# AUTOMATIC REGISTRATION OF SATELLITE IMAGERY BY MATCHING LINEAR FEATURES 

Mohamed Ibrahim Zahran<br>Associate Professor of Surveying and Photogrammetry<br>Faculty of Engineering at Shoubra, Benha University


#### Abstract

This paper presents a proposed approach of automatic image to image registration based on matching edge segments. Here, edge segments are extracted from the two images and their signatures are generated using chain coding. Segment signatures are utilized in a matching procedure to find common parts of edge segments. Cross correlation matching is then employed to refine the positions of midpoints of those common parts in the two images. The matching results can then used to compute the transformation parameters necessary for the transformation of the image to be matched.

The test images consist of two satellite images of different platforms and with different resolutions. One image is a QuickBird image with 0.6 m ground sample distance and the other image is a SPOT-5 image with a 2.5 m ground sample distance. The images cover the same area which is a somewhat urban area with insignificant height variation. The results reached by applying the proposed approach on the test images have proved the validity and efficiency of matching edge segments in automatic image-to-image registration.


Keywords: Image Registration, Chain Coding, Edge Detection, Image Matching, Correlation Techniques.

## 1. INTRODUCTION

In many cases, images of one area that are collected from different sources must be used together. To be able to compare separate images pixel by pixel, the pixel grid of each image must conform to the other images in the data base. The tools for rectifying image data are used to transform dissimilar images to the same coordinates system. Registration is the process of making an image conforms to another image [2]. A map coordinates system is not necessarily involved. To carry out image-to-image registration automatically, corresponding features in both images are to be found in an automated procedure, which is known as image matching in digital photogrammetry. The common features in the two images, resulted by matching, can then be used to determine the transformation parameters required for the registration process. The image to be registered is finally transformed from its own coordinate system into the coordinate system of the reference image. Resampling by using any of the known resampling methods would be needed during the transformation process.

Matching is the most fundamental problem in digital photogrammetry that is required in most photogrammetric procedures. In the matter of fact, matching is a complicated problem so that there is no general solution that works well on all types of imagery, different object spaces and varying illumination conditions. Several matching methods have been proposed. The main difference is the selection of matching features and the way their similarity is determined.

Area-based matching, or Intensity-based matching, is one of the most popular methods. This method determines the correspondence between two image areas according to the similarity of their gray level values [4,11]. The cross correlation and least squares correlation techniques are well-known techniques for area based matching. In the first technique, a small image patch is selected in one image and compared with image patches of the same sizes within a search window of the other window. The maximum cross-correlation factor serves as the similarity criterion. In least-squares matching, the gray level differences between template and matching windows are minimized by applying a transformation to the matching window. The matching is done in an iterative procedure; the parameters calculated during the initial pass are used in the calculation of the second pass and so on until an optimum solution is determined.

In feature-based matching, features are extracted from images before they are matched. The process of matching is usually performed by comparing feature attributes or descriptors such as orientation, gradient, shape, etc [9,12]. Topological and geometrical relations among features help constraining the large space of possible mappings among the features. Symbolic matching refers
to methods that compare symbolic descriptions of image features. Symbolic descriptions can be implemented as graphs, trees or semantic nets relating derived image features $[1,10]$.

## 2. FEATURE-BASED MATCHING

In contrast to intensity-based matching, feature-based matching does not compare gray values directly rather it compares derived quantities, referred to as features that may include points, lines, regions, or such abstract quantities as moments. Features are more suitable matching primitives than gray values, leading, in general, to more robust matching methods. For example, less strict conditions apply for approximate locations of conjugate matching primitives. Moreover, features are more closely related to objects which eventually appear on a map than the original gray values. A gray value can mean anything. However, an edge may refer to an object boundary and as such is more meaningful in terms of the desired end product.

Feature-based matching proceeds in three steps. First, appropriate features are extracted from the images. The second step is to find corresponding features based on similarity and consistency criteria. Similarity depends on the feature attributes whereas consistency depends on the degree to which the mapping function is satisfied. The mapping function describes the relationship between the images to be matched. The last step is to check the consistency of the matching results. The various feature-based matching techniques follow these three steps but differ with respect to extracted features, similarity and consistency measures, and mapping functions.

