AUTOMATIC TIE-POINT GENERATION FOR OBLIQUE AERIAL IMAGERY: AN ALGORITHM

Seth Weith-Glushko, Graduate Student
Carl Salvaggio, Associate Professor
Rochester Institute of Technology
College of Science
Chester F. Carlson Center for Imaging Science
54 Lomb Memorial Drive
Rochester, NY 14623-5604
saw7167@cis.rit.edu
salvaggio@cis.rit.edu

ABSTRACT

Tie points are pixels that describe the geographic position of the same information in two or more images. They are used in a variety of applications such as geometric image transformations, image mosaicking and three-dimensional model generation. Unfortunately, present methods used to select tie-points are computationally inefficient and prone to error. Moreover, the methods are best suited for imagery taken at nadir. This study examines the feasibility and effectiveness of a new tie-point generation algorithm that utilizes inertial navigation system data and the Laplacian of Gaussian spatial filter. Initial results show that the algorithm can provide matches with sub-pixel accuracy in images that exhibit changes in rotation, scale, and translation. The algorithm can also produce matches that exhibit small errors when applied to images that have the distortions mentioned above plus changes in perspective.

INTRODUCTION

The Digital Imaging and Remote Sensing Laboratory at the Rochester Institute of Technology has created a new algorithm that will automatically generate tie points, or corresponding pixel locations, among two images collected from different sensing positions. To test the feasibility and effectiveness of the algorithm, prototype code was generated in the IDL programming environment. Five test data sets containing oblique aerial imagery were created. Each data set contained two images with a different geometry configuration including a set that exhibits a translation change in the vertical direction, a translation change in a horizontal direction, a mild perspective change (less than a 30° change in orientation), a severe perspective change (roughly a 90° change in rotation) and an excessive perspective change (roughly a 180° change in rotation). If appropriate matching points were generated across each data set, the algorithm can be considered feasible. To test the effectiveness, the algorithm was executed using varying maxima LoG errors from 10% to 25% and geophysical coordinate errors from 10% to 90%. After execution, quality metrics were determined.

The first quality metric to be applied was a visual assessment of the final matched points generated for the two images. By marking points in the image where matched points occurred, a qualitative determination of the effectiveness can be made. The second quality metric consists of a number of plots used for visualization of the data. These plots present the number of generated point sets, the number of matched point sets generated after the point matching has executed, the number of matched points after the full algorithm has executed, and the mean root-mean-squared distance error after algorithm completion, as a function of LoG threshold. For the sake of brevity, only the plot of the mean root-mean-squared distance error will be presented.

