# SATELLITE IMAGE REGISTRATION AND IMAGE STITCHING

Dhanyamol Antony
Dept. of Computer Science
SCT College of Engineering
Trivandrum, India
dhanyamolantony@gmail.com

Subu Surendran
Dept. of Computer Science
SCT College of Engineering
Trivandrum, India
subusurendran@gmail.com

*Abstract*— **Satellite image registration and stitching system is used to create panoramas of satellite images. Panorama is a picture made by combining a series of photos into one large picture so that we will get a full view of an area. The paper proposes a stitching technique based on image registration. Here we geometrically align one image onto another image of the same scene taken at different time, from different viewpoints or by different sensors. Image stitching algorithm take the alignment estimates produced by the registration algorithm and blend the images in a seamless manner. It also deals with potential problems such as blurring or ghosting caused by parallax and scene movement as well as varying image exposures.**

**Keywords- Image registration; Corner detection; Random sample consensus (RANSAC), Alpha Blending**

## I. INTRODUCTION

Panoramic image stitching has wide applications in the areas like literature and commercial applications. Image stitching system will modify the perspective of images and blend them, so that the photographs can be aligned seamlessly. Image stitching methods are classified into two categories- direct and feature based methods [1]. In this paper we describe a stitching method based on image registration, depends on feature extraction. A large number of image registration algorithms have been developed over recent years. These algorithms are of two categories based on matching methods- Area based methods (ABM) and Feature based methods (FBM) [2]. The template matching is a common area based methods. This method uses a distortion function that measures the degree of similarity between the template and the image. Typical distortion measures include Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC), etc. Taking into account the robustness, researchers often adopt NCC. Since the basic template matching algorithm need to be calculated at each positions of the reference image, the computational time is very long.

Fast template matching [8] improved the speed of the template matching by using Fast Fourier Transform (FFT) algorithm. In feature based methods, registration algorithms are combinations of the following three components: feature points detection, feature points matching, and calculation of transformation matrix. The advantages of feature points are low computational complexity, fast calculation, insensitivity to illumination changes and is also invariant to image rotation and scale. So most systems use image registration based on feature points. Here we propose a feature based image stitching system for panorama creation of satellite images.

The rest of the paper is organized as follows. In section II we describe the panorama creation algorithm and the different steps involved in it. Section III will give an overview to the result obtained from experimental results. In section IV, we present conclusions and ideas for future work.

## II. PANORAMA CREATION

Panorama creation has mainly four different steps: feature extraction, feature matching, homography estimation and blending. These four steps can be summarized as follows:

**Algorithm: Panorama creation**
Input: n satellite images

1. Apply image processing techniques to remove the noise or blurring caused during image acquisition.
2. Extract interest points from all n images.
3. For each pair of images:
    a. Select ‘m’ matching images that have the most feature matches to the current image

b. Select most matching pair of features between the two images.
c. Find geometrically consistent feature matches using RANSAC to solve for the homography between pairs of images
d. Find connected components of image matches
e. Render panorama using alpha blending

Output: Panoramic image

### II.1 *FEATURE EXTRACTION*

The first step for panorama creation is to detect the feature points. The most appropriate ways to identify the feature points in two-dimensional space are corner detection algorithms. Lots of modifications of such algorithms exist. They differ in accuracy, detection rate, localization, repeatability rate, robustness to noise, speed etc.

Here we assume corners and edges as feature points. These points are known as interest points. A large number of corner detectors are available nowadays and Harris corner detector [3] is used here. The advantages of Harris detector over other detectors are simple computation, stability of the extracted feature points etc. One of the first corner detectors was developed by Hans P. Moravec in 1977. Moravec defined the concept of "points of interest" in an image and concluded these interest points could be used to find matching regions in different images [9]. Moravec called the feature points as interest points, points where there is a large intensity variation in all direction. It could also be used to identify distinct regions in images.

Harris detector was developed by Chris Harris and Mike Stephens in 1988, a modification to Moravec operator. It helps to identify both corners and edges in the image frames. Harris corner detector is based on the local auto-correlation function of a signal; where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions [3]. Here it will calculate the differential of corner score with respect to direction. If the intensity changes in all direction with respect to a point, it indicates a corner and if in any one direction indicates an edge. Auto-correlation function will relate a matrix A and averages the derivatives of the signal in a window around a point (x,y).

A(x,y ) =

$$\begin{bmatrix} \sum_{w} (\mathrm{I_x}(\mathrm{X_k}, Y_\mathrm{k}))^2 & \sum_{w} \mathrm{I}_x(\mathrm{X_k}, Y_\mathrm{k})\mathrm{I_y}(\mathrm{X_k}, Y_\mathrm{k}) \\ \sum_{w} \mathrm{I}_x(\mathrm{X_k}, Y_\mathrm{k})\mathrm{I_y}(\mathrm{X_k}, Y_\mathrm{k}) & \sum_{w} (\mathrm{I_y}(\mathrm{X_k}, Y_\mathrm{k}))^2 \end{bmatrix}$$

where I(x,y) is the image function and $(x_{k,}y_k)$ are points in the window W around (x,y).

