GRAPH-BASED MATCHING OF STEREO IMAGE FEATURES

Olaf Hellwich and Wolfgang Faig
Department of Surveying Engineering
University of New Brunswick
Fredericton, N.B.
Canada, E3B 1W4
olaf@atlantic.cs.unb.ca

Abstract:

A feature-based stereo vision method was developed. It utilizes a neighbourhood graph in order to control the search for matching edge pairs. Information about neighbourhood relations supports the establishment and the reliability of matches. The matching method does not exploit any geometric constraints and is therefore applicable to a wide range of related problems, such as finding correspondence between images and maps. The most important advantages of the matching algorithm are its independence from human interaction and the use of curved edge segments.

The proposed method was implemented in the form of a modular software package on a Sun workstation using the XWindow user interface. The complete software package forms a softcopy photogrammetric program package. Its modular nature provides the possibility to exchange parts in order to adjust it to future technological developments, and use it for further stereo vision research. The features, advantages and problems of the proposed method are discussed. Digitized stereo image pairs for the investigation of automobile accidents are used to illustrate the method. For objects with distinctive edges, 80 to 90% of the resultant densely distributed feature pairs were found to be correct.

Key Words: Feature Extraction, Image Matching, Stereoscopy

1. Introduction

Since computers are getting cheaper and faster, and computer vision research has made progress in various vision tasks, it appears to be feasible to solve the stereo vision problem of applied photogrammetry without the use of stereo plotters, based on digital image processing. This goal is being approached with the development of digital photogrammetric workstations which solve the stereo vision problem fully digitally, and to a varying degree, automatically (Schenk and Toth, 1992; Dowman et al., 1992; Miller et al., 1992).

The proposed matching method for digitized stereo image pairs is mainly intended to be used in industrial close-range photogrammetry. As examples for non-metric imagery, images of cars which were involved in accidents were evaluated.

In order to provide a user friendly tool which can be applied by non-specialists, the amount of user intervention during the matching process has to be limited. However, as it cannot yet be expected to solve all matching problems with the existing means of computer vision, the possibility for interactive verification and correction of the matching results has been provided. These two aspects secure the usefulness of the developed software for practical applications.

1.1 Previous Work

Originally feature-based approaches to the stereo vision problem were considered for application. In the following discussion the extraction and matching of edges is reviewed.

McIntosh and Mutch (1988) developed a matching method for straight lines. The lines are extracted from line-support regions which provide parameters describing the edges. The matching algorithm compares the parameters of candidate edges for matching. A match function results in a high similarity value, when the two edges belong to a correct match. Each parameter of the edges contributes to the similarity value according to an empirically set weight. The algorithm allows the use of the epipolar constraint, but does not rely on it.

Pong et al. (1989) developed a matching method for topographic structures consisting of edges and regions. Arc segments, i.e. long thin edge regions which are not necessarily straight, are extracted from the images and linked by a region growing process. The regions, separated from each other by the arc segments, are found by assigning a unique label to each maximally connected group of non-arc pixels. In both images, sequences of edges and regions on an epipolar line are generated. The algorithm establishes a match between these sequences. The matching criteria are the orientation of the edges and the intensity of the regions. After the evaluation of all epipolar lines, the algorithm finds unique matches for the edge or region segments.

Ayache and Faverjon (1987) proposed a method of prediction and propagation of hypotheses applied to a graph-based description of images. The images are first reduced to neighbourhood graphs of edge segments. The line segments are generated by a polygonal approximation of edge elements in the images. The neighbourhood relations between the edge segments are established with the help of a bucketing technique. An image is subdivided into rectangular grid meshes. All edges which belong partially or completely to the same grid mesh are considered neighbours. The constraints used by the stereo matching algorithm are: epipolar constraint, geometric similarity constraint, continuity constraint and uniqueness constraint. The experimental results of Ayache and Faverjon show weaknesses where the objects or object parts have curved contours.

A graph-based matching approach more elaborate than Ayache and Faverjon's technique was presented by Horaud and Skordas (1989). An image is described in form of a relational graph. In the matching process not only the features and their attributes are evaluated, but also the relationships between nearby features are considered. In order to find a mapping function between the images which considers relations between features in

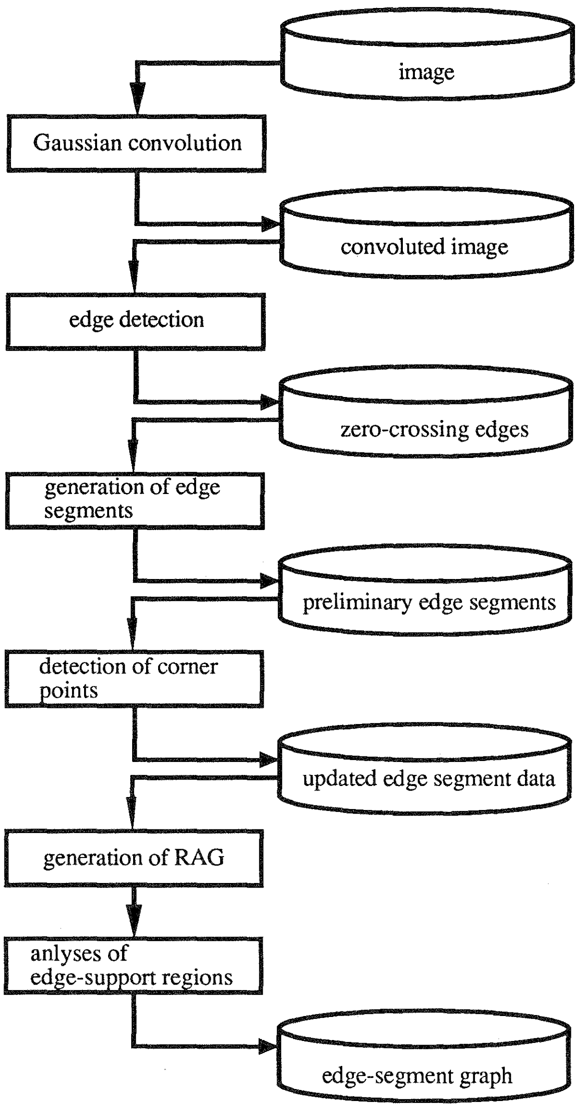


Fig 1.1. The matching procedure. Generation of a neighbourhood graph.

