中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-08-009

An Online Boosted People Counting System for Electronic Advertising Machines

Duan-Yu Chen, Chih-Wen Su, Yi-Chong Zeng, Shih-Wei Sun, Wei-Ru Lai, and Hong-Yuan Mark Liao

September 4, 2008 || Technical Report No. TR-IIS-08-009

http://www.iis.sinica.edu.tw/page/library/LIB/TechReport/tr2008/tr08.html

An Online Boosted People Counting System For Electronic Advertising Machines

Duan-Yu Chen[a], Chih-Wen Su[a], Yi-Chong Zeng[a], Shih-Wei Sun[a], Wei-Ru Lai[b], and Hong-Yuan Mark Liao[a]

[a]*Institute of Information Science, Academia Sinica, Taiwan*

Department of Communications Engineering, Yuan Ze University, Taiwan

[a]{*dychen,* lucas, *yichongzeng, swsun, liao*}@iis.sinica.edu.tw

[b]*wrlai*@saturn.yzu.edu.tw

Abstract

This paper presents a novel people counting system for an environment in which a stationary camera can count the number of people watching a TV-wall advertisement or an electronic billboard without counting the repetitions in video streams in real time. The people actually watching an advertisement are identified via frontal face detection techniques. To count the number of people precisely, a complementary set of features is extracted from the torso of a human subject, as that part of the body contains relatively richer information than the face. In addition, for conducting robust people recognition, an online boosted classifier trained by Fisher's Linear Discriminant (FLD) strategy is developed. Our experiment results demonstrate the efficacy of the proposed system for the people counting task.

Keywords: people counting, video surveillance

1. Introduction

Counting the number of people in a public place (e.g., a street, shopping mall, or subway station) over time is very important in many real-world applications. For instance, a crowd gathering at a specific place may indicate an unusual situation or event. On the other hand, counting the number of people in a shopping mall may provide valuable information for optimizing trading hours, as well as evaluating the attractiveness of some shopping areas. In this paper, we focus on counting the number of people watching a TV-wall advertisement monitor or an electronic billboard. With the advent of intelligent cameras and the increasing capabilities of video surveillance, automation of people-counting is now technically possible.

In recent years, a great deal of research [2-3, 20-21] has been directed at providing more accurate people counting methods. Generally, the developed methods can be categorized into two types: people detection-based and feature-based approaches. In people-based approaches, once people have been detected, they can be counted easily. For instance, in the W4 system proposed by Haritaoglu et al. [17], shape information is used to identify individuals; while Viola et al. [18] employ boosted classifiers to detect pedestrians by using appearance and motion clues. The main problem with these approaches is that their applicability is limited. In some cases, such as when people walk next to each other and/or occlude each other, the detection/tracking process may fail. In contrast, feature-based approaches do not include a people detection step, but try to transform the people-counting problem into some feature space using computer vision techniques. Typically, these methods extract features based on edge density [19], edge orientation [20], the number of moving pixels [21-22][19], blob size [2-3][20], fractal dimension [23] or multiple clues [24] to estimate the number of people in a scene. Then, a classifier, such as a trained neural network, is applied to perform classification based on the extracted features.

In existing approaches, the number of people counted is an approximation of the actual

number of people in the field of view of the camera. For instance, Chan et al. [3] first localize motion areas and then separate the areas into individual blobs. Then, the sizes of blobs are estimated to determine the number of people. However, for an electronic advertisement billboard, the advertising agent may ask: "How many people actually watched the advertisement in the last five hours?" Current people-counting systems cannot answer this question. The unique feature of the proposed people-counting system is that it does not repeatedly count an individual if he/she watches an advertisement for a long period. Clearly, to solve the problem, a face recognition system must be built. This requirement is very different from existing people-counting systems.

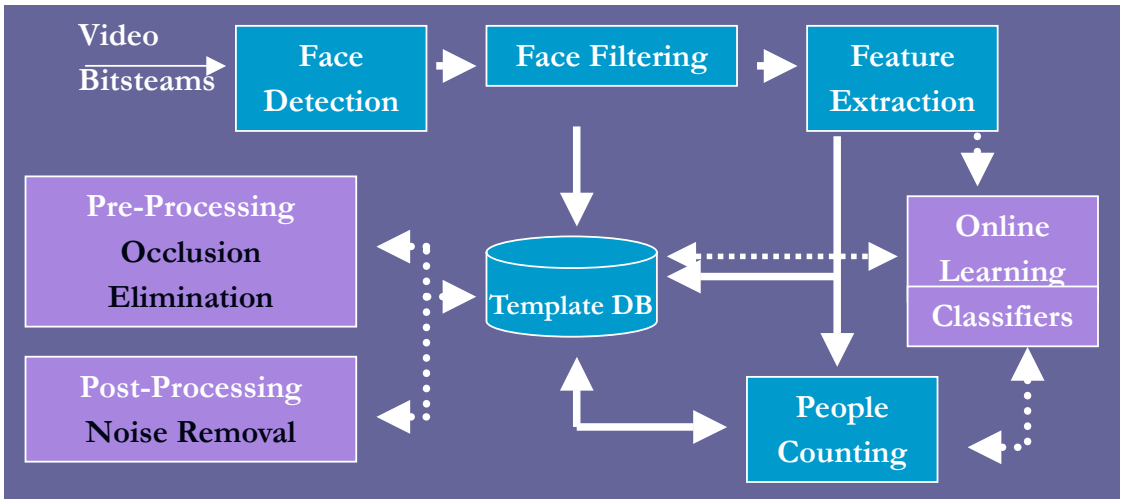

Fig.1. Overview of the proposed people counting system

In this paper, we propose a people counting system that comprises a face recognition module and a first-in first-out face database. The system architecture is illustrated in Fig. 1. First, the system performs a face detection step to identify people who are actually watching the advertisement, rather than simply standing in the area. Since some non-facial regions may be detected accidentally, we design a face filtering process to verify that a detected region is in fact a face part. Then, we extract features directly from that region. However, since the information in the pure face portion is insufficient, we extract features from the torso region to compensate for the deficiency. To ensure robust people recognition we have developed an

online boosted classifier based on Fisher's Linear Discriminant (FLD) criterion. By using the features extracted from the face and part of the torso region, the system can effectively execute the people counting task.

The remainder of this paper is organized as follows. In Section 2 we explain how face detection and face filtering are performed. Section 3 describes the feature set used for human subject recognition. In Section 4, we discuss the online boosted classifier, which is trained by using Fisher's Linear Discriminant criterion. We then detail our experiment results in Section 5, and summarize our findings in Section 6.

