Introduction

The rapid market penetration of medium- and high-resolution bitmapped displays has highlighted the practical difficulty of manipulating low-resolution digital halftone images (Perry & Wallich, 1985; Stoffel & Moreland, 1981). Most such devices are used primarily for presenting text and therefore have only one bit/pixel display capacity (Baldauf, 1985). Nevertheless it is occasionally useful to be able to show continuous-tone pictures, and several techniques for digital halftoning have been reported (Limb, 1969; Jarvis *et al*, 1976; Witten & Neal, 1982). Although present and projected screen resolutions obviate high-quality rendering, the rough representations that are possible have many applications. Easily recognizable facsimiles of human faces, for example, can be depicted on a binary matrix as small as 75×90 (Klein & Metz, 1978). Unfortunately once an image has been converted to the halftone representation it becomes very difficult to manipulate and is effectively frozen (Stoffel, 1982). This paper addresses the important problem of scaling pictures which are intended for display in halftone format.

The picture to be scaled may be presented in either greyscale or bilevel form. In the first place, of course, any photographic pictures will have been digitized into a greyscale representation. However, in many situations this representation may have been converted to a bilevel one through any of a number of halftoning techniques, and it may be impossible or inconvenient to obtain the original greyscale digitization. A major result of this research is that reduction from a bilevel representation need not be significantly inferior, in terms of the perceptual result, than scaling from a greyscale original. Thus if pictures are to be displayed on a bitmapped screen in a variety of different sizes, they can be halftoned in the maximum-size format and reduced directly from this bilevel version when required. The greyscale original can safely be discarded, offering significant savings in storage and handling costs because only one version need be saved. Consequently, although the techniques presented here permit scaling from either greyscale or bilevel images to a bilevel halftone, we emphasize the use of bilevel source images since this is likely to be the most popular mode in practice.

The next section introduces basic concepts of discrete raster images and discusses the general problem of scaling halftones. Following that we summarize the standard treatment of the interpolation problem as it applies to continuous-tone images. The subsequent section applies interpolation to bilevel images. One ubiquitous problem in image processing is the appearance of unwanted moire fringes and other artifacts, and this is avoided by the use of a Peano scan as described next. Finally we present and discuss some representative results of the scaling method that has been developed.

Scaling raster images

A raster image is a two-dimensional array of pixels which are sampled values of a continuous original picture. Sampling in the spatial domain alone yields the *continuous-tone* raster representation of the picture, where each pixel is assigned a real number to describe its intensity (see Figure 1). However, any given output device can show only a finite number of shades of grey at each pixel, and therefore the intensity must be quantized to one of the grey levels which are available on the display. Such a representation, quantized in intensity as well as spatially sampled, is known as a *greyscale* image. Its fidelity depends on the number of available grey levels and the method by which quantization is performed. Typically there are 256 levels, represented by integers; quantization is by rounding or truncation to the nearest level; and no degradation is noticeable between the continuous-tone and quantized versions.

In the case of an output device which is only capable of showing two levels of grey (ie either black or white), however, the quantization error becomes very significant. In general it is *not* sufficient simply to round the intensity value to the nearest level. Instead a halftone representation is used to approximate the greyscale picture by varying the density of dots over each small local area. This can produce the same perceptual effect as a continuous-tone original because of the trade-off between greyscale sensitivity and frequency response of the human eye (Stoffel & Moreland, 1981).

Several methods have been reported for scaling binary images, but all are tailored to such applications as fonts, icons, and other drawings which are inherently bilevel (Abdou & Wong, 1982; Casey *et al*, 1982; Ulichney & Troxel, 1982). Such algorithms identify higher-level features, notably outlines or edges, apply interpolation to them, and reconvert the scaled image description to pixel form. However, there is a world of difference between images which are inherently bilevel and ones which are halftoned representations of continuous-tone pictures. In a halftone the dot pattern provides local approximations to the tones of the original picture. Unlike inherently bilevel pictures, description in the language of outlines is meaningless. Even if higher-level features could be identified, they would not necessarily convey any information about the image, but could be artifacts of the halftoning process. Thus existing scaling algorithms are simply inappropriate for halftone pictures.

It seems clear that any reducing technique for halftone pictures must proceed by performing an integration operation on the source image, scaling the integrated version, and re-converting the scaled version to the bilevel representation for display (Pratt, 1978). Any interpolation scheme which does not involve integration and instead operates on a purely local basis runs the risk of slipping between the cracks and missing important features by accident. Ideally, perhaps, one could use precise knowledge of the original halftoning process to reconstruct an integrated version which closely approximates the greyscale image that is being represented. This work, however, makes the (realistic) assumption that such knowledge is unavailable. Moreover, because scaling from an original greyscale version is only marginally better than scaling from a halftone, such knowledge would make little difference to the result.