THEORY

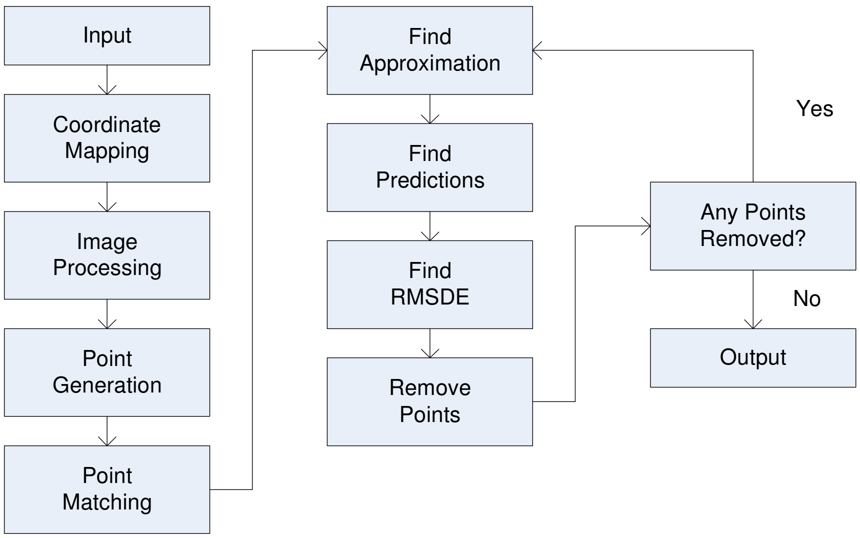


Figure 1 - Overview of the baseline algorithm

The baseline algorithm to generate tie points consists of a series of intermediary stages where small processing tasks take place. An overview of the algorithm can be seen in Figure 1. The input consists of two images that exhibit a geometric change. Both images are mapped to their representative geophysical coordinates (i.e. latitude and longitude). Image processing is performed to enhance the effect of the Laplacian of Gaussian (LoG) filter. Point generation occurs by applying the LoG filter to each image and thresholding the results at an arbitrary value. The thresholded images are run through the point matching algorithm. A point is considered matched if the difference between the geophysical coordinates and LoG values at corresponding pixels are within a user-specified error tolerance.

The main algorithm then enters an iterative process. The polynomial approximation problem is solved to match the candidate points from image one to the candidate points in image two. The root mean square distance error (RMSDE) is calculated for each matched pixel set. Matched points with an RMSDE greater than the mean RMSDE plus one standard deviation of the set of matched pixels are removed. If any matches are removed, an additional iteration is carried out; otherwise, a list of matching points is output.

Input and Coordinate Mapping

Input for the algorithm consists of two true-color images. Once the images are read in, geophysical coordinate mapping occurs. Geophysical coordinate mapping involves calculating the latitude and longitude of an imaged point associated with a pixel's x- and y-coordinate values.

Image Processing

Once coordinate mapping has been performed, image processing takes place. Image processing must be done to make each image similar in overall brightness distribution for the purpose of normalizing the data to increase the point generator performance. There are two kinds of image processing performed: color space reduction and histogram specification. If the two input images are true-color, the equation below is employed to transform those images to grayscale. This equation comes from the NTSC standard for luminance. (Mai, 2000) The grayscale transformation is carried out since the Laplacian of Gaussian operator is not applicable to multi-band imagery.

$$Y_{x,y} = (0.3 * R_{x,y}) + (0.59 * G_{x,y}) + (0.11 * B_{x,y}) \quad (1)$$

Y represents the luminance image; R, G, and B represent the digital count value for each pixel coordinate in the red, green and blue color channels of the image.

Histogram specification involves transforming the histogram of one image to match the histogram of another. Figure 2 highlights this process. The histogram is taken for both images. Treating the normalized histogram as the probability distribution function, a cumulative distribution function (CDF) is calculated. The mathematical derivations for the PDF and the CDF are seen in Equations 2 and 3.

$$PDF(x) = \frac{n_x}{N} \quad \text{where} \quad x = 0,1,2,...,255 \tag{2}$$

$$CDF(x) = \sum_{a=0}^{x} PDF(a) \quad \text{where} \quad 0 \le a \le 255 \tag{3}$$

n_x represents the number of pixels that have the digital count specified by x while N represents the total number of pixels within the image.

