A Train Station Surveillance System: Challenges and Solutions

Burak Ozer
Engineering Consultant
Perkasie, USA
e-mail: ozerburakpa@gmail.com

Marilyn Wolf
The School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, USA
e-mail: marilyn.wolf@ece.gatech.edu

***Abstract*—In this paper, we describe our multi-year development effort to create a smart-camera surveillance system for use in train stations. We address the problem of activity analysis in very crowded areas and offer solutions to the detection/tracking problems in one of the most crowded environments. The smart camera system is installed in several train stations around Tokyo, Japan and is currently deployed analyzing crowd movement and individual gestures of the passengers in the stations. The paper summarizes the technical problems we faced during our research /development stages as well as during system deployment. The hierarchical smart camera system, which can detect the overall crowd movement in the higher level and individual activities in the lower level, is developed by Verificon Corp., a Princeton University startup company. We developed smart camera algorithms and spent several years commercializing the technology in a joint project with Yokogawa Electric. A railway company was the lead customer on this project. Verificon system passed several rounds of testing, running for several months on live data in several Tokyo train stations and was deployed by the railway company. This project has led to several extensions of the original technology.**

Keywords-surveillance; tracking; gesture recognitionin

I. INTRODUCTION

Smart cameras analyze video to create high level descriptions of the scene and alert security personnel based on the warning/alarm level. The principle of multi-camera surveillance systems is to monitor the intrusion or undesired behavior in a security room by dedicated security personnel. Since the personnel cannot pay constant attention to the monitors, the effectiveness of such systems is limited. On the other hand, smart camera systems increase effectiveness by calling attention to the cameras that capture intrusion or identify undesired activities, reduce the need for monitors (by activating the vision from the necessary camera on the monitor), reduce the need for supervisory personnel (by increasing the number of effectively controllable cameras per personnel), etc. Different approaches have been developed for object detection and tracking. However, most of these methods and systems are suitable to detect individual objects with low occlusion and high resolution levels. Furthermore, the successful extraction of object of interest (OOI) depends in general on the complexity of the background scene or the availability of the background scene without any foreground objects. Thus, there is a need for systems and methods for improved foreground object extraction, detection and tracking for smart camera systems.

There are different approaches for monitoring crowd movement and tracking individuals in very crowded scenes. Rodriguez *et al.* [1] presents a method for tracking in unstructured scenes, Ali *et al.* [2] proposes a method for tracking in structured scenes, where there is one direction of motion in the scene. Khan *et al.* [3] uses a multi-view approach for tracking. The scope of our paper is summarizing problems that we faced during different deployment stages of a train station surveillance system based on our smart camera technology and we believe that some of the solutions that we provided may be used as part of other systems/algorithms including some of the above mentioned papers. We designed algorithm modules to find real-time solutions and improve system performance as well as increase correct gesture detection rates.

The first prototype smart camera system was developed on TriMedia video processing boards at Princeton University. At later stages, during commercialization of the software, the algorithms are designed for RISC processors for Windows and Linux environments as well as for FPGA acceleration. The current system is running on DaVinci evaluation boards. One of the major problems was efficient system design for real-time processing. Processing enormous amount of data obtained from a very crowded scene in real-time is a hardware challenge.

Some other challenges are; changing camera characteristics that vary from unit to unit, changing lighting (low/high light levels, high contrast, flicker effect due to fluorescent light, shadow, automatic exposure correction), changing color balance, camera motion (due to pan-tilt-zoom, wind, shaking due to another object, e.g. train), perspective change, camera calibration, reflection, abrupt lighting changes. Figure 4 shows the flow chart of the gesture recognition/activity analysis algorithm. In the next section we will summarize the problems we faced in the project in more detail. Section 3 covers the smart camera algorithm and solutions offered by the system for the gesture recognition application. In Section 4 we show some experimental results. The reader should note that the frames are not from train stations due to security and disclosure agreements. Some basic algorithm blocks that are used in the current system can be found in our previous work [7].

II. CHALLENGES

Train station lighting creates several types of lighting challenges. One of the basic problems is the flickering effect due to fluorescent lighting. The flicker effect causes a constant lighting change especially in the train stations on the reflective surfaces. Another challenge is the shadow elimination. Shadows, especially on low saturated areas, are very hard to eliminate. The suit colors of the passengers are sometimes detected as shadow areas by the system due to poor lighting conditions. The lighting as well as camera parameter changes affect the gesture detection rate directly, e.g. when a train arrives at a platform the platform camera parameters are changing rapidly which causes poor detection rates. Mixed indoor outdoor lighting is responsible for many technical challenges. The solutions for the lighting changes are embedded in the background elimination part of the algorithm. For several cameras, where the lighting change is occurring constantly we also added pre-processing modules, e.g. gamma correction module, to the algorithm. Some of these problems are shown in the following figures (Fig. 1, 2, and 3).

