Cut-Out Image Mosaics

Jeff Orchard Craig S. Kaplan*
David R. Cheriton School of Computer Science
University of Waterloo

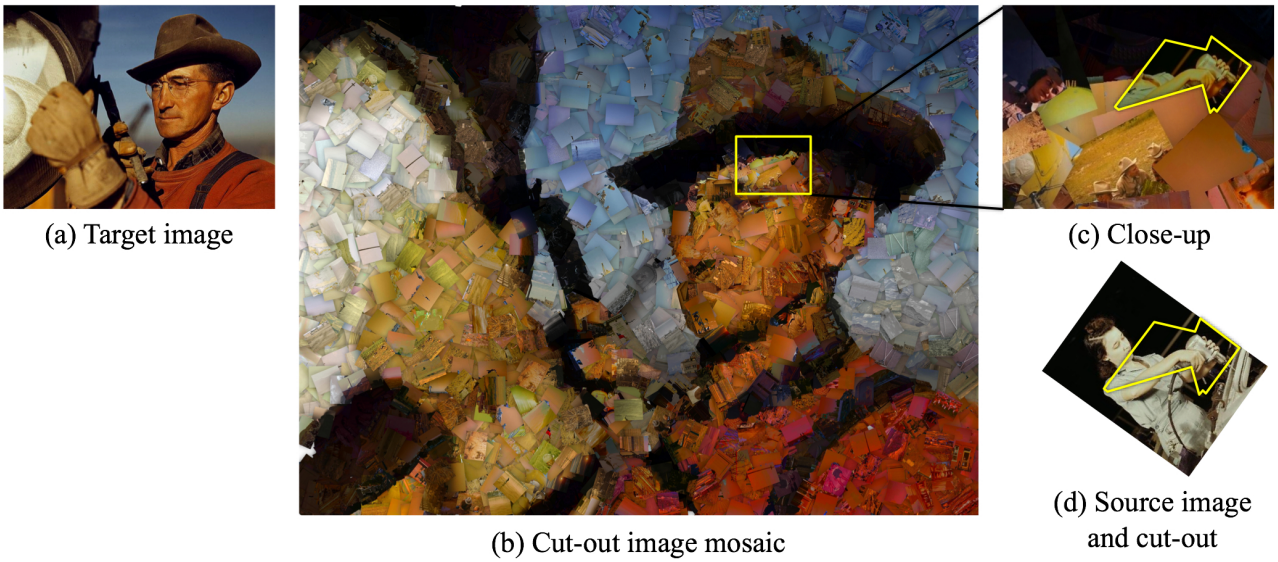


Figure 1: *Cut-out image mosaic involving various target tile shapes, colour correction, and the ability to select sub-images from images in the source database. The target and source images were taken from the Library of Congress collection on flickr*$^{\text{TM}}$.

Abstract

An image mosaic is a rendering of a large target image by arranging a collection of small source images, often in an array, each chosen specifically to fit a particular block of the target image. Most mosaicking methods are simplistic in the sense that they break the target image into regular tiles (e.g., squares or hexagons) and take extreme shortcuts when evaluating the similarity between target tiles and source images. In this paper, we propose an efficient method to obtain higher quality mosaics that incorporate a number of process improvements. The Fast Fourier Transform (FFT) is used to compute a more fine-grained image similarity metric, allowing for optimal colour correction and arbitrarily shaped target tiles. In addition, the framework can find the optimal sub-image within a source image, further improving the quality of the matching. The similarity scores generated by these high-order cost computations are fed into a matching algorithm to find the globally-optimal assignment of source images to target tiles. Experiments show that each improvement, by itself, yields a more accurate mosaic. Combined, the innovations produce very high quality image mosaics, even with only a few hundred source images.

Keywords: image mosaic, registration, non-photorealistic, least-squares, Fourier transform, assignment problem.

1 Introduction

The first image mosaics were large murals formed by placing thousands of coloured tiles [Battiato et al. 2006]. Inspired by these works of art, today the term "image mosaic" refers to a stylized representation of a large image, the "target" image, formed by piecing together a collection of carefully chosen smaller images called "source" images. The target image is subdivided into small pieces, called "target tiles" (typically rectangles), and each tile is filled with a source image that approximates the tile's contents.

Image mosaics communicate at two disparate scales. These two scales act as a symbolic divide, so that the target image is conceptually set apart from the contents of the tiles that comprise it. This dichotomy provides a rich environment for combining images that either suggest the same message from two different perspectives, or supply contrasting viewpoints. For example, an image of a car might be made out of pictures of the employees that manufactured it, or pictures of bicycles. Image mosaics are a powerful medium for conveying such split-level messages.

Every image mosaic must strike a balance between the opposing

*e-mail: {jorchard,cskaplan}@uwaterloo.ca

NPAR 2008, Annecy, France, June 09–11, 2008.
© 2008 ACM 978-1-60558-150-7/08/0006 $5.00

goals of *accuracy* and *discernability*. *Accuracy* seeks to reproduce the target image as closely as possible; *discernability* seeks to ensure that each of the source tiles is legible. It is easy to trade off between these two goals by controlling tile sizes, but more interesting to attempt to achieve accuracy and discernability simultaneously. In this paper we introduce a novel method for assigning source images to tiles with much higher accuracy than previous approaches. This extra margin of quality allows us to achieve overall accuracy comparable to previous techniques but with larger individual tiles, thereby increasing discernability.

