A REAL-TIME HYBRID METHOD FOR PEOPLE COUNTING SYSTEM IN A MULTI-STATE ENVIRONMENT

Ali Rezaee
Department of Engineering, Khayam University of Mashhad, Falhi street, Mashhad, Iran

Hojjat Bagherzadeh, Vahid Abrishami
Department of engineering, Azad University of Mashhad, Mashhad, Iran

Hamid Abrishami
Khorasan Science And Technology Park, Mashhad, Iran

Keywords: Detecting and tracking people, PCA, Human shaped objects, Feature extraction, Dynamic-VCM, Crowd estimation, Solitude scenes.

Abstract: Detecting and tracking people in real-time in complicated and crowded scenes is a challenging problem. This paper presents a multi-cue methodology to detect and track pedestrians in real-time in the entrance gates using stationary CCD cameras. The proposed approach is the combination of two main algorithms, the detecting and tracking for solitude situations and an estimation process for overcrowded scenes. In the former method, the detection component includes finding local maximums in foreground mask of Gaussian-Mixture and Ω-shaped objects in the edge map by trained PCA. And the tracking engine employs a Dynamic VCM with automated criteria based on the shape and size of detected human shaped entities. This new approach has several advantages. First, it uses a well-defined and robust feature space which includes polar and angular data. Furthermore due to its fast method to find human shaped objects in the scene, it's intrinsically suitable for real-time purposes. In addition, this approach verifies human formed objects based on PCA algorithm, which makes it robust in decreasing false positive cases. This novel approach has been implemented in a sacred place and the experimental results demonstrated the system's robustness under many difficult situations such as partial or full occlusions of pedestrians.

1 INTRODUCTION

Detecting and tracking of moving objects has been at the core of visual surveillance applications for both scientific and industrial research in the past few years. People tracking naturally play a key role in any visual surveillance system. Its application ranges from security monitoring, pedestrian counting, traffic and pedestrian control, detection of overcrowded situations in public buildings to tourist flow estimations.

People counting methods fall into two major categories, the first category is based on people detection and the second one uses feature-based algorithms.

The former implies the detection of pedestrians in order to count them. For instance, Goncalo Monteiro et al. (Monteiro and Peixoto, 2006) use Haar-like features and AdaBoost algorithm to discriminate pedestrians which can work in a real time manner; however, as a result of only considering spatial filters it has a high false positive rate. Viola et al. (Viola and Jones, 2003) compute both spatial and temporal rectangle filters by efficiently using the integral image technique, and model a pedestrian classifier using a variant of AdaBoost. The system is trained based on full human figures and due to this fact the system needs to have the whole body figure to work properly, therefore it has problems in detection of occluded or partial human figures. Ken Tabb et al. (Tabb and

Rezaee A., Bagherzadeh H., Abrishami V. and Abrishami H. (2010).
A REAL-TIME HYBRID METHOD FOR PEOPLE COUNTING SYSTEM IN A MULTI-STATE ENVIRONMENT.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 119-126
DOI: 10.5220/0002734601190126

K.a.D., 2004) proposed a method which employs Active Contour Models (ACM) to detect moving objects and neural networks to classify the shapes obtained by ACM as either 'human' or 'non-human'. This technique is in contrast with other methods of shape description which relies on having a one to

Figure 1: A sample input frame and the output results of different steps of algorithm.

one correspondence between landmark points on the shape model and the current contour, and still suffers from occlusion. A.Koschan (Koschan,S.K.K, 2002) uses Active Shape Models (ASM) as human shaped objects detector; in addition the colour information contributes to the solution of occlusions. Nevertheless, the tracking of a person becomes rather difficult if the image sequence contains several moving persons with similar shape and the task may fail if the person is partially occluded. These approaches seem to fail in situations where people walk next to each other and/or occlude one another; however, Zhao and Nevatia (Zhao and Nevatia, 2001) employ Markov chain Monte Carlo technique as a method for finding the omega pattern, formed by the head and shoulders, which can overcome the occlusion problem but the complexity of MCMC method is an obstacle against working in a real time manner.

