Automated Surveillance and Detection of Foreign Stationary Objects

N.G Kiewiet*, Y. Roodt†, H. Roos†, W.A Clarke†,
HyperVision Research Lab
University of Johannesburg
South Africa
*Email: kiewietn@gmail.com
†Email: {yukor,hansr,willemc}@uj.ac.za

***Abstract*—CCTV systems are frequently monitored manually by a human observer. This human observer is typically responsible for dealing with tens or hundreds of cameras at a time. Potential security threats may easily be missed by the system's human operators due to fatigue or being overwhelmed by the amount of change in the images. The timely detection of security threats is an important attribute for any security system. A robust algorithm for detecting potential threats from a surveillance video is presented.**

***Index Terms*—GPGPU, video surveillance, illumination invariant, noise rejection.**

I. Introduction

Security is a primary concern for public transport hubs such as train stations, bus stations, airport terminals and city councils. This is due to the important role that public transport has on relieving congestion in cities. The effectiveness of public transport security in this regard is related to how the traveling public perceives the security when deciding whether to make use of public transport.

Video surveillance is an important tool in increasing the security in public transport areas. Security is increased in actual terms by using surveillance footage as evidence after an event has occurred. Furthermore additional surveillance cameras affects how the traveling public perceives the increase in security.

A traditional method employed in detecting these security threats involves an operator manually monitoring many different camera feeds for any security violations. After 20-40 minutes of surveillance a human operator may suffer from a condition called "operator blindness" [1,2]. This is a phenomenon where the ability of the operator to detect security threats is severely reduced by fatigue.

This method requires a large staff of operators to function efficiently. Compounding the problem is the fact that due to public demands for increased safety, the amount of Closed Circuit Television (CCTV) systems has increased. This substantially increases the volume of data generated daily making it unfeasible to analyse manually with human operators.

Previous approaches to automated security surveillance systems include the CROwd Management with Telematic Imaging and Communication Assistance (CROMATICA) and the follow up project, PRo-active Integrated systems for Security MAnagement by Technological, Institutional and Communication Assistance (PRISMATICA) [1,3].

The security issues being detected by the computer aided system are foreign stationary objects. These are objects that remain stationary for a suspicious amount of time. An example of foreign stationary objects to be detected include loiterers, beggars, vagrants and unattended luggage. As with detecting any security concern the detection speed and accuracy is important.

The general approach to foreign object detection consists of generating a reference image which represents the background and subtracting it from an incoming frame to obtain all the foreground objects [4]. This method suffers from an oversensitivity to changes in illumination which results in false detections [5]. Analysing the movements of these foreground objects can indicate which objects are stationary [1]. Another method involves thresholding images to their level line representations and determining what pixels remain stationary for a certain history of images [3]. Both these methods are vulnerable to:

- Contrast changes: which results in false detections,
- Occlusion: due to camera position stationary objects may be occluded by a crowd or other objects, and
- Motion: small movements by seemingly stationary people, i.e. small movements of feet, hands and head occur frequently.

In Section 2 of the paper we describe the theory used in designing a foreign stationary object detector. Section 3 contains the stationary foreign object detector algorithm and its implementation. The experimental setup and its effectiveness are described in Section 4 and Section 5. In Section 6 we summarize the work and provide suggestions for future work.

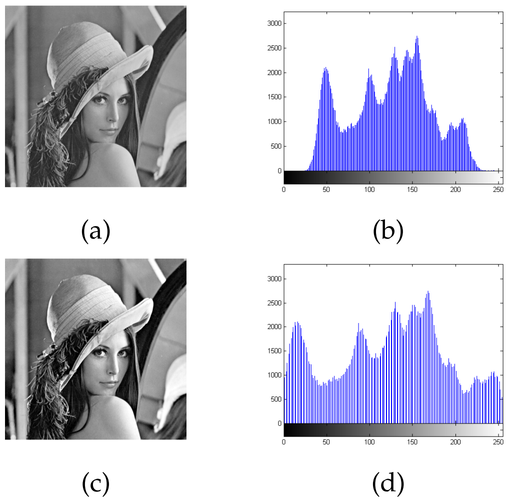


Fig. 1. (a) Original image. (b) Original image histogram. (c) Contrast enhanced image (d) Contrast enhanced image histogram.

II. Theory

A. Image Contrast Normalization

Image enhancement techniques aim to improve the visual appearance of an image. This improves the ability of the system to analyse the image information [6].