Point features are usually referred to as interest points. The popularity of point features in photogrammetry stems from the fact that the matching is easier than that of more complex features. Interest points can be characterized in terms of distinctness, stability, invariance, uniqueness and interpretability [1]. An interest operator satisfying these criteria is usually implemented in three steps. First, optimal windows in the image are located based on the average gradient magnitude, for example. Second, the image function within the selected windows is categorized by using statistical tests. Third, optimal points are located within the selected windows using suitable estimation techniques. Finding conjugate interest points is usually performed by intensity-based matching.

A line is richer in information than a point. Extracting lines begins with detecting local edges which are then aggregated into more globally defined lines using various grouping criteria. Local edges are detected using an edge operator. Many edge detection methods are described in the literature. There are several
differential edge operators which are either first difference operators or second difference operators [7,8].

Since intensity changes occur at different scales in an image, a good edge detector should satisfy two criteria. First it should be a differential operator, taking either a first or a second derivative of the image. The other criterion is the capability of acting at any desired scale. The Laplacian of the Gaussian (LoG) combines the scale dependency with differentiating the image [5,6]. A Gaussian function serves to smooth the image. The Gaussian is limited in the spatial and frequency domain, which is its most unique characteristic compared to other smoothing functions. As a differencing operator, the Laplacian is chosen. The two operations are combined and the result is the LoG operator. Edges are detected at the zero crossings, i.e., the locations in the convolved image at which the sign changes. Although the LoG has many merits it still suffers from two major drawbacks. Dislocalization is one problem. In the case of non-linear edges the zero crossings will be displaced at the corners of the edge [10]. Another problem arises from the property that the zero crossings form closed contours. Therefore, a problem occurs when edges intersect.

Another popular edge operator was devised by Canny [8]. It is similar to the LoG but belongs to the class of directional derivatives. Canny imposes three criteria for an ideal edge operator. The first criterion is good detection, i.e., low probability of wrongly marking non-edge points and low probability of failing to mark real edge points. The second criterion concerns good localization, i.e., points marked as edges should be as close as possible to the center of the true edges. Thirdly, for a single edge point, only one response should be triggered. In order to extract the matching primitives, Canny's edge detector is used in this paper to find edges in the two images under consideration. Here, the image is convolved with the Gaussian function. The first directional derivative of the convolved image is computed. Upon that the gradient magnitude at each point of the convoluted image is calculated. A non-maximum suppression is performed in the direction of the gradient. The resulting edge image is thresholded in order to eliminate false edges. At last, a fine to coarse technique can be applied to mark additional edges.

Edge operators produce an output at each pixel. To produce discrete edges, this output must be thresholded and the pixels over the threshold labeled as edge pixel. Real image edges usually cause a region of pixels around the edge to be over the threshold. A thinning operation is thus employed so that only the maximum response in the direction perpendicular to the edge is called an edge [4]. Extracted edges are seldom continuous due to image noise and scene scraps. A linking step fills in small edge gaps based on the examination of pixels in the gap or on heuristics such as gap size.

Following the edge thinning and linking processes, a suitable representation or description of edges is to be generated utilizing attributes of edge points. A known method to represent a boundary line is using discrete boundary codes, a coding method of continuous contour with a sequence of numbers, each number corresponding to a segment direction [3,5]. This representation is compact, invariant to translation, but depends on rotation and scaling transformations. However, the difference in the code reflects the turning angles, which are invariant to the transformations. In tangential representations $(\psi-s)$, the tangent $\psi$ angle is encoded as a function of arc length $s$. This is similar to differential chain codes, but don't have to be pixel-to-pixel. In radial representations $(r-s)$, the distance $s$ from the center is encoded as a function of arc length $s$.

Two interesting description methods of closed contours are the Fourier theorybased method and the Moment theory-based method [8]. The Discrete Fourier Transform (DFT) or the Fourier Series (FS) are generally used to describe the shape feature from its boundary. They give a sequence of complex coefficients called Fourier descriptors. These coefficients represent the shape of an object in the frequency domain where the lower frequencies symbolize its general contour, and the higher frequencies represent the details of its contour. Fourier descriptor representation is compact, accurate and also invariant to geometric transformations. The Moment theory-based method uses region-based moments to characterize the contour of an object with a set of parameters that are invariant to geometric changes. Different matching strategies can then be applied on those edge descriptors.

