

Design and Validation of a System for People Queue Statistics Estimation

Vasu Parameswaran, Vinay Shet, Visvanathan Ramesh

Abstract Estimating statistics of people queues is an important problem for many businesses. Monitoring statistics like average wait time, average service time and queue length help businesses enhance service efficiency, improve customer satisfaction and increase revenue. There is thus a need to design systems that can automatically monitor these statistics. Systems that use video content analytics on imagery acquired by surveillance cameras are ideally suited for such a monitoring task. This chapter presents the systematic design of a general solution for automated visual queue statistics estimation and its validation from surveillance video. Such a design involves the careful consideration of multiple variables such as queue geometry, service-counter type, illumination dynamics, camera viewpoints, people appearances etc. We address these variabilities via a suite of algorithms designed to work across a range of queuing scenarios. We discuss factors involved in the systematic validation of such a system such that realistic performance assessment over a wide range of operating conditions can be ensured. We address validation, evaluation parameters and deployment considerations for this system and demonstrate the performance of the proposed solution.

Vasu Parameswaran

Nokia Research Center, Palo Alto, CA. Work performed when author was at Siemens Corporate Research, Princeton, NJ, e-mail: vasu.parameswaran@nokia.com

Vinay Shet

Image Analytics and Informatics, Siemens Corporate Research, Princeton, NJ, e-mail: vinay.shet@siemens.com

Visvanathan Ramesh

Goethe University, Frankfurt/Main, Frankfurt, Germany . Work performed when author was at Siemens Corporate Research, Princeton, NJ, e-mail: Ramesh@fias.uni-frankfurt.de

1 Introduction

Managing human queues is important for a variety of businesses. Amusement parks need to handle queues of people waiting their turn for rides, airports need to manage people waiting to be screened by security, retail stores need to serve patrons waiting to pay for their purchases. If such queues of people are not actively monitored by the business, it often results in lower customer satisfaction, inefficient resource allocation and loss of revenue. There is thus a need, to not only monitor queue statistics in real-time but also to better understand trends that emerge in such queues over time. Automated video content analytics provide one means to queue design monitoring systems that are capable of doing both.

This chapter presents the systematic design and validation of a general solution for a video-surveillance-based, automatic, visual queue statistics estimation. The design of such a system requires the careful consideration of several variables, which include the type of queue (e.g. snaking queue, linear queue etc), the type of service counters (check-in counters, baggage screening counters etc.), level and dynamics of illumination, camera viewpoint, people appearances (with and without bags) etc. These variabilities are addressed by decomposing the main task into two sub-tasks, and combining their outputs. The subtasks are: (a) Queue size estimation, and (b) Service time estimation. We show that these modules can be combined together to analyze a range of queuing scenarios. The two tasks are solved using algorithmic components for crowd count estimation, and person detection & localization. In order to obtain a realistic assessment of the performance of a solution, it is necessary that a wide range of operating conditions be covered while keeping validation costs to a minimum. In this chapter we discuss the validation process we followed for a particular deployment of the proposed queue solution, the evaluation metrics used, and the results of the process for our solution.

A complete queue statistics monitoring system needs to perform several tasks, (i) estimate the number of people waiting in the queue, (ii) estimate the average service time at all the service counters, (iii) in case of multiple service counters, detect whether they are operational, and (iv) estimate the expected waiting time. A critical component in the system is one that estimates the number of people waiting in the queue. Prior research has focused on two related problems - crowd density estimation and crowd count estimation. Crowd density estimation involves devising measures related to the overall crowdedness of the scene while crowd count estimation, as the term implies, involves estimating the number of people in the scene. Both tasks involve using image features such as the weighted area of foreground blobs in the image and their mapping to either crowdedness or crowd count estimation. Approaches differ based on the measure that is used to characterize the crowd, features selected on image and the type of learning used to map features to the measure.

Paragios and Ramesh [14] use a discontinuity preserving Markov Random Field based approach to perform smooth foreground segmentation that combines spatial constraints with intensity modeling. They then combine the change detection map with a geometry module that performs soft-auto calibration to estimate a crowded-

ness measure in a given region. In [1] Chan et al describe a system to estimate the size of an inhomogeneous crowd by first segmenting the crowd into components of homogeneous motion using mixtures of dynamic texture-motion models. They extract a set of simple holistic features from these segmented regions and learn the correspondence between number of people in each segment and the extracted features using Gaussian process regression. In [9], Kong et al describe a learning based approach for view point invariant crowd counting. They first extract edge orientations and blob size histograms as features. They compute a density map measuring relative size of individuals and global scale measuring camera orientation and use it to normalize the extracted features. They learn the relationship between extracted features and crowd density using linear fitting and neural networks. In [8] Kilambi et al describe a system to estimate the counts of people in groups and track them. They learn the mapping between scene and camera calibration parameters to the counts of people visible in the image. Zhao and Nevatia [23]) use a generative model for the formation of image silhouettes and infer the number and positions of people using the well-known Markov Chain Monte Carlo method.