Image contrast normalization is an image enhancement technique that expands the range of intensity values in an image. This will have the effect of increasing the visual contrast between two areas of uniform densities.

The linear contrast stretching algorithm changes the contrast according to a linear function [4]. This function is given by Eq. 1 for grayscale images:

$$I(x, y) = 255 \times (I_0(x, y) - min)/(max - min) \tag{1}$$

Where,

- $I(x, y)$ denotes the new intensity value at coordinate (x, y);
- The constant 255 represents the range of intensity levels of image I;
- $I_0(x, y)$ the original intensity value at coordinate (x, y);
- With min and max representing the intensity values at the 1^{st} and 99^{th} percentile in the image intensity histogram.

In Fig. 1 the effect of an image normalization operation on an image is shown along with the corresponding histogram. As can be seen the full range of contrast is now being used to depict the image.

B. Background Estimation with Kalman Filter

Background estimation is frequently used in many video and image processing applications. Within the context of detecting foreign stationary objects, background estimation will be used in order to implement a background subtraction scheme to obtain the foreground objects. Further processing will then be able to obtain the foreign stationary objects.

There exist 4 general assumptions when creating a reference image [7].

- Images are taken from a stationary camera with fixed focal length;
- The environment consists of mostly static objects;
- Slow variations in illumination may occur;
- Occlusion of the background by some foreground objects.

The Kalman filter consists of a set of equations that implement a predictor-corrector estimator which minimizes the error covariance [8]. The following equations represent a background estimator based on a Kalman filter approach where $B(k, p)$ represents the background intensity at time k and position p. The white zero-mean Gaussian noise representing the model error is given by $\mu(k, p)$, $I(k, p)$ is the input image intensity and $\eta(k, p)$ is the white zero-mean Gaussian noise representing measurement error.

$$B(k + 1, p) = B(k, p) + \mu(k, p) \tag{2}$$

$$I(k, p) = B(k, p) + \eta(k, p) \tag{3}$$

As can be seen from Eq. 2 and Eq. 3 a temporal pixel-wise approach is used [9]. The Kalman filter time update (Eq. 4 and Eq. 5) and measurement update (Eq. 6, 7 and 8), are shown in vector form below.

$$\mathbf{x}_k^- = \mathbf{F}\mathbf{x}_{k-1} \tag{4}$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F} + \mathbf{Q} \tag{5}$$

$$\mathbf{L}_k = \mathbf{P}_k^-\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{6}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{L}_k\left(z_k - \mathbf{H}\mathbf{x}_k^-\right) \tag{7}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{L}_k\mathbf{H})\,\mathbf{P}_k^- \tag{8}$$

Where,

- $\mathbf{x}_k^-$ represents the *a priori* background estimate at time k;
- $\mathbf{F}$, $\mathbf{H}$ is the system and measurement matrices respectively;
- The filter gain is given by $\mathbf{L}_k$;
- $\mathbf{P}_k^-$ represents the *a priori* error covariance matrix;
- The measured input is given by z_k;
- The process noise covariance and measurement noise covariance is given by $\mathbf{Q}$ and $\mathbf{R}$ respectively;
- $\mathbf{x}_k$ and $\mathbf{P}_k$ represents the *a posteriori* background estimate and error covariance matrices respectively, and
- $\mathbf{I}$ is the identity matrix.

Seeing as this is a recursive structure to the solution no data storage is required, this means that the system produces a new estimate of the background as soon as another input image is available.

Before a filter update operation occurs, the residual $z_k - \mathbf{H}\mathbf{x}_k^-$, is checked. If this residual is greater than a threshold value a sudden change occurred in the scene. The threshold value for the residual is given by:

$$\epsilon(k, p) = \gamma \sqrt{\Delta(k)} \tag{9}$$

The value γ is determined by the confidence interval given by the χ^2 test for statistical significance of $I(k, p) = \hat{B}(k-1, p)$. The residual error covariance matrix is represented by $\Delta(k)$. When the residual is greater than the above threshold the filter becomes inconsistent with its assumed statistics and the measurement is rejected. Equation 7 is used to estimate the background with the following gain.

$$L_k = P_{00}\left(P_{00}^2 + V\right)^{-1} \tag{10}$$

Where L_k is the filter gain, P_{00} is the first element in the error covariance matrix and V is the square of the residual. The measurement and process noise covariance are assumed to be constant.

C. Accumulator

A method for filtering objects in a scene based on the amount of time they existed in the scene is needed to detect foreign stationary objects.