The Eigen values of this matrix will provide a good indication to detect whether a particular point is a flat, edge or a corner point. If this matrix has a large Eigen value then an interest point is detected Now the rank of matrix A can be calculated. Matrix with rank one and rank two indicates an edge and a corner respectively [4].

### II.2 *FEATURE MATCHING*

The feature extraction will results in a number of interest points. The images should be matched somehow using these interest points. Maximum correlation rule might be applied for this purpose. The peculiarity of feature point is that it can be differentiated from its neighboring image points. It helps to match corresponding points in another image. ie, the neighborhood of a feature should be different from the neighborhoods obtained after a small displacement[5]. A second order approximation of the dissimilarity between an image window and a slightly translated image window is given by

$$D(\Delta x, \Delta y) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \mathrm{M} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \rightarrow \mathrm{(I)}$$

$$\mathrm{M}=\iint_{W} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \mathrm{w(x, y)dxdy} \rightarrow \mathrm{(II)}$$

In order to ensure that no displacement exists for which D is small; the eigen values should both be large. This can be achieved by enforcing a minimal value for the smallest eigen value or alternatively for the following corner response function $R=detM-k(trace\ M)^2$. Here k is set to 0.04 (a suggestion of Harris). The quality of feature is ensured by tracking the neighborhood. In the case of tracking, it should be ensured that the features can be tracked from one video frame to another. Here each window of pixel around every feature point of first image is analyzed and matched with every feature point in the second image. Points which have maximum correlation are taken as corresponding pairs.

In practical situations a large number of feature points may be extracted. In this case we have to limit the number of points selected from each image, before match them. This is achieved by only selecting the corners with a value above a certain threshold. This threshold can be tuned to yield the desired number of features. In some scenes most of the strongest corners are located in the same area, in such cases we have to ensure that every part of the image has sufficient number of corners. However some points might be wrongly matched. The algorithm can be summarized as follows:

1. Convert the two images into grayscale.
2. Generate correlation matrix
3. Select points with maximum correlation measures
4. Construct the lists of matched point indices
5. For each point in the first set of points.
    a. Get the point in the second set of points with which the point in first image has a maximum correlation measure.
6. Extract matched points.
7. Create matching point pairs.
8. Stop

*II.3 HOMOGRAPHY ESTIMATION*

Feature matching will result in two images with correlated points. At this point we have to define a model which can translate points from one set to the other. Some kind of image transformation called projective transformation is applied for projecting one of the two images on top of the other. A homography matrix matching the two images is used for this purpose. Homography matrix is a 3×3 matrix consists of homogeneous coordinates with 8 degrees of freedom[10].

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

Using homogeneous coordinates each pixel position $< x, y >$ is represented as a tuple in the form, $< x\ , y, w >$ where w is the scale parameter. Here we fix 'w' as 1 for simplification. i.e.

$$I_{xy} = < x, y, 1 > \rightarrow \text{(III)}$$

Projective transformation is implemented by using simple matrix multiplication by using homogeneous coordinates as follows.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \rightarrow \text{(IV)}$$

Once all projected points are computed, original system can be obtained by dividing each point by its homogeneous scale parameter and then dropping the scale factor.

$$\langle \frac{x', \quad y', \quad w'}{w'} \rangle = \langle \frac{x'}{w'}, \quad \frac{y'}{w'}, \quad 1 \rangle = \langle x'', \quad y'' \rangle \rightarrow \text{(V)}$$

Homography matrix is constructed by using set of correlated points. RANSAC (RANdom SAmple Consensus) method might be used for this purpose [6]. It estimates parameters of a mathematical model from a set of observed data which contains outliers. It is a global transform model; identify the wrong correspondence in the images. Infact each image contains an overlapping area and a non-overlapping area whose shape must be preserved [6].

*II.4 RANSAC ALGORITHM*

The maximum correlation rule might result in some outliers, the points which do not fit in an estimated mathematical model. This might severely affect the estimated transformation and must be avoided.

RANSAC algorithm handles this problem by classifying the points into inliers and outliers [6]. A threshold $\tau$ is fixed in order to ensure that all inliers are fit in the model and not deviated from the model than $\tau$. The procedure for RANSAC described in [7] can be abridged as follows:

1. Select randomly the minimum number of points required to determine the model parameters.
2. Solve for the parameters of the model.
3. Determine how many points from the set of all points fit with a predefined tolerance.
4. If the fraction of the number of inliers over the total number points in the set exceeds predefined threshold $\tau$, re-estimate the model parameters using all the identified inliers and terminate.