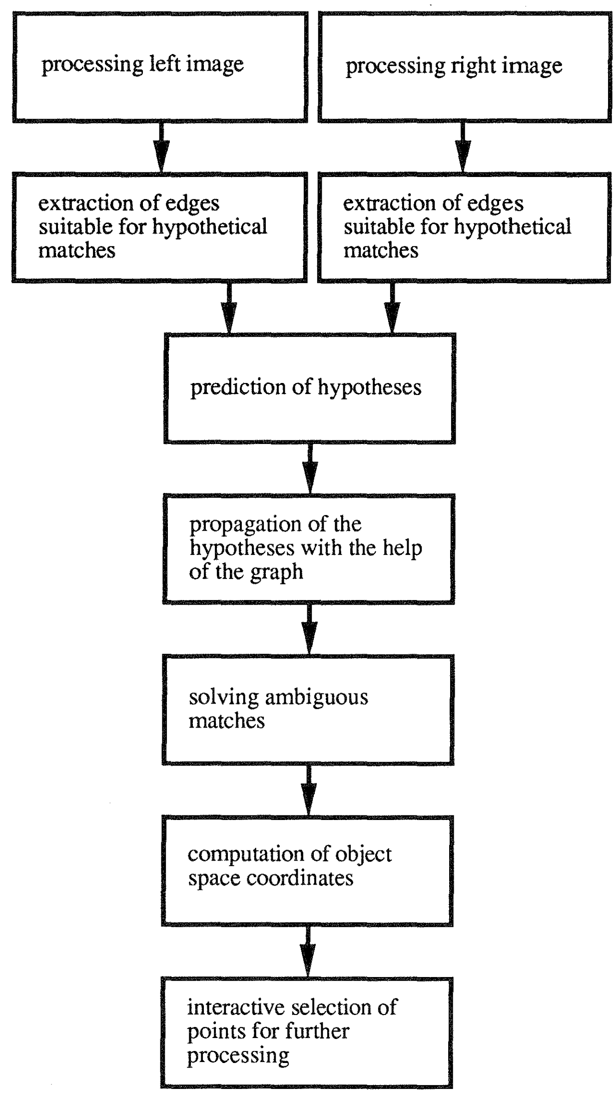


Fig. 1.2. The matching procedure. Graph-based matching and derivation of object space coordinates.

the left and in the right image, a correspondence graph is built. The stereo correspondence problem is therefore formulated in form of a search for maximum cliques in a graph. Horaud and Skordas perform an exhaustive as well as a simplified heuristic search in the correspondence graph.

The approaches reviewed here had to be improved concerning the matching of curved lines. An algorithm was developed that avoids the use of the epipolar constraint and nevertheless provides matches efficiently.

An overview about the developed stereo method is shown in flow diagrams. First, both images are processed separately (see Fig. 1.1). When two edge neighbourhood graphs have been generated, then the matching is conducted (see Fig. 1.2).

2. Edge Detection

An edge in a digital image occurs when the intensity values of neighbouring pixels are significantly different (Haralick, 1984). The edge detection is performed in a two step procedure. First the image is smoothed by Gaussian convolution. Then the edges are computed. Here, Haralick's (1984) method for the detection of step edges from zero crossings of the second directional derivative was implemented according to the suggestions given by Hummel and Lowe (1989).

The main problem in the detection of edges from zero crossings of the second derivative of the brightness function is due to the fact that the second derivative is zero for step edge pixels as well as for pixels where the brightness function values are constant. Therefore, the zero-crossing pixels have to be tested for edge quality. This is done according to Berzins (1984) who has noted that a zero crossing is a gradient maximum, if and only if

$$\frac{\partial f}{\partial n} \cdot \frac{\partial^3 f}{\partial n^3} < 0 \qquad (2.1)$$

where $\frac{\partial}{\partial n}$ is the directional derivative.

3. Segment and Corner Generation

The edge pixels and non-edge pixels are given in the form of a binary image. It is not explicitly known how edge pixels are connected to long chains of edge pixels, and where these chains begin or end. Therefore, it is necessary to generate chains of connected edge pixels.

A chain of pixels along an edge will be called an edge streak. We define a corner as a discontinuity in the direction of an edge streak, which subdivides the edge streak into edge segments. Corner points of the edge streaks support the finding of correct matches.

3.1 Generation of Edge Streaks

Generation of edge streaks can be organized in the form of a minimum cost search (Shu, 1986). The general case of combining pixels to form edge streaks can be explained as follows (see Fig. 3.1):

Pixel P is part of an edge streak. The combination of P with one of the neighbouring pixels p_0 to p_7 which are not yet part of an edge streak is considered.Thus, the costs C of the combinations (P, p_0), (P, p_1), ... (P, p_7) have to be computed. The combination (P, p_{min}) which results in the minimum cost C_{min} among the costs $C(P, p_i)$ is candidate for a link operation. If the cost $\tilde{C}(P, p_{min})$ is less than a user-defined threshold C_t, the link (P, p_{min}) is conducted.

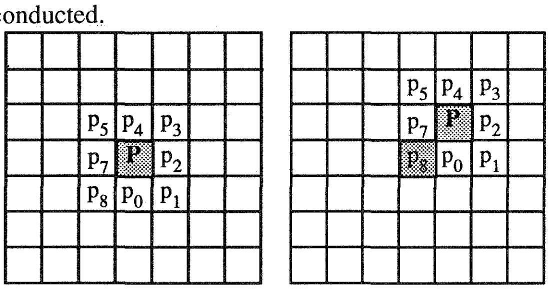


Fig. 3.1. Minimum-cost search for edge streak generation. The search operation is shown twice.

As can be seen, two cost functions C and $\tilde{C}$ are used in order to decide about a link (P, p_i). The cost C is determined by the difference in gradient direction, the difference in gradient magnitude, a non-edge pixel property and the divergence from straightness of the generated edge streak. The cost components resulting from the non-straightness of an edge streak is only considered for the finding of the minimum cost C_{min}, but not for the comparison with the threshold C_t, as there are no objections against the generation of an edge streak that includes a corner point.

3.2 Detection of Edge-Streak Corners

Corner points are of special importance for the derivation of correct matches between edge segments from two images. Firstly, corner points are descriptive parameters which contribute to the distinctiveness of an edge feature. Thus, they improve the feature matching. Secondly, once a match of two segments has been established, corner points can serve the purpose of mutually fitting the two edge curves in order to get point-oriented information.