Accumulating a history of images according to the following formula will satisfy the above condition,

$$A(k) = w \times A(k-1) + (1-w) \times I(k) \tag{11}$$

Where,

- $A(k)$ is the accumulated image;
- w represents a constant weight, and
- $I(k)$ is the new image to be accumulated.

The value k represents the k^th image accumulated. Equation 11 can be seen as a time domain low pass filter, where the constant w controls the rate at which new information is added to the accumulator.

From this observation it is possible to tune the accumulator according to Eq. 11 to eliminate objects not stationary for at least a particular amount of time. Subtracting the accumulator from the estimated background will result in detected foreign stationary objects.

D. Local Image Binarization

Illumination and reflectance plays a key role in the success of image segmentation using thresholding techniques. Variable thresholding based on local image properties are able to suppress factors such as noise and non-uniform illumination [4]. Using the standard deviation, σ_{xy}, and mean, ρ_{xy}, of a local neighbourhood of pixels, A_{xy}, gives information on local contrast and average intensity of that neigbourhood. A threshold value can then be calculated using the following equation:

$$F(x, y) = \begin{cases} 1 & : g(x, y) > a\sigma_{xy} + b\rho_{xy} \\ 0 & : otherwise \end{cases} \tag{12}$$

Where, $F(x, y)$ is the resulting binary image with a and b representing constants determined with *a priori* application information. The image undergoing thresholding is represented by $g(x, y)$.

TABLE I
PERFORMANCE COMPARISON BETWEEN CPU AND GPU [14,15].

	2.93GHz Intel Core i3	**ATI Radeon HD5750**
Computation	12.79 GFLOPS	1.008 TFLOPS
Memory Bandwidth	21 GB/s	73.6 GB/s
Price(2010)	\$113	\$79

E. GPGPU

GPGPU is an acronym for General Purpose Computation on Graphics Processing Units and relates to using the parallel processing ability of the GPU for non-graphics related problems. The raw processing power and low cost of modern GPUs coupled with its rapidly expanding programmability have made it an attractive platform for image processing applications [10]. The GPU also experiences a rate of computational power growth greater than that of Moore's Law for CPU's [11]. Table I compares the relative performance and cost between the CPU and GPU.

To harness the capabilities of the GPU, an understanding of its architecture is needed. The GPU is a highly parallel dedicated processing device which has a computational structure called the *graphics pipeline*. This pipeline has been recently transformed into a more flexible programmable pipeline structure which allows user defined stream programs instead of fixed-function operations [10].

An algorithm design consideration that needs to be taken into account is the fact that stream processors have gather functionality but limited scatter functionality [11]. Gather is defined as a read-operation from different memory addresses while scatter would be the writing-operation to different addresses. A method of implementing the scatter functionality is to rewrite the problem in terms of a gather operation and using the vertex processor to scatter. Read and write access to textures is only available to kernels, but outside of these kernels, data can be transferred to and from GPU and CPU. This allows for applications which are not strictly parallel to utilize the GPU for its parallel segments, transferring its output to the CPU, and continuing processing on the host computer.

III. Foreign Stationary Object Detection Algorithm

As mentioned previously the processing rate of security threats by a system is an important factor in a security system's practicality. For this reason the stream processors on a GPU are used in the implementation of the interactive foreign stationary object detector. The foreign stationary object detector will only be implemented in non-waiting areas with a stationary camera of fixed focal length. The algorithm was designed to mitigate the effect of varying illumination while being robust to the effect of noise. The system must be able to detect objects that remain stationary for at least 2 minutes. All processing was done using the GPU stream processors with the exception of part of the contrast normalization function. The contrast normalization function requires determining the image intensity histogram, which requires cooperation with the CPU. A CPU independent algorithm for obtaining the histogram does exist but was not considered to maintain simplicity [12]. The high level functional block diagram of the algorithm is given in Fig. 2.

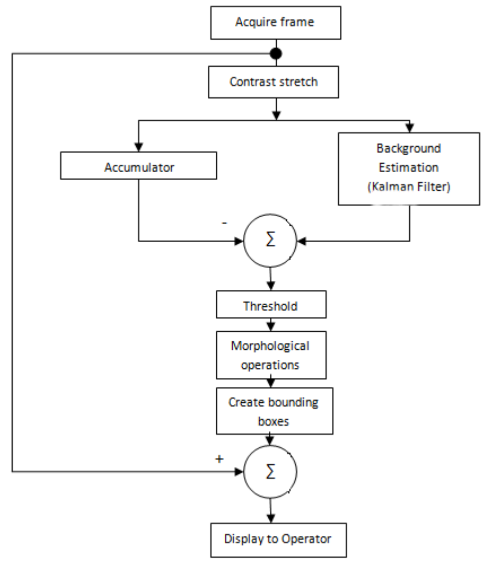


Fig. 2. Overview of the functional components of the foreign stationary object detector.