One way to improve the accuracy of a mosaic is increase the number of images in the source database. As the database grows, so does the probability of finding a close match for a target tile. However, more source images require more processing time to create the mosaic. Also, not every mosaic should be produced using thousands or tens of thousands of images. Occasionally, limited resources or thematic constraints may restrict us to a few hundred source images.

Previous work has sought methods to speed up the matching process to support efficient search over large image databases. Typically, these approaches summarize a source image as a "signature" consisting of a few numbers. Matching can then be done at the signature level. The most common technique is to partition the source image into a regular grid and average the contents of each grid cell down to a single colour [Di Blasi et al. 2006; Silvers 2001; Tran 1999; Zhang 2002]. Di Blasi et al. [2006] partition each image into a 3×3 grid and compute the average red, green and blue values within each grid cell, yielding a signature of 27 numbers. They arrange their source image database into an antipole tree data structure, reducing the number of comparisons needed to find a close match. Zhang [2002] partitions source images into four cells. Each cell is represented by a coarse histogram, and the histograms are reduced in turn to binary signatures. Finkelstein and Range [1998] use the wavelet signature approach of Jacobs et al. [1995] to accelerate their mosaic method. The wavelet signature records the largest coefficients of the wavelet transform of an image.

These aggregate signatures place a bound on the smallest image features that can be used to compare source images to target tiles. For example, the 3×3 signatures of Di Blasi et al. mean that their algorithm will be blind to any features smaller than one ninth of a source image. For a mosaic consisting of an $M \times N$ grid of tiles, accuracy can effectively be no better than that of a $3M \times 3N$ image.

These algorithms could clearly benefit from a finer-grained comparison metric, which could find correspondences between features at any size. Still, without a huge database, the chances of finding close matches is lower. The odds are greatly improved if we allow the possibility of matching a target tile to a *portion* (or cut-out) of a source image, as in Figure 1(d). Given a user-defined target tile, we would consider all possible shifts (translations) of the target tile's footprint within the source image. This change would logically multiply our source database by a large constant factor (the number of possible shifts for each image), yielding more opportunities for closer matches with a relatively modest database. Of course, we must deal with the seeming increase in computational complexity. In Section 2, we present an FFT-based approach that can feasibly compute fine-grained matching quality for all shifts.

A complementary strategy to improve mosaic accuracy is to allow for colour shifting by applying a global scaling factor to the colour components of an image. Doing so can alter the colour composition of an image dramatically, while still depicting geometric content [Di Blasi et al. 2005]. Finkelstein and Range [1998] also applied a colour shift, though their shift was potentially different at each target pixel. This pixel-wise method can introduce phantom features in the source images, even rendering them unrecognizable; in other words, it increases accuracy at the expense of discernability. We show in Section 2 how our technique can be extended to compute an optimal colour adjustment for any source image assignment.

Assuming that we can feasibly compute the matching cost of every shift of every source image against every target tile, we are still left with the question of selecting the particular assignment of source images to tiles that will produce a pleasing overall result with high quality. The natural choice of a greedy algorithm is not always best. In Section 3 we discuss our assignment algorithm, which makes a global choice based on all matching costs.

We describe our implementation in Section 4. In Section 5 we present results that isolate the effects of these various improvements, and show that our technique helps to improve accuracy.

2 Evaluating matching cost

Let us assume that we are given a target image $\vec{T}(i,j)$ from which we wish to construct a mosaic. (We use uppercase italic letters to denote images, and include an arrow over an image name when pixels in that image are vectors as opposed to scalars.) Fix a single tile shape in the mosaic, and represent it via a characteristic function $W(i,j)$ that is 1 inside the tile and 0 everywhere else. That is, the non-zero part of W indicates the portal through which the source image and target image are compared. Fix a source image $\vec{S}(i,j)$. We now consider the problem of computing $C(a,b)$, the image dissimilarity over all tile pixels between the target image and the source image shifted by (a,b). A reliable metric is the weighted sum of squared differences (SSD) cost function

$$C(a,b) = \sum_{i,j} \|\vec{S}(i-a, j-b) - \vec{T}(i,j)\|^2 W(i,j), \quad (1)$$

where $\|\cdot\|$ represents the Euclidean distance between two colours and the summation is taken simultaneously over all i and j.

We may also wish to allow for colour shifting by scaling the different colour components in $\vec{S}$ to best match the target $\vec{T}$. We let $\mathbf{D}$ be a diagonal matrix with a scaling factor for each colour component of $\vec{S}$, and rewrite the cost function as

$$C(a,b,\mathbf{D}) = \sum_{i,j} \|\mathbf{D}\vec{S}(i-a, j-b) - \vec{T}(i,j)\|^2 W(i,j). \quad (2)$$

Given a source image $\vec{S}$, it is infeasible to find values of (a,b) and $\mathbf{D}$ that minimize $C(a,b,\mathbf{D})$ by direct computation of Equation 2. If $\vec{T}$ and $\vec{S}$ are roughly $N \times N$ pixels in size, computing the cost function for a single shift (a,b) requires $\mathcal{O}(N^2)$ time. However, there are also roughly N^2 possible shifts, meaning that it would take $\mathcal{O}(N^4)$ time to find the optimal (a,b), and even then we would need to compute $\mathbf{D}$. Bear in mind also that we will eventually want to compute the matching costs for all possible source-image/target-tile pairs. If there are P source images and M tiles, we would expect a brute force algorithm to take $\mathcal{O}(N^4PM)$ time in total.

