

Use of Affine Transformation For Registration of Image Sets

Connor McCullough, University of Miami

Abstract—This paper covers transformation of an image related to another through affine registration, so the two can be concatenated into a panoramic image. The process involves finding a set of matching points in the two images based on mutual features, using these points to generate a transformation matrix, transforming all the points in the first image to a destination image, interpolating the values to the integer indices in the new image, and overlapping the final two images. While some imperfections may arise from use of the affine model, the effects of these may be reduced by crossfading the overlapping regions of the two final images.

Keywords—registration, transformation, affine, interpolation, panorama, stitching

I. INTRODUCTION

Image transformations have a variety of applications, including removal of distortion caused by lenses, as well as image stitching to create panoramic photos. The two main classes of linear transformations are affine transformations, which have six degrees of freedom, and projective transformation, which have nine degrees of freedom. The images in this project are related via affine transformations, which are made up of the operations of rotation, translation, magnification, and shear. The process of stitching images together to create a panorama consists of mapping features in both images and determining their coordinates in each, determining a transformation matrix from use of at least 6 points, transforming the entire image based on this transformation, interpolating the values for non-integer index points, and creating a composite of the two images.

II. CHOOSING REFERENCE POINTS

The coordinates of mutual features in both images were chosen randomly using the `ginput` function in MATLAB, which creates an array of points that the user has clicked on. Points must be chosen from a variety of locations all over the image, not just locally, to obtain the optimal transformation matrix. Because there are six variables in the affine transformation matrix, at least 6 points must be chosen so that all the variables can be solved for. However, additional points may be used for increased accuracy.



Fig. 1. Matching Points Mapped onto both images

III. CREATING TRANSFORMATION MATRIX

Each of the sample points in the original image and image to transform to are stored in separate arrays, made up of a row of X coordinates, Y coordinates, and ones, giving two $3 \times N$ matrices. The two matrices are related via product of the 3×3 transformation matrix:

$$\hat{X}_{3 \times N} = H_{3 \times 3} X_{3 \times N} \quad (1)$$

By solving for H using the right pseudo-inverse, the following matrix equation is obtained:

$$H = \hat{X}_{3 \times N} X_{N \times 3}^T (X_{3 \times N} X_{N \times 3}^T)^{-1} \quad (2)$$

Because the matrix H is an estimation of the affine structure, it may not be completely accurate. A more accurate, but more difficult to implement approach would be to use Singular Value Decomposition. When the transformation matrix was calculated, a test matrix was created by multiplying the resulting transformation matrix H by the original image coordinate matrix X, and calculating the difference between the result of this test, and the original destination coordinate matrix Xhat. The result was errors of up to 10 pixels in a 750x1000 image, or a percent error of 1.33%. This is due to both the fact that not every single pixel can fit the matrix, due to the manual error of plotting points by hand and the deviation from the affine model. Also, the method of solving for H is merely an estimation that may not be entirely accurate.

IV. MAPPING POINTS TO NEW IMAGE

The final step is to use the transformation matrix to map all the points from the original image into its destination. Due to use of discrete indeces in MATLAB, problems may arise when mapping coordinates into the new image. A value in the original image may be mapped to a decimal coordinate in the destination. Therefore, the value at the indeces must be interpolated based on the surrounding known values.



Fig.2. Second image, and transformed first image before stitching

Last, there must be a method of overlapping the images so they smoothly transition between each other. While proper technique in obtaining the matrix and transforming the image will create a smooth transition, there may inevitably be some inaccuracies due to the image's deviation from the affine model, and changes in perspective. For example, in the test image, the placement of the bushes seen through the lattice arch is different, so the images could not possibly line up 100% accurately. However, there are work arounds for this issue. A simple solution to concatenating the overlapping portions of the two images is to multiply the intensity of the two images by .5 wherever they overlap.



Fig. 3. Stitched images with overlapped regions multiplied by .5.

However, this makes it easy to see where the two images transition between each other and between overlapping. A better solution would be for a smooth transition between a single image and overlapping image, and to reduce the area that the two images are overlapping at equal ratios. To solve this, a mesh grid based on the normalized distance from the center of the original image (0) and the border of the far edge of the overlapping region (1) was created. A second matrix was created that is the first matrix subtracted from 1. This creates a crossfade of the two overlapping regions and ensures that the overall “gain” of the two is 1, so that the brightness is not affected.



Fig. 4. Stitched images using crossfading function.

CONCLUSION

By using an affine transformation matrix, two images with matching features can be registered together. This process consists of mapping coordinates of matching features to generate a transformation matrix, mapping all the points in the original image to its destination using the transformation matrix, using interpolation to map to integer indeces, and crossfading the overlapping portions of the image together. Several points of innacuracy were determined throughout the project, mostly in the generation of the matrix. However, by properly crossfading the overlapping regions of the image together, some of the effects of these errors can be made less visible.