## 3. DISCRETE BOUNDARY ENCODING

One popular approach for representation or description of extracted edges is the use of discrete boundary codes. This encoding is similar in concept to the use of curvature, in the sense that it develops a local measure of the boundary orientation as the curve is traversed (i.e. as a function of curve length). Thus, given a set of discrete boundary orientation (and perhaps length) primitives, a polygonal representation of the boundary is used to generate the code. Since discrete samples are considered here, the resultant parameterization is in terms of a sequence or chain of discrete descriptors, or chain codes.

Referring to Figure 1, a piecewise linear approximation to a digital edge segment is developed using a set of orientation-only primitives. Therefore, the chain encoding approach is similar to generation of a syntactic description of the boundary, using the primitives shown in the figure [8].

An alternate mechanism for viewing this approach is derived from a neighborhood matrix with each neighbor coded to correspond to the primitives in the figure. This matrix appears as follows:

| 4 | 3 | 2 |
| :--- | :--- | :--- |
| 5 |  | 1 |
| 6 | 7 | 8 |

Having assigned the chain code of edge segment points in certain direction (say clockwise), the chain code of the points in the reversed direction (anticlockwise) can also be determined. Let an edge segment that has n number of points with the chain codes assigned starting from the first point (point 1) until reaching the last point (point $n$ ). The chain code for the segment points in the reversed direction, arranged from point n to point 1 , can be given as follows:

Reversed chain code at any point $\mathrm{k}=($ Chain code at the point k$)-4$

If the resulted value of the reversed code is zero or negative, add 8 to it. The codes of the points are then arranged from point $n$ to point 1 to get the right coding of the segment.

Therefore, for the edge segment of Figure 1, the reversed chain code results by applying Equation 1 can be written as:

$$
555344455665533322233221187811
$$

However, the order of the computed code is to be reversed to give the right code, starting from point n and ending with point 1 . Thus, the right code has the following final structure:

$$
118781122332223335566554443555
$$

Three parameters are considered when a chain code is created for a curve. They are start point, the step between two consecutive sample points along the curve, and the tracking direction on which the curve is traced. Choosing a different starting point usually leads to a different (shifted) coding result. Using a step of one between two consecutive sample points assures accurate representation of the shape. Rotation has similar effect in coding as choosing a different start point. However, the angles no matter how the curve is geometrically transformed remain unchanged. Since the number in the chain code represents the direction, the difference in the code reflects the turning angles. They are invariant to the transformations.

## 4. THE PROPOSED APPROACH

This section presents a proposed approach of automatic image to image registration based on matching edge segments. Here, edge segments are extracted from the two images (reference image and matched image) and their signatures are generated using chain coding. Segment signatures are utilized in a matching procedure to find common parts of edge segments representing corresponding features. Below are the procedures to be followed for the implementation of this approach.

