Reconstruction of Solid Shape[1]

Tomáš Pajdla
Computer Vision Laboratory
Czech Technical University
CZ-121 35 Praha 2, Karlovo nám. 13
pajdla@vision.felk.cvut.cz

Research report No. K335-1995-100

November 16, 1995

Czech Technical University
Faculty of Electrical Engineering
Department of Control
Computer Vision Laboratory

[1]This research was supported by the Grant Agency of the Czech Republic, grants 102/93/0954, 102/95/1378 and European Union grant Copernicus No. 1068 RECCAD.

Preface

This work is submitted to fulfil the reqirements set upon the finishing of the first period of author's graduated studies at the Czech Technical Univerity, Prague, Faculty of Electrical Engineering, Department of Control Engineering. This work has been carried out under the supervision of Mr Doc. Ing. Václav Hlaváč, CSc. who also supervises my doctoral studies.

Abstract

Traditional approaches to geometrical modelling of real objects build upon human interaction with computer systems. An alternative method allowing to build complex geometrical models by "looking" at real objects is proposed in this work. Our approach to the model reconstruction is, in general, akin to an autonomous exploration. The system scans the object from different viewpoints and in-assistadly constructs its geometrical model. Such a system consists of a 3-D scanner providing samples from object's surface, a controlled tilt and slant table carrying the object, and a computer program automatically building geometrical models from measured data.

This work aims at clearly defining the problem. It also comments on related approaches and topics in the literature and presents a vision of a complex modelling system.

Contents

1 Introduction

This work investigates representation and automatic acquisition of 3-D solid shape model by the means of local measurement of its geometry. Our ultimate goal is to develop an automaton that will be able to infer an abstract mathematical model of shape from an example of a 3-D object and represent this model in the computer. Such a model should allow to manipulate, view, and "edit" the shape of the object under scrutiny.

Understanding the shape of the objects around us has always been an interesting topic. From the times of old Greek mathematicians, people build and maintain a complex axiomatic theory, geometry, describing shapes of objects that can be seen, touched, or only imagined as pure abstractions of real objects. Till the second half of 20th century, however, abstract models of shape have always been a matter of human beings and their imagination. Real objects were observed and measured and their abstract description was then manually constructed by some more or less clever *designer* in the form of drawings or symbolic mathematical language. Such an approach has in last few decades found many applications in Computer Aided Design (CAD) where designers construct the geometry of objects with the aid of a computer using more or less restricted geometrical language depending on the generality of the programs at hand. The needs for advanced tools supporting representation, manipulation, and visualisation of geometry gave rise to *Computer Graphics*.

Computer Vision has also started a few decades ago but with the aim to provide methods allowing machines to *see* and *learn* about their environment. Already early outstanding works in this field concentrated towards understanding the shape of objects as well as sensors (eyes) used to see shape. Stereo vision allowing humans to see depth was thoroughly studied [1] and great deal of attention has also be paid to understanding shape from one shaded image. Later, the whole gamut of approaches, so called *Shape from X*, to reconstruction of the shape was invented and new reconstruction methods are still being found.

Recently, for instance, the concept of stereo has been generalized [2, 3]. Two cameras in known relation can be replaced by many cameras (or one moving camera) in unknown relative positions with unknown calibrations. Allowing for uncalibrated visual system greatly reduces costly calibration procedures and certainly eases demands on the system when implemented in praxis. On the other hand, great cost is paid by loosing Euclidean geometry which is replaced by geometry affine or even projective. It means that the coordinates of points delivered by an uncalibrated reconstruction differ from the "correct" Euclidean coordinates by some unknown, albeit same, affine or projective transformation.

Due to inevitable selfocclusion of solid objetcs all "vision based" shape reconstruction methods can deliver only partial measurements if taken from one view point. In order to construct a complete model of an object, a fusion of data measured from different viewpoints is called for. This work proposes an *automatic system for building complex geometrical models* by "looking" at real objects. Our approach to the complete model reconstruction is akin to an autonomous exploration, i.e. the system scans the object from different viewpoints and in-assistadly constructs its geometrical model. Such a system consists of a 3-D scanner providing samples from object's surface, a controlled tilt and slant table carrying the object, and a computer program automatically building geometrical models from measured data.

This report is organized as follows. In Section 2 we define the class of objects which we study and the notion of their shape. The model of shape and its representation in the computer is discussed in Section 3. In section 4 we continue with the presentation of methods used to measure surfaces of objects. Section 5 comments on previous works closely related to geometrical modelling from measured data. Finally, Section 6 clearly defines the problem complete model construction form partial observations and discusses some possible approaches.

2 Solid object and its shape

The object and its shape can be defined in a number of ways. It suits to our purposes to adopt the approach of Koenderink [4] who defines the shape operationally, i.e. in connection to an intended task: *"Shape is operationally defined. Thus things do not have a shape the way Santa Claus has a red suit"*. We would like to stress that there is therefore no mathematical theory of shape. In this work, however, we define the shape from the geometer's point of view, hence the theory of surfaces can be applied.

