G. Vosselman

Relational Matching

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
W-7500 Karlsruhe, FRG

Juris Hartmanis
Department of Computer Science
Cornell University
5149 Upson Hall
Ithaca, NY 14853, USA

Author

George Vosselman
Institute of Photogrammetry, University of Stuttgart
Keplerstr. 11, W-7000 Stuttgart 1, FRG

Dissertation der Landwirtschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn
D 98

CR Subject Classification (1991): I.2.8, I.5.4, H.1.1

ISBN 3-540-55798-9 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-55798-9 Springer-Verlag New York Berlin Heidelberg

Typesetting: Camera ready by author/editor
45/3140-543210 - Printed on acid-free paper

Preface

This book is the result of an elementary study on relational matching. Relational matching is a method for finding the best correspondences between structural descriptions. In computer vision it is widely used for the recognition and location of objects in digital images. For this purpose, the digital images and the object models are represented by structural descriptions. The matching algorithm then has to determine which image elements and object model parts correspond.

This study particularly focuses on the evaluation of the correspondences. In order to find the best match, one needs a measure to evaluate the quality of a match. This measure usually quantifies the similarity between the image and the model elements. This strategy is based on the assumption that corresponding elements will have similar characteristics (like size, shape, etc.). This study reviews the evaluation measures that have been suggested over the past few decades and presents a new measure that is based on information theory. This new measure is integrated into tree search methods that are utilized to find the best match.

The resulting relational matching theory hence combines matching strategies, information theory, and tree search methods. Because the reader may not be familiar with all aspects, comprehensive introductions are given to these topics.

I would like to thank my supervisor, Prof. Wolfgang Förstner, for the pleasant cooperation and the many interesting discussions we had. I would also like to thank the German Research Society, which financed the Special Research Program "High Precision Navigation" (SFB 228) at the University of Stuttgart. The research for this thesis was performed within the image processing project of this research program.

Stuttgart, June 1992 George Vosselman

Contents

I Introduction to relational matching

Part I

Introduction to relational matching

1 Computer Vision and Matching

Computer Vision is the kind of image processing that results in information about the spatial and physical properties of the objects that have been recorded in digital images [Rosenfeld and Kak 1976, Ballard and Brown 1982]. It has outgrown all other areas within digital image processing and is the dominating theme on most conferences on image processing.

Traditionally Computer Vision research was done by electrotechnical engineers and physicists, but quite soon many other disciplines like medicine, mathematics, computer science, psychology, photogrammetry and other engineering disciplines joined in. This resulted in a very broad domain for applications of Computer Vision. Although the fields of applications often are very different, the image processing techniques that are used and the problems that have to be solved have very much in common. One of these common problems is the *correspondence problem*, or *matching problem*.

1.1 Correspondence problems

The correspondence problem is the problem of finding the corresponding features out of two or more data descriptions. It is one of the central and most difficult problems in Computer Vision: surface reconstruction from stereo images requires corresponding points from two images, recorded objects can only be recognized by matching image descriptions with object models and automatic navigation systems have to match images to digital maps.

The data of these images, object models, or maps can be described at different levels of abstraction.

At the lowest level images are described by their grey values. Algorithms matching small patches of grey value images have been developed to determine parallaxes [Helava 1976, Förstner and Pertl 1986] or to measure coordinates of signalized points by matching the image with an artificial mask.

At the next level features like points, lines and regions that can be extracted from the images are used for the matching. Surface reconstruction of recorded objects or terrains can be performed by matching the features of stereo images [Marr and Poggio 1979, Barnard and Thompson 1980, Hahn 1989]. Tracking features through sequences of images is used for the calculation of paths covered by vehicles. Image sequences can also be used for building up a three dimensional scene description [Gennery 1977, Moravec 1977]. Comparison of image features with features of digital maps is applied to automated map revision, and computer guided cartography makes use of general models to interpret image features [Nagao *et al.* 1979, McKeown *et al.* 1985, Fua and Hanson 1988].