1. Canny's edge detector is used to find edges in each of the two images. The detected edge points are aggregated into edge segments utilizing both gradient magnitude and orientation.
2. Having the resulted edge images, a row-after-row scanning process can be followed to catch any of the end points of each edge segment. Having found an end point, the chain code is assigned to the points of each edge segment until reaching the other end. The centroid of each segment is computed by averaging the coordinates of segment points. Coordinates and chain code of each segment point as well as centroid coordinates are recorded.
3. Each edge segment in the reference image is matched against candidate edge segments in the matched image. All edge segments whose centroids are located within a specified search window in the matched image are considered to be Candidates. The size of the search window can be computed using available information regarding the image formation process.
4. The matching process of two segments checks the similarity of chain codes of their points. The matching starts by comparing point codes while laying the shorter segment along the longer segment such that the beginning points of the segments coincide. The shorter segment is slipped one pixel at a time along the longer segment. In each relative position of the two segments, the points having similar chain codes are recorded and counted.
5. The matching score due to matching two segments is evaluated as the ratio of number of points of similar chain codes to the number of points of the shorter segment. The highest matching score, if it exceeds a selected threshold, indicates a match.
6. In some cases, depending on edge dissimilarities in the two images and circumstances of the edge aggregation process, the two segments to be matched are sorted in a reversed order. To overcome this problem, the matching is retried with shorter segment slide along the reversed version of the longer segment. The chain codes of the reversed version are determined using the procedure explained in Section 3.
7. In consequence of the matching process, the common parts in each two matched segments, which have points of similar codes, are extracted and recorded. The disparity values are computed for the extracted common points. This is to ascertain that the figural continuity is preserved after applying the matching procedure.
8. The coarse locations of any of the common points, of each two matched edge segments, are to be determined precisely by an intensity-based matching procedure. Here the location of common point in one image is fixed and the exact corresponding location in the other image is found automatically, to sub-pixel accuracy, using prototype cross correlation program. The input data to the program are the pixel coordinates of the selected common point in the two images, which determine the centers of the reference and search windows. The corresponding location of common point in the other image is found, to sub-pixel accuracy, by fitting a two-dimensional polynomial to the nine pixels centered at the position with the highest correlation, and searching for the maximum.
9. The precise location of the common points in the two images determines the parameters of the transformation needed to bring the images into alignment.
10. The image to be registered is finally transformed from its own coordinate system into the coordinate system of the reference image.

## 5. EXPERIMENTATION

The test images consist of two satellite images of different platforms and with different resolutions. The images cover the same area in the region of Kofra, Libya. It is a somewhat urban area with insignificant height variation. One image is a panchromatic QuickBird image with 0.6 m ground sample distance. Figure 2 shows a patch of the image, covering about 0.36 km by 0.36 km ground area. The other image is SPOT- 5 image with a 2.5 m ground sample distance. This image is resampled to generate an equivalent image with a 0.6 m ground sample distance, in order to be comparable with the QuickBird image. Figure 3
exhibits a patch of the generated image, covering the same ground area shown in Figure 2. The two patches are cut out from the original test images such that there are shifts of few pixels in the coordinates of corresponding feature points.

The edge segments are extracted by applying Canny's edge detector and the edge tracking algorithm to the two image patches, using MATLAB software package. Figure 4 and Figure 5 illustrate the resulted binary edge patches containing extracted edge segments. Due to the variation in radiometric and geometric characteristics of the two patches, extracted edge segments of corresponding features look differently. They are not exact copies of each other. Also, corresponding edges sometimes combine different features of the object space. Thus the interest is to find those common parts of the edge segment representing corresponding features.

The first task after extracting edge segments in both image patches is to assign the right signature to each segment using the chain coding. Edge segments that have more then twenty points are only considered in the coding process. For each of the two edge patches, a row-after-row scanning process is adopted to look for any of the end points of each edge segment. Having found an end point, the chain code is assigned to the points of each edge segment until reaching the other end. Coordinates and chain code of each segment point as well as coordinates of segment centroid are recorded. Figure 6 and Figure 7 illustrate the edge patches containing edge segments having more than twenty points.

Having coded each edge segment in both edge patches, the task now is to match edge segments to find each two conjugate segments. The matching process is carried out using the procedure described in Section 4. A prototype program is developed to implement the edge matching procedure. The size of the search window is taken as 25 pixels. The threshold for the matching score is selected as 0.7 . This means that at least $70 \%$ of the points of each of any two matched edge segments have the same chain code. The common points, points of similar chain code, of each two matched segments are extracted and recorded. The figural continuity is checked throughout the computation of the disparity values of resulted common points.

For each two matched edge segments, the location of the middle of common points in the reference patch is fixed and the exact corresponding location in the other patch is found automatically, to sub-pixel accuracy, using prototype cross correlation program. The pixel coordinates of the middle common point in the two images are employed as centers of the reference and search windows. The size of the reference window is specified as 11 pixels by 11 pixels. The size of the search window is chosen as 31 pixels by 31 pixels. A matching threshold of 0.7 is considered.

## 6. RESULTS AND ANALYSIS

The numbers of extracted edge segments that have more then twenty points are 487 in the Quickbird patch and 513 in the Spot patch. Each segment in the Spot patch is matched against its matching candidates in the Quickbird patch adopting a search window size of 25 pixels and matching threshold of 0.7 . Figure 8 and Figure 9 show the matched segments in each of the two patches.

