OPTIMIZATION OF THE QUALITY OF COMPUTER-AIDED TILE DESIGN BY THE DEVELOPMENT AND IMPLEMENTATION OF NON-LINEAR ENHANCEMENT ALGORITHMS

by F. Ferrer (*). M. J. Bartual(*), V. Candela (**), A. Marquina(**)

(*)VERNÍS, S.A.
(**) Department of Applied Mathematics. Valencia University. Spain.

ABSTRACT

A present facet in improving the quality of the design process for ceramic floor and wall tile consists of optimizing computer image-processing tools, such as computer programmes (filters), since ceramic decoration is nowadays mainly carried out with the aid of computer-aided design systems.

The first part of the study provides an introduction to the image digitalization system, the basic concepts of image processing and convolution filters.

The development of a non-linear algorithm is presented, which allows performing the convolution operation for image enhancement, without further noise being introduced. The choice of the model is justified, as well as the reasoning which led to its derivation.

1.- INTRODUCTION TO AN IMAGE PROCESSING SYSTEM

Design plays an increasingly important part in developing ceramic products, as the success of a product is to a large extent based on its technical and aesthetic characteristics.

Since roughly ten years ago, computer systems have become an essential part of design departments in the ceramic sector, due to the industry's need to create models for aesthetically innovative ranges at great speed.

These computer programmes can be put into two categories: the ones that work on vectorial designs, in which the images are composed of lines and curves and the ones that work on *bit* maps, in which the image is made up of a fine grid of *pixels*. This present work was carried out in the field of the latter kind of applications.

In computer-aided design systems, image processing takes place by means of four processes: input, visual representation, manipulation and output.

1.1.- Image capture and digitalization:

In these systems, the first stage is the capture and digitalization of the image by input devices (scanners, video cameras, CD photo systems, digital cameras), which transform the information contained in the image (luminous intensity, colour) into electrical signals, which will thereafter be processed [1].

A digitalizer converts an image into a suitable numerical code for a digital computer. The most commonly used digitalizer in the ceramic sector is the scanner.

A scanner is basically made up of an optical system which explores the photograph and examines the luminosity of each dot of the image.

This exploring process is done by photoelectric cells, yielding analog-type data, which by means of an analog-digital converter, are transformed into digital information (a binary number), which can be processed and stored by the computer.

Depending on the quality of the photoelectric cells, there are several types of scanners. The ones that handle shades of grey, assign a value 0 to black and 255 to white, through the remaining values that correspond to different shades of grey. A maximum of 256 is chosen because the eye cannot easily discern more shades of grey. These readers are the least commonly used kind, since one of the most important items for ceramic design, namely colour, cannot be handled.

In the case of colour scanners, the system separates the colour components into the basic values which form colours as perceived in human vision: red, green and blue. Each one of these three primary colours has a range of values from 0 to 255. When the 256 possible values of red are combined with the 256 possible values of green and the 256 values of blue, the total number of possible colours lies around 16.7 million (256 x 256 x 256) [2].

The scanner's resolution depends on the accuracy of its photoelectronic system, and is measured by the number of *pixels* per inch. The greater the number of pixels, the greater the definition of the image.

The main problem faced by colour scanners is the great amount of information that can be contained by an image of this kind. On scanning at 300 dots per inch, for example, a 20 x 20 cm tile will contain 5.6 million *pixels*, for each of which 3 *bytes* of information are required. The resulting image thus takes up 16.7 MB of storage capacity.

1.2.- Visual representation of images in the computer:

After the scanner has thoroughly explored the image, it is conveyed to the computer, where it can be manipulated as the designer wishes.

The way the images obtained by the scanner are visualised on the screen depends on the monitor and the type of graphics card installed. Normally the monitor has much less resolution than the scanner. To work with the images obtained by the scanner, the image control *software* is able to convert the image into a format that is visible on screen. This is done by grouping several *pixels* together into just one, and calculating an intermediate value.

Before starting to retouch images, the monitor has to be calibrated, as otherwise the final image could turn out to be very unlike the one shown on the screen.

1.3.- Image processing in the computer:

The image input to the computer contains all kinds of different information: shapes, colours, textures, brightness, contrast, etc. With computer applications one or several types of information can be extracted, ignoring the others. All these operations together form what is known as processing.

1.3.1.- Digital image filtering techniques. Convolution.

The simplest operations of all those used for image treatment are the *pixel* or dot operations. This type of function applies a transformation to the luminance of each dot in an image, a function which does not depend on any data except for the prior value of the luminance involved.