The recursive nature of the algorithm means that data storage is limited to only the preceding state of the system. When a new image becomes available it is enhanced using a linear contrast normalization operation and used to update the background estimator and accumulator. The absolute difference between the estimated background and the accumulator is transformed into a binary image according to Eq. 12. This binary image contains, if any, the objects that remained stationary for at least 2 minutes.

The morphological erosion operation is used to eliminate any small disconnected detected areas which may have resulted from noise. This binary image is then segmented by recursively calculating the maximum and minimum coordinates of each connected component. A bounding-box is then constructed from these coordinates.

When a foreign stationary object is detected and it remains stationary for an extended period of time, the background estimator will assimilate the foreign object in the next few update images. At this point the object will no longer be detected.

A. Implementation

The implementation of the stationary foreign object detector was done using the hyper-vision framework to interface with the GPU [13]. Due to the architecture of the stream processors, several passes may be required for each of the algorithm's components. A pass can be defined as the operation of a shader on one or more textures resulting in a single texture output. After each pass, data is written into a texture which may be used by the next pass.

The Kalman filter was used to estimate the background every 20 frames using a temporal pixel-wise approach. The system matrix, $\mathbf{F}$, and measurement matrix, $\mathbf{H}$, of the Kalman filter was set to $\left[\begin{smallmatrix}1,1\\0,1\end{smallmatrix}\right]$ and $[1,0]$ respectively during the implementation. The process noise covariance, $\mathbf{Q}$, and measurement noise covariance, $\mathbf{R}$, was set to a constant value of 0.01 and 200.0 respectively. The high value for the measurement noise covariance forces the Kalman filter to trust the estimated measurements, $\mathbf{x}_k^-$, more than the actual measurement, z_k. This results in the filtering out of temporary illumination changes and movement.

Updating the state of the accumulator occurs every 5 frames. The accumulator weight is based on the length of time an object needs to be stationary before it is detected. As previously mentioned, the contrast normalization function was processed both on the CPU and GPU. The histogram can be constructed by using a shader to threshold the input image and outputting the amount of pixels remaining after each pass. For n histogram buckets, n passes with the histogram shader are needed. A shader is used to implement the contrast enhancement formula per pixel.

Morphological operations are used to reduce the effect of detected noise in the binary image. Erosion is applied to the binary image containing the detected foreign stationary objects to eliminate any small disconnected areas. This is then followed by a morphological closing operation to create larger detected areas and to fill any gaps left by the thresholding function. The structuring elements in both morphological operations are 3×3 blocks.

A conceptual form of the foreign stationary object detector is given in Fig. 3. The accumulator contains the background as well as the abnormal stationary objects and any false-positives. The shaded area shows the

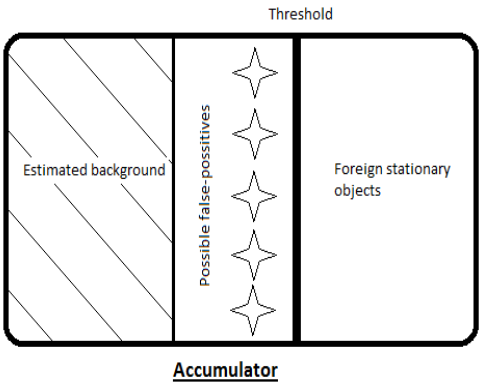


Fig. 3. Conceptual object distribution for the accumulator.

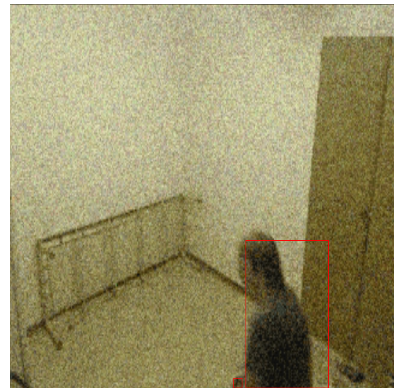
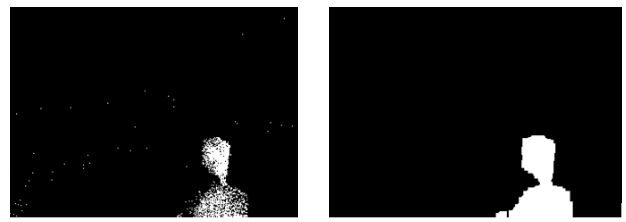


Fig. 4. Detected foreign stationary object.

