A Correlation Chip for Stereo Vision

R. A. Lane, N. A. Thacker and P. A. Ivey.
Dept. of Electronic and Electrical Engineering,
University of Sheffield
Mappin Street, Sheffield, United Kingdom.

Abstract

This paper gives an example of how computer vision algorithms can be reformulated to exploit correlation based hardware without compromising the underlying principles of the algorithm. The work shows results of the "stretch-correlation" algorithm for calibrated stereo depth estimation and goes on to discuss the development of a convolution chip for implementing this algorithm. The motivation for the chip and its applicability to other computer vision algorithms is also discussed.

1 Introduction

In many cases computer vision algorithms use feature extraction as a preprocessing stage for the higher levels of processing. The justification being the flexibility and statistical properties which are inherent in, for example, edge and corner data. More specifically, the use of edge-string extraction in stereo matching algorithms is seen by many as the most robust technique for "difficult" stereo problems. Where difficult refers to the class of problem for which the grey-level data is not consistent between stereo viewpoints, and the amount of object deformation between views is large [7]. However, if real-time computer vision is to be a viable proposition for applications such as robot control, it is equally as important to have a computationally efficient solution as it is to have a statistically robust algorithm. From a computational perspective the use of high-level primitives, such as edge-strings, inevitably leads to a requirement of general purpose computer architectures to manipulate the necessary high-level data structures, thus the computational efficiency is reduced. However, greater efficiency can be achieved by exploiting the regular ordering of data in images with vector based operations.

In the case of stereo vision, correlation based approaches are used extensively by the photogrammetry community. However, these show a lack of robustness for difficult stereo problems due to the unsuitability of absolute grey-level similarity measures. Edge string based algorithms remove this problem by manipulating quantities which are more directly related to the underlying 3D structure of the scene rather than the illumination.

The aim of our work is to develop a stereo vision system which reconciles the contradictory objectives of algorithmic accuracy, robustness and computational efficiency by examining edge-string matching and reformulating it into a convolution based implementation. This has involved the development of an algorithm called "stretch-correlation" and the development of a chip to perform the vector

BMVC 1994 doi:10.5244/C.8.63

acceleratable aspects of the algorithm. The specification of the chip has included the requirements of general purpose functionality.

The development of a variety of Video Signal Processing devices [8] has been prolific in recent years, and has grown to meet the requirements of image processing functions such as image sequence encoding. Whilst in theory these devices offer great potential for implementing other algorithms, it is apparent that the requirements of this market only partially intersects the requirements of computer vision. For example, a general requirement of the mass market is that images are processed at a rate of 25-30 Hz, whereas with computer vision it is important that the numerical properties of the algorithm are preserved, at the expense of the image throughput capability if necessary.

2 Algorithm Classifications

Rationalising the requirements of a subset of target algorithms forms the initial stage of general purpose hardware design. In addition to core arithmetic operations, it is necessary to examine the data access requirements of any algorithm. For our chip this included the ability to perform all vector acceleratable aspects of the stretch-correlation algorithm. Additionally, we have also attempted to provide support for general purpose image processing functionality, based on the results of an algorithms survey [10]. A summary of part of this report is given below:

- A broad range of basic arithmetic operations (including multiplication)
- 1D and 2D accumulations with variable kernel sizes
- Efficient variable bit-length calculations

Taken alone these computational requirements justify the use of a large silicon area, highspeed, fine grain SIMD-like architecture [1] recently designed in our group. Additionally, the survey concluded that convolution can be subdivided into categories based on the locality and uniformity of data access. The following classification contains three categories in ascending order of data bandwidth:

- Image convolved with single fixed mask
- Image convolved with infrequently varying coefficients
- Image convolved with frequently varying coefficients

Whilst the first two categories are generally supported by commercial convolution chips, the third category, which includes algorithms such as arbitrary image warps, requires special consideration. In particular, for a VLSI design the only practical solution requires a large on-chip coefficient store. For this reason we decided to design a second chip with less programming flexibility, but a high coefficient bandwidth. As we will explain, the demands of our stereo vision algorithm fall within the functional domain of this processor.

3 Stretch-Correlation Algorithm

3.1 Description

Block correlation based stereo algorithms map well onto convolution based hardware, but in their simplest form provide data which is inaccurate due to the region based disparity quantisation. With the addition of window shaping and hierarchical processing [4, 5, 6] block quantisation effects can be alleviated, but the dependence on grey-level consistency between stereo views causes a lack of robustness in environments where illumination is not ideal. In contrast, edges represent the underlying three dimensional structure of the scene, and are a more reliable match primitive.