An example of such operations is *complementation,* which consists of replacing the luminance value in each dot for that of its complementary value. The effect obtained is the digital negative of the image. (Figure 1.)

Area operations mean that more than the function and the luminance of the dot have to be known, as they modify the appearance of each *pixel* in terms of the luminance value taken by the surrounding dots.

There are operations that can be implemented directly, that is, without any need to convolute the image.

One of these, for example, is the *median filter*. In this operation, the surrounding area of each *pixel* is taken, and the central *pixel* is replaced by the median value of the surrounding area. The effect obtained by applying the median filter is a softening of the image, getting rid of isolated dots, which are generally attributable to noise, without blurring the ones at the edges too much. (Figure 2.)

Convolution calculates a new light intensity for a *pixel* from the light intensities of the neighbouring *pixels*. Each neighbouring *pixel* contributes in a percentage of its value to the calculation of the new *pixel*. The concept for defining this process is called "weighting", and is done by the use of the convolution nucleus. Each element of the convolution nucleus is called a *coefficient of convolution* [3]. The following diagram schematically depicts a filter based on a convolution nucleus and the way this should be applied to the convolution of an image.

Schematic diagram of the convolution filtering process

The best way to explain convolution is to take a simple example. In the preceding figure we can see that the *nucleus* of the convolution filter is a 3 x 3 matrix which contains *fixed numbers*, which are the filter constants:

$$FILTER = \begin{Bmatrix} f_{-1,-1} & f_{0,-1} & f_{1,-1} \\ f_{-1,0} & f_{0,0} & f_{1,0} \\ f_{-1,1} & f_{0,1} & f_{1,1} \end{Bmatrix} = [f]$$

The filter calculates a new value for the light intensity of each *pixel*, by means of the information from the neighbouring *pixels*:

If we consider the matrix of light intensities of the source image:

$$[S] = \begin{pmatrix} S_{-1,-1} & S_{0,-1} & S_{1,-1} \\ S_{-1,0} & S_{0,0} & S_{1,0} \\ S_{-1,1} & S_{0,1} & S_{1,1} \end{pmatrix}$$

new value in (0,0)= $(f_{-1,1} * S_{-1,-1}) + (f_{0,-1} * S_{0,-1}) + (f_{1,-1} * S_{1,-1}) + (f_{-1,0} * S_{-1,0}) + (f_{0,0} * S_{0,0}) + (f_{1,0} * S_{1,0}) + (f_{-1,1} * S_{-1,1}) + (f_{0,1} * S_{0,1}) + (f_{1,1} * S_{1,1})$

If we consider the matrix:

$$[f] = \begin{pmatrix} 1 & 9 & 1 & 9 & 1 & 9 \\ 1 & 9 & 1 & 9 & 1 & 9 \\ 1 & 9 & 1 & 9 & 1 & 9 \end{pmatrix}$$

this is the so-called low-pass filter, which consists of calculating the arithmetic mean of all the light intensities of the neighbouring *pixels*.

A high-pass filter would be:

$$[f] = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Convolution filters are linear in the sense that the reference nucleus [f] is a matrix of constants and does not change, whatever part of the image this is applied to.

When the reference matrix changes its content, depending on the light intensities to which it is applied, then the filter is called non-linear.

The size of the nucleus matrix can be greater than 3 x 3, which could be considered the lowest efficient value.

The study and effect of images filters can be further found in monographs such as [4].

1.3.2.- Detection filters: gradients and Laplacian

One problem of great interest in image processing is the enhancement of profiles and their detection. There are two fundamental concepts underlying profile enhancement and detection, and these are:

1) The *Gradient* of a function g(x,y) of two variables is:

$$\text{grad}(g) = \sqrt{\left(\frac{dg}{dx}\right)^2 + \left(\frac{dg}{dy}\right)^2}$$

2) The *Laplacian* of g

$$\mathcal{L}(g) = \frac{d^2 g}{dx^2} + \frac{d^2 g}{dy^2}$$

One way to implement these filters can be found according to the convolution schematic.

For example, to implement the *directional gradient* of an image, this can be done by means of one of the following reference filters:

$$[f] = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad [f] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad [f] = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Horizontal Horizontal/Vertical Vertical

The effect of the gradient filter is a binary digital image that provides the borders of the image (see Figure 3).

A discrete way to calculate the *Laplacian* would be from

$$\frac{d^2 g}{dx^2} = g(x+1)-2g(x)+g(x-1)$$

$$\frac{d^2 g}{dy^2} = g(y+1)-2g(y)+g(y-1)$$

$$\mathcal{L}(g) = -4g(x,y)+g(x+1,y)+g(x-1,y)+g(x,y+1)+g(x,y-1)$$

This formula can be rewritten with the convolution filter scheme, which would provide the following reference matrix:

$$[f]^{\sim} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

The effect of the Laplacian filter can be seen in Figure 4.