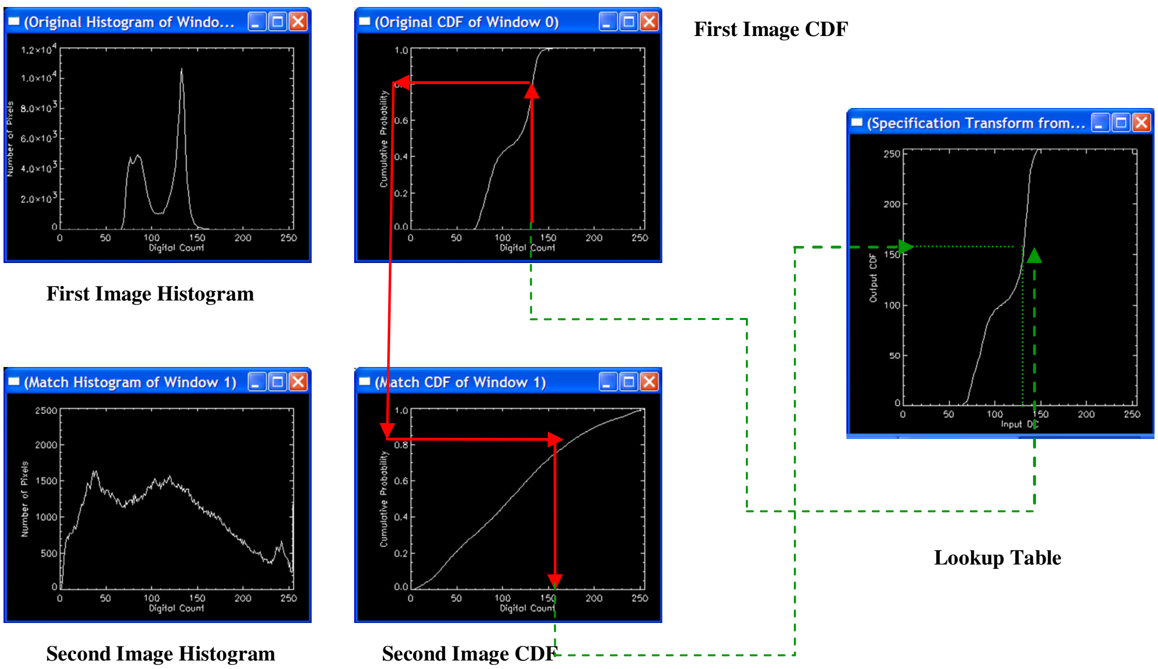


Figure 2 - Histogram specification processing

Using an arbitrary input grayscale value, an output grayscale value is calculated using the path shown in Figure 2. A lookup table is formed by following this path for each digital count of the input image and applied to the image to be transformed. The result is images with similar grayscale value distributions, independent of image structure. (Gonzalez et al., 2002) It is important to note that although image content is changed, these changes are discarded once the matched points have been found.

Point Generation

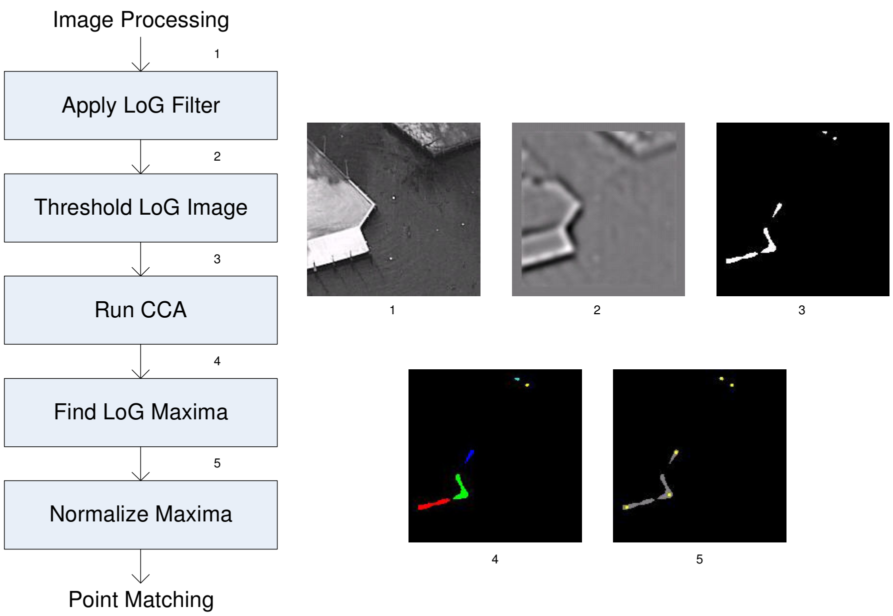


Figure 3 - Overview of the point generation stage

The point generation follows the logic flow shown in Figure 3. The first step is the application the Laplacian of Gaussian (LoG) spatial filter to an image. The LoG spatial filter is not a single filter that can be convolved with an image. Instead, it is the application of a Gaussian spatial filter followed by the Laplacian spatial filter. The second step is a threshold of the LoG image at an arbitrary value. This arbitrary value is image-independent and affects the number of generated points in this stage. As such, the value will need to be modified by user on an image-to-image basis. An appropriate value can be found by executing the algorithm and lowering the threshold value if too few points are generated. Walli found that by applying a threshold to an LoG image, points with high frequency detail (i.e. edges) could be isolated and defined. The theory behind this practice is that these points will be similar across images of a common point. (Walli, 2003)

