

Colour Edge Enhancement

K.T. Gribbon, D.G. Bailey, and C.T. Johnston

Institute of Information Sciences and Technology, Massey University
Private Bag 11-222, Palmerston North, NEW ZEALAND

k.gribbon@massey.ac.nz, d.g.bailey@massey.ac.nz, c.t.johnston@massey.ac.nz

Abstract

It is useful to perform edge enhancement prior to segmentation in order to sharpen edges that have been degraded due to image blur from area sampling. A non-linear filter is often preferred because linear filters can introduce ringing on edges and amplify high-frequency noise. Rank-based approaches are effective but unfortunately rank is not defined for vector-valued pixels in colour images. This paper details the derivation of a colour edge enhancement filter based on distances, which are well defined in vector space. The filter was successfully implemented on an FPGA and some of the issues involved with this are discussed.

Keywords: FPGA, edge enhancement, rank filter, colour image segmentation

1 Introduction

Segmentation is the process of partitioning an image into several disjoint areas, regions or objects containing uniform feature characteristics such as colour or texture [1]. It is an important step in image analysis as it forms the first abstraction of high-level information from raw pixel values [2]. There are a number of well documented segmentation techniques for grey-scale images. Many of these can be extended to colour images by directly applying the segmentation technique to one or more of the individual components in an appropriate colour space and aggregating the results in some fashion [3]. Unfortunately, there is no universally agreed upon combination of segmentation technique and colour space. The choice is application dependent.

One problem that is common to all segmentation

techniques regardless of colour space is image blur. This is an unavoidable result of area sensing within the image capture system. Consider an ideal step edge between two regions, F_1 and F_2 of constant intensity or colour. As a result of area sampling, the boundary pixels are neither one intensity (or colour) nor the other as in Figure 1a. In a grey-scale image, peaks in the histogram (Figure 1b), represent the predominant intensity of the regions F_1 and F_2 .

Edge boundary pixel values are a linear combination of pixel values representative of the two regions, F_1 and F_2 that lie on either side of the edge. This can be expressed as

$$\mathbf{f}_3 = \alpha \mathbf{f}_1 + (1 - \alpha) \mathbf{f}_2 \quad \text{where } 0 < \alpha < 1 \quad (1)$$

where \mathbf{f}_3 is a particular edge boundary pixel value and α is a weight that depends on the exact position of the edge falling on the boundary pixel. While the example in Figure 1 may be solved by selecting an

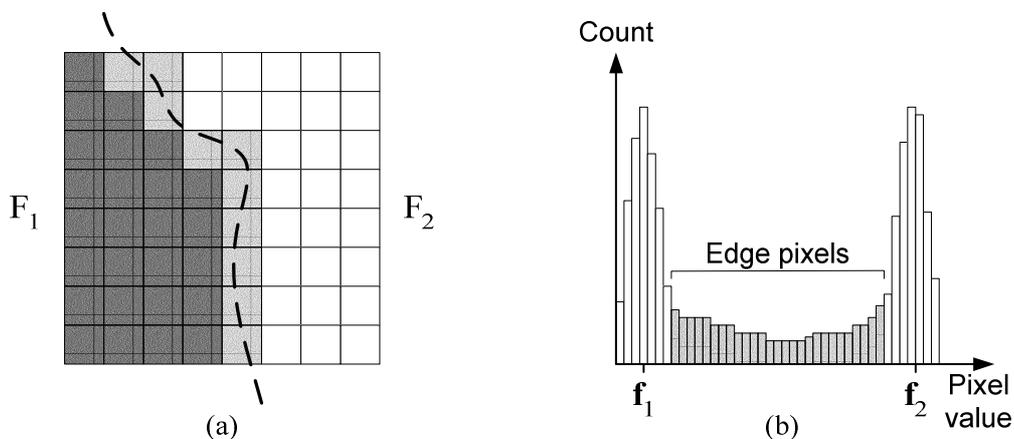


Figure 1: Step edge and corresponding intensity histogram of regions F_1 and F_2 of pixel value \mathbf{f}_1 and \mathbf{f}_2

appropriate threshold, the problem becomes more complicated when segmenting multiple objects with different colours. Here, multiple thresholds are often used but as thresholding does not use spatial information to guarantee that segmented regions are contiguous [3], the edge pixels may be assigned to a region completely unassociated with either of the adjacent regions.

