Flexible and Robust Image Matching by Region-Based Approach

THESIS

Flexible and Robust Image Matching by Region-Based Approach

Caihua Wang

Graduate School of Electronic Science and Technology
Shizuoka University, Japan
July 1996

To

my Parents and Xiaomei

Acknowledgments

I would like to express my gratitude to my supervisor Prof. Keiichi Abe for his encouragement, guidance and valuable discussion during the my doctoral course, without which this thesis would not be possible. I would also like to express my gratitude to Prof. Ohtsubo, Prof. Nakatani and Porf. Watanabe for their helpful comments and valuable advice that greatly appreciated in the preparation of this thesis.

I wish to acknowledge to Japanese Ministry of Science and Education, whose generous scholarship during the academic years 1993.1 – 1886.10 greatly supported my study, research as well as my living in Japan.

I am also grateful to my fellow students in my laboratory for their help on many occasions. Especially I would like to thank Andreas Held for much kindly help and valuable advice he had given to me when he was a senior schoolmate in my laboratory, and Takahiro Sugiyama for his stimulating discussions and much appreciate technical support. I would also like to thank the staff of the Department of Computer Science, Information Faculty, Shizuoka University, for their help and support on many occasions.

Last but not least, my thanks go to my parents and my sisters for their understanding and encouragement during the long term when I am studying in foreign country. Special thanks go to my wife, Xiaomei, for her support and understanding during the three years doctoral course.

Abstract

Image matching, that is, establishing correspondence between the selected primitives in two or more images so that the matched primitives are projected from the same physical elements in the three dimensional world, is a basic and critical problem in many applications of computer vision. In the present thesis, a region-based approach, that is, an approach which uses regions as matching primitives, is proposed to match two images flexibly, robustly and efficiently. Such flexible and robust matching technique, which can obtain appropriate matching results for images in presence of noise, deformation, occlusion and inconsistency in feature exaction in two images, is essential for real applications of computer vision.

According to the difference in the purposes and the problem settings in various applications, image matching problems are classified into three types: 1) stereo matching for obtaining the disparity map by fusing the pixels in the two stereo images along epipolar lines; 2) image correspondence or registration of two images of the same scene but taken from different viewpoints where the relative positions are not constrained, nor known; and 3) image matching for object identification, location or recognition. In this work, the image matching problems in all the cases are investigated from the region-based matching paradigm, and for each type of the problem, an algorithm of region-based approach is proposed.

In stereo image matching, instead of matching each individual pixels in the images, the patches, which correspond to the regions projected from continuous surfaces so that they can be matched continuously between two image, is the regarded as matching primitives. To find the matching of such patches, we divide the range of the depth in the scene into small intervals, and find continuous matchings of segments on the epipolar line in each small interval. Then the continuously matched segments are merged into patches, by considering the continuity between epipolar lines. The

matchings of patches obtained in individual intervals, which may partially conflict or overlap with each other, are integrated to obtain the matchings of patches in the whole scene by considering the consistency among the matchings of patches.

For image correspondence in a general case, we propose a robust region matching method which utilizes both local region structure similarity and global consistency of the region matchings. The method consists of two stages. In the first stage, local ternary relations among regions, that is, three triangles composed of the centers of gravity of the region in question and its three nearest neighbors, are used together with chromatic features of regions to establish region correspondences. In the second stage, the optical flow generated by the camera movement, which can be estimated using all obtained correspondences, is used as the global consistency constraint to validate individual correspondences. The ones which do not obey the optical flow are detected as mistakes and then corrected using the estimated optical flow.

A region-based syntactic approach are also attempted for the region matching problem. The regions and their relations in an image are represented by a Region Adjacency Graph(RAG), in which the nodes correspond to the regions, the edges between nodes represent the region adjacency, and both nodes and edges are characterized by some features of them, such as the average color for a region, length of boundary between two regions. The problem of matching the regions projected in two images is formulated into the problem of finding the optimal inexact graph matching between two RAGs. It is noticed that a RAG is an attributed *planar* graph, and it has some specific properties. Therefore, by utilizing these properties, we can construct an efficient algorithm to find the optimal inexact matching between two RAGs. This syntactic approach can be extended for object identification, location and recognition, where matchings between the models of objects and an input image should be attempted.

Contents

CHAPTER 1

Introduction

1.1 Historical Review of Image Matching

1.1.1 Image Matching

Image matching, that is, establishing correspondence for the selected primitives in two or more images so that the matched primitives are projected from the same physical elements in three dimensional world, is one of the oldest tasks in computer vision research. This is because many basic tasks in computer vision involve image matching as an unavoidable and critical problem. For example, in the stereopsis for constructing three dimensional information of the scene, the depth information of each primitive in an image can be computed by triangulation only after its correspondence to that in the other image was established by matching two stereo images. Other examples include time-varying images analysis where the images taken at different time are matched to detect the changes in the scene; image retrieval searching in image database for an image of a particular scene or containing particular objects; image identification to judge whether two images are taken from the same scene, object location which determines the location of an object by matching a sample image of the object with a part of the image which may contain the object, and so on.

Image matching is also one of the most active fields which attracts enormous research efforts in computer vision nowadays, because it is still an open problem of essential importance, which have not been solved completely yet. Image matching is made difficult by several complications in that problem. First, it is a reversal problem of perspective projection and therefore ill-conditioned. That is, a point in an image may be projected possibly from anyone of the points along a line in the scene which passes through the image point and the center of the lens of camera, but in another image, all those points in the scene will project to distinct points. It means that for a given point in one image, there are often many possible matchers of it in another image. Second, due to the inaccuracy of measurements and the effects of variation in illumination between two images, even if the primitives in two images are projected from the same physical element in the scene, they may have features different in some way. Third, due to occlusion, there are almost certainly some regions of half-occluded points which appear in only one image and, consequently, have no match in the other image.

According to the difference of the purposes and the problem settings in various applications, image matching problems are classified into three types: 1) stereo matching to obtain the disparity map by fusing the pixels in the two stereo images, where a priori knowledge about the camera settings is available so the epipolar line constraint, which can be derived from camera settings, can be used to restricts the possible matches of a point in one image to those lying in the corresponding epipolar line in the other image. 2) image correspondence or registration of two images of the same scene but taken from the different viewpoints which are neither constrained nor known; and 3) object identification, location or recognition by matching an image of an object with a sample image or a model of the object or with the image taken from another viewpoint.

There are many approaches[1-35] proposed to deal with image matching problem, in which the majority are proposed for stereo matching[1-26], and only a small

part of them deals with image correspondence in general case[27-35]. There are also a few methods[36-38] of matching a sample image of an object with a part of the input image proposed for object location and recognition. Matching of a model of an object with an image[38-49] is deeply depending on the model used for representation of the object, that is not concerned substantially in this thesis.

1.1.2 Stereo Matching

All the algorithm proposed for stereo matching can be characterized from three aspects: the constraints they impose on the images; the primitives they select for matching; and the matching strategy they employ.

Constraints: Because even after the epipolar constraint was applied to restrict the matchers of a given primitive to that lying on the corresponding epipolar line in the other image, there are always too many candidates of matchers for it, many algorithms proposed previously impose also various constraints on images to simplify the huge search space for matching. Kanade and Ohta[12], Bensrhair[11] and Li[18] employed the ordering constraints along corresponding epipolar lines so that the optimal matchings of primitives between the corresponding epipolar lines in two images can be found efficiently by dynamic programming. The continuity of the disparity constraint, which is directly related to smoothness of the scene surfaces, is used in most of the existing stereo matching techniques[1-26] for suppressing some impossible matchings. The figural continuity constraint is introduced by Grimson[10] to utilize disparity continuity along the boundaries of the surfaces in the scene.

Matching Primitives: Selection of primitives for matching has a special importance in stereo matching, because using more discriminating primitives which can be distinguished from each other more easily can restrict the search for matching a given primitive to fewer candidates which have similar features. Hannah[5], Mori[6], Kanade[7] and some other method [3,4] utilized SSD (sum of squared difference)

measurement to match every pixel by comparing the density distributions in a local neighborhoods of a pixel in one image with that in a corresponding neighborhoods of a pixel in the other image, instead of by comparing the gray values of the two pixels only. The interesting points matching methods, which extract some points with specific intensity pattern in their neighborhood from images as matching primitives, also compute the correlation between the neighborhoods of two corresponding pixels in two images to measure the matching similarity of them. Marr[9] suggested that the zero-crossings should be used as matching primitives upon the psychophysical evidence on human vision system, and Grimson[10] gave an implementation of Marr-Pioggio theory. Connected linear features such as line segments and edge segments are preferred by many proposed techniques[18], because those features are usually more stable to photometric variations and are less in number, that will lead to more reliable and efficient matching algorithms. A few methods by Marapane[16] and Lee[20] used regions as matching primitives.

Matching Strategies: In addition to imposing constraints on images and selecting discriminating features for matching, several control strategies have been employed in many stereo matching techniques to reduce further the matching ambiguity and enhance the matching efficiency. In coarse-to-fine strategy used by Marr[9], Grimson[10], Vleeschauwer[26] and others[20], the matchings obtained in a coarser scale of resolution is used to guide and limit the search for the matching of primitives at finer scales. Marapane[16] and Hoff[24] employed a multi-primitive hierarchal matching strategy which used more than one abstract levels of primitives, such as pixels, edges and regions, simultaneously. In structural stereo approaches proposed by Horand[19] and Boyer[25], the relational structure information between primitives is used to eliminate the ambiguity of matchings.

1.1.3 Image Correspondence in General Case

This problem is generalized from the stereo matching by removing the epipolar constraint so that we can deal with other applications such as image retrieval, scene identification, motion with large frame interval, and other tasks in which image correspondence is required but the relative positions of the viewpoints are unknown. In principle, all the feature-based stereo matching techniques can be extended to the general case, but the increase of the dimension of search space for matching and the ambiguity of the matching will be too severe for most of those approaches.

Contrary to those for stereo matching, there are only a small number of approaches proposed for image matching in the general case. Kahl[28] and Skea[29] used a control point matching approach to find the maximum matching of points, that is subjected to satisfy some geometric invariant properties in two images. Strickland[30] proposed an edge matching approach, in which the edges in a image are considered as a network where the cross points of edges are presented as nodes and the edge segments between the cross point are presented as branches. A relaxation scheme is used to find the matching of branches and nodes in two images with the maximal similarity between the structures of two networks of the edges in two images. Weng[35] proposed a hierarchal multi-primitive matching approach, which adopted local rigid movement constraint, that is, the displaces of the matched primitives must obey the movement of a rigid object locally.

Chen and Nakatani[33] proposed a region-based image matching approach using color feature of regions and the region adjacency relations. In their work, the region structure is represented by region tables, in which the first item contains a region which is concerned and the ith item contains a list of regions which are away from the concerned region in distance i (that is, they are separated from the concerned region by $i-1$ regions in between). The region correspondence is carried out matching the regions at similar levels in the region table.

1.1.4 Object Identification, Location and Recognition

Though this third type of image matching problem is not dealt with substantially in this thesis, a short review of it is in order.

The matching problem involved in object identification, location and recognition is essentially a problem of finding a correspondence between a model of the object and the image (or a part of it) in which the object is projected. There are many methods proposed for matching of models with images, where the type of model used for representation of the object will greatly influence the matching procedure. The majority of model matching methods falls in the following classes:

1. *Template matching.* Template matching[36-38] has been a traditional approach to two dimensional object identification, location and recognition, where the the object is represented by a sample image of it, called template. The most simple template matching program just slide the template over the image and find the place at which the template is matched with the image best. A parametric template matching approach are proposed in [38] to deal with not only the translations of the pattern in images, but also the rotation and scaling. Recently, deformable template matching[36] are proposed to deal with shape deformation caused by the projections from different viewpoints.

2. *Shape matching.* This approach are proposed for recognition of 2D objects or planar objects. In this case, the object is considered to be decisively described by the features of its contour, that is, its 2D shape. Matching the model of an object with the image is to finding the correspondence of the contour in the model to that extracted from the image[39-42].

3. *Local feature matching.* In this approach, the object model is described by a set of local features, which are always selected as point features such as corner or vertices. The same features are detected from the image, and then

the number of local features, which may be projected in the images from the object features under certain transformation, are counted. This is carried out by the technique called *geometric hashing*[43-49]. The transformation my be affine transformation[44], general transformation of translation, rotation and scaling[43], or perspective transformation[45].

4. *Structural matching.* An object is described by some features and the relational structure of the features. The problem of object recognition becomes the problem of matching two structures, one represent the model of the object and the other is extracted from the image. This problem is generally described as a so called consistent labelling problem in [64], and a solution of state space searching was given. There are also many graphic approaches[62,63,65-67] proposed for this problem, where the features and their relational structure of the object was represented as attributed graph and the then find a maximal inexact matching between two graphs.

5. *Global feature matching.* Murase[49] proposed a appearance learning methods which learns all the appearances of the objects, and each appearance (the image of the object taken from certain viewpoint) of the object is approximated by its eigenvalue. In the recognition stage, the eigenvalue of an input image of the object is first computed and then are compared with that of the appearances learned in the previous stage. The appearances which is closest to the input image are regarded as the corresponding appearances of the observed object.

1.2 Research Objective: Region Matching

Efficient and robust image matching technique for establishing correspondence between features from two perspective projected or temporarily and/or spatially shifted images is essentially important for real applications of computer vision. For such an

efficient and robust matching, regions are more appropriate as features than edges or points, because using regions as matching primitives has the advantages (1) that there are much fewer regions in an image than other primitives such as edges, (2) that regions possess more information (such as area, average color, shape and moment) which supplies higher discriminating capability of regions, and (3) that they are also more stable against occlusions than, for instance, edges[31]. These advantages enable a more efficient and reliable matching by using regions, and the result can be used as guiding information for matching of more detailed primitives, such as edges and corners, when it is necessary. The objective of this thesis is to describe a region-based approach proposed for efficient and robust image matching.

In the numerous literatures on image matching, only a few address the region-based techniques, and various deficiencies can be found in them. The earlier methods which only use region features (Marapane[31] and Trivedi[24], Fuh[32]) suffered from the matching ambiguity when there are similar regions in the images. Lee[20] computed the matching probability of a region and its corresponding candidates using region features, and then improved them by relaxation based on the pair of similar triangles in two images. However, the relation of similar triangle is neither stable nor discriminating, because the triangle of the centers of gravity of triple regions can hardly keep in similarity and, on the other hand, two triples of regions of two similar triangles may be unrelated. The recent method proposed by Chen[33] tried to use structural information of region adjacency relations, but neither the region adjacency relation is meaningful and stable for regions of separated objects, nor the region table used there is an inherent way to describe the relations of region structure similarity. Another problem of the proposed region-based techniques is that they can obtain only sparse disparity from region correspondence, when they are applied to stereopsis.

In this thesis, we will describe three region-based image matching methods. The first is a region-based part-to-whole stereo matching algorithm, in which the patches to be matched continuously between two images (that is, the regions which are

projected from the same continuous surfaces in the scene) are regarded as matching primitives. The second is a region-based technique for image correspondence in general case, which consists of two stages: the matching stage that establishes region correspondence using both features similarity and local structural similarity of regions, and the validating and correcting stage which utilizes the global consistency of the whole obtained correspondences to detect and correct some apparent mistakes in the derived local matching results. The third is a region-based syntactic approach to image matching problem, where the regions and their relations in an image are represented by a Region Adjacency Graph (RAG), and the problem of matching the regions in two images are formulated into the problem of finding the optimal inexact graph matching between two RAGs. This syntactic approach can be extended for object identification, location and recognition, where matchings between the models of objects and an input image should be attempted.

1.3 Outline of the Thesis

This thesis is organized as follows. In Chapter 2, the stereo matching problem is considered. Stereo matching is represented as a problem of finding between two images the matching of the regions which are projected form the continuous surfaces in the scene. Such a region is equivalent to a patch which can be matched continuously in two images. A part-to-whole approach is described in this chapter, which first finds patch matchings in each small interval of depth by dynamic programming with the continuity of matching being taken into account, and then integrates the matchings of patches in each individual interval to obtain the matchings of patches for the whole scene, by considering the consistency among the matchings.

In Chapter 3, a region matching method is proposed for image correspondence in general case where the epipolar constraint is not available. The method utilizes both local region structure similarity and global consistency of the region matchings

to suppress the matching ambiguity occurred when only the features of regions are used. The local ternary relations among regions, that is, three triangles composed of the centers of gravity of the region in question and its three nearest neighbors, are used to describe the local structure of regions. In Chapter 4, the optical flows generated by the camera movement are estimated from the whole of the region correspondences obtained by the method described in Chapter 3, under the assumption that the majority of them are guaranteed to be correct. The estimated optical flow, which should be followed by every correspondence of regions, is used as the global consistency constraint of region matchings to detect and correct apparently mistaken correspondences.

Chapter 5 gives a region-based syntactic representation of region matching between two images. The regions and their relations of the objects in an image are represented with a Region Adjacency Graph (RAG), in which the nodes correspond to the regions, the edges between nodes represents the region adjacency, and both nodes and edges are characterized by some features of them, such as the average color of region, length of adjacency of edge, etc. The problem of matching the regions in two images are formulated into the problem of finding the optimal inexact graph matching between two RAGs. In this chapter, the RAG is shown to be an attributed *planar* graph, and some of its properties are noticed. From these properties, we can obtain that an attributed RAG can be partitioned into biconnected components connected by articulation nodes as a tree, and within the biconnected components, the edges of each node must be matched with order reserving. Utilizing these properties, an efficient algorithm to find the optimal inexact matching between two RAGs is developed. In Chapter 6, we summarize what we attained, and discuss the remaining problems of the proposed methods as well as possible extensions of them for wider range of applications, including the extension of the region-based syntactic approach for the purpose of object location, identification and recognition.

CHAPTER 2

Stereo Matching: A Region-Based Part-to-Whole Approach

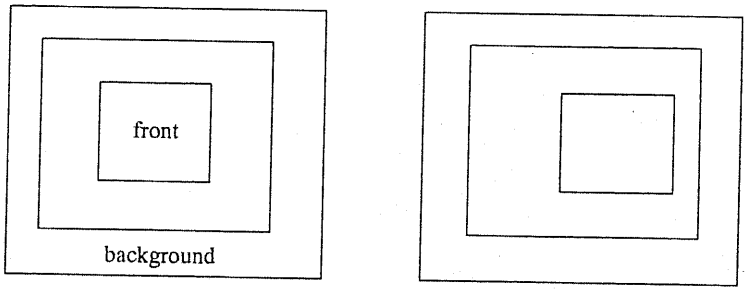
2.1 Introduction

For the past decades, a major portion of the research efforts in computer vision has been directed towards the study of three-dimensional structure of scenes using machine analysis of images. Stereo vision is essentially a process to construct three-dimensional structure of the scene which has two major advantages over other methods such as laser range finding. First, it is a passive method, which means that it does not alter the environment. Second, it is potentially better for high resolution of three dimensional description of moving objects or changing scenes in less than a thousandth of a second, whereas the current laser range finding technology usually requires objects to be relatively motionless.

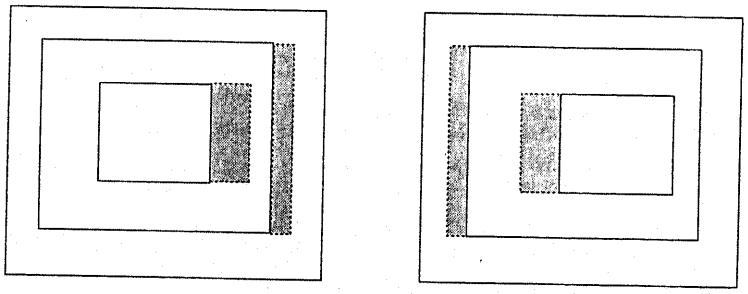
The central task in stereo vision is stereo matching, that is, finding correspondences of pixels in two images. Although big efforts have been made for stereo matching for decades, it is still an open problem due to its complexity. There are many methods proposed for stereo matching, and most of them fall into the two

categories: the feature-based approach[1,9-26] and the area-based approach[2-8].

In the feature-based approach, features such as regions, edges, corners, or zero crossings of Laplacian of Gaussian extracted from images are used as matching primitives. As the number of features is always much smaller than that of pixels, the feature-based matching can be carried out efficiently, but at the same time only with sparser disparity[1] information. Usually, interpolation is applied to compute the disparity for non-feature pixels, but it has a flaw. When a part of a homogeneous region (or nearly homogeneous region where no feature can be extracted) is occluded in another image, as shown as the shaded areas in Figure 2.1, the disparities of pixels between the edges can not be computed correctly just by interpolating from the disparity of the matched edges.



a) The edges matched.



b) The regions to be machted.

Figure 2.1. Interpolating disparity in regions using disparity obtained from matchings of edges.

[1]Disparity is defined as the locational shift of the corresponding pixels which are projected from the same scene point in two images.

Another problem of the feature-based approach is that it requires that the same features should be extracted in both images. However, in real applications the features extracted from images are neither complete nor consistent in two images. The error arisen from the feature extraction will bring about the error in matching of features in turn.

On the other hand, the area-based approach matches each pixel in an image to a pixel in the other image by comparing the density distributions in a window centered at the two pixels to be matched. This approach can generate dense disparity, but there are three problems to be solved. First, it is very computationally expensive when the size of the window is large. Second, it is not easy to determine the size of the window. When the window is too small, correlations computed from the window will have not enough ability to discriminate the correct matching of a pixel from its matching candidates. When the window is too big, the correlation of pixels near the edge of a surface which are different in depth from other neighbor surfaces will be low and they may not be matched although they should be. The approach using an adaptive window[2] had been proposed to solve this problem, but it is computationally expensive due to its iterative nature. Third, the continuity constraint employed in this approach requires that the disparity should change smoothly over the image. This condition is not satisfied when the objects locate in a large range of depth in the scene or the change of viewpoint is large.

In this chapter, a new approach for stereo matching to obtain dense disparity is described. In that approach, we assume that the scene is composed of some piecewise connected surfaces. Thus the images of the scene taken from different viewpoints can be considered of containing some patches which can be matched continuously between two images. These patches correspond to the regions of the surfaces in the scene, but they need not to be homogeneous regions, that is, regions of pixels with the same gray value or color. Our objective is to find out the matching of these patches.

Similar to that when we look at object near to us we do not care about the

objects far from us or vice versa, in our method we first try to find possible matchings of patches of surfaces located within a small interval of depth in the scene at one time. In the second step, we integrate these matchings of patches in each small interval of depth to generate the matching of patches of the whole scene. The matchings of patches obtained within each interval of depth may be overlapped or conflicting with each other. Thus we consider the relation among the matchings of patches and select the best-matched one among the overlapped or conflicting matchings of patches as the unique matching of the patches.

2.2 Finding Matchings of Patches in Each Interval

As shown in Figure 2.2, a point in the scene projects to different points in the left and the right images. The shift of the location of the point p' in the right image relative to that of p in the left image along the epipolar line, shown as vector v, is called the disparity between the corresponding points p and p'.

According to perspective projection theory, an interval of depth is equivalent to an interval of disparity. Therefore, here we will use intervals of disparity instead of depth. We divide the range of disparity into some small intervals and find the matchings of patches within each interval of disparity.

We assume that the surfaces concerned should be projected to a region in the images with its width not less than w. If we divide disparity into some intervals not larger than w, then objects within such a small interval of depth will be projected to two images in the same order along an epipolar line, which is assumed to be horizontal here. Therefore dynamic programming can be used to find the matchings of patches along the epipolar lines in each interval of disparity.

In this section, we will first consider how to find the optimal matching of continuous segments of the pixels along the epipolar lines, by matching two sequences of pixels with continuity being taken into account. Then we merge the matched segment

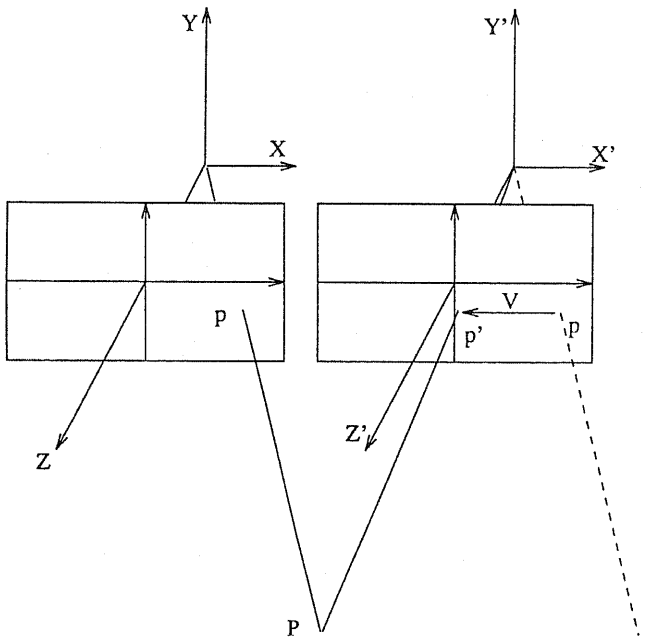


Figure 2.2: The disparity of corresponding points in two images

in each epipolar line into patches, considering the continuity between epipolar lines.

2.2.1 Matching in Epipolar Lines

Let $a_0a_1a_2.....a_m$ and $b_0b_1b_2.....b_n$ be two sequences of pixels along an epipolar line in the left and right images. For each pixel, we regard its color (their RGB value) as its features. Define the cost of matching two pixels as their *chromatic distance* in RGB space.

$$d(a_i, b_j) = \sqrt{(R(a_i) - R(b_j))^2 + (G(a_i) - G(b_j))^2 + (B(a_i) - B(b_j))^2} \quad (2.1)$$

If a pixel has no matching pixel in the other image, a special matching $null$ is assigned to it. This means that the pixel is occluded in the other image. A constant cost $M/2$ is assigned to the cost of an occlusion, where M is the maximal chromatic distance allowed for matched pixels. That is, if two pixel a_i and b_j are quite different in color so that the chromatic distance between them is over M, then they will never be

matched because matching both of them with $null$ has less cost.

As noticed above, in a small interval of disparity, $\{a_i\}$ and $\{b_j\}$ must be matched with order reserving. That is, if a_i is matched by b_j, then a_{i+1} can only match with b_k where $k > j$ (here the case where more than one pixels may match to one pixel in the other image is not considered). That is shown in Figure 2.3.

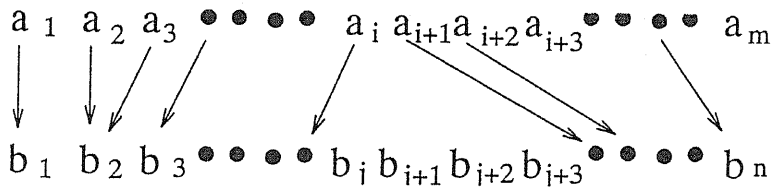


Figure 2.3: Matching of pixel sequences with order reserving

The optimal one-to-one matching of two pixel sequences with order reserving can be computed using dynamic programming[54,55,57] as follows:

$$\begin{aligned} C(i,j) \;=\; \min\{\; & C(i-1,j-1) + d(a_i,b_j), \\ & C(i-1,j) + M/2, \\ & C(i,j-1) + M/2\} \end{aligned} \tag{2.2}$$

The optimal matching is obtained as the matching which yields $C(m,n)$.

Under the constraint that two sequences are matched within a disparity interval $R_k = [d_k, d_{k+1}]$, for every matched pair of pixels (a_i, b_j), $d_k \leq j - i \leq d_{k+1}$ should hold. Thus only a small part of $\{C(i,j)\}$ satisfying the above condition needs to be computed.