The third step involves the execution of connected components analysis (CCA) on the resulting bi-level image. Connected components analysis is a process by which ``blobs'' of maximum-valued digital counts are categorized into a region. A blob is any collection of non-zero valued pixels surrounded by zero-value pixels. ``Blobs'' are identified and pixels that belong to the ``blobs'' are marked as belonging to that ``blob.'' A visualization of this process can be seen in Figure 4.

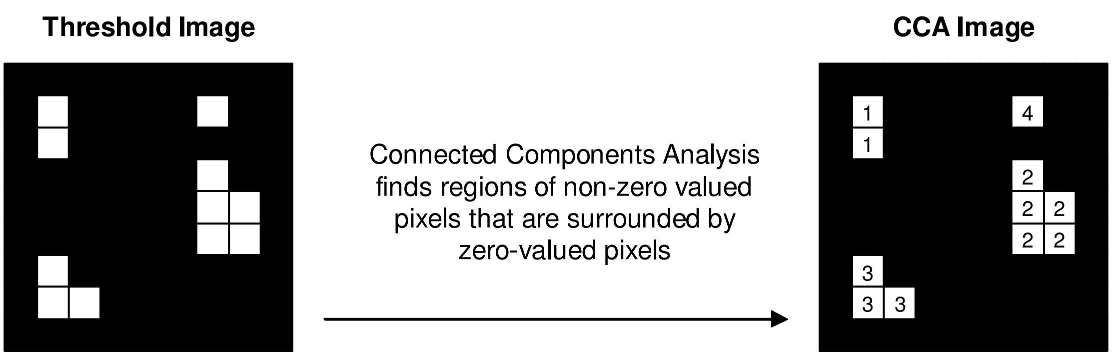
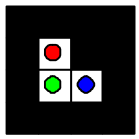


Figure 4 - An example run of connected components analysis

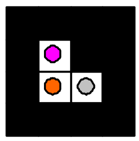
The fourth step is a sequential search of each region defined by connected components analysis. Within each region, the pixel with the highest absolute LoG value is found and marked. Finally, those pixels that have been marked have their LoG values normalized and are passed to the next stage of the process. An example of this process can be seen in Figure 5.

Region 1



- Pixel Location: (14,12); LoG Value: 239.45
- Pixel Location: (14,13); LoG Value: 1246.26 **MAXIMUM**
- Pixel Location: (15,13); LoG Value: 573.50

Region 2



- Pixel Location: (134,120); LoG Value: 985.43
- Pixel Location: (134,121); LoG Value: 987.04 **MAXIMUM**
- Pixel Location: (135,121); LoG Value: 535.49

Hence, after normalization with largest LoG value of the maximums (1246.26),

Point 1 – Pixel Location: (14,13); LoG Value: 1
Point 2 – Pixel Location: (134,121); LoG Value: 0.792