Figure 1 depicts the various image representations and transformations that have been mentioned. This paper is concerned with the two reducing transformations (shown as thick lines), and the development applies equally to reduction from the greyscale or bilevel representations.

Interpolation

Scaling a sampled function, whether two-dimensional or otherwise, is equivalent to resampling the original continuous function. This can be accomplished by first reconstructing the original, and then sampling it on a different set of points. The two processes are usually performed in a single step by interpolation, as discussed in detail by Pratt (1978). Here we take a slightly different approach by integrating over each pixel-sized square. The purpose of this is to ensure that the overall picture intensity is preserved exactly.

Consider a continuous intensity function $f(x, y)$ sampled at unit intervals along the x and y axes in an $m \times n$ grid. The resulting representation is an array of intensity values $f(i, j)$, $i = 0 \cdots m-1$, $j = 0 \cdots n-1$. The energy of an individual pixel is

$$F(i,j) = \int_{i-0.5}^{i+0.5} \int_{j-0.5}^{j+0.5} f(i,j)\,dx\,dy = f(i,j),$$

since each pixel occupies a unit square.

Magnification or reduction is equivalent to a change of scale, replacing the original $m \times n$ grid with a new $M \times N$ one. The ideal re-sampled set of values will be

$$f(X_I, Y_J), \quad \text{where } X_I = \frac{m-1}{M-1} \times I, \ Y_J = \frac{n-1}{N-1} \times J; \quad I = 0 \cdots M-1, \ J = 0 \cdots N-1.$$

To reconstruct the original function $f(x, y)$ from the samples $\{f(i,j) \mid i = 0 \cdots m-1, j = 0 \cdots n-1\}$, an interpolating function $g(x, y)$ should be chosen and the original function approximated by a sum of the sampled values weighted by $g(x, y)$ translated to center at the individual points:

$$f^*(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} g(x-i, y-j)\, f(i,j).$$

A number of interpolating functions are possible, such as sinc, Gaussian, cubic spline, bilinear, and uniform. Naturally there is a trade-off between complexity (manifested in computation time) and interpolation error, as discussed in detail by Pratt (1978). In most cases the interpolating function is "separable", in other words its effects in the x and y directions are independent: $g(x,y) = h(x)\,h(y)$. Once the continuous function $f^*(x, y)$ has been reconstructed, scaling can be performed by resampling at the new grid points X_I, Y_J to yield a new set of values $f^*(X_I, Y_J)$.

The total energy of an individual pixel in the new grid will be

$$F^*(X_I, Y_J) = \int_{X_{I-0.5}}^{X_{I+0.5}} \int_{Y_{J-0.5}}^{Y_{J+0.5}} f^*(x,y)\,dx\,dy = \int_{X_{I-0.5}}^{X_{I+0.5}} \int_{Y_{J-0.5}}^{Y_{J+0.5}} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} g(x-i, y-j)\, f(i,j)\,dx\,dy,$$

in terms of the original sampled function values $f(i, j)$. Assuming that the interpolating function is separable, and interchanging the order of integration and summation, this can be re-written

$$\begin{aligned} F^*(X_I,Y_J) &= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \int_{X_{I-0.5}}^{X_{I+0.5}} \int_{Y_{J-0.5}}^{Y_{J+0.5}} h(x-i)\, h(y-j)\, f(i,j)\,dx\,dy \\ &= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} f(i,j) \int_{X_{I-0.5}}^{X_{I+0.5}} h(x-i)\,dx \int_{Y_{J-0.5}}^{Y_{J+0.5}} h(y-j)\,dy. \end{aligned} \qquad (1)$$

This expression will of course be a real number and results in a greyscale picture. For bilevel presentation it is necessary to reconvert it back to halftone.

Combining interpolation with halftoning

It is possible to combine the operations of interpolation, integration and halftoning into a single process. The pixels in the target image are mapped on to the original and visited one after another. An estimate is made of the blackness, or energy, of the corresponding area of the original image. These estimates are summed, and if the total so far exceeds a pre-determined threshold value, the current target pixel is blackened and the total reduced by a corresponding amount. The effect is to maintain a running estimate of the error between the integrated energy of the pixels already visited in the target image and the integrated energy of the same area in the original. Pixel values are chosen to minimize the accumulated error at each stage.

Here is a fragment of Pascal code for the operation on one pixel, with coordinates *XI*, *YJ*. The reconstructed continuous intensity function $F^*(X_I, Y_J)$ is called *Fstar*(*XI*, *YJ*), and its calculation according to the expression derived above will be discussed shortly.

```
const  threshold = · · ·
var    error, XI, YJ : integer;

       . . .

error := error + Fstar(XI, YJ);
if error ≥ threshold
       then
       begin
           set(XI, YJ);       {* turn on pixel XI, YJ *}
           error = error – threshold
       end
       else
           reset(XI, YJ);     {* turn off pixel XI, YJ *}
       . . .
```

Threshold is chosen so as to equalize the intensities of the target and original images, and is set to the energy of a black area whose size is that of one pixel in the new image. It is easy to verify that this is $(m \times n)/(M \times N)$ times the greyscale value of a black pixel, assuming (as is conventional) that the interpolation function is normalized to integrate to 1.