2.2.2 Taking One-to-many Matching into Account

For a slant surface, as shown in Figure 2.4, the projected region in one image may be narrower or wider than the projected region in the other image. In other words, a slant

surface may project to a segment of n connected pixels on the epipolar line in one image, and project to a segment of m pixels on the corresponding epipolar line in the other image, where $n \neq m$. Therefore, the segment of n pixels will correspond to the segment of m pixels in the other image. In this case, we must allow one pixel in one image to be matched to more than one pixels in the other image, so that a narrower region can match to a wider region, or vice versa.

The optimal matching of two sequences of pixels with one-to-many and many-to-one matchings being allowed can be computed as follows:

$$\begin{aligned} C(i,j) \;=\; \min\{ & \; C(i-1,j-1) + d(a_i,b_j), \\ & \; C(i-1,j) + d(a_i,b_j), \\ & \; C(i,j-1) + d(a_i,b_j), \\ & \; C(i-1,j) + M/2, \\ & \; C(i,j-1) + M/2\} \end{aligned} \tag{2.3}$$

The second and third items the Equation 2.3 are computed for one-to-many matching in the pixel sequences. We can show that only if a one-to-many matching occurred, the second or third item will be the minimum of Equation 2.3. For example, consider the second item with the case that b_j was matched to $null$ before the matching of a_i and b_j. In this case,

$$\begin{aligned} C(i-1,j) + d(a_i,b_j) &= (C(i-1,j-1) + M/2) + d(a_i,b_j) \\ &\geq C(i-1,j-1) + d(a_i,b_j) \end{aligned} \tag{2.4}$$

That is, the second item will be suppressed by the first item in this case.

2.2.3 Continuity in the Matching

Because a continuous surface projects to continuous segments in the epipolar lines in both the left and the right images, these segments projected from a surface should

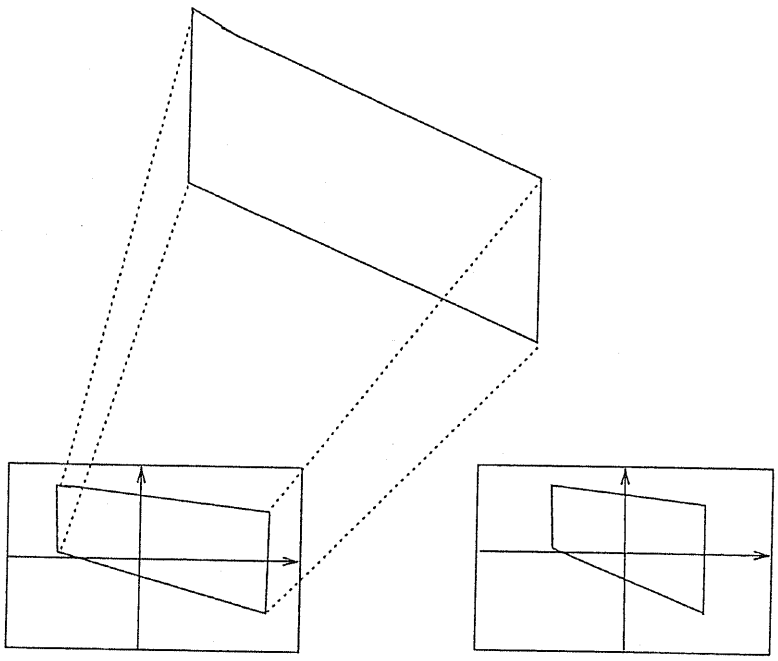


Figure 2.4: The regions of a slanting surface in two images

be matched continuously. Two segments P and P' in an epipolar line are said to be matched continuously if the following conditions are satisfied:

1. For $\forall p_i \in P$, there exists $p'_i \in P'$ so that $p_i \longrightarrow p'_i$ ($\longrightarrow$ means 'matches') . The same is true for $\forall p'_i \in P'$.

2. For any p_i, $p_j \in P$ and their corresponding pixels p'_i and p'_j, if p_i and p_j are adjacent, i.e., $Dist(p_i, p_j) = 1$, then $Dist(p'_i, p'_j) \leq \delta$. The same is true for P'.

where $Dist(a, b)$ stands for the distance between two pixels a and b, and δ is a number to control the allowed change of depth of two neighboring pixels on a continuous surface. In the following, we consider the case of $\delta = 2$.

The optimal matching with the least discontinuity of two sequences of pixels on an epipolar line can be obtained by imposing a large matching cost when a discontinuity occurred in the matching. According to the above definition of matching continuity, the discontinuity of matching occurs in the following cases.

1. $a_i \rightarrow b_j$, when $b_{j-1} \rightarrow null$ or $a_{i-1} \rightarrow null$

2. $a_i \rightarrow null$ or $b_j \rightarrow null$, when $a_{i-1} \rightarrow b_{j-1}$.

these two disconnected points occur in pairs, one at the left edge of matched patch and one at its right edge. So we need to impose a discontinuity cost on just either case, say, on the first one. Another case of violating the conditions of matching continuity is that more than two pixels matched to one pixel (in case of $\delta = 2$).

Suppose the cost imposed for a discontinuity be a constant D. Appending this cost into the computation in (2.3) yields:

$$\begin{aligned} C(i,j) \quad = \quad \min\{ \quad & C(i-1,j-1) + d(a_i,b_j) + g(i-1,j-1), \\ & C(i-1,j) + d(a_i,b_j) + h_1(i-1,j), \\ & C(i,j-1) + d(a_i,b_j) + h_2(i,j-1), \\ & C(i-1,j) + M/2, \\ & C(i,j-1) + M/2\} \end{aligned} \tag{2.5}$$

where

$$g(i-1,j-1) = \begin{cases} D & \text{if } C(i-1,j-1) = C(i-2,j-1) + M/2 \\ & \quad \text{or } C(i-1,j-1) = C(i-1,j-2) + M/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

It means that, if the optimal match until (a_{i-1}, b_{j-1}) is arrived by deleting a_{i-1} or b_{j-1}, then the disconnected cost is imposed for matching a_i and b_j.

$$h_1(i-1,j) = \begin{cases} D & \text{if } C(i-1,j) = C(i-2,j) + d(a_{i-1},b_j) \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

It means that, if the optimal match until (a_{i-1}, b_j) is given by matching a_{i-2} and a_{i-1} with b_j, then an additional cost for discontinuity is imposed if a_i is still matched to b_j.

$$h_2(i,j-1) = \begin{cases} D & \text{if } C(i,j-1) = C(i,j-2) + d(a_i, b_{j-1}) \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

The meaning of $h_2()$ is similar to that of $h_1()$.

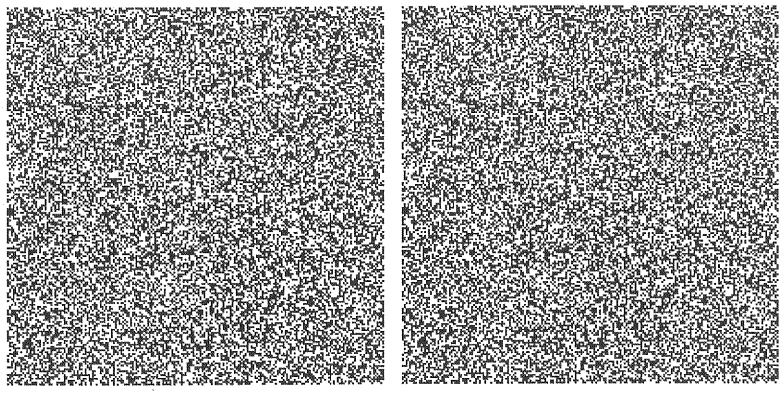
We can see from (2.5) that in the case of $D > M$, if a discontinuity have occurred when a_i and b_j are going to be matched, then all the pixels after a_i and b_j in the two sequences will never be matched. This is because matching a_i and b_j after a discontinuity will be imposed of a cost D additional to the matching cost, which is greater than the cost of deleting them. So the deletion of them will be chosen, and the same will be repeated for the following pixels, leading to the whole subsequences thereafter un-matchable. However, when some subsequences of considerable length after a_i and b_j can be matched, the cost of matching on those successive pixels, even added with D, may become less than the cost of deleting all of them, and thus the former should be chosen.

Therefore, we modified (2.5) so that this possibility occurs, resulting in the following dynamic programming using two computing arrays. The first array $C(i,j)$ is computed similarly to (2.5), where just the deletion of pixels is considered after a discontinuity has occurred. The second array $F(i,j)$ is computed for possible matchings of pixels after the discontinuity. When $F(i,j)$ gets less than $C(i,j)$, $C(i,j)$ takes the value of $F(i,j)$.

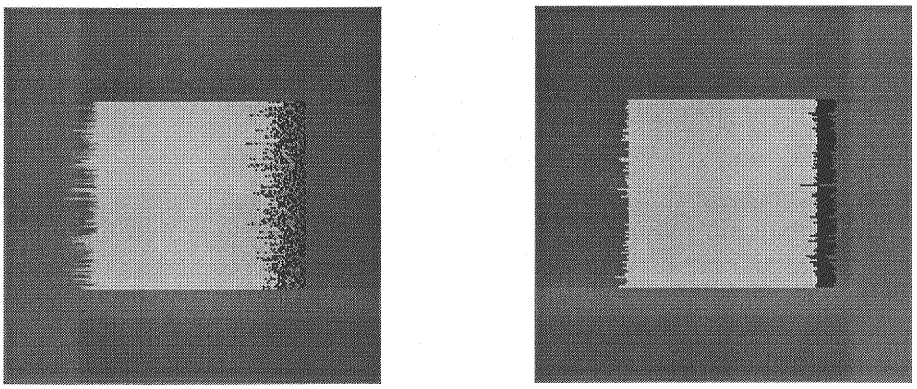
$$\begin{aligned} C(i,j) \;=\; \min\{\; & F(i,j) \\ & C(i-1,j-1) + d(a_i, b_j) + g(i-1,j-1), \\ & C(i-1,j) + d(a_i, b_j) + h_1(i-1,j), \\ & C(i,j-1) + d(a_i, b_j) + h_2(i,j-1), \\ & C(i-1,j) + M/2, \\ & C(i,j-1) + M/2\} \end{aligned} \tag{2.9}$$

$$
\begin{aligned}
F(i,j) \;=\; \min\{\;& F(i-1,j-1) + d(a_i,b_j), \\
& C(i-1,j-1) + d(a_i,b_j) + g(i-1,j-1) \\
& F(i-1,j) + d(a_i,b_j) + h_1'(i-1,j), \\
& F(i,j-1) + d(a_i,b_j) + h_2'(i,j-1)\}
\end{aligned} \tag{2.10}
$$

where $h_1'()$ and $h_2'()$ are defined similar to $h_1()$ and $h_2()$, though they are defined on $F(i,j)$.



a) Left and right images



b) Matching reuslts: no continuity considered

c) Matching reuslts: continuity considered

Figure 2.5: Random dot stereo images of a square over background.

Figure 2.5 illustrates the effects of taking continuity of matching into account. Figure 2.5(a) shows a pair of random-dot-stereo images of a scene where a square

is floating over the background. Figure 2.5(b) shows the disparity obtained by the matching without continuity being considered. The disparity is shown as a gray-scale image, with brighter grey-level showing larger disparity. The black part shows the occluded part. Figure 2.5(c) shows the disparity obtained when the continuity is considered.

2.2.4 Continuity between Epipolar Lines

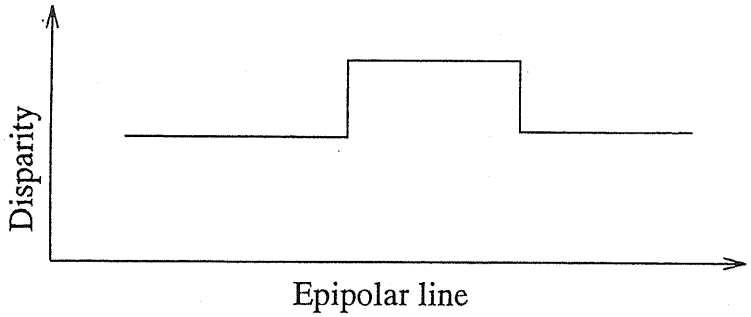
In general, when the interval of disparity is small, the optimal continuous matching of segments of pixels in epipolar lines can be considered to be the unique matchings in the interval. However, if occlusions exist, there may be ambiguity in some matchings.

Consider the case of random-dot-stereo images. As shown in figure 2.6(b), B'_{0_l} and B_0 are parts of the same surface, but B'_{0_l} is occluded in another image. If B'_{0_l} is consistent with the left part A_{1_l} of the surface A_1 which is located in front of B_0, then the following two matchings are possible.

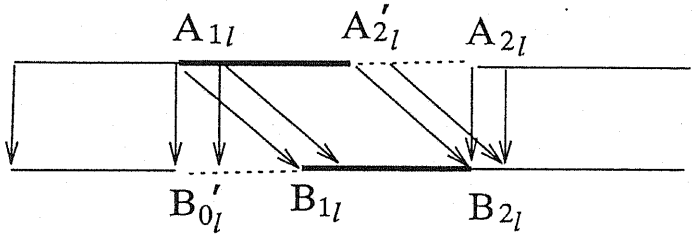
1. $A_{1_l} \longrightarrow B_{1_l}, B'_{0_l} \longrightarrow \{null\}$, that is, A_{1_l} is matched correctly as the left part of the front surface, and B'_{0_l} is regarded as occluded.

2. $A_{1_l} \longrightarrow B'_{0_l}$, $B_{1_l} \longrightarrow \{null\}$, that is, A_{1_l} is matched to a part of the back surface B'_{0_l} and B_{1_l} is regarded as occluded.

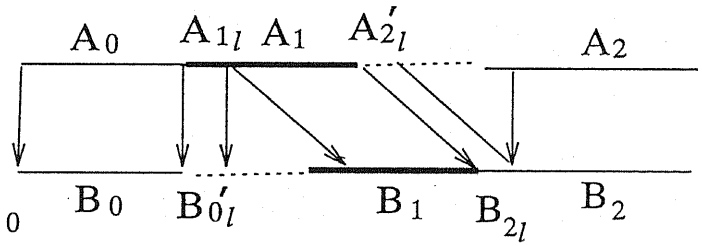
Which matching is obtained depends on the order in which the matching is performed. For example, we will obtain the matchings $A_{1_l} \longrightarrow B'_{0_l}$, $A'_{2_l} \longrightarrow B_{2_l}$, $\{null\} \longrightarrow B_{1_l}$ and $A_{2_l} \longrightarrow \{null\}$ as shown in figure 2.6(c), by proceeding the matching from left to right. On the other hand, if we proceed the matching from right to left, A_{0_r} and B_{1_r} will be mismatched to B'_{0_r} and A'_{2_r}, respectively, but A_{1_l} and B_{2_l} will be matched correctly. From these two sets of matchings obtained by performing matchings in different directions, we can select the correct ones by considering the continuity of matchings within each epipolar lines and the continuity between vertically adjacent epipolar lines.



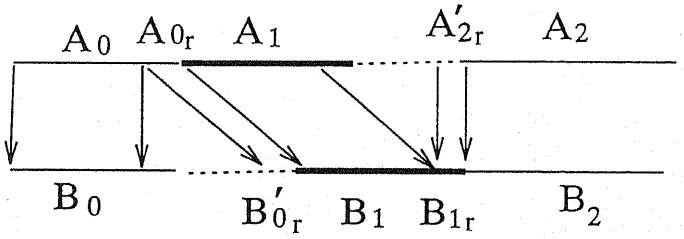
a) The surfaces in the scene



b) Ambiguity in matchings



c) The matchings from left to right



d) The matchings from right to left

Figure 2.6: Complementary matchings of two pass matching.

Denote i, j as the vertical and horizontal coordinates of a pixel, respectively. Let $\{P(i,j)\}$ and $\{Q(i,j)\}$ be the disparity of pixel (i,j) obtained from the matchings performed from left to right and from right to left, respectively, and let $\{B(i,j)\}$ be the set of pixels where $P(i,j) \neq Q(i,j)$, i.e.,

$$B(i,j) = \begin{cases} 1 & \text{if } P(i,j) \neq Q(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Let $\{D(i,j)\}$ be the disparity which we want to compute. For every pixel such that $B(i,j) = 0$, the disparity is unique, so $D(i,j) = P(i,j) = Q(i,j)$. For a pixel in $\{B(i,j) = 1\}$, we select a disparity from $P(i,j)$ and $Q(i,j)$, so that the following conditions are satisfied.

(i) Continuity within an epipolar line: $|D(i,j-1) - D(i,j)| \leq \delta$ or $|D(i,j+1) - D(i,j)| \leq \delta$.

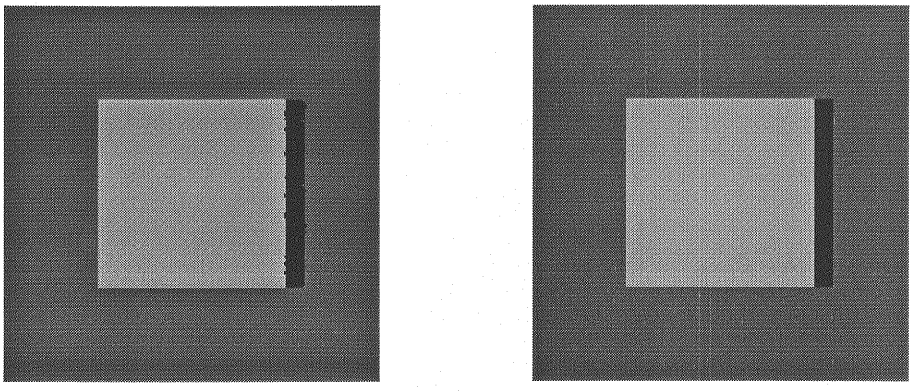
(ii) Continuity between epipolar lines: $|D(i-1,j) - D(i,j)| \leq \delta$ or $|D(i+1,j) - D(i,j)| \leq \delta$

The detailed procedure is:

1. Compute $\{P(i,j)\}$, $\{Q(i,j)\}$, $\{B(i,j)\}$, $\{D(i,j)\}$ as stated above. $D(i,j) = \infty$, if $B(i,j) = 1$.
2. Scan $\{B(i,j)\}$, and for every pixel such that $B(i,j) \neq 0$ do:
   (a) Set $D(i,j) = P(i,j)$. If the above conditions are satisfied, then $B(i,j) = 0$ and go to (d).
   (b) Set $D(i,j) = Q(i,j)$. If the above conditions are satisfied, then $B(i,j) = 0$ and go to (d).
   (c) $D(i,j) = \infty$.
   (d) Process next $B(i,j)$.

3. repeat steps 2 and 3 until all $B(i,j)$'s become zero.

Figure 2.7(a) shows the result obtained from Figure 2.5(c) by applying this process. Figure 2.7(b) shows the true disparity.



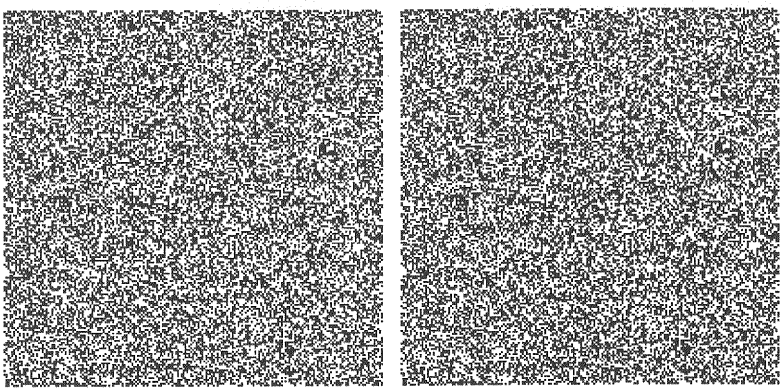
a) Disparity obtained b) The true disparity

Figure 2.7: Disparity after processed between epipolar lines for the random dot stereo images in Figure 2.5(a)

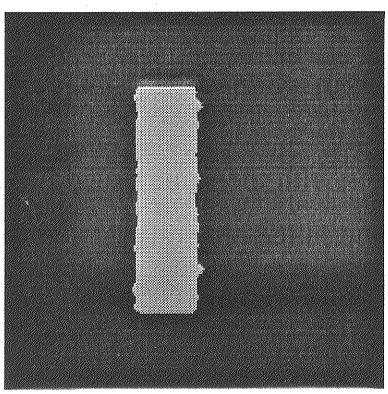
2.3 Integrating Matchings of Patches

If all the objects in the scene project to the images with the same order along the eppipolar lines, then the matching process described in the previous section will be sufficient for obtaining the matchings of patches in the whole scene by treating the whole range of depth as one interval. However, when the scene contains objects spread in a large range of depth and/or the distance between viewpoints is large, the left-right relation of the regions of two objects in different depth may change in two images. Figure 2.8(a) shows an example of random dot stereo images. These images are generated from a scene which contains two vertical rectangles in front of the background. The right rectangle in the left image locates much nearer to the viewer so that it came to the left of the farther one in the right image. In this case, each

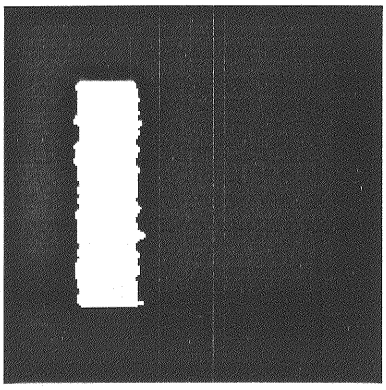
rectangle can only be matched with the disparity (depth) interval where it locates, and integration of each matching generates the structure of the whole scene. Figure 2.8(b) and (c) show the matching results in the intervals R_1(the interval of disparity from 10 to 19) and R_9 (disparity from 90 to 99), respectively. In each interval, a rectangle is matched. In this section, we will consider the integration of such matchings obtained within individual disparity intervals, in a general case.



a) Left and right images



b) matching reuslt in R_1



b) matching reuslt in R_9

Figure 2.8: Random dot stereo images of two vertical rectangles.

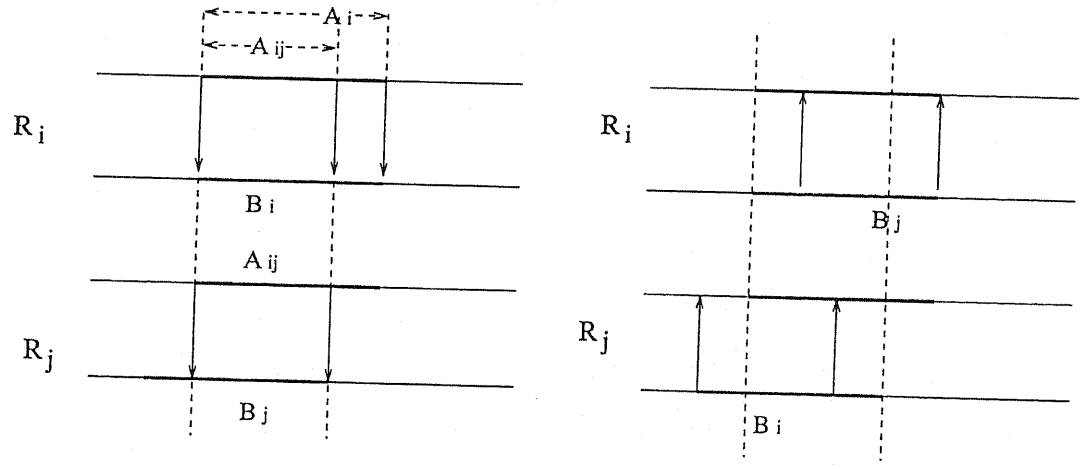
2.3.1 Selecting the Most Mature Matchings of Patches

A patch in one image should be matched to a unique patch in the other image. However, the matchings of the patches obtained from the individual intervals of depth may not satisfy this condition. For example, when there exist more than one objects with similar color, which are projected on the same epipolar lines but locate in different depths, the projected region of the objects in one image may match to different regions (in different depths) of other similar objects in the other image. For an object which projects to large homogeneous regions, while it is matched completely in the depth interval where it exists, it can also be matched partially in the neighboring intervals. Among those contradicting matchings, we select as the desirable one the matching of patches which is matched most maturely, that is, the matching with the lowest matching cost and the biggest matched area.

Consider a surface S which exists in interval R_i of depth in the scene. Suppose that S projects to patch P in the left image and to patch P' in the right image. P and P' are completely matched in interval R_i. This matching is denoted as $A_i \longrightarrow A_i'$, where $A_i = P$ and $A_i' = P'$. In another interval R_j, P and P' are matched only partially. The matching in R_j is denoted as $A_j \longrightarrow A_j'$, where $A_j \subset P$. Figure 2.9(a) shows this situation by segments on epipolar lines.

Let $A_{ij} = A_i \cap A_j \neq \emptyset$ be the intersection of A_i and A_j, and assume that A_{ij} is matched to B_i and B_j in R_i and R_j, respectively. If $B_i \neq B_j$, they are conflicting matchings for A_{ij}. In general, we can expect that the cost of the correct matching $A_{ij} \longrightarrow B_i$ in R_i is lower than $A_{ij} \longrightarrow B_j$ in R_j, so we can select the one with the lowest cost as final matching of A_{ij}. However, the cost of $A_{ij} \longrightarrow B_i$ may be equal to that of $A_{ij} \longrightarrow B_j$ when P is a homogeneous region, and may become larger under some conditions of illumination.

Instead of comparing the cost of $A_{ij} \longrightarrow B_i$ and $A_{ij} \longrightarrow B_j$, we consider both these two matchings in R_i and R_j, simultaneously. We can see in Figure 2.9(b) that



a) The overlapping mathings b) The matching of B_j and B_i in R_i and R_j

Figure 2.9: The overlapped matching of patchs in two intervals.

in R_i both B_i and B_j are matched well, but in R_j, B_i can only be matched partially while B_j is matched well. This is because that B_i and B_j are parts of patch P' and that P' is matched with P more maturely in R_i than in R_j.

Using the fact above, we can recompute the matching cost of conflicting matchings of $A_{ij} \longrightarrow B_i$ and $A_{ij} \longrightarrow B_j$ as follows.

$$M_i(A_{ij}) = (C_i(A_{ij}, B_i) + C_i(X_i, B_j))/2 \tag{2.12}$$

$$M_j(A_{ij}) = (C_j(X_j, B_i) + C_j(A_{ij}, B_j))/2 \tag{2.13}$$

where $M_k(P)$ is recomputed matching cost for patch P in R_k. $C_k(A, B)$ is the average of the matching costs of pixels in A and B matched in R_k. X_i and X_j are the corresponding patches of B_j and B_i in R_i and R_j, respectively. In R_i, as both B_i and B_j are well matched, $M_i(A_{ij})$ will be low. On the other hand, in R_j, B_i is matched only partially and the unmatched part will have high costs (cost of deletion), so $M_j(A_{ij})$ will be larger.

The computation above are stated for a whole patch. However, when depth

of a surface changes gradually along the vertical direction, the whole patch may not be completely matched in one interval. Therefore, instead of carrying out the above computation for a whole patch, we just recompute the matching similarity with respect to a part of it, that is, the computation above is carried out by using the concerned epipolar line and the neighboring lines below and above it.

This recomputation is repeated for all the patches matched in two neighboring intervals R_i and R_{i+1}. After this recomputation of matching costs, the costs of matchings of patches in the correct depth interval remain in low values, but the costs of matchings of patches in other intervals become larger. Finally, from the conflicting matchings of patches, we select the one with the lowest recomputed cost as the unique matching.