Our final goal is to provide a method for reconstruction (copying) objects of real world by the means of physical measurements, a computer and a replication machine, e.g. a milling machine. For practical reasons we focus on objects originating in human production and serving to special human purposes. Furthermore we restrict ourselves to objects that are created in technical production by using force or milling. We also admit natural objects but only if theirs properties roughly corresponds to artificial objects and so that they can be successfully copied by using methods common in industrial production.

Our definition of the object is driven by the following observations:

- The properties that we are interested in, i.e. physical properties of investigated object. In this respect, the object is *a compact set in 3-D Euclidean space that does not change its spatial arrangement when manipulated in usual physical conditions.*

- The sensors which are common in computer vision and which are limited so that only outer visible surface of an object can be measured. It means that *the object is from our perspective defined by its outer visible surface.*

- The interest is in macroscopical features of objects and therefore its *surface seems to be continuous and piecewise smooth.* We are not interested in any microscopical effects but it is required that the measurement is dense enough to measure properly all geometrical features on the surface.

By the shape of the object, which is of primary interest here, we understand *a closed, orientable, continuous, piecewise smooth, 2-dimensional regular submanifold (surface) embedded in* $\mathbb{R}^3$. We say that two objects have the same

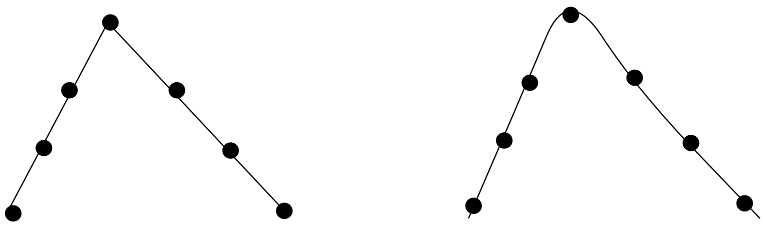

Figure 1: The set of points on the surface can be modeled by piecewise smooth or smooth manifold.

shape if their surfaces differ only up to some Euclidean transformation. Since the object can be measured only by some physical measurement, we also assume that their surface is closed and bounds some volume. Let us mention here that no 3-D data acquisition method which is later considered provides closed surface from one measurement. Closed surface (shape of the object) emerges only as a result of connecting more measurements together what is the main goal of this work.

3 Representation of shape

In the previous section, the shape of an object was given as a piecewise smooth submanifold. Here we claim that any really mensurable shape can be also well modelled by a smooth manifold. The reason is that we anyway can measure only discrete points from object surface. It is therefore not possible to say whether there is really a discontinuity or only a high curvature, see Figure 1.

Let us discuss different definitions of a submanifold. First we define a notion of manifold as it is given in [5].

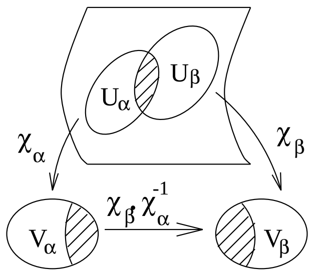

Figure 2: Coordinate charts on a manifold.

Definition: An *m-dimensional manifold* is a set M, together with a countable collection of a subsets $U_\alpha \subset M$, called *coordinate charts*, and one-to-one functions $\chi_\alpha : U_\alpha \to V_\alpha$ onto connected open subsets V_α of $I\!R^m$, called *local coordinate maps*, which satisfy the following properties:

(a) The coordinate charts cover M:

$$\bigcup_{\alpha} U_\alpha = M$$

(b) On the overlap of any pair of coordinate charts $U_\alpha \cap U_\beta$ the composite map

$$\chi_\beta \circ \chi_\alpha^{-1} : \chi_\alpha(U_\alpha \cap U_\beta) \to \chi_\beta(U_\alpha \cap U_\beta)$$

is a smooth (infinitely differentiable) function.

(c) If $x \in U_\alpha, \tilde{x} \in U_\beta$ are distinct points of M, then there exist open subsets W of $\chi_\alpha(x)$ in V_α and $\tilde{W}$ of $\chi_\beta(\tilde{x})$ in V_β such that

$$\chi_\alpha^{-1}(W) \cap \chi_\beta^{-1}(\tilde{W})$$

is empty.

There are number of ways how to define a submanifold. A submanifold S in $I\!R^3$ can naturally be defined in implicit form by the vanishing of a continuous, smooth function $F : I\!R^m \to I\!R$, which satisfies maximum rank condition everywhere where it vanishes and has defined first derivatives [5]:

$$S = \{F(\mathbf{x}) = 0\}, \tag{1}$$

Implicit formulation defines target manifold by a constraint F applied to some other, host, manifold. The continuity and smoothness of F guarantees that the resulting submanifold behaves reasonably, i.e. there are no selfintersections, it is orientable, regular, etc. This is great advantage over other alternative definitions. Geometry as well as topology of the manifold are given just by setting one equation.