A solution to this problem is to apply an edge enhancement filter prior to segmentation of the image. The filter compensates for the blur by sharpening edges. For grey-scale images, rank-based filters are most suited to this task because they avoid introducing ringing on edges and high frequency noise amplification that results from application of a linear filter [4,5]. In this paper, we extend rank-based edge enhancement to colour images and describe some of the issues encountered when implementing this filter at real-time video rates on an FPGA. Section two describes the operation of the edge enhancement filter for grey-scale images and its extension to colour images. Implementation issues for the FPGA design will be given in section three. Finally, conclusions will be presented in section four.

2 Edge Enhancement Filter

A spatial filter is a local operator whose output value is some function of the surrounding neighbourhood pixel values [6]. The neighbourhood is typically called a window and can be of any shape but is predominantly rectangular or square. Filtering the input image consists of moving the window over the image pixel-by-pixel and calculating the output of the filter at each window location.

One approach to non-linear edge enhancement is to use the values within the local window to estimate the regions F_1 and F_2 on either side of the edge. Two values, $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$ are chosen from within the window that are considered to represent F_1 and F_2 . The value of the centre pixel within the window is then set to the nearer of these values as in equation (2)

$$\hat{\mathbf{f}}_{centre} = \begin{cases} \mathbf{f}_1 & \text{if } \|\mathbf{f}_{centre} - \hat{\mathbf{f}}_1\| < \|\mathbf{f}_{centre} - \hat{\mathbf{f}}_2\| \\ \mathbf{f}_2 & \text{if } \|\mathbf{f}_{centre} - \hat{\mathbf{f}}_1\| \geq \|\mathbf{f}_{centre} - \hat{\mathbf{f}}_2\| \end{cases} \quad (2)$$

where $\hat{\mathbf{f}}_{centre}$ is the output pixel value and $\|\mathbf{f}_i - \mathbf{f}_j\|$ is the distance between \mathbf{f}_i and \mathbf{f}_j . For grey-scale images this is just the absolute difference, but for colour images, the L_2 or Euclidean norm is used. For grey-scale images, this effectively classifies the pixels in the valley region of Figure 1b to either of the peaks.

2.1 Rank-based edge enhancement filter

For grey-scale images, if there is only one edge within the window, then the minimum and maximum pixel values provide good estimates of the pixel values in the regions F_1 and F_2 . In the presence of noise, however, the minimum and maximum are most likely to be corrupted by the noise and therefore become less representative of the regions F_1 and F_2 . It has been shown in [5] that selecting rank positions closer to the median reduces this noise sensitivity. As the output pixel value is within the window, no new pixel values are generated and ringing is prevented [5].

Ranking enables selection of $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$ as long as the window encompasses an edge. If the centre of the window is not an edge pixel, then it will be very close in value to one or the other of the selected representatives and is part of the region. If the centre pixel is an edge pixel, then there will be at least one and usually two or three pixels from each side of the edge within the window, enabling their selection as $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$. The boundary pixel is therefore assigned to one or other of the two regions.

2.2 Vector Ranking

Extending ranking to colour images is not straightforward as there are no natural concepts of rank or ordering for p -channel data¹. For ranking to be defined, the multi-channel data must be reduced to a scalar value that is a function of one or more of the component values of the pixel [7], according to

$$d_i = d(\mathbf{x}_i) \quad (3)$$

where d_i is the output scalar value, d is some reduction function $d: \mathbb{R}^p \rightarrow \mathbb{R}$, and \mathbf{x}_i is the multi-channel input pixel for $1 \leq i \leq N$ pixels in the window.