5. Otherwise, repeat steps 1 through 4 (maximum of N times).

The important parameter to be initialized is the distance threshold $\tau$ for which a point is declared inlier or outlier. Without knowledge of the distance of the outliers, the only way to determine this parameter is by experimentation [7]. The number of iterations N can be determined theoretically by the following formula,

$$1-\rho = (1-\omega^{n})^{N} \quad \rightarrow \text{(VI)}$$

*II.5 ALPHA BLENDING*

Once the images are aligned and unwanted objects removed, the two images have to be blended. Alpha blending can be used here. This is done from the center of one image to the other. A translucent foreground color is combined with a background color, thereby producing a new blended color. The degree of the foreground color's translucency may range from completely transparent to completely opaque. If the foreground color is completely transparent, the blended color will be the background color. Conversely, if it is completely opaque, the blended color will be the foreground color. Of course, the translucency can range between these extremes, in which case the blended color is computed as a weighted average of the foreground and background colors.

Alpha Blending can be accomplished by blending each pixel from the first source image with the corresponding pixel in the second source image. The equation for doing the blending is,

**Final pixel = alpha * (First image's source pixel) + (1.0-alpha) * (Second image's source pixel)**

Here the value of alpha is the blending factor specifies percentage of colors from the first source image used in the blended image. All the pixels in the above equation should be in RGB888 format, or 8 bits for red, 8 bits for green, and 8 bits for blue color. If the bits from most significant to least significant are laid out, it would result in

$$\overset{bit\ 23 \qquad\qquad\qquad\qquad\qquad bit\ 0}{\text{RRRRRRRRGGGGGGGGBBBBBBBB}}$$

The outline for blending method in [11] is as follows:

1. Select the two images say foreground and background.
2. For each pixel:
   a. If source pixel is transparent, just return the background.
   b. Else calculate the final pixel value using the formula:
      *Final pixel = alpha * (First image's source pixel) +(1.0-alpha) * (Second image's source pixel)*
      *alpha = alpha_bg + alpha_fg –*
      *alpha_bg * alpha_fg*
3. Stop

## III. RESULTS AND DISCUSSION

Our image stitching system is well suited for all types of images including the satellite images. It will stitch arbitrary number of images in a seamless manner. The system supports images of different formats like JPEG, TIFF, GIFF, PNG etc. It does not perform very well on images with very different lighting conditions. In order to avoid this problem, we should normalize the two images before applying the method. The processing time of the proposed system vary with the size of image. The performance can be evaluated by comparing the execution time of various images with different size and formats. Table 1 shows such a comparative study conducted on a system with configuration: Intel Core i5 processor, x64-based PC with a 4GB RAM.

| Image type | Image format | Image size | Execution time |
|---|---|---|---|
| Camera image | JPEG | 60kb | 1.29sec |
| Camera image | GIFF | 62kb | 1.13sec |
| Satellite image | TIFF | 12MB | 47.53sec |

## IV. CONCLUSION AND FUTURE WORK

This system is well suited for the automatic panorama creation for a pair of satellite images. However, as it stands, it can be used to create full panoramas using an arbitrary number of images. It could also be used to create image mosaics in which the correct order of the images is not known beforehand, by considering a minimum threshold of correlation between two images before attempting a match. This way, only images with strong correspondences are stitched at a time, and a full mosaic could again be constructed recursively.

The system could be extended to solve "*multiple image super-resolution"* problem. The problem is to create panoramas with a higher resolution by taking multiple images of the same scene with sub-pixel displacements. It could also be extended for video stitching. This is a straightforward generalization of multiple-image stitching, the potential presence of large amounts of independent motion, camera zoom and the desire to visualize dynamic events impose additional challenges.

## V. REFERENCES

[1] R. Szeliski. Image alignment and stitching: Autorial. Technical Report MSR-TR-2004-92, Microsoft Research, December 2004.
[2] Gang Hong, Yun Zhang. Combination of feature-based and area-based image registration technique for high resolution remote sensing image[C].Geoscience and Remote Sensing Symposium,Barcelona, pp. 377-380, July 2007.
[3] The Harris Corner Detector Konstantinos G. Derpanis kosta@cs.yorku.ca October 27, 2004
[4] Evaluation of interest point detectors Cordelia Schmid, Roger Mohr and Christian Bauckhage Inria rhˆone-alpes, 655 av. De l'europe, 38330Montbonnot, france cordelia.schmid@ inrialpes.fr
[5] Feature matching, "http://www.cs. unc.edu/~marc/tutorial/node51.html"
[6] Thomas Colleu1, Jian-Kun Shen2, Bogdan J. Matuszewski2, Lik-Kwan Shark2, Claude Cariou1 Feature-Based Deformable Image Registration with RANSAC Based Search Correspondence.
[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004
[8] Lewis J P. Fast Template Matching[C].Canadian Image Processing and Pattern Recognition Society. Quebec, pp. 120-123, May 1995.
[9] Moravec Operator, "http://kiwi.cs.dal.ca/~dparks/ CornerDetection/moravec.htm"
[10] Homography Estimation by Elan Dubrofsky B.Sc., Carleton University, 2007
[11] Alpha Blend Algorithm, "http://www.daniweb.com /software-development/c/code/216791/alpha-blend-algorithm"