However, we can greatly speed up this computation by considering the problem in the frequency domain. To see how, we first separate Equation 2 into expressions for each of three colour components k, obtaining

$$C_k(a,b,D_k) = \sum_{i,j} \left[D_k S_k(i-a, j-b) - T_k(i,j)\right]^2 W(i,j)\ , \quad (3)$$

where $k \in \{1, 2, 3\}$, S_k and T_k are scalar-valued images, and D_k is a scalar. Note that the total cost is now given by

$$C(a, b, \mathbf{D}) = C_1(a, b, D_1) + C_2(a, b, D_2) + C_3(a, b, D_3). \quad (4)$$

For each value of k, we now expand Equation 3, giving us

$$\begin{aligned} C_k(a, b, D_k) &= \sum_{i,j} D_k^2 S_k^2(i - a, j - b)W(i, j) \\ &-2 \sum_{i,j} D_k S_k(i - a, j - b)T_k(i, j)W(i, j) \\ &+ \sum_{i,j} T_k^2(i, j)W(i, j) \, . \end{aligned} \quad (5)$$

The last term in Equation 5 is easy to compute because it does not depend on (a, b). The first two terms are correlation terms, and can be posed as convolutions simply by reflecting S_k. Letting $\overline{S}_k(i, j)$ equal $S_k(-i, -j)$, we reformulate Equation 5 as

$$\begin{aligned} C_k(a, b, D_k) &= D_k^2 \left(\overline{S}_k^2 * W\right)(a, b) \\ &-2 \; D_k \left(\overline{S}_k * (T_k W)\right)(a, b) \\ &+ \sum_{i,j} T_k^2(i, j)W(i, j). \end{aligned} \quad (6)$$

Here we use $*$ to denote the convolution operator and we have dropped the explicit dependence on (i, j) in the convolution terms for brevity.

Rephrasing the equation in this way allows us to transfer some of the work into the frequency domain. Let $\mathcal{F}$ represent the discrete Fourier transform operator. Then the convolution terms in Equation 6 can be written

$$\begin{aligned} \overline{S}_k^2 * W &= \mathcal{F}^{-1}\left\{\mathcal{F}\{\overline{S}_k^2\}\, \mathcal{F}\{W\}\right\} \\ \overline{S}_k * (T_k W) &= \mathcal{F}^{-1}\left\{\mathcal{F}\{\overline{S}_k\}\, \mathcal{F}\{T_k W\}\right\} \, . \end{aligned} \quad (7)$$

Using the Fast Fourier Transform [Cooley and Tukey 1965], we can compute $\mathcal{F}$ and $\mathcal{F}^{-1}$ on $N \times N$ images in $\mathcal{O}(N^2 \log N)$ time. This computation dominates the total effort of computing the convolution, since the Fourier coefficients can be multiplied pointwise in only $\mathcal{O}(N^2)$ time. The resulting algorithm is significantly more efficient than the brute force $\mathcal{O}(N^4)$ approach (though previous experiments suggest that it is memory intensive [Orchard 2005]). Note that we need not compute $\overline{S}_k$ (a reflected version of S_k) explicitly before computing its FFT; we can equivalently take the complex conjugate of $\mathcal{F}\{S_k\}$.

The FFT has been used in similar image registration problems in the past [Fitch et al. 2005; Friston et al. 1995; Kilthau et al. 2002; Orchard 2005]; once the cost function is evaluated for all shifts, simply scanning to find the minimum element reveals the optimal translation. Dalal et al. used it in an NPR packing context to compute the optimal shift of a geometric primitive within a region [Dalal et al. 2006]. In their case, the two images being compared were the characteristic function of the primitive and the distance transform of the region.

These convolution terms, once evaluated, form the coefficients for a cost function that is simply a quadratic function of D_1, D_2 and D_3. Since these variables are decoupled in the cost function, their optimal values can be found by taking the derivatives of Equation 4 with respect to each s_k, setting them to zero, and solving for D_k. We obtain the optimal D_k value for a given shift (a, b),

$$D_k = \frac{\overline{S}_k^2 * W}{\overline{S}_k * (T_k W)} \, . \quad (8)$$

Note that the numerator and denominator of Equation 8 are both functions of (a, b), and the division is intended to be element-wise, thus yielding a value of D_k for each possible offset (a, b). Using these optimal D-values, we evaluate our cost function $C(a, b)$ for each (a, b). Finding the global minimum is simply a matter of scanning through the N^2 costs to find the smallest. The location of the lowest cost indicates the optimal shift (and corresponding values for D_1, D_2 and D_3).

Similarly, we can compute the optimal shift for other colour correction schemes. Suppose, for example, that we only want to allow D_1 to vary, while holding D_2 and D_3 fixed. We can compute the cost for all possible shifts under this colour correction scheme simply by setting D_2 and D_3 to 1, and recomputing Equation 4. All the necessary convolution terms were already evaluated for the higher-order colour correction, so the amount of additional computation required for this alternate scheme is small.

3 Choosing an assignment

Once individual costs have been computed for all pairs of a target tile and a source image, we must use these costs to construct an *assignment*, a choice of image for each tile.

A simple approach with very high accuracy is a greedy algorithm with replacement. We consider the target tiles independently. For each one, we choose the source image with the lowest overall cost. Most previous mosaicking techniques use this variety of assignment. However, while its accuracy may in some sense be considered optimal, it is prone to distracting artifacts. Areas of roughly constant colour will tend to be paved with copies of the same source image, creating a repeating pattern that reduces the aesthetic appeal of the final mosaic. Some previous approaches have mitigated this problem by preventing two copies of a source image to occupy adjacent tiles [Di Blasi et al. 2006].