There are a number of different forms of the reduction function in equation (3). One method is to project the vector onto a single component. For colour images, this may include choosing one of the channels from RGB space or the intensity channel from HSI space as the corresponding one-dimensional subspace. Another method involves interleaving the bits from each colour channel into a single scalar value [8]. For RGB space this can be achieved using

$$R_{N-1}G_{N-1}B_{N-1}R_{N-2}G_{N-2}B_{N-2} \dots R_0G_0B_0 \quad (4)$$

where N is the number of bits per channel. The performance of these methods is highly dependent

¹ For processing of colour images, p is typically three i.e. RGB, HSI, and YUV colour spaces. However, there are colour spaces with other dimensions such as CMYK ($p = 4$)

upon the correlation between the colour channels. In addition the above methods require a priori information concerning the relative importance of the different components in the input image. Even when this is available, artefacts can appear in the output image when colour spaces with highly correlated intensity and colour information, such as RGB space, are used [9].

Therefore a scalar metric in vector space must be used so that correlation between the colour channels can be taken into account. [10] suggested using the aggregated distance of each point from all other points as one form of the reduction function in equation (3). Distance is a well defined function between two vectors and can be calculated by taking the vector norm. While the vectors are not able to be ranked in any meaningful way, [9,11] showed that the colour median can be defined as the vector that minimises the sum of the distances to all other vectors for a window of N vector-valued pixels, $\mathbf{x}_i, i = 1, 2, \dots, N$.

$$\mathbf{x}_{med} = \left\{ \mathbf{x}_j \mid \min_j \sum_{i=1}^N \|\mathbf{x}_j - \mathbf{x}_i\| \right\} \quad (5)$$

where $\|\cdot\|$ is usually the L_2 or Euclidean norm.

2.3 Colour Edge Enhancement Filter

Although the median can be defined for a set of vectors such as colour pixels from within a 2-D window, ranking cannot be used to determine $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$ because minimum and maximum are not defined for vectors. However, consider a 1-D three-element window such as Figure 2a or Figure 2b situated over a multi-channel edge. From equation (1) the centre pixel value will be between the other values in the window. Thus at a multi-channel edge, the pixel value at the centre of the window, \mathbf{f}_3 , is always the vector median and the pixel values on either side of the centre. \mathbf{f}_1 and \mathbf{f}_2 , can therefore be considered as estimates for $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$. [12] have made use of this in their multi-channel edge enhancement filter, by performing the enhancement of equation (1) only when \mathbf{f}_3 is the vector median within the window.

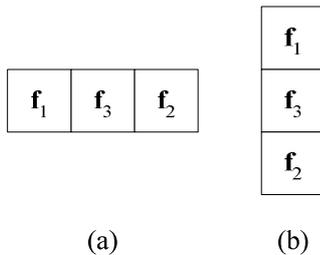


Figure 2: 1-D three-element windows

For a three-element window from Figure 2, equation (5) becomes

$$\mathbf{f}_3 = \mathbf{x}_{med} \quad \text{if} \quad \begin{cases} \|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 - \mathbf{f}_2\| < \|\mathbf{f}_1 - \mathbf{f}_2\| + \|\mathbf{f}_1 - \mathbf{f}_3\| \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 - \mathbf{f}_2\| < \|\mathbf{f}_2 - \mathbf{f}_1\| + \|\mathbf{f}_2 - \mathbf{f}_3\| \end{cases} \quad (6)$$

Equation (6) simplifies to

$$\mathbf{f}_3 = \mathbf{x}_{med} \quad \text{if} \quad \begin{cases} \|\mathbf{f}_3 - \mathbf{f}_2\| < \|\mathbf{f}_1 - \mathbf{f}_2\| \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| < \|\mathbf{f}_2 - \mathbf{f}_1\| \end{cases} \quad (7)$$