The code works as follows. *Error* should be initialized anywhere between 0 and *threshold*. *Fstar* will return a value in the same interval. At the end of the code *error* will once again lie in that interval. It represents the cumulative discrepancy between the interpolated pixel values already encountered and the pixel values deposited so far, and the effect of the **if** statement is to keep this discrepancy always between 0 and *threshold*. Since 0 and *threshold* are the energies of white and black pixels respectively, the cumulative blackness of the bilevel image is always accurate to within one pixel.

Scanning the image

Any scanning regime that visits each target pixel exactly once will suffice to minimize the overall intensity error and preserve the total brightness of the picture. However, a good method will minimize the error over every area of the image, no matter how small. It will also suppress spurious texture patterns, along with moire fringes and other artifacts. Finally, negligible computation should be necessary to locate the next pixel.

The Peano scan exhibits these desirable properties (Peano, 1890; Hilbert, 1891; Witten & Wyvill, 1983), and has been used to good effect in a number of image processing applications (eg Witten & Neal, 1982; Stevens *et al*, 1983; Lempel & Ziv, 1986). It divides the plane into four quadrants and visits all pixels within one quadrant before moving on to the next. This property holds recursively, each quadrant being subdivided into four, right down to individual pixels (Figure 2).

The algorithm given above ensures that the cumulative energy of the bilevel image is always within one pixel's worth of the cumulative energy of the original over any continuous segment of the scan. The fact that the Peano curve exhausts each recursively-defined quadrant before moving on to the next means that the total energy over every quadrant is accurate to within one pixel. This is a remarkable guarantee that the cumulative error introduced by the algorithm is low over both large and small regions of the image.

Interpolating functions

The process of scaling can be done in a single pass using the Peano scan. The code fragment given above will be executed $M \times N$ times, once for each pixel in the target. Each time, the function *Fstar* must be executed, involving the calculation of expression (1). (Note that henceforth the interpolation function is assumed separable.) The integrals of h can be calculated in advance for each pair (I, i) and (J, j), and stored for use when scaling pictures. Even so the execution of *Fstar* will involve the evaluation of the double sum, one operation for each pixel of the source image. In practice, however, this enormous computation is avoided by sharply truncating the interpolating function h. This also reduces the number of integrals of h that must be stored, for if $h(x)$ is zero for $|x| > k$, then

$$\int_{X_{I-0.5}}^{X_{I+0.5}} h(x-i)\, dx$$

is non-zero only when $X_{I-0.5} - k \le i \le X_{I+0.5} + k$.

We have experimented with three interpolating functions, ***nearest neighbor***, ***average intensity***, and ***bilinear***. More sophisticated interpolation is inappropriate because of the high granular noise exhibited by all halftone images.

Nearest neighbor interpolation, the crudest method, is used as a basis for evaluating the other schemes. Figure 3 illustrates the new, scaled grid superimposed in thick lines on the original one. Nearest neighbor takes the closest original pixel to the center of the new one and uses its value (black or white) for the new pixel. It requires no multiplications, since normalization can be accomplished by choosing the threshold to be 1, the energy of a black pixel.

Average intensity interpolation estimates the energy throughout the area covered by the new pixel. This involves not only summing over a number of old pixels, but also weighting partially-covered pixels in accordance with their overlap area. For example, in Figure 3 nine pixels are involved, one having a full contribution and the others partial ones. This scheme corresponds to the interpolating function

$$h(x) = \begin{cases} 1 & -0.5 \le x \le +0.5 \\ 0 & \text{otherwise,} \end{cases}$$

and the integrals of h provide weights which account correctly for the contributions of partially-covered pixels. Note that no smoothing is introduced: the area attributed to one (old) pixel does not overlap with that of any other.

For reduction down to a scaling factor of 0.5, at least 4 and at most 9 old pixels contribute to each new one. In general, an average of $(1+\frac{m}{M})\,(1+\frac{n}{N})$ old pixels contribute to each new one. The calculation of (1) requires one addition and two multiplications for each old pixel (i, j) in the summation.

Bilinear interpolation is accomplished by the function

$$h(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{otherwise.} \end{cases}$$

This performs smoothing by including some energy outside the region covered by the new pixel. It also weights energy at the center of each pixel more heavily than that at the edges. Because its spread is wider than that of the average intensity method, the number of old pixels that contribute to each new one increases to $(2+\frac{m}{M})\,(2+\frac{n}{N})$. As before, the interpolation coefficients are evaluated using the integrals of expression (1).

