Recognizing Parameterized Models Using 3D Edges [1]

Ian Reid and J. Michael Brady

Department of Engineering Science
Oxford University, OX1 3PJ, UK

Abstract

We present extensions to ideas we have described previously [1] for recognizing instances of object classes. An object class is modelled by a boundary representation, a set of model parameters, a set of constraints on the parameters, and a set of feature constraint tables, which represent pairwise geometrical constraints between features. A method for automatic generation of these tables from a boundary representation is described. An object's pose and class parameters are recovered by the algorithm using 3D edge fragments extracted from a stereo pair of images.

1 Introduction

Our work is based on two important concepts to come from the vast amount of literature relating to the problem of object recognition from 3D (range) data; constrained search of an interpretation tree and viewpoint independent, binary geometric measurements between features [2].

When dealing with geometrically fixed objects, bounds on the geometric measurements can be precomputed for each pair of model features and stored in *feature constraint tables* or FCTs. However this useful data structure becomes much more difficult to compute when we consider parameterized models because the bounds are no longer simple numbers; they are functions of the parameters which are not necessarily quantifiable until run-time. In section 2 we describe how *symbolic* FCTs can be generated automatically for parameterized models. Section 3 gives details of the edge constraints used and recent recognition results using stereo edge fragments.

2 Building FCTs automatically

An FCT entry is a pair of bounds on a pairwise geometric measurement involving upper and lower bound functions f_u and f_l, respectively, of model features M_i and M_j and a parameter vector p:

$$\mathrm{FCT}_{ij} = [f_l(M_i, M_j, p), f_u(M_i, M_j, p)]$$

For geometrically fixed objects, the value of p is known and so the FCT consists of constant entries. We showed in [1] that the idea of a FCT can be extended for p

[1] We acknowledge the use of a version of the TINA system and thank AIVRU, at University of Sheffield for this. Thanks also to Rupert Curwen for help with the experiments.

BMVC 1991 doi:10.5244/C.5.49

unknown by allowing an entry to be a reference to an atomic expression instead of a constant.

Since an FCT entry must be an atomic expression, the possibly large expressions for f_l and f_u must be simplified. In some cases this requires just the application of algebraic simplification rules, but in others we need to impose model constraints on the process (e.g. the expression $\min(x, y)$ can only be simplified if we have a constraint on the ordering of x and y). In the event that simplification cannot produce an atomic expression, we add a new constraint, $\langle\text{newvar}\rangle = \langle\text{expr}\rangle$, to the model constraints. This guarantees that the function expressions can always be simplified to atomic form.

In [1] simplification was performed by hand. This was both time consuming and error prone. We noted then though, that Mathematica [3] is useful tool for our problem; it is a system designed for symbolic computation and has in-built rules for simplifying symbolic expressions and mechanisms for the addition of mathematical relationships in the form of transformation and substitution rules. In the following we illustrate its application to our problem, however space restrictions permit only a simple example to give a flavour of the method.

Suppose that from the boundary representation (B-rep) of a model we generate the bounds for a particular geometric measurement to be the expressions

$$a - b \quad \text{and} \quad c - b, \tag{1}$$

(not necessarily in that order), and in addition we have model constraints $a, b, c \geq 0$ and $a = b + c$. These two constraints together imply a number of straightforward but powerful relations. Firstly, $a = b + c \Rightarrow a - b = c$; we enforce this by converting the constraint into a list of Mathematica substitutions: [2]

$$\{a - b \to c, -a + b \to -c, a - c \to b, -a + c \to -b, b + c \to a, -b - c \to -a\}$$

Secondly, $a = b + c$ and $a, b, c \geq 0 \Rightarrow a \geq b$ and $a \geq c$. Positivity of a parameter is asserted by a function Pve which we define appropriately (e.g. one of the rules in its definition is $\text{Pve}[-x] :=$ not $\text{Pve}[x]$). We also define rules for finding the maximum of two expressions:

$$\begin{aligned}
\text{Mx}[x, x] &:= x \\
\text{Mx}[0, x] &:= x \text{ if } \text{Pve}[x] \text{ else } 0 \\
\text{Mx}[-x, -y] &:= -\text{Mn}[x, y] \\
\text{Mx}[-x, y] &:= y \text{ if } \text{Pve}[x] \text{ and } \text{Pve}[y] \\
\text{Mx}[a + x, a + y] &:= a + \text{Mx}[x, y] \\
\text{Mx}[ax, ay] &:= a\text{Mx}[x, y] \text{ if } \text{Pve}[a] \text{ else } a\text{Mn}[x, y] \\
\text{Mx}[x, a + y] &:= x \text{ if } Mx[x, y] == x \text{ and not } \text{Pve}[a] \\
\text{Mx}[x, a + y] &:= y + a \text{ if } Mx[x, y] == y \text{ and } \text{Pve}[a]
\end{aligned}$$

Mn, for minima, is defined similarly. Then the substitutions $\text{Mx}[a, b] \to a$ and $\text{Mx}[a, c] \to a$ enforce the ordering constraints above and simplified, ordered bounds can be found for the expressions (1), above: $[-b + c, c]$. The addition of a constraint $c = b + d$ is sufficient to enable the reduction of the bounds to atomic form. Other constraints are similarly enforced by substitution rules.

[2] For clarity, Mathematica expressions are given in a pseudo-code.

