MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

11—5 Finding line segments in range images

Concettina Guerra *
Università di Padova, Italy
and Purdue University, USA.

Valerio Pascucci †
Lawrence Livermore Nat. Lab., USA.

Abstract

The problem of fitting straight lines to a number of data points arises frequently in image processing and computer vision. In this paper, we consider two approaches to solve the line detection problem in range images. The first is a simple 3D approach that finds the best line among a randomly selected subset of lines each through a pair of edge points. The second approach, based on tabu search, optimizes the fit of the lines to the points. We present experimental results obtained with the two approaches on range images of polyhedral objects.

1 Introduction

The problem of fitting straight lines to a number of data points arises frequently in image processing and computer vision. Many approaches have been proposed and implemented for the case of 2D images. Successful approaches for intensity images do not necessarily represent reasonable solutions for range data. For instance methods based on the Hough transform and its variants are effective for 2D line detection for their insensitivity to noise and gaps in the lines [3]. However, since their memory and computation requirements grow with the number of parameters necessary to define a geometric primitive they do not seem adequate for the detection of lines in 3D space.

Not much work has been done on the detection of lines in range images; most of the research on the extraction of primitives from range data has focused on the segmentation of data into planar patches or into the more complex second-order degree surfaces [7]. Given an image segmented into planar regions, the line segments can be found by tracing and linking the boundary points between adjacent faces in the labeled image. However, the results of such an approach are generally not satisfactory and need to be refined due to the often imperfect results of the segmentation itself. One of the common errors of the range data segmenters is to produce over-segmentation, that is multiple detection of a single surface. Furthermore, often the detected boundaries between adjacent surfaces are distorted and noisy.

One approach to line detection in range images proposed in the literature [2] reduces the problem of line detection in 3D to a 2D problem by first segmenting the range data into planar patches and then finding the boundaries of each planar face. This is accomplished by mapping the edge points assigned to a face into a 2D binary image and applying a 2D Hough transform to find straight line segments that are the boundaries of the planar face. This method finds the same line more than once, since a line is at the boundary of more than planar face of an object; thus further processing is needed to eliminate the duplicates.

In this paper we consider a direct 3D line detection problem stated as follows. Given a set Γ of n points in 3D space and a non-negative constant ϵ, determine the line that is at a distance at most ϵ from the maximal number of points of Γ. The above problem is solved repeatedly for the extraction of multiple lines after the removal of the points that are found close to the best line. For line detection in range images, the input set Γ is chosen as the set of edge points.

In the following we first discuss a simple approach to solve the 3D line detection problem that determines the best line among all lines defined by pairs of input points. We then present a more robust optimization technique based on *tabu search* that explores a larger set of candidate lines. Both strategies have high computational requirements and may become impractical for applications involving large datasets of points. To reduce the complexity of the algorithms *random sample consensus* (RANSAC) [1] [4] is used so that the search can restricted to a random selected subset of pairs of points.

The rest of the paper is organized as follows. In section 2 we describe a simple approach to line detection on range images. We show that a proper selection of edge points that only takes into account crease edges may lead to better results and also speed up the computation. In section 3 we briefly review the tabu search paradigm for solving optimization problems and show how it can be applied to the problem of detecting lines in three-dimensional point sets. Section 4 describes a post-processing step that is needed to eliminate gaps present in the lines or to remove possible duplicate lines. Finally, we conclude with future work.

*Address: via Gradenigo 6a, 35131 Padova, Italy. E-mail:guerra@dei.unipd.it

†Address: P.O. Box 808, L-560, Livermore, CA 94551 USA. E-mail:pascucci@llnl.gov

2 A simple approach

A simple approach to line detection selects the optimal line among those that are defined by pairs of points in the dataset. It does so by determining for each line passing through a pair of edge points the number of other points lying within the expected measurement error ϵ from that line; the best line is the one that corresponds to the maximum computed value. The number of pairs $\frac{n\times(n-1)}{2}$ is generally very large for any reasonable value of n. If all pairs of points have to be considered the overall $O(n^3)$ time complexity would be prohibitive. However, in practice a number of pairs much less than the above is acceptable. A relatively small subset of pairs of points randomly chosen can yield results that are within a given bound from the optimal.

