Science real and relevant conference 2010
Proceedings homepage://www.conference.csir.co.za/

Scintillation mitigation for long-range surveillance video

JP Delport[1]

[1]CSIR Defence, Peace, Safety and Security, PO Box 395, Pretoria, 0001

*Corresponding author: JP Delport, jpdelport@csir.co.za

Reference: DS07-PO-F

Abstract

Atmospheric turbulence is a naturally occurring phenomenon that can severely degrade the quality of long-range surveillance video footage. Major effects include image blurring, image warping and temporal wavering of objects in the scene. Mitigating these effects, while preserving motion not caused by turbulence, can increase the effectiveness of a camera system designed for long-range surveillance.

The parallel processing performance, high memory bandwidth and programmability of modern Graphics Processing Units (GPUs) make them ideal platforms for implementing the image processing algorithms required for this task. This work presents the results of de-blurring and de-warping algorithms, implemented on the GPU, that are approaching real-time performance on high-resolution digital surveillance video. The algorithms are being developed and implemented in a collaborative effort between Armscor, the CSIR and students at the University of Johannesburg.

1. Introduction

Long-range video surveillance footage can be adversely affected by atmospheric effects. Specifically, atmospheric turbulence, or scintillation, can cause blurring and warping in images, as can be seen in Figure 1. The figure shows a montage of parts of 16 consecutive video frames captured by a long-range surveillance camera. Apart from the low contrast, it can be seen how the image has been blurred and how the edge of the tower has been warped by atmospheric turbulence. The consecutive images were also captured in less than a second, which gives an indication of how dynamic the influence of the scintillation is.

By correcting for the blurring and warping, the effectiveness of a surveillance system can be improved. This paper focuses primarily on an implementation of a de-warping algorithm for atmospherically degraded video footage, but some results of a de-blurring algorithm are also provided. The algorithms were implemented on a Graphics Processing Unit (GPU) that has proven ideal for these types of image processing work thanks to its parallel processing performance, high memory bandwidth and programmability.

The paper is structured as follows: Section 2 gives some references to related work, Section 3 briefly describes an implemented de-blurring algorithm, Section 4 provides details and results of the de-warping algorithm, and finally, Section 5 discusses some future work and concludes the paper.

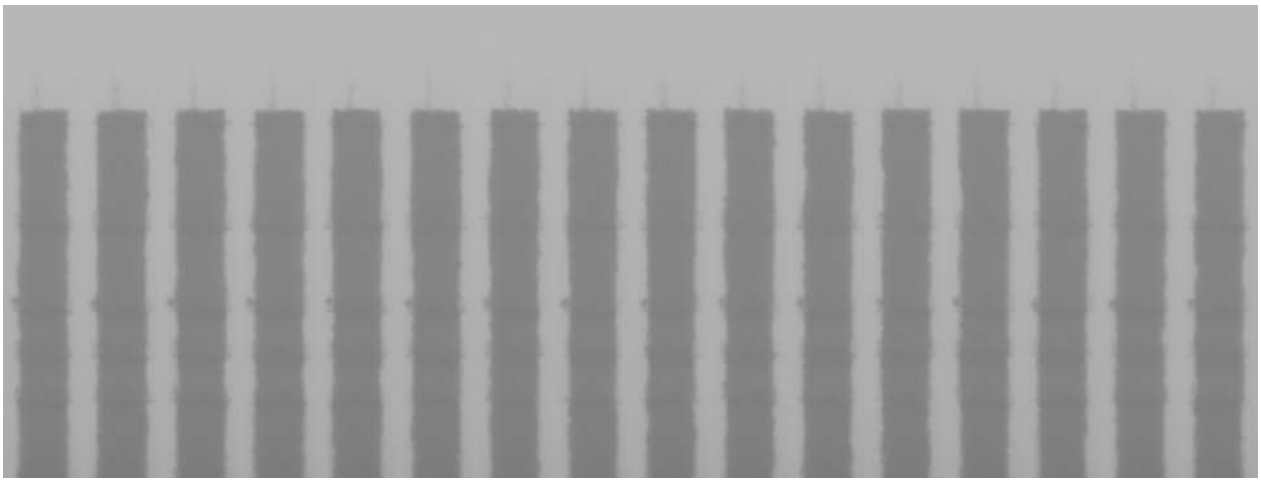

Figure 1: A montage of parts of 16 consecutive video frames captured by a long-range surveillance camera showing the dynamic influence of scintillation on the objects in the images

2. Related work

Mitigation of scintillation effects has been studied in the areas of astronomy (Li and Simske, 2009) and long-range surveillance (Robinson, 2009; Li, Smith and Mersereau, 2006; Li and Simske, 2009; Frakes, Monaco and Smith, 2001). The body of work can be divided into algorithms that operate on single images and others that operate on video data. In single images, mitigation of blurring effects is normally the goal; while in video sequences, mitigation of blurring as well as warping effects can be addressed.