The main drawback of implicit definition consists in complexity of F for nontrivial objects. Each shape is best described by its own and only by its own function F which is, of course, unknown and often so very complicated that there is no chance to find it. There is even no way to represent it analytically.

In special applications, however, like is curve and surface evolution, F can be approximated by their values sampled on a finite 2-D or 3-D grid. For instance, Sethian and Osher in [6] represent curves as equilevels of a 2-D function $F(x, y)$. The function $F(x, y)$ is represented by its values sampled at discrete grid points x_i, y_i. This representation becomes intractable in 3-D.

The alternative to the implicit definition is the explicit, or parametric, definition of a submanifold.

Definition: Let M be a smooth manifold. A *submanifold* of M is a subset $N \subset M$, together with a smooth one-to-one map $\phi : \tilde{N} \to N \subset M$ satisfying the maximum rank condition everywhere, where the *parameter space* $\tilde{N}$ is some other manifold and where ϕ has defined derivatives. The map ϕ is often called an *immersion* and defines the parameterization of the submanifold N.

Parametric definition and representation is often used in computer graphics and geometric modelling. Its main advantage is that it is easy to represent arbitrary surfaces in a uniform manner. It is enough to choose representation of parametric domain $\tilde{N}$ and the immersion ϕ.

There is a lot of freedom in selecting $\tilde{N}$ and therefore it can be represented without any difficulties in a simple way as a, for instance, triangular mesh. Immersion ϕ is often chosen to be from some class of parametric functions, e.g. polynomials, defined locally over the triangles in the mesh. The polynomials are simple and all manifolds can be represented by them if they are defined over the triangles, with desired precision. If higher precision is required, more triangles have to be added.

Unfortunatelly, the construction of ϕ to have desired properties, like continuity, is not as simple task. For instance, it has been already shown by Herron [7] that it is impossible to construct closed, non-degenerate C^1 surface of arbitrary topology just by stitching B-splines patches at their boundaries. The notion of geometric continuity has been introduced in order to find a parametric representation of surfaces with smoothly varying normal over arbitrary domains. Generalized B-spline surfaces of arbitrary topology, as proposed by Hoppe and DeRoose [8], can model closed surfaces, yet still imposing restrictions on the connectivity of B-spline control meshes[1]. Another scheme was given by Veltkamp in [7] where by splitting Bezier triangles, G^1 closed continuous surface can be constructed.

The important aspect of explicit representation stems from the fact that parametric domain and immersion are represented separately. The parametric domain can not be constructed arbitrarily if it is in the form of a manifold. For instance, the triangulated mesh representing $\tilde{N}$ must have the same topology as N if this is infinitelly differentiable. It means that the representation in the computer has to be kept in a proper format. The problems with topology might appear when two surface patches are merged together. Merging must ensure that the resulting representation endow the structure of manifold. This is not a simple task.

In our work, $\tilde{N}$ is constructed as a triangulated mesh. This choice is primarily driven by the fact that every surface allows triangulation. The approximation of ϕ, $\bar{\phi}_i$, can be then defined over each triangle as a linear interpolation of its vertices. This representation naturally supports measured data if the points measured from surfaces are used as vertices of the triangulation.

4 Visual Surface Reconstruction

This section comments on available techniques that deliver measurements of a 3-D surface. By *visual surface we understand a part of a surface that is visible from a point in space.* Since it is always the case that only a part of the surface is visible and mensurable. Due to self occlusion only visual surface can be seen, never the whole object surface.

[1] The mesh is either entirely constructed from 4-sided patches or exactly four faces can meet at a vertex.

4.1 Reconstruction of points

A large number of depth perception methods, called *Shape from X*, have recently appeared as the outcome of the intensive research carried out in the area of active and passive range sensing [9, 10, 11, 12]. For our purposes, only relatively precise and dense measurement is benefitial and therefore we will shortly discuss the principles of Shape from Stereo and Shape from Structured Light as these are currently the most promising approaches.

Stereo vision uses two perspective cameras placed at different locations in space to look at the same 3-D point. Each camera therefore provides a light ray in the space that projects the 3-D point onto the retinas by connecting the point, the centers of projection, and the point in the retina, Figure 4. The intersection of two rays then readily gives the desired 3-D coordinates of the point in space. Computing the coordinates of 3-D points by intersecting the rays passing through the point's image projections is usually called triangulation. Bottleneck of this approach consists in automatic extraction of corresponding projections from images. This correspondence turns out to be rather difficult problem as it is not easy to predict how the projections are related without knowing where the original point is with respect to the cameras. The use of an

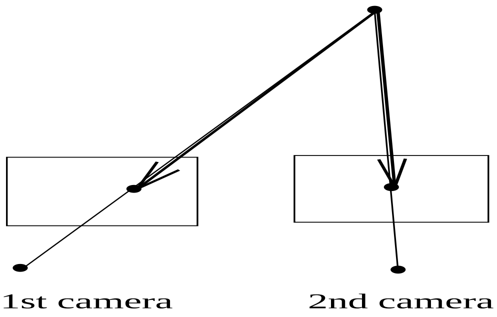

Figure 3: *Stereo measures 3-D point by intersection of rays emanting from two displaced cvameras.*

