Object Recognition from Range Data Prior to Segmentation *

S. Z. Li

Department of Electronic and Electrical Engineering
University of Surrey
Guilford, Surrey GU2 5XH, U.K.

Abstract

We propose an unconventional approach to range image understanding. It is unconventional in the sense that object matching and recognition is performed before the image is segmented. The motivation for this is our own perception of problems, which we will analyze in the paper, arising from the conventional paradigm of recognition after segmentation. The proposal consists of two levels: the representational level and the computational level. We illustrate how to represent scenes and models using attributed relational graphs (ARGs). The ARG representation describes surface points, rather than segmented primitives, as its nodes, their invariant properties and invariant relations between them. Object matching and recognition, as well as image segmentation, is determined by the optimal point-wise mapping from the image ARG and the model ARG. We formulate the problem of finding such mapping as one of constrained optimization. The solution is computed using the relaxation labeling method. We present some experiments of such optimal mapping.

Keywords — Invariant representation, shape matching, optimization.

*S. Z. Li is currently with the School of EEE, Nanyang Technological University, Singapore 2263.

Introduction

Three-dimensional computer vision aims at understanding the 3D world by analyzing input images. Marr [1] believed that *depth* information should be made explicit to achieve higher vision goals. Most computer vision research has been based on this belief. However, unlike human vision which is able to infer depth quickly and easily from intensity images, achieving 3D computer vision goals from intensity images presents formidable difficulties due to lack of constraints for solving visual computation problems therein. As sensor technology advances, range images from commercial range scanners [2] have become available. A range image or depth map is an array of pixel values which represent the distances from the sensed focus plane to object surfaces within the field of view. It provides explicit range information about visible surfaces on the image grid, and is largely independent of the illumination conditions of the environment and the reflectivity of the objects being sensed. Therefore, many perceptual problems arising from intensity images do not occur with range images. Because of these advantages, the last few years have seen increasing interest in 3D vision from range images. Despite the advantages, however, range image understanding still remains difficult and presents theoretic and pragmatic challenges [3, 4].

Nearly all of existing range image understanding approaches, *e.g.* those described in [5, 6, 7, 8, 9, 10, 11, 12, 13], follow the traditional computer vision paradigm developed from intensity image understanding (see [3] for a survey). The procedural flow of the paradigm is basically as follows [14]:

1. segmenting image(s),
2. constructing the scene description from the segmented image(s), and
3. matching and recognizing the scene to models by comparing the constructed scene description with model descriptions.

The construction of the scene description and object matching and recognition are always performed after segmentation.

Why Segmentation

This scenario seems to be necessary for intensity image understanding. Recovering the scene geometry, or computing the geometric description of the scene, from intensity images attempts to invert the optics by which the image is formed. Finding solutions to such inverse problems presents tremendous difficulties owing to the complexity of the image formation. At best, probably only a sketch of the description can be made from the image. This sketch is a simplified and abstracted picture of the scene, such as a description derived from the image(s). This is a substantial step intensity image understanding has to take before matching and recognition.

However, with available range data, the necessity of the scenario sequence becomes questionable. The formation of range images differs substantially from that of intensity images. Unlike intensity data Range data contains explicit, intrinsic depth information about visible surfaces of objects in the scene. It is much easier to compute true physical, geometric descriptions of object surfaces. In this case, it is feasible to build such descriptions directly from range data without segmentation though it cannot be done easily from intensity data. This is the advantage that we should best exploit from range data.

Argument for Segmentation

What are the benefits from segmentation, anyway? Segmentation aims at abstracting images into simplified "meaningful" entities or primitives by removing much of the information contained in the original image while retaining what is thought to be significant information. This results in abstract features or primitives such as feature points, lines, surface regions and volume bodies. Images as well as model objects are then described by these primitives, their properties and relations. Object matching and recognition is performed by comparing such descriptions based on some criterion of similarity or goodness of fit.

Segmentation offers the following advantages: Firstly, it provides a basis for a simplified sketch of the scene contained in the images. This is obviously so for intensity images. Pixel values of intensity images do not describe the physical shapes of the objects imaged and thus extracting descriptions about the object shapes directly from pixel intensities is very difficult. Segmentation then becomes a necessary step for deriving object descriptions. Secondly, segmentation reduces computational complexity tremendously compared with direct operation on raw data. For example, after segmentation, an image of size 512×512 can be simplified into, say, a few dozens of primitives. This makes the problem of matching tractable on available computational machinery. Thirdly, its subsequent primitive-based descriptions have made it possible to exploit existing AI techniques such as searching and reasoning to perform higher level vision tasks. The first advantage applies mainly for intensity images. The second point above is a definite advantage for both kinds of images, this is particularly true on sequential machines. The third depends on techniques used for the computation.

Argument against Segmentation

Despite the advantages, segmentation may be problematic. The problems occur at two levels: (1) the representation of objects and (2) the computation of image segmentation. (Note: at the representational level, segmentation means image segmentation and model decomposition; at the computational level, it means image segmentation only).

As far as the problem in representation is concerned, some information important for matching may have been undesirably removed by segmentation. As a result, descriptions derived based on segmented primitives may be inadequate to constrain the matching. In the following, we take curvature-sign based segmentation [15], which can be one of the best segmentation schemes for the purpose of matching and recognition from range data, as an example of demonstrating this problem.