Algorithms for the processing of video data can be further divided into off-line processing algorithms and those that attempt to perform correction in real time. The focus of this work is on real-time correction of live video. Further, some algorithms naturally lend themselves to parallel implementations and these algorithms can greatly benefit from the processing power available in modern GPUs.

Image pyramid processing algorithms (Adelson, Anderson, Bergen, Burt and Ogden, 1984; Strengert, Kraus and Ertl, 2006) naturally lend themselves to GPU implementations and can be used for interpolation (Marroquim, Kraus and Cavalcanti, 2007). Specifically, we create a related algorithm to interpolate a vector field.

3. De-blurring algorithm

This section presents a very brief and high-level overview of the de-blurring algorithm as implemented in Robinson (2009). The de-blurring work has been transferred to the CSIR in a knowledge transfer session with the student (Bezuidenhout, de Villiers, Delport, Bachoo, Duvenhage, Hattingh, Calitz and Jeebodh, 2010). Blur in the image can be modelled as an Optical Transfer Function (OTF) and one such model has been created by Hufnagel and Stanley (Robinson, 2009; Li et al., 2006; Li and Simske, 2009). The OTF derived by Hufnagel and Stanley is shown in Equation 1.

$$H(u,v) = e^{-\lambda(u^2+v^2)^{\frac{5}{6}}} \tag{1}$$

H represents the blur in the frequency domain as a function of the horizontal (u) and vertical (v) frequency variables. The λ parameter must be estimated (or chosen) to match the blur present in the image.

Once the OTF has been computed, it can be used in a Wiener restoration filter of the form:

$$W = \frac{1}{H}\left(\frac{|H|^2}{|H|^2 + SNR^{-1}}\right) \tag{2}$$

where W is the frequency domain Wiener filter and SNR is an approximation of the signal-to-noise ratio in the input image. The filter is applied by calculating the 2D Fast Fourier Transform (FFT) of the input image, multiplying it by the Wiener filter and then taking the inverse FFT of the result.

Figure 2 shows what can be achieved by this filter after selecting appropriate values for λ and SNR. The generation of the OTF, the FFT and its inverse, as well as the multiplication of the image and filter, have been implemented on a GPU to speed up execution.