Figure 1. Challenge examples; shadow, reflection, changing background

Figure 2. Challenge examples. Left: Shadow in low saturation areas. Right: Sudden lighting changes.

The background may change for several reasons. In terms of temporal characteristics, these can be classified as repetitive and constant changes. For instance, movement of moving signage cause pixels/blocks having different background models in a repetitive way while standing passengers in front of restrooms change the background model during a longer time interval. Different solutions have been proposed to address this issue. One drawback of these algorithms is that it is not easy to determine the temporal model for background elimination in terms of absolute times. Our objective is to model a change in the background scene by classifying the change according to the duration the change occurs and the temporal characteristics of the change. For this purpose, two time parameters are used: "duration and forgetting factor". Duration determines the number of frames the pixel/block is assigned to a background model. Forgetting factor determines the last time the pixel/block is assigned to a background model. In repetitive models, the duration of two background models are similar while the forgetting factor varies repetitively for these two models. In relatively constant changes, the duration gets longer and forgetting factor smaller for the new background model while the forgetting factor of other background models for the corresponding area gets larger and the duration remains constant. For example, in one application, the objective may be to classify standing passengers as foreground objects to gather information such as how long a passenger spends at a certain location while in another application passengers standing at another location may be considered as background objects. Calibration of each camera is another challenge in a crowded train station. There are several calibration algorithms in the literature, however, a detailed calibration algorithm that takes all the camera parameters into account is hard to find. The camera position and the topology of the train station; escalators, stairs, ramps make the calibration a more challenging problem. Tracking is affected by occlusion. Especially in a crowded train station, tracking multiple persons for a period of time, is a very difficult task. The algorithm should be able to find the gesture of the person(s) during this tracking period and identify different gestures of the same person. We developed a graph based tracking algorithm that solves several occlusion problems and tracks the individuals.

Figure 3. Lighting changes, reflections.

III. SOLUTIONS

Fig. 4 shows the general algorithm for gesture recognition. The solutions for the above mentioned problems are embedded in several algorithm blocks. This section covers these blocks as well as solutions for the problems.

A. Background Elimination

A multivariate background model uses color and temporal attributes to extract foreground regions. Several sub-algorithms including but not limited to shadow elimination, flicker elimination, background adaptation under lighting/background changes, etc. are performed at this stage. As we mentioned above, some cameras with constant lighting changes, e.g. platform cameras facing the railroad directly, need pre-processing steps. A combination of these algorithms is implemented as a pre-processing

module; gamma correction, histogram equalization, using different color spaces, etc.

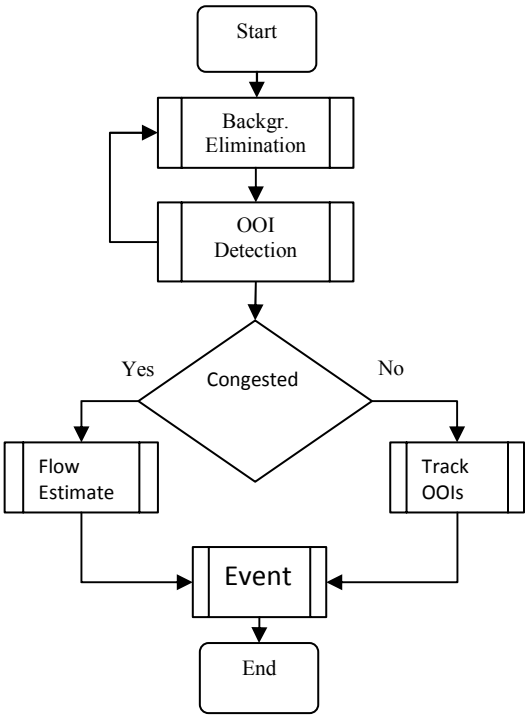

Figure 4. Algorithm

There are different ways of eliminating flicker effect, finding the flicker frequency or smoothing in time are some of these solutions. We used a de-flickering algorithm that smoothes the illumination component of the fluorescent light. Most of the lighting changes are corrected during the background elimination part described below. Background elimination starts with initialization and training. The training process is performed for each frame until the training criteria are satisfied. The statistics are computed for each pixel (or block) at each frame. The statistics of the background models consist of the mean and covariance of the model components and the duration of the background model (i.e. the number of frames where the pixel/block is assigned to the corresponding background model). Furthermore the forgetting factor is computed and its change is observed. Each pixel/block may have more than one background model and is assigned to a background model if the comparison criteria based on the color components is satisfied.

