LARGE-SCALE INFOGRAPHIC IMAGE DOWNSIZING

Ruihua Ma
Institute for Infocomm Research, Singapore
21, Heng Mui Keng Terrace, Singapore 119613
Telephone: (65)68743112
ruihua@i2r.a-star.edu.sg

Gurminder Singh
Naval Postgraduate School, USA
Monterey, California 93943, USA
Telephone: (01)831-6563041
gsingh@nps.navy.mil

ABSTRACT

'Content Repurposing' is a way to convert existing multimedia content to suit mobile devices (handphones, PDAs, handheld PCs etc.), which have a lower display capability. As long as images are concerned, the main task is to downscale images and images of different types must be tackled differently in order to preserve as much visual content as possible. While in general most images stand fairly well downscaling, infographics (charts, maps, etc.) and other images containing thin structures are delicate to handle. Conventional methods tend to average out thin structures which usually convey critical information. Specifically designed techniques present various kinds of limitations. In this paper, we present a novel approach for infographic image downsizing. It relies on the concept of 'thin structure', the determination of which is related to the required downsizing scale. The algorithm is general, robust, without any limitation on the images. It is efficient because all operations are local. It works in an iterative way to achieve arbitrary large-scale downsizing. Result examples on real-world images are given to show the effectiveness of the approach.

1. INTRODUCTION

Image downsizing is a common operation in image processing and finds its application in many areas. One recent application is 'Content Repurposing'[1], which converts existing multimedia content to suit mobile devices (handphones, PDAs, handheld PCs etc.), which have a lower display capability. While in general most images stand fairly well downsizing, infographics (charts, maps, etc.) and other images containing thin structures are delicate to handle, even more so when large-scale downsizing is concerned.

Conventional interpolation-then-thresholding based methods tend to average out thin structures which usually convey critical information (**Figure 2**). Other specifically designed methods are mostly based on line detection [3-13]. The idea is, by finding and modelling thin lines explicitly, appropriate methods can always be worked out. There are two options: one is to perform the detection on the entire image to obtain some kind of parameterized representation of the lines; the other is to conduct analysis in a local neighbourhood. Most existing methods fall into the second class. Instead of explicit parameterization, these methods use templates for thin line detection. This approach works fine with images containing only simple line types (fixed width, straight, no crossing and intersection, etc.). But its robustness is greatly reduced with more complex graphical objects in real-world applications. **Figure 1** shows one example. Early works addressing this problem have the disadvantages such as handling only binary images, imposing restrictions to the types of lines and downsizing scales, low robustness, as well as often high computational complexity. Most of all, the very key 'thin structure/line' was not understood. In fact, 'thinness' (of lines or other structures) is a relative concept, i.e., relative to the desired downsizing scales. More specifically, taking a line as example, it should be considered 'thin' and needs special treatment only if its width is smaller than the sampling step, i.e., the downsizing scale.

Figure 1. A typical infographic image: stock chart.

To overcome the problems discussed above, we present a robust and efficient technique for infographic images. It is based on the key concept 'thin structure'. We organize the paper as follows. In Section 2 we present in detail our novel technique for infographic image downscaling. Test results are presented in Section 3 to show the effectiveness of the method. Finally we conclude in Section 4 and discuss the direction for future improvement.

2. THE NOVEL APPROACH

We adopt an iterative downsizing scheme. In fact, we learnt from previous works that direct arbitrary scale downsizing is complex and not robust, whereas it is much easier to devise simple and robust method for small-scale downsizing. Therefore, we devise a number of such algorithms, called 'core downsizing algorithms' (CDSAs), which are capable of robust downsizing for a scale of 2 or smaller. The CDSAs are applied in an iterative manner to achieve arbitrary downsizing scale. In this way, we obtain a robust downsizing method for arbitrary downsizing scales.

