CONSTRUCTIVE THEORY OF FUNCTIONS, Varna 2002
(B. Bojanov, Ed.), DARBA, Sofia, 2003, pp. 354-361.

An Algorithm for Recognition of Linear Surfaces

Žarko Mijajlović and Dragan Urošević *

This paper is a part of the research on a project concerning the analysis of 3D contours. Namely, the problem of pattern recognition is becoming very actual, especially in connection to automation and robotization of industrial production. Therefore, this problem is considered in many mathematical disciplines: discrete mathematics, geometry, numerical mathematics, topology, combinatorics, computer science, etc. Besides various interesting mathematical contents, there are very important and rather concrete applications. All these led to the high activity in this field: publication of scientific articles and books, software development and construction of specialized facilities. A good review of this subject and the bibliography can be found for example in [5].

1. The Project

At this moment a project under the name *Mathematical methods and algorithms for recognition of 3D contours* is ran jointly by the Mathematical Institute and the Mathematical faculty in Belgrade. The research subject of the project is the development of mathematical methods and design of algorithms for the recognition of a class of surfaces in the three-dimensional space represented by a finite set of points. Let us say few words about the research in the project.

The use of equipment for 3D scanning (3D scanners) in modern production processes and other areas of industry and technology is of the growing importance. 3D scanner are used for scanning surfaces of real three-dimensional objects. After scanning of an object having a contour C, the image of its surface is obtained, i.e., a finite set S of points, where points are represented by their spatial coordinates x, y, z. In most cases, S is the only information we have about the contour C. However, in many cases we need to know various mathematical properties of the contour C: geometrical, analytical, differential, fractal, etc. Reconstruction and recognition of the contour C according to the

*This paper was presented at the workshop "Multiscale approximations (wavelets, splines, RBF) and applications to Image and Signal Processing" accompanying CTF 2002.

image S is a hard problem in general, and for its solution various mathematical methods are necessary. In our project, we are focused mainly on polygonal surfaces, then on the second order surfaces, and in some cases on surfaces of irregular shape. Another type of problems that we are interested in is the degree of similarity of scanned objects in respect to their geometrical types.

In our research we use the following methods: signal processing (digital image processing), fractal geometry, numerical mathematics and machine vision. Besides customary operations necessary in the early stage of the analysis of the image S (smoothing, clustering, and noise filtration), we use other methods and algorithms. These techniques include edge detection (for example Sobel and Roberts edge detector), as well as methods and notions of fractal geometry (for example, fractal dimension, fractal images of surfaces, fractal compression). The first phase of our research includes the use of software packages *Mathematica 4.0* and *Matlab 6.0*. Occasionally, we use Lindermayer systems (a language for self-similarity). In the later phase of our project we shall implement so developed algorithms. For obtaining the experimental data, we use an industrial 3D scanner, provided by the German company MEL Micro Electronic GMBH (Munich, Germany). Also, we expect that we shall have the opportunity to use a relatively fast multiprocessor computer that Mathematical Institute in Belgrade expects to have in the near future.

2. The Scanner

The main source for our analysis of 3D images is a 3D scanner. Namely, we are using a scanning device M2D, produced by MEL Mikroelektronik, for

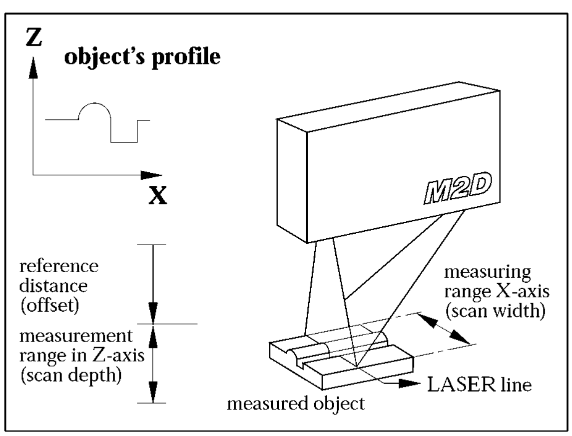

Figure 1

acquisition of data needed to build 3D images. We are going to give a short description of this device in order to have an idea about the size and the resolution of the image, as well as possible applications.

