3rd International Conference on Sensing Technology, Nov. 30 – Dec. 3, 2008, Tainan, Taiwan

Reconfigurable Foveated Active Vision System

Donald G Bailey
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
D.G.Bailey@massey.ac.nz

Christos-Savvas Bouganis
Department of Electrical and Electronic Engineering
Imperial College London
London, United Kingdom
ccb98@imperial.ac.uk.

Abstract—**Foveal images have variable spatial resolution, enabling a significant reduction in image size and volume. When combined with a high resolution input image, it enables almost instantaneous electronic panning and tilting. This paper proposes an FPGA based foveated imaging system where the spatial resolution may be adjusted from frame to frame. Several Cartesian mapping functions are considered in terms of pixel shape and efficient implementation on an FPGA. It is shown that a 2-pass mapping based on the L_∞ distance metric provides the best tradeoffs.**

Keywords- active vision, foveal vision, FPGA, real time image processing, embedded vision

I. INTRODUCTION (HEADING 1)

In recent years image sensors have reached very high capacity making 5M pixels in one sensor a commodity. This allows the acquisition of high resolution images, which usually has a positive impact on the overall performance of many computer vision algorithms. However, it also creates a computational burden in processing systems when real-time constraints have to be met and the whole information from the sensor has to be processed. Moreover, in certain applications, such as tracking and pattern recognition, it is not so important to maintain the same resolution across the image sensor as to have a wide field of view and have high resolution only on specific regions of the sensor. Thus, a trade-off between high resolution and processing power has been created.

Multi-resolution techniques are usually employed to address the above problem, where recently space-variant or foveating image sensors have been introduced that address the problem in its origin. These sensor architectures have variable spatial resolution across the surface of the sensor targeting data reduction without a severe impact to the final performance of the application. In [1] Martinez and Altamirano have demonstrated that a data reduction by a factor of 22 can be achieved without significant degradation in the performance of their tracking algorithm. Active vision techniques are then used to ensure that the high resolution part of the sensor corresponds to the region of the scene where it can be most effective. This is usually in the centre of the region of interest.

Pattern recognition and tracking are two of the domains where a foveating image sensor has been successfully employed. Wilson and Hodgson [2] addressed the problem of pattern recognition using a space-variant sensor, where a space-variant sensor system has employed in [3,4] for tracking.

Current approaches are divided into systems where the fovea can be electronically configured, and those that feature a fixed topology. In the latter approach, the image sensor can not be reconfigured to place the high resolution region to a different region of the sensor and if this is necessary it should be performed by mechanically steering the camera.

This paper presents an approach that is based on the first category. It focuses on the implementation of such a sensor using field programmable gate arrays (FPGAs) and off-the-self CMOS imaging devices. A reconfigurable system based on an FPGA is proposed and the resource utilization of the FPGA for mapping a set of possible topologies for a space-variant sensor is investigated.

II. RELATED WORK

Many configuration topologies have been introduced in the literature for spatially variant imaging. Most are inspired by the human vision, having a resolution that decreases with the distance from the centre of the sensor. In [5], Bandera and Scott propose the use of rectangular and hexagonal lattices with varying resolutions, while in [2] Wilson and Hodgson use a log-polar mapping inspired by the sampling structure of the human retina. In [6], Traver and Pla consider the trade-offs in the sensor topology when a log-polar mapping is targeted.

Although the log-polar mapping is the most widely used mapping, one of its drawbacks is the non-linear effects that the polar coordinate system introduces to the regions of the sensor that do not align with the optical axis. Another problem is the singularity and consequent blind-spot in the centre of the fovea.

In [7], a foveate wavelet transform is proposed as a variable resolution technique based on masking within the wavelet transform, for data compression. The main benefit of the transform is that it preserves the linearity, by using only low and high pass filters to perform the mapping.

Recently, research has focused on sensors that can provide many fovea regions rather than having a fixed topology. Camacho et. al. [8] proposed a multi-resolution topology where the fovea region can be shifted across the sensor emulating the saccadic eye movements. In [9] they generalised this idea to multiple foveal regions and moving object detection.

There are four main approaches for acquiring an image of variable spatial resolution.

978-1-4244-2177-0/08/$25.00 © 2008 IEEE

Optics approaches: In [10], Huniyoshi et al. achieved a variable spatial resolution by using a regular CCD sensor combined with a specific lens that mimics acuity of the human visual system. More recently, Hua and Liu [11] proposed an approach that uses two off-the-shelf image sensors combined with a beam splitter to create a foveating imaging system.

