# Hierarchical Matching Techiques for Automatic Image Mosaicing

C.L Begg, R Mukundan

Department of Computer Science, University of Canterbury, Christchurch, New Zealand
clb56@student.canterbury.ac.nz, mukund@cosc.canterbury.ac.nz

## Abstract

This paper looks at image mosaicing using hierarchical matching techniques as a way to create a novel view of a scene quickly. Several parameters including error function, region match and constraints are evaluated to find the fastest and most accurate mosaics. Constraints are shown to be very important to intensity matching, hue matching is more accurate than intensity matching, and hue matching with the intensity correction produces overall better results.

**Keywords**: Image-based Rendering, Mosaicing, Cylindrical Panorama, Seamless Tiling

## 1 Introduction

Computer graphics and computer vision are two important areas in computer science. Movies are requiring higher quality graphics, as are computer games. Tele-reality [1]—remote (time or space) visualisation of a scene from reality—is becoming increasingly used in engineering, science and medicine. Image-Based Rendering [2] is a set of methods that take images from the real world and produce novel (new, but not trivially derived) views of the scene. Image-Based Rendering is a cross-over link between computer vision and computer graphics. There are a number of View Synthesis Techniques, including:

- Image Warping or Morphing [3]
- Image Mosaicing
- Stereopsis
- Three and Multiple Camera Geometry [4]

Some of these methods are used to extract depth values and structural information from an image, and can therefore recover a partial 3D model of the scene.

Of these techniques, this paper will be focusing on Image Mosaicing and areas associated with it.

Image mosaicing creates larger images by "stitching" together many smaller images. This is often done by hand to create long panoramas using photographs. This unfortunately leaves visible joins between photos (called seams) and often the images do not line up correctly.

Computer can manipulate images quickly and easily re-project the images so they line up. Previously, a user was required to point out common points to the computer so it could do the re-projection. In the last 11 years, much research has been undertaken in how the computer can find corresponding points. Some of the methods created depend on exact measurements of the camera(s) and their positions and orientations. Other methods look for features in the images[5], and some only look at the colour values of the pixels in the images. Almost all of the algorithms assume uniform illumination and/or some epipolar geometry[6].

When the two assumptions about uniform illumination and epipolar geometry are not made, more source images are available for use in mosaicing and this decreases the need for preprocessing. It also allows images to be taken on one day and extra images taken on another day to fill in gaps in the mosaic.

The significance of the lack of epipolar geometry is that any two images that have—or appear to have—an overlap, can be stitched together. No prior knowledge of the (approximate) location or size of the overlap is known. No assumption can made as to which image is the left and right, or top and bottom, or even which of those two cases the input will be.

## 2 Hierarchical Image Matching

Hierarchical Image matching is a directed search technique using Gaussian pyramids. The lowest resolution is searched to find the best solution by some metric, and the position of that solution defines the area to be searched in at the next resolution higher. Pseudo-code for this algorithm is given in figure 1.

```
create image pyrimid
set min and max x and y
for each level starting at the lowest
                                  res
    calcuate error for each possible
                             overlap
    find region of best match
    set new min and max x and y based
                            to region
create new image with overlap using x
                  and y of the middle
                  of the region
```

Figure 1: Pseudo-code for hierarchical image matching

For this paper, two image types were investigated for matching: Intensity and Hue.

## 2.1 Hierarchical Intensity Matching

Hierarchical intensity matching was the first automatic mosaicing methods attempted using the algorithm shown in figure 1. Only the intensity was used to create the image pyramid and used in matching. The final image is created using the original colour images and the calculated best overlap.

### 2.1.1 Error Function.

Three error functions were evaluated, average absolute difference, root-mean-square of difference, and root of the average absolute difference of squares.

The first function is the simplest

$$error = \sum ABS(v_1 - v_2)/n \tag{1}$$

Root-mean-square is

$$error = sqrt(\sum (v_1 - v_2)^2/n) \tag{2}$$

Root of average absolute difference of squares

$$error = sqrt(\sum (ABS(v_1^2 - v_2^2))/n) \tag{3}$$

### 2.1.2 Region Matching.

Three region matching algorithms where tried.

The first looks for the single lowest error.

The second looked for the region of 9 neighbouring overlaps that give the lowest average error.

The last is the same as the second, but attempts to penalise regions that have very small overlaps (typically less than 20 pixels in overlapping region). This last region match as created to try and counter act the behavior of the previous algorithm shown in the results.

### 2.1.3 Gaussian Pyramid Depth.

The resolution of the smallest image that matching was performed on was controlled and the speed and accuracy of the match was looked at. The size of the smallest image has a large impact on the speed, as every possible overlap is checked (within the constraints below), and only 121 overlaps per level.

Obviously the greater the depth, the smaller the initial image and the faster the matching occurs, but it is possible for a mismatch to occur, precluding a good match.

### 2.1.4 Matching Constraints.

By constraining the initial possible match, the matching process is sped up and the accuracy is improved by not allowing areas that shouldn't match but has low error value.

This is a more strict method of preventing very small overlaps than the third region matching function in section 2.1.2.

## 2.2 Hierarchical Hue Matching

Hierarchical hue matching follows the same algorithm to hierarchical intensity matching, but using the hue to match on, instead of intensity. It has been suggested that this method might be more robust than intensity matching. Because the intensity is not used in the matching process, it can be used to calculate an intensity correction that should make seams less visible.

The region matching and constraints were also added to hue matching.

### 2.2.1 Error Function.

Hue is not a linear number, it is a cycle that has values between 0 and 360 degrees, and 0 and 360 are the same value. This complicates the error function, because it can not be a direct number comparison.

The two error functions tried were average absolute shortest difference and average distance between the hue-saturation points.