Note that during the initial training part, object detection and tracking are not performed. The training part of the background modeling enables learning the scene to improve the reliability and robustness of detection and tracking algorithms. The training is based on the statistics of the background models, e.g. comparing the covariance and the duration of highest weighted background model with predetermined values. The objective is to eliminate any static foreground object. The adaptation phase continues to distinguish temporary and permanent changes in the scene. At the end of training, if the background model has a high covariance of the color components and/or has small duration, it is eliminated. During the training phase, some foreground objects may be static for some time. If the object is static for a long duration during the training phase, it is assumed as a background object. Therefore, the training duration defines the criteria to distinguish temporary and permanent objects. After the training, the background model is updating by using exponential moving average with forgetting factor. If the forgetting factor is large, the new update is dominant. Otherwise, the average part is dominant.

Each background model is updated when the pixel/block is classified as the corresponding background area. If the pixel/block is classified as the foreground object, a new temporary background model is computed and the model is updated in the consecutive frames. If this temporary background model has covariance smaller than a predetermined threshold and duration longer than a predetermined threshold, it is assigned as a new background model. The background models for each pixel are eliminated if the forgetting factor is higher than a threshold. It is the objective of this system to have a flexible model for background-foreground classification. For instance, a surveillance system at a railway station may aim to monitor only people while everything else is considered as background. For this purpose a feedback from the object detection algorithm is used to classify other objects in the scene as background objects.

Furthermore, the object detection part is based on modeling combinations of different segments. A segment next to the object of interest may be misclassified as foreground region. If this segment is not classified as a part of the object of interest, the feedback is used to classify the segment as a background object. This feature helps to improve cases such as shadow removal. The output of this background elimination step is further processed by an additional step that combines connected component and morphological operations.

B. Finding Objects of Interest

The result of background elimination algorithm is used with high-level knowledge of the scene to extract region of interest where the motion analysis is done. The high level knowledge may be introduced by the user as a masking process or by performing a priori training to detect predetermined object of interest in a scene. The latter is done automatically to label the regions by using similar scenes. For example, monitoring areas in a train station may consist of platform and railway areas. The object detection may be done only in the predefined regions of interest. Different events are then defined according to the object detection results in different regions of interest.

OOI detection finds the existence of objects by using multiple algorithms, for example, moment invariants of DCT blocks is used in this block, a similar method is given in [4], [5] for human detection. Next block in Figure 4 decides the switching between flow detection and individual object tracking based on the OOI number. The user can set this threshold, e.g. if the OOI number is larger than 20 go to flow analysis block. The output of this part enables estimation of congestion level in the scene. Depending on the congestion level in the scene (or parts of the scene), group or individual motion analysis are performed. Different applications may be implemented afterwards. For example, flow estimation algorithm described in the next section may be used together to give an alarm if the standing person number per area exceeds a threshold. To compute the congestion level, in each block, the areas where OOI is detected are marked as one. If the ratio of areas marked as one to the total areas is greater than a predetermined threshold, it is assumed that the area is congested and the level of congestion per area is computed. Note that the block sizes are adapted according to camera calibration results.

C. Flow Analysis

Flow analysis is performed when the video shot includes a large number of people. Flow analysis requires performing numerical differentiation of sets of pixels to determine optical flow gradients. Optical flow is computed only on sets of foreground pixels. Once the basic flow per set of pixels is computed, the resulting motion vectors are then grouped to find larger features of motion, such as large groups of people who move together. After grouping, the motion vectors are analyzed to detect such events. Flow analysis combined with the patented OOI detection by using the moment invariants enables the user to find the direction of the crowd (8-directions), the number of people, the moving/standing percentage, crowd densities (number of people/area), flow rate, etc. at different locations of the scene for different resolution levels. Flow analysis does not require detailed analysis of the individual due to moment invariants which enables the user to detect human from his upper body.

D. Tracking

Tracking is achieved through graph matching algorithm by using low level attributes to high-level semantics. The probabilistic matching of OOIs to the models allows the system to perform well under heavy occlusion. Event detection is application dependent. Hidden Markov Models, neural networks and several other algorithms are tested and implemented for different applications. Some of the events that the current system can detect are walking/standing, falling, bending, fighting, unattended object, etc.

The tracking algorithm in this part is based on shape, color and motion patterns of object of interests (Fig. 5). The aim is to use simple and fast algorithms to match low-level image features of segments in different frames while using a sophisticated algorithm to combine/segment the regions to connect them to high-level attributes. The outcome of this algorithm may be used for motion analysis including articulated movements, gestures, rigid movements for event classification. The objective of this system is to provide robust and reliable tracking of OOIs even in the case of occlusion and view changes.