Results

Figures 4–6 show a number of examples of reduction using the scaling method developed. In each case the upper picture shows the halftoned version of a 256×256 original image. All halftones use the Peano scan technique reported earlier (Witten & Neal, 1982). The other images in each Figure are scaled to 192×200 pixels. At the left is the result of scaling the halftoned version directly; on the right is the result of scaling the original greyscale image. The method developed above is used for reduction in both cases. Because the original greyscale values are preserved, very much more information is used to produce the rightmost pictures than the leftmost ones.

Pictures (b) and (c) of Figure 4 illustrate the problems that occur with nearest-neighbor interpolation — which is not really "interpolation" at all. Each target pixel depends upon just one source pixel, and line segments are missing from various parts of the circles. Pictures (d) and (e) show how average-intensity interpolation solves the problem by producing much more uniform lines. When rendered using bilinear interpolation (not shown) the circles remained uniform but appeared noticeably fuzzier because of the smoothing effect. On all other pictures the bilinear method achieved no perceptible improvement in picture quality, despite its higher computational cost. Consequently the remaining figures were all produced using the average-intensity method.

Figure 5 shows a striped pattern which emphasizes the difference between scaling from a halftone (b) and scaling from a greyscale image (c). Both versions are quite acceptable, and artifacts are avoided. However, slightly higher frequency resolution is apparent in (b). In fact, considering the very much greater amount of information available in the greyscale original, it is astonishing that the effect is not more marked. That the difference is minimal in practice is illustrated by Figure 6 which shows a portrait. It is doubtful whether anything is lost by working from the halftone instead of the greyscale original.

In all cases it is evident that the overall intensities of the original and scaled versions are the same. If desired, the brightness can easily be altered by adjusting the threshold.

Conclusions

A method has been described which enables a halftone image to be reduced to an arbitrary size, which may be different in x and y. It combines in a single step interpolation, integration and reconversion into halftone. The Peano scan is used to visit the pixels one by one and guarantees that the intensity error is minimized over both large and small areas of the image. Another benefit is that spurious textured patterns and other artifacts are suppressed. The amount of computation involved is small and realistic, involving just a few multiplications for each pixel of the reduced image. Satisfactory results have been obtained with the simple "average intensity" interpolating function.

A striking result of the work is that the reduced pictures are not noticeably different whether one begins with a bilevel or greyscale original. Consequently if pictures are sometimes viewed on a bitmapped screen at a given size and sometimes reduced, little is lost by discarding the greyscale original and reducing directly from the maximum-size bitmap.

Acknowledgement

This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

Abdou, I.E. and Wong, K.Y. (1982) "Analysis of linear interpolation schemes for bi-level image applications" *IBM J Research and Development, 26,* 667-680.

Baldauf, D.R. (1985) "The workhorse CRT: new life" *IEEE Spectrum, 22* (7) 67-73, July.

Casey, R.G., Friedman, T.D., and Wong, K.Y. (1982) "Automatic scaling of digital fonts" *IBM J Research and Development, 26,* 657-666.

Hilbert, D. (1891) "Ueber die stetige Abbildung einer Linie auf ein Flachenstuck" *Math Annalen, 38,* 459-460.

Jarvis, J.F., Judice, C.N., and Ninke, W.H. (1976) "A survey of techniques for the display of continuous tone pictures on bilevel displays" *Computer Graphics and Image Processing, 5,* 13-40.

Klein, E. and Metz, H.J. (1978) "Evolution in image science" in *Advances in digital image processing,* edited by Stucki, P., pp 3-20. Plenum Press.

Lempel, A. and Ziv, J. (1986) "Compression of two-dimensional data" *IEEE Trans Information Theory, IT-32* (1) 2-, January.

Limb, J.O. (1969) "Design of dither waveforms for quantized visual signals" *Bell System Technical J, 48,* 2555-2582.

Peano, G. (1890) "Sur une courbe, qui remplit toute une aire plane" *Math Annalen, 36,* 157-160.

Perry, T.S. and Wallich, P. (1985) "Computer displays — new choices, new trade-offs" *IEEE Spectrum, 22* (7) 52-59, July.

Pratt, W.K. (1978) *Digital image processing.* Wiley.

Stevens, R.J., Lehar, A.F., and Preston, F.H. (1983) "Manipulation and presentation of multidimensional image data using the Peano scan" *IEEE Trans Pattern Analysis and Machine Intelligence,* 520-, September.

Stoffel, J.G. and Moreland, J.F. (1981) "A survey of electronic techniques for pictorial image reproduction" *IEEE Trans Communications, COM-17* (12) 1898-1925, December.

Stoffel, J.C. (1982) *Graphical and binary image processing and applications.* Artech House, Chapter 9: 493-494.

Ulichney, R.A. and Troxel, D.E. (1982) "Scaling binary images with the telescoping template" *IEEE Trans Pattern Analysis and Machine Intelligence, PAMI-4* (3) 331-335.

