

Histogram Painting for Better Photomosaics

Brandon Lloyd, Parris Egbert
Computer Science Department
Brigham Young University
{blloyd | egbert}@cs.byu.edu

Abstract

Histogram painting is a method for applying local histogram transformations to an image without introducing discontinuities at the boundary of the transformed region. Using interactive tools similar to those found in standard paint programs, the histogram from one part of an image can be “painted” onto another. This is useful for improving the results of seam elimination algorithms in photomosaicing applications. When local intensity distributions differ significantly at tile edges, a noticeable shift in appearance may remain even after the seam is removed. Histogram painting can be used to reduce these differences before removing the seam. The technique can also be used to remove shadows from an image by “painting” the histogram from lit regions onto the shadowed portions.

Keywords: seam removal, shadow removal, histogram matching.

1. Introduction

Images acquired from satellite or aerial photography have a limited field of view. Many overlapping image tiles are often placed together to form a larger photomosaic. Visible seams between individual tiles destroy the illusion that the photomosaic is one continuous image. Seams can arise from intensity differences along the edges of the tiles caused by variations in atmospheric conditions, season, time of day, or changing surface features. Even when the tiles are photographed at nearly the same time, differences in film, scanning, or post-processing can cause subtle intensity variations that create visible seams. A simple solution to the problem is to perform a graduated cross-fade between adjacent tiles within the region of overlap. For this to produce satisfactory results, the features within the overlap region must line up perfectly. If they are mis-registered at all, the cross-fade will create a distracting ghosting effect. Unfortunately, a cross-fade with images acquired by traditional techniques will almost always cause ghosting because of the perspective distortion introduced by the camera.

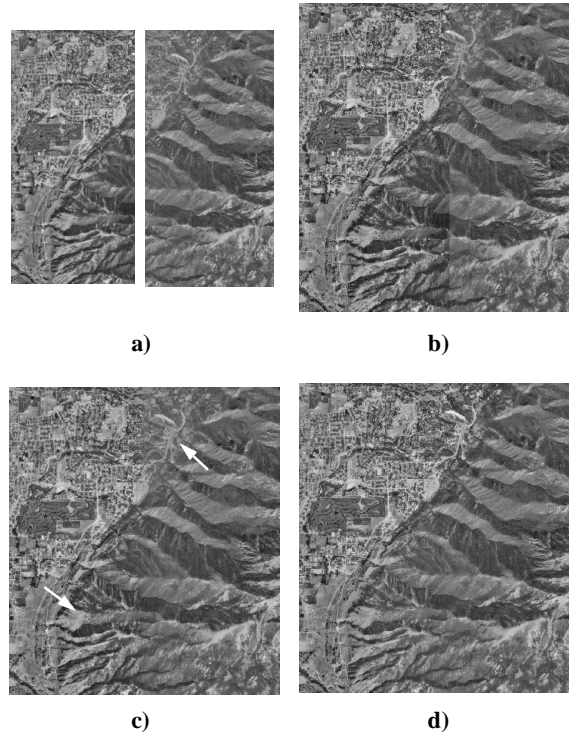


Figure 1: **a)** Two overlapping tiles. **b)** Mosaic without seam elimination. **c)** Though seam elimination improves the mosaic significantly, the contrast differences in the areas indicated by the arrows reveal the border between the tiles. **d)** A uniform mosaic created by using histogram painting.

More sophisticated algorithms can perform seamless blends between two image tiles without creating ghosting artifacts. However, if there are differences in the contrast or the intensity distributions of the tiles, one will still be able to distinguish the individual tiles in the photomosaic. Though the seam in Figure 1c is blended well, even the subtle difference in contrast between the two halves of the single image betrays the fact that the image is actually composed of two images blended together. Milgram [5] suggested performing a histogram equalization on both images before removing the seam. Histogram matching [4], which is a generalization of histogram equalization, could also be used to coerce the histogram of one image to

match the other, or to match both to a predetermined histogram. The drawback of such an approach is that these transformations are applied globally to the whole tile and this may not always be desirable. The two tiles in Figure 1, for example, come from a larger context wherein the city regions have the appearance of the tile on the left and the mountain regions have the appearance of the tile on the right. We would like to leave those regions as they are in order to keep them consistent with surrounding tiles. The inconsistent regions are the small city area in the upper left corner of the right tile which has too little contrast, and the mountain area in the lower right of the left tile which has too much. Global transformations like histogram equalization and histogram matching cannot deal with this situation. Furthermore, even if a global transformation could be performed, local intensity distributions may still differ after the transformation.