2.3.2 Judgment for Occlusion

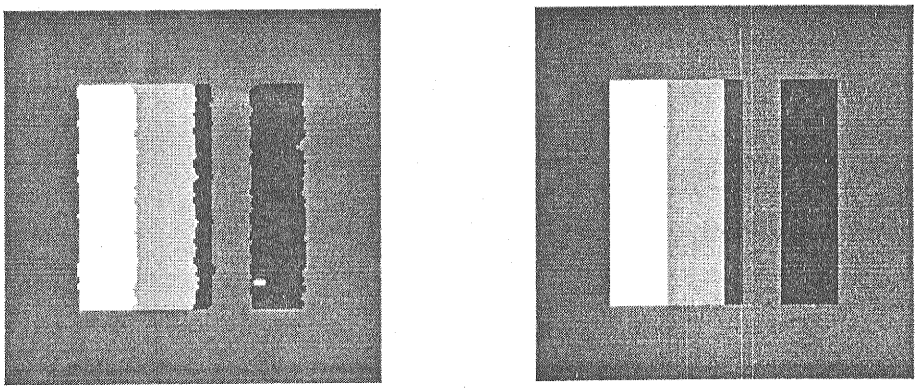
In the above processing, a unique correspondence is selected for each patch in one image, say left image. However, different patches in left image may match to the same patch in the right image. For example, suppose that a scene has a nearly constant colored background and a part of the background is visible in the left image and is occluded in the right image. This occluded part of background may match to another part of the background in the right image, which will be also matched by a correct correspondence in the left image. For those many to one matchings of patches, we must decide that which patches are matched correctly and which patches are occluded.

Suppose that P_1 and P_2 are two patches in left image, and P' is a patch in right image so that $P_1 \longrightarrow P'$ and $P_2 \longrightarrow P'$. Here we simply compare the cost of $P_1 \longrightarrow P'$ and $P_2 \longrightarrow P'$, and select the one with the lower cost as the correct matching and regard the other as occluded patch.

2.4 Experimental Results

We applied our method to several random-dot-stereo images and some real images.

The final result of matching for the random dot stereo images in Figure 2.8(a) are shown in Figure 2.10(a). The true disparity of them are depicted in Figure 2.10(b).



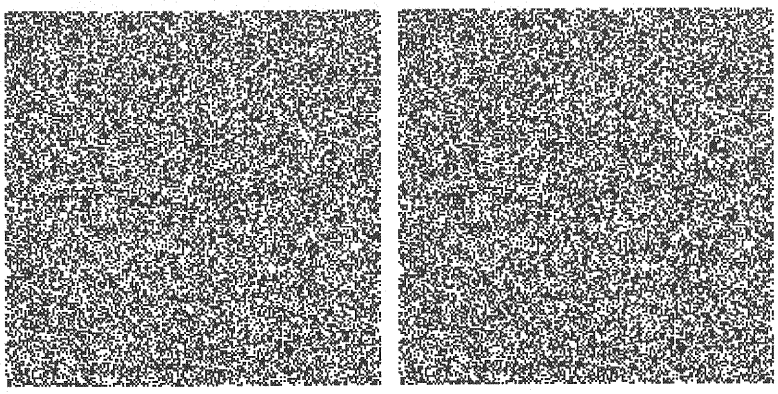
b) matching reuslts c) true disparity

Figure 2.10: Result for images in Figure 2.8(a)

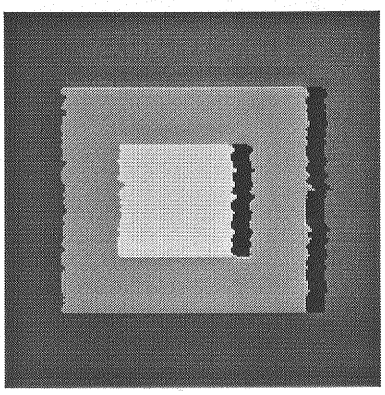
Figure 2.11(a) shows another pair of random dot stereo images which contain two squares floating over the background, one of them is in the front of the other. Figure 2.11(b) shows the matching result of them, and (c) depicts the true disparity.

Figures 2.12(a) shows a pairs of stereo images of the real world obtained form the "Database of Multi-viewpoint Image for Research" by Ohta et al., Tsukuba University, Japan. These images are complicated since the objects in the scene located in a large range of depth. The patch matching similarity in several intervals are shown from (b) though (f). The final result is shown in Figures 2.12(e). We can see that method generated reasonable results for this complicated scene.

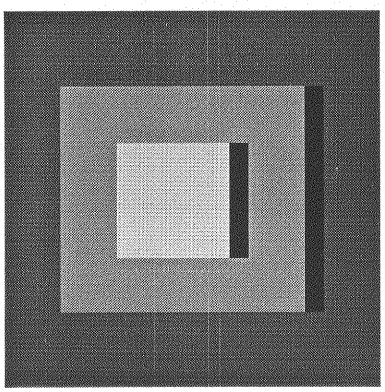
Figure 2.13 shows the experimental results of another pair of images from the "Database of Multi-viewpoint Image for Research". This pair of images looks not so complicated, but there exist some surfaces with nearly constant color whose depth changes gradually. Reasonable results are obtained by our method.



a) Left and right images

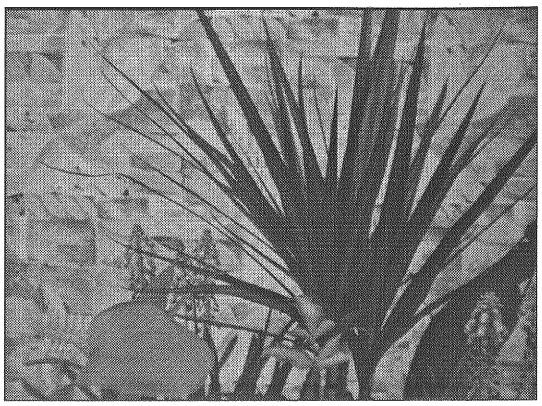
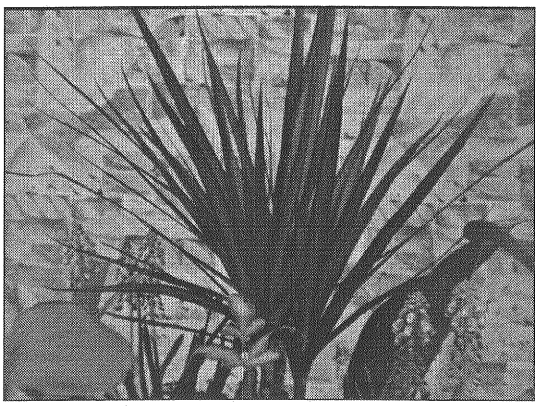


b) Disparity obtained

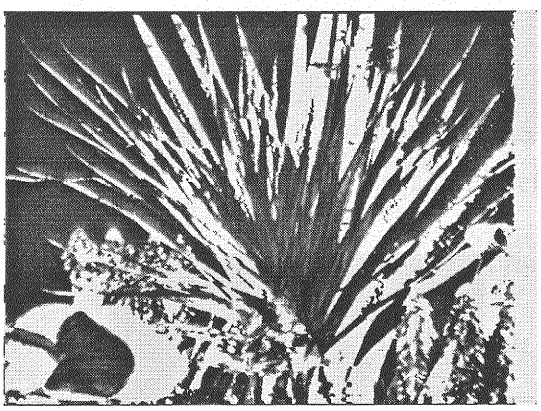


c) The true disparity

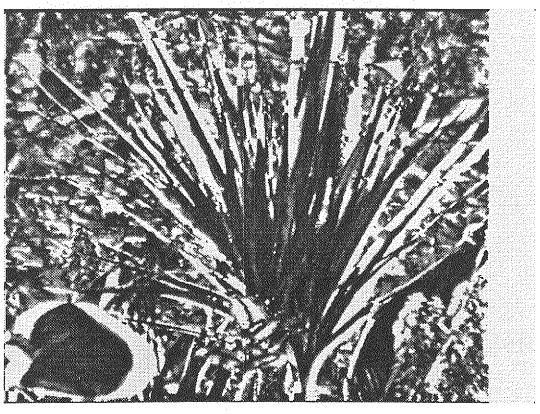
Figure 2.11: Random dot stereo images of two squares.

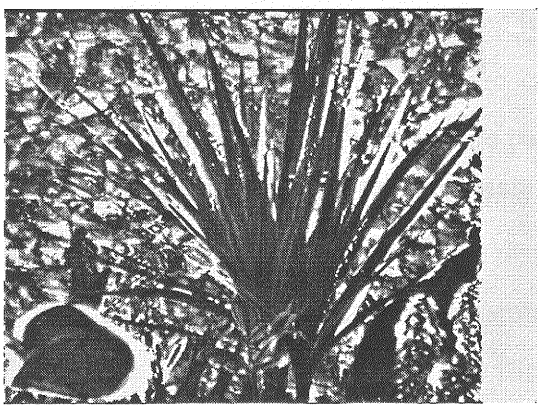
a) Left and right images



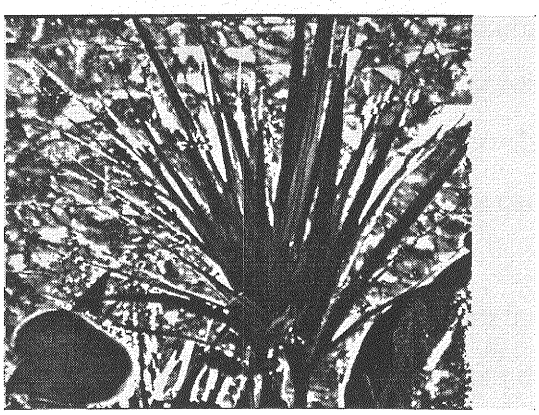
b) Matching in interval 0



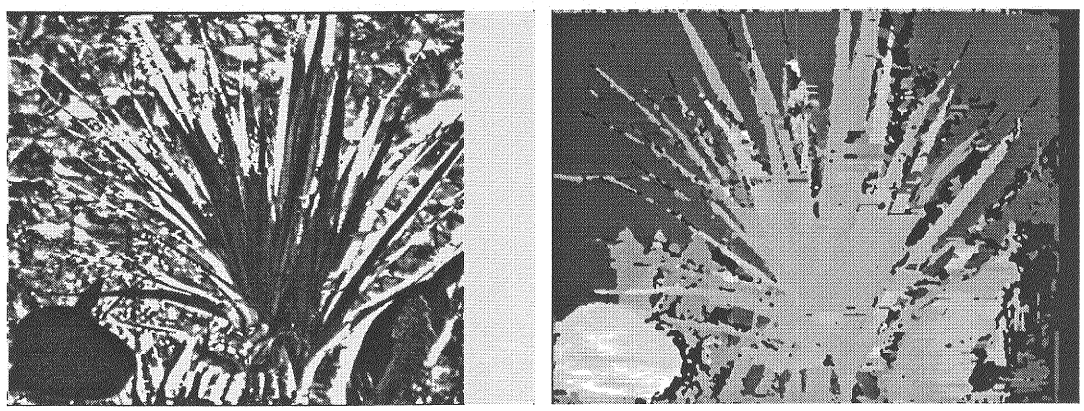
c) Matching in interval 3



d) Matching in interval 4

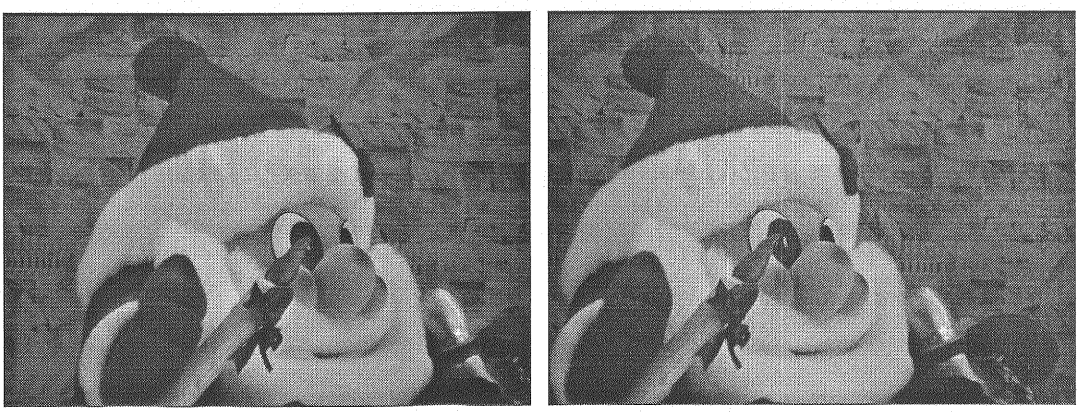


e) Matching in interval 5

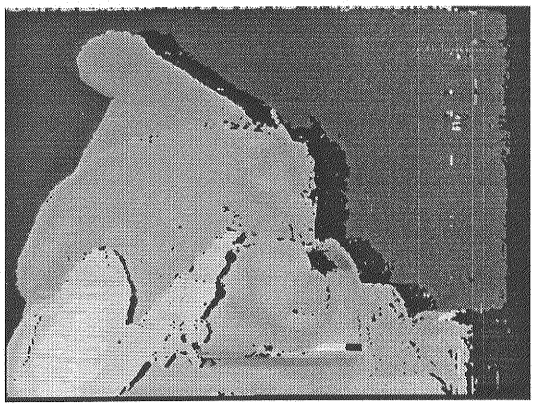


e) Matching in interval 6 e) The obtained disparity

Figure 2.12: Experimental result for stereo images of plants



a) Left and right images



b) The obtained disparity

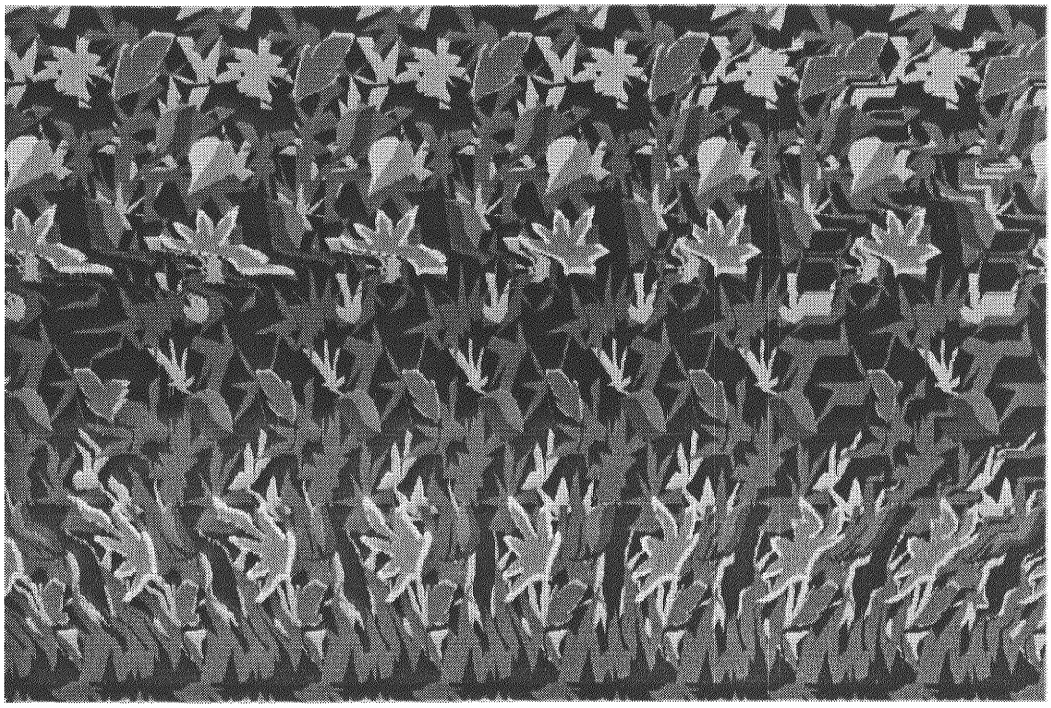
Figure 2.13: Disparity from two stereo images of a doll

The computation for the images in Figures 2.12 and 2.13 (size of 640 $\times$ 480, full color) takes about 20 minutes (user time on SUN SPARC-10), though it would be much decreased by various considerations for speeding up.

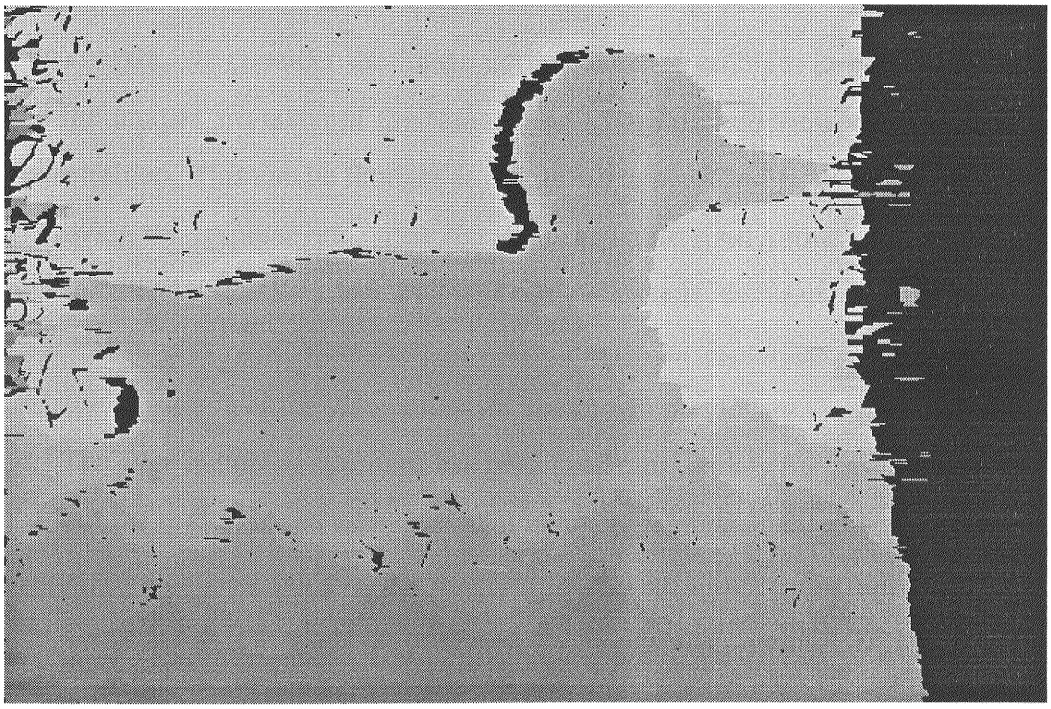
Figure 14(a) shows a picture which are called as magic card. Some other structure will be revealed from the picture if we see it in the way that the sight of one eye is shifted from that of the other eye. In principle, this is equivalent to the stereo fusion. Figure 14(b) shows the result "seen" by our algorithm, that is, the result of matching the picture with itself with a nonzero shift.

2.5 Summary

In this chapter, we proposed a new approach for stereo matching to obtain dense disparity. In this approach the patches which can be matched continuously between two images are regarded as matching primitives. The stereo matching problem was then represented as a problem of finding consistent matchings of such patches in the scene. This is done in a part-to-whole approach which consists of two steps. In the first step, we divide the range of disparity of the scene into some small intervals, and use dynamic programming, where continuity of the matching was taken into account, to find continuous matchings of patches in each interval of disparity. In the second step, we integrate these matchings of patches to generate a consistent set of matching of patches of the whole scene, by considering vertical continuity between adjacent epipolar lines as well as consistency and competition among matchings in different depth intervals, and selecting the best-matched one among the overlapped or conflicting matchings of patches. Experimental results on random dot stereo images and the images of real scene showed the effectiveness of our method.



a) A magic card



b) The matching result

Figure 2.14: Experimental result on a magic card

CHAPTER 3

Region Matching Based on Similarity of Local Structure

3.1 Introduction

In this chapter, we consider the image correspondence problem in general case, in which the images are taken of the same scene from different viewpoints which are not constrained. This can be thought as a problem generalized from stereo matching by removing the epipolar line constraint. This generalization allows us to be able to deal with other applications such as image retrieval, scene identification, motion with large frame interval, and other tasks in which image correspondence is required but the relative positions of the viewpoints are not known previously.

The edge matching approach has been used most widely for image correspondence, but the region-based approach, that is, using regions as matching primitives, are more promising for our purpose here, for the reason stated in Section 1.2.

There are a few methods proposed for region matching[20,31-33]. While the early works by Marapane[31] and Fu[32] only used the features of regions for matching, the recent one by Chen[33] tried to use structural information of region

adjacency relations, which unfortunately tends to be meaningful and stable only for regions belong to the same object. Lee[20] computed the matching probability of a region with its corresponding candidates using region features, and then improved them by relaxation based on the pair of similar triangles in two images. However, the relation of similar triangles is neither stable nor discriminating, because the triangle of the centers of gravity of triple regions can hardly keep in similarity and, on the other hand, two triples of regions composing two similar triangles may be irrelevant.

In this chapter, we describe a region approach which utilizes both the features of the regions and the local structures of regions to suppress the matching ambiguities occurred when only the features of regions are used. The local ternary relations among regions, that is, three triangles composed of the centers of gravity of the region in question and its three nearest neighbors, are used to describe the local structure of the region which is concerned. The matching similarity of a given region in one image with a corresponding candidate in the other image are computed with both feature similarity and local structural similarity between them, and from the corresponding candidates, the one having the highest matching similarity is selected as the most suitable matching of the given region.

3.2 Segmentation of the Images

Assume that the objects and surfaces in the scene are composed of some faces which have uniform reflecting property so that they are projected to nearly homogeneous regions of color in the images. Therefore, the images can be segmented into regions, which are sets of connected pixels with similar color.

All the pixels in the image are first classified into some clusters so that all the pixels in one cluster have similar color. This is done by clustering[73] the pixels with their chromatic features, that is, the three coordinates of their color in HSV space. Here the HSV model proposed by Abe and Yagi[71] is adopted for the chromatic

space.

The image is segmented initially based on the clusters obtained, that is, the pixels in each cluster are assigned with a same label indicating the cluster. Because two pixels which have the same able are just just the pixels which have similar color in the image, so there is no guarantee for the pixels which have the same label to be connected with each other. This means that the pixels of the same label do not compose a unique region. Therefore, all the pixels need to be relabeled so that all the pixels having the same label are connected with each other to form a region.

After relabeling of the pixels, all the regions, that is, the sets of connected pixels with the same label, in the image will have distinct label. Due to noise and local distortion of the chromatic feature of pixels, some meaning-less small regions may appear. These small regions are merged to the one of regions neighboring them by the following two steps.

In the first step, small regions with the area less the T_0 are first merged to the adjacent region with the longest common boundary, regardless of their chromatic features. That is, for a region R_i that $a(R_i) < T_0$, where $a()$ stands for the area, merge it to R_k such that

$$C(R_i, R_k) = \max_{R_j \in N(R_i)} \{C(R_i, R_j)\}$$

where $C(R_i, R_j)$ stands for the length of the common boundary between R_i and R_j, and $N(R_i)$ is the set of the regions adjacent to R_i.

In the second step, the regions with area greater than T_0 but less than T_1 are merged to their neighboring regions, by considering their chromatic similarity and the length of their common boundary. That is, a region R_i such that $T_0 < a(R_i) < T_1$, can be merged to a region R_j adjacent to R_i, if they satisfy the following conditions:

1. R_i and R_j have similar chromatic features, that is, the distance between the average colors of R_i and R_j in the HSV space is less than a given threshold

T_c. The distance between color A and B in the HSV space is computed as:

$$D_c(A,B) \equiv \sqrt{(V_A - V_B)^2 + S_A^2 + S_B^2 - 2S_A S_B \cos(H_A - H_A)} \quad (3.1)$$

where V_C, S_C and H_C are the value of brightness, saturation and hue of color C, respectively. See [71] about how V_C, S_C and H_C can be computed from the RGB values of C.

2. The length of the common boundary between R_i and R_j is not shorter than one third of the length of the perimeter of region R_i.

All the pairs of adjacent regions which satisfy the above conditions are merged sequentially in the order of the length of the common boundary normalized by the length of the perimeter of the region to be merged. This is carried out by the following steps:

1. For each region R_i, let $N'(R_i)$ be the set of the regions which are adjacent to R_i and the pair of (R_i, R_j) satisfies the above two conditions. Find the region R_{i_m} from $N'(R_i)$ so that

$$l_n(R_i, R_{i_m}) = \max_j l_n(R_i, R_j)$$

where $l_n(A,B)$ stands for the length of the common boundary between A and B.

2. From all the pairs $\{R_i, R_{i_m}\}$, find the pair $(R_i^*, R_{i_m}^*)$ which has the longest length of the common boundary normalized by the length of the perimeter of R_i. That is,

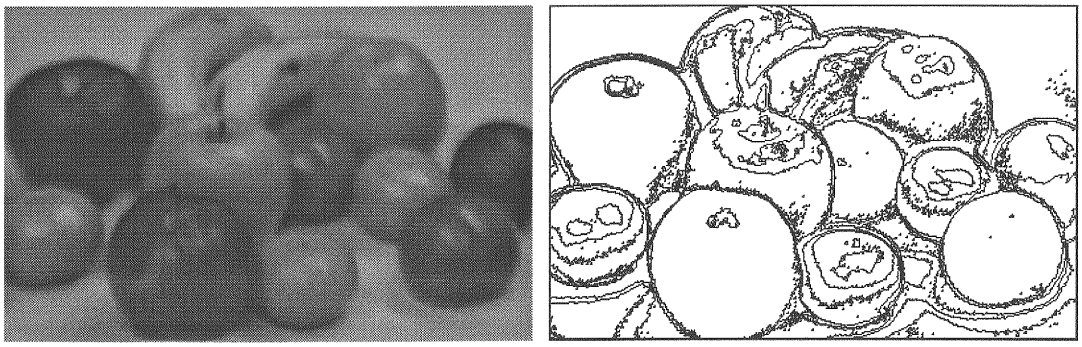
$$l_n(R_i^*, A_{i_m}^*) = \max_{R_i} l_n(R_i, R_{i_m})/l(R_i) \quad (3.2)$$

where $l(R_i)$ stands for the length of the boundary of region R_i.

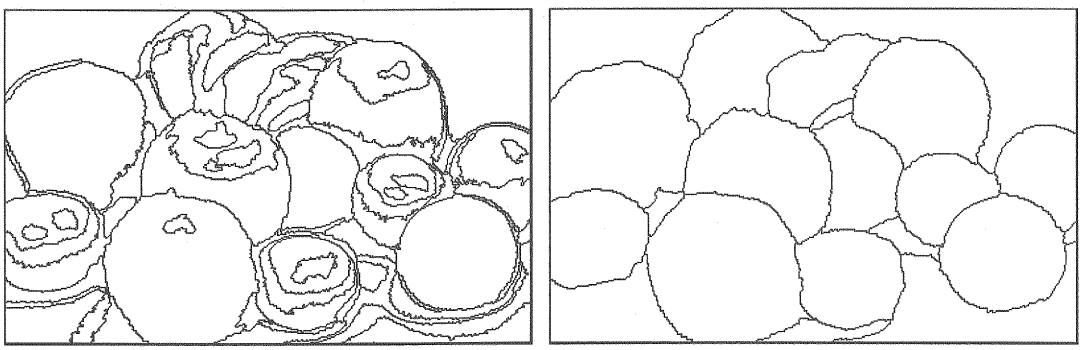
3. Merge R_i^* to $R_{i_m}^*$

The above procedure is repeated for all the regions which satisfy the region merging conditions.

Figure 3.1 shows the results obtained in the region segmentation procedure. Figure 3.1(a) shows an input image and the results of clustering in the HSV space is shown in Figure 3.1(b). Figure 3.1(c) shows the result after the regions smaller than T_0 pixels (where $T_0 = 100$) are merged in the first step. Figure 3.1(d) shows the final result of the segmentation of the image in Figure 3.1(a) after the region merging in the second step finished (where $T_1 = 4000$, $T_c = 0.10$).



a) Input images (bright values) b) Result of clustering



c) Result after the 1st merging step d) Result after hte 2nd merging step

Figure 3.1: Segmentation of color image

3.3 Local Structure of Regions

For each region obtained above, some features of them, such as area, average color, moments and center of gravity, are computed in advance for the use in matching procedure.

