Automatically Detecting and Tracking People Walking Through a Transparent Door with Vision

Nathalie M. El Nabbout, John S. Zelek, and David Clausi

Abstract

There has been a growth in demand for surveillance equipment to monitor people in indoor as well as outdoor environments. Furthermore, using guards to watch surveillance screens all the time is highly inefficient and thus automation of human monitoring can be more accurate and produce cost savings. The problem is challenging if we choose to use a passive non-invasive sensor such as vision. The specific problem we investigate is tracking people through a sliding glass door. This is challenging because of the transparent door and both the door and person are moving. The method we have chosen consists of tracking coherent motion field clusters. The video frames are preprocessed, corner features are extracted and matched over frames, and the background trajectories are learnt. Finally, the test sequences are processed to obtain the trajectories of the various image features and those are classified based on the background model into foreground and background trajectories. The proposed method was tested on a set of real data with varying scenarios, and illumination as well as noise changes with a success rate approaching 95% correct classification into either background or foreground even if the tracker lost track of the entering person.

1. Introduction

In the recent past, there has been considerable focus in advancing algorithms in surveillance applications concerning monitoring moving objects such as pedestrians and cars. Such applications mostly deal with real-life environments where conditions are constantly changing. However, obtaining moving objects from dynamic backgrounds is a very difficult problem; thus most researchers simplify the problem by assuming that the background is stationary. This assumption permits the use of statistical techniques for background modeling which results in effective differentiation of a foreground moving object from the constant background [1]. This assumption, though, does not hold in all outdoor and indoor environments.

The main goal of this project is to implement a computer vision technique to solve the problem of differentiation between a human entering the store (foreground) and the dynamic movement of the store's sliding door (background) and subsequently detecting and tracking the person. Most stores have a surveillance camera mounted facing the store's main entrance to monitor who is entering. In this particular problem, the assumption is that the store keepers are only concerned by who enters the store and not by who leaves it.

2. Related Work

Among the most common methods of motion extraction is by using background subtraction and then tracking the remaining scene. Background subtraction can take place via a pixel-to-pixel fashion by subtracting the current image from a reference image in the case of static backgrounds [2]. However, in the case of a very dynamic overlapping background such as the one addressed in this paper, a different approach is necessary.

Yang and Ahuja [3] present an algorithm for extracting and classifying two-dimensional hand gesture motion based on motion trajectories. Salient features are extracted in their algorithm and represented by color and geometry. Those regions are later used to generate trajectories that describe the dynamic characteristics of the hand gestures [3]. Finally, a neural network is used to learn and classify the hand gestures into each of the 40 different American Sign Language hand gestures with the success rate of around 96% based on the test sets used. However, salient features are not always easy to detect when there is a dynamic background.

Zhu, Avidan, and Cheng [1] propose a corner-based background model that allows the detection of moving-

978-0-7695-3153-3/08 $25.00 © 2008 IEEE
DOI 10.1109/CRV.2008.28

objects in dynamic backgrounds. Their algorithm undergoes three major steps. First, the Harris Corner Detector is used to obtain the feature points representing each frame of their image sequence. The Harris corner points are then represented as SIFT-like descriptors. Thus, the SIFT-like features are used to model the entire scene instead of the foreground moving objects [1]. Since SIFT features are illumination, rotation and translation invariant, any slight variations caused by natural phenomena such as ripples or trees swaying would be tolerated by SIFT descriptors. Based on this resulting feature set, the image sequence is classified into foreground and background.

In this paper, a new approach is proposed by modeling the background motion instead of the human motion and then classifying all motions detected in the image sequence into either background or foreground.

3. Methodology

In this section, the details of the implemented algorithm are discussed. Data sets used are described, methods for preprocessing images are explained, and background / foreground modeling is clarified.

3.1. Data Sets

The data set used in this project mainly consists of five-minute video sequences of a sliding glass door in a store. The door only opens when people are close to enter and this creates the problem of extracting the foreground (person) from a dynamic background (sliding door). Figure 1 shows sample data frames.

Figure 1: Both images are samples from data sets used.

3.2. Method Implemented

The method used to extract the foreground person from the background is based on four major steps: preprocessing the image frames, extracting the interest points, modeling the transparent door dynamic background trajectories by learning the movement of door's corner points, and finally classifying the different trajectories into foreground or background.