To obtain *enhancement* from the Laplacian, it is enough to consider the reference matrix

$$[f]^{\sim} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

Horizontal enhancement is obtained by

$$[f] = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

and *vertical enhancement* by

$$[f] = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

1.3.3.- The Fourier transform and the Gibbs phenomenon

The fundamental hypothesis of Fourier analysis is as follows:
"Any signal can be broken down into an infinite number of sinusoidal components defined by their frequency, amplitude and phase".

In practice, this breakdown is done in a finite number of components, which entails some advantages and drawbacks:

Under this hypothesis, the Fourier transform is an operation that allows us to work in a simpler way in a frequency space (domain).

The Fourier transform gives us the so-called "spectrum" of the signal, which includes the separation of the frequencies, and for each of these we have the information on amplitude and phase.

We can revert the process and reconstruct the original signal from the frequencies, their amplitudes and phases. This process is done by the so-called inverse Fourier transform.

Since our representation of the signals is discrete, (by means of a finite number of points), the discrete Fourier transform considers a finite number of frequencies, and the highest of these will determine a type of distortion known as "aliasing", which confuses information given in the same phase.

The effect of the filter for focusing edges based on the Fourier transform is shown in Figure 5.

The fundamental applications of the Fourier transform are as follows:

1) To analyze a signal for analog transmission. The Fourier transform is calculated and from the signal a finite number of frequencies are transmitted along with their amplitude, to be subsequently reconstructed.

2) This means that the operation with convolutions can become very straightforward, as it transforms this spatial operation into a simple multiplication in the frequency domain.

3) The calculation of the Fourier transform is painstaking as this requires a number proportional to N^2 (multiplications), where N is a number of several thousands. Still, this calculation can be speeded up to obtain a number proportional to $N \log_2 N$ multiplications by means of an intelligent algorithm called the FFT - Fast Fourier Transform.

Disadvantages of the FT:

1.- Since the signals contain jumps, the Fourier analysis produces the unpleasant Gibbs phenomenon (see the enlargement of Figure 5) which consists of introducing oscillations and noise around these discontinuities, when one proceeds to reconstruct the signal, from a finite number of frequencies.

2.- The Fourier transform separates the frequencies, but is not local in amplitude, which implies that every error made in every frequency is perceived in all the other frequencies.

2.- THE AIM OF THIS WORK:

The reproduction of the original images can deteriorate for different reasons, but the effects produced are mainly of two types. The first of these is the introduction of the so-called "white noise", which generally gets in through an error in the data translated by the scanner either by numerical mistakes in the coding or decoding of the data. White noise is well-known by television viewers, as the "snow" that forms on the screen. Just as important is the blurred effect, loss of contrast or haze, which is the other commonly found type of image deterioration. In this work we will deal with the second of these phenomena.

The study shows how a new image-processing tool can be constructed, (one that is not offered in software packages for image processing), which is useful for improving the quality of ceramic design. This consists of a filter which gets rid of blurring and recovers profiles, without introducing the snow effect or other types of noise.

Blurring does not generally arise in computer problems, as occurs with noise, but in "physical" ones. There can be a wide range of difficulties, from a possible lack of focusing of the image at the start, to the loss of resolution in the sensors, going through any kind of disturbance of the medium (such as air ionization). In spite of all the care taken with the apparatus, the number of images that deteriorate in any process does not cease to be surprising. It is furthermore well known, that many filters used in software packages for image processing also produce blurring. A simple enlargement or filters that remove noise, for example, often result in a loss of contrast or profiles.

It is indeed true that most software packages for image processing include enhancement programmes, but, as set out below, these do not solve the problem that arises. On one hand it can be said that noise and blurring are opposites in the sense that one arises when an attempt is made to get rid of the other. But the filters found on the market are of a linear nature, which we have already seen means producing the Gibbs phenomenon. Nevertheless the main disadvantage is the difficulty of defining the profiles, which are lost as the image blurs and cannot be recovered in the enhancement process, because they are not detected. The resulting image enhancement depends to a large extent on the profile detector.

It is thus clear that the problem of enhancement is far from solved. The importance of overcoming this drawback to obtain minimally satisfying results forces one to look into new approaches for solutions.