VLSI approaches: Targeting a small form factor and reduced power, researchers have focused on manufacturing image sensors that have variable spatial resolutions [12,13]. The main drawback of these approaches is the fixed topology. Those that are reconfigurable have limited degrees of configuration.

Software emulation: Emulating a variable resolution sensor using software has been adopted by many researchers due to the low cost and high flexibility that it offers. However, for real-time applications, the required processing time limits its applicability.

Hardware emulation: A widely used approach is the use of a standard CCD sensor combined with a VLSI or FPGA that maps the image to a topology that emulates a variable resolution sensor. Camacho et. al. [9] proposed a system that employs an FPGA to perform the mapping of an adaptive fovea sensor. Arribas and Macia [14] proposed an implementation of log-polar mapping using an FPGA that was able to achieve real-time performance. In [15], the VASI (variable acuity super-pixel imager) system is described. It uses a CCD sensor and an FPGA for real-time image processing. The system supports multiple fovea, but only two levels of resolution are possible. More recently, Martinez and Altamirano [1] proposed an FPGA pipelined architecture that transforms Cartesian images to a foveated image.

This paper proposes an approach that belongs to the hardware emulation category. It proposes a reconfigurable system using an off-the-shelf CMOS sensor combined with an FPGA that performs the mapping to a foveated image. The original contributions of this paper are:

- several possible mappings for a spatial variant image sensor are investigated;
- the FPGA resource requirements of those mappings are outlined for real-time operation;
- the design enables almost instantaneous pan and tilt under electronic control within a wide field of view;
- a small, light weight system enables embedded applications, for example mobile robotics.

III. BASIC PRINCIPLE

To reduce the need for a pan-tilt head, we propose using a high resolution (3M or 5M pixel) CMOS sensor with a wide angle of view. However, within this frame, we only read out data from within a smaller window. The window may be repositioned within the frame under programme control. This provides the equivalent of a very fast pan and tilt because the window may be repositioned from one frame to the next. As the camera does not move, both the latency and motion blur associated with physically panning or tilting the camera is avoided.

While it is possible to process such data in real time on an FPGA, any algorithm that requires multiple frames will require significant off-chip memory, with its consequent bandwidth bottleneck. To enable on-chip storage, the volume of data, hence image size, must be reduced considerably.

The simplest way to reduce the data volume is to reduce the image size, and hence resolution. The consequence of this is a loss of information that may be critical in many applications. Often, high resolution is only critical in small regions of the image. A foveated window, inspired by the human visual system, provides a balance between high resolution, and large data volume. Within the window, the high resolution is maintained in the centre, with the resolution decreasing towards the periphery. Compared to a uniform resolution image, the increase in resolution in the centre comes at the expense of a decrease in resolution at the periphery.

The architecture for accomplishing this is shown in Fig. 1. The FPGA is used to set the window position, apply the foveal mapping of the input data, and process the resulting image to extract key features for window control.

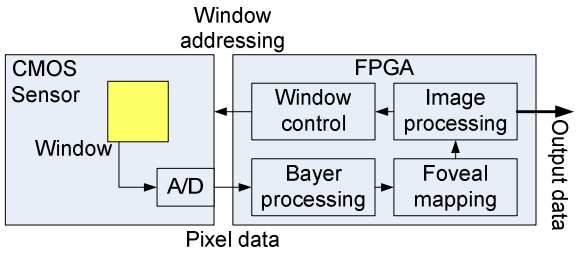

Figure 1. Proposed system architecture.

A. Foveal Mappings

Rather than use a log-polar mapping, which is based on a polar coordinate system, we propose to maintain Cartesian coordinates in the warped image. Let u be the distance from the centre of the input window and f be the distance from the centre of the foveated image. The foveal mapping can then be defined either in terms of the forward mapping

$$f = \mathrm{map}_f(u) \tag{1}$$

or the reverse mapping

$$u = \mathrm{map}_r(f)\,. \tag{2}$$

The magnification, M, at any point is the ratio between the output and input pixel distances:

$$M_f(u) = M_r(f) = u/f\,. \tag{3}$$

Related to the magnification is the acuity, A, which is the effective resolution of a pixel. The acuity is given from the slope of the mapping,

$$A(u) = \frac{du}{df} = \frac{d\,\mathrm{map}_r(f)}{df} = 1 \Big/ \frac{d\,\mathrm{map}_f(u)}{du}. \quad (4)$$

There are relatively few constraints on the mapping function The mapping should be monotonic, and there is little point in having an output pixel larger than an input pixel. This implies

$$0 < A(u) \le 1. \quad (5)$$

To map an *N*x*N* input window to a *w*x*w* output image also requires

$$\begin{aligned} \mathrm{map}_f(0) &= \mathrm{map}_r(0) = 0 \\ M_f(N/2) &= M_r(w/2) = w/N \end{aligned} \quad (6)$$

If the fovea is in the centre of the image, then

$$M(0) = A(0) = 1.0 \quad (7)$$

although this is not actually a constraint; the resolution in the fovea does not have to correspond to the input resolution, and the best resolution does not have to be in the centre of the output image.