Figure 1 shows the four basic stages of the stretch-correlation stereo algorithm. Firstly image rectification is used to allow the easy application of the epipolar constraint, this requires precise camera calibration data [9]. This stage presents a major computational load. Our algorithm embodies edge matching in a correlation implementation by using preprocessing stages to enhance non-horizontal edge information whilst suppressing noise. This takes the form of gaussian smoothing the images with a 1 pixel standard deviation kernel, and taking first order horizontal differences (similar to the first stages of Canny). The correlation stage of the algorithm uses window shaping in the form of either block stretching or shearing. The enhanced image blocks are resampled through a range of "stretch" values, using linear interpolation on the gaussian smoothed images, this forms an extra search dimension in addition to the horizontal displacement. The window shaping process is demonstrated for the simplistic case of a single edge in the image block in figure 2. The purpose of the block stretching/shearing is to allow a linear disparity gradient to exist within each block thus defining a first order model of distortion between views.

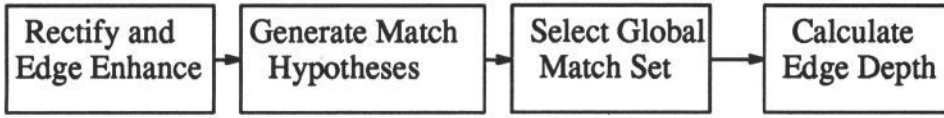

Figure 1: The Stretch Correlation Algorithm

The correlation stage of the algorithm can be seen as a hypothesis generator which works solely on the local figural consistency constraint at a block level. Each block produces a correlation surface from which an ordered list of potential matches is obtained by considering all maxima up to a threshold. The threshold is determined by placing a cut on the characteristic signal distribution for the correlation score, thus allowing the selection of a specific S/N ratio. At this stage a "loose" global support constraint based on disparity gradient (DG) is applied [7], such that a block must at least receive some support from neighbouring blocks. A DG limit of 2 is the minimum required to enforce ordering. Unsupported hypotheses are thus rejected and other hypotheses are examined. This stage requires a small amount of high-level processing.

Once a block match has been established, the depth at all edgels is calculated

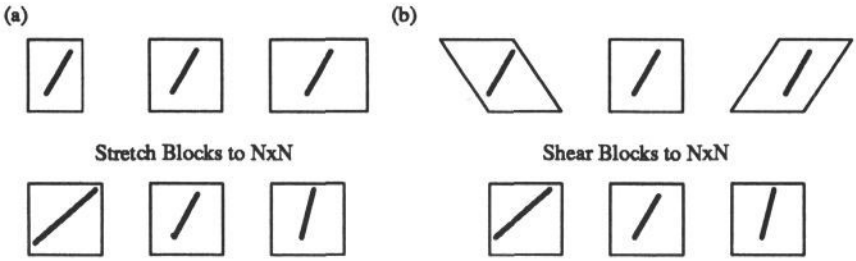

Figure 2: The Stretching and Shearing Process

using the two parameters obtained from the matching stage: horizontal disparity at block-centre and stretch/shear value from the window shaping.