Another class of approaches directly detects humans in the scene. Approaches here tend to fall primarily in two categories: those that detect the human as a whole and those that detect humans based on part detectors. Among approaches that detect humans as a whole, Leibe et.al [10] employs an iterative method combining local and global cues via a probabilistic segmentation, Papageorgiou et. al. [13] uses SVM detectors, Felzenszwalb [4] uses shape models, and Gavriilla [6, 5] uses edge templates to recognize full body patterns. A popular detector used in such systems is a cascade of detectors trained using AdaBoost as proposed by Viola and Jones [19]. Such an approach uses haar wavelets features and has been successfully applied for face detection in [19]. In [20] Viola and Jones applied this detector to detect pedestrians by augmenting haar wavelets with simple motion cues for enhanced performance. Another popular feature is the histogram of oriented gradients, introduced by Dalal and Triggs [2] who used it with a SVM based classifier. This was further extended by Zhu et. al [24] to detect whole humans using a cascade of histograms of oriented gradients. Tuzel et al [18] use covariance matrices to represent humans and learn a classifier on a Riemannian manifold.

Part based representations have also been used to detect humans. Mikolajczyk et al. In [11] Mikolajczyk et al divide the human body into seven parts and for each part, a Viola-Jones approach is applied to orientation features. [12] divides the human into four different parts and learns SVM detectors using Haar wavelet features. Wu and Nevatia [21] use edgelet features and learn nested cascade detectors [7] for each of several body parts and detect the whole human using an iterative probabilistic formulation. Mohan et.al. [22, 21, 10] follow up low level detections with high level reasoning that allows them to enforce global constraints, weed out false positives, and increase accuracy. Shet et al [16] use part based detectors with logical reasoning to fuse the results into scene-consistent human hypotheses.

Our queue monitoring system is expected to have high accuracy, easy deployment, and easy configuration. It is expected to work well under a wide variety of operating conditions including different queue structures, viewpoints and person

appearances. Furthermore, the system is expected to be deployed in a fixed-camera setting allowing the possibility of performing background maintenance and change detection. Due to these factors, we chose not to use methods that are based primarily on image-features and machine learning. We instead chose to use and build upon foundational algorithms designed for a fixed-camera setting.

The rest of the article is organized as follows: In section 2 we decompose the queue statistics monitoring problem into its constituent components and make explicit our assumptions. In section 3, we identify several sources of variation that can adversely affect such a queue monitoring solution. In the following sections we describe techniques that help us handle these variability. In section 4 and section 5, we describe our approach for crowd count estimation and people detection & localization respectively. In section 6 we describe the queue monitoring system that combines all these technologies. In section 7 we list issues that need to be considered during deployment. We then describe how the solution is validated in section 8. Finally, in section 9 we conclude with a review of the solution including rationales for various choices made, a discussion of the results and insights obtained from the evaluations, current limitations of the solution and items for future work.

2 Problem Decomposition

One can model a queue as consisting of one or more queue-zones where people wait to be served, and one or more counter zones where people are provided service (see figure 1). We assume that the amount of time each person requires to get service at a counter is a random variable S with probability density: $p_s(t)$. The total time that



Fig. 1 Queue and Counter Zones

a given person needs to wait in the queue before he gets served is a function of two variables - the number of people ahead of him in the queue, and the time taken for each person to be served. If there are N people in the queue, the expected waiting time for a person just entering the queue at the end, T_{avg} , is given by:

$$\begin{aligned} T_{avg} &= E(S_1 + S_2 + \dots + S_N) \\ &= N\bar{S}_N \\ &\approx N \int_0^{\infty} s p_s(s) ds \end{aligned} \quad (1)$$

where S_i are independent samples drawn from the service time density p_s and \bar{S}_N is the sample mean. As we discuss later, the number of people in queue may not be known precisely and instead also come from a probability density function $p_N(x)$. In this case the waiting time is given by:

$$T_{avg} \approx \left(\int_0^{\infty} x p_N(x) dx \right) \left(\int_0^{\infty} s p_s(s) ds \right) \quad (2)$$

Thus, a queue solution can be based upon two modules - a queue module (QM) that analyzes images from one or more cameras observing the people waiting in queue, and a counter module (CM) that analyzes images from one or more cameras observing the counter areas. QM estimates the number of people in the queue while CM estimates how long each person waits in a counter zone to complete service.

Modeling the problem in this manner offers several advantages in the design of a solution. Firstly, there is no implication that the solution necessarily depends upon detecting and tracking multiple individuals throughout the queue. Secondly, there is no explicit assumption made on the queue structure (e.g. straight-line queue, snaking queue, or combinations) - one just needs to know the number of people waiting in the queue. If the structure of the queue is known apriori, we could exploit it and obtain better accuracy. However, in our analysis we assume that the queue structure is unknown. Thirdly, there are no explicit assumptions made on the imaging conditions, e.g. camera position, person appearances, illumination level, etc. We incorporate robustness to these variabilities into the design of QM and CM which are based on low level analytics. Lastly, decomposing the queue problem as described here enables the re-use and adaptation of available algorithms for crowd density estimation, e.g. [14] and person detection e.g. [3] for QM and CM. In the following sections, we describe use and adaptation of such algorithms in greater detail.