3.- THE MATHEMATIC MODEL:

In optical engineering, it is well known that blurring can be mathematically modelled by means of parabolic differential equations, of which the best-known is the heat equation:

$$\boxed{\begin{array}{l} u_t = a u_{xx} \\ u(x,0) = u_0(x) \end{array}} \qquad \boxed{a>0 \quad t>0}$$

where $u_0(x)$ would in this case be the function that indicates the value of the light intensity of the source image in the *pixel* x, and u_t and u_{xx} respectively represent the first derivative in respect of time and the second spatial derivative of the solution function.

The parameter $a>0$ is related to the rate at which contrast is lost and blurring increases.

After a certain time, t_0 , the solution of the equation, $u(x,t_0)$ represents the light intensities of the image that is obtained.

A first way to tackle the enhancement problem could consist of inverting the previous problem, that is:

Given $u(x,t_0)$, (the image that we have), find $u_0(x) = u(x,0)$, by verifying $u_t = a\, u_{xx}$ for any $0<t<t_0$.

Even without knowing t0, it might be possible to solve the equation "backwards" (reverse time) and stop when a sufficiently highlighted image is obtained.

The matter is unfortunately not quite as simple as that, as the heat equation is not generally reversible (that is, its inverse is an "improperly set" mathematical problem: it is unsteady in the sense that a small error in the data can give rise to a large error in the solution, and there is not only one solution). To illustrate this a physical example could be used: if one has a piece of paper and this is burned, (knowing all the conditions) it can be determined what it will be like after a certain time. On the other hand, if we are given a burnt piece of paper it is impossible to know what that paper was originally like. (There are an infinite number of shapes and sizes which could produce the same burnt sheet). The process is physically irreversible.

The physical example is also useful for explaining why the heat equation is used as a model: on very hot days outdoor scenes are not so well defined as on cold days, when these images are sharper.

This impossibility to comprehensively reverse the heat equation (and parabolic equations in general) means that we have to search for other models to obtain enhancement.

The idea is to "locally invert" the heat equation by means of a non-linear process. This can be done by a recently derived Osher-Rudin model ([5]), which consists of the equation:

$$\boxed{\begin{array}{l} u_t + F(u_{xx})\,[u_x] = 0 \\ u(x,0) = u_0(x) \end{array}}$$

(where in this case $u_0(x)$ is not the original function but the deteriorated one) and it can be observed that there are several important aspects to the behaviour of its solution:

A) The solution of this model is of the kind known as "steady" (for example, the solution no longer varies in a visually detectable way from a t_0)

B) The solution creates jumps in the points of inflection (the ones where u_{xx} changes sign).

C) The solution has the same contrast as the original, in the sense that the local maximums and minimums are maintained.

Point (B) is important since the inflection curves match the profiles in two dimensions. The solution of the proposed equation thus gives us a steady image (so that one does not have to be concerned about how long is required t_0 to complete the resolution), which corresponds to the deteriorated image, but with the detectable enhanced profiles.

Function F is the one that carries out the task of detecting the profiles and has to comply with two conditions:

1) It must be continuous, in the mathematical sense, so that the problem has a stable solution.

2) F(v) must have the same sign as v.

Nevertheless, and in spite of any function with these properties taking us to the desired solution, the choice of F is the key to final quality (it allows the steady state to be reached sooner or later, and finding or not finding certain profiles).

4.- DESCRIPTION OF THE ALGORITHM:

In order to solve the equation, two factors were taken into account: speed and quality. Both points tend generally to be opposed and a balance has to found between final product quality and the time required to obtain it.

This equation, in two dimensions, is as follows:

$$\boxed{u_t + F(£\,(u))\,|\,grad(u)\,| = 0}$$

where £(u) is the Laplacian of u, and grad (u) its gradient.

Our first attempts turned to the mathematical model. First F has to be chosen to detect the largest possible number of profiles. For this purpose we took

$$F(v)=\text{signo}\ (v)=\begin{cases} 1, & v>0 \\ 0, & v=0, \\ -1, & v<0 \end{cases}$$

and we convoluted this with different nuclei in order to "soften" the function (to make it possible to differentiate this with no problems), up to:

$$F(v)=\frac{v}{1+|v|^2}$$

The softer the F, the steadier the solution (less possibility of error arising), but the longer it takes to detect deteriorated profiles, so that the choice of F is made in accordance with the deterioration of the image. For the examples tested, it sufficed to use the last function.

We also worked with the Laplacian. The one we used is of the form $au_{xx} + bu_{xy} + cu_{yx} + du_{yy}$, where a, b, c and d are parameters which adjust the horizontal, diagonal and vertical deterioration. As we do not have any data in the cases studied, it was assumed that a=b=c=d=1. If additional data were available (for instance if one could roughly follow the direction of the screen), the parameters could be readjusted.

