An assessment of the Edge Detection & Region Segmentation facilities as currently offered within the Image Processing package for MathCAD and those offered by SHV.

Introduction.

Having been actively connected with image handling & processing for approaching 40 years, I was recently delighted to have the opportunity to explore the Image Processing package currently provided as an 'add-on' for use with MathCAD. In general I find this package to comprise a satisfyingly varied set of functions for image manipulation & possible interpretation. However, when I come to the section headed 'Edge Finders', I find a need to make a number of major observations, as follows:-

- The first observation concerns a comment which appears in a number of places within the section - viz. "Edge detection is used to enhance image readability for certain types of features that depend on boundaries". A point I would like to make here is that by far the majority of natural images are primarily ***composed*** of 'boundaries' - either, on the one hand, boundaries between zones or 'regions' of tolerably constant or very slowly varying brightness &/or colour or, on the other hand, effective boundaries between zones of different texture! So I consider that edge detection is a major imperative part of ***any*** normal image analysis, rather than, as might be inferred by the foregoing wording, on relatively rare occasions!
- The majority of edge finders offered in the MathCAD package are based on one or other form of sensing either preferentially horizontal or vertical edges or an RMS composite of the two. Since local edges in typical images may occur at virtually any orientation, this restriction must inevitably lead to major orientation-dependent effects on outputs. Even the few edge finders which attempt to allow for orientation effects appear only to sense some form of edge 'strength' based on a combination of the outputs from horizontal & vertical components, thereby being essentially ***scalar*** output parameters with no encoded vector orientation component.
- The 'Laplacian Edge Finders' are essentially 2D second derivative operators. As such, they are elsewhere widely publicised as being very useful for their property of yielding a zero crossing at the location of each edge in the image. Whilst this is true for 'ideal' images - i.e. images having coarse & rather simple structure - it can readily be demonstrated that, for complex 2D structure, the zero crossings generated may not coincide with edges at all (see, for example, Ref. 1, Chapter 4.1 and Figs. 4.1, 4.2 & 4.17)! In addition, for certain types of image structure 'false' edges may also be created (e.g. Ref. 1, Fig. 4.3)!
- If Laplacian Edge Finders are to be used at all, the best form of function in the 'Laplacian family' to yield an approximation to 'circular symmetry' in its operation is one which is not provided by the MathCAD package - viz. what might be labelled 'Laplace12(M)', where the weightings are

  $$\begin{matrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{matrix}$$

  This form of Laplacian, if used following a suitable 2D Gaussian blurring operation, essentially amounts to the DOG edge detection concepts popularised by David Marr *et al* in the early 1980's (the said DOG approach being, of course, subject to the same potential problems as already mentioned for Laplacians). See, for example, Ref. 1, Chapter 3.7 for some discussion on this.
- The Canny Edge Detector is a much more powerful & sound approach, being based on a 2D Gaussian convolution followed by horizontal & vertical ***first*** derivative processing and thresholding. As such, it potentially provides a much more reliable edge locator. I have recently

been given to understand that this detector has tended to become the standard method of edge detection at this time. However, as far as I can ascertain, even this detector is essentially ***scalar*** in its outputs.

The main purpose of this report is to start with the premise that the Canny detector is currently the 'favourite' approach for edge detection, together with region segmentation being an entirely separate operation. These capabilities are then compared with what ***should*** be possible.

An assessment of the capabilities of the Canny Edge Detector

It is now over 20 years since the Canny Edge Detector was first proposed as an important improvement to the various techniques for edge detection available at that time (Ref. 2). During the intervening years it appears that the said edge detector has become essentially the preferred standard method, superceding a considerable variety of edge detection methods available prior to that time (most of which are, I am delighted to find, nevertheless still available for experimental use within the MathCAD Image Processing package). Whilst it is certainly true that the Canny approach is substantially preferable to any of the previously existing approaches, it is my strong personal opinion that it still falls far short of what ***should*** be achievable. My reason for this opinion is as a result of having carried out a widely varying, critical study of what are essentially edge detection capabilities of human vision (which is itself essentially a 2D discrete sampling imaging system) - see particularly Ref. 3, Chapters 4 & 5 for a rather detailed summary of all this. Against this range of capabilities, even the Canny edge detection falls far short of what is potentially possible in several directions as listed below. These shortfalls, originally the subject of a rather thorough ***theoretical*** appraisal previously carried out by me in the mid 1980's, have been confirmed by recent practical studies working with the Canny Edge Detector as packaged as part of the Image Processing additions to MathCAD.

- It should be possible to measure the local position of a local edge fragment to something better than 0.1 spatial sampling increments (i.e. better than 0.1 pixels). This compares with local edge position being sensed by the Canny detector (and virtually all other commercial edge detectors) to the nearest pixel only.
- local edge orientation should be able to be sensed (directly) to better than 1 or 2 degrees. It seems that, for the Canny detector, local edge orientation is only estimated, at best, as nearer horizontal or nearer vertical (with no output specifically to state this). This is despite the fact that Canny's original paper acknowledged the desirability for orientation to be sensed to the nearest 30 degrees from a mathematical standpoint!
- local edge contrast should be recordable to a high level of sensitivity. Although the magnitude of local edge contrast must be sensed somewhere within the Canny algorithm in order to compare with input threshold values, it is only apparent in the output in terms of the dual (hysteresis) threshold. It is presumed that one reason for this is that, in the absence of an accurate & balanced means of sensing local edge orientation, any output of local edge strength would be subject to some very significant uncertainty, dependant on the actual local edge orientation present.
- it should be possible to measure local edge sharpness to a reasonable accuracy. Human vision does this rather well. In the Canny detector there is no apparent ability to sense local edge sharpness. Again this is perhaps of no surprise, since again such a measure would require an adequate means of dealing with circular symmetry.

- it should be possible to estimate local edge curvature directly. It is strongly believed that such an ability is one of the most important components in human visual ability to recognise the form of objects (see Ref. 3, Chapter 5, also Chapters 12.6 & 13). Since there is no direct measure of local edge orientation available from the Canny detector, there is essentially ***no*** potential capability of measuring local curvature.
- where there are a pair of spatial images overlaid - as in stereo (binocular) viewing - it should be possible to measure the local differential edge locations to better than about 0.1 pixels. This is the sort of sensitivity which is evident in human vision by which by far the majority of our awareness of stereo depth is possible (see, for example, Ref. 1, Chapter 10.1 & Ref. 4). Alternatively, in the case of temporal image sequences, it should be possible to measure local motion orthogonal to the local edge between consecutive frames to better than 0.1 pixels. Again, in human vision there is an acute sense of awareness of very subtle local movements (although I know of no thorough experimental study which has confirmed the magnitude in ***absolute*** terms). In the Canny detector (or, for that matter, any of the other commercially available edge detectors), since local edge position is only sensed to the nearest pixel and there is no direct measure of local orientation, any attempt to provide a measure of local edge displacement between pairs of frames (for either local stereo depth discrimination or local motion sensing) must, of necessity, be very crude (i.e. in terms of integer numbers of pixels).

So what can be done about this series of substantial shortfalls in practical against potential performance? The answers to by far the majority of these shortfalls have been available ***conceptually*** for some 15 years, although a practical, Windows-friendly package has only become available during the past few months. This latter package is as part of a computer vision facility SHV (Simulated Human Vision), the history of which will be discussed below.

The evolution of SHV (Simulated Human Vision).

Starting some years before Canny first published his improved edge detector - and based on previously collected performance data for the human eye, together with emerging knowledge on the detailed physiology of the human eye and visual tract - work was progressing at Sowerby Research Centre of British Aerospace (B.Ae.) to endeavour to construct various algorithms which simulated the early processes of human vision. For this work, since it was quickly realised that the detector matrix of the eye (the retina) was essentially a ***hexagonal*** matrix, a means had to be found whereby processing was effectively carried out in hexagonal space whilst still being ***practically*** carried out using conventional square matrices. When the Canny Edge Detector publication became known, the eye simulation work was already well advanced and was showing rudimentary capabilities of achieving the high & versatile sensitivities of human vision on the computers of the day. As such, it was considered absolutely necessary to assess the relative similarities & differences of our approach and that of Canny. A short summary of the conclusions from that exercise are given in Chapter 4.6 of Ref. 1. It was found that the basic starting points for the two approaches were very similar but, as things stood at the time, we felt that our approach was being taken considerably further than the reported work of Canny.

As our work progressed, it was reported in a series of papers and, largely through lack of available effort at the time, any possible further developments of the Canny approach were not explored by us. It was somewhat of a surprise, therefore, that, coming across the Canny approach as a widely used edge detector some 20 years later, I find that what is now offered is essentially the same as that of which I became aware in the mid 1980's!

By 1990 our research & development work was essentially complete and in 1992 the whole set of inter-related concepts were published under one cover as Ref. 1. By this time it was considered that the sum total of inter-related capabilities - which were ***all*** basically dependent on the underlying approach to ***edge*** detection - not only explained many of the known (and sometimes almost unbelievable) properties & capabilites of the human visual system, but also allowed a convenient means of inter-relating edge detection & region segmentation - previously generally considered as two opposing approaches to scene description. It was also found possible to demonstrate a near identity of processing functions for the handling of paired-frame analysis of either stereo (spatial) disparities or optical flow (temporal) disparities.

Now of course it had to be expected that, since the computer vision processes were essentially being carried out on a serial computer, whereas human vision had the big advantage from a processing standpoint of having a very largely ***parallel*** spatial processing capability, the total processing to achieve the very high sensitivity & versatility was distinctly slow. Hence in those days it was impractical to consider the total processing of images larger than about 256 x 256 pixels at most - and even then the processing time was of the order of tens of minutes. However, over the intervening years - with the massive increases in computer speed and memory availability, together with potential implementation of modern machine code routines - both the processing speed and the size of images which could be handled have increased greatly.