The second category uses the image processing statistical methods instead of detecting people for the counting task. These methods apply different features of objects which can be the blob size (Masoud and Papanikolopoulos, 2001) , (kong, Gray and Hai, 2006), (Aik and Zainuddin, 2009), the Fractal Dimension (Rahmalan, Nixon and Carter, 2006), the bounding box (Masoud and Papanikopoulos, 2001), and also edge density (kong, Gray and Hai, 2006), (Villamizar and Sanfeliu, 2009). These methods can be employed for real time application but they have lower accuracy than the methods in the first category.

In this paper we explorer an alternative technique based on a novel integration of multiple hypotheses for the detecting and tracking of human head-shoulder regions in order to count them in entrance gates which brings this method into the first category. In addition, for the crowd situations, we employ an estimation method which uses spatial features i.e. blob size, edge density and orientation, which places this component into the second one. The algorithm does not produce unique trajectories, but we show that after a one-time estimation of a systematic correction factor based on manually labelled ground truth data, accuracies up to 99 % can be achieved for real-world scenarios. A snapshot of our results is shown in Fig. 1.

The outline of this paper is as follows. Section 2 first gives a brief description of the system, in addition reviews the different algorithms and their role in this approach. We illustrate a detailed analysis of our real-world tests of the system in section 3. And finally we conclude the paper in Section 4.

2 SYSTEM DESCRIPTION

Most of the previous works, assume that pedestrians, regardless of their clothes and hairstyles, display a typical Ω-like shape which is formed by their heads and shoulders. But in some areas like the sacred places where religious people wear special clothes, other potential head candidates can come into account, namely in an Islamic place most of women wear a long black veil and clergymen wear a special hat which can result in different shape of heads, like O or Λ. Based on this fact, beside employing the Ω-like shapes extraction for finding heads, we also notice O-like and Λ-like shapes. An efficient feature vector for demonstrating the head shape features also have been developed.

Lots of accurate methods like Zhao et al. (Zhao and Nevatia, 2001) suffer from time complexity and do not fit into real time constraints. Since most of counting applications are needed to be real time, we apply a further pre-processing step and also a trained PCA in order to find heads while reducing the processing time.

In order to find heads, first a foreground map, based on Gaussian Mixture Models (GMM) (Stauffer and Grimson, 1999) is used to segment the objects from the background which can overcome the known problems of adaptive background models.

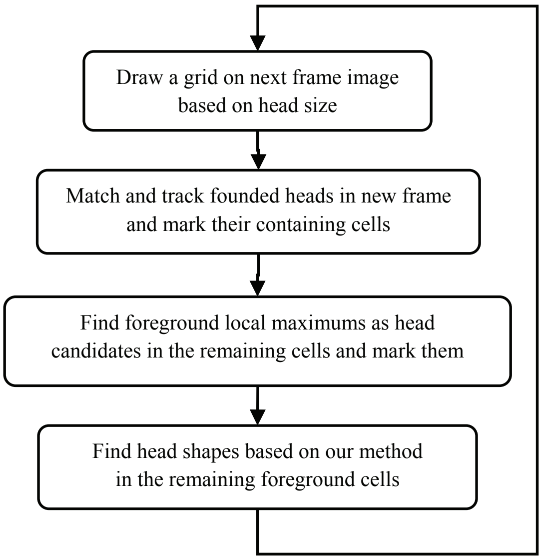


Figure 2: The block diagram of our algorithm.

Then a dynamic sized mask is applied to eliminatethe unwanted background cues. Next, to find the foreground local maximums as head candidates, we apply the Zhao et al. method (Zhao and Nevatia, 2001). Furthermore with our PCA based head shape detection algorithm on foreground pixels, we find heads.