Finally, we get to the algorithm used. The equation is non linear, as the coefficient of grad(u), F(L(u)), clearly depends on the same solution. One way to tackle the problem would be to consider the equation locally (subdividing the image) and consider this to be locally linear (F(L(u)) constant in each part). The other is to attempt to apply strictly non-linear methods which already proved their worth in other types of problems. In particular, we used ENO and PHM methods ([6],[7]) to implement the algorithm, these being of high order, so that the quality of the calculated solution is high, and though not being so fast as the linear methods, or the non-linear first order methods, their respective rates are often quite alike.

Flow diagram of the non-linear enhancement algorithm:

SOURCE IMAGE

⇩

SELECTION OF PIXELS

⇩

$$[S]=\begin{bmatrix} S_{-1,-1} & S_{0,-1} & S_{1,-1} \\ S_{-1,0} & S_{0,0} & S_{1,0} \\ S_{-1,1} & S_{0,1} & S_{1,1} \end{bmatrix}$$

⇩

CALCULATION OF THE COEFFICIENTS OF THE REFERENCE MATRIX FROM «S» LIGHT INTENSITIES

⇩

$$[f_S]=\begin{bmatrix} f_{-1,-1} & f_{0,-1} & f_{1,-1} \\ f_{-1,0} & f_{0,0} & f_{1,0} \\ f_{-1,1} & f_{0,1} & f_{1,1} \end{bmatrix}$$

⇩

CONVOLUTION OF [f] CON [S]

Note that $[f_s]$ depends on [S]

5.- EXPERIMENTAL RESULTS:

The effect of the filter presented in this work on the same image on which the edges were focused by means of the Fourier transform can be seen in Figure 6.

The Gibbs phenomenon appears in the form of very dark or very light points, as can be observed in Figure 7.1. This phenomenon is not found when the enhancement filter that we have prepared is used, as shown in Figure 7.2. It can also be seen that the enhancement obtained by our filter is consistent with the original image, in the sense that all the profiles are enhanced in an even manner, unlike the Fourier type filters, where some zones are excessively enhanced and others are left almost unchanged. In the same way, we must stress the fact that in Figure 7.1 the light intensities deteriorated, whereas these were maintained in Figure 7.2.

The filter has been tried out on other images and designs, yielding similar results.

6.- CONCLUSIONS:

The algorithm presented suppresses blurring caused by problems in the optical equipment or by other problems of image deterioration, without producing additional noise.

Furthermore, the algorithm leaves a margin of freedom for the softness of the detector and directional parameters, which means better adaptation to each specific problem. The filter can thus be adjusted to the glaze diffusor coefficients. These possibilities are at present in the research stage.

7.- REFERENCES:

[1] A. Domingo; Tratamiento digital de imágenes. Published by ANAYA, 1993

[2] A. Droblas, S. Greenberg. Manual de Photoshop. Published by McGraw-Hill, 1994

[3] C. Watkins, A. Sadun, S. Manreka; Modern Image Processing. Published by Academic Press, 1993.

[4] R.H.T. Bates, M.J.McDonnel; Image Restoration and Reconstruction. Published by Oxford University Press, 1989.

[5] S. Osher and L. Rudin, *Feature-Oriented image enhancement using shock filters*, Sinum. vol. 27, (1990) pp. 919.

[6] A. Harten, B.Engquist, S. Osher and S. Chakravarthy, *Uniformly High Order Accurate Essentially Non-oscillatory Schemes III*, J. Comput. Phys., v. 71 No. 2, (1987), pp. 231-303.

[7] A. Marquina, *Local Piecewise Hyperbolic Reconstructions for Nonlinear Scalar Conversation Laws*, UCLA CAM Report No. 25-(1989), SIAM J. Scientific Comp., v. 15, (1994) pp. 892-915.