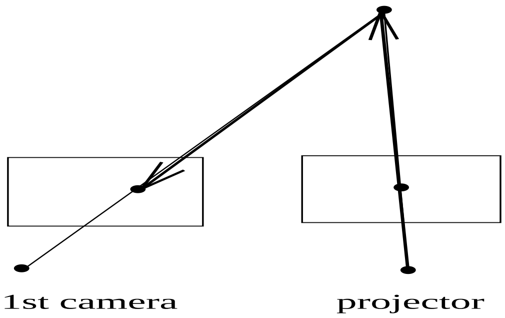

Figure 4: *Active range finder replaces one camera by a projector of rays.*

active controlled illumination (structured light) can remove or at least alleviate "chicken and egg" correspondence problem. Usually, one camera is replaced by a source of light rays and a coding scheme allowing unique identification of the rays from the image of the illuminated scene. For instance, the most trivial setup would use just one ray at a time so that the image had contained maximally one bright point. Such a setup leads to very trivial solution to the matching of projections. However, this naive solution turns out to be very impractical since the one whole image has to be captured and processed to measure one 3-D point. More advanced but still easy and efficient ray coding scheme projects planar pencil of rays at a time, so that an intersection curve, light trace, can be seen in each image, Figure 3. Ray identification is still unique albeit much faster extraction is enabled, see [13].

Figure 5 shows the principle of the Laser Plane Range Finder that projects a light plane at a time. See also [13, 14] for detailed description of active ranging systems that we developed at the Computer Vision Laboratory, Czech Technical University, Prague and ESAT, Katholieke Universiteit, Leuven. Figures 6

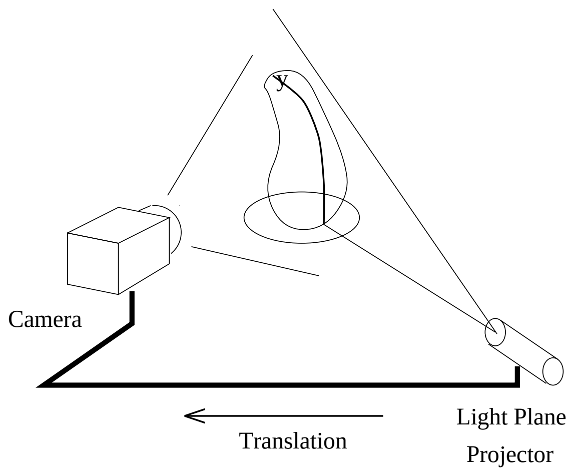

Figure 5: Laser Plane Range Finder projects a plane of active rays.

and 7 show two examples of visual surface reconstruction using a structured illumination. Points on the surface are connected by the rectangular grid which is created in the process of reconstruction.

All mentioned systems use the same principle for computing the points in space, i.e. triangulation of rays passing through two projections of a point. The difference consists in finding the correspondences, not in computing the coordinates of 3-D points. From this point of view, there is no difference between active and passive range sensors and therefore the same analysis of sensor geometries can be used.

4.2 Reconstruction with or without the calibration

In principle, the triangulation can be done only if we exactly know how the pair of cameras or the camera and the ray projector are related and what are their calibrations (focal length, aspect ratio, etc.) The spatial relation of the cameras and their calibrations give altogether the calibration of a stereo–rig. If the calibration of a stereo–rig is not known or if it is not known exactly, computer programs can deliver some coordinates but these are not anymore "true" Euclidean coordinates as desired.

In the following text, the vectors will be denoted by bold lower characters as $\mathbf{u}$, whereas matrices will be represented by italics capital characters as A. $\mathbf{u}^T$ then stands for the transposition of a column vector $\mathbf{u}$.

A camera projects a point $\mathbf{x} = (x, y, z, 1)^T$ from an Euclidean space to a point $\mathbf{u} = (u, v, 1)^T$ in the image by the mapping

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} p/r \\ q/r \end{pmatrix} \quad \begin{pmatrix} p \\ q \\ r \end{pmatrix} = M\ \mathbf{x}, \tag{2}$$

where u, v stand for affine image pixel coordinates of detected point, i.e. the coordinates that are actually available in the image, $p,\ q,\ r$ are linear homoge-