3.2.1. Image Preprocessing. As in every real life scenario, video sequences tend to contain a lot of clutter and additional excessive irrelevant information that should be removed because it conflicts with the useful data that needs to be extracted from the image. In order to do that, erosion and dilation using vertical line structuring elements were applied in order to accentuate the black edges of the door and remove unnecessary details such as the shelf details on the right side of the images. This was followed by a close operator with a disk structuring element of size of 4 pixels to remove any holes remaining. This also serves to emphasize the person's size making humans larger objects to detect. Figure 2 illustrates the result.

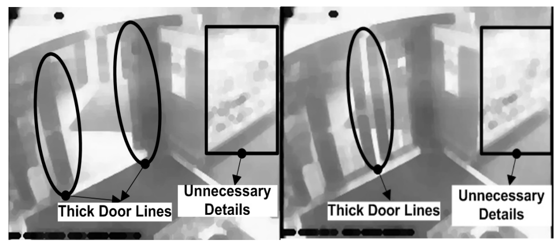

Figure 2: Preprocessed images showing the doors became thick lines and shelves' useless details were removed.

3.2.2. Interest Point Extraction. Features that need to represent the background should be limited yet be able to represent a door. The simplest manner to represent a door is by using corner points to illustrate the moving edges of the door. Those corner points are extracted and matched from one image to another to obtain door movement trajectories. Therefore, for training the classifier, point behavior is learned – explained in following sections - whereas in test images, points throughout the image are tested to see whether their behavior falls under the category of the sliding door or not. A corner point can represent any object that is entering through the door. The objective of this project is not to actually recognize the object as a human; instead, the goal is to identify that there is an entering object which is not the door and obtain its trajectory. This is done by testing the point related to this object against the learned-sliding door behavior – clarified in next sections.

Harris Corner Detector: Implementation

The Harris Corner Detector is known for its invariance to rotation, scale, illumination variation, and image noise [4]. The detector is essentially based on the local auto-correlation function which computes the local changes of the image regions with patches shifted by minute amounts in various directions [4,5,7].

When the Harris Corner Detector was implemented on the pre-processed frame images, many unnecessary points were detected. To decrease the number of points attained, corner strengths were extracted for each corner point and those with strengths exceeding maximum of mean corner point strength were filtered out to be used as illustrated in Figure 3.

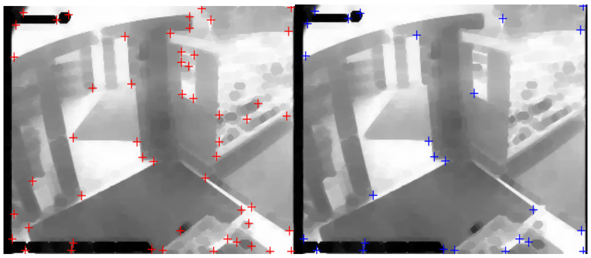

Figure 3: Left image shows all the corner points detected in the image. Right image shows corners with highest strengths

Furthermore, since the camera is stationary and mounted in front of the main entrance door, a region of interest (ROI) can be constrained once - at time of installation of the device - without affecting the practicality of the system. This can be done once irrespective of the camera position in the store. The ROI – shown in figure 4- selected is at the bottom part of the transparent sliding door because:

1) In the case of a background, it is sufficient to have the lower moving points representing the dynamic door edges to indicate that it is door since the general motion of the door's lower part is the same as the motion of the door at other parts.
2) If there is a human passing through the door, he/she should necessarily pass in the door's lower part region. Thus, any entrance should be identified by the Harris detector as additional corner points between the open door at this stage.
3) The fewer corner points that should be tracked, the simpler the classification and the better its reliability and speed of performance.

Matching Harris Corners

So far the image features (corners) were extracted from individual images. The next step is to match the different corners obtained from frame to frame. The motion between images can be assumed to be relatively small. Therefore, the location of the feature should not change widely between two consecutive frames. Each feature from the first frame that is closest to a certain feature detected in the second frame is considered as the same feature translated. The association between the points that are obtained in the first frame and in the second frame is dealt with by using a Nearest Neighbor (NN) classifier based on Euclidean distance. For strong edges, not many points are lost during the course of detection. Lost ones are replaced with points from the set of high strength.

3.2.3. Background Model Learning. Up until this point, the images were preprocessed and image features – namely corners – were detected and matched between pair of frames. The next step is to obtain a trajectory of the background points over the data sequence and learn their behavior over time. This is required to classify whether the behavior of certain trajectories, found in the test images, belong to the background or foreground model.