2.1. Method of Operation

The Scanner M2D provides 2-dimensional measuring of profile heights. The measured objects may consist of various materials. A laser line is projected onto the target and the distance to several points of the object is measured by triangulation. The diffuse reflected light from the target is projected onto a 2-dimensional CCD-Array, built into the scanner. The Scanner is suitable for the geometric profile recognition and distance measurement. The height (Z-axis) by distance (X-axis) is measured. The result is the contour of the object at the point of measurement. Every 20 ms a complete scan of 283 points of measurement, or every 40 ms a scan with 566 points of measurement is done. The measurement range works from 5...1200 mm (X-axis) and 8...600 mm (Z-axis).

2.2. Typical Applications

Precise guidance of handling-robots: disc mounting, positioning of car-bodies, gap adjustment in the automotive industry; welding seam tracking and quality assurance. Fast measurement of: fast moving objects, gaps, edges, width an depth of slots, 3D shapes and profiles, SMD components in the production process. Parts handling: collision avoidance, object presence and positioning.

3. Filtering

In designing algorithms for working with a set of data which represents an image, usually two basic goals are present. The first one concerns various transformations of the image, as of resizing, sharpening, distorting, blurring, lightening, etc. The second one concerns determination of some key constants "of the semantical nature" assigned to the image. For example, if the image represents an ellipsoid, then these constants would be the lengths of axes of the ellipsoid. Of course, there are other problems concerning images, for example ray tracing, image compression, computer generating images, the theory of fractal images and fractal dimensions, etc.

Let us consider for the moment the first type of algorithms. Any application of transformation on an image based on these algorithms are called *filtering*, while the transformation itself is called *a filter*. In most cases, before we start some mathematical analysis of the image, firstly some filters are applied, for example noise reduction and smoothing of data. To be specific, let us assume that a gray image $\mathcal{S}$ is represented by a matrix $S = ||s_{ij}||_{m\times n}$. Here, (i, j)

represents a pixel of the picture, while s_{ij} stands for the level of the grayness assigned to the pixel (i, j). If G is the set of all possible levels of grayness, say $G = \{0, 1, \dots, 255\}$ (if $G = \{0, 1\}$ then $\mathcal{S}$ will be called bitmapped image), then $s_{ij} \in G$, and a filter would be any map F which transforms the level of grayness s_{ij} of the pixel (i, j) into another level s'_{ij}. Let us consider in more details how some of the common filters are constructed.

According to the previous paragraph, at the first sight we would like to define a filter as a map $F : (i, j, s_{ij}) \to (i, j, s'_{ij})$. However, the assignment of new levels of grayness to the pixel (i, j) obviously depends on the neighboring pixels, i.e., it is context sensitive. For example, the averaging filter, closely related to the reduction of the contrast, would be defined by

$$F(i, j) = A \bullet (i, j) = \mathrm{int}\left(\frac{1}{9}\sum_{p=i-1}^{i+1}\sum_{q=j-1}^{j+1} s_{pq}\right) \tag{1}$$

where int is the integer part function, $A = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, and $\bullet$ is the obvious external operation applied on (i, j).

Thus, in this case the filter is defined by a 3×3 matrix A which selects pixels neighboring the pixel (i, j) and by (1) computes the new level of grayness at (i, j). We shall call the matrix A the core of the linear filter F. Thus, following just described idea, in general a linear filter F is defined by the core, a $(2n + 1) \times (2m + 1)$ matrix $A = ||a_{ij}||$, and

$$F(i, j) = A \bullet (i, j) = \sum_{p=-m}^{m}\sum_{q=-n}^{n} a_{ij} s_{i+p, j+q} \tag{2}$$

We are supposing that $A = \begin{bmatrix} a_{-m,n} & \cdots & a_{0,n} & \cdots & a_{m,n} \\ \vdots & & \vdots & & \vdots \\ a_{-m,0} & \cdots & a_{0,0} & \cdots & a_{m,0} \\ \vdots & & \vdots & & \vdots \\ a_{-m,-n} & \cdots & a_{0,-n} & \cdots & a_{m,-n} \end{bmatrix}$

For example, the effect of the filter "sharpening from the left",

$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & -12 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ is given at the Figure 2.

