Interpreting Severe Occlusions in Region-Based Stereo Vision

by

Vladimir Tucakov

B.Sc. (honors, Computer Science) Brock University 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

we accept this thesis as conforming
to the required standard

The University of British Columbia

December 1997

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
Vancouver, Canada

Date Dec 23 1997.

DE-6 (2/88)

Abstract

This thesis presents a theory of correspondence matching in region-based stereo vision under severe occlusions. The general purpose of stereo vision is to perform distance measurements based on correspondence between slightly offset cameras. Region-based stereo vision performs matching in terms of regions in each image. Regions are generally obtained using a segmentation algorithm and matching is done by comparing the properties of the regions. The goal of this thesis is to analyze and explicitly model the effects of occlusions on the region-based stereo matching process.

In order to explicitly model only occlusions it is assumed that segmentation can be done correctly and that properties of the regions can be altered only by occlusions. In order to systematically solve the problem, occlusions are separated into categories depending on the effect they had on the properties of the regions. Each occlusion type is modeled in terms of a number of constraints. A constraint satisfaction process is introduced in order to identify the existence and kind of occlusion. The matching process allows establishing one-to-one, one-to-none, one-to-multiple, as well as, multiple-to-multiple region correspondences.

A software system has been developed in order to test the correctness of the presented theory. The system allows the user to create a virtual scene with severe occlusions. Stereo images of the scene are then generated and passed onto the stereo algorithm. The system is able to correctly identify all occlusion types and reliably establish correct correspondences between regions. Further, the system was able to produce explanations for the match or lack of a match between regions. The system has also been successfully tested using real stereo images.

Contents

Acknowledgements

Alan Mackworth, I would like thank you for allowing me to explore my creativity as a researcher. My thesis turned out exactly as I wanted. Thanks.

Jim Little thank you for taking the time to read my work. I appreciate all your help.

David Lowe, you are a well of motivation and positive thoughts for me.

Don Murray, Cullen Jennings, Jochen Lang, Rod Barman, Stewart Kingdon; you guys had the patience to listen to my ideas and give me meaningful comments. I still think that Quake-Wake is a good one !

Jeff Beis, Dan McReynolds, Michael Horsch, Bob Price; thank you for dropping your work every time I decided to bug you with my thoughts. I am sure it wasn't easy.

Valerie McRae, thank you for being my window to the outside world when I am all wrapped up in thoughts.

Dan Razzell, thank you for backing up the file system.

My family, thank you for your patience, support and love.

VLADIMIR TUCAKOV

The University of British Columbia

December 1997

Chapter 1

Introduction

Robustness and versatility are the two most important attributes of any good computer vision algorithm. Major problems in computer vision have been addressed by many researchers, however robust and versatile solutions to problems such as visual correspondence, object recognition, tracking, segmentation and so forth have not yet been found. This thesis will not offer a solution to the general computer vision problem, it will however propose a way of thinking that concentrates on producing computer vision systems that are more robust and versatile.

Many computer vision problems are often considered as a subset of a larger problem. For example, low-level vision deals with properties within images, not considering the scene that the image was obtained from. Mid-level vision uses results of low level vision in order to extract 2D and 3D geometric structures that are relevant to the scene. The task of high-level vision is to incorporate these geometric structures into objects and to model the processes and relationships between them. In general, the interaction between these levels of abstraction has been modeled in terms of propagation of information from lower to higher levels of abstraction. This bottom-up flow of information is illustrated in Figure 1.1.

Figure 1.1: The standard approach to computer vision

While bottom-up propagation of information is reasonably well understood, top-down propagation of information has been generally neglected. While higher level information is the ultimate goal of computer vision, it is also a mean of achieving robustness and versatility of the overall system. High level information is important because it can influence the processes at lower levels of abstraction. Information from higher levels of abstraction can be propagated by selecting the algorithms that should process information at lower levels of abstraction, and by parameters that should be given to these algorithms. Top-down propagation of information is illustrated in Figure 1.2.

Ideally, a vision algorithm should work successfully in all domains. Practice however, shows

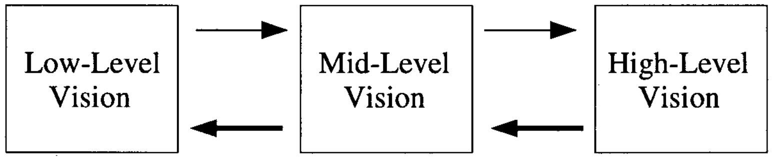


Figure 1.2: Approach taken in this thesis

that vision algorithms do not work well under all circumstances. One approach to solving this problem is to produce algorithms that gracefully degrade, as the underlying assumptions are violated. Another approach is to select precisely the algorithm and parameters that will work well in a constrained environment. Finally, an alternative is to use an algorithm that is capable of determining when its results are incorrect and more importantly report on the reason for failing in producing the correct result. The word failure will be used to refer to the reason for failing to produce correct results by a vision algorithm

Modeling failures is the main thrust of this thesis. Information about the failures enriches the information propagated in the bottom-up direction. Furthermore, reason for failure is instrumental in selecting algorithms and parameters in the top-to-bottom approach. By identifying failures within a computer vision system, and by being able to compensate for the effects caused by the failure, the overall system becomes more robust and versatile.

1.1 Failures in Stereo Vision

This thesis considers the problem of correspondence matching in region-based stereo vision under severe occlusions. The purpose of stereo vision is to perform distance measurements based on correspondence between images obtained from slightly offset cameras. Region-based stereo vision performs matching in terms of regions in each image. Regions are generated using a segmentation algorithms and matching is done by comparing the properties of the regions. The goal of this thesis is to analyze and explicitly model the effects of occlusions on the region-based stereo matching process.

Standard region-based stereo algorithms establish correspondence between regions by selecting the pair of regions that are most likely to belong to the same part of the scene. The matching is done by defining a function that calculates the difference between regions. The pair with the least difference is proposed as the correct match.

The problem with standard region-based stereo approaches is that properties of a region can be greatly altered due to occlusions. Consider the example in Figure 1.3. The properties, such as area, width and shape, of the regions corresponding to the triangle in the scene are significantly different due to the occlusion.

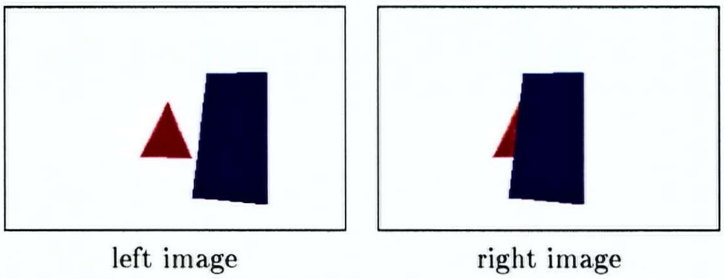


Figure 1.3: Properties of matching regions significantly altered by an occlusion

Occlusions may also cause a region in one image to correspond to more than one region in the other image. Figure 1.4 shows an example where a surface is fully seen in the right view, yet it is seen as two regions in the left view.

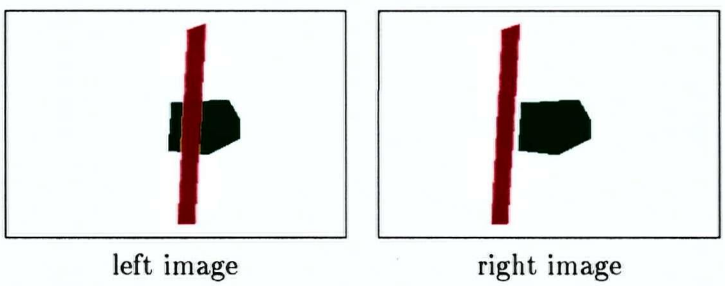
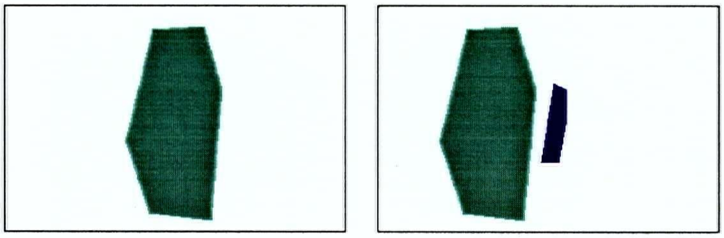


Figure 1.4: A region in one image corresponds to two regions in the other image

Finally, occlusions may cause a region in one image to have no corresponding region in the other image at all. Figure 1.5 shows an example where a surface is seen in the right view, yet it is completely unobservable in the left view. While a correspondence can not be established, it is possible to determine a distance range which the fully occluded surface must be in.

In order to explicitly model only occlusions it was assumed that segmentation can be done



left image right image

Figure 1.5: Situation in which correspondence is not possible due to an occlusion

correctly and that properties of the regions can be altered only by occlusions. In order to systematically solve the problem, occlusions were separated into categories depending on the effect they had on the properties of the regions. Each occlusion type was modeled in terms of a number of constraints. A constraint satisfaction process was introduced in order to identify the existence and kind of occlusion.

1.2 Outline of the Thesis

Chapter 2 presents the stereo vision problem and discusses the approaches taken to solving it. Chapter 3 outlines the region-based approach in more detail and formulates it in term of a constraint satisfaction process. Chapter 4 is the technical core of this thesis; it enumerates kinds of occlusion failures, and presents constraints that need to be satisfied in order to recognize that an occlusion is occurring. Chapter 5 discusses how scenes with multiple occlusions can be solved. Chapter 6 presents the experimental results. Chapter 7 offers directions for future work and a brief conclusion.

Chapter 2

The Stereo Vision Problem

The purpose of stereo vision is to perform range sensing of the environment from two or more slightly offset points of view[1]. The range sensing is done by identifying corresponding points between images and using triangulation to derive the distance. Stereo vision has been researched for decades and there have been many attempts at solving the problem. In this section I will outline these methods and discuss the advantages and disadvantages of these approaches.

The basic idea behind most stereo vision algorithms is to identify a primitive in one image and compare it to a number of primitives in the other image. The primitives that can be matched are governed by the epipolar and range constraints discussed in Appendix A. The kind of primitives used for matching define the kind of the general stereo method. For example, *correlation or area-based* methods use a small patch of the image, such a small rectangle. *Feature-based algorithms* use results of a lower level image processing algorithm, such as edges, line segments or corners, for matching between images. *Region-based* algorithms use results of region segmentation algorithms to establish the stereo correspondence.

The matching between primitives is established based on the amount of difference or similarity between the properties of the primitives. Ideally, the primitives in both images would be identical and matching would require that the properties are unchanged in both views. The nature of the stereo problem is such that the primitives may look radically different from the two stereo points of view. The difference is caused by three pitfalls:

- **False primitive extraction:** The primitives chosen by the method does not correspond to the same portion of the scene. For example, in area based correlation, rectangles corresponding may not necessarily correspond to the same portion of the scene.

- **Scene structure:** Occlusions, foreshortening, specularities, repetitive features and lack of texture can affect the performance of stereo algorithms.

- **Camera distortion and noise:** The cameras used for acquiring the images may alter the appearance of the primitives. Perspective, radial and tangential distortions are examples of

[1] It could be argued that purpose of stereo vision is in foreground/background segmentation or surface and boundary detection. The underlaying functionality, however, is range sensing, and the task is application specific.

the camera effects. Cameras also may have random noise in acquiring the images, as well as colour bleeding in case of colour cameras.

While all these problems exist, the general approach to solving the stereo vision correspondence problem has been to define a single function that determines the most likely match candidate. In general the likelihood of the match candidate has been expressed by a single number. All match scores were then compared and the most likely candidate was selected as the correct match.

The problem with this approach was that it was not always possible to determine the correct match. Therefore, validation techniques were developed to eliminate the false matches. Validation caused the depth information to be sparse. The solution to the sparseness was found in interpolation between the valid measurements in order to assign values to the invalid pixels.

While other methods such as Binocular difrequency [16], phase-based energy matching [24],[33], energy-based regularization [62], [6] exist, the *Match, Validate, Interpolate* process is characteristic of most stereo vision methods.

2.1 Correlation Methods

The correlation methods [6, 28, 27, 29, 36, 44, 57, 66] were the first and most robust solutions to the stereo vision problem. The matching primitive is a rectangular patch of the image with a center at the pixel that a match is determined for. A patch from one image is compared to patches in the other image such that epipolar and range constraints are satisfied. The match is established by finding the patch with the best correlation score. The correlation score can be determined in a number of ways, such as SAD-Sum of Absolute Differences, SSD - Sum of Squared Differences, Mean-Squared Differences and Ordinal Measures [9].

None of these similarity measurement functions can guarantee that the highest score is the correct match. Therefore, validation methods are required to determine the correctness of the matches.

Correlation methods can be validated using a number of empirical observations. Nishihara and Poggio [46] have shown that the probability of a mismatch goes down as the size of correlation window and the amount of surface texture increase. Therefore, one way of validating the match results is by setting a threshold on the correlation score. When the score is above the threshold the result is considered to be valid, otherwise it is considered to be false. Anandan [2] considers the score in the neighborhood of the potential match. If the score is peaked enough then the match is considered valid.

This kind of threshold validation method has been considered for all types of stereo matching. The problem with these methods is that they do not always identify the invalid pixel matches. They are developed on empirical observations that allow the validation to put most of the match failures under one umbrella.

More sophisticated methods of validation involve cross correlation. The idea is to perform correlation twice by reversing the roles of the two images and consider as valid only those matches for which we measure the same disparity. This method has been used by Chang et al. [15], Fleck [23] and Fua [25]. The approach allows more reliable identification of false matches since the likelihood of establishing the same false match in both directions of search is small. While this method has proven to be effective it does not provide the reason for the match failure. While the likelihood of establishing a false valid match is small, it still exists.

Even if the validation is done correctly the disparity images become too sparse and need to be interpolated. The task of interpolation is to assign values to the invalid pixels based on the available information and assumptions about the structure of the scene. Interpolation approaches by Terzopoulos [61] and Szeliski [60] used only the depth information from the matching stage. These methods are able to introduce depth discontinuities and discard incorrect matches based on the smoothness assumption. The interpolation method by Fua [25] incorporates edge information to obtain more precise discontinuities in the depth results.

Correlation methods are currently the most robust solution to the stereo vision problem. They perform well in a wide range of scenes and lighting conditions, which makes them well suited for mobile robotics navigation [64]. Correlation methods are simple and they run efficiently both on specialized parallel hardware[29], [14], as well as single CPU standard processors. Efficiency of correlation methods can further improved if motion of the cameras is known [63].

Correlation methods, however, are not perfect. They fail under occlusions, in textureless areas, under large perspective foreshortening and on specular surfaces. The most serious problem with correlation stereo is that most algorithms use a fixed size mask on the whole image. A fixed size mask can not in all cases represent the same part of the scene. For example, if a mask overlaps a depth discontinuity with an occlusion. The actual contents of the mask in one image will be different than the contents of the mask in the other image, since different portions of the occluded surface will be seen from the two different points of view. The typical solution to the problem of selecting the mask size is to perform matching with several mask sizes and select the best match, proposed by Kanade and Okutom [34] and also by Little [38]. Similarly, Boykov et al. [12] proposed a method that is equivalent to changing the shape of the correlation mask. The problem of perspective foreshortening still exists, where to represent the same part of the scene the mask sizes and shapes would have to be different in each image.

2.2 Feature-Based Stereo Algorithms

Feature-based algorithms use results of a lower level image processing algorithm, such as edges, line segments or corners, as primitives for matching between images. Unlike correlation methods, with feature-based algorithms it is possible to extract primitives that correspond to the same part of the scene since the shape and size of the primitive are data driven. The properties of the feature-based primitives, however, can be radically different. The difference in the properties is caused both by the inaccuracy in extracting the same part of the scene as well as the scene phenomena such as occlusions and perspective distortion.

The similarity measurement in feature-based stereo approaches depend on the kind of feature used in matching. Feature matching algorithms by Marr and Poggio [40] and Grimson [30] used edges as primitives. Edge pixels were compared based on the direction and intensity of the edge gradient. This approach suffered from the impoverished information that is captured in a single edge pixel.

More sophisticated methods consider line segments as primitives [58],[42], [3]. Line segments are richer in properties since the length and orientation can be considered in matching. The drawback of using line segments is that it is assumed that the environment predominantly consists of straight edges. Line segment matching did not allow for modeling of failures caused by occlusions, misbehaviors of the edge detection or line segment formation.

Other feature-based stereo algorithms include matching based on corner features [45],[43],[7]. While corners can easily represent the same part of the scene and their properties are quite stable to image noise, the amount of information that can be represented by a corner is limited. Therefore, a comparison based on corner properties will result in many false positive matches. In the case of corners, the issue is generally to determine why so many corners are similar, rather than determining the reason why the features are so different.

Feature-based stereo algorithms can also take advantage of validation methods such as thresholding or cross correlation. Other more sophisticated methods have been suggested that use rigidity checking [41].

Dynamic programming approaches [4],[48] have been proposed to allow global evaluation of the match solution. The problem of matching was cast as a problem of minimizing a cost function. These methods relied on ordering and uniqueness constraints.

Feature-based algorithms produce disparity images that are even more sparse than ones obtained by correlation methods. Delaunay triangulation [10],[49] is a common method used for establishing surfaces from a highly sparse data. This method can be used under the smoothness assumption in order to reduce the amount of information required to represent a surface.

Feature-based algorithms are useful in environments where particular features are dominant, such as straight lines in an office scene. Feature-based algorithms have the potential of being faster than correlation based algorithms because fewer primitives need to be compared. Corner-based algorithms work more reliably because of the stability of the corner features with respect to the viewing position. On the other hand, feature-based algorithms produce sparse disparity maps relying heavily on interpolation. Another problem with feature-based algorithms is that the features between images are more likely to appear similar than to appear greatly different. This particularly applies to corner matching since they are not rich with information.

2.3 Region-Based Stereo Algorithms

Region-based stereo methods are not common in stereo vision research. The approach is to segment the stereo images and establish correspondence by comparing the properties of the obtained regions. Regions produced by segmentation algorithms can be very rich with information. The size, shape, the average intensity of the region can be considered in the matching process. Further, other image processing algorithms can be applied on the regions in order to establish additional information. The advantage of considering regions is that primitives are more likely to be different. Therefore, the task is to determine the reason for the difference in the properties of the primitives, rather than disambiguating between many primitives that are similar.

Boyer and Kak [11] proposed solving the stereo correspondence problem through structural description of the images. The idea was to use a low level vision algorithm to extract descriptions of the images and establish correspondence in terms of them. Descriptions of an image consisted of a number of primitives. Each primitive was characterized by a number of attributes and by the interrelationship of the primitives. The task of the algorithm was to establish correspondence between extracted primitives. The matching was done by finding a pair of primitives that had the closest attributes as well as preserved the interrelationship.

An information-theoretic inter-primitive distance measure was defined to measure the similarity of primitives and to account for the statistical behavior of the image-to-image distortion process. A measure of relational inconsistency was developed to allow graceful degradation of the matching process.

Structural stereopsis is a computationally less intensive due to a smaller search space. The size of the search space is smaller due to the smaller number of primitives that need to be matched.

Structural stereopsis is rich with attributes that can be used in an intelligent manner during the matching process. On the other hand area and feature-based approaches to stereo are relatively poor with information. For example Grimson's [30] stereo algorithm considers only the polarity of a zero crossing in the output from an operator applied to the image.

Following Boyer's paper Lee and Lei [35] developed a method for region matching and depth finding in stereo aerial photography The approach was to segment the image into a number of regions. Then the correspondence between regions is established using local features of the region using similarity moments. The matches are then refined using global matching by angle consistency. The depth is recovered by establishing correspondence between corners of the matched regions.

While feature-based algorithms generally find the boundaries of surfaces, the region-based algorithms extract the surfaces. Region-based stereo algorithms are similar to the feature-based algorithms because they are also capable of representing the same part of the scene in terms of primitives. While the same parts of the scene can be represented with the primitives, the properties of the primitives are generally different. Again, the differences can be caused by the properties of the algorithm or by the properties of the scene.

While research in full 3D reconstruction of the scene using region-based stereo is not voluminous, region-based stereo algorithms have been used as preprocessing step to correlation or feature-based algorithms [53]. Region-based stereo algorithms have also been used in cooperative segmentation of images [54]. Region-based stereo algorithms have been used in specialized applications such as robotic manipulation [11] or aerial imaging [35] and have not been used in general scene reconstruction.

2.4 Opportunity for Improving Previous Work

In this thesis I consider that the purpose of stereo vision is to produce accurate and dense depth maps. In most previous work the accuracy of the depth maps has often been traded off for the density of the depth maps. In this thesis I am proposing a method that will increase both the density and the accuracy of the depth maps.

In order to produce dense depth maps it is necessary to establish correspondence for as many pixels as possible. Some of the above described techniques, such as feature and line segment-based algorithms, do not produce dense depth maps and therefore heavily rely on interpolation and triangulation post-processing. The reason for producing sparse depth maps is that matching is done on surface boundaries and corners, rather than surfaces themselves.

Area-based algorithms, on the other hand, can produce more dense depth maps, because matching areas allows for comparing properties of surfaces rather than just the boundaries. In cases where there is no surface texture, for example, area-based algorithms also establishing depth measurements only at surface boundaries.

Region-based algorithms are surface oriented methods and are therefore are most likely to produce dense depth maps. Region-based algorithms are similar to area-based algorithms in that the primitives matched are both regions of an image. The difference is in that the regions in area-based algorithms are of fixed size and shape, yet in region-based algorithms the shape and size of regions are data driven. If we assume that regions of an image can be extracted such that they correspond to surfaces in the scene, then the region-based approach becomes very powerful. The power of the region-based approach allows reasoning about why two regions in different views of the scene may or may not belong to the same surface. In other words, regions-based matching allows for explaining the cause for the difference in the appearance of the surface in two images.

Producing the explanations for the difference in the appearance between views has not been a major focus of the previous work in stereo vision. Explanation of the reason for the difference in the appearance is an instrumental part of producing a robust and versatile system. Therefore I choose to use region-based stereo vision because I believe that it has the greatest potential of producing accurate and dense depth maps.

