Inferring Volumetric Descriptions from Slice Data

Emanuele Trucco
Department of Artificial Intelligence
University of Edinburgh
Forrest Hill, Edinburgh, Scotland

Abstract

This paper describes a simple but powerful technique for inferring volumetric descriptions of articulated objects from parallel planar cross-sections or *slices*. A two-step algorithm is described: first the data are segmented into volumes or *parts*, then volumetric primitives are fitted to the parts. The segmentation is completely model-independent. It exploits the regularities derived from differential geometry to generate a set of perceptually plausible object subparts which, in principle, could be described by *any* volumetric primitives. *Sticks* and *blobs* are used to express the reconstructed model, which is meant to be suggestive more than literal. The model specifies type, connectivity, position and orientation of the subparts. Attractive features of the approach include its simplicity, its stability with respect to variations of imaging conditions, and the model-independent nature of volumetric segmentation.

Introduction

This paper describes a technique for inferring volumetric descriptions of articulated objects from a variable number of cross-sections or *slices*. Automatic acquisition of model-based object descriptions from planar slices is relevant in several domains, including analysis of medical tomographic data [1], inspection of industrial parts [2] and object recognition [3,4]. We assume the slices are parallel and registered.

It can be in general difficult to adopt surface segmentation techniquesi which compute explicitly curvatures, because wide gaps between slices can lead to considerable errors in the curvature estimates. We keep therefore our conjectures about the nature of the surfaces to a minimum, just expecting them to satisfy the convexity and transversality assumptions, introduced in the next section. Using this regularities, we build hypotheses of plausible cross-sections of parts going through each slice. We then coalesce such cross-sections to form consistent 3-D volumes.

The models used to describe the segmented parts are a subset of the volumetric primitives of Fisher's Suggestive Modelling System or SMS [5], the *stick* and the *blob*. The stick represents straight or bent elongated volumes, e.g. a pen or a finger. The blob represents compact structures with no privileged elongation, e.g. a human head.

The rest of this paper is organized as follows. First the assumptions made in our work are presented and motivated. Then our algorithm and its implementation for recovering SMS descriptions from slices are introduced. Some experimental results are reported and discussed briefly. The last section mentions some related work.

Assumptions

The following assumptions are made in order to detect boundaries between object subparts:

(a) Transversality This states basically that interpenetrating parts, arbitrarily shaped, form contours of concave discontinuities of their tangent planes [6]. Transversality suggests to segment parts along loci of negative minima of each principal curvature [7]. The problem could be tackled satisfactorily by curvature-based segmentation techniques if the data were dense [8]. We'll sketch in the next section how the subparts recovered by our technique are separated by contours which approximate the loci of minima in negative curvature found by a simple but effective technique.

(b) Convexity. Convexity is used in an intuitive more than mathematically precise sense. It means that the surfaces of the parts forming the objects must be formed by convex or parabolic points. From the mathematical point of view, this hypothesis could be better renamed *non-concavity*.

Incidentally, the inter-slice distance must be informative enough, so that discontinuities across slices are due to object discontinuities and not to insufficient sampling frequency. All experiments presented were run with sets of 128 slices.

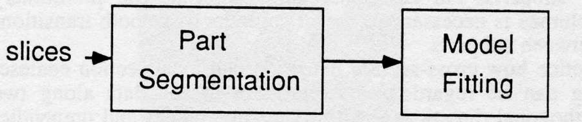

Fig.1 - The two-stage algorithm devised.

The Algorithm

The input to the algorithm is a sequence of parallel planar slices of an articulated object. Each slice is an image giving the intersection of the object with a plane (cross-section). The output is a file describing the structure of the object, that's type, parameters, position, orientation and connectivity of its subparts. The description file is written in SMS [5], a language which allows to express models in terms of line, surface and volumetric features. Only volumetric features are inferred here.

The algorithm is divided in two sequential stages: *model-independent segmentation* and *model fitting* (see figure 1).

Stage 1: Segmentation

The *segmentation* is meant to recover the object subparts exploiting the transversality regularity. This is carried out in three steps.

Cross-section detection: a linear piecewise approximation is computed from the cross-sectional curve on each slice and subsequently segmented into plausible cross-sections of parts going through the slice. The linear piecewise approximation is computed as follows. The curve is segmented near local minima of curvature. The segmentation points are selected by a scale-independent maximum-deviation algorithm. At each iteration, a curve arc is approximated by a straight line through its endpoints and split at the maximum deviation point, until the length of the approximating line is less than a threshold or the arc approximates satisfactorily a straight line. The ratio of deviation to length is used as a measure of the significance of the straight line approximation. Although less accurate than those involving explicit curvature estimates, deviation-based algorithms for curve segmentation are fast, simple and efficient, and give results in good accord with human judgement [9]. The linear piecewise approximation is then segmented at minima of concave portions. Such minima are found trivially. Notice how the whole procedure can be regarded as an efficient approximation of a curvature-based detection of negative extrema.