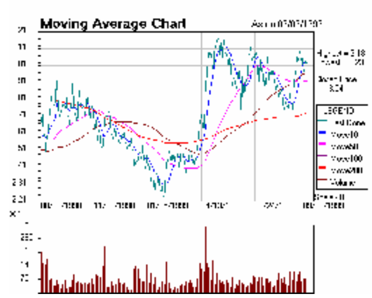
The CDSAs preserve 'thin structures' through preservation of 'thin pixels'. A thin pixel is defined to be 'thin' if it tends to be lost after downsizing for a given scale (<=2) but needs to be preserved to maintain structural information in the image. A thin structure is a group of pixels of which at least one is a thin pixel. 'Thin structure' is a concept more general than 'thin line'. Here restriction on the shape of image objects is totally removed. In addition to thin structure preservation, the resulting downsizing algorithms are very simple and fast, because to tell whether a pixel is thin is much simpler than detecting a (thin) line, as can be seen below.

The overall process is composed of two steps. The first step is to determine the background color. Then the core downsizing algorithms (CDSAs) are applied iteratively to the image to achieve the required downsizing scale.

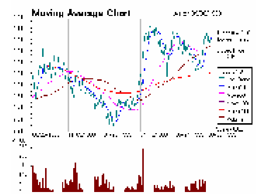
2.1 Background Color Determination

We assume that input infographic images' background is monochrome, which holds for most of the images. The background color is the most populous and the most widely distributed in the images. It is determined as follows. Firstly the color histogram is computed. The bounding box for each color is also computed at the same time. Then the color bins of the histogram are sorted in the ascending order. The one of which the bounding box is more than half of the image size is assigned as the background color of the image.

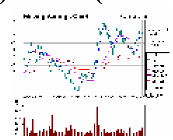
2.2 The Iterative Scheme



(a) s = 2.0 (260x211)



(b) s = 3.0 (173x141)



(c) s = 4.5 (115x94)

Figure 2. Chart downsizing using bilinear interpolation. Scales are 2.0 to 4.5.

The core downsizing algorithms fall into two categories, according on the actual downsizing scales (s_H, s_V), S_H being the desired scale in the horizontal direction and S_V the desired scale in the vertical direction. The first category consists of 3 CDSAs, performing downsizing by a scale of 2 (integer): CDSA2 in both horizontal and vertical directions, CDSA2H only in the horizontal direction, and CDSA2V only in the vertical direction. The second category also consists of 3 CDSAs, performing downsizing by a scale smaller than 2 but bigger than 1 (non-integer): CDSA1x in both horizontal and vertical directions, CDSA1xH only in the horizontal direction, and CDSA1xV only in the vertical direction.

The iterative scheme works as follows:

If S_H >= 2 and S_V >= 2, then call CDSA2 iteratively and update S_H and S_V (divided by 2), until S_H < 2 or S_V < 2;

If S_H >= 2 and S_V < 2, then call CDSA2H iteratively and update S_H, until S_H < 2;

If S_H < 2 and S_V >= 2, then call CDSA2V iteratively and update S_V, until S_V < 2;

If S_H = 1.x and S_V = 1, then call CDSA1xH;

If S_H = 1 and S_V = 1.x, then call CDSA1xV;

If S_H = 1.x and S_V = 1.x, then call CDSA1X.

2.3 The Core Downsizing Algorithms (CDSAs)

All the CDSAs start with pixel classification for the image to downsize for the given scales. Then the value of each pixel in the target image (downsized) is determined according to pixel configuration in its corresponding reference block (see later).

2.3.1 Pixel Classification

This step determines whether a foreground pixel is thin. The output is a map indicating the type of all the pixels of the input image. For each pixel to classify, a 3x3 block is defined, with the pixel in question at the center (Figure 3). Further, four sub-blocks, each being 2x2 and containing the center pixel, are defined. If the center pixel is a background pixel, it is not a thin pixel. If it is a foreground pixel, in each sub-block, a counter C is established that counts the total number of pixels that have the same color as the pixel to classify (center). The center pixel is labelled as 'thin pixel' if the following conditions are satisfied:

- all of the 4 counters (C's) must be less than 4;
- at least one of the C's is greater than 1.