At the end of 1990 the present author - who was mainly responsible for the original development of the computer vision simulation software and also was the author of the composite book (Ref. 1), together with the previous book covering human visual performance limits (Ref. 3) - retired from full-time work. However, with the aid of his son - himself a computer consultant - he has been able to implement the majority of the concepts covered in the book in a PC friendly form, as part of this implementation his son having made use of DLL's for further streamlining of the processing. The result is that we have recently been able to produce a user-friendly suite of software which carries out edge detection which, in turn, yields a tabulation of sub-pixel local edge position (to better than 0.1 pixels), local edge orientation (to better than 1 degree), local edge contrast, local edge sharpness, local edge chromaticity difference (where appropriate) and, for spatial or temporal image pairs, local edge displacement orthogonal to the local edge orientation to better than 0.1 pixels. Furthermore it is found that it is possible to consider such total processing of images comprising several million pixels in a matter of a few seconds at most.. It is readily possible to generate appropriately coded pictorial representations of any of these tabulated parameters other than the sub-pixel positional data on a 'nearest pixel' basis by means of a set of small support programs. In addition, in order to permit full appreciation visually of the ***sub-pixel*** positional & orientation capabilities, a dedicated supplementary piece of software has been developed which maps the edge data to the nearest one tenth pixel, together with the local orientation, at a linear scale up of X10. This software is also provided as part of the complete suite.

As a direct resultant of the measurement of the set of local edge characteristics during the actual edge detection process, a number of other processes can also be carried out automatically actually during the edge detection. These include region segmentation based directly on the edges detected (with region closure being semi-intelligently carried out where necessary based on the first derivatives of 2D grey level distribution in the vicinity of local edges) and, for paired frame analysis, local fragmentary stereo or optical flow maps. In addition, supplementary software has been developed which permits ready derivation of region segmentation maps (in terms of either region boundaries, 'cartoon' reconstructions of original input images or region labels - the latter similar to the output from 'reg_grow' in MathCAD), edge curvature maps and, for paired frame analysis, forms of region segmentation based on local flow or local stereo disparity.

For reliable region maps to be generated by SHV in the case of complex images containing both high & low contrast important structure, it is desirable to set the edge detection threshold to be very low. For such images this is considered to be the only really reliable way to ensure that artifactual region segmentation doesn't take place. This latter problem is akin to the sort of problem which motivated Canny to develop his dual threshold technique in order to provide a capability for 'bridging' local gaps in profiles where the strength of the first derivative fluctuated about a chosen single threshold. Whilst the Canny 'hysteresis' threshold approach has considerable merits for images where there is local edge strength fluctuation, but the mean edge strength remains roughly constant, it can be far from satisfactory where a complex image has structure of both high and low contrast which is all of importance.

The low threshold region maps thus generated by SHV from typical complex scenes tend initially to be very rich in detail, frequently involving several hundred or even several thousand regions. A method was therefore developed (and described in the Ref. 1, Chapter 15.8), whereby progressive region merging (or thinning) could be carried out in a semi-intelligent manner based on an understanding of the size / contrast threshold relationships known to exist in human vision and as modelled in Ref. 3, Chapter 7 & Ref. 5. From a reading and limited practical study of the 'reg_grow' function available in the MathCAD image processing suite, it would appear that this latter function uses rather similar (semi-intelligent) properties to the SHV 'merging' function. However, there it would seem that any similarity ends. The 'reg_grow' function appears to be specifically designed to form regions ***directly*** from an original image, based solely on the distributed local grey level properties (i.e. essentially regardless of local edge structure). The SHV 'merge' function, on the other hand, is designed to interrogate the region borders derived originally from edge analysis and to combine, progressively, adjacent regions which are 'visually similar' based on the properties of the common boundary. This is believed to be more closely related to region merging carried out at higher levels of the human visual tract.

One other major difference seen between the SHV 'merge' function and the MathCAD 'reg_grow' function is that the only output from the 'reg_grow' function appears to be a region label map, whereas in SHV processing - both before and after carrying out a region merging process - a full tabular record of the properties both of all regions and all region boundaries is stored, from which it is possible to reconstruct either a region label map, a region boundary map or a 'cartoon' representation of the original input image. These tabular records include some very simple statistics for each region segmented - in particular the mean grey level (and chromaticity, where appropriate) and the size.

Having taken the SHV merge step in order to reduce the number of regions to a manageable number, it should then be possible to employ the existing MathCAD function 'shape_features' in order to generate a number of useful additional local feature (i.e. region) characteristics for virtually any original image or for any selected features from an original image in order to supplement the ***basic*** region statistics which are automatically stored as part of the SHV processing.

Curvature maps - which, it appears, can ***only*** be readily generated from an edge detection function having the richness of output available from SHV - are believed to be an absolutely essential data bank for permitting any sensible sort of automatic form ***recognition*** or ***identification***. See Ref. 3, Chapters 5, 12 & 13, plus Ref. 1, Chapter 12, for substantial discussion on this topic. These higher level image interpretation tasks are believed to be a necessary part of the majority of reliable practical image analysis, the simple ***detection*** process being very often entirely inadequate on its own!

High fidelity stereo or optical flow maps are further important outputs from many scenes which involve either 3D spatial distributions or any temporal motion. If such distributions are limited to magnitudes of several pixels, together with a necessity for point to point 'matching', then extraction of useful data can be both highly cumbersome and of dubious accuracy. Substantial discussion & demonstration of what can be achieved from edge detection which can measure both sub-pixel displacement and local edge orientation is to be found in Ref. 1, Chapters 5, 9 & 10.

In keeping with other known properties of human vision, the first derivative processing in SHV is carried out using only ***partial*** derivative early data (that is, the positive and negative components of the conceptually Laplacian process are initially processed independently). This permits the derivation of both first & second derivative early outputs for further processing (thus providing the options for further processing which are the types of output from both main types of classical edge detector - i.e. first derivative or second derivative - see, for instance, Ref. 1, Chapters 2.5 & 3.7 for discussion). It has been found that, the ***first*** derivative outputs (effectively as for Canny's detector, but sensed and analysed at 30 degree increments of orientation - a facility only made fully possible by the pseudo-hexagonal array processing developed for SHV - see Ref. 1, Chapter 3.4) are the most appropriate for basic edge sensing. On the other hand, the ***second*** derivative outputs (with maximum gradients cued by the first derivative peaks - see Ref. 1, Chapter 5.10 - rather than their own zero crossings, in order to avoid the potential problems associated with classical second derivative operators for certain complex image situations) are most appropriate for stereo & motion sensing.

In addition to the various basic types of processing summarised above, it is possible to carry out entire processing with inclusion or exclusion of a number of local interactive weighting processes which are known to operate within the human visual tract, such options yielding a number of different resolution outputs permitting a 'play-off' of signal against noise. Furthermore, an option also exists for single frame processing of ***line*** images, as opposed to ***edge*** images, by means of bypassing the effectively 2D first derivative step of the main processing but otherwise using all other parts of the total process. This option provides what is considered to be a capability normally totally missing from classical edge detection facilities, but which is very evidently readily available in human vision. After all, a square, for instance, is perceived as a square, whether it is an outline image or a solid shape!

If desired, it is possible to generate 'part processed' bitmap images at a variety of stages through the complete processing chain, these 'part processed' representations being available either in the pseudo-hexagonal image format or a 'recovered' square matrix format. Such 'part processed' images include the luminance distribution as offered to the main processes (i.e. after any early preprocessing such as blurring, scaling etc.), a 2D first derivative map, a 2D second derivative map, effects of secondary local weighting functions, initial fragmentary edge sensing, local edge strength distribution, local edge sharpness distribution and, for paired-frame processing, local edge displacement (disparity). Of course, whereas the final output edge data are capable of generating sub-pixel 'super-resolution' output images, all such partially processed images have, of necessity, to be limited in ***spatial*** representation to 'nearest pixel'.

Some practical examples of comparisons between outputs from a Canny detector process and an equivalent processing using SHV.

There follow a few practical demonstrations of outputs from the MathCAD version of the Canny detector and the equivalent outputs from SHV processing.

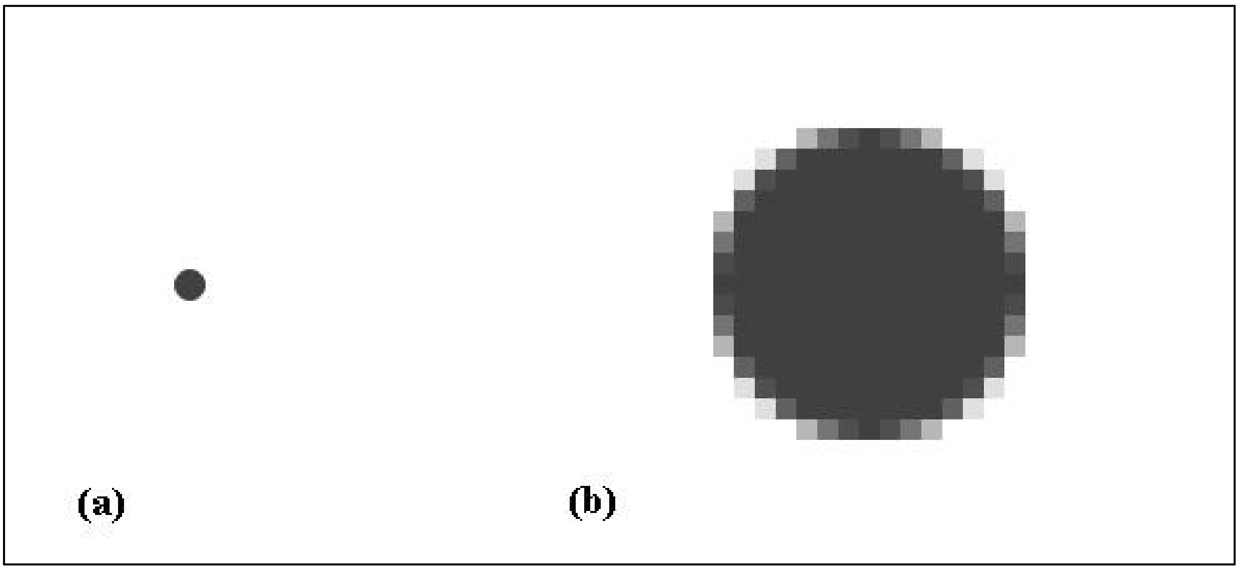


Fig. 1. A simple disc image of diameter 15 pixels. (a) as normally viewed; (b) highly zoomed to show the grey level structure around the perimeter, in which is encoded the sub-pixel trend of the edge.