We might instead consider a greedy algorithm *without* replacement, in which each source image can be used only once (or any prescribed number of times, if we include multiple copies). This approach works, but is unlikely to find an optimal assignment; the quality of the result depends crucially on the incidental order in which the target tiles are considered.

Given that we compute the costs of all pairs of target tiles and source images, we have all the information we need to achieve a globally optimal assignment without replacement. Observe that the costs describe a complete bipartite graph whose vertices represent the target tiles and source images. Each edge records the cost of using a particular image for a tile. Finding the globally optimal assignment is then equivalent to computing the minimum weight matching of this graph. Efficient algorithms exist for this so-called *assignment problem*. We use a Matlab-based implementation by Markus Bühren [2007] of the Kuhn-Munkres algorithm [Munkres 1957].

4 Implementation

In this section, we outline how the above optimization framework is built into an image mosaicking method. Our general exhaustive search strategy is simply to consider every target-tile/source-image pair, and find its minimum SSD solution (optimal shift and colour correction). Given M target tiles and P source images of size $N \times N$, the exhaustive evaluation of all such comparisons takes $\mathcal{O}\left(MPN^2 \log N\right)$ operations. Once all the comparisons are made, we know the full ranking of all source images for each target tile, and can pick and choose them as we see fit.

The above strategy to optimize the SSD was implemented in Matlab (MathWorks Inc., Natick, Massachusetts). In addition to the target and source images, the user supplies an image of the target tiles. Each tile is drawn in a unique colour; pixels of that colour will be extracted to produce the W images below.

Suppose $\vec{S}$ is a source image and $\vec{T}$ is a piece of the target image (the same size as $\vec{S}$) that contains a target tile. We define the the alpha image W to be the same size as $\vec{S}$ and $\vec{T}$ with a value of one for pixels inside the target tile, and a value of zero outside (the tiles provided by the user must therefore be small enough to fit inside the source images). The following pseudocode efficiently computes the optimal offset and $\mathbf{D}$-values.

1. Compute $\mathcal{F}\{\overline{S}_k\}$, $\mathcal{F}\{\overline{S}_k^2\}$, $\mathcal{F}\{T_k W\}$, $\mathcal{F}\{W\}$, and $\sum_{ij} T_k^2 W$ for $k = 1, 2, 3$.
2. Compute $Q_k = \mathcal{F}^{-1}\{\mathcal{F}\{\overline{S}_k^2\}\mathcal{F}\{W\}\}$ and $R_k = \mathcal{F}^{-1}\{\mathcal{F}\{\overline{S}_k\}\mathcal{F}\{T_k W\}\}$ for $k = 1, 2, 3$. Note that Q_k and R_k are both images, with a value for each possible offset (a, b).
3. Compute $D_k = \frac{R_k}{Q_k}$ for $k = 1, 2, 3$, where the division is on an element-by-element basis.
4. Compute $C = \sum_{k=1}^{3} D_k^2 Q_k - 2 D_k R_k + \sum_{ij} T_k^2 W$.
5. Scan C for its minimum element. Record the coordinates of the minimum element as well as its corresponding D_k-values. These give the $(a, b, \mathbf{D})$-values that optimize the SSD cost function in Equation 2.

We consider a number of performance factors in building our source image database. The Fourier Transforms of the source images are used repeatedly, once for each target tile. Hence, we compute the source-image Fourier Transforms once, store them, and re-use them.

All the source images we use in our experiments have a 4:3 aspect ratio. As a pre-processing step, the source images are scaled down to either 60×80 or 80×60. In terms of speed, it helps if many of the source images are the same size because the target tile image, alpha image and source image all have to be the same size to use the FFT method outlined above. Having only two different source image sizes means we can compute and store the Fourier Transforms for our target tiles ($\mathcal{F}\{T_k W\}$ and $\mathcal{F}\{W\}$), and reuse them. If a different sized source image is encountered, the appropriate target-tile Fourier Transforms can be computed on the fly, but with a performance penalty.

Once each target tile is assigned a source image, we use higher-resolution versions of both the target image and source images to build the actual mosaic. The optimal offsets and colour correction parameters are all re-computed for this higher-resolution rendering. Fine source image details are more discernible in the final high resolution mosaic.

In our experiments, all image manipulation is performed in YIQ colour space rather than RGB (we expect similar results using HSB). While the theory outlined in section 2 allows for three degrees of freedom (DOF) when performing colour correction, our experiments use schemes with only two, one or zero degrees of freedom. In the 2-DOF scheme, we use one parameter to scale Y (luminance), and the other parameter to scale both I and Q (hue). In the 1-DOF scheme, only Y is scaled. Thus, for each target-tile/source-image pair, three possible colour correction schemes are evaluated and optimized. For M target tiles and P source images, we compute a total of $3MP$ matches. Clearly, the higher-DOF matching costs will be less than or equal to the lower-DOF costs. However, the optimal parameters are recorded for all three colour correction schemes and stored for later use.

5 Experiments

We present three experiments to illustrate the benefits of each of the optimizations incorporated into our mosaicking method.