At the highest level the data descriptions that are matched not only contain features but also the interrelationships between the features. Problems like three-dimensional object recognition and location [Binford 1982], navigation (mapping an image to a map) [Faugeras and Price 1981, Nevatia and Price 1982] can only be solved when the global context of the features is known. The topological and geometrical relations between features contain important information that is needed to constrain the large space of possible mappings between the features.

Because of these constraints, such *relational descriptions* can be matched without having a priori knowledge about the spatial relationship between them. Matching methods applied to lower level data descriptions need such knowledge, e.g. the relative orientation between two images resulting in the epipolar constraint. If such information is not available or not good enough only high level data descriptions can be matched. Reflecting the recent interest in robust methods, the computer vision literature of the last few years shows an increased number of publications dealing with high level representations (e.g. [Mohan and Nevatia 1989, Straforini *et al.* 1990]).

This thesis is an investigation into the *relational matching method.* Apart from *relaxation labeling* this method is the only method that can match relational descriptions. In contrast to the relaxation labeling, the relational matching method always finds the best mapping between the features of the descriptions. This thesis discusses and further develops the theories related to the relational matching method.

1.2 Relational matching theory

Theories, that state a problem, define the optimal solution and allow the derivation of an algorithm to find it, are sparse in computer vision. Many algorithms are designed on an ad hoc basis and have lots of tuning parameters (e.g. weight factors) which are hard to interpret and which optimal values often depend on the data the algorithms are to process. Many algorithms work, often only tested on a few images, without really understanding why they do work. A sound theory behind the algorithms is missing. Already several authors have complained about this situation and have urged the necessity of theory development, despite the complexity of computer vision problems [Haralick 1985, Price 1985].

Maybe due to their complaints, there is an increase in the number of papers founding new algorithms on the probability theory and the information theory. The information theory thereby gives an interesting alternative view on probabilities. Using principles from these well established theories, solutions could be derived to quite some fundamental problems in computer vision. Leclerc [1988], e.g., presented a new approach to image segmentation based upon the "minimum description length" principle of the information theory. Using the same principle, Fua and Hanson [1988] evaluate the detection and location of houses and roads in aerial images which are defined by generic models. Boyer and Kak [1986] defined an information-theoretic measure for comparing relational descriptions and Wallace and Kanade [1989] reported on a new clustering scheme which could be used for object recognition or perceptual grouping of features in a data set.

In this thesis, an attempt is made to describe and further develop the relational matching method using the information theory. Relational matching compares two relational descriptions. It has to find the best mapping from features in one description to the features of the other description. In order to find the best mapping, it has to measure the similarity between the features which are mapped to each other. This requires an evaluation function, a measure for comparing relational descriptions. We will discuss the evaluation function of Boyer and Kak [1986] and show that, although this function already is much better than previously published measures, it still has some drawbacks. Optimizing this function may lead to non-optimal mappings. Based on this analysis we will suggest a new evaluation function that eliminates these drawbacks.

Having an evaluation function, we can now look for the best mapping. Relational matching problems are solved with *tree search methods* that have been developed in the domain of the artificial intelligence. The tree search methods utilize the evaluation function together with heuristics to efficiently select the best mapping from the large space of possible mappings. As the value of the new function has to be maximized, whereas the values of usual functions have to be minimized, the tree search methods will be slightly adapted for the purpose of the new evaluation function.

The extraction of relational descriptions from the raw data (e.g. grey value or range images) of course is important to the matching method, because good descriptions, i.e. abstract descriptions without segmentation errors, are easier to match than bad ones. The description extraction will, however, not be a primary topic of this thesis. The stress will be laid upon the evaluation function and the search method of relational matching.

1.3 Organization of the thesis

Before going into the aspects of relational matching, chapter 2 first gives an overview and classification of the matching methods used in computer vision in order to have a better view on the characteristics of relational matching and to be able to recognize the advantages and disadvantages of relational matching. After that, chapter 3 gives a more formal definition of the relational matching problem and reviews the early development of relational matching in literature. Chapters 2 and 3 prepare for chapter 4 that outlines several problems in matching relational descriptions and that presents the contributions of this thesis in solving them. Chapter 4 concludes the introductory part of this thesis. The reader familiar with the matching literature may skip chapters 2 and 3, or have a short look at the notation introduced in chapter 3, and continue with the specific topics of this thesis that will be discussed in chapter 4.