Consider the 1D surfaces in Figure 1. On the left side of the figure are two different surfaces. Based on their curvature sign, which is the same as the sign of the second derivative in this

Figure 1: Different surfaces have the same results of segmentation based on the curvature.

1D case, both surfaces give the same segmentation which consists of a convex patch and a flat patch. If these surfaces can be considered similar in shape and should be classified as the same, then look at the other two on the right. The two surfaces are defined by $y_1(x) = e^{-x}\sin(x)$ and $y_2(x) = \cos(x)$, respectively, and are perceptually different in their shapes. Their second derivatives are $y_1''(x) = -2e^{-x}\cos(x)$ and $y_2''(x) = -\cos(x)$, respectively. These too have the same segmentation which is determined by the sign of $-\cos(x)$ in both cases. This is because the magnitude is lost after segmentation and mutual differences between points within a segmented primitive (a patch in these two examples) are removed. In these situations, the two surfaces are identical in terms of the segmentation and this can cause constraint inadequacy for matching. This problem has also been commented in [3]. Although patch attributes and relations, such as average curvature, surface area, curvature extrema and normal angle and distance between extremal points and so on, can be used as additional constraints, these features can change substantially given different occlusions.

These examples may exaggerate the real situation. For instance, polyhedra, cylinders and spheres, which have constant or piecewise constant curvatures, can be well described by segmentation based on curvature-signs. Recognition after segmentation can work well probably only in such special cases. Our task is to recognize free-shaped solid objects.

In the above discussion, we have assumed the segmentation could always be computed as desired. In reality, difficulties in segmenting images into primitives have been frustrating researchers for decades. It can often be the case that segmentation algorithms fail to produce desired results. It is sometimes difficult even to give a clear definition of segmentation. Take again the curvature sign based segmentation as an example. There is a clear definition of the sign of a quantity in theory. Clearly, this definition can not be used in noisy situations, sometimes even not in noise-free situations. What people do in practice is to classify surfaces as (1) very positively curved, (2) very negatively curved or (3) flat and slightly curved [15, 16, 17]. The signs are practically defined with respect to a small positive number, *i.e.* a threshold. Nonetheless, the difference between highly curved surfaces and slightly curved surfaces still remains fuzzy because that the threshold involved can not give rise to an adequate control of resulting surfaces in the presence of noise.

Recognition Before Segmentation

The above issues have motivated us to propose an unconventional approach to range image understanding in which the problems in representation could be overcome and the difficulties in computation of image segmentation is avoided. In our approach, the representations of both image and models are built upon un-segmented surface points and the matching takes place between these points. The proposal consists of two levels: the representational level and the computational level.

We represent both the scene in the image and model objects with attributed relational graphs (ARGs). However, unlike conventional ARGs based on segmented primitives [18, 19, 20, 14, 21, 22, 12, 13], an ARG here consists of raw surface points as nodes; similar to the conventional ARGs, the nodes in our ARG are attributed by unary node properties and constrained by binary relations between nodes. The properties and relations we use are chosen to be invariant under translation and rotation (we do not consider scale change a problem because the real scale of objects is usually at hand with range data), and stable to occlusion. They include (1) mean curvature and Gaussian curvature [15, 10] as the unary properties, and (2) the angle between surface tangents and the distance between surface points as binary relations. No clues of discontinuities are needed for matching in the proposed approach.

Clearly, the ARG built directly from surface data rather than segmented primitives is richer in information about the physical surfaces of objects since there is little loss of information. Thus, such an ARG should provide more powerful constraints for object matching and recognition based on shapes.

Object matching and recognition is determined by a node-wise mapping from the image ARG and the model ARG. We formulate the problem of finding such mapping as one of constrained optimization. The mapping optimizes a criterion which defines the similarity, or goodness of fit, between the two ARGs in terms of their properties and relations. The criterion is maximized by using the relaxation labeling method [23, 24].

Because each node corresponds to a surface point, the mapping actually maps from image surface points to model surface points. Once the mapping is found, matching and recognition is done, so is image segmentation. The pose of every matched object in the image can be easily determined based on the mapping found by using, *e.g.* minimum squared error fit between the matched image and corresponding models.

Our proposal has two obvious weaknesses: first an explosion of computational costs, which is particularly true on any conventional sequential machines, and second the problem of local optima. The former is due to the increase in the number of ARG nodes to be matched and the nature of the relaxation labeling method we use. To relieve the cost, we could use only a crucial subset of surface points rather than all of them in computing the mapping. Nonetheless, our algorithm is inherently parallel and distributive and well suited for implementing on SIMD architectures. The latter concerns an unsolved problem, that of global optimization which has attracted great deal of attention and in-depth research. It is not clear how serious this latter problem will turn out to be with true range images though we have found little difficulty in finding the global optimum according to the experiments we have done so far in small scale. The convexity properties of the criterion functional needs investigating.

Related Work

There are close relations between the curvature sign based approach to range image understanding, such as described in [11, 10, 12, 13] and our approach. The curvatures, angles and distances are used as constraints for matching. The difference is that the conventional ARG is built on segmented curvature-sign (+, 0, -) patches where the magnitude is lost; in contrast, our ARG is built on surface points (pixels) and the curvatures are not quantized.