2.1 Assumptions

The design of algorithms for QM and CM makes the following assumptions that are reasonable for a typical surveillance scenario:

1. The camera is static (although the scene may be dynamic)

2. Vertical lines in the world project to vertical lines in the image. Extension to the non-vertical projection case is in principle straightforward.
3. The projected width and height of a person of average size as a function of image position is provided. Since this is partial calibration information and not a full 3D geometric transformation, we refer to our scenes as ‘quasi-calibrated’, and the functions as ‘quasi-calibration’.

A static camera allows the possibility of maintaining a statistical model of the ‘empty’ scene, i.e. the scene devoid of foreground objects, and of computing a change image, which is a binary mask $B(x,y)$ that corresponds to foreground objects in the scene. Background maintenance and change detection are rapidly maturing sub-fields within video surveillance. We assume the availability of modules for generating B .

3 Robustness to Variability

Several sources of variability influence the images coming into the queue processing system which QM and CM need to be robust to. Mainly, the following are the classes of variable:

1. Illumination (slow and fast changing, local and global)
2. Camera viewpoint (top-down and general)
3. Variation in person appearances (pose, articulation, etc)

For fast and slowly changing local and global illumination, we use our prior work on illumination-robust change detection [15], [17]. Taken together, the methods effectively deal with most of the illumination changes found in a typical queue monitoring environment. Therefore the binary mask $B(x,y)$ can be expected to be largely free of illumination effects. Camera viewpoint and person appearance variability are handled by the respective modules for QM and MM. The scene background is maintained over time and updated conservatively. In other words, rather than updating the background at a specific exponential adaptation rate at all times which causes foreground objects to become blended into the background, the background is updated only when it is determined with high probability that the change is due to illumination.

In the next two sections we describe two key components of the overall solution, namely crowd count estimation and person detection & localization. Subsequently, we describe the visual queue analysis system.

4 Crowd Count Estimation

The task of the crowd count estimation component is to estimate the number of people in a given region of interest in the scene. We adapt and improve the crowd density estimation work of [14] to carry out the task.

Denote the width by $w(y)$ and height by $h(y)$. We work with the y axis pointing downwards as in an image. Let the width and height of the image be W and H respectively. We propose that the crowd size C can be expressed as a weighted area of B :

$$C = \sum_{i=1}^H \sum_{j=1}^W \theta(i)B(i, j) \quad (3)$$

Paragios and Ramesh [14] choose $\theta(i) = 1/(w(i)h(i))$. Although this is approximately position invariant and a reasonable weight function, in this work, we derive a weight function that incorporates position invariance explicitly. Assume that there is one person in the scene and that the person is modeled by a rectangle with its top left corner at (x, y) . In this case we seek a function $\theta(\cdot)$ such that:

$$\sum_{i=y}^{f(y)} \theta(i) \sum_{j=x}^{x+w(f(y))} B(i, j) = 1 \quad (4)$$

Here $f(y)$ is the y coordinate of the person's foot $f(y) - y = h(f(y))$. Let the y coordinate of the horizon be y_v which can be obtained by solving $h(y) = 0$. The smallest y coordinate for a person's head we consider is given by :

$$y_0 = \max(0, y_v + \epsilon) \quad (5)$$

Let y_{max} be the maximal head position above which the feet are below the image. Equation 4 applies for positions $y_0 \leq y \leq y_{max}$. For $y > y_{max}$ the weighted sum is adjusted to the fraction of the visible height of the person. We thus have $(H - y_0 + 1)$ equations in as many unknowns and the linear system of equations can be solved to yield $\theta(\cdot)$. Although this is in principle correct, the equations do not enforce smoothness of $\theta(y)$ and hence the resulting weight function is not typically smooth (see figure 2). We could remedy this problem using regularization (e.g. Tikhonov) but we found the following method quite effective in our case: We first define the cumulative sum function

$$F(y) = \sum_{t=y_0}^y \theta(t) \quad (6)$$

The inner sum in equation 4 is $w(f(y))$ and hence equation 4 can be written as

$$F(f(y)) = F(y) + \frac{1}{w(f(y))} \quad (7)$$

This is a recurrence relation in F . We arbitrarily set $F(H) = 1$ and obtain F at sparse locations: $y = \{H, H - h(H), \dots, y_0\}$. Next we interpolate F using a cubic spline and finally obtain θ as follows:

$$\theta(y) = F(y) - F(y - 1) \quad (8)$$

Figure 2 shows the estimate obtained by this recurrence method and the original system. Also shown is the projected area inverse function used in [14]. $\theta(\cdot)$ will be dependent upon scene-geometry but it needs to be calculated only once at system setup time. The quasi-calibration prior coupled with position invariance helps us deal with a variety of viewpoints, thereby satisfying the viewpoint robustness requirement.