Chapter 3

Region-Based Stereo Vision

In this thesis we adopt the idea of region-based stereo matching. The purpose of the approach is to segment each stereo image into a number of regions such that the pixels within the region belong to the same surface. The regions are then analyzed for a number of distinctive features in terms of size, shape, spectral and textural content and relative position to other regions in the image. Regions from one image are compared to regions in the other image. Matching is done by using constraints imposed by the stereo problem and the fact that the images are obtained simultaneously.

By the process of segmentation we are abstracting an image from a raster of pixels to a number of regions that have richer characteristics. By considering regions, as opposed to pixels, we are reducing the number of image primitives used in matching. At the same time we are increasing the dimensionality of the search space in which the matching is done. Increasing the dimensionality of the search space increases the reliability of the produced matches. Further, decreasing the number of match candidates reduces the amount of computation needed.

Region-based stereo is not a new method, and certainly does not perform any more robustly than other methods. However, it has the advantage that it allows easier modeling of the failure in matching due to the richness of information encapsulated in the regions. This thesis will discuss in detail occlusion failures that occur in region-based stereo and show how the failures can be resolved and appropriate actions can be taken to produce a robust result.

3.1 Region Segmentation

Segmentation does not have a universally correct solution and therefore may be considered unsuitable for generating primitives for stereo matching. At this point I will make an important assumption about segmentation:

The segmentation for the stereo correspondence problem must be such that it identifies the same parts of the scene in the images taken from slightly offset cameras, not withstanding effects of the scene and camera geometry, such as occlusions and perspective.

In other words, the segmentation algorithm will produce regions that correspond to the same part of the scene in both stereo images. While this assumption may seem too strong, the matching problem is far from trivial. The future work section suggests direction in overcoming problems of segmentation as yet another kind of failure.

3.2 Region Matching as Constraint Satisfaction

The basic approach in region-based matching is comparing the characteristics of the regions under a number of constraints inherent in stereo vision. Generally, matches are established by identifying pairs of regions whose characteristics are the most similar and do not violate constraints inherent in stereo vision. In this section I formally define stereo vision constraints and the similarity measurement between regions.

The following notation will be used to represent a region:

$$R = \begin{bmatrix} R_{area} \\ R_{RGB} \\ R_x \\ R_y \\ R_{moments} \end{bmatrix}$$

where R_{area} is the number of pixels in the region, $R_{RGB} = (R_{ave}, G_{ave}, B_{ave})$ are the average RGB values, $R_x = (x_{min}, x_{max})$ are the minimum and maximum x values, and similarly $R_y = (y_{min}, y_{max})$ are the minimum and maximum y values, $R_{moments} = (moment_1, moment_2)$ are the first and second moments of the region.

The segmentation therefore produces a set of regions in each image:

$$R^l = \{R_i,\, i = 1..n^l\}, R^r = \{R_i,\, i = 1..n^r\}$$

where R^l and R^r are all regions in left and right respectively, and n^l and n^r are the number of regions in each image.

The purpose of the matching process is to determine which regions in one image could correspond to the regions in the other image given two kinds of constraints, the stereo constraints and size/shape constraints. The results of the matching process are stored in a binary matrix M:

$$M_{i,j} = C_{stereo}(R_i^l, R_j^r) \wedge C_{size/shape}(R_i^l, R_j^r), 0 < i < n^l, 0 < j < n^r$$

where C_{stereo} and $C_{size/shape}$ are the stereo constraints and size, shape constraints respectively, given a pair of regions from the left and right image.

3.2.1 Stereo Constraints

The constraints imposed by the definition of the stereo problem are the epipolar and disparity range constraints[1]. These constraints define the location of the region in the image where matches are possible. The matches between regions that violate these constraints are eliminated from M. The stereo constraint is defined with the following equation:

$$C_{stereo}(R^l, R^r) = C_{epipolar}(R^l, R^r) \wedge C_{disprange}(R^l, R^r)$$

where $C_{epipolar}$ and $C_{disprange}$ are epipolar and disparity range constraints respectively, and R^l is a region in the left and R^r a region in the right image.

For simplicity we will assume that the optical axis of the cameras are parallel and that the epipolar lines are aligned and horizontal. The epipolar constraint can therefore be written as following:

$$C_{epipolar} = R^l_{y_{max}} > R^r_{y_{min}} \wedge R^l_{y_{min}} < R^r_{y_{max}}$$

That is, as long as there is any vertical overlap between two regions along the epipolar lines the epipolar constraint is satisfied. So, the regions with no vertical overlap are rejected.

The disparity range constraint depends on the baseline distance between the stereo cameras and the focal length of the lenses [31]. For each pixel in one image there exists a range along the epipolar line in the other image that a matching pixel can be found. In the case of regions the disparity range constraint is satisfied if there is any overlap between regions given the values of the disparity range. If for all values of the disparity range there is no overlap between regions then it is not possible that they belong to the same surface. The disparity range constraint is defined with the following equation:

$$C_{disprange}(R^l, R^r) = D_{min} < (R^l_{x_{min}} - R^r_{x_{min}}) < D_{max} \vee D_{min} < (R^l_{x_{max}} - R^r_{x_{max}}) < D_{max}$$

[1]Epipolar and range constraints are described in Appendix A.

where (D_{min}, D_{max}) is the disparity range.

Stereo constraints, as formulated above, can be viewed as hard constraint which are used to eliminate impossible matches. This formulation of the stereo constraints does not involve any parameters and its purpose is to perform elimination, rather than strict matching.

3.2.2 Size and Shape Constraints

Although stereo constraints can prune a number of possible matches they are not usually sufficient to identify the correct match. Therefore additional information about the shape and size of the region are used. We consider the following measures:

- Average RGB pixel values within the segment.
- Size of the segment in x and y directions.
- Number of pixels within the segment.
- First and second moment.

The possible matches between regions were further eliminated by imposing constraints on the maximum difference between each characteristic of the region. A vector $MaxDiff$ was introduced, such that a maximum deviation was defined for each measure of size and shape. The size and shape constraint was then defined by the following equation.

$$C_{size/shape}(R^l, R^r) = \forall \left| R^l_{domain} - R^r_{domain} \right| < MaxDiff_{domain}, domain \in Domains$$

$$Domains = \{area, RGB, x_{size}, y_{size}, moments\}$$

The values of the $MaxDiff$ vector define the maximum deviation of region characteristics in a given domain. For example, a red region in the left image cannot possibly match a green region in the right image. On the other hand a region that was L pixels long in the left image can be $L/2$ pixels long in the right image, due to occlusion.

The above formulation of constraints on size and shape, similarly to stereo constraints, allow only for elimination of matches between regions that have significantly different characteristics. Two problems remain, the problem of multiple matches and the problem of no matches. In some scenes it is possible that there will exist regions that have very similar characteristics. In general the likelihood of multiple matches decreases as the dimensionality of the feature vector increases. The next section gives a brief solution to this problem. A more serious problem occurs when the characteristics comparisons are conservative and prune out matches that should be established.

The values of the vector $MaxDiff$ have to be set empirically to fit the requirements of matching and the context of the scene. It should be noted that if a constraint is violated it does not necessarily mean that the match is not possible. For example the epipolar constraint can be violated with a vertical occlusion, yet the regions can belong to the same object.

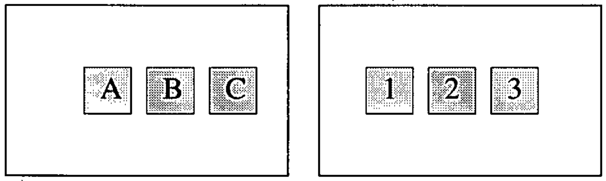
3.2.3 Multiple Match Elimination

Stereo, size and shape constraints eliminate matches that are not possible given the vector $MaxDiff$. The elimination process, however, can leave multiple matches between regions. Multiple matches can be eliminated using the method of arc consistency [39], if the matches are not symmetric between the two images. Consider that a region R_i^l in the left image matches multiple regions in the right image, say a set $X = R_x^r, M(i,x) = true$.

If there is one and only one region in the set X, say R_j^r that matches only R_i^l, then $M(i,j) = true$, and $M(i,x) = false$, $x \neq j$. This process of elimination of multiple matches is repeated until no false matches could be eliminated.

Figure 3.1 shows an example of situation where multiple matches are encountered. Consider that regions A,B and C are results of segmentation in the left image and regions 1,2 and 3 are results of segmentation in the right image. The matrix $M1$ is the result of constraint satisfaction process, where 1s mean a possible match and 0s mean impossible matches. The segments lie along the same epipolar line, they have identical shape and size as well as the same color. We consider that the disparity range constraint was able to eliminate a number of false matches, however there are still multiple matches between regions.

The arrows in the figure show one iteration of the above outlined multiple match elimination algorithm. In the first iteration the matches between region 3 and regions A and B are eliminated due to the fact that region C can match only region 3. Therefore we obtain the matrix $M2$. In the



M1

	1	2	3
A	1	1	1
B	0	1	1
C	0	0	1

M2

	1	2	3
A	1	1	0
B	0	1	0
C	0	0	1

M3

	1	2	3
A	1	0	0
B	0	1	0
C	0	0	1

Figure 3.1: Example of multiple region matches

next iteration of the algorithm we eliminate the match between region 2 and A because region B matches only the region 2. Therefore we obtain a the final matrix $M3$ that does not have multiple matches.

In some scenes it is possible to have a configuration of regions that can not be solved using arc consistency. There exit a number of ways of solving this problem, such as increasing the dimensionality of the feature vector that describes the region by applying additional processing of the image. The thrust of this thesis is, however, on the problem of matching when correspondence can not be established due to over constraining.

3.3 Explaining Constraint Violations

In the stereo matching process imposing strict constraints leads to a greatly reduced number of possible matches between images. In most previous work this problem has been handled by either not establishing a match at all or by establishing matches between regions on the nearest neighbor principle.

The approach taken in this thesis differs in that it allows establishing correspondence between regions that may be violating some of the constraints. The correspondence is however, established only if the violation of the constraints can be justified in terms of a failure supported by new evidence introduced through the previous matches.

Let us consider one of the examples outlined in the introduction. In this example constraint satisfaction can easily match the rectangular segments because none of the constraints are violated. On the other hand the triangle can not be matched because width, height, and area are significantly different between the two views of the triangle.

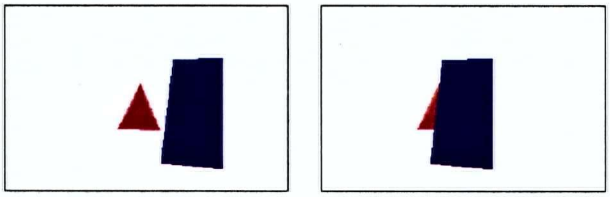


Figure 3.2: Example of an occlusion problem

The correspondence between the two views of the triangle can be established if it can be shown that the rectangle is occluding it. The rectangular regions are matched, therefore they can be placed in the 3D model. Further, the contour of the left corner of both triangular regions can be matched, placing the visible part of the triangle in 3D. Since the position of the triangle in such a 3D model is behind the rectangle we have a justification for the missing part of the triangle in the right view. If this is the only explanation, then the match between regions can be established, even though it previously violated the constraints.

Explaining the reasons for constraint violations is the main idea in this thesis. The rest of the thesis concentrates on establishing matches under occlusions by identifying and modeling all possible kinds of occlusions.

Chapter 4

Interpreting Occlusions

This chapter concentrates on match failures caused by occlusions. In order to understand occlusions we will enumerate all occlusion types that may occur in a constrained environment. It will then be shown how these occlusion types can be detected and used to establish appropriate matches. Further, it will be discussed how distance measurement can be established even if a match is not possible.

In order to analyze match failures due to occlusion, we consider an environment which disallows other kinds of failures. In other words, we assume that without occlusions matches can be done reliably and accurately. The following are the assumptions about the domain:

- The image of the scene is easy to segment into regions that correspond to different surfaces.
- The surfaces in the scene are flat, and perpendicular to the optical axes of the cameras.
- There exists a dominant background that can be easily identified.

While these assumptions may be false in the real world, we need to make them in order to understand the nature of the occlusion failures.

There has been previous work in detecting depth discontinuities and occlusions [37, 26]. These approaches used the ordering and uniqueness constraint in correlation based algorithms. Work in dynamic programming based stereo [18], and layered representation of scenes [65],[1],[19],[8] are related to the interpretation of occlusions. The approach described in this thesis is different because it defines occlusions from the aspect of region-based stereo matching. Due to the level of abstraction this work more similar to Cooper's casual scene understanding [17].

4.1 Occlusion Types

In order to systematically study occlusion we need to enumerate several types of occlusions. We will first consider a situation where there exists one fully visible surface, *the occluding surface* and another surface that is occluded, *the occluded surface.* The regions corresponding to the surfaces will be referred to as *the view* of the surface. For example, "the left view of the occluding surface" refers to the region in the left image that corresponds to the fully visible surface. We now define two kinds of views that the occluded surface may have:

- **Better view,** the view in which the occluded surface is seen better, i........e. larger.
- **Worse view** the view in which the occluded surface is more occluded or invisible.

The surface in the better view can be seen in one of the two following ways:

- **No occlusion:** The occluded surface is fully visible in the better view.
- **Partial occlusion:** The occluded surface is partially occluded in the better view. [1]

Similarly, the worse view of the occluded object can be seen in three different ways:

- **Partial occlusion:** where the surface is partially seen.
- **Full occlusion:** where the surface is fully occluded.
- **Split occlusion:** where the occluded surface is imaged in terms of two or more separate regions.

Further, the occluded surface may appear on opposite sides of the occluding surface in the two views. For example, the better view of the occluded surface may be on the left of the occluding surface, while the worse view may be on the right side of the occluding surface. In Figure 4.1 we present all possible combinations of the above outlined kinds of occlusions.

4.1.1 Comments on Occlusion Cases in Figure 4.1

In Figure 4.1 we outline the possible occlusion cases. The figure is organized such that the column represents a particular instance of the better view in the occlusion case. Similarly, the rows correspond to cases in which the worse view has the same characteristic. For example, the second row, *Worse View/Left Partial* means that the worse view of the occluded surface is seen on the left of the occluding surface. Similarly, the first column, *Better View/Left None* means that the better view of the occluded surface is seen on the left of the occluding surface.

A certain symmetry can be observed in Figure 4.1. For example, Better View/Left None, Worse View/Left Partial is conceptually equivalent to Better View/ Right None, Worse View/Right

[1] Better view can have split occlusion as well, however it will not be discussed until later sections.

Partial. The boundary of this symmetry is depicted by the thicker line separating the table in lower right and upper left section. The modeling of occlusion will be done in terms of one half of this table, and the full domain will follow from the symmetry.

There are four entries in Figure 4.1 that are left blank. These entries are impossible since the better view is smaller than the worse view. For example in the first row Left Partial is in the better view, while Left None is in the worse view. Since, None means no occlusion, then it is a better view than Partial.

Having a split occlusion in both left and right view has been omitted because it will be discussed in more detail in subsequent sections.

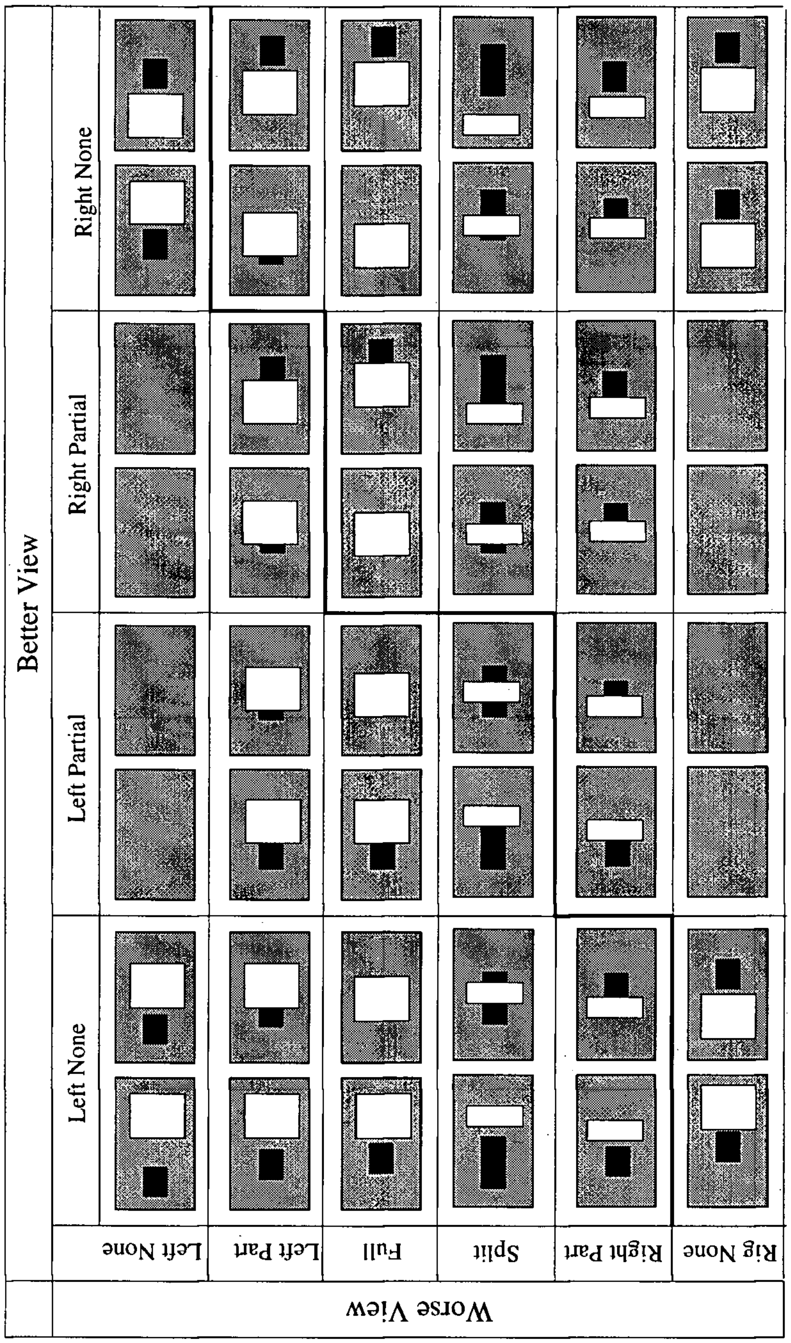


Figure 4.1: Possible cases of occlusion between two separable surfaces

4.2 Modeling Occlusions

Successful modeling of occlusions will result in establishing a plausible explanation for all failures to match using strict stereo constraints. In other words for every region in one view of the scene there must be a match and an explanation why strict stereo constraints failed. If a match cannot be established an explanation must be provided. The following is an outline of the procedure that will be followed:

1. identify the unoccluded surfaces and establish correspondence
2. identify the unmatched regions
3. identify occlusions and establish correspondence depending on the type of occlusion
4. identify the occluded parts of the surface in the better view
5. identify possible locations of the fully occluded surfaces

The most difficult task in matching under occlusion is establishing that an occlusion is taking place, and identifying the type of occlusion. In order to resolve the type of occlusion we identify the following subproblems:

- **Shared Contours:** cases in which the match between contours of both occluded views can be used to establish correspondence. All cases that have full view of the surface in the better view and are not fully occluded in the worse view fall into this category. Similarly, partial occlusions in both views that are on the same side of the occluding object are in this category.
- **Possible Shared Contours:** Partial occlusions in both images that are on opposite sides of the occluding object are in this category. The partially occluded surfaces in the better view and split occlusions in the worse view are also in this category.
- **Full occlusions:** cases in which no correspondence can be established.

4.3 Solution to Shared Contours Problem

Certain occlusion types allow the same contour of the surface to be imaged both in the better and the worse view. If this is the case, then it is possible to establish the match between the regions based on the shape of the contour. Partial and some split occlusions are in this category and the solutions will be described in detail in the following sections.

4.3.1 Weak Stereo Constraints

Occlusion can cause serious changes in the appearance of the surface. These changes inevitably cause strict stereo constraints to fail. In order to solve the occlusion problem with shared contours we need to identify the stereo constraints that are not violated. The following are *weak stereo constraints* that are not violated under shared contour occlusions:

- **Colour Constraint:** regardless of the occlusion type the imaged reflectance of the surface should be the same in both images.
- **Weak Range Constraint:** there has to be some horizontal overlap between regions along the epipolar line.
- **Weak Epipolar Constraint:** there has to be some vertical overlap between regions along the epipolar line.

4.3.2 Partial Occlusions

Due to partial occlusion other standard stereo constraints can be easily violated. For example, in Figure 4.2 the triangle is occluded by the rectangle. The colour of the region that corresponds to the triangle is the same in both views. The weak epipolar and range constraints are also not violated. On the other hand area, width and height constraints are clearly violated and a simple match cannot be established.

In order to establish correspondence between the two views of the triangle we need to show that there is a partial occlusion. To do this we need to show the following:

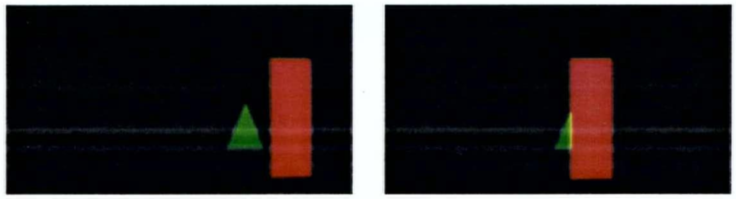


Figure 4.2: Partial occlusion

- **Partial Contour Match:** There exists a partial contour match between the two views of the triangle. In this case the match between the left corner in each view can be established.
- **Occluder Exists:** We need to show that there exists one or more surfaces that are positioned in the scene such that the surface is occluded.
- **Only Explanation:** The partial occlusion is the only explanation for the violated stereo constraints.

Example of Partial Occlusion

Consider an example of partial occlusion presented in Figure 4.3. In this example the vertical rectangle is occluding the horizontal rectangle. In order to establish partial occlusion we first perform the straight forward matching using the standard stereo constraints. The views of the vertical rectangle match without violating any constraints. On the other hand, the views of the horizontal rectangle do not match due to the change in the width and consequently the change in area and 1st and 2nd moments. Therefore the match between the views of the horizontal rectangle cannot be established without an explanation.