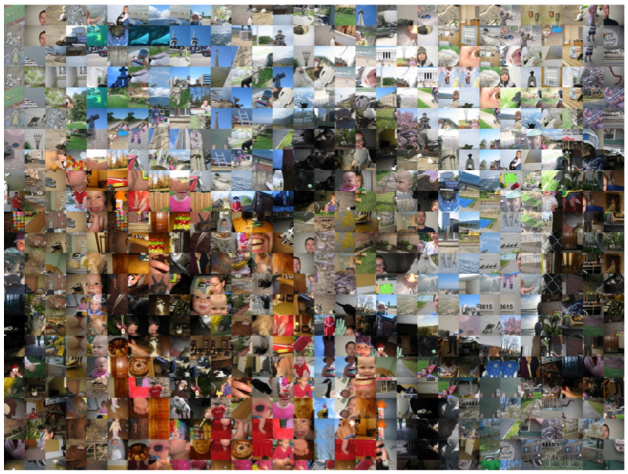
In the first experiment, our goal is to demonstrate that the selection of an optimal sub-image allows for more accurate matching of the target tiles. For comparison, we ran both our method and the method of Di Blasi et al. [2006]. The Di Blasi method includes source images in their entirety, rescaling as necessary to fit the aspect ratio of the target tile. To compare our method directly to that of Di Blasi, we used square tiles of size 40×40 pixels. Also, neither method performed any colour correction.

Figure 2 illustrates the advantage of selecting optimal sub-images. Both methods use the same source database of 400 images, allowing each image to appear at most twice. The mosaic produced using our method differs from the target image by a root mean squared error (RMSE) of 50.1, while the mosaic by Di Blasi differs by 67.1. The figure also shows a sample target tile, and the source image selected by each of the two methods. The Di Blasi method is incapable of matching the fine-grained structure of the target tile. In our proposed method, the location of the source sub-image can be fine-tuned to match the target tile. In the case of Figure 2(f), the outline of the baby's face is an excellent match for the contour of the circle on the shirt.

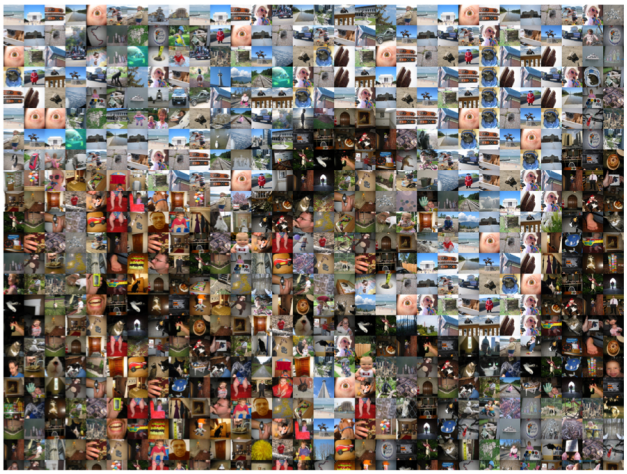
The second experiment is designed to demonstrate the effect of colour correction. Figure 3 shows three mosaics produced by our method, each mosaic using successively higher-order colour correction. The mosaics were created using a source database of 400 images, where each image could be used up to three times, each time with different colour correction parameters (where applicable). The tile shapes are puzzle pieces, randomly generated using splines so that no two puzzle pieces are exactly the same. The beveled edges of the puzzle pieces are added in a post-processing step.

With no colour correction (Fig. 3(a)), the RMSE is 46.3. Allowing only luminance correction yields an RMSE of 30.3. If we allow both luminance and hue to be adjusted, the RMSE drops to 24.9. The image inset in Fig. 3(b) shows tiles (puzzle pieces) filled with source sub-images that have been darkened drastically to match a dark target tile. The inset of (c) shows a tile containing sky and clouds. The optimization process chose a negative scaling value for the hue correction, turning the blue sky to orange.

In the third experiment, we study the effect of solving the global assignment problem instead of using a greedy algorithm to assign source images to target tiles. Figures 1 and 4 show mosaics that we have dubbed "photo-heaps", similar in appearance to joiners [Zelnik-Manor and Perona 2007], or coverpop [Bumgardner 2008]. To create a photo-heap mosaic, we cover most of the image canvas with randomly-placed rectangles of size 50×30. The unoccluded portion of each rectangle is then used as a target tile; there are 1849 target tiles in each mosaic in Figure 4, some of which are almost entirely occluded. To compare the target tiles to our source images, we rotate each rectangle along with its target-image contents to align it with the coordinate axes, considering both portrait and landscape orientations. Hence, each target tile is matched to each source image twice – the rectangles fit inside the 80×60 and 60×80 source images in either orientation. Once a source image is assigned to a target tile, the selected source image is then rotated back so that it looks like a rotated photograph in the result. We also add a subtle drop-shadow effect to enhance the layered appearance.



(a) Proposed method without colour correction (RMSE = 50.1)

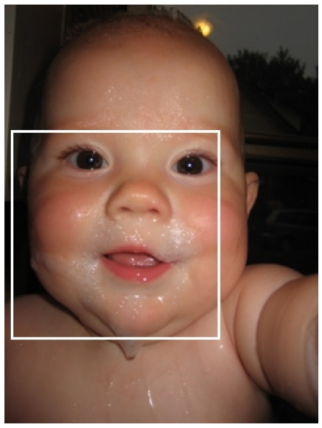


(b) Method of Di Blasi (RMSE = 67.1)