Obtaining Corner Trajectories

For the learning phase, the region where corner points are extracted is restricted to the lower part of the door – refer to Figure 4. Reasons for this restriction were given previously. The data set used for the background model did not include people in order not to have any interference with the learning of the background motion. The background images without people can be easily obtained as people quickly move out the area and the door shuts after they have left the scene. In case there is no image without people, the background can be obtained by applying a Gaussian mixture model [8]. In the ROI, all trajectory points that are found by the enhanced Harris corner detector (one that returns only the strongest corner points explained in previous section) are tracked (Refer to Figure 3). The trajectories of the strongest points in the ROI are shown in Figure 5. They were found by obtaining the Harris points in the first image of the sequence and matching them to the next frame's corner points based on proximity. Similarly, each consecutive two frames were matched to each other.

Figure 4: The region of interest for the background image.

Many trajectories can be found in the background images; however, many of them are just constant points with no particular useful information. Each of these trajectories has a slope that can be calculated and converted into the orientation of the line segment. Those orientations are shown by the histogram in the right image of Figure 5. Constant points over time are recorded as 0° orientation.

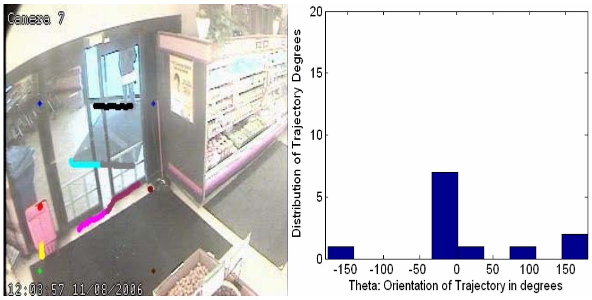

Figure 5: Plot of all trajectories in the region of interest for the purpose of learning a background model shown in left image. Right image is the histogram showing the number of trajectories per orientation in degrees.

This gave further insight into how each point is traversing from frame to another and gave a general description of the motion of the sliding door. Figure 6 illustrates this function.

The orientation of each trajectory segment is learnt over different background images and the angles of the best trajectories that represent the door are considered. Over different frames, the orientations with 0° were omitted because most of them pertained to constant points. The remaining trajectories were considered and the average of their orientations was computed. This is explained in more details in the following section.

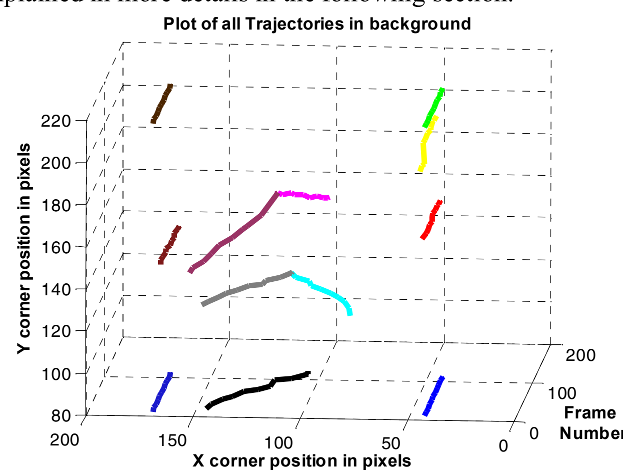

Figure 6: Plot of all trajectories in the region of interest for the purpose of learning a background model. Point (0,0,t) corresponds to the upper left corner of image Frame t.

3.2.4. Foreground/Background Classification. Up to this point, the background model has been found by obtaining the trajectories of the corner points of the door and learning their sliding behavior. The next step is to identify on any given image whether the trajectories found correspond to foreground or background.

<u>Obtaining Trajectories in Test Images</u>

The initial steps to obtain the trajectories in the test images are similar to those performed when learning the background model. The Harris corner points all over the test image are obtained and matched from frame to frame and a trajectory is obtained. Figure 7 is a sample image that shows the trajectories that are obtained over time. 3D plots corresponding to all trajectories found in the region of interest are generated as illustrated in Figure 8. Many trajectories shown in Figures 7 and 8 have a semi-constant behavior (moving only a few pixels throughout the sequence), movement similar to the door or that of the door, and movement of the person. Thus, each of those should be classified in order to come up with a viable trajectory that represents the foreground. The classification procedure and conditions are explained in the next section.

