RECOGNITION OF PARAMETRISED MODELS FROM 3D DATA

Ian Reid

Engineering Science Department, Oxford University

Parks Road, Oxford, OX1 3PJ

ian@robots.oxford.ac.uk

ABSTRACT

This paper describes work done as part of the Oxford AGV (Autonomous Guided Vehicle) project [2] towards recognition of classes of objects to be encountered in a factory environment. We address the problem of recognising an object from range-data observations as an instance of a parametric model class, and determining the values of the class parameters for that instance of the model, and the pose of the object. We represent an object *class* as a map from an underlying shape, a set of parameters, and some constraints on these parameters, to an instance of the class. A search of the interpretation tree is combined with a constraint network to determine the legal interpretations and parameter values using observations on a instance of the class. We demonstrate the feasibility of this approach using polyhedral models and simple range-image features (position, surface normal observations).

INTRODUCTION

This paper addresses the problem of recognising an object from range-data observations as an instance of a parametric model class, and determining the values of the class parameters for that instance of the model, and the pose (by this we mean the 6 positional degrees of freedom) of the object.

The literature covering the problem of object recognition from either 3D (range) or 2D (intensity) data is extensive. Landmark work in this field includes [1, 8, 5, 9]. Generally such systems represent an object as a set of geometric features which are matched to features in an image. The matching rigidly enforces geometric constraints between features in order to reduce the size of the search space, which is inherently large. Unfortunately this results in separate models for objects which humans would typically classify as being of the same type. Similarly, objects with parts which move relative to one another are not easily represented in these systems. These difficulties mean that there has been much less research into recognition of parametrised models. Major work in this field includes [3, 7].

Our work bears resemblance to Grimson's extension (to simple parametrisations of models [7]) of his earlier work with Lozano-Pérez [8] which is based on searching and pruning an interpretation tree using geometric constraints. However our system – which combines an interpretation tree search with a constraint network based on the ideas of Fisher and Orr [6] – provides a more general framework for parameter determination.

Like Grimson and many others, we use a data-driven search of an interpretation tree as the basis for the match between the observed data and the model. Reference-frame-independent, scalar measurements can be derived from pairs of observed features; henceforth these will be called *image-measurements*. Matching proceeds by checking each new assignment of an observed feature to a model feature for consistency with the model and with the previous assignments in the current interpretation. Two assignments $p_{i_1} \rightarrow f_{j_1}$ (observed feature i_1 to model feature j_1) and $p_{i_2} \rightarrow f_{j_2}$ are *consistent* if the image-measurements derived from p_{i_1} and p_{i_2} lie within the range of possible values derived from the model features f_{j_1} and f_{j_2}.

In [8] the bounds for each image-measurement are precomputed and stored in look-up tables for efficient access. For parametrised models, offline computation of the bounds is not possible because the bounds depend on the values of the parameters. However we can choose to use image-measurements which are related to the parameters in well defined ways. We use a constraint network to represent the relationships between bounds on image-measurements and model parameters, and look-up tables which point to variables in the network (replacing the bounds table).

The following two sections show how such a network is defined and used to control the search of the interpretation tree and to determine object parameters.

REPRESENTATION

Our approach models an object using:

- a boundary representation (the faces of the object);

- a set of parameters (not necessarily independent);

- a set of constraints on the parameters.

As a simple example, consider a class of pyramids. This might be represented by the five faces, the parameter r (the ratio of its height to its base), and the constraint that the height be less than two times the base but greater than half the base; ie $0.5 \leq r \leq 2.0$.

The constraints and the boundary model define a class of object over the parameter set. Our aim is to recover the actual values (or ranges of possible values) of the parameters for an instance of a class in a scene, assuming no *a priori* knowledge about its pose (nor indeed, the extent to which it is occluded).

A sup/inf constraint network of the type described in [6] is compiled from the model's constraint set, in which each node is one of the following:

- a constant;

- a variable;

- an operator which performs a given operation on its input(s) (currently $+$, $-$, $*$, $/$, sin, cos, tan and their inverses and min and max are supported in our implementation).

Each variable V in the network has an associated interval (denoted by $B[V]$) giving the current upper and lower bounds of the variable. Each model parameter is a variable in the network. In addition, each bound for the image-measurements is a variable in the network. Although this could result in very large numbers of variables in the network, in practice many image-measurements are bounded by the same variables (which are sometimes the parameters themselves). References to the network variables corresponding to the image-measurement bounds are precomputed and stored in look-up tables.

The network is constructed so that the constraints and relationships between the parameters, and the relationships between the image-measurement bounds and parameters are satisfied by the network's topology. The topology is static for a given model (i.e. object class) and therefore may be precompiled.

