Tina Memo No. 1994-002
Presented at: BMVC, York, Sept. 1994.

A Correlation Chip for Stereo Vision.

R.A.Lane and N.A.Thacker.

Last updated
1 / 12 / 1998

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

1 abstract

This paper shows 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.

2 Introduction

In many cases computer vision algorithms use feature extraction as a preprocessing stage for the higher levels of processing. The justification for the use of feature based represention comes from the flexibilty and the 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 where the grey-level data is not consistent between stereo viewpoints, and the amount of object deformation between views is large. [7],[edge-string]. However, if real-time computer vision is to be a viable proposition for applications such 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. Once general purpose processing has become a prerequisite the computational efficieny required for real-time vision starts to become more difficult to achieve. In general, efficient image manipulation can only be achieved by exploiting the regular ordering of data in images using vector based operations such as correlation.

In the case of stereo vision, before the development of edge-string algorithms, correlation based approaches were used extensively to establish correspondence. However, the lack of robustness of these techniques for difficult stereo problems was due to the unsuitability of absolute grey-level similarity measures for determining correspondence when illumination of the objects differs apprerciably between views. Edge string based algorithms removed this problem by manipulating quantities which were 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 taking the essence of edge-string matching and reformulating this into a convolution based implementation. This work has involved the development of an algorithm called "stretch-correlation" which is essentially a reformulation of edge-string based algorithms, and the development of a chip to perform all of the vector acceleratable aspects of the stretch-correlation algorithm. The specification of the chip has involved, as much as was possible, the inclusion of general purpose functionality making it capable of performing many other image processing and computer vision operations. The chip can, therefore, be classed as a computer vision processor. The justification for the chip and its architecture are discussed below.

The development of a variety of Video Signal Processing devices [8] has been prolific in recent years, and in the main has grown to meet the requirements of image processing functions such as motion compensation and image coding. Whilst in theory these devices offer great potential for implementing other algorithms, it as apparent that the requirements of this market only partially intersects the requirements of computer vision. For example, the basic requirement of the mass market is in general that images are processed at a rate of 25-30 Hz, whereas in contrast, our approach to computer vision is that the underlying algorithm is not compromised as a consequence of implementation. It is important that the numerical properties of the algorithm are preserved, at the expense of the image throughtput capability if necessary.

3 Algorithm Classifications

Rationalising the requirements of a subset of target algorithms forms one of the initial stages of general purpose hardware design. In addition to the 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. In addition we have also attempted to provide support for general purpose image processing functionality, based on the results of an algorithms survey [1]. A summary of the conclusions of this report, regarding general purpose algorithmic requirements, 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 [Ref] recently designed in our group. However, in addition to the core arithmetic operations, classifications based on data access requirements can be formed. The survey concluded that convolution can be subdivided into catergories based on the locality and uniformity of data access. The following classification contains three catergories in ascending order of data bandwidth:

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

While the first two categories may be supported with standard levels of communication bandwidth and data casheing stratergies the third category, which includes algorithms such as non-raster devolvable image warps, requires special consideration. In particular, for a VLSI design the only practical solution requires a large on-chip coefficient store. This is clearly at odds with the previous computational specification. 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.

4 Stretch-Correlation Algorithm

4.1 Description

As we have alreay said, 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 aleviated, but the dependance on grey- level consistency between stereo views causes a lack of robustness in nonideally lit environments. In contrast edges represent the underlying three dimensional structure of the scene, and are a more reliable match primitive.

Figure 16.1 shows the four basic stages of the stretch-correlation stereo algorithm. The first stage is epipolar realignment which is required to reduce the number search dimensions in the correlation stage, and requires precise camera calibration data optimsed to remove epipolar errors [9]. This stage (image rectification) presents a major computational load as will be discussed later. Our algorithm embodies edge matching in a correlation implementation by firstly only attempting to match blocks of the image which contain edges and secondly by using preprocessing stages to enhance non-horizontal edge imformation 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 trivial case of a single edge in the image block in figure 16.2. The purpose of the block stretching/shearing is to allow a linear disparity gradient to exist within each block. This provides subpixel location for edge based data which obeys our first order model of figural distortion between views. We have, therefore, addressed the two problems in correlation based stereo: accuracy and robustness.