2. Face Detection and Face Filtering

In this phase, we first use the support vector machine based (SVM-based) face detector developed by Kienzle et al. [4] to perform the face detection task. Then, we apply the proposed filtering process to remove false positives detected by the above face detector.

2.1 SVM-based Face Detector

To count the number of people watching an advertisement on a TV billboard, it is necessary to detect frontal part of faces because the people are "really watching" the advertisement. Therefore, face detection is the first important step to be accomplished. To satisfy the real-time requirement, we adopt the SVM-based face detector developed by Kienzle et al., which can provide fast approximations of support vector decision functions. It has been shown that an SVM can provide highly accurate object detection results [26]. Research conducted to speed up kernel selection has focused on developing new ways to reduce the number of expansions, i.e., the number of support vectors, and the number of operations needed to compute the similarities between a support vector and the input.

To reduce the number of support vectors, we use Burges' reduced set approach [5]. In an SVM, the decision rule for a test pattern X is represented by [5]

$$f(X) = sign\left(\sum_{i=1}^{m} y_i \alpha_i k(X_i, X) + b\right), \tag{1}$$

where $\{X_i\}, i \in [1, m]$ is the set of support vectors, y_i is the label for each support vector, α_i is the corresponding coefficient, k is the kernel function, and b denotes the bias. The decision surface induced by f is a hyperplane in the reproduced kernel Hilbert space associated with k. The corresponding normal can be approximated using a reduced set $\{Z_j\}$, $j \in [1, m']$, where the size of the reduced set $m' < m$. Therefore, the new decision function f' of the reduced set is denoted by

$$f'(X) = sign\left(\sum_{j=1}^{m'} \beta_j k(Zj, X) + b\right), \tag{2}$$

where $\{\beta_j\}$, and $j \in [1, m']$ is a new corresponding coefficient set for the reduced set $\{Z_j\}$. We analyze the complexity of the kernel function as follows. When one classifies image patches of size $h \times w$ using features with plain gray values, the decision function requires $h \times w$ operations for each pixel. If a filter is linearly separable, the computational complexity of the filtering operation can be reduced from $O(h \times w)$ to $O(h + w)$ per pixel by computing

$$J = [I * a] * b^T, \tag{3}$$

where I is an input image; J denotes an output image; and a and b are column vectors decomposed from the filter mask H defined by

$$H = ab^T. \tag{4}$$

Eq.(3) decomposes the computation of the convolution of two dimensional patches into two independent convolution operations with mask sizes $h \times 1$ and $1 \times w$, respectively. For generalization, we consider the singular value decomposition of H:

$$H = \sum_{i=1}^{r} s_i u_i v_i^T, \tag{5}$$

where r is the rank of H; s_i is the ith singular value of H; and u_i and v_i denote the ith columns of orthogonal matrices U and V, respectively. Accordingly, the linear filter can be evaluated as a linear combination of r separable convolutions by

$$J = \sum_{i=1}^{r} s_i \left[I * u_i \right] * v_i^T \ . \tag{6}$$

As a result, the computational complexity is reduced from $O(h \times w)$ to $O(r \cdot (h + w))$.

Combining the above two solutions for complexity reduction, the approximation of the decision rule can be defined by

$$f'(X) = sign\left(\sum_{j=1}^{m'} \beta_j k(U_{j,r} S_{j,r} V_{j,r}^T, X) + b \right). \tag{7}$$

Therefore m' and r allow the system to find a balance between efficiency and accuracy. In the experiments, we evaluate the performance of face detection by changing the number of separable filters, i.e., to determine the effect of changing the value of r. Figure 2 shows an example of face detection using this approach. In the figure, most of the faces can be detected correctly, but there are some false positives. Removing false positives from the detected frontal face set is necessary since the detected results (no matter whether they are correct or incorrect) will be further analyzed to compute the number of viewers. Since the subsequent face recognition module will consume a great deal of computational power, a pre-processing step to filter out false positive faces is very important. In the next section, we describe an effective approach for removing false positives from the set of detected face candidates.

Fig.2. Demonstration of face detection using an SVM-based approach [4]

2.2 Filtering False Positive Faces

Face detection has attracted a great deal of attention in the past decade. Well-developed face detection techniques, such as OpenCV [10] and Kienzle et al. [4], can achieve success rates of 80%-90%. However, their detection rate of false positives is in the 10%-20% range. In our system, we adopt Kienzle et al.'s approach to detect potential face candidates for our task. To distinguish between a real face and non-faces, we need to train the system to know what a "real face" looks like. In the follow sub-sections, we first explain how to use a method based on principal component analysis (PCA) to better represent a face and a non-face, and then describe the verification process.

A. Training Process

In the training process, the face detection module extracts face-like images from test videos. For our experiments, we randomly selected 200 face images and 200 non-face images as training images. The images were then scaled down to 64×64 pixels, and the intensity was normalized by

$$A'_{i,j} = \frac{A_{i,j} - \mu_{i,j}}{\sigma_{i,j}}, \quad i \in \{\text{F, NF}\}, \tag{8}$$

where the suffix i denotes a face class (F) or a non-face class (NF); $A_{i,j}$ and $A'_{i,j}$ represent the j-th original image and the j-th normalized image, respectively; and $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation of $A_{i,j}$, respectively. The goal of performing intensity normalization is to reduce the error caused by varying lighting conditions and backgrounds [11]. The average images ($\bar{A}_i$) of a face class and a non-face class are calculated according to

$$\overline{A}_i = \frac{1}{M}\sum_{j=1}^{M} A'_{i,j}, \quad i \in \{\mathrm{F}, \mathrm{NF}\}, \tag{9}$$

where M=200. We then apply PCA to calculate the set of eigenfaces corresponding to the training data. Related works on eigenfaces can be found in [11-14]. Using the source code in [11] to calculate the eigenfaces directly, we obtain 200 eigenfaces (ϕ_F) and 200 non-eigenfaces (ϕ_{NF}). The weighting vectors of all the training images are measured by

$$\begin{aligned} \omega_{i,j,k} &= \left(A'_{i,j} - \overline{A}_i\right)\cdot \phi_{i,k}, \\ \Omega_{i,j} &= [\omega_{i,j,1}\ \omega_{i,j,2} \cdots \omega_{i,j,N}]^{\mathrm{T}}, \quad i \in \{\mathrm{F}, \mathrm{NF}\}, \end{aligned} \tag{10}$$

where $\phi_{i,k}$ denotes the k-th eigen-image, and $\omega_{i,j,k}$ and $\Omega_{i,j}$ represent the k-th weighting value and the weighting vector of $A_{i,j}$, respectively. Every weighting vector consists of 200 weighting values; hence, N=200.