FIGURE 1 • POSITIVE-NEGATIVE OPERATIONS

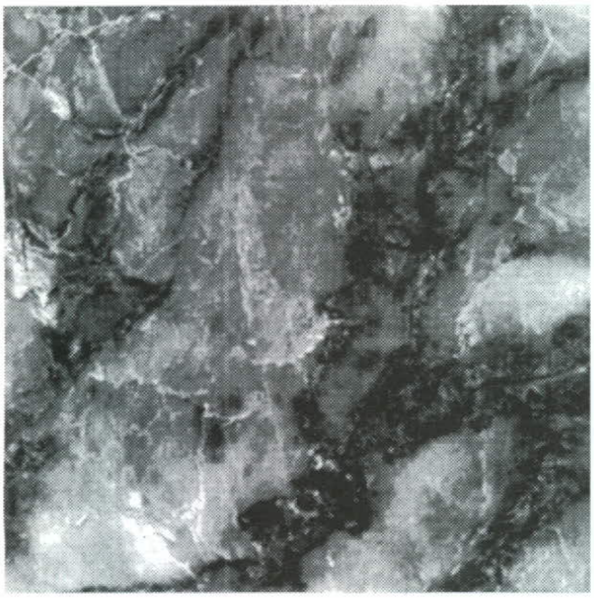

Fig. 1.1 - Positive image in grey scale

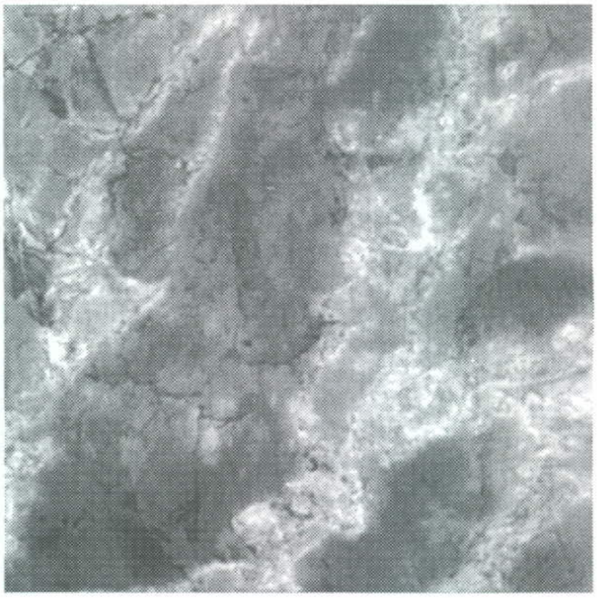

Fig. 1.2 - Negative image in grey scale

FIGURE 2: MEDIAN OPERATIONS

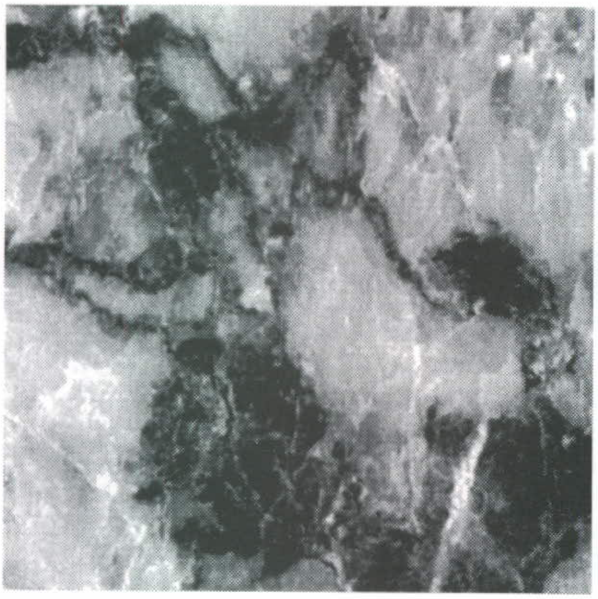

Figure 2.1: Positive C.M.Y.K. colour image