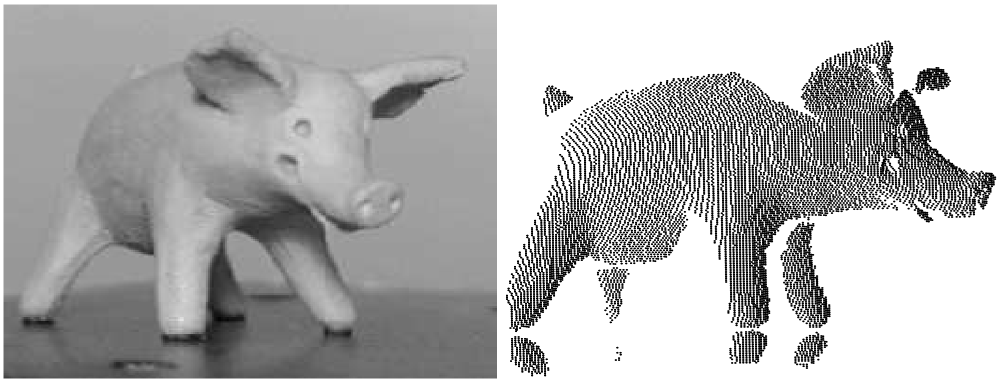

Figure 6: *Gray level image of a pig and its 3-D reconstruction using Laser Plane Range Finder.*

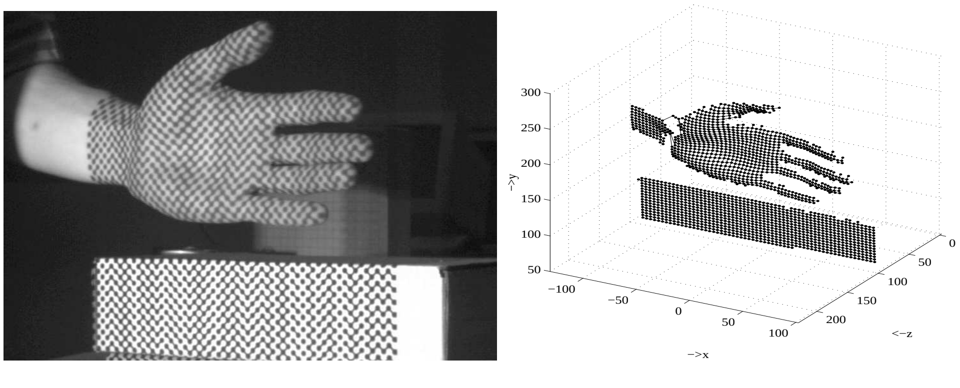

Figure 7: *Gray level image of a human hand and its 3-D reconstruction using Binary Illumination Coded Range Finder.*

neous coordinates of the pixel. $x,\ y,\ z$ represent Euclidean coordinates of a 3-D point that projects into the pixel (u, v). Linear projection matrix $M \in I\!R^{3\times 4}$ can be decomposed through QR decomposition,

$$M \;=\; K\ R\ \Big(\ I \mid -\mathbf{t}_C\ \Big), \tag{3}$$

into the product of a rotation matrix R and a upper triangular matrix K containing the internal camera calibration parameters. $\mathbf{t}_C$ stands for the position of camera projection center. The entries of K are called *internal camera calibration parameters.*

Let $\mathbf{x}$ represents a point in an 3-D Euclidean space, then its projections $\mathbf{u}$ and $\mathbf{u}'$ into two cameras given by camera matrices M and M' are given by

$$\mathbf{u} \;\simeq\; M\,\mathbf{x} \quad \mathbf{u}' \;\simeq\; M'\,\mathbf{x}. \tag{4}$$

The symbol $\simeq$ in (4) means that the projections are determined only up to an unknown scale factor. Equations in (4) can be rewritten, see [15] for details, to form a linear homogeneous system with respect to $\mathbf{x}$

$$\begin{pmatrix} u\ \mathbf{m}_2^T & - & v\ \mathbf{m}_1^T \\ u\ \mathbf{m}_3^T & - & \mathbf{m}_1^T \\ u'\mathbf{m}'^T_2 & - & v'\mathbf{m}'^T_1 \\ u'\mathbf{m}'^T_3 & - & \mathbf{m}'^T_1 \end{pmatrix} \mathbf{x} \;=\; 0, \tag{5}$$

where $\mathbf{m}_i^T$ and $\mathbf{m}'^T_i$, $i = 1\ldots 3$, are i–th rows of M and M' respectively. The 4×4 matrix on the left hand side of equation (5) has rank 3 if $\mathbf{u}$ and $\mathbf{u}'$ are really projections of the same point.

To reconstruct the $\mathbf{x}$ we need to invert the projection through the knowledge of M and M' by solving (5) for $\mathbf{x}$. However, both M and M', one of them, or only some of their entries need not be known. It happens if we do not know the internal parameters of the cameras or their relative displacement.

In [2] it is shown that if the calibration of the rig is unknown, the reconstructed coordinates $\tilde{\mathbf{x}}$ differ from correct Euclidean ones $\mathbf{x}$ by some unknown projective transformation H,

$$\tilde{\mathbf{x}} \;=\; H\,\mathbf{x}. \tag{6}$$

H ranges from Euclidean over affine to projective transformation depending on the level of knowledge about the calibration and number of views available.

Notice, that H is the same for all points for one position and calibration of the cameras. It is completely independent from the world that is measured. Often, though the sensor is not calibrated, it is fixed, i.e. H is unknown but the same for the whole sequence of measurement, say within measuring one object from different viewpoints.