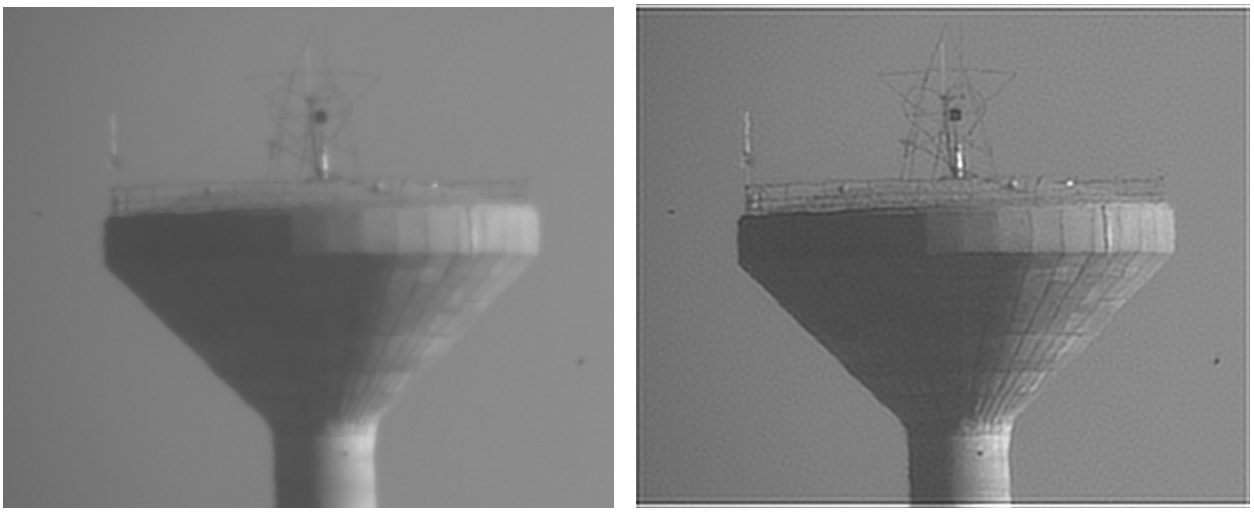

Figure 2: De-blur results: the left image shows an unprocessed incoming video frame and the right image shows the image after the application of the Wiener restoration filter

4. De-warping algorithm

Scintillation can cause quite severe warping of features in long-range surveillance images. For example, in Figure 1 and Figure 2 it can be seen how straight edges of the towers are distorted by the atmosphere.

This section describes an implementation of a specific algorithm to mitigate the warping effects of scintillation. Note that if some choices of parameters and procedures appear arbitrary, it is because they were determined in a rapid experimental process. Future work could address a more exhaustive search for alternatives.

The algorithm has four major steps, namely:

1. Find a sparse set of matching regions in a range of consecutive images.
2. Average the pixel offsets over the range of input images.
3. Convert the sparse set of matches into a match for every pixel (dense set).
4. Move each input pixel by the calculated offset.

Each of these steps is described in the following sections and thereafter some results are presented.

4.1 Region matching

Ideally, if one assumes a stationary camera observing a stationary object, a salient object feature would fall onto the same pixel of the camera sensor in every video frame. Atmospheric scintillation causes these features to move around in the image from one video frame to the next; however, it seems reasonable that a feature could be located in a constrained region around its ideal position.

The region-matching algorithm tries to find, for every pixel in the current image, a matching pixel in the previous 15 input video images. As a first step, pixel neighbours in the input image are evaluated to determine if, for a specific pixel, there are some local details surrounding it. Figure 3 shows how eight values surrounding the current pixel are sampled in image space. By sampling at pixel borders, the GPU automatically calculates an average of the adjacent pixels. The difference in intensity between the centre pixel and the eight neighbour values are calculated, and if fewer than four differences fall below a threshold, the input pixel is not considered further during the matching process.

The reasoning behind omitting pixels that fail the local detail test is two-fold. Firstly, these are normally uniform regions where de-warping would not be noticed by the user and secondly, these regions are difficult to match and generally provide false positives.

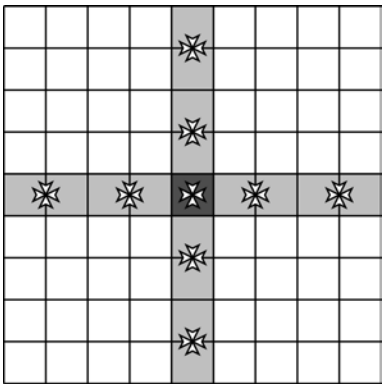

Figure 3: Neighbour sampling points (for region matching) around a centre pixel

Next, for every pixel that passed the local detail test, a window in each of the 15 previous video frames is considered and the best match in each frame is recorded. Matching is again performed using the sampling mask in Figure 3 and the sum of squared differences between the current image and an earlier image is calculated for every window position in the earlier image. The window position with the smallest sum is selected as the offset of the input pixel in the specific earlier frame. As a further refinement, pixels for which the smallest sum of squared differences are larger than a threshold can be discarded.

The output of this step is a sparse set of pixel offsets, in two dimensions, for each of the 15 previous video frames. More specifically, 15 textures (matching the input image dimensions) are created on the GPU where a pixel's red value specifies the x-offset, the green value specifies the y-offset and the blue value specifies if a valid match was determined.

4.2 Offset averaging

The 15 offset maps determined in the previous step are combined to form a single offset image (or texture). The only variable used in this step is to set a threshold for the number of valid offsets that must be present for a valid output offset to be generated. By requiring matches in a large number of earlier frames, larger movements that are not caused by scintillation can be excluded from the de-warping.

The format of the output texture is the same as that described in the previous section. Currently, a simple geometric average of incoming valid offset values is calculated for every pixel, but it is planned in future to experiment with other operators, e.g. the median of input values.