By running the edge matching program, only 111 Spot segments have got matches in the Quickbird patch. No double matches have been detected; each segment of the 111 spot segments has only one match in the Quickbird patch. Looking at the disparity values of the common points of each two matched segments, it was found that they satisfy the figural continuity. However, there are two Spot segment pairs where each pair is matched to the same segment in the Quickbird patch. That is due to that each pair consists of two nearly horizontal segments that are close to each other.

By employing a prototype cross correlation program, the corresponding locations of the middle common point of each two matched edge segments in the both patch are refined. 9 matched segments are excluded due to their proximity to the image borders so that the search window is located fully or partially outside the image. Thus, only 102 middle point pairs are considered. Out of those 102 point pairs, 96 points have matching scores exceeding the selected threshold. Table 1 lists the pixel coordinates of the middle points in Spot patch and their conjugates in Quickbird patch obtained by applying correlation matching. The coordinate differences between the two coordinate sets as well as the estimated coefficient of correlation are also included in the table.

By performing an affine transformation between the two matched sets of coordinates, root mean square errors of 0.34 pixel and 0.24 pixel are resulted in the column and row directions, respectively. Those figures indicate the accuracy of the entire matching process given that the two patches have different radiometric characteristics. The resulted parameters of the affine transformation are: $\mathrm{a}=0.9973, \mathrm{~b}=-0.0010, \mathrm{c}=-2.195, \mathrm{~d}=-0.0011, \mathrm{e}=1.0003$ and $\mathrm{f}=-1.7078$. This is based on the affine model that $x^{\prime}=a x+b y+c$ and $y^{\prime}=d x+e y+f$ with ( $x, y$ ) point coordinates in the Spot patch and ( $x^{\prime}, y^{\prime}$ ) point coordinates in the Quickbird patch. x - and y-coordinates correspond to column- and rowcoordinates, respectively, of the points listed in Table 1. The resulted transformation parameters can then be used to transform the Quickbird patch into the pixel coordinate system of the Spot patch.

## 7. CONCLUSIONS

A proposed approach of automatic image to image registration based on matching chain-coded edge segments are presented and tested. The approach does not need any manual interaction and therefore has the potential to be integrated into an automatic workflow. With respect to the results of experiments made on the test imagery, matching chain-coded edge segments is found successful in identifying conjugate points. The matching ambiguity occurs mainly at having entirely straight edge segments with no critical points at all, especially in the cases they are partially existed or they are close to each other in the same image.

## REFERENCES

[1] Haralic, R. M. and Shapiro, L. G. (1992) "Computer and Robot Vision." Addison-Wesley, Inc., Reading, MA.
[2] HILD, H., HAALA, N. and FRITSCH, D. (2000) "A Strategy For Automatic Image To Map Registration." International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B2, pp. 287-294.
[3] Li D. and Simske S. (2002) "Shape Retrieval with Flat Contour Segments" http://citeseer.ist.psu.edu.
[4] Mikail, E.M., Bethel, J.S. and McGlone, J.C. (2001) "Introduction to Modern Photogrammetry." John Wiley \& Sons, Inc., New York.
[5] Parker, J. K. (1994) "Practical Computer Vision using C." John Wiley \& Sons, Inc., New York.
[6] Pal, N. R. and Pal, S. K. (1993) "A Review of Image Segmentation Techniques." Pattern Recognition, Vol. 26, No. 9. pp. 1227-1294.
[7] Pratt W. K. (1991) "Digital Image Processing" John Wiley \& Sons, Inc., New York.
[8] Schalkoff, R. J. "Digital Image Processing and Computer Vision." John Wiley \& Sons, Inc., New York (1989).
[9] Schenk T., Li, J.C. and Toth, C. (1991) "Towards An Autonomous Systems for Orienting Digital Images." Photogrammetric Engineering and Remote Sensing, Vol. 57, No. 8, pp.1057-1064.
[10] Schenk T. (1999) "Digital Photogrammetry" Laurelville, OH: TerraScience.
[11] Wolf, P.R. and Dewitt, B.A. (2000) "Elements of Photogrammetry." McGraw Hill, Inc., New York.
[12] Zhou, X. and Dorrer E. (2000) "Non-Linear Scale And Orientation Free Correlation Matching Algorithm Based On Edge Correspondence." International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B3, pp. 1054-1062.