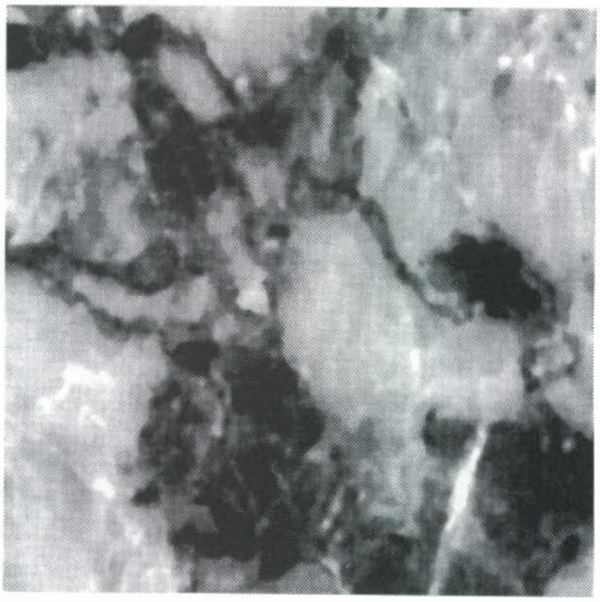

Figure 2.2: Median filter: radius 2 pixels

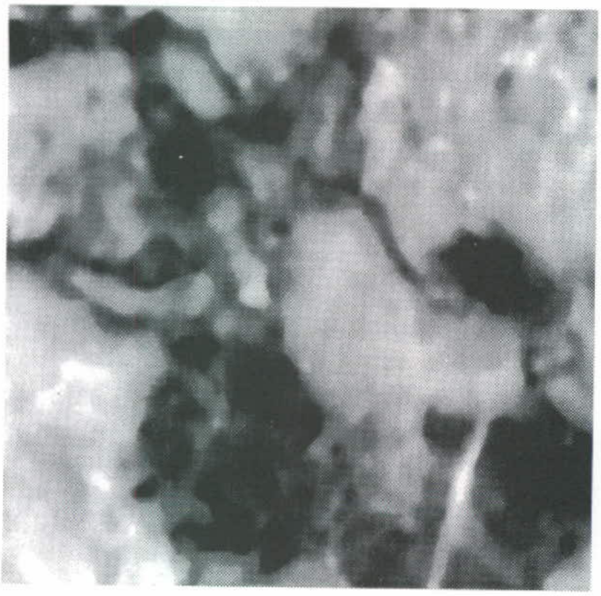

Figure 2.3: Median filter: radius 4 pixels

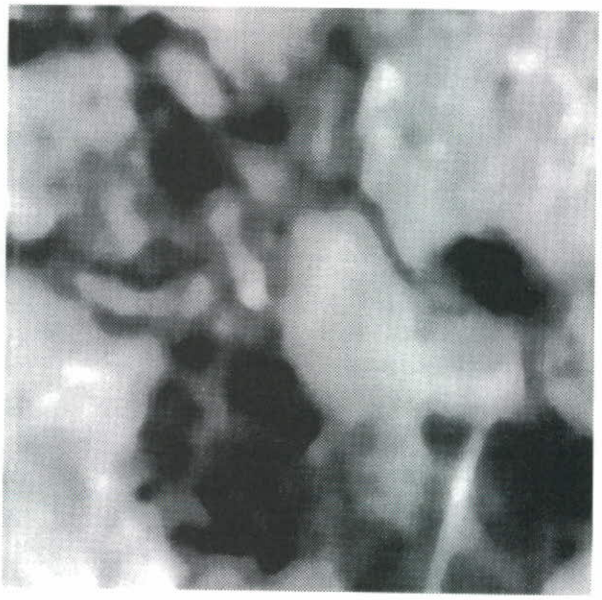

Figure 2.4: Median filter: radius 6 pixels

FIGURE 3: GRADIENT OPERATION: TRACING PROFILE

Figure 3.1: Original image in grey scale.