The result of this step is a collection of planar cross-section hypotheses for each slice. The contour of each cross-section hypothesized is represented by the linear piecewise approximation (a polygon) and is enclosed between two concavities of the cross-sectional curve. Notice that this assumes that every concavity is just a separator (non-object) between two convex objects.

Local section coalescing: this step checks local shape continuity across slices to form preliminary volumes. Cross-section hypotheses lying on adjacent slices are projected on the same plane and their degree of overlap is computed by a O(nlogn) polygon intersection algorithm [10]. Two sections are coalesced if their overlap exceeds a user-defined threshold. In our experiments, this threshold was set to 0.5 in most cases. A similar technique was devised by Boissonat for surface reconstruction from slice data [11]. Low overlap figures indicate shape discontinuities and therefore contours of subparts. An additional exploration of the preliminary volumes is necessary to detect contours of smooth transitions between subparts.

Notice how cross-section detection and local section coalescing can be regarded as exploration of the data along two orthogonal directions - within the slice planes and perpendicularly to them.

Overfragmentation analysis: Bent subparts can be erroneously segmented by the overlap analysis owing to their orientation (see figure 4). This step identifies and solves such situations (called *snake cuts*), glueing together again chunks of overfragmented parts. The axes of the two volumes involved are estimated locally near the joint by means of a simplified SAT technique. Two border sections (one per volume) are then computed by slicing the two subvolumes involved with slices perpendicular to the estimated axes. Shape continuity is then tested on the resulting sections and the two subparts merged if sufficient continuity (overlap) is found.

The overall result of the segmentation stage is a final orientation-independent set of volumes or *subparts* and a *subpart connectivity graph*, to be fed into the model fitting procedure. In practice, the algorithm detects contours of non-concave parts along two perpendicular directions, along and across the planar slices. Apart from sampling considerations, sources of imprecision include the fact that the critical points detected along one direction can become weak for particular object orientations. The combined action of the two directions, however, makes up for it. Another error is introduced by the fact that cross-sectional data are parallel slices. This implies that segmented parts are, in general, truncated by extremal slices. The truncation involves usually only a small terminal portion of the part, but results in underestimated subpart diameters. A correction is not difficult to introduce.

Stage 2: Model Fitting

The second stage, *model fitting*, generates the SMS file describing the subparts identified by the segmentation. Ten parameters are recovered for each part: type, position, orientation and structural parameters (length, thickness and bent radius for a stick, three main radii for a blob).

For fitting purposes, blobs are modelled by ellipsoids and sticks by (possibly) bent cylinders. A part is first classified as stick or blob by means of an heuristic comparing the maximal and minimal lengths along the main axes. Then the axis of the parts is computed. Blobs pose no serious problems for axis computation; since there is no privileged elongation direction, the blob's axes are assumed to be straight and parallel to the main axes. On the contrary, the axis of a stick can be bent. Notice that the current version of SMS allows only a circular axis approximation for bent sticks. If the stick was assembled by the overfragmentation analysis its axis is obviously bent and a suboptimal circle fitting algorithm [12] is used to fit a SMS bent stick to it. The algorithm selects a subset of axis points and fits a circle to them exactly. If the stick was not overfragmented then a linearity test is run on an initial estimate of the axis. A least squares fit is then performed on the sections's centroids to recover straight axes. Suboptimal circle-fitting is adopted again for bent axes.