4.3 Sparse to dense offset interpolation

The processing performed so far resulted in a sparse offset map for each input pixel. The map can potentially be used as is for de-warping the input image, but then a tessellation of the geometry onto which the input image is mapped would have to be performed (because we do not have an offset for every pixel or for the vertex matching the pixel). Our approach converts the sparse offset map into a dense map by means of interpolation using an image pyramid (Adelson et al., 1984; Strengert et al., 2006; Marroquim et al., 2007) approach that fits quite well onto GPUs.

The processing is split into an *analysis* and *synthesis* phase (also called a push and pull phase). During the analysis, up to four valid input offsets are combined (by averaging valid values) into a single output value. The output plane dimensions are half those of the input plane, so we are moving up the pyramid, as shown in Figure 4. By continuing this process, we arrive at a single offset value at the top of the pyramid.

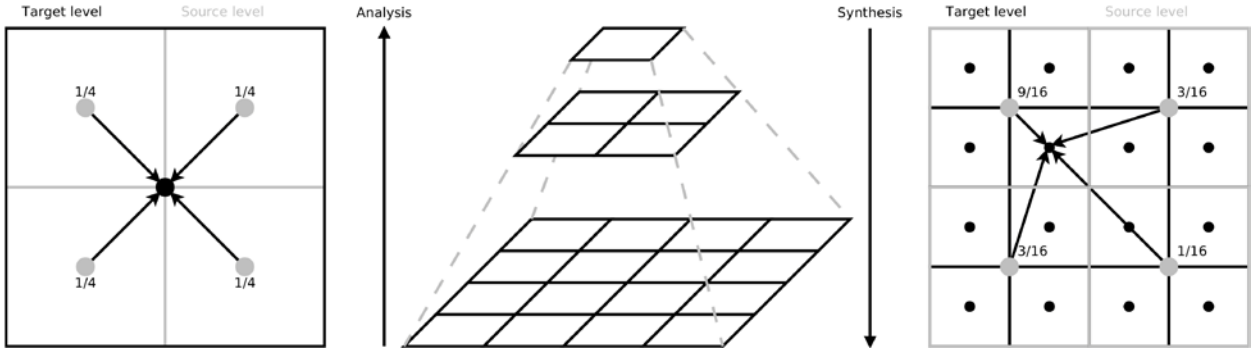

Figure 4: The analysis and synthesis phases in an image pyramid interpolation approach

It can be seen that the process would only work correctly if the input image dimensions are powers of two. To achieve this, the input image is rendered to a power of two dimensioned texture, with blank areas filled by the edge values in the input image.

The synthesis phase now generates missing values at finer resolutions (larger dimensions) using the calculated values present in the coarser resolutions (smaller dimensions), so we are moving down the pyramid, as shown in Figure 4. From Figure 4 it can be seen that an output pixel's value is determined from four input pixels. An output pixel always has four neighbours at the coarser resolution: one neighbour is always closest; two others are equally spaced but further away than the closest one, and the remaining one is the furthest away. The contributions by the neighbours are weighted according to increasing distance by values of 9/16, 3/16 and 1/16 respectively. The weight values match those used during Catmull-Clark subdivision (Marroquim et al., 2007).

4.4 Image de-warping

The output of the previous processing step is a dense map that contains the x- and y-offsets by which every input pixel should be adjusted. The map can be utilised to de-warp the image in a variety of ways, some of which include the following:

- The image can be treated as unconnected pixels and each pixel can be drawn independently at the offset specified in the map. This approach can cause tearing in the output image.

- A dense grid can be used onto which the pixels are mapped. The vertices of the grid are then manipulated using the offsets and the GPU takes care of stretching the input pixels.
- The map can be interpreted to mean that a given output pixel should be fetched from the negative offset location in the input image.

We implemented the final approach and, even though it is an approximation, it provides quite pleasing results. On the GPU, this approach translates into using the offset map during texture fetches when drawing the final image.

4.5 Results

Figure 5 shows the processed version of the input images in Figure 1. The figure shows how the edges of the tower are now much more uniform and straight. The figure can, unfortunately, not sufficiently convey the reduction in temporal wavering when the image sequence is viewed as live video.