We present a method called histogram painting which is used to apply local histogram transformations. The most important contribution of this paper is a method for applying the local transformation without introducing sharp discontinuities at the boundary of the transformed region. For photomosaicing applications, histogram painting allows the user the local control necessary to match the intensity distributions more closely near the tile edges. This improves the results of seam elimination algorithms which are subsequently used to remove the intensity discontinuities. Since hand adjustments to a large number of tiles can be tedious, we also present an automated algorithm that can be applied when the tiles have a significant amount of overlap.

2. Previous Work

The construction of a photomosaic begins with the registration of image tiles. Brown [2] has written an excellent survey of registration techniques. Once the images are lined up correctly, additional processing is often necessary to create a seamless photomosaic. Several researchers have explored methods for eliminating seams. Milgram [5] suggested an algorithm for selecting a seam within the overlapping region between two tiles such that the intensities match well at the seam. The seam path is constructed one row of pixels at a time by assigning a cost to each position on the row based on the intensity differences between the two tiles on either side of the position and the deviation from the already constructed path. Dynamic programming is used to determine the path with the lowest cost. Discontinuities may still occur at the selected seam points. These are eliminated by smoothly ramping intensities over a small range across the seam. Shiren et. al [7] later improved the technique by extending the seam-point searching algorithm to two dimensions within the overlapped region. Araújo and Leite [1] use watersheds of the difference image in the overlapping region to efficiently calculate a seam along lines of maximum correlation. Peleg [6] used iterative relaxation to compute a seam elim-

ination function over each tile, which when added to the original tiles, adjusts the intensities of each pixel to ensure that they match at the edges. The function is constrained to be as smooth as possible in the interior of the tile to minimize visible artifacts.

Burt et. al [3] used a multi-resolution algorithm to create seamless transitions between images that may be completely unrelated. They observed that in a graduated crossfade between two images, if the width of the transition zone T is large compared to the wavelength of the highest spatial frequencies in the image, the ghosting effect will be introduced. On the other hand, if T is small compared to the wave length of the lowest prominent frequency, a visible, albeit somewhat blurred step in intensity may result. To blend two images together with neither steps nor ghosting, the cross-fade must be performed separately on different spatial frequency bands with a transition width appropriate for the frequencies contained in each sub-band. The images to be blended are first decomposed into two image pyramids. The highest frequency sub-band is at the base of the pyramid. Each level above the base contains the sub-band with half the frequency of the level below it. A new image pyramid is created by blending each level of the two source image pyramids with a transition width appropriate for the frequencies in that level. Adding together each level of this new image pyramid yields the final, seamless image.

While all of these algorithms may be used to remove tile seams, none of them alone is sufficient to achieve consistent intensity distributions across tiles, especially heterogeneous tiles that consist of significantly different features such as the mountains and city regions of Figure 1. We use histogram painting to first achieve a smooth histogram transition in intensity distribution between image tiles and then apply any of the aforementioned seam elimination techniques to remove intensity discontinuities at the edges.

3. Histogram Painting

Histogram painting refers to the application of a local histogram transformation interactively with the mouse or stylus. The user begins by selecting a source region within the area that will be modified and a target region which has the desired intensity distribution. A mapping is created that will transform the histogram of the source area to the histogram of the target region. Once the mapping is established there are two techniques to apply the transformation. For simple regions, the user outlines the area to modify and specifies a transition width. The pixels in the center of the region are fully transformed with a smooth, uniform-width transition at the region boundary. The other application technique affords the user more precise control. Using brushes of varying sizes and opacities the user can control the amount of transformation that is applied. Figure 2 shows an image modified with the histogram painting tool. Both application methods use an off-screen