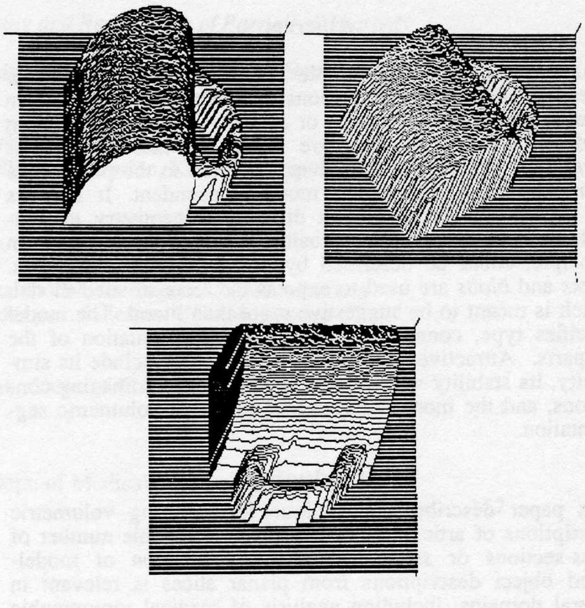

Fig.2 - Range images of a cup in different orientation

Experimental Results

We have implemented the above algorithm in C on a Sun3 workstation. We report in this section the results of two experiments. The input slices were obtained from 128x128 full range images, sliced by hand at different resolutions. The number of slices used in the experiments was 128.

A cup - figure 2 shows several images of a cup in different orientations with respect to the slices. An example of the SMS description files generated by the program is given in the appendix. Figure 3 shows a 3-D pictorial rendering of the reconstructed model. Parameter variations (position, orientation, shape) indicated good stability to rotation. The

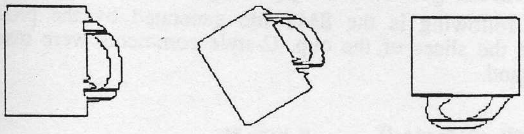

Fig.4 - Overfragmentation (*snake cuts*) in different orientations

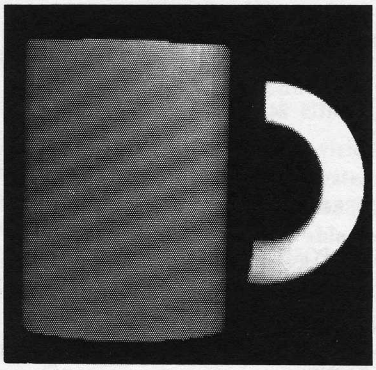

Fig.3 - Computer-generated reconstructed model

handle was always overfragmented (figure 4) and successfully recovered by the overfragmentation analysis. The circular shape of the handle in figure 3 is due to the current SMS convention for the axis of a bent stick (circular arc only). Notice however that a more accurate model could be fitted to the parts obtained after the segmentation step. The correct connectivity is preserved (see connectivity section in SMS description, appendix), although a certain amount of shrinking is introduced. The present prototype does not include any correction for this shrinking.

A toy mouse - figure 5 shows a plasticine mouse and figure 6 its sliced version. Figure 7 is a 3-D rendering of the recovered subpart model. Some shrinking has been introduced again, owing to a few border slices being regarded as noise for each subpart.

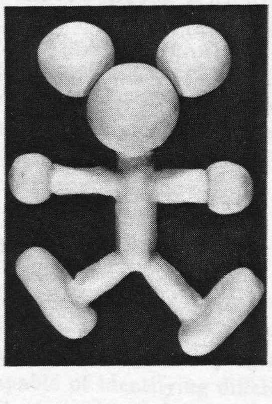

Fig.5 - Plasticine toy mouse

Discussion

We have presented a two-stage algorithm for recovering a SMS volumetric description from tomographic slices of objects and a first prototype implementation. Attractive features of the algorithm include model-independent segmentation of the data into subparts, based on a few general assumptions; rotational stability; the ability of correctly describing arbitrarily curved objects in terms of parts (for the segmentation phase); simplicity and computational inexpensiveness.

Each step of the segmentation algorithm is local in nature and therefore suitable for parallel implementation.

Some Related Work

An example of industrial application is given by White [2] who presents a two-step system for reconstructing accurate models of cylindrical pipes in different orientation from a sequence of slices. Grimson and Lozano-Perez [13] discuss a technique for identifying simple cylinders in range data by fitting ellipses to arbitrary cross-sections. Although robust, such techniques seem confined to simple cylinders. An interesting example of medical application is Soroka's system [1], which infers linear straight generalized cylinders from tomographic slices. The technique cannot cope with bent subparts and seems prone to overfragmentation. A certain interest has recently been directed to the problem of inferring volumetric models from full range images, in particular fitting models to segmented parts [14,8,15]. The methods proposed are usually mathematically elegant and well-founded but computationally expensive, involving optimization and variational techniques.