Figure 5 - An example of the sequential search

Point Matching

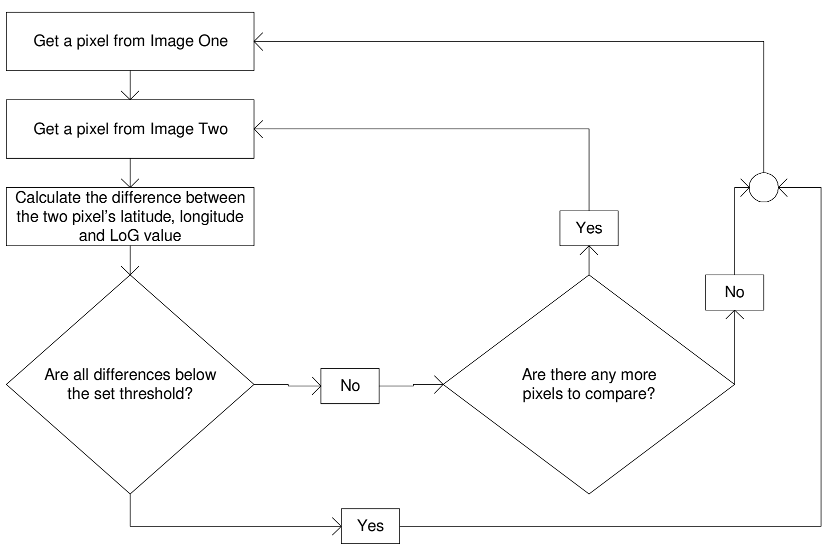


Figure 6 - An overview of the pixel comparison process

Point matching occurs using an iterative process, as seen in Figure 6. Every point in the first image is compared to every point in the second image. For each candidate point set, three differences are calculated. The first calculation finds the difference between the latitude associated with the first candidate point and the latitude associated with the second candidate point. The second calculation finds the difference between the longitude associated with the first candidate point and the longitude associated with the second candidate point. The third calculation finds the difference between the LoG value associated with the first candidate point and the LoG value associated with the second candidate point. If each difference value is within an arbitrary error value, the pixel is considered matched.

Once this process has completed, it is possible to have multiple pixel matches in the second image for one pixel in the first image. If this is the case, the average x- and y-coordinates of all of the points in the second image is calculated. The average coordinates are used as a match to the pixel in the first image. This process is illustrated in Figure 7.

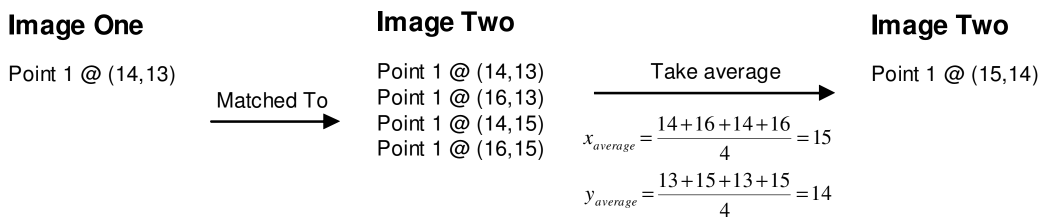


Figure 7 - An example of the averaging process

Find Approximation, Predictions and RMSDE

The final stage in the tie-point generation algorithm is an iterative process. The first step is a geometric transformation of the coordinates in one image to register it with the other image. A global polynomial distortion model is used to perform the transformation. Mathematically, this model is defined as Equations 4 and 5.

$$x_m = a_{00} + a_{10}x_{ref,m} + a_{01}y_{ref,m} + a_{11}x_{ref,m}y_{ref,m} + a_{20}x^2_{ref,m} + a_{02}y^2_{ref,m} \tag{4}$$

$$y_m = b_{00} + b_{10}x_{ref,m} + b_{01}y_{ref,m} + b_{11}x_{ref,m}y_{ref,m} + b_{20}x^2_{ref,m} + b_{02}y^2_{ref,m} \tag{5}$$

x_m and y_m represents matched image coordinates in one image; x_{ref} and y_{ref} represent matched image coordinates in the other image; and a_{nm} and b_{nm} represent constants to be solved for. n and m are indicative of the powers of the x and y terms seen in the equations, respectively. It is important to note that there are unique equations for each matched point set. As such, there are multiple linear equations. These equations can be written in matrix form as seen below.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} 1 & x_{ref,1} & y_{ref,1} & x_{ref,1}y_{ref,1} & x^2_{ref,1} & y^2_{ref,1} \\ 1 & x_{ref,2} & y_{ref,2} & x_{ref,2}y_{ref,2} & x^2_{ref,2} & y^2_{ref,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{ref,m} & y_{ref,m} & x_{ref,m}y_{ref,m} & x^2_{ref,m} & y^2_{ref,m} \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{11} \\ a_{20} \\ a_{02} \end{bmatrix} = W \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{11} \\ a_{20} \\ a_{02} \end{bmatrix} \tag{6}$$

