A PIPELINED ARCHITECTURE FOR THE CANNY EDGE DETECTOR

B.P.D.Ruff

GEC Research Ltd.
Long Range Research Laboratory
Hirst Research Centre
East Lane
Wembley
Middlesex, HA9 7PP, UK

ABSTRACT

Presented here is an architecture for implenting a sub-pixel resolution edge detector based upon the second difference of a Gaussian filtering. A pipelining approach allows many simple operations to be performed in parallel across the video data stream so allowing a throughput at video rate, 10MHz. The output is intended describe edges in terms of an 8-bit strength and orientation as well as 8-bit Cartesian sub-pixel offsets giving a possible resolution of 1/50th of a pixel edge position.

I. INTRODUCTION

The edge data produced is intended as input to a stereo image analysis system[1] in real time, that is at video rate. The stereo algorithm operates through matching edge elements,'edgels', from the edgemaps of two binocular stereo grey level images assumed for the purpose of this discussion to have a parallel camera geometry, i.e. the ratster lines in the video signal are parallel to the epipolar lines, implying that corresponding edgels lie on the same raster in the two images. The edge matching algorithm, *PMF*[2] , requires an edge location accuracy greater than the course 0.5 pixels produced by naive edge detectors. This accuracy is provided through the implementation of the *Canny* edge on a discrete digital system with arithmetic precision tailored to the tolerance demanded by the edge matcher. Other factors influencing the accuracy of the operator are the speed restrictions for a 10MHz operating frequency for arithmetic operations and the limited range of existing fast digital signal processing VLSI circuits[4].

As compared to simpler algorithms such as the Sobel edge detector, the Canny operator is much more sophisticated and correspondingly more complex and computationally intensive. Performing the whole algorithm in one or a few clock cycles is not possible. A general outline of the Canny is given in Fig.1. Detailed in Fig.2 is an overview of a pipelined implementation of the algorithm showing the various computations required sub-divided into the major operations. The edge operator consists of an initial convolution of the image with a two dimensional Gaussian mask with some standard deviation sigma. After passing a gradient operator over the image, points of inflection are located in a process called non-maximal suppression. The edge map produced is then sieved to remove spurious edges in a process known as hysteresis. The data output by the edge operator is the cleaned edge map consisting of six pieces of information:

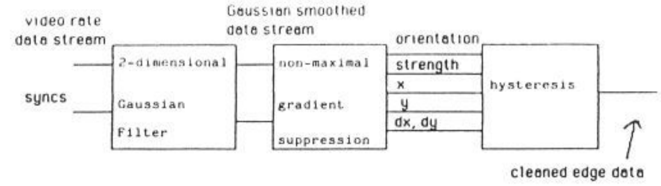

Figure 1 Digital video data is first conovolved with a two dimensional Gaussian, then non-maximal gradient suppression removes all but inflections on the smoothed intensity surface. Direction and sub-pixel offsets are calculated and then Hysteresis grows back noisy breaks in edges.

1. the edge strength
2. edge orientation
3. x coordinate
4. y coordinate
5. x sub-pixel offset
6. y sub-pixel offset

The x-y position may be assumed to be implicit in the timing of the video-bus connecting the stages of the pipeline.

II. CANNY OPERATOR

Refering to Fig.1 the Canny operator may be thought to contain three stages. These are; two dimensional

† The design of special purpose hardware was undertaken as part of the Alvey programme IKBS 025, "3D Surface Representations and 3D Model Matching from Stereo".

AVC 1987 doi:10.5244/C.1.19

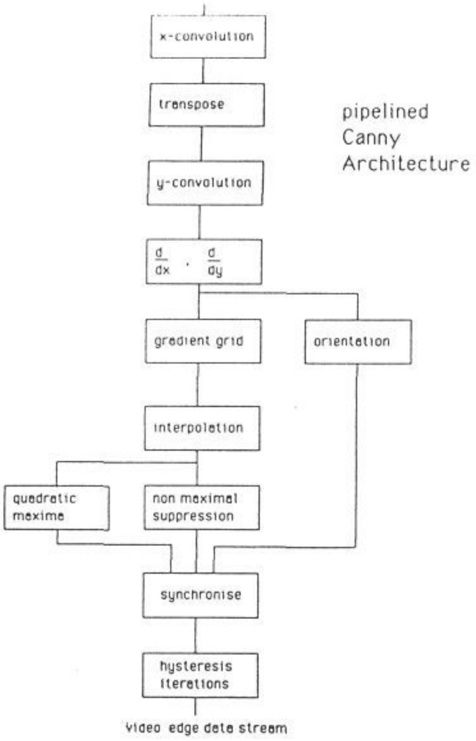

Figure 2 Data flow is shown from top to bottom. X-convolution followed by a transposition of the image and a subsequent identical convolution forms the two dimiensional Gaussian filtering. After a gradient operator, orientation is calculated in parallel with interpolation and quadratic solution. Non-maximal suppression decides upon the existance of an inflection.

