotes a feature vector component of each block in the 3×3 block. Using the feature components indicated in Fig. 3(b), Fig. 4 shows two cases of feature vector distributions that are computed from 3×3 neighboring blocks. The feature vector distribution is expressed in terms of the weighted pixel value as a function of the feature vector index (44×9). The weighted pixel value is defined by the weighted sum of gray levels of pixels corresponding to each edge orientation index, in which the weight is determined by the edge magnitude. In the first row, only two feature vector components in nine blocks are considered, thus the center block is assumed to be a little bit outside edge area. On the other hand, in the second row the center block lies in the middle of edge area.

3.2. Feature vector matching

K-nearest neighbor algorithm classifies objects based on the closest training examples existing in the feature space. The algorithm is a very simple method that requires less computational load for real-time processing. To compare the similarity with dataset, we employ the Tanimoto metric [7] that takes on values between 0 and 1, in which 1 represents an identity similarity between two feature vectors. Fig. 5 shows the result of feature matching with each body part segmented as head, torso, and legs.

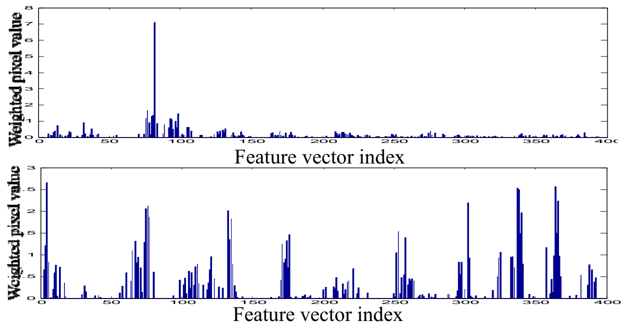


Fig. 4. Examples of the weighted pixel value

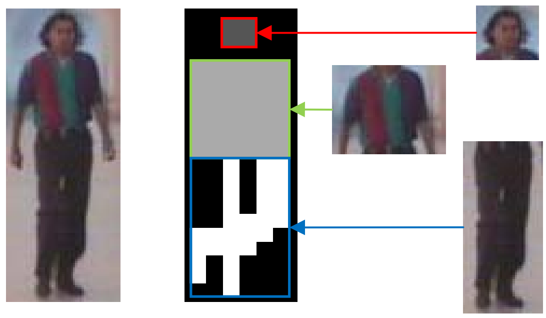


Fig. 5. Feature matching result of each body part segmented as head, torso, and legs.

We also use features in the current image that is not employed for training. A complex image with many people and difficult pose is hard to detect and segment people at first. However, after a number of features are trained, detecting people in the complex image is quite possible.

4. EXPERIMENTAL RESULTS AND DISCUSIONS

During training, to classify the feature vectors into head, torso, and legs, we add a tag. We indicate each part by different gray level and bounding box color as shown in Fig. 5, in which the most similar feature vector has a similarity measure larger than 0.5. We divide an image into 6×6 blocks.

Experimental results are shown in Fig. 6. The first column shows original images (384×288) from CAVIAR dataset. The result of foreground detection is shown in the second column. We set bounding boxes in those images but there is an incorrectly detected one. We should eliminate it and split a bounding box into classified body part boxes, such as head (red), torso (green), and legs (blue). Fig. 6(c) is the segmentation result of body parts. Most people are detected but few parts are missed because they occupy too

(a) (b) (c)

Fig. 6. Experimental results. (a) Original video sequence. (b) Foreground detection. (c) Human detection and segmentation of body parts as head (red), torso (green), and legs (blue).

small area to detect. The proposed algorithm is to be improved to detect people far from a camera.

The average computation time to extract people is 0.8 sec using MATLAB on an AMD Athlon 64X2 Dual Core Processor 3800+ 2.01GHz machine, from which we easily verify the performance of the proposed method. The proposed method can be applied to real-time applications using a fast implementation program.

5. CONCLUSION

We have presented a framework for detecting human and for segmenting body part as head, torso, and legs in real video sequences. The proposed algorithm makes use of the edge orientation around edge area. Edge orientation can be applied to describing the human shape information: human body has various orientation components at one's head and vertical or diagonal orientation components at one's torso and legs. We also use shape information in construction of feature vectors by dividing an image into a number of subimages. Experimental results demonstrate that the proposed features using edge orientation map effectively detect human. Further research will focus on the improvement of a classification method for reliable detection of people in more complex scenes.

Acknowledgment This work was supported by the Second Brain Korea 21 project.

6. REFERENCES

[1] J. Rittscher, P. Tu, and N. Krahnstoever, "Simultaneous estimation of segmentation and shape," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 486–493, San Diego, CA, USA, June 2005.

[2] H. Wang and D. Suter, "Tracking and segmenting people with occlusions by a sample consensus based method," in *Proc. Int. Conf. Image Processing*, vol. 2, pp. 410–413, Genova, Italy, Sept. 2005.

[3] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *Proc. Int. Conf. Pattern Recognition*, vol. 3, pp. 1187–1190, Hong Kong, China, Aug. 2006.

[4] Y. Ran, I. Weiss, Q. Zheng, and L. Davis, "Pedestrian detection via periodic motion analysis," *Int. Journ. Computer Vision*, vol. 71, no. 2, pp. 143–160, Feb. 2007.

[5] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[6] CAVIAR dataset: http://homepages.inf.ed.ac.uk/rbf/CAVIAR.

[7] S. E. Umbaugh, *Computer Imaging: Digital Image Analysis and Processing*. Boca Raton, Florida: CRC Press, USA, 2005.