The second condition is to prevent an isolated pixel from being classified as 'thin' and preserved over iteration. But in images containing numbers such as statistical charts, this condition should be removed because the points in numbers bear critical information and need to be retained.

All the background pixels are not thin by default.

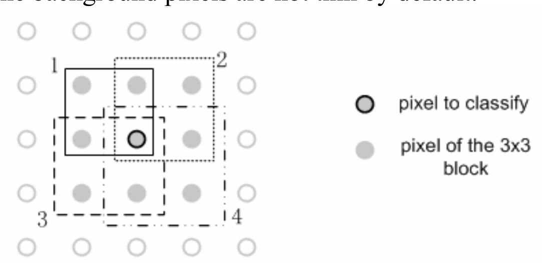


Figure 3 Pixel classification

2.3.2 Reference Block and Reference Pixels

Given the location (i, j) of an objective pixel in the target image and the downsizing scales s_H and s_V, the corresponding sampling position (x, y) is computed as follows:

$$\begin{cases} x = s_H * (i-1) + 1 \\ y = s_V * (j-1) + 1 \end{cases} \quad (1)$$

where we assume the origin of the image coordinates system is (1, 1). The sampling position (x, y) defines a so-called 'reference block' which contains at most 4 pixels. The top and/or left pixel is called 'primary reference pixel', written as ($i_{primary}$, $j_{primary}$); the rest of the pixels in a reference block are called 'secondary reference pixels', written as ($i_{secondary}$, $j_{secondary}$). ($i_{primary}$, $j_{primary}$) is computed as follows:

$$\begin{cases} i_{primary} = \lfloor x \rfloor \\ j_{primary} = \lfloor y \rfloor \end{cases} \quad (2)$$

where $\lfloor \bullet \rfloor$ is a floor operator. Therefore, if both s_H and s_V are integers, then so are i_{ref1} and j_{ref1}, and the reference pixel is the pixel at the sampling position itself. If one of s_H and s_V is not integer, the reference pixel is the nearest pixel at the top-left of the sampling position. The '*reference block*' is called so because the value of the objective pixel in the reduced image depends totally on it. In other words, objective pixel determination is entirely a *local* operation. Furthermore, during this process, a primary reference pixel has priority over secondary reference pixels.

The shape and size of the reference block depends on the downsizing scales. If $s_H > 1$ and $s_V > 1$, the reference block is a 2x2 block. If $s_H = 1$ and $s_V > 1$, the reference block is a 1x2 block (vertical). If $s_H > 1$ and $s_V = 1$, the reference block is a 2x1 block (horizontal). See **Figure 4**.

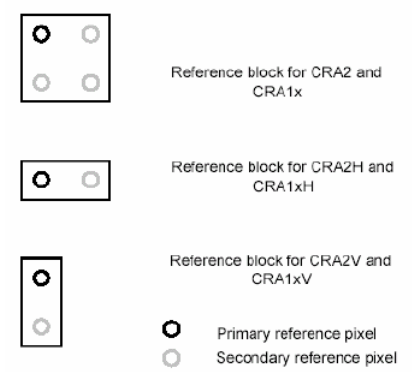


Figure 4. Reference blocks and reference pixels.

2.3.3 The Core Downsizing Algorithms with Integer Downsizing Scales

Because of paper limitation, we describe only CDSAs with integer downsizing scales, including CDSA2, CDSA2H and CDSA2V. First, the image to be downsized goes through the pixel classification, as described previously. Then comes the assignment of each of the objective pixels in the reduced image. For each pixel, its reference block is first determined. To assign a value to the objective pixel, two cases are considered separately:

- **The primary reference pixel is a thin pixel**. The objective pixel is assigned the value of the primary reference pixel. Here the simple nearest-neighbor rule is applied.
- **The reference pixel is not a thin pixel**. Logical OR operation is performed with respect to the secondary reference pixel(s) in the reference block. If all of them are non-thin pixels, then the objective pixel is assigned the value of the primary reference pixel. If one of them is a thin pixel, the objective pixel is assigned the value of that secondary reference pixel (the first one in some chosen order in case of 2x2 reference block).