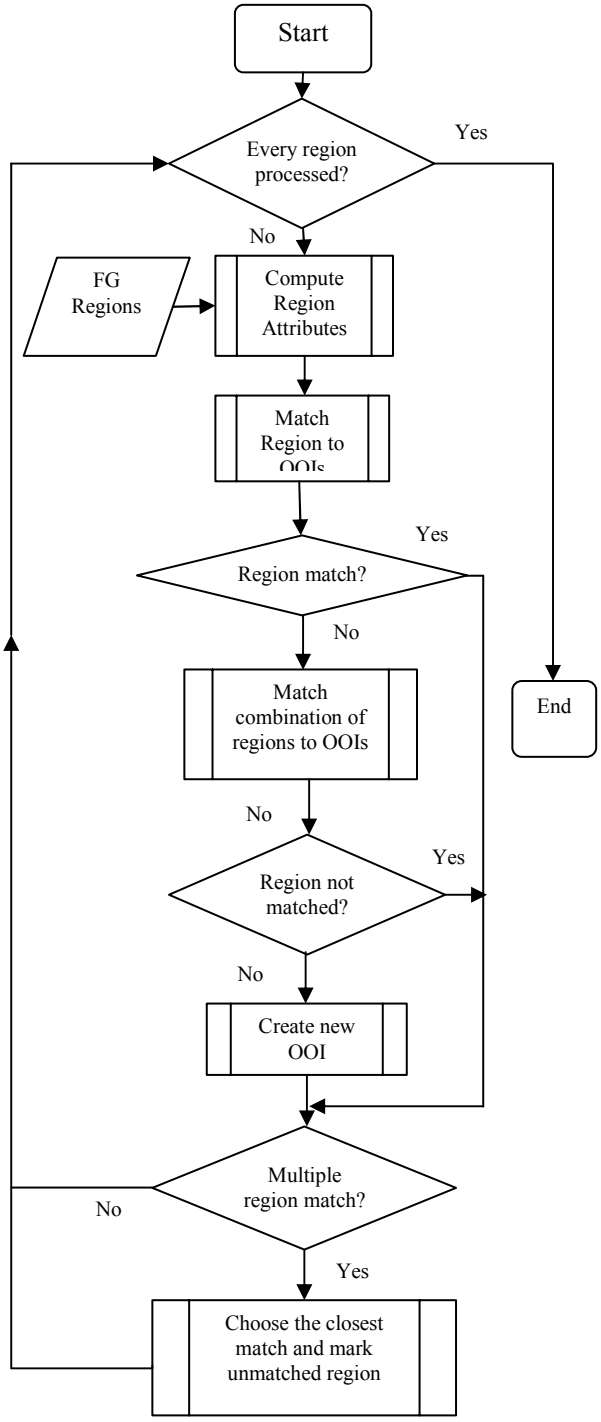

Figure 5. Tracking Process

OOIs in an active tracking list are tracked over a period of time. There are several options to remove OOIs from the active tracking list. In one option, if the OOI is assumed to

be out of the view of the monitoring area, it is no more maintained in the tracked objects table if the OOI is not detected for a pre-determined number of frames. Furthermore, a high level knowledge of the scene will be used for this purpose (e.g., estimating the enter/exit areas of the scene by using the first frame and last frame locations of detected OOIs). However, the list may keep the OOI's information even when object is out of the view area for two cases: 1) When the same OOI is assumed to enter/exit the area of view (e.g., when people id's are kept based on face recognition or other identity based recognition algorithms when high resolution of frames are available). 2) When coordination of multiple cameras are implemented where an OOI exits the view area of a camera and enters that of another neighbor camera.

E. Event Detection

Event detection is based on certain rule sets and specs provided by the railroad company. Some of the events and gestures that need to be detected are standing, walking, bending, fighting, falling, drunk walking and unattended object finding. The system is trained for each of the gestures and HMMs are used as basis for gesture recognition. Besides HMMs the system uses a combination of 3D information, body proportions changes during tracking interval, aspect ratio changes as well as interaction of OOIs. For example the rule set for fighting needs several gesture sets of different OOIs, on the other hand, a drunk person rules set is dependent on gestures of a single OOI for several tracking periods. Raising an alarm depends on this stage, for example detecting drunk person won't raise any alarm, but detecting a drunk person and falling will trigger the alarm flag.