Witten, I.H. and Neal, R. (1982) ''Using Peano curves for bilevel display of continuous-tone images'' *IEEE Computer Graphics, 2* (3) 47-52, May.

Witten, I.H. and Wyvill, B. (1983) ''On the generation and use of space-filling curves'' *Software — Practice and Experience, 13*, 519-525.

Captions for figures

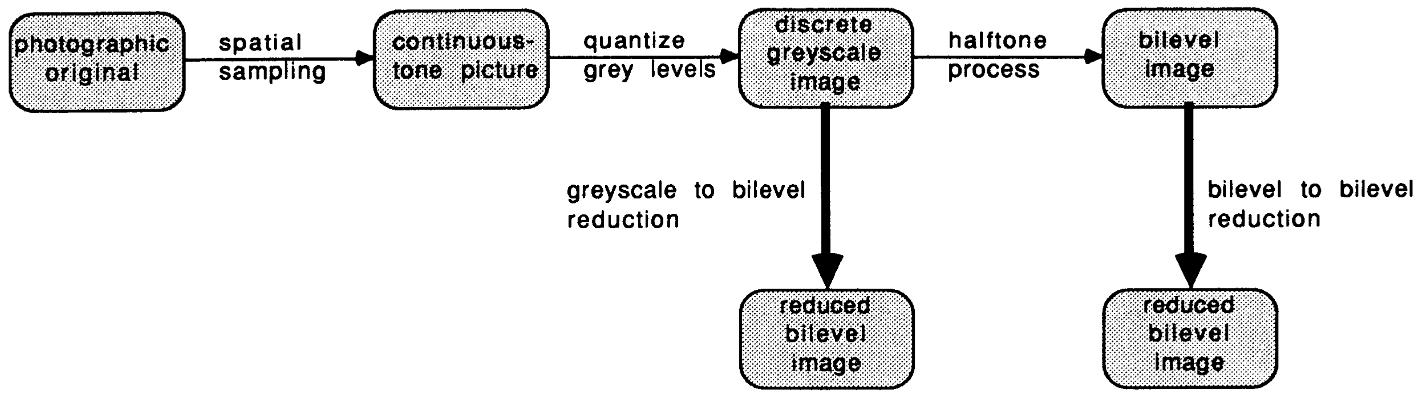

photographic original
spatial sampling
continuous-tone picture
quantize grey levels
discrete greyscale image
halftone process
bilevel image
greyscale to bilevel reduction
reduced bilevel image
bilevel to bilevel reduction
reduced bilevel image

