AQTR 2004 (THETA 14)
2004 IEEE-TTTC-International Conference on Automation, Quality and Testing, Robotics
May 13 – 15, 2004, Cluj-Napoca, Romania

Improved quad-tree generation of ordered dithering mask

Radu-Lucian Lupşa
Babeş-Bolyai University, Faculty of Mathematics and Computer Science
rlupsa@cs.ubbcluj.ro

Abstract

Rendering grayscale images on printers requires transforming images form grayscale to bi-level black and white. Ordered dithering is the fastest algorithm for this transformation. It has been made popular with the blue noise mask algorithm. We propose here a novel algorithm for producing the mask for ordered dithering.

1 Introduction

Rendering a grayscale image on a printing device requires that the image be transformed from grayscale to black and white. Therefore, the gray levels must be transformed into a larger or smaller density of black points on the white background (paper). The process is commonly known as dithering or halftoning.

In spite of the increased computing speed, the quest for good halftoning algorithms is still open, as ever larger images are required to be printed at ever higher resolutions.

A very fast halftoning algorithm is the ordered dithering. The dithering algorithm starts form the gray-level image , and a mask (either the size of the image, or tiled to that size). Each pixel in the image is compared to the corresponding mask value. If the pixel value is greater, the corresponding pixel in the output image is set to 1, otherwise to 0. The halftoning itself is very simple and very fast; on the other hand, generating the mask is a difficult problem, that had been unsolved for a long time. Now, the most popular solution is the so-called *blue noise mask*, which consist in a direct filter of the Fourier transform with a blue noise filter. Unfortunately, the algorithm is very slow, and moreover the method is patented.

2 The simple quad-tree algorithm

The algorithm we propose here is an extension of the quad-tree algorithm given in [5]. We give here a brief description of the quad-tree algorithm.

The mask to be constructed must be square, and the sides length, n, must be a power of 2.

The mask is divided into four sub-squares, recursively divided again until the size of one pixel. Those sub-squares form a quad-tree.

Now, the mask construction goes as follows: For each value from 0 to $1 - \frac{1}{n^2}$ in increments of $\frac{1}{n^2}$, choose a path (see below) from the top of the quad-tree towards a leaf. The value is then written to that leaf. For each node the path pass through, a random permutation of the 4 sons is choosen, and each time the path reaches that node the next son is choosen as the next node for that path. Each time a cycle of four path is completed, a new random permutation is generated for that node.

3 Improuved mask generation techniques

The advantage of the quad-tree algorithm is, besides its simplicity and speed, the following property: when halftoning an uniform gray image, for any square corresponding to a node of the quad-tree, the ratio between the number of white pixels and the total number of pixels is equal, whithin one pixel, to the gray level. However, this property holds only for the squares that correspond to the nodes of the quad-tree; for the others squares, the difference can be quite large, and, as a result, artefacts an apear, especially on large and uniform zones of the image.

This paper describes an approach to reducing the deviation from an uniform pattern. The ideea is a combination of the quad-tree algorithm and the blue-noise filtering.

Like the quad-tree algorithm, this algorithm fills in the mask with values from 0 to $1 - \frac{1}{n^2}$ in increments of $\frac{1}{n^2}$, and for each value a path is constructed in the quad-tree, form the root to a leaf, and the value is written in the matrix in the element corresponding to the leaf of the quad-tree. However, we modify the algorithm for choosing, when the path reaches an internal node, which node will be the next along the path.

In the original algorithm, when the path reaches for the first time a given node, the next node is randomly choosen from the four sons; the next time it is choosen from the remaining three (the son that was choosen the first time is no longer a candidate), the third time we have to choose between he remaining two sons, and finally the fourth time we choose the last son, and then we start over the algorithm. All the random choices give equal probability to each son.