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 based 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

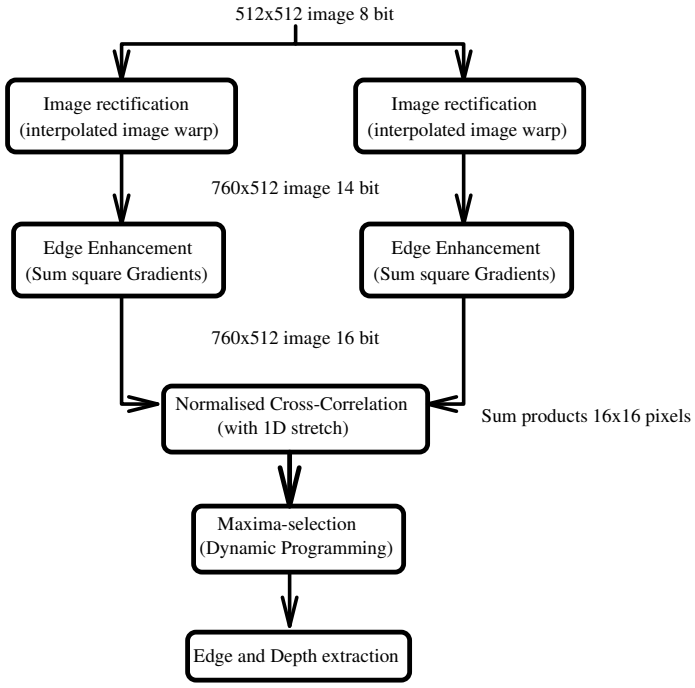

Figure 1: The Stretch Correlation Algorithm

by placing a cut on the characteristic signal distribution for the correlation score, thus allowing the selection of a specific signal-to-noise ratio. At this stage a "loose" global support constraint based on a disparity gradient is applied [7], where loose means that a block must at least recieve some support from neighbouring blocks. A disparity gradient limit of 2 is the mamimum required to enforce ordering. Unsupported hypotheses are thus rejected and other hypotheses are examined with a single pass philosophy. Whilst this stage does require high-level processing, the overall overhead is negligable in comparison to the hypothesis generation stage.

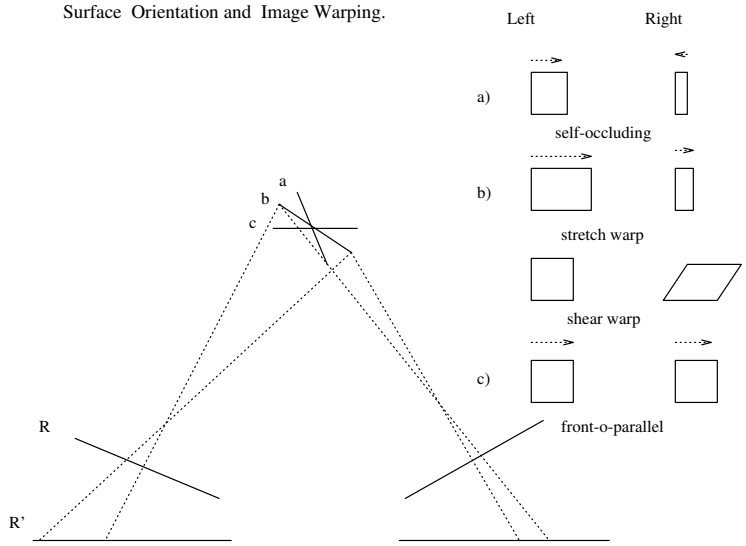

Figure 2: The Stretching and Shearing Process

Once a block match has been establised the depth at all edgels is calculated using the two parameters obtained from the matching stage: horizontal disparity at the block centre and a stretch/shear value from the window shaping.