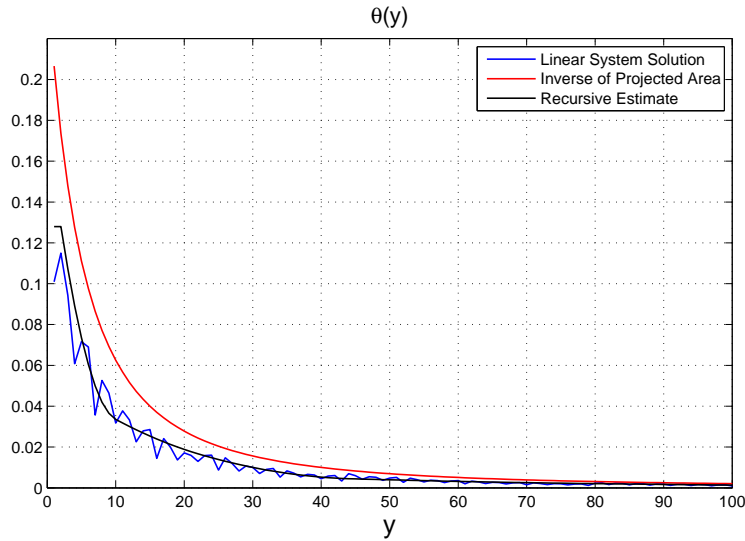


Fig. 2 Weight function (calculated versus projected area based)

For $N > 1$, C , the crowdedness measure obtained from equation 4 above will approximately equal N if the people do not occlude each other. However, if there are occlusions, C will not be unique, but can be described by a probability distribution function $P(C|N)$. The entropy of $P(C|N)$ will depend upon the camera angle and be lowest for a top-down view. We estimate this pdf by simulating N humans in various configurations and degrees of overlap and calculating C using the resulting binary image and the scene-specific weight function we calculated in section 2. This process allows the inclusion of more detailed human body models and specific perturbation processes to the binary image. For example, if the sensor noise characteristics are available we could incorporate them into the simulation process. Similarly, structured perturbations such as shadows, reflections, and carried luggage can also be introduced into the simulation process, allowing the scaling up to more complex

scene conditions. While we do not attempt to mathematically aim for invariance to such perturbations, including them into the simulation and mapping process allows us to be robust to them. The essential output of the simulation process is an estimate $P(C|N)$. Let the maximum number of people in the scene be N_{max} . The simulation process produces $P(C|i), 1 \leq i \leq N_{max}$. At runtime, Bayes rule is used to find the posterior:

$$P(N|C) = \frac{P(C|N)P(N)}{\sum_{i=1}^{N_{max}} P(C|i)P(i)} \quad (9)$$

We further reduce the computational burden at run time by storing the posterior (rather than the likelihood) in a look up table. Hence, all that needs to be done at run-time is the calculation of C and a lookup into the table to obtain $P(N|C)$. We also approximate $P(N|C)$ as a normal distribution and simply store the mean and standard deviation in the table. This has been found to work quite well in practice.

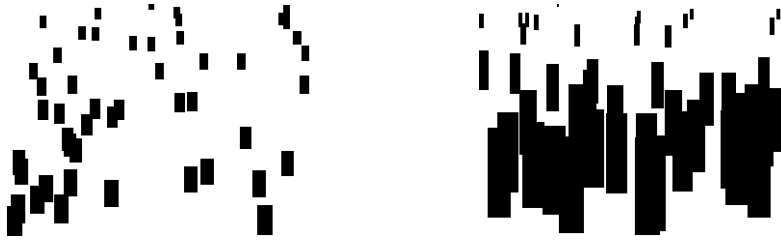


Fig. 3 Example Images showing high-ceiling (left) and low-ceiling (right) camera positions

Figure 3 shows simulated images from a camera position higher up in the ceiling and a general view (top row). Figure 4 shows $P(C|N)$ for $N = 5$ and $N = 10$ for the respective views using simulations. As expected, the separation between the two pdfs is better for the approximately top-down view, which also has a lower variance. It can be noted that there is a slight overestimate introduced due to the weight-function calculation procedure we used, resulting in C not exactly equaling N . The overestimate does not have a practical impact because we use equation 9 to estimate N from C . It can be recalled that most view-based training is done automatically at system setup-time by simulating different numbers of people-silhouettes in different configurations and establishing the distribution $P(C|N)$. While simulations can take account for a lot of variability in people appearances, we still found it necessary to fine-tune the crowd count output to handle unmodeled effects. This is done by comparing the output to manually annotated ground truth and estimating a scale and offset that can best match the two. In practice, while the amount of data needed for the fine-tuning is small, it is essential that different levels of crowding be present in the data. The validation numbers reported in section 8 are from fine-tuning the

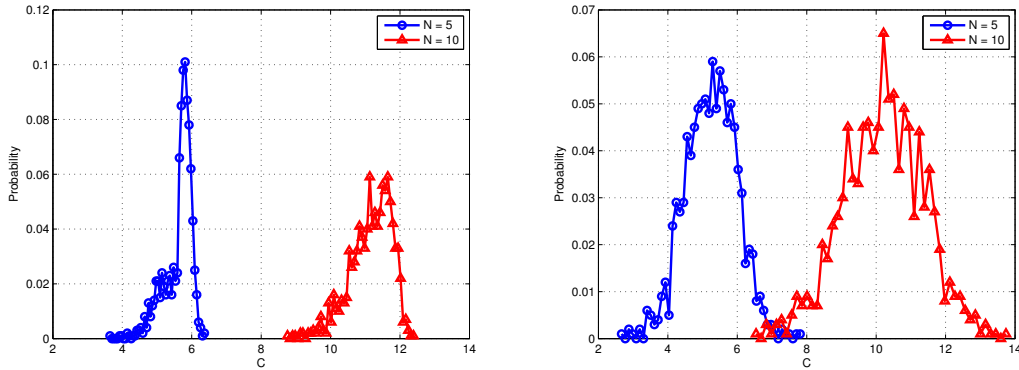


Fig. 4 Example results for $P(C|5)$ and $P(C|10)$ using simulations for high-ceiling (left) and low-ceiling (right) camera positions

mapping based on a few hours of ground truth data covering different levels of crowding.