Using this result and in an analogous fashion to equation (2), the output of the filter is given by

$$\hat{\mathbf{f}}_3 = \hat{\mathbf{f}}_3(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3) = \begin{cases} \mathbf{f}_1 & \text{if} \quad \begin{cases} \mathbf{f}_3 = \mathbf{x}_{med} \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| < \|\mathbf{f}_3 - \mathbf{f}_2\| \end{cases} \\ \mathbf{f}_2 & \text{if} \quad \begin{cases} \mathbf{f}_3 = \mathbf{x}_{med} \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| \geq \|\mathbf{f}_3 - \mathbf{f}_2\| \end{cases} \\ \mathbf{f}_3 & \text{otherwise} \end{cases} \quad (8)$$

where $\hat{\mathbf{f}}_3$ is the new centre pixel value, and \mathbf{f}_1 , \mathbf{f}_2 and \mathbf{f}_3 are the pixel values from Figure 2. Equation (8) effectively states that if \mathbf{f}_3 is the vector median of the three-element window, the output is the pixel value closest to \mathbf{f}_3 in terms of distance.

The filter described in equation (8) has a number of properties:

- It is similar to a rank filter in that it generates no new values. The output is one of the pixels in the window.
- By definition of equation (8) the filter enhances edges in one direction only. Two-dimensional enhancement requires two passes and a rotation of the window structure by 90 degrees between the passes to enhance edges in both the vertical and horizontal directions.
- Isolated noise pixels or fine line structures are not affected by the filter because in this case the centre pixel is not the median, so no enhancement occurs. Noise in the image can be smoothed by first applying a 1-D three-element median filter before filtering with the edge enhancement filter [12].

2.4 Two-dimensional Colour Edge Enhancement

The edge enhancement filter described above may be extended to 2-D filtering. Extension to two dimensions requires determination of the predominant direction of the edge. A 'cross' type window structure, shown in Figure 3 is used for this purpose. It is basically a combination of the two 1-D three-element window structures of Figure 2.

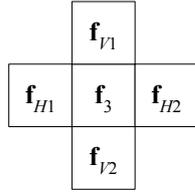


Figure 3: Window structure for 2-D filtering

The window in Figure 3 is scanned over the entire image. For horizontal edges, \mathbf{f}_{H1} and/or \mathbf{f}_{H2} may themselves be edge pixels, rather than good representatives for the regions F_1 and F_2 . As a result, the distance $\|\mathbf{f}_{H1} - \mathbf{f}_{H2}\|$ is less than $\|\mathbf{f}_{v1} - \mathbf{f}_{v2}\|$ because \mathbf{f}_{v1} and \mathbf{f}_{v2} will be on opposite sides of the edge. A similar argument may be made for vertical edges where $\|\mathbf{f}_{v1} - \mathbf{f}_{v2}\|$ is less than $\|\mathbf{f}_{H1} - \mathbf{f}_{H2}\|$. Therefore, by comparing these distances we can select either the horizontal or vertical window as appropriate. The output of the 2-D filter is given by

$$\hat{\mathbf{f}}_3 = \begin{cases} \hat{\mathbf{f}}_3(\mathbf{f}_{v1}, \mathbf{f}_{v2}, \mathbf{f}_3) & \text{if } \|\mathbf{f}_{H1} - \mathbf{f}_{H2}\| < \|\mathbf{f}_{v1} - \mathbf{f}_{v2}\| \\ \hat{\mathbf{f}}_3(\mathbf{f}_{H1}, \mathbf{f}_{H2}, \mathbf{f}_3) & \text{if } \|\mathbf{f}_{H1} - \mathbf{f}_{H2}\| \geq \|\mathbf{f}_{v1} - \mathbf{f}_{v2}\| \end{cases} \quad (9)$$

2.5 Edge Test

While the centre pixel value, \mathbf{f}_3 satisfying equation (1) will be the vector median, the converse is not necessarily true. The vector median is testing if \mathbf{f}_3 is closer to both the window endpoints \mathbf{f}_1 and \mathbf{f}_2 than the distance between \mathbf{f}_1 and \mathbf{f}_2 . Consider all of the points that are within a distance $\|\mathbf{f}_1 - \mathbf{f}_2\|$ of \mathbf{f}_1 in 3-D vector space. Such points will be within a sphere of this radius, centred on \mathbf{f}_1 . Consider a similar sphere centred on \mathbf{f}_2 . All points, \mathbf{f} , inside the intersection of these spheres will be the median when taken together with \mathbf{f}_1 and \mathbf{f}_2 .

