Semi-Automated mosaic algorithm for close range photogrammetry

A.A.SHAKER* M.M.M. HASAN** N.I. ELGENDY***

1. Abstract

Image registration has a lot of applications in the photogrammetric community. The traditional techniques depends on the choice of the matched points manually., which is tedious less accurate and time consuming Other techniques depends on the automatic search of these matched pointes which is accurate and fast but has the lack of accurate choice if the image has some typical features which needs to manually delete these wrongly matched features This paper present a new technique that correlates typical features through a guided searching technique at the overlapping area, then applying some geometrical constraints for these detected features. This technique is proved to be faster and need no data editing after matching. Because it only examines two potential small parts of the two images rather than processing the hole image each time to correlate each potential feature in the base image. The site is at Kingston upon Hull –UK for two typical buildings having the same architecture and colors.

2. Introduction

The registration problem is the process by which a newly captured (input) image is transformed to match the orientation and scale of the previously captured (base) image. Image registration requires an intensive computational complexity on a very huge number of data represented by the resolution of the image which requires a lot of memory space to solve. For this reason a high performance registration techniques is of special importance especially with continuous increase of the image resolution. Image registration also requires the existence of common points in the two images. These common points either matched manually, which is time consuming and less accurate or automatically which is accurate and fast.

There are several techniques for image registration could be found in the literature and categorized in [1] & [4]. All of these literatures are covering three basic elements related to the registration process. The three basic elements of the registration method considered on these techniques are, feature type, search strategy and similarity measure.

In this paper the feature is a huge number of corners at the base image produced using Harris corner detector. For speeding up the calculation, this huge number of corners is reduced through an algorithm to pick only the corners lies on the edges of buildings as produced by Canny's edge detector as discussed in [5]. The search strategy is done through a geometrically guided search.

---

* Prof., Shoubre Faculty of Engineering.
** Lecturer, Shoubre Faculty of Engineering.
*** Lecturer, Shoubre Faculty of Engineering.

This geometrically guided search is achieved by determining the relative position of the two images relative to base image coordinate system. Then, cropping a pre defined small window size from the base image at each corner (sub image1) and a bigger one at the nearly corresponding position at the input image (sub image2). The second window should be larger enough to contain the feature picked at the base image. The similarity measure is the normalized cross correlation function within the Matlab signal processing toolbox. It will be used at two stages in this algorithm. First, to locate the relative position of each two successive images, second, to correlate the feature included in sub image1 at the base image to the features included in sub image2 at the input image by sliding neighborhood operations as discussed in the next section.

3. Program Description

The program should run first at the folder that contains the images will be processed. The images names should always ended by serial numbers representing the sequence in which they will be processed Then it will start by asking the user about the name of the image series by prompting the user with (Enter the image series name and extension [ ]:). The user should write the name if the image series names excluding the end numbers i.e. HPIM "the camera used is an HP digital camera" and the extension (e.g. jpg) separated by comma and between two square brackets. In this instant it will read all the images begin with the series name and have the extension written at the prompt.

The program will then prompt with (Enter the numbers of the images will be processed [] :). If the user wants to process a random numbers of images each pair of them has overlap area, he should write these capturing sequence numbers separated by spaces and between square brackets. An Enter keystroke without writing any numbers will process all images that have the same name and extension with the sequence they already saved with. It will then display the first two images in the image sequence as shown in Figure 1. The user will then be asked to crop an area out from the input image prompting the user with (Please crop any part at the overlapping area gives best correlation'). This process, cropping, limits the search to this area which will save calculation time.

As mentioned earlier, the similarity measure used will be the Matlab normalized cross correlation function. The restriction for using this function is that one of the two images matrices should be smaller in both dimensions than, and should also be included in the other image matrix for the correlation to be meaningful [5]. A part of image2 (sub image2) should be marked using the mouse. This part should be uniquely defined in both images, should not be extremely changed by changing the camera position or obstructed by a target not exists in the first scene and smaller in both dimension than image1. Nevertheless the program will keep prompting with the same prompt until a correlation of 60% (experimentally decided) between the chosen part and any part of image1 exist.