The proposal of using pre-segmentation based representation has been raised before. Fischler [25] once suggested that some isomorphic/iconic representation is essential, and that the use of this type of representation, together with a global maximization paradigm for imposing a meaningful structure on a scene, could avoid many of the problems in the symbolic, *i.e.* the segmentation based, paradigm.

The structural representation, like the ARG, has been used in computer vision to represent models and scenes [18, 19, 20, 14, 21, 22, 26, 12, 13]. Many problems in computer vision, such as image sequence analysis, stereo correspondence, feature labeling, object recognition, scene understanding, *etc.* can be seen as examples of ARG matching. Inexact matching between images and models is commonly performed by explicit search approaches or by using implicit search methods such as relaxation labeling [27]. A goodness of fit is usually formulated to measure the optimality of matching.

In the following, we will present the proposed approach firstly at the representational level, secondly at the computational level and thirdly at the algorithm level and analyze the complexity of the algorithm. After that, we demonstrate some experiments on the matching and recognition of surfaces in 1D depth signals. Finally, we present a discussion.

Representation Based on Invariants

We are interested in surface-based representations because range image pixels directly correspond to the geometric locations of visible surface points. The availability of the locations of physical surfaces allows us to compute intrinsic and invariant properties and relations between surface points. These invariants enable us to compare object shapes based on our knowledge of geometry and play an important role in matching and recognition.

(Note: In this paper, we will use upper-case notations to refer to model quantities and lower-case notations to image quantities. For example, we will use $\mathbf{d}$ to denote the set of image surface points and $\mathbf{D}$ to denote the set of model surface points.)

Figure 2: The angle and distance relations between surface points.

A 3D surface can be described by a vector of three functions of two parameters. We use $\mathbf{z} = [z_1(x,y), z_2(x,y), z_3(x,y)]^T$ to represent a surface where z_1, z_2 and z_3 are functions of two parameters describing the location of 3D surface points with respect to three independent axes. A special form of the surface function is the graph surface $\mathbf{z} = [x, y, z(x,y)]^T$ also known as a Monge patch [28] where x, y are the two coordinates of a parameter plane. It is the conventional representation of an image. We assume that one surface function defines one surface. There is only *one* surface in the image since it can be expressed by *the* image surface function $z(x,y)$. Similarly, we have the notation $\mathbf{Z} = [Z_1, Z_2, Z_3]^T$ for the model surface.

The image surface $\mathbf{z}$ is then sampled into a set of m surface points denoted by

$$\mathbf{d} = \{1, 2, ..., m\} \tag{1}$$

Note that the sampling can be either homogeneous or inhomogeneous. In a homogeneous sampling scheme, the sampled points are evenly distributed on the image lattice, for example. In an inhomogeneous sampling scheme, highly curved surface areas are sampled densely while flat areas need be sampled only sparsely. An inhomogeneous sampling can be properly done only in modeling where object shapes are known in advance. Sampling schemes could be designed more rigorously based on Shannon's sampling theorem.

A model surface is sampled into a set of M points $\mathbf{D} = \{1, 2, ..., M\}$. We include into the set of model points a special point 0, or $NULL$, to which noisy image points and points of un-modeled surfaces should be matched. Thus the set of model points is

$$\mathbf{D} = \{0, 1, 2, ..., M\} \tag{2}$$

Multiple model surfaces can be merged into one surface which we shall call the universal model surface. Suppose there are $K = 1, ..., N$ model objects in the universal model and model K is sampled by M_K points such that $\mathbf{D}_K = \{1, 2, ..., M_K\}$. The set of universal model points is simply the union of all the individual sets: $\mathbf{D} = \cup_{K=1}^{N} \mathbf{D}_K \cup \{NULL\}$ with $M = \#\mathbf{D} = 1 + \sum_{K=1}^{N} M_K$ points.

The set of surface points are described by their unary properties and are linked, or constrained, to each other by binary relations between them. These properties and relations have to be viewpoint-invariant to make matching much easier. The invariant unary relations (properties) we use are mean curvature denoted by h; and Gaussian curvature denoted by k. They

Figure 3: Nodes i, j and k, their properties and relations.

encode information on local surface shape and are invariant to rotation and translation [28, 15]. The binary relations we use are the angle between each pair of tangent planes; and the distance between each pair of surface points. These two sources of binary constraints are illustrated on the left half of Figure 2 in which $\mathbf{z}$ is the surface function, $\theta = \theta(i, j)$ is the tangent angle between surface points i and j ($\mathbf{n}(i)$ and $\mathbf{n}(j)$ are the normal vectors) and $d = dist(i, j)$ is the distance between the two points. These relations encode constraint on surface shapes in a more global sense than the curvature. The relative distance also tends to impose, via continuity, a constraint on the ordering of mapping. The angle is invariant to 3D translation, rotation and scale changes; the distance is invariant to translation and rotation but not to scale changes. However, the real scale of the world is known if range data are available. Relations θ and d are stable to occlusion. Note that the angle is "directed"; it can vary from 0 to 360 degrees as illustrated in the right half of the figure. These properties and relations can computed from the raw surface function at points of interest where the image surface function is $\mathbf{z}$ while the universal model surface function is $\mathbf{Z}$.