B. Verification Process

The verification process proceeds as follows. Each examined image is scaled down to 64×64, and the intensity is normalized by applying Eq.(8). According to Eq.(10), to measure two weighting vectors of the images Ω_{F} and Ω_{NF} for face and non-face classes, respectively, we calculate the Euclidean distances between Ω_{F} and $\Omega_{\mathrm{F},j}$ and select the minimum distance (ε_{F}). Similarly, the minimum distance ($\varepsilon_{\mathrm{NF}}$) is computed from the Euclidean distances between Ω_{NF} and $\Omega_{\mathrm{NF},j}$. The minimum distance can be derived by

$$j^* = \arg\min_{j\in\{1,2,\cdots,M\}} \left\|\Omega_i - \Omega_{i,j}\right\|^2,$$
$$\varepsilon_i = \left\|\Omega_i - \Omega_{i,j^*}\right\|^2, \quad i \in \{\mathrm{F}, \mathrm{NF}\}. \tag{11}$$

If ε_{F} is smaller than $\varepsilon_{\mathrm{NF}}$, the image under examination is a face; otherwise, it is a non-face.

3. Feature Extraction

When counting the number of people actually watching an advertisement on a TV billboard, some strict criteria need to be satisfied. First, one has to make sure the people are really "watching." Second, some people may like the advertisement very much and watch it for a long while, but they cannot be double counted. In these circumstances, the extracted faces should have a frontal orientation and the chosen feature set should have strong discriminative power so that correct assessments can be made. Since each person only has a few distinct facial features (eg., nose, mouth and eyes), we propose extracting a separate set of features from part of the torso so that the complete feature set will contain richer information. As people usually adopt a frontal orientation when watching a public TV advertisement, we extract features from the front of each individual's torso. The features extracted are the shape context [7] and the kernel weighted region saliency. We describe these features in Section 3.1 and Section 3.2, respectively.

3.1 Shape Context [7]

The shape context descriptor for a point on a shape is the histogram of relative polar coordinates of all other points on the shape. Basically, this descriptor provides global discrimination. The corresponding points on two similar shapes usually have similar shape contexts. This characteristic enables us to solve the shape correspondence problem as an optimal assignment problem. Point correspondences between two shapes are thus established by minimizing the point matching costs, i.e., the x^2 test statistic for histograms. Global optimal correspondences can be found by minimizing the sum of the individual matching

errors. The above-mentioned correspondence matching problem can be solved by a bipartite graph matching algorithm that enforces a one-to-one point matching process. Therefore, the shape distance, D, [7] is estimated as the weighted sum of the image appearance distance D_{ac}, the shape context distance D_{sc}, and the bending energy D_{be} as follows:

$$D = w_1 D_{ac} + w_2 D_{sc} + w_3 D_{be}\,, \tag{12}$$

where w_i denotes the weighting of its corresponding distance. D_{ac} is the appearance cost, defined as the sum of squared brightness differences in Gaussian windows around corresponding image points:

$$D_{ac}(P,Q) = \frac{1}{n}\sum_{i=1}^{n}\sum_{\Delta\in Z^2} G(\Delta)\left[I_P(p_i+\Delta) - I_Q\left(T\left(q_{\pi(i)}\right)+\Delta\right)\right]^2\,, \tag{13}$$

where I_P and I_Q are the gray-level images corresponding to P and Q, respectively; Δ denotes some differential vector offset; G is a windowing function, which is usually a Gaussian and $\{P_i\}, i \in [1,n]$ is a point set of P. The distance is computed after the thin plate spline (TPS) transformation T has been applied to warp the images into alignment as much as possible; and $\pi(i)$ is the permutation of points *q(i)* of Q resulting from minimizing the costs of all pairs of points of P and Q.

D_{sc} is used to measure the shape context distance between shapes P and Q as the symmetric sum of the shape context matching costs over the best matching points, i.e.,

$$D_{sc}(P,Q) = \frac{1}{n}\sum_{p\in P}\arg\min_{q\in Q} C(p,T(q)) + \frac{1}{r}\sum_{q\in Q}\arg\min_{p\in P} C(p,T(q)), \tag{14}$$

where

$$C\left(p_i,q_j\right) = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_i(k)-h_j(k)\right]^2}{h_i(k)+h_j(k)}\,. \tag{15}$$

$h_i(k)$ and $h_j(k)$ denote the K-bin normalized histogram at p_i and q_j, respectively. The distance of the bending energy D_{be} corresponds to the minimal amount of transformation needed to

align the shapes *P* and *Q;* thus, it is equivalent to minimizing the bending energy I_f

$$I_f = \iint_{\Re^2} \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right] d_x d_y \,, \tag{16}$$

where

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} \eta_i U\left(\left\| (x_i, y_i) - (x, y) \right\| \right),$$

and the kernel function $U(s)=s^2 \log s^2$ and $U(0)=0$. η_i is a weighting for the point (x_i,y_i) and $\|\cdot\|$ denotes the 2-norm operation. The detailed derivation of the above method can be found in [8].

We use the shape context to match the shapes of parts of the torso, as shown in Fig. 3. Figs. 3(a) and 3(b) are the same part of a person's torso taken automatically at different time instants. Although the shapes are similar, there are some subtle differences. Compared to Fig.3(a), Fig.3(b) is blurred and has a translation to up-right. After proper bending by minimizing D_{be}, we obtain the image in Fig.3(d), which is the warped shape of Fig.3(a). The shapes in Fig.3(d) and Fig.3(b) match well, as shown in Fig.3(c). In this case, the person shown in Fig.3(a) and Fig.3(b) would be considered the same person.