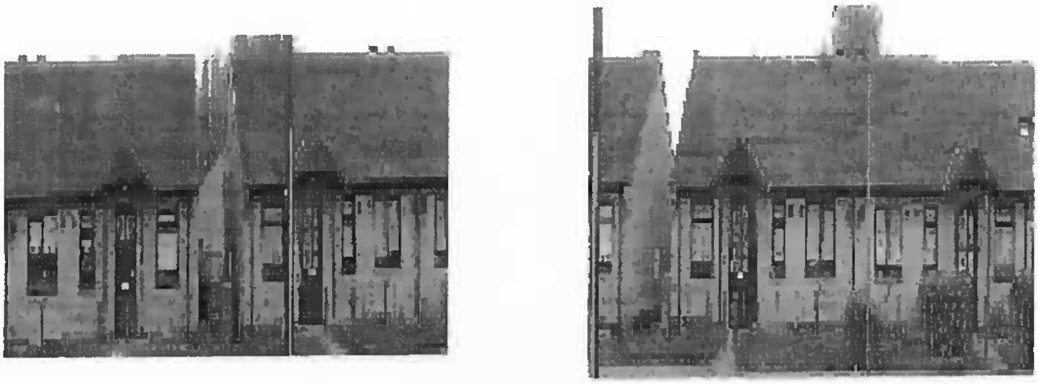


Figure 1: Base images (right) and Input image (left)

At the correlation operation, the program correlates the sub image2 matrix with all matrices combinations at the base image that equal in size to sub image2 using sliding neighborhood operation. A sliding neighborhood operation as seen in Figure 2 could be visualized as moving the sub image2 hanged from its center on each pixel at the base image in a column wise direction. The value of each pixel in the output matrix (correlation matrix) is calculated as the correlation value of the corresponding input pixel's neighborhood and base pixel's neighborhood. The neighborhood is the number of pixels, surrounding that pixel, (central pixel) forming either a rectangular or square block

Assuming the base image matrix as shown in Figure 2 (left) is a 6x5 and sub image2 is a 2x3 matrix (bold rectangle at the right) , the center of the 2x3 matrix neighborhood will be correlated to each pixel in the 6x5 matrix by the way shown in Figure 2 with its center marked with a dot. For any *(m, n)* neighborhood matrix, the center pixel is positioned at row $m+1/2$ and column $n+1/2$ rounded towards $-\infty$. This means that the center of sub image2 will be the pixel (1,2).If any of the rows or columns on the base image sub matrices needs padding to match the dimension of the input sub image2 matrix it will be padded by zeros. The padding in our analogy will be applied when correlating sub image2 matrix to any sub matrix at imge1 with its center lay at the first column, last columns and last row. The dimension of the correlation matrix using this analogy will be 7x7 to compensate for the padding of one row and two columns. The number and position of the padding rows or columns could be computed from the following equations followed by their respective values in the example:

Pre padding rows = center row-1 =1-1=0
Pre padding columns = center column-1 =2-1=1
Post padding rows = m-center row =2-1=1
Post padding columns= n-center column =2-1=1

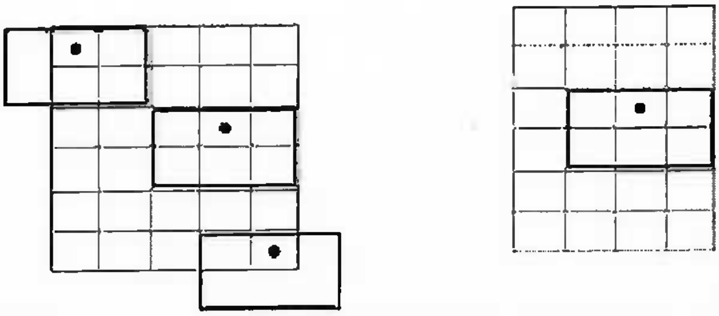


Figure 2 :6x5 base image matrix (left) and 2x3 input image matrix (right).

Knowing the position of maximum correlation at the correlation matrix relative to its coordinate system and the position of the input sub image2 matrix center relative to the input image coordinate system, an approximate estimation of the two elements shift vector (x_o, y_o) is formed and the position of input image relative to the base image could be calculated. At the analogy above suppose that this 2x3 sub image1 is matched with the second bold rectangle (middle right) with its center at row 3 and column 5 in correlation matrix. The shift vector will be

x_0=5-2 -Pre padding columns = 2

y_0=3-1- Pre padding rows =2

Figure 3 shows sub image2 as chosen by the user and the matched part of the base image both are appended in one figure according to their approximate orientation to each other.