<u>Classification of Trajectories</u>

To classify each of the obtained corners in the test image into background (door) or foreground (entering person), several criteria were tested.

Figure 7: Left test image: Plot of all the trajectories at frame 50. Right test image: Plot of all trajectories at frame 99.

Criteria 1: Constancy and Extreme Randomness

As any solution to a problem, several assumptions should be made in order to simplify the problem. The first assumption is that constant points and random trajectories present no valuable information. Therefore, those points/trajectories can be removed from further consideration. The method to remove constant points is to study their displacement over time and if the scrutinized point is moving only a few pixels over the whole sequence range, then simply remove it. As for randomness, the variance of the corner points over time is taken. If the variance is large, then this is an indication that the points are not tracking anything moving at a practical pace.

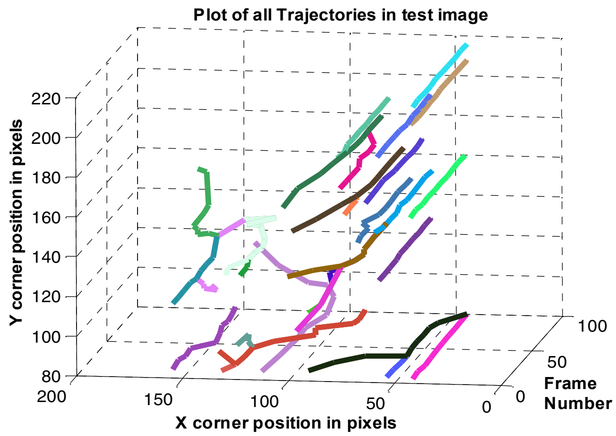

Figure 8: All the trajectories of the test image found plotted against their x and y positions in pixels and across frames. Point (0,0,t) corresponds to the upper left corner of Frame t.

Criteria 2: Backward Movement

Having removed the constant and the random points, we are left with some group of points that can either represent the door or the foreground object. Since the foreground objects of concern are actually entering objects into the store, then any trajectory that shows backward behavior indicates points moving out of the store. In order to eliminate these points, the displacement of the corner positions are computed and summed. If they are negative in the y-direction (points closer to bottom right corner of image moving to upper left corner of image), it indicates objects are moving out and therefore are eliminated from study.

Criteria 3: Comparison of Remaining Points to Background Movement

Any remaining points after criteria 1 and 2 are worth studying. The remaining points can represent either the background sliding door or an entering object. The method to compare the points is by resorting to slopes or line orientation (theta). The door's trajectories' orientations (found during the background modeling stage) are compared to the orientation of the other trajectories found in the various test images. If the test trajectories' orientations do not fall within a particular range of τ_{max} and τ_{min} of the modeled background door, then it would be classified as a foreground trajectory. In this project the range ±20° was used for τ_{max} and τ_{min}. Figure 9 shows the case when various trajectories are obtained in a test image prior to any application of criteria. Many of the trajectories that are constant, random, or belonging to the background are filtered out when the criteria are imposed. The red bars of the histogram illustrate the range that fall within the background trajectories. The green bar around 0° orientation includes many points that are constant. The rest also undergo filtering due to criteria 1 and 2. Figure 10 is the resulting trajectory when the criteria are applied. A histogram of orientations, Figures 9 and 10, showing the number of trajectories having a particular theta is plotted for pre-filtered and post-filtered cases. Figure 11 is a summary of the algorithm.

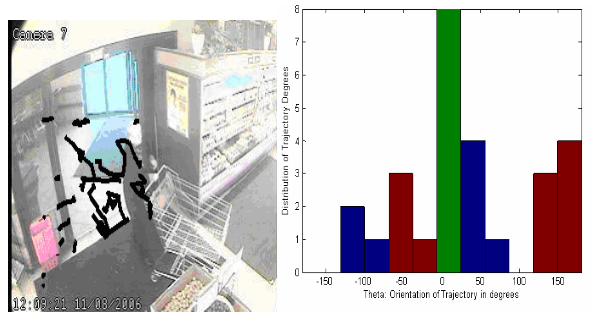

Figure 9: Left image illustrates all trajectories obtained in the region of interest prior to use of any criteria; therefore, there are trajectories that belong to background (like the ones next to the door), foreground, constant points, and random lines. Right image is a histogram of different orientations (x-axis: -180° to 180°) corresponding to the trajectories found.