Figure 1

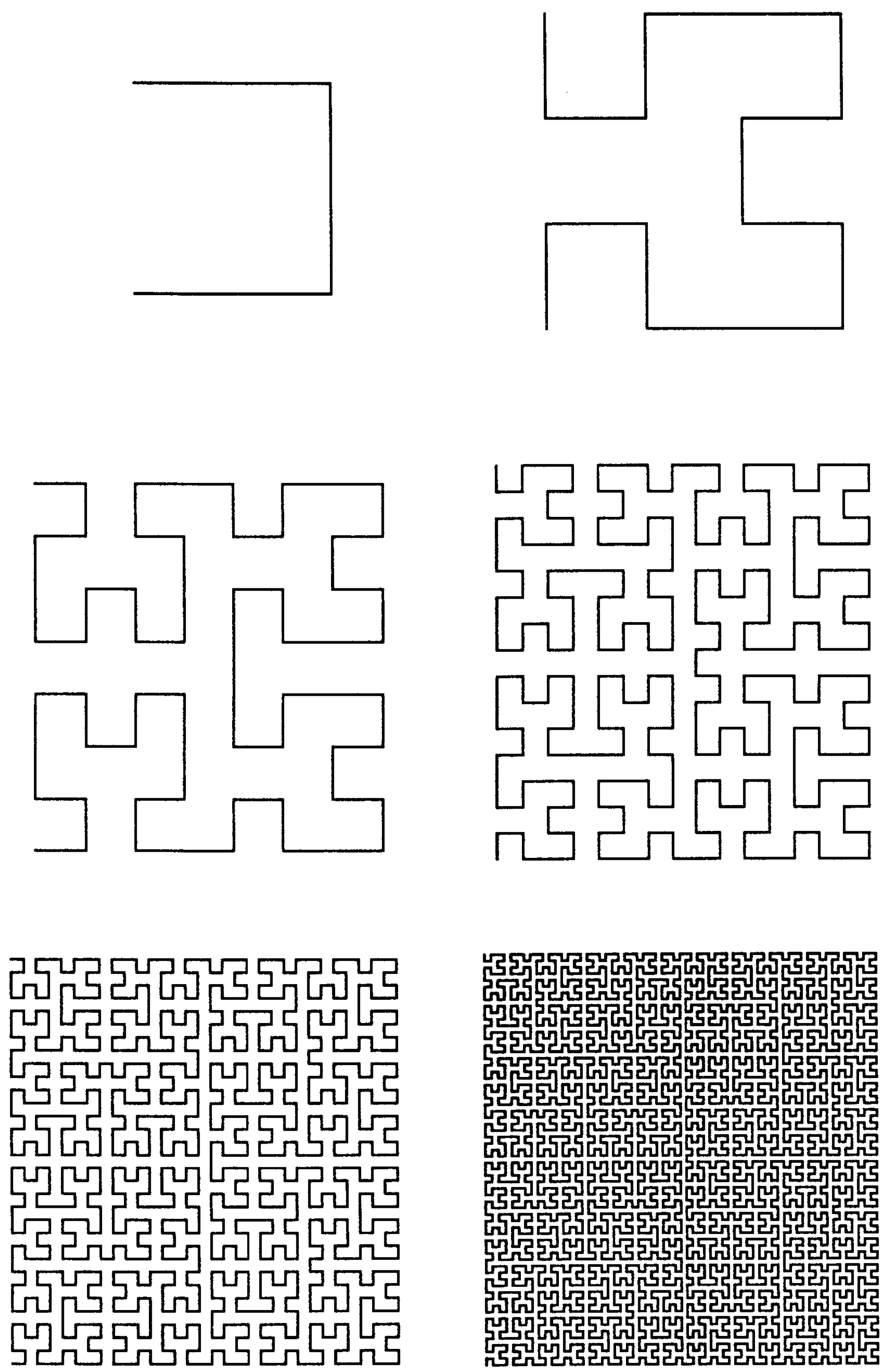
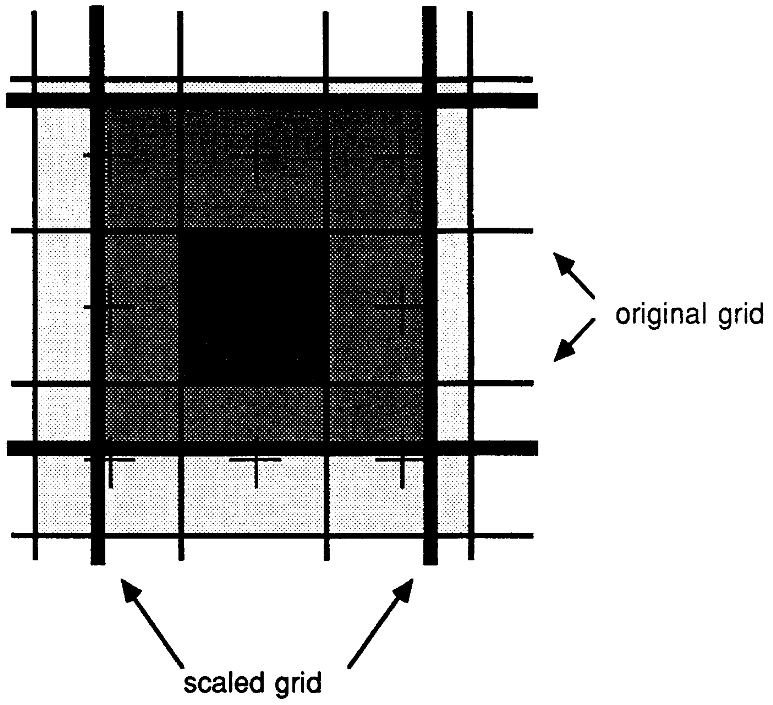