(c) Target image

(d) Target tile

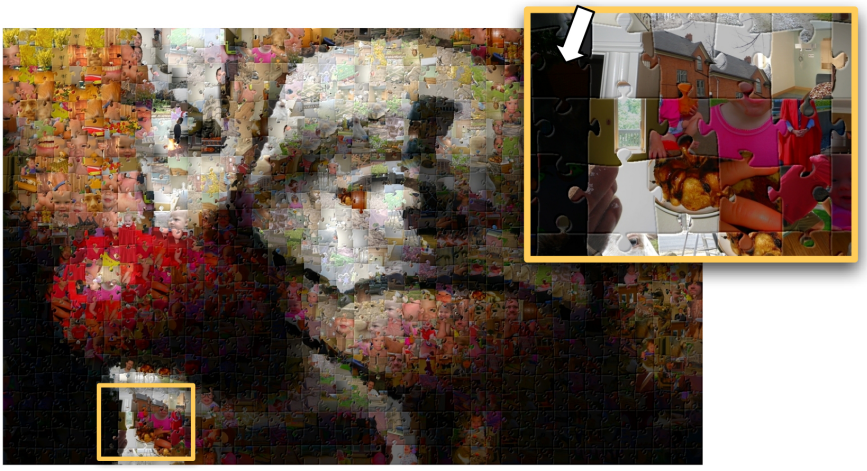
(e) Di Blasi match



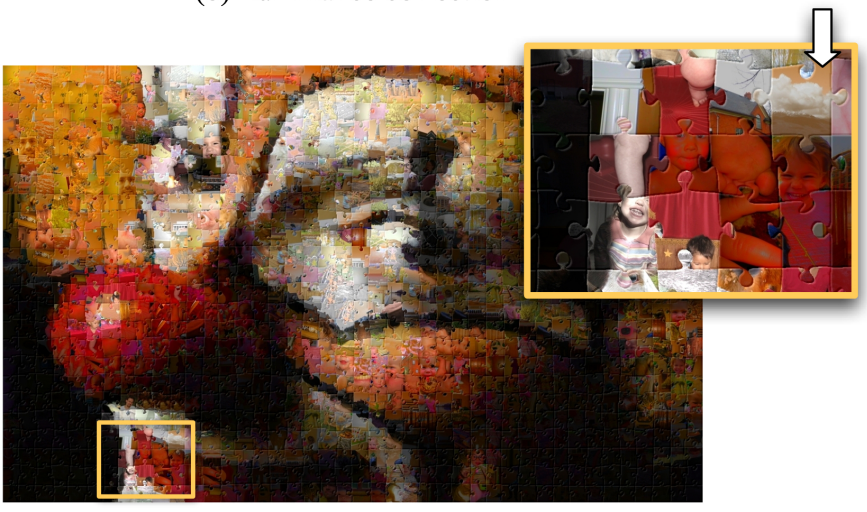
(f) Match found by proposed method

Figure 2: *Effect of choosing an optimal source sub-image. Our mosaic is shown in (a), generated using a grid of squares. Each source image was used at most twice. A comparable mosaic of Di Blasi et al. is shown in (b), using the same grid of squares. Both mosaics used the same source image database of 400 images without colour correction. The target image is shown in (c). The area indicated by the arrow is shown in a close-up in (d). The source image used by Di Blasi is shown in (e), while the source sub-image used by our method is shown outlined in (f). The RMSE for (a) is 50.1, while the RMSE for (b) is 67.1.*

(a) No colour correction



(b) Luminance correction



(c) Luminance and hue correction

Figure 3: *Effect of colour correction. In (a), no colour correction was done. In (b), the optimal luminance scaling was used for the reconstruction. The arrow shows a mosaic tile that was severely darkened. In (c), the optimal luminance and hue scaling were used. The arrow shows a mosaic tile that depicts blue sky as orange, indicating that the optimal hue factor is a negative number. In each mosaic, each of the 400 source images could be used up to three times. In (b) and (c), each of the three occurrences of a source image could have different colour-correction parameters.*

(a) Greedy Algorithm (RMSE=93.3)

(b) Global Matching Algorithm (RMSE=63.1)

Figure 4: *Effect of greedy versus global assignment. The mosaic in (a) was created by assigning source images to target tiles in order of increasing cost (greedy algorithm), resulting in an RMSE of 93.3. The mosaic in (b) was created by solving the global assignment problem between source images and target tiles, resulting in an RMSE of 63.1. The source database contained 422 cat images, where each image could be used up to six times (three types of colour correction with two orientations). The insets show a close-up of the outlined portion.*

In the greedy algorithm, all computed costs are sorted in increasing order. Starting from the lowest cost, source images are assigned to target tiles. The globally optimal matching solution is computed by considering all target-tile/source-image pairs simultaneously, choosing the pairings to minimize the SSD cost over the entire mosaic. In Figure 4, each source image can be used up to six times since there are three colour-correction schemes (0, 1 and 2 degrees of freedom), and two orientations (portrait and landscape).

The low quality of the greedy mosaic in Figure 4(a) stems from the fact that many of the target tiles are quite small (because of occlusion). These small tiles yield small SSD costs not only because they have fewer pixels, but also because their size makes it easier to find close matches for them. Hence, the greedy algorithm fills the small tiles first, losing the opportunity to find good matches for the larger, more visually important tiles.

The globally optimal matching method takes advantage of the fact that many source images would fit a small tile fairly well. Finding a good match for a larger tile is more difficult. Hence, the global solution puts more weight on the larger tiles, thereby spreading the error to find the best overall assignment to minimize the mosaic RMSE.

Speed tests were performed using our Matlab-based implementation on an Apple MacBook with a 2.3GHz Intel DuoCore processor and 2GB of RAM. Generating the low-resolution version of the mosaic in Figure 4 required five seconds to pre-process the 422 source images, about 3.4 hours to evaluate all the target-tile/source-image matching costs, and 42 minutes to solve the matching problem.