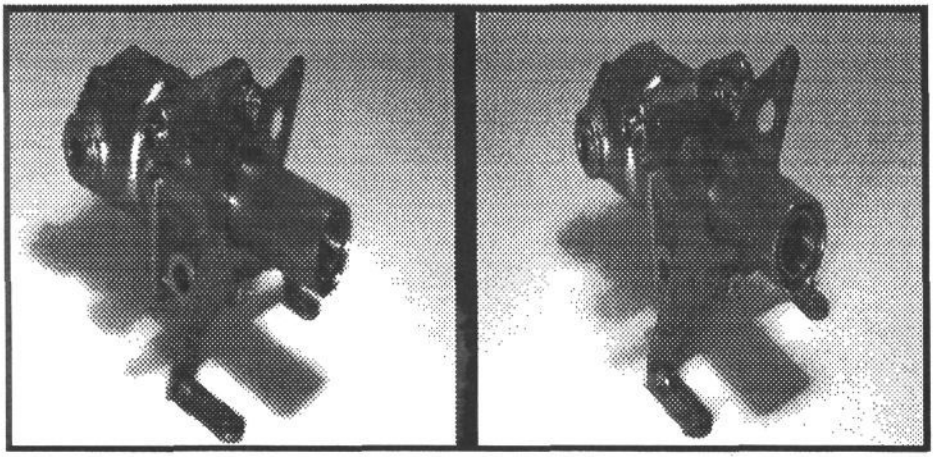

Figure 3: Shaft Assembly Image Pair

3.2 Algorithm Performance and Computational Requirements

Figure 3 shows a typical difficult problem for which the stretch-correlation algorithm is intended. Figure 4 shows a 3D reprojection of non-horizontal edge data obtained from: (a) the stretch correlation algorithm, compared to the results obtained from: (b) the PMF edge-string based algorithm. It can be seen from this qualitative data that the results of block correspondence from the stretch-correlation algorithm are comparable to a typical edge-string based algorithm in terms of the quantity of grossly incorrect data. There is also sufficient location accuracy to allow unambiguous edge string matching.

The stretch-correlation algorithm has been statistically evaluated in comparison to other correlation based techniques. The criterion used were edge location accuracy, quantity of returned edge data and disambiguation ability [3]. It was found that disambiguational ability was comparable to current Euclidean distance

Figure 4: (a) Stretch-correlation, (b) Edge-string based algorithm

methods [2] with significant improvements with respect to location accuracy. The stretch correlation algorithm returned edge data to an accuracy of 0.8 pixels RMS error, compared to nonwindow- shaping techniques which typically had a 1.1 pixel RMS error, while at the same time it returned a larger quantity of matched edge data.

This section examines the low-level manipulations of our stereo algorithm, and shows where redundancy has been exploited. The image rectification stage of the algorithm requires a perspective reprojection of pixel coordinates with sub-pixel interpolation. The perspective reprojection takes the form of eqns 1 and 2.

$$(u_w, v_w, w) = R(x, y, f_1) \tag{1}$$

$$(x_R, y_R, f_2) = (u_w f_2/w, v_w f_2/w, f_2) \tag{2}$$

where R is a rotation matrix, x, y and x_R, y_R are the original and rectified image coordinates and f_1 and f_2 are the initial and rectified camera focal lengths. Equation 2 represents a difficult reprojection to implement as a 2 pass raster scan due to the division by w (see section 4.1) and a major computational overhead. The image interpolation process is performed by resampling of the source image by convolution with offset masks as shown in eqn 3. Subpixel interpolation can be performed to an accuracy of 1/8 of a pixel in both of the x,y dimensions using 64 8× 8 off-centre masks. The data bandwidth involved in this process implies that the coefficient data must all be stored on-chip. Edge enhancement, gaussian smoothing and rectification are all efficiently combined into this convolution/interpolation stage.

The stretch-correlation stage can be considered as correlating with a resampled template for each image block and for each stretch value. The resulting dot-product calculation is normalised with eqn 4 and can be expressed by eqn 5. By rearranging eqn 5 we can obtain an expression for the correlation measure which contains two reusable partial summation terms as in eqn 6, this reduces the

computation required for the correlation stage by a typical factor of 5 compared to a template based approach. This implies the need for 1D convolution support.

The edge detection stage of the algorithm extracts the nonhorizontal edge positions by the application of the simple heuristic operator expressed in eqn 7. This stage provides the data bandwidth reduction necessary for further efficient high-level processing.

$$P_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{i,j}^{a} I_{m+i,n+j} \qquad N = 8, a \in \{0..63\} \tag{3}$$

$$c^2 = \sum_{i=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{j=\frac{-N}{2}}^{\frac{N}{2}-1} L_{m+i,n+j}^2 \sum_{i=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{j=\frac{-N}{2}}^{\frac{N}{2}-1} R_{m+i,p+j}^2 \qquad N = 16 \tag{4}$$

$$x = \frac{1}{c} \sum_{i=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{j=\frac{-N}{2}}^{\frac{N}{2}-1} (\alpha_k L_{m+i,n+k} + \beta_k L_{m+i,n+k+1})\, R_{m+i,p+j} \tag{5}$$