5 Person Detection and Localization

The goal of this component is to detect and localize people in the scene. Similar to the crowd count estimation component we start with the change image $B(x,y)$. We use our previously published method [3] to carry out the task of detecting and localize people. For completeness, we briefly review the essentials of the algorithm leaving out mathematical details which can be found in [3]. The input to the algorithm is the binary change mask image B and the quasi-calibration. The output is the number and location of people. For an example, please see figure 5. A key goal in

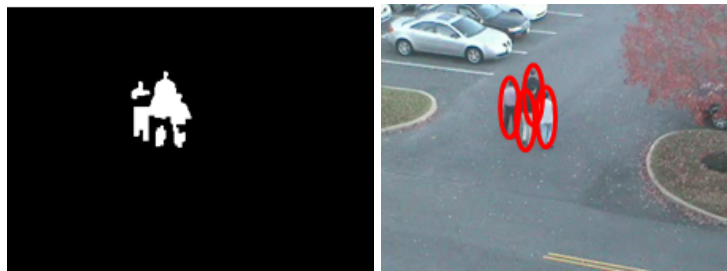


Fig. 5 Crowd Segmentation Input and Output

the work was fast operation so that real time operation is possible. To this end, the

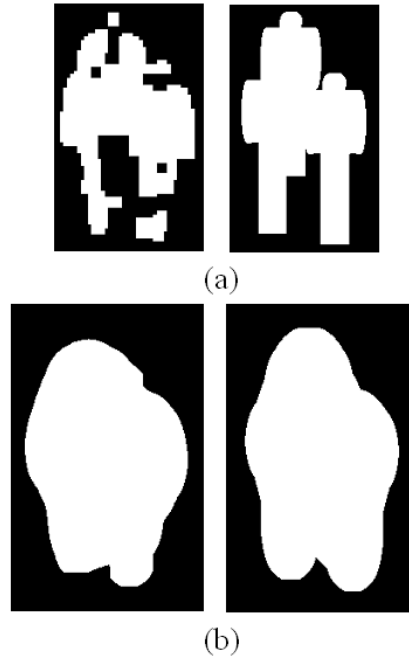


Fig. 6 Comparison of two input images of same parameter. (a) input image: left is from real video (b) right is from simulation. (b) filtered output.

solution is calculated in two stages: First, a quick guess is calculated which maps the size and shape of the foreground silhouette (e.g. as seen in the left image in figure 5) to the number and positions of people in the silhouette. We refer to this stage as an *indexing* stage which produces a solution that is quite close to the true solution. The approximate solution is refined, as necessary, using a Markov Chain Monte Carlo based search method (specifically the method of [23]).

Key sources of variability are the relative poses of people in the silhouette, their articulation, and errors in change detection due to camouflage (i.e. where the foreground and background appearances are similar) and sensor noise. These variabilities are handled by filtering the change mask using a morphological filter of rectangular shape. The filtered shape is represented by Fourier descriptors of the contour. Figure 6 shows the filtering process on a real image and an idealized image. The resulting filtered shapes are quite similar in appearance and result in similar values of the Fourier descriptors. Prior to deployment, given the quasi-calibration, the system goes through a training phase where it generates silhouettes containing different numbers of people, articulations and positions and builds a look up table that maps Fourier descriptors of the morphologically filtered silhouettes to the number and positions of people in the silhouette. In order to limit the size of the look up table, the

system has an upper bound of six people in a given blob (note that the scene may have several such blobs).

6 The Queue Analysis System

We are now ready to describe the overall queue analysis system. As discussed in section 2 the queue analysis system can be composed by combining two modules - a queue module (QM) that estimates the number of people waiting in the queue and a counter module (CM) that estimates the service time, i.e. the time required to obtain service at a counter. QM is built using the crowd count estimation component to estimate a probability distribution over the total number of people waiting. As crowd counts vary slowly over time we also found it useful to low-pass filter the crowd count estimate such that sudden changes in value were disallowed. In the scenario where a single queue is covered by multiple cameras, the probability distributions obtained from each camera are combined together.

CM is built using the person detection & localization component. Firstly zones are configured on the scene that correspond to regions where the person waits for service (e.g. as in figure 1). The person detection & localization component is used to localize people within the defined service counter zones. Typically only one person stands in the zone at a time. The system estimates the amount of time that a person waits in the service counter zone, i.e. the “service time”, and maintains statistics of this quantity over time. Errors in this process can arise from intermittent missed detections as well as false alarms. Such errors are sporadic and random in nature. These errors are handled quite effectively by a temporal smoothing process that makes use of two parameters: One is the minimum inter-arrival time between two people. Another is the minimum service time. Provided that the probability of successive miss detections over the minimum inter-arrival time, and the successive false alarm rate over the minimum service time are negligible, the system can recover from both types of error. These parameters were set based on training data from the installation.