This intersection is a disc-shaped region in 3-D vector space, the cross-section of which is illustrated in Figure 4. Clearly, points in \mathbf{f} can deviate significantly from the path between \mathbf{f}_1 and \mathbf{f}_2 and still be the median. The vector median test of equation (6) is therefore not very discriminating.

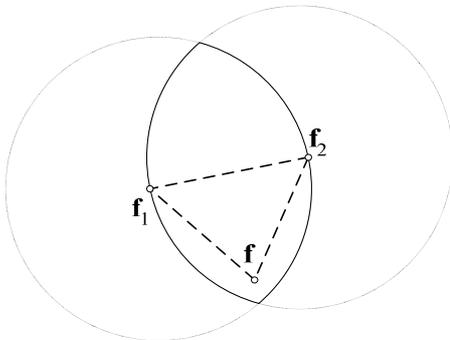


Figure 4: Median region between 2 points \mathbf{f}_1 and \mathbf{f}_2

Rewriting equation (1) in relative terms (distances) rather than pixel values gives

$$\|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 + \mathbf{f}_2\| = \|\mathbf{f}_1 - \mathbf{f}_2\| \quad (10)$$

Any small deviation of \mathbf{f}_3 from the path between \mathbf{f}_1 and \mathbf{f}_2 will violate this equation. Such a deviation may be the result of noise, or even inevitable quantisation errors. Rearranging and adding a threshold to account for this gives

$$\|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 - \mathbf{f}_2\| - \|\mathbf{f}_1 - \mathbf{f}_2\| < T \quad (11)$$

where T is some scalar threshold. In 3-D vector space, equation (11) describes the interior of an ellipsoid with foci at \mathbf{f}_1 and \mathbf{f}_2 . Replacing the test in equation (8) with equation (11) gives

$$\hat{\mathbf{f}}_3 = \begin{cases} \mathbf{f}_1 & \text{if } \begin{cases} \|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 - \mathbf{f}_2\| - \|\mathbf{f}_1 - \mathbf{f}_2\| \leq T \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| < \|\mathbf{f}_3 - \mathbf{f}_2\| \end{cases} \\ \mathbf{f}_2 & \text{if } \begin{cases} \|\mathbf{f}_3 - \mathbf{f}_1\| + \|\mathbf{f}_3 - \mathbf{f}_2\| - \|\mathbf{f}_1 - \mathbf{f}_2\| \leq T \\ \text{and} \\ \|\mathbf{f}_3 - \mathbf{f}_1\| \geq \|\mathbf{f}_3 - \mathbf{f}_2\| \end{cases} \\ \mathbf{f}_3 & \text{otherwise} \end{cases} \quad (12)$$

The tests of equations (6) and (11) are compared in Figure 5.

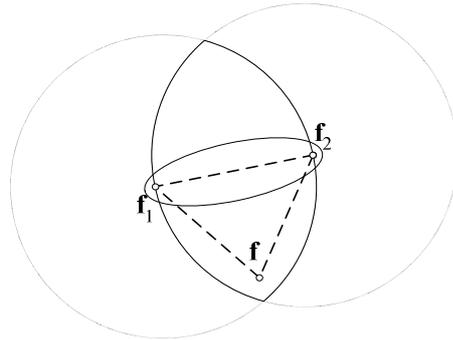


Figure 5: Comparison of regions formed by equations (6) and (11)