As we have already mentioned, we assume that the object is indeed a part of Euclidean space and that the change of sensor's viewpoint is therefore due to an Euclidean transformation of the object. Let $\mathbf{x}$ here represent coordinates of an arbitrary point of the object as expressed in some Euclidean coordinate system attached to the sensor taking the first view and $\mathbf{x}'$ stands for similar coordinates in the second view. Then by the above assumption

$$\mathbf{x}' = E\,\mathbf{x}, \tag{7}$$

where E stands for 4×4 matrix of the Euclidean transformation moving the sensor from the first viewpoint to the second one. Now, let us assume that both points $\mathbf{x}$ and $\mathbf{x}'$ are part of a visible surface and thus reconstructed by the sensor as the points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ respectively. From (6) and (7) it follows that

$$H^{-1}\tilde{\mathbf{x}}' = E\,H^{-1}\,\tilde{\mathbf{x}} \tag{8}$$

$$\tilde{\mathbf{x}}' = H\,E\,H^{-1}\,\tilde{\mathbf{x}}. \tag{9}$$

The matrix

$$T = H\,E\,H^{-1} \tag{10}$$

is not a general projective transformation, but a conjugate to a Euclidean transformation matrix E. There are two important properties of T namely that

- T and E have same eigenvalues,
- and the eigenvectors of T are transformed eigenvectors of E according to (6),

which allows to do affine or Euclidean reconstruction [16] by matching partial reconstructions of an overlapping data acquired from different viewpoints.

We have presented the methods used in Computer Vision for the visual surface reconstruction. Particularly, we have shown that there is no difference between passive and active ranging methods as far as perspective cameras are used as adequate mathematical models of projection. It is clear that any system aiming at construction of a complete surface from partial observations should allow for different geometries ranging from Euclidean over affine to projective one.

The last part of this section have also suggested that even though partial observations can only be reconstructed up to some projective or affine transformation, if put together Euclidean geometry might emerge.

4.3 Visual surfaces from point measurements

Our ultimate goal is to measure a visual surface, not only points on it. However, our sensors are discrete by their construction,[2] hence implicitly providing only points. We need therefore to interpolate or approximate these points to form a surface. Very straightforward and simple method might use proximity of points to construct their triangulation. In other words, measured points are used as vertices and close points are connected so that they altogether form a triangulated patch. Such an approach implicitly assumes that the level of detail is much coarser that the sampling of the surface allowing thus to replace shape between close points by their palanar approximation.

[2] A CCD camera consists of discrete elements

In general, it is not easy to generate such triangulation over an unorganized set of points in 3-D since triangles must satisfy certain constraints to form simplicial complex. See [7] for 3-D point set boundary computation based on γ-Graphs, [17] for 3-D point set triangulation using a smoothness criterion, and [18] for Delaunay triangulation based boundary computation. In some cases, e.g. when cameras are used, however, this task can be made easier.

If points are to be reconstructed by their projection onto a plane, reconstructed visible surface must be mapped to an image plane. The set of 3-D points sampled from the surface is thus in one-to-one correspondence with its projection to the image. In such a case, the triangulation of 3-D points can be defined in terms of a 2-D triangulation of their images, see Figure 8. Delaunay triangulation (sometimes called Dirichlet or Thiessen) can be a good choice because it provides the most equiangular triangulation in the plane and the "least rough" triangulation of the surface[3] [19]. Besides that, it can be computed with an $O(NlogN)$ algorithm. Of course, other planar triangulations might be used as well. Range finders often provide some relations between the points.

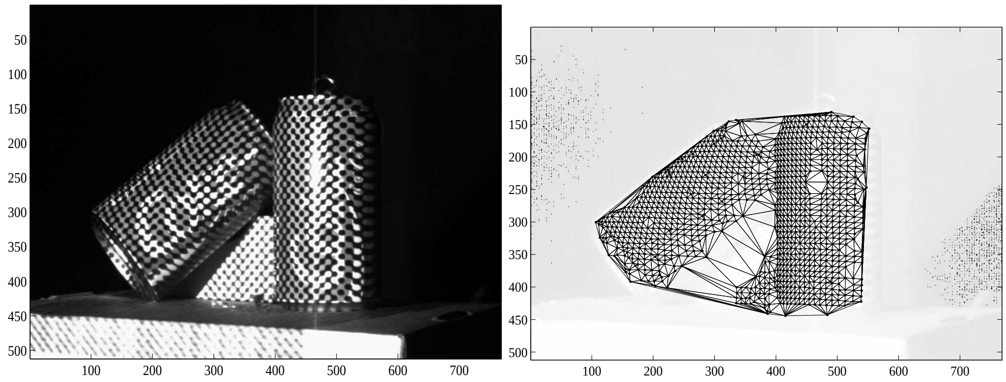

Figure 8: The image of a CocaCola can illuminated by a structured light and 2-D Delaunay triangulation of reconstructed points.