The second part of this thesis contains old and new theories about relational matching. The method is described in general terms as a method that matches two relational descriptions. This description applies to all kinds of matching problems, e.g. image to image matching, image to model matching, etc. Because the information theory will play an important role in the chapter on the evaluation function, chapter 5 deals with the basic elements of the information theory. We will also derive some useful properties of information that will be used in the subsequent chapters. In chapter 6 we will review existing functions for comparing relational descriptions and develop the new evaluation function. Chapter 7 describes the tree search methods and shows how the new evaluation function can be combined with these methods.

In the third and last part of this thesis we show how the developed theories can be applied to the problem of locating three-dimensional objects in digital images. This problem is just one of the many problems that can be solved by relational matching and serves to show the new theories into some more detail. In chapter 8 we will describe the method that has been used to extract the relational descriptions from grey value images. In chapter 9 we specify the evaluation function for the object location problem into more detail and describe methods by which the components of this function can be calculated. The employed search method and heuristics and their performance are discussed in chapter 10.

Chapter 11 finally gives a summary of the results achieved in this thesis and an outlook to further improvements of the relational matching method.

2 A classification of matching methods

Matching algorithms may be analyzed by posing three fundamental questions:

- **"What kind of data is matched?"** As sketched in the previous chapter the data which is to be matched can be described at several levels of abstraction. This level of representation strongly influences the definition and performance of the matching algorithm.

- **"What is the best match?"** All matching procedures define the best match to be the one which optimizes the value of some evaluation function.

- **"How to find the best match?"** The central part of a matching algorithm describes how to find the match with the optimal value of the evaluation function.

After working out these questions, we will describe the use of multi-level descriptions in matching. This increasing popular technique speeds up the search and affects both the data description and the search method. Finally, we describe a number of well known matching methods along the lines defined above and discuss the advantages and disadvantages of the different algorithms.

2.1 Data description

A matching method matches two data descriptions, or, more precisely, it tries to find the best mapping between the basic elements of the descriptions. The data descriptions range from matrices of pixel grey values to relational descriptions.

2.1.1 Primitives

The basic elements are called the primitives of the descriptions. There are many types of primitives. In case of an image, the most obvious primitive type is the pixel. For many matching methods, however, a description by pixels is much to large to perform a search for the best match within a reasonable time limit. Such methods then use descriptions with other, more compact, types of primitives by which the essential information of the image can be stored in a smaller amount of primitives. These are primitives like points, lines or regions[1]. At this level one can also compose descriptions of object models or maps, using the same types of primitives. For describing three dimensional models or range images one can also use volumetric primitives like generalized cylinders.

[1]Such primitives can be extracted by a large number of image segmentation methods. An example is given in chapter 8.

The primitives are described by their characteristics. These are called attributes. Similarity in attribute values is the major guide to find the best primitive to primitive mapping. A primitive can be described by the names and the values of its attributes. Thus, as an example of the simplest primitive, one could describe a pixel no. 83 with row coordinate 119, column coordinate 34 and intensity 75 by primitive p_{83}

$$p_{83} = \{ \text{(row 119) (column 34) (intensity 75)} \}$$

and region no. 5 defined by the centroid coordinates, its surface size and border length by primitive p_5

$$p_5 = \{ \text{(centre-row 203) (centre-column 138) (surface 3892) (border 294)} \}$$

The attributes used in the descriptions above are all numerical, but, clearly, they can also have a symbolic nature, like for instance the polygon primitive below

$$p_2 = \{ \text{(length 32.4) (closed true) (shape circular)} \}$$

describing a closed polygon representing a circle with circumference 32.4.