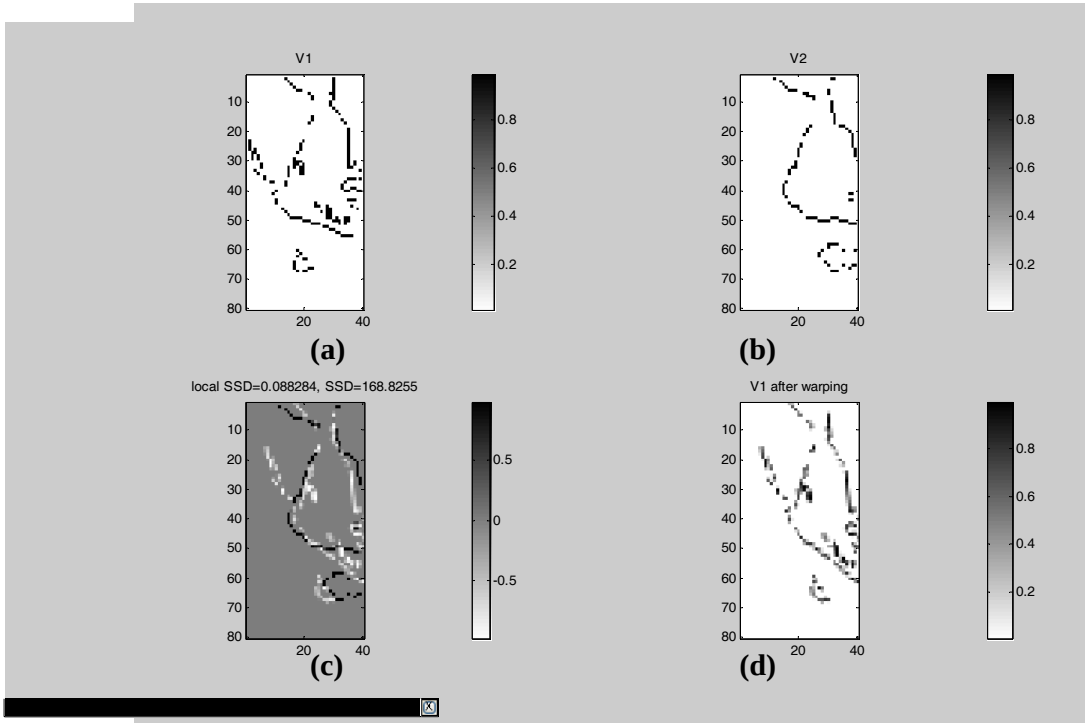

Fig. ces between two shapes are found by minimizing the point matching costs. (a) and (b): the same part of the torso taken at different time instants; (c) shape matching using the shape of (b) and the warped version in (d).

3.2 Kernel Weighted Region Saliency

In addition to the edge-based shape context feature, a region-based color feature is extracted by computing both the global region saliency and the local color information. We propose a technique that performs template matching by matching the salient regions between distinct templates. We use the kernel weighted region saliency to generate a compact template signature. The process for deriving the kernel weighted region saliency is illustrated in Fig. 4. First, the original image I is thresholded to a binary image by using a non-parametric and unsupervised method of automatic threshold selection [9]. An optimal threshold is selected by maximizing the discriminant measure of separability of the resultant gray level histogram. The resultant binary image is shown in Fig. 4(b). We then apply the Euclidean distance transform on the binary image as follows:

$$\tilde{I} = \min_{p(x,y)\in P} \left\{ (x-i)^2 + (y-j)^2 \right\}, \tag{17}$$

where P is the region with intensity value 0 and pixel (i,j) belongs to the region with intensity value 1. Fig. 4(c) shows the transformed image $\tilde{I}$ after applying the Euclidean distance transform. The salient regions are the areas with binary intensity value 1; thus, we compute a new image whose pixel value is inversely proportional to that of $\tilde{I}$ by

$$\hat{I}(x,y) = e^{-(\tilde{I}(x,y))} \cdot I(x,y), \tag{18}$$

where $I(x,y)$ is the original pixel intensity of image I. It is used as a weighting function to characterize the color information. The resultant new image, $\hat{I}$, i.e., the kernel weighted region saliency, is illustrated in Fig. 4(d). In this figure, the salient regions in the template are well localized. The examples in Figs. 4(e)-4(g) are salient regions extracted from different individuals. Clearly, these regions provide sufficient information to make distinctions. Figure 5 shows a pair of processed templates. The image in Fig.5(b) is blurred and the human subject is translated and scaled from Fig.5(a). However, the resulting kernel weighted salient regions of the image pair are still similar, which indicates that the proposed feature has good discriminating power based on color regions.

To match shapes by using salient regions, a distance measure d_{rs} is defined as

$$d_{rs} = \sum_{i=1}^{b} \left\{ \left| \hat{I}_p(x,y) - \hat{I}_q(x,y) \right| \ (x,y) \in B_i \right\}, \tag{19}$$

where $\hat{I}_p$ and $\hat{I}_q$ are the templates obtained after applying Eq.(18). B_i is the block obtained by first normalizing a template to a pre-defined size and then partitioning it into b blocks of equal size. By using the distance measure, the difference in the spatial relationships of region saliency between a pair of templates, i.e., $\hat{I}_p$ and $\hat{I}_q$, can be computed. In addition, by controlling the parameter b, we can adjust the degree of tolerance in translation and scaling.

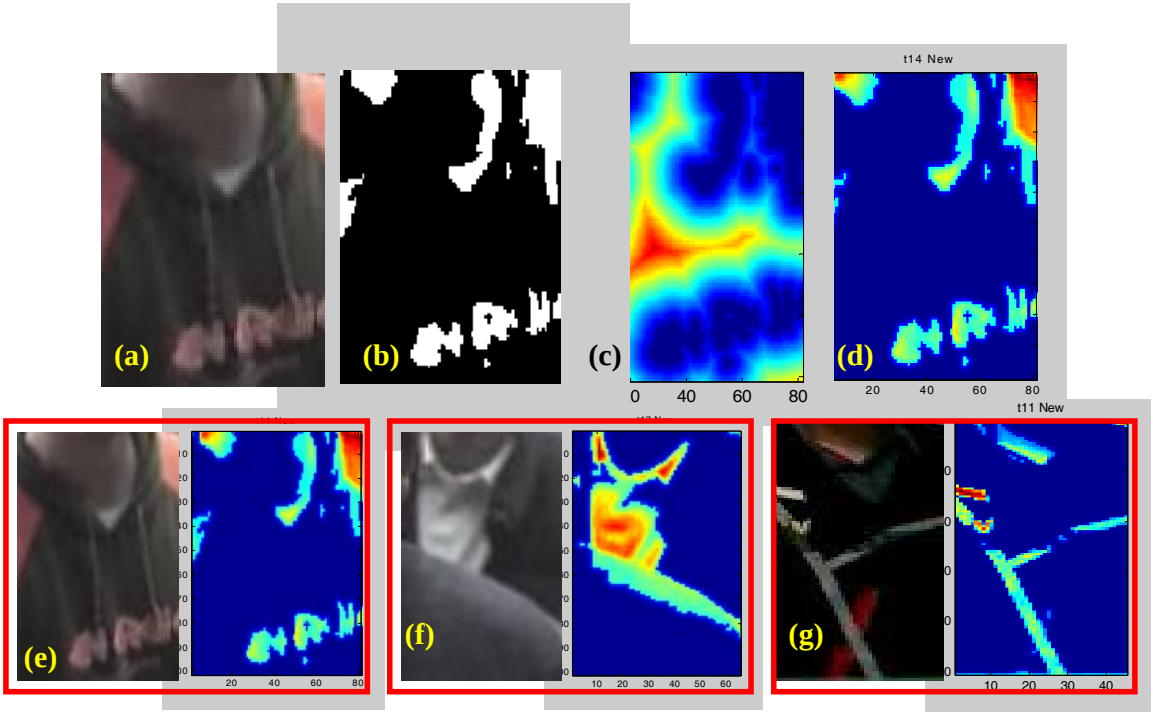

Fig. 4. Demonstration of the proposed kernel weighted region saliency: (a) original image; (b) an adaptively binarized image; (c) the Euclidean distance transform of (b); (d) the resulting kernel weighted region saliency. (e)-(g) the salient regions possess effective discriminative abilities.