As what the early region-matching methods did, we may be able to determine a correspondence of a given region in one image by finding the region in another image which is most similar to the former in features. However, when the scene contains some similar surfaces, the features of them may be merely different. Moreover, in the images in presence of variation in illumination, photometric distortion or noise, for a given region in one image, any one of its corresponding candidates may happen to be most similar in features to the given region. Therefore, it is nonsensical to determine the correct correspondence for a given region by choosing the one most similar in features among the candidates which are merely different in features.

To reduce the ambiguity of region matching, we suggest that instead of matching regions only by their features individually, a group of regions should be considered. That is, the matching for a given region in one image can be established more stably and reliably if we require that the given region and its corresponding one in the other image are not only similar in features, but also have similar neighboring regions which surround them with similar configurations. The configuration of the neighboring regions around the concerned region is represented by the local structure of regions, which is described in the follows.

3.3.1 Ternary Relation of Regions

It is noticed by Lee[20] that as the local relation, the ternary relation among three regions is more determinant than the binary relations between two regions, because ternary relation also determine the relative position between each other. The ternary relation of three regions is represented by a triangle, which is composed of the centers

of the gravity of three regions as its vertices and the lines connecting the centers of gravity of the regions as its edges. As shown in Figure 3.2, under the assumption that the viewpoint changed not so greatly, if three regions in one image are corresponded correctly by a triple of regions in the other image, then the triangle composed by the centers of gravity of the triple regions in two images would remain similar.

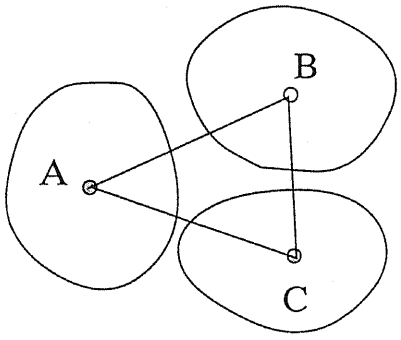


Figure 3.2: Ternary relations among regions

It is shown that when the ternary relation is represented by the angles and edges of the triangle, the ternary relation of regions is invariant under the transformation of rotation and translation of the image[34].

The ternary relation of region is used to represent the local relational structure of regions in the images. Consider a region A which is concerned and three regions A_1, A_2 and A_3 which are neighboring with A. As shown in Figure 3.3, the relation of A and each two regions from A_1, A_2 and A_3 can be represented by a triangle, and the three triangles $\triangle AA_1A_2$, $\triangle AA_2A_3$ and $\triangle AA_1A_3$ describe the configuration of the four regions completely. This set of three triangles composed of the concerned region A and its three neighboring regions is defined as the local structure of A. If A is corresponded correctly to a region B in the other image, we can expect that B has a similar local structure as that of A.

For a given region A in one image, the corresponding region in the other image is found as the region which is similar to A in features as well as in local structure.

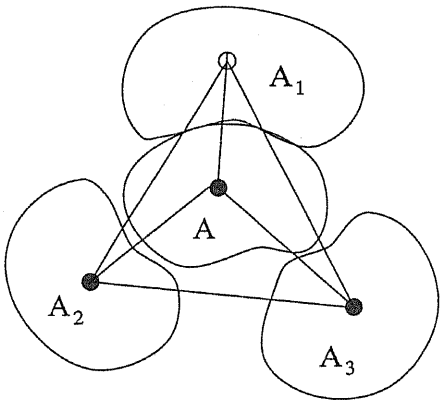


Figure 3.3: Local relational structure of regions

3.3.2 Similarity of Local Structure

Let A_i be the concerned region and (A_i^1, A_i^2, A_i^3) be the three neighboring regions of A_i in image $\mathcal{A}$. Let $\{B_{i_k}\}$ be the candidates of correspondence of A in image $\mathcal{B}$. Let us consider that when A is supposed to be matched with B_{i_k}, how well the local structure of A can also be matched with that of B_{i_k} in image $\mathcal{B}$.

For each region A_i, the local ternary relations formed by three triangles, which are composed of A_i and two of its three nearest neighbors A_i^1, A_i^2 and A_i^3, are constructed. In order to compute the local structural similarity of region A_i and its corresponding candidate B_{i_k}, we map also A_i^1, A_i^2 and A_i^3 to their corresponding candidates $B_{i_k}^1$, $B_{i_k}^2$, $B_{i_k}^3$. This can be thought as mapping the three triangles to other three triangles in the other image. Figure 3.4 shows an example where (A_i^1, A_i^2, A_i^3) are matched to one triple of corresponding candidates of them.

For each pair of a triangle and the mapped one, we compute their similarity in terms of edge similarity, angle similarity, and vertex (region) similarity. The sum of similarities of three pairs of the triangles is regarded as the relational similarity between A_i and B_{i_k} under the specified mapping $\mathcal{P}$ of $(A_i^1, A_i^2, A_i^3) \xrightarrow{\mathcal{P}} (B_{i_k}^1, B_{i_k}^2, B_{i_k}^3)$. It is computed as:

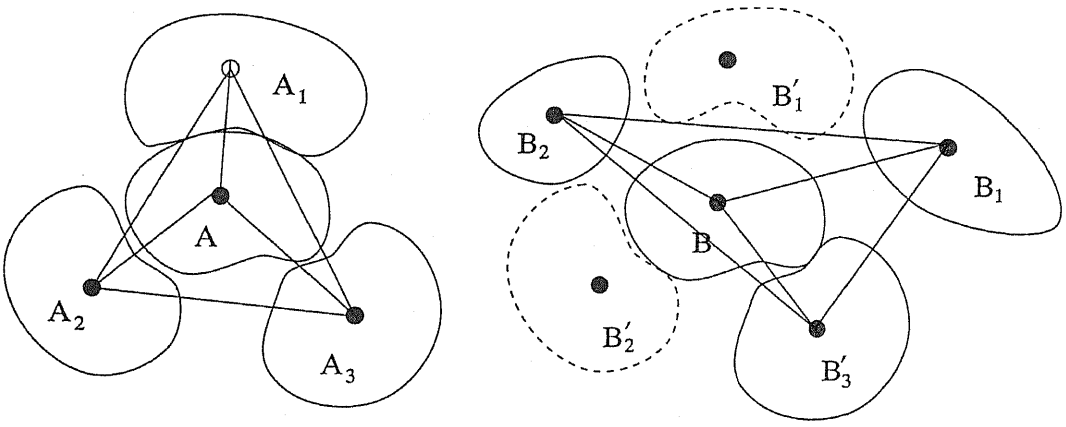


Figure 3.4: Matching of the local structure.

$$
\begin{aligned}
S(A_i, B_{i_k}, \mathcal{P}) &= S_t(\triangle A_i A_i^1 A_i^2, \triangle B_{i_k} \mathcal{P}(A_i^1)\mathcal{P}(A_i^2)) + \\
&\quad S_t(\triangle A_i A_i^1 A_i^3, \triangle B_{i_k} \mathcal{P}(A_i^1)\mathcal{P}(A_i^3)) + \\
&\quad S_t(\triangle A_i A_i^2 A_i^3, \triangle B_{i_k} \mathcal{P}(A_i^2)\mathcal{P}(A_i^3)) \\
&= S_t(\triangle A_i A_i^1 A_i^2, \triangle B_{i_k} B_{i_k}^1 B_{i_k}^2) + \\
&\quad S_t(\triangle A_i A_i^1 A_i^3, \triangle B_{i_k} B_{i_k}^1 B_{i_k}^3) + \\
&\quad S_t(\triangle A_i A_i^2 A_i^3, \triangle B_{i_k} B_{i_k}^2 B_{i_k}^3)
\end{aligned}
$$

Here $S_t(\triangle abc, \triangle def)$ is the similarity between two triangles:

$$
\begin{aligned}
S_t(\triangle abc, \triangle def) = S_v(\triangle abc, \triangle def) S_e(\triangle abc, \triangle def) \\
S_a(\triangle abc, \triangle def)
\end{aligned}
$$

In this formula, $S_v(\triangle abc, \triangle def)$ is the matching similarity between two triangles, that is initially computed by the product of the chromatic similarities of three pairs of vertices of the two triangles:

$$
S_v(\triangle abc, \triangle def) = S_c(a, d) S_c(b, e) S_c(c, f)
$$

$S_e(\triangle abc, \triangle def)$ is the similarity of edges of the two triangles and is computed by

$$S_e(\triangle abc, \triangle def) = S_l(\overline{ab}, \overline{de}) S_l(\overline{ac}, \overline{df}) S_l(\overline{bc}, \overline{ef})$$

where S_l stands for similarity of length between two line segments and is computed as $S_l(l_1, l_2) = 4l_1 l_2/(l_1 + l_2)^2$.

$S_a(\triangle abc, \triangle def)$ is the angle similarity between two triangles and is computed by

$$S_a(\triangle abc, \triangle def) = 1 - (|\angle bac - \angle edf| + |\angle abc - \angle def| + |\angle acb - \angle dfe|)/360$$

We map A_i^1, A_i^2 and A_i^3 to all of their possible candidates, and compute the maximum of the relational similarity between A_i and B_{i_k}, which is treated as the structural similarity $\tilde{S}(A_i, B_{i_k})$ between A_i and B_{i_k}.

$$\tilde{S}(A_i, B_{i_k}) = \max_{\mathcal{P}} S(A_i, B_{i_k}, \mathcal{P})$$

Figure 3.5 shows the local structures of two regions which corresponds to the maximal local structural similarity between A and B.

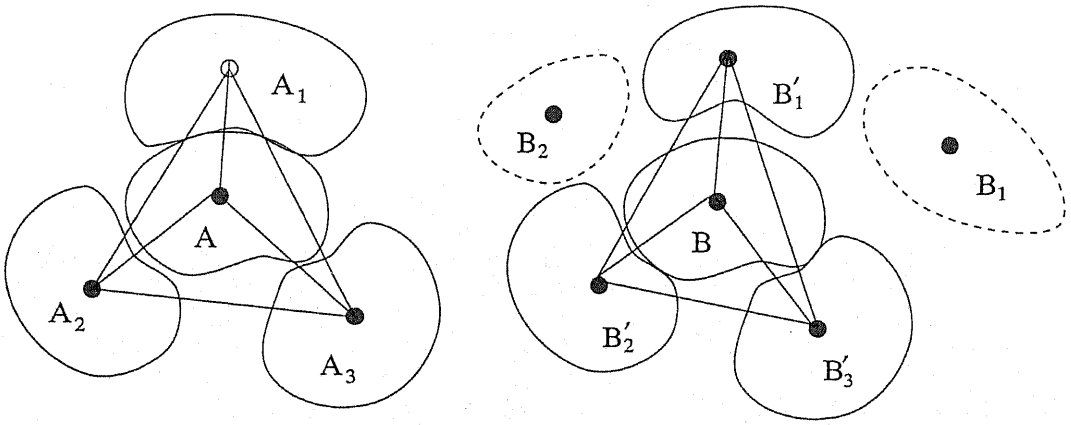


Figure 3.5: Local structure matched with maximal similarity.

3.3.3 Consideration for Region Occlusion

In the computation of the above, if one of the three regions A_1, A_2 and A_3, which are the nearest neighboring regions of A in image $\mathcal{A}$, is occluded in the other image, the local structure of A may not be matched well by that of the corresponding region B in image $\mathcal{B}$, in spite of that the local structure of A and other three neighboring regions, which do appear in image $\mathcal{B}$, can be matched with that of B with high structural similarity.

In order to compute local structural similarity more robust against occlusion of regions, we select four neighboring regions of A for computing local structural similarity. For A and each triples of the four neighboring regions, a local structural similarity are computed by the method stated in the previous section. Among the structural similarities computed with four triples of the neighboring regions, the maximal one is selected as the local structural similarity of A and B.

3.4 Region Matching Using Local Structure Similarity

In this section, we will describe the procedure of matching regions by using both their feature similarity and local structure similarity. The idea is to establish the initial matching for each region using their features, and then improve the matchings using similarity of local structure of regions.

3.4.1 Initial Matching by Chromatic Similarity

For each region A_i in one image $\mathcal{A}$, the corresponding candidate regions $\{B_{i_k}\}$ in the other image are first found out as the ones which have color and area similar to A_i. In other words, $\{B_{i_k}\}$ satisfy the following conditions:

1. $1/T_a \leq a(A_i)/a(B_{i_k}) \leq T_a$ and
2. $D_c(A_i, B_{i_k}) \leq T_c$.

where $a()$ stands for area, and $D_c()$ is the *chromatic distance* in the modified HSV space[72] between two regions, as defined by Equation (3.1). T_c and T_a are certain thresholds and set to 0.07 and 4.0, respectively, in the experiments.

For each region A_i and its corresponding candidate B_{i_k}, compute their *chromatic similarity* by $S_c(A_i, B_{i_k}) = 1 - D_c(A_i, B_{i_k})/T_c$. The chromatic similarity between a region and its matching candidate is regarded as the initial matching similarity of them. That is,

$$S^{(0)}(A_i, B_{i_k}) = S_c(A_i, B_{i_k})$$

where $S^{(m)}(A, B)$ denote the similarity between the regions A and B in the m-th iteration to be described below.

For each region A_i in image $\mathcal{A}$, we determine the initial correspondence of it by selecting the region which has the maximal matching similarity with A_i.

$$S^{(0)}(A_i, B_i^*) = \max_k S^{(0)}(A_i, B_{i_k})$$

The same computation is also carried out for each region in image $\mathcal{B}$.

3.4.2 Relaxation Based on Local Structure

The computed initial matching similarity between a region A_i and its corresponding candidate B_{i_k} is then improved iteratively, using the local structural similarity, by the following steps:

1. For each region A_i in image $\mathcal{A}$ and a corresponding candidate B_{i_k} in image $\mathcal{B}$, compute the structural similarity of them by the method stated in the previous section. Here the similarities of the vertices of the triangles are computed using the the region matching similarity obtained in the last iteration. That is, the S_v, which is involved in the computation of local structure similarity, is computed

by

$$S_v(\triangle abc, \triangle def) = S^{(n-1)}(a,d)S^{(n-1)}(b,e)S^{(n-1)}(c,f) \qquad (3.3)$$

2. After all structural similarities between a region A_i and its corresponding candidate B_{i_k}, which is denoted as $\tilde{S}(A_i, B_{i_k})$, have been computed, a new matching similarity $S(A_i, B_{i_k})$ between A_i and its corresponding candidate B_{i_k} is calculated by normalizing $\tilde{S}(A_i, B_{i_k})$ by $\sum_j \tilde{S}(A_i, B_{i_j})$.

   All the processes and computations above are also carried out for every region B_j in image $\mathcal{B}$.

3. Integrate the matching similarities computed for two images by

$$S^{(n)}(A_i, B_j) = (S(A_i, B_j) + S(B_j, A_i))/2$$

4. For each region A_i in image $\mathcal{A}$, we determine the correspondence of using the newly obtained matching similarities. That is, select the corresponding candidate which has the maximal new matching similarity with A_i as the correspondence of it.

$$S^{(n)}(A_i, B_i^*) = \min_k S^{(n)}(A_i, B_{i_k})$$

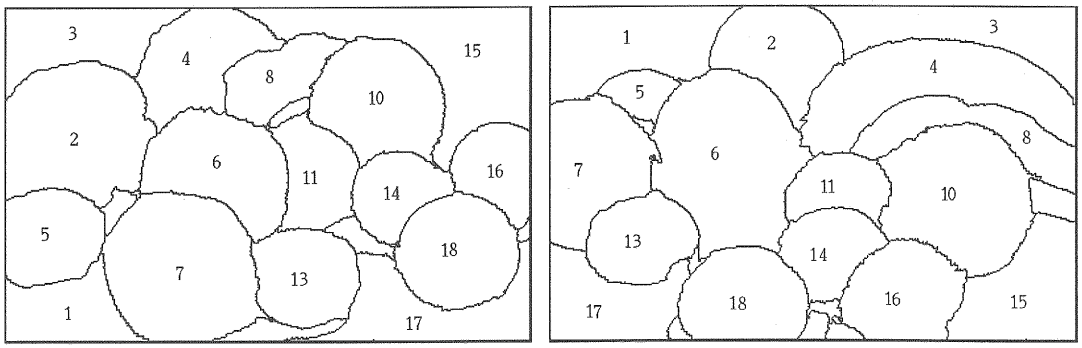
Steps 1—4 are repeated until the results of matching obtained in step 4 do not change.

3.5 Experimental Results

Matching results of the above method for two pairs of images are shown in Figure 3.5 and Figure 3.6. Figure 3.5(a) shows two input color image of fruits, and the results of region segmentation and correspondence of the input images are shown in Figure 3.5(b). The same number indicates the regions matched each other. Figure 3.6(b)

shows the experimental result on another pair of color images of toys in Figure 3.6(a). In the left image of the matching results, the regions numbered by a same number which has distinct subscripts of lower case letters of alphabet, like 11_a and 11_b, shows that more than one regions in the left image correspond to a same region in the right image.

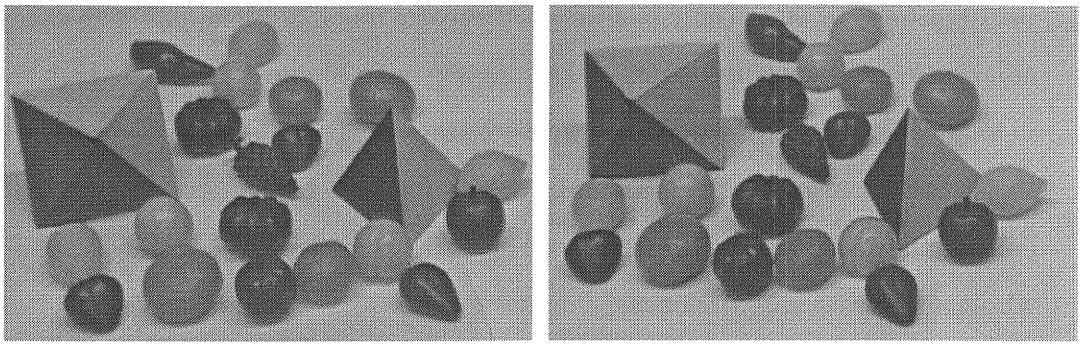
a)Input images (only bright values are shown)



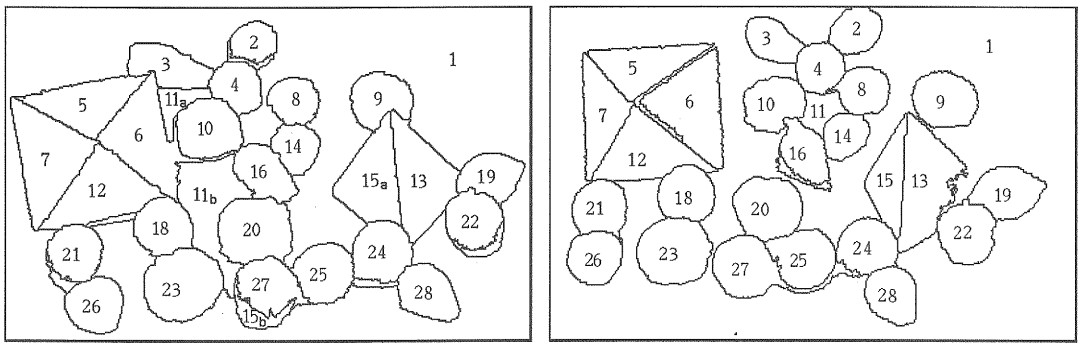
b) Results of region segmentation and matching

Figure 3.6: Region matching results for the images of fruits

The parameters which are required by the matching algorithm are set to the followings in the experiments. T_0, the threshold for the area of regions which can be merged in the first step of region merging (see Section 3.2), is set to 100 for both images in Figure 3.6 and 3.7. T_1, another threshold of region area used in the second step of region merging, is selected as 4000 and 800 for the images in Figure 3.6 and



a) Input images (only bright values are shown)



b) Results of region segmentation and matching

Figure 3.7: Region matching results for the images of toys

3.7, respectively. T_c, the threshold of chromatic similarity for selection of matching candidates (see Section 3.4.1), is selected 0.07 for both images in Figure 3.6 and 3.7. The times required for matching process are 9.5 seconds 44.8 seconds (user time on SUN SPARC-10) for the images in Figure 3.6 and 3.7, respectively.

Although the matching algorithm worked fairly well in these cases, due to the robustness of our ternary relations compared with binary relations such as region adjacency, there are still some mistakes. As shown in Figure 3.7, regions 11_a and 11_b in the left image corresponded incorrectly to region 11 in the right image, due to the inconsistency of region segmentations in two images. In the next chapter, we will consider how to detect and correct these mistakes using the global consistency of the whole obtained correspondences.

CHAPTER 4

The Global Consistency of Region Matching

4.1 Introduction

In the region correspondences obtained by the region matching method described in the previous chapter, although there may be some mistakes due to occlusion of regions or inconsistency of region segmentation between two images, as shown in Figures 3.6 and 3.7, but the majority of them can be expected to be correct. In this chapter, We will describe a method of detecting and correcting the mistaken ones by using the global consistency of the whole correspondences of regions.

Based on the assumption that the scene does not change while the camera has moved, if the correspondences are correct, then the locational shifts between the corresponding regions in two images can be regarded as the movements of the objects in the image plane due to the camera movement. It means that these locational shifts should follow the optical flow generated by the camera movement. The locational shifts of the center of gravity of regions obtained from the region correspondences for images in Figure 3.7 are shown in Figure 4.1. From the figure we can observe that the

locational shifts of the correct correspondences are rather regular in the optical flow while those of the incorrect correspondences violate the whole regularity apparently.

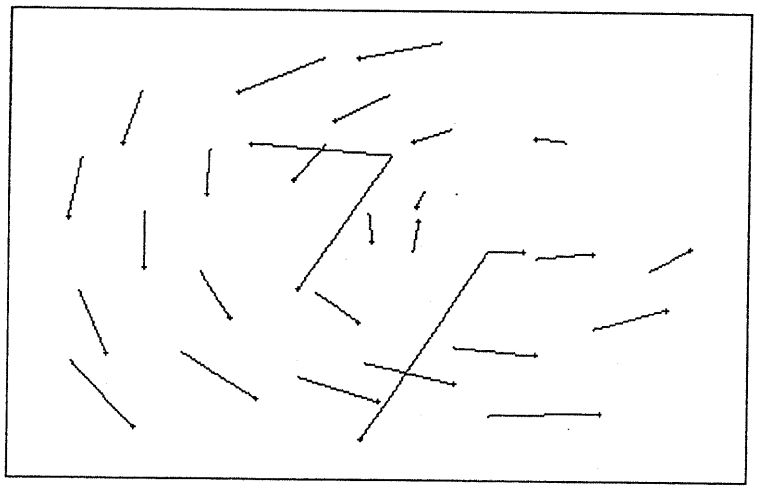


Figure 4.1: Locational shifts of regions

Under the hypothesis that most of the obtained correspondences are correct, we can estimate the parameters of the camera movement using all the obtained correspondences. Then we check each correspondence with the estimated parameters of the camera movement and judge a correspondence to be a mistake if its locational shift in two images is far from the one predicted from the estimated parameters.

4.2 Estimation of the Parameters of Camera Movement

In principle, the parameters of camera movement can be estimated from the correspondences of several points in two perspective images, provided the correspondences are known. There are many methods proposed for this purpose, and they may be classified into two categories: continuous approach[56-60] and discrete approach[51-55]. With the continuous approach, small inter-frame motion and smooth optical flow fields are required, and the motion parameters are analyzed in terms of velocity. With the discrete approach, the motion parameters are analyzed in terms of the displacements

of the corresponding points in two images, which are allowed to be relatively large. In our application, it is supposed that the displacements of the corresponding regions in two images may be large, so the discrete approach is more appropriate for our objective.

Given the exact correspondences of eight points in two images, the motion parameters can be computed uniquely from the point correspondences; that is known as *eight point algorithm*[51-53]. However, in the practice such as our case, there are usually more than eight correspondences of feature points in presence of noise, so the computed motion parameters are different depending on which eight feature points are used. There are many methods[51-55] proposed to deal with this general case of motion parameter estimation and a general treatment can be found in the book by Kanatani[50].

4.2.1 Model of Camera Movement

As shown in Figure 4.2, take an XYZ-coordinate system fixed to the camera in such a way that the origin O corresponds to the center of the lens, which is called as the viewpoint, and the Z-axis corresponds to the optical axis of the camera. Suppose that the camera first rotated around the X, Y and Z axes by α, β and γ degrees, respectively, and then translated along X, Y and Z by $-\Delta X$, $-\Delta Y$, $-\Delta Z$, respectively.

Equivalently, we can think that the scene, which is assumed to be static, moves rigidly in the opposite direction relative to the camera. It is know[54] that the three dimensional rigid motion of the scene is equivalent to a rotation by an angle θ around an unit vector $l = (l_1, l_2, l_3)$, followed by a translation ΔX, ΔY, ΔZ. The rotation

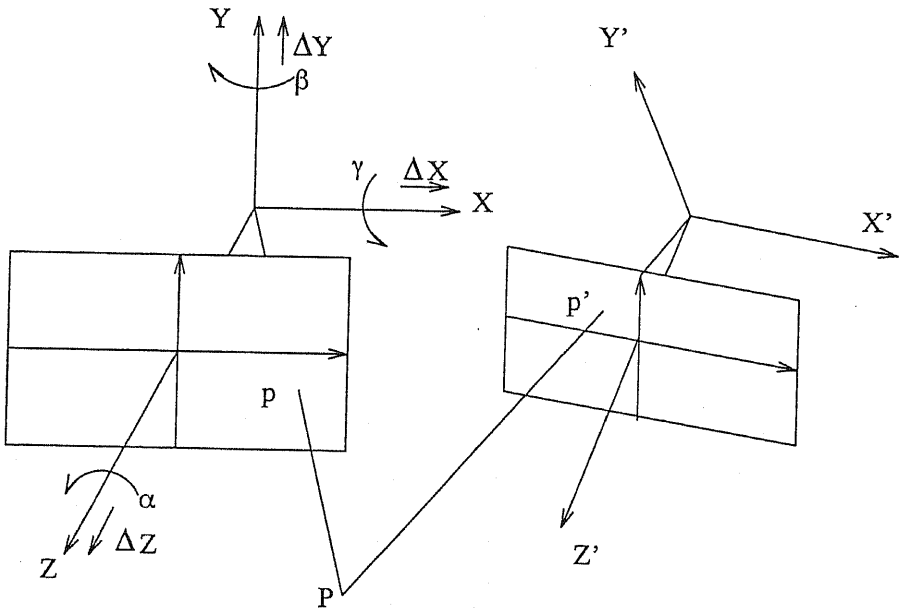


Figure 4.2: Model for camera moving

can be represent by a rotation matrix R which is given by

$$R = \begin{pmatrix} \cos\theta + l_1^2(1-\cos\theta) & l_1 l_2(1-\cos\theta) - l_3\sin\theta & l_1 l_3(1-\cos\theta) + l_2\sin\theta \\ l_2 l_1(1-\cos\theta) + l_3\sin\theta & \cos\theta + l_2^2(1-\cos\theta) & l_2 l_3(1-\cos\theta) - l_1\sin\theta \\ l_3 l_1(1-\cos\theta) - l_2\sin\theta & l_3 l_2(1-\cos\theta) + l_1\sin\theta & \cos\theta + l_2^2(1-\cos\theta) \end{pmatrix} \tag{4.1}$$

Let (X, Y, Z) be the scene-space coordinates of a point P before the motion, and (X', Y', Z') be the coordinates of P after the motion, then the motion can be represented by

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T \tag{4.2}$$

where T is called as the translation vector which is defined as following

$$T = \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix}$$

In our application, instead of the rotation angles, the rotation matrix is employed to describe the rotation, because the rotation matrix is sufficient for the purpose of estimating the optical flow generated by the camera movement. The rotation matrix will be denoted as follows:

$$R = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$$

4.2.2 Estimating the Parameters of Camera Movement

Let (x_i, y_i) and (x_i', y_i') be the image points projected from (X, Y, Z) and (X', Y', Z'), respectively. We can show that (x_i, y_i) and (x_i', y_i') must satisfy the following relation (see Appendix 1), which is equivalent to the epipolar constraint.