Figure 2



■	nearest-neighbour pixel
▩	new pixel coverage
░	average-intensity pixels

Figure 3

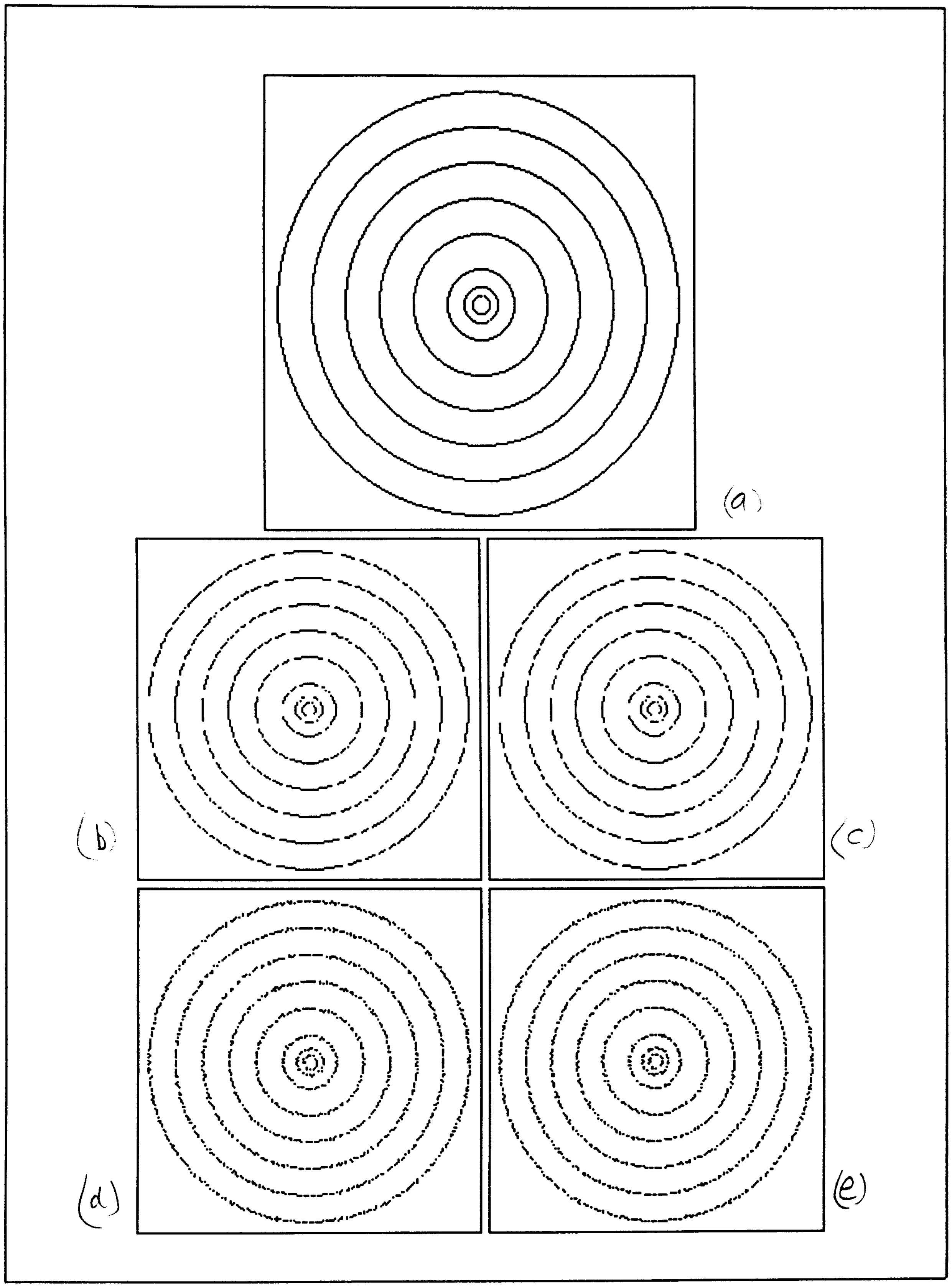