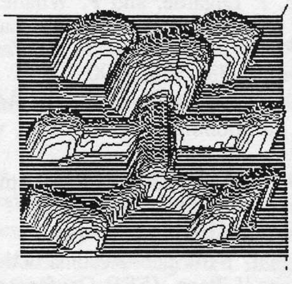

Fig.6 - Range image of the toy mouse

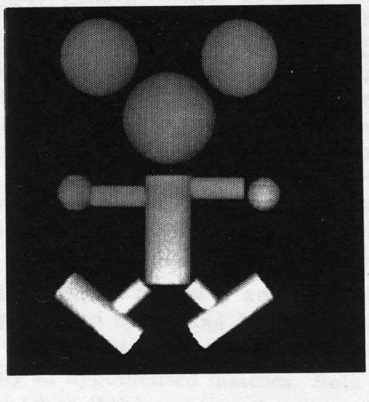

Fig.7 - Computer-generated reconstructed model

Acknowledgments

Thanks to Bob Fisher for his constant advice. KP Naidu supplied the range data for the plasticine mouse made (with great fun) by the author. This work has been supported by a "Stimulation Action" Grant of the Commission of the European Communities.

References

References

1. B. Soroka, "Generalized Cones from Serial Sections," *Computer Graphics and Image Processing*, vol. 15, pp. 154-166, 1981.
2. S. White, "Technology Innovations and Product Design Issues in Machine Vision," *Proc. British Machine Vision Conference*, pp. xi-xvi, 1990.
3. E. Trucco, "Inferring Convex Subparts from Slice Data," *Pattern Recognition Letters*, 1990.
4. E. Trucco, *Automatic Acquisition of Volumetric Models from Slice Data,* University of Edinburgh, 1991. Department of Artificial Intelligence Research Paper 519.
5. R. Fisher, "SMS: a Suggestive Modelling System for Object Recognition," *Image and Vision Computing*, vol. 5, pp. 98-104, 1987.
6. M. do Carmo, *Differential Geometry of Curves and Surfaces,* Prentice-Hall, 1976.
7. D. Hoffman and W. Richards, "Parts of Recognition," *Cognition*, vol. 28, pp. 65-96, 1985.
8. F. P. Ferrie, J. Lagarde, and P. Whaite, "Recovery of Volumetric Object Descriptions from Laser Rangefinder Images," *Proc. First European Conference on Computer Vision*, pp. 387-396, 1990.
9. M. Fischler and R. Bolles, "Perceptual Organization and Curve Partitioning," *IEEE-PAMI*, vol. 8, pp. 100-105, 1986.
10. S. Kundu, "A New O(nlogn) Algorithm for Computing the Intersection of Convex Polygons," *Pattern Recognition*, vol. 20, pp. 419-424.
11. J-D. Boissonat, "Surface Reconstruction from Planar Cross-Sections," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 393-397, 1985.
12. T. Pavlidis, *Algorithms for Graphics and Image Processing,* Computer Science Press, Rockville, Maryland, 1982.
13. W. Grimson and T. Lozano-Perez, "Finding Cylinders in Range Data," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 202-207, 1985.
14. A. Pentland, "Automatic Extraction of Deformable Part Models," *International Journal of Computer Vision*, vol. 4, pp. 107-126, 1990.
15. F. Solina and R. Bajcsy, "Recovery of Parametric Models from Range Images," *IEEE-PAMI*, vol. 12, pp. 131-147, 1990.

Appendix

The following is the SMS file generated by the program from the slices of the cup. C-style comments were inserted by hand.

```
(STICK stick0                 /* body */
    LENGTH          83.75
    CROSS_RADIUS 30.81
    BEND_RADIUS  0
    (NONE)
)

(STICK sn_stick321            /* handle */
    LENGTH          79.68
    CROSS_RADIUS 6.32
    BEND_RADIUS  24.19
    (NONE)
)

(PRIMARY_ASSEMBLY cup         /* positions & connect. */

    (DEFAULT_POSITION AT
        TRANSLATION (40, -80, 500)
        ROTATION RST (HALFPI, 0, 0)
    )

    (VARS (NONE))

    ((ASM_ALT (
            (PLACED_FEATURE stick0 AT
                TRANSLATION  (76, 24, 0)
                ROTATION RST (1.57, 0, 0)
                SCALE          1.0
            )

            (PLACED_FEATURE sn_stick321 AT
                TRANSLATION  (34, 40, 0)
                ROTATION RST (3.14, 0, 0)
                SCALE          1.0
            )
    )))

    ( (CONNECTED  stick0 sn_stick321) )
)

(VDFG cup (          /* visibility information */
                              /* front view */
    (VIS_GROUP (stick0 sn_stick321)
     TAN_GROUP (NONE)
     CONNECT_CONSTRAINTS (NONE)
     NEW_FEAT_CONSTRAINTS (NONE)
     POSITION_CONSTRAINTS (NONE)
     ))
)
STOP
```