Intensity Based Image Mosaicing

Vinod D.S.
Dept.of ISE, SJCE
Mysore, India

Vidyullatha Prakash

ShivaPrakash M.

Akshatha R. Bhat

Shanmugam Kannan

Abstract: Image Mosaicing is useful in a variety of vision and computer graphics applications like 3d vision, photogrammetry, satellite imagery, video images etc. The basic reason for Image Mosaicing is that camera vision is limited to 50 by 35 degrees, and Human Vision is limited to 200 degrees. By using Image Mosaicing we can ideally have 360 by 180 degrees vision. The modified version of Intensity Based Image Mosaicing algorithm has been proposed. Through this algorithm seams that appear while stitching images are not eliminated. This algorithm works both for single viewpoint as well as multiple viewpoints. The implementation aspect of the algorithm is discussed along with test results. The modified version algorithm for Image Mosaicing using Euclidean Warp and Bilinear Interpolation has been used for the purpose of comparison. We have used our own image data sets for experimenting.

Key–Words: Image Mosaicing, Panoramic Image, Bilinear Interpolation, Euclidean Warp, Intensity Based Image Mosaicing

1 Introduction

Image mosaics are collection of overlapping images together with coordinate transformations that relate the different image coordinate systems. By applying the appropriate transformations via a warping operation and merging the overlapping regions of warped images, it is possible to construct a single image covering the entire visible area of the scene. This merged single image is the motivation for the term “mosaic”. Image mosaicing can be done in a variety of ways. There are many algorithms to do image mosaicing [2], [3]. Usually, the algorithms differ in the Image registration process.

The algorithm using ”Euclidean Warp and Bilinear Interpolation” [1] [5], used for the sake of comparison here is very simple and works by using Euclidean warping, using which the image registration parameters are extracted. Once the transformation matrix is formulated, the nearest neighbor interpolation technique is used.

The algorithm ”Intesnity Based Image Mosaicing” is proposed. The basic goal of the program is to stitch a series of flat images together to produce a continuous panoramic image (a mosaic) [4]. The difficulty here is that in order for the images to fit together some of the images must be warped (transformed) to adjust for the difference of perspective in the images. The initial estimate of the transformation is usually very imprecise and results in a very sloppy match between two images. That is an error minimization method (Marquardt-Levenberg) is employed for this purpose.

2 Image Mosaicing using Euclidean Warp and Bilinear Interpolation

After observing the problem domain, that is the specifications of the area on which we are working, assimilation of these basic fundamentals is essential for thorough understanding of the algorithm. We chose Euclidean warp of transformation and Bilinear Interpolation in estimating the missing values.

2.1 Euclidean image warp

Image warping is in essence a transformation that changes the spatial configuration of an image [1]. Using this definition, a simple displacement of an image by few pixels in the x-direction would be considered a warp. The euclidean warp is also called euclidean similarity transform involving four parameters as in equation 1.

$$P = [s, \alpha, t_x, ty] \tag{1}$$

- s= Scaling Factor,
- α= Rotation angle,
- t_x and t_y The translation in x and y direction respectively.

Let $p = (x, y, 1)$ denote a position in the original image, I and $p\prime = (x\prime, y\prime, 1)$ denote the corresponding position in the warped image, $I\prime$ (both in homogeneous coordinates). Looking at one pixel, we can write a simple linear transformation, equation 2. where T denotes the composite transformation matrix.

$$p\prime = T \times p \tag{2}$$

Instead of warping a single point we could warp the whole image of n points, equation 3.

$$P = \begin{pmatrix} x_1 & x_2 & \ldots\ldots & xn \\ y_1 & y_2 & \ldots\ldots & yn \\ 1 & 1 & \ldots\ldots & 1 \end{pmatrix} \tag{3}$$

Then equation 2 becomes equation 4.

$$P\prime = T \times P \tag{4}$$

Due the discrete nature of raster images, one is in no way ensured that each input pixel exactly maps to an out pixel. Consequently backward warping is mostly performed i.e. from the output image to the input image. Since T is square and has full rank we can easily compute the inverse transformation as in equation 5.

$$P = T^{-1} \times P\prime \tag{5}$$

In mosaicing, the transformation between images is often not known beforehand. Approximate estimation of the transformation required between two images to be merged is done. The estimated transformation is done by providing the points of correspondence in each of the images. In order to recover the transformation we rearrange the warping equation 2 so that the warping parameter is the vector t in equations 6 and 7.

$$p\prime = Z \times t \tag{6}$$

$$\begin{pmatrix} x\prime \\ y\prime \\ 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 & 0 & 0 \\ y & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} s \times \cos(\alpha) \\ s \times \sin(\alpha) \\ t_x \\ t_y \\ 1 \end{pmatrix} \tag{7}$$

The warping parameters are now obtained by solving the linear system equation 7. Therefore at least two points are required for solving. Therefore matching points on each image is to be estimated.

2.2 Bilinear Interpolation

Interpolation is defined as estimation of a missing value by taking an average of known values at neighbouring points. The more adjacent pixels included while interpolating, more accurate in estimation. But this comes at the expense of computational time. Bilinear interpolation includes best of both worlds, it is accurate as well as fast. Hence, Bilinear Interpolation is used here.