Once the people are located in each individual image, it is necessary to track them across frames. This is achieved by using our robust implementation of Vector Coherence Mapping (VCM) (Quek and R.B.a.F, 2006) named Dynamic-VCM.

In overcrowded scenarios, counting people individually would be a time consuming process, and the accuracy would also decrease. These are two big deficiencies in a real-time application. Furthermore, in order to overcome this issue, the proposed system automatically determines whether the situation is a crowded one or a solitude one. In the former case, our method employs the head detection and tracking approach and in the latter, we utilize a crowd estimation algorithm.

The organization of our algorithm is as follow as shown in block diagram in Fig. 2:

1- Track the heads found from the last frame and mark their containing cells. Certainly if this is the first frame, we do not have any found heads.

2- After that, we find the foreground mask local maximums as head candidates in the remaining unmarked cells. We use the fast Zhao (Zhao and Nevatia, 2001) method in this step. If there are only walking persons in the scene, we are sure that the local maximums of foreground mask are heads of peoples, if not, we do not use this step.

3- Finally, detecting head shapes in the remaining cells. Our head detection is based on (Zhao and Nevatia, 2001) idea but it is completely different. We use different and more efficient feature vector, and also use fast PCA method, instead of slow HMM method.

2.1 Pre-processing

In order to find the head candidates, (Zhao and Nevatia, 2001) used a linear search method on the foreground pixels which can be really time consuming and also is an obstacle especially in real-time applications. In addition, as a mean to increase the performance of algorithm, we have added a pre-processing step which works as follow; Moreover, as the result of installing the camera in a specified angle, the captured images have a special angle and also the size of heads are nearly the same in the image.

First, we divide the image into a grid, and this grid division is in such a way that more than one head won't be able to fit in each cell at most. So in the searching method, if we find a head in a cell, we won't look for any other heads in that cell. Furthermore this head can be found by any of the methods namely tracking the previous frame, finding foreground local maximums, or the head shape detection as described below.

2.2 Finding Foreground Local Maximums

This method detects the heads which are on the boundary of the foreground (Zhao and Nevatia, 2001). As a result of putting cameras way above the ground, the head of a human has the least chance to be occluded and in addition the human is small; therefore this algorithm has a high detection rate. The basic idea is to find the local vertical peaks of the boundary. These peaks are being considered as

(a) (b)

Figure 3: The process of finding local maximums. (a) The result of method. (b) Foreground mask.

humans if they are within a region along the boundary whose size can be counted as the size of a human; however, the peaks which don’t have enough foreground pixels in that region will be eliminated. Flat peaks are also allowed in a way that the head top is the centre of the flat peak. This process has been shown in Fig. 3.

2.3 Feature Extraction

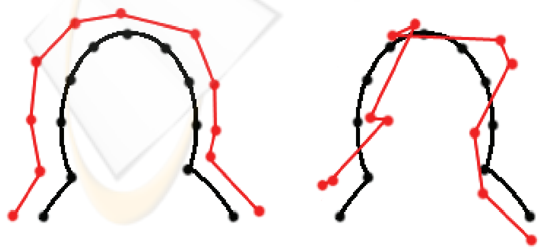
Experiences from previous works show that most of the methods as pedestrian detector which use features, are based on finding the Ω shaped objects as head or the shape of the whole human figure in the edge map image. In most crowded scenes, body occlusion can cause some parts of body not to be seen and makes its shape deficient. Nevertheless, the shape of human head is almost unique and tends to be like Ω in fact by setting up the camera in a good angle (near to vertical) we do not have any occlusion on heads.