The corners of the chain-coded edge streaks are detected according to the algorithm of Beus and Tiu (1987). It is based on an algorithm developed by Freeman and Davis (1977).

4. Generation of an Edge Neighbourhood Graph

In this approach the distribution of edges over an image and the relations between edges are controlling and supporting the matching.

The algorithm for edge neighbourhood detection is based on the philosophy that an edge segment "owns" all pixels of an image which are situated closer to it than to any other edge segment. The set of pixels owned by an edge segment constitutes an edge region. The Region Adjacency Graph RAG of the edge region image is the desired edge neighbourhood graph.

The edge regions are found by growing layers of neighbouring pixels around the edge segments. At the beginning of the process all edge segment pixels are considered initial pixels of the edge regions. The set of four-neighbouring pixels of the edge-region pixels which is not yet integrated into the region, is the layer of pixels surrounding an edge region. The region growing process successively integrates layers into all regions. First, one layer is grown around each region. The pixels of a layer are only integrated into a region when they do not already belong to another region. This region would be considered being a next neighbour. After one layer is grown for all segments the sequence of regions is reversed. This makes up for the fact that regions which grow first, potentially grow faster than regions which grow later. Subsequently, further layers are grown, and the process is repeated until all image pixels belong to certain regions, or until a predefined maximum number of layers was handled. The latter provides the possibility not to include the whole image into processing, but to stop the layer growing when the distances between edge segments are too large for neighbourhood relations.

The RAG is build according to the algorithm shown in pseudo-code in Fig. 4.1. It could also be regarded as a contour-based dilation algorithm for non-binary images. Fig. 4.2 shows the final state of a RAG generation.

```
1.  For all regions {
        If a pixel of another region is contained
        in the surrounding layer of the region {
            Register that this region is adjacent to the
            processed region.
        }
        Grow the region such that the surrounding layer is
        integrated into the region according to the rule
        described above.
    }
2.  If all image pixels that are to be evaluated
    are members of an edge region, exit.
3.  Invert the order of the regions and execute step 1. again.
```

Fig 4.1. Algorithm for the generation of a Region Adjacency Graph.

4.2 Establishing Edge Parameters

Matches of edge segment pairs are established by comparing parameters which describe the edge segments. All of these edge parameters are contained in the brightness function at and in the vicinity of an edge segment. Scanning the vicinity of an edge segment, the search for next neighbours can be used in order to extract descriptive and relational edge parameters from the brightness function.

Descriptive parameters describe the properties of an edge segment and, hence, make them distinctive from other segments. The descriptive parameters are derived by an analysis of edge-support regions. The problem of which edge pixel belongs to the edge support region and

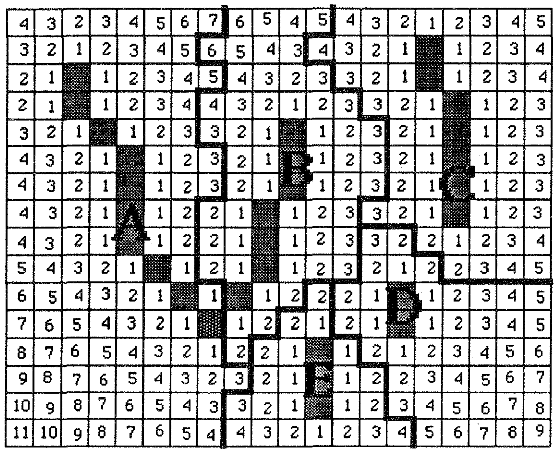


Fig. 4.2. Final state of the region growing by layers. The edge segment pixels are shown in gray. The labels surrounding the edge segments are the layer indices. The bold lines are borderlines between edge regions. Next neighbours are A-B, A-E, B-C, B-D, B-E, C-D and D-E.

which does not, is solved by evaluating the cost of the integration of the pixel in question into the edge-support region.

Relational parameters give information about the neighbourhood relations between an edge and adjacent edges. They have either topologic or geometric content. Relational parameters are determined by the next-neighbour relations of the edge segments.

4.2.1 Evaluation of Edge-Support Regions

The edge-segment pixels as well as the pixels in the vicinity of an edge segment form the edge-support region which characterizes an edge. The edge-support property of pixels has to be determined efficiently. Here, the arbitrary decision was made that only pixels belonging to an edge region established during the generation of the RAG can be members of the edge-support region. This enables us to establish the edge-support regions along with the RAG. Each pixel that is integrated into an edge region is tested for membership in the edge-support region. This is done by a cost evaluation similar to the minimum-cost search conducted during the generation of edge segments. If the cost of integration is less than a user-defined threshold, the pixel will be regarded as an edge-support pixel and will take part in a statistical evaluation for the determination of descriptive edge parameters.

The evaluated cost components are

1. the difference δ_g of the gradient direction of the pixel in question and the average of the eight neighbouring pixels,
2. the difference δ_{gs} in gradient direction between the pixel evaluated and the closest pixel of the edge segment (chain),
3. the difference Δ_{ms} in gradient magnitude between the pixel in question and the closest pixel of the edge segment,
4. the distance d of the pixel from the closest pixel of the edge segment.

The fourth cost component is not self-explanatory. It was introduced as a simple way of keeping the edge support region limited to the immediate vicinity of the edge segment. Otherwise, the edge-support regions are often fragmented and partially unconnected with the edge segment.

The cost is computed according to

$$c = c_g + c_{gs} + c_{ms} + c_d \qquad (4.1),$$

where

$$c_g = 100 \cdot \delta_g \qquad (4.2),$$

$$c_{gs} = 100 \cdot \delta_{gs} \qquad (4.3),$$

δ_g, δ_{gs} in radians,

$$c_{ms} = \Delta_{ms} \qquad (4.4),$$

Δ_{ms}: difference in gradient magnitude in percent of the gradient magnitude of p_n,

$$c_d = \Delta x^3 + \Delta y^3 \qquad (4.5),$$

Δx, Δy: x-, y-difference between pixel and closest pixel of the segment chain.

Although the distance component c_d initially appears to be odd, it proved to be very useful as it is neglegibly small for short distances, but imposes a heavy influence on the cost of pixels far away from the edge segment.

The descriptive edge parameters derived from the edge-support regions are

1. maximum brightness,
2. minimum brightness,
3. contrast (maximum - minimum brightness),
4. width (support region size / segment length),
5. steepness (contrast / width),
6. average brightness in the edge-support region.