The image ARG can then be constructed after these properties and relations are computed. It is a triple denoted by

$$\mathbf{g}(\mathbf{d}, \mathbf{r}_1, \mathbf{r}_2) \tag{3}$$

In this notation, $\mathbf{d}$ is a set of nodes, each node corresponding to a surface point; $\mathbf{r}_1 = [\mathbf{r}_1(1), \ldots \mathbf{r}_1(m)]$ is a vector of properties in which

$$\mathbf{r}_1(i) = [r_1^{(1)}(i), r_1^{(2)}(i)] = [h(i), k(i)] \tag{4}$$

is the vector of properties at i consisting of the mean curvature h and the Gaussian curvature k at the point; $\mathbf{r}_2 = [\mathbf{r}_2(i, j)]_{m \times m}$ is a matrix of binary relations in which

$$\mathbf{r}_2(i, j) = [r_2^{(1)}(i, j), r_2^{(2)}(i, j)] = [\theta(i, j), d(i, j)] \tag{5}$$

is the vector of relations between i and j consisting of the angle and the distance between the points. Figure 3 shows three nodes, i, j and k, of an ARG, their unary properties and binary relations.

Figure 4: Mapping from image to models

Similarly, we can build the model ARG $\mathbf{G}(\mathbf{D}, \mathbf{R}_1, \mathbf{R}_2)$. The computation of $\mathbf{R}_1$ and $\mathbf{R}_2$ for the model ARG can be done using two kinds of procedures: (1) CAD modeling packages [29] and (2) procedures used for computing $\mathbf{r}_1$ and $\mathbf{r}_2$ from images. The former can be used to build a whole 3D object model. The latter can be only applied to one of multiple views of models; the ARGs built from the multiple views may or may not be merged.

In the case of multiple model objects, the models are merged into the universal model and are represented by the universal model ARG. Nodes (points) in the universal ARG inherit their original unary properties. However there are no relations between surface points of different model objects. We define relations in the universal model ARG as follows

$$\mathbf{R}_2(I,J) = \begin{cases} \text{true values of relations} & \text{if } I, J \text{ come from the same model} \\ NONE & \text{otherwise} \end{cases} \tag{6}$$

That is, $\mathbf{R}_2(I,J)$ is computed from the model surfaces if both points I and J are from the same model; otherwise it is assigned a special value $NONE$, meaning there are no physical relations between I and J. Special treatment is needed in matching the $NONE$ relation, as will be discussed later.

Mapping from the Image to Models

In this approach, object matching and recognition, as well as segmentation of objects/background and overlapping objects are determined by a mapping from image surface points to model surface points. This is illustrated in Figure 4 where $\mathbf{d}$ denotes the set of image surface points and $\mathbf{D}$ denotes the set of points of the universal model surface. An image, shown on the left of the figure, consists of sub-parts of model objects (textured regions) and background (the white region). Anything yet to be modeled is treated as part of the background. There can be multiple objects in the universal model, as shown on the right, one of them being the $NULL$ model (the white region). The mapping, shown in arcs associating the image to the universal model, maps regions of images into regions of, possibly different, models. The background of the image is mapped

Figure 5: Mapping from image surface points (pixels) to model surface points.

to the $NULL$ model. More exactly the mapping takes place from the image ARG to the model ARG because the mapping is constrained by the properties and relations of the ARGs as will be illustrated shortly. The concept of the mapping is further illustrated in Figure 5. Three object models are shown on the top of the figure. An overlapping scene is shown in the lower-left. A camera views in the downwards direction, and takes an image as shown in the lower-right. This image contains the depth of visible 3D surfaces of the overlapping objects (shown in the height of the 1D signal). The mapping is a point-wise rather than patch-wise.

There can be numerous mappings from the point set $\mathbf{d}$ to the point set $\mathbf{D}$. Only one (or a few) of them gives the best interpretation. That is the one which optimally satisfies some criterion in terms of the properties and relations of the two ARGs. Before giving the criterion explicitly, we present in Figure 6 an example of a result obtained by maximizing the criterion. The figure contains two objects (bottom curves). The objects are rotated and superimposed and are viewed from above the superimposed or occluded scene. The image (the top curve) consists of the noisy depth of the visible part of the surfaces of the superimposed objects and some background regions. The mapping from image pixel points to model points is shown in lines linking the image and the models. It is based on similarity between surface shapes; it also tends to be organized in an orderly fashion, the ordering being a feature of Kohonen's self-organization maps [30]. The background points (the six left-most and three right-most points in the lowest part of the image) are mapped to 0, or $NULL$ (the mapping to $NULL$ is not shown

Figure 6: The optimal mapping from a 1D range image (top curve) to two object models (bottom curves).

in the figure). Derivatives for computing the curvature and the tangent angle are estimated using finite difference without any preprocessing like smoothing.

Computation of Optimal Mapping

We wish to find a mapping from $\mathbf{d}$ to $\mathbf{D}$ which maximizes a criterion called the gain. The gain is a functional of the mapping and measures the global relational similarity, or goodness of fit, between $\mathbf{g} = (\mathbf{d}, \mathbf{r}_1, \mathbf{r}_2)$ and $\mathbf{G} = (\mathbf{D}, \mathbf{R}_1, \mathbf{R}_2)$ given the mapping.