1. Simple small disc image.

Fig. 1 shows a simple small disc image, both as normally viewed and as highly zoomed. Note here that, when highly zoomed, the edges of the disc exhibit intermediate grey levels between foreground and background level. This is typical of all natural greyscale images, but the subtle edge structure is not normally visible when viewed at normal viewing scale!

If this image is offered to the Canny detector, the resultant output, both as normally viewed and highly zoomed is shown at Fig. 2. This output from the Canny detector can be ***inferred*** to be a simple disc (in the absence of any possible alternative). However, the 'outline' of the disc is exceedingly 'raggy'.

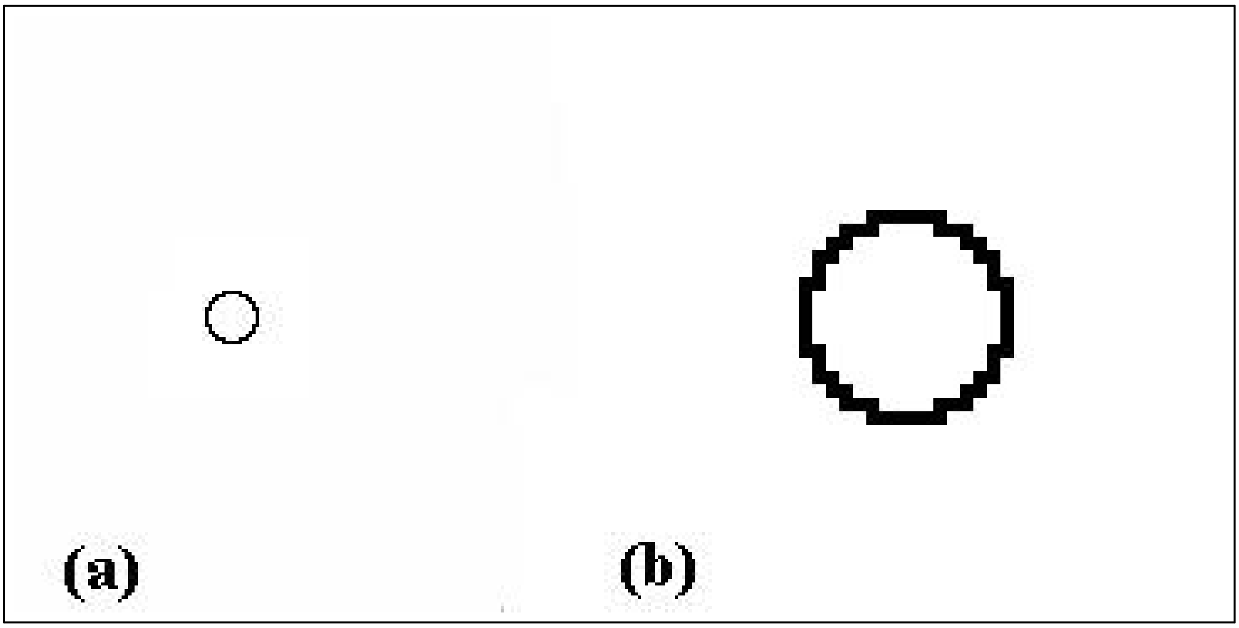
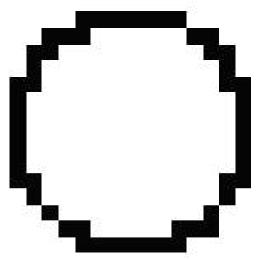


Fig. 2. Output from the MathCAD version of the Canny Edge Detector. (a) direct; (b) highly zoomed to show the 'tesselated' structure.

Figs. 3 & 4 show the resultant output from an equivalent SHV detection process. When the SHV output is represented at the original pixellation scale (Fig. 3), either as normally viewed or highly zoomed, it looks vaguely similar to the Canny output - i.e. again a very 'raggy' outline. However, if the full output data from SHV are plotted on a matrix representing tenths of pixels of the original, with the individual data points being represented as short bars centred on the computed position and of orientation depicting the local computed orientation, the result is as shown in Fig. 4. It should immediately be obvious that the full nature of the disc outline has actually been extracted.



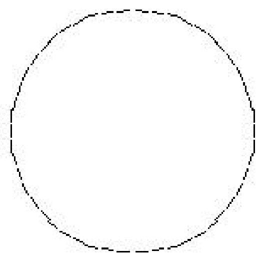


Fig. 3. A highly zoomed representation of the SHV output for the 15 pixel disc image when presented at a single pixel resolution.

Fig. 4. The SHV output plotted at a resolution of 0.1 pixels, showing both the sub-pixel positional data and a coarse representation of the orientation.

2. Small star image.

Next let us consider another small object, but one which has well-defined structure. Fig. 5 shows an image of a small, high contrast star, both at normal viewing scale and highly zoomed. Again note here that, when highly zoomed, the edges of the star are characterised by intermediate grey levels.

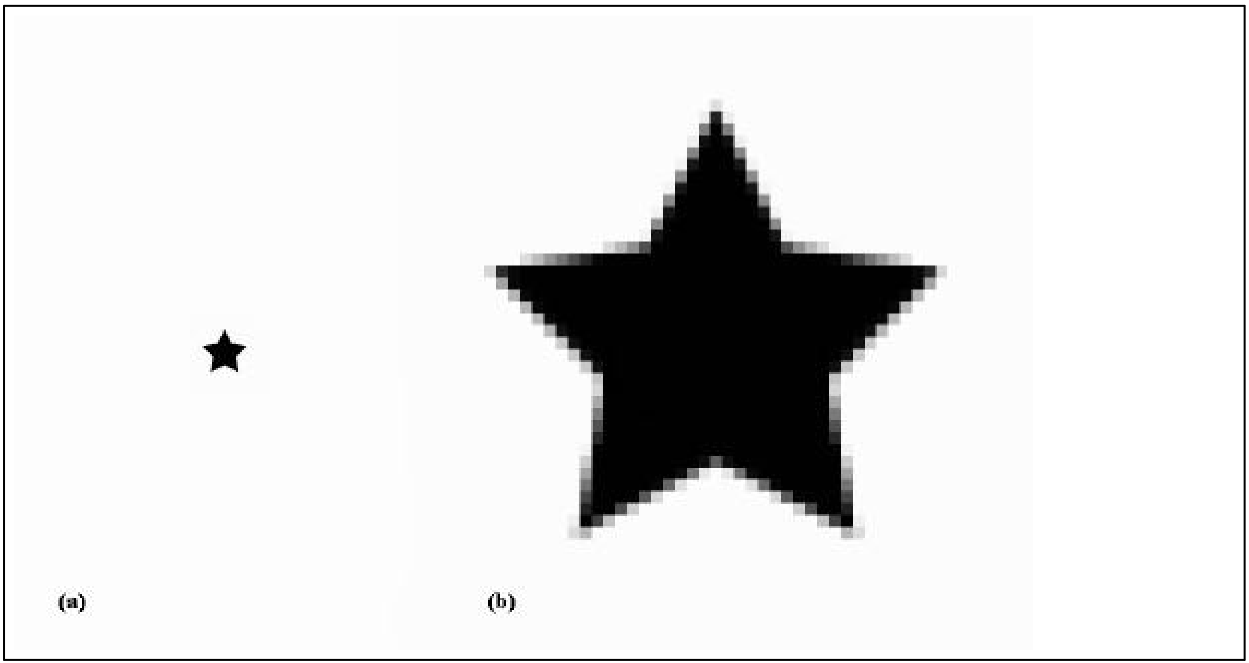


Fig. 5. A small star image. (a) the star as normally viewed; (b) a highly zoomed version showing the grey scale structure around the edge containing the sub-pixel edge data.

Fig. 6 shows the output from edge detection by the Canny operator. Here the output from the Canny detector is again rather raggy, as must be expected when the output is limited to nearest pixel around the perimeter.

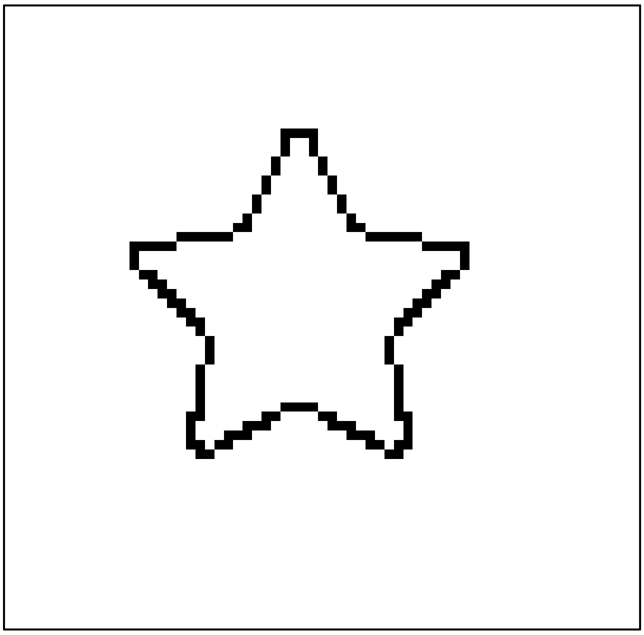


Fig. 6. The star output from the MathCAD version of the Canny Edge Detector.

When processed through SHV, the output at original pixel scale is again similarly raggy to the Canny output (Fig. 7). However, this time, when the output data are plotted on a tenth original pixel scale image (Fig. 8), the form of the star is very clearly evident, with the exception of the points, which take on a rounded form. This rounding of corners & sharp points is an artifactual effect which is inevitable when using any input blur to limit processing noise (as can be shown to be necessary in order to satisfy optimum signal / noise criteria) and is nothing to do with the actual edge detection. In looking at figure 8, it should again be remembered that the rather smooth flow of the edge is without ***progressive smoothing***, but is, rather, a plot of the actual individual point to point outputs from SHV.