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} T_{\times} R \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \tag{4.3}$$

where R and T are the rotation matrix and translation vector defined in the previous section, respectively, and $\times$ stands for vector production. That is, $T_{\times} R$ is defined as

$$T_{\times} R = \begin{pmatrix} 0 & -\Delta Z & \Delta Y \\ \Delta Z & 0 & -\Delta X \\ -\Delta Y & \Delta X & 0 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \tag{4.4}$$

If we treat $T_{\times} R$ as a matrix E as defined below:

$$\begin{pmatrix} 0 & -\Delta Z & \Delta X \\ \Delta Z & 0 & -\Delta Y \\ -\Delta X & \Delta Y & 0 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} = E \qquad (4.5)$$

then Equation (4.3) can be rewritten as a linear equation of the elements of matrix E as

$$x'xe_1 + x'ye_2 + x'e_3 + y'xe_4 + y'ye_5 + y'e_6 + xe_7 + ye_8 + e_9 = 0 \qquad (4.6)$$

The matrix E is called the *essential matrix*.

It is obvious that if the exact correspondence of eight points between the images before and after camera movement are known, a nonzero solution of e_i's, $i = 1, ..., 9$ can be determined up to a scale by solving the linear system of Equation (4.6). Furthermore, if six points among the eight points in the scene are not coplanar, the nonzero matrix E can be decomposed uniquely to a rotation matrix and a translation vector by singular value decomposition (SVD) of E[52,53].

However, in the case of region correspondence, the centers of gravity of the pairs of matched regions are used as the correspondence points. The correspondence of those centers of gravity of the matched regions may not be the exact one, because the displacement of the center of gravity of a region between two images is determined not only by the movement of the scene relative to camera, but also due to the inconsistency of region segmentation in two images or a partial occlusion of the region in one image. Moreover, some mistakes are also contained in the obtained region correspondences. Therefore, a robust method to estimate camera movement parameters from the correspondences in presence of noise is required.

In our application, we have usually much more than eight correspondences of regions which can be used. For the majority of those correspondences, which are guaranteed to be correct by the matching method described in the previous section, although the correspondences of their centers of gravity may not satisfy Equation

(4.6), they can be expected to deviate from it by errors with mean of zero. Thus the least-square method can be used to compute the essential matrix E using the correspondences of centers of gravity of the pairs of matched regions. That is, letting $\{(x_i, y_i)|i = 1, ..., n\}$ and $\{(x'_i, y'_i)|i = 1, ..., n\}$ be the centers of gravity of region in two images, respectively, and supposing that (x_i, y_i) corresponds to (x'_i, y'_i), compute E by

$$\min \sum_{i=1}^{n} (x'_i x_i e_1 + x'_i y_i e_2 + x'_i e_3 + y'_i x_i e_4 + y'_i y_i e_5 + y'_i e_6 + x_i e_7 + y_i e_8 + e_9)^2$$

Due to the noise in correspondences of points, there is no guaranty that the obtained matrix E is exactly an essential matrix, that is, it can be decomposed completely into a rotation matrix and a translation vector. Therefore, the residual decomposition of E proposed by J. Weng[54] is used to compute the rotation matrix R and the translation vector T from the obtained E.

First, from the definition of the essential matrix, we can obtain:

$$\begin{aligned} E^T \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} &= \left[\begin{pmatrix} 0 & -\Delta Z & \Delta Y \\ \Delta Z & 0 & -\Delta X \\ -\Delta Y & \Delta X & 0 \end{pmatrix} R \right]^T \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \\ &= R^T \begin{pmatrix} 0 & \Delta Z & -\Delta Y \\ -\Delta Z & 0 & \Delta Y \\ \Delta Y & -\Delta Y & 0 \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \end{aligned} \tag{4.7}$$

That is, the translation vector must be orthogonal with the vectors which are

composed of the elements in three columns of the essential matrix E.

$$\begin{pmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} = 0 \tag{4.8}$$

The matrix E obtained by the least-square method was proved[54] to have rank 2. Therefore, we can determine the direction of T by Equation (4.9) using the matrix obtained.

Next, the rotation matrix R is computed by minimizing the error between obtained matrix E and the essential matrix defined in (4.4), which is represented by the vector product of the translation vector with the rotation matrix, where R is subjected to be rotation matrix.

$$min \left\| \begin{pmatrix} 0 & -\Delta Z & \Delta X \\ \Delta Z & 0 & -\Delta Y \\ -\Delta X & \Delta Y & 0 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} - \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \right\| \tag{4.9}$$

Finally, the relative depths Z of the points of each correspondence can be computed by minimizing the following.

$$\min \left\| Z' \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} - ZR \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - T \right\| \tag{4.10}$$

4.3 Detection of Mistaken Correspondences

In this section, we consider how to detect the mistaken correspondences after the parameters of camera movement have been estimated. The idea is to validate the displacement between the centers of gravity of each individual correspondences of regions in two images with the camera movement parameters estimated from all

the correspondences. If the displacement of a correspondence of two regions is not coherent with the camera movement, then this correspondence is regarded as mistaken matching. Two approaches to do this are described in the following.

4.3.1 Error in Image Plane

When the parameters of camera movement are known, for a given points in one image, the range of the location where the corresponding point should be in the other image can be predicated from the parameters of camera movement. As shown in Figure 4.3, if a point (x, y) in one image has a correspondence (x', y') in the other image, then (x', y') should be on the line segment from (x'_0, y'_0) to (x'_1, y'_1).

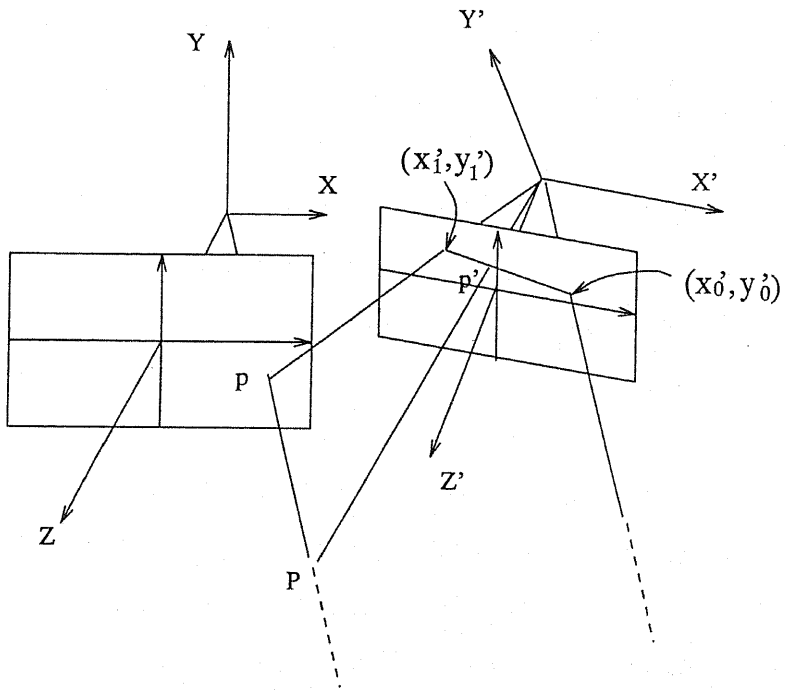


Figure 4.3: Corresponding segment on Epipolar line

The points (x'_0, y'_0) and (x'_1, y'_1) are determined by the parameters of camera movement as follows:

$$\begin{pmatrix} x'_0 \\ y'_0 \end{pmatrix} = \begin{pmatrix} (r_1x + r_2y + r_3)/(r_7x + r_8y + r_9) \\ (r_4x + r_5y + r_6)/(r_7x + r_8y + r_9) \end{pmatrix}$$

If $\Delta Z \geq 0$, then

$$\begin{pmatrix} x_1' \\ y_1' \end{pmatrix} = \begin{pmatrix} (x_0' + \Delta X)/(1 + \Delta Z) \\ (y_1' + \Delta Y)/(1 + \Delta Z) \end{pmatrix}$$

otherwise

$$\begin{pmatrix} x_1' \\ y_1' \end{pmatrix} = \begin{pmatrix} x_0' + \Delta X \\ y_1' + \Delta Y \end{pmatrix}$$

Let (x_i, y_i) and (x_i', y_i') are the centers of gravity of two corresponding regions, (x_i', y_i') should lie near the line segment from (x_0', y_0') to (x_1', y_1'), which is determined by the point (x_i, y_i), and the parameters of camera movement R and T, as shown in the above. So the error of the correspondence can be computed as the distance from (x_i', y_i') to that line segment. The distribution of the errors of the correspondences computed in the image plane can be expected to be bimodal, because the correct correspondences should have small error but the mistaken ones will have large error which appear as the outliers in the whole distribution.

Table 4.1 shows the error in the image plane for each correspondence of regions in the region corresponding results shown in Figure 3.7. We separate the outliers from the valid correspondences using the thresholding method proposed by Kittler and Illingworth[61]. The threshold value computed is also shown in the table. It can be seen that the mistaken correspondences for region 11_b and 15 can be detected as mistaken correspondences, because the errors in the image plane of them are larger than the threshold computed. However, the mistaken correspondence for 11_a could

not be detected by this method.

Table 4.1: Error of each correspondence in the image planes under the estimated parameters

Corresp.	Error	Corresp.	Error	Corrsp.	Error
1	0.0207	11_a	0.0030	20	0.0044
2	0.0028	11_b	0.0338	21	0.0199
3	0.0000	12	0.0156	22	0.0007
4	0.0000	13	0.0003	23	0.0162
5	0.0221	14	0.0006	24	0.0095
6	0.0122	15_a	0.0029	25	0.0133
7	0.0254	15_b	0.0804	26	0.0179
8	0.0003	16	0.0011	27	0.0132
9	0.0007	18	0.0092	28	0.0123
10	0.0016	19	0.0016	Threshold : T=0.030	

4.3.2 Error in Parameter Estimation

Another way to detect mistaken correspondences is based on an analysis of the error in the estimation of camera movement parameters. It is obvious that there will be some error in the estimation of the camera movement parameters from their true values of them. The error may arise for two reasons. The first reason is that even if a given region in one image is corresponded correctly to a region in the other image, there will be some error in the displacement of centers of gravity of the corresponded regions, due to the inconsistency of region segmentation in the two images or lightly partial occlusion of the region in one image. These correspondences cause error in the epipolar constrain Equation (4.6) which is used to estimate the essential matrix,

but the error of this type can be expected to be smaller than the errors from the second reason below.

The second reason is that there may be some mistakes in the obtained region correspondences. These mistaken correspondences are not consistent with the camera movement, so they will contribute a lot to the error in the parameter estimation. Therefore discarding a mistaken correspondence will let the estimated parameters draw nearer to the true values than discarding a correct one. This leads to the method of detecting mistaken correspondences described in the following.

For each correspondence of regions, the parameters of camera movement are reestimated by discarding it from the whole correspondences, and the perturbation of the reestimated parameters from the old ones is computed. Let P_i be the value of the perturbation of the parameters estimated by discarding the ith correspondence, then the set $\{P_i\}$ will be bimodal, one consisting of smaller values related to discarding a valid correspondence and the other consisting of larger values related to discarding an outlier. Table 4.2 shows the perturbation of the reestimated parameters for each correspondence of regions in the region correspondence results shown in Figure 3.7.

Similar to what was done in the first method, the outliers in the perturbations of estimated parameters by discarding a single correspondence can be separated by thresholding. The threshold computed by the method of Kittler and Illingworth is also shown in Table 4.2. In this table, we can see that all the mistaken correspondences are detected, but at the same time the correct correspondence of region 1 and 5 is also

overdetected.

Table 4.2: Perturbations of estimated parameters by discarding each correspondence in Fig. 4(a).

Discarded Correspondence	Perturbation in parameters	Discarded Correspondence	Perturbation in parameters
1	0.0576	15_a	0.0024
2	0.0133	15_b	0.3040
3	0.0134	16	0.0025
4	0.0015	18	0.0003
5	0.0382	19	0.0031
6	0.0069	20	0.0027
7	0.0002	21	0.0007
8	0.0049	22	0.0097
9	0.0130	23	0.0019
10	0.0009	24	0.0035
11_a	1.7139	25	0.0066
11_b	0.1016	26	0.0058
12	0.0026	27	0.0077
13	0.0001	28	0.0241
14	0.0053	Threshold: T = 0.030	

It seems that the second method yields overdetection while the first one does underdetection. We prefer the second method because the overdetected correspondences can be reverified in the correction stage.

4.4 Correcting Mistaken Correspondences

For each outlier of the correspondences detected as above, correction is carried out as follows.

First, the estimated optical flow of each correspondence is computed using the estimated parameters of camera movement and the estimated relative depths. For the outliers detected, the depths are substituted by the average depth. The estimated optical flow for Figure 3.7 is shown in Figure 4.4.

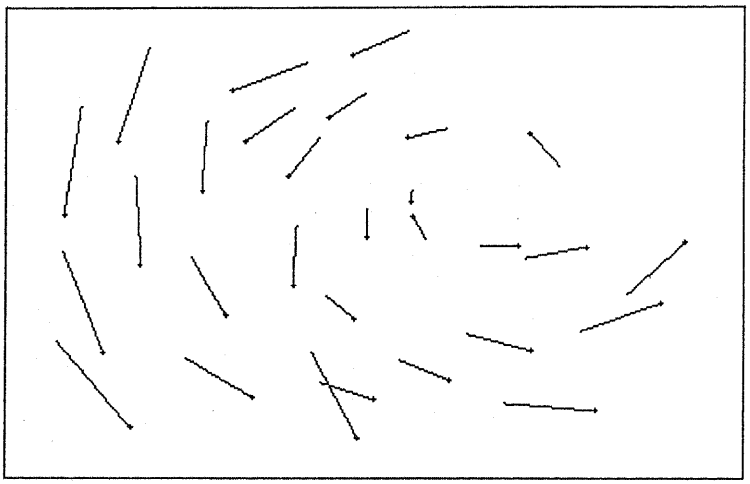


Figure 4.4: Estimated optical flow

For a region A_i whose correspondence is judged as an outlier, if there are some other corresponding candidate B_{i_k} are consistent with the estimated optical flow of A_i, then we select the one which has the highest matching similarity with A_i as the new correspondence of A_i. We say that B_{i_j} is consistent with the estimated optical flow of A_i if the vector from the center of gravity of A_i to that of B_{i_j} lies in an area from $-10°$ to $10°$ relative to the direction of the estimated optical flow, and the length of the vector is within $\frac{2}{3}$ to $\frac{3}{2}$ of the length the estimated optical flow.

If an outlier of correspondences could not be corrected in the step above, we try to use the ternary relation of regions to correct it. For a region A_i whose correspondence is judged as an outlier, from its neighbors select two nearest regions

A_i^1, A_i^2 whose correspondences are validated in the validating step. By mapping A_i^1 and A_i^2 to their correspondences and the triangle, which consists of the centers of gravity of A_i, A_i^1 and A_i^2, simultaneously onto the other image, we can estimate the approximate location $(\hat{x}_i, \hat{y}_i)$ where the center of gravity of the corresponding region of A_i should be. We find regions which are similar to A_i in color, within a disk with the center locating at $(\hat{x}_i, \hat{y}_i)$ and the radius equal to half the length of the longer of the two edges from $(\hat{x}_i, \hat{y}_i)$ of the mapped triangle. Among these regions, the one has the biggest area in the disk is assigned as the correspondence of A_i.

A region A_i whose correspondence is an outlier which could not be corrected in the above steps is judged to have no correspondence.

The corrected correspondences of Figure 3.7 are shown in Figure 4.5, where the outliers are checked out with the second method. A '0' means that no corresponding region exists in the other image. The overdetected correspondence are reverified in the first step.

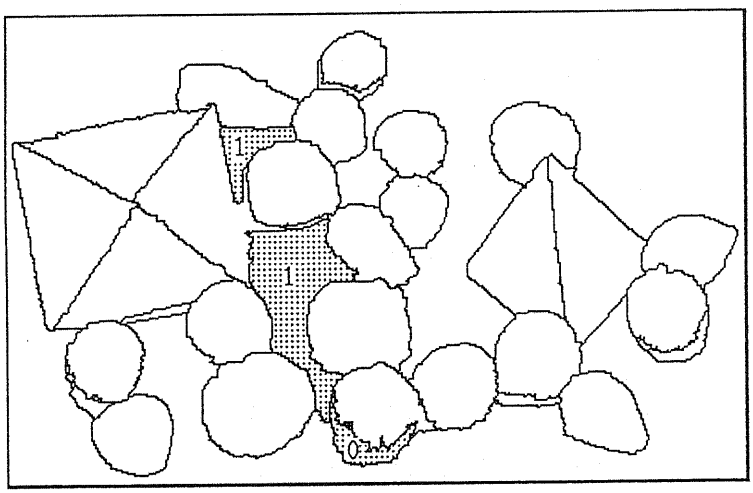


Figure 4.5: Corrected correspondences(for the left image)

4.5 Summary

we propose a robust region matching method which utilizes both local region structure similarity and global consistency of the region matchings. Our approach adopted a two-stage matching strategy, that is, a matching stage followed by a validating and correcting stage. Region correspondences are first established using both feature similarity and local structure similarity of regions. The local structure are represented by ternary relations among regions, which was shown to be stable and discriminant in local region configurations.

Also using local structure similarity will improve the region correspondence results so that the majority of them are guaranteed to be correct, but there may be some mistaken correspondences still. In the second stage, the optical flow generated by the camera movement is estimated from all the obtained correspondences and is used to detect and correct some apparent mistakes in the matching results. Experimental results show the effectiveness of our method.

CHAPTER 5

Region Correspondence by Inexact Attributed Planar Graph Matching

5.1 Introduction

In this chapter, we will attempt a syntactic representation of regions and their relations in the images and try to match regions in two images based on the syntactic representations of them. The syntactic approach to region matching should be encouraged because a syntactic representation of the regions in images carries both the local features of each region and the global relations of the regions. Region matching on a syntactic representation can achieve the global stability of matchings. Furthermore, a syntactic representation of regions in image is more similar to the representation of models, such as aspect representation. Therefore, the syntactic approach has the potentiality of the extension for object location, identification and recognition.

Chen and Nakatani[33] tried an approach of region matching using region structure similarity. The region structure is represented by region tables, in which the header (0-th) item contains a region which is concerned and the ith item contains a list of regions which are away from the concerned region in distance i (that is, they

are separated from the concerned region by $i-1$ regions in between). Obviously, the region table is not an inherent way to describe the relations of regions.

Another more natural way is to describe the relational structure of regions with an attributed graph, and then find the optimal inexact matching of two attributed graphs[62-65]. Many approaches such as state space search[62-65] are proposed to deal with the the problem of inexact matching of general attributed graphs, but they are generally computationally expensive. Chen[74] simplified the state space searching method of Fu[65] by finding the optimal spanning tree matching between two graphs, but there the relations between the son nodes descendant from the same parent node are ignored.

In this chapter, we propose an efficient graph matching approach for finding region correspondence of images. In this approach, we consider the topological relation of region adjacency, and represent the regions and their relations in an image with a Region Adjacency Graph(RAG). In the images which are taken from a same scene from different viewpoints, although the region and their relations in the images are not completely consistent with each other due to the change of viewpoint and the inconsistency of the region segmentation results in the images, but we can expect that the relations of the regions are still similar. Thus the optimal region correspondences between two images can be considered to be the correspondences of regions with maximal region similarity and region adjacency similarity. This is equivalent to finding the optimal inexact matching of two RAGs.

Although the computationally expensive approaches proposed for the general graph matching can be applied to inexact matching of RAGs, but we noticed that a RAG is a special kind of graph -- an attributed *planar* graph, and that the relation of region adjacency assigns some specific properties. Utilizing these properties specific to RAGs, we can construct an efficient algorithm to find the optimal matching between two RAGs.

In the next section, we give an operational definition of inexact matching of

RAGs that takes region merging into account, and formulate region correspondence into inexact matching of attributed *planar* graphs. In Section 5.3, we mention some properties which are specific to planar graphs and topological adjacency relation of regions. In Sections 5.4 -- 5.6, an efficient algorithm to search for the optimal inexact matching of two RAGs is constructed based on these properties. Some experimental results of our method are shown in Section 5.7.

5.2 Description of the Problem

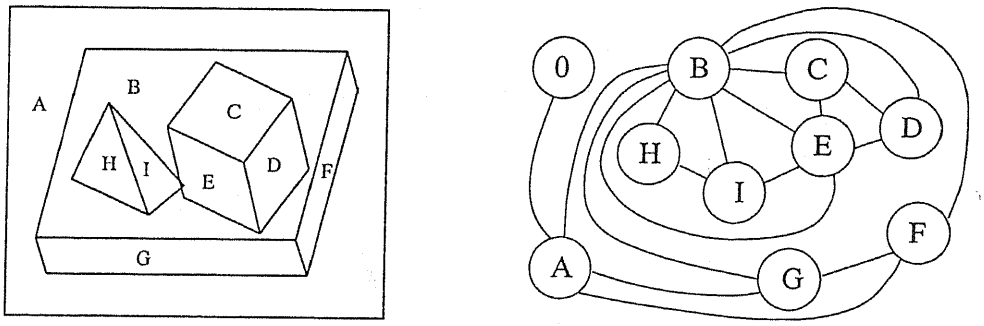
The regions and their adjacency relations in an image can be described as an attributed graph denoted as

$$G = \{\mathcal{N}, F, \mathcal{E}, H\}$$

where $\mathcal{N} = \{N_i\}$ is the set of nodes which correspond to regions in the image, $F = \{(f_1(N_i), ..., f_m(N_i))\}$ denotes the set of feature vectors which characterize the regions, $\mathcal{E} = \{E_j\}$ is the set of edges between nodes which represents the region adjacency, and $H = \{(h_1(E_j), ..., h_n(E_j)\}$ denote the set of feature vectors which characterize the edges. Hereafter we denote the edge connecting nodes N_i and N_j in terms of node pair (N_i, N_j).

We refer to the graph which represents the regions and their adjacency relations in an image as the *region adjacency graph*, or as the RAG simply. A RAG is a planar graph because we can embed it in a plane without edge crossing, by drawing each node in its corresponding region and edges passing through the boundary between the corresponding (adjacent) regions in the image plane. An example is shown in Figure 5.1.

In general, a complete consistent matching(or isomorphism) of two RAGs is impossible, because change of viewpoint and inconsistent segmentations in two images may cause occlusion of regions, discrepancy of region features and inconsistency of adjacency relations of regions. So an inexact matching between two



(a) Regions in an image (b) Its RAG

Figure 5.1 A planar graph describing regions and their adjacent relations(0 stands for the frame of the image).

attributed planar graphs must be considered.

Definition 1 An *inexact matching* of two attributed planar graphs G and G' is an isomorphism of two attributed planar graphs which are obtained from G and G' by performing a sequence of graph transforming operations such as node substitutions, node deletions, node mergings, edge substitutions and edge deletions.

In contrast to the existing definitions[62-66], our definition does not include the operations of node insertion and edge insertion, but we perform the operations of node deletion and edge deletion on *both* graphs, instead of performing all the operations on only one graph. It is clear that our definition is equivalent to the former ones because inserting a 'null' node or 'null' edge in one graph is equivalent to deleting it from the other. One more difference of our definition from the others is that the merging of regions is taken into account, as a single region in one image may be segmented into two or more regions in the other image.

For attributed RAGs, the graph transforming operations can be described as follows:

1. Node substitution. Substituting a node N_i in graph G by a node N'_k in G' (or the reverse) means matching region N_i with region N'_k.

2. Node deletion. Deleting a node N_i (or N'_k) from graph G (or G') means N_i (or N'_k) does not appear in the other image due to an occlusion or some other reason.

3. Node merging. Only two nodes N_i and N_j having an edge between them (i.e., adjacent) can be merged into one region. Merging N_i and N_j also means substituting them by a node N'_k in G' at the same time.

4. Edge substitution. When two node pairs (N_i, N_j) and (N'_k, N'_l) are matched, the edge between N_i and N_j should be substituted by the edge between N'_k and N'_l (or the reverse) if both edges exist.

5. Edge deletion. When two node pairs (N_i, N_j) and (N'_k, N'_l) are matched, but there exists an edge only in one graph, it should be deleted.

As the edges in the RAGs represent region adjacency, deleting an edge will cause an insertion of another edge in the graph, as shown in Figure 5.2.

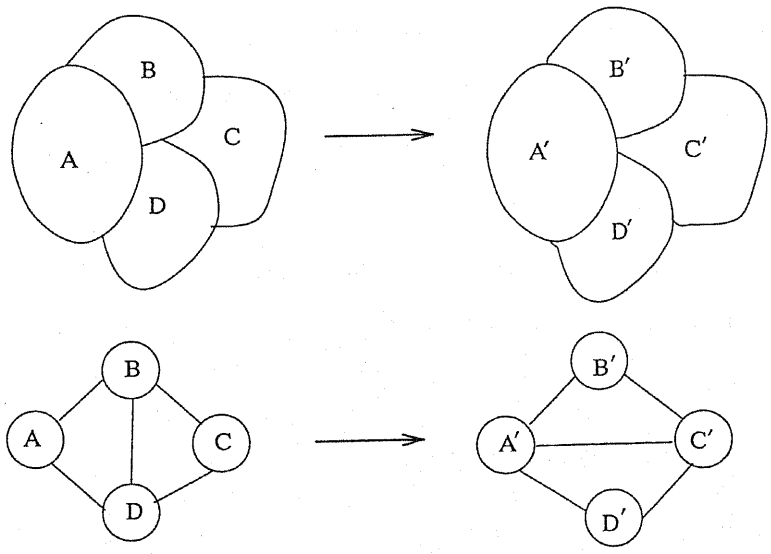


Figure 5.2 Effect of edge deletion

Generally, deletion and substitution of edges also happen in node merging and node deletion.

1. When two nodes N_i and N_j are merged (say that N_j is merged into N_i), the edge between them should be deleted, and all the edges $(N_j, *)$ should be substituted by $(N_i, *)$.

2. When a node N_i is deleted, the edges connected with it should also be processed. In our method, we assume that node deletion happens when a region is occluded by another region N_j, so all the edges $(N_i, *)$ should be replaced with $(N_j, *)$.

Figure 5.3(a) shows the cases of region deletion. Suppose that the small region X vanishes in the other image, then the relations of the regions will change, that is also shown in Figure 5.3(a). The corresponding graph transforming operations on the graphs are shown in Figure 5.3(b).

