RELATIONAL MATCHING FOR AUTOMATIC ORIENTATION

Woosug Cho

Department of Geodetic Science and Surveying
The Ohio State University
Columbus, Ohio 43210

Commission III, Working Group 3

KEY WORDS: Relational matching, A* search, Heuristics, Binary tree, Cost and Benefit function, Interest operator, Relative orientation.

ABSTRACT

The objective of this research is to investigate the potential of relational matching in one of the fundamental photogrammetric processes, the orientation of a stereopair. The automatic relative orientation procedures of aerial stereopairs have been investigated. The fact that the existing methods suffer from approximations, distortions (geometric and radiometric), occlusions, and breaklines is the motivation to investigate relational matching which appears to be a much more general solution.

An elegant way of solving the initial approximation problem by using distinct (special) relationship from relational description is suggested and experimented. Two evaluation functions (cost and benefit function) with the same relational descriptions are investigated. Special attention is given to the solution for relational matching when a large number of features are involved. To speed up the relational matching procedure, unit ordering and modified forward checking are incorporated into the proposed relational matching scheme. In addition, an optimal way of constructing local binary relations is implemented.

The detection of erroneous matching is incorporated as a part of proposed relational matching scheme. Experiments with real urban area images where large numbers of repetitive patterns, breaklines, and occluded areas are present prove the feasibility of implementation of the proposed relational matching scheme.

The investigation of relational matching in the domain of image matching problem provides advantages and disadvantages over the existing image matching methods and shows the future area of development and implementation of relational matching in the field of digital photogrammetry.

1. INTRODUCTION

One of most fundamental tasks in photogrammetry is to find conjugate features in two or more images, which is commonly referred to as the matching problem. In conventional photogrammetry, the matching problem is solved by a human operator who identifies conjugate features in two or more images without conscious effort, in real time. The human visual system is easily able to form a stereo model and to describe the scene content in a highly symbolic fashion. In digital photogrammetry, the matching problem, which is called image matching problem in this study, is yet far from being solved fully automatically. The most persistent problems are occlusions, foreshortenings (relief distortions), breaklines (discontinuities in surface) and nonlinear radiometric differences among the images [Doorn et al. 1990, Zilberstein 1992].

The image matching problem can be described as comparing a specific feature in one image with a set of other features in the other image and selecting the best candidate, based on the similarity measure between feature descriptions. The feature description can be described at different levels of abstraction. Depending on the level of feature description, the image matching methods are usually divided into the three groups: area-based matching, feature-based matching, and relational matching. For a detailed description of the area-based and feature-based matchings, the reader is referred to the papers [Schenk 1992, Haralick and Shapiro 1992].

In computer vision, relational matching has been used for problems like object recognition and location, scene analysis, and navigation. Recently, relational matching began to gain attention in digital photogrammetry [Vosselman 1992, Zilberstein 1992, Shahin 1994, Tsingas 1994].

As the name suggests, relational matching seeks to find the best mapping between two relational descriptions. Relational description consists of not only features but also geometrical and topological relationships among the features. In order to find the best mapping, relational matching has to employ the measure of similarity while mapping one relational description into the other relational description. The measure of similarity between two relational descriptions can be achieved by an evaluation function which is usually defined as a cost function or benefit (merit) function. The cost function is to be minimized and is zero if two relational descriptions are identical. Unlike a cost function, the benefit function is to be maximized; and it achieves a maximum when two relational descriptions are best matched.

The motivation for proposing a relational matching scheme in this paper stems from the fact that the method is much less sensitive to many factors which are limiting the existing image matching methods. Such factors include approximations, distortions (geometric and radiometric), and occlusions. Consequently, relational matching appears to be a much more general solution.

2. FEATURE EXTRACTION

Point features provide the most stable geometry for relative orientation. The extraction of distinct points such as corner points is a basic procedure in digital photogrammetry and computer vision. There has been much research in the field of distinct point detection [Moravec 1977, Förstner 1994, Tang and Heipke 1994]. These previous works show that the Moravec operator and the Förstner interest operator perform best for real images. The Förstner interest operator was chosen because of its salient features such as rotation invariant and subpixel accuracy. For a more detailed description of the Förstner interest operator the reader is referred to Förstner and Gülch [1987].

Linear feature extraction (edge detection) plays a crucial part in digital photogrammetry and computer vision. Boundaries of objects tend to show up as intensity discontinuities in an image (edges). An edge operator algorithm is designed to detect local edges within small spatial extents. The computer vision and image processing literature is abound with edge operators, see, e.g. [Ballard et al. 1982, Haralick and Shapiro 1992].

To obtain the straight lines with corner points as their end points, the straight line extraction algorithm must be well behaved around the corner points and should produce as long and straight lines as possible. In addition, the algorithm must have good geometric precision(localization). After investigating existing linear feature extraction algorithms, a simple algorithm which fulfills to some extent all the necessities described above is developed.

The proposed straight line extraction algorithm is based on two properties:

1. good geometric precision (localization).
2. lines are as straight and long as possible.

The first property can be achieved by applying the 2×2 Robert gradient operator which is optimal among 2×2 operators [Haralick and Shapiro 1992]. The computed gradient magnitude is thresholded by a free parameter which is estimated from the whole image content. This weight threshold step prevents generating weak and less meaningful straight lines. The second property, long and straight line, can be obtained by the analysis of the chain of edge pixels which results from edge following. The edge following can be implemented in two different ways:

- gradient magnitude based approach,
- gradient orientation based approach.

The paper [Burns et al. 1986] shows that the gradient orientation varies relatively less over the intensity surface than the gradient magnitude. Thus, the gradient orientation based approach of edge following is selected. Next, the chain of edge pixels is analyzed to extract a long and straight line from the chain. The algorithm for extracting a long and straight line from the chain of edge pixels is mainly based on an algorithm of Douglas and Peuker [Ballard and Brown 1982]. With the Douglas and Peuker algorithm, the straightness of line can be achieved by a small threshold for norm distance.