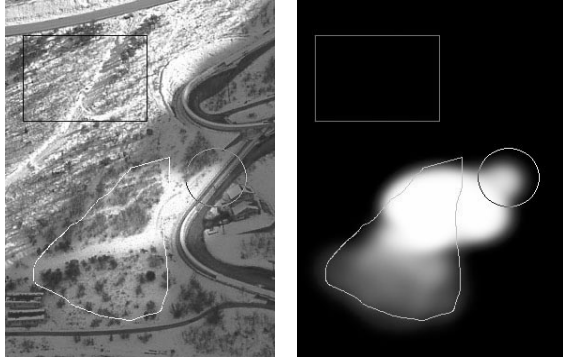


Figure 2: a) A view of a typical histogram painting operation. The regions in the rectangle and oblong outline are used for the target and source histograms respectively. b) The transition map created by the user’s brush strokes.

image called the transition map to control the degree to which each pixel is transformed.

3.1 The Transition Map

The transition map is used in conjunction with histogram matching to produce the intensity transformations. The intensity in the transition map is used to determine how much to transform the corresponding pixels in the image. Pixels in white areas undergo the full histogram transformation while black areas remain untouched. Gray areas are only partially transformed. Figure 2b shows the transition map used for the image in Figure 2a. Both application techniques modify the values in the transition map and update the pixels in the image accordingly. For the region with a fixed transition width, the transition map is created by first drawing the region filled with white on a black transition map. The map is then filtered with a gaussian filter with a radius equal to the specified transition width. For brushes, gaussian kernels are splatted on the transition map as the brush is applied. The brushes can also be used to “erase” the transformation by subtracting the kernel. Painting operations continue to manipulate the same transition map until the painting operations are “frozen.” At this point the transition map is cleared, the transformed pixel values are written to the image, and a new histogram transformation must be selected. We interactively update the display in response to user brush strokes, using a two-dimensional look-up table (LUT) with which to calculate transformed pixel values. The LUT is indexed by the intensity of the pixel in the original image and the corresponding pixel in the transition map. This table is generated with histogram matching.

3.2 Histogram Matching

The goal of histogram matching is to transform the intensity values of one image in such a way that its histogram matches that of another image. A basic result from statis-

tics shows how this can be done. A random variable X with probability density function (pdf) $f(x)$ can be transformed into a random variable with uniform distribution by using its cumulative density function (CDF) as a transformation function. The CDF for X is:

$$F(x) = \int_0^x f(t)dt$$

The inverse of the CDF of X will transform a uniform random variable U on the interval $[0,1]$ into a random variable with the same distribution as X .

$$F^{-1}(U) = X$$

$F^{-1}(u)$ may not exist if $F(x)$ has regions where the function is flat. To ensure a unique inverse for each x we use the function:

$$F^*(u) = \{\max(x) | f(x) = u\}$$

To transform X into a random variable Y with distribution $g(y)$ we first use $F(x)$ to transform X into a uniform random variable. Then we use $G^*(y)$ to transform from the uniform random variable to $g(y)$. The complete transformation is given by:

$$h(X) = G^*[F(X)] \quad (1)$$

The process is represented graphically in Figure 3.

By rescaling pixel intensities to the range $[0,1]$ and normalizing the area under the histogram curve to 1.0 we can transform the histograms of the source image and the target image into pdfs $f(x)$ and $g(y)$ respectively. Using Equation 1 we can create a LUT that will transform the distribution of intensities of the source image to that of the target image.