3 Edge-based recognition

An edge segment is represented as $d = (e, p_m, l)$, where p_m is the midpoint of the segment, l is the length and e is unit vector in the direction of the edge. In addition we define $p_1 = p_m + (l/2)e$ and $p_2 = p_m - (l/2)e$, the endpoints of the segment. The measurements between edges d_i and d_j used are:

edge-angle $e_i.e_j$

edge-dist if (e_i and e_j parallel) $\mathrm{sgn}(e_i.e_j)(v.v - (v.e_j)^2)^{\frac{1}{2}}$ else $v.\frac{e_i \wedge e_j}{|e_i \wedge e_j|}$

edge-proj-1 $p_{i_1}.[e_j \wedge \frac{e_i \wedge e_j}{|e_i \wedge e_j|}]$ and $p_{i_2}.[e_j \wedge \frac{e_i \wedge e_j}{|e_i \wedge e_j|}]$

edge-proj-2 $p_{j_1}.[e_i \wedge \frac{e_j \wedge e_i}{|e_j \wedge e_i|}]$ and $p_{j_2}.[e_i \wedge \frac{e_j \wedge e_i}{|e_j \wedge e_i|}]$

where $v = p_{i_m} - p_{j_m}$ is a vector between the edges. Image measurement edge-dist is the perpendicular distance between the lines on which the fragments lie and edge-proj gives upper and lower bounds on the distance of edge d_i to the plane containing d_j and normal to $e_i \wedge e_j$.

A sample object class parameterized by $\{w_1, w_2, w_3, d_1, d_2, d_3, d_4, d_5, d_6, h_1, h_2, h_3\}$ is depicted in figure 1 (this parameter set is not independent; e.g. it entails the constraint $w_1 = w_2 + w_3$). The constraints on model parameters

$$\begin{array}{ccc} w_1 \geq 0.5d_1 & , & w_1 \leq 1.5 * d_1 \\ h_1 \geq 0.4w_1 & , & h_1 \leq 1.0 * w_1 \end{array}$$

were added to the model definition to restrict the relative dimensions of an object somewhat.

A calibrated stereo pair of three instances of this class was captured and stereo edge fragments extracted using PMF stereo [4]. Edges less than a certain length (assumed unreliable) and those in the background were removed manually. The remaining edge fragments were clustered into three sets of mutually orthogonal edges and an interpretation tree search conducted to determine legal interpretations and parameter values. Finally the object pose was computed by substituting the parameter estimates into the B-rep and computing a best fit for the transformation from model to sensor coordinates.

Figure 3 shows wire frames for the three (computed by substituting the parameter values into the B-rep) superimposed on the left camera image in the computed poses. Figure 4 gives the computed ranges of the parameters for each of the three interpretations. We can use these ranges to distinguish between the different instances; if the parameter ranges for the same parameter in two different interpretations are different then we can say that the two represent different instances of the same class.

References

[1] I. D. Reid. Recognition of parametrised models from 3d data. In *Proceedings of the IAPR Workshop on Machine Vision Applications*, pages 347–350, 1990.

[2] 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.

[3] Stephen Wolfram. *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley, 1988.

[4] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, (14):449–470, 1985.

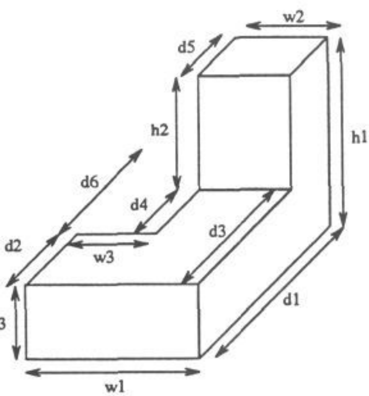

Figure 1: A sample object class and possible parameterization.

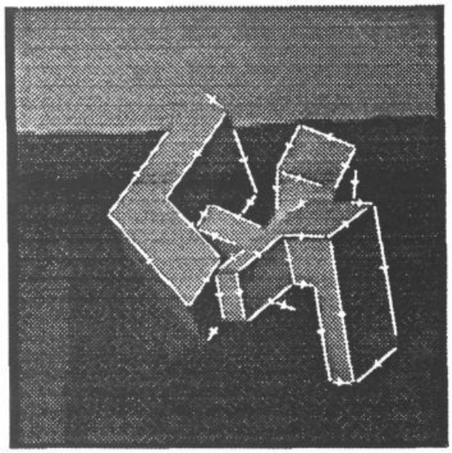

Figure 2: Three instances of the object class with stereo data superimposed on the left camera image.

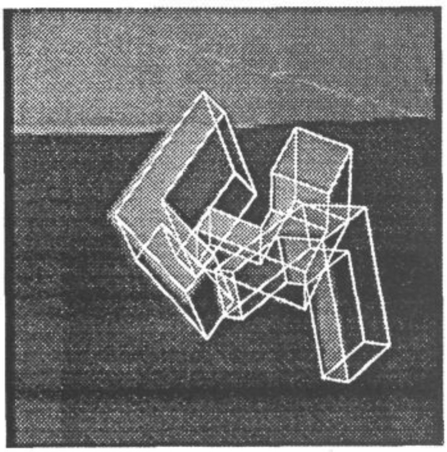

Figure 3: Wire frames generated by substituting the computed parameter values into the B-rep, superimposed over the left camera image in the computed pose (no hidden line removal).

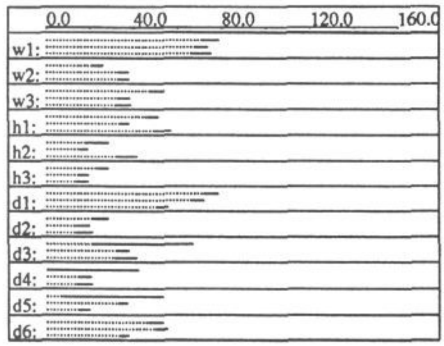

Figure 4: The solid bars indicate the computed ranges for the parameters. For each parameter the ranges from top to bottom correspond to the interpretations in figure 3 from left to right.