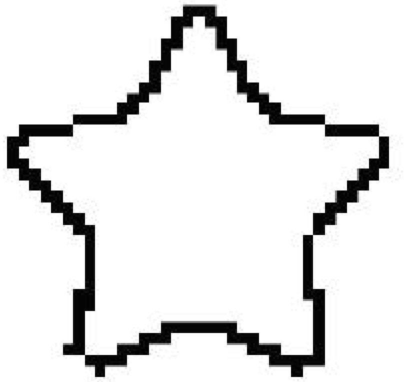


Fig. 7. The output from SHV plotted as nearest pixel at single pixel resolution.

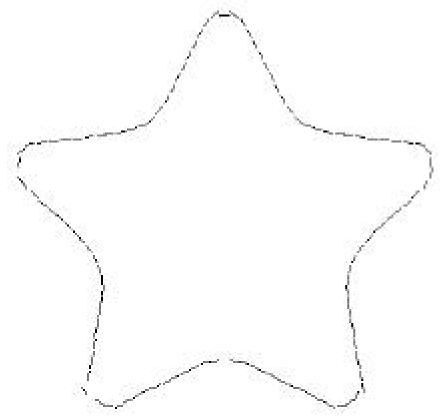


Fig. 8. As figure 7, but plotted at 0.1 pixel resolution.

If the sub-pixel positional data and the local orientation data are further processed through a supplementary software in order to yield an ***ordered*** profile rather than a raster scan output, the ordered profile local orientation data may be further processed to yield local ***orientation difference*** data for each edge point. This data (automatically tabulated during the supplementary process) may then be plotted on a similar plot to that of Fig. 8 as lines orthogonal to the local edge trends, these lines being of length proportional to the local orientation difference. The result of such a further processing is shown at Fig. 9, where it can be seen that the relative curvature trends are very clearly evident.

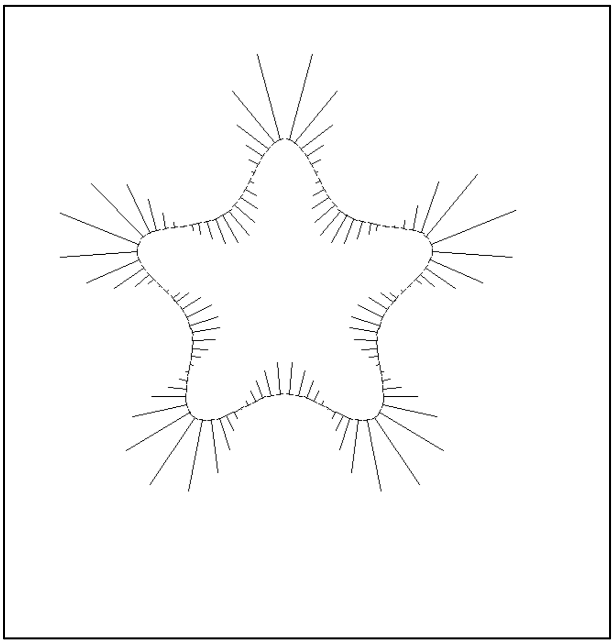


Fig. 9. The star output plotted at 0.1 pixel resolution after supplementary processing to generate local curvature. Relative curvature illustrated as lines orthogonal to local edge.

3. Small dodecagon.

As a further demonstration of the fidelity of edge orientation sensing, a dodecagon of size effectively the same as a disc image of approximately 30 pixels diameter was processed in the same way as the small star. The original image at normal viewing size and zoomed by a factor of 10 are shown as Fig. 10. It should be clearly evident that, although there is visual evidence of the 12 'points' at normal viewing size, it is quite difficult to determine the exact structure from the enlarged image.

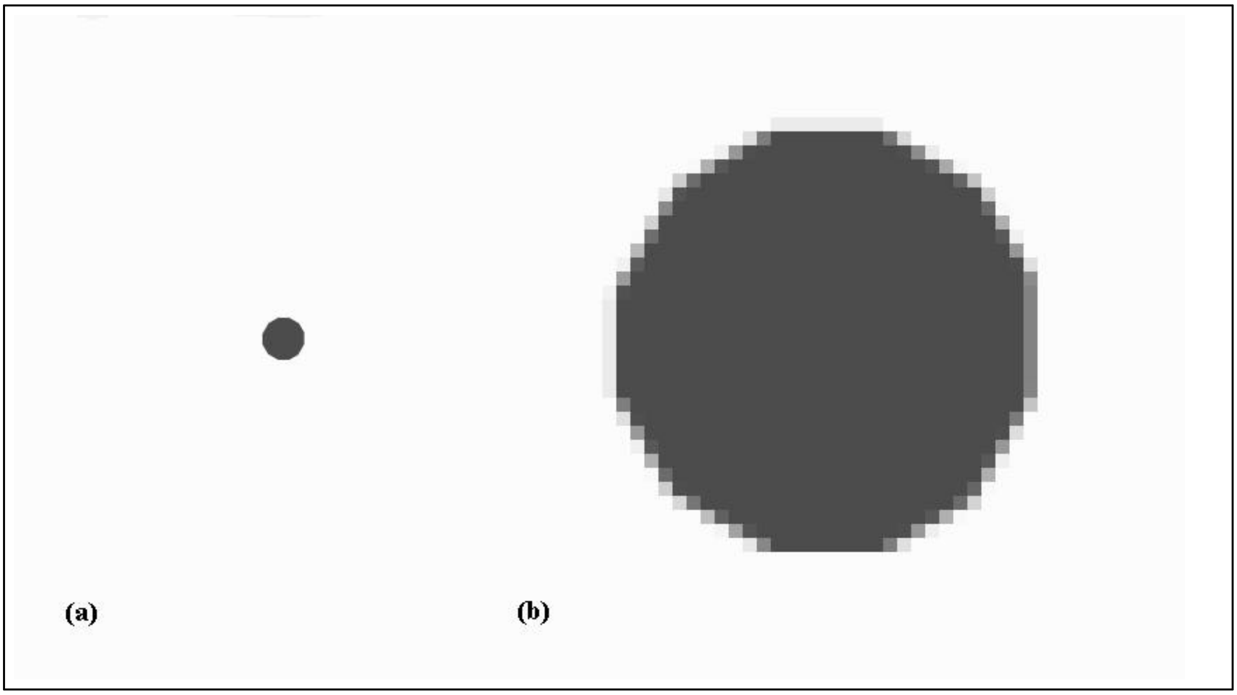


Fig. 10. A dodecagon input image of approximate 'diameter' 30 pixels. (a) normal viewing size; (b) highly zoomed to show the graded grey level structure around the edge.

If the image is processed using the Canny detector with a Gaussian blur of standard deviation 1.3 pixels (as considered to be approximately the same as that employed in human vision), the results are as shown in Fig. 11. These are basically the same as for processing of a 30 pixel diameter circle.

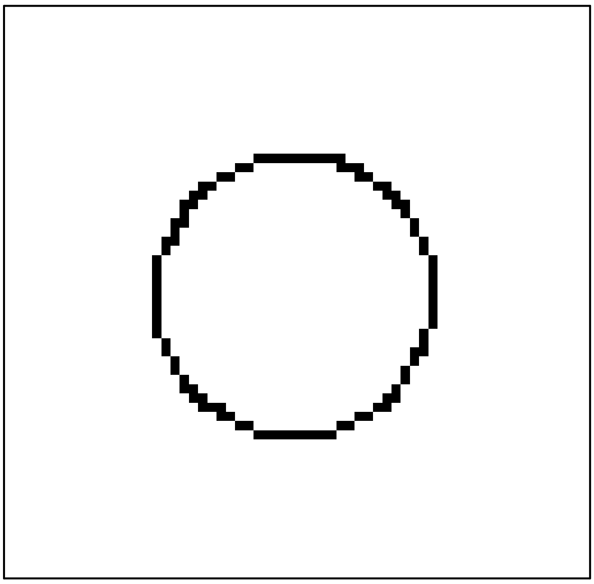


Fig. 11. The dodecagon output from the MathCAD version of the Canny Edge Detector.

If the same image is processed through SHV, then even on a one-tenth pixel plot it is quite difficult to see the 12 points (Fig. 12).

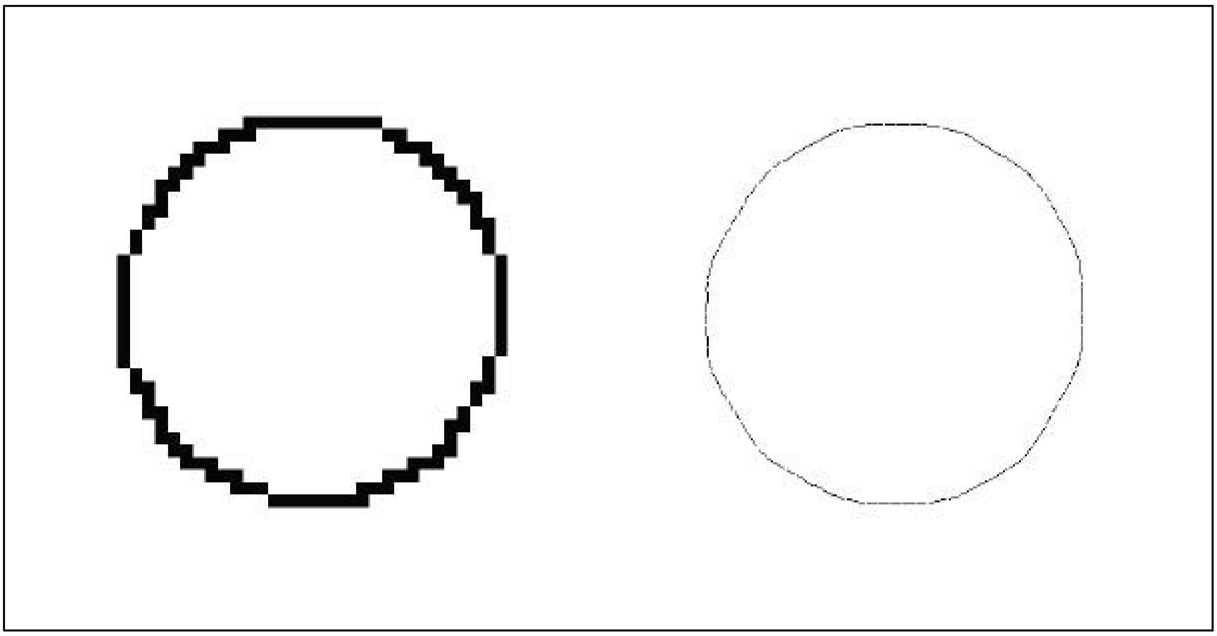


Fig. 12. Single pixel and 0.1 pixel resolution plots of the dodecagon output from SHV.