The determination of the number k of trials (pairs) that are guaranteed to produce good results with high probability has been considered for the extraction of various parametric shapes and problems [4] [11] The value k is chosen as follows. Let w be the probability that a single randomly selected data point is within ϵ distance from the line. Let z be the probability that at least one of the random selections is error-free. z is a function of w and k:

$$z = 1 - (1 - w^2)^k$$

Thus, the value k is given by:

$$k = \frac{\log(1-z)}{\log(1-w^2)}.$$

In our system, the edge points are found by the scanline approximation algorithm [8, 9]. There are several edge detection algorithms developed in the literature for range images. We have chosen the scanline algorithm because of its simplicity and its high execution speed. Furthermore, this edge detector finds the jump and crease edge points. This information may be useful in reducing the amount of post-processing needed to eliminate duplicate lines. Jump edges are generated by occlusion planes. A line detection algorithm tends to produce separate lines corresponding to jump edges on both sides of the planes; the lines are adjacent on the range grid, but are distant in 3D space and therefore detected as different lines. By using the information about the location of the jump edges, this problem is overcome.

3 Tabu search

Tabu search (TS) [5, 6] is a powerful optimization technique that has been used to solve a variety of complex combinatorial problems. One of the main components of TS is the use of adaptive memory: during the search, local choices are guided by the past history of the search. Restrictions are imposed by making reference to the memory structures, which store forbidden or tabu alternatives. This prevents solutions from the recent past from being revisited. Other important components of tabu search are *intensification* and *diversification*. Intensification encourages moves historically found good; diversification encourages moves to solutions that differ significantly from those seen in the past.

We have applied the basic TS paradigm to the line detection problem and obtained results of better quality than using the above simple approach. The quality of a line is defined as number of edge points within ϵ distance from the line. The method does not restrict the lines to pass through pairs of edge points and therefore may find a better fit of lines to points especially for long line segments. The TS involves exploring a dynamic neighborhood of a solution. Given a solution r_{ij}, that is a line segment with endpoints P_i P_j, a "move" operation moves either or both P_i and/or P_j in a small 3D neighborhood. More precisely, a move changes the value of one or more coordinates of either P_i or P_j or both by ϵ. Tabu moves correspond to changes in coordinates recently applied and are excluded. Starting from an initial random solution, the procedure repeatedly selects the best local move as the next solution to explore. Furthermore, it keeps track of the best known solution over time and updates the list of tabu moves.

The procedure uses *diversification techniques* that allow to consider moves not in the local neighborhood. When the objective function remains constant over a given time, a larger value of ϵ is introduced. Furthermore, if the objective function does not decrease for a number of consecutive iterations, a critical event occurs and the search is restarted with a new initial solution. The method continues to generate solutions and over time the best known solution continues to improve.

The TS for line detection has been successfully applied to reasonably small sets (hundreds of points) from biological datasets, in particular for the determination of linear segments associated to secondary structures in proteins. For applications in vision involving range data the number of boundary points is generally high (few thousands points) and the extra work required to optimize the model fitting is justified when the level of noise and distortion is high and high accuracy is required.

4 Merging segments

The procedures described above to find the segments that best fit the data may produce an over-segmentation, that is more segments than those actually present in the image. To overcome this problem, a post-processing phase is needed to eliminate spurious segments or to merge contiguous segments.

Next we describe how to merge two 'close' segments. The following two criteria define close segments:

- the lines containing the two segments have approximately the same slope.
- the distance between the two segments is below a given threshold.

The distance between two segments if defined as the minimum distance between the two lines containing the segments, if this minimum distance is achieved by points internal to the two segments; otherwise, it is defined as the minimum of the distances between the extreme points of the two segments.

Once two segments P_iP_j and P_kP_l are found close according to the two previous criteria, they are replaced by a single segment defined as follows. Let

$$P_m = \frac{w_{ij}(P_i+P_j)/2 + w_{kl}(P_k+P_l)/2}{w_{ij}+w_{kl}}$$

where w_{ij} (w_{kl}) is the number of input points within ϵ distance from $\overline{P_i, P_j}$ $(\overline{P_k, P_l})$. In other words, a segment is given a weight that is the number of edge points associated to the segment. This weighting factor penalizes short or sparse segments. Let e_{ij} (e_{kl}) be the unit vectors of r_{ij} (r_{kl}). The new segment r' that results from merging r_{ij} with r_{kl} lies on the line through P_m with the unit vector e_m:

$$e_m = \frac{w_{ij}e_{ij} + w_{kl}e_{kl}}{w_{ij}+w_{kl}}$$

To determine the endpoints of r' we project P_i, P_j P_k and P_l on the line of r' and choose the two projections that are the farthest apart.

5 Experimental results

The programs LD_RANGE and LD_RANGE_TABU that implement the proposed approaches are written in C. We have tested the programs on 40 range images of polyhedral objects acquired by an ABW structures light scanner. [1] Figure 2 shows the output of the program using the simple approach presented in section 2 on the image abw.test.9 of figure 1. Figure 3 shows the output of TS on the same image abw.test.9. From the figures, the improvements of the TS in terms of fewer fragmented segments are clear. The execution times on a SUN Sparcstation20 are 0.3s and 0.6s for the simple and TS approaches, respectively.