We tried two approaches for generating intermediate histogram transformations. First, we tried using an interme-

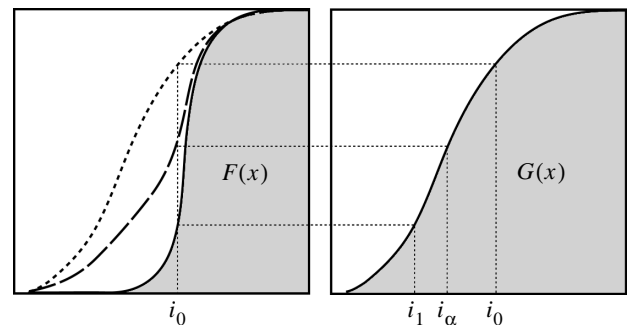


Figure 3: Matching histogram CDF $F(x)$ to $G(x)$. The intensity i_0 is mapped to i_1 by the transformation. The dashed line is the linearly interpolated CDF with parameter $\alpha = 0.5$. With simple linear interpolation, i_α would lie right in the middle of i_0 and i_1

diate target CDF by linearly interpolating between the source CDF and the target CDF, with the value in the transition map as the interpolation parameter. We also tried linearly interpolating between the identity LUT and the LUT generated by the histogram match to create an intermediate LUT. We prefer the second approach, not only because it is simpler and easier to calculate, but more importantly, the intensities change at a constant rate with respect to the interpolation parameter. This makes the histogram painting tools easier to use because their behavior is more predictable.

For color images we perform histogram matching separately on the red, green, and blue channels. This does not always yield expected results. The user's intuition of histogram painting is that it makes the intensities of one area look like those of another. The user expects to see the dominant color of the source region be replaced by the dominant color of the target region. However, when the channels are histogram matched separately, colors present in neither of the regions can emerge. There is also a uniqueness problem. Consider two identical images, one in color and the other in grayscale. If we attempt to recolorize the grayscale with histogram painting by using the color image as the target the operation will fail because different colors may map to the same grayscale intensity. Despite its limitations, this simplistic approach to handling color works very well for our application of cleaning up aerial photography in photomosaics.

4. Automatic Edge Histogram Blending

The seam elimination algorithms work best when the adjacent images have similar intensity distributions in the area of overlap. If the tiles have a significant amount of overlap, we can use an automated algorithm to make a smooth transition from the histogram of one image to the other within the overlap region. The transition map is formed by creating a smooth linear ramp across the region. The transition map need not be created explicitly as its only purpose is to provide the interpolation parameter for the histogram blend, and this can be computed on-the-fly. For images with homogeneous features, it is sufficient to use the entire overlapping region in each image for the source and target histograms. If there are heterogeneous features in the overlapping region, however, it is better to use an algorithm that adapts to changes in the local histogram. We obtained good results by using a sliding window for the histogram computation as shown in Figure 4. At each step the source and target histograms are taken from a small window in the overlapping region. The histogram transformation LUTs are calculated as usual. The mapping is used to transform n columns of pixels in the center of the window. The window is then shifted down n columns. The process is repeated along the length of the overlapping region. The histograms can be efficiently updated by adding the pixels on the leading edge of the sliding window

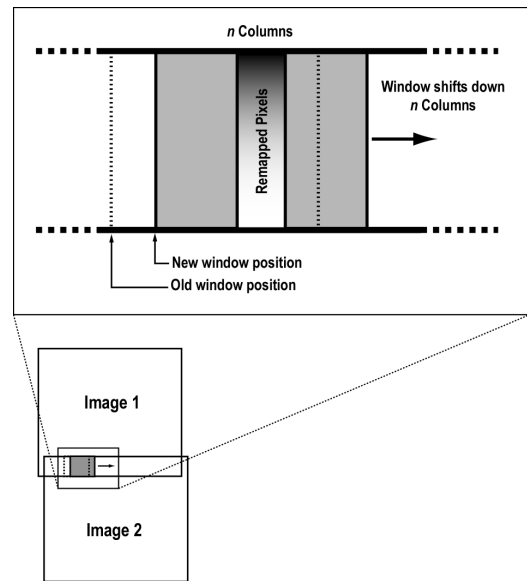


Figure 4: Sliding window used in automatic edge histogram blending. Histograms are taken from the shaded region. Only the narrow band of pixels in the center of the region is transformed.

and subtracting those on the trailing edge. The choice for n depends on the images. If the intensity distributions change rapidly along the length of the overlapping region and n is too large, the algorithm can create noticeable steps. If n is small then the histogram transformation LUTs must be calculated more often and the process takes longer. We achieved satisfactory results using a value of n that is approximately one-tenth of the width of the sliding window.