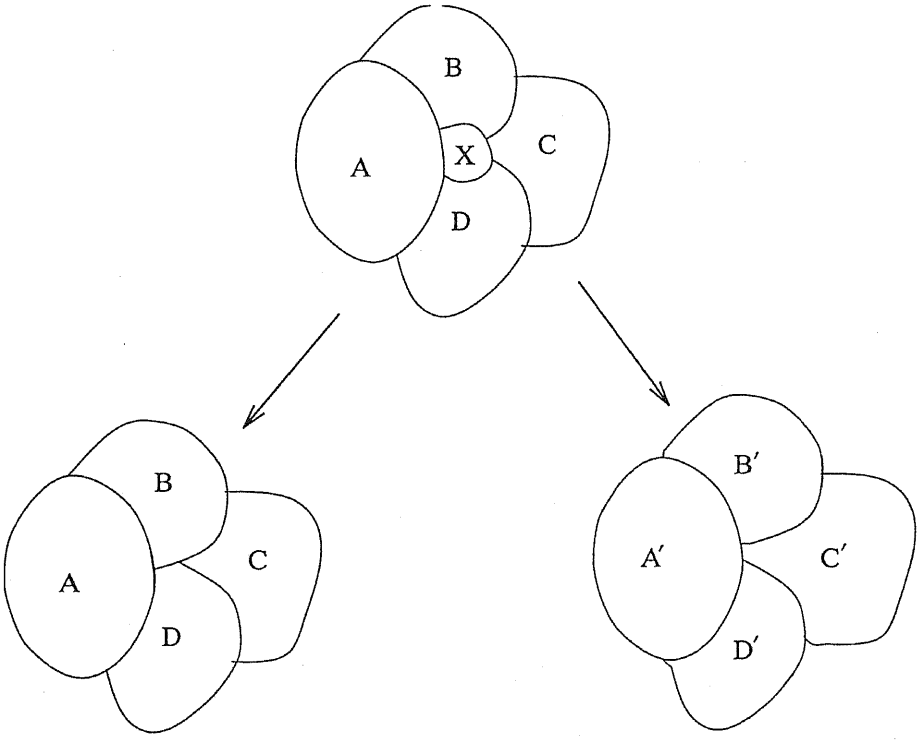
Two cases of node merging as well as their corresponding graph transforming operations on the graphs are shown in Figure 5.4 and Figure 5.5. The two cases of node merging seem different with respect to A. However, they are just equivalent when we take nodes D in Figure 5.4 and A in Figure 5.5 as concerned nodes, respectively.

For each operation performed on the graphs, a cost can be associated with it. For an inexact matching of two attributed planar graphs, its cost can be computed by summing up the cost of all the graph transforming operations which are performed on both graphs to convert them into two isomorphic graphs.

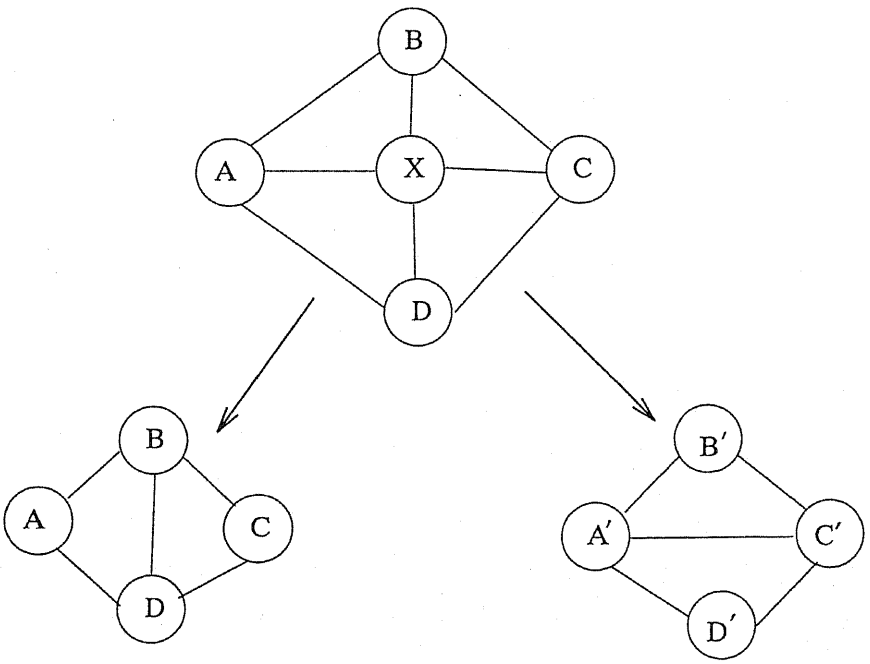
Definition 2 The *optimal inexact matching* of two attributed planar graphs is the inexact matching which has the minimal matching cost.

5.3 Properties of RAG

For convenience of illustration, we first mention necessary terminologies from graph theory and some properties of planar graphs[67]. Some special features of region adjacency relations are also stated.



a) Region deletions



b) The graph transforming operations

Figure 5.3 Two cases of deleting a region

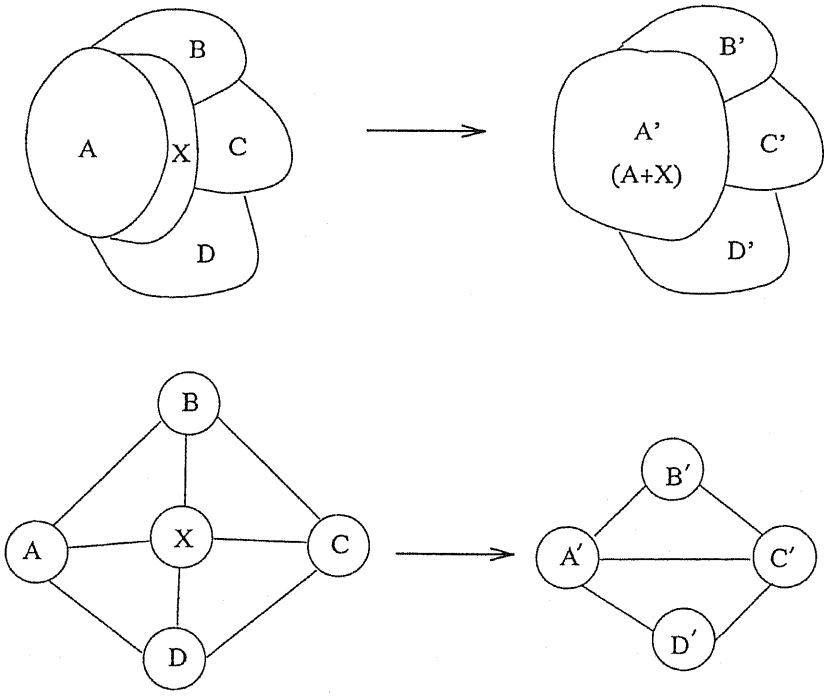


Figure 5.4 Region merging: case 1.

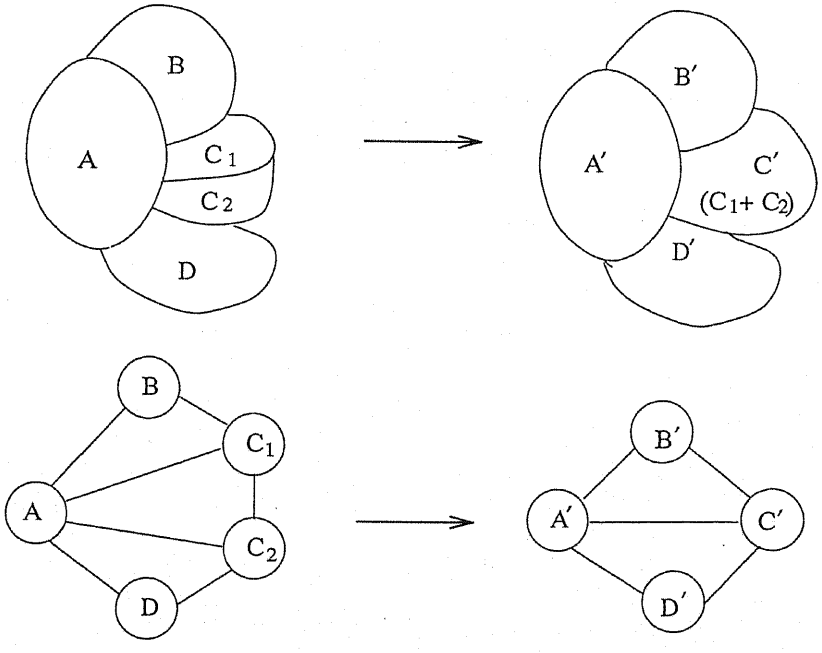


Figure 5.5 Region merging: case 2.

Articulation node If there is a triple of distinct nodes N_1, N_2, N_3 in graph G such that N_3 lies on every path which connects N_1 and N_2, then N_3 is an articulation node of G.

Biconnected graph A graph G is biconnected if for each triple of distinct nodes N_1, N_2 and N_3 in G, there exists a path between N_1 and N_2 such that N_3 is not on this path.

Biconnected component A biconnected component in a graph is a maximal subgraph which is biconnected.

When the regions and their adjacent relations are considered, a region N_i appears as an articulation node if and only if N_i is a region which encircles one or more connected sets of regions. For instance, in Figure 5.1 region B encloses two sets of regions $\{C, D, E\}$ and $\{H, I\}$. In this situation, any region in these sets can be connected only through N_i to the regions in the other sets or the regions which lie outside of N_i (e.g. A, F, G in Figure 5.1).

An attributed planar graph can be partitioned into the biconnected components and a tree; the latter consists of the articulation node and the virtual nodes which represent the biconnected components. In this partition, an articulation node (e.g., B in Figure 5.6) appears not only in the tree, but also as a node in the biconnected components (e.g., V_2 and V_3) which are encircled by the articulation region, and in the biconnected component (e.g., V_1) which contains the articulation nodes. In a biconnected component, only the articulation node have edges connecting with the nodes outside the component. The root of the tree is assumed as the node which corresponds to the frame of the image (node 0). The partition of the graph in Figure 5.1 is shown in Figure 5.6.

A biconnected component can further be partitioned into triply connected components biconnected by biarticulation nodes, which are defined as follows.

Biarticulation node If there is a quadruple of distinct nodes N_1, N_2, N_3 and N_4 in a graph G such that every path between N_1 and N_2 passes through either N_3 or

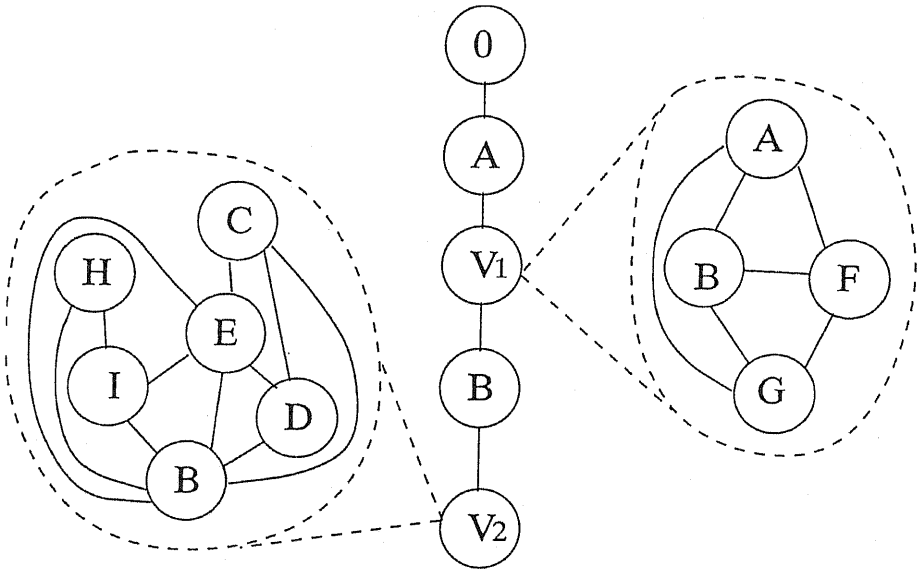


Figure 5.6: The partition of the graph in Figure 5.1

N_4, then N_3 and N_4 are biarticulation node pair of G.

Triply connected graph A graph G is triply connected if for each quadruple of distinct nodes N_1, N_2, N_3 and N_4 in G, there is a path between N_1 and N_2 such that neither N_3 nor N_4 is on this path.

Triply connected component A triply connected component in a graph is a maximal subgraph which is triply connected.

A planar graph can be drawn in a plane without edge crossing. Drawing a planar graph in a plane without edge crossing is called its *embedding* in a plane.

Property 1 A triply connected planar graph has exactly two embeddings in a plane. One of the embeddings is obtained from the other by reversing the order of all edges around each vertex[67], that is, they are mirror images each other.

In region matching, if we exclude the special case that one image is a mirror image of the other, then each triply connected planar component has only one embedding in a plane. This fact implies that if a node in a triply connected component in a RAG is matched to a node in a triply connected component of the other RAG, then their edges (and the nodes attached in the other end of the edges) arranged in the

image planes can only be matched in the same order. That is, suppose that a node N in a triply connected component of G is matched to a node N' in a triply connected component of G', and let $\{(N, N_i)\}$ and $\{(N', N'_j)\}$ are the sets of edges emanating from N and N' in the triply connected components, subscripted clockwise on the image planes, respectively, then the following rule must be satisfied.

$$N_i \leftrightarrow N'_j \Longrightarrow N_{i+1} \leftrightarrow N'_{j+k}$$

where $\leftrightarrow$ denotes a match and $k > 0$. N_{i+1} stands for the edge next to N_i in clockwise order. N'_{j+k} means an edge in the frontward of N'_j in clockwise order. We will call this rule *order reserving*.

For the biarticulation nodes in RAGs, it can also be shown that if two biarticulation nodes are matched in the graph isomorphism, then the edges emanating from them can only be matched with order reserving.

Property 2 Let (N, N_1) and (N, N_2) be two neighboring edges of node N in a biconnected component of a RAG, then there must exist an edge between N_1 and N_2.

This property is self-evident from the notion of region adjacency. From property 2, we can obtain:

Property 3 If only one embedding of a triply connected component in RAG is considered, that is, the mirror version of the embedding is ignored, then a biconnected components in RAG has only one embedding. (The proof is given in Appendix 1.)

Basic RAG Let N_i be a node in a biconnected component B in G, and let $S(N_i) = \{N_{i_1}, N_{i_2}, ..., N_{i_m}\}$ be the sets of nodes subscripted clockwise around N_i in B. The subgraph which consists of nodes $\{N_i\} \cup S(N_i)$ and the edges between those nodes is called as a *basic RAG(or simply, BRAG)* at N_i (Figure 5.7).

From Property 3, we can obtain:

Property 4 If two biconnected components in RAGs have been made isomorphic, then the edges of matched nodes in their BRAGs must be matched in the same

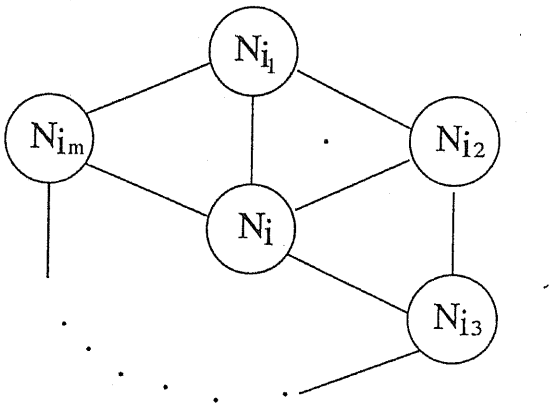


Figure 5.7: A basic RAG

order.

5.4 Framework of the algorithm

From the discussion in the previous section, we can obtain that: an attributed RAG can be partitioned into biconnected components connected by articulation nodes. Regarding a biconnected component as a virtual node, a RAG can be treated as a tree, in which the virtual nodes are connected by the articulation nodes. Matching two RAGs is to build two isomorphic trees in which each biconnected component in one tree is isomorphic to its corresponding one in the other tree. Thus the matching of two RAGs can be considered at two levels: (1) the matching at the tree level where biconnected components are regarded as virtual nodes, and (2) the matching within a biconnected component where the order reserving constraint can be used.

In overview, our algorithm searches for the optimal matching in the following way. First, all the pairs of nodes from two graphs that are similar in feature are identified as the pairs of nodes that may possibly be matched. Then, from each pair, we try to change original graphs into two isomorphic attributed planar graphs by performing the graph transformation operations with the minimal cost. Finally, after

every node pair is tried, the isomorphism with the minimal cost among all is selected as the optimal matching between two attributed planar graphs.

The algorithm consists mainly of two levels, the lower level is to match the nodes in two biconnected components and the upper is to match two biconnected components at articulation nodes which they are connected with. When a pair of nodes from two graphs is selected to be the starting pair, the two isomorphic graphs are first built within the biconnected components which contain the two start nodes. As shown in Figure 5.6, an articulation node also appears in the biconnected components which it connected with, so building two isomorphic subgraph within two biconnected components will results in matching of some articulation nodes involved. These matched pairs of articulation nodes then trigger matching of the remaining biconnected components connected with them. This is done at two levels: a tree matching where a biconnected component are regarded as a virtual node, and the node matching within two biconnected components that is activated by the tree matching. Because a biconnected component in one image may be separated into more than one biconnected components in the other image, the splitting or merging of the virtual nodes must be taken into account.

To match the nodes in two biconnected components, we first build two basic isomorphic planar graphs, each of which is a subgraph consists of the concerned node, the nodes connected to it and all the edges between them, by performing graph transformation operations of the minimal cost on both of the two attributed planar graphs. The two basic isomorphic graphs make some other nodes matched. From these matched nodes, we can build in turn other basic isomorphic planar graphs with the minimal cost. If there is no conflict of node matchings obtained from the basic isomorphic planar graphs, their combination provides two isomorphic planar graphs with the minimal cost. If a conflict occurs, the newly obtained matching is adjusted according to the previous match, as to be explained in section 3.4, and the pair of the concerned nodes of the basic isomorphic planar graphs which generated

the conflicting match is set to a starting pair so that the superseded match can appear earlier in the searching starting from the new starting pair.

Once the matching of the nodes of two biconnected components are completed, we try to match the remaining biconnected components connected with the pairs of articulation nodes which has been matched in the biconnected components. For every such pair, if there is a pair of biconnected components which are the parent nodes of articulation nodes in the tree, then the two biconnected components are tried to be matched. For the other biconnected components, which appear as the child nodes of the articulation nodes in the tree, a tree matching, which recursively invokes the matching of nodes in two biconnected components, is performed.

The matching procedures at the tree level and biconnected component level are described in the following sections.

5.5 Matching at the Tree Level

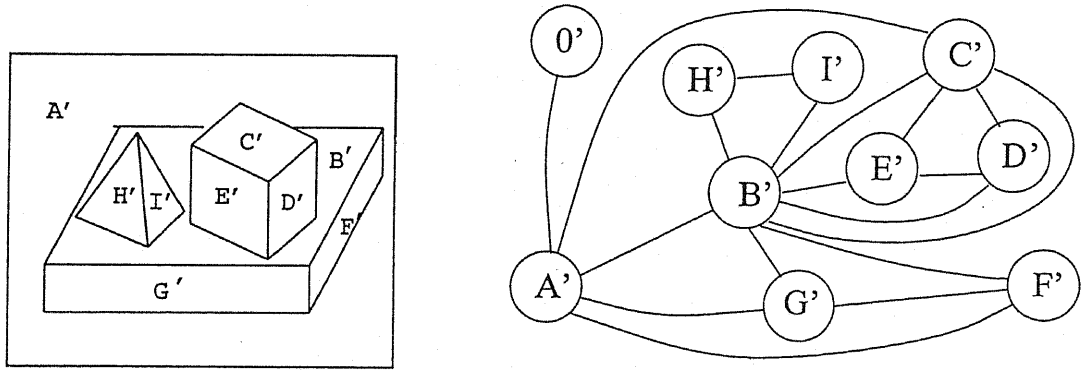
Let $T = \{\{A_i\}, \{V_k\}, \mathcal{E}\}$ and $T' = \{\{A'_j\}, \{V'_l\}, \mathcal{E}'\}$ denote the tree structure of the two RAGs, where $\{A_i\}$, $\{V_k\}$ and $\mathcal{E}$ denote the sets of articulation nodes, biconnected components, and the edges in one RAG, respectively, and $\{A'_j\}$, $\{V'_l\}$ and $\mathcal{E}'$ are the correspondings of those in the other RAG. Matching T and T' is to build two isomorphic trees $\overline{T} = \{\{\overline{A_i}\}, \{\overline{V_k}\}, \overline{\mathcal{E}}\}$ and $\overline{T'} = \{\{\overline{A'_j}\}, \{\overline{V'_l}\}, \overline{\mathcal{E}'}\}$ from T and T' by performing graph transforming operations described in the previous chapter. In the obtained isomorphic trees $\overline{T}$ and $\overline{T'}$, it is self-evident that:

1. A virtual node $\overline{V_k}$ in $\overline{T}$ can only match to a virtual node $\overline{V'_l}$ in $\overline{T'}$, and the same for articulation nodes.

2. If two biconnected components $\overline{V_k}$ and $\overline{V'_l}$ are matched, then their parent articulation nodes, say $\overline{A_i}$ and $\overline{A'_j}$ must also be matched.

3. If two articulation nodes $\overline{A_i}$ and $\overline{A'_j}$ are matched, then the biconnected components $\{\overline{V_{i_k}}\}$ which connect with $\overline{A_i}$ can only match to $\{\overline{V'_{j_l}}\}$ connecting with $\overline{A'_j}$.

Notice that although the isomorphic trees should have one-to-one match of virtual nodes, in the original RAGs two or more biconnected components in one tree may be merged into one biconnected components in the other tree. Therefore the many-to-one or one-to-many matching of virtual nodes in the tree matching must be taken into account. As an example, Figure 5.8(a) shows an image of the same scene in Figure 5.1, but taken from a different viewpoint. Figure 5.8(b) shows its RAG. Because region I' touches to region E' and region C' sticks out from region B', there is only one biconnected components in this RAG. Hereafter we will consider the matching of two RAGs in Figure 5.1 and 5.8 as a running example of the matching algorithm.



(a) Regions in an image (b) Its RAG

Figure 5.8 Another example of image and its corresponding RAG

An exhaustive strategy is applied to search for the optimal matching of two RAGs at the tree level. First, any two biconnected components V_i and V'_j are assumed possible to be matched if there are two nodes $N_{k_i} \in V_k$ and $N_{l_j} \in V_l$ which are similar in features. From this start point, we search all the ways in which the two RAGs can be transferred into isomorphic trees, satisfying the three conditions mentioned

above. Among all the pairs of isomorphic trees built from a starting pair, the one with minimal graph transforming cost is regarded as the optimal matching of two RAGs with respect to the specific starting point. When the search is carried out for every starting point, the isomorphic trees obtained with minimal graph transforming cost is regarded as the optimal matching of two RAGs.

Building isomorphic trees from a specific starting pair of biconnected components V_k and V_l' is carried out in the following steps.

5.5.1 Building two Isomorphic Biconnected Components

If V_k and V_l' are to be matched as one-to-one, we need only to find out the optimal matching between the nodes in V_k and V_l', because the matching of V_k and V_l' are independent with the matching of the other parts of the trees. However, in general V_k or V_l' could match to a part of the other, so any local maximal *subgraph* isomorphism of one biconnected component in the other can be thought as their matching.

Definition 4 A local maximal subgraph isomorphism between V_k and V_l' is two maximal isomorphic biconnected components which can be built from V_k and a part of V_l' by performing graph transforming operations, or vise versa.

Figure 5.9 shows a case of local maximal subgraph isomorphism of V_2 in V_2', which are shown in Figure 5.6 and Figure 5.8, respectively.

For V_k and V_l', all local maximal isomorphic subgraphs $\overline{V_k}$ and $\overline{V_l'}$ are built (the details will be described in Section 5.5). Starting from each local maximal isomorphic subgraphs $\overline{V_k}$ and $\overline{V_l'}$, we search for the isomorphic trees which can be built from the original RAGs. This is done in the following steps.

5.5.2 Reorganize the Trees

Building two isomorphic biconnected components $\overline{V_k}$ and $\overline{V_l'}$ from V_k and V_l' may cause V_k or V_l' to be split into two or more biconnected components. Note that

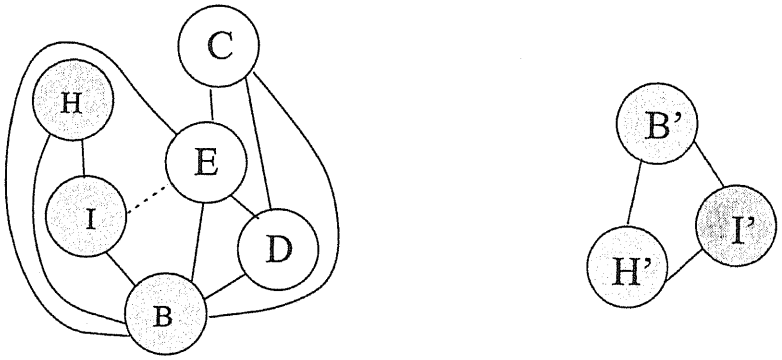


Figure 5.9: A maximal subgraph isomorphism of V_2 and V_1'

in Figure 5.10 the edge (I, E) was deleted in building a local maximal isomorphic biconnected components between V_2 and V_2'. This causes V_2 to be split into two biconnected components. The RAG containing the biconnected component should be reorganized, as shown in Figure 5.10.

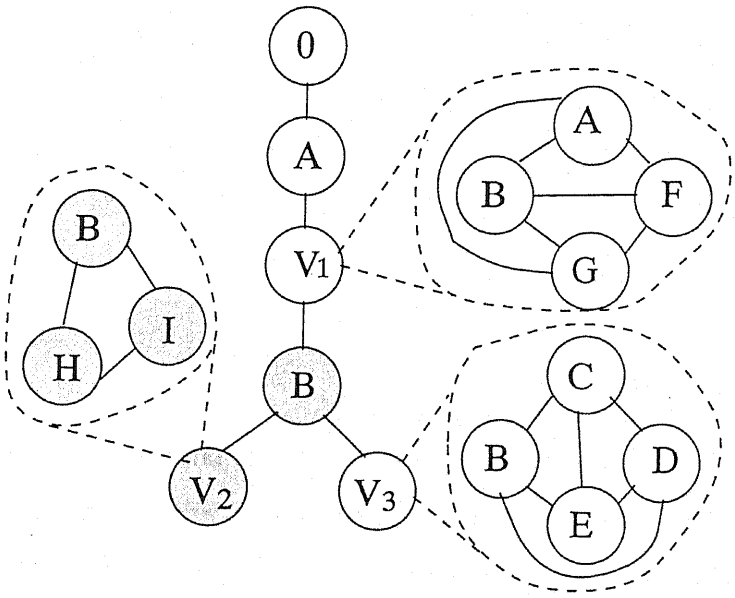


Figure 5.10: Splitting of biconnected component

5.5.3 Matching the remaining nodes of matched articulation nodes

When two isomorphic biconnected components $\overline{V_k}$ and $\overline{V_l'}$ are build, some articulation nodes $\{\overline{A_i}\}$ and $\{\overline{A_j'}\}$ involved in them are also matched. For each pair of matched articulation nodes, if there are some nodes which are connected with them and are not matched yet, then we try to match them.

1. Match the parent articulation nodes.

   If both of the parent nodes of $\{\overline{A_i}\}$ and $\{\overline{A_j'}\}$ are articulation node, then the parent articulation nodes A_p and A_p' are to be matched also.

2. Match biconnected components of A_i and A_j'.

   If there are some biconnected components not yet matched $S_v = \{V_m \mid m = 1, ..., M\}$ and $S_v' = \{V_n' \mid n = 1, ..., N\}$ connected with A_i and A_j', respectively, then the biconnected components in S_v must match to those in S_v'.