We formulate the problem as one of continuous relaxation labeling. Use

$$f = \{f(i,I) \in [0,1] \mid (i,I) \in \mathbf{L}\} \tag{7}$$

to denote the state of the mapping $\mathbf{g} \to \mathbf{G}$, where

$$\mathbf{L} = \mathbf{d} \times \mathbf{D} = \{(i,I) \mid i \in \mathbf{d}, I \in \mathbf{D}\} \tag{8}$$

is the set of correspondences between nodes i of the scene ARG and nodes I of the model ARG. State

$$f(i,I) \in [0,1] \tag{9}$$

represents the certainty with which $i \in \mathbf{d}$ is mapped to $I \in \mathbf{D}$

$$f : i \stackrel{f(i,I)}{\longrightarrow} I \tag{10}$$

A constraint on f is

$$\sum_{I:(i,I)\in\mathbf{L}} f(i,I) = 1 \qquad \forall i \in \mathbf{d} \tag{11}$$

The optimal mapping f^* maximizes a global gain functional

$$E(f^*) = \max_{f \in \mathbf{S}} E(f) \tag{12}$$

where $\mathbf{S}$ is the final unambiguous solution space

$$\mathbf{S} = \left\{ f \mid f \in \{0,1\}^{\mathbf{L}}, \sum_{(i,I) \in \mathbf{L}} f(i,I) = 1 \quad \forall i \in \mathbf{d} \right\} \tag{13}$$

in which $f(i,I) = 1$ or 0.

We define the gain given the mapping f by the functional

$$E(f) = \sum_{(i,I) \in \mathbf{L}} \sum_{(j,J) \in \mathbf{L}, j \neq i} \pi(i,I;j,J) f(i,I) f(j,J) \tag{14}$$

In the above, $\pi(i,I;j,J)$ is the interaction or compatibility between the matches (i,I) and (j,J). We choose it to be of the form

$$\pi(i,I;j,J) = \begin{cases} [\pi_2(i,I;j,J)\pi_1(i,I)\pi_1(j,J) - H]\pi_s(i,I;j,J) & \\ \qquad \text{if } I \neq 0,\ J \neq 0 \text{ and } \mathbf{R}_2(I,J) \neq NONE & \\ \pi_0 & \\ \qquad \text{if } I = 0 \text{ or } J = 0 & \\ 0 & \\ \qquad \text{if } \mathbf{R}_2(I,J) = NONE & \end{cases} \tag{15}$$

where $\pi_2(i,I;j,J) \in (0,1]$ measures the similarity between the binary relations $\mathbf{r}_2(i,j)$ of the scene ARG and $\mathbf{R}_2(I,J)$ of the model ARG; $\pi_1(i,I) \in (0,1]$ measures the similarity between the unary relations $\mathbf{r}_1(i)$ and $\mathbf{R}_1(I)$; $H \in [0,1)$ (we set $H = 0.2$) is a bias controlling the excitatory ($\pi > 0$) and inhibitory ($\pi < 0$) behavior of π; $\pi_0 \in [0,1)$ (we set $\pi_0 = 0.01$) is a number controlling the tendency with which scene nodes are mapped to 0; and $\pi_s(i,I;j,J) \in (0,1]$ localizes interaction spatially. The unary similarity is defined by

$$\pi_1(i,I) = \exp(-\sum_{k=1}^{2} |r_1^{(k)}(i) - R_1^{(k)}(I)| / \sigma_1^{(k)}) \tag{16}$$

where $\sigma_1^{(k)} > 0$ ($k = 1,2$) are parameters. We can choose $\sigma_1^{(k)} = 0^+$ for symbolic unary properties to give rise to symbolic interaction, or choose it to be an appropriate number for numerical properties. The binary similarity is defined by

$$\pi_2(i,I;j,J) = \exp(-\sum_{k=1}^{2} |r_2^{(k)}(i,j) - R_2^{(k)}(I,J)| / \sigma_2^{(k)}) \tag{17}$$

where $\sigma_2^{(k)} > 0$ ($k = 1,2$) are some parameters. Empirically, we find it necessary to limit the interaction to a certain spatial scope; the farther two points are apart, the smaller the interaction should be. The spatial localization is implemented by defining the localization term

$$\pi_s(i,I;j,J) = \exp(-d(i,j)/\sigma_s) \tag{18}$$

where $\sigma_s > 0$ is a parameter controlling the rate of the attenuation and $d(i,j)$ is the Euclidean distance between surface points i and j.

Algorithm for the Constrained Optimization

The optimization in Equation 12 is constrained by Equation 11. We perform the constrained maximization using relaxation labeling methods [23, 24]. Given the compatibility function π and an initial assignment $f^{(0)}$, the algorithm iterates the following two steps. First, the gradient at time t is computed from

$$q^{(t)}(i,I) = \partial E(f)/\partial f^{(t)}(i,I) = 2\sum_{(j,J)} \pi(i,I;j,J) f^{(t)}(j,J) \tag{19}$$

and second the assignment is updated by an appropriate scheme $f^{(t+1)}(i,I) \leftarrow \mathbf{\Psi}(f^{(t)}, q^{(t)})$ with $\mathbf{\Psi}$ being some updating operation. However instead of updating the whole of $f^{(t)}$ in one go, we update the $f^{(t)}(i)$ independently and in parallel for $i = 1,...,m$, *i.e.* for all image points, where $f^{(t)}(i) = \{p(i,I) | I \in \mathbf{D}\}$ is the i^{th} vector component of $f^{(t)}$. The updating is governed by the following rule

$$f^{(t+1)}(i) = f^{(t)}(i) + h_i u^{(t)}(i) \qquad \forall i \tag{20}$$

where $u^{(t)}(i)$ is the updating direction determined by the gradient projection as described in [24] and $0 < h_i < 1$ is a normalization factor which keeps $f^{(t+1)}(i)$ in the feasible solution space. It should be noted, however, that the gradient-based algorithm is not guaranteed to find the global solution. Readers are referred to [24] for more details of the continuous relaxation labeling algorithm.