(a) Digital Edge Segment


111788811221177766677665543455
(c) Edge Segment Code

Figure 1: Chain Encoding of A Digital Edge Segment


Figure 2: Quickbird Image Patch of the Test Area


Figure 3: Spot Image Patch of the Test Area


Figure 4: Edges detected in Quickbird Patch


Figure 5: Edges detected in Spot Patch

Figure 6: Edges of more than 20 points in Quickbird Patch


Figure 7: Edges of more than 20 points in Spot Patch


Figure 8: Matched Edges of Quickbird Patch


Figure 9: Matched Edges of Spot Patch

Table 1: Pixel Coordinates of Middle Points of Matched Segments in Spot Patch and Their Conjugates in Quickbird Patch Located by Matching

| Seg. <br> No. | Coordinates of <br> Middle Points in <br> Spot Patch |  | Coordinates of <br> Matched Points in <br> Quickbird Patch |  | Differences in <br> Coordinates |  | Correlation <br> Coefficient |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | column | row | column | row | column | row |  |
| 1 | 528 | 29 | 524.8 | 26.8 | 3.2 | 2.2 | 0.86 |
| 2 | 438 | 13 | 434.9 | 11.5 | 3.1 | 1.5 | 0.96 |
| 3 | 107 | 52 | 104.5 | 50.2 | 2.5 | 1.8 | 0.82 |
| 4 | 238 | 35 | 235.0 | 32.8 | 3.0 | 2.2 | 0.73 |
| 5 | 255 | 29 | 251.9 | 27.0 | 3.1 | 2.0 | 0.88 |
| 6 | 242 | 40 | 239.2 | 37.9 | 2.8 | 2.1 | 0.84 |
| 7 | 389 | 57 | 385.5 | 54.7 | 3.5 | 2.3 | 0.80 |
| 8 | 403 | 55 | 399.1 | 52.9 | 3.9 | 2.1 | 0.89 |
| 9 | 289 | 19 | 286.0 | 17.2 | 3.0 | 1.8 | 0.94 |
| 10 | 458 | 54 | 454.3 | 52.0 | 3.7 | 2.0 | 0.77 |
| 11 | 471 | 28 | 467.5 | 26.0 | 3.5 | 2.0 | 0.84 |
| 12 | 84 | 53 | 81.7 | 51.1 | 2.3 | 1.9 | 0.89 |
| 13 | 23 | 53 | 20.5 | 51.8 | 2.5 | 1.2 | 0.91 |
| 14 | 469 | 60 | 464.7 | 57.7 | 4.3 | 2.3 | 0.78 |
| 15 | 153 | 59 | 150.8 | 57.2 | 2.2 | 1.8 | 0.79 |
| 16 | 116 | 87 | 113.3 | 85.2 | 2.7 | 1.8 | 0.88 |
| 17 | 280 | 87 | 276.8 | 85.4 | 3.2 | 1.6 | 0.84 |
| 18 | 449 | 79 | 445.4 | 76.6 | 3.6 | 2.4 | 0.81 |
| 19 | 111 | 95 | 108.2 | 93.3 | 2.8 | 1.7 | 0.81 |
| 20 | 46 | 99 | 43.8 | 97.4 | 2.2 | 1.6 | 0.88 |
| 21 | 84 | 102 | 81.1 | 100.3 | 2.9 | 1.7 | 0.90 |
| 22 | 551 | 102 | 547.7 | 99.7 | 3.3 | 2.3 | 0.81 |
| 23 | 91 | 111 | 88.6 | 109.2 | 2.4 | 1.8 | 0.95 |
| 24 | 459 | 118 | 455.4 | 116.0 | 3.6 | 2.0 | 0.81 |
| 25 | 360 | 120 | 356.6 | 118.1 | 3.4 | 1.9 | 0.86 |
| 26 | 327 | 127 | 323.9 | 124.9 | 3.1 | 2.1 | 0.90 |
| 27 | 521 | 118 | 517.3 | 116.2 | 3.7 | 1.8 | 0.83 |
| 28 | 554 | 122 | 550.5 | 119.7 | 3.5 | 2.3 | 0.86 |
| 29 | 301 | 120 | 297.8 | 117.8 | 3.2 | 2.2 | 0.91 |
| 30 | 413 | 133 | 410.1 | 130.9 | 2.9 | 2.1 | 0.69 |
| 31 | 243 | 99 | 240.0 | 96.7 | 3.0 | 2.3 | 0.87 |
| 32 | 367 | 145 | 363.7 | 142.9 | 3.3 | 2.1 | 0.84 |
| 33 | 517 | 139 | 513.3 | 136.7 | 3.7 | 2.3 | 0.88 |
| 34 | 549 | 141 | 545.5 | 138.7 | 3.5 | 2.3 | 0.86 |
| 35 | 60 | 157 | 57.9 | 154.9 | 2.1 | 2.1 | 0.87 |
|  |  |  |  |  |  |  |  |