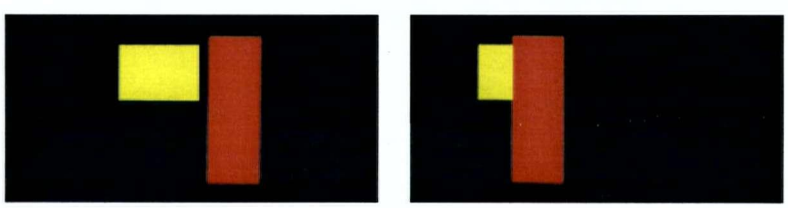


Figure 4.3: Resolving a Partial Occlusion

In order to show that the explanation may be partial occlusion we will first establish partial contour matches. Figure 4.4 shows the two possible partial contour matches. The contour matching procedure is explained in Appendix B.

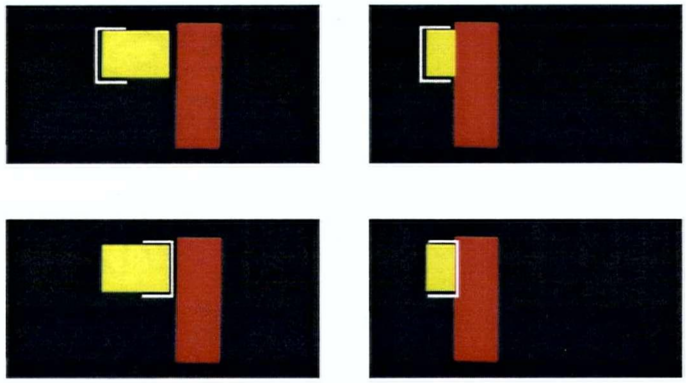


Figure 4.4: Possible partial contour matches

The second step is to establish which surface is causing the occlusion. We can establish the better view of the occluded surface by comparing the size of the regions in two views. It can be observed that the left view is the better view since the area of the region is larger. The next step is to overlap the region in the better view better on top of the worse view based on the partial contour matches. The overlapping is shown in Figure 4.5.

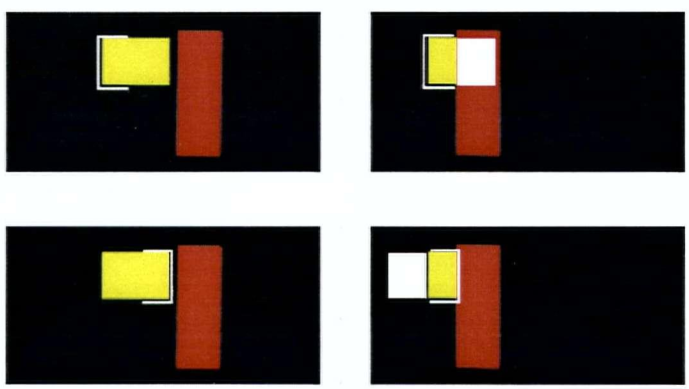


Figure 4.5: Overlapping the better view based on the contour match

The white areas in Figure 4.5 represent the missing parts of surface. In the case of right contour match we find that the missing part overlaps the background. On the other hand the left contour match makes the missing overlap the taller rectangle. Contour matches establish a potential displacement between the views of the occluded surface. By comparing the right contour displacement with the displacements of the background we find that the background cannot be the occluder. On the other hand, the displacement of the left contour is less than the displacement of the taller rectangle. Therefore the tall rectangle can be the occluder. Since there are no other possible explanations we conclude that the wider rectangle is partially occluded by the taller rectangle.

4.4 Split Occlusions

Split occlusions occur when a surface is occluded in such a way that it is imaged in terms of two separate regions in the worse view. Figure 4.6 presents an instance of split occlusion. The most important property of the split occlusions is that one region in the better view can be matched to two or more regions in the worse view.

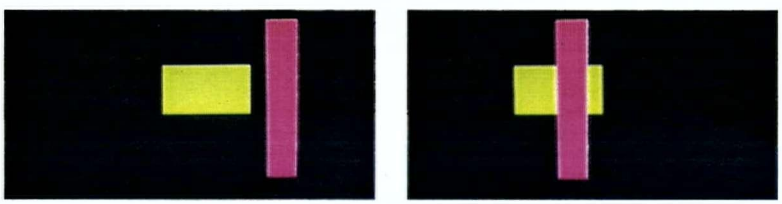


Figure 4.6: Example of a split occlusion

In order to establish matches between regions based on split occlusions we need to show that the following constraints are satisfied:

- **Weak stereo constraints** are not violated
- **Establish partial contour matches** between the unoccluded view of the surface and the regions in the other image that may be split due to an occlusion.
- **Ensure that the distances are preserved** between the positions of the positions of the partial contour matches in each image.
- **Identify the occluding surface** which is in front of the occluded surface and produces the split regions.

- **Only explanation;** the split occlusion is the only explanation for the violated stereo constraints.

4.4.1 Example of a Split Occlusion

Consider the scene in Figure 4.6. In this scene the brighter rectangle is occluded by the darker rectangle. The occlusion is such that in the worse view the occluded rectangle is imaged in terms of two disjoint regions. The correct interpretation of the scene requires that two regions in the worse view both match only one region in the better view.

To establish a split occlusion we first establish correspondence between regions that do not violate any of the stereo constraints. In this case the darker rectangle is easily matched. At this point we have one region in the left image and two regions in the right image that are still unmatched. It can be shown that neither of the regions can be matched as partially occluded surfaces, as in section 4.3.2. At this point we need to establish partial contour matches between regions. Figures 4.7 and 4.8 shows the possible matches between the two regions in the right image and region in the left image. Figure 4.7 shows two contour matches that are not suitable for explaining split occlusions. The reason for the lack of explanation is that the displacement implied by these contour matches show that the surfaces are at different distances from the cameras. On the other hand, in Figure 4.8 we find that the contours imply that the surfaces are at the same displacements. This implies that it is possible that two regions in the right image and the region in the left image belong to the same surface.

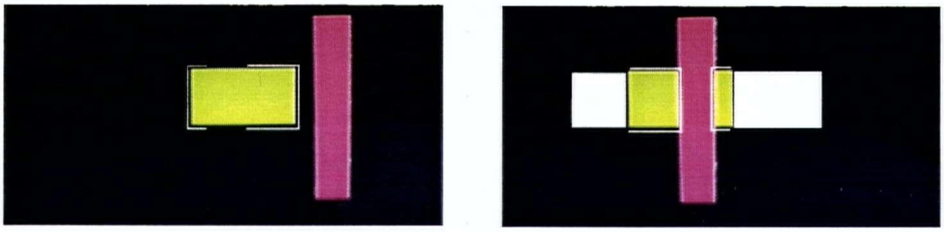


Figure 4.7: Contour matches NOT suitable for explaining split occlusions

Given that it is possible to have correspondence between the regions in the right image and the region in left image we need to identify the missing part of the surface. This is done by

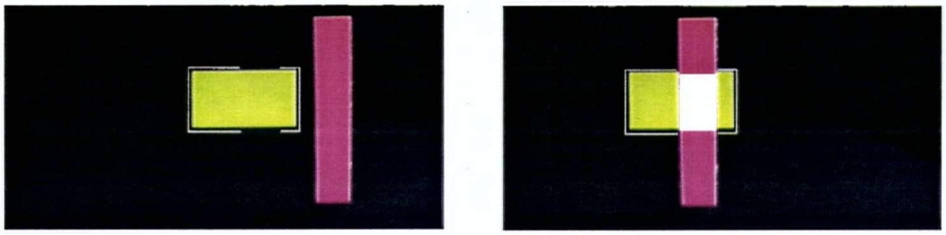


Figure 4.8: Contour matches suitable for explaining split occlusions

overlapping the regions from the split view on top of the region in the better view. The parts of the region in the better view that are not overlapped are then projected to the corresponding position in the worse view. Figure 4.8 shows this in terms of a white rectangle, which corresponds to the missing part of the occluded surface.

Now that we have identified the missing part of the surface we can inspect the region corresponding to it in the worse view. In Figure 4.8 we find that the missing part of the surface corresponds to a part of the darker rectangle. The darker rectangle was matched earlier and its disparity is known. Similarly, the contour matches of the occluded surface yield a disparity measurement. Now it is possible to compare the disparities and show that the disparity of the occluding rectangle is greater than the disparity of the occluded rectangle. Therefore it is possible to show that the darker rectangle produces a split occlusion of the lighter rectangle.

4.5 Solution to Full Occlusion

In the case of full occlusion there does not exist a correspondence. Therefore the stereo constraints are not applicable in the standard matching sense. In order to show that there exists a full occlusion we need to show that the observable part of the surface can be hidden behind another surface. Therefore we need to show the following:

- **Identify possible occluders:** the possible occluders are regions that overlap the occluded region as it is slid along the epipolar line.
- **Occluder in front:** in order to show that the surface is fully occluded we need to show that the occluding candidates are in front of the occluded surface. This means that the disparity

of the occluding surface must be greater that the disparity of the occluded surface, given the hypothetical position of the missing region.

- **Identify better view:** it is important to establish if the whole surface is seen in the better view. If it fully seen, then it is possible to establish the size of the surface needed to fully occlude the surface. If the occluded surface is not fully observed it may be impossible to determine the size of the occluding surface.

4.5.1 An Example of Full Occlusion

In Figure 4.9 we present an example of full occlusion. In the left view we have three regions that correspond to surfaces in the scene. In the right view only the rectangular surfaces can be seen. The triangular surface is hidden behind one of the rectangular surfaces. Since the rectangles are fully observed it is possible to establish correspondence between the views of the rectangles. By using the scales on top of the images in Figure 4.9 we measure a disparity of 1 for the higher rectangle, and disparity of 2 for the wider rectangle.

The triangle is now slid along the epipolar line and positions at which there is full overlap between the rectangles and the triangle are recorded. The dotted triangles represent some of the possible positions triangle.

The disparity of the triangle when occluded by the wider rectangle ranges from 0 to 0.5. This can be observed by measuring the position of the tip of the triangle against the scales in both images. On the other hand, the disparity when the triangle is hidden behind the higher rectangle is 2.5. The conclusion that can be made is that the triangle can be hiding behind the wider rectangle since the disparity of the triangle is smaller than the disparity of the wider rectangle. On the other hand the triangle cannot be hiding behind the higher rectangle because the disparity of the triangle is greater than the disparity of the higher rectangle. In other words, in order for these disparities to be correct, the triangle would have to be in front of the rectangle, in which case it would be imaged.

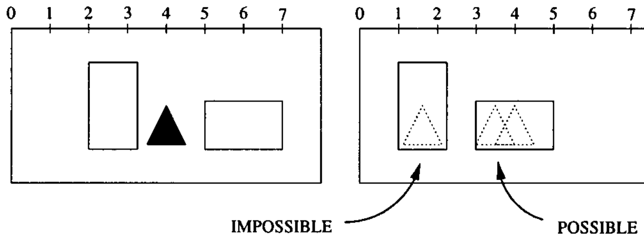


Figure 4.9: An example of full occlusion

4.6 Occluded Better View

The task of forming an explanation of an occlusion becomes more difficult if the better view of the occluded surface is also occluded. There are four typical situations of this problem which are shown in Figure 4.10. In all four cases shown the better view of the surface is in the left view of the stereo scene. The better view of the occluded surface may not be imaging the whole surface. In other words the better view is possibly also occluded. (We say possibly, because it may be the case that the boundary between the occluder and the occluded surface is also the true boundary of the occluded surface).

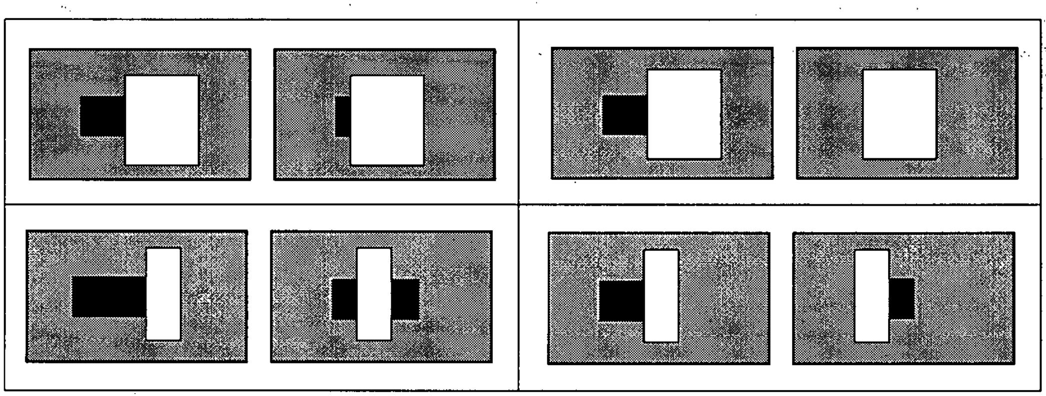


Figure 4.10: Possible situations when better view of the surface is occluded

4.6.1 Width Constraints

Let us consider the specific example where the better view is partially occluded and the worse view is fully occluded, see Figure 4.11. In this situation it is not possible to determine the width of the occluded surface. It is possible that the occluded surface is no larger than what can be seen from the better view. On the other hand, is possible that the surface is larger since the rest of the it can not be observed.

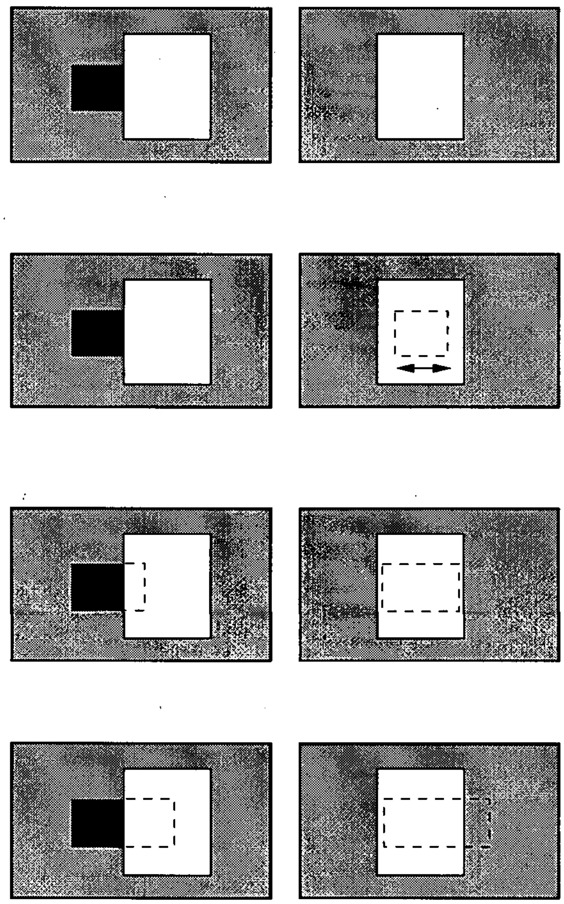


Figure 4.11: Width constraints on occluded surfaces

While the size of the occluded part of the surface can not be determined, there exists a range of sizes that it must fall within. Figure 4.11 shows constraints on the size of the occluded surface. If it is assumed that the occluded surface is fully observed in the better view, it is possible to determine the disparity range such that the occluding surface is fully hiding it.

On the other hand, if we assume that the occluded surface is not fully observed in the better view then the maximum width of the surface is constrained by the width of the occluding surface.

Figure 4.11 shows a width of the occluded surface that is not possible. If the occluded surface is wider than the occluding surface, then the occluded surface would not be fully occluded.

The hypothesized size of the occluded surface influences the number of displacements it can possibly be at. In this situation higher level knowledge can be used to determine stricter boundaries on the size of the region, and consequently influencing the distance measurement.

4.6.2 Shape Constraints

When a surface is occluded in both stereo views it is not possible to establish the exact shape of the surface. It is, however, possible to establish constraints on the shape of the surface. An example of constraints on shape of an occluded surface are presented in Figure 4.12. Again, the shape of the occluding surface, the displacement and the shape of the visible part of the surface are variables of the constraint on shape of the occluded part of the surface.

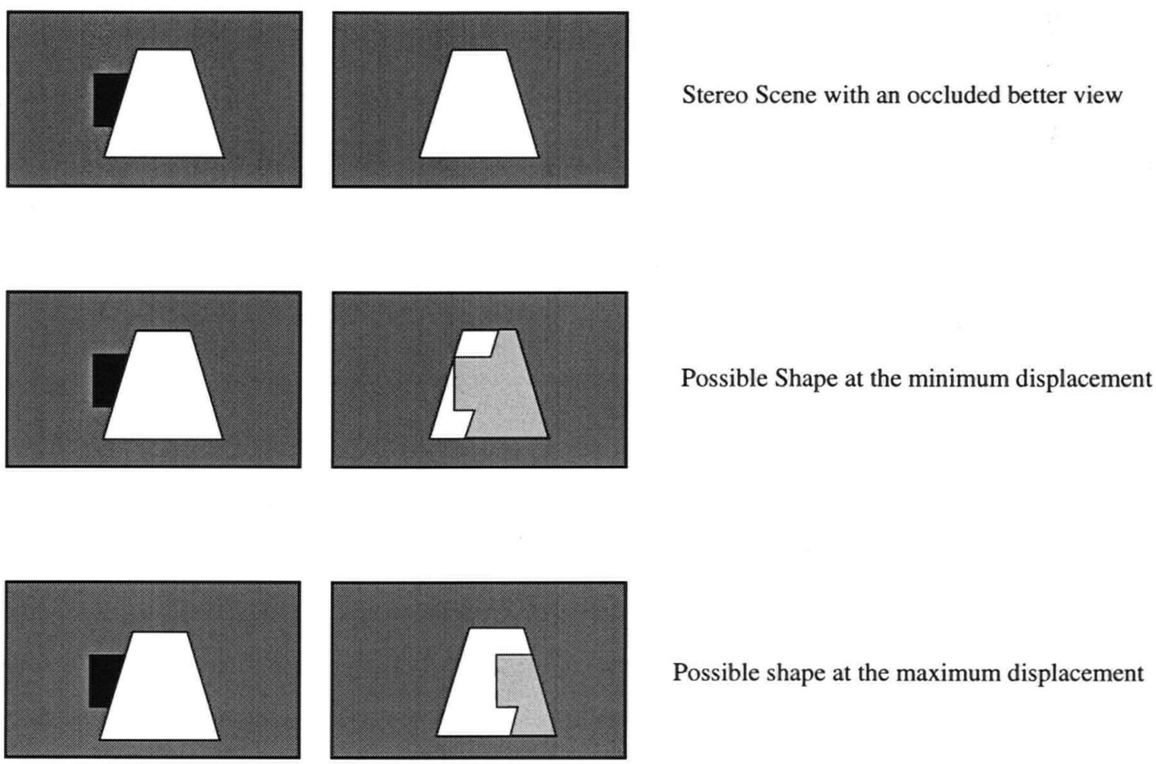


Figure 4.12: Shape constraints on occluded surfaces

4.6.3 Occlusions on Opposite Sides

In some occlusion cases it is not possible to establish reliable partial contour matches. Consider the example in the Figure 4.13. In this example it is possible to establish a partial contour match that has a plausible explanation which may not be correct.

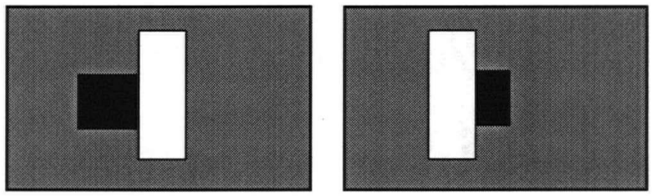


Figure 4.13: Partial Occlusion in both views

Figure 4.14 shows two interpretations of the scene. The top example shows the interpretation of the scene as a partial occlusion, where a contour match is established. While this is a correct interpretation it is not the only one. The bottom example shows that it is possible that the contour match is incorrect due to partial occlusion in the left image. In this case a unique displacement can not be established, however constraints on ranges and shapes can be derived as discussed in sections 4.6.1 and 4.6.2.

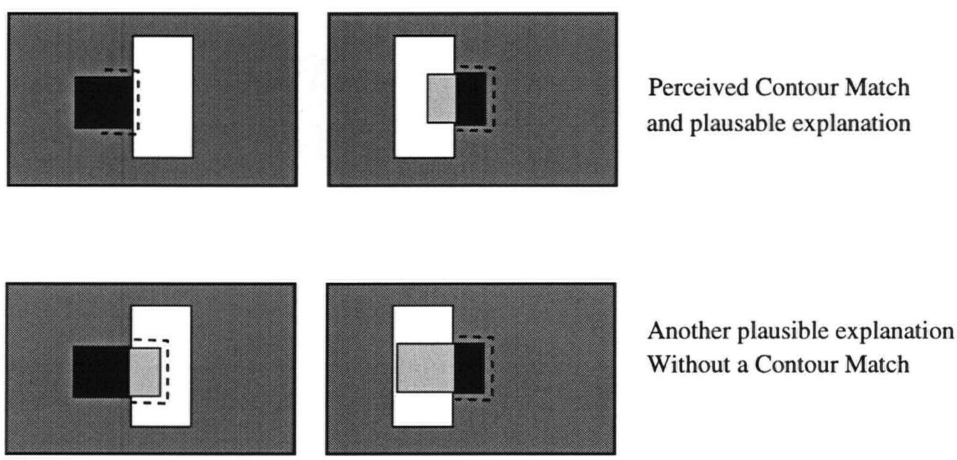


Figure 4.14: Two possible interpretations

4.6.4 Partial and Split occlusions

An instance in which the better view is partially occluded and worse view is split occluded is another example where partial contour matches may be unreliable. Consider the example in Figure 4.15. In this example there exists only one contour match that does not include an occlusion boundary. Section 4.4 shows that one contour match is not sufficient to describe a split occlusion.

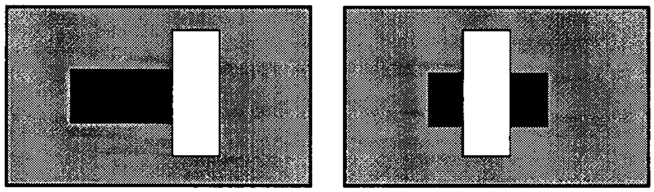


Figure 4.15: Split-Partial Occlusion

Figure 4.16 describes the process involved in disambiguating the scene. First, the better view is overlapped on top of the worse view based on the contour match. The better view overlaps the occluding surface and a part of the right region of the worse view. We can check that the occluding surface is in front of the occluded surface by comparing the disparities. The right region of the split occlusion is now hypothesized to be a part of the occluded surface. The part of this region is appended to the region corresponding to the better view. The new region is overlapped on top of the better view. We now establish that the missing part is occluded and we can conclude that we have a case of split occlusion.