The suggested algorithm is simple, computing efficient and does not require any postprocessing such as thinning. Since the 2 × 2 Robert gradient operator is implemented, it interacts well with the physical boundaries. However, because the operator window size is small, it may suffer from the noise in an image. It must be noted that the success of matching depends heavily on good feature descriptions. However, the primary concern of this work is not feature extraction but relational matching and its related procedures.

2.1 Feature Postprocessing

The interest point operator fails to detect corner points of feature lines that intersect outside of the threshold angle. The gradient disturbance around corner points also causes the failure of detecting corner points. Some of those missing corner points can be recovered by using extracted straight lines. When two or more straight lines extracted by the suggested algorithm meet at a point which is not detected as a corner point by the interest point operator, the point is considered as a corner point. However, if the angle between straight lines is less than 30 degrees, the point is not considered as a corner point and also two straight lines are not taken as one long straight line.

Due to noise, low resolution of the image, and shadow cast by a building, the line following does not reach the corner points detected by the interest point operator or passes over the corner points. To solve this problem, a corner point is searched inside the area A which is created by width w and distance d. The search for a corner point is based on the following two rules:

1. Proximity.
2. Forward primary.

Based on the proximity rule, the search starts from the end point of a straight line by checking its 8 neighborhood. The forward primary rule means that the search follows primarily the extending line direction. If a corner point is not found after the search moves one pixel forward along the line, it moves one pixel backward along the line. This procedure repeats until a corner point is found or to the end of search area. When the search finds more than one corner point, the corner point that is closest to the line and is in the forward direction is selected based on the two rules.

3. PROPOSED RELATIONAL MATCHING SCHEME

The proposed scheme of relational matching utilizes straight line primitives and their binary relations. A potential problem of relational matching is a large search space which may result in unacceptable computation cost. Two measures are introduced to prune the large search space. For one, only a limited number of binary relations between line primitives are allowed. The second measure is to determine good initial approximations.

Because the two relational descriptions do not match precisely (noise, occlusion, etc.), inexact matching and nil mapping is used. The evaluation functions have been widely used to guide the heuristic search to find the solution. To compare the behavior of a heuristic search with respect to evaluation functions, two evaluation functions (cost and benefit functions) are implemented. The heuristic tree search method A* with heuristics (unit ordering and modified forward checking) is implemented to find the mapping between two relational descriptions.

3.1 Description of Primitives and Relations

The relational description consists of primitives and relational tuples among the primitives. Three primitives are used in this study: (1) Open straight line, (2) Half-open straight line and (3) Closed straight line. Each straight line may have none, one or two corner points at its ends. An open straight line has no corner points, the half-open straight line has one corner point, and the closed straight line has two corner points. Each line primitive has three attributes: Length, Orientation and Contrast. The units of length and orientation are in pixels and degrees, respectively. The contrast could be 0 or 1 depending on the direction of line gradient and line orientation. Figure 1 illustrates the attributes of the line primitives.

Three 2-D binary relations constitute the relational description:

1. Central distance: the distance between the centers of two straight lines.
2. Short distance: the shortest distance between two straight lines.

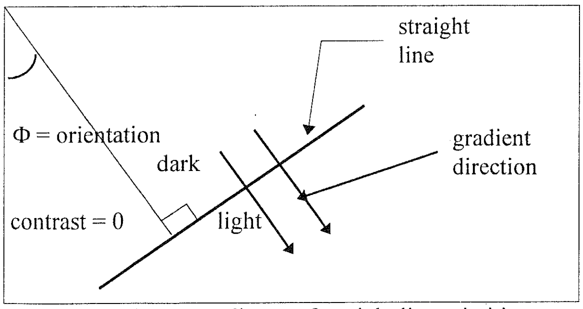


Figure 1: The attributes of straight line primitive.

3. Angle: the angle between two straight lines.

Figure 2 illustrates the binary relations and their attributes. All three relations are necessary for unambiguously describing the spatial relationship between line primitives. It must be noted that all attributes of the straight line primitives and of the binary relational tuples implemented are orientation-invariant quantities except for line primitive orientation. Obviously, the proposed procedure cannot be used for matching stereopairs with large rotations. However, the relational descriptions described above are valid for the aerial images where small rotation exists in two images.

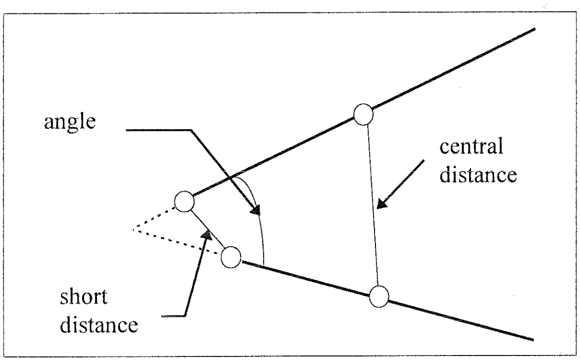


Figure 2: Three 2-D binary relations and their attributes.

3.2 2-D Binary Relations from 2-D Tree Technique

Each primitive is only allowed to have a small number of binary relations with its neighboring primitives. Since the density and distribution of line primitives varies over the image, the straight-forward approach of using a circle, centered around the midpoint, would not do a good job. Instead, the following approach is chosen. Its goal is to subdivide the image into rectangles such that every rectangular area has the same number of line primitives.

Two-dimensional (2-D) tree approach is suitable to divide the image into a number of small rectangular planes that contain a certain number of primitives. The primitives contained in a rectangular plane have binary relations with the primitives in the neighboring rectangular planes. In this way, the number of binary relations can efficiently be manipulated. The subdivision of a 2-D image plane and its corresponding 2-D tree are represented in Figure 3. In Figure 3, the black dots represent the centers of line primitives and each subdivision is designed to have 4 or 5 line primitives in this example.