Figure 3:Subimage2 (left) and Matched sub image1 (right)

The offset calculated above is used to shift the coordinate of the input image to the base image coordinate system. The intersection polygon representing the overlapping area polygon (a) as shown in Figure 4 could be calculated as

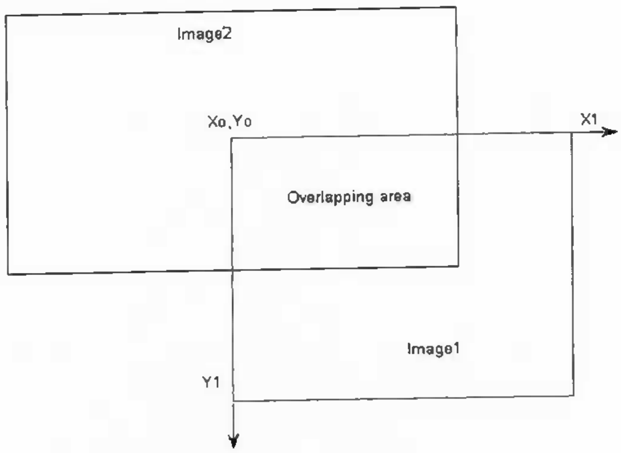


Figure 4 Overlapping area

$$a = I\left([\vec{x}_1,\vec{y}_1],[\vec{x}_2,\vec{y}_2]\right)$$

Where:

$\vec{x}_1 = [1, n_1, n_1, 1]$

$\vec{y}_1 = [1, 1, m_1, m_1]$

$\vec{x}_2 = [x_o + 1, x_o + n_2 - 1, x_o + n_2 - 1, x_o + 1]$

$\vec{y}_2 = [y_o + 1, y_o + n_2 - 1, x_o + n_2 - 1, y_o + 1]$

m_1, n_1 **: The number of rows and columns at image1 respectively.**

m_2, n_2 **: The number of rows and columns at image2 respectively.**

Figure 5 The program displays the two overlapping sub images (ovr_image1 and ovr_image2) after cropping them automatically from the original images using the above equations and appended in one figure.

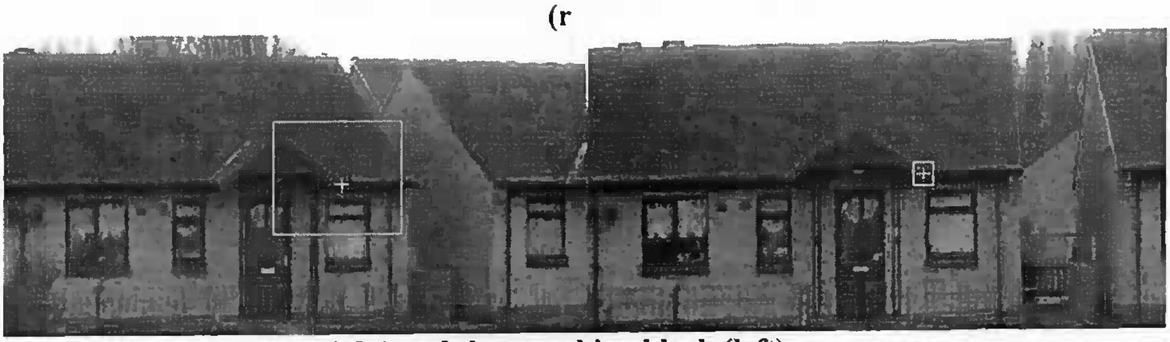
Figure (5): The two overlapping sub images ovr_image1 (right) and ovr_image2 (left).

To increase the speed of the program, the tracking stage will be done only on the overlapping areas. At this stage the program will locate the features corners at ovr_image1. Harris's corner detector has been used for this purpose. To decrease

the huge number of corners detected and to choose some corners easy to be tracked, only the points at the edges of ovr_image1 will be processed. The edge points are easy to track because since they are detected on the edge of one image it will be most probably detected at the other image.