Computational Complexity

The structure of the computation is illustrated in Figure 7. The computation consists of four stages:

1. Model building or model learning, *i.e.* the computation or construction of the universal model ARG $\mathbf{G}$ from model object surfaces (this can be done off-line),
2. Image measuring, *i.e.* the computation of the image ARG $\mathbf{g}$ from the image surface,
3. Computing the compatibility function or connections from $\mathbf{G}$ and $\mathbf{g}$, and
4. Relaxation labeling or computing the maximal mapping.

The computation in the last two stages takes place in the box marked "Mapping Network" in the figure. There are $m \times (M+1)$ units in the mapping network, the number being the size of $\mathbf{L} = \mathbf{d} \times \mathbf{D}$. The computation done at each unit, say (i,I), is to compute the gradient $q(i,I)$ in Equation 19. The updating operation takes place among units in each row, say i, using the updating rule in Equation 20. It is parallel for all i, *i.e.* for all points in the image. The updating terminates when no more changes occur in the state f of the network. The final f is usually unambiguous [24].

The computation in the first two stages is cheap compared to that of the later two stages. The computation in stages 3 and 4 is expensive in both space and time for conventional computers.

Figure 7: The structure of the computation.

Suppose, for example, we want to implement the algorithm for a task with an image of size $m = 100 \times 100 = 10^4$, a model of size $M = 100 \times 100$ and with 10 models in the universal model base. Then there are about $m \times 10M = 10^9$ units in this implementation. If we use one byte for a connection (compatibility coefficient), the storage requirement for all connections is $\sum_{K=1}^{10}(10^4 \times 10^4)^2/2 = 5 \times 10^{16}$ bytes! The time cost is obviously high for combinatorial optimization in a sequential implementation. The main consumption is in computing the gradient (Equation 19) and the gradient projection operation [24].

However, the computation could be done in near real time if we run the algorithm on a massively parallel distributed architecture, such as the Connection Machine, with a sufficient number of processing units. All connections, and the gradients, can be computed simultaneously for all $i = 1, ..., m$ and $I = 0, 1, ..., M$. Updating, *i.e.* the computation of the mapping, takes place simultaneously for all $i = 1, ..., m$. Now the time complexity depends mainly on the number of iterations required for the algorithm to converge. This number is dependent on m and M, the number of nodes in the scene ARG and the universal model ARG, respectively. Our observation is that this number is about a low order polynomial function of $m + M$ (see a table at the end of the next section). If an appropriately sized implementation takes a few hundred steps to converge, then it would be reasonable for real-time applications.

Figure 8: The optimal mapping from a 1D range image (top curve) to two model objects (bottom curves). Left: the image contains one of the two objects. Right: the image contains both objects.

Experiments

Surfaces in all experiments presented in this paper were 1D depth signals. From these signals, we computed the curvature $\kappa(i)$ at every point i, compressed its range using a logarithmic function (see below) and used the compressed quantity as the unary property

$$r_1^{(1)}(i) = \mathrm{sign}(\kappa(i)) \log(1 + 100 * |\kappa(i)|)$$

where $\mathrm{sign}(x)$ was the sign of x. The compression might not be very crucial for matching. We also computed the angle

$$r_2^{(1)}(i,j) = \theta(i,j)$$

(in degrees) between two tangent lines and the distance

$$r_2^{(2)}(i,j) = d(i,j)$$

(in pixels) between two points as the binary relations. Initially, we set

$$f^{(0)}(i,I) = \pi_1(i,I) / \sum_{I=0}^{M} \pi_1(i,I)$$

and then applied the relaxation labeling algorithm for the constrained optimization. The derivatives used to compute the angle and the curvature were estimated using finite difference without any preprocessing such as smoothing and thus the estimates were very noisy. Despite this, the algorithm still successfully performed the matching and recognition.

On the left in Figure 8, there are two model objects (bottom curves). One of them is pure linear (polygonal) and the other is a mixture of curved and linear. The scene (the top

Figure 9: The optimal mapping from a 1D range image (top curve) to two model objects (bottom curves). The image contains two instances of one of the two objects.

curve) consists of a rotated, partially occluded and noisy version of one of the objects and some background points (the lowest four points in the image).The mapping is shown by the lines linking the image and the models. The four background points are mapped to 0 (not shown). On the right of the figure, the two model objects are the same as those on the left, but the scene contains one more objects. The resulting mapping maps some pixels to one model surface, some to the other and some (background) to $NULL$. By comparing it with the case on the left side, we can see that the mapping in this case is the sum of the result from the previous case and the mapping to the object added to the scene; adding another object to the scene has disturbed the rest of the mapping very little.

More results are demonstrated in Figures 9-11. The scene in Figure 9 contains two instances of one model object. The resulting mapping maps correctly the non-background image pixels to the surface points of the instantiated model. The two model objects in Figure 10 correspond to the two surfaces on the right of Figure 1. The difference between the left and the right of the figure is that the image on (the top of) the left contains two impulses or discontinuities. However, this does not significantly affect the global results. Figure 11 shows two mappings to six model objects.