In the interest of brevity, the implementation details of the approach are skipped. Each rectangular plane is called a bucket. In summary, the nodes at the lowest level of a 2-D tree are neighborhoods if one of following six conditions are fulfilled :

1. Their parent nodes are the same.
2. Their nodes are the same and their parent nodes are different.
3. Their nodes are different and their parent nodes (*Left* and *Right*) are different, and grandparents are the same and their *row* bounds overlap.
4. Their nodes are different and their parent nodes (*Up* and *Down*) are different, and grandparents are the same and their *column* bounds overlap.
5. Their nodes are different, their grandparent nodes (*Left* and *Right*) are different, and their *row* bounds overlap.
6. Their nodes are different, their grandparent nodes (*Up* and *Down*) are different, and their *column* bounds overlap.

The conditions described above are valid for a binary tree with depth 3. If the depth of the binary tree gets deeper, the conditions can be easily expanded in an alternating manner.

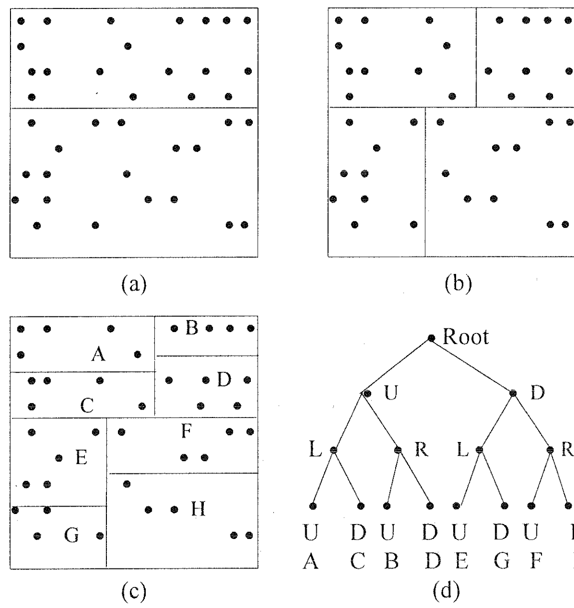


Figure 3: A subdivision of 2-D image plane and corresponding 2-D tree.

3.3 Special Relation

The half-open and closed line primitives have one and two corner points at the ends, respectively. These two line primitives may share the same corner points as their end points. In that case, the identical corner point is called node point. Now, the binary relations between two line primitives which share the same corner point at the end are called node relations in this study. Since the node relations are distinct from the other binary relations the node relations are particularly suited to be matched beforehand in order to determine approximations.

Each node relation has three attributes:

1. Angles between two line primitives (α, β).
2. Orientation between two node points (Φ).
3. Distance between two node points (d).

Figure 4 illustrates the node relations and their attributes. Whereas the angles (α,β) and distance (d) are orientation-invariant quantities, orientation (Φ) is orientation variant. Thus, the node relation cannot be used for matching a stereopair with a large kappa angle.

All pairs of line primitives with node points are extracted and all combinations of node relations are created. Relational matching is performed by A* search with forward checking. For the matched node points, the mismatch detection process is then performed, which is discussed in next section. Using the matched node points, the initial approximation between two relational descriptions can be estimated. The initial approximation is represented by the base vector (shift) $b = (b_{row}, b_{col})^T$ between two relational descriptions. The base vector with matched m node relations is computed as

$$b_{row} = \frac{\sum_{k=1}^{m} (r_{row_k} - l_{row_k})}{m}, \quad b_{col} = \frac{\sum_{k=1}^{m} (r_{col_k} - l_{col_k})}{m}. \tag{1}$$

where r and l stand for right and left image, respectively. The base vector as an initial approximation serves the mapping function to find the corresponding line primitives in a reduced search space.

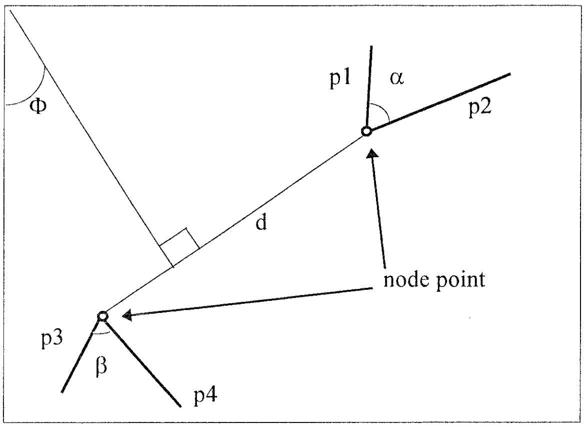


Figure 4: The node relation with its attributes.

3.4 Evaluation Function and Heuristics

The evaluation function guides the search algorithm to a solution, based on the similarity measure between two relational descriptions. There are two ways of estimating the similarity measure using the cost function: distance measure approach and conditional probability approach. The distance measure approach that utilizes the absolute differences between geometric attributes of conjugate primitives (relational tuples) is implemented in this research. Details of the conditional probability approach can be found in [Boyer and Kak 1988]. A benefit function estimates the support (or benefit) that the attributes of corresponding primitives and relation tuples give to the mapping. The mutual information is implemented as a benefit measure [Vosselman 1992]. The unit of information is *nat*, that is the base of the logarithm is e. It is assumed that all attributes of all primitives and relational tuples are independent of one another so that total measure of cost and benefit between two relational descriptions can be computed simply by summing up each cost (benefit) measure among primitives and relational tuples

Scale, different viewing directions, surface orientation, topography undulation and relative heights of objects cause the geometric attribute values and the geometric relations to be different. Thus, it is necessary for the matching to know these variations. In this study, the variation of two relational descriptions is estimated by an analytical function with a few critical parameters. The result of this analytical function approach is used to compute a cost measure as well as a benefit measure using the conditional probability function for mutual information. For the analytical function, the collinearity equation is chosen because of easy manipulation of parameters of interest. For the lack of space, the implementation details of the analytical function approach are skipped. For a more detailed description of the analytical function approach and evaluation function the reader is referred to Cho[1995].