Table 1 (Continued)

| Seg. <br> No. | Coordinates of <br> Middle Points in <br> Spot Patch |  | Coordinates of <br> Matched Points in <br> Quickbird Patch |  | Differences in <br> Coordinates |  | Correlation <br> Coefficient |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | row | column | row | column | row |  |  |
| 36 | 59 | 157 | 56.8 | 154.9 | 2.2 | 2.1 | 0.86 |
| 37 | 120 | 152 | 117.1 | 150.2 | 2.9 | 1.8 | 0.81 |
| 38 | 180 | 158 | 177.0 | 155.7 | 3.0 | 2.3 | 0.88 |
| 39 | 68 | 161 | 65.8 | 159.1 | 2.2 | 1.9 | 0.85 |
| 40 | 244 | 157 | 240.9 | 154.6 | 3.1 | 2.4 | 0.87 |
| 41 | 358 | 182 | 354.7 | 179.8 | 3.3 | 2.2 | 0.85 |
| 42 | 535 | 183 | 531.2 | 180.9 | 3.8 | 2.1 | 0.88 |
| 43 | 549 | 204 | 545.2 | 201.9 | 3.8 | 2.1 | 0.88 |
| 44 | 192 | 193 | 188.9 | 190.8 | 3.1 | 2.2 | 0.94 |
| 45 | 304 | 200 | 300.8 | 197.7 | 3.2 | 2.3 | 0.87 |
| 46 | 494 | 206 | 490.2 | 204.1 | 3.8 | 1.9 | 0.82 |
| 47 | 381 | 229 | 377.2 | 226.9 | 3.8 | 2.1 | 0.88 |
| 48 | 516 | 259 | 512.1 | 256.6 | 3.9 | 2.4 | 0.85 |
| 49 | 570 | 234 | 566.3 | 231.6 | 3.7 | 2.4 | 0.85 |
| 50 | 403 | 245 | 399.2 | 242.6 | 3.8 | 2.4 | 0.85 |
| 61 | 23 | 254 | 21.1 | 252.5 | 1.9 | 1.5 | 0.89 |
| 52 | 423 | 245 | 419.3 | 242.8 | 3.7 | 2.2 | 0.88 |
| 53 | 283 | 260 | 279.7 | 258.1 | 3.3 | 1.9 | 0.90 |
| 54 | 116 | 314 | 113.5 | 312.4 | 2.5 | 1.6 | 0.85 |
| 55 | 143 | 285 | 140.2 | 282.9 | 2.8 | 2.1 | 0.85 |
| 56 | 29 | 299 | 26.9 | 297.6 | 2.1 | 1.4 | 0.82 |
| 57 | 324 | 296 | 320.1 | 293.9 | 3.9 | 2.1 | 0.89 |
| 58 | 237 | 316 | 233.9 | 314.0 | 3.1 | 2.0 | 0.87 |
| 59 | 306 | 322 | 302.3 | 319.7 | 3.7 | 2.3 | 0.84 |
| 60 | 114 | 314 | 111.6 | 312.4 | 2.4 | 1.6 | 0.85 |
| 61 | 122 | 324 | 119.5 | 322.1 | 2.5 | 1.9 | 0.91 |
| 62 | 518 | 317 | 514.1 | 314.9 | 3.9 | 2.1 | 0.90 |
| 63 | 203 | 353 | 199.9 | 351.0 | 3.1 | 2.0 | 0.90 |
| 64 | 540 | 329 | 536.1 | 326.4 | 3.9 | 2.6 | 0.87 |
| 65 | 303 | 354 | 299.2 | 351.4 | 3.8 | 2.6 | 0.89 |
| 66 | 438 | 350 | 433.9 | 347.7 | 4.1 | 2.3 | 0.91 |
| 67 | 88 | 369 | 85.6 | 367.6 | 2.4 | 1.4 | 0.91 |
| 68 | 489 | 345 | 485.0 | 343.0 | 4.0 | 2.0 | 0.90 |
| 69 | 377 | 347 | 373.7 | 345.0 | 3.3 | 2.0 | 0.93 |
| 70 | 518 | 373 | 514.1 | 371.0 | 3.9 | 2.0 | 0.89 |
|  |  |  |  |  |  |  |  |