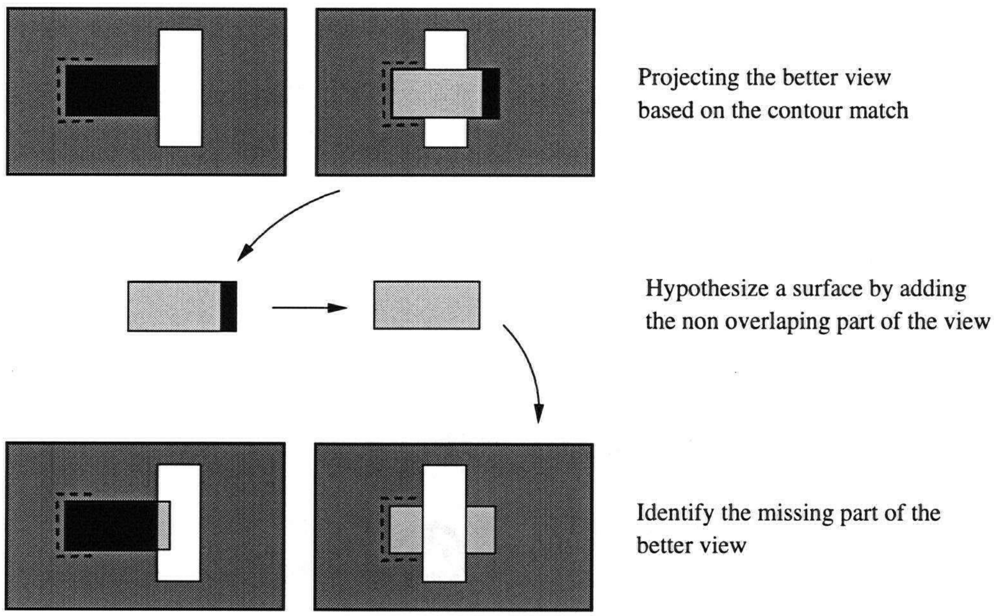
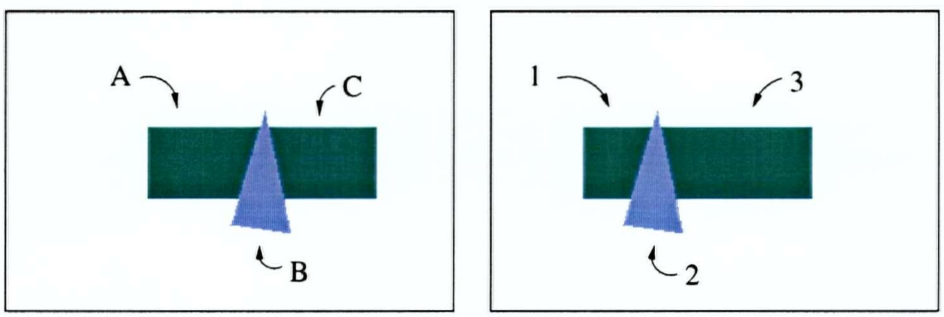


Figure 4.16: Resolving the Split-Partial Occlusion

4.7 Multiple Region Matching

In this chapter I have so far discussed the matching between a region in one image and one, multiple or no regions in the other image. In this section I will discuss the case in which the surface is occluded in such a way that it is imaged in terms of more than one region in both images. None of the previously described methods are sufficient to establish correspondence under this scenario. Consider the example in Figure 4.17. In this scene there are two surfaces, the closer one which is fully imaged into regions B and 2 that can be matched based on strict stereo constraints. On the other hand, regions A,C and 1,3 can not be matched based on the previously discussed constraints, such as partial or split occlusion constraints.



left image   right image

Figure 4.17: Example of a scene with multiple to multiple region matching

For example, regions A and 1 have a partial contour match, however the match between them can not be interpreted as a partial occlusion because the missing part of region A can not be fully accounted for. This is due to the fact that region 3 would have to be matched before it can be determined if the missing part of it region A is behind that surface. Similarly, regions 3 and C also have a partial contour match, however the missing part of region C also can not be fully attributed to an occluding surface. In this case it is necessary to establish multiple region match. The next section describes in detail process of establishing correspondence between multiple regions.

4.7.1 Multi-match Collections

In order to present the algorithm for multiple region matching I will define the following multi-match collection terminology:

- *root_match* is the one-to-one match between two regions that have a partial contour match. This match will be used as a reference for the whole multiple-to-multiple correspondence.
- *left_all* and *right_all* are collections of regions that are believed to belong to the same surface from the left and right view respectively. The collection retains the relative position of regions.
- *total_shape* is a collection of regions that reflects the shape of the surface as perceived from the left and right cameras. *total_shape* is equivalent to the union of *left_all* and *right_all*.
- *left_diff* and *right_diff* are the difference between *total_shape* and *left_all* and *right_all* respectively. *left_diff* and *right_diff* may be seen as the part of the surface that is seen in the "other image" and needs a justification for missing in "this image".

If we consider the example in figure 4.17, a possible *root_match* would be the match between regions A and 1, due to a partial contour match. *left_all* would be the collection containing regions A,C and *right_all* would be the collection containing regions 1 and 3. Figure 4.18 shows *left_all*, *total_shape*, *right_all*, *left_diff*, *right_diff* as they should be at the end of the multiple region match. The next section presents the algorithm that computes these results.

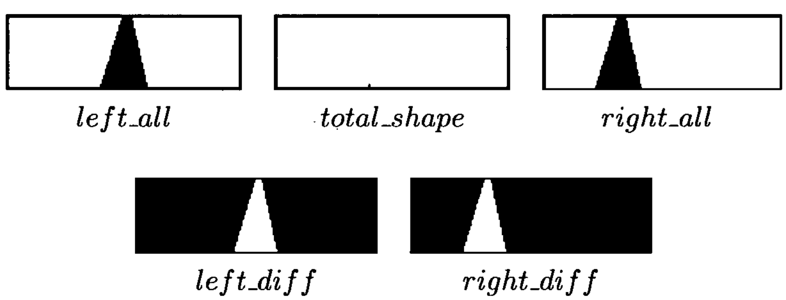


Figure 4.18: Examples of the terminology as related to the scene in Figure 4.17

4.7.2 Multiple-match Algorithm

The following is the algorithm for multiple region matching. The inputs to this algorithm are two regions (root regions) that have a partial contour match. The output of the algorithm is either TRUE, FALSE or WAITING. TRUE means that the for the given root regions there exists a multiple match interpretation. FALSE means that a multiple match interpretation is not possible. WAITING means that the result can not be established because the algorithm is waiting for a region to be matched first.

1. Establish the *root_match* of based on a partial contour match between two regions.
2. Initialize the $left\_all$ and *right_all* with the left and right root region respectively.
3. Initialize *total_shape* by adding the left and right root regions.
4. Determine $left\_diff$ and $right\_diff$ by subtracting *right_all* and $left\_all$ from *total_shape* respectively. (Note that ordering of left and right)
5. Overlap $left\_diff$ and $right\_diff$ on the left and right image respectively based on the contour match.
6. Identify overlapping regions that do not belong to regions that form $left\_all$ or *right_all.*
7. Sort overlapping regions into matched and unmatched regions.
8. If any matched region is behind the hypothesized surface, return FALSE.
9. If there are no unmatched regions go to 14.
10. If the unmatched region does not satisfy weak stereo constraints, return WAITING.
11. Add unmatched regions to $left\_all$, *right_all* and *total_shape.*
12. Go to 4
13. Establish multiple match between regions in $left\_all$ and *right_all*
14. Return TRUE.

4.7.3 Example of the Algorithm Execution

In this section I will step through the algorithm using the example presented in Figure 4.17. I will assume that the match [B,2] was established under strict stereo constraint.

The first step in the algorithm is to identify the root of the multiple match. The *root_match* is set to be the match between region A and region 1. The *left_all* and *right_all* are initialized by regions A and 1 respectively. And *total_shape* is set to the sum of *left_all* and *right_all*. The *left_diff* and *right_diff* are set to the difference between *total_shape* and *right_all* and *left_all* respectively. Figure 4.19 shows the status of the multi-match collections. Figure 4.20 shows the overlap of *left_diff* and *right_diff* on the image.

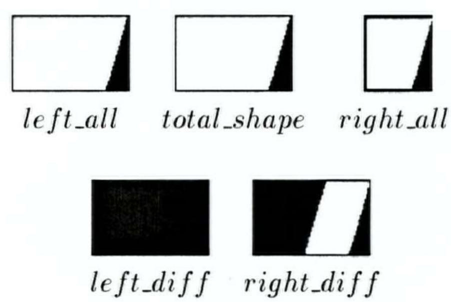


Figure 4.19: First iteration: state of the multi-match collections

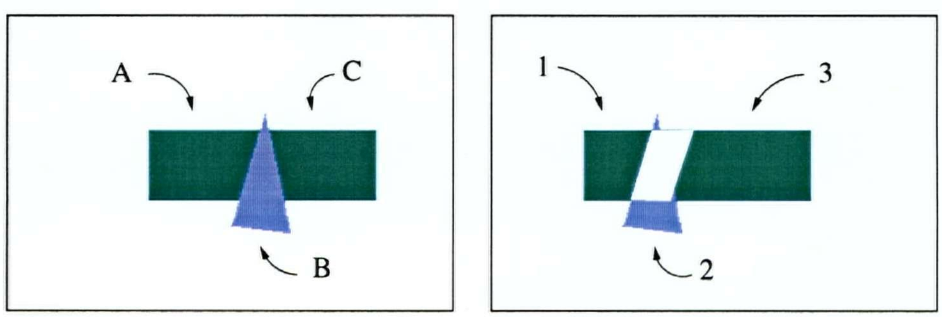


Figure 4.20: First iteration: overlapping *left_diff*,*right_diff*

In the second iteration $left\_diff$ and $right\_diff$ were overlapped on the right and left image respectively. It was determined that the overlaps included regions 2 and 3. Region 2 is matched to region B, and the disparity is larger that the one of the root match. Region 3 on the other was not matched, and it did satisfy the weak stereo constraint. Therefore region 3 was added to the $right\_all$ and $total\_shape$. Figure 4.21 shows the status of the multi-match collections once region 3 was added. Figure 4.23 shows the overlap of the new $left\_diff$ and $right\_diff$ on the image.

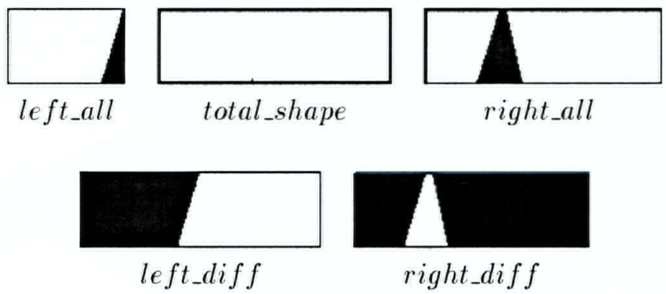


Figure 4.21: Second iteration: state of the multi-match collections

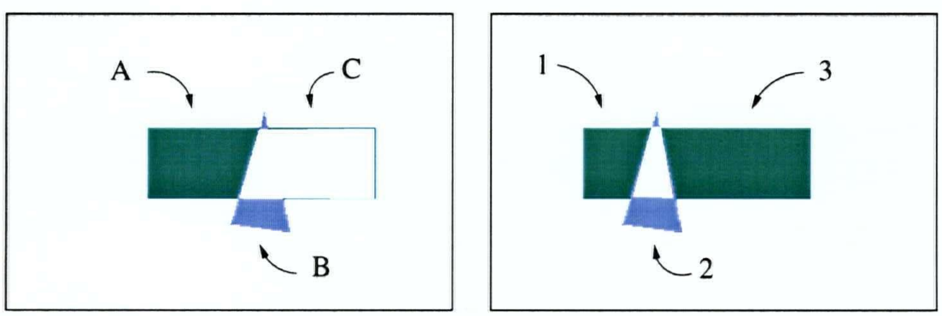


Figure 4.22:

Figure 4.23: Second iteration: overlapping $left\_diff$,$right\_diff$

In the third iteration $left\_diff$ and $right\_diff$ were overlapped on the right and left images and the overlapping regions were determined. In this case regions B and C were found as overlaps. B was determined to be an occluder. On the other hand C, was not matched and did not violate the weak stereo constraint. Therefore C was added to the $left\_all$ and $total\_shape$. Figure 4.24 shows status of the multi-match collections after the addition of the region C. Figure 4.25 shows the overlap of the new $left\_diff$ and $right\_diff$ on the image.

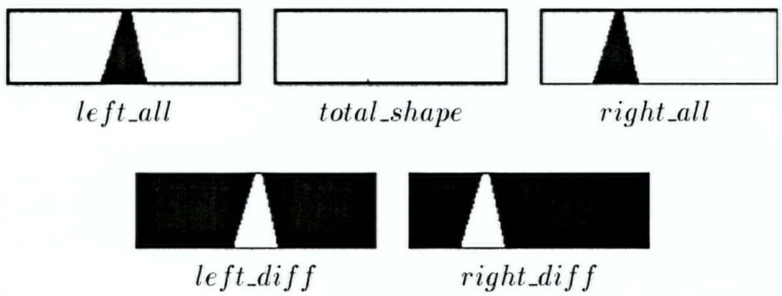


Figure 4.24: Third iteration: state of the multi-match collections

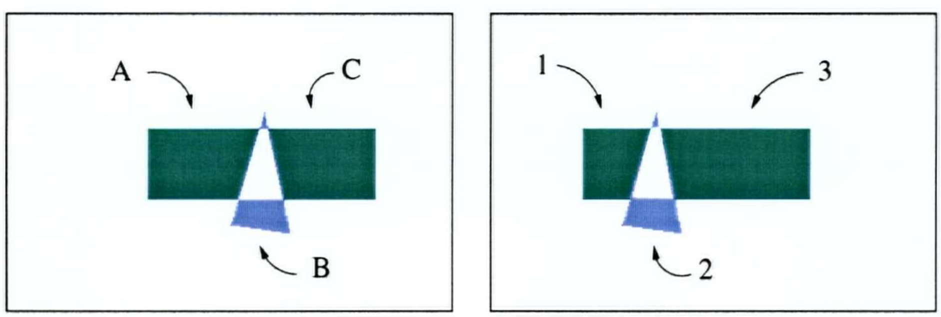


Figure 4.25: Third iteration: overlapping $left\_diff$,$right\_diff$

The process is repeated including region 3 to $right\_all$ and $total\_shape$. At this point no other overlaps were found, and the multiple match constraint was satisfied. The match was therefore [(A,C),(1,3)] with the root in [A,1].

4.8 Multiple Match Generalization

Multiple match constraint may be seen as the generalization of the partial and split occlusion constraints. Partial occlusion is equivalent to one iteration of the multiple match, where no overlapping regions are found and therefore $left\_all$ and $right\_all$ are just the initially matched regions. On the other hand, split occlusions are equivalent to the multiple region matching where either $left\_all$ or $right\_all$ contain only one region and the other one contains more than one region. Split occlusion is stronger because in order to add extra regions partial contour matches are required.

In this chapter I outlined a method for establishing correspondence between regions casted by occlusion between two surfaces. In the following chapter I deal with problems encountered when there are more than one surface.

Chapter 5

Solving Multiple Occlusions

In the previous chapter I discussed match failures due to occlusions. The chapter outlined methods for intensifying the kind of occlusion and resolving the position of the occluded part of the surface. Resolving the occlusion problems was outlined on case basis. In this chapter I will outline a method for resolving scenes in which there are multiple occlusions. Further, I outline possible problems with the solution as and suggest appropriate remedies.

5.1 Inverse Painters Algorithm

The matching problem is a constraint satisfaction problem as discussed in section 3.3. The following is an overview of the approach:

1. Establish correspondence between regions that do not violate any strict stereo constraints.
2. Perform arc consistency checking to establish unique matches.[1]
3. Perform failure explanation on remaining segments.
4. Select matches that have only one explanation.
5. If no new matches are established go to step 7
6. Go to 2
7. The end

This method is related to the well known "painters algorithm" in graphics. The painters algorithm creates an image of the scene by drawing furthest object first. The objects are then drawn into the scene in "further to closer" order. The scene is displayed once the closest object has been drawn. This way the closer objects are occluding the objects further away.

The algorithm outlined above is similar to the painters algorithms, however it works in the opposite direction since the matches are established from the closest surface to the furthest one away. The closest object can be matched using the strict stereo constraints, since it cannot

[1]It is possible that, in some cases, arc consistency will not produce a unique solution, however that problem is left for future work.

be occluded. [2] Other matches can then be established by occlusion explanation from the closest surface to the one furthest away.

Consider the example in Figure 5.1. In this scene we find that there are multiple occlusions in the worse view.The surface A is the closest to the cameras, surface B is occluded by the surface A, and finally surface C is occluded by the surface B. In this scene the above outlined algorithm will have to perform three iterations before all surfaces are matched.

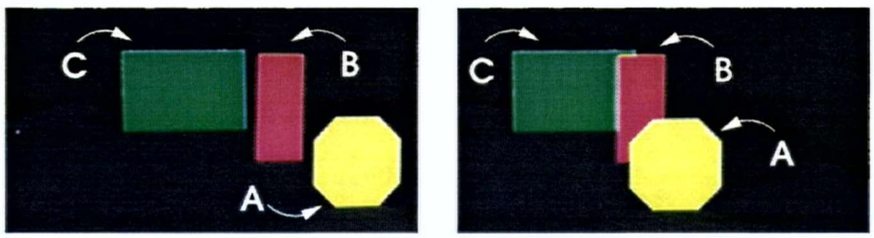
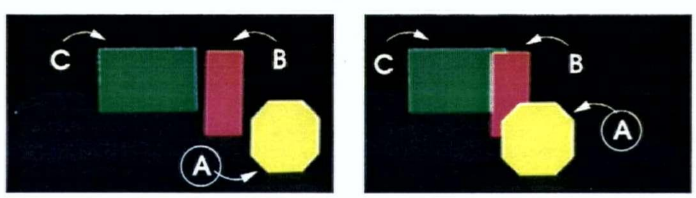


Figure 5.1: Scene with multiple occlusions

In the first iteration of the algorithm left view of the surface A is matched with with the right view of the surface A based on strict stereo constraints. Figure 5.2 shows the matrix of match results under strict stereo constraints (O's mean that the constraints are satisfied and X's mean that they are not). The circled letters show that region has been matched.



right view

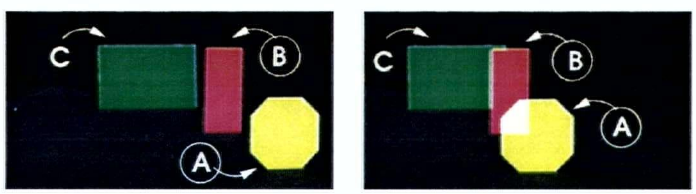
left view		A	B	C
	A	O	X	X
	B	X	X	X
	C	X	X	X

Figure 5.2: 1st Iteration: Strict Stereo Constraints

[2]The closest surface may be only partially imaged and therefore not satisfy the stereo constraints, however the match can be resolved as a partial occlusion

Next, we are interested in explaining the match failure of the remaining regions. For simplicity we will consider only partial occlusions. In order to show that a partial occlusion is taking place we need to show that weak stereo constraints are not violated, there is shared contour, and the missing part of the surface is behind an already matched surface. At this point in the constraint satisfaction surface C can not be matched. Views of the surface C no not violate the weak stereo constraints, they have partial contour matches, however the missing part of the surface in the right view belongs to the unmatched surface B. In other words, we cannot show that the missing part of the surface C is behind the surface B, until we know where surface B is.

On the other hand, match between views of the surface B can be explained as a partial occlusion because the missing part of the surface is behind surface A which is matched. This match is depicted in Figure 5.3.

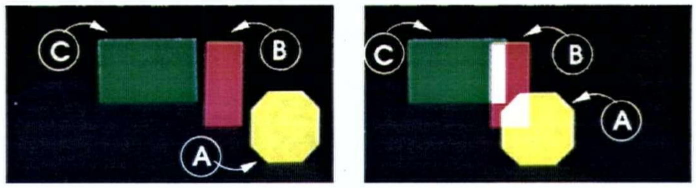


right view

left view	A	B	C
A	O	X	X
B	X	O	X
C	X	X	X

Figure 5.3: 2nd Iteration: Partial Match, Surface B matched

Finally, surface C can be explained as a partial occlusion because the missing part belongs to the surface B which now is matched and is in front of the surface C. The final result is displayed in the figure 5.4.



right view

left view

	A	B	C
A	O	X	X
B	X	O	X
C	X	X	O

Figure 5.4: 3rd Iteration: Partial Match, Surface C matched

5.2 Problem of Multiple Explanations

If we assume that the failure constraints are implemented correctly it is still possible that more than one explanation can be established. Consider the example in figure 5.5. In this example the rectangle and the triangle are matched based on the strict stereo constraint. The problem however, arises from the fact that it is possible that the two triangles belong to different surfaces. It is possible that the surface that is seen in the left image is hidden by the rectangle in the right image and similarly the surface in the right image can not be seen in the left image due to a full occlusion.

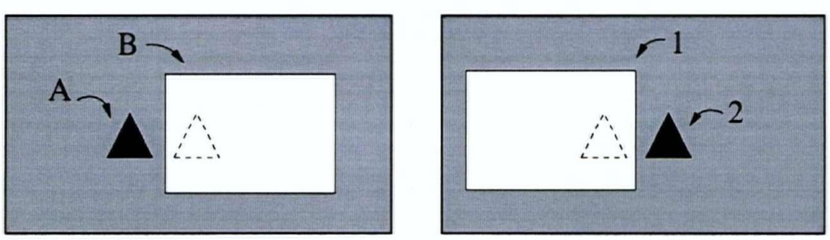


Figure 5.5: Problem with multiple interpretations

Figure 5.6 shows the result of constraint satisfaction process for the strict match and full occlusion constraints. The arrows in the full occlusion constraint indicate that there is a full occlusion between the regions that correspond to the row and the column of the matrix. The occluding surface is pointed at by the arrow.