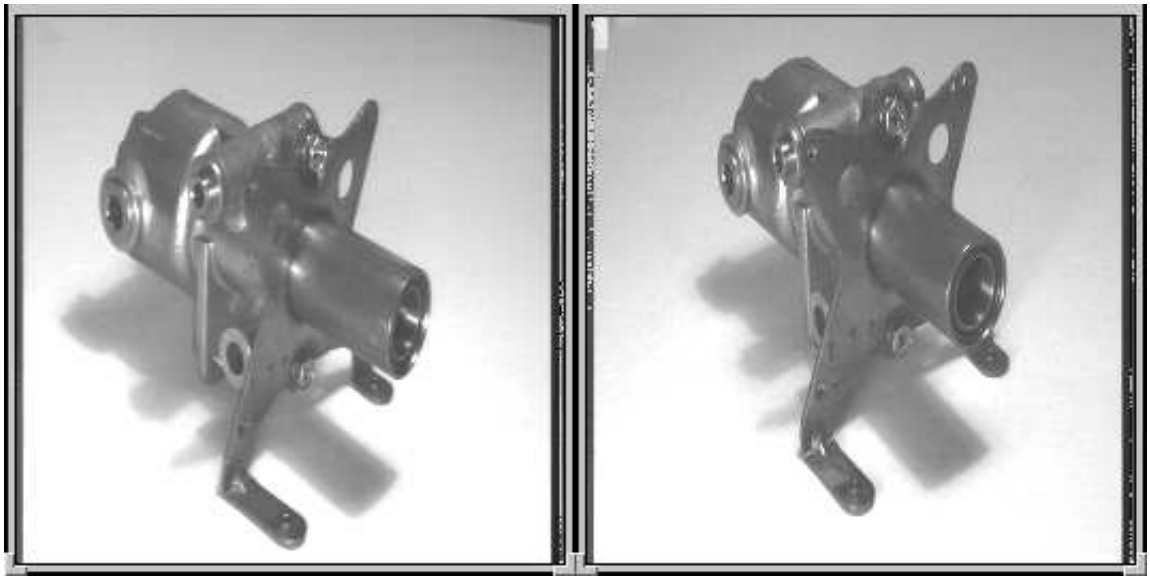

Figure 3: Shaft Assembly Image Pair

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

5 Algorithm Performance

Figure 16.3 shows a typical difficult stereo problem, for which the stretch-correlation algorithm is intended. Figure 16.4 shows a 3D reprojection of non-horizontal edge data obtained from the stretch correlation algorithm (left), compared to the results obtained from the PMF edge-string based algorithm (right). 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 a comprehensive manner in comparison to other correlation based techniques. The criterion used were edge location accuracy, quantity of returned edge data and disambiguation ability [3]. Summarising our experiments it was found that disambiguational ability was comparable to current Euclidean distance methods [2] with significant improvements with respect to location accuary. 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, but, significantly the stretch correlation algorithm returned a larger quantity of matched edge data.

6 Computational Requirements

The purpose of developing low-level algorithms is to enable the use of vector acceleration hardware to provide an efficient solution for real-time problems. This section examines the fundamental manipulations of our stereo algorithm, and shows where redundency has been exploited.

The image rectificaion stage of the algorithm requires a perspective reprojection of pixel coordinates with sub-pixel interpolation. The perspective reprojection takes the form

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

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

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. This equation represents a nonraster devolvable image warp due to the process of perspective forshortening imposed by the division with w. This presents a major problem in terms of high bandwidth nonuniform data access of the source image and leads to the requirement of a non-raster processor. Also, the image interpolation process is ideally performed by resampling of the source image by convolution with offset masks as shown in eqn 16.3. 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 this 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 value of stretch. The resulting dot-product calculation is normalised with eqn 16.4 and can be expressed by eqn 16.5. By rearranging eqn 16.5 we can obtain a new expression for the correlation measure which contains two reusable partial summation terms as in eqn 16.6, this reduces the computation required to compute the correlation stage by a factor of 5 typically compared to a template based approach. This suggests the need for 1D convolution capabilities if support hardware is going to exploit this method.

The edge detection stage of the algorithm extracts the nonhorizontal edge positions by the application of the simple heuristic operator expressed in eqn 16.7. This stage provides the data bandwidth reduction neccessary for subsequent (high level) processing stages and could easily be supported on a standard general purpose processor.

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

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

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

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

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

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

Equations 16.3 to 16.7 clearly demonstrate that our essentially edge based stereo algorithm can be implemented as a set of 1D and 2D multiply accumulate operations. To enable the amount of computation required at each stage to be put into perspective, the following list gives a breakdown in terms of the core operations: additions/subtractions and multiplies:

Image rectification, edge enhancement : WHn^2 *mults and adds*

Cross Correlation : $\frac{5}{2}\ WHR\gamma\ mults\ and\ adds$

Edge Detection : $2WH \times 8\ subtractions$

Where WH=Image width $\times$ height, n=gaussian kernel size=8, R=search range, γ=ratio of blocks containing edges to blocks which do not.