The process of segmentation is an important step toward the determination of semantic content of an image. A part of this process is an edge detection,

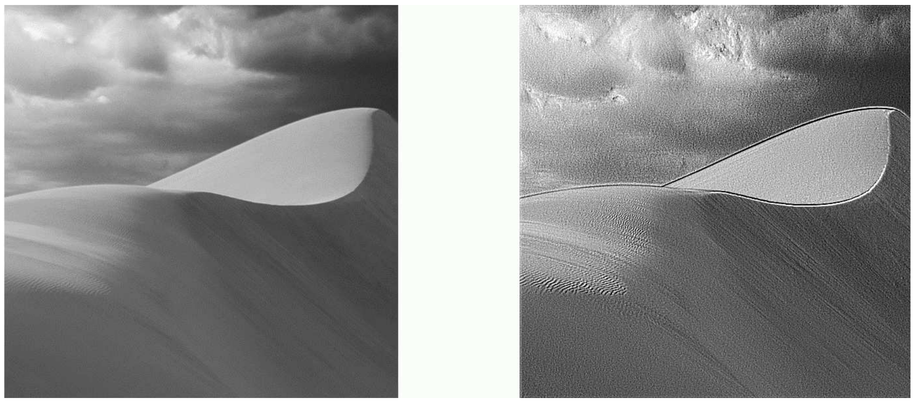

Figure 2

as a procedure for classifying pixels within the image as belonging to a certain type of texture. The common edge detection filters are:

Sobel edge detector given by the pair of matrices

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad D_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix},$$

and *Roberts edge detector* also given by the pair of matrices

$$D_x = \tfrac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad D_y = \tfrac{1}{2}\begin{bmatrix} 0 & 0 \\ -1 & -2 \end{bmatrix}.$$

Most edge detectors are based on discretization of the first or the second derivative. Namely, both derivatives nicely exhibits edges in stepwise signals.

So far about definitions and examples of linear filters. Of course there are nonlinear filters, such are the most of rendering filters (lens flare, lightning, 3D transforms, etc). However, for linear filters other questions can be stated. We shall mention here two basic problems.

– Which linear filters F are invertible? That is, for what F there is a linear filter H such that for all images $\mathcal{S}$, $HF(S) = S$.

– Some filters F introduce noise in the image. We can restate this fact in terms of information theory. We may assume that every image $\mathcal{S}$ carries some information. Then S may be considered as a sequence of discrete stochastic variables which assigns to every pixel the level of grayness. Thus, we can assign to S the entropy e, and to the filtered image the entropy e'. Obviously, one can measure the level of introduced noise by measuring the change of the entropy e. For invertible F, one can prove $e = e'$. Also, one can prove that for a bitmapped image, it's entropy is greatest if the number of 1-bits is about equal to the number of 0-bits of the image.

4. Contour Recognition

Our algorithm for recognition of linear surfaces represented by a set of experimentally collected data – coordinates of points that belong to the surface, is not limited only to data collected by 3D scanners. Namely, besides 3D scanners today there are various types of equipment able to scan three dimensional surfaces, for example georadars, ultrasound detectors etc. By emitting some kind of electromagnetic rays (for example optical rays by use of laser diodes), scanner is able to collect the space coordinates of points on the scanned surface and usually one more data – the intensity of the reflected ray. Therefore, we arrive to the following pure mathematical problem:

Let $S \subset R^4$ be a finite set of quadruples of real numbers, and suppose that the projection set $S' = \{(x, y, z) | \ (\exists u \in R)(x, y, z, u) \in S\}$ is the set of coordinates of points of a two-dimensional surface $\mathcal{S}$ in R^3. By examining the set S', classify the surface $\mathcal{S}$.