Figure 6 shows the neighborhoods included inside a window with a pre specified size (13x13 pixels) around each one of the corners detected at ovr_image1(marked by + on the right).These neighborhoods will be the feature required to be tracked at ovr_image2 (the same as sub image2 above). Since the orientation of the ovr_image2 is now approximately known, a block larger in size in both dimensions than the first window with its center at a pre computed azimuth and distance from the window center at ovr_image1 will be containing the feature we searching for (the same as sub image1 above). Any feature in the searching block matches the feature included in the first window under a certain correlation threshold decided by the user (80% in this paper) will be the matching feature (marked with + on the left).

Figure 6: Feature window to be tracked
(r



ight) and the searching block (left)

Figure 7 shows the matched points in both images after the correlation process connected by lines

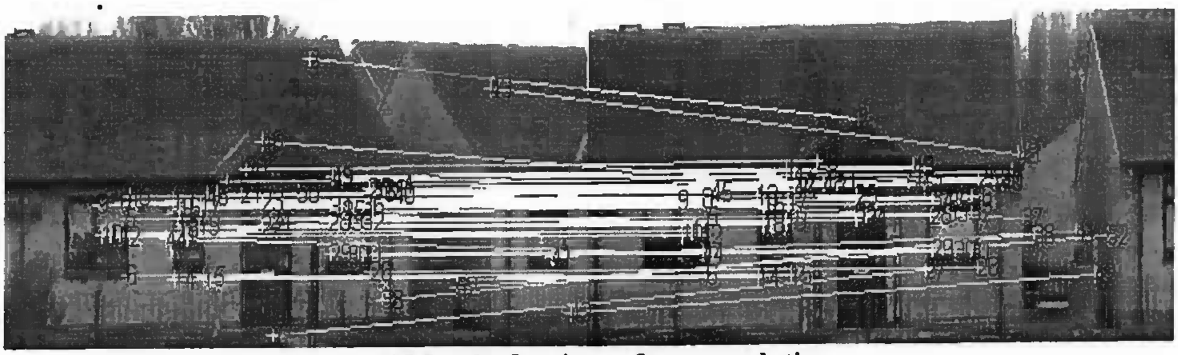


Figure 7 Matched points after correlation

The program in order to perform the above stage will prompt the user by [Enter the minimum correlation threshold []:]. The user is either type a number different than 80 or press the Enter key to proceed with the 80% correlation threshold.

As seen in figure 7 there are some outliers or wrongly tracked corners. To remove these outliers the program will proceed with the following data clustering process. In this process, the azimuths and distances between each two corresponding points are computed. Then bins both the azimuths and distances into three equally spaced containers and return the number of elements included in each container. The azimuths or distances that are not within the maximum number of points cluster will deleted and accordingly the points they are computed from. The standard deviation for the azimuths or distances within the maximum number cluster will then be computed. Since both images were taken from two different view points, the azimuths and distances of the lines connecting the matched points might not be equal. As a criterion of removing outliers in this paper, any azimuth or distance exceeds seven times the standard deviation will be considered an outlier and will also be removed from this cluster. Figure 8 shows the histogram for both the azimuths (left) and distances (right) in our case with the vertical axis represents the number of observation and horizontal axis represents the azimuth in degrees and distances in pixels respectively. Figure 9 shows the final points after clustering. The above procedure will be repeated if two groups of azimuths or distances obtained. The condition for taking the second cluster is that the number of azimuths or distances exceeds 60% the number of the first cluster or equal to it. This case might occur if the camera position at one image was not at the center of the scene resulting in difference scale between the beginning and end of the image and the other image is the same or nearly orthogonal.

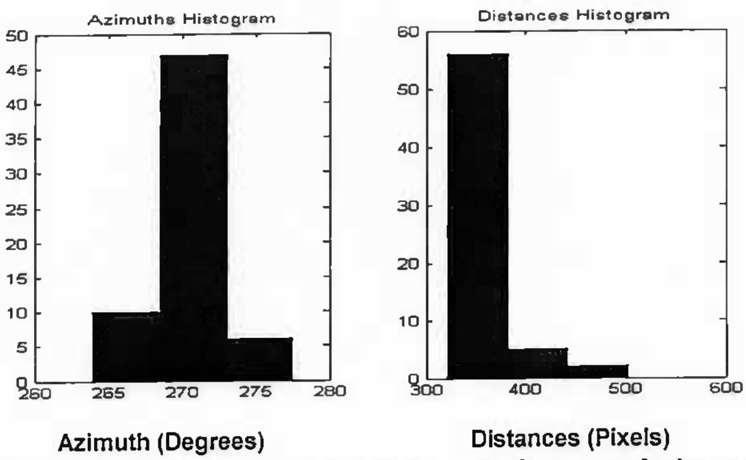


Figure 8: Histogram of azimuths and distances after correlation matching.