5. Shadow Removal

One of most serious intensity distribution disparities between adjacent image tiles occurs when one tile is shadowed and the other is lit. Histogram painting can be used to remove shadows under certain conditions. First, there must be enough detail in the shadowed region. Applying histogram painting to a completely black shadow will only create a lighter blob, because it cannot restore detail that was not present in the original image. Second, shadow removal works best for large shadows like those created by a mountain slope or cloud cover. It is far too tedious to paint out many small shadows such as those cast on a flat rocky terrain with the sun low in the sky. Third, we found that it was easier to handle soft rather than hard shadow edges. When the histogram from a lit region is painted onto another lit region, the pixels are brightened excessively. It is extremely difficult to prevent this from happening with a hard edge because if the brush crosses the edge by even one pixel there will be a bright halo surrounding the region that was shadowed. Soft edges are easier to handle because they do not require as much preci-

sion. One last consideration is that the objects in “relit” portions of the image will not cast shadows if these were not present in the original image. This may actually be desirable if the image is going to be used as input into a classification algorithm or as a texture map on a model with novel lighting.

Shadow removal begins by picking regions in lit and shadowed areas of the image to use for the target and source histograms, respectively. It is important that these regions be as similar as possible and include all the features in the shadowed region. For example, in Figure 6a we picked regions that contained both trees and road. The roads do not look right if we paint them with a histogram transformation obtained from regions consisting only of trees because the road-colored pixels are not contained in the target and source histograms. Once we have established histogram transformation we proceed with the painting operations. Large interior regions can be painted quickly using a large brush at full opacity or the simple region fill tool. The edges require more precision. Soft shadow edges are not as dark as the interior and require less transformation. A small brush with low opacity provides the necessary control. The brushes are additive so that running the brush back and forth over a region slowly brings the transformation to just the right level as the intensity builds up in the transition map. If we overshoot and a region becomes too bright we slowly back off the transformation by a similar operation with a low opacity

erase tool. Using this method it is possible to create a seamless transition between shadowed and lit areas.

6. Results

Figure 5 shows a seamless 8GB, meter-per-pixel resolution photomosaic of a stretch of land nearly 90 miles long. The data is used for an interactive terrain visualization program. Each tile was first histogram matched to a representative tile with similar features in order to equalize the tile brightness and contrast levels globally. We then used histogram painting to make the necessary large-scale adjustments at a tile’s edges to match it with its neighbors. Automatic edge matching was then applied to all edges before using a seam removal algorithm.

Figure 6 shows two images partially shadowed by a mountain and by clouds. Using histogram painting these shadows have been completely removed in just a few minutes.

7. Conclusions and Future Work

Existing seam elimination algorithms successfully remove intensity discontinuities, but this is insufficient to create photomosaics in which the constituent tiles are indistinguishable. The intensity distributions must match as well. Histogram painting allows the histograms to be changed locally with smooth transitions into untouched areas of the