(Note: In all the mappings presented in these figures, the objects in the images have the same order (from left to right) as those in the models. This is to avoid the mapping from looking confusing (different groups of mapping crossing one another). The approach is independent of the order of the appearance of objects.)

Figure 10: The optimal mapping from a 1D range image (top curve) to two model objects (bottom curves). The two model objects have the same curvature sign as discussed previously.

Figure 11: Optimal mappings to six model objects.

Parameters used in the experiments are summarized in Table 1. Table 2 gives an impression of computational times for a sequential implementation in C on a Sun4 workstation, where N is the number of model objects, N_t is the number of iterations for the relaxation labeling process to converge and the times listed are average times in seconds. The times increase drastically as the number of nodes increases. So is the space. For example, there are $m = 48$ image points and $M = 96$ model points in the last experiment for mapping to six model objects. It requires $[48 \times (96+1)]^2$, *i.e.* 22×10^6, bytes for the storage of $\pi(i, I; j, J)$.

Parameter	H	π_0	$\sigma_1^{(1)}$	$\sigma_2^{(1)}$	$\sigma_2^{(2)}$	σ_s
Equation No.	15	15	16	17	17	18
Value	0.2	0.01	20	100	2/3	10

Table 1: Parameters for binary relations.

N	M	m	$M+m$	N_t	User Time	Sys Time
2	32	32	64	40	60.0	1.5
4	64	32	96	85	295.0	8.5
6	96	48	144	120	2420.0	19.0

Table 2: Computational times spent in a sequential implementation on a Sun4 workstation.

Discussion

We have proposed an unconventional approach to range image understanding in which object matching and recognition is performed before the image to be understood is segmented. This proposal is motivated by the problems arising from the conventional paradigm of recognition after segmentation. We use attributed relational graphs (ARGs) to represent objects. An ARG takes surface points rather than segmented primitives as its nodes, with the points constrained by their invariant properties and inter-relations. Object matching and recognition is performed by finding the mapping from the scene ARG nodes to the model ARG nodes which maximize a goodness of fit between the two ARGs in terms of their node properties and relations. No clues of discontinuities are needed in the proposed approach for surface matching.

The mapping associates points of similar shape between the image surface and the model surfaces and tends to be organized in an orderly fashion. Because the mapping is point-wise, it dictates image segmentation, the segmentation identifying figure/background and overlapping objects. The pose of every matched object in the image can be easily determined based on the mapping found by using, *e.g.* minimum square error fit between the matched image and corresponding models.

The matching is performed directly between the image points rather than segmented primitives, such as feature points, lines or patches. This distinguishes the proposal from most of other work in the conventional paradigm. Direct matching based on the geometric shapes of objects is possible with range images where the 3D positions of visible object surface points are available and thus intrinsic and invariant properties and relations can be computed directly from the image data.

The representational advantage of recognition before segmentation is that information important for matching and recognition is preserved so that the object descriptions derived provide more powerful constraints for matching and recognition. Another advantage is that the computational difficulties in segmenting images, which have been frustrating computer vision researchers for decades, are avoided.

We have presented a number of experiments on matching and recognizing overlapping arbitrarily-shaped 1D signals. These experiments have proved our promise to match and recognize range images before segmentation and demonstrated the effectiveness of the proposed approach to match rotated, partially occluded and noisy shapes, at least in one-dimension. The up to second derivatives for computing measurements such as the curvature and the tangent angle are estimated using finite difference without smoothing the signals; thus the computed measurements must be very noisy. Furthermore, the connections $\pi(i, I; j, J)$ are quantized into at most 256 levels so that they can be stored using one byte. Despite these, the program has successfully performed the matching and recognition based on these noisy measurements. This demonstrates the property of error tolerance of the approach. The program has worked quite stably for a range of different scenes.

Two obvious disadvantages of the proposal are: It is expensive both in time and space for conventional computers; it may lead to solutions which correspond to unfavorable local optima. To reduce the costs, we could use only some of the image pixels rather than all of them as the ARG nodes to be labeled. For example, given an image of size 256×256, we can compute the curvatures and the angles using the whole image to get good estimates. After that, sample part, *e.g.* 32×32, of the pixels as the ARG nodes, their properties and relations. The sampling could be denser in more highly curved areas and sparser in flatter areas. This strategy can also be used in model ARG building. The computational costs both in time and space can then be considerably reduced. Although it has not been a problem so far in our small-scale experiments, it is worthwhile to analyze the convexity properties of the functional. A Graduated Non-Convexity (GNC) algorithm [31] could be devised to approach the global optimum, for example, by varying parameter H or π_0 from 0 to its target value.

The involved parameters have so far been chosen by trial-and-error. Their selection could be done by a learning procedure. The learning here can be stated as follows: given two ARGs g and G and the optimal mapping f^* from g to G, consider $E(p \mid \mathbf{g}, \mathbf{G}, f^*)$ as the functional of a vector p of parameters, $p = [\sigma_1(1), ..., \sigma_1(K1); \sigma_2(1), ..., \sigma_2(K2); H, \pi_0]$; find the optimal p^* such that E is maximized. Alternatively, the optimal p^* could be determined using a training scheme: given a sequence of $\mathbf{g}_1$, $\mathbf{g}_2$, ..., and corresponding f_1^*, f_2^*, ..., adjust p based on $E(p \mid \mathbf{g}_k, \mathbf{G}, f_k^*)$ so that the final p^* best classifies all patterns.