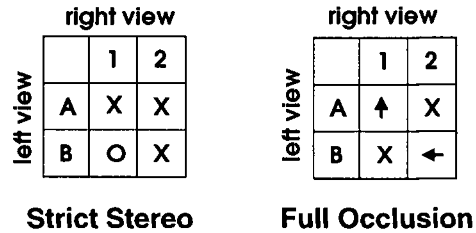


Figure 5.6: Conflict between strict stereo match and full occlusion

Given the available information it is not possible to determine the correct solution to the scene. Several solutions to this problem can be proposed:

Specialized processing: It is possible that there is additional information in two images that can be further analyzed in order to establish weather the two regions belong to the same surface. An example of such processing would be based on correlation between the triangles in order to establish the difference in the fine textures of the regions.

Higher Level Concepts: It may be possible to use higher level constraints such as symmetry, straight line continuity, object volume and so on, in order to heuristically arbitrate between the explanations.

Domain Knowledge: By adding knowledge about the environment in which the images are acquired it is possible to rule out additional explanations. For example, existence of gravity and knowledge of approximate position of the cameras relative to the direction of gravity can greatly improve the results, since objects that are not supported from below can be ruled out as solutions.

5.2.1 Constraint Ordering

The above outlined solutions to multiple explanations of stereo correspondences are an exhaustive approach to solving the problem. These approaches introduce additional processing requirements as well as require in depth research of this particular problem. In this thesis I adopt a constraint ordering scheme that allow to reduce the amount of computation as well as simplify the matching problem.

The constraint ordering identifies the likelihood of various interpretations of the stereo correspondence. The following list presents the ranking of occlusion types:

1. Strict Stereo Constraint
2. Partial Occlusion Constraint
3. Split Occlusion Constraint
4. Multiple Match Constraint
5. Full Occlusion Constraint

This list implies that if there is sufficient evidence for two interpretations of occlusions, the constraint with higher rank will be picked over the constraint with the lower rank. For example, the triangles in Figure 5.5 will be interpreted as one surface that is matched under strict stereo constraint, rather than two surface that are fully occluded in the two views.

This ordering is picked from empirical observations on the likelihood of interpretations. For example, it is unlikely that two regions will satisfy the strict stereo constraints yet belong to two different surfaces. The ordering was also influenced by the amount of information contributed to the disambiguation process. For example, establishing that the occlusion is strict, partial, or split, can place the surface at a well defined 3D position, while full occlusion can not precisely place the surface.

Ordering of the constraints is important from the efficiency perspective. The above ordering also represents the amount of computation necessary to establish that the constraint is true. For example, strict stereo constraint performs a small number of characteristics comparisons, while full occlusions require overlapping (pixel to pixel comparison) of the the region for a full range of disparities. Similarly, split occlusions can leverage from the partial contour matches already established by computations required by the partial occlusion constraint.

Chapter 4 presented the theory of identifying occlusion failures, this chapter extended this theory for multiple occlusion identification. Appendix E discusses the complexity of the algorithm outlined in this chapter. The following chapter presents the implementation and experimental results of this theory.

Chapter 6

Implementation and Results

6.1 Software System Description

In order to test the theory described in this thesis I developed an elaborate software system. The software system was designed to accept two stereo images, segment them into regions and establish correspondence between segments. The system encoded stereo constraints outlined in chapter 3 and occlusion failure constraints in chapter 4. The system produced matches between regions, and the associated constraints that were satisfied in order to establish the correspondence. In some cases, such as full occlusions, the the system would not be able to establish a match, however it would present evidence for the lack of a match. In cases where the correspondence could not be established, the system reported on possible locations of the missing surface. Based on the geometry of the stereo camera setup the matches were then interpreted in terms of three dimensional positions of surfaces.

Aside from accepting images and producing matches between regions, the system was also designed to produce an interactive system that allows the user to create a complex synthetic scene with multiple occlusions. The system would then create images that correspond to the views from two stereo cameras. These stereo images were then analyzed by the software system and matches were established. By pointing and clicking the user was able to inspect characteristics of regions, see the matching regions, set the constraints under which the matching occurs and view the explanation for matches.

The software system was implemented in the Java programming language in order to allow easy access to the users over the Internet. The implementation details of the system are described in Appendix D.

6.2 Experimental Results

The performance of the system was intensively tested using non-trivial images that do not violate the underlying constraints about the environment. The system was expected to segment the images, and establish correspondence between regions in the images. Aside from the correspondence between regions, the results included the explanation for why the regions are matched, as well as why the regions could not be matched.

The images were obtained in two ways, synthetically using the scene editor and real images obtained by a stereo vision setup. The synthetic images were used to exhaustively test the correctness of the system. The system was also tested with stereo images of real scenes that were compliant to the restrictions imposed in this thesis. To achieve easy segmentation based on colour, objects placed in the scene were all of different hue. While all objects place in real scenes were not planar, they allowed easy approximation to a flat surface. Further, all objects were placed such that they were fronto-parallel to the optical axes of the cameras. The only exception was the table that objects were placed on. Since, the table was not a fronto-parallel surface it was ignored in the experiments.

While much effort was put into presenting the results in a systematic and coherent fashion the best way of inspecting the results is by running the system itself as described in Appendix D.

6.2.1 Simple Scenes

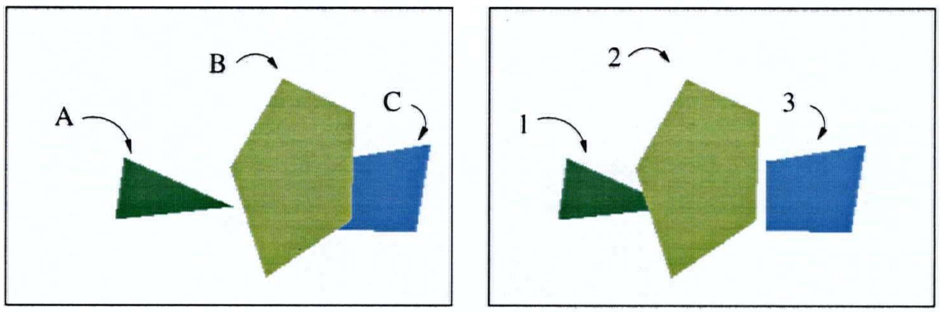
In this section we illustrate the correctness of the system by solving scenes with basic types of occlusion. This section also defines notation and builds intuition needed to interpret the results.

The simplest example of the system's performance can be found in Figure 6.1 which presents a scene with partially occluded surfaces. The arrows pointing at the regions of the image represent the result of the segmentation process. The regions in the left image are labeled with characters and regions in the right image are labeled with numbers. The matches between the regions are presented in Figure 6.2.

Figure 6.2 consists of five matrices corresponding to five types of constraints. Matrix entry 'O' means that regions that correspond to the row and column of the matrix are matched under the constraint that corresponds to that matrix. Similarly, 'X' means that they can not be matched under this constraint. Split occlusion matrices have more than one entry per row or column. Left split occlusion matrix will have more than one 'O' per row. This means that a region in the left image corresponds to more than one region in the right image. Similarly, right split matrix may have more than one 'O' per column. Therefore, a region in the right image corresponds to more than one region in the left image. Finally, the full occlusion matrix may contain the letter 'X' meaning that there is no full occlusion between the two regions, or an arrow meaning that a full occlusion is possible. The direction of the arrow indicates the occluding region. If a matrix has a large grey cross mark, it means that all entries contain an 'X'.

Partial Occlusions

Figures 6.1 and 6.2 show that region B in the left image is matched to the region 2 in the right image under strict stereo constraints. Further, region A matches region 1 and C matches 3 under the partial occlusion constraint.



left image right image

Figure 6.1: Example of a segmented scenes with partial occlusions

Strict Stereo (right view / left view)

	1	2	3
A	X	X	X
B	X	O	X
C	X	X	X

Partial Occlusion (right view / left view)

	1	2	3
A	O	X	X
B	X	X	X
C	X	X	O

Full Occlusion (right view / left view)

	1	2	3
A	X	X	X
B	X	X	X
C	X	X	X

Left Split Occlusion (right view / left view)

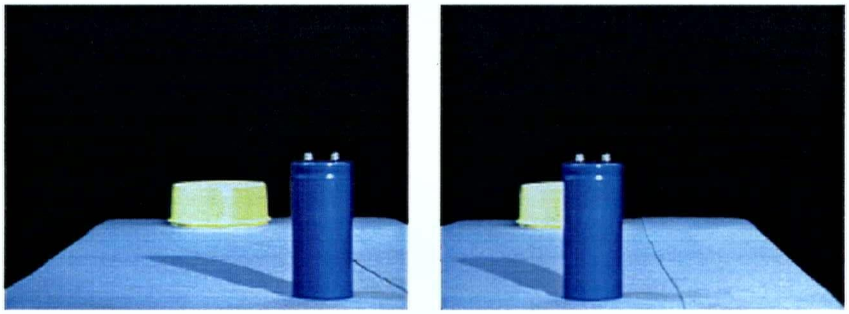
	1	2	3
A	X	X	X
B	X	X	X
C	X	X	X

Right Split Occlusion (right view / left view)

	1	2	3
A	X	X	X
B	X	X	X
C	X	X	X

Figure 6.2: Constraints associated with the Figure 6.1

Figure 6.3 shows a scene with two objects, where the closer object is partially occluding the object further away. The segmentation of images was done by recursive segmentation described in Appendix C. The segmentation process was guided by manually setting segmentation parameters Images in Figure 6.3 were first segmented based on intensity in order to separate the objects from the background. Then the images were segmented based on hue in order to separate between individual objects. The surface that objects are placed on is identified as the background and is not considered in the matching process.



left image right image

Figure 6.3: Real scene with partial occlusion

The views of the closer object in the scene are matched based on the strict stereo constraints. The views of the occluded object are matched based on the partial occlusion constraint, due to the fact that there exists an occluder for a partial contour match. The partial contour match between the views of the occluded object and the missing part of the occluded object are displayed in Figure 6.4.

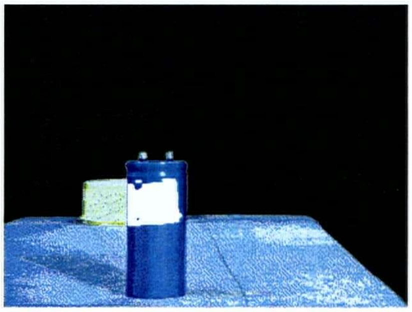
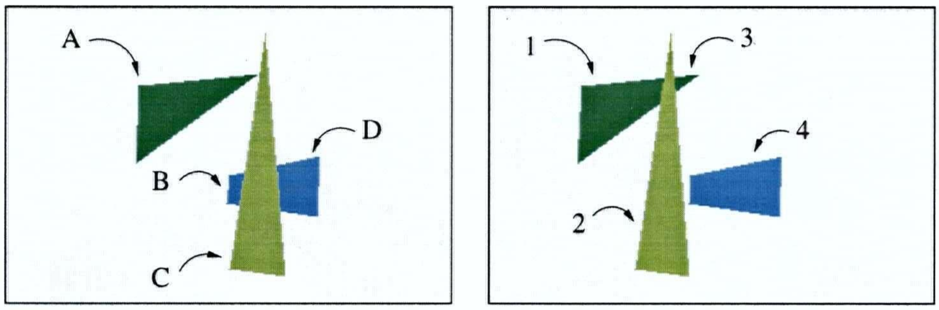


Figure 6.4: Establishing partial occlusion: *the missing part of the occluded object is displayed in white*

Split Occlusions

Figure 6.5 presents a scene with surfaces that are split-occluded. The system correctly matches the regions and identifies the type of occlusion as a split occlusion. It should be noted that the better view of the occluded surface is matched to two regions in the worse view.



left image   right image

Figure 6.5: Example of a correctly solved scene with split occlusions

Strict Stereo (right view across, left view down)

	1	2	3	4
A	X	X	X	X
B	X	X	X	X
C	X	O	X	X
D	X	X	X	X

Partial Occlusion (right view across, left view down)

	1	2	3	4
A	X	X	X	X
B	X	X	X	X
C	X	X	X	X
D	X	X	X	X

Full Occlusion (right view across, left view down)

	1	2	3	4
A	X	X	X	X
B	X	X	X	X
C	X	X	X	X
D	X	X	X	X

Left Split Occlusion (right view across, left view down)

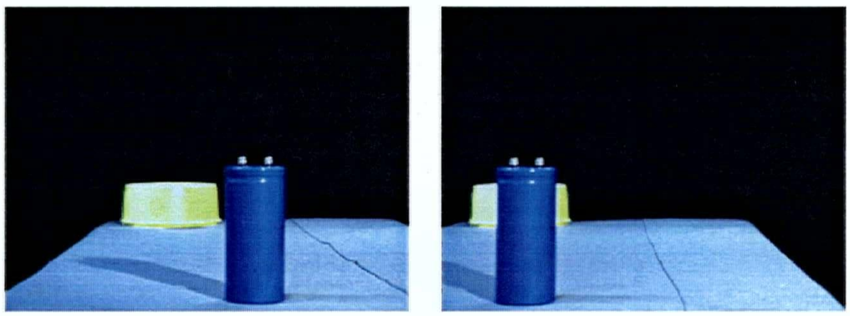
	1	2	3	4
A	O	X	O	X
B	X	X	X	X
C	X	X	X	X
D	X	X	X	X

Left Split Occlusion (right view across, left view down)

	1	2	3	4
A	X	X	X	X
B	X	X	X	O
C	X	X	X	X
D	X	X	X	O

Figure 6.6: Match constraint associated with the scene in Figure 6.5

Figure 6.7 shows a scene that contains a split occlusion. The images were segmented out by separating the background based on intensity, and the objects were segmented based on hue. The occluded object is imaged in terms of one region in the left image, and in terms of two regions in the right image.



left image　　　　　　right image

Figure 6.7: Real scene with split occlusion

The closer object in the scene is matched based on strict stereo constraints. Figure 6.8 show the partial contour matches between views of the occluded object, as well as the missing part in the occluded view.

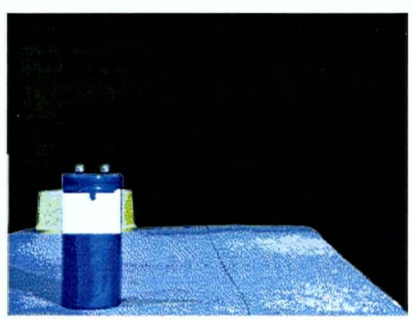
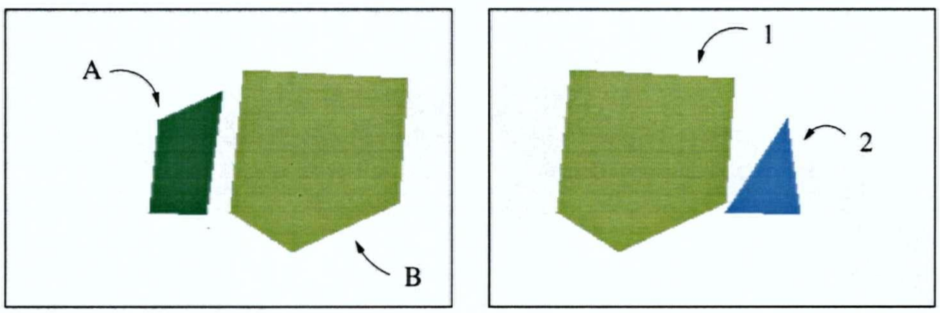


Figure 6.8: Establishing split occlusion *the missing part of the occluded object is displayed in white*

Full Occlusions

Figure 6.9 depicts a scene in which two surfaces are fully occluded. The system correctly identifies surfaces that are fully occluded, identifies the occluding surface and the disparity range in which occluded surface must be in. It was determined that region A must be in the range between 0 and 5, and region B must be in the range between 0 and 8.



left image　　　　right image

Figure 6.9: Example of a correctly solved scene with full occlusion

Strict Stereo (right view / left view)

	1	2
A	X	X
B	O	X

Partial Occlusion (right view / left view)

	1	2
A	X	X
B	X	X

Full Occlusion (right view / left view)

	1	2
A	↑	X
B	X	←

Left Split Occlusion (right view / left view)

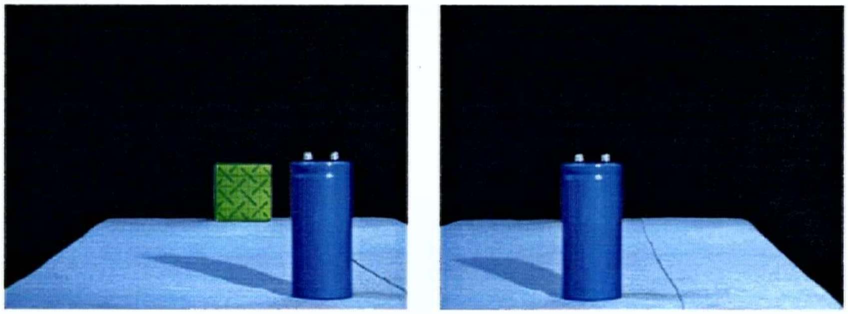
	1	2
A	X	X
B	X	X

Right Split Occlusion (right view / left view)

	1	2
A	X	X
B	X	X

Figure 6.10: Match constraint associated with the scene in Figure 6.9

Figure 6.11 shows a scene in which there are two objects and one of the views object further away is fully occluded by the closer object. Segmentation is done by separating the objects from the background based on intensity, and objects are separated based on hue.



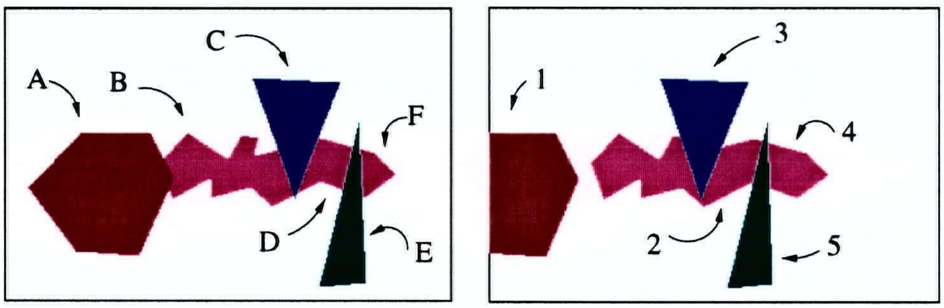
left image right image

Figure 6.11: Real image with full occlusion

In this scene it was determined that the further object is fully occluded in the right image. While correspondence between the view of the occluded object can not be established, it was possible to determine the disparity range in which the match must lie in. In this case, the occluded object lies in the interval between 41 and 45 pixel disparities.

6.2.2 Elaborate Scenes

Figure 6.12 shows a scene with multiple to multiple region correspondence. This scene contains four surfaces, two surfaces are fully imaged and are therefore matched based on the strict stereo constraint, one surface is partial occluded due clipping of field of view, and matched under partial occlusion constraint. On the other hand, the furthest surface is imaged in terms of three regions in the left image and in terms of two regions in the right image.



left image right image

Figure 6.12: Example of multiple region matching

Figure 6.13 shows the matches between the regions in the images. In the figure cells without a character in them, represent that the constraint was not satisfied, in other words, a blank is equivalent to an "X". The non-occluded triangular regions are matched based on strict stereo constraints. The left most surface is matched based on partial occlusion constraint, where the missing part of the surface in the right image is not visible due to the field of view.

While matches (A,1), (C,3), (E,5) are established due to the strict stereo and partial occlusion constraints, the rest of the regions were grouped and correspondence was established based on the multiple match constraint. The multiple match table in the Figure 6.13 shows the correspondence between the remaining regions. Character "O" shows that match between region B and 2 is the root of the multi-match correspondence. Regions D,F and 4 are part of the multiple match correspondence. The multiple region correspondence defines a surface. Figure 6.14 shows the shape of the surface from the left and right view as well as the total imaged shape of the surface.

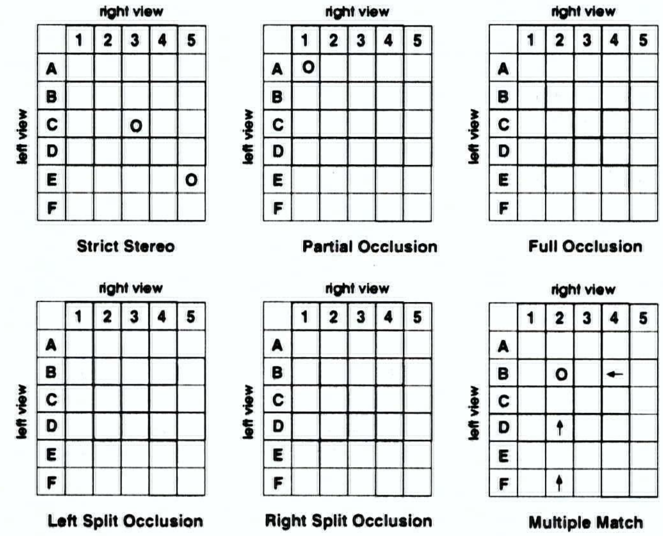


Figure 6.13: Match constraints associated with the scene in Figure 6.12

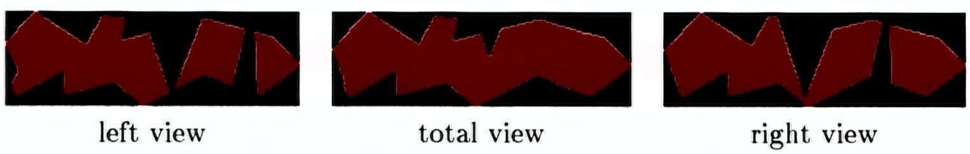
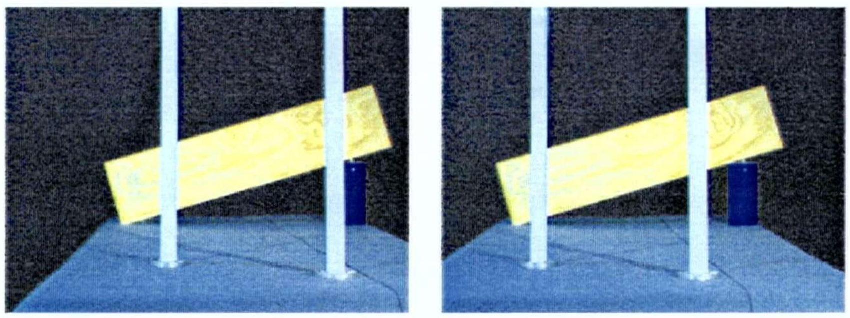


Figure 6.14: Shape of the surface established through multiple region correspondence

Figure 6.15 shows a real scene with a case of a multiple region match. The images were segmented out based on intensity in order to extract the object from the background and individual objects were separated by segmentation in hue. The vertical poles were matched by strict stereo constraints. The wooden board was imaged in terms of three regions in each image. The left-most region in each image were established as the root of the multiple region match. The other four regions were added to the total view of the surface. Figure 6.15 shows the total shape of the surface, as well as the view of the surface from the left and the right image.