To speed up the searching process in a tree, two well known heuristics are implemented: unit ordering and forward checking. The search space in the image matching problem is rather large. One way of reducing it is by ordering unit primitives. Tree search suffers from many backtrackings and explores fruitless paths when unit primitives at higher levels of a tree have many possible candidates. Therefore one is interested in ordering the tree in such a fashion that unit primitives with fewer label primitives are ordered at higher levels of the tree.

Forward checking examines the consistency of current unit-label pair with future unit-label pairs below the current level in a tree. The forward checking procedure is modified to be suitable for the image matching problem in this study. While examining future with current unit-label pairs, the modified forward checking counts and stores the number of valid future pairs at the current level. This number is stored in a 2-D table: row for unit primitives and column for label primitives. At the current level of a tree, the modified forward checking sums up the number of previous and future unit-label pairs. The total number obtained by the modified forward checking is utilized while the search tree tries to find a solution.

3.5 Matching Scheme

Since the descriptions of the two images of a stereopair differ (due to noise, different viewing angles, difficulties in feature extraction and occlusion, etc.), inexact relational matching must be employed. The extent of nil mapping must be controlled, especially when the evaluation function is a cost function. If the tree search maps all unit primitives to nils, the total cost measure is zero and this mapping provides a trivial solution. As for the benefit function, nil mapping does not contribute any benefit to the mapping. However, the extent of its usage is not actually a concern because any unit-label pairs that benefit the mapping are more preferable than nil mapping.

The proposed relational matching scheme utilizes the A* search with heuristics such as unit ordering and the modified forward checking. Since the image matching problem reaches the solution at the bottom of the tree, A* search is selected in this study. Although the proposed matching scheme employs the heuristic search A* with the modified forward checking and unit ordering, the search space is too large to reach the solution in a reasonable amount of time. To speed up the convergence of the solution, two relational descriptions are locally matched first, and then a globally consistent solution is achieved using a relation consistency check. Figure 5 illustrates the entire procedure of the proposed relational matching scheme.

3.6 Mismatch Detection

The proposed matching scheme matches the line primitives in two relational descriptions. Even though some matched pairs satisfy the interrelationships with other matched pairs, there could be mismatches due to a large search window, segmentation error, and missing corresponding features. These mismatches can be detected by two steps: a geometric approach and a radiometric approach.

The geometric approach employs the affine transformation between two matched line primitive pairs. It is assumed that rotation between two relational descriptions is small or estimated from the matched line primitives. The detailed procedure is the following.

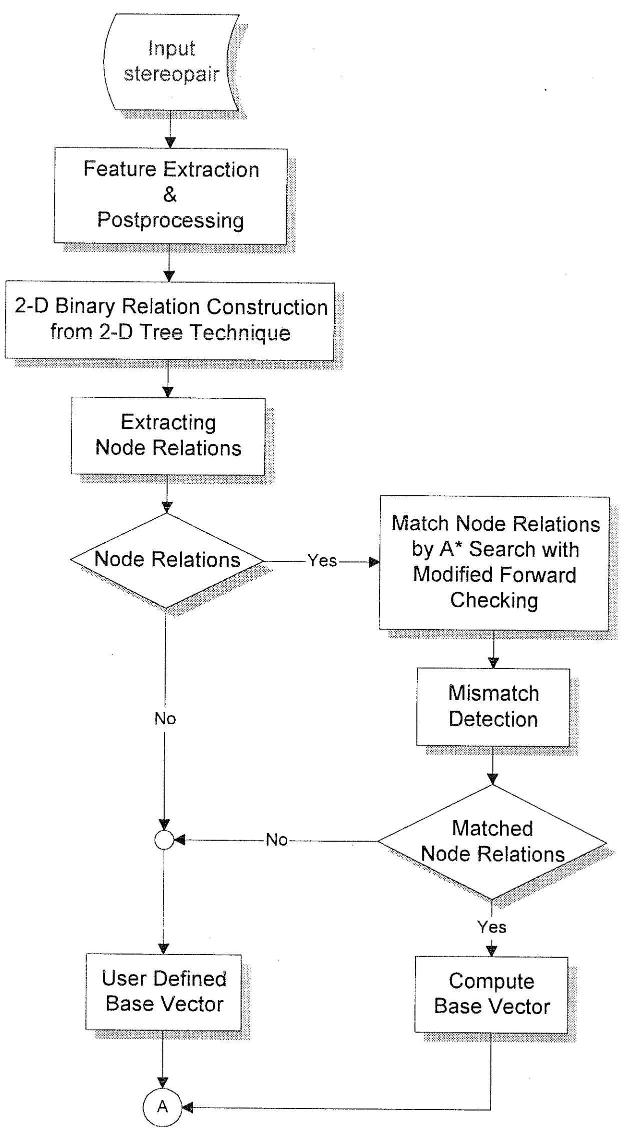
1. Compute the affine transformation between two sets of matched end points.
2. Estimate the residuals in y coordinates between the original points and the transformed points and compute the standard deviation of residuals in y coordinates.
3. Eliminate the points whose residuals in y coordinates exceed three times the standard deviation.

In this approach, the y coordinates of the matched points are only considered because the x coordinates correspond to the object height of the points.

While the geometric approach is designed to detect the blunder-like mismatches, the radiometric approach is to detect mismatches more rigorously. The radiometric approach employs the correlation technique using the normalized correlation coefficients for the matched end point pairs. Any matched point pairs satisfying the condition $\rho \leq \rho_{threshold}$ are eliminated from the set of conjugate point pairs. The correlation coefficient threshold $\rho_{threshold}$ is set to 0.6 in this study..