For reasons of completeness, the other descriptive edge parameters are also listed here, namely:

1. segment length (derived during edge-streak generation and corner detection),
2. segment direction (direction from beginning to end of a segment),
3. segment curvature (derived during corner detection).

4.2.2 Parameterization of Neighbourhood Relations

The relational parameters of the edge description are determined when the next neighbourhood of two edge segments is detected as the RAG is generated. The relational parameters are

1. the closest distance between two pixels of the segment chains (i.e. the layer index),
2. direction from the center of the chain to the center of the neighbouring chain,
3. perpendicularity,
4. parallelity,
5. collinearity (This property is not detected precisely. If the edge segments belong to the same edge streak, they are assumed to be "collinear" for matching purposes.),

6. side (left or right).

5. Graph-Based Feature Matching

The fundamental idea for the matching of edge feature pairs is that any property of the edge features can be compared with the help of a match function. In this way the properties contribute to a measure of similarity of the candidate features. As no constraints have to be fulfilled, one or several properties that differ for candidate feature pairs do not prevent the algorithm from identifying a correct match, as long as the overall similarity can be proven with a high similarity value.

5.1 The Match Function

The comparison of candidates for matching is conducted with a match function which evaluates feature parameters (McIntosh and Mutch, 1988):

$$s = \frac{\sum_i (w_i \cdot v_i)}{\sum_i w_i} \qquad (5.1)$$

where v_i = similarity value for a parameter pair,
w_i = weight of parameter i.

The result of the match function is a similarity value s of the features compared.

The similarity values v_i of the parameters are computed according to the nature of the specific parameters. Generally, the computation is conducted according to (after McIntosh and Mutch, 1988):

$$v_i = \frac{min(|p_{jL}|, |p_{kR}|)}{max(|p_{jL}|, |p_{kR}|)} \qquad (5.2)$$

where p_{jL} and p_{kR} are the instantiations of the compared parameters.

For some parameters, equation (5.2) is not suitable.

The similarity value for segment directions t_L, t_R is computed as (McIntosh and Mutch, 1988):

$$v_t = \frac{\delta - |t_L - t_R|}{\delta} \qquad (5.3)$$

where t_L, t_R are the directions of the left and right segment in $\frac{rad}{100}$.

The value δ is interpreted as the maximum allowable difference between the directions compared. If the magnitude of the difference $|t_L - t_R|$ is larger than δ, then the similarity value v_t will be negative. δ is computed according to:

$$\delta = \delta_{max} \cdot \frac{50}{min(50, l_L + l_R)} \qquad (5.4)$$

where δ_{max} is set to $80 \frac{rad}{100}$,
l_L, l_R are the lengths of the left and the right segments.

This means that δ is generally $80 \frac{rad}{100}$; a value that has been determined empirically. δ is increased when the sum of the segment lengths is less than 50 pixels as directions of short segments agree considerably less than directions of longer segments.

For the propagation step of the generation of hypotheses, for both the left and the right edge segments L and R, "parent" nodes in the edge neighbourhood graphs are known. This provides also the possibility to compare relational parameters. The similarity value of two relations r_L, r_R between parent and neighbour nodes is computed according to:

$$v_r = \frac{1}{5}\left(\frac{\delta - |t_{rL} - t_{rR}|}{\delta} + \frac{min(d_{rL}, d_{rR})}{max(d_{rL}, d_{rR})} + C + P + S\right) \qquad (5.5)$$

where t_{rL} = direction between related edges in the left graph,
t_{rR} = direction between related edges in the right graph,
d_{rL} = distance between related edges in the left graph,
d_{rR} = distance between related edges in the right graph,
δ = computed value according to (5.4),

$$C = \begin{cases} 100 \text{ if both left and right relation are referring to edges belonging to the same edge streak} \\ 50 \text{ if both left and right relation are referring to edges not belonging to the same edge streak} \\ 0 \text{ otherwise} \end{cases},$$

$$P = \begin{cases} 100, \text{ if } r_L, r_R \in \mathbf{P} \\ 100, \text{ if } r_L, r_R \in N \\ 100, \text{ if} (r_L, r_R \notin \mathbf{P} \vee r_L, r_R \notin N) \\ 0, \text{ if} ((r_L \in \mathbf{P} \wedge r_R \in N) \vee (r_L \in N \wedge r_R \in \mathbf{P})) \\ 50 \text{ otherwise} \end{cases},$$

$$S = \begin{cases} 100; \text{ if } r_L, r_R \in \mathbf{L} \\ 100, \text{ if } r_L, r_R \in \mathbf{R} \\ 0 \text{ otherwise} \end{cases}.$$

Here $\mathbf{P} = \{r \mid r \text{ is a relation of parallel edge segments}\}$,
$N = \{r \mid r \text{ is a relation of perpendicular edge segments}\}$,
$\mathbf{L} = \{r \mid r \text{ is rel. of a parent node and a left child node}\}$,
$\mathbf{R} = \{r \mid r \text{ is rel. of a parent node and a right child node}\}$.

Thus, the similarity of two relations depends on the direction and the distance between the related segments, edge-streak membership and topology.

The parameters compared by the match function are determined by the domain to which the matching method is applied; in this case the application domain is stereo imagery. For other domains such as matching of digitized maps, the parameters p and, if necessary, the parameter similarity functions v_p could be adapted to the specific problems. Image geometry could also be taken into account as a parameter of the match function.

5.2 The Matching Procedure

The matching process consists of three steps: (1) the prediction of hypotheses, (2) the propagation of the predicted hypotheses based on the edge neighbourhood graph, and (3) the solution of conflicts between concurrent connected components. Ayache and Faverjon (1987) used a similar algorithm.

5.2.1 The Prediction of Hypotheses

In the prediction step a set of particularly obvious matches is attempted. First, sets of comparatively distinctive edge segments are extracted from the edge neighbourhood graphs using a selection function that weights the segment parameters.

The selection function derives a measure of distinctiveness d for each edge segment. The averages of the weight function values of the user-defined parameters are computed:

$$d = \frac{\sum_{i=1}^{n} w_i(p_i)}{n} \qquad (5.6)$$

where $p_i =$ argument of parameter i,
$w_i() =$ weight function for parameters p_i,
$n =$ number of parameters evaluated.

A weight function w_i is given in digitized form as a set of x-values defining a standard function curve.