However, when supplementary profile ordering and orientation difference computation is carried out, the resultant plot showing local orientation differencing not only displays the 12 local peaks where expected, but also exhibits a quite remarkable ***consistency*** of profile form around the entire figure (Fig.13).

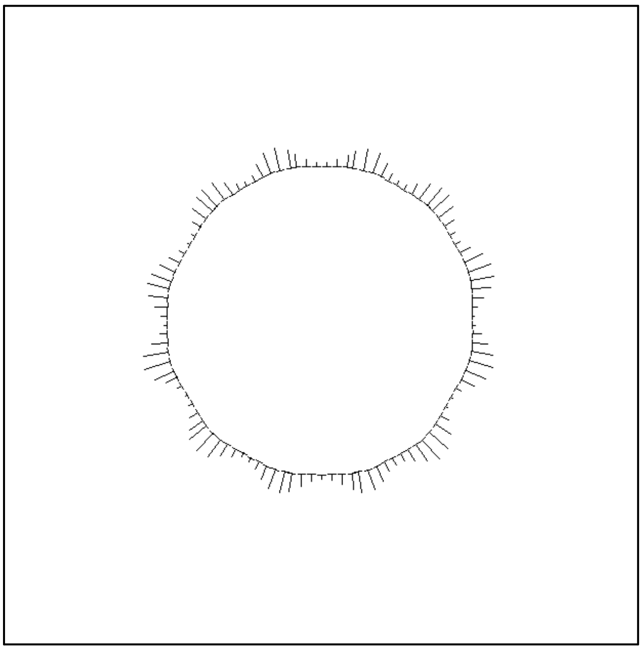


Fig. 13. 0.1 pixel resolution plot of the local curvature ouptut from SHV for the dodecagon.

3. Typical fingerprint (MathCAD image 'fingerp.tif').

In order to compare Canny & SHV processing for an image which is demonstrated in the MathCAD documentation, the MathCAD image 'fingerp.tif' (Fig. 14) was processed by both methods (using essentially the same blur and basic thresholding as demonstrated in the MathCAD documentation).

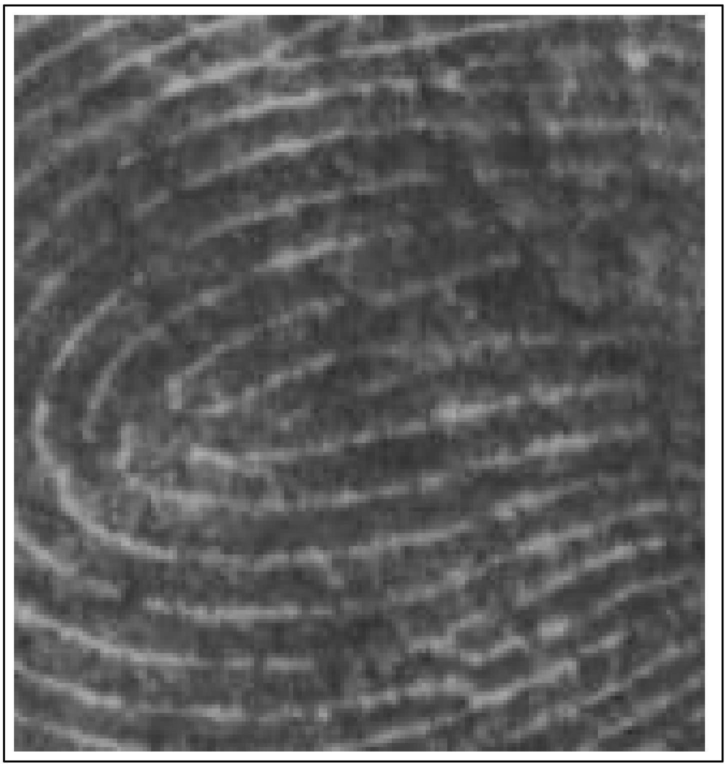


Fig. 14. The input image for a fingerprint as provided in MathCAD.

The output from the Canny detector was as shown in Fig. 15. This is essentially the same as that which can be found in the MathCAD supplementary document 'Canny.mcd'.

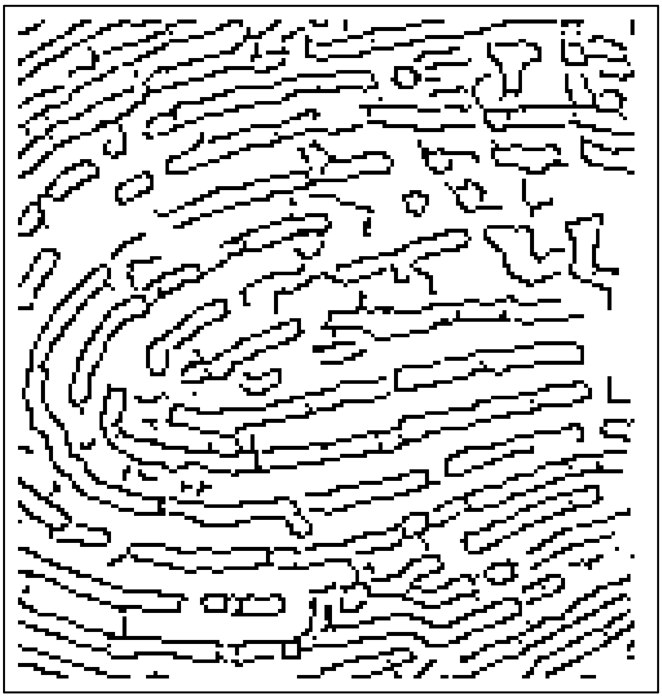
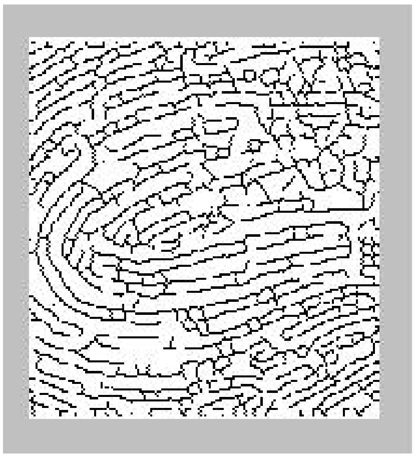


Fig. 15. The output edge map as recomputed using the Canny Edge Detector in MathCAD.

When the same blur and roughly the same thresholding are applied for a SHV process and the output is presented as a simple ***line art*** output, the results are very similar (Fig. 16). If, however, the SHV output is plotted at normal scale as a ***greyscale*** output (Fig. 17), then (it is claimed) a considerably greater amount of 'intelligence' emerges, related to the local ***edge contrast*** mapping. Such extra 'intelligence' may or may not be of additional value, dependent on precisely what the output data are being used for.



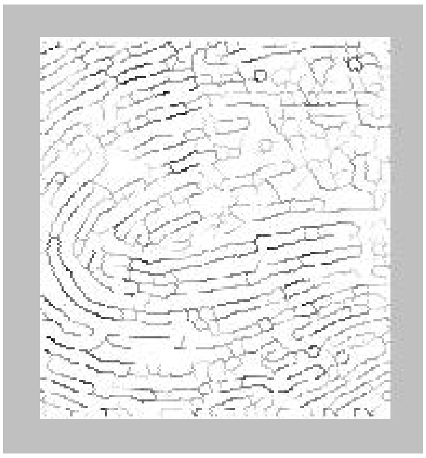


Fig.16. Line art output from Fig. 14 using SHV.

Fig. 17. As fig. 16, but showing local edge contrast as shades of grey.

It almost goes without saying that a one-tenth pixel orientation plot, as for the simple images, yields a considerable amount of further information (which is potentially available for ***visual*** fingerprint inspection, but is most definitely ***not*** available for conventional automatic inspection! To illustrate this for the ***whole*** of the input 'fingerp.tif' image would lose visual data due to the limitations of reproduction on a standard A4 print format, so instead I have chosen to illustrate the output from SHV for a portion of the original just to the left of centre. This is shown in Fig. 18 as an enlargement of a single pixel resolution representation of the SHV output and as Fig. 19 plotted at an 0.1 pixel resolution.