3. EXAMPLES OF DOWNSIZING RESULTS

The new algorithm is applied to the stock chart image in Figure 1, with same scales as in **Figure 2**. The result is shown in **Figure 5**. It can be seen that fine structures in the images are well preserved, at all the scales, in sheer contrast with conventional methods. As can be seen, line structures are well preserved over different scales. We notice that while curves and lines are well preserved, types survive less well, especially with a larger scale. This is rather expected. In other words, to solve the problem properly, OCR techniques should be called up here. Also, when the scale becomes bigger, lines start to aggregate and the whole chart may become unreadable. In fact, this shows that there is a limit on the applicable scale. It is determined inherently by the property of the original image, i.e., the local density of thin pixels. The algorithms have been applied to a large number of other infographic images and the results are all very satisfactory.

4. CONCLUDING REMARKS

In this paper, we have presented a robust downscaling algorithm for infographic images. It is specially suited for large scale downsizing for mobile devices, with arbitrary scales in both dimensions. An inherent feature of the algorithm is that it can handle color images, instead of limiting itself to binary images, as opposed to other techniques. The algorithm is very simple and fast, very suited to online processing in multimedia repurposing proxy. However, types and dense thin structures are beyond the present method. For future work, we will develop techniques to separate types from the graphic components and using OCR to recognize types. Further, we will provide a measure which will tell the largest scale applicable by the current technique.

5. REFERENCES

[1] C. Christopoulos, T. Ebrahimi, V.V. Vinod, J.R. Smith, R. Mohan, and C.S. Li. MPEG-7 Application: Universal Access Through Content Repurposing and Media Conversion, ISO/IEC JTC1/SC29/WG11 MPEG99/M4433, Seoul, March 1999.

[2] J. Smith; R. Mohan; C. Li, "Content-based Transcoding of Images in the Internet," *ICIP,* 1998.

[3] Wakabayashi *et al.* Binary image reduction method US Patent 5,054,099 (October 1, 1991)

[4] Endoh *et al.* Method and apparatus for producing binary picture with detection and retention of plural binary picture blocks having a thin line pattern including an oblique line. US Patent 5,121,222 (June 9, 1992).

[5] Hirabayashi *et al.* Image reducing apparatus. US Patents 5,138,672 (August 11, 1992).

[6] Hirabayashi *et al.* Image reducing apparatus. US Patents 5,309,524. May 3, 1994.

[7] Yoshida et al. Image reduction system. US Patent 5,159,468 (October 27, 1992)

[8] Ferracini *et al.* System and method for scaling a digital image. US Patent 5,335,295 (August 2, 1994)

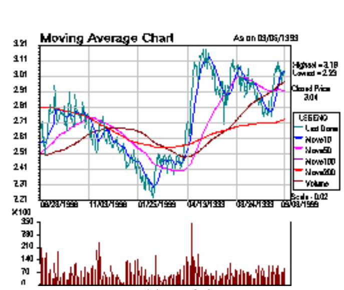
[9] Kato *et al.* Pixel density converting apparatus. US Patent 5,351,137. (September 27, 1994).

[10] Hino, et al. Method of scaling an image capable of line width preservation. US Patent 5,539,534 (July 23, 1996)

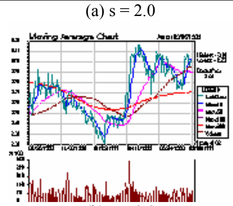
[11] Hirabayashi *et al.* Image reduction with fine-line protection. US Patent 5,680,225 (October 21, 1997)

[12] Iguchi et al. Apparatus and method for image reduction. US Patent 5,901,254 (May 4, 1999)

[13] Akagi *et al.* Image processing device and method for detecting and removing fine lines in an image pattern. US Patent 5,946,416 (August 31, 1999)



(a) s = 2.0


(b) s = 3.0

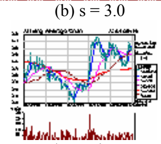

(c s = 4.5

Figure 5. Chart downscaling using our method.