A description may use different types of primitives (e.g. primitives describing polygons and primitives describing regions). In such cases, the primitive type may also be considered a symbolic attribute to the primitive:

$$\begin{aligned} p_2 &= \{ \text{(type polygon) (length 32.4) (closed true) (shape circular)} \} \\ p_5 &= \{ \text{(type region) (centre-row 203) (centre-column 138)} \\ &\quad\ \text{(surface 3892) (border 294)} \} \end{aligned}$$

Data descriptions, that merely are lists of primitives described by their attribute names and values, are called *feature based descriptions.*

2.1.2 Relations

Each primitive in a feature based description is a description of a small part of the image, map or model. All parts are considered to be completely independent of each other. Clearly, such descriptions lack all contextual information, whereas this kind of information may be very useful to a matching algorithm.

Consider, for example, the above descriptions with polygons and regions. If, in addition to the attributes of the primitives, it would be known that polygon p_2 is (a part of) the contour of region p_5 and one would like to find the corresponding polygon and region in another description, one can restrict the search to those pairs of a polygon and a region which not only have attributes similar to those of p_2 and p_5, but which also share the relationship that the polygon is (a part of) the contour of the region.

Thus, it is useful to extend the description of the primitives by a description of the interrelationships among the primitives. Such a description of primitives and their interrelationships is called a *structural* or a *relational description.*

The relationships can be represented in relation tuples. The relation "contour" e.g. is a binary relation between a polygon primitive and a region primitive. Thus, the "contour"-relation tuple of polygon p_2 and region p_5 would be

$$\{ \ p_2 \ p_5 \ \}$$

Collecting all pairs of polygons and regions for which the contour relation holds, the set of these binary tuples could look like

$$(\ \{ \ p_2 \ p_5 \ \} \ \{ \ p_2 \ p_4 \ \} \cdots \{ \ p_1 \ p_7 \ \} \ )$$

Finally, one has to mark that this tuple list concerns the relation contour, which is just one relation out of many possible relations. This is usually done by making a pair of the relation name (e.g. contour) and the relation tuple list. E.g.

$$(\ \text{contour} \quad (\ \{ \ p_2 \ p_5 \ \} \ \{ \ p_2 \ p_4 \ \} \cdots \{ \ p_1 \ p_7 \ \} \ ) \ )$$

Just like the primitives are described by their attributes, the relation tuples may also have attributes. In case of the contour relation, for instance, one may want to know the percentage of the region contour that is covered by the polygon, or the minimum distance between the polygon and the centre of the region. Storing both, the relation pair would be

$$\begin{array}{lll} (\ \text{contour} & (& \{ \ p_2 \ p_5 \ (\ \text{coverage } 100 \ ) \ (\ \text{distance } 10.3 \ ) \ \} \\ & & \{ \ p_2 \ p_4 \ (\ \text{coverage } 34 \ ) \ (\ \text{distance } 5.7 \ ) \ \} \\ & & \qquad \ldots \\ & & \{ \ p_1 \ p_7 \ (\ \text{coverage } 100 \ ) \ (\ \text{distance } 25.4 \ ) \ \} \ ) \ ) \end{array}$$

Such relational descriptions tell a lot more about the data than the feature based descriptions do. They are needed for those matching problems that can not be solved without contextual information.

2.1.3 The image as a function of coordinates

Although it is possible to specify a grey value image by a set of pixels, with every pixel having a row coordinate, a column coordinate and a grey value attribute, the grey value is often looked upon as a function of the row and column coordinates:

$$g(r,c) \qquad r = 1,2,\ldots,N_r \qquad c = 1,2,\ldots,N_c \tag{2-1}$$

When extending this description with an interpolation rule, the grey value can be considered a function over a continuous space of row and column coordinates. This interpretation is essential to some image to image matching methods, namely the area based matching methods.

2.2 The match evaluation function

The match evaluation function is a function on two data descriptions which has to guide the search method in finding the best match. The best match between two descriptions is the mapping for which the corresponding primitives of the descriptions show the best similarity in their attributes and, in case of relational descriptions, their relations.

