Computer-Aided Design 36 (2004) 501–510

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Applying knowledge to reverse engineering problems

Robert B. Fisher

School of Informatics, University of Edinburgh, Edinburgh, UK

Received 12 February 2003; received in revised form 15 May 2003; accepted 19 May 2003

Abstract

This paper summarizes a series of recent research results made at Edinburgh University based on projects that apply domain knowledge of standard shapes and relation ships to solve or improve reverse engineering problems. The problems considered are how to enforce known relationships when data fitting, how to extract features even in very noisy data, how to get better shape parameter estimates and how to infer data about unseen features.

Keywords: Range data; Reverse engineering; Constraints

1. Introduction

Traditional processes for reverse engineering objects and structures from 3D datasets have been initially data (e.g. triangulated models) and parametric surface (e.g. quadric surface) driven. These approaches has been successful for simple parts, but have resulted in reconstructions that have ‘frozen-in’ errors. Typical errors are surfaces at incorrect relative positions or artifacts arising from noisy or missing data.

For several years our research group at Edinburgh University has been exploring ‘knowledge-based’ techniques to help overcome these and other problems. The underlying theme behind this set of techniques is the exploitation of general knowledge about the domain of objects being reconstructed. The reconstruction process is not ‘model-based’ reverse engineering, as then there would be no point to building the models—this would not be ‘reverse engineering’. On the other hand, the knowledge is not arbitrary, because the objects that humans construct are not arbitrary: the shapes of most normal objects follow standard conventions arising from tradition, utility or engineering design. This is a ‘knowledge-based’ approach.

We argue that exploiting this extra knowledge allows improved reverse engineering. This paper presents several different examples of the general approach, summarizing results from the full publications, which are cited within and can be found at: http://www.dai.ed.ac.uk/homes/rbf/publications.html.

One of the assumptions underlying the work summarized here is that the reverse engineering/reconstruction process need not be fully automated. Computers are good at data analysis and fitting; humans are good at recognizing and classifying patterns. Thus we are working in a cooperative problem solving paradigm, where a human might hypothesize that a given relationship holds (e.g. two surfaces are potentially parallel) and the computer can either help verify the relationship (e.g. calculate the probability that they are parallel) or compute some parameter that results from the relationship (e.g. the separation between the surfaces).

From these general ideas, we have been exploring techniques to improve reverse engineering of objects from three-dimensional (3D) point data sets. These main themes are explored in the sections that follow:

1. There are many constraints on feature relationships in manufactured objects and buildings. Exploiting these constraints improves the recovery of object models.
2. General shape knowledge can allow recovery even when data is very noisy, sparse or incomplete.
3. Complete data acquisition can be impossible in practice, but inference of much occluded data is possible.
4. Euclidean fitting is now fast enough to be practical and gives better results.

E-mail address: rbf@inf.ed.ac.uk (R.B. Fisher).

0010-4485/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0010-4485(03)00158-1

5. Many of these recovery problems require discovery of shape and position parameters that satisfy the knowledge-derived constraints. Evolutionary search methods can be used to do this search effectively.

2. Constrained reverse engineering

Parts have standard feature relationships

One of the cornerstones of the recent research in our laboratory has been constrained recovery of 3D shapes from 3D point cloud data. There has been much previous research on curved surface shape estimation, based either on the Euclidean distance [6] or variants of the algebraic distance [22]. Given the shape bias arising from the algebraic distance, researchers have also developed a general quadric surface extension to the algebraic distance using a gradient based weighting [52] or a shape specific approximation [31]. These fitting approaches were for single surfaces. In our case, we have used a constrained algebraic distance approach that applies shape specific constraints on all of the individual surfaces. Within the same framework, we also simultaneously apply constraints that encode standard feature relationships such as alignment of surfaces, colinearity of features, etc. This constrained reverse engineering technique has been applied to both industrial parts and architectural scenes.

The key issue is how to incorporate shape and design constraints into shape fitting of 3D data. Our current approach is to formulate shape fitting as constrained least-squares problem. If:

- $\vec{p}$ specifies the parameter vector for feature shapes and positions
- $\mathscr{H}$ is the least-squares shape error matrix
- $C_i(\vec{p})$ are constraints over the parameters
- λ_i are penalty costs

and then minimize:

$$\vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} + \sum_i \lambda_i C_i(\vec{p})$$

The first term is a least-squares fitting term that ensures that model surfaces lie close to the image data. The second term encodes the penalties for constraint violations. The linear least-squares error term can also be a non-linear Euclidean distance (or other) error term. Minimizing this error is generally a non-convex problem, so we initialize $\vec{p}$ to be the standard least-square solution and $\lambda_i = 0$ and then apply numerical optimization methods. We then incrementally enforce the constraints by increasing penalty costs λ_i and re-minimizing until the constraints are satisfied to the desired tolerances. The gradual increase ensures that the solution stays near the least-square solution and also helps avoid local minima. Experiments show that solutions initialized from different randomly perturbed starting points converge to a small cluster of nearby solutions.

While we have only experimented with constraint functions $C(\,)$ that use the square of the error in the constraint, one could also use a gated function that produces zero error if the constrained relationship is within a given tolerance. If this form were used, our gradient based optimization method would need to be modified as there is a discontinuity at the tolerance point. One possible approach is to use evolutionary methods mentioned in Section 6. Then the constraint can be simply ignored in the evaluation function if the specified tolerance is satisfied.

We have applied this approach to engineering parts modeled by planar and quadric surfaces [55,56]. The surfaces are extracted from range images or point clouds by a segmentation process based on (1) shape discontinuity detection, (2) boundary constrained noise smoothing, (3) principal curvature based local shape classification and finally (4) quadric surface fitting. A recent comparison in Ref. [26] concluded that in many ways our single image planar surface segmentation algorithm had the best performance among current algorithms.

The part shown in Fig. 1 has constraints between planar, cylindrical and conical surfaces. Seven shape relation constraints were applied. All constraints can be satisfied while still maintaining close surface fitting. Applying the constraints also improves shape parameter recovery. For example, the top cylindrical surface has the true radius of 60 mm. Initial least-square quadric fitting estimated an elliptic cylinder radius of 33–46 mm. Adding the relationship constraints resulted in a circular cylinder radius estimate of 59.54 mm.

One can also apply the approach [57] to enforcing inter-surface boundary constraints between freeform and quadric surfaces, while also trying to minimize surface fitting error.

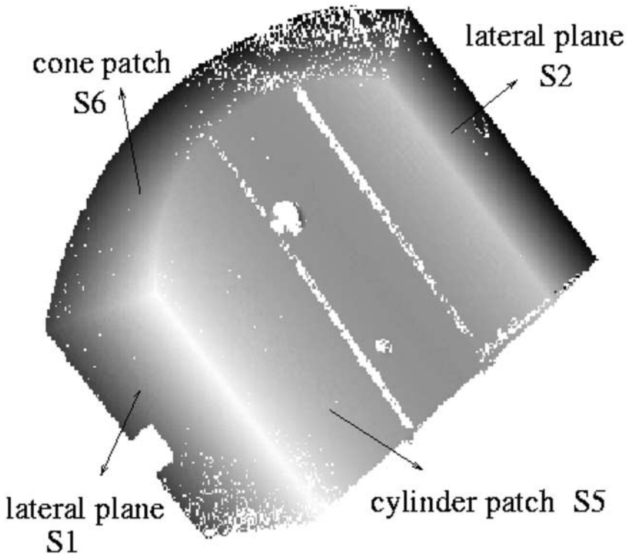

Fig. 1. Constrained quadric surface recovery.