   As noticed in Section 5.2, a biconnected components V_m descendant from an articulation node A_i is a set of adjacent regions enclosed by region A_i. As shown in Figure 5.6, V_m may be merged not only to a biconnected component within A_i, but also to those out of A_i in the other RAGs. Thus a child biconnected component of A_i may be matched not only with the child biconnected component but also with the parent biconnected component of A_j' in the as other image.

   For any $V_m \in S_v$ and $V_n \in S_v'$, because they contain $\overline{A_i}$ and $\overline{A_j'}$ which are already matched, we can build the local maximal isomorphic biconnected components between V_m and V_n' by starting from $\overline{A_i}$ and $\overline{A_j'}$, that is described in Section 6.3.

   Let $C_b(V_m, V_n')$ stands for the cost of the local maximal isomorphic biconnected components built from V_m and V_n', $C_w(S_v, S_v')$ stands for the minimal cost for matching all the biconnected components in S_v and S_v'. We can compute

$C_w(S_v, S'_v)$ recursively as following.

$$C_w(S_v, S'_v) = \min_{(m,n)} \{C_b(V_m, V'_n) + C_w(S_v \setminus \{V_m\} \cup S_m, S'_v \setminus \{V'_n\} \cup S'_n) \quad (5.1)$$

where $\setminus$ stands for the set difference and S_m and S'_n are the sets of remaining parts of the biconnected components after the partial matching of V_m and V'_n.

In the example, there are two not-yet-matched virtual nodes connected with B and one connected with B' which are matched in the previous steps. Two cases should be considered here: either V_1 or V_3 would be matched V'_1 first. Let us consider the case that V_3 and V'_1 be matched first. The matching of V_3 and V'_1 is shown in Figure 5.11.

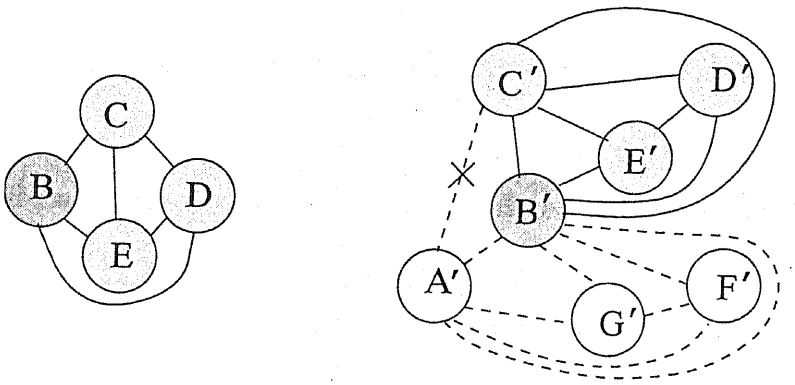


Figure 5.11 Matching in biconnected components
starting from matched articulation nodes

Note that the edge (C', A') will be deleted during building the local maximal isomorphic subgraphs between V_3 and V'_1. This will cause again V'_1 to be split into two biconnected components, as shown in Figure 5.12.

In the figure, only one biconnected component adjacent to B or B' remained to be matched. Matching V_1 and V'_1 results in complete matching of the biconnected components adjacent to B and B'.

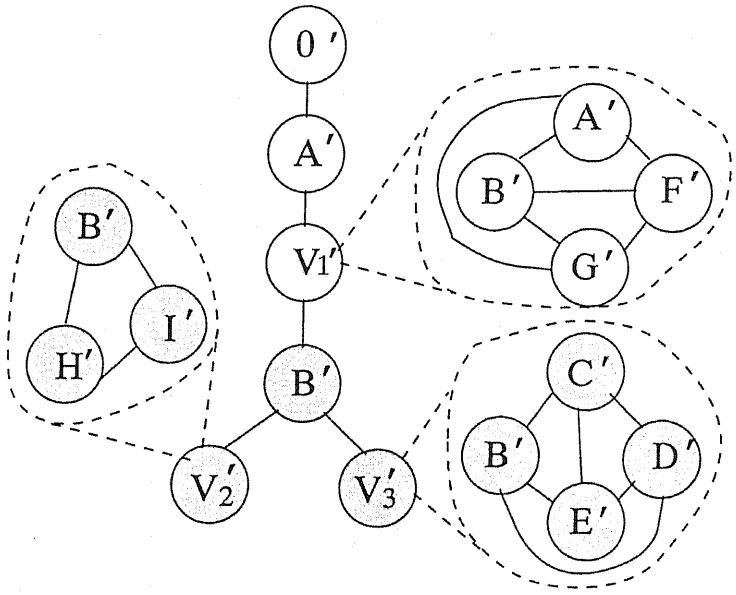


Figure 5.12: Remaining biconnected component

If we start from the second alternative, that is, matching of V_1 and V_1', a similar splitting of V_1' occurs and essentially the same result is obtained.

5.5.4 Extending the Matching of Trees.

Matching of S_v and S_v' may cause some other articulation nodes to be matched newly. Then the step 4 is repeated for the newly matched articulation nodes to proceed the matching of the trees.

In the example, in both cases of matching the remaining biconnected components V_1 and V_3 with V_3', the articulation nodes A and A' will get matched next. Repeating the processing in step 4 will results in two completely isomorphic RAGs.

5.6 Matching within Biconnected Components

Now consider matching nodes and edges within two biconnected components to find a local maximal subgraph isomorphism of V_k in V_l'. Because V_k may match to any

part of V_l', all the local maximal subgraph isomorphism should be searched for.

Any two nodes $N_i \in V_k$ and $N_j' \in V_l'$ which have similar features are thought to be possibly matched and regarded as the initial matched node pair of a local maximal isomorphic subgraphs of V_k and V_l' to be built. The process of building a local maximal subgraph isomorphism within V_k and V_l' from each initial matched pair of nodes N_i and N_j' is in the following.

1. For any starting pair N_i and N_j', build two maximal isomorphic BRAGs of them by performing graph transforming operations with the minimal cost (to be described in the next section).

   The pairs of nodes $\{(N_{i_k}, N_{j_k}') \mid k = 1, ..., n\}$ matched by isomorphism of the two BRAGs at N_i and N_j' are put into a queue Q.

2. Fetch an unprocessed pair (N_{i_k}, N_{j_k}') from Q. If N_{i_k} or N_{j_k}' is an articulation node, go to step 6.

3. Build two isomorphic BRAGs at N_{i_k} and N_{j_k}' independently of the matching obtained so far. If the pairs of nodes matched by the isomorphism of the BRAGs at N_{i_k} and N_{j_k}' do not conflict with the previous ones, the isomorphic subgraphs of the two biconnected components, which are the unions of isomorphic BRAGs obtained until now, can be extended to include the two isomorphic BRAGs at N_{i_k} and N_{j_k}'. The newly obtained matchings of nodes are put into Q.

4. If a newly obtained matched node pair conflicts with any previously matched pair, a conflict processing (to be described in Section 5.6.3) is performed to handle the newly obtained matching.

5. mark (N_{i_k}, N_{j_k}') as processed.

6. Steps 2 through 5 are iterated until all the pairs in Q are processed.

5.6.1 Matching of BRAGs

Let N_i and N'_j be nodes in two biconnected components of G and G', and let $S(N_i) = \{N_{i_1}, N_{i_2}, ..., N_{i_m}\}$ and $S(N'_j) = \{N'_{j_1}, ..., N'_{j_n}\}$ be the nodes of the BRAGs at N_i and N'_j, subscripted clockwise around N_i and N'_j, respectively. As stated above, nodes in $S(N_i)$ and $S(N'_j)$ should be matched with order reserving while the two BRAGs are changed into isomorphic graphs. Thus the minimal cost of the graph transforming operations that make the two BRAGs isomorphic can be computed with an algorithm similar to the one for circular string matching[68-70].

Suppose that the two isomorphic BRAGs are to be built starting from N_{i_1} and N'_{j_1}, and let $C(k,l)$ be the minimal cost of the graph transforming operations performed in order to make the two subgraphs $\{N_i, N_{i_1}, ..., N_{i_k}\}$ and $\{N'_j, N'_{j_1}, ..., N'_{j_l}\}$ isomorphic, then $C(k,l)$ can be computed recursively as follows:

$$\begin{aligned} C(k,l) \;=\; \min\{\; & C(k-1,l-1) + C_{mt}(N_{i_k}, N'_{j_l}), \\ & C(k-1,l) + C_{de}(N_i, N_{i_k}), \\ & C(k,l-1) + C_{de}(N'_j, N'_{j_l}), \\ & C(k-1,l) + C_{dn}(N_{i_k}), \\ & C(k,l-1) + C_{dn}(N'_{j_l}), \\ & C(k-1,l-1) + C_{dm}(N_{i_k}), \\ & C(k-1,l-1) + C_{dm}(N'_{j_l}), \\ & C(k-1,l) + C_{mg}(N_{i_k}), \\ & C(k,l-1) + C_{mg}(N'_{j_l}) \quad \} \end{aligned} \tag{5.2}$$

where $C(0,0)$ is set to 0 in advance.

$C_{mt}(N_{i_k}, N'_{j_l})$ is the cost of substituting (or matching) N_{i_k} by (or with) N'_{j_l}, defined as:

$$C_{mt}(N_{i_k}, N'_{j_l}) = \begin{cases} D(N_{i_k}, N'_{j_l})/D_0 & \textit{if } D(N_{i_k}, N'_{j_l}) < D_0 \\ D(N_{i_k}, N'_{j_l})^2/D_0^2 & \textit{otherwise} \end{cases} \tag{5.3}$$

where we used as $D(A, B)$ a *chromatic distance* computed in HSV space using the average color of region A and B in the color image matching experiments to be described in Section 6.4. D_0 is a threshold.

$C_{de}(N_i, N'_{i_k})$ is the cost of deleting the edge (N_i, N'_{i_k}), which is defined as

$$C_{de}(N_i, N_{i_k}) = \frac{1}{1 + exp(-T(l - p\bar{l}))} \tag{5.4}$$

where l is the length of the common boundary between N_i and N_{i_k}, $\bar{l}$ is the average length of the common boundaries between N_i and the regions adjacent to N_i, p is a factor which takes a value between 0 and 1, and T is a factor to control the gradient of the function.

$C_{dn}(N_{i_k})$ is the cost of deleting N_{i_k} in the following two cases:

1. All the other regions adjacent to N_{i_k} except N_i are also adjacent to N_i.

2. N_{i_k} is regarded as occluded by $N_{i_{k-1}}$ or $N_{i_{k+1}}$. In this case, deleting N_{i_k} causes $N_{i_{k-1}}$ to be adjacent with $N_{i_{k+1}}$.

In both cases, $C_{dn}(N_{i_k})$ is defined as

$$C_{dn}(N_{i_k}) = A(N_{i_k})^2 / A_0^2 \tag{5.5}$$

where $A(N)$ stands for the area of region N and A_0 is a constant to be determined depending on the coarseness of the regions in the given images.

$C_{dm}(N_{i_k})$ is the cost of deleting N_{i_k} in case that N_{i_k} is regarded as occluded by a node N_p that is adjacent to N_{i_k} but different from N_i, $N_{i_{k-1}}$ and $N_{i_{k+1}}$. In this case, deleting N_{i_k} causes N_p to be adjacent to N_i, and so N_p may match to N'_{j_l}. Therefore, the cost $C_{dm}(N_{i_k})$ is computed as follows.

$$C_{dm}(N_i) = C_{dn}(N_{i_k}) + C_{mt}(N_p, N'_{j_l}) \tag{5.6}$$

$C_{mg}(N_{i_k})$ is the cost of merging the node N_{i_k} into a node N_{i_q}, where $p < k$ and N_{i_q} is the last node matched in the optimal matching between $\{N_{i_1}, ..., N_{i_{k-1}}\}$

and $\{N'_{j_1}, ..., N'_{j_l}\}$. Without loss of generality (the proof is given in Appendix 2), we assume that N'_{j_l} is the node matched with N_{i_q}. Then the cost $C_{mg}(N_{i_k})$ can be computed by

$$C_{mg}(N_{i_k}) = \begin{cases} 0 & \text{if } N_{i_k} = N_{i_q} \\ C_{mt}(N_{i_k}, N'_{j_l}) + C_a(N_{i_k}, N'_{j_l}) & \text{otherwise} \end{cases} \quad (5.7)$$

where $C_a()$ is the cost calculated for the case when the total area of merged nodes is more than that of their corresponding node N'_{j_l}. It is defined as:

$$C_a(N_{i_k}, N'_{j_l}) = \begin{cases} 0 & \text{if } A(N_{i_k}) + \sum A(N_{i_q}) \leq A(N'_{j_l}) \\ ((A(N_{i_k}) + \sum A(N_{i_q}) - A(N'_{j_l})) & \\ \quad /A(N'_{j_l}))^2 & \text{otherwise} \end{cases} \quad (5.8)$$

where the summation is taken over the nodes N_{i_q}'s which are matched to N_{j_l}.

Note that no cost is imposed for the edge deletion and substitution caused by node merging or node deletion.

While $C(k, l)$ is calculated by Eq.(5.2), the operation which produces the optimal partial matching is also recorded. When $C(m, n)$, the final minimal cost, is computed, we can trace back from $C(m, n)$ to get the optimal matching starting at the specified node pair.

Because $S(N_i)$ and $S(N'_j)$ are circular sequences, we have to shift one of the starting nodes against the other and repeat the computation above, and find the minimal cost among the ones obtained by starting from different starting node pair in the sequence.

5.6.2 Matching at Articulation Nodes

As described in Section 5.6.1, the optimal matching of the remaining biconnected components connected with a pair of matched articulation nodes (A, A') can be computed by Eq.(5.2), which calls the node matching within two biconnected components.

For two biconnected components B_i and B'_j, which are connected with A and A' respectively, a procedure similar to the one described in Section 6.3.1 can be applied, except that the matching starts from a matched pair of articulation nodes (A, A'). Because the edges of an articulation node in a biconnected component represents the boundary of the biconnected component, they will be split when the biconnected component are separated into more than one biconnected components in the other image. Thus the initial isomorphic basic planar graphs at A and A' should be built by a *partial* matching of the nodes surrounding A and A'.

Similar to the procedure in Section 5.6.1, we regard the two BRAGs at A and A' as two attributed circular strings. Let $S(A) = \{A_i \mid i = 1, ..., m\}$ and $S(A') = \{A'_j \mid j = 1, ..., n\}$ be the sets of nodes surrounding A and A' in the biconnected components B_i and B'_j, respectively. Assume that the nodes are subscripted in clockwise order. Without loss of generality, suppose $m \leq n$. Then we make two non-circular strings, one is the non-circular string of $S(A)$, and the second is the non-circular string $2S(A')$, that is, $S(A')$ repeated twice. The partial matching between circular strings $S(A)$ and $S(A')$ becomes the problem of finding the optimal substring matching[68] of $S(A)$ in $2S(A')$.

The procedure for computing the minimal matching cost described in Section 6.3.1 can be applied to this case with the following changes.

1. $C(0, i)$ is set to 0 for $i \leq n$ and to ∞ otherwise.

2. The minimal cost is computed by

$$\min\{C(m, j) \mid j = 0, ..., 2n\}.$$

5.6.3 Processing of Conflicts

Since the two isomorphic graphs of two biconnected components are built by yielding iteratively two isomorphic BRAGs at the matched nodes *independently*, the matchings

of nodes in a BRAGs pair may conflict to the matching obtained previously. Our algorithm introduced a simple strategy to deal with the conflicts, in order to avoid processing the two possibilities in parallel.

Let (N_i, N'_j) be the pair of matched nodes and suppose that two isomorphic BRAGs at N_i and N'_j have just been yielded. Assume that a newly obtained matching (N_{i_k}, N'_{j_l}) in the isomorphism of the two BRAGs at N_i and N'_j conflicts to a matching, say (N_p, N'_{j_l}) where $N_p \neq N_{i_k}$, which is obtained previously. In this case, we keep the matching (N_p, N'_{j_l}) and adjust (N_{i_k}, N'_{j_l}) in the newly yielded isomorphic BRAGs by deleting the edges (N_i, N_{i_k}) and (N'_j, N'_{j_l}). The cost $C(N_i, N'_j)$ of the isomorphism of the two of basic planar graphs at N_i and N'_j is also adjusted as

$$\begin{aligned} C(N_i, N'_j) \leftarrow\ & C(N_i, N'_j) + C_{de}(N_i, N_{i_k}) \\ & + C_{de}(N'_j, N'_{j_l}) - C_{mt}(N_{i_k}, N'_{j_l}) \end{aligned} \tag{5.9}$$

At the same time, the pair (N_i, N'_j) is put into the starting point list for search, so that the abandoned matching (N_{i_k}, N'_{j_l}) can be established earlier in the later search starting from (N_i, N'_j).

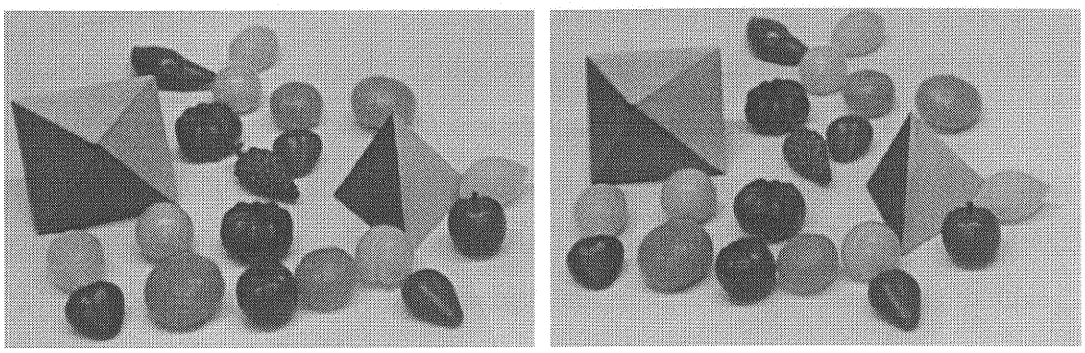
5.7 Experiments

We applied our algorithm to various kinds of color images, which are shown in Figure 5.13(a), Figure 5.14(a), Figure 5.15(a) and Figure 5.16(a). The images were first segmented into regions which consist of connected pixels of similar color, and a unique label is assigned to each region in either of the two images. For each region, several features, such as the average color, area, center of gravity, are computed. Then all the boundaries of every region are followed to establish the region adjacency lists. In general, only one adjacency list is needed for one region, but for the regions which encircle n sets of adjacent regions in it, $n + 1$ lists is needed. The first one is used to record its adjacency relations of the regions surrounding it, and the other n lists are for

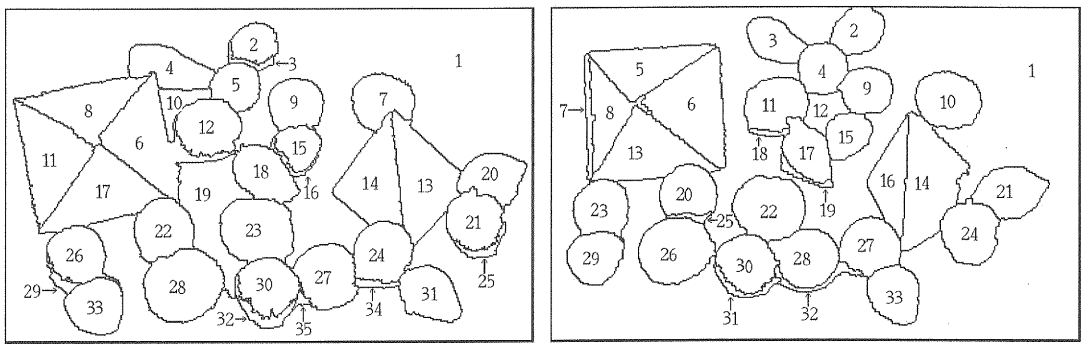
describing the boundary regions of the n "holes" in it. With this structure, it is easy to identify articulation nodes as the regions which have more than one boundaries.

Our algorithm runs on the graph representation converted from the adjacency lists. The results of matching for the test images are shown in Figure 5.13(b), Figure 5.14(b), Figure 5.15(b) and Figure 5.16(b), respectively. The regions in two images with the same number are regions matched each other. A region corresponds to '-1' means that the region has been deleted in the searching, and a region corresponds to '-2' means that the region remains unmatched in the search and is deleted at the end of search.

The values of parameters in the cost computations of graph transforming operations are selected as: $D_0 = 0.1, T = 20, p = 0.2$ and $S_0 = 200$. The computation time (user time on SUN SPARC-10) are 85 sec. and 110 sec., respectively.

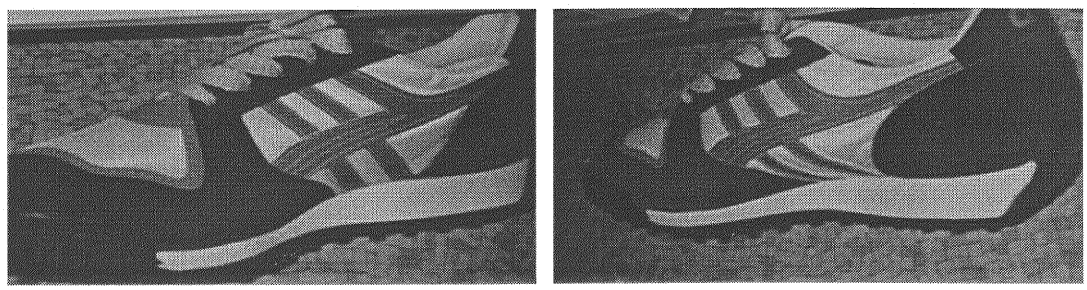


(a) Input images (only bright values are shown)

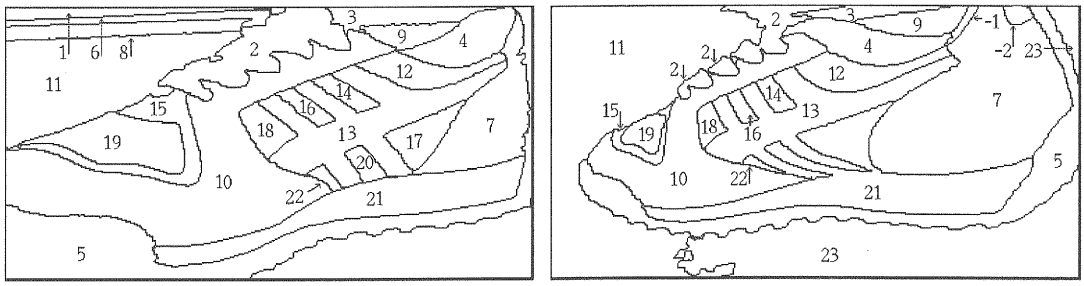


(b) Results of region correspondence

Figure 5.13: Results of region correspondence of images of toys



(a) Input images (only bright values are shown)



(b) Results of region correspondence

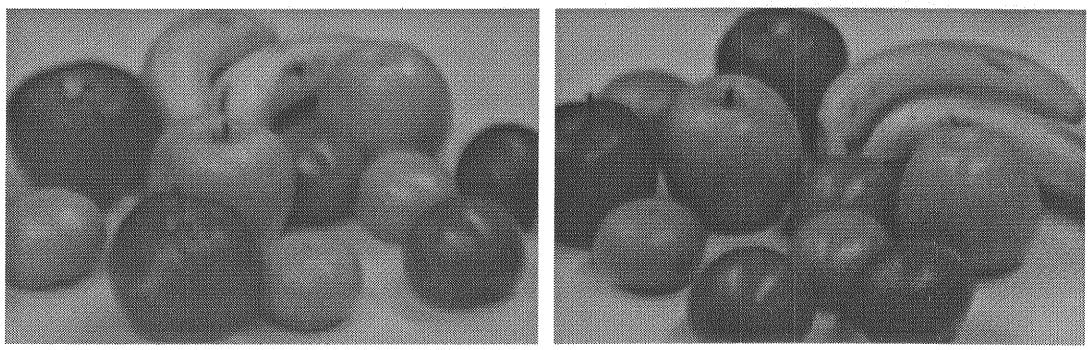
Figure 5.14: Results of region correspondence of images of shoes

5.8 Summary

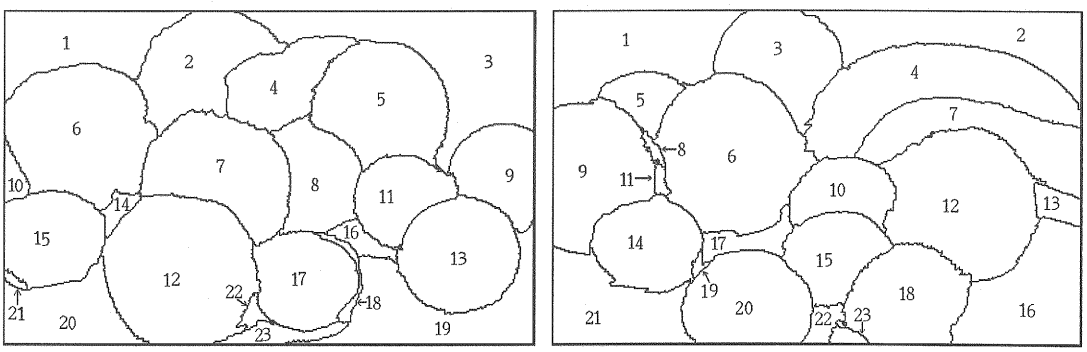
We formulated region correspondence problem between two images into inexact matching of two attributed planar graphs, and gave an operational definition of it. Some properties of the planar graph and of the topological relations of regions are noticed, in order to construct an efficient algorithm to find the optimal inexact matching of two attributed planar graphs.

Because our algorithm searches for the optimal matching only from the starting pairs of nodes which are similar in features, we expect that the final matching would be only approximately optimal in some cases. However, the experimental results show that the algorithm yields sufficiently good region matching in most cases.

There is a limit of our methods: for an object close to the camera, because it will move so much in the two images that adjacency relations between the regions for this object in two images will be wholly different, causing the deletion of the region to make the cost less. Therefore, these regions in images may not be matched correctly by our algorithm. Comparing the features of the deleted regions in two images may solve the problem.



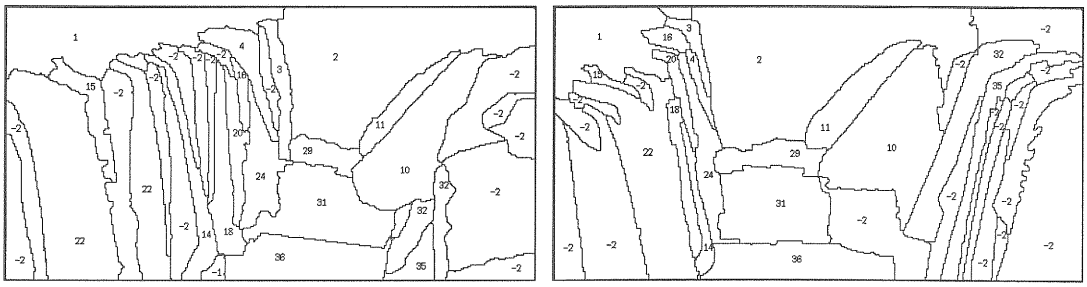
(a) Input images (only bright values are shown)



(b) Results of region correspondence

Figure 5.15: Results of region correspondence of images of fruits

(a) Input images (only bright values are shown)



(b) Results of region correspondence

Figure 5.16: Results of region correspondence of images of clothes

CHAPTER 6

Conclusion and Future Works

In this thesis, I indicated out that a flexible and robust matching technique for matching two temporarily and/or spatially shifted images in presence of noise, deformation, occlusion and inconsistency in feature extraction, is essential for real applications of computer vision. I showed that the region-based approach is more promising than other feature-based approaches to this problem, because (1) there are always much fewer regions in an image than other primitives such as edges, (2) regions possess more information which supplies higher discriminating capability of regions, and (3) they are also more stable against noise, deformation and occlusion than, for instance, edges.