Zhao features (Zhao and Nevatia, 2001) for head detection have been used by most of previous works. It first selects some good points on the contour of head and uses their positions and their normal vector angle as a feature vector model. It then checks if each point can be the top of a head by employing (1):

$$S(x,y) = \left(\frac{1}{k}\right)\sum_{i=1}^{k} e^{-\lambda D(\vec{m}_i)}\left(\vec{v_i}.\vec{O}\left(\vec{C}(\vec{m}_i)\right)\right) \quad (1)$$

where $\{\vec{m}_1, \dots, \vec{m}_k\}$ and $\{\vec{v}_1, \dots, \vec{v}_k\}$ are denoted as the positions and the normal units of the model points respectively, when the head top is at (x,y). A point can be considered as the top of a head if S is lower than a threshold.

The head detector (Rahmalan, Nixon and Carter, 2006) employs the distance transform in order to find the corresponding points. For each point of the model, if it exists on the edge map, the point itself will be considered as the point of model.



(a) (b)

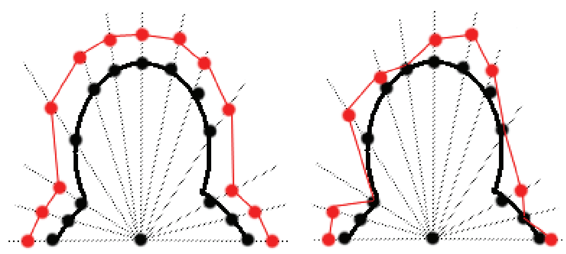
Figure 4: A well-defined and improper samples of the model point and its corresponding with the same values for S parameter by the equation (1).

On the other hand, the nearest point on the edge map will be counted as the point of model. Fig. 4.a illustrates a sample of well-defined corresponding points, in which the black points are that of the model, and the red points are the discovered corresponding points. Assuming that the angle of normal vector of each point is good enough, the calculated value for S is approximately around 0.8. Nevertheless, in real-life scenario the points are not as good as mentioned. Due to only considering the nearest point on the edge map and not taking into account their spatial position to each other, the obtained points may not have a great relation towards each other. As a result of not having a great quality edge map, there might be some missing edge points or existing a huge amount of accumulated edge points in a special place of the image.

Fig. 4.b shows an improper sample of these points. As it can be seen the obtained edge points have the same distance as Fig. 4.a, but they have no semantic relations. But if accidentally the angle of normal vector of points satisfies the criteria, which may occur a lot, the value of S in this image has the same value as that of in Fig. 4.a.

In the proposed method, in order to locate the position of corresponding points, we use a different method in which instead of using the top head as the coordinate point, we consider the centre of Ω-shaped model. In addition, our selected points are being chosen in a way that each point has a particular angle towards the reference point; moreover the angles between each two successive points are equal.

In order to find the corresponding points of the model point, these two conditions should be satisfied. First for each model point the corresponding point should have the same angle towards the reference point and second its distance should be equal or greater than the model point towards the reference point. In a situation which no



(a) (b)

Figure 5: A well-defined and improper samples of the model point and its corresponding based on our proposed method.

corresponding point can be found, the algorithm assumes that this part of edge has been eliminated and considers the model point as corresponding point. A limited number of missing edge point is permissible and exceeding this limitation results in rejection of labelling the corresponding point as the head object. The proposed feature extraction method makes the points relation semantically strength and decreases the probability of forming zigzag shapes which were the shortcomings of the Zhao's (Zhao and Nevatia, 2001) .Figure 5-a illustrates a set of well-defined points with a great value for S parameter and Figure 5-b shows the worst obtained set of points with a lower value. As it can be seen there are 4 missing points in figures b.

The S value can be determined by (2):

$$S(x,y) = \frac{1}{k}\sum_{n=1}^{k} e^{-\lambda(r(\vec{m}_i)-r(\vec{n}_i))}(\vec{O}(\vec{m}_i).\vec{O}(\vec{n}_i)) \quad (2)$$

where $\{\vec{m}_1, \ldots, \vec{m}_k\}$ are the model points, $\{\vec{n}_1, \ldots, \vec{n}_k\}$ are corresponding points, $\overrightarrow{O(x,y)}$ is the N-Vector of (x,y) and λ is a constant coefficient which is equal to 0.25. In order to calculate the n_i points, we simply perform a linear search on the m_i corresponding line.