Gaussian Smoothing , *Non-maximal* Gradient , and *Hysteresis.* As an insight into the working of the operator, if a one dimensional step edge is considered, then convolution with a one dimensional mask smoothes the step into a round ramp. Passing a differencing operator over the ramp the gradient may be determined at the pixel centres and an inflection located corresponding to the edge. The position of the edge is located to high resolution through the anti-aliasing effect of the Gaussian smoothing. Differentiation is achieved by convolution with the row matrix (-1 0 1). The sub-pixel offsets to the edge are found firstly by fitting a quadratic curve to the edge gradient and its two nearest neighbours and then solving geometrically for the maximum. The strength of the edge is not interpolated but is the nearest pixel gradient to the edge.

It then remains to remove the spurious edges. The hysteresis algorithm creates two masks initially, one containing eges above some lower threshold and the other above some upper threshold. To the 'strong' edges above the upper threshold after one iteration, adjacent edgels in the lower threshold mask are appended. In this way the effects of noise in the reduction of edge strength are removed as broken edge segments are 'grown' back together, while spurious edgels between the thresholds and not connected even indirectly to strong edges are discarded.

III. GAUSSIAN SMOOTHING

Conceptually, the simplest approach to two dimensional Gaussian convolution is to convolve the image with a large two dimensional mask. For an N by N image this requires N^2 multiplications and N^2 -1 additions. Implementaion of this using discrete multipliers is unacceptably large, though such multiplication may be possible in future with the advent of systolic array multipliers. Fortunately the Gaussian convolution allows a special decomposition into a cascade of two one dimensional convolutions, one a column matrix and the other a row matrix. Simply G2d = Gx * Gy. For a 40x40 convolution then 80 instead of 1600 multiplications need be performed. With the use of VLSI FIR filters, with say 8 multipliers per chip, i.e. 8-tap, then the use of five filters to produce a forty tap filter allows the convolution to be implemented within an acceptable size.

An overhead incurred with this method lies in the need to change the order of the image scan from raster scan to a column first scan. This is achieved by first writing the raster scan image into a frame buffer and then reading it out column first while the next image is written into a separate frame buffer. The accuracy of the arithmetic used in the FIR filters reflect the accuracy required in the edge position and to some extent edge orientation. In practice 14 bit filter coefficients and 10-bit intermediate data stored in a frame buffer between convolutions for 8-bit image data provides an accuracy of 0.018 pixels for an ideal simulated test image. This is indistinguishable from a floating point implementation of the edge operator applied to the same image. For practical reasons, 9-bit intermediate data and 9-bit filter coefficients are to be used providing a marginally degraded performance which in real images is indistinguishable from the above implementation, giving an error of 0.019 pixels. The error of ~0.02 pixels is within the tolerance of the edge matcher. The angular error in orientation is about 1.5° ·

IV. NON MAXIMAL SUPPRESSION

With reference to Fig.2, the video stream data is firstly buffered to produce a 3x3 grid of pixels so that a simple differentiation convolution may be performed to produce gradient components in x and y directions. The gradient convolutions are simply:

$$\frac{\partial I_{jk}}{\partial y} = I_{jk} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{\partial I_{j_k}}{\partial x} = I_{jk} * (-1\ 0\ 1)$$

Where I is a 3x3 matrix centred on the image at x= j,y= k.

This is the first computational unit in the non-maximal suppression pipeline. The gradient components are used to calculate the edge orientation and strength, the latter to be buffered in a 3x3 array of edge strengths to be used in the next unit. The orientation of the edge is used to select pairs of gradients within the array that lie to either side of an imagined extension of the edge normal through the central element in the array.The two pairs of gradients selected are used to interpolate the supposed edge strength along the edge normal between the strengths in each pair. The interpolation is a linear weighting of the two strengths taken as a moments calculation.

$$g_{interpolated} = g_1 . u + g_2 . (v - u)$$

Where g1 and g2 are the two gradients to either side of the normal, and u and v are the components of the gradient at the centre of the array. The pipelined architecture for this is shown in Fig.3. This shows the close mapping of the algorithm onto the architecture in a data flow manner, though synchronous clocking replaces token exchange. If ga and gb are the two interpolated gradients about the putative edge, then for an inflection to occur the conditions gcentral > ga , and gcentral > gb must be satisfied. In