F. Hardware

Converting floating point calculations to fixed point calculations was the major challenge at this stage. Second-generation software runs on Linux platforms in real-time. The third generation software is adapted to run on DaVinci video processing boards with TMS320C64 fixed point DSP. Right now, the systems in Tokyo train stations are running on DaVinci boards. The fourth generation software will run on custom designed hardware designed by the railway company itself. We also carried parts of the algorithm on FPGA platform and the reader can find details of this effort in our paper [6].

The output; ID of the individual, his/her gesture, position in 3D, speed of movement (in meters/frames), unattended object, person's height and width (in meters), number of people, flow direction of people, etc., can be sent to a remote location with an IP address. The system is used for indoor and outdoor train platforms as well as inside the stations. A platform, e.g. an area of the size 200 meters by 50 meters is usually covered by one camera. Each camera is connected to a video processing board. Oblique views allow a single camera to cover a larger area, reducing the number of cameras required. The deployed system passed 9 phases of evaluation. Each phase has been evaluated by Yokogawa and railway company and Verificon adjusted its algorithm and developed new algorithms to achieve the pre-set specifications by Yokogawa. The software also passed several multi-month test phases that are evaluated by the railway company. Each research phase brought new improvement to the detection rate and each phase's output is compared with the previous phase's output for the same sequences. After Phase 9, the system achieved a correct gesture detection rate of 80 percent. This rate is dependent on the scene and gestures that needs to be detected.

IV. RESULTS

The results in this section are shown on different videos obtained from two locations other than train stations. Due to security issues and disclosure agreements we show some results on the videos obtained from a jewelry store and an art exhibition.

Figure 6. Top view camera results. The system detects "walking" gesture of individuals and tracks them with direction information.

The system has been tested in several stations around Tokyo area. Tokyo, Ueno, Yokohama stations are some of the locations where the railway company tested the system before deployment. Each test ran for 3-6 months after each major change/improvement in the algorithm based on the research phase. There were 9 research phases during development. The output format changes based on the customer needs. Person number, time, his-her gesture, number of people at a specific location, standing-moving person number are some of the outputs sent to the command center or to a specified network address. The reason of choosing an art gallery and a jewelry store for our tests is to show some of the similar problems that we faced in the train stations, e.g. occlusion, lighting changes, glare, reflection in a more controlled environment. Although the gesture detection rates get higher in these controlled environments for particular gestures, e.g. standing, walking (around 90 per cent correct detection); it is hard to make a direct comparison for other gestures, e.g. falling, drunk walking since the rule sets in these environments are different.

Figure 7. Ellipse representation of tracked visitors in an art gallery.

Fig. 6 shows an example frame from top view. The output arrows show the 8-directional moving direction of the visitors in an art exhibition. Fig. 7 shows the same art gallery from different cameras where the tracked visitors are represented by ellipses. Although the occlusion is not as heavy as in a train station, the frames are good examples for how the algorithm is responding under occlusion. Fig. 8 and 9 show tracking results in a jewelry store under reflection, glare, shadow, and lighting changes. Fig. 10 is a picture that we took in one of the stations. The cameras are equipped with our smart camera algorithm.

Figure 8. Customer tracking in a jewelry store.

Figure 9. Employee tracking in a jewelry store.

Figure 10. Train station cameras equipped with the smart camera algorithm.

V. CONCLUSIONS

In this paper we described some challenges we faced during development and deployment of a smart camera technology based surveillance system. We described our solutions to these problems which may also help to other researchers in the area and give an idea about similar systems.

REFERENCES

[1] M. Rodriguez, S.Ali, and T. Kanade, "Tracking in Unstructured Crowded Scenes," 12th IEEE Int. Conf. on Computer Vision, 2009.

[2] S. Ali and M. Shah, "Floor Fields for Tracking in High Density Crowd Scenes," ECCV, 2008.

[3] S.M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," ECCV, 2006.

[4] B. Ozer, W. Wolf, "Human Detection in Compressed Domain," IEEE International Conference on Image Processing 2001, Thessaloniki, Greece, October 2001.

[5] C.H. Lin, T. Lu, W. Wolf, B. Ozer, "A peer to peer application for real-time gesture recognition," ICME 2004.

[6] Schlessman, J.; Cheng-Yao Chen; Wolf, W.; Ozer, B.; Fujino, K.; Itoh, K., "Hardware/Software Co-Design of an FPGA-based Embedded Tracking System", Computer Vision and Pattern Recognition Workshop, June 2006.

[7] B. Ozer, T. Lu, W. Wolf, "Design of a real-time gesture recognition system: high performance through algorithms and software", IEEE Signal Processing Magazine, Volume 22, Issue 3, May 2005 Page(s):57 – 64.