2.4 Head Shape Detection

Head Shape detection algorithm is divided into two parts. A fast contour searching method which detects candidates for Ω-like shaped objects within the foreground mask; it is based on principles described by Zhao and Nevatia (Zhao and Nevatia, 2001). A pedestrian contour is represented as model consisting 19 points as described in 2.2. Then these head candidates have to be verified, in which the verifying method (Zhao and Nevatia, 2001) used a Hidden Markov Model which suffers from time complexity, and we used a fast method based on PCA (Principle Component Analysis) as follows.

In (Zhao and Nevatia, 2001), the basic theory is based on this fact that head and shoulder of pedestrian is Ω-shaped objects, but on the other hand, mostly in religious countries, some clergymen wear special hats which make their heads and shoulders to be O-like objects, and also most of women wear a black veil which makes them to be more Λ-like. Therefore, in according to the mentioned reasons, we employ three different model point datasets for every each one of Ω, O and Λ shaped objects as shown in Fig.6.

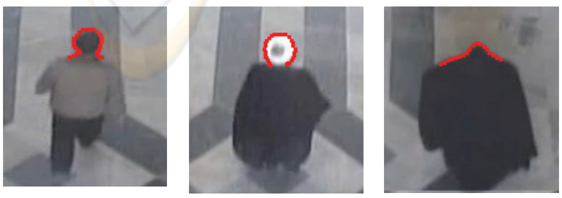


Figure 6: Three samples of Ω, O and Λ shaped objects.

First in order to train the PCA, we manually extract three set of head shape objects, and then for each head compute the feature vector as described in 2.2, and use them as our training data sets. And by training those extracted data sets, the PCA space for each data set is being made separately.

For each data set, all corresponding image patches are vectorized to form a data matrix X_{ij}, where each row of it is a vectorized image patch. ($X_{ij} = \{x_{ij1}, x_{ij2}, \ldots, x_{ijK}\}$ where k is the number of images)

For each X_{ij}, the covariance matrix, or the total scatter matrix, is computed using the equation (3):

$$\Sigma X_{ij} = \sum_{k=1}^{K}(x_{ijk} - \mu_{ij})(x_{ijk} - \mu_{ij})^T \text{ a} \qquad (3)$$

where μ is the sample mean of each patch. Applying Principal Component Analysis (PCA) on ΣX_{ij}, the eigenvalues of the covariance matrix are calculated and the first m largest eigenvalues, $\lambda_1 \ldots \lambda_m$ and their zassociated eigenvectors, $e_1 \ldots e_m$ are selected to produce a projection matrix for that region . Using the projection matrix, PCA projects the original image space of each region into a low-dimensional space, while preserving as much information as possible using much fewer coefficients.

For each training head image, the PCA projected coefficients of all its regions are concatenated to form the descriptor of the face. This descriptor can directly be used along with a distance metric in the recognition phase to identify the query faces.

(a) (b)

Figure 7: A snapshot of the input frame (a) and its head detection result (b).

2.5 Dynamic-VCM Tracking

The tracking process is based on Vector Coherence Map (VCM) (Quek and R.B.a.F, 2006), a correlation based algorithm that tracks iconic structures (in this case templates) in image sequence. The repetitive processing steps in our framework ensure that effectively we track tokens and not merely image regions. In our case, tokens are being considered the coherent clusters of motion vectors lying on the featured points of Omega.