All possible combinations of significant edge segments (s_{iL}, s_{jR}) from the left and right edge neighbourhood graphs are compared using the match function (5.1). The set of mutually best matches $\boldsymbol{M}$ is considered for further processing. "Mutually best" means that both the assignment of the best matching right segment to the left segment as well as the assignment of the best matching left segment to the right segment result in the same pair of edge segments. When the similarity measure $m(s_{iL}, s_{iR})$ is greater than a user-defined threshold, the match is accepted as an initial match of a hypothesis.

5.2.2 The Propagation Algorithm

In the propagation step, connected components are generated by searching the edge neighbourhood graphs for further potential matches, starting with the initial hypothetic matches. The neighbouring edges of the initial match (p_L, p_R) are compared in all combinations. Two classes for the acceptance of a match found are defined as follows:

I. matches which

A. are elements of the set of mutually best matches $\boldsymbol{M}$ or matches where one segment s_i is matched to a segment s_j which is matched to a segment s_k belonging to the edge streak to which s_i also belongs,

and

B. have a match function value greater than a threshold t_1:

$$m(s_L, s_R) > t_1$$

The matches are regarded as valid matches of a connected component.

II. matches which fulfill condition A as before and

B. have a match function value greater than a threshold t_2, and less than the threshold t_1:

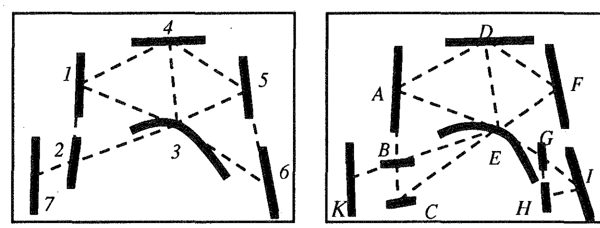
$$t_1 > m(s_L, s_R) > t_2$$

These matches are not considered being members of the set of connected components, but they are used for further propagation of matches in order to find more matches of class I.

The propagation strategy is based on a breadth-first search in a tree. The search starts by comparing neighbours of the segments of an initial hypothetical match. Thus, the initial match is the root node of a tree. Its child nodes are the matches of classes I and II among the neighbours of the initial segments. As a breadth-first search is conducted in the graph, the first generation child nodes are established in direct sequence. Then, the second generation child nodes are searched by comparing all neighbours of a first generation match. This means that the breadth-first tree degenerates to a linear list with generation indices.

In every generation, all matches of classes I and II become further "tree" nodes of the next generation, if they are not already part of the tree as a node of an earlier or of the same generation. The propagation is ended as soon as no further tree nodes can be generated, or if there are more than three generations of class II nodes in sequence.

Fig. 5.1 shows two neighbourhood graphs to be matched, Fig. 5.2 gives an example for the generation of the propagation tree, and Table 5.1 shows the class I and class II matches established during the propagation.



left image right image

Fig. 5.1. Two edge neighbourhood graphs.
The most distinctive edges are *3* and *E*. An initial match of a hypothesis could be the match (*3*, *E*).

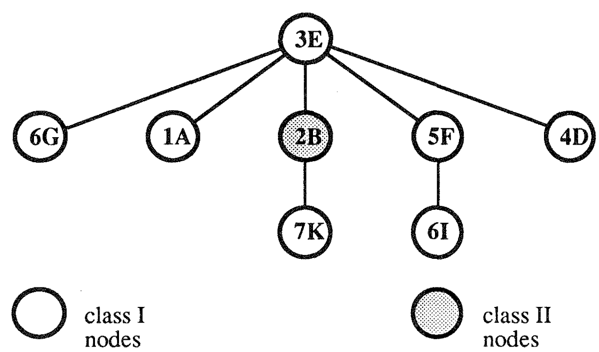


Fig. 5.2. The propagation tree.
Even though the edge segments 2 and B do not seem to match very well (see difference in direction), match (*7*,*K*) was found.

parent node	class I matches	class II matches	discarded matches
(3,E)	(6,G) (1,A) (5,F) (4,D)	(2,B)	
(2,B)	(7,K)		
(5,F)	(6,I)		
			(6,G)

Tab. 5.1. Matches from hypothesis propagation.
Match (*6*, *G*) was overwritten by match (*6*, *I*).

During the build-up of the propagation tree, the potential matches of class I are considered to belong to a connected component. Ambiguous assignments are solved at the end of the processing of a hypothesis. If multiple matches occur between different edge streaks, the match with the greater similarity value is preferred.

When a match which is found during the propagation of a hypothesis is the initial match of another hypothesis, the other hypothesis is discarded. This saves computation time as the second hypothesis would result in a connected component similar or even equal to the one generated.

According to the details and features described before, the propagation algorithm was implemented as shown in Figures 5.3 and 5.4.

```
for all hypotheses {
    treat initial match ( pL, pR) as root node;
    while tree not completely processed {
        Match_Neighbours_of (pL, pR);
        for both edge neighbourhood graphs {
            for all segments {
                discard ambiguous matches;
            }
        }
    }
}
```

Fig. 5.3. Propagation algorithm.

```
Procedure Match_Neighbours_of (pL, pR) {
    compute match function values for NL x NR;
    for all left neighbours {
        if mutually matched {
            if ( m(sL, sR) > t1) {
                if (sL, sR)∉ tree, Make_Tree_Node;
                Make_Match;
                check for not yet processed hypothesis;
            } else if (m(nL, nR) > t2) {
                if (sL, sR)∉ tree, Make_Tree_Node;
            }
        } else if sL is matched to a right segment sR which is
        matched to another segment tL of the edge streak of
        sL {
            if ( m(sL, sR) > t1) {
                if (sL, sR) ∉tree, Make_Tree_Node;
                Make_Match;
                check for not yet processed hypothesis;
            }
        } else if sR is matched to a left segment sL which is
        matched to another segment qR of the edge streak of
        sR{
            if ( m(sL, sR) > t1) {
                if (sL, sR) ∉tree, Make_Tree_Node;
                Make_Match;
                check for not yet processed hypothesis;
            }
        }
    }
}
```

Fig. 5.4. Algorithm for the procedure "Match_Neighbours_of".

The procedure Match_Neighbours_of does the comparision of neighbouring nodes of an established match. Procedure Make_Tree_Node adds a node to the propagation tree. The procedure Make_Match adds a match to the edge neighbourhood graph. During the tree generation, several entries per edge segment can occur. Afterwards, the main procedure discards the ambiguous matches in the connected component.