Figure 1 shows an example of how the measurement of the angle between one of the side faces and the bottom face can be connected to the parameter r of the pyramid so that observed data can propagate through the network.

ALGORITHM

The interpretation tree is a dynamic structure, growing as consistent assignments are added and shrinking if they are found to be inconsistent. The variables in the (static topology) network hold the current estimates of the parameter values and image-measurement bounds for the current state of the interpretation tree. The state of the network is therefore used to guide the tree search. Consistency of image-measurement M with the assignments $p_{i_1} \rightarrow f_{j_1}$ and $p_{i_2} \rightarrow f_{j_2}$ is guaranteed by
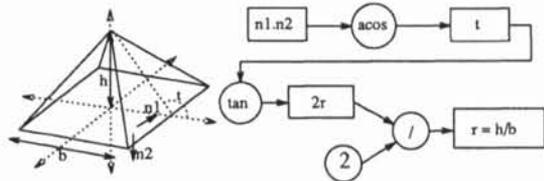

Figure 1: Pyramid example: (a) a pyramid with observations n_1 and n_2 on a side and on the base respectively; (b) network connections for propagation of image-measurements to model parameters. The squares represent variable nodes, the circles represent operation and constant nodes.

enforcing

$$\inf B[V_l] \leq M \leq \sup B[V_u]$$

where V_l and V_u are network variables (not necessarily the same) corresponding to upper and lower bounds respectively on image-measurement M and are determined using the look-up tables (see examples 1 and 2). In practice an interval around M, $I(M)$ is considered because M can only be determined up to certain error limits; the test above then becomes an intersection test.

If an assignment is consistent then the image-measurements derived from it are used to update the network:

$$\sup B[V_l] = \min\{\sup B[V_l], \sup I(M)\}$$
$$\inf B[V_u] = \max\{\inf B[V_u], \inf I(M)\}$$

Thus progressive refinements of the legal bounds of the image-measurements are performed; the measurements are used as updates on the current state of the network and allowed to propagate, refining other bounds and possibly highlighting an inconsistency in the current interpretation.

In order to maintain integrity in the network while backtracking during the tree search (caused by an inconsistency), each variable has a local stack associated with it. When a potential match is deemed consistent, each variable saves its current value on its stack. In the event that the interpretation turns out to be inconsistent, the old network state can be popped from the stack.

Our implementation currently considers position, surface normal features, denoted by $p_i = (v_i, n_i)$ for the ith observed feauture. The image-measurements derived from these which we use are:

$$M1 \equiv \arccos n_{i_1}.n_{i_2};$$

$$M2 \equiv (v_{i_1} - v_{i_2}).n_{i_2};$$

$$M3 \equiv (v_{i_2} - v_{i_1}).n_{i_1}.$$

Two examples based on the object shown in figure 2 illustrate the use of the measurements.

Example 1: Measurement $M1$ is consistent with the match p_1 to face 1 and p_2 to face 9 if

$$B[V] \cap I(\arccos n_1.n_2) \neq \emptyset$$

348

Figure 2: This object can be parametrised by the variables shown. Surfaces are marked with numbers.

where V is a variable connected in the network so that $t = \pi - V$. The look-up table entries for for the upper and lower bounds on the angle between faces 1 and 9 are both pointers to the variable V.

Example 2: Measurement $M2$ is consistent with the match p_1 to face 2 and p_2 to face 3 if

$$\sup I((v_2 - v_1).n_1) \geq 0$$
$$\text{and} \quad \inf I((v_2 - v_1).n_1) \leq \sup B[d_2]$$

where the look-up table entry for the upper bound on the distance of a point on face 3 to the plane defined by face 2 is a pointer to the variable d_2. The first condition is a form of sidedness; it says that all points on face 3 lie on the positive side of the plane defined by face 2. The second condition arises from the fact that the length of face 3 is d_2.

Once a complete legal interpretation has been found, the object pose (i.e. transformation from sensor to model coordinate frame) can be calculated. The state of the network determines the object parameter values. These are substituted into the generic model to give a model of the class instance, then the method described by [8] is used to estimate the pose. Occasionally the bounds on some parameters may not be sufficiently tight to estimate part of the transformation accurately. The addition of more observed data will, in general, result in a refinement of the bounds.

RESULTS

Results for the object shown in figure 3 are given in this section. Surface normals are estimated using a least-squares fit from the range-image at the points indicated. Two interpretations are found by the algorithm, corresponding to the symmetry of the object.

Table 1 shows the pose and parameter ranges determined. The bounds on the parameters are indicated by the outer two small vertical lines; the interval between the lines is the set of parameter values consistent with the observed data. The small vertical dash in between the bounds is the actual value of the parameter for the instance in the range-image; in each case the true value lies between the bounds computed by the algorithm. Some of the intervals have been narrowed to the point where the parameter is almost exactly determined (e.g. h_2 and w_3).