Another consideration is how to define the distances *u* and *f*. Different definitions will affect the nature of the distortion introduced. Let the coordinates relative to the centre of the input window or output image respectively be defined as (x_u, y_u) and (x_f, y_f). A radial Euclidean mapping, using the L_2 distance metric would then be

$$f = \sqrt{x_f^2 + y_f^2} = \mathrm{map}_f\left(\sqrt{x_u^2 + y_u^2}\right). \quad (8)$$

With a radial mapping, the angle of the point relative to the centre of the image is unchanged. This means that the *x* and *y* coordinates are both scaled by the magnification for the given radius:

$$\begin{aligned} x_f &= x_u M_f(u) \\ y_f &= y_u M_f(u) \end{aligned} \quad (9)$$

A computationally simpler transform may be obtained by using the L_∞ or chessboard distance metric, again with radial scaling using (9):

$$u = \max\left(|x_u|, |y_u|\right). \quad (10)$$

An even simpler transform may be obtained by considering the mapping separable (rather than radial) and independently mapping *x* and *y*:

$$\begin{aligned} x_f &= \mathrm{map}_f(x_u) \\ y_f &= \mathrm{map}_f(y_u) \end{aligned} \quad (11)$$

With the separable mapping there are different magnifications in each of the coordinate directions.

Fig. 3 compares the effects of these three mappings for *N*=512 and *w*=64 with a uniform reduction in resolution (*M*=1/8). Panels (c), (d) and (e) all have the same magnification function:

$$\mathrm{map}_r(f) = f + \tfrac{7}{32} f^2 \quad (12)$$

but with the different distance measures. Panel (f) has the same distance measure as (d) but with a larger foveal region:

$$\mathrm{map}_r(f) = f + \tfrac{7}{32768} f^4. \quad (13)$$

The acuity as a function of distance for these maps is shown in Fig. 2. The larger the size of the fovea, the smaller the acuity at the periphery.

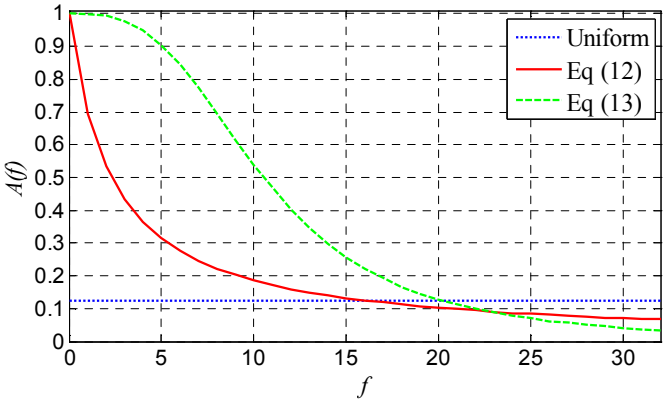

Figure 2. Acuity for the different fovea mappings.

The size and shape of the pixels depends on both the mapping, and their position within the output image. The pixels are only square in the centre of the fovea, and along the diagonals of the separable mapping. The longest dimension of the pixel is given by the acuity, however for the radial mappings, the shortest dimension is given by the magnification so the pixels in the foveal image generally have better resolution tangentially than radially. The change of aspect ratio with positions for the radial mappings is more uniform, so that standard image processing techniques are more likely to work correctly on the smaller image. This is less so for the separable mapping.

In the foveal images, objects appear least distorted with the L_2 mapping although horizontal and vertical lines become curved. However, the overall resolution with this mapping is lower because of the regions in the corners without data.

Figure 3. Example foveal mappings: Central column: low resolution mapped images, enlarged here to show details; side columns, the mappings shown in the input space to show the effect of resolution reduction. (a) original Lena (512x512); (b) uniform reduction to 64x64; (c) L_∞ mapping; (d) separable mapping; (e) L_2 mapping; (f) separable mapping, with a larger fovea than (d).

IV. FPGA IMPLEMENTATION

To gain the maximum benefit from the reduction in data volume, it is necessary to warp the image as the pixel data is streamed from the sensor. Most commonly, a reverse mapping is used to perform image warping; reverse mapping determines, for each output pixel, the corresponding location in the input image, using some form of interpolation to handle fractions of pixels [16]. This requires random access to the input image, which would require significant memory for frame buffering, and introduce additional latency into the transformation.