5.2.3 Solution of Conflicts Between Different Connected Components

Which of the conflicting hypotheses is preferred is based on the strength of prediction of a hypothesis. The connected component with the largest number of matched segment pairs is considered to be the correct one.

When two matches are conflicting, the match belonging to the stronger hypothesis is always preferred. Then the strength of the weaker hypothesis is reduced by one. During the process of solving contradictions, only the original strengths of prediction are compared. This avoids that the order in which the hypotheses are evaluated for ambiguous matches influences the result of the process.

After the solution of conflicts between concurring connected components, the remaining strengths of prediction are compared with a user-defined threshold. The threshold should be set according to the image type and the number as well as reliability of the matches needed. Hypotheses weaker than the thresholding strength are discarded completely, as those hypotheses often are not based on correct matches (Ayache and Faverjon, 1987).

6. Pointwise Matching and Object Space

6.1 Establishment of Point Correspondence

Once the edge segments are matched, correspondence between linear groups of points is established. Yet it is not determined how the points on the edges form pairs of matching points. The task of finding these point pairs is non-trivial, as there are several influences which are resulting in a nonlinearity of the function that maps points from one segment curve to the other. These influences include perspective distortion and occlusion. The parameters controlling the mapping between two chain-coded curves are not considered rigorously here, but three simple cues are used to establish a pointwise correspondence.

These methods assume that the segment points have certain distinctive characteristics which allow for the determination of a measure of fitting for different positions while the shorter curve is sliding pixel by pixel along the longer curve. It is always assumed that the transformation function from one chain-coded curve to the other one is linear with slope 1. This means that both curves are assumed to have the same scale and that chain code elements are assumed to be equally long. Thus, the assignment of matching pixels is one-to-one.

The cues used for curve fitting are an alignment according to corresponding corner points, an alignment according to perpendicular neighbouring edge segments and a least squares fitting. They are applied in the order in which they are mentioned here. If one method is applied successfully, the other methods are not pursued.

6.2 Area-Based Matching with Subpixel Accuracy

The curve fitting results in a pixel by pixel correspondence between the curves. An area-based matching algorithm with subpixel accuracy is applied to the end points of the edge segments and to one point in the middle of long segments.

The matching algorithm of Kanade and Okutomi (1990) was implemented. The algorithm iteratively derives disparity updates and uses a varying window size in order to minimize the uncertainty of the disparity.

6.3 Interior Orientation and Model Formation

For the interior orientation, the fiducial marks on the image frame are digitized by the operator. Coordinates are transformed from the pixel coordinate systems to the image coordinate systems by affine transformations. The formulae for this affine transformation are given, e.g. by Moffit and Mikhail (1980, pp. 291-295, pp. 592-593).

As the images have been taken with a terrestrial photogrammetric camera, the parameters of the relative orientation are known. The image coordinates of corresponding points which determine the disparities d_x and d_y as

$$\begin{aligned} d_x &= x_L - x_R \\ d_y &= y_L - y_R \end{aligned} \qquad (6.1)$$

can be used according to the collinearity condition to compute object space coordinates.

The formulae for the computation of spatial coordinates X, Y, Z according to the collinearity condition are given e.g. by Kraus (1982, Chapters 2.3 and 4.11).

Final results of the matching method are the spatial coordinates of points on the surface of the object. They can be used for further applications.

6.4 A Set of Uncertainty Measures

In order to estimate the quality of the matching results, there is a need for a reliability measure. The problem to provide such a measure for a vision algorithm that encounts both for accuracy as well as robustness is known to be hard (Förstner and Ruwiedel, 1992). In this method, a measure which is meaningfully including all aspects of the matching (e.g. edge detection, feature matching and subpixel-accurate point matching) in a coherent numeric system cannot be provided. Instead of that a set of uncertainty measures each making a statement about the outcome of a specific step in the processing is available.

The match function (5.1) gives a similarity measure s about the overall similarity of the two edge segments compared. The higher the similarity measure the more accurately the edge segments correspond. In this sense a high s-value can indicate a high probability of correctly matched edge features. Nevertheless, it does not prove a correct match. Correctly matched edges will most probably have a high similarity value. However, if the object has several similar features, e.g. regular patterns, wrongly matched features will also obtain a high s-value.

To draw conclusions about the correctness of a match becomes much easier when in addition to the similarity measure of feature pairs the strength of a connected component is considered. When many matches have been found during the propagation of a hypothesis, the probability that these matches are correct is high. This property is used in order to solve ambiguities occurring between different hypotheses.

The measure available to assess the quality of a matched pair of points is the uncertainty of the disparity update $\sigma_{\Delta d}$ as a result of the area-based matching algorithm of Kanade and Okutomi (1990). With the help of $\sigma_{\Delta d}$ the accuracy of the object space coordinates can be estimated.

Summarizing, it can be stated that the similarity measure of the feature matching in combination with the strength of the connected component allows conclusions about the reliability of the matching, whereas the uncertainty of the disparity mainly contains information about the accuracy of the matching. A more comprehensive investigation about the robustness of object space coordinates derived from matched edge features was done e.g. by Faugeras et al (1992). The derivation of meaningful reliability measures directly resulting from a matching method without any comparison with results of other methods or true data has to be left to future research. Methods and approaches to solve parts of the problem can be found in Förstner and Ruwiedel (1992).

7. Implementation and Experimental Results

The developed feature-based matching method was implemented on Sun workstations using the X-Window user interface and applied to imagery for car accident investigation. Final goal of the measurements was the determination of the depths of dents in the car bodies.

7.1 The AMORPH Stereo Vision Software Package

The amorph stereo vision system provides a digital photogrammetric software package. Although its main components are determined by the implementation of the proposed stereo vision method, it contains the tools to interactively perform digital photogrammetric operations, such as visualization and modification of digital images; input, selection and subpixel matching of corresponding points; interactive verification and correction of the matching results; interior orientation and model formation as well as selective output for further processing. Thus, the software can be used without depending on the developed stereo method. The main areas of application of the AMORPH software are basic digital photogrammetry and the use as an environment for stereo vision research.

The AMORPH software is designed in a strictly modular manner. General enhancements as well as modifications of the graph-based stereo vision method (e.g. based on recent research results) are easily implementable.