The mosaic in Figure 5 is the result of a modified version of the same mosaicking framework. Instead of searching for the best translational offset of each source image, we consider rotated versions of each source image. The source image database consists of 600 randomly-selected images from the flickr$^{\text{TM}}$ "squared circle" group. All the target tiles are circles, laid down in back-to-front order. We turn the problem of finding the best source-image rotation into the problem of finding the best shift (along the θ axis) by transforming all the target tiles, target characteristic functions (W), and source images to a polar representation. This problem is easier to solve than the previous version because it has only one shift parameter, rather than two. As a result, the optimal shift can be found using the one-dimensional FFT instead of the two-dimensional FFT, changing the running time from $\mathcal{O}\left(MPN^2 \log N\right)$ to $\mathcal{O}\left(MP\left(N^2 + N \log N\right)\right)$.

6 Conclusions and Future Work

We have shown that it is computationally feasible to incorporate deeper levels of complexity and optimization into the production of image mosaics. Our method handles arbitrarily shaped target tiles, and finds the optimal offset for each source image so that it most resembles the target tile. The FFT is used to accelerate the computation. Moreover, this framework also accommodates colour correction with minimal overhead. Combining these improvements with a globally optimal target-tile/source-image assignment algorithm yields mosaics that more accurately depict the target image while maintaining an impressive level of discernibility for the details within the source images.

The potential of the alpha image W has not been showcased in this paper. We use it simply as a binary mask to define each target tile shape. However, its role in the SSD cost function is that of an arbitrary weighting function that could be used to assign continuous weights to any pixel in the target tile. For example, the binary W used in this study could be convolved with a simple windowed-averaging kernel to achieve a tapered drop-off of weights on the borders of the tile pieces. This change reduces the influence of the image content near the borders of the tile. Conversely, a higher weight could be assigned to the tile edges in an attempt to enforce continuity of edges across tile borders. Blending across edges might then obscure the seams between source images.

The cut-out image mosaic method outlined here is admittedly slower than many other mosaicking methods. Our implementation is not entirely optimized, and an implementation in C or C++ might yield speed gains. However, the bulk of the processing is in computing the inverse FFT, and Matlab has an efficient implementation of this operation. Irrespective of such optimizations, we hope that the improved mosaic quality justifies the additional effort.

The matching method here stems from an image registration method, finding the optimal parameters to register each source image to each target tile. In our experiments, we used translation