Finally, equation 2 is used to estimate the expected waiting time of people in the queue.

7 Deployment considerations

The algorithmic components that make up the queue monitoring system are designed for general views. In principle they can thus be deployed at sites where cameras have already been installed. However, where customer projects allow, it is useful to choose camera positions for best overall effectiveness. Generally, the more the number of cameras deployed, the better the coverage of the queue becomes. However, computational as well as economic constraints often limit the number

of cameras that can be deployed in a real world setting. An optimal setup is one that minimizes the number of cameras required, maximizes the coverage of these cameras, and maximizes the accuracy of the system. A top-down view optimizes these different considerations (also recall from figure 3 that top-down views result in lower uncertainty for crowd count estimation). Such views are easiest to achieve in installations sites with high ceilings. Figure 7(a) provides a good example of such a camera set up. In installation sites where ceiling height is low, cameras need to be positioned in an oblique set up. Figure 7(b) provides an example of such an installation. Such oblique views provide smaller coverage and thus such situations typically need a higher number of cameras.



Fig. 7 Camera view of (a) high ceiling mounted camera and (b) low ceiling mounted oblique camera

For economic reasons, it is impractical to install a set of cameras, observe the goodness of the camera set up, and iteratively refine camera placement and numbers. Camera installations in major businesses like airports typically involve structural modifications to the building that make it impractical to perform anything but minor adjustments post installation. This implies that the initial recommendation provided to the camera installer needs to be as optimal as possible. One possible approach to accomplish this is to perform 3D modeling of the installation site and use this 3D model to plan for the optimal number and placement of cameras.

To construct 3D models, it is possible to use any existing CAD drawings of the building or even detailed measurements of different structures such as distances between walls, height of ceiling etc. Such 3D models can easily be constructed using 3D modeling software such as AutoCAD and even modern 3D game engines. Figures 8 shows example screen shots of the point of view of the proposed camera installation when viewed within such a 3D model. Use of such 3D models to plan the number and placement of cameras greatly minimizes costly errors in camera setup. Care must however be taken to account for structures in the building not typically modeled in CAD drawings. Examples of such structures include overhead signage in an airport (as can be seen in Figure 7 (b)), ceiling mounted display monitors etc.

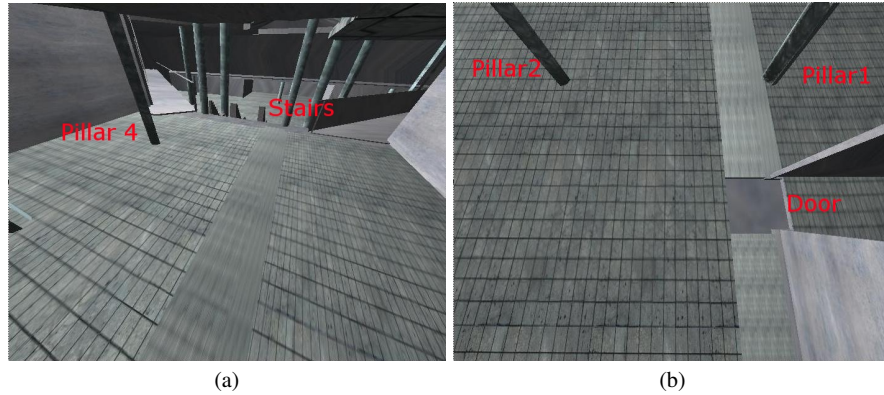


Fig. 8 Simulated camera views from 3D model of target installation site.

8 Validation process

The queue analysis system was validated on data from a baggage screening checkpoint at an airport. The queue in this scenario spanned multiple large areas and required five cameras whose views were mostly non-overlapping. On the other hand, a single camera was sufficient to view the baggage screening zones (shown in figure 1). The system was required to flag situations when the average waiting time exceeded a maximum value or was lower than a minimal threshold. In general, validation requires balancing two competing needs: On the one hand, in order to obtain a realistic assessment of the performance, validation data needs to be representative of the range of situations that will occur in practice. It needs to cover different levels of illumination, different sizes of crowd, different queue formations, and different wait times. On the other hand, data collection and ground truthing costs need to be kept as low as possible. To this end, we selected eight cumulative hours per camera that spanned a range of operating conditions. Figure 9 shows an example from the camera covering the largest number of people. The variability shows low and high crowding, strong illumination effects, and evolving queue structures.

As crowd counts are expected to vary slowly over time, we reduced ground truthing costs by annotating at a low frequency (typically every 1000 frames, which corresponds to about 40 seconds) and interpolating the data for intervening times. Annotations were done for the following four quantities:

1. Number of people.
2. Average Service Time.
3. Waiting Time.
4. The number of counters open and closed.

Although there were a total of five cameras viewing the queue, most of the time, people were concentrated in two cameras, the results for which are shown in figure 10. The left column shows the trend for the number of people seen in the camera.



Fig. 9 Different Scene Conditions Included for Validation. Clockwise starting from top-left: Low Crowding, High Crowding, Strong Illumination Effects, Different Queue Structures.