TABLE II
Detections of Foreign Stationary Object Detector (FSOD)

Detector Type	No. of foreign stationary situations	Number of detections	Non detections	Erroneous detections
FSOD	28	25 (89.29%)	3 (10.71%)	1
Aubert et al.	436	427(98%)	9(2%)	0



(a) (b)

Fig. 5. (a) Unformatted binary image. (b)Morphologically enhanced binary image.

absolute difference between the estimated background and the accumulator. A threshold is applied to remove any false detections, such as objects which have not been stationary for at least two minutes.

IV. Experimental Setup

A surveillance video of a non-waiting area, such as a passageway was created to test the ability of the system to detect foreign stationary objects. The noise rejection of the system was tested by corrupting the surveillance footage with white Gaussian noise. The added noise was zero mean Gaussian with a standard deviation ranging from 10% to 70% of the maximum intensity value. Gaussian noise was chosen since it affects the images intensity value. The foreign stationary object detector uses the difference in greyscale intensity of the accumulator and estimated background to detect any objects. Adding noise that constantly changes the input images intensity values will test the systems ability to detect stationary objects.

To test the illumination invariance of the algorithm, a surveillance video of a scene with controlled variable lighting conditions was created. The total amount of erroneous and non-detections of the system was measured under different levels of illumination. The level of illumination was measured from the same position using a digital lux meter.

The processing performance of the system was measured using software timers and measuring the average processing frame rate for different resolution input videos. The test bed specifications are as follows:

- Processor: AMD Athlon X2 250 at 3.0 GHz;
- Memory: 4.0 GB DDR3;
- Operating System: Windows 7 Ultimate 64bit;
- Graphics: NVidia GTX295 with 1792 MB video memory.

V. Results

Assessment of the system is done by manually recording the approximate position in the image and the time that the object remains stationary for all foreign stationary objects in the test video set. Results of the system were then compared with the recorded manual observations to obtain the detection rate of the system.

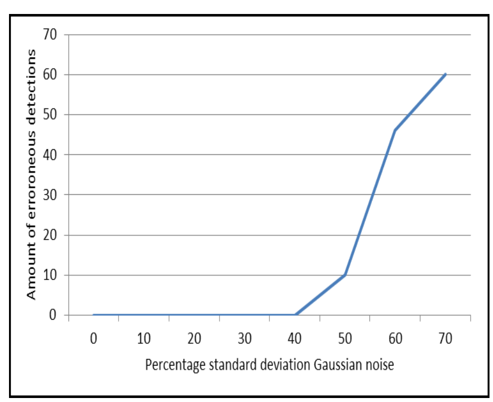


Fig. 6. Noise rejection results with increasing corruption by Gaussian noise of test video.

This is given by the total amount of foreign stationary objects detected by the system versus the total amount of manually recorded foreign stationary objects. The differences observed give the amount of non-detections (manual detections not detected by system) and the number of erroneous detections (system detected areas without real-life justification). The results from Table II were obtained from 6783 frames of surveillance video, 10 minutes.

The results in Table II show that a competitive detection rate, 89.29%, can be achieved compared to [1,3]. A detected foreign stationary object is shown in Fig. 4. The erroneous detections were caused when sharp illumination changes occurred in small areas. Global illumination changes didnt affect detection rate after the background estimator stabilized. During the noise tests the amount of erroneous detections did not increase until the test video was corrputed by Gaussian noise with standard deviation of 50% and greater as shown in Fig. 6. This is due to the morphological erosion function which eliminates any objects smaller than the structuring element. The effect of this in reducing the amount of erroneous detections can be seen when comparing the unformatted binary image and the morphological enhanced binary image shown in Fig. 5. This binary image was obtained with a 10% standard deviation in white Gaussian noise.