7.2 Accident Investigation Imagery

Images of cars are characterized by well-determined curved edges of various shapes as well as specular reflections in some parts. All images were taken outdoors under common conditions for close-range photogrammetry. No special measures were observed. The images were taken with a 24 x 36 mm^2 camera on colour slide film. They were digitized with an Apple McIntosh slide scanner using the DeskScan software for the FrameGrabber card. The images have a resolution of about 600 x 400 pixels.

The first picture shows the image pair of a grey car (Fig. 7.1). The damaged door of the car is used as the object of demonstration.

The generated edge neighbourhood graph can be seen in Fig. 7.2. Lines symbolize the edge segments. The underlying grey regions were generated by the layer growing algorithm in order to find neighbouring regions. Region neighbourhoods are indicated by dashed lines. The pixels with dots surrounding the edge segments are the edge-support regions from which the edge parameters are derived.

The final matches are displayed in Fig 7.3. The match numbers enable the alert eye to identify matched edge segments. The unlabeled edge segments are unmatched.

Another image pair can be seen in Fig. 7.4 while Fig. 7.5 shows the resulting matches on the object of interest.

Table 7.1 gives the computation times for each processing step on a Sun sparc IPX workstation. Unfortunately, the times are influenced by partially uncontrollable parameters of network performance, as some data had to be transferred from a remote station. However, it is obvious that the matching of graphs is extremely fast in comparision to the preceeding processing of the single images.

Intensive tests of the matching method resulted in about 80 to 90% correct matches among the matches found on the objects of interest.

8. Conclusions and Recommendations

A feature-based stereo matching method was developed using a graph structure to control the search for matches and to exploit relational properties for the identification of matches. The features used are edges of variable shapes.

Fig. 7.2. Edge neighbourhood graph.

Special features of the matching method are

1. *the use of curved edge segments*
2. *an efficient method for the generation of edge neighbourhood graphs and the analysis of edge-support regions*
3. *the avoidance of the epipolar constraint for objects with distinctive edges*

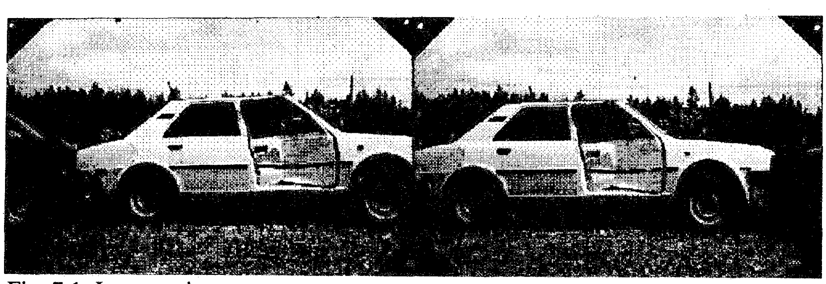


Fig. 7.1. Image pair

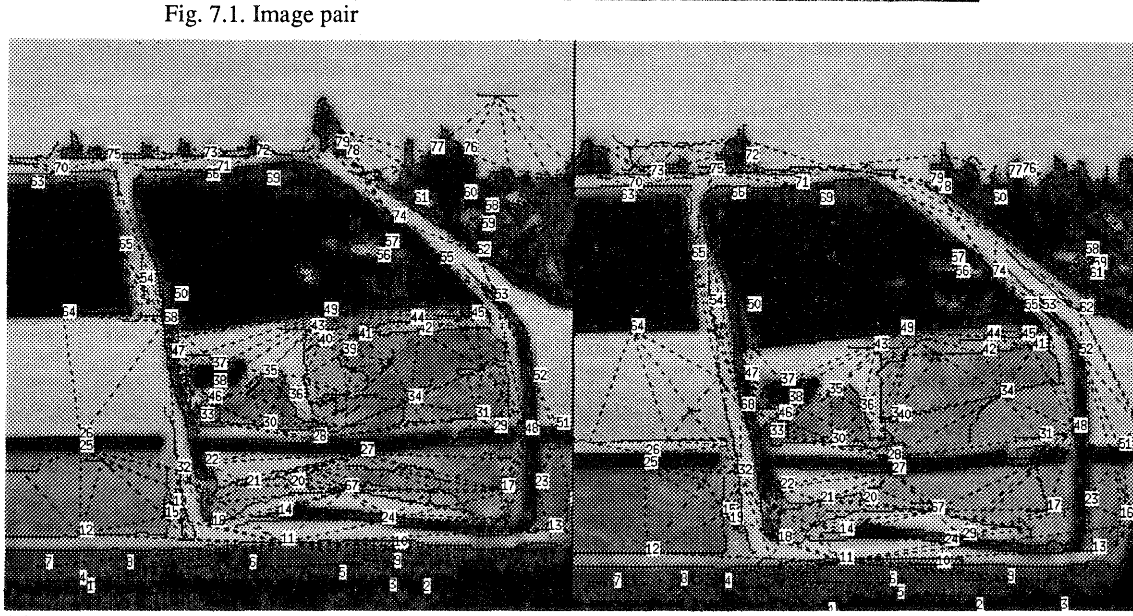


Fig. 7.3. Matched segments

	processing step	time [s]
	Gaussian convolution computation of zero crossings edge streak generation unselection of insignificant edge streaks corner detection generation of edge neighbourhood graph	22.0 10.9 4.7 1.7 2.0 36.3
Σ	processing 1 image	77.6
	processing 2 images	155.2
	prediction propagation establishing point correspondence	0.1 3.1 1.0

Table 7.1. Computation times for an image of 230 x 230 pixels on a Sun sparc IPX station.

8.1 Future Improvements

The generation of edge streaks as well as the detection of chain-code corners proved to be rather sensitive to changes in the image intensity function. Thus, the path of an edge streak as well as the detectability and detection of a corner at a junction in the image is uncertain. This problem can be substantially reduced by an extension of the corner detection towards the more general idea of junction nodes in the graph structure. Junction nodes are points where several edge segments can begin or end. Edge streak corners are considered being a type of junction nodes. The other type is a T-junction, where one edge streak begins or ends on another edge streak.