The problem is with the definition of similarity. If we would only have one type of attribute with a numerical value, e.g. the grey value of a pixel primitive, this is still easy: one may say that two pixels are similar if the absolute difference of the grey values is small. However, in the case of two numerical attributes, e.g. the average grey value and the size of a region, this already becomes less trivial. What is more important: similar grey values or similar sizes? One may solve this problem by weighting the attribute values according to their standard deviations. But the similarity definition becomes even more complicated if one wants to combine these numerical attributes with symbolical attributes, like a predicate indicating whether a polygon is open or closed or a polygon shape classificator with categories "straight", "sinoidal" and "polynomial".

Looking for the mapping with the best similarity value (however this value may be defined) often results in a very time consuming search. Therefore, the space of possible mappings needs to be reduced. This can be achieved by imposing constraints. Two types of constraints have to be discerned: hard constraints and soft constraints. Hard constraints define the limits of the search space. In case the similarity function is a cost function, mappings outside these limits have infinitely high costs. Soft constraints, on the contrary, do not define which mappings are possible and which are not, but define a relative preference over the space of possible mappings. Soft constraints are used to implement heuristics in the evaluation function, which tell the search method where it will be more likely to find the best mapping.

2.2.1 Similarity measures

Similarity of two descriptions is usually defined as a cost function or a distance function. These costs are to be minimized and are zero only if both descriptions are identical. The costs of the mapping are defined by the similarity of the attribute values of the primitives (and relations tuples) that are mapped to each other. Usually, all attributes of all primitives and relations tuples are considered independent. Then, having defined the costs of a difference in attribute values, the costs of one primitive (or relation tuple) correspondence can be defined as the sum of the attribute costs, summed over all attributes of that primitive (or the relation tuple). Similarly the costs of a mapping is the sum of the costs of all primitive (and relation tuples) correspondences.

Thus suppose we have two feature based descriptions P and Q where both descriptions are sets of primitives $\{p_1, p_2, \ldots, p_N\}$ resp. $\{q_1, q_2, \ldots, q_N\}$. Further suppose that all primitives are described by N_a attributes a_k with values v_k and that a mapping h is given which maps the primitives of P to the primitives of Q, such that, when $h(p_i) = q_j$, primitive q_j of description Q is considered to be the corresponding primitive of primitive p_i of description P. Then the costs of the instantiation of p_i with q_j are defined as the sum of the attribute correspondence costs over all attributes.

$$\text{costs}(p_i, q_j) = \sum_{k=1}^{N_a} \text{costs}(v_k(p_i), v_k(q_j)) \tag{2-2}$$

where $v_k(p_i)$ denote the value of the kth attribute of the ith primitive of description P. The costs of the mapping h can be defined by:

$$\text{costs}(h) = \sum_{i=1}^{N} \text{costs}(p_i, h(p_i)) \tag{2-3}$$

What is left is the definition of the costs which are imposed if the attribute values of two corresponding primitives are not the same. Let us first consider numeric values. If there is only one attribute, the absolute or the square of the difference between the two values is often taken as the distance measure. This results in the L_1 resp. the L_2 norm. Problems arise if there are several different attributes. For instance, two feature based image descriptions exist of regions. All region primitives are ellipses which are described by two attributes: a roundness attribute which is the quotient of the shorter and the longer semi-axis and a size attribute which is the number of pixels within the region. All values of the roundness attribute are somewhere between 0 and 1, whereas the sizes of the regions may vary from a few pixels (say 10) to the number of pixels in the image. Clearly the values of the latter attribute are much larger and the difference between two region size attribute values will usually also be much larger than the difference between the values of two roundness attributes. Thus, when taking the costs of an instantiation of two region primitives to be the sum of the absolute or square values of the differences of the roundness and size values, the influence of the roundness attribute will be marginal and the best mapping will be the mapping with the best similarity in region sizes.

To get a more balanced measure, frequent use is made of attribute value transformations so that the range of the attribute values is about the same for all attributes. This may be achieved by scaling the values v into the interval [0,1] (a) or in case of Gaussian distributed attributes by normalizing the distribution (b)

$$(a)\ \ v' = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \qquad\qquad (b)\ \ v' = \frac{v - \mu_v}{\sigma_v} \tag{2-4}$$

An even more sophisticated transformation is used by the Mahalonobis distance which builds a square sum which also corrects for dependencies between the attributes.