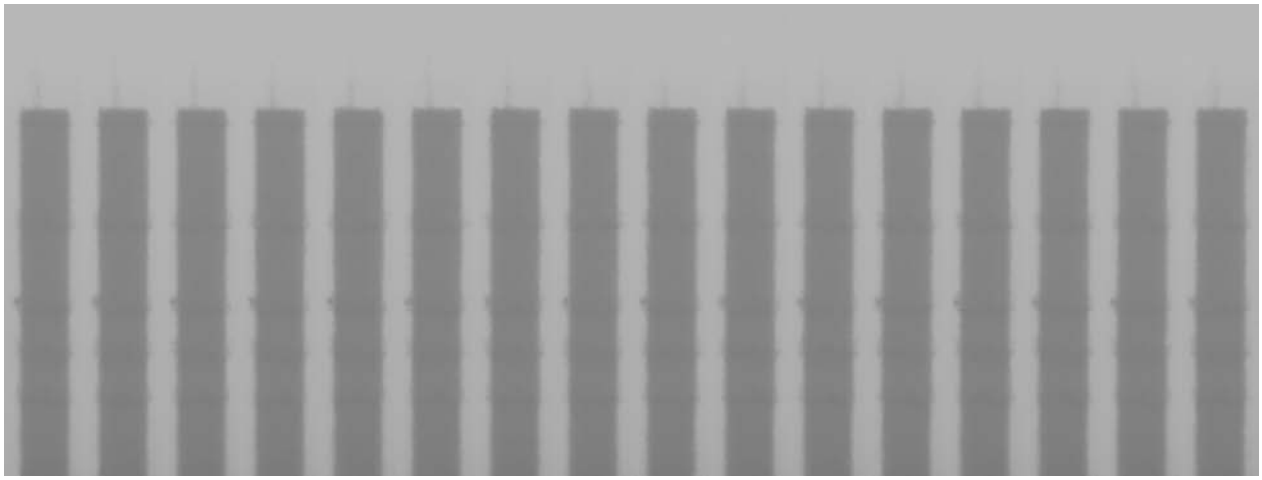

Figure 5: A montage of parts of 16 consecutive video frames captured by a long-range surveillance camera after de-warping has been applied

Figure 6 shows an input image, the de-warped version as well as the sparse map of offsets as calculated after the offset averaging processing step.

5. Conclusion and future work

This paper presented some details and results of working GPU implementations of specific de-blurring and de-warping algorithms that can be used to mitigate the effects of scintillation in long-range surveillance video footage. The algorithms form part of an experimental demonstrator and are continually being optimised and refined. The GPU implementations are approaching real-time execution speeds on monochrome megapixel-sized images captured at 20 Hz.

Many areas of future work have been identified. The parameters of the de-blurring stage can probably be estimated during execution, but currently it is foreseen that the system would allow a user to manually adjust these values, much like the contrast and brightness of an image can be adjusted.

More detailed experimentation at all stages of the de-warping algorithm can be performed. The algorithm must also be tested against a wider range of input sequences to verify whether currently selected parameters would work in a wider range of cases. Alternatives in the image de-warping step of the

algorithm also have to be investigated, especially where the output of this stage is to be used as the input to a super-resolution algorithm that requires sub-pixel accuracy.