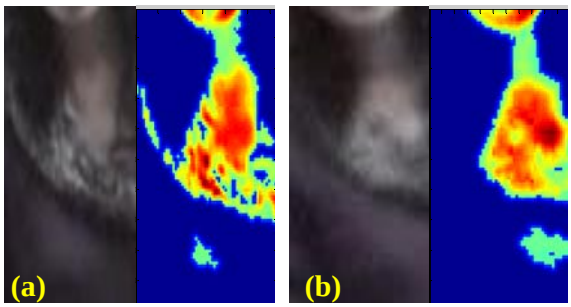

Fig. 5. Salient regions detected in the same person are similar, even though they were taken at different time instants.

4. Online Boosted People Counting

An intrinsic characteristic of a video-based people counting system is that the pose of a human target inevitably changes over time. Most existing methods need to address the pose change problem before performing template matching, i.e., they must pre-define a set of thresholds to ensure good functionality of the feature set. However, it is extremely difficult to find an appropriate feature set that can fit all changes. For example, in Fig. 6, even the same

person filmed at different time instants will exhibit some appearance changes. To resolve this problem, we propose a template matching algorithm that has online appearance learning ability. We describe the learning algorithm in Section 4.1, and then discuss the corresponding database that supports the online learning mechanism in Section 4.2.

Fig. 6. Even the same person filmed at different time instants will exhibit some changes in appearance.

4.1 Online Learning Algorithm

We use Fisher's Linear Discriminant criterion [15] to determine a projection orientation so that the two classes will be properly separated. Under Fisher's criterion, each element of a feature vector is viewed as a coordinate in a high-dimensional feature space. A linear projection based on the criterion is used to project a torso template from the original high-dimensional space to a new feature space with much smaller dimensions. In the new feature space, the ratio of the determinant of the between-class scatter to that of the within-class scatter is maximized.

For a template, suppose we have a set of n ($n=n_1+n_2$) d-dimensional feature vectors $f_1,\ldots f_n$, which consist of n_1 positives of F_1 and n_2 negatives of F_2. If we form a linear combination of the components h_i, we can obtain the scalar dot product by

$$z_i = v^t f_i, \tag{20}$$

and a corresponding set of n projected points $z_1,\ldots,z_n$ divided into two subsets, Z_1 and Z_2. Geometrically, if $||v|| = 1$, each z_i is the projection of the corresponding h_i onto a line in the direction of v. The Fisher Linear Discriminant process employs the linear function shown in

Eq. (20) for which the criterion function

$$J(v) = \frac{|\mu_1 - \mu_2|^2}{s_1^2 + s_2^2} \tag{21}$$

is maximized. The v maximizing $J(\cdot)$ yields the best separation between the two projected sets. Here, μ_i is the mean of the projected feature vector (Z_i) of set F_i; and s_i^2, the scatter of the projected feature vectors, is defined by

$$s_i^2 = \sum_{z \in Z_i} (z - \mu_i)^2, i = \{1,2\}. \tag{22}$$

Thus, $(1/n)(s_1^2 + s_2^2)$ is an estimate of the variance of all the feature vectors, and $s_1^2 + s_2^2$ is the total within-class scatter of the projected samples.

According to the generalized Rayleigh Quotient [16], the objective function $J(\cdot)$ in Eq.(21) can be defined as

$$J(v) = \frac{v^t S_B v}{v^t S_V v}, \tag{23}$$

where S_V is the within-class scatter matrix defined by

$$S_V = S_1 + S_2, \qquad S_i = \sum_{f \in F_i} (f - \mu_i)(f - \mu_i)^t, \tag{24}$$

and S_B is the between-class scatter matrix defined by

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t. \tag{25}$$

Therefore, we can obtain the solution for v that optimizes $J(\cdot)$ by

$$v = S_V^{-1}(\mu_1 - \mu_2). \tag{26}$$

Fig.7 illustrates the principle of Fisher's criterion. The accumulated positive and negative samples captured by our system over time can be well separated by projecting their high-dimensional feature vectors on to an optimized projection line.

In our approach, the templates collected from the same person are labeled positive samples

and those collected from other subjects are labeled negative samples. Thus, a one-against-all binary classifier is learned online for each person (class) when he/she appears in the field of view. When a new template x appears, it is first examined by all classifiers to determine if it belongs to one (or more than one) of them. If x belongs to more than one class, its class $C(i)$ is determined by

$$C(i) = \arg\max_i v_i x^t + b_i , \tag{27}$$

where b_i is the bias of the ith classifier. However, if x does not belong to any of the classes, a new class is formed. In the training process, it is essential that a template database be maintained for training classifiers. In the next section, we explain how this database is maintained.

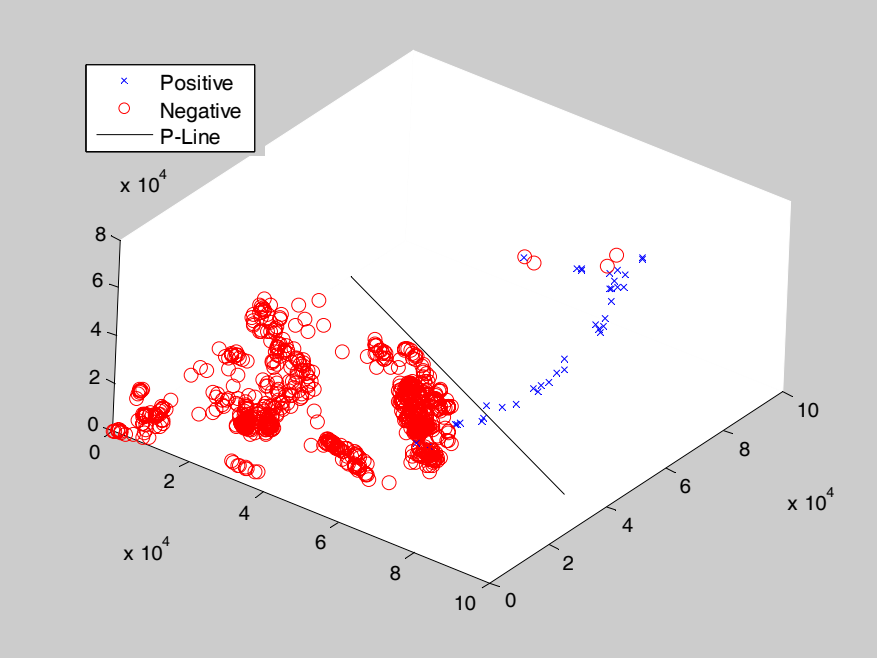

Fig. 7. Classification of input data using Fisher's Linear Discriminant function: a high-dimensional feature vector combined with shape contexts and kernel weighted region saliency is projected on to the projection line to separate positive and negative samples.