For bilinear interpolation, weighted average of two translated pixel values for each output pixel value is used.

2.3 Algorithm 1

The algorithm can now be summarized in the following steps.

1. Load two input images (color/grayscale)
2. Show input images and prompt for correspondence..
3. Choose two matching points in both the images in the same order.
4. Estimate parameter vector t using Euclidean Warping.
5. Construct the transformation matrix.
6. Warp incoming corners to determine the size of the output image (in to out).
7. Do backwards transform (from out to in).
8. Re-sample pixel values with bilinear interpolation.
9. Offset and copy original image into the warped image.
10. Show the result.

3 Intensity based image mosaicing

The basic goal of the program is to stitch a series of flat images together to produce a continuous panoramic image (a mosaic). The difficulty here is that in order for the images to fit together some of the images must be warped (transformed) to adjust for the difference of perspective in the images. The initial estimate of the transformation is usually very imprecise and results in a very sloppy match between two images. That is why we must employ an error minimization method (Marquardt-Levenberg). Some of the assumptions made are:

- The camera conforms to the pinhole model camera.
- The objects in the images are sufficiently far away to be approximated by planar surfaces.
- When defining the points of correspondence, four corresponding points are selected in exactly the same order in both images. Failure to comply with this assumption will result in unpredictable warping.
- While defining points of correspondence, any three points in the same image that lie approximately on the same line is not selected. Because the matrix would then be ill-formed as it is difficult to define a unique projective transform matrix based on four points three of which are on the same line. Failure to comply with this assumption will result in unpredictable warping.

3.1 To find projective transform for two images

To find a projective transform, four corresponding points are selected on both the images. If the set of points (x_i, y_i) and $(x_i\prime, y_i\prime)$ and for $1 <= i <= 4$ are corresponding points. Then the corresponding points can be represented in terms of via the 8-parameter projection matrix as in equation 8.

$$\begin{pmatrix} x_1\prime \\ x_2\prime \\ x_3\prime \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (8)$$

Equation 8 amounts to equation 9.

$$\begin{aligned} x\prime &= \frac{x_1\prime}{x_3\prime} = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}} \\ y\prime &= \frac{x_2\prime}{x_3\prime} = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}} \end{aligned} \quad (9)$$

The linear system of equations can then be created, involving all eight points is in equation with h_{33} is assumed to be 1. As pointed out in Single View and Two-View Geometry, there is a case for which the true solution for h_{33} is 0. Then, the column vector H will be a poor quality estimate of the actual transform matrix.

Based on just the points of correspondence the transform does not guarantee any kind of precision matching of features at one of the images with features in another image. In fact, the error can be quite large. This is overcome by Marquardt - Levenberg minimization to correct the values of the matrix.

3.2 To blend the images together

Any kind of blending of the two images is simply a weighted combination of the pixels of those images. If when blending two images we just assign equal weights to the pixels in the areas of overlap, that would produce a satisfactory result but unfortunately if there's even a very a small error in matching of the features then pixel mismatching at the edges would give away the fact that we are blending two different images. In order for the edges not to be visible, we want the intensities of both images to be weighted less as the pixel locations get closer to its edges. That way, when an averaged sum is computed to find out the final intensity of the image as in equation 10.

$$Intensity = \frac{(W_1 \times I_1 + W_2 \times I_2)}{(W_1 + W_2)} \quad (10)$$

- W_1 and W_2 are weights assigned to first and second image pixels respectively,
- I_1 and I_2 are intesnities of first and second images pixels respectively.

If the pixel location is closer to the center of one image but is near the edge in another image, then the pixel whose location is near the center of its image is will get a greater contribution in the blended image's pixel intensity.

3.3 Algorithm 2

The Algorithm can be summarized in the following steps:

1. Load the images.
2. The images are stored in an appropriate buffer.
3. The matching points are chosen, 4 in number in both the images.
4. The First image is distorted to match the features in Second image.
5. The transformation matrix is calculated.

6. The Marquardt Levenberg Minimization Algorithm is used to minimize the intensity difference and calculate the final transformation matrix.

7. Warp incoming corners to determine the size of the output image (in to out).

8. Do backwards transform (from out to in).

9. Re-sample pixel values with bilinear interpolation.

10. Offset and copy original image into the warped image.

11. The blending gradient is built.

12. The resultant images are mixed to get a complete image.

13. Show the result.

4 Results

4.1 Results for Mosaicing using euclidean warp and bilinear interpolation

4.1.1 Experiment 1

The input images are fig.1 (a) and fig. 1 (b), the mosaiced image is fig. 1 (c). The input images confirm to single view point and are taken with the camera being rotated at the optical center. On careful observation of the Mosaiced image , one can find that there is almost near perfect mosaicing. Seam is visible only near the lower right corner. The stone fountain, the benches, the lights etc., have been perfectly mosaiced, minor discrepancy exists in the upper right corner, where the trees have been mosaiced.