or,

$$\vec{X} = \vec{W}\vec{A} \tag{7}$$

$$\vec{Y} = \vec{W}\vec{B} \tag{8}$$

These constants can be solved for by using the matrix inverse and using the minimum number of required matched points (9) as below:

$$\vec{A} = \vec{W}^{-1}\vec{X} \tag{9}$$

$$\vec{B} = \vec{W}^{-1}\vec{Y} \tag{10}$$

or using the pseudo-inverse when there are more than 9 matched points. (Walli, 2003)

$$\vec{A}' = (\vec{W}^T\vec{W})^{-1}\vec{W}^T\vec{X} \tag{11}$$

$$\vec{B}' = (\vec{W}^T\vec{W})^{-1}\vec{W}^T\vec{Y} \tag{12}$$

As part of the algorithm, the previous two equations will be used more extensively because more than the minimum number of points is desired. After the matrices have been defined, the transformation is applied to one image to register it with the other.

Once the transformation has taken place, a quality metric called the root mean square distance error (RMSDE) is calculated. Here, the geometrically transformed points are compared against the matched points in the first image using the equation below.

$$RMSDE = \sqrt{(x_{predicted} - x_{matched})^2 + (y_{predicted} - y_{matched})^2} \tag{13}$$

$x_{predicted}$ and $y_{predicted}$ represent the predicted coordinates from the transform while $x_{matched}$ and $y_{matched}$ represent the coordinates found from the point matching algorithm. Next, first order statistics (mean and standard deviation) are performed on all RMSDE values calculated.

Remove Points and Output

Using these first order statistics, a cut-off RMSDE value is calculated as the mean RMSDE value plus one standard deviation. If any matched point sets have an RMSDE greater than this value, the matched point set is considered not matched and removed from further consideration. If there were point sets that were deleted and more than 18 matched point sets remain, an iteration has been completed and starts again with a new geometric transformation using the remaining matched point sets. Otherwise, the iterative process is complete and a list of matching points is output.

RESULTS

Five test data sets were created. The five data sets exhibit geometric changes such as horizontal translation (Test Case 4), vertical translation (Test Case 5), slight perspective change (Test Case 6), severe perspective change (Test Case 7) and excessive perspective change (Test Case 8). For each of these test data sets, a plot of mean RMSDE at the completion of the algorithm as a function of LoG threshold was generated. To generate these results, the algorithm was run on a Toshiba Satellite notebook computer, containing a 2.8 GHz Pentium 4 Mobile processor and 512 MB of RAM. Four runs of each test case were made: 10% LoG threshold baseline, 25% LoG threshold baseline, 10% LoG threshold variant and 25% LoG threshold variant.