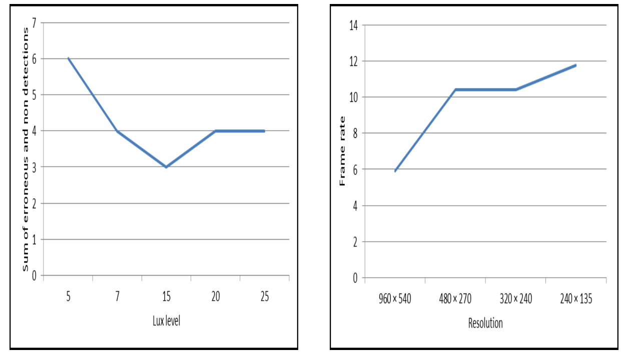
The illumination invariant test result given in Fig. 7(a), shows that the amount of errors detected by the system stabilizes after a 20 lx increase in illumination at the stationary point of measurement. The variance in the total amount of erroneous and non-detections amounted to 1.2 during the illumination invariance test. The single outlier of 6 reflects the highest error rate during the lowest illumination level of 5 lx. The small variance in total errors suggests that the system is robust in terms of varying illumination conditions. The amount of stationary objects detected remained constant throughout the test.

The processing performance at different input image resolutions is depicted in Fig. 7(b). A frame processing rate of 10Hz was achieved using a resolution of 320×240 which is higher than comparable systems [1].

VI. Conclusion

The foreign stationary object detection system described herein is able to aid operators in the detection of potential security threats based on the time that the objects remain stationary. A competitive detection rate was achieved when compared to similar systems described in [1] and [3]. The robustness of the system in the face of added white Gaussian noise was shown. The illumination invariant properties of the system was also demonstrated.

Employing a GPU in implementing the foreign stationary object detector, free's up the CPU for other logic operations. The GPU architecture is also more easily scalable than the CPU only implementations.



(a) (b)

Fig. 7. (a) Number of erroneous and non-detections compared to different illumination conditions. (b) Average processing frame rate at different resolutions.

An improved method of detecting objects based on the time an object remains stationary may require using an autoregressive model. Other future work may include the dynamic estimation of the measurement and process noise error covariances of the background estimator.

References

[1] S. A. Velastin, B. A. Boghossian, M. A. Vicencio-Silva, *A motion-based image processing system for detecting potentially dangerous in underground railway stations* Transportation research part C, 2006.

[2] F. Zhu, *A Video-based Traffic Congestion Monitoring System Using Adaptive Background Subtraction*, Second International Symposium on Electronic Commerce and Security, 2009.

[3] D. Aubert, F. Guichard, S. Bouchafa, *Time-scale change detection applied to real-time abnormal stationarity monitoring*, Real time Imaging, 2004.

[4] R. C. Gonzales, R. E. Woods, *Digital image processing*, third edition, Pearson International, 2008.

[5] M. Cheng, C. Wang, H. Wu, *Real-time Detection of Slow Moving and Temporarily Stationary Objects*, IEEE International Conference on Systems, Man, and Cybernetics, 2006.

[6] W. K. Pratt, *Digital Image Processing 4^{th} Edition*, John Wiley and Sons, Inc., Los Altos, California, 2007.

[7] M. Boninsegna, A. Bozzoli, *A tunable algorithm to update a reference image*, Signal Processing: Image Communication, 2000.

[8] G. Welch, G. Bishop, *An introduction to the Kalman filter*, IEEE SIGGRAPH 2001.

[9] J. Scott, M. A. Pusateri, D. Cornish, *Kalman filter based video background estimation*, Applied Imagery Pattern Recognition Workshop, 2009.

[10] M. Houston, *General-purpose computation on graphics hardware*, Proceedings of SIGGRAPH 2007.

[11] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, T. J. Purcell, *A survey of general purpose computation on graphics hardware*, Computer graphics forum, Volume 26-1, page 80-113, 2007.

[12] O. Fluck, S. Aharon, D. Cremers, M. Rousson, *GPU histogram Computation*, Siemens Corp. Research, SIGGRAPH, 2006.

[13] Hyper Vision Research Lab, *Hyper Vision GPU framework*, http://www.hypervisionlab.com/, Accessed 10 April 2010.

[14] Intel, *Specifications of Intel core i3 530*, Available from: http://ark.intel.com/Product.aspx?id46472, Accessed 30 October 2010.

[15] AMD, *ATI Radeon HD5750 graphics specifications*, Available from: http://amd.com/us/products/desktop/graphics/ati-radeon-hd-5000/hd-5750/Pages/ati-radeon-hd-5750-overview.aspx, Accessed 30 October 2010.