4.2 Maintenance of the Dynamic Template Database

To maintain the dynamic template database (tDB), we develop an update rule. Since the processing time is one of the major concerns of this system, the template database cannot be

too large; otherwise, the template matching process will require too much time. We assume the maximum size of the template database is equal to the maximum number of people that can stand in a camera's field of view. For the data structure of the database, we use the first-in first-out queue mechanism. Therefore, if someone stands in front of a camera for an extended period, he/she will not be counted repeatedly. On the other hand, if someone leaves the field of view for a short time such that and his/her old template has been removed from the queue, then he/she will be counted again. Fig. 8 shows the algorithm devised for handling the proposed dynamic template database. In the algorithm, the process *Match*($O_{i,t}$,O_j) is an important step because it determines if a template $O_{i,t}$ is reliable enough to be selected as a training sample for the same person. The similarity between two templates is measured by

$$\eta\left(O_{i,t}, O_j\right) = e^{-(\alpha_1 \cdot D_s + \alpha_2 \cdot D_t)}, \tag{28}$$

where α_i is the weight for its i^{th} corresponding distance. D_s is the spatial distance defined by $D_s = \beta_1 \cdot D + \beta_2 \cdot d_{rs}$, which is a linear combination of the distance of the shape context D and the kernel weighted region saliency d_{rs} with weights β_1 and β_2, respectively. The weighting sets α and β are determined empirically based on extensive experiments; and D_t is the time interval between $O_{i,t}$ and O_j. If $\eta(\cdot,\cdot)$ is larger than a pre-defined threshold, a new template, i.e., a new human subject, is found. We then add it to the queue if there is room. However, if the queue is full, the template that has stayed for the longest time must be removed. In addition, a post-processing step is proposed to prevent counting noise, which occurs when a candidate appears and then suddenly disappears from the camera's view. A distance D_N is devised to measure whether a candidate causes noise, where

$$D_N = \frac{1}{|O_i|} \sum_s \left(\tau_{O_{i,t}} - \tau_{O_{i,s}}\right). \tag{29}$$

Here, τ is the instant that the template O_i appears; and | O_i | is a counter that counts how many times O_i has appeared already. By using the updating rule, we can maintain the

dynamic database efficiently.

Template Database Updating Rule

Input

N_{TDB}: the size of Template DB (tDB) is pre-defined
/*It can be adjusted/trained based on how large the camera's field of view is.*/

$\{O_{i,t}\}$: detected faces in the current frame t

Output

Number of People: *PeopleCounted*

Algorithm

```
if Match( O_{i,t}, O_j ), then update the time flag T_{Oj} of O_j (T_{Oj} =t)
                                   /* {O_j} is the set of templates in tDB */
else                               /* a new person is detected*/
    if |tDB|< N_{tDB}
        Push O_{i,t} to tDB;
    else                           /* tDB is full. Remove the template that
                                       has stayed
                                       longest */
        Remove the O_j with the smallest time flag T_{Oj} from tDB;
    Push O_{i,t} to tDB;
    end
    PeopleCounted = PeopleCounted + 1;
```

Fig. 8. The algorithm for maintaining the dynamic template database

5. Experiment Results

We used an empirical method to determine the number of separable filters that should be used in face detection. In our experiment, we used between 1 and 5 filters. Fig. 9 shows the performance of the face detector in terms of precision and recall when different numbers of separable filters were used. Considering the tradeoff between the precision and recall rates, it is clear that the best performance was achieved when both precision and recall were 87% and the number of separable filters was 3. Examples of face detection results using different settings of r are shown in Fig. 10.

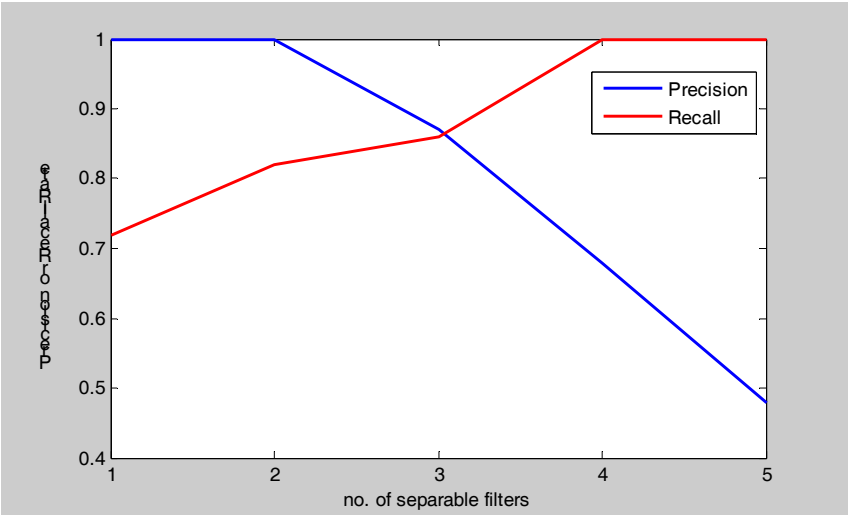

Fig. 9. Determining the number of separable filters in the face detector by evaluating the precision and recall rates

(a) $r = 1$ (b) $r = 2$

(c) $r = 3$ (d) $r = 4$

Fig. 10. Faces detected using different numbers of separable filters.

To evaluate the performance of the face detector, 5,071 candidates were detected from the test video, which contained 4,538 faces and 533 non-faces. In the experiment, we used different videos for training and testing. In addition, we applied a PCA-based method to filter out non-faces that were accidentally included by the face detection process. The performance

results of the face filtering process are listed in Table 1. The proposed method correctly recognized 4,430 faces and 477 non-faces among the 5071 test images. The success rate was 96.77%, which was higher than the initial success rate of 89.49%. (Originally, only 4,583 faces and non-faces were correctly recognized, as the proposed face filtering process was not applied). On the other hand, 108 faces and 56 non-faces were falsely recognized. This result is also better than the initial results. (Originally, the false detection rate was 10.51%, and the new false detection rate was 3.23%). The face filtering process successfully removes the majority of false positives and significantly reduces the computation time required by the subsequent face recognition process.