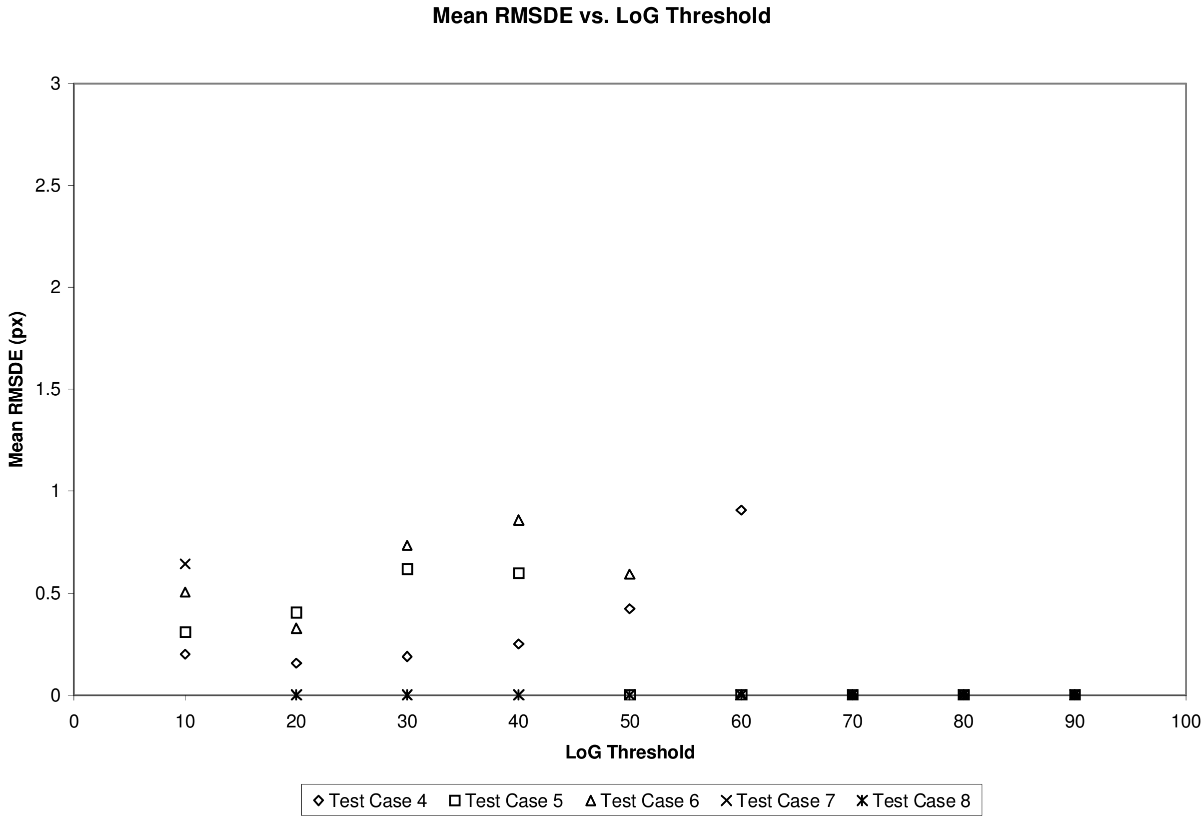


Figure 8 - Mean RMSDE as a function of LoG threshold, 10% maxima error

DISCUSSION

A number of trends become evident when viewing the data. First, as the LoG threshold increases, the number of generated and matched points decreases. Likewise, the mean RMSDE at the completion of the algorithm increases. This is due to the fact that as the threshold is increased, there are fewer points to mitigate the higher errors in certain point sets. Another interesting trend concerns the type of geometric change there is within each set. As the perspective changes more, or there is less of a translation change within the imagery, the smaller the range in which a LoG threshold can lie in order for point matching to occur. This can be seen as one progresses from Cases 5 to 8.

Coinciding with this trend is an apparent increase in the final mean RMSDE of each set. As one progresses from Cases 4 and 5 which exhibit sub-pixel accuracy (mean RMSDE ~ 0.2), one moves on to Cases 7 and 8 which start to exhibit severe errors (mean RMSDE ~ 1 and greater). This is best exemplified by visual analysis of each set. In Cases 4 and 5, the matched points coincide perfectly across the two images. Continuing on to Case 6, the matched points continue this pattern, however, in localized regions of the image, there is more than sub-pixel error present. This trend continues in Case 7 where there are a number of regions with error present. Finally, in Case 8, the errors in the matches become extremely evident. Upon closer examination, one notices that pixels that lie in the ground plane are correct but pixels exhibiting any significant elevation do not. Due to the coordinate mapping process, which is projective, latitudes and longitudes are assigned to those pixels as if they were physically located at ground level. In the case of the buildings, these pixels obviously do not reside on ground level and as such, are prone to error.