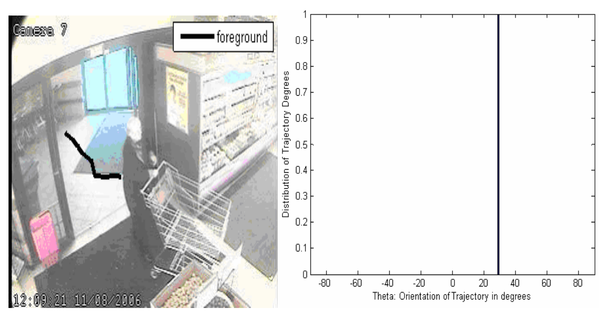

Figure 10: Left image illustrates the remaining trajectory after application of the three criteria and the classification of trajectory into foreground. The right image is a histogram of orientations (x-axis: -90° to 90°) corresponding to the trajectories found. Note that the starting point of an image is upper left corner which explains why the orientation in the image looks different than the plotted one of the right which was based on a Cartesian coordinate system.

4. Results

In this section, the results of the proposed algorithm are presented and then analyzed.

4.1. Method Results

Ninety-five different test sets each comprising of at least 100 frames with varying entering people cases and changing illumination and noise effects were performed. Several of those data sets are shown in detail followed by an overall performance evaluation. For each test set, the remaining corner points after the first two criteria are checked against the background model.

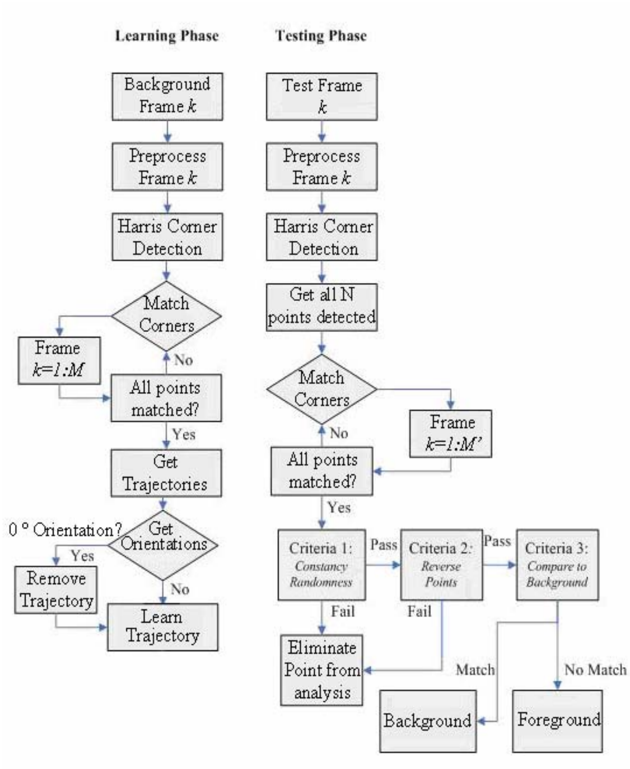

Figure 11: Shown above is the summary of the algorithm.

In case no match occurs between the remaining points and the background model, a legend is displayed on the plot stating that the trajectories belong to the foreground object; otherwise, the legend appears confirming that the is a background trajectory and thus belongs to the dynamic door modeled. The results of the test sets are shown in what follows.

4.1.1. Example Test 1. Figure 12 shows a set of frames with the first one showing all the trajectories forming without any application of criteria and the second frame demonstrating the result after the criteria were tested for and classification decision processed. It can be seen that in the case of this test set, the trajectories were correctly classified as foreground.

Figure 12: First frame shows all trajectories prior to the three criteria. The second image shows the final result.

4.1.2. Example Test 2. The second test is one that purely consists of background images. This serves to check whether the background model correctly captures the behavior of the door and if criteria applied would result in the correct trajectory classification. First frame in Figure 13 shows all the trajectories being found without use of the three criteria. The second image shows that after the criteria were applied, almost all the trajectories were removed due to being semi-constant. Only one trajectory remained classified as background. Note that the remaining trajectory is that of the left side of the door. Although the other side was not continuously detected by the Harris corner detector, the background was still correctly classified.

Figure 13: First frame shows all the background points tracked. The second image shows the results of applying all three criteria.