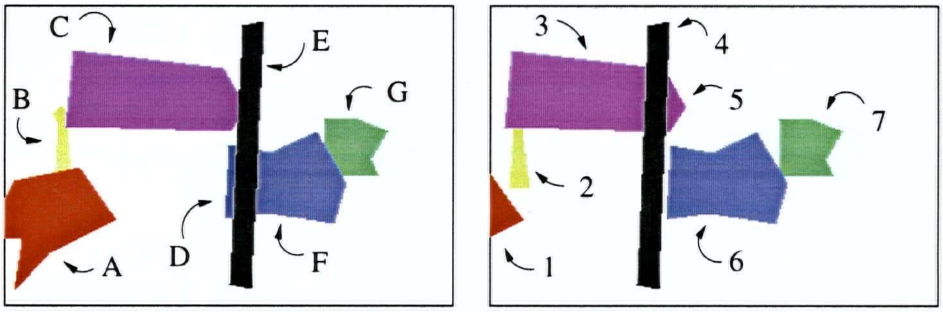
left image right image

Figure 6.15: Scene with a multiple region matching

left view total view right view

Figure 6.16: Shape of the surface established through multiple region correspondence

Figure 6.17 shows a scene with multiple occlusions of different types. This example illustrates that the algorithm correctly identifies occlusion types when they need to be identified in a closer to further sequence.



left image right image

Figure 6.17: Multiple occlusions of different types

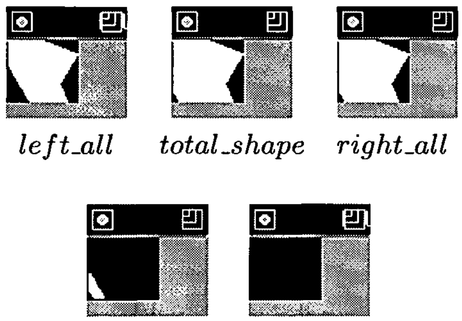
Regions E and 4 were matched based on strict stereo constraints. The regions A and 1 are matched based on the partial occlusion constraint since there exists a contour match and the missing part of the surface is clipped by the field of view. The rest of the regions at this point were not matched since an interpretation could not be found. In the next iteration split occlusions [(D,F),6] and [C,(3,4)] were established since the missing part of the surfaces could be attributed to the occluding surface established by the matches [E,4]. In the next iteration, matches [B,2] and [G,7] were established based on partial occlusion constraint. At this point correspondence between all regions was established and the matching process was over.

The system allows the user to ask for an explanation for the result of a match between two regions. For example the output from the system for the explanation of the match between region G and region 7 is:

```
Constraint: root = TRUE because  of
  Constraint: strict_stereo = FALSE because  of CONJUNCTION
    area constraint = FALSE
    epipolar constraint = TRUE
    Constraint: range = UNKNOWN
    width = TRUE
    height = TRUE
    first moment = TRUE
    second moment = TRUE
    color = TRUE
  Constraint: partial occlusion = TRUE because  of CONJUNCTION
    Constraint: no match exists = TRUE because  of NEGATION
      MatchExistConst = FALSE
    Constraint: not noise = TRUE because  of NEGATION
      Noise = FALSE
    Constraint: not noise = TRUE because  of NEGATION
      Noise = FALSE
    Constraint: weak stereo = TRUE because  of CONJUNCTION  size = 2
      color = TRUE
      bright = TRUE
    partial match = TRUE because of
    Exists: contour match = TRUE
      MissingOcclConst: occluder size = 2
      MissinOcclConst = WAITING
      MissingOcclConst: occluder size = 1
      MissingOcclConst: occ_disp= 29 disp = 20
      MissinOcclConst = TRUE
```

The output is formated such that each line represents a constraint. The constraint may have a name, value and type. For example, the second line is a *strict stereo constraint* that is *not satisfied* due to a *conjunction* of a number of constraints. The following lines of text that are indented represent constraints which form the conjunction of the strict stereo constrain. Similarly, partial occlusion is a conjunction of a number of constraint. Finally, the first line shows if all constraints have been evaluated.

This output, therefore, means that strict stereo constraint could not be satisfied due to a violation in the area constraint between regions G and 7. On the other hand, partial occlusion constraint is satisfied because there the regions have not been matched yet, the regions are not noise (specified by the size of the region), weak stereo constraints are satisfied, and there exists a contour match such that there is a surface that explains the missing part of region G. Aside from the output the system also displays the shape of the regions, the total shape and the missing part of the occluded surface, as shown in Figure 6.18.



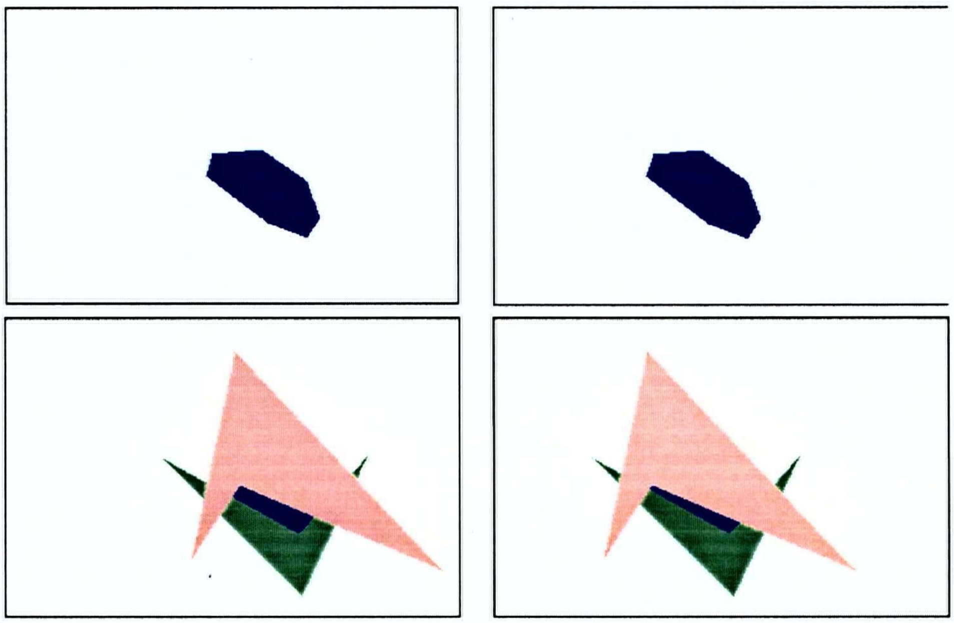
left_all *total_shape* *right_all*

left_diff *right_diff*

Figure 6.18: Explanation of the reason for matching regions G and 7 under partial occlusion constraint

6.2.3 Challenging Scenes

Consider the scene presented in Figure 6.19, the top pair of images shows stereo view of an object, the bottom pair shows the same view when two additional surfaces were added in the scene. The new surface were placed such that the true boundary of the object is not seen.



left image right image

Figure 6.19: Matching based on occluding vs. bounding contours

The algorithms correctly identifies correspondence between images for the regions that correspond to the two closest surface. The regions that belong to the furthest object are matched based on partial occlusion constraint. The match however is not correct because the partial contour match was established on the occluding edge rather on the bounding edge. In this case it should be recognized that the displacement based on this match may not be correct, however it does provide with a minimum distance that the farthest surface has to be at. The minimum distance therefore is equivalent to the second farthest object.

Scene presented in Figure 6.20 consists of two surfaces. The images however do not have sufficient evidence in order to piece together the surface farther away. This example contains all kinds of occlusions discussed in this thesis. The system developed is successful in identifying the following interpretations:

Strict Stereo Constraint:	(G,5)
Partial Occlusion:	(F,4)
Split Occlusion:	([C,D],3)
Multiple Match:	([A,B],[1,2])

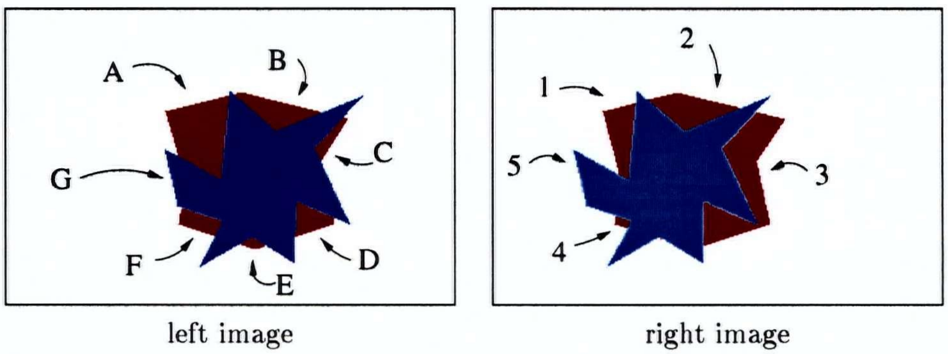


Figure 6.20: Severely occluded surface

In this case there is not enough evidence that all matched regions belong to the same surface. On the other hand, the output of this algorithms could then be very easily used in a higher level process that could take advantage of the fact that all the regions have similar spectral properties and could belong to the same surface.

6.3 Summary of Results

The system I developed shows that the theory of region-based stereo correspondence developed in this thesis is robust to occlusions. The robustness is achieved by identifying the the occlusion type, and based on this information, establishing depth measurements for occluded parts of the scene.

The density of the depth images obtained using this method is maximal, because pixels that belong to the foreground object, in both images, are all assigned a depth value. In case of fully occluded objects it was possible to assign a range of depths that the object must be in.

Since all pixels are given a depth value, it is not necessary to perform any kind of interpolation on the depth values. Interpolation in general implies an approximation to the real values and often results in blurred boundaries between objects. The approach taken in this thesis produces crisp boundaries.

Matching multiple regions is a novel approach that has not been discussed in the stereo vision literature. The advantage of allowing matching between multiple regions in one image to a number of regions in the other image is that the obtained depth map eliminates some of the shadowing effects that standard stereo vision algorithms have. In other words, the obtained depth maps may have multiple levels of accurate depth values.

The clipping effect of the field of view is a common problem in matching between stereo images. The approach in this thesis allowed accounting for these clipping effects as occlusions. This way surfaces that are not fully visible from one camera could still be placed in 3D.

The next chapter discusses the relevance of the results to the computer vision research and outlines possible extension to the approach taken in this thesis.

Chapter 7

Discussion, Possible Extensions and Conclusion

7.1 Discussion and Possible Extensions

This thesis analyses region based stereo matching in a very restricted domain. While the algorithm presented performs well in this restricted domain, it requires further extension for less restricted domains. Work presented in this thesis can be extended to become more suitable in a wide range of scenes. In order to achieve this it is necessary to lift the restrictions and model failures that are introduced by each less restricted domain. In this chapter I address the main issues related to lifting these restrictions and outline future improvements on the method. Finally, I conclude by summarizing achievements in this thesis and their impact on future research.

7.1.1 Segmentation Issues

The main assumption made in this thesis is that images are segmented out into regions that correspond to the surfaces in the scene. In many scenes achieving this is not trivial.

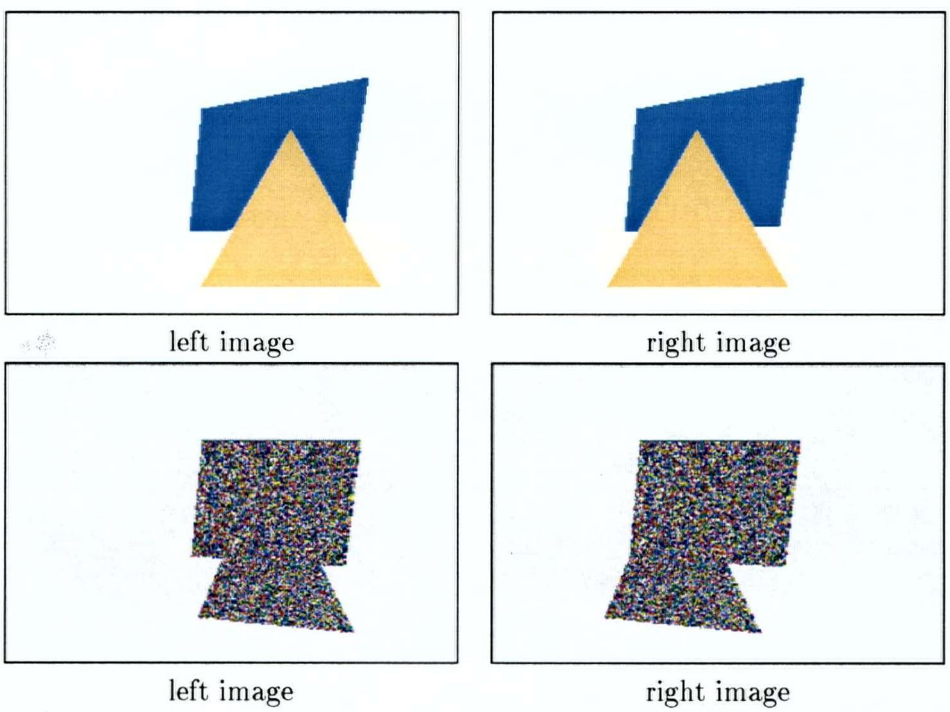


Figure 7.1: Example of a scene with same surfaces, yet different textures

Consider the scene in Figure 7.1. In this figure there are two pairs of stereo images of the same scene. The difference between scenes is that surfaces in the top pair can be segmented out trivially, while the surfaces in the bottom pair can not be segmented based on a single view. In the segmentable case matching algorithm presented in this thesis will work perfectly. On the other hand, the random dot images do not contain any monocular evidence for the surfaces and therefore the algorithm will not be able to establish correspondence between the images.

On the other hand, algorithms such as area based correlation would do very well on the random dot images, while on the textureless surfaces they would get reliable matches only around the borders of the images. This set of images illustrates the domains in which some algorithms are better than others. The framework presented in this thesis allows for incorporating the appropriate algorithm depending on the kind of scene encountered.

In order to deal with the scenes that has portions that can not be segmented into surface , a new kind of failure description must be introduced. If this kind of failure is detected segmentation into surfaces could be done based on the results of the area based correlation, if there exists enough texture.

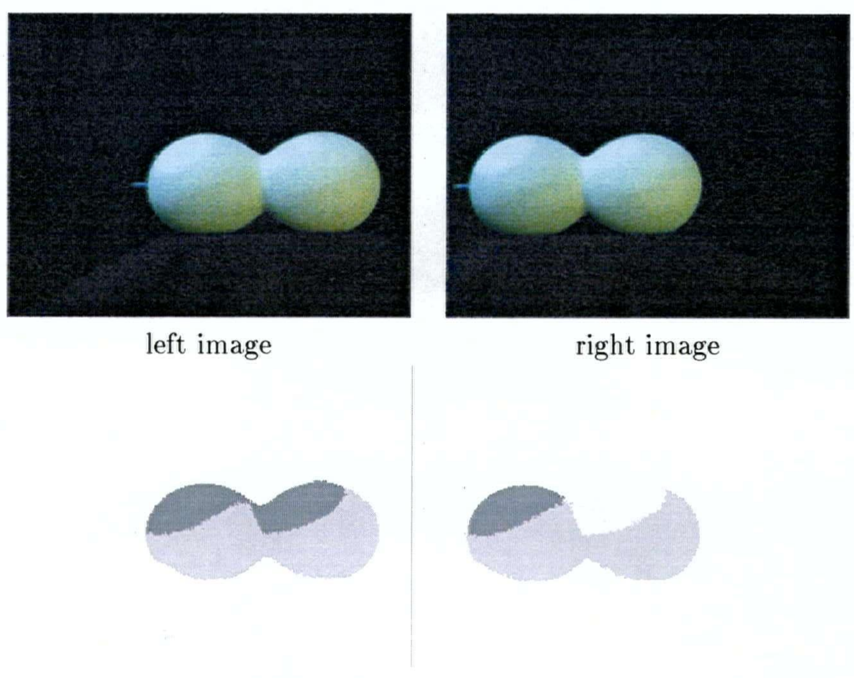


Figure 7.2: Example of segmentation mis-behavior

Aside from scenes that can not be segmented at all, there are cases where the segmentation is only partially correct and therefore not suitable for matching. For example, Figure 7.2 shows an example where a surface may be segmented out in terms of more than one region. Further, it is possible, as shown in the figure that the surface in the left image may be segmented out in terms of two region, while the surface in the in right image may be segmented into three regions. In this case multiple-to-multiple matching scheme can be used, however it would be much better if the stability of the region is determined prior to matching.

In this example there is also the issue that the surface considered is not planar. Region-based matching can be used to establish correspondence between the object itself, however it would not be suitable for determining the shape of the surface. Methods such as correlation and/or shape from shading would be more appropriate for determining the shape of the surface. Region-based matching would still be useful in identifying the part of the image that specialized operation should be performed on.

The assumption that a single image can be segmented into regions is not very realistic. A more realistic segmentation algorithm would be one that is required to produce regions that include the whole surface, but may include more than one surface in a region. Figure 7.3 is an example of a segmentation result that would include more than one surface in a region. Appendix F discusses how methods from this thesis can be applied to improving segmentation.

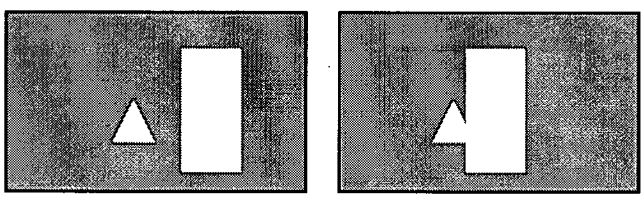
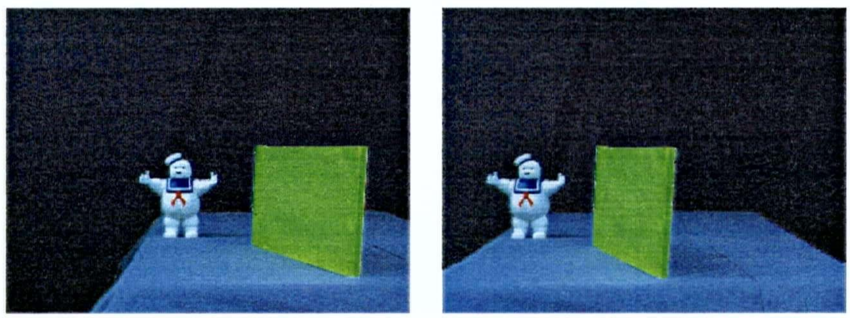


Figure 7.3: Segmentation artifact that may include more than surface in a region

7.1.2 Surface orientation

This thesis assumes that all surfaces in the environment are fronto-parallel to the optical axis of the cameras. This, however, is simply not true in most natural scenes (an exception being some aerial photography of urban scenes). Therefore, lifting the fronto-parallel restriction is one of the most important tasks for the future work.

If the surface is is not fronto-parallel, then the regions that the surfaces will map onto is guaranteed not to have the same shape in both images. The strict stereo constraint would therefore fail and other constraint would not be able to explain the cause for the change in shape.



left image   right image

Figure 7.4: Example of a non-fronto-parallel surface

Consider the example in Figure 7.4 where a surface in the scene is not fronto-parallel[1].The properties of the regions that correspond to the rotated surface are significantly different between the two views. The match between the regions would violate a number of constraints including the width, area, and first and second moments. In this case the matching process will not be able to produce a match between the two view of the surface since there does not exist an occluder that would justify the change in the properties.

In order to solve the problem of perspective it is necessary to understand the constraints on the changes in the shape of the regions. As with occlusions, contours of the regions could be explored in order to establish correspondence between parts of regions.

If there exists texture within the region, correlation methods could be used to establish correspondence between points within the region. In this case uniqueness and ordering constraints can be imposed to improve the results. One would have to be careful not to use correlation methods if the difference between viewing angle from the two views is very large.

The accuracy with which the orientation of the surface can be established may greatly vary.

[1]The surface that objects are placed on is also in this category however it is ignored for simplicity

Further, it may be possible that occlusions can be mistaken for perspective foreshortening. It would be important to establish the relationship between perspective effects and occlusions, and develop a theory for disambiguation between the two.

7.1.3 Lighting effects

Segmenting may become more difficult when we allow lighting to be such that surfaces cast shadow on each other. Shadows may introduce boundaries between regions that belong to the same surface. Shadows, however, are not a significant problem in matching, because they will appear in both images. Shadows, if detected, can identify the approximate location of the light source. Knowledge of the approximate position of the light source can introduce additional constraints that can be used to disambiguate the interpretation of the scene.

Specularities are a combined artifact of lightning and surface properties. Specularities are a serious problem in stereo matching because they can not be perceived at the same position of the surface. It is, however, important to identify that a part of the image corresponds to a specularity. There are a number of probabilistic methods that can be used to determine that a regions is a specularity. In this case the matching between specularities, even if satisfied under strictest constraints, should not be used to produce a depth measurement.

Kind of lighting conditions, diffuse light, point lighting and so on would have to be analyzed on individual basis. Multiple light sources would also be taken in account.

7.1.4 Arbitration

Lifting constraints allows for multiple interpretations of the scene. As mentioned earlier, perspective foreshortening could be mistaken for an occlusions. In cases where there are multiple interpretations it is necessary to choose the most likely one. In order to identify the right interpretation it is necessary to incorporate knowledge about the domain that images are obtained in.

General knowledge about the shape and size of the surface, volume and size of objects that surfaces belong to, usual location of the light source, phenomena such as gravity and rigidity can be used to disambiguate between interpretations. For example, if the size of the largest surface in an environment is known, then there exists an upper bound on how far a surface can be, given that size of the region it corresponds to is known. Further, in cases where information about the environment is not available it is possible to use heuristic reasoning such as picking the simplest interpretation of the scene. Much of work done in 1970s in fields of artificial intelligence and computer vision are very relevant to issues discussed in this thesis. Resurrection of some of the older methods combined with the more modern knowledge of computer vision may prove to be very synergistic.