4. EXPERIMENTS AND RESULTS

To assess the feasibility of the proposed relational matching scheme, a software prototype was developed and experiments



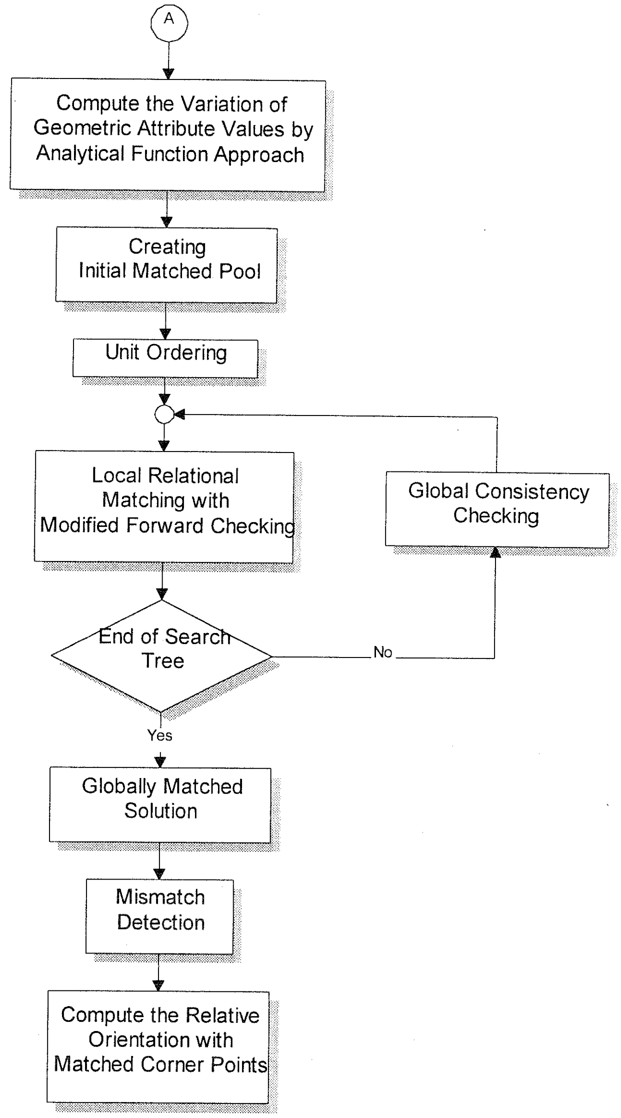


Figure 5: Flowchart of the proposed matching scheme.

with two stereomodels have been performed. Because of a limited space, the author describes one of data sets used, and reports and analyzes the major results. All computations, such as generating image pyramids, image subsampling, feature extraction, and relational matching were performed on an Intergraph workstation InterPro 6000.

The stereopair consists of two digitized images depicting the campus of The Ohio State University at a scale 1: 4000. The diapositives were scanned by the Intergraph Photoscan with a resolution of $30\mu m$. An image pyramid was generated using a Gaussian kernel. Images with a resolution of 512×512 pixels were used to extract corner points and straight lines. Figure 6 shows the stereopair superimposed with corner points.

4.1 Feature Extraction

Corner points and straight lines were extracted from the images. For the Förstner interest operator, $f = 1.5$, $q = 0.75$, 97% confidence level and 7×7 window size, also for nonmaxima suppression. The linear straight lines were obtained by the straight line extraction algorithm developed. There are four free parameters required in this algorithm: minimum line distance, norm distance, and gradient orientation and magnitude threshold. In this experiment, the minimum distance, norm distance, and orientation and magnitude threshold are 15 pixels, 1.5 pixels, 45°, mean gradient magnitude, respectively.

It is important to use an orientation-invariant gradient operator, because the Förstner interest operator is also orientation-invariant. Several gradient operators were tested and was found that the 2×2 Robert operator performs best with the Förstner interest operator. Most corner points are detected well by the interest operator. However, a closer examination of the OSU campus model reveals that some corner points remained undetected where the gradient distribution around the point is not symmetrical.

After the feature postprocessing, each straight line was named (closed, half-open, and open) based on the number of corner points connected to the line. The number of straight lines of the three different types after the feature postprocessing is listed in Table 1. Figure 7 depicts the results of this postprocessing.

Table 1: Number of straight lines of three different types after postprocessing.

Line Type	Left Image	Right Image
Closed line	56	68
Half-open line	141	148
Open line	139	127

4.2 2-D Tree Binary Relation Construction

In the experiment, each bucket is designed to have four or five line primitives. Since the binary relations are constructed locally, the set of binary relational tuples for the label primitives must be larger than that of unit primitives. There is a chance that some label primitives may not have binary relations in their set of binary relational tuples which correspond to those of unit primitives. The nonexisting binary relational tuples in the binary relation set for label primitives results in assigning unit primitives to nils even though the unit and label primitives are conjugate features. This problem can be solved by reducing the depth of the 2-D binary tree of the label primitive set by one, which is equivalent to making each bucket for the label primitive set having twice the primitives that the unit primitive set has. Thus, each bucket for the label primitive set is allowed to have 9 or 10 line primitives.

As listed in Table 2, the left image of the OSU stereopair have a number of line primitives less than the right images, thus the set of line primitives of the left image is taken as unit primitive set. As shown in Table 2, the left image (unit primitive set) has a 70% reduction and the right image (label primitive set) has a 30 % reduction.

Table 2: The reduced number of binary relations using the 2-D tree technique.

Closed and Half-open Line	Left (unit)	Right (label)
Line Primitive Number	197	216
Maximum of Binary Relation No.	19,306	23,220
2-D tree Technique	5,690	15,741

4.3 Node Relation Matching

For the OSU stereopair, the unit primitive set (left image) has 22 node relations and the label primitive set has 26 node relations. To match two node relation sets, the proposed relational matching scheme with cost function was implemented. The cost function was selected because it tends to match as many as possible due to the controlled usage of nil mapping. The mismatch detection was then performed on those matched node points. In Figure 8, the matched node relations after the mismatch detection are shown. Since the node relations are distinct from one another, there were no wrong matches for the OSU stereopair. Using equation (1), the base vector b = (-0.25, -186.375) was computed. The processing time for node relation matching is 9.5 seconds for the OSU stereopair.