Table 1. The results of face filtering with the proposed method

		Face	Non-face
Test	Face	4430	56
	Non-face	108	477
Total		4538	533

Fig. 11. A test dataset containing several people with varied appearances and frequent occlusions

To evaluate the overall performance of the proposed people counting system, we used a long test video that contained many events. Fig. 11 shows some frames of test video. In the test video, a large number of human subjects moved frequently in the field of view, and mutual occlusions between the subjects occurred frequently. We implemented the proposed system using Matlab 7.0 with a 1.83GHz Intel CPU. The frame rate of the video was approximately

3-5 fps. Figure 12 shows a snapshot of the people counting process. In the figure, the horizontal axis represents the frame number of the test dataset, and the vertical axis represents the accumulated number of people counts. The blue, red, and green curves indicate the counts without post-processing, the counts with post-processing, and the ground truth, respectively. Without using post-processing to filter out noise, it is clear that the accumulated number of people counts would vary abruptly with a sudden increase and decrease of counts. For instance, in the interval between frame 300 and 400, two peaks appear almost consecutively. While applying the post-processing step, the accumulated number of people counts increased smoothly without abrupt variations because candidates that only appeared for a short time were labeled as noise and filtered out. Therefore, our system executes the people counting task in a stable and accurate manner.

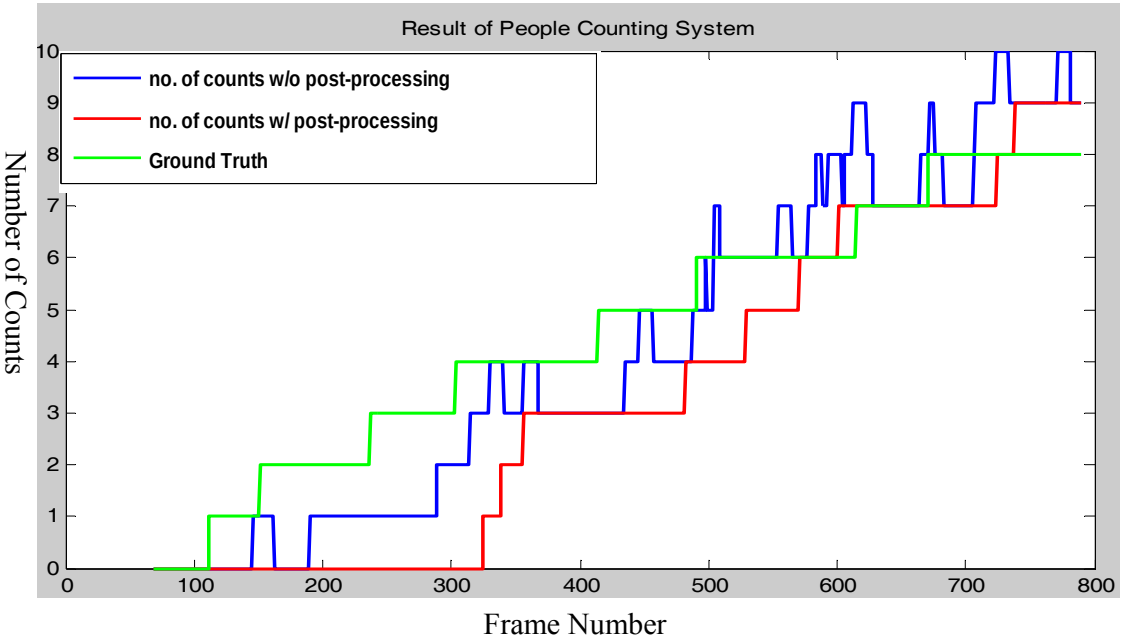

Fig. 12. Demonstration of the proposed people counting mechanism

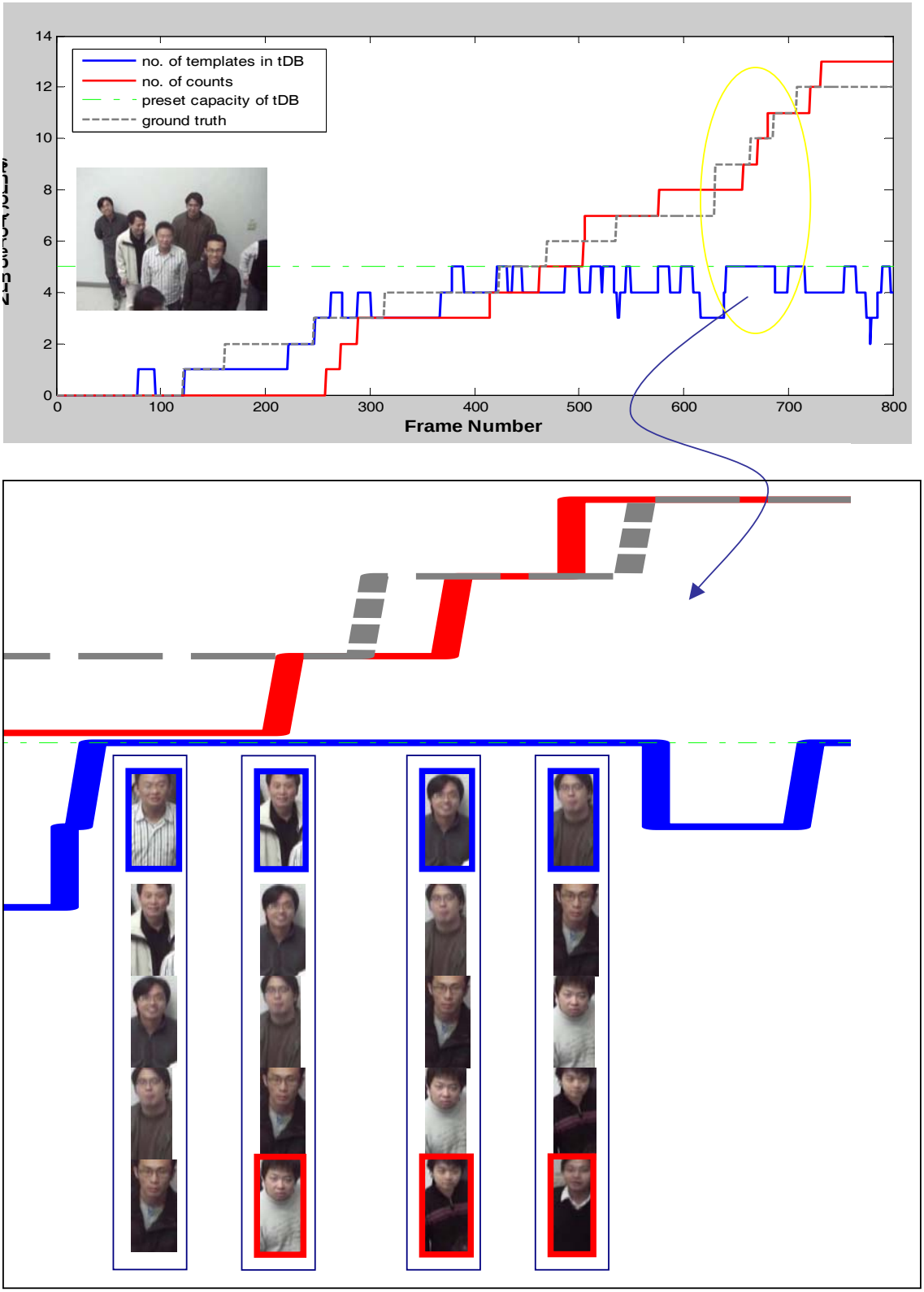

Fig. 13. Demonstration of the process for maintaining the dynamic template database. The templates enclosed by blue bounding boxes are at the top of tDB and will therefore be removed if space is required for new templates.

Fig.13 shows how the dynamic template database is updated during the people counting process. In this experiment, we assume that the maximum number of people that can stand in the field of view is five (the preset capacity of the queue indicated by the green dotted line). It is clear that the number of templates (the blue solid line) in the queue increases when a new template is detected and decreases when a template is regarded as noise. Moreover, at the bottom of Fig.13, a snapshot of the states of the queue illustrates how the push and pull operation works. A template with a blue bounding box indicates that it is at the head of the queue and will be removed if space is required for a new template (in a red bounding box). In this test video, 8 human subjects moved frequently in the field of view and 4 of them left the field for a short time and then returned. Therefore, the ground truth of the total number of counts should be 12. Our system returned a result of 13 because one of the subjects changed his appearance by removing his coat. The system was affected by this action because it relies on the color and shape of the torso. We ran simulations (using videos) for a large amount of test data, and the average successful detection rate was close to 90%.

Fig. 14 shows the GUI interface of the proposed people counting system. The accumulated number of people counts is shown on the left-hand side. For statistical purposes, the "watching time" of each person is illustrated with a bar chart at the bottom (left-hand side) of the GUI. On average, the precision of the people counting system was about 89% (verified by a manual counting process). In addition, our system was set up at an exhibition conference site (SecuTech 2008) for three days and the accumulated number of people counts was 187 (April 16, 2008), 206 (April 17, 2008) and 163 (April 18, 2008). The system ran real-time demonstrations and the ground truth was counted manually by two individuals working independently (8 hours per day). The average success rate was close to 90% over the 3-day period. The environment setting and system demonstration of our people counting system at SecuTech are shown in Fig. 15.

Fig. 14. The GUI interface of our people counting system

Fig. 15. System demonstration and exhibition environment setting at the SecuTech Expo 2008

6. Conclusion