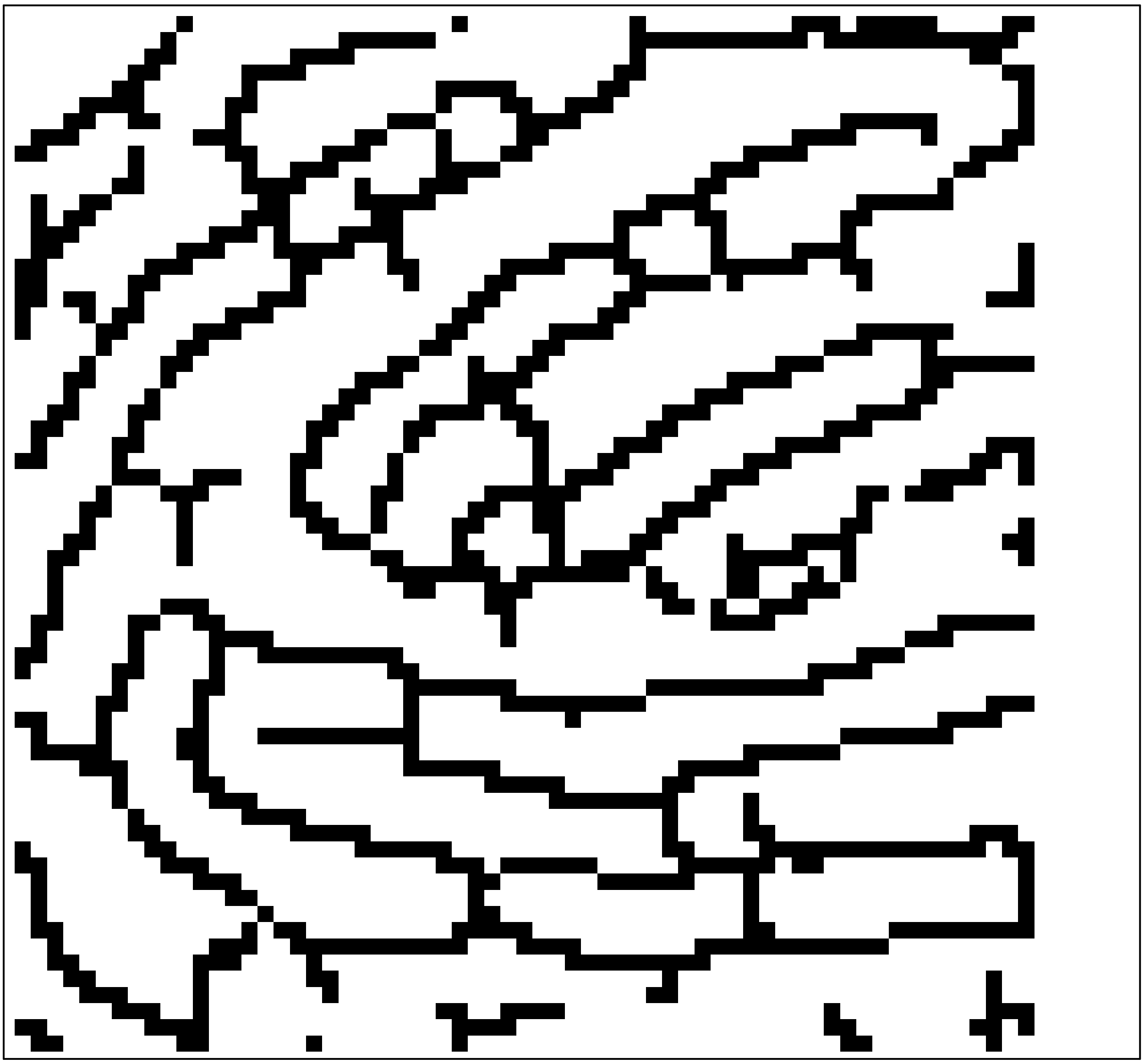


Fig. 18. The SHV output for a 64 x 64 pixel portion of Fig. 14 cropped from left of centre plotted at single pixel resolution.

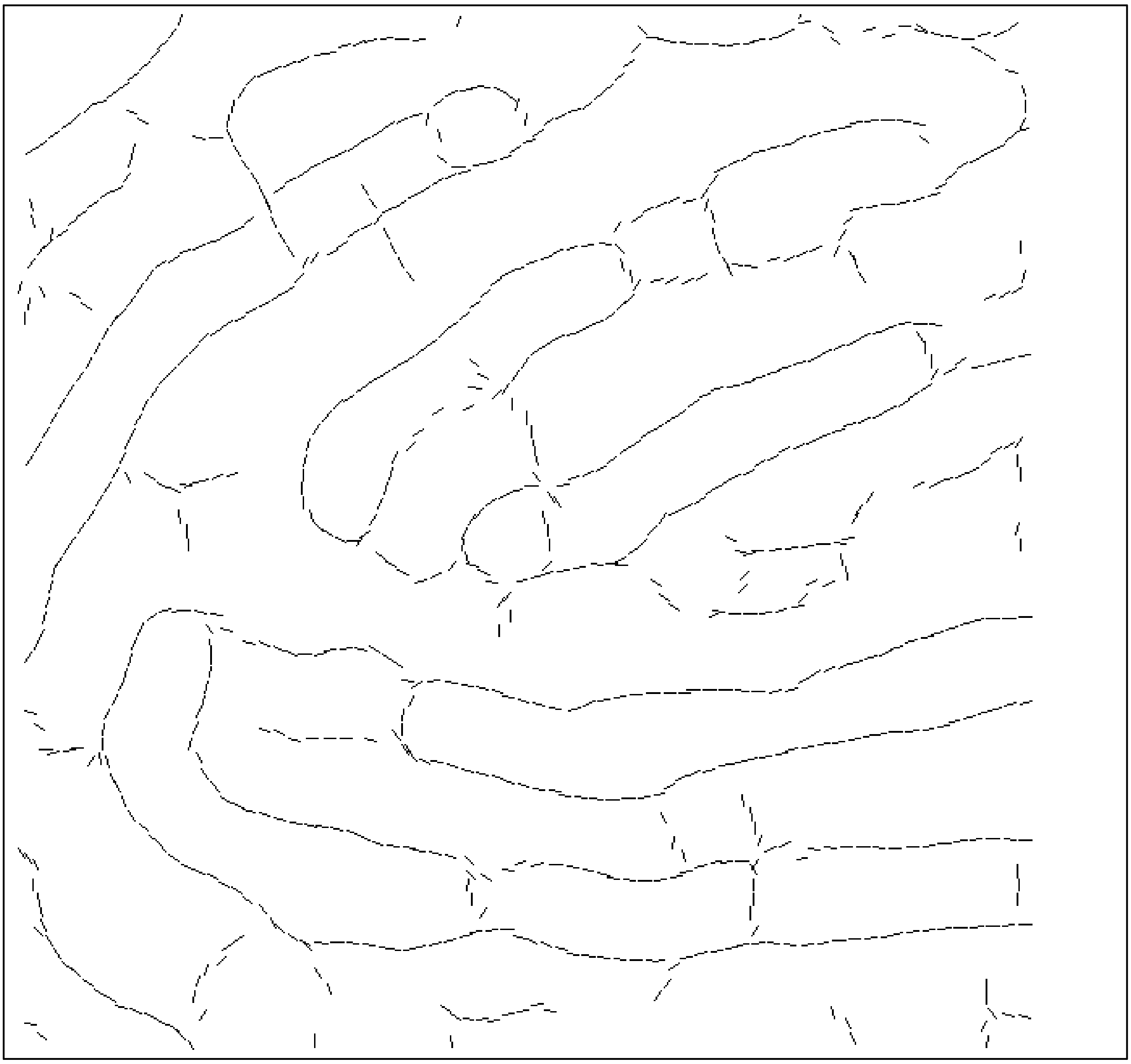


Fig. 19. The same portion of the SHV output from Fig. 14 when plotted at 0.1 pixel resolution.

4. Portrait.

When it comes to more complex images, one must argue that each & every image will have its own subtle characteristics. It therefore becomes exceedingly difficult to carry out any rigorous comparisons. Nevertheless, as an attempt to illustrate the combined increase of 'intelligence' which can be extracted from typical complex images using SHV, a typical portrait (Fig. 20) which has been used in years gone by for demonstration of capabilities of forerunners of SHV has been reprocessed using closely similar blur & thresholding by both the Canny detector and SHV.

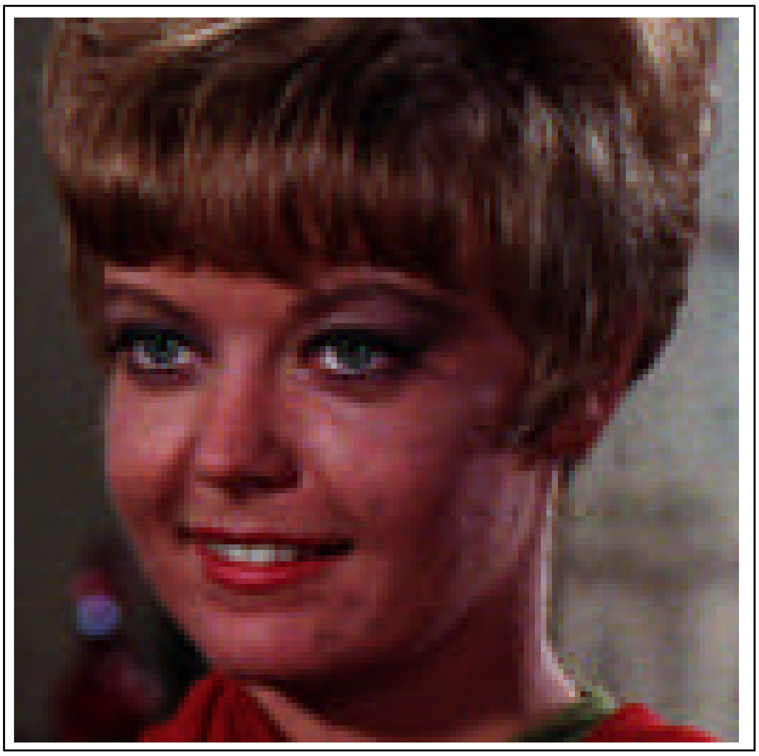


Fig. 20. A typical colour portrait with an input resolution of 128 x 128 pixels.

Since this image is only of 128 x 128 pixels, one might assume that it cannot really be expected to yield a particularly fine detail of edge structure. But when one ***views*** the image one has a distinct impression of a very ***rich*** structure contained therein, despite the very sparse pixellation. What then can SHV do for this type of complex image?

The image was first processed ***directly*** using both the Canny Edge Detector and SHV (in both cases using settings considered to be both typical and comparable). When the outputs are presented at the original pixel scale, the richness of edge fragments in the single pixel resolution representations is similar (Fig. 21).