4.4 Matching Results

The matching result with benefit function only is shown in Figure 9 because of the lack of space. The matching result statistics are listed in Table 3. As shown in Table 3, two evaluation functions perform quite similarly except for the computation times.

While the relational matching starts the search, it tries to find a path with a minimal cost and also a minimum number of nil mappings. Even though the search tree recognizes the cost of the current path is higher than that of the solution path already found, it must keep expanding the subtrees as long as the number of nils in the current path is smaller than the path that is already found. This property causes poor performance of the cost function. The search tree finds a path which has a minimum cost, but if the other path has a smaller number of nil mapping, then the search tree takes the path as the solution path. Since the cost function cannot cope efficiently with the nil mapping, the cost evaluation function suffers from the computation complexity by expanding unnecessary subtrees.

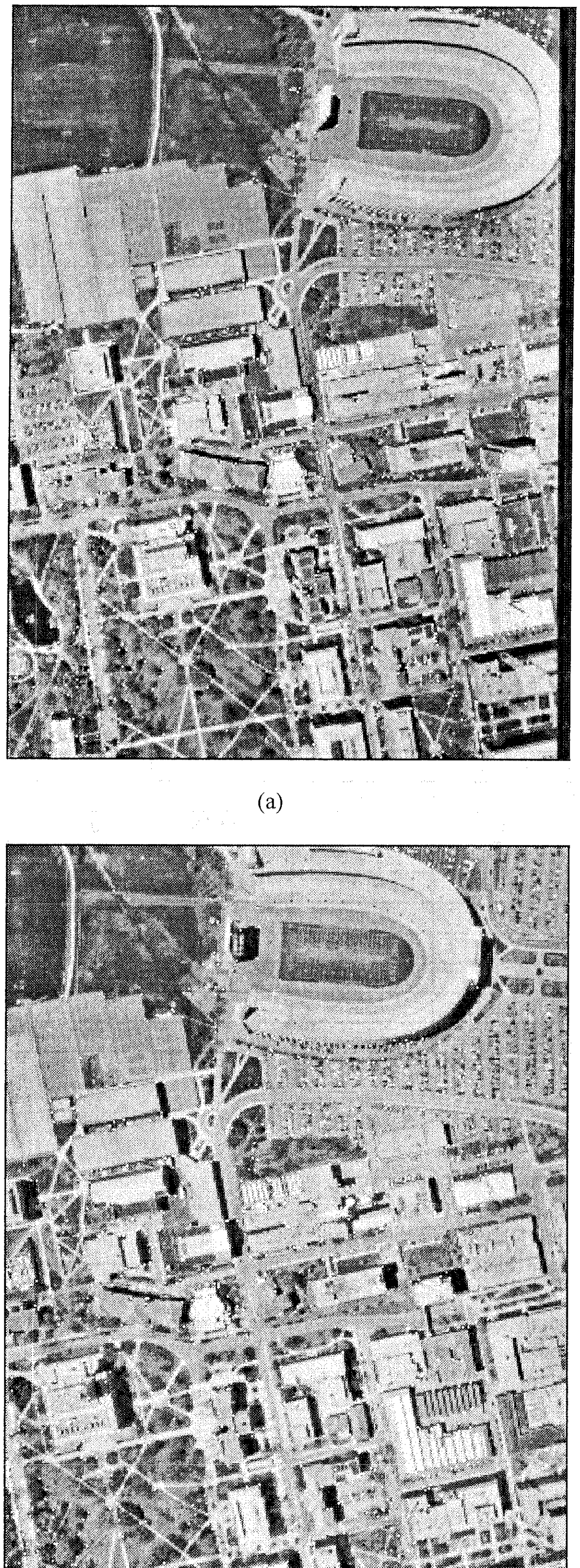
Unlike the cost function, the benefit function does not have a nil mapping problem from the point of computation complexity. When the tree search with benefit function finds a path which has more benefit than the solution path already found, it takes the path as a solution path without considering the number of nil mappings in the current path. This property of the benefit function makes the search tree backtrack when the estimated benefit is smaller than the benefit of the solution path. This characteristic of the benefit function leads to better performance than cost function.

Relational matching with both evaluation functions has several bad matches in which relations are held only locally, but not globally. These wrong matches are obtained when unit primitives have no corresponding label primitives in a set of label primitives but have label primitives which happen to have the relations compatible with other relations among the primitives. Using the relationally matched line primitives, the relative orientation between two images was computed. The matched end points were extracted from the matched lines and then those points were checked through the mismatch detection. Table 4 shows the matched end point numbers after the mismatch detection.

The matched points after mismatch detection were used for computing the relative orientation between two images. Table 5

Table 3: The matching results from the proposed relational matching scheme.