The blue line shows the actual number of people while the red line shows the number of people estimated by the system. The right column shows the error histograms for the first two cameras. Overall, it can be seen that the ground truth and the estimates agree fairly well. For the first camera (top row) which holds up to 30 people, the absolute error was less than 6 people for 96% of the time, and the average absolute error was 1.72 people. For the second camera (bottom row) which holds up to about 100 people, the absolute error was less than 8 people for 96% of the time, and the average absolute error was 3.25 people.

Although the counts agree reasonably well in practice, upon closer examination, it can be seen that there are errors at certain times. We traced these errors to two main sources. Firstly some errors occurred due to insufficient calibration. Recall that we only ‘quasi-calibrated’ the scene. Radial distortion contributed to errors when people were near the boundary of the scene (the heavy-crowd case). Secondly, sudden illumination changes due to sunlight streaming in from the ceiling were sometimes not fully compensated by the system, which resulted in residual error (especially

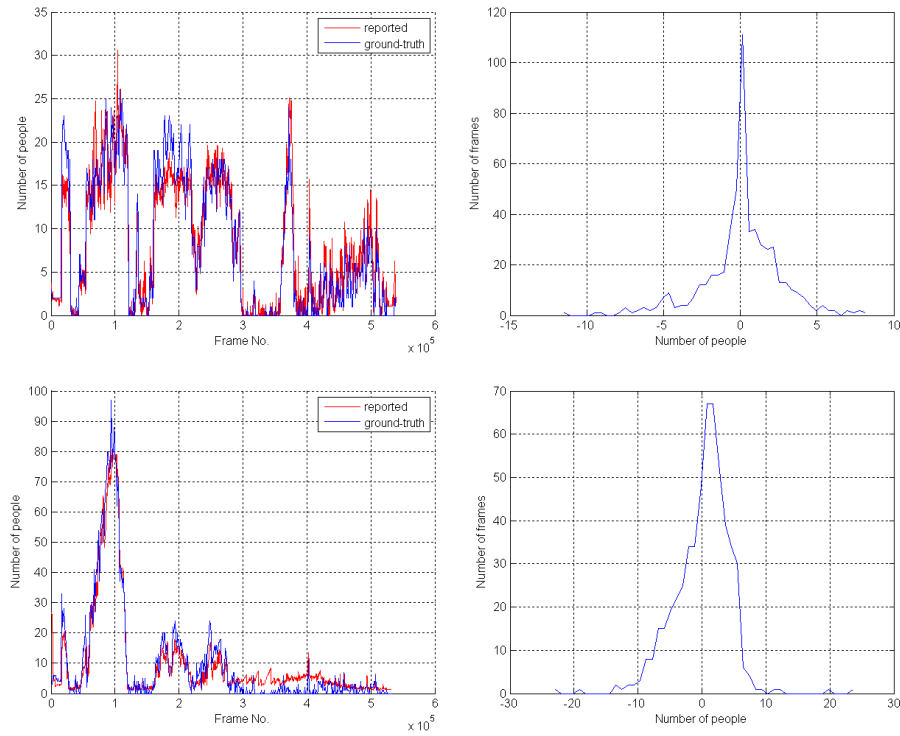


Fig. 10 Crowd Count Comparison and Error Histograms (best viewed in color)

for the second camera). Nevertheless the amount of error was small enough to have negligible impact on the overall waiting time.

There are two parts to service time estimation. Firstly, the time required for a person to pass through a given counter zone is needed (i.e. counter specific service time). Secondly, it is also required to determine the open/closed state of each service counter. For the specific deployment scenario under consideration, namely baggage screening at an airport, it was determined externally that the per counter-screening time held fairly constant at about 15 to 17 seconds on average. This is to be expected because the type of ‘service’ is the same. Consequently, it was decided to use the externally determined service time at this particular deployment of our queue solution. Determining the open/closed state of each counter was still necessary, and for this task we used the person detection and localization algorithm of section 5 to determine if a person occupied a given counter zone. The accuracy of counter open/close detections was found to be 96%. Finally, figure 11 shows the error histogram for the average waiting time estimation. The average absolute error was found to be 0.8736 minutes. Recall that the waiting time is a statistic derived from the queue and counter modules, which in turn rest upon the illumination robust change detection module. It is therefore to be expected that errors in any of the four-

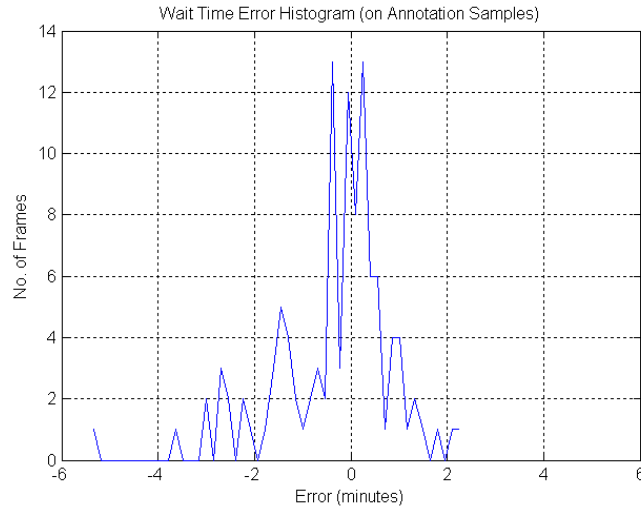


Fig. 11 Wait Time Error Histogram

dational modules will propagate up to the waiting time calculation. Besides the two sources of error described above that adversely impacted the crowd count estimation component, for this particular deployment, we also traced some error down to the fluctuations in the service time. Nevertheless, as the impact on the overall waiting time was minimal, this source of error was ignored for this particular deployment.