$$x = \frac{1}{c} \sum_{j=\frac{-N}{2}}^{\frac{N}{2}-1} \alpha_k \sum_{i=\frac{-N}{2}}^{\frac{N}{2}-1} L_{m+i,n+k} R_{m+i,p+j} + \beta_k \sum_{i=\frac{-N}{2}}^{\frac{N}{2}-1} L_{m+i,n+k+1}\, R_{m+i,p+j} \tag{6}$$

$$P_{k,l} \textit{ is edge if} \{P_{k,l} < P_{i,j}\} \textit{has fewer than 3members} \tag{7}$$

$$i \in \{k-1, k, k+1\} \textit{ and } j \in \{l-1, l, l+1\},\ \ i,j \neq k,l$$

Equations 3 to 7 clearly demonstrate that our edge based stereo algorithm can be implemented as a set of 1D and 2D multiply accumulate (MAC) operations. Summarising, even for modest image sizes 256 × 256 and sparse edge data, the total computation is $> 10^7$ MAC operations per image pair.

4 Chip

4.1 Design Requirements

Image reprojection can be performed as a two-pass raster scan algorithm by generalising with $[x'\ y'] = [X(x,y)\ Y(x,y)]$, where (x,y) in the source image maps to (x',y') in the output image. By firstly keeping y constant, x' can be calculated from $x' = X(x,y)$, and secondly by calculating $y' = Y(x',y)$. For general reprojection (with many-to-one pixel mappings) finding the inverse of x' can become complex. Additionally, for a two-pass algorithm, the interpolation scheme must be devolvable in x and y. To maintain generality (with respect to other image processing functions) it was decided that image warping would be supported by employing a single-pass inverse mapping scheme. This implied that the processor must be able to efficiently access the source image in a nonraster based stream

and be capable of storing and applying a large number of arbitrary 2D filter kernels. This formed the basic requirement of the chip and dominated its design such that the list of requirements for a general purpose image processor had to be slightly compromised. However, the design of the chip has attempted to address issues which are specifically relevant to the demanding problem of non-linear image warping common in computer vision. The full list of requirements is given below:

- Minimum 8×8 pixel convolution kernel with minimum 8 bit coefficients and 16 bit image data.
- No intermediate truncation of results.
- Must support raster and nonraster based processing.
- Must deliver rectified 512×512 images at around 10Hz.
- Must support 1D and 2D accumulation.
- Coefficients must be local and changeable every multiplication cycle
- Must be easy to program and incorporate into systems design.

We feel that this chip covers a significantly large enough domain to be classified as a general purpose computer vision processor and can be regarded as a complementary device to that described in [1].

4.2 Architecture and Programming

Figure 5 shows the major functional components of the chips datapath which will be fabricated on a 1um process and will clock at 20 MHz. It consists of 8 multiply-accumulators (MACs) which produce 8 1D dot-products every 8 clock cycles, and a final accumulator which is only used in 2D mode. Two address generators produce addresses for both input and output data at up to 20MHz. Two onchip RAMs exist for mask coefficients and image data caching. The coefficient RAM can store 64 8x8 2s complement coefficient masks which, for the case of image rectification, represents the ability to interpolate using a gaussian mask at 64 subpixel locations on the 2D image plane.

Support for nonraster based processing is provided by the input image caching system which at any point in time holds a valid readable 8x8 pixel window of the source image. This 8x8 pixel window is fed onto the multipliers at a rate of one 8 pixel column (128 bits of image data) every clock cycle. Over 8 such clock cycles, a new 8 pixel row or column is being assembled at a rate of 1 pixel (16 bits of image data) every clock cycle from offchip RAM, into the currently writable area of the cache. To perform this task the image cache uses a novel dual memory ping/pong arrangement which operates in a row and column wise manner, such that at any given time certain areas of the cache are readable, whilst the remaining areas are writable. This caching scheme ensures that, as long as the window on the source image only ever moves by a maximum of one in any direction between points of application, the 8 multipliers will not stall.