Therefore, in order to process the streamed input image, it is necessary to use a forward mapping, which determines where in the output image each input pixel maps to. A common problem with using the forward mapping is holes in the output, where there are output pixels with no corresponding inputs. This is not an issue with a foveal mapping because the acuity is always less than or equal to 1, resulting in a many to one mapping.

Associating a single input pixel with each output pixel can result in aliasing. Aliasing may be reduced by averaging all of the input pixels associated each output pixel, which is equivalent to having larger pixels on the image sensor. Conceptually, each input pixel is mapped to an accumulator corresponding to a pixel in the foveal image. If the input pixel spans the boundary of two (or four) low resolution pixels, then that pixel value must be split among two (or four) accumulators.

The block diagram for this is shown in Fig. 4. Each pixel in the input stream is mapped to a corresponding output position using the X and Y map logic blocks. Input pixels that straddle multiple output pixels are weighted accordingly, before being added to an appropriate accumulator. The diagonal edges of the radial mappings mean that it is necessary to maintain accumulators for two rows of pixels. When an output pixel is complete (all associated input pixels have been accumulated), it is passed to the normalization block where the accumulated pixel value is divided by the number of pixels before being stored in the image buffer.

In terms of expensive resources, the pixel splitting block requires 4 multipliers, and the normalization block requires a divider. The division may be pipelined if necessary to meet speed requirements. The forward mapping can be provided as a lookup table in BlockRAM with entries at the input resolution. For the L_2 mapping, the square root may be avoided by having the lookup table give the magnification factor as a function of u^2. An additional 2 multiplications would be required to scale the input coordinates to give the warped pixel position.

If the mapped pixel boundaries are made horizontal and vertical (as they are with the separable mapping, and can be with a small modification to the L_∞ mapping) the logic in Fig. 4 may be halved by using a 2-pass warp. A 2-pass warp first gets the pixels into the correct column, and then when a column is complete it is mapped into the correct row [16]. The two passes may be pipelined, avoiding the need to buffer the intermediate image. The mapping for the warp only needs to be specified at the output resolution, considerably reducing the memory required for storing the mapping. Only one bank of registers and two accumulators are required for the 2-pass warp. The splitting is performed in 2 stages, reducing the number of multipliers there to two.

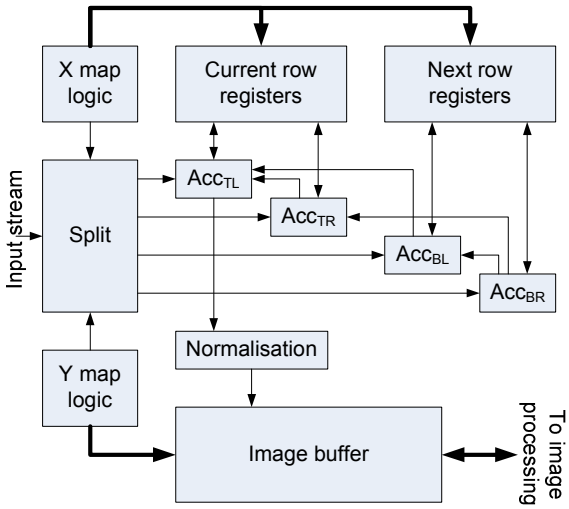

Figure 4. Image warping engine.

The mapping logic is the smallest for the separable mapping, although only slightly more is needed for a 2-pass L_∞ mapping. The L_2 mapping is the most complex. For these reasons, it is proposed that a 2-pass L_∞ mapping be implemented because the pixel shape is more uniform than with the separable mapping.

Multiple foveal maps may be predefined and selected depending on the application. The low storage of the warped images and modest resource requirements for implementing the mapping also mean that multiple warp engines could be implemented in parallel if the processing requires different resolutions for different steps.

V. APPLICATIONS

The proposed framework can be employed in several fields of computer vision such as tracking, pattern recognition, and compression. In the first two cases, the fovea region should be placed on the object of interest that needs to recognised or tracked. As the object moves from the centre of the fovea, this may be detected, and the position of the fovea dynamically adjusted in the next frame to enable the object to be tracked, or to provide a centered image to simplify pattern recognition.

For the last case a saliency detection algorithm is usually employed [17,18]. These algorithms model the human attention model and estimate the regions of the image that attract the human attention and thus should be given more resolution than the other regions.