For images containing only sparse edges $n^2 \approx R\gamma$, and for only modest image sizes $\approx 256 \times 256$, the total computation is $> 10^7$ multiplies and $> 10^7$ additions per image frame pair.

7 Chip

7.1 Design Requirements

The image interpolation scheme outlined in section 16.6) and non-raster source data access inherent in image rectification lead to the requirement for the chip which we are now developing. These factors dominate the design of the chip such that the list of requirements for a general purpose image processor had to be compromised. However, the design of the chip has attempted to address issues which are specifically relevant to the demanding problem of non-linear image warp 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 changable every multiplication cycle
- Must be easy to program and incorporate into systems design.

Given these requirements we feel that this chip covers a significantly large enough domain to be classified as a general purpose computer vision processor and should be regarded as a complementary device to that described in [Ref].

7.2 Architecture and Programming

Figure 16.5 shows the major functional components of the chips datapath. The architecture of the chip, which will be fabricated on a 1um process and will clock at 20 MHz. It consists of 8 multiply-accumulators which produce 8 1D dot-products every 8 clock cycles, and a final accumulator which is used only in 2D mode. Two address generators produce addresses for both input and output data at upto 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.

The support for nonraster based processing is provided by the input image caching system which at any point in time holds a valid 8x8 pixel patch on the input image and the next new 8 pixel row or column of image data. The image cache uses a novel dual memory ping/pong arrangement which operates in a row and column wise manner, this allows a new 128 bits (8 x 16 bit image data values) of image data to be read every 8 clock cycles. The resulting stored image represents a barrel shifted version of the required 8x8 data window in the input image (Figure 6). This offset has to be taken out during convolution by a combination of addressing and by barrel shifting the 8 sets of 8 bit coefficient data onto the appropriate multiplier. This configuration provides an effective factor of 7/8 reuse of data (best case) reducing the required input image data bandwidth by factor a factor of 8. The net efficiency of this casheing stratergy depends upon the details of the scan path in the input image which will be discussed further below.

One of the requirements for this chip was that it should be easy to use. The chip requires a host controller for the simple tasks of resetting and loading coefficient masks. This processor would be ideally placed for finishing up final stages of non vector acceleratable processing such as the feature extraction in our application. Coefficient and