3 Implementation Issues

An example FPGA implementation of the edge enhancement filter detailed above has been completed. Real-time enhancement at video rates constrains the design into performing all of the required calculations for each pixel at the pixel clock rate (one pixel every 40 ns for VGA output). A pipelined approach is thus needed to allow the filter output to be evaluated over several stages. Figure 6 shows the pipeline for the edge enhancement filter. In addition, memory bandwidth constraints mean that only a single pixel can be accessed per clock cycle [13]. Thus the requirement for simultaneous access to five pixels (Figure 4) from the input image per clock cycle requires that two rows of image data be buffered. Additional hardware for row buffering and scanning the window must also be implemented within the FPGA.

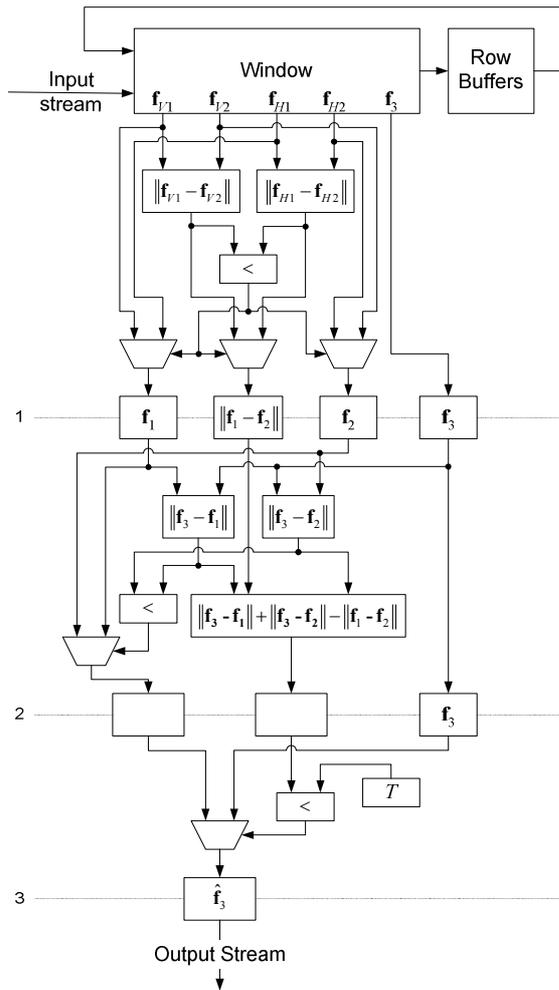


Figure 6: Pipeline for edge enhancement filter

The first pipeline stage calculates the distance between the endpoints of the 1-D horizontal and vertical windows that make up the window structure in Figure 4 and selects the appropriate 1-D window. In stage two, the remaining distances, $\|f_3 - f_1\|$ and $\|f_3 - f_2\|$ are calculated and the nearest representative selected. In the final stage, the three distances are used to determine whether or not the centre is a boundary pixel and selects the output accordingly.

3.1 Distance metric

The choice of the distance measure used in the edge enhancement filter becomes important when real-time processing is concerned. Euclidean distance (L_2 norm) is preferred due to its isotropic behaviour [14]. Unfortunately the square and square root operations of the Euclidean distance measure make a real-time hardware implementation very expensive.

Other distance measures include city-block distance (L_1 norm) and chessboard distance (L_∞ norm). These metrics are simpler to calculate than the Euclidean distance but are non-isotropic, which affects the region formed by the test in equation (11). Points of

constant L_1 distance form a square in 2-D vector space and the surface of a cube in 3-D vector space, while points of constant L_∞ distance form a diamond shape in 2-D vector space and the surface of an octahedron in 3-D vector space.

Even in the presence of noise f_3 lies approximately on the path between f_1 and f_2 and predominantly in the same direction. Therefore distortion introduced by the non-isotropic nature of the L_1 and L_∞ norms affects the three points in the same way and therefore will have little effect on the relative distances. With real-time processing constraints in mind it is useful to calculate the less costly L_∞ distances.

