PHOTO-STENCILS

Jeff Orchard

University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Ontario, Canada N2L 3G1
jorchard@uwaterloo.ca

ABSTRACT

Similar in nature to an image mosaic, a photo-stencil is a rendering of a binary target image by an arrangement of smaller binary stencil images. This paper presents a method for efficiently creating photo-stencil images, making use of the Fast Fourier Transform to speed up costly computations. The resulting photo-stencils are a new and interesting form of non-photorealistic rendering.

Index Terms— mosaics, Fourier, matching, registration

1. INTRODUCTION

Many forms of non-photorealistic rendering are artistic in nature. For example, image mosaics [1, 2, 3, 4, 5, 6] endeavour to reconstruct a target image by assembling a collection of intelligently chosen, and smaller, source images.

A different take on the mosaicking theme is object packing, such as the NPR packing of Dalal [7] and jigsaw image mosaics [8]. The objective of object packing is to optimally fit together a set of image objects to form a composite object. For example, one could create the shape of a slice of bread by cobbling together smaller images of food.

A photo-stencil is a rendering of a binary target image by placing – indeed, packing – small, binary stencil images. In this paper, we outline an efficient method for computing photo-stencil images using the Fast Fourier Transform.

2. METHOD

Suppose you are given a binary target image, \( g \), and a binary stencil image, \( f \). The fundamental goal of the stencilation process is to place \( f \) onto \( g \) such that the white part of \( f \) rests completely inside a white region in \( g \), as shown in Fig. 1. If we suppose the foreground (white) pixels have an intensity of 1, and the background (black) pixels have an intensity of 0, then we can achieve our white-on-white overlay by finding the offset \((a, b)\) to apply to \( f \) such that the cross-correlation,

\[
C(a, b; f, g) = \frac{\sum_{ij} f_{i-a,j-b} g_{ij}}{\left(\sum_{ij} f_{ij}^2 \sum_{ij} g_{ij}^2\right)^{1/2}},
\]

is maximized. If there is a white patch in \( g \) that completely encompasses the white part of \( f \), then there will be a shift \((a, b)\) such that

\[
C(a, b; f, g) = 1.
\]

This is the maximum value that \( C \) can attain, and we denote \((a, b)\) a legal position of the stencil. If there is no such region of \( g \), then \( C(a, b; f, g) \) will be strictly less than 1 for all shifts \((a, b)\).

Evaluating (1) by brute-force for every possible integer offset is computationally expensive. Notice, however, that the numerator can be expressed as a convolution, and therefore computed using the Fast Fourier Transform (FFT) [9].

\[
\sum_{ij} f_{i-a,j-b} g_{ij} = \mathcal{F}^{-1}\left\{\mathcal{F}\{f\}\mathcal{F}\{g\}\right\}_{ab},
\]

where the function \( \mathcal{F} \) represents the FFT, and the notation \( \mathcal{F}\{f\} \) represents the complex conjugate of \( \mathcal{F}\{f\} \) (see [10] for a detailed explanation). Of course, to perform this computation, we need the FFT of our stencil image \( f \) and our target image \( g \).
After appropriately shifting the stencil image $f$, it is subtracted from the target image $g$. This operation can equivalently be done in the frequency domain.

However, once we have those, we can compute the cross-correlation for all possible integer shifts at the cost of a simple inverse FFT.

Here is how the method proceeds. Once we have found a place to put one stencil, we choose another stencil and find a place for it, being careful not to allow it to overlap with any of the previously-placed stencils. We continue this process until we cannot find a place suitable to fit any of the stencils.

When a stencil is placed, its “footprint” is removed from the target. That is, the region where the white part of $f$ landed in $g$ becomes black since that part of the target image has now been accounted for. In practice, we simply subtract the translated version of $f$ from $g$ (shown in Fig. 2); this works because we have insisted that the white part of $f$ land only on a white part of $g$. If that were not the case, simple subtraction would result in negative binary pixel values.

Moreover, this subtraction can be done in the frequency domain, since the Fourier transform is a linear operator. The advantage of doing the subtraction in the frequency domain is that it avoids the FFT and inverse-FFT pair of transforms required to get the spatial-domain representation of $g$. Hence, we only have to take the FFT of $g$ once at the beginning of the process.

As each stencil is chosen and placed, the remaining portion of $g$ – what is left after subtracting all the previously-placed stencils – is used in later steps. Hence, the remaining portion of $g$ records the binary content of the target image that is still available for stenciling.

As one would expect, larger stencils have fewer legal locations than smaller stencils. For this reason, our method has a parameter that allows the user to bias the order of stencil selection: preferring large stencils first, or small stencils first.

### 3. RESULTS

Figure 3 shows two photo-stencils of a target image. The stencil set consists of three different sizes of lower-case letters from the English alphabet. The stencilation in (b) was created with a greater initial preference for larger stencils than the stencilation in (c), resulting in (b) having more large-scale stencils than (c). These photo-stencils were generated on a 2.1GHz Intel Dual-core MacBook with 2GB of RAM. Running time was approximately 180 seconds.

Figure 4 shows a photo-stencil of an image of Albert Einstein. The stencil set consists of three different sizes of 41 mathematical typographical symbols.

### 4. CONCLUSIONS

The non-photorealistic photo-stencil rendering method is efficient and generates interesting images. The method can be implemented in the frequency domain, including image comparison (cross-correlation), shifting, and subtraction.

Possible extensions to this work include optimal rotation of stencils, as well as a graylevel (non-binary) version.
Fig. 3. Sample photo-stencils, showing two different stencil-size selection strategies.

Fig. 4. Einstein target and stencilation.
5. REFERENCES