We have proposed an online boosted people counting system in which an efficient face detector combined with our proposed face filter is employed for the subsequent real-time application. In addition, we developed a new feature set and use it to recognize human subjects in the camera's field of view. To achieve robust recognition of people, we employ an online boosted classifier trained by using Fisher's Linear Discriminant (FLD) criterion. In this study, we applied the proposed scheme to people watching a TV-wall advertisement, and showed that the people counting task can be performed effectively by recognizing part of the face and part of the torso. Our experiment results demonstrate that the proposed system can achieve an 89% success rate in real-time.

References

[1] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for Cooperative

Multisensor Surveillance," *Proc. of IEEE*, Vol. 89, No.10, pp. 1456-1477, Oct. 2001.

[2] D. Roqueiro and V. A. Petrushin, "Counting People using Video Cameras**,"** *MDM/KDD'06*, August 20, 2006.

[3] A. B. Chan, Z. S. Liang, and N. Vasconcelos, "Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking," Proc. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008.

[4] W. Kienzle, G. Bakir, M. Franz and B. Scholkopf, "Face Detection - Efficient and Rank Deficient," *Advances in Neural Information Processing Systems*, Vol. 17, pp. 673-680, 2005.

[5] C. J. C. Burges, "Simplified support vector decision rules," Proc. of *International Conference on Machine Learning*, pages 71–77, 1996.

[6] P. Viola, and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, 2004.

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 24, pp. 509-522, April 2002.

[8] F. L. Bookstein, "Principal Warps: Thin-Plate Splines and Decomposition of Deformations," *IEEE Transactions on Pattern Analysis and Machine Learning*, Vol. 11, No. 6, pp. 567-585, June 1989.

[9] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, Jan. 1979.

[10] Open source computer vision library, http://www.intel.com/technology/ computing/ opencv/.

[11] Eigenface tutorial, http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm.

[12] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for The Characterization of

Human Faces," *Journal of the Optical Society of America A*, vol.4, no.3, pp.519–524, March 1987.

[13] M. Kirby and L. Sirovich, "Application of The Karhunen-Loeve Procedure for The Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, no.1, Jan. 1990.

[14] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," Proc. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp.586-591, June 1991.

[15] R. A. Fisher, "The Use Multiple Measures in Taxonomic Problems," *Annals of Eugenics*, Vol. 7, pp.179-188, 1936.

[16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, John Wiley and Sons, Inc., 2001.

[17] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No.8, August 2000.

[18] P.Viola, M.J. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *International Conference on Computer Vision*, 2003.

[19] S.-Y. Cho, T. W. S. Chow, and C.-T. Leung, "A Neural-Based Crowd Estimation by Hybrid Global Learning Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 29, No. 4, August 1999.

[20] D. Kong, D. Gray, and H. Tao, "Counting Pedestrians in Crowds Using Viewpoint Invariant Training," *British Machine Vision Conference*, 2005.

[21] R. Ma, L. Li, W. Huang, and Q. Tian, "On Pixel Count Based Crowd Density Estimation for Visual Surveillance," Proc. *Conference on Cybernetics and Intelligent Systems*, Singapore, 1-3 December 2004.

[22] N. Paragios, and V. Ramesh, "A MRF-based Approach for Real-Time Subway

monitoring," *Computer Vision and Pattern Recognition*, 2001.

[23] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin, "Estimating Crowd Density with Mikowski Fractal Dimension," *International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[24] C. S. Regazzoni, A. Tesei, "Distributed data fusion for real-time crowding estimation," *Signal Processing*, Vol. 53, pp 47-63, 1996.

[25] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, Vol. 35, No. 4, pp. 399 – 458, Dec. 2003.

[26] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 349-361, April 2001.

Acknowledgements

This work is supported in part by the Ministry of Economic Affairs under Contract No. 96-EC-17-A-02-S1-032, and the National Digital Archives Program under Contract No. NSC 96-2422-H-001-001.