7.1.5 Learning

Lifting individual constraints on the domain and introducing solutions to the new effects is a tedious process. It is possible that there exists an underlying pattern that could be discovered after several iteration of refining the matching algorithms. Therefore, the next challenge can be viewed in terms of automating the process of lifting constraints. Many methods developed in the field of machine learning may be applicable in achieving this task.

Future work may include interaction with a user as a teacher providing examples to the system. The user could be specifying the matches between images and the computer system would infer the matching rules. This avenue would be particularly interesting because it would allow computer vision programming on a "show examples" principle, rather than on hard coding. Higher level generalization of lower level computer vision concepts would greatly help achieving this goal.

7.2 Conclusion

In 1982 Ballard and Brown [5] made the following observation about the correspondence problem:

One solution is to identify world features, not image appearance, in the two views, and match those (the nose of a person, the corner of a cube). However, if three-dimensional information is sought as help in perception, it is unreasonable to have to do perception first in order to do stereo.

In this thesis I argued a somewhat opposite point of view. The thrust of my thesis was on utilizing higher level knowledge in order to produce more robust and versatile results. I argued that top-to-bottom flow of information is just as important as the bottom-up flow of information. The process of selection of appropriate algorithms and their parameters was seen as the instrumental process in propagating information from higher levels of abstraction to lower levels of abstraction. Identifying reasons for algorithm failures and explicit modeling of these failures was the focus of this thesis.

Failures in region-based stereo vision due to severe occlusions were the specific problem considered as an example of failure modeling. Partial, split and full occlusions were defined as failures that may cause standard region-based algorithms to fail. Identification of these failures allowed matching between multiple regions, establishing the shape of the surface that could not be determined form a single view of the scene. Matching between multiple regions is unique to this thesis, and exhibits the advantage of region based matching over other stereo vision algorithms. Approach taken in this thesis also allows 3D placement of a surface in cases when a correspondence can not be established at all, due to a full occlusion. While in other approaches unmatched surface would not be placed in 3D at all, in my approach unmatched surface are placed in 3D, often with small errors.

The algorithms presented in this thesis are designed to deal with a specific problem, namely occlusions in region-based stereo vision. As discussed in this chapter these algorithms do not offer a universal solution to the image correspondence problem. This thesis, however, does provide a framework for a structured interfacing between computer vision algorithms at various levels of abstraction. I believe that this framework will be useful in not only stereo matching, but also general correspondence problems such as correspondence in motion matching and arbitrary scene view correspondence.

Appendix A

Stereo Constraints

A.1 Epipolar Constraint

Given a known imaging geometry, the epipolar constraint defines a line in one image along which a match can be found for a point in the other image [20]. Consider the point P^R in the Figure A.1. P_1, P_2, P_3 are some of the possible points in the scene that may be projecting into the point P^R in the right image, assuming the pinhole imaging model with pinholes at the point S^R for the right image and point S^R for the left image. Points P_1^L, P_2^L and P_3^L are the projections of the points P_1, P_2, P_3 onto the left image. The epipolar constraint forces the points P_1^L, P_2^L and P_3^L to lie along a straight line. Similarly, for a point in the left image there exists an epipolar line in the right image that the matching point must be on.

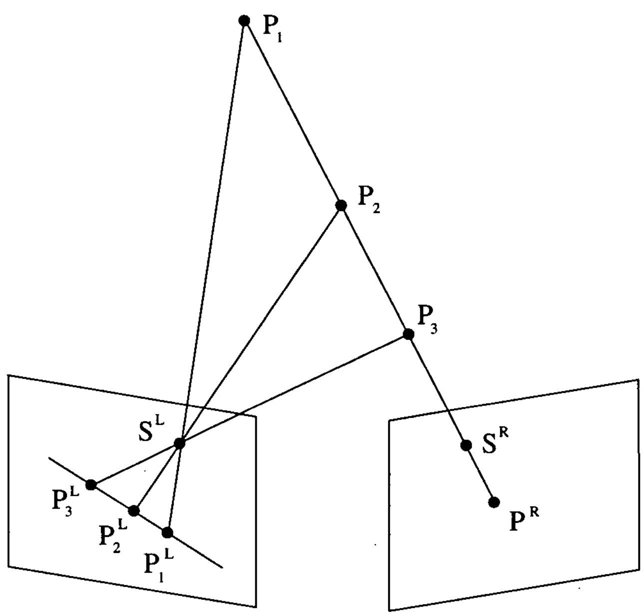


Figure A.1: Epipolar Constraint

The implication of the epipolar constraint is that the search for a matching point in one image needs to be done only along one line in the other image.

A.2 Range Constraint

The range constraint imposes restrictions on what part of the epipolar line it is possible to find a match. In Figure A.2 the point P_{min} represents the projection of the point P when it is the closest to the right camera. Point P_{max} represents the projection of the point P when it is at infinity. The range along the epipolar line between P_{min} and P_{max} is the possible position of the projection of the point P. Therefore, the search for the match should be restricted only to that interval in the image.

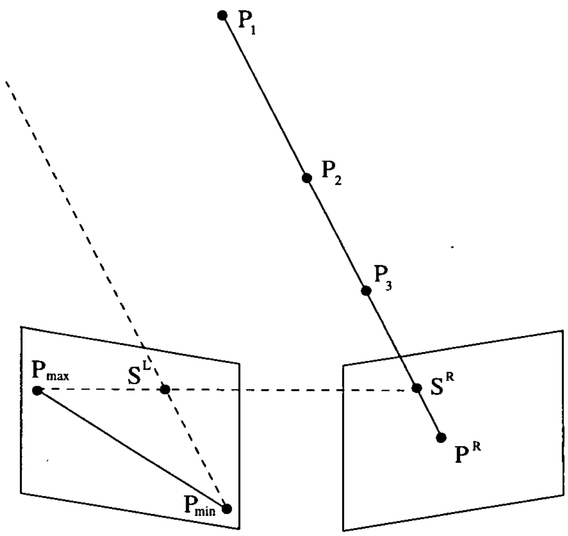


Figure A.2: Range Constraint

A.3 Uniqueness Constraint

Under uniqueness constraint, a point in one image should have at most one match in the second image. This constraint is true if objects in the scene are opaque. Uniqueness constraint is useful in reducing the number of possible correspondences.

A.4 Continuity Constraint

Continuity constraint limits the world to smooth surfaces. The main assumption is that the depth values in the result of stereo correspondence change smoothly. While this assumption is not true in all scenes, it is useful for techniques such as dynamic programming stereo.

A.5 Ordering Constraint

Ordering constraint as also an assumption about the world that is often true in regular stereo scenes. The assumption is that the order of the corresponding points is preserved along the epipolar line. This assumption is not true under severe occlusions, however it is useful for many standard stereo vision techniques.

Appendix B

Contour Matching

In this chapter I will outline the method used for establishing the correspondence between the contours of regions. The problem at hand is: given two regions in different stereo views, produce their contours and identifying parts of the contours that may correspond to the each other. We assume reliable segmentation and epipolar geometry.

The contours can be represented as a circular array of (x,y) image coordinates.

$$C_i = x_i, y_i : i = [0..S]$$

Where x, y are the coordinates of one pixel and S is the number of pixels in the contour.

The contour can be obtained by recording an arbitrary edge point of a region and tracing the boundary in a clock-wise direction. Since the contour array is circular and the first element is arbitrarily chosen, any pixel of the left contour may match any pixel of the right contour. Therefore, any two pixel can be compared based on two criteria:

- The contour pixels match does not violate the epipolar constraint.
- The clockwise tangent vectors are similar by orientation and direction.

This can be written as:

$$T(C_i) = atan(\delta x_i, \delta y_i)$$

$$F(C_i^{left}, C_j^{right}) = |T(C_i^{left}) - T(C_j^{right})| < \epsilon^T \wedge |y_i^{left} - y_j^{right}| < \epsilon^y$$

Where T is the direction of the tangent vector, and F is a boolean value that indicates if two contour pixels match.

We can now establish a matrix of pixel matches:

$$M_{ij}(C^left, C^right) = F(C_i^{left}, C_j^{right}), i = [0..S^{left}]j = [0..S^right]$$

We are interested in consecutive pixels that satisfy the tangential and epipolar constraint F. Therefore we need to find contour segments $L = (i, j, l)$, where i and j are the the first pixel in the first matching pixels of the left and right contour segments respectively and where l is the length of the matching contour segment.

A contour segment must satisfy the following criteria:

$$\begin{aligned} L(C^{left}, C^{right}, i, j, l) \rightarrow \quad & \forall k = [0..l], M_{(i+k)(j+k)}(C^{left}, C^{right}) = true \\ & \wedge M_{(i-1)(j-1)}(C^{left}, C^{right}) = false \\ & \wedge M_{(i+l+1)(j+l+1)}(C^{left}, C^{right}) = false \end{aligned}$$

Finally, the result is a set of contour segments that are of a significant length, ie.

$$L(C^{left}, C^{right}, i, j, l) \ where \ l > \epsilon^{l}$$

Appendix C

Segmentation Algorithm

The correspondence between images was determined in terms of region segmentation of the images. The main requirement of the segmentation for the correspondence problem is that the same parts of the scene are segmented into regions in both stereo images. The segmentation algorithm should ensure that the difference of shape and size of the regions are only an effect of the scene and camera geometry, such as occlusions and perspective. In other words, algorithms that significantly alter the shape or size of a region when viewed from different viewpoint are not acceptable. The segmentation algorithm should also be efficient.

The process of regions segmentation is a well researched yet a difficult field of computer vision. There are many standard methods for image segmenting; threshold based techniques [50], [51], [56], segmentation by region growing [13], [21], splitting and merging [32], relaxation [55]. Evaluation and comparison of these algorithms is a particularly difficult problem. There are a number of papers that address this problem from the viewpoint of cytology [52], [67]. Analytical evaluation of segmentation is difficult since no formal solution to image segmentation has been found yet. An alternate evaluation of segmentation algorithms is done in a number of empirical methods. One of them is the empirical goodness methods such as, goodness based on intra-region uniformity, inter-region contrast and goodness based on region shape. Empirical evaluation is also done based on discrepancy methods, such as discrepancy based on the number of miss assigned pixels, discrepancy based on the number of objects in the image, and discrepancy based on feature values of segmented objects.

In this thesis we are interested in consistent performance of the algorithm with the change of viewing position. In other words, the inaccuracies of the segmentation algorithm are acceptable as long as they are consistent in both views of the stereo setup.

The segmentation within the scope of the stereo problem has two advantages: a) the difference in the position of the cameras is relatively small resulting only in small changes in the reflectance of the surfaces in the scene b) the stereo images obtained at the same time ensuring that the illumination of the scene is equal in both images. It will be assumed that the aperture of cameras used to obtain the cameras is equal or the differences can be compensated for.

It is well known that color histograms of an object are invariant to translation and rotation along the viewing axis and change only slowly under change of angle of view, change in scale, and occlusions [59]. In the case of stereo vision we need to be concerned about translation, the change of angle of view and occlusions. Therefore, a well known color segmentation algorithm by Ohlander et al. [51] was chosen.

Ohlander's colour segmentation algorithm used histograms obtained from multiple models of the colour space. A criteria was developed in order to pick model that has the most distinctive peak. Two thresholds are then determined on each side of the peak. The images is then thresholded by classifying pixels that have values between the two threshold values. This classification produces regions that belong to the found peak and regions that do not. The algorithm was recursively applied to each one of the new regions. The selected peak for each new region is selected depending on the model in which a peak was found. Therefore the performance of the algorithm is directed by the data. The segmentation stopped when no significant peaks in the histograms could be found.

C.0.1 Modification to Ohlander's algorithm

In order to perform correct matching it was important to identify regions in both stereo images that correspond to the same surface in the scene. This was achieved by coordinating the performance of the algorithm on both images. The segmentation of both stereo images was done simultaneously.

In the first iteration of the segmentation algorithm histograms for both images were determined. The best peaks for each image were found. In some cases the color models in which the peaks were found were not the same. Therefore, the segmentation of the images was not appropriate for performing matching between regions. The algorithm was therefore forced to pick one thresholding colour model. The best peaks from the two images were then compared using Ohlander's criteria. The thresholding range of the better peak was then set for both images. The thresholding was done on both images using the same thresholding range.

After thresholding, newly formed regions in each image were matched to their corresponding regions in the other image. The process was then repeated by coordinating the threshold values between the matched regions.

Appendix D

Java Implementation

In order to test the theory described in this thesis I developed an elaborate software system. The software system was designed to accept two stereo images, segment them into regions and establish correspondence between segments. The system encoded stereo constraints outlined in chapter 3 and occlusion failure constraints in chapter 4. The system produced matches between regions, and the associated constraints that were satisfied in order to establish the correspondence. In some cases, such as full occlusions, the the system would not be able to establish a match, however it would present evidence for the lack of a match. In cases where the correspondence could not be established, the system reported on possible locations of the missing surface. Based on the geometry of the stereo camera setup the matches were then interpreted in terms of three dimensional positions of surfaces.

Aside from accepting images and producing matches between regions, the system was also designed to produce an interactive system that allows the user to create an elaborate scene with multiple occlusions. The system would then create images that correspond to the views from two stereo cameras. These stereo images were then analyzed by the software system and matches were established. By pointing and clicking the user was able to inspect characteristics of regions, see the matching regions, set the constraints under which the matching occurs and view the explanation for matches.

The software system was implemented in the Java programming language [22] in order to allow easy access to the users over the Internet. A demonstration of the system can be fount on the Internet at: http://www.cs.ubc.ca/spider/tucakov/mscthesis.html

D.0.2 Scene Editing

In order to test the performance of the system easily and exhaustively a scene editor was developed. The scene editor allowed the user to place a number of surfaces in the scene. The surfaces were such that they did not violate the constraints imposed by this thesis. That is, the surfaces were frontal parallel to the optical axis of the cameras. Further, all surfaces were of constant color. It was assumed that there existed diffuse lighting.

Creating a scene involved defining the shape of surfaces in the scene and placing them in the virtual space in front of the cameras. Using a graphical user interface the user would define a two dimensional shape of the surface and define its color. Figure D.1 shows the user interface used for defining the shape and color of the surface.

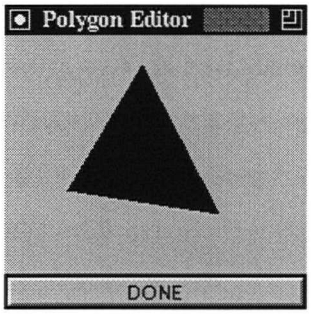
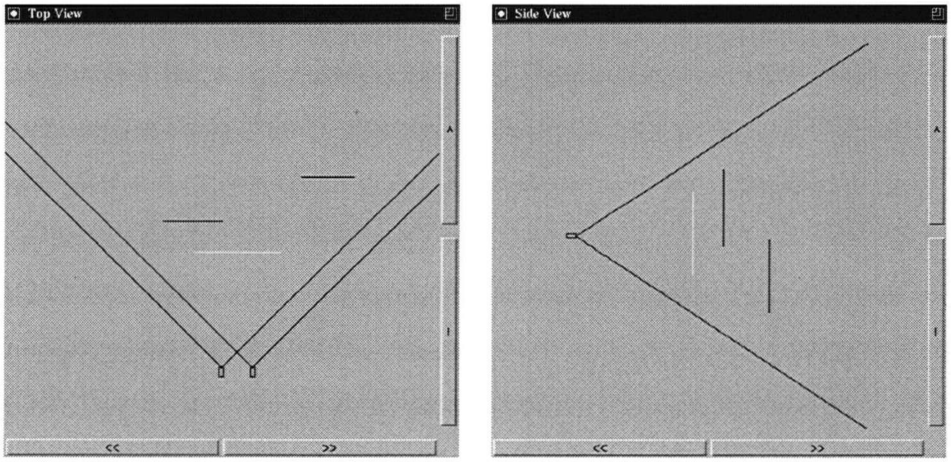


Figure D.1: Editing the shape of the surface

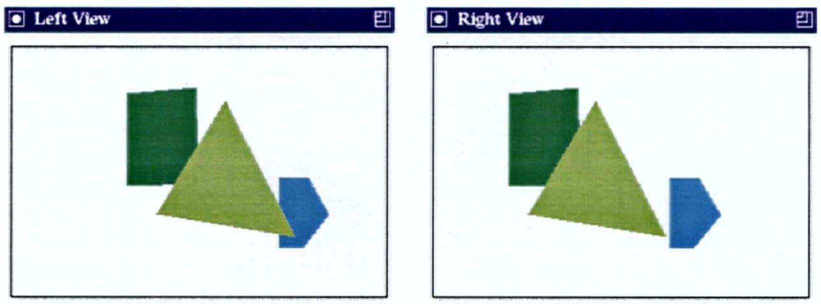
Once a polygon was defined the user could place it in the scene by using a graphical user interface that depicted the top and side view of the scene. This user interface can be seen in Figure D.2; the rectangles represent the cameras and the lines coming out of cameras represent the field of view of each camera. The parallel lines in the field of view of the cameras represent the position of the user defined surfaces. Lines coming out of the cameras represent the boundaries of the field of view of the cameras.



top view side view

Figure D.2: Interface for surface positioning

The user could select a surfaces and position it anywhere in the scene. While the position of the surface is edited the system generated views from the virtual stereo cameras. This enabled the user to inspect the generated stereo images while editing the scene. Figure D.3 shows the stereo view of the scene.



left camera view right camera view

Figure D.3: View from the two virtual cameras

Once the scene was finished the user would request scene editor to create image files that are then passed onto the stereo algorithm.

D.0.3 Image Segmentation and Region Viewing

Images created by the scene editor or obtained from stereo cameras can be passed on to the stereo algorithm. The first step of the stereo algorithm is to segment out the images into regions. This is done by loading either grey-scale or RGB images and using a segmentation algorithm similar to the recursive region splitting by Ohlander et al. [47]. The details of the segmentation algorithm are described in Appendix C. The system allows the user to select between manual or automatic segmentation. Automatic segmentation works reliable on synthetic images, however real images in some cases need a user intervention. Segmentation is assumed to be correct therefore segmentation by the user is acceptable.

Once the segmentation is done the results are displayed to the user. By pointing and clicking the user can now select a region and obtain information about the characteristics of that region. Figure D.4 shows the user interface that displays both the original images, in the first row, and the corresponding segments in the second row. The colours or regions in the segmented images is selected such that there are no two regions of same colour.

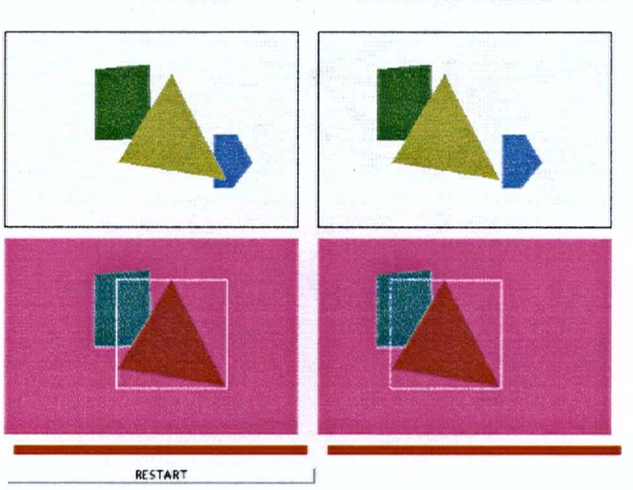


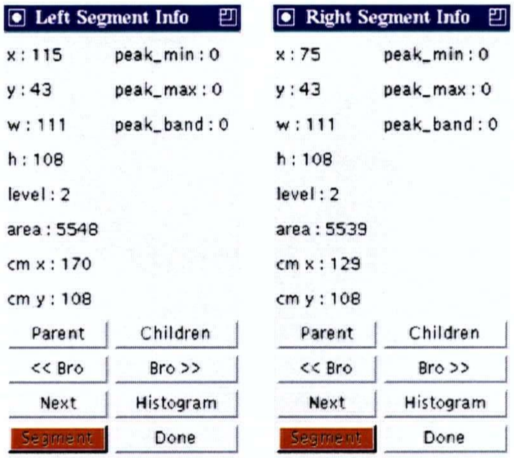
Figure D.4: Displaying the segmentation results

When the user clicks on a region a white bounding box is drawn around the region to indicate that it is selected. The selected regions can then be inspected by using a user interface that displays values of various properties of the region. Figure D.5 shows the information that corresponds to the regions selected in the Figure D.4.

Being able to easily access the information about the regions can greatly ease the debugging of code.

D.0.4 Viewing Matches

Once the images are segmented out into regions the user can request the system to produce matches between the images. The system performs matching as described in the previous chapters. The results of matching are displayed in terms of a matrix. In figure D.6 we see results of matching using strict stereo constraints. The rows of the matrix correspond to regions in the left image and columns correspond to the regions in the right image. By clicking on the cells in the matrix the user identifies matches between the regions. Selecting a cell in the match matrix also selects regions in the segmented images allowing the user to inspect which regions the cell corresponds to.



left region right region

Figure D.5: Information about the selected regions

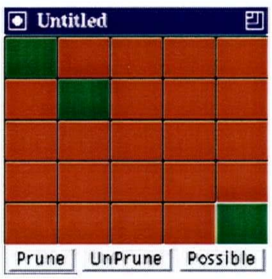


Figure D.6: Region Match Matrix, rows correspond to regions in the left image and columns correspond to regions in the right image

The matching process may determine that a match between two regions can be established since a certain kind of occlusion was determined, say partial occlusion. The user can request from the system to display the evidence that partial occlusion is taking place. In this case the system would numerically display values of region characteristics that are relevant to the weak stereo constraints, such as the comparison of the position of the bounding boxes and the average color of the region. The system would also display the partial contour match between the regions and it would display the missing part that is lost due to the occlusion.

Appendix E

Complexity Analysis

The performance of the algorithm outlined in this thesis depends heavily on the contents of the images. The number of regions segmented, number of occlusions, kinds of occlusions and the shape and size of regions all influence the efficiency of the algorithm.