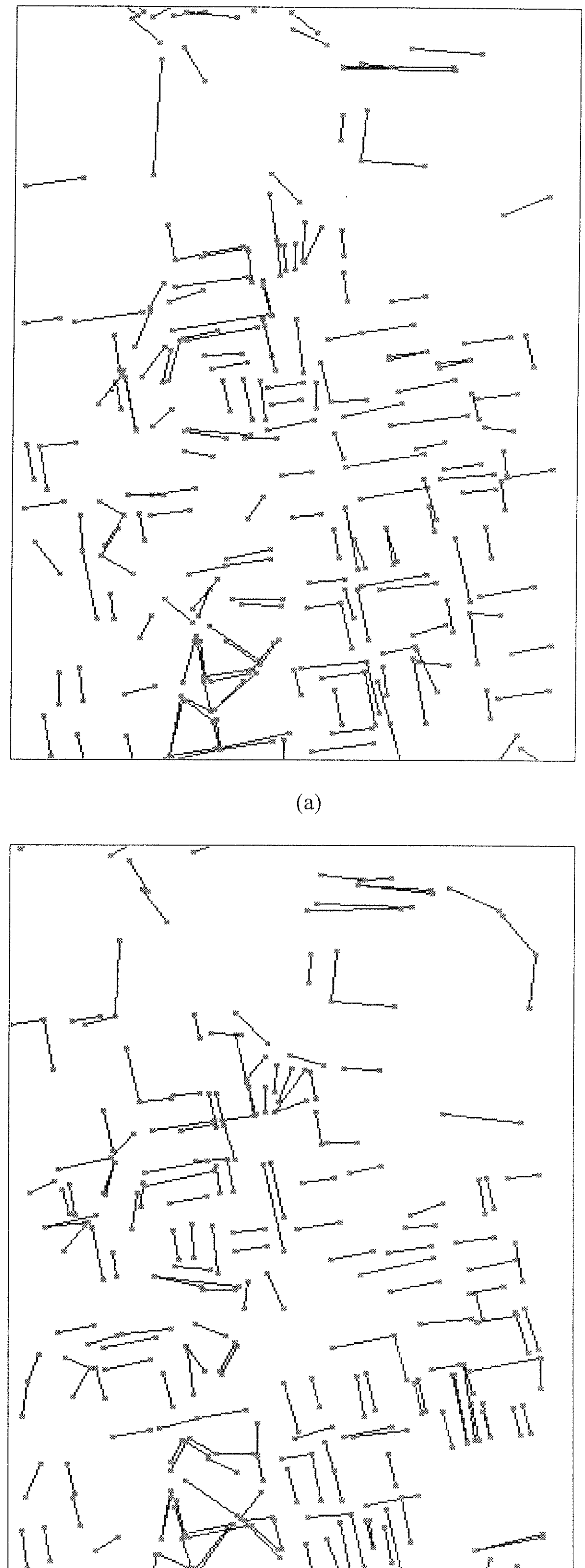
Statistics	Cost Function	Benefit Function
Total unit primitives number	197	197
Matched line primitives number	103	107
Good match	90	97
Bad match	13	10
Correct matching percentage	87.379	90.654
Computation time (second)	1428	1079



(a)

(b)

Figure 6: OSU left (a) and right (b) images superimposed with interest corner points: 97% confidence using 7×7 operator window size, 7×7 non-maxima window size, $q = 0.75$, and $f = 1.5$.



(a)

(b)

Figure 7: OSU left (a) and right (b) images with extracted corner points and straight lines after feature postprocessing.

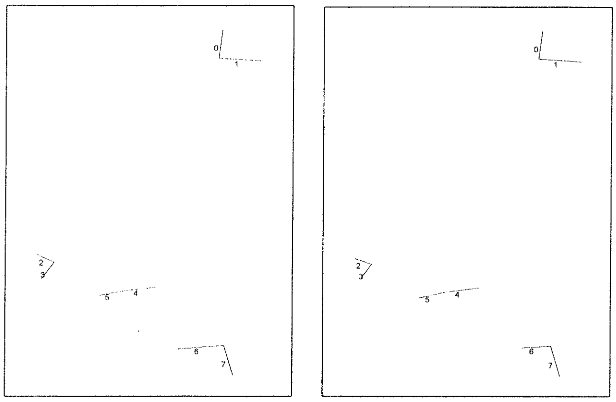
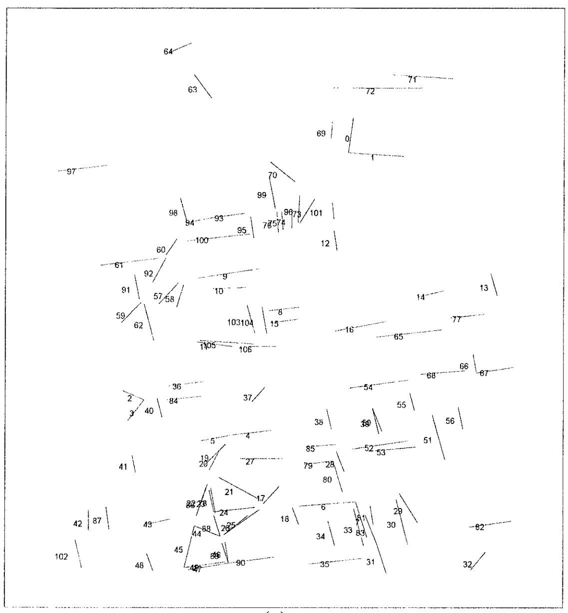
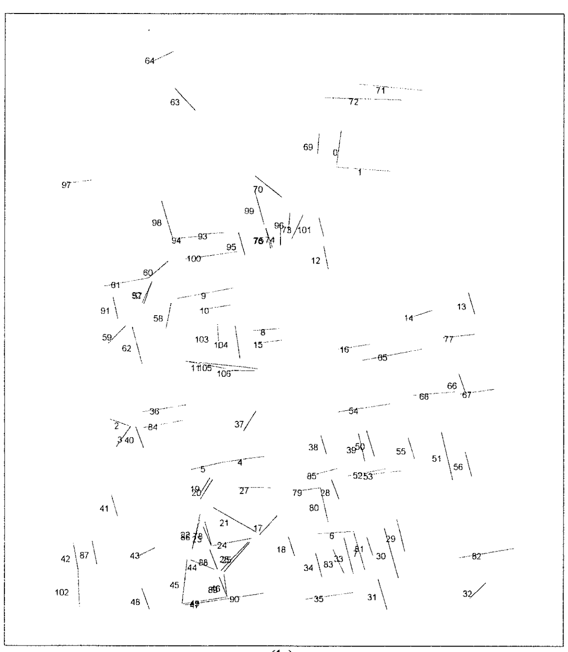


Figure 8: The matched node relations



(a)



(b)

Figure 9: Matched line primitives with benefit function: (a) left (unit) image and (b) right (label) image.

Table 4: Number of matched end points after mismatch detection.