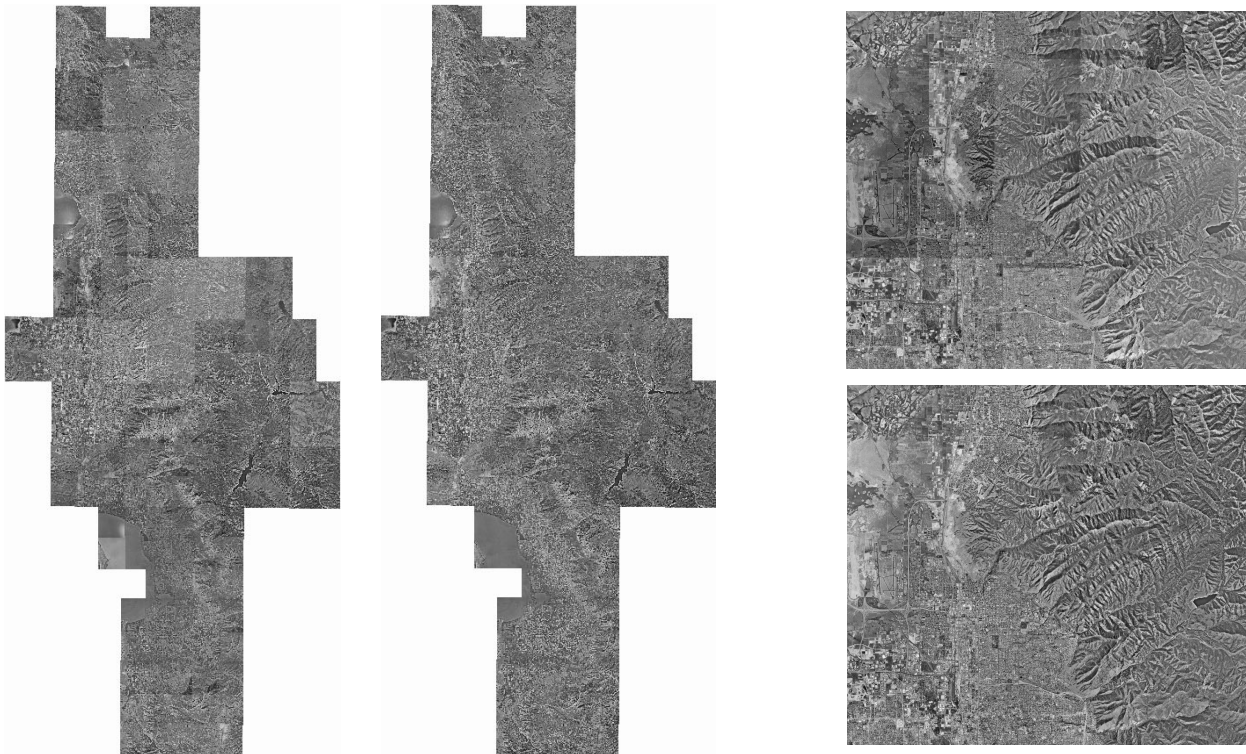


Figure 5: A comparison of a distant and closeup view of a large data set before and after histogram transformations. Automatic edge histogram blending was applied to all of the tiles.

image. Histogram painting can remove shadows, contrast differences, and other inconsistencies that reveal tile locations in a photomosaic. We present a simple method utilizing a transition map to perform local pixel transformations without introducing discontinuities at the boundaries of the transformed regions.

There are several ways to make histogram painting easier to use. It can be quite tedious to get shadow edges just right. It would be useful to detect shadow edges and automatically generate the transition map. When modifying large areas with subtle histogram shifts it would also be useful to use the local histogram around the current position of the brush in order to better adapt to local histogram changes. Finally, we would like to investigate better ways of performing histogram transformations on color images.

References

- [1] F. Araújo Jr. and N. J. Leite, A Morphological Algorithm for Photomosaicking, *Proceedings VIII European Signal Processing Conference*, Trieste, Italy, 1996, pp. 181-186.
- [2] L. Brown, A survey of image registration techniques, *ACM Computing Surveys*, vol. 24, no. 4, 1992, pp. 326-376.
- [3] P. Burt and E. Adelson, A Multiresolution Spline With Application to Image Mosaics, *ACM Transactions on Graphics* vol. 2, no. 4, 1983, pp. 217-236.
- [4] K. R. Castleman, *Digital Image Processing*, (Upper Saddle River, New Jersey: Prentice Hall, 1996), pp. 91-93.
- [5] D. Milgram, Adaptive Techniques for Photomosaicking, *IEEE Transactions on Computers*, vol. 26, 1975, pp. 1175-1180.
- [6] S. Peleg, Elimination of Seams for Photomosaics, *IEEE Conference on Pattern Recognition and Image Processing*, 1981, pp. 426-429.
- [7] S. Yang, L. Li, and G. Peng, Two Dimensional Seam-Point Searching in Digital Image Mosaicking, *Photogrammetric Engineering and Remote Sensing*, vol. 55, no. 1, 1989, pp. 49-53.



a)



b)



c)



d)

Figure 6: a) and c) Regions of images shadowed by mountains and clouds respectively. b) and d) The same regions with shadows removed.