Let $\mathcal{P}^t = \{p_i^t\}_{i=1}^N$ be the set of interest points detected in image I^t at time t. For a particular interest point p_i^t in image I^t, we can estimate its new position in image $I^{t+\delta t}$ by computing the correlation of the neighbourhood of p_i^t in $I^{t+\delta t}$. We use ADC (Agarwal & Sklansky 1992; Quek 1994; 1995) to perform the correlation. Hence the $\mathcal{N}$ is given by (4):

$$\mathcal{N}(p_i^t)[m,n] = \sum_{j=x_i^t-N}^{x_i^t+N} \sum_{k=y_i^t-N}^{y_i^t+N} \left| I^t(j,k) - I^{t+\delta t}(m+j, n+k) \right|$$

$$-D_x < m < D_x;\ -D_y < n < D_y \qquad (4)$$

where $p_i^t = (x_i^t, y_i^t)$, $2N+1$ is the size of the correlation template, and D_x, D_y define the maximal expected x and y displacements of p_i at $t + \delta t$ respectively.

We define the Vector Coherence Map (vcm) at p_i^t to be (5):

$$vcm(p_i^t) = \frac{1}{\sum_{j=1}^{|p^t|} W_i^t(p_i^t)} \sum_{j=1}^{|p^t|} \mathcal{N}(p_j^t) \times W_i^t(p_j^t) \qquad (5)$$

where $0 \le W_i^t(p_i^t) \le 1$ is some weighting function of the contribution of the $\mathcal{N}(p_i^t)$ of point p_j^t on the vector at p_i^t.

By manipulating $W_i^t(p_j^t)$, we can enforce a variety of spatial coherence constraints on the vector field. Hence the vcm implements a voting scheme by which neighbourhood point correlations affect the vector v_i^t at point p_i^t. We can convert this into a 'likelihood-map' for v_i^t by normalizing it, subject to a noise thresholdT_{vcm} given by (6):

$$|vcm(p_i^t)| = \frac{vcm(p_i^t)}{peak'\left(vcm(p_i^t)\right)} \qquad (6)$$

where

$$peak'\left(vcm(p_i^t)\right) = \begin{cases} max\left(vcm(p_i^t)\right) \\ \quad if\ max\left(vcm(p_i^t)\right) > T_{vcm} \\ \infty \qquad otherwise \end{cases}$$

$|vcm(p_i^t)|$ therefore maps the likelihood of terminal points for vectors originating from p_i^t due to neighbourhood point correlations.

Due to need a real time tracking process, we used a dynamic method to calculate the $\mathcal{N}$. In a way that instead of computing the $\mathcal{N}$ for each point p_i^t, we once compute the $\mathcal{N}$ array for the featured points of Ω_i^t and then extract the $\mathcal{N}$ vector of each individual p_i^t. With employing of this dynamic technique, the proposed VCM tracking approach has time cost less than half of the original VCM.

2.6 Crowd Estimation

The number of people in the scene doesn't increase or decrease suddenly, so in order not to lose information of the previous scene as changing from the solitude state to the crowded one and vice versa we need to apply this process gradually. But the question is how someone can detect the time to change between these two states. In the proposed method, we divided the whole area of tracking into multiple horizontal stripes, in a way that in each stripe only one person can fit within the strip's width. In addition, in the scene a fast and reliable algorithm estimates the population in each strip which is between 0 and 1 and it's called the crowd coefficient. In order to obtain this coefficient, we employ the proposed method in (Kong, Gray and Tao, 2006) which utilizes two different features. The features obtained from the foreground map and the ones achieved differently from the edge orientation map.

The proposed method uses the foreground map in order to find the blob size and then establish a blob size histogram as given by (7):

$$H_b(i) = \sum_k S(k) \mid B(i) \le S(i) \le B(i+1) \qquad (7)$$

where $H_b(i)$ and $B(i)$ denote the size and count for the ith bin, and $S(k)$ is the size of foreground blob k.