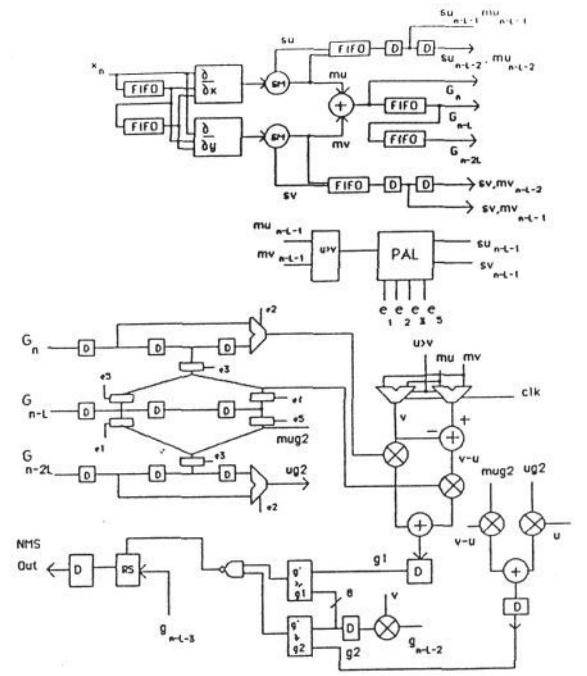

Figure 3 Top: Data is differentiated in X and Y directions and buffered to present three rows simultaneously to the 3x3 pixel selector. The output of the selector is firstly two pairs of pixels to be interpolated between and secondly the centre pixel. Non-maximal suppression is then a comparison between the magnitude of the central and two interpolated gradients masking non-maxima with zeroes.

order then to determine the sub-pixel offsets to the edge, a quadratic is fitted to the three gradients above and the displacement of the maximum from the central gradient calculated from the formula:

$$\delta x = \frac{g_a - g_b}{2(2g_{central} - g_a - g_b)}$$

$$\delta y = \delta x * \tan\theta$$

where theta is the orientation of the edge.

The orientation, strength, sub-pixel offsets are then synchronised in their pipeline position and passed to the hysteresis unit.

V. HYSTERESIS

In accordance with the pipelined architecture of the edge detector, the hysteresis algorithm is implemented as a pipeline. Each iteration operates in parallel with its predecessor to produce the very high number of logical operations per second required, around 1440 million operations. For each iteration a 3x3 grid is considered containing three bit planes (fig.4). One contains the lower thresholded mask, another the upper thresholded mask, and a third mask generated from the two above masks and whose output is the following iterations upper thresholded mask, or 'strong edge mask'. If NM is the generated mask, then the operation performed each clock cycle from an initially clear mask on a 3x3 grid is:

$$NM_{jk}=NM_{jk}+(UT_{11}.LT_{jk});\ 0 \le j \le 2,\ 0 \le k \le 2$$

where '+ ' denotes logical OR'ing, and UT and LT are upper and lower threshold masks. The next mask output bit is then fed into the input of the proceeding iterations upper thresholded mask. The next iteration's lower thresholded mask receives its input from the output of the lower threshold mask and thus is unchanged. The generated mask is fed with zeroes. Note that no edge segments can be invented, only unconnected weak edges are removed. For each iteration two lines of the bit masks must be buffered before passing to the next iteration. In this way 3 bits of two line buffers, or 24 bits for 8 iterations are needed. For standard 8-bit buffers, 6 8-bit line buffers as well as several PALs will provide all the logic and buffering for the algorithm.

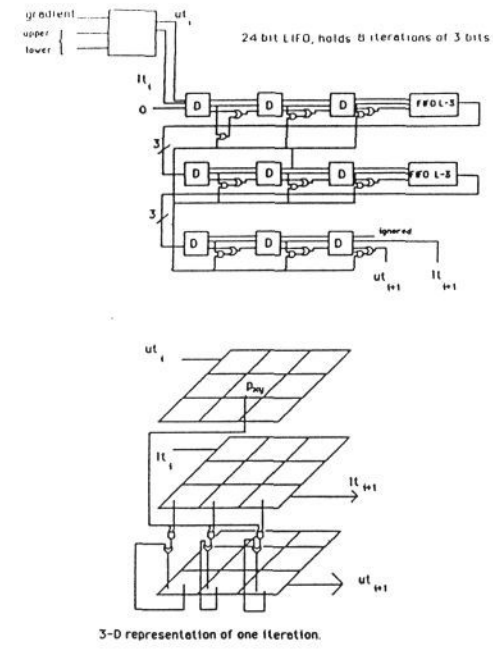
gradient
upper
lower
ut
24 bit LIFO, holds 8 iterations of 3 bits
lt
D
FIFO L-3
ignored
ut
lt
p_{xy}
3-D representation of one iteration.

Figure 4 Three bit D-type latches store the three bit-planes representing strong edges, weak edges, and next stong edge bit map. Shown in 3-D are these planes with the logic shown for the operation only for the front pixels. The operation is next mask OR'edwith itself and the product of the central strong edge and the weak edge bit.