4 Summary and conclusions

It is useful to perform edge enhancement prior to segmentation in order to sharpen edges that have been degraded due to image blur from area sampling. A non-linear filter is often chosen because linear filters can introduce ringing on edges and amplify high-frequency noise. A rank-based approach was derived but unfortunately rank is not defined for vector-valued pixels in colour images.

The vector median however, can be used as the basis of a 1-D colour edge enhancement filter [12]. This paper extends this to two dimensions by comparing distances in a 2-D window. It was shown that the vector median is not a very discriminating test for an edge. Consequently, enhancement can occur erroneously. This paper presented a stricter test, given by equation (11), based on the fundamental definition of an edge, given by equation (1).

An example FPGA implementation of the edge enhancement filter for colour images has been successfully implemented. The paper discussed the issues involved in selecting an appropriate distance metric for real-time processing at video rates. The L_1 and L_∞ norms, although less computationally intensive, are also non-isotropic in nature and distort the edge test. However, since all pixels lie on the same line and in the same direction such distortion will be minimal. Therefore their use in this real-time implementation can be justified.

5 Acknowledgments

The Celoxica University Programme for generously providing the DK2 Design Suite.

6 References

- [1] Castleman, K. R. *Digital Image Processing*, New Jersey: Prentice-Hall, 1996.
- [2] Park, S. H., Yun, I. D., and Lee, S. U., Color image segmentation based on 3-D clustering: morphological approach *Pattern Recognition*, vol. 31, no. 8, pp. 1061-1076, 1998.
- [3] Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J., Color image segmentation: advances and prospects *Pattern Recognition*, vol. 34, no. 12, pp. 2259-2281, 2001.
- [4] Bailey, D. G., Hodgson, R. M., and McNeill, S. J., "Local filters in digital image processing," *Proceedings of the 21st National Electronics Conference*, Christchurch, New Zealand, pp. 95-100, 1984.
- [5] Hodgson, R. M., Bailey, D. G., Naylor, M. J., Ng, A. L. M., and McNeill, S. J., Properties, implementations and applications of rank filters *Image and Vision Computing*, vol. 3, pp. 3-14, 1985.
- [6] Gonzales, R. C. and Woods, R. E. *Digital Image Processing*, Upper Saddle River, New Jersey: Prentice-Hall, 2002.
- [7] Comer, M. and Delp, E., Morphological Operations for Color Image Processing *Journal of Electronic Imaging*, vol. 8, pp. 279-289, 1999.
- [8] Chanussot, J., Paindavoine, M., and Lambert, P., "Real time vector median like filter: FPGA design and application to color image filtering," *Image Processing, Proceedings. ICIP 99. International Conference on*, pp. 414-418 vol.2, 1999.
- [9] Astola, J., Haavisto, P., and Neuvo, Y., Vector median filters *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678-689, 1990.
- [10] Barnett, V. , Ordering of Multivariate Data *Journal of the Royal Statistical Society Series a-Statistics in Society*, vol. 139, pp. 318-354, 1976.
- [11] Astola, J., Haavisto, P., Heinonen, P., and Neuvo, Y., "Median type filters for color signals," *Circuits and Systems, 1988., IEEE International Symposium on*, pp. 1753-1756 vol.2, 1988.
- [12] Tang, K., Astola, J., and Neuvo, Y., Multichannel edge enhancement in color image processing *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 5, pp. 468-479, 1994.
- [13] Gribbon, K. T. and Bailey, D. G., "A Novel Approach to Real-time Bilinear Interpolation," *Second IEEE International Workshop on Electronic Design, Test and Applications*, Perth, Australia, pp. 126, 2004.
- [14] Barni, M., Buti, F., Bartolini, F., and Cappellini, V., A quasi-Euclidean norm to speed up vector median filtering *Image Processing, IEEE Transactions on*, vol. 9, no. 10, pp. 1704-1709, 2000.