9 Conclusions

This chapter presented the design and validation of a system for automatic, video based, people queue statistics estimation. In particular, we designed a system that estimates statistics over the total number of people waiting, the service time per counter, the number of operational service counters, and the waiting time. Such statistics are very useful not only for optimized resource allocation and improved customer satisfaction, but also to understand trends over longer periods of time, thus providing valuable input for business intelligence.

In practice, a queue monitoring system is expected to handle successfully a wide variety of variation in scene conditions arising due to several classes of influencing variable. Mainly these include illumination, queue structures, person appearances, viewpoints, etc. We followed a systematic approach for the design of the system. We first modeled the queue at an abstract level that allowed us to identify the need for two key modules - a queue monitoring module that estimated the total number of people waiting in the queue, and a service-counter monitoring module that estimated the service time per counter as well as determined which service counters

are open. The modules are in turn built upon the algorithmic components of crowd count estimation, and person detection & localization. Robustness to illumination variation is largely achieved by a foundational change detection module we have previously reported in [15] and [17], allowing us to focus on the remaining variables including queue structure, person appearance, and viewpoint. By design the algorithmic components do not need knowledge of the queue structures and are hence automatically robust to different queue formations. We imposed position invariance into the formulation, which, along with quasi-calibration information allowed us to achieve robustness to different viewpoints. The training process for both algorithmic components included simulations of different person appearances and allowed us to achieve robustness to variability in person appearances. We also discussed important factors to consider for the deployment of the queue monitoring system. We validated the queue monitoring system on representative data from a real airport. The data spanned a wide variety of conditions expected under normal operation of the system at the airport. The results demonstrated the overall effectiveness of the queue monitoring system.

We believe that several factors contributed to the successful design of the queue monitoring solution: (1) The systematic approach for the analysis of a queue and the design of the solution based on that analysis that also helped avoid the need to solve harder problems such as long range tracking of people in crowded situations, (2) The systematic enumeration of key sources of variability in the scene and ensuring that they are all addressed either mathematically or via training, and (3) A modular approach which allows us to ‘divide and conquer’ the space of variability as well as more easily pinpoint weaknesses and areas for further work on the system.

Acknowledgements The authors would like to thank Imad Zoghliami and Thomas Baum for helpful discussions and valuable feedback as well as Ningping Fan for his contribution to the project. The authors would also like to thank the airport authorities for providing permissions to use the images included in this article.

References

1. A. B. Chan, Z. S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *Proc. IEEE IEEE Conference in Computer Vision and Pat-tern Recognition (CVPR)*, pages 1–7, June 2008.
2. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR05*, pages I: 886–893, 2005.
3. L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghliami. Fast crowd segmentation using shape matching. *Proc. ICCV*, 2007.
4. P. Felzenszwalb. Learning models for object recognition. In *CVPR01*, pages I:1056–1062, 2001.
5. D. Gavrilu. Pedestrian detection from a moving vehicle. In *ECCV00*, pages II: 37–49, 2000.
6. D. Gavrilu and V. Philomin. Real-time object detection for smart vehicles. In *ICCV99*, pages 87–93, 1999.
7. C. Huang, H. Al, B. Wu, and S. Lao. Boosting nested cascade detector for multi-view face detection. In *ICPR04*, pages II: 415–418, 2004.

8. P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding*, pages 43–59, 2008.
9. D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. *Proc. British Machine Vision Conference (BMVC)*, 2005.
10. B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE CVPR'05 in , San Diego, CA*, pages 878–885. sp, may 2005.
11. K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, May 2004.
12. A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, April 2001.
13. C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. *Intelligent Vehicles*, pages 241–246, October 1998.
14. N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. *Proc. IEEE IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, pages I:1034–1040, 2001.
15. V. Parameswaran, M. Singh, and V. Ramesh. Illumination compensation based change detection using order consistency. *Proc. IEEE CVPR*, 2010.
16. V. Shet, J. Neumann, V. Ramesh, and L. Davis. Bilattice-based logical reasoning for human detection. In *CVPR*, 2007.
17. M. Singh, V. Parameswaran, and V. Ramesh. Order consistent change detection via fast statistical significance testing. *Proc. IEEE CVPR*, 2008.
18. O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1713–1727, 2008.
19. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001.
20. P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV03*, pages 734–741, 2003.
21. B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *ICCV*, Oct 2005. Beijing.
22. T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. *CVPR*, 2:459–466, 2003.
23. T. Zhao and R. Nevatia. Segmentation and tracking of multiple humans in crowded environments. *IEEE PAMI*, 30(7):1198–1211, 2008.
24. Q. Zhu, M. Yeh, K. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR06*, pages II: 1491–1498, 2006.