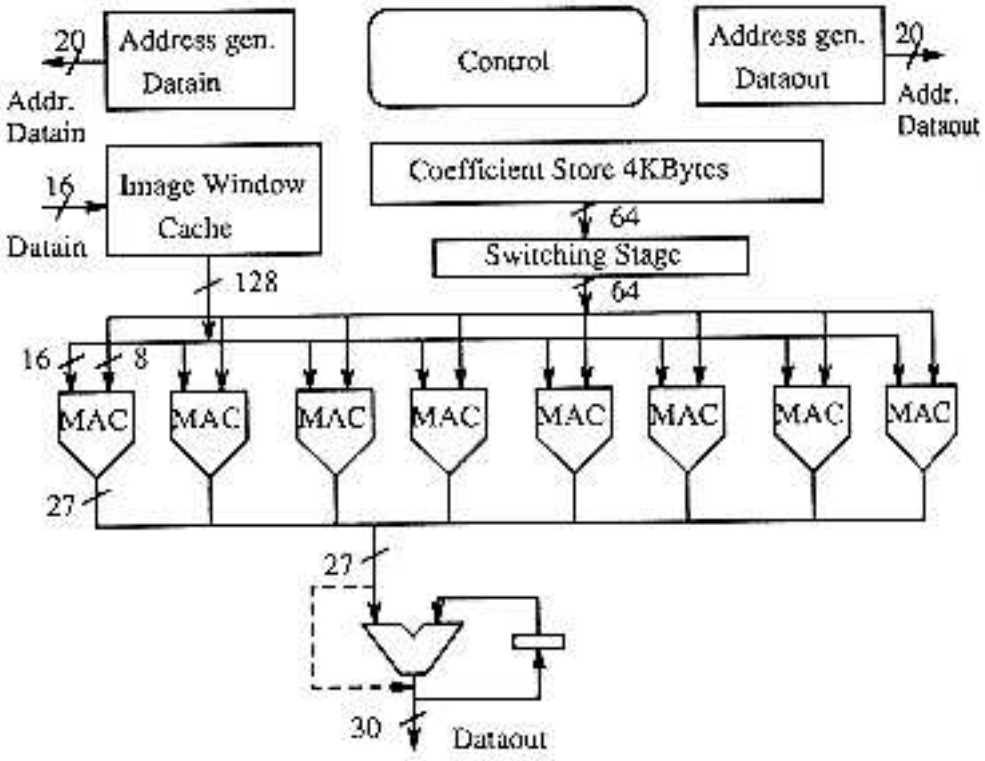

Figure 5: Chip Architecture

register loading is simplified by making the chip appear like a static RAM to its host controller when in a reset mode. Many nonraster based algorithms, including image rectification, can be formulated as a set of XY-vectors defining the kernel movement around the source image (Figure 7), and a set of mask identifiers to select the required mask. Our chip is programmed in this manner. The chip can move the applied location of the coefficient kernal by up to 16 pixels in x and y, but efficient reuse of data relies on small shifts. Any shift vector greater than 1 pixel will cause the multiplication pipeline to stall while the input cache is loaded. 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 run at effectively the optimum rate (20/8 MHz output pixels).

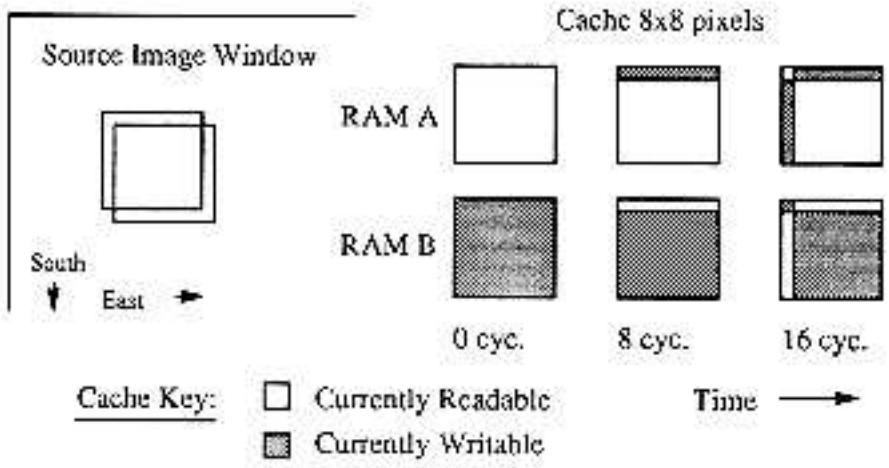

Figure 6: Image Cache

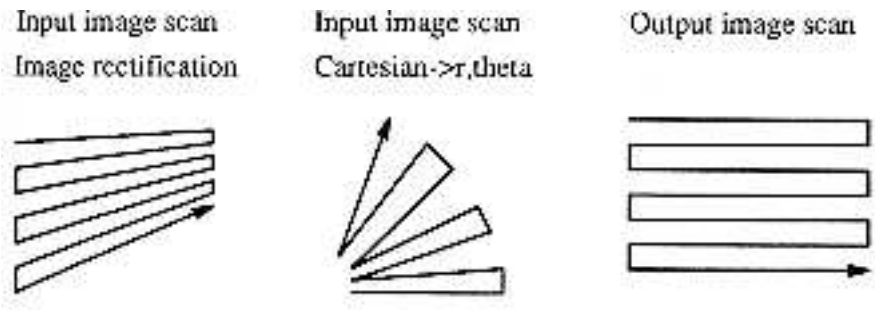

Figure 7: Typical Input/Output Scanpaths

8 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. In doing so we have shown that, with thought, 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 this hardware can be developed by making use of 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 applications such as communication or entertainment as these applications put the 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. The computationaly intensive parts of many computer vision algorithms could be implemented on the chip described in this paper and we believe that powerful and efficient general purpose processors for computer vision are feasible.

References

[1] Evans S.J., Thacker N.A., Yates R.B., 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.

[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 Stereo-comparison". 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.