In the studying of this problem, we should consider the origin of data contained in the set S': The surface $\mathcal{S}$ is not an ideal mathematical surface, the measured data are just approximations of true coordinates of points, and some external influences on the measurement process are possible. Therefore, a part of recognition process must be some preparation of row data as of filtering and smoothing. For example, the intensity (the fourth coordinate u) may be used to omit a point from the set of data if the intensity at the point is to weak or to strong, or if it differs to much from the neighboring points.

We proposed an algorithm for recognition of linear surfaces $\mathcal{S}$ represented by the set S. Namely, we suppose that the scanned surface $\mathcal{S}$ is linear, i.e., it consists of polygonal surfaces. The algorithm finds these polygonal surfaces by determining the finite set of nodes – points at which the polygonal lines, boundaries of the polygonal surface, breaks into segments. The central part of the algorithm is a procedure of moving through the set of data a regression line determined by a selected number of points from the surface $\mathcal{S}$. One of the advantage of the algorithm is that the smoothing of the collected data set S is not necessary. Also, we have a good control in the process of recognition by the possibility of fine tuning of various parameters of the algorithm (e.g. number of points which determine the regression line, the rate of changing of slope of the regression line, etc). A software for 3D scanners is developed based on this algorithm, and good results were obtained in comparison to some programs supplied with commercial 3D scanners.

5. Appendix: Description of the Program

The program based on the algorithm has the following functions: **1.** Plots coordinate data on the screen (left upper corner of the screen). **2.** Plots luminosity data (left lower corner of the screen). **3.** Filters data by use of two criteria: intensity (luminosity) of the laser ray, and large oscillation of data of near points. **4.** Plots filtered coordinate data on the screen (right upper corner of the screen). **5.** Finds node (breaking) points of so obtained curve (contained in the output file). **6.** Plots node points and the linear (polygonal) approximation of the curve (right lower corner of the screen).

The program is written in C and it is designed to work on PC platform (MS DOS, WIN9x, NT).

Parameters of the program

Input file – name of the input file, *Output file* – name of the output file.

Number of points in the cluster. The program works with clusters – sets of points of the size represented by this number. This number also controls small variations (errors) in the measurement. Namely, if this number is bigger, the lesser is the influence of these errors ("the roughness of the curve") on the final computations of nodes.

Tolerance of slope change. The program is trying to find the change of slope of the curve (the angle between two consecutive segments). This change indicates to the program that a node is in the examined cluster. The tolerance of the slope change is the number of verifications that the suspected point is a node. Therefore, if this number is larger, it is more certain that the displayed point is a node. This number also controls small variations of the propagation of the curve ("local errors on the scanned object"), as of small holes, hills and displacements of parts of the curves. Therefore, if this number is lesser, the program will find such disruptions of the curve, otherwise they will be omitted in the final computation.

Tolerance of slope. The program is trying to find not only the change of the slope of the curve, but also angles between two consecutive segments that approximate two consecutive parts of the curve. The tolerance of the slope controls the lower bound for this angle.

Tolerance of intensity. This number controls the lower limit of the luminosity of points. If this number is small, points of weak luminosity are filtered. If this number is large points of weaker luminosity are considered in the final computation.

References

[1] D. Ballard and C. Brown, "Computer Vision", Prentice Hall, 1982.

[2] P. J. Burt, Fast filter transforms for image processing, *Computer Graphics and Image Processing* **16** (1981), 20–51.

[3] J. F. Canny, Finding edges and lines in images, Technical report 720, MIT, June 1983.

[4] I. M. Gelfand, "Lectures on Linear Algebra", Nauka, Moscow, 1966.

[5] M. J. Tarner, J. M. Blackledge, and P. R. Andrews, "Fractal Geometry in Digital Imaging", Academic Press, 1998.

[6] H. Wechsler, "Computational Vision", Academic Press, Boston, 1990.

Žarko Mijajlović

Faculty of Mathematics
University of Belgrade
Studentski trg 16
11000 Belgrade
YUGOSLAVIA
E-mail: zarkom@mi.sanu.ac.yu

Dragan Urošević

Mathematical Institute SANU
Knez Mihailova 35
11000 Belgrade
YUGOSLAVIA
E-mail: durosevic@mi.sanu.ac.yu