	Cost Function	Benefit function
No. of matched points from relational matching	96	103
No. of matched points after radiometric mismatch detection	42	44
No. of matched points after geometric mismatch detection	41	44

shows the results of automatic relative orientation from the proposed relational matching scheme with two evaluation functions and an experienced human operator. As shown, the results of automatic relative orientation were less accurate than that of a human operator. However, the results of automatic relative orientation are less than one pixel accuracy, which is accurate enough for relational matching to be used as a means of providing the approximation for a highly accurate matching method such as area based matching.

Table 5: Results of automatic relative orientation.

	Cost function	**Benefit function**	**Analytical plotter**
Pixel resolution	480μm	480μm	
Number of points	40	44	9
Xo (mm)	90.681	90.618	90.586
Yo (mm)	-2.439	-2.450	-2.393
Zo (mm)	150.182	150.175	150.341
ω (deg)	0.644	0.647	0.604
ϕ (deg)	-1.183	-1.210	-1.327
κ (deg)	359.151	359.168	359.064
Maximum residual	758.7	722.6	10.3
Standard deviation	329.1	304.5	4.5

5. CONCLUSIONS

In this study, relational matching with two evaluation functions was implemented to solve the image matching problem, specifically automatic relative orientation. As shown, the proposed relational matching scheme provided reliable results of automatic relative orientation for urban area images containing breaklines, occlusions, and repetitive pattern by using three different binary relationships between two straight line primitives.

In the proposed relational matching scheme, locally consistent relations were extended to a globally consistent solution. From a practical point of view, this is a way of obtaining a solution for a large set of primitives. Furthermore, the 2-D tree technique was developed to speed up the tree search and manipulate the local binary relations efficiently. Implementing heuristics such as unit ordering and the modified forward checking also helped the relational matching reach a solution without expanding unnecessary subtrees.

The investigation of two evaluation functions in relational matching showed that the benefit function performs little better than the cost function, particularly for the domain of the image matching problem. One way of obtaining an initial approximation between two images with little rotation was developed and implemented. It showed that the distinct and prominent relations such as node relations were a good way of solving the initial approximation problem. Despite the limited experiments in this study, the results confirm that relational matching successfully deals with many problems in urban area images.

This research, however, showed that there were some problems in relational matching. First, the computation expenses are too heavy to obtain the solution in a reasonable amount of time, compared to existing matching methods. Since relational matching is a combinatorial optimization process, the computing complexity is inevitable. Second, it behaves in a poor manner when there are no conjugate features in a set of features. Finally, the relationships between primitives must be well constrained one another in order to achieve a reliable solution. A highly abstract contextual information is a key to reaching a reliable solution.

From the experiments in this study, the following issues are identified for future research.

- A feature extraction algorithm which detects and interacts well with the physical boundaries must be explored. As shown in this study, wrong matches are always acquired when there are no corresponding feature primitives.
- The higher the abstraction of feature description, the better and faster the relational matching reaches a solution. A way of acquiring a high level of abstraction of feature description prior to relational matching must be explored.
- Some attributes in the primitives and relations are orientation variant. In order for the proposed relational matching scheme to be implemented for more general cases, those attributes should be replaced with orientation invariant ones.
- In this study, two descriptions are realtionally matched. The proposed relational matching scheme can be extended to match more than two descriptions simultaneously so that it could be utilized for multiple image matching problems such as automatic aerotriangulation.
- The computation complexity of relational matching in the form of a tree must be reduced by investigating other existing search methods such as relaxation technique, maximal clique and simulated annealing.
- The proposed relational matching scheme implemented in this research has the potential of being used for updating maps, object recognition and navigation. Further investigation and developments are required in the near future.

REFERENCES

Ballard, Dana H. and Brown, Christopher M. (1982) *Computer Vision.* Prentice-Hall, Inc.

Boyer, K. L., Kak, A. C. (1988) Structural Stereopsis for 3-D Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 10, No. 2, March, 144-166. ,

Burns, J. B., Hanson, A. R., and Riseman, E. M. (1986) Extracting Straight Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 8, No. 4, July, 425-455.

Cho, Woosug (1995) Relational Matching for Automatic Orientation. Ph.D Dissertation, The Ohio State University, Columbus, Ohio.

Doorn, B., Agouris, P., Al-Tahir, R., Stefanidis, A. and Zilberstein, O. (1990) Digital Stereo Matching: In Perspective. Technical Notes in Photogrammetry No. 10, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio.

Förstner, Wolfgang and Gülch, E. (1987) A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features. *ISPRS Intercommissiom Workshop*, Interlaken, June, 281-305.

Haralick, Robert M. and Shapiro, Linda G. (1992) Computer and Robot Vision. Addison-Wesley Publishing Company, Inc.

Moravec, H.P. (1977) Toward Automatic Visual Obstacle Avoidance, *Proceedings of the 5th Joint Conference on Artificial Intelligence*, Cambridge, 584.

Schenk, T. (1992) Algorithms and Software Concepts for Digital Photogrammetric Workstations. Report No. 418, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio.

Shahin, Fayez. (1994) An Automatic Correspondence for Digital Images Using Linear Features. Ph D Dissertation, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio.

Tang, L. and Heipke, C. (1994) An Automatic Procedure for the Relative Orientation of Aerial Images. Proceedings on Integration, Automation, and Intelligence in Photogrammetry, Remote Sensing and GIS. 3rd Liesmars Colloquium, Wuhan, October, 49-58.

Tsingas, Vassilios (1994) A Graph-Theoretical Approach for Multiple Feature Matching and Its Application on Digital Point Transfer. Munich Commission III, Institute for Photogrammetry, University of Stuttgart, 865-871, Stuttgart, FRG.

Vosselman, George (1992) Relational Matching. Lecture Notes in Computer Science 628, Springer-Verlag, Berlin Heidelberg.

Zilberstein, Ofer (1992) Relational Matching for Stereopsis. *International Archives of Photogrammetry and Remote Sensing,* Commission III, Washington D.C., 711-719.