Binary Coded Range Finder [14], for instance, constructs 4-connected mesh as a byproduct of its performance, see Figure 7. This mesh is constructed on the basis of proximity of points in image. Links are established to minimize the Euclidean distance between neighboring image points in north, west, east, and south quadrants respectively.

In the case of Laser Plane Range Finder [13] a surface is reconstructed in planar scanlines. The scanlines are put one next the other by translating camera and scanner. This suggest to parametrize the surface across the scanlines by the motion of camera–projector rig; parameter t in Figure 10. The second parameter can be introduced by parametrizing the surface along the scanlines, Figure 9. In our setup, there is a ono-to-one relation between rows in the image (in fact all lines parallel to rows, not only the discrete set of rows) and a pencil

[3]The least rough means that among all possible triangulations in the plane the integral $\int_S (\nabla z(x,y))^2 ds$ is minimal, where the gradient is computed on a linear interpolation over triangles expressed as a function over the domain of triangulation.

of rays emanating from the focus F_P of the projector. All points in one scanline can be parameterized by their rays while the rays can be parameterized by the row coordinates of their rows.

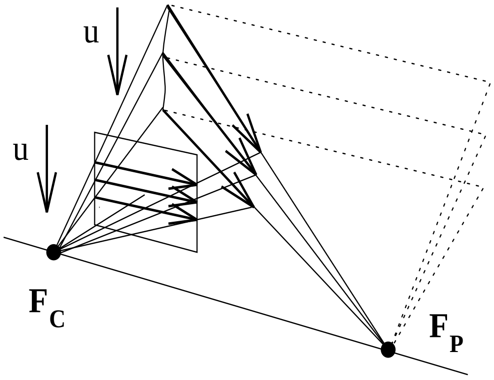

Figure 9: *The parameterization of the rays emanating from the projector.*

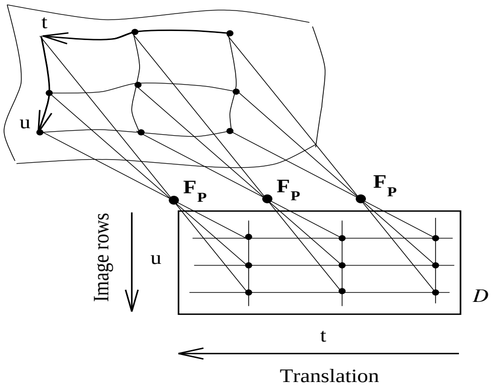

Figure 10: *Surface parameterization is composed from ray parameterization and the translation parameterization.*

This parameterization allows easy construction of 4-connected mesh following parametric curves just by connecting the points found in neighbouring rows of the image and points in neighbouring scans with the same image row coordinate. The assumption of surface continuity is implemented as the restriction on the distance of neighbouring points. Only the neighbours closer than a predefined ϵ are considered to lie on one surface next to each other. The points with no close neighbours are assumed to be outliers and are removed from measured data.

The message of this section is that sensors itself often provide useful, however partial, information about the connectivity o points. We believe that this information make reconstruction easier. Therefore we are bound to use it as much as possible.

4.4 Observability of Solid Shapes

Not all objects are fully observable due to inevitable selfocclusion of solids. Therefore a surface point can be measured by triangulation only if it is simultaneously visible from at least two distinct viewpoints **p** and **q** in space, Figure 11, under limited angle α. Now, let $\mathcal{H}$ is the convex hull of the surface and let us assume that the 3-D scanner is always outside $\mathcal{H}$. Then it is easy to see that only thouse points can be measured that

1. lie on $\mathcal{H}$, or

2. are visible from at least two points on $\mathcal{H}$ under angle that is greater than some *minimal observability angle* α_{min}, see Figure 11.

The above restriction naturally classifies the objects into two classes. In the first class, there are α–*observable* objects while int the second one α–*unobservable* objects fall. Observable objects can be measured completely. In practice, we often meet unobservable objects. It means that we will not be able to scan their curface completely and we can not use closeness of the reconstructed surface as a criterion for termination of the modelling process.

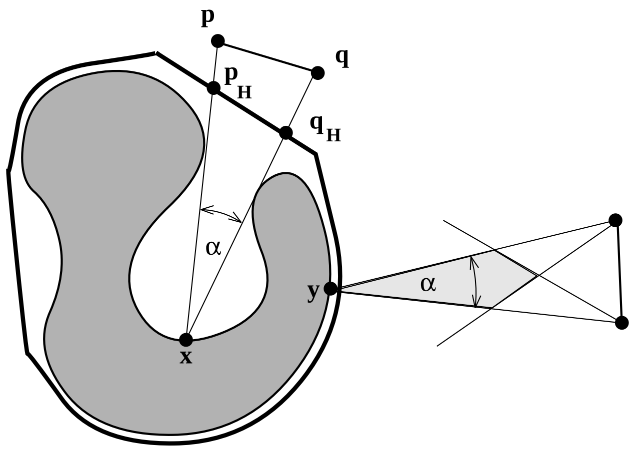

Figure 11: Mensurable point must be visible from at least two views points under angle greater or equal to α.