In the complexity analysis I will assume that the image is already segmented out and that the properties of the regions are established.

E.1 Overall algorithms complexity

In this section I outline the complexity of establishing matches between all regions in the stereo images.

E.1.1 No occlusions

The best case scenario is when the matching is done on a scene with no occlusions. In this case it is sufficient to check the strict stereo constraints. Therefore the complexity of the algorithm is:

$$O(n_l n_r c)$$

where n_l and n_r are the number of regions in the left and and right images c is a constant amount of processing required to check the strict stereo constraint. Since c is a constant and $n_l \simeq n_r$ then the complexity is:

$$O(n^2)$$

where $n = max(n_l, n_r)$.

E.1.2 With occlusions

If there are occlusions but there are no multiple occlusions, the algorithm passes through the match matrix once to establish strict stereo matches and once to identify the kind of occlusion. Therefore the complexity is:

$$O(n^2) + n^2 C_{occlusions}$$

where $C_{occlusions}$ is the complexity of identifying the occlusion type for a pair of regions, and $n = max(n_l, n_r)$.

If there are multiple occlusions then the algorithm does one pass for strict stereo constraints, and multiple passes through the matrix in order to identify the kind of occlusion. Therefore complexity with multiple occlusions is:

$$O(n^2) + n^2 n^2 C_{occlusions} = O(n^2) + n^4 C_{occlusions} \tag{E.1}$$

where $C_{occlusions}$ is the complexity of determining the occlusion type between a pair of regions. The second n^2 term allows for the worst case in which each pass resolves only one occlusion.

E.1.3 Complexity of individual occlusion verifications

In this section I will present the complexity analysis of each individual occlusion constraint satisfaction. There exist two fundamental operations used in establishing the kind of occlusion:

- Partial contour matching
- Overlapping of regions based on the contour match.

E.1.4 Partial contour matching

Partial contour matching has to be done only once between pairs of regions that satisfy the weak stereo constraints. The complexity of partial contour matching between two regions depends on the perimeter length of the regions:

$$O(L^l L^r) = O(L^2)$$

where L^l and L^r the perimeter of the region in the left and right image respectively, and $L = max(L^l, L^r)$.

E.1.5 Overlapping

The complexity of overlapping between two regions, given a contour match, is a function of the area of the regions:

$$O(2(A^l + A^r)) = O(4A) = O(A)$$

where A^l and A^r are the area of the left and right regions, respectively, that are being overlapped, and $A = max(A^l, A^r)$.

E.1.6 Partial Occlusions

In order to solve if match between two regions can be established based on partial occlusion constraint, we need to perform partial contour matching, and overlapping based on the contour matches.

$$O(L^2) + n_{contours}O(A)$$

where L is the perimeter of the larger region, A is the area of the larger region, and $n_{contours}$ is the number of contour matches. The number of contour matches can not be greater than the perimeter of the larger region, therefore the complexity can be rewritten as:

$$O(L^2) + LO(A)$$

$$O(L(L + A))$$

E.1.7 Split Occlusions

In order to establish split occlusions it is necessary to identify all regions that satisfy the weak stereo constraint. Then it is necessary to identify regions that have contour matches that preserve the disparity. Finally an overlap step is performed to identify the split occlusion.

$$O(n) + nO(L^2) + num_{contours}nr + nO(A)$$

where n is the maximum number of regions, L is the maximum perimeter of a region, $n_{contours}$ the number of contours form the contour matching procedure, r is the stereo search range and A is the maximum area of a region. If $n_{contours}$ is replaced with L we get the formulation, and since r is a constant:

$$O(n) + nO(L^2) + Ln + nO(A)$$

$$O(n(1 + L^2 + L + A))$$

$$O(n(L^2 + A))$$

E.1.8 Full Occlusion

In order to establish full occlusions we need to slide and compare the region for the distance of the stereo range. Therefore the complexity is:

$$O(rA) = O(A)$$

where r is the value of the disparity range (a constant) and A is the maximum area of a region.

E.1.9 Multiple Region Matching

Multiple region matching starts by establishing a root match which requires establishing a partial contour match (partial contour match is done only once). Based on this contour match an overlap is done and new regions are added to the aggregate. The overlapping is repeated for as long as there are regions that can be added to the aggregate. The complexity of multiple matching is:

$$O(L^2) + n_{contours}nO(A)$$

where L is the maximum perimeter of a region, $n_{contours}$ is the resulting number of contours, n is the maximum number of regions and A is the maximum area of a region. If $num_{contours}$ is replaced with L we get:

$$O(L^2 + LnA)$$

E.1.10 Doing all occlusions

In order to establish the complexity of checking for all kinds of occlusions types we need to add up the cost of each individual test:

$$O(L(L+A)) + O(n(L^2+A)) + O(A) + O(L^2+LnA)$$

$$O(L^2 + LA + nL^2 + nA + A + L^2 + LnA)$$

$$O(L^2(2+n) + A(L+n+1+Ln))$$

$$O(L^2n + A(L+n+Ln))$$

As upper bounds, if there are P pixels in the image, for the maximum area A we can use P, and for the number of regions we can use P and for maximum perimeter we can use P, then:

$$O(P^2P + P(P + P + PP))$$

$$O(P^3 + P(P + P + P^2))$$

$$O(2P^3 + 2P^2)$$

$$O(P^3)$$

where P is the number of pixels in the image.

Finally, substituting in the above formula into the overall expression E.1 we get:

$$O(n^2) + n^4 C_{occlusions}$$

$$O(P^2) + P^4 O(P^3)$$

$$O(P^2 + P^7 + P^6)$$

$$O(P^7)$$

where P is the number of pixels in an image, and number of regions n is replaced with P.

It should be noted that $O(P^7)$ is the upper bound on the complexity of the algorithm. This is however not a tight bound on the complexity. Empirical results show that the average case of the algorithm is in the order of $O(P^2)$ to $O(P^3)$.

Appendix F

Segmentation of Unseparable Surfaces

This appendix explores the solutions to the matching problem when segmentation does not work perfectly. More specifically, we analyze the case where surfaces in the scene do not have a clear boundary in the image they produce. Therefore one region in the image may represent more than one surface in the scene. Figure F.1 illustrates an example where in the right view the surfaces are not separable.

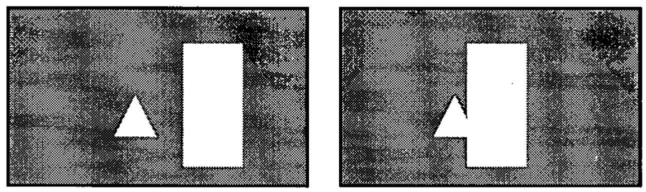


Figure F.1: Two surfaces that are not separable in the better view

The main cue in discovering unseparable surfaces will be partial contour matches and the notion of occlusions. Therefore, we will allow occlusions in our scenes. We will still require that a region in the image belongs to one or more full surfaces. That is, a single surface will not be imaged is terms of separate regions. We will also disallow perspective distortion, shading, texture and so on.

The problem of non-separable surfaces can be separated into two subproblems. The easier one, when physical separation exists in the better view and the more difficult problem, where separation does not exist in neither views.

F.1 Separable in the better surface

When the surfaces are separable in the better view the problem is relatively straight forward. The actual shape of the surfaces can be determined from the better view and using the partial contour matches it is possible to determine the surface configuration. More precisely, in order to establish a match based on unseparable surface failure we need to show that the following constraints hold:

- **Weak Stereo Constraints** are not violated

- **Establish partial contour matches** between the regions that are physically separated and the region in which the surfaces are not separable.

- **Show that the contours do not overlap**

- **Overlap possible;** it is important to show that given the partial contour matches the aggregate due to unseparable surfaces can be produced.

- **Only Explanation;** the unseparable surfaces are the only explanation.

F.1.1 Example of separable better view

Consider the scene in the figure F.1. In the left image we can observe two separated regions, one that correspond a triangle and one that corresponds to a rectangle. On the other hand, in the right image we observe only one region that is a aggregate of the triangle and the rectangle.

We can show that the weak stereo constraints hold for match between the rectangle and the aggregate, as well as the match between the rectangle and the aggregate. We can also find the partial contour matches between the regions in the left image and the region in the right image. Figure F.2 shows the contours that can be established between regions. Further, we observe that the contours do not overlap.

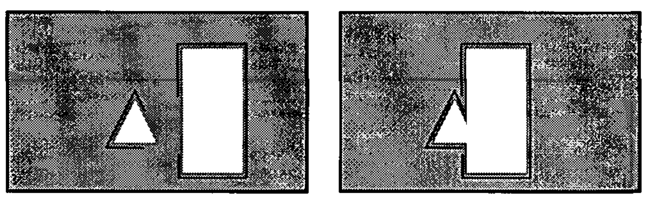


Figure F.2: Contour matches between the non-separable surfaces

We can now overlap the regions from the left view on top of the aggregate in the right view. We find that aggregate is fully overlapped showing that it indeed could be composed of these regions. Finally, we compute the disparities based on the partial contour matches. The disparities allow us to determine the distance of the surfaces. In this example the rectangle has a greater disparity, therefore it is closer to the cameras. Based on this information we can introduce a split of the aggregate into two regions as shown in figure F.3.

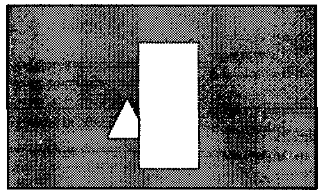


Figure F.3: The perceived edge between the two surfaces

F.2 Scenes with both views Non-Separable

Scenes in which both views have aggregates of non-separable surfaces are difficult to resolve. In some cases however it is possible to recover some information.

F.2.1 Example of Non-Separable views

Figure F.4 is an example of a scene in which two surfaces can not be separated in both views. In this case we can determine that there are two surfaces, we can determine their distance from the cameras. We can also determine the constraints on the shape of the surfaces.

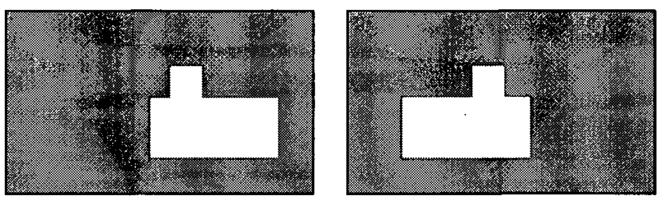


Figure F.4: Two unseparable surfaces in both views

In order to interpret this scene we first establish all partial contour matches between the two views. As shown in Figure F.5a, we can establish two partial contour matches, represented by dotted and dashed lines. By overlapping regions based on the contour matches we can establish the constraints on the shape of each surface. Figure F.5b shows the shape constraints corresponding to the top surface, and Figure F.6c shows the shape constraints corresponding to the bottom surface.

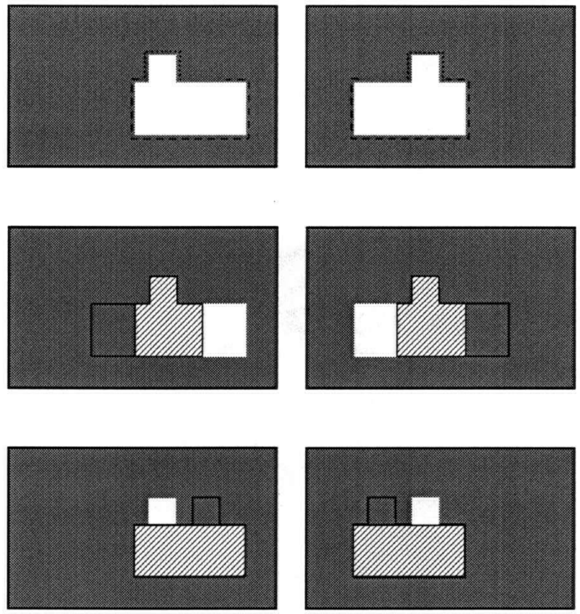


Figure F.5: Matching partial contours

Now we can establish parts of the views that can not belong to the surfaces. In Figure F.6a we identify the parts of the views that can not belong to the lower surface. This information is obtained from parts of the view that do not belong to the top surface, based on constraints outlined in Figure F.6b. Similarly we can identify the parts of the view that can not belong to the bottom surface. By combining the two we obtain areas of the view that must belong to the one of the surfaces, as well as areas that can belong to either, see Figure F.6c.

The areas of the view that can belong to either surface leave the actual shape of the surface ambiguous. While the shapes of the surfaces are not fully defined there exists an additional constraint. This constraint imposes that projection of every ambiguous pixel in the view must intersect at least one of the surfaces.

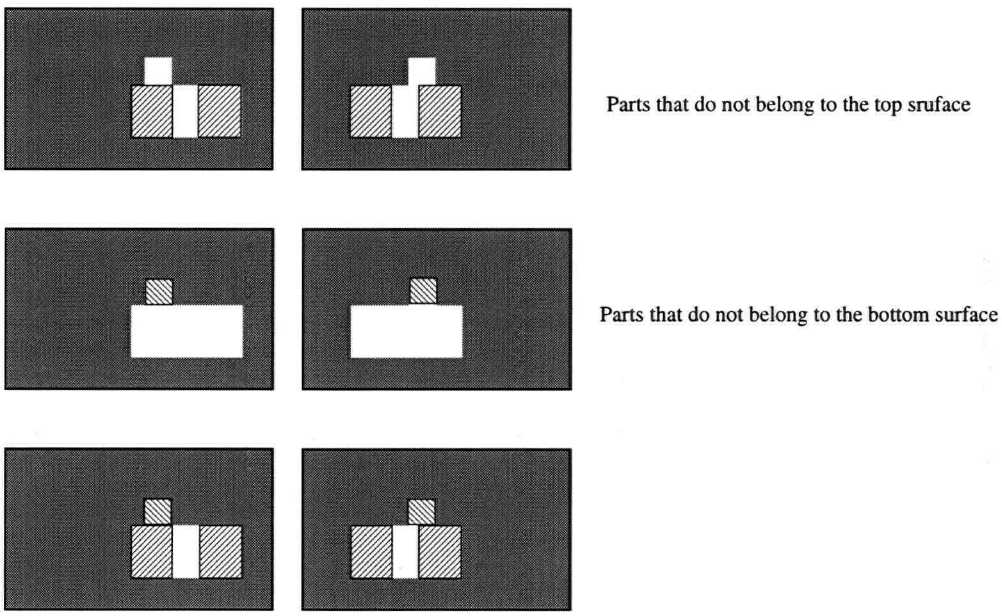


Figure F.6: Finding the virtual edges

Bibliography

[1] E. Adelson. Layered representations for vision and video. In *Representation of Visual Scenes*, Cambridge, MA, June, 1995.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[3] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1:107–131, 1987.

[4] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proc. 7th Int. Joint Conf. on Artificial Intelligence*, pages 631–636, Vancouver, 1981.

[5] D. H. Ballard and C. M. Brown. *Computer vision.* Prentice Hall, Englewood Cliffs, NJ, 1982.

[6] S. T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3:17–32, 1989.

[7] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:333–340, 1980.

[8] P.N. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *Proc. 4th International Conference on Computer Vision*, pages 431–438, 1993.

[9] D.N Bhat and S.K. Nayar. Ordinal measures for visual correspondence. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1996*, pages 351–357, June 1996.

[10] J. D. Boissonnat, O. D. Faugeras, and E. L. Bras-Mehlman. Representing stereo data with the Delaunay triangulation. In *Proc. IEEE Conf. on Robotics and Automation, 1988.*

[11] K. L. Boyer and A. C. Kak. Structural stereopsis for 3-d vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):144–166, March 1988.

[12] Y. Boykov, O. Veksler, and R. Zabih. Disparity component matching for visual correspondence. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997*, pages 470–475, Puerto Rico, 1997.

[13] C.R. Brice and C.L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3):205–226, 1970.

[14] H. Bulthoff, J. J. Little, and T. Poggio. A parallel algorithm for real-time computation of optical flow. *Nature*, 337:549–553, February 1989.

[15] C. Chang, S. Chatterjee, and P.R. Kube. On an analysis of static occlusion in stereo vision. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1991*, pages 722–723, 1991.

[16] J. J. Clark and P. D. Lawrence. Determination of disparity gradients with binocular diffrequency measurements. TR-4, UBC Dept. of Electrical Engineering, Joint Robotics and Teleoperator Laboratory, Vancouver, BC, 1985.

[17] P.R. Cooper, L.A. Birnbaum, and M.E. Brand. Causal scene understanding. *Computer Vision and Image Understanding*, 62(2):215–231, September 1995.

[18] I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs. A maximum-likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[19] T. Darrell and A.P. Pentland. Robust estimation of a multi-layered motion representation. In *Workshop on Visual Motion*, pages 173–178, 1991.

[20] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

[21] J. Feldman and Y. Yakimovsky. Decision theory and artificial intelligence: I. a semantics based region analyzer. *Artificial Intelligence*, 5:349–371, 1974.

[22] D. Flanagan. *Java in a Nutshell.* OReilly and Associates, Inc., 1996.

[23] M.M. Fleck. A topological stereo matcher. *International Journal of Computer Vision*, 6(3), August 1991.

[24] D.J. Fleet, A.D. Jepson, and M.R.M. Jenkin. Phase-based disparity measurement. *Computer Vision Graphics and Image Processing*, 53(2):198–210, March 1991.

[25] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. In *Machine Vision and Applications*, pages 35–49, 1993.

[26] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14:211–226, 1995.

[27] M. A. Gennert. Brightness based stereo matching. In *Proc. 2nd International Conference on Computer Vision*, 1988.

[28] Michael A. Gennert. *A Computational Framework for Understanding Problems in Stereo Vision.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[29] Walter Gillett. Issues in parallel stereo matching. Master's thesis, Massachusetts Institute of Technology, 1988.

[30] W.E.L. Grimson. Computational experiments with a feature based stereo algorithm. AI-Memo-762, MIT AI Laboratory, Cambridge, MA, 1984.

[31] B.K.P. Horn. *Robot Vision.* McGraw Hill, 1986.

[32] S.L. Horowitz and T. Pavlidis. Picture segmentation by a directed split and merge procedure. In *International Conference on Pattern Recognition*, pages 424–433, 1974.

[33] A.D. Jepson and M.R.M. Jenkin. The fast computation of disparity from phase differences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1989*, pages 398–403, San Diego, 1989.

[34] M. Kanade, T. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. In *Image Understanding Workshop*, 1990.

[35] H. Lee and W. Lei. Region matching and depth finding for 3d objects in stereo areal photographs. *Pattern Recognition*, 23(1/2):81–94, 1990.

[36] Martin D. Levine, D. A. O'Handley, and G. M. Yagi. Computer determination of depth maps. *Computer Graphics and Image Processing*, 2(4):131–150, October 1973.

[37] J. J. Little and W. E. Gillett. Direct evidence of occlusion in stereo and motion. *Image and Vision Computing*, 8(4):328–340, November 1990.

[38] J.J. Little. Accurate early detection of discontinuities. In *Vision Interface 92*, pages 97–102, May, 1992.

[39] A. K. Mackworth. Constraint satisfaction. In *The Encyclopedia of AI*. John Wiley, New York, second edition, 1991.

[40] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 15 October 1976.

[41] D.P. McReynolds and D.G. Lowe. Rigidity checking of 3d point correspondences under perspective projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1174–1185, 1996.

[42] G. Medioni and R. Nevatia. Matching images using linear features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:675–685, 1984.

[43] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th Int. Joint Conf. on Artificial Intelligence*, page p. 584, Cambridge, MA, 1977.

[44] K. Mori, M. Kidode, and H. Asada. An iterative prediction and correction method for automatic stereocomparison. *Computer Graphics and Image Processing*, 2:393–401, 1973.

[45] R. Nevatia. Depth measurement by motion stereo. *Computer Graphics and Image Processing*, 5:203–214, 1976.

[46] H.K. Nishihara and T Poggio. Stereo vision for robotics. In *ISRR83 Conference*, Bretton Woods, HN, 1983.

[47] R. Ohlander, K. Price, and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313–333, 1978.

[48] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:139–154, 1985.

[49] F. Preparata and M. Shamos. *Computational Geometry.* Springer-Verlag, 1985.

[50] J.M.S. Prewitt. Object enhancement and extraction. In *Picture Processing and Psychopictorics*, pages 75–149, 1970.

[51] Keith Price R. Ohlander and D.R. Reddy. Picture segmentation using a recursive splitting method. *Computer Graphics and Image Processing*, 8:313–333, 1978.

[52] S. Ranade and J.M.S. Prewitt. A comparison of some segmentation algorithms for cytology. In *International Conference on Pattern Recognition*, pages 561–564, 1980.

[53] S. Randriamasy and A. Gagalowicz. Region based stereo mathing oriented image processing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1991*, pages 736–737, Lahaina - Maui, Hawaii, 1991.

[54] S. Randriamasy and A. Gagalowicz. Direct cooperation between region-based stereo matching and top-down segmentation. In *Proceedings of ICOM'93*, 1993.

[55] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems, Man and Cybernetics*, 6(6):420–433, June 1976.

[56] S.A. Shafer and T. Kanade. Recursive region segmentation by analysis of histograms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1166–1171, 1982.

[57] B. Shahraray and M. K. Brown. Robust depth estimation from optical flow. In *Proc. 2nd International Conference on Computer Vision*, 1988.

[58] L. A. Spacek. Edge detection and motion detection. *Image and Vision Computing*, 4:43–56, 1986.

[59] M. Swain and D.H. Ballard. Color indexing. *International Jurnal of Computer Vision*, 7(1):11–32, 1991.

[60] R. Szeliski. *Bayesian modeling of uncertainty in low-level vision.* Kluwer Academic Press, Norwell Ma, 1989.

[61] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:129–139, 1986.

[62] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.

[63] V. Tucakov and D.G. Lowe. Temporally coherent stereo: Improving performance through knowledge of motion. In *Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation*, pages 1999–2006, Albuquerque, NM, 1997.

[64] V. Tucakov, M. Sahota, D. Murray, A. Mackworth, J. Little, S. Kingdon, C. Jennings, and R. Barman. Spinoza: A stereoscopic visually guided mobile robot. In *Proceedings of the Thirteenth Annual Hawaii International Conference of System Sciences*, volume 5, pages 188–197, 1997.

[65] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Trans. Image Processing*, 3(5):625–638, September 1994.

[66] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144, 1987.

[67] Y.J. Zhang. A survey of evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, August 1996.