The modified algorithm, instead of choosing with equal probability the next node among the candidats, gives to each a probability proportional to the inverse of a penalty. Before expalining how the penalty is calculated, let's remark that, if we must produce an uniform gray of $\frac{k}{n^2}$, the pixels corresponding to the first k paths will be white, and the remainining pixels will be black. Ideally, the Fourier transform of the halftoned image will have blue-noise characteristics [2]: except for the 0 frequency coefficient, the low frequency coefficients will be low, then the coefficients will attain a maximum around and finally the coefficients will fall to a constant value. Three characteristics are important: that the spectrum be symmetrical around zero — so that the halftoning pattern be isotropical, that the low frequency coefficients be small — so that there be no dark or light "stains", and that there are no "spikes" in the rest of the spectrum — so that there are no visible regularities.

The penalty shows "how bad" the resulting halftoning pattern would be. The penalty is computed as follows:

- First, we consider a matrix, corresponding to the level in the quad-tree just below that where the choice must be made; the matrix has an element for each node of the quad-tree of the *next* level. That is, four of its elements would correspond to the four sons between which the choice must be made. The value of each element is the number of values already written to the mask, in positions corresponding to descendents of that node.

- Next, we compute the discrete Fourier transform of the matrix above. If the matrix is larger than 8×8, we consider only the 8×8 region centered on the four nodes between which we must choose.

- We notice at this point that, the matrix of which we compute the Fourier transform contains only two distinct values, and moreover the difference between them is 1. We compute the *reference frequency* as $f_r = \sqrt{n_p}$, where n_p is either the number of elements having the larger value, or the number of elements having the smaller value, whichever is smaller.

- Finally, the penalty takes into account the magnitude of the Fourier coefficients corresponding to frequencies below the reference frequency — they must be as small as possible — plus the variation of the rest of the coefficients — they must not have spikes and must be radially symmetric.

4 Comparisons and conclusions

Figure 1 compares the halftoning patterns for graylevels of 50%, 75% and 96%, for the following penalty functions:

a) original algorithm [5], which is equivalent of the special case with penalty=1

b) penalty is the sum of the Fourier coefficients corresponding to frequencies below $f_r/2$

c) penalty is the square of the largest Fourier coefficient for a frequence below $f_r/2$, plus the sum of the difference between consecutive Fourier coefficients, taken in increasing frequency order.

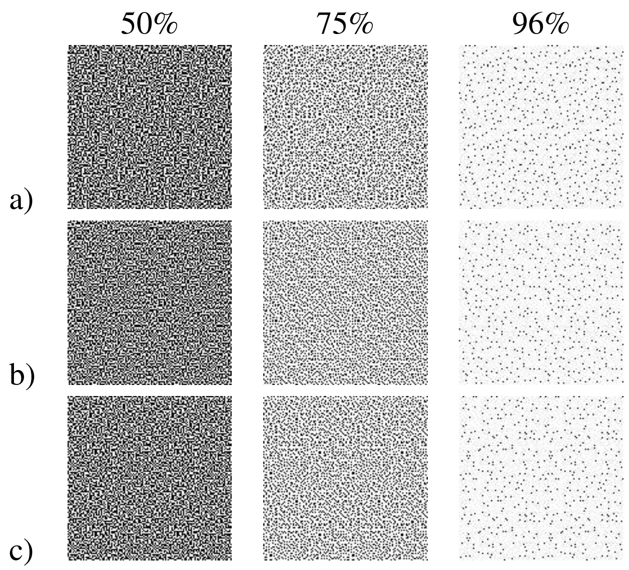

Figure 1: Unform gray comparisons

References

[1] VICTOR OSTROMOUKHOV *A Simple and Efficient Error-Diffusion Algorithm* Computer Graphics Proceedings, Annual onference Series, 2001, p. 567–572

[2] T. MITSA, K. J. PARKER *Digital halftoning using a blue noise mask* Journal of the Optical Society of America, 1992

[3] R. W. FLOYD, L. STEINBERG *An adaptative algorithm for spatial grey scale* Proc. Soc. Inf. Display, 17:75–77, 1976

[4] RADU LUPSA, DANA LUPSA *A Color Dithering Algorithm* Proc. of ICAM3, publ. in Buletinul Stiintific al Universitatii Baia Mare, vol. XVIII, nr. 2, 2002

[5] RADU LUPŞA *Quad-tree construction for ordered dithering mask* Buletinul Universităţii Politehnica Timişoara, ian. 2004