Figure 3: A sample object: (a) a parametrisation; (b) a computer generated range-image (lighter corresponds to closer).

The rotation is given as a rotation direction vector, and an angle in radians. The true values are shown in parentheses. The translation is not given; some of the parameter bounds have not been narrowed sufficiently to give an accurate estimate. This is to be expected since a minimum of observed data has been used for the experiment.

	Interpretation 1	Interpretation 2
w1		
w2		
w3		
h1		
h2		
h3		
d1		
d2		
d3		
d4		
d5		
d6		
Rotation vector	0.70398 (0.70406) 0.56317 (0.56339) 0.43272 (0.43230)	0.52995 (0.53414) -0.81925 (-0.82571) -0.21905 (-0.22125)
Angle	2.95430 (2.75654)	0.93452 (0.93632)

Table 1: Results for object in figure 3

On a Sun4 workstation the entire algorithm, including verification that the matched faces are visible in the final pose estimate, required seven seconds.

DISCUSSION AND CONCLUSIONS

A scheme for the recognition of object *classes* has been presented which makes use of a network of constraints to guide matching and determine model parameters. The feasibility of the scheme has been demonstrated using position, surface normal data.

We have not yet attempted complete automation of the model generation. Specifically, the precomputation of look-up tables is currently performed manually. This is time consuming and error-prone. Most of the necessary techniques for automatic symbolic manipulation have already been developed [3, 4], so this should pro-

vide a tractable but interesting research problem.

Our approach is general enough to allow for data other than position, surface normal pairs. In particular we make use of good quality, dense range-maps. Therefore, we are currently investigating a scheme which uses both planar and quadratic surface data, and edge data. This should result in tighter constraints and fewer ambiguities.

Our algorithm demonstrates good performance on an average speed machine (Sun4). However the current program is grossly inefficient in its network evaluation. Tests have shown that the network computation would be reduced by 80% or more using a data-flow implementation. We expect to port the network software to transputers where a parallel data-flow network will be the natural implementation.

Currently we assume all observed data to be from the object; i.e. *a priori* segmentation. This problem is overcome by [8] using a *null face*; an assignment which is always consistent. This could be incorporated into our work without difficulty, but has a number of disadvantages. A null face assignment is interpreted as an observation which does not correspond to any part of the model under consideration. Its use precludes negative constraints of the type "this observation was not physically possible under the current assumptions about the object pose and model". We are investigating ways to overcome the need for *a priori* segmentation without the disadvantages assosiated with the null face technique as implemented by [8].

In order to retain a very simple tree search algorithm, we require the boundary model representing the underlying shape of the object to have a fixed number of surfaces. Hence, a disadvantage of our approach is its inability to classify objects with differing numbers of surfaces as instances of the same class (not to be confused with the case where only some of the surfaces can be seen). An example of this limitation can be seen by considering a model of an industrial pallet. Ideally this should constrain the pallet to have between 5 and 7 horizontal slats of equal width with equal spacing and have 3 or 4 supporting beams at right angles to the slats, and so on (see figure 4). Our approach does not extend to parametrising the number of slats or supports. We are investigating ways to overcome this problem.
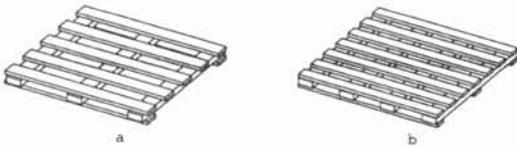

Figure 4: Pallets with differing numbers of surfaces.

REFERENCES

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

[2] Michael Brady and et. al. Progress towards a system that can acquire pallets and clean warehouses. In *Proceedings of the 4th International Symposium on Robotics Research*, pages 359–374, 1987.

[3] R. Brooks. Symbolic reasoning among 3-d models and 2-d images. In J.M. Brady, editor, *Computer Vision*. North Holland, 1981.

[4] C. I. Connolly, D. Kapur, J. L. Mundy, and R. Weiss. Geometer: A system for modeling and algebraic manipulation. In *Proceedings of the DARPA IU Workshop*, pages 797–804, 1989.

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

[6] R. B. Fisher and M. J. L. Orr. Solving geometric constraints in a parallel network. In *Proceedings of the 3rd Alvey Vision Conference*, pages 87–95, 1987.

[7] W. E. L. Grimson. Recognition of object families using parametrized models. In *Proceedings of the 1st International Conference on Computer Vision*, pages 93–101, 1987.

[8] W. E. L. Grimson and Tomas Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[9] David G. Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision*, 1(1):57–72, 1987.