The first function can be defined for each pixel as

$$error = MIN(ABS(h_1 - h_2), ABS(360 - h_1 + h_2)) \tag{4}$$

The second function for each pixel is

$$error = SQRT((x_1 - x_2)^2 + (y_1 - y_2)^2) \tag{5}$$

where

$$\begin{aligned} x_n &= s_n \times cos(h_n) \\ y_n &= s_n \times sin(h_n) \end{aligned} \tag{6}$$

#### 2.2.2 Intensity Correction.

Because the intensity is independent of the hue being matched on, it is possible to use the final overlapping region to calculate an intensity correction and apply it to the images. The correction is calculated by finding the average difference in intensity between the two images inside the overlap, and adding that value to every pixel in the second image.

### 2.3 Comparison of Intensity and Hue Matching

A comparison between intensity and hue matching was made using the results from the two matching methods.

## 3 Experimental Results

Evaluating the performance of some of the experiments is subjective when it comes to the visibility of seams and accuracy of matching.

### 3.1 Hierarchical Intensity Matching

#### 3.1.1 Error Function.

There was no difference in the position matched when using any of the error functions, even in combination with some of the variation below. The first functions is the simplest, and fastest to compute, and was used for the other results.

#### 3.1.2 Region Matching.

The region matching algorithm made a very large difference when it was changed.

The one lowest error position worked well in about 50% of cases.

The second method looking for a region of 9 neighbouring positions with the lowest average error did work well for some input image pairs, but often matched on a single row or column of pixels, as shown in figure 2[1].

[1] All images are available in colour at http://llnz.dyndns.org/gallery/lmtt/

The third method trying to penalise regions that have very small overlaps did improve the results from the second method, but did not completely solve the problem. This method was also much slower than the first method.

#### 3.1.3 Gaussian Pyramid Depth.

The depth of the pyramid does make a large difference in both time and accuracy. Each level further makes the process around much faster down to the constant overhead of 0.14s.

| Depth | Time elapsed |
|---|---|
| 7 | 0.140s |
| 6 | 0.150s |
| 5 | 0.200s |
| 4 | 0.350s |
| 3 | 2.160s |
| 2 | 17.590s |
| 1 | 4 min, 11.32s |

Accuracy is not greatly affected by the depth. For 2 mega-pixel images, no change in final matching position is found until around the 7th level, where the images are 5x7 pixels.

#### 3.1.4 Matching Constraints.

Adding constraints to the matching position made a very large difference to the mosaics.

A set of typical constraints were created that work for most image sets. The left-most point of the second image must occur between 50% and 95% of the width of the first image, and the top-most point within 20% of the height either up or down of the top of first image.

There is also a small reduction in time taken for matching, due to there being a smaller number of overlaps to check at the first level.

### 3.2 Hierarchical Hue Matching

#### 3.2.1 Error Function.

The error function for hue matching made considerable difference to the final mosaic created.

The first function is slightly faster to compute, but does not match well as often as the second function. Neither function matched correctly all the sample input images that were tried.

#### 3.2.2 Intensity Correction.

The intensity correction does make a very large difference to the quality of the final mosaic. Most seams are completely invisible, and the rest have

Figure 2: Region of 9 mismatch, typical

(a) Original hue matched image

(b) Intensity corrected hue matched image

Figure 3: Intensity corrected mosaic using hue matching

obvious perspective distortion as can be seen in figure 3.

### 3.3 Comparison of Intensity and Hue Matching

As shown in figure 4 is an example of intensity and hue matched seams. This result is typical of the comparisons done between the two methods. Hue matching in most cases does provide a better match, and due to the intensity correction, a more seamless mosaic. Some exceptions were found, such as when the hue does have a large contrast of values.

## 4 Discussion and Further Work

Constraints do add an assumption that had not been present in previous automatic matching that was done in this project. The assumption is that there is an order to the input images. For example, the first input image is the left-most image, and the second image is the right-most. For most applications this assumption will not cause problems.

There are many areas that need to be looked at. Instead of hierarchical/Gaussian pyramid matching, a 2d sliding window model needs evaluation, as do feature extraction methods, which would also allow for automatic perspective corrections. In Hue matching, more research is needed to evaluate error functions to find ones that work more often. Another area for future work is to look at matching on hue and intensity, and matching gradients and edges instead of individual pixels.

## 5 Conclusions

The error metric in intensity matching does not make a significant difference, region matching does make some difference without constraints. Constraints make a very large difference to the speed and accuracy of mosaicing, when correctly set.

Hue matching does work, with some error metrics working better than others. Intensity correction does make the seams much less visible—only perspective distortion is apparent.

Hue matching gives a better final mosaic than hierarchical intensity matching, due to being more robust to noise and a better descriptor for the scene.

## References


[1] R. Szeliski, "Image mosaicing for tele-reality applications," Tech. Rep. CRL 94/2, Cambridge Research Lab, Digital Equipment Corp, May 1994.

[2] S. B. Kany, "A survey of image-based rendering techniques," Tech. Rep. CRL 97/4, Cambridge Research Lab, Digital Equipment Corp, August 1997.

[3] G. Wolberg, "Recent advances in image morphing," *Proc. Computer Graphics International*, 1996.

[4] O. Faugeras and Q. Luong, *The Geometry of Multiple Images.* MIT Press, 2001.

[5] Y. Kanazawa and K. Kanatani, "Image mosaicing by stratified matching," *Image And Vision Computing*, vol. 22, p. 11, Feb 2004.

[6] A. Zisserman, "Geometric framework for vision," 1997.

(a) Intensity image (b) Hue image

Figure 4: Comparison of Intensity and Hue Matching