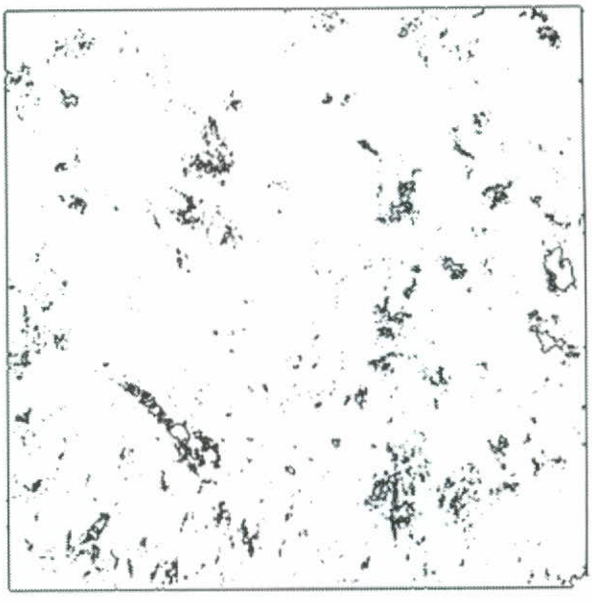

Figure 3.2: Gradient operation: level 255

FIGURE 4: MADE-TO-MEASURE FILTER, LAPLACIAN

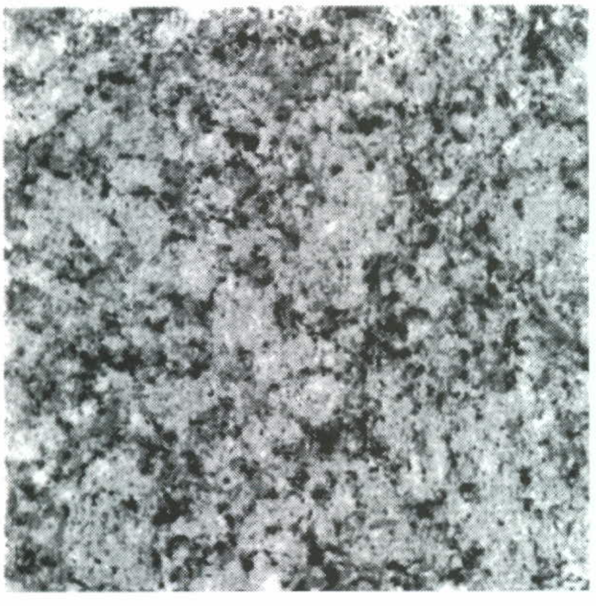

Figure 4.1: Original image

Figure 4.2: Made-to-measure filter: Laplacian

FIGURE 5: EDGE FOCUSING OPERATION

Figure 5.1: Original CMYK colour image

Figure 5.2: Edge focusing filter

FIGURE 6: ENHANCEMENT FILTER

Figure 6.1: Enhancement filter, 10 iterations.

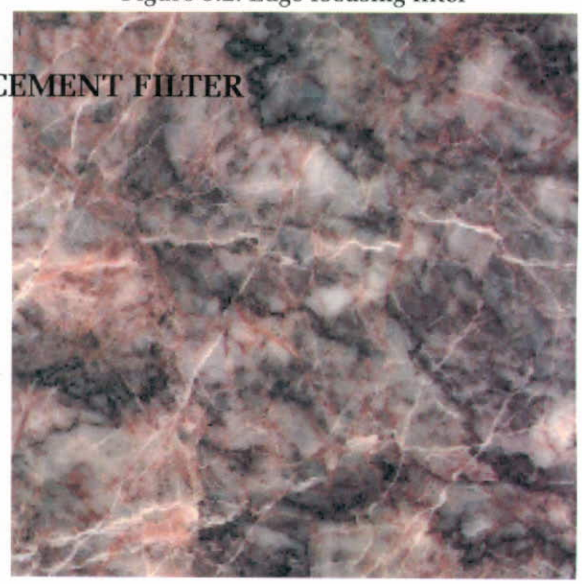

Figure 6.2: Enhancement filter, 20 iterations.

FIGURE 7: ENHANCEMENT AND EDGE FOCUSING FILTER

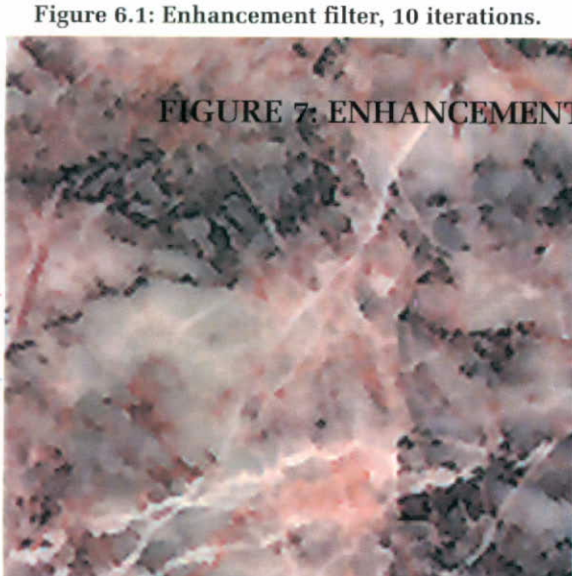

Figure 7.1: Enlarged section of the edge focusing filter.

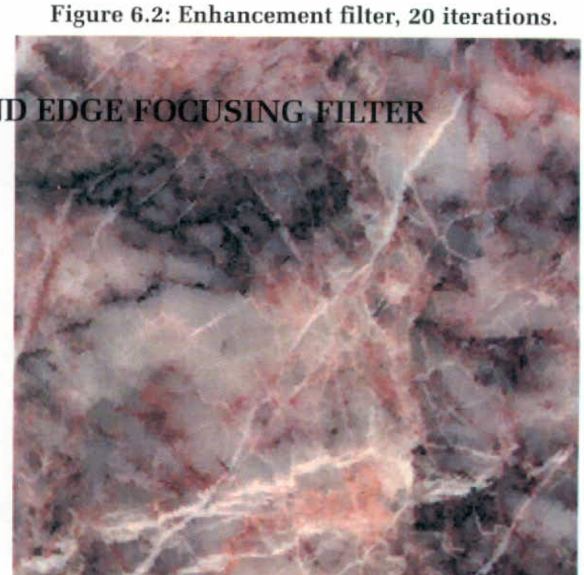

Figure 7.2: Enlarged section of the enhancement filter: 20 iterations.