The definition of a variety of mapping functions can be incorporated in the proposed system (even at run-time) and thus an optimum selection of mapping function can be performed given the application of interest. As a result, an improved performance can be expected compared to existing

approaches that provide a fixed mapping function. Moreover, the resource usage for the mapping scales linearly with the width of the fovea window, allowing the proposed system to be used under a large range of applications with different specifications of the fovea.

VI. CONCLUSIONS

It is shown that a foveal mapping can reduce the image size significantly, reducing the storage and processing time. Several mapping approaches have been explored and it is shown that using an L_∞ distance metric provides the best compromise in terms of uniformity of mapping and resources.

An embedded system, consisting of a sensor and FPGA is possible, enabling high speed, active vision to be implemented without a mechanical pan-tilt platform. Further research is required on the imaging algorithms for processing the foveated images to implement both tracking and recognition algorithms.

VII. REFERENCES

[1] J. Martinez and L. Altamirano, "FPGA-based pipeline architecture to transform Cartesian images into foveal images by using a new foveation approach", in *IEEE International Conference on Reconfigurable Computing and FPGA's*, San Luis Potosi, Mexico, pp 1-10 (2006).

[2] J.C. Wilson and R.M. Hodgson, "A pattern recognition system based on models of aspects of the human visual system", in *International Conference on Image Processing and its Applications*, Maastricht, Netherlands, pp 258-261 (1992).

[3] Y. Xue and D. Morrell, "Adaptive foveal sensor for target tracking", in *Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, vol 1 pp 848-852 (2002).

[4] Y. Cui, S. Samarasekera, Q. Huang, and M. Greiffenhagen, "Indoor monitoring via the collaboration between a peripheral sensor and a foveal sensor", in *1998 IEEE Workshop on Visual Surveillance*, Bombay, India, pp 2-9 (1998).

[5] C. Bandera and P.D. Scott, "Foveal machine vision systems", in *IEEE International Conference on Systems, Man and Cybernetics*, Cambridge, MA, vol 2 pp 596-599 (1989).

[6] V.J. Traver and F. Pla, "Designing the lattice for log-polar images", in *11th International Conference on Discrete Geometry for Computer Imagery*, Naples, Italy, LNCS vol 2886 pp 164-173 (2003).

[7] J. Wei and Z.-N. Li, "On active camera control and camera motion recovery with foveate wavelet transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 23(8) pp 896-903 (2001).

[8] P. Camacho, F. Arrebola, and F. Sadoval, "Shifted fovea multiresolution geometries", in *International Conference on Image Processing*, Lausanne, Switzerland, vol 1 pp 307-310 (1996).

[9] P. Camacho, F. Arrebola, and F. Sadoval, "Multiresolution sensors with adaptive structure", in *24th Annual Conference of the IEEE Industrial Electronics Society*, Aachen, Germany, vol 2 pp 1230-1235 (1998).

[10] Y. Huniyoshi, N. Kita, K. Sugimoto, S. Nakamura, and T. Suehiro, "A foveated wide angle lens for active vision", in *International Conference on Robotics and Automation*, Nagoya, Japan, vol 3 pp 2982-2988 (1995).

[11] H. Hua and S. Liu, "Dual-sensor foveated imaging system", *Applied Optics*, 47(3) pp 317-327 (2008).

[12] R.J. Vogelstein, U. Mallik, E. Culurciello, R. Etienne-Cummings, and G. Cauwenberghs, "Spatial acuity modulation of an address-event imager", in *11th IEEE International Conference on Electronics, Circuits and Systems*, Tel-Aviv, Israel, pp 207-210 (2004).

[13] R. Etienne-Cummings, J. Van der Spiegel, P. Mueller, and M.-Z. Zhang, "A foveated silicon retina for two-dimensional tracking", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(6) pp 504-517 (2000).

[14] P.C. Arribas and F.M.-H. Maciá, "FPGA implementation of a log-polar algorithm for real time applications", in *Conference on Design of Circuits and Integrated Systems*, Mallorca, Spain, pp 63-68 (1999).

[15] V.I. Ovod, C.R. Baxter, M.A. Massie, and P.L. McCarley, "Advanced image processing package for FPGA-based re-programmable miniature electronics", in *Infrared Technology and Applications XXXI*, Orlando, FL, SPIE vol 5783 pp 304-315 (2005).

[16] G. Wolberg, *Digital image warping*. Los Alamitos, CA: IEEE Computer Society Press (1990).

[17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11) pp 1254-1259 (1998).

[18] Y. Liu, C.-S. Bouganis, and P.Y.K. Cheung, "A spatio-temporal saliency framework", in *IEEE International Conference on Image Processing*, Atlanta, GA, pp 437-440 (2006).