The edge orientation map is used to create $E_b(i)$; an edge orientation histogram which sums the number of pixels at each 22.5 degrees interval. Since

(a) (b)

Figure 8: Samples of changing from the solitude situation to the crowd one.

the number of vertical edges in the scene is strongly correlated with the number of pedestrians present, the first and the last bin are the peaks. After extracting features and obtains the histograms, we use PCA algorithm which is more efficient and decrease the number of misclassification. In order to train the PCA, we normalize the size of each bin by dividing it to the number of pixels in the image to obtain more distinguishable spaces. Therefore the crowd coefficient is given by (8):

$$C_b = \max_i Dis(F_b(i), P(i)) \qquad (8)$$

where $F_b(i) = \{H_b(i), E_b(i)\}$ is the feature vector and $P(i)$ is the coefficients vector of ith trained space.

With inspecting the entering strip of the scene, we can detect the time to change between the solitude state and the crowd one. And also, with the help of parameters related to size of body, we can calculate the speed ratio of the population. Based on the statistical analysis, the speed ratio is in relation with crowd coefficient. In a way that the more the population is, the less the speed ratio will be.

In this method, we define the track area and the crowd area in the scene which in solitude situations are defined dynamically and complement each other, so that sum of these two areas is always equals to the whole scene. Assume it's a solitude situation and it's changing to the crowd state now. We start to expand the crowd area in proportion to the speed ratio, so that this region grows as the people move as shown in Fig. 8. In the mean time, we check the entering strip continuously and whenever the crow coefficient decrease enough; the track region starts to spread out. When the exiting strip is a part of the crowd region, in order to count people we operate based on the coefficient of the exiting strip and the speed ration by the given equation (9):

$$N = \propto (1 - C_n)/\gamma \qquad (9)$$

where $\propto, \gamma$ are based on the environmental conditions which control the output result.

Figure 9: A snapshot of the captured image and the result of the algorithm.

And on the other hand, when the exiting strip is a part of the solitude region, the counting method is as described before.

3 EXPERIMENTAL RESULTS

We evaluate our algorithm quantitatively by experimenting on one of the biggest Islamic holy places frames with more than 70 gates and over 3 million people incoming per month. The gates have outdoor situation such as both heavy and light shadows, cloudy and rainy weather, and etc. The scene has been captured with a CCD camera at a frame rate of approximately 13 fps at 640 x 480 pixels. (Fig. 9)

Fig. 10 shows the evaluation of counting results for a one hour period on a day between 10:10 a. m. and 2:10 p.m. The accumulated count demonstrates that the automatic count is overestimated.

However, the limited time domain error cannot be referable and does not provide meaningful results for larger time domains. Fig. 11, shows the relative error percent over times which provide a more meaningful way for analyzing error.

It can be inferred that, the relative error can be decreased to 2% for a counting interval of 1 hour and 1% after 2 hours. The accuracy of automatic Counting is really noticeable and is up to 99% which is displayed in Fig. 11.

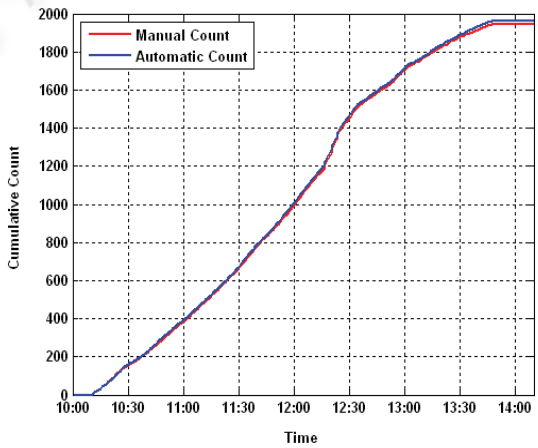


Figure 10: Accumulated Counts.

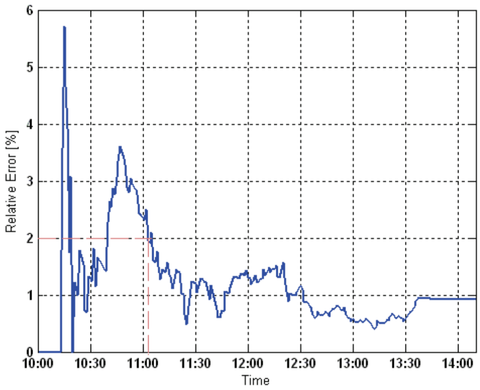


Figure 11: Relative Error.