4.1.3. Example Test 3. In this particular test, the histogram of the lady entering the store is difficult to distinguish from the white shelves and black carpet since she is wearing a white jacket and black pants. For this reason, the tracker loses track of her at some point when the histograms become unclear. However, although the tracking does not continue until the end, the algorithm is still able to classify the short distance trajectory as belonging to the foreground. The first image in figure 14 below shows all trajectories before application of criteria and the second shows the remaining trajectories after criteria are carried out resulting in correct classification. Note that more than one trajectory remained with foreground qualities.

Figure 14: First image is before criteria application and the second is the successful classification result.

4.1.4. Example Test 4. In this test sequence, a lady enters with a shopping cart. It can be seen that the classification that took place was correct in spite of having the shopping cart complicate the tracking.

Figure 15: First image is that before criteria application and the second is the correct result.

4.1.5. Illumination tests. Different tests were performed with varying illumination. Gamma, which weighs towards brighter values when it is below 1 and towards darker values when it is above 1, was changed with values 0.3, 0.5, 0.7, 1, 1.2, 1.5, 2, and 5. The algorithm performs best between values of 0.5 to 2. At values of 0.3 some corner points disappear and tracking becomes more difficult. On the other hand, at value 5, anyone wearing black would not be distinguished as well as with lower values of gamma.

At low levels of gamma, the image becomes washed out and fewer corners can be detected and tracked. Figure 17 (left) is a misclassified image as background. In the image, the only trajectories found were similar to that of the door trajectory and there were no other sufficient corners in this image to correct the mistaken classification. Figure 17 (right) shows another failed case where no trajectory was found since the man's clothes are too similar to the background causing histogram confusion.

Figure 18 shows the overall performance of the algorithm as percentage of correct classification versus the changing gamma values.

Figure 16: Left image with Gamma =0.3. Right image Gamma =2. Both are classified correctly as foreground.

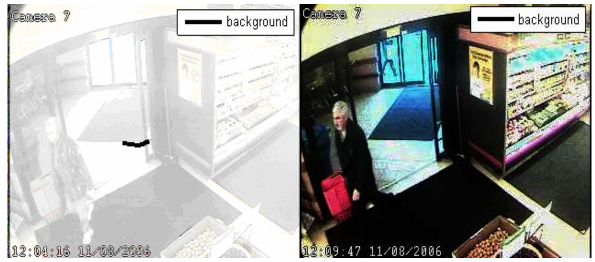

Figure 17: Failed Cases: Left image: incorrectly classified as background under gamma 0.3 Right Image: image incorrectly classified as background under gamma 5.

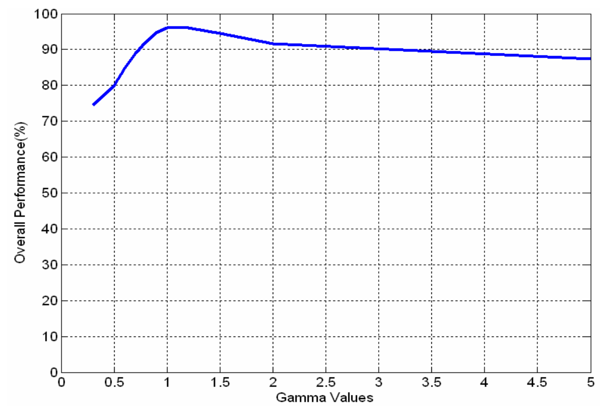

Figure 18: Overall performance of the algorithm in % (y-axis) versus the changes of gamma values: 0.3 to 5 (x-axis).

4.1.6. Noise Tests. Different tests were also performed with varying noise levels. Noise altered between three main types: Salt and Pepper noise with noise density varying from 0.01 to 0.1, Gaussian noise with zero mean and variance ranging between 0.01 and 0.1, as well as speckle noise with the multiplicative factor also changing from 0.01 to 0.1. Given that the original images are already noisy, adding even more noise ruins the image substantially. As the noise increases, more corner points are detected due to noise and the trajectories start containing more random jumps or becoming segmented. For this reason, the performance decreases as noise increases. Below are some final results of various noisy images followed by the averaged overall performance of the algorithm (percentage of correct classification) with respect to the three different noise types mentioned.