4.1.2 Experiment 2

The input images are fig.2 (a) and fig. 2 (b), the mosaiced image is fig. 2 (c). In this experiment , both the input images are of the same visual scene, with the exception that the second image is tilted about 60 degrees with respect to the first. Because of the tilting, there is addition of new features, such as some more part of the trees as well as the road come into the visual scene. In the Mosaiced image, the black region indicates the region where no mapping is done. This is considerably a large area due to the fact that the input images respect the same visual scene. The first image is taken as the base image, therefore it is found that the mosaiced image is also based on the first image. Hence the temple is not tilted in the Mosaiced image.

(a) First image

(b) Second image

(c) Mosaiced image

Figure 1: Fountain image

4.1.3 Experiment 3

The input images are fig.3 (a) and fig. 3 (b), the mosaiced image is fig. 3 (c). Seams are clearly visible in the Mosaiced image. The images are taken using a single point of view with the camera being rotated about its optical center.

4.1.4 Inference

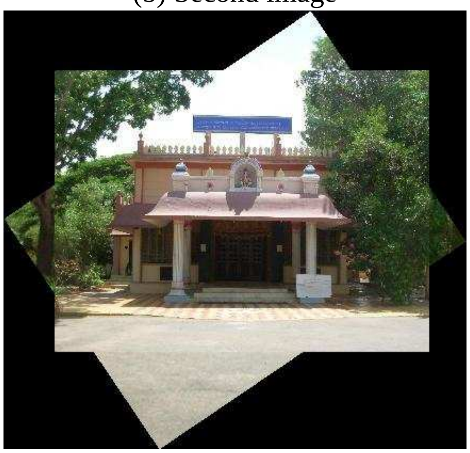
From this we can come to the conclusion that:

1. It is not necessary that seams need be present always.

2. Seams arise due to the minute intensity differences between the two images; since the algorithm does not include a blending scheme to minimize the intensity difference.

(a) First image

(b) Second image



(c) Mosaiced image

Figure 2: Temple image

3. This paves the way for algorithm Intensity Based Mosaicing.

4.2 Intensity Based Image Mosaicing

4.2.1 Experiment 4

The input images are fig.1 (a) and fig. 1 (b), the mosaiced image is fig. 4. The input images confirm to single view point and are taken with the camera being rotated at the optical center. Comparing fig. 4 with fig. 1 (c) one can find that there is perfect mosaicing. No seam is visible. The stone fountain, the benches, the lights etc., have been perfectly mosaiced.

(a) First image

(b) Second image

(c) Mosaiced image

Figure 3: Entrance image

Figure 4: Mosaiced fountain image

4.2.2 Experiment 5

The input images are fig.2 (a) and fig. 2 (b), the mosaiced image is fig. 5. The black region indicates the region where no mapping is done. This is considerably a large area due to the fact that the input images respect the same visual scene. Comparing 5 and 2 (c) in the mosaiced image. In the implementation of the first algorithm we have taken the first image as the base image, therefore we find that the Mosaiced image is also based on the first image. Hence the temple is not tilted in the Mosaiced image. But here the image is tilted as the second image is the one being manipulated and not the first one.

Figure 5: Mosaiced temple image

Figure 6: Mosaiced entrance image

4.2.3 Experiment 6

The input images are fig.3 (a) and fig. 3 (b), the mosaiced image is fig. 6. There is no seam at all in fig 3 (c). Looking closely at the middle of the mosaiced image, one could see slight blurring. This is due to the blending which has taken place. Also, the mosaiced image is not perfectly rectangular due to the usage of perspective transformation, unlike algorithm 1 which used similarity transformation.

4.2.4 Inference

From the results presented we can come to the conclusion that:

1. Seams are not present.
2. Perfect Mosaicing is taking place.
3. This algorithm works not just for single view point but for multiple view point but only with translation.

5 Conclusion

The results explained in the previous sections show that:

1. Our algorithms do the requisite work with the advantage of being simple and easy.
2. What we have done requires minimal human intervention unlike other approaches.
3. We have worked on color images and gray scale.
4. We have worked on both single point of view images and multiple point of view with translation, scaling as well as rotation.
5. Ultimately, we can conclude that we have designed and implemented two unique algorithms for construction panoramic image mosaics which provide wide, all-encompassing views, even exceeding human vision.

References:

[1] Heckbert Paul, *Fundamentals of Texture Mapping and Image Warping*, Masters thesis, CS Division, U.C. Berkeley, June 1989

[2] Richard Szeliski and Heung-Yeung Shum, *Creating Full View Panoramic Image Mosaics and Environment Maps*, Computer Graphics (SIGGRAPH'97 Proceedings), pages 251-258, August 1997.

[3] Richard Szeliski, *Image Alignment and Stitching: A Tutorial*, Foundations and Trends in Computer Graphics and Vision Vol. 2, No 1 (2006) pages 1104

[4] Getian Ye, *Image Registration and Super-resolution Mosaicing*, Doctoral thesis School of Information Technology and Electrical Engineering, Australian Defence Force Academy University College, The University of New South Wales September 2005

[5] Mikkel B Stegmann, *Image Warping*, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, October 2001