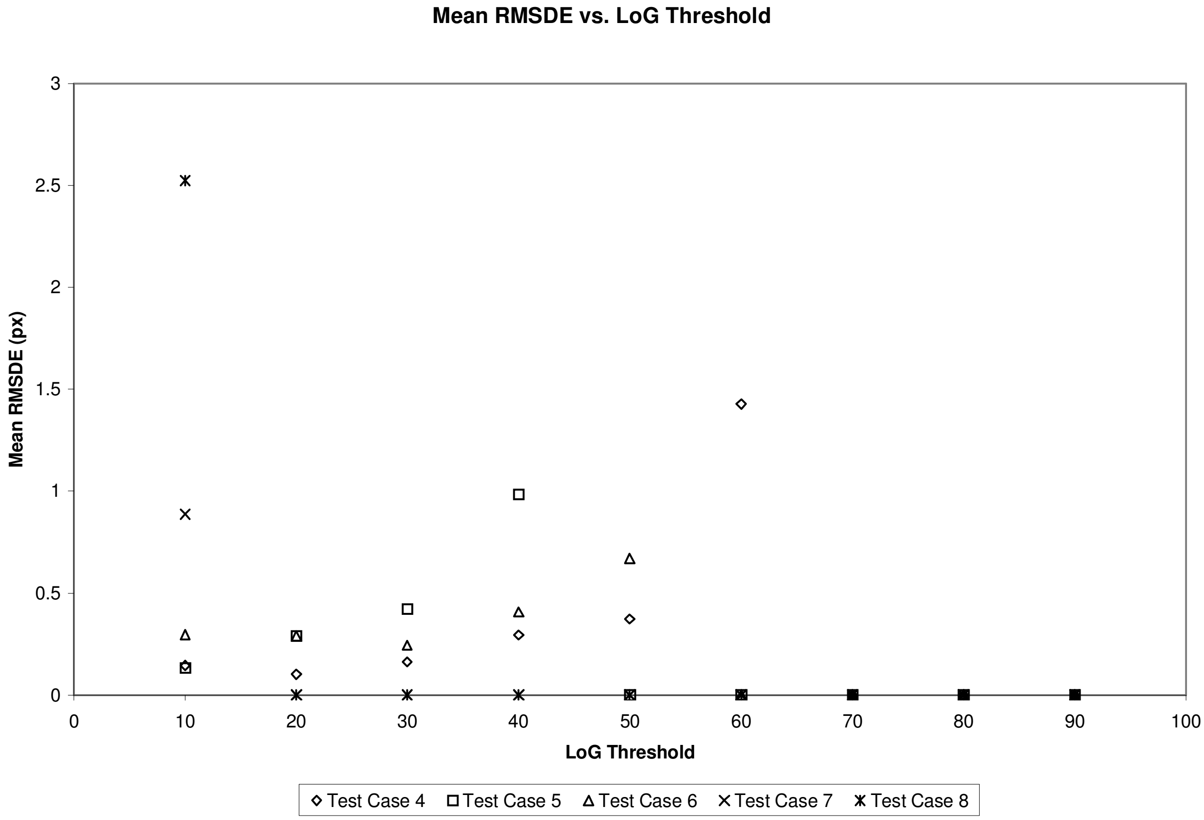


Figure 9 - Mean RMSDE as a function of LoG threshold, 25% maxima error

CONCLUSIONS

All in all, the tie point generation algorithm has proven to be feasible and effective. First, it is computationally efficient. With execution times that span from one to twenty minutes, the algorithm proves feasible within this regard. Second, it has shown that matched points with sub-pixel accuracy can be generated on images that exhibit changes such as translation and perspective. However, care must be taken when executing the algorithm on imagery that exhibit excessive perspective changes. Due to the image content, the number of matched points made could be less than the number desired.

REFERENCES

Gonzalez, Rafael C. and Richard E. Woods (2002). *Digital image processing*, 2nd Ed., Prentice Hall, Upper Saddle River, New Jersey, 200 p.

Mai, Luong Chi., 2000. *Introduction to computer vision and image processing*, URL: http://www.netnam.vn /unescocourse/computervision/12.htm, Institute for Information Technology, Vietnam (last date accessed: 24 Feb 2004).

Salvaggio, C. (2003). *Digital image processing I notes*, URL: http://www.cis.rit.edu/people/faculty /salvaggio/courses/1051-461/1051-461.pdf, Rochester Institute of Technology, Rochester, New York (last date accessed: 18 May 2003).

Walli, Karl C. (2003). *Multisensor image registration utilizing the LOG Filter and FWT*, Masters Dissertation, Rochester Institute of Technology, Rochester, New York, 200 p.