Figure 4

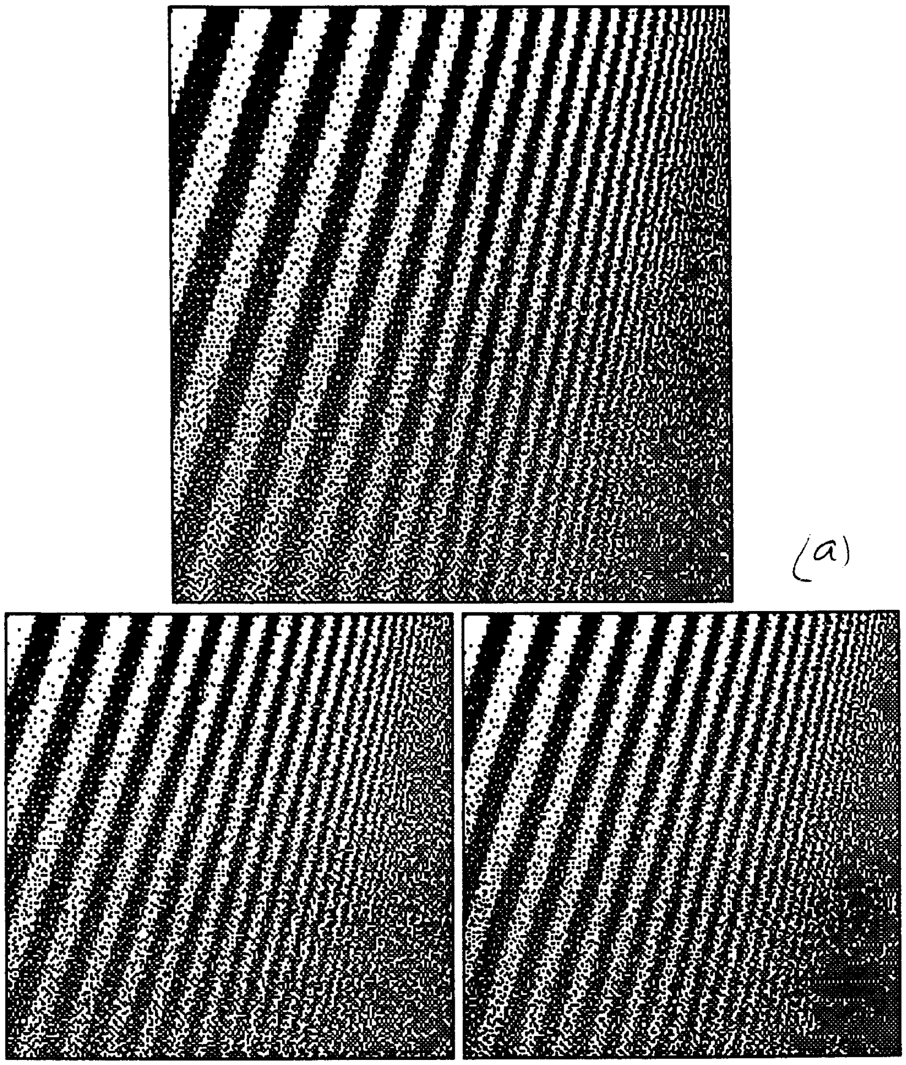


Figure 5

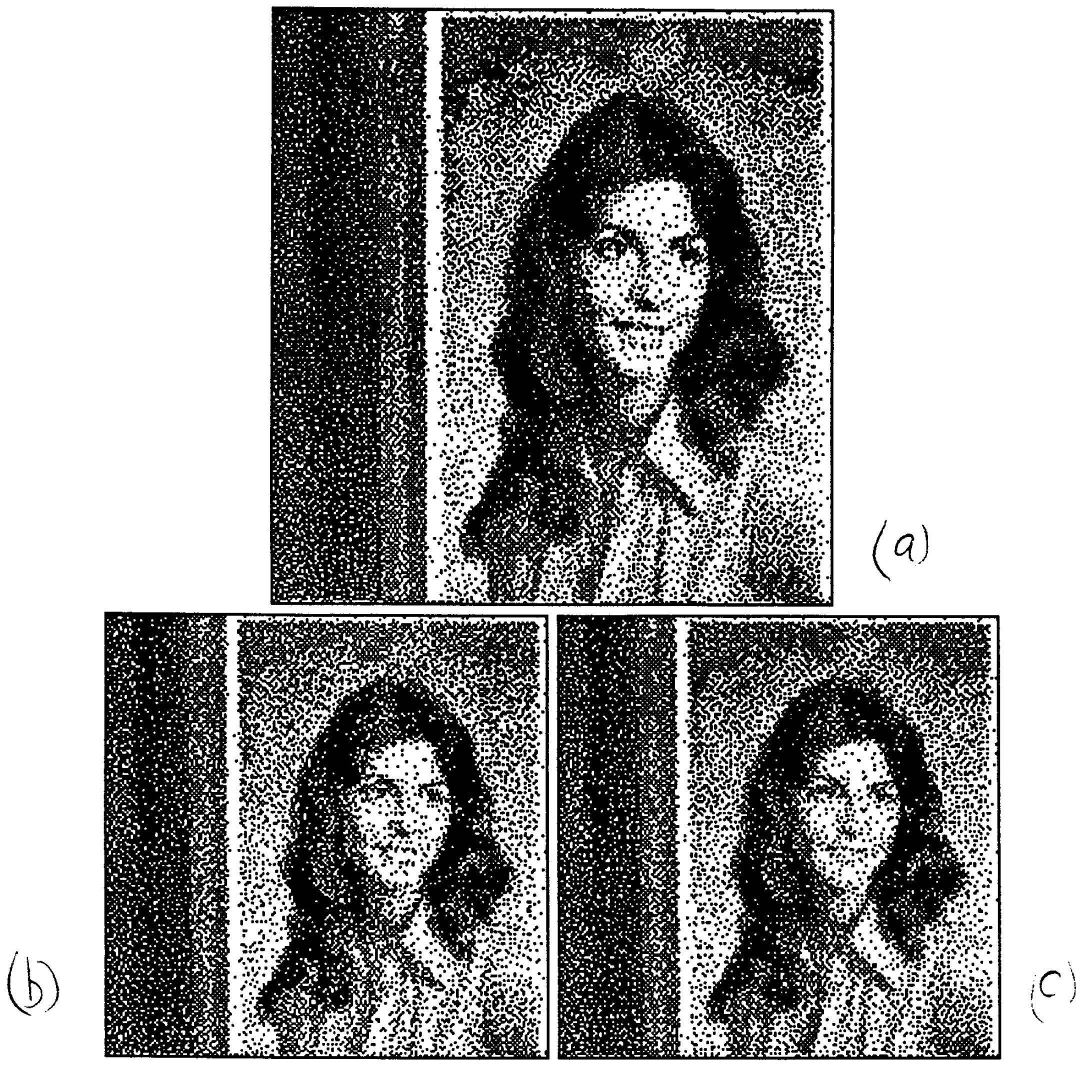


Figure 6