These linear transformations are to assure that all attributes are treated as equally important, i.e. on the average contribute a same amount to the evaluation function. However, this may not be the optimum. For example, suppose that two colour images are represented by regions which are described by their size and their average hue value. If we want to match two such descriptions of images taken from different positions, it is clear that the values of the size attribute are not invariant against a change in the position of the camera whereas the colours of the recorded objects, and thus the hue attribute values, will be constant. Hence, similar hue values contain more information about the correctness of a mapping than similar region sizes and, thus, the hue attribute should have a greater contribution to the overall similarity value than the size attribute. Many researchers therefore choose a weighting factor for each type of attribute to indicate the importance of similarity of this attribute. Assuming that the attributes of corresponding features will have the same values, the weights are functions of the correctness of this assumption. The costs of an instantiation of two primitives then becomes the weighted sum of the attribute costs:

$$\text{costs}(p_i, q_j) = \sum_{k=1}^{N_a} w_k \cdot \text{costs}(v_k(p_i), v_k(q_j)) \tag{2-5}$$

where w_k is the weight of the kth attribute. The new evaluation function that will be developed in this thesis does not require such weights. The contributions of the attributes to the overall measure will already reflect their importance for the matching.

Beside the absolute or the square of the attribute value difference, there is a third approach for defining the costs of an attribute correspondence. This approach uses the probability theory and requires that the conditional probability (density) functions are known for all attributes. A conditional probability function $P_a(v_2|v_1)$ of attribute a defines how likely it is that the attribute of a primitive will take the value v_2 when it is known that the attribute of the corresponding primitive has value v_1. This conditional probability can be converted to a cost measure by taking the negation of the logarithm:

$$\text{costs}(v_1, v_2) = -\log P(v_2|v_1) \tag{2-6}$$

This cost definition is also called the conditional information of v_2 by v_1. Summing up these costs means multiplying of probabilities under the logarithm and minimizing the costs is equivalent to maximizing the product of all probabilities. Under the assumption that all attributes of all primitives are independent, the mapping with the fewest costs is the mapping with the highest likelihood: the maximum likelihood mapping.

If there is only one attribute which is Gaussian distributed and it is expected that the attribute values of corresponding primitives are the same, the minimization of the square sum of the value differences is also known to give the maximum likelihood estimate. Thus, in this case, the square sum may be regarded a special case of the conditional information, which uses a more general probabilistic approach. Similarly, the maximum likelihood estimation of Laplacian distributed variables is found by minimizing the sum of the absolute value differences.

Attributes of a description do not have to be numerical. They also may have symbolic values. The similarity between two symbolic values, of course, can not be determined by a difference between the values. The simplest approach is to test if the two values are the same and to add a penalty to the total costs if they are not. This is a rather crude method. The costs of a correspondence of a symbol "a" with a symbol "b" are the same as the correspondence of symbol "a" with a symbol "c". Both correspondences are penalized by the same amount since the values are different. It may, however, well be that e.g. correspondence ("a","b") is more likely than correspondence ("a","c"). This brings us to another method which can be used for judging correspondences of symbolic attributes. If one can determine the conditional probabilities for all attributes (e.g. $P(\text{"b"}|\text{"a"})$ and $P(\text{"c"}|\text{"a"})$), one will be able to differentiate between the likely and the less likely correspondences of attributes with symbolic values.

For descriptions which use both numeric and symbolic attributes, it thus can be concluded that the only sound way to measure similarity between two descriptions is a probabilistic approach. In chapter 6 we will further explore the advantages and problems of the conditional information and present a new probabilistic measure.

2.2.2 Constraints

Usually a priori knowledge is available to the matching process in the form that certain combinations of attribute values are considered completely incompatible. E.g. when attribute value v is considered completely incompatible with attribute value v', all primitives $q_j \in Q$ having attribute value v' will be discarded when looking for the corresponding primitive of primitive $p_i \in P$ having attribute value v. This may be called a hard constraint: under no condition p_i with attribute value v will be mapped to q_j with attribute value v'. If the evaluation function is a cost function, the costs of such a mapping would be indefinitely high.