4 CONCLUSIONS

In this paper, we have a new approach for simultaneously detecting and tracking pedestrian in a crowded scene; the gateways of a sacred place acquired from a stationary camera. Our contribution to this work mainly includes a dynamic size mask which is being employed to remove the undesirable background regions that are labelled as foreground by the Gaussian Mixture Models, the PCA based algorithm which is utilized to detect the head shape objects, the Dynamic-VCM in order to track the objects in real time manner and finally using statistical features and PCA to estimate the population in crowded scenes. The experimental results on real-life data show the robustness and accuracy of our method.

This novel approach could be extended in following fields. 1) We are interested in extend the system to detect and track other classes of objects (e.g., cars). It can be enabled by reconstructing the state space of PCA. 2) We are determined to improve our system in a way that it will be enable to switch between the detecting and tracking and the crowd estimation method in case of overcrowded states.

ACKNOWLEDGEMENTS

This work was supported by Khorasan Science and Technology Park (KSTP) and the Dotis Research and Development Inc.

REFERENCES

Tabb, K.a.D., Neil and Adams, Rod and George, Stella, 2004. Detecting, Tracking & Classifying Human Movement using Active Contour Models and Neural Networks. in *Tabb2004.*

Koschan, A., S.K.K., Paik, J.K., Abidi, B., Abidi, M., 2002. Video Object Tracking Based On Extended Active Shape Models With Color Information. In *1st European Conf. Color in Graphics, Imaging, Vision.* University of Poitiers.

Viola, P., Jones, M.,Snow, D., 2003. Detecting pedestrians using patterns of motion and appearance. In *Proc. 9th Int. Conf.Computer Vision, pages 734–741, 2003.*

Monteiro, G., Peixoto, P., Nunes, U., 2006. Vision-based Pedestrian Detection using Haar-Like features. In *Robotica 2006 - Scientific meeting of the 6th Robotics Portuguese Festival, Portugal*

Zhao, T., Nevatia, R., Fengjun Lv, 2001. Segmentation and tracking of multiple humans in complex situations. In *CVPR2001.*

Masoud, O., Papanikolopoulos, N.P., 2001. A novel method for tracking and counting pedestrians in real-time using a single camera. In *IEEE Trans. on Vehicular Technology, vol. 50, no. 5, pp. 1267-1278*

Kong, D., Gray, D., Tao, H., 2006. A Viewpoint Invariant Approach for Crowd Counting. In *ICPR'06, pp.1187-1190, 18th International Conference on Pattern Recognition.*

Rahmalan, H., Nixon, M. S. and Carter, J. N. 2006. On Crowd Density Estimation for Surveillance. In *International Conference on Crime Detection and Prevention*, London UK.

Quek, R.B.a.F., 2006. Accurate Tracking by Vector Coherence Mapping and Vector-Centroid Fusion. In *ICPR'06, International Journal Of Computer Vision, 2002.Recognition.*

Stauffer, C., Grimson, W, 1999. Adaptive Background Mixture Models for Real-Time Tracking. In *CVPR'99 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

Kong, D., Gray, D., Tao, H., 2006. A Viewpoint Invariant Approach for Crowd Counting. In *ICPR'06, The 18th International Conference on Pattern Recognition*

Villamizar, M., Sanfeliu, A., Ansrade-Cetto, J., 2009. Local Boosted Features For Pedestrian Detectiobn. In *Pattern Recognition and Image Analysis, Spriner Berlin / Heidelberg*

Eng Aik, L., Zainuddin, Z.,2009. Real-Time People Counting System using Curve Analysis Method. In *The International Journal of Computer and Electrical Engineering.*