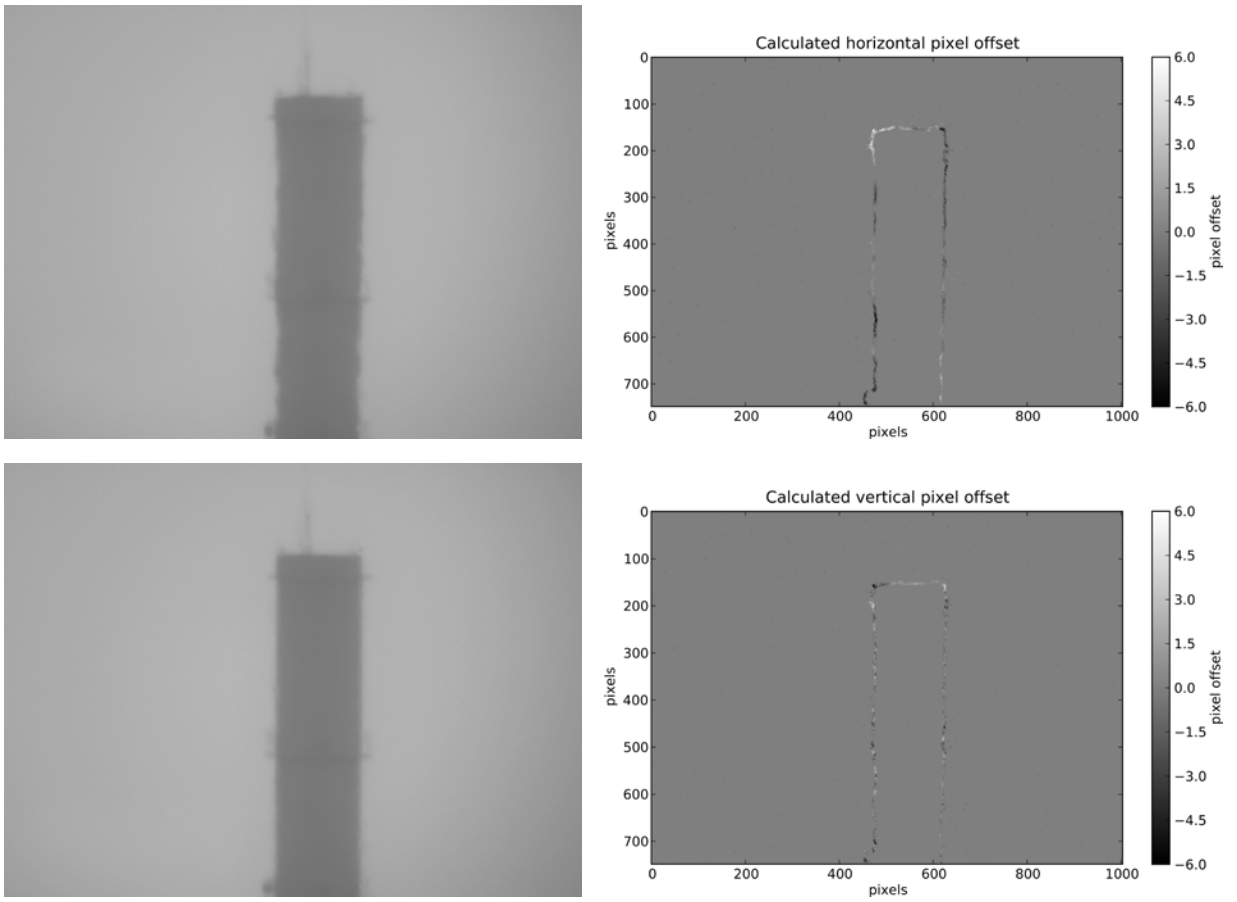

Figure 6: De-warping: the raw input image is shown at the top left; the bottom left shows the de-warped version of the input; the top right represents the calculated horizontal pixel offsets and the bottom right the calculated vertical offsets

6. Acknowledgement

Financial support for this work and for the collaboration with universities and students was provided by the Armscor LEDGER programme.

7. References

Adelson, E., Anderson, C., Bergen, J., Burt, P. and Ogden, J. (1984), 'Pyramid methods in image processing', *RCA Engineer* **29**(6), 33–41.

Bezuidenhout, D., de Villiers, J., Delport, J., Bachoo, A., Duvenhage, B., Hattingh, Z., Calitz, R. and Jeebodh, P. (2010), A proven collaboration model for impact generating research with universities, *in* 'CSIR science real and relevant'.

Frakes, D., Monaco, J. and Smith, M. (2001), Suppression of atmospheric turbulence in video using an adaptive control grid interpolation approach, *in* 'IEEE international conference on acoustics speech and signal processing', Vol. 3.

Li, D. and Simske, S. (2009), 'Atmospheric turbulence degraded image restoration by kurtosis minimization', *IEEE Geoscience and Remote Sensing Letters* **6**(2).

Li, D., Smith, M. and Mersereau, R. (2006), An adaptive model for restoration of optically distorted video frames, *in* 'Proceedings of SPIE', Vol. 6065.

Marroquim, R., Kraus, M. and Cavalcanti, P. (2007), Efficient point-based rendering using image reconstruction, *in* 'PBG'07: Proceedings of the Eurographics Symposium on Point-Based Graphics', pp. 101–108.

Robinson, P. (2009), Exploration of the mitigation of the effects of terrestrial atmospheric turbulence in long-range video surveillance, Master's thesis, University of Johannesburg.

Strengert, M., Kraus, M. and Ertl, T. (2006), Pyramid methods in GPU-based image processing, *in* 'Vision, modeling, and visualization 2006: proceedings, November 22-24, 2006, Aachen, Germany', p. 169.