Is the proposed approach applicable to range images on a 2D grid? In the image case, two curvatures, *i.e.* mean curvature and Gaussian curvature, are available, which is an advantage; the angle is between two tangent planes rather than between tangent lines while the distance is still between two surface points. However, there is one more dimension in images than in signals, which is a disadvangage. It is unclear whether the gain from the advantage can compensate the loss due to the disadvantage.

Is it applicable to sparse range data? Our opinion is that it could be if the objects contained in the image are polyhedral but not if they are free-formed. The reason is that much information about planar surface shapes can largely be recovered from sparse data if the objects are polyhedral; however, this is not true about free-formed surface shapes.

Acknowledgement

I would like to thank my supervisor Josef Kittler for his supervision and comments on this work. I would also like to thank Graeme Jones, Ali Khan, Maria Petrou and Geoff Nicholls in particular for their helpful comments. The work was supported by a scholarship from the British Council.

References

[1] D. Marr. *Vision.* W. H. Freeman and Co, San Francisco, 1982.

[2] R. A. Jarvis. "A perspective on range finding techniques for computer vision". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):122–139, 1983.

[3] J.P. Brady, N. Nandhakumar, and J.K. Aggarwal. "Recent progress in the recognition of objects from range data". *Image and Vision Computing*, 7(4):295–307, 1989.

[4] R. C. Jain and A. K. Jain. "Report on Range Image Understanding Workshop, East Lansing, Michigan, March 21–23, 1988". *Machine Vision and Applications*, 2:45–60, 1989.

[5] M. Oshima and Y. Shirai. "Object recognition using three-dimensional information". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(4):353–361, July 1983.

[6] P. J. Besl and R. C. Jain. "Three-Dimensional object recognition". *Computing Surveys*, 17(1):75–145, March 1985.

[7] R. C. Bolles and P. Horaud. "3DPO: A three-dimensional part orientation system". *International Journal of Robotic Research*, 5(3):3–26, Fall 1986.

[8] O. D. Faugeras and M. Hebert. "The representation, recognition and locating of 3D objects". *International Journal of Robotic Research*, 5(3):27–52, Fall 1986.

[9] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, 1987.

[10] B.C. Vemuri, A. Mitiche, and J.K. Aggarwal. "3D object representation from range data using intrincic surface properties". In K. Takeo, editor, *Three Dimensional Machine Vision.* Kluer, Boston, MA, 1987.

[11] A. K. Jain and R. Hoffman. Evidence-based recognition of 3D objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(6):783–802, November 1988.

[12] T. J. Fan, G. Medioni, and R. Nevatia. "Recognizing 3D objects using surface descriptions". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1140–1157, November 1989.

[13] S. Z. Li. "Towards 3D vision from range images: An optimization framework and parallel networks". *CVGIP: Image Understaning*, 55(3):231–260, May 1992.

[14] H. Ballard and M. Brown. *Computer Vision.* Prentice-Hall, 1982.

[15] P. J. Besl and R. C. Jain. "Invariant surface characteristics for 3D object recognition in range images". *Computer Vision, Graphics and Image Processing*, 33:33–80, 1986.

[16] Richard L. Hoffman and Anil K. Jain. "Segmentation and classification of range images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):608–620, September 1987.

[17] S. Z. Li. "Invariant surface segmentation through energy minimization with discontinuities". *International Journal of Computer Vision*, 5(2):161–194, 1990.

[18] M. Fischler and R. Elschlager. "The representation and matching of pictorial structures". *IEEE Transactions on Computers*, C-22:67–92, 1973.

[19] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. "A versatile computer-controlled assembly system". In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 298–307, 1973.

[20] L. S. Davis. "Shape matching using relaxation techniques". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(1):60–72, January 1979.

[21] L. G. Shapiro and R. M. Haralick. "Structural description and inexact matching". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3:504–519, September 1981.

[22] B. Radig. "Image sequence analysis using relational structures". *Pattern Recognition*, 17(1):161–167, 1984.

[23] A. Rosenfeld, R. Hummel, and S. Zucker. "Scene labeling by relaxation operations". *IEEE Transactions on Systems, Man and Cybernetics*, 6:420–433, June 1976.

[24] R. A. Hummel and S. W. Zucker. "On the foundations of relaxation labeling process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):267–286, May 1983.

[25] M. A. Fischler. "On the representation of natural scenes". In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 47–52. Academic Press, 1979.

[26] E. Bienenstock. "Neural-like graph-matching techniques for image-processing". In D. Z. Anderson, editor, *Neural Information Processing Systems*, pages 211–235. Addison-Wesley, 1988.

[27] B. Bhanu and O. D. Faugeras. "Shape matching of two-dimensional objects". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):137–155, March 1984.

[28] M. Lipschutz. *Differential Geometry.* McGraw-Hill, 1969.

[29] B. Bhanu and C-C. Ho. "CAD-based 3D object representation for robot vision". *Computer*, 20(8):19–35, August 1987.

[30] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1988.

[31] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, MA, 1987.