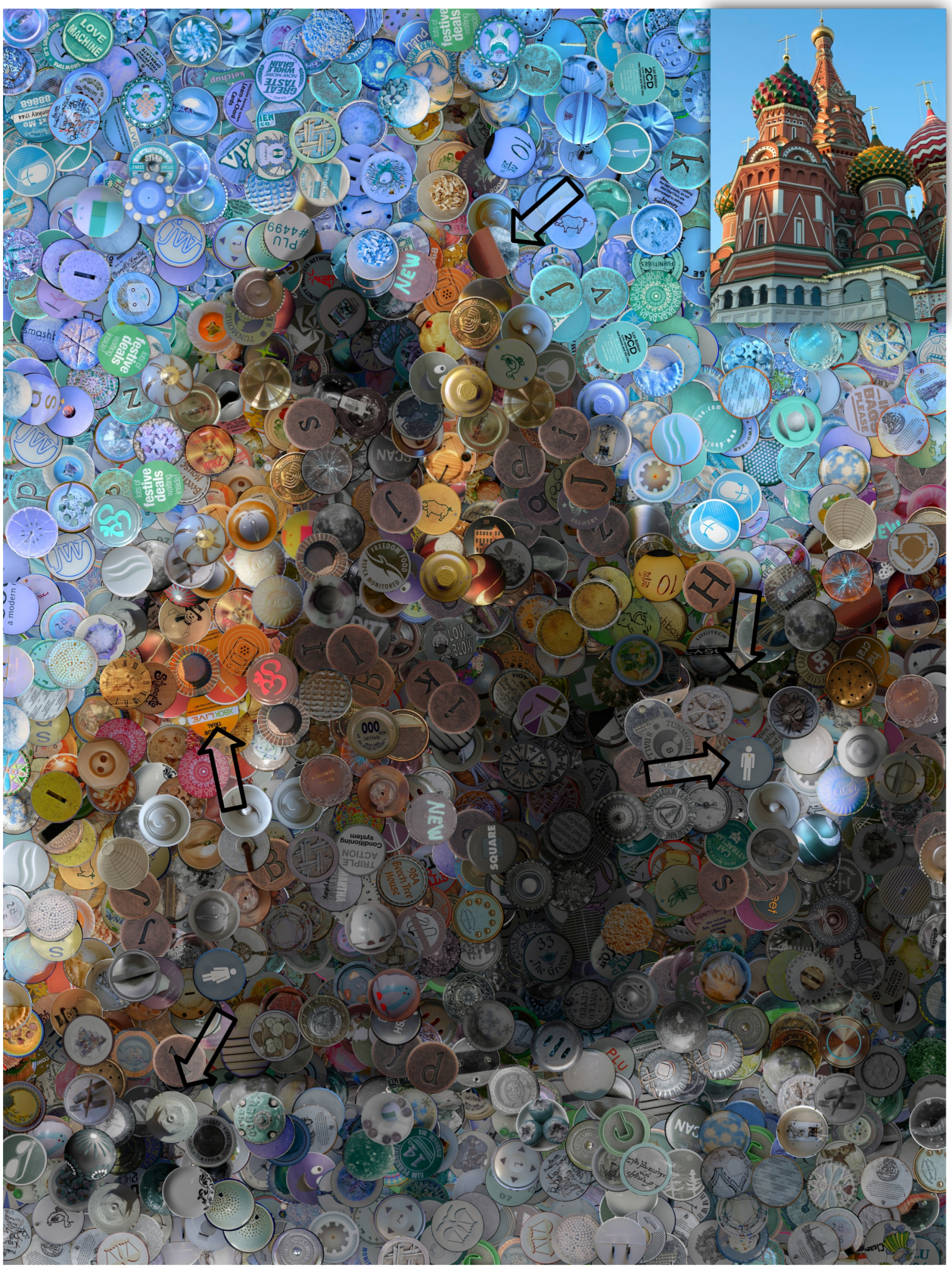


Figure 5: *St. Basil's Cathedral rendered using a source database of 600 circle images from the flickr™ "squared circle" group. Each source image was used up to three times (once for each colour-correction scheme). The five arrows indicate some places where the source image and rotation angle provide a good fit for the underlying target.*

alone, or rotation alone. However, one could consider more complex spatial transformations such as rigid-body motion (translation and rotation together). Some work exists in rigid-body registration using the fast SSD computational framework [Omanovic and Orchard 2006; Orchard 2007]. However, we should expect that adding more dimensions to our parameter space will make the search take longer.

Finally, we are considering the use of the singular value decomposition (SVD) [Golub and Van Loan 1996] to narrow our search of the source database by comparing each target tile to the singular vectors associated with the largest singular values. However, it is not clear that the same fast SSD method can be used in this framework.

Acknowledgments

We would like to thank Adam Finkelstein for assistance during this project. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Canada Foundation for Innovation. We also thank the contributors to the flickr "squared circle" group who made their work available under a Creative Commons license.

References

BATTIATO, S., DI BLASI, G., FARINELLA, G. M., AND GALLO, G. 2006. A survey of digital mosaic techniques. In *Proc. of Eurographics Italian Chapter Conference*, G. Gallo, S. Battiato, and F. Stanco, Eds.

BÜHREN, M., 2007. Functions for the rectangular assignment problem. Matlab File Exchange `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6543&objectType=file`, August.

BUMGARDNER, J., 2008. flickr^{TM}: faces in places. `http://www.coverpop.com/pop/flickr_facesinplaces/`, February.

COOLEY, J. W., AND TUKEY, J. W. 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation 19*, 90, 297–301.

DALAL, K., KLEIN, A. W., LIU, Y., AND SMITH, K. 2006. A spectral approach to NPR packing. In *Non-Photorealistic Animation and Rendering*, ACM, 71–78.

DI BLASI, G., GALLO, G., AND PETRALIA, M. P. 2005. Puzzle image mosaic. In *Proc. of IASTED/VIIP2005*.

DI BLASI, G., GALLO, G., AND PETRALIA, M. P. 2006. Smart ideas for photomosaic rendering. In *Proc. of Eurographics Italian Chapter Conference*, G. Gallo, S. Battiato, and F. Stanco, Eds.

FINKELSTEIN, A., AND RANGE, M. 1998. Image mosaics. In *EP '98/RIDT '98: Proceedings of the 7th International Conference on Electronic Publishing, Held Jointly with the 4th International Conference on Raster Imaging and Digital Typography*, Springer-Verlag, London, UK, vol. LNCS 1375, 11–22.

FITCH, A. J., KADYROV, A., CHRISTMAS, W. J., AND KITTLER, J. 2005. Fast robust correlation. *IEEE Trans Image Proc 14*, 8 (August), 1063–1073.

FRISTON, K. J., ASHBURNER, J., FRITH, C. D., POLINE, J.-B., HEATHER, J., AND FRACKOWIAK, R. S. J. 1995. Spatial registration and normalization of images. *Human Brain Mapping 33*, 165–189.

GOLUB, G. H., AND VAN LOAN, C. F. 1996. *Matrix Computations*. Third Edition, Johns Hopkins University Press, Baltimore, MD.

JACOBS, C. E., FINKELSTEIN, A., AND SALESIN, D. H. 1995. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95*, 277–286.

KILTHAU, S. L., DREW, M. S., AND MÖLLER, T. 2002. Full search content independent block matching based on the fast Fourier transform. In *Proc. International Conference on Image Processing*, vol. 1, I–669–I–672.

MUNKRES, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics 5*, 1 (March), 32–38.

OMANOVIC, M., AND ORCHARD, J. 2006. Efficient multimodal registration using least-squares. In *Proc. of the International Conference on Image Processing and Computer Vision (IPCV'06)*, H. R. Arabnia, Ed., vol. 1, 230–233.

ORCHARD, J. 2005. Efficient global weighted least-squares translation registration in the frequency domain. In *ICIAR'05, Lecture Notes in Computer Science, LNCS*, M. Kamel and A. Campilho, Eds., vol. 3656, 116–124.

ORCHARD, J. 2007. Efficient least-squares multimodal registration with a globally exhaustive alignment search. *IEEE Trans Image Proc 10*, 16 (October), 2526–2534.

SILVERS, R. S., 2001. Digital composition of a mosaic image. Canadian Patent 2226059, October.

TRAN, N. 1999. Generating photomosaics: an empirical study. In *Proc. of the 1999 ACM symposium on Applied computing*, ACM Press, San Antonio, Texas, 105–109.

ZELNIK-MANOR, L., AND PERONA, P. 2007. Automating joiners. In *Non-Photorealistic Animation and Rendering*, ACM.

ZHANG, Y. 2002. On the use of CBIR in image mosaic generation. Technical Report TR 02-17, University of Alberta, July.