The concept of the matching method is the establishment of matches by comparing descriptive and relational parameters. This way of matching works well as long as the parameters are distinguishable. As soon as several edge segments are equal in several parameters (e.g. in an image of a brick wall), minor disturbances of noise or texture edges in the graph structure weaken the relational information which is one of the few distinguishable parameters in the propagation. The traversal of the graph using noise edge pairs prevents the algorithm from finding correct matches. The problem can be solved with the help of an evaluation of disparity gradients. It is assumed that disparity changes slowly when surfaces are continuous. This property is also valid for one side of occluding edges. It can be used as a relational parameter during the propagation step. Disparity gradients between the match to be established and the last match of class I among the predecessor nodes in the propagation tree should be small in order to support the evidence for a correct match. For the disparities in y-direction this results in an integration of epipolar geometry into the match function, as y-disparities should be zero while the epipolar lines are horizontal.

References

Ayache, N. and Faverjon, B. [1987], "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments", *International Jounal of Computer Vision*, pp. 107-131.

Berzins, V. [1984], "Accuracy of Laplacian Edge Detectors", *Computer Vision, Graphics and Image Processing*, Vol. 27, pp. 195-210.

Beus, H. L. and Tiu, S. S. H. [1987], "An Improved Corner Detection Algorithm Based on Chain-Coded Plane

Fig. 7.4. Image pair

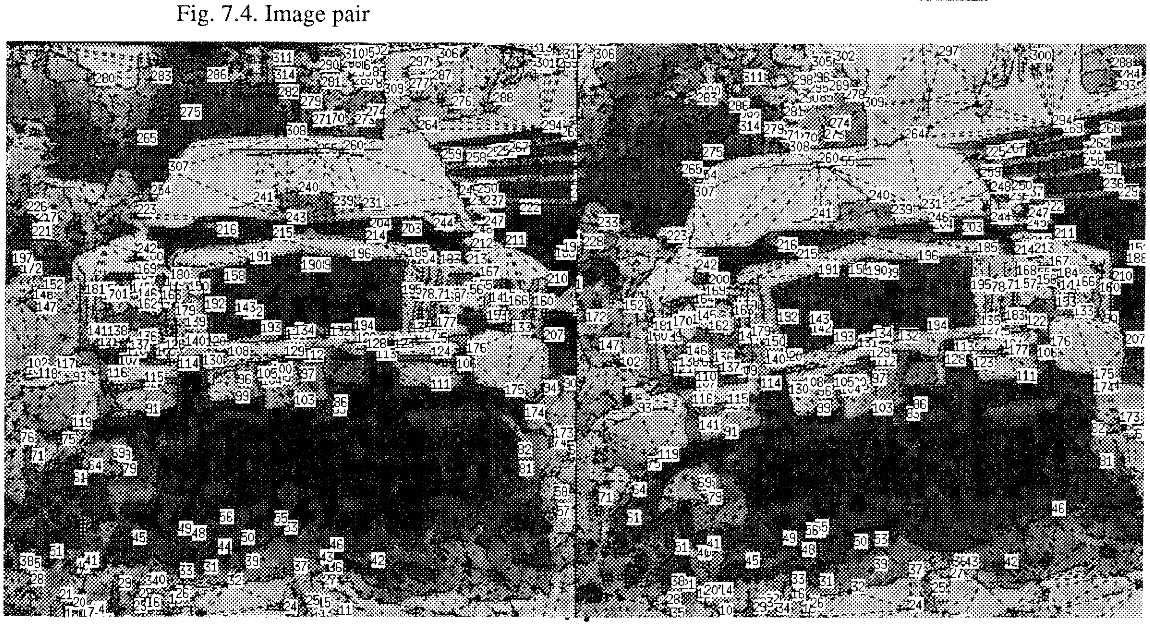


Fig. 7.5. Matched segments

Curves", *Pattern Recognition*, Vol. 20, No. 2, pp. 291-296.

Dowman, I. J., Ebner, H., Heipke, C. [1992], "Overview of European Developments in Digital Photogrammetric Workstations", *Photogrammetric Engineering and Remote Sensing*, Vol. LVIII, pp. 51-56.

Faugeras, O., Fua, P., Hotz, B., Ma, R., Robert, L., Thonnat, M., Zhang, Z. [1992], "Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms", in: Förstner, W. and Ruwiedel, S. (Eds.), *Robust Computer Vision*, Herbert Wichmann Verlag, Karlsruhe 1992.

Förstner, W. and Ruwiedel, S. (Eds.) [1992], *Robust Computer Vision*, Herbert Wichmann Verlag, Karlsruhe.

Freeman, H. and Davis, L. S. [1977], "A Corner-Finding Algorithm for Chain Coded Curves", *IEEE Transactions on Computers*, pp. 297-303.

Haralick, R. M. [1984], "Digital Step Edges from Zero Crossing of Second Directional Derivatives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 1, pp. 58-68.

Horaud, R. and Skordas, T. [1989], "Stereo Correspondence Through Feature Grouping and Maximal Cliques", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 11, pp. 1168-1180.

Hummel, R. and Lowe, D. [1989], "Computational Considerations in Convolution and Feature-Extraction in Images", in: Simon, J. C. (ed.), *From Pixels to Features*, Proceedings of a workshop held at Bonas, France, 22 - 27 August 1988, pp. 91-102.

Kanade, T. and Okutomi, M. [1990], *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment,* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Kraus, K. [1982], *Photogrammetrie*, Band 1, Grundlagen und Standardverfahren, Ferd. Dümmlers Verlag, Bonn.

McIntosh, J. H. and Mutch, K. M. [1988], "Matching Straight Lines", *Computer Vision Graphics and Image Processing*, Vol. 43, pp. 386-408.

Miller, S. B., Helava, U. V., Devenecia, K. [1992], "Softcopy Photogrammetric Workstations", *Photogrammetric Engineering and Remote Sensing*, Vol. LVIII, pp. 77-84.

Moffit, F. H. and Mikhail, E. M. [1980], *Photogrammetry*, 3rd edition, Harper & Row, New York.

Pong, T.-C., Haralick, R. M., Shapiro, L. G. [1989], "Matching Topographic Structures in Stereo Vision", *Pattern Recognition Letters*, Vol. 9, pp. 127-136.

Schenk, T. and Toth, C. K. [1992], "Conceptual Issues of Softcopy Photogrammetric Workstations", *Photogrammetric Engineering and Remote Sensing*, Vol. 58, No. 1, pp. 101-110.

Shu, S.-P. [1986], *ISES - An Image Segmentation Expert System for Aerial Photographs Based on the Use of Edge and Texture Features*, Doctoral Thesis, Rensselaer Polytechnic Institute, Troy, NY.