6 Conclusions

We have presented two approaches to the line detection problem in range images and shown results on range images of polyhedral objects. Although the TS requires higher execution times, the results obtained with this strategy are generally better than those obtained by a simpler strategy. We have used the line segments extracted from range images as the inputs to a matching algorithm that finds correspondences between line segments in two 3D objects. When TS was chosen to generate the line segments, generally a better match was found.

As a final note, the segments extracted by this method do not generally correspond to a valid boundary representation of an object [10]. One necessary condition satisfied by a valid boundary representation is that the boundary segments either are disjoint or intersect at a common vertex. This is often violated by the output of the algorithm. Obtaining a valid boundary will be subject of future work.

[1]The images are available from http://marathon.csee.usf.edu/range/segcomp/SegComp.html.

References

[1] C. Bolles and M. K. Fischler. A RANSAC-based approach to model fitting and its applications to finding cylinders in range data. In *Proc. 7th Int. Joint Conf. on Art. Intel.*, 1981.

[2] B. A. Boyter and J. K. Aggarwal. Recognition with range and intensity data. In *Proceedings of the Workshop on Computer Vision: Representation and Control, MD, 1984*, pages 112–117. IEEE, 1984.

[3] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, Jan. 1972.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[5] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, Boston, 1997.

[6] A. Hertz, E. Taillard, and D. de Werra. Tabu search. In *Local search in combinatorial optimization*, pages 121–136. Wiley & sons Ltd., 1997.

[7] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–17, July 1996.

[8] X. Y. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scanline grouping. *Machine Vision Applications*, 7(2):115–122, 1994.

[9] X. Y. Jiang and H. Bunke. Robust and fast edge detection and description in range images. In *MVA'96 IAPR Workshop on Machine Vision Applications*, pages 538–541, Japan, 1996.

[10] A. A. G. Requicha. Representation for rigid solids: Theory, methods, and systems. *Computing Surveys*, 12(4):437–464, 1980.

[11] G. Roth and M. D. Levine. Extracting geometric primitives. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 58(1):1–22, July 1993.

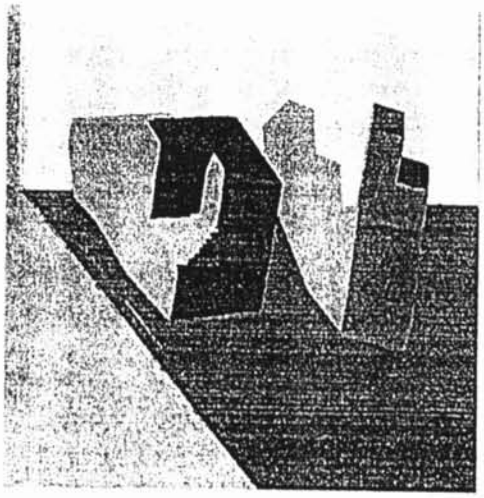

Figure 1. The range image abw.test.9

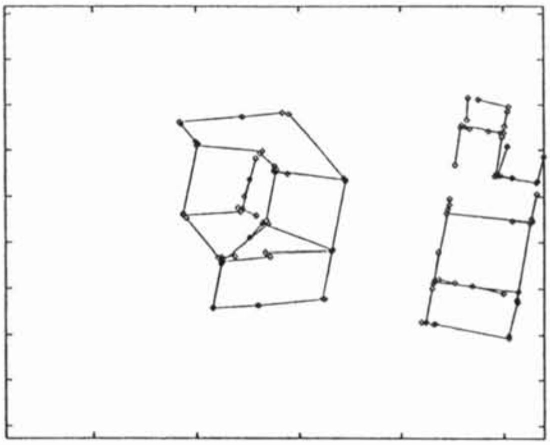

Figure 2. The results of the simple line detection algorithm described in section 2 on abw.test.9

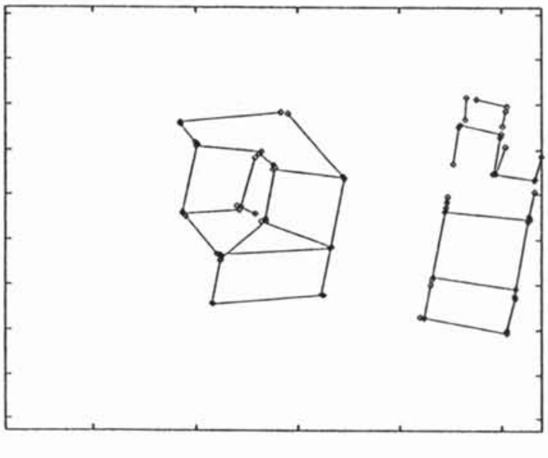

Figure 3. The results of tabu search on abw.test.9