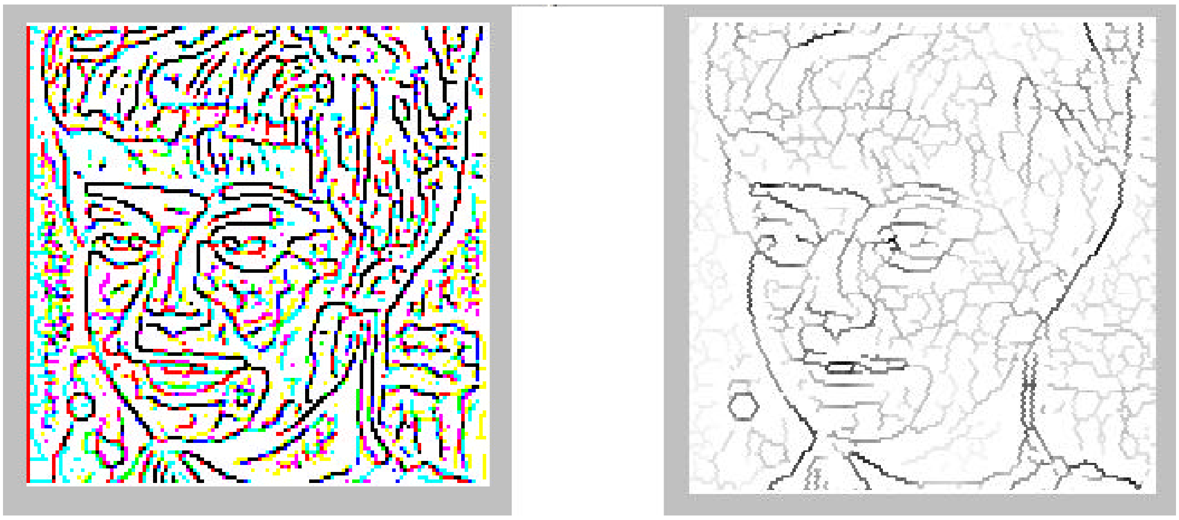


Fig. 21. Direct outputs from processing Fig. 20. (a) Canny; (b) SHV (single pixel resolution).

There are, however, significant differences in form of presentation - coloured line fragments in the Canny output indicating local zones where colour difference is more prominent than brightness difference, whereas grey level representations of local edge contrast are a main factor for SHV. In both outputs the inference that only minimal detail can be extracted from such a low resolution input image tends to be confirmed. However, in the case of the SHV output, since a region segmentation has automatically been carried out during the edge detection process and the full edge & region statistical data have been stored in the standard output files, it is also possible to produce a reconstruction of the original image. Such a reconstruction is shown as Fig. 22 where, although subtleties of detail which can be visually discerned in the original have been lost, nevertheless the reconstruction does recover a fair representation of the original.

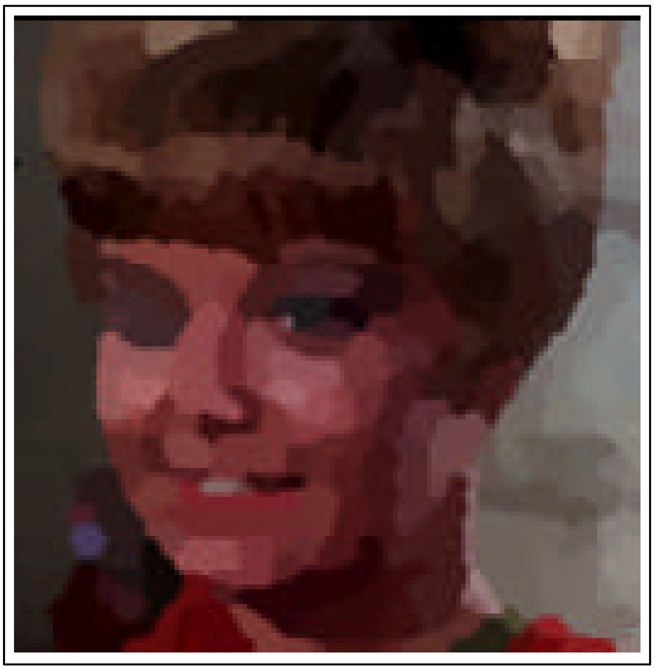
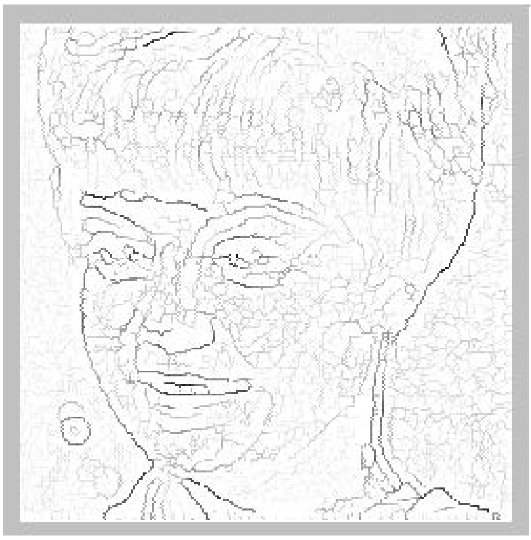


Fig. 22. Reconstruction of the image of Fig. 20 from the edge-based region segmentation and stored data obtained in support of the SHV output of Fig. 21b.

It was found, as part of our studies into human visual threshold performance, that, for viewing of discretely sampled images of complex scenes, optimum visual performance was achieved when the individual pixels of such images subtended between 2 & 4 retinal receptor spacings (see Ref. 5). A facility for 2D cubic interpolation of complex sampled images was therefore developed in order to interpolate semi-intelligently by X2 linear in both X & Y dimensions (see Ref. 1, Chapter 8.3), this subsequently becoming an integral part of SHV (for use prior to offering the image to the main edge detection processes, if considered necessary).

If this 2D cubic interpolation facility is first used on the image of Fig. 20, this being then followed by normal SHV edge detection processes, it is found that a considerably greater amount of subtle detail becomes available (Fig. 23)! If a reconstruction is then carried out from the region statistical data now available from the interpolated input image, this reconstruction is found to bear a considerably better resemblance to the original (Fig. 24)!



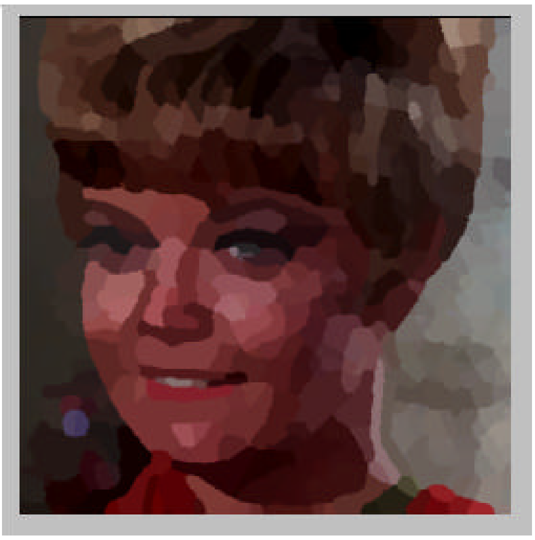


Fig. 23. SHV single pixel edge output after cubic interpolation of the input image by X2.

Fig. 24. Reconstruction from edge-based region segmentation associated with Fig. 23.

The improvements of information extraction / recovery need not stop here. If, instead of a X2 linear 2D cubic interpolation, one applies a X4 linear 2D cubic interpolation (also directly available as an integral part of the SHV process), again followed by standard SHV processing and region reconstruction, then even more impressive information extraction / recovery is possible (Figs. 25 & 26).

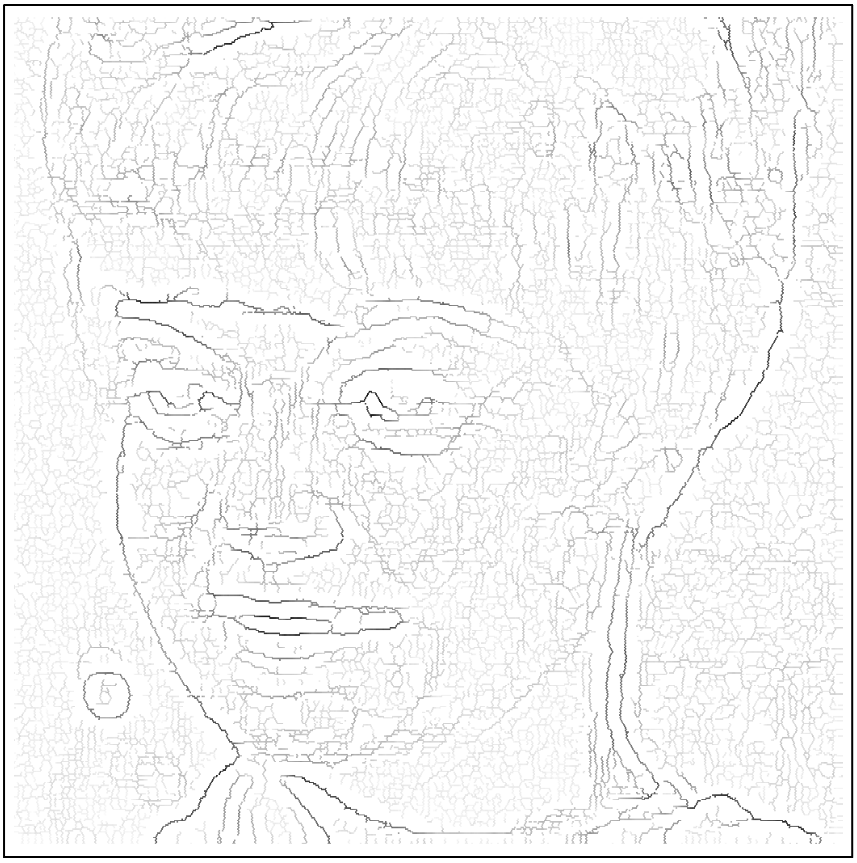


Fig.25. The output from SHV at single pixel resolution after initial cubic expansion of Fig. 20 by X4 linear.