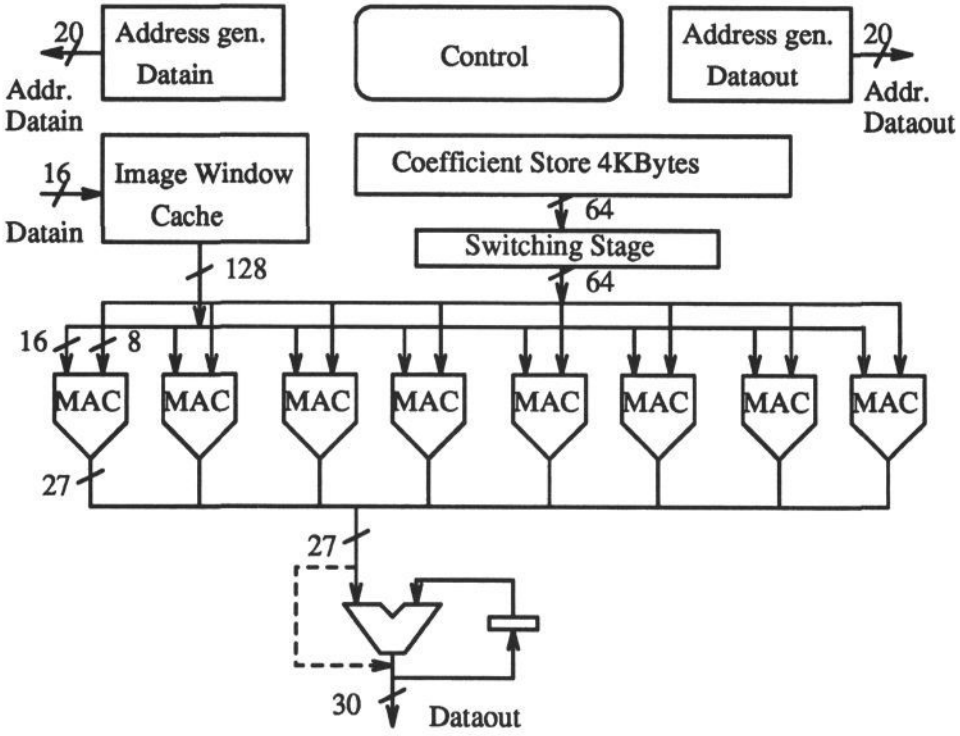

Figure 5: Chip Architecture

Figure 6 shows how the image cache (implemented as 8 physical onchip RAMS, but conceptualised here as two virtual RAMs A and B) is divided into dynamically changing readable and writable areas as the source image window moves first by 1 pixel south and then by 1 pixel east. As a consequence of caching the source image in this way, the origin (top left-hand corner) of the 8x8 pixel window becomes displaced across the cache in both x and y directions. Accounting for the x-offset is easily performed by addressing, however, the y-offset has to be taken out by barrel shifting in parallel the 8 coefficient values which form the second operand in the multiplication stage. This caching technique allows each 16 bit pixel read onto the chip to be reused indefinitely, thus reducing the image onchip read bandwidth by a factor of 8 typically.

The chip has been designed to be easy to use and requires a host controller for the simple tasks of resetting and loading coefficient masks. Coefficient and register loading is simplified by making the chip appear like a static RAM to its host controller. Many nonraster based image warps can be formulated as a set of XY-vector steps in the source image, and a set of mask identifiers to select the required mask. Our chip is programmed in this manner. Programs consist of a list of 16 bit instructions of the following form:

$$prog \;=\; mask\ id. \in \{0..63\}\ yvec \in \{0..15\}\ xvec \in \{0..15\} + \; HALT$$