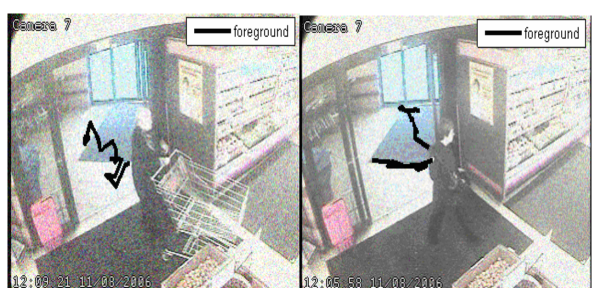

Figure 19: Left image is the result of Gaussian noise 0 mean, 0.01 variance. Right image is result of speckle 0.01.

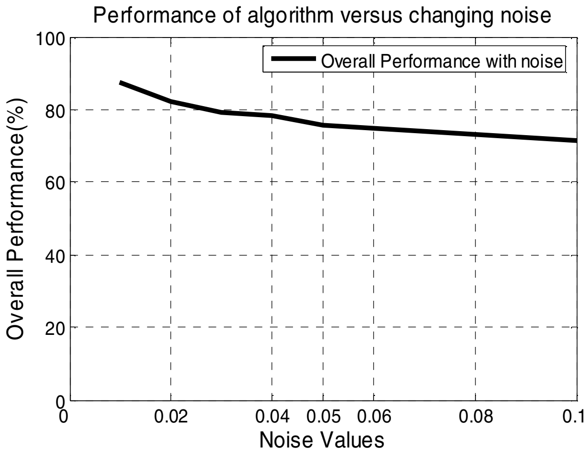

Figure 20: Overall performance of the algorithm in % (y-axis) versus increasing noise (x-axis).

4.2. Evaluation of Results

Among the 95 test sets (each of around 100 images) that were used to test algorithm, almost 95% of the trajectories were classified correctly given no major illumination or noise was in the image. With extreme illumination, performance decreased to around 85% (on average for low and high gammas). Figure 18 is a graph illustrating the overall performance of the algorithm with respect to changing gamma values. The classification of trajectories worked regardless if the person was not tracked all the way and therefore needed limited information (few frames) in order to come to a decision whether the frame was a foreground or a background image. However, tracking faced some problems with illumination and similarity of histograms to background histograms which was shown in Figure 17 (left and right images respectively). There are two cases when the classification results in an incorrect decision: when the tracking is incorrect (like in Figure 17 (left) and when there is high noise) which results in incorrect classification or no trajectory was tracked and thus there was nothing to classify like in Figure 17(right). Otherwise, even if the trajectory was short, the classifier was able to correctly classify trajectories. As for noise, the performance decreased to around 75% with extremely noisy values (Figure 20). Solutions to improve the algorithm under noise are still being worked on. Computation time for each frame is around 2 seconds. This is relatively fast for MATLAB.

5. Future Work

Although the technique used for this project was for the purpose of modeling the glass sliding door background, a similar concept can be expanded to include revolving doors, hinged-doors as well as opaque doors. Each of these doors can be represented by a certain trajectory type and if learned, can be used to distinguish between any type of store background model and a foreground object Since ROI is constrained at the time of installation, camera perspective does not pose an issue for the implementation of this technique. Furthermore, the illumination and histogram similarity problem can be solved by constructing an illumination-invariant tracker that would not lose track of the person. Finally, the project can be expanded to include counting people which would be commercially useful.

Acknowledgment

The authors would like to thank GEOIDE for their financial support.

References

[1]: Zhu, Q; Avidan, S.;Cheng, K, "Learning a Sparse, Corner-Based Representation for Background Modeling", IEEE (ICCV), Vol. 1, pp.678-685, 2005

[2]: Wang, L; Hu, W; Tan, T. "Recent developments in human motion analysis", Pattern Recognition, Vol. 36, No. 3, pp.585-601, 2003

[3]: Yang, M ; Ahuja, N. "Recognizing hand gesture using motion trajectories". IEEE (CVPR), vol. 1, pgs 466-472, June 1999

[4] Moravec, H.. "Obstacle avoidance and navigation in the real world by a seeing robot rover". Report Carnegie-Mellon University, Robotics Institute, 1980.

[5] Schmid C.; Mohr R. Bauckhage. Evaluation of interest point detectors". IJCV, 37(2):151–172, 2000.

[6]: Lowe. "Object recognition from local scale invariant Features". ICCV, 1999.

[7]: Harris C., Stephens M.. "A combined corner and edge detector". Fourth Alvey Vision Conference, pgs 147–151, 1988

[8]: Jepson A.; Black M "Mixture Models for Optical Flow Computation" CVPR, pp. 760-761