Three algorithms of region-based matching approach have been proposed for different problems of image matching. The first one was proposed for stereo matching where epipolar line constraint is available and the other two algorithms are proposed for image correspondence in general case where the relative positions of viewpoints are not known. The third one is possibly applicable also to image matching for object identification, location or recognition, with some extensions.

In Chapter 2, I have described a region-based part-to-whole approach for stereo matching. In this approach, the regions projected from continuous surfaces in the scene were regarded as matching primitives, and we notice that those regions are

equivalent to the patches which can be matched continuously between two images. The stereo matching problem was then represented as a problem of finding consistent matchings of such patches in all the scene. A part-to-whole approach was adopted to do this. First, we divided the range of disparity into some small intervals, and used dynamic programming, with the continuity of matching being considered, to find continuous matchings of segments on an epipolar line in each interval of disparity. Then the matched segments were integrated into matched patches by considering vertical continuity between adjacent epipolar lines. Finally, we integrated the matchings of patches obtained in individual intervals to generate a consistent set of matching of patches of the whole scene, by selecting the best-matched one from the overlapped or conflicting matchings of patches.

The patches used in our approach is equivalent to regions, that implies that patches are more stable and discriminating than other primitives such as pixels or edges. Therefore, it can be expected to obtain more stable and reliable matching result by our method than by the methods using other primitives. On the other hand, the patches are established by continuous matching of connected pixels, so dense disparities are obtained simultaneously when the patches are matched. Thus the disadvantage that the feature-based approaches can only obtain sparse disparities was overcome by our approach.

Another characteristics of the proposed approach is that we need not to segment the images into homogeneous regions beforehand, that is always a too hard task for the existing segmentation techniques to do in the case of natural images which are always complicated and over-textured. This characteristics allowed our approach to be applicable to some complicated scenes. The results on some experimental images showed the effectiveness of our method, but it need to be compared with the other methods under the ground truth data.

For image correspondence in general case where the epipolar constraint is not available, the region-based approach is regarded to be a robust and efficient approach

to image matching problem. It also plays an important role in a hierarchal multi-primitives system of image matching, where the region matching results are used as guidance information for the matching of more detailed primitives such as edges and edgels. The reliability of the region matching results is critical in such a hierarchal multi-primitives image matching system, because the error occurred in the region matching stage will be brought to the next stages of matching of other primitives.

In Chapter 3 and Chapter 4, we proposed a reliable region-based image matching approach which consists of two stages. In the first stage, a local ternary relation among regions was used together with chromatic features of regions to suppress the matching ambiguity occurred when only the features of regions are used. By this method, more reliable region correspondences could be established. In the second stage, the optical flow generated by the camera movement, which can be estimated using all the obtained correspondences, was used as a global consistency constraint to validate individual correspondences, and the ones which do not obey the optical flow were detected as mistakes and then corrected using the estimated optical flow.

In fact, in the applications of object identification, location and recognition, the scene can be thought consist of some objects which have some structure. For the image matching problem in such applications, we suggested that a syntactic representation of the images should first be attempted to describe the structure of the objects in the images, and then carry out a structural matching on those syntactic descriptions of images. Essentially, the regions in such images corresponding to the faces of objects in the scene, and they possess much more semantic information than other primitives, such as edges. So the regions are more appropriate for semantic representation of objects in the images.

In Chapter 5, we attempted a region-based syntactic approach for the region matching problem in the general case. We represented the regions and their relations in an image with a Region Adjacency Graph(RAG), and then formulated the problem of region matching into a problem of finding the optimal inexact graph matching

between two RAGs. We showed that a RAG is an attributed *planar* graph, and noticed that they have some specific properties. Utilizing these properties, we could construct an efficient algorithm to find the optimal inexact matching between two RAGs.

Some future works can be enumerated as follows:

1. *Stereo matching:* In the proposed stereo matching approach, we adopted a strategy of merging the continuous segments matched on each epipolar lines to obtain the patch matchings. The horizontal continuity in the patch matching is achieved by extending the dynamic programming to consider the matching continuity. However, the vertical continuity in the patch matchings is only improved by utilizing the complementarity of the segment matchings obtained in different matching orders, that is, the matching obtained in the order from left to right and that obtained in reverse order. A more sophisticated way to obtain the optimal matching of patches with both horizontal and vertical continuity should be considered in the future work.

   Another problem is that the effectiveness of the algorithm should be evaluated explicitly. Although the results on some experimental images showed the effectiveness of our method, it need to be compared with the other methods under the ground truth data.

2. *Image matching in general case:* The region matching in the general case is carried out on the output results of region segmentation. Although our algorithm allow inconsistency in region segmentation, but it can not work well with so bad results of image segmentation. Other primitives such as edge may be used jointly with the region primitives.

   Another reason of necessity for using other primitives jointly is that we can only obtain sparse and rough correspondences between two images from the results of region matching. In general, we can only obtain the correspondence

of the centers of gravity of the corresponding regions, which may be too spare and too inaccurate for some applications. Therefore, matching of more detailed features, such as edges, corners or the interior points in the regions, is required. In such a top-down hierarchal matching approach, interactive use of the hierarchal primitives must be considered.

3. *Extending the region-based image matching approach to model matching:* The region-based structural matching approach was applied to the general image matching by now, however, it is possible to be extended for object location, identification and recognition, because a lot of objects can be described by the their surfaces and the structural relations between the surfaces, such as aspect representation, which is very similar to the structural representation of regions in the images. However, for such further applications, some problems such as matchings between the models of objects and an input image, which may be represented as inexact subgraph isomorphic, should be investigated.

   The structural relations of surfaces contains not only the topologic relations (that is, the adjacency relations) of the object, but also the geometric relations, such as the paralleling relation or orthogonal relation of two surfaces. Up to now, most of the matching techniques used either topologic relations or geometric relations in the images. However, while the topologic relations in an object are relatively stable with perspective transformation, but they are neither discriminant for the configuration of the regions involved, nor stable to inconsistency of region segmentation. On the other hand, geometric relations are discriminant for configuration of regions and relatively stable against the inconsistency of region segmentation, but they are not invariant under perspective transformation. We suggest that both the geometric and topologic relations should be used jointly in describing the model of the objects and be utilized in the matching process.

APPENDIX A

The Relation of Corresponding Points in Two Images

The relation of the scene-space coordinates (X, Y, Z) of a point P before motion, and coordinates (X', Y', Z') of P after motion can be represented by

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T \tag{A.1}$$

where T is called as translation vector which is defined as following

$$T = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

and

$$R = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$$

Transform these equations so that it can be represented by the image points

(x, y) and (x', y') and thsir depthes as

$$Z' \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = Z \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Eliminate Z' from the above eqution, we can obtain:

$$x' = \frac{(r1x + r2y + r3)Z + X}{(x7x + r8y + r9)Z + Z} y' = \frac{(r4x + r5y + r6)Z + Y}{(x7x + r8y + r9)Z + Z}$$

Elimination of Z from the above eqution results in the Epipolar constraint.

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} \begin{pmatrix} 0 & -Z & X \\ Z & 0 & -Y \\ -X & Y & 0 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

For a given point (x, y) in a image, let

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 0 & -Z & X \\ Z & 0 & -Y \\ -X & Y & 0 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \qquad \text{(A.2)}$$

then the corresponding point (x', y') in the other image must sutisfy $Ax' + By' + C = 0$, that is, (x', y') is on the line of $Ax + By + C = 0$.

APPENDIX B

Proof of Property 3 of RAG

Let N be a node in a biconnected component and let $E = \{(N, N_i) \mid i = 1, ..., n\}$ be the edges emanating from N in the biconnected component, subscripted clockwise on the image planes. The following cases can be considered:

1. $n = 2$.

   In this case, there is only one way to arrange the edges as a circlar sequence in a plane.

2. $n > 2$ and all the nodes in $S(N) = \{N_i \mid i = 1, ..., n\}$ are distinct.

   In this case, because there is an edge between any two neighboring nodes N_i and N_{i+1} (Property 2), the subgraph composed of nodes of $\{N\} \cup S(N)$ and the edges between the nodes is a triply connected subgraph. Accordingly, Property 3 is held.

3. $n > 2$ and a node appears twice in $S(N)$.

   Suppose that $N_i = N_j$, $i \leq j$ and we can partitions the edges into two disjoint groups as $E1 = \{(N, N_1), ..., (N, N_i), (N, N_{j+1}), ..., (N, N_n)\}$ and $E2 = \{(N, N_{i+1}), ..., (N, N_{j-1}), (N, N_j)\}$. We can obtain:

Lemma A1 In any embedding of the edges in a plane, any edge in $E1$ (except (N, N_i)) can not be neighboring an edge in $E2$ (except (N, N_j)).

Proof: If a node N_k in $E1$ is neighboring a node N_l in $E2$, then due to the Property 2 of RAGs, there must exists an edge (N_k, N_l). So the subgraph composed of nodes of $\{N\} \cup S(N)$ and the edges between the nodes must be a triply connected subgraph, and N is not a biarticulation node.

Lemma A1 means that if there are more than one embeddings of the edges of $E1$ and $E2$ in a plane as RAGs, then the order of the edges $E1$ or $E2$ can changed only within $E1$ or $E2$, respectively.

Lemma A2 For the edges in $E1$ (or $E2$) and the subgraph composed of the nodes $\{N, N_1, ..., N_i, N_{j+1}, N_n\}$ (or $\{N, N_{i+1}, ..., N_{j-1}\}$), either case 1 or case 2 will be hold. That is, only the oclockwise embedding of the edges should be considered if the mirror version is ignored.

From Lemma A1 and A2, we obtain that only the clockwise of the embedding of the edges of N should be considered.

4. The other case that $n > 2$ and a node appears more than twice in $S(N)$ can be proved inductively using the reasoning above.

APPENDIX C

About Node Merging

Suppose that N'_{j_q}, $q < l$ is the last matched node in matching of $\{N_{i_1}, ..., N_{i_{k-1}}\}$ and $\{N'_{j_1}, ..., N'_{j_l}\}$.

1. The cost of the matching by merging N_{i_k} in this case is computed as:

$$\begin{aligned}
C'(k,l) &= C(k-1,l) + C_{mt}(N_k, N'_{j_q}) + C_s(N_{i_k}, N'_{j_q}) \\
&= C(k-1,q) + C_{de}(N'_{j_{q+1}}) + \cdots + C_{de}(N'_{j_l}) + \\
&\quad C_{mt}(N_{i_k}, N'_{j_q}) + C_s(N_{i_k}, N'_{j_q}) \\
&= C(k-1,q) + C_{mt}(N_{i_k}, N'_{j_q}) + C_s(N_{i_k}, N'_{j_q}) + \\
&\quad C_{de}(N'_{j_{q+1}}) + \cdots + C_{de}(N'_{j_l}) \\
&\geq C(k,q) + C_{de}(N'_{j_{q+1}}) + \cdots + C_{de}(N'_{j_l}) \\
&\geq C(k,q+1) + C_{de}(N'_{j_{q+2}}) + \cdots + C_{de}(N'_{j_l}) \\
&\cdots \\
&\geq C(k,l-1) + C_{de}(N'_{j_l})
\end{aligned}$$

So the merging of N_{i_k} in this case need not to be considered.

2. N'_{j_l} is regarded as the last matched node while it is not so. In this case the cost is:

$$\begin{aligned} C'(k,l) &= C(k-1,l) + C_{mt}(N_k, N'_{j_l}) + C_s(N_{i_k}, N'_{j_l}) \\ &= C(k-1,l-1) + C_{de}(N'_{j_l}) + C_{mt}(N_{i_k}, N'_{j_l}) + C_s(N_{i_k}, N'_{j_l}) \\ &\geq C(k-1,l-1) + C_{mt}(N_{i_k}, N'_{j_l}) \end{aligned}$$

That is, the cost computed for this case will not influence the matching cost $C(i,j)$.

Bibliography

[1] R. Dhond and J. K. Aggarwal: Structure from stereo – A review, *IEEE Trans. on SMC*, Vol. 19, No. 6, pp. 1489-1510, 1989

[2] T. Kanade and M. Okutomi: A stereo matching algorithm with an adaptive window: Theory and experiment, *IEEE Trans. on PAMI*, Vol. 16, pp. 920-932, 1994

[3] R.C. Bolles, H.H. Baker, and M.J. Hannah: The JISCT stereo evaluation, *Proc. of Image Understanding Workshop*, pp. 263-274, 1993

[4] H. H. Baker and T. O. Binford: Depth from edges and intensity based stereo, *Proc. of IJCAI*, pp. 631-636, 1981

[5] M. J. Hannah: SRI's baseline stereo system, *Proc. of IUW*, pp. 201-208, 1980

[6] K. Mori, M. Kidode and H. Asada: An Iterative Prediction and Correction Method form Automatic Stereopsis, *Journal of CVIP*, Vol. 2, pp. 393-401, 1973

[7] T. Kanade and M. Okutomi: A stereo matching algorithm with an adaptive window: Theory and experiment, *IEEE Trans. on PAMI*, Vol. 16, pp. 920-932, 1994

[8] J. R. Jordan and A. C. Bovik: Using Chromatic Information in Dense Stereo Correspondence, Pattern Recognition, Vol. 25, No. 4, pp. 367-383, 1992

[9] D. Marr: *Vision*, San Francisco, CA: Freeman, 1982

[10] W. E. Grimson: Computational Experiments with a Feature Based Stereo Algorithm, *IEEE Trans. on PAMI*, Vol. 7, No. 1, pp. 17-34, 1985

[11] A. Bensrhair P. Miche and R. Debrie: Fast and automatic stereo vision matching algorithm based on dynamic programming method, *Pattern Recognition Letters*, Vol. 17, pp. 457-466, 1996

[12] Y. Ohta and T. Kanade: Stereo by intra- and inter- scan-line search using dynamic programming, *IEEE Trans. on PAMI*, Vol. 7, No. 2, pp. 139-154, 1985

[13] R. Mohan and G. Medioni: Stereo Error Detection, Correction and Evaluation, *IEEE Trans. on PAMI*, Vol. 11. No. 2, pp. 113-120, 1989

[14] U. R. Dhond and J. K. Aggarwal: Stereo matching in the presence of narrow occluding objects using dynamic disparity search, *IEEE Trans. on PAMI*, Vol. 17, No. 7, pp. 719-734, 1995

[15] D. Brockelbank and Y. H. Yang: An Experimental Investigation in the Use of Color in Computational Stereopsis, *IEEE Trans. on SMC*, Vol. 19, No. 6, pp. 1365-1383, 1989

[16] S.B. Marapane and M.M. Trivedi: Multi-primitive hierarchical (MPH) stereo analysis, *IEEE Trans. on PAMI*, Vol. 16, pp. 227-240, 1994

[17] M. S. Lew and T. S. Huang: Learning and Feature Selection in Stereo Matching, *IEEE Trans. on PAMI*, Vol. 16, No. 9, pp. 869-881, 1994

[18] Z. N. Li: Stereo correspondence based on line matching in Hough space using dynamic programming, *IEEE Trans. on SMC*, Vol. 24, pp. 144-152, 1994

[19] R. Horaud and T. Skordas: Structural matching for stereo vision, *Proc. 9th Int'l. Conf. on Pattern Recognition*, pp. 439-445, 1988

[20] H. J. Lee and W. L. Lei: Region matching and depth finding for 3D objects in stereo aerial photographs, *Pattern Recognition*, Vol. 23, No. 1/2, pp. 81-94, 1990

[21] C.V. Stewart and J.K. MacCrone: Experimental analysis of a number of stereo matching components using LMA, *Proc. of 10th ICPR*, Vol. A, pp. 254-258, 1990

S. B. Marapane and M. M. Trivedi, "Region-Based Stereo Analysis for Robotic Applications", *IEEE Trans. on Syst Man Cybern*, Vol. 19, No. 6, pp. 1447-1464, 1989

[22] X. Hu and N. Ahuja: Matching Point Features with Ordered Geometric, Rigidity, and Disparity Constraints, *IEEE Trans. on PAMI*, Vol. 16, No. 10, pp. 1041-1049, 1994

[23] R. Mohan, G. Medioni and R. Nevatia, Stereo Error Detection, Correction and Evaluation, *IEEE Trans. on PAMI*, Vol. 11, No. 2, pp. 113-120, 1989

[24] W. Hoff and N. Ahuja: Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection, *IEEE Trans. on PAMI*, Vol. 11, No. 2, pp. 121-236, 1989

[25] K. L. Boyer and A. C. Kak: Structural Stereopsis for 3-D Vision, *IEEE Trans. on PAMI*, Vol. 10, No. 2, pp. 144-166, 1988

[26] D. D. Vleeschauwer: An intensity-based, Coarse-to-fine Approach to Reliable Measure Binocular Disparity, *CVGIP: Image Understanding*, Vol. 57, No. 2, pp. 204-218, 1993

[27] R. S. Mitra and N. N. Murthy: Elastic, Maximal Matching, *Pattern Recognition*, Vol. 24, No. 8, pp. 747-753, 1991

[28] D. J. Kahl, A. Rosenfeld and A. Danker: Some Experiments in Point Pattern Matching, *IEEE Trans. on SMC*, Vol.10, No. 2, pp.105-116, 1980

[29] D. Skea, I Barrodale, R. Kuwahara and R. Poeckert, A Control Point Matching Algorithm, *Pattern Recognition*, Vol. 26, No. 2, pp.269-276, 1993

[30] R. N. Strickland and Z. Mao: Computing Correspondence in a sequence of non-rigid shapes, *Pattern Recognition*, Vol. ??, No. ?, pp.901-912, 1992

[31] S. B. Marapane and M. M. Trivedi: Region-Based Stereo Analysis for Robotic Applications, *IEEE Trans. on SMC*, Vol. 19, No. 6, pp. 1447-1464, 1989

[32] C. S. Fuh, P. Maragos and L. Vinlent: Visual Motion Correspondence by Region-Based Approaches, *Proc. Asian Conf. on Computer Vision*, pp. 784-789, 1993

[33] Y. Chen and H. Nakatani: Image Region Correspondence by color and Structural Similarity, *IEICE Trans. on Information and Systems*, Vol. E76-d, No. 4, pp. 429-436, 1993

[34] S. Z. Li: Matching: Invariant to Translation, Rotation and Scale Changes, *Pattern Recognition*, Vol. 25, No. 6, pp. 583-594, 1992

[35] J. Weng, N. Ahuja, and T.S. Huang: Matching two perspective views, *IEEE Trans. on PAMI*, Vol. 14, pp. 806-825, 1992

[36] A. K. Jain, Y. Zhong and S. Lakshmanan, "Object Matching Using Deformable Template", *IEEE Trans. on PAMI*, Vol. 18, No. 3, pp. 267-277, 1994

[37] V. A. Anisimov and N. D. Gorsky: Fast Hierarchical Matching of an Arbitrarily Oriented Template, *Pattern Recognition Letters*, Vol. 14, No. 2, pp. 95-101, 1993

[38] D. W. Paglieroni, G. E. Ford and W. M. Tsujimoto: The Position-Orientation Masking Approach to Parametric Search for Template Matching, *IEEE Trans. on PAMI*, Vol. 16, No. 7, pp. 740-747, 1994

[39] M. Bhanu and O. D. Faugeras: Shape Matching of Two-dimensional Objects, *IEEE Trans. on PAMI*, Vol. 6, No. 2, pp. 137-156, 1984

[40] N. Ueda and S. Suzuki: Learning Visual Models from Shape Contours Using Multiscale Convex/Concave Structure Matching, *IEEE Trans. on PAMI*, Vol. 15, No. 4, pp. 337-352, 1993

[41] C. A. Rothwell and A. Zisserman: Planar Object Recognition using Projective Shape Representation, *International Journal of Computer Vision*, Vol. 16, pp. 57-99, 1995

[42] S. Sclarroff and A. P. Pentland: Modal Matching for Correspondence and Recognition, *IEEE Trans. on PAMI*, Vol. 17, No. 6, pp. 545-561, 1995

[43] Y. Lamdan and H. Wolfson: Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, *Proc. of Image Understanding Workshop*, pp. 238-249, 1988

[44] R. Hummel and H. Wolfson: Affine Invariant Matching, *Proc. of IUW*, pp. 351-364, 1988

[45] D. M. Gavrila and F. C. A. Groen: 3D Object Recognition from 2D Images using Geometric Hashing, *Pattern Recognition Letters*, Vol. 13, pp. 263-278, 1992

[46] M. C. K. Yang and J. S. Lee: Object Identification from Multiple Images Based on Point Matching Under a General Transformation, *IEEE Trans. on PAMI*, Vol. 16, No. 7, pp. 751-576, 1994

[47] O. Munkel: Aspect-trees: Generation and Interpretation, *Computer Vision and Image Understanding*, Vol. 61. No. 3, pp. 365-386, 1995

[48] S. Zhang, G. D. Sullivan and K. D. Baker: The Automatic Construction of a View-Independent Relational Model for 3D Object Recognition, *IEEE Trans. on PAMI*, Vol. 15, No. 6. pp. 531-544, 1993

[49] H. Murase: Visual Learning and Recognition of 3-D Objects from Appearance, *International Journal of Computer Vision*, Vol. 14, pp. 5-24, 1995

[50] K. Kanatani: *Geometric Computation for Machine Vision*, Oxford Science Publications, 1993

[51] H. C. Longuet-Higgins: A Computer Algorithm for Motion Reconstruction a Scene from Two Projections, *Nature*, Vol. 293, No. 10, pp. 133-135, 1981

[52] R. Y. Tsai and T. S. Huang: Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces, *IEEE Trans. on PAMI*, Vol. PAMI-6, No. 1, pp.13-27, 1984

[53] X. Zhuang: A Simplification to Linear Two-view Motion Algorithm, *Computer Vision, Graphics, and Image Processing*, Vol. 46, No. 1, pp. 175-178, 1989

[54] J. Weng and T. S. Huang: Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation, *IEEE Trans on PAMI*, vol. 11, No. 5, pp. 451-476, 1989

[55] J. Philip: Estimation of Three-dimensional Motion of Rigid Objects from Noisy Observations, *IEEE Trans. on PAMI*, Vol. 13, No. 1, pp. 61-66, 1991

[56] A. R. Bruss and B. K. P. Horn: Passive Navigation, *Computer Vision, Graphics and Image Processing*, Vol. 21, pp. 3-20, 1983

[57] K. Prazdny: Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving, *Computer Vision, Graphics and Image Processing*, Vol. 17, pp. 238-248, 1981

[58] E. D. Micheli, V. Toree and S. Uras: The Accuract of the Computation of Optical Flow and of the Recovery of Motion Parameters, *IEEE Trans. on PAMI*, Vol. 15, No. 5, pp. 343-447, 1993

[59] R. Hummel and V. Sundareswaran: Motion Parameter Estimation from Global Flow Field Data, *IEEE trans. on PAMI*, Vol. 15, No. 5, pp. 459-476, 1993

[60] G. J. Young and R. Chellappa: Statistical Analysis of Inherent Ambiguities in Recovering 3-D Motion from a Noisy Flow Field, *IEEE trans. on PAMI*, Vol. 14, No. 10, pp. 995-1013, 1992

[61] J. Kittler and J. Illingworth: Minimum Error Thresholding, *Pattern Recognition*, Vol. 19, pp. 41-47, 1986

[62] H. Bunke: Inexact graph matching for structural pattern recognition", *Pattern Recognition Letters*, Vol. 1, No. 4, pp. 245-253, 1983

[63] M. A. Eshera and K. S. Fu: A Graph Distance Measure for Image Analysis, *IEEE Trans. on Syst Man Cybern*, Vol. 14, No. 3, pp. 398-408, 1984

[64] L. G. Shapiro and R. M. Haralick: Structural Descriptions and Inexact Matching, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 3, No. 5, pp. 504-519, 1981

[65] A. Sanfeliu and K. S. Fu: A Distance Measure Between Attributed Relational Graphs for Pattern Recognition, *IEEE Trans. on Syst Man Cybern*, Vol. 13, No. 3, pp. 353-362, 1983

[66] E. K. Wong: Model Matching in Robot Vision by Subgraph Isomorphic, *Pattern Recognition*, Vol. 25, No. 3, pp. 287-303, 1992

[67] J. E. Hopcroft and R. E. Tarjan: Isomorphism of Planar Graphs, *Complexity of Computer computation*, R. E. Miller and J. W. Thather Eds., Plenum, New York, pp. 131-152, 1972

[68] H. Bunke and U. Buhler: Recent Advances in String Matching, *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke Ed., World Scientific Publishing, pp. 3-21, 1992

[69] M. Maes: On a Cyclic String-To-String Correction Problem, *Info. Process. Letters*, No. 35, pp. 73-78, 1990

[70] W. H. Tsai and S. S. Yu: Attributed String Matching with Merging for Shape Recognition, *IEEE Trans. on. PAMI*, Vol. 7, No. 4, pp.453-462, 1985

[71] D. Yagi, K. Abe and H. Nakatani: Segmentation of Color Aerial Photographs Using HSV Color Models, *Proc. of MVA*, pp. 367-370, 1992

[72] R. A. Wagner and M. J. Fischer: The string-to-string correction problem, *Journal of ACM*, Vol. 21, pp. 168-173, 1974

[73] K. Takahashi, H. Nakatani and K. Abe: Color Image Segmentation Using ISODATA Clustering Method, *Proc. of 2th ACCV*, Vol. I, pp. 523-527, 1995

[74] Y. L. Chen: Image Region Correspondence Based on Structural Similarity and Color Information, Doctor Thesis, Shizuoka University, 1993

A. 関連論文

A1. C. Wang and K. Abe, Region Correspondence for Color Scene Images Taken from Different Viewpoints, Proc. of Machine Vision Application'94, pp. 26-29, 1994

A2. 王、阿部：二つの視点から撮影した同一情景の領域間対応づけ, 情報処理学会論文誌, Vol. 36, No. 10, pp. 2253-2262, 1995

A3. C. Wang and K. Abe: Region Correspondence by Inexact Attributed Planar Graph Matching, Proc. 5th ICCV, pp. 440-447, 1995

A4. 王、阿部：奥行き距離における段階的なマッチング結果の統合によるステレオ マッチング, 静岡大学大学院電子科学研究科研究報告, Vol. 17, pp.77-87, 1996

A5. C. Wang and K. Abe: Stereo Matching by Integrating Piecewise Surfaces Matched in Subranges of Depth、 To appear in Proc. of 13th ICPR, 1996

B. その他の論文

B1. K.Abe, F. Mizutani and C. Wang: Thinning of Grey-scale Images with Combined Sequential and Parallel Conditions for Pixel Removal, IEEE Trans. Systems, Man, and Cybernetics, Vol. 24, No. 2, pp. 294-299, 1994

B2. C. Wang and K. Abe: A Method for Gray-Scale Image Thinning: The Case Without Region Specification for Thinning, Proc. of 11th ICPR, pp. 404-407 1992

C. 口頭発表

C1. 阿部、王： 画像のマッチング, 平成５年度東海支部連合大会講演論文集, pp. S-67-S-68, 1993

C2. 王、阿部： オプティカルフローによる画像領域マッチング結果の修正, 1994 年度電子通信学会春季大会講演論文集, p. D-565, 1994

C3. 王、阿部： 二つの視点から撮影した同一情景の領域間対応づけ, 画像の認識・理解シンポジウム講演論文集 II, pp. 89-96, 1994

C4. 王、阿部： 属性つき平面グラフの最大類似マッチングを用いた領域間対応づけ, 信学技報, PRU-127, No. 2, pp. 71-78, 1995

C5. 王、阿部： 奥行き距離における段階的なマッチング結果の統合によるステレオ マッチング, MIRU'96 (採録決定)