The chip can move the applied location of the coefficient kernel by up to 16

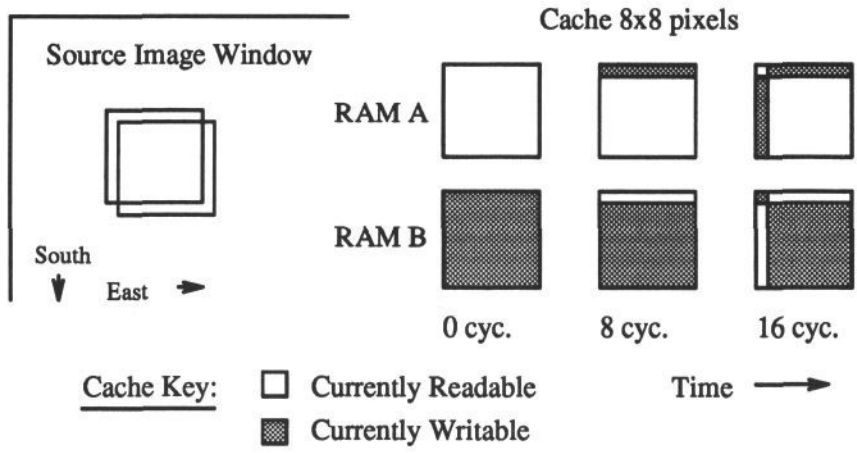

Figure 6: Image Cache Ping/Pong over time

pixels in x and y, but efficient reuse of data relies on small shifts. A shift vector greater than 1 pixel will cause the multiplication pipeline to stall while the input cache is loaded. Figure 7 shows typical scan paths through the source image which could be used to perform image rectification and a cartesian to polar transformation, with minimum stalling. In the case of image rectification the output image can be scaled such that 99% of all shift vectors are 1 or 0 in either x or y. Thus the processor will effectively run at the optimum rate of 20/8 MHz output pixels in 2D mode or 20 MHz output pixels in 1D mode.

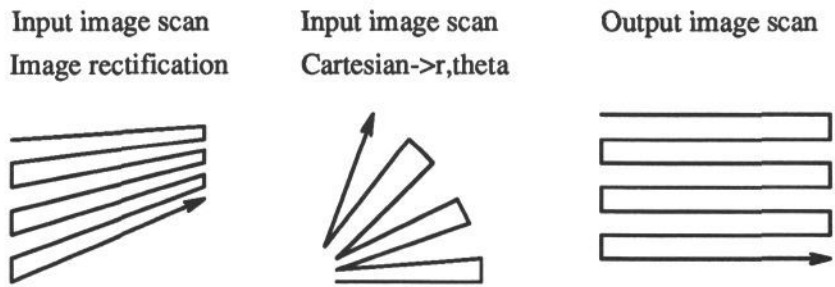

Figure 7: Typical Input and Output Image Scanpaths

4.3 Conclusions

We have presented a summary of a correlation based stereo vision algorithm designed to make use of the same constraints exploited in more robust edge-string algorithms. We have shown that an existing feature based computer vision algorithm can be reformulated for specialised hardware and thus enable efficient implementation of algorithms for real-time applications. With current academic support for VLSI design under such schemes as Eurochip these devices can be

developed with available design packages. Once developed, this hardware would bring real potential for commercial exploitation of machine vision research. Such hardware is, however, unlikely to emerge in the industrial sector for communication or entertainment, as these applications place emphasis on data throughput rather than computational accuracy. Hardware development must be done without compromising algorithmic performance and preferably in a way that has a wide range of possible applications. Typical applications include image warps, convolutions, neural network calculations and filter operations. The computationally intensive parts of such applications could be implemented on the chip described in this paper, and we believe this work demonstrates that powerful and efficient general purpose processors for computer vision are feasible.

References

[1] Evans S.J., Thacker N.A., Yates R., Ivey P.A., "A Massively Parallel Vector Processor for Image Communications". Proc IEEE Image'Com 93, 303-308, 1993.

[2] Inria. "A Parallel Algorithm that Produces Dense Depth Maps and Preserves Image Features". Research Report No. 1369 1191.

[3] Lane R.A., Thacker N.A. and Seed N.L. "Stretch-Correlation as a Real-Time Alternative to Feature Based Stereo Matching Algorithms". Image and Vision Computing Journal in print 1993.

[4] Masatoshi Okutomi and Takeo Kanade. "A Locally Adaptive Window for Signal Matching". Intl. Jour. of Computer Vision 7:2, 143-162, 1992.

[5] Mori K., Kidode M. and Asada H., "An Iterative Prediction and Correction Method for Automatic Stereocomparison". Computer Graphics and Image Processing 2, 393-401, 1973.

[6] Otto G.P., Chau T.K.W., "A "Region Growing" Algorithm for Matching of Terrain Images". Proc. 4th AVC 123-128, 1988.

[7] Pollard S.B., Mayhew J.E.W., Frisby J.P. "PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit". Perception 14, 449-470, 1985.

[8] Sailesh K Rao et al., AT&T Bell, Labs. "A Real- Time P*64/MPEG Video Encoder Chip". IEEE International Solid-State Circuits Conf. 32-33, 1993.

[9] Thacker N.A., Courtney P., "Online Stereo Camera Calibration", AI Vision Research Unit, University of Sheffield.

[10] Thacker N.A., Ivey P.A., "An Assessment of Image Processing and Computer Vision Algorithms Suitable for VLSI Implementations", ESG, Dept. EEE, University of Sheffield, Report 93/3, 1993.