Table 1 (Continued)

| Seg. <br> No. | Coordinates of <br> Middle Points in <br> Spot Patch |  | Coordinates of <br> Matched Points in <br> Quickbird Patch |  | Differences in <br> Coordinates |  | Correlation <br> Coefficient |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | column | row | column | row | column | row |  |
| 71 | 274 | 389 | 270.6 | 387.0 | 3.4 | 2.0 | 0.86 |
| 72 | 192 | 404 | 188.7 | 402.0 | 3.3 | 2.0 | 0.89 |
| 73 | 353 | 403 | 349.4 | 401.1 | 3.6 | 1.9 | 0.92 |
| 74 | 59 | 402 | 56.5 | 400.3 | 2.5 | 1.7 | 0.85 |
| 75 | 523 | 429 | 518.8 | 426.9 | 4.2 | 2.1 | 0.87 |
| 76 | 391 | 429 | 387.4 | 427.0 | 3.6 | 2.0 | 0.94 |
| 77 | 86 | 432 | 83.7 | 430.5 | 2.3 | 1.5 | 0.90 |
| 78 | 232 | 434 | 228.6 | 431.9 | 3.4 | 2.1 | 0.88 |
| 79 | 404 | 438 | 400.5 | 436.0 | 3.5 | 2.0 | 0.93 |
| 80 | 500 | 450 | 496.4 | 447.4 | 3.6 | 2.6 | 0.83 |
| 81 | 167 | 440 | 163.8 | 438.2 | 3.2 | 1.8 | 0.87 |
| 82 | 463 | 471 | 459.2 | 469.0 | 3.8 | 2.0 | 0.83 |
| 83 | 488 | 488 | 484.2 | 486.1 | 3.8 | 1.9 | 0.92 |
| 84 | 258 | 484 | 254.7 | 482.2 | 3.3 | 1.8 | 0.90 |
| 85 | 455 | 502 | 451.2 | 499.9 | 3.8 | 2.1 | 0.84 |
| 86 | 437 | 498 | 433.2 | 495.9 | 3.8 | 2.1 | 0.82 |
| 87 | 409 | 507 | 405.5 | 505.1 | 3.5 | 1.9 | 0.89 |
| 88 | 298 | 503 | 294.9 | 501.0 | 3.1 | 2.0 | 0.80 |
| 89 | 351 | 473 | 347.2 | 470.9 | 3.8 | 2.1 | 0.86 |
| 90 | 37 | 521 | 32.9 | 520.0 | 4.1 | 1.0 | 0.95 |
| 91 | 36 | 521 | 31.7 | 520.0 | 4.3 | 1.0 | 0.96 |
| 92 | 98 | 542 | 95.5 | 540.7 | 2.5 | 1.3 | 0.78 |
| 93 | 250 | 530 | 246.6 | 528.1 | 3.4 | 1.9 | 0.89 |
| 94 | 404 | 527 | 400.3 | 525.1 | 3.7 | 1.9 | 0.90 |
| 95 | 396 | 538 | 392.4 | 536.1 | 3.6 | 1.9 | 0.86 |
| 96 | 505 | 542 | 501.0 | 539.8 | 4.0 | 2.2 | 0.89 |