VI. SOME RESULTS

An edge map of a simple scene containing several geometrical objects is shown in fig.5. For this scene a standard deviation of 1.0 for the Gaussian was chosen, with upper and lower threshols of 9 and 5 for the hysteresis. An expanded view of the ringed corner is given in fig.6 with a superimposed grid of 1 pixel on a side to give an indication of the reliiability of sub-pixel resolving power using a finite 9-bit implementation of the Canny operator. Note that the locational performance is degraded at the corner, a common feature of Gaussian or Laplacian of Gaussian based operators.

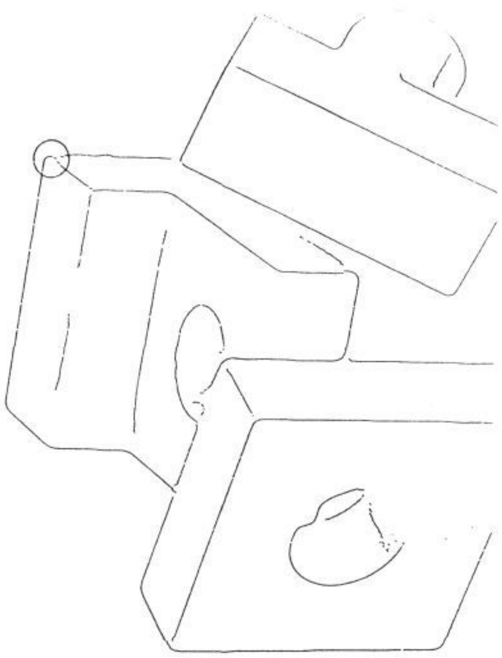

Figure 5 Edge map of widget scene with sigma= 1.0. The small circle shows the position of the junction in figure 6.

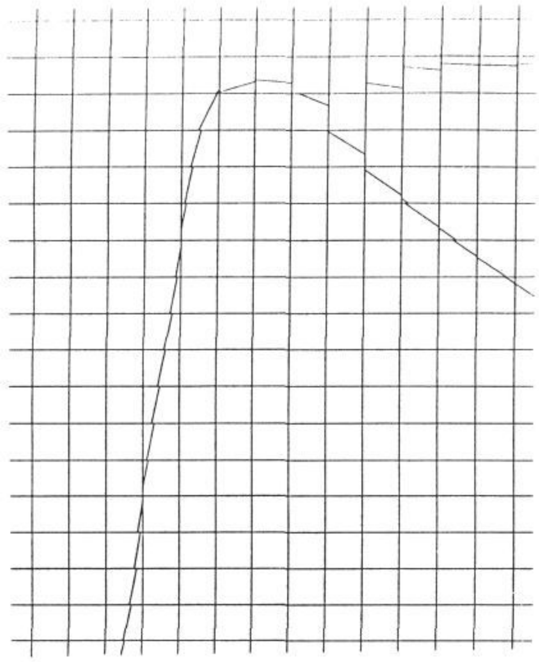

Figure 6 This edge map shows squares demarking individua pixels. The edgels are represented in position and direction only by the lines within the boxes.

VII. DISCUSSION

The use of a pipelined architecture for edge detection is particularly appropriate since information flow in these algorithms is unidirectional, allowing data further into the algorithm to be processed independantly of previous and future outputs.

In the Gaussian convolution, current DSP chips such as Zoran's 9x9 bit 8-tap FIR filter[5, 6]
both operating at 20MHz allow a degree of ease in building large digital filters impossible or very cumbersome using individual multiplier ICs, or if multiplier accumulators are used, too slow. The speed performance of video rate algorithms precludes ths use of digital signal processors such as the TMS32010 or parallel computers such as the DAP or transputer unless many processing elements are used. A solution then is to develop an architecture that is mapped directly onto the architecture of the algorithm so allowing many low level logical and arithmetic operations to be performed in parallel, though such an architecture is clearly application specific and thus inflexible unless many ALU's are liberaly connected to allow a programmable architecture, a possibility in the future of opto-electronics.

References

1. B F Buxton, Murray D W, Mayhew, J E W, Pollard, S B, and Pridmore, T, "Computer vision algorithms for early visual processing - I: initial considerations, image convolution, smoothing, restoration and early stereo visual processing," HRC Report 16,976A, GEC Research Ltd, Hirst Research Centre, Wembley, March 1986.
2. Stephen B Pollard, John E W Mayhew, and John P Frisby, "PMF - A Stereo Correspondence Algorithm Using a Disparity Gradient Limit," *Perception*, vol. 14, no. 4, pp. 449-470, 1985.
3. J F Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI– 8, no. 6, pp. 679– 698, November 1986.
4. "DSP : A Technology in Search of Applications.," *Computer Design*, November 1986.
5. "ZORAN ZR33891 Data Sheet," *ZORAN Corp, Peabody, MA. USA..*
6. "MA7180 Data Sheet," *Marconi Electronic Devices Ltd., Lincoln.*