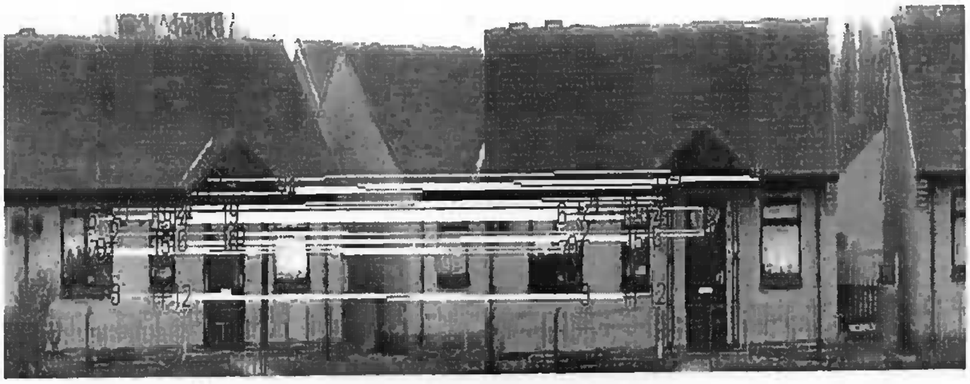


Figure 9: Matched points after azimuth and distance threshold.

The coordinates of the final matched points after clustering are shifted again to follow the coordinate system of the original images as shown in Figure 10 and Figure 11. The percentage of the finally matched points with respect to the chosen corners at the base image is 40% with a mean correlation of 92%

Figure 10: Image1 with the matched points overplayed

Figure 11: Image2 with the matched points overplayed

Figure 12 shows the resulting mosaic after performing the 2-D spatial transformation of the input image to match the orientation and scale of base image. The projective transformation as in [5] is used with the above matched points are used as common points

Figure 12 Mosaic of the two images

4. Conclusion and future work

This paper presents an efficient algorithm for high performance semi automatic images mosaic. This has been attained by reducing the time of registration through reducing the search data by dealing with sub images rather than processing the hole images. The sub images selection is performed through a geometric guided search that searches for the feature at a pre computed position. The similarity measure used is the normalized cross correlation. The resulting matching points are then subjected to a clustering process based on histogram analysis

The algorithm was evaluated for feature tracking on real images and the results shows that almost 40% of corners detected at the base image are matched with 92% mean correlation value Most of the rejected features are not similar at the two successive images due to the change of the camera position

Future wok will be directed towards solving the problem of the existence of multiple positions for the maximum correlation at the correlation matrix. This case arises when there are exactly similar features at one image and accordingly in the other image. This will convert this algorithm to an automatic one because it is the only step which needs the user interaction

It would be also one of the tasks to develop an automatic block size (at the input image) selection algorithm. The process by which the block size is minimized at each matched point detection process until the block size at image2 equals the window size (feature) at image1. This will serve two purposes, first the search will be close to the corresponding feature which will make the program predict the correct feature, and second, it will also affect the efficiency and speed of the algorithm by reducing the time of the correlation process.

The azimuths and distances clustering of the lines connecting the matched points are based on a histogram clustering using a pre defined number of clusters; this part of the program should be changed to automatic clustering.

Finally, a complete investigation of this tracking method requires a more evaluation by comparing it experimentally with those proposed in the literature and through using a large number of image sequences with different capturing situations.

5. References

[1] Bruce D. Lucas and Takeo Kanade (1981),. An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence, pages 674-679,.

[2] Carlo Tomasi and Takeo Kanade.(1991), Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132,

[3] Jianbo Shi and Carlo Tomasi.(1994), Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600.

[4] Prachya Chalermwat (1999),. High performance Automatic Image Registeration For Remote sensing). PhD theises, Georrg Mason University_Fairfax,Verginia.

[5] The Math Works, Inc (2004), Matlab Image Processing Toolbox, the Math Works, Inc., Natick, MA