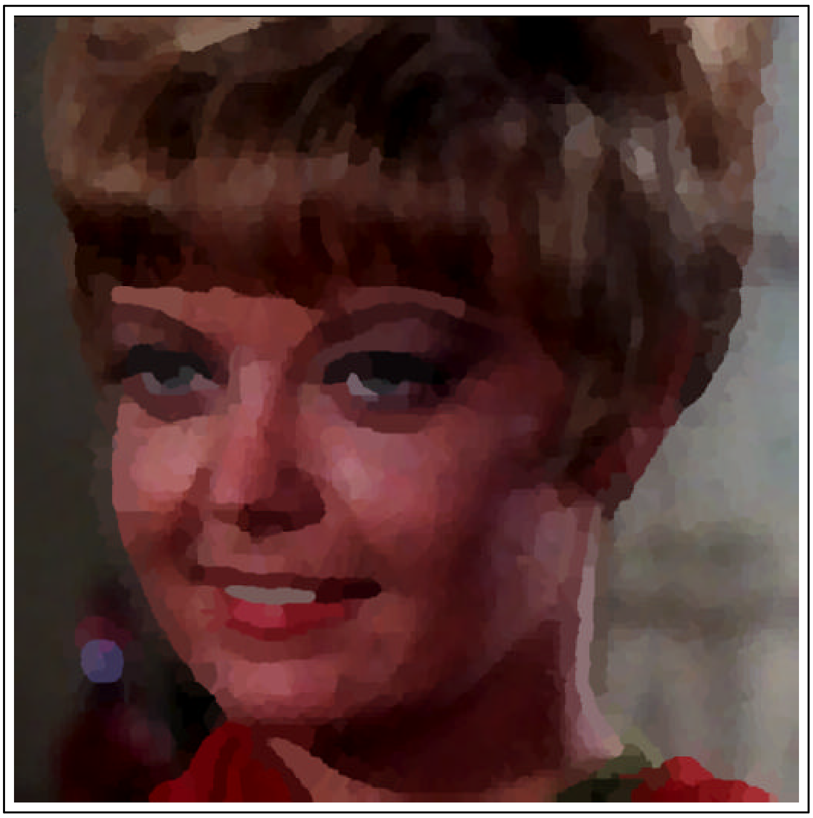


Fig.26. The reconstruction derived from the egde-based region segmentation data associated with Fig. 25.

Fig. 26 is considered to be a rather satisfying reconstruction of the original, particularly bearing in mind that the said original still only contained 128 x 128 picture points! With any such reconstructions it should also be remembered that one has available the ***total*** data related to each and every picture point on the original image (as segmented by the processes) and that, although the sub-pixel local edge data are not specifically used in the reconstruction, the full region segmentation as derived from the sub-pixel local edge data ***are*** all necessary in order to implement the reconstruction.

5. Simple shape motion or stereo.

Although it is obviously impossible to provide any sensible ***comparison*** between a current MathCAD process and SHV for local motion or stereo (since no direct process is currently available within MathCAD for such processing), it is considered of use to provide just one simple illustration of the power of the SHV paired-frame processing for very small local edge displacements. For simplicity of description and visualisation, I have chosen to demonstrate the sensing of local orthogonal components of displacement, point by point, around a small disc image which has been subjected to approximately 0.7 pixels of diagonal movement between the two images. The pair of input images are shown at Fig. 27 where, even when highly zoomed, the differences around the edges are somewhat difficult to perceive.

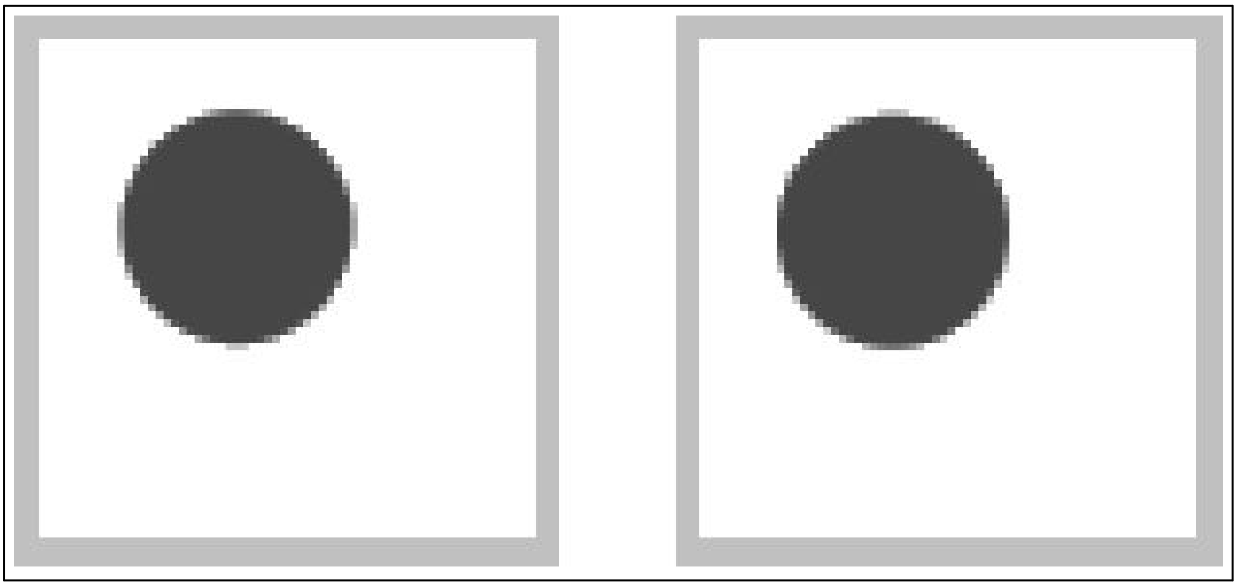


Fig. 27. The input pair of 30 pixel diameter disc images, where the between frame motion is 0.5 pixels in each of X & Y dimensions.

The output of a paired-frame process is then represented as a one-tenth pixel 'super-resolution' edge map, with the magnitude of local orthogonal edge displacement being shown as short radial lines at each sub-pixel edge point (Fig. 28).

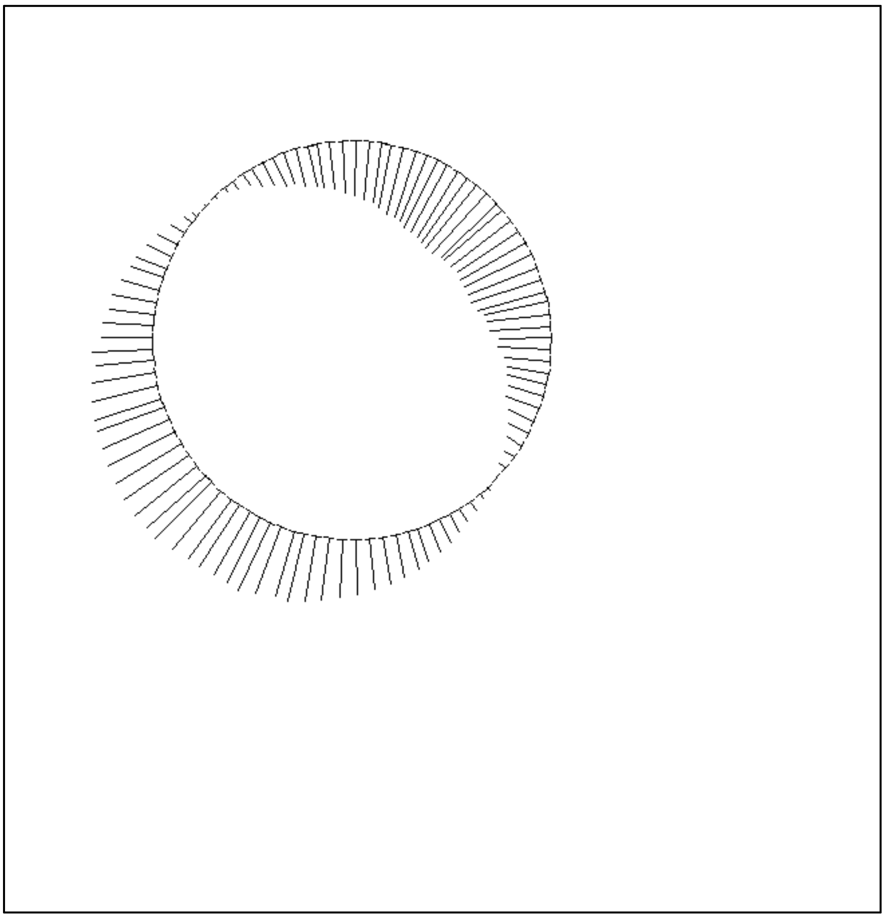


Fig.28. An 0.1 pixel resolution plot of the edge map obtained by SHV from the pair of images shown in Fig. 27, with the local motion orthogonal to the edge overlayed.

It should be easy to appreciate that, if a disc image is moved diagonally, then two opposite diagonal portions of the edge of the disc will show orthogonal movement equivalent to the imposed movement, whilst the two quadrature portions of the edge will show ***no*** orthogonal movement (i.e. they will be moving ***tangentially***). Other parts of the edge will have intermediate orthogonal movement according to a sine law. It will be seen that the output map well illustrates this sinusoidal trend, where all magnitudes derived apart from the two extreme diagonal portions are of smaller magnitude than these extremes. In looking at this output it must be remembered that the extremes themselves are representations of only approximately 0.7 pixels motion!

A much fuller discussion on the whole of this concept of sensing of sub-pixel components of movement, together with an extended theoretical treatise, are to be found in Chapters 6, 9 & 10 of Ref. 1.

Conclusions.

It is hoped that the foregoing observations & discussions have provided a fair comparison of what seem to be the currently available commercial facilities for edge detection & region segmentation found in MathCAD, as compared with much higher fidelity capabilities which can be achieved by drawing heavily on knowledge of the capabilities of human vision.

It is felt that, with possibly minimal effort, it might be possible to incorporate at least some of the extended capabilities now available within SHV into the Image Processing package of MathCAD. This would have the dual benefit of making some of these extended facilities available to a wide research & development population as a part of what is acknowledged to be a widely accepted facility in MathCAD, whilst at the same time providing a form of outlet for what are recognised as (in themselves) a rather specialised set of functions which are difficult to market in isolation.

References.

[1] Overington I. (1992), ‘Computer Vision - a unified, biologically-inspired approach’, Elsevier Science Publishers B.V..
[2] Canny J.F. (1983), 'Finding edges and lines in images', MIT Tech. Report No. 720, Cambridge, Mass., USA.
[3] Overington I. (1976), 'Vision and Acquisition', Pentech Press, London.
[4]. Berry R.N. (1948), 'Quantitative relationship among vernier, real depth and stereoscopic depth acuities', J. Exp. Psychol., *38*, 708.
[5]. Overington I. (1982), 'Towards a complete model of photopic visual threshold performance', Opt. Eng., *21*, 002.