5 Related work

Maars theory motivated $2\frac{1}{2}$D surface reconstruction as a step towards reconstructing the geometry of complete objects. The works of Terzopoulos [20], Stevenson and Delp [21], Blake and Zissermann [22], Sinha and Schunck [23], and Szelisky and Terzopoulos [24, 20] solve reconstruction of surfaces which can be expressed as functions of two parameters. However, they did not address the reconstruction of closed surfaces. Contrary to a visual surface reconstruction, reconstructing closed surfaces requires to put multiple observations together while constraining its representation to be topologically correct.

This section gives an taxonomy of different problems related to modeling from measured data and a survey of related published approaches. We focus on the design of surfaces with prescribed topologies, on the problem of recovering a "valid" surface from scattered data, and on construction of a closed surface from a collection visual surfaces measured on one object from different viewpoints.

5.1 Designing closed surfaces with prescribed topology

Here we assume that scattered data points in 3-D space are available. The task is to find a closed surface of a given topology, i.e. a genus of the surface is known, passing through the points.

General approach to design of surfaces with prescribed topologies has been reported by Ferguson, Rockwood, and Cox in [25]. They build on the theory developed by Firby and Gardiner [26] stating that *a closed, orientable surface* *of positive genus g has a marked polygon with* $4 \times g$ *sides.* This allows to construct global parameterization of the surface using multiply periodic functions defined over the polygon. An embedding is then obtained as a solution of a 2–dimensional boundary value problem where the points provide the constraints forcing the surface to pass through them.

The theory does not handle sphere and all surfaces homeomorphic to it, hence they had to be treated by some different technique. Two other drawbacks of the above approach come to mind. First, coordinates of points at the boundaries of the marked polygon have to be supplied. Second, embeddings are computed by means of solving a boundary problem using finite elements and thus giving rise to a computationally intensive process.

And yet another issues which remain opened are how one can impose local discontinuities in orientation or how to replace the interpolation by an approximation and, of course, how to step from closed, complete surfaces to surfaces with boundaries.

5.2 Interpolating points by a valid surface

It is assumed that exact 3-D data points sampled from a surface are available. The task is to find a valid boundary (surface) through all points.

Veltkamp in [7] gives an algorithm for reconstruction of a closed simple polyhedral boundary from scattered points. It is stated that a brute force algorithm testing all possible combinations has at least $O(n^{5\ n})$ complexity (n is the number pf points) what is clearly intractable. Moreover, the boundary need not be unique. A γ-Neighbourhood Graph representation[4] is constructed and then a heuristic constriction[5] is applied in order to prune the γ-Neighbourhood Graph until all data points lie on its hull. The constriction starts with the convex hull and the deletion rule prefers "shallow" hull triangles to be removed first. The method is compared with other constriction approaches and its behaviour turns out to be better [6]. The whole process has at least $O(n^3)$ costs.

The main drawback of the method comes from the fact that a valid hull has to be always simple thus avoiding all topologies except spherical one.

5.3 Approximating points by a valid surface

Here, the task is to find a valid surface approximating scattered, noisy, 3-D data points.

Hoppe, DeRose et al. [8] developed the system that is able to reconstruct a valid surface approximating unorganized noisy points sampled from a surface.

---

[4] γ-Neighbourhood Graph in 3-D generalizes Delaunay tetrahedralization in the following manner. Each triangle in the tetrahedralization has the intersection of two balls of radii r_1, r_2, touching their vertices, empty. If, let say, r_1 is taken as the radius of the ball touching all four vertices of the tetrahedron then by changing r_2 the whole parametric class of graphs can be constructed ranging from the empty graph to the complete graph. For $r_1 = r_2$ a γ-Neighbourhood Graph boils down to a Delaunay tetrahedralization.

[5] The constriction is the process of deleting hull triangle from a graph in such a way that a hull of the new graph remains properly defined.

[6] It is not clear how they evaluate the goodness of results. Visually, I guess.

The key idea in their approach is to introduce the surface as a zero set of the signed distance function defined in the neighbourhood of point samples[7]. The signed distance is constructed by the following procedure. Each point $\mathbf{p}_i$ from the input data set gets assigned a plane that locally approximates points in its k–neighbourhood, i.e. its k closest points. The planes are represented by centers of gravity of the points in k–neighbourhoods, $\mathbf{o}_i$, and by their normals. Then, planes are consistently oriented (orientability of the surface is assumed). The absolute value of the distance of a point $\mathbf{p}$ to the surface is given by its distance to the plane, theirs center of gravity is closest to $\mathbf{p}$. The signum is given by the orientation of the plane. Once the distance is defined, the zero set is approximated by simplicial surface using the marching cubes algorithm.

The important features of the approach that make it very attractive can be found in the ability to represent any topology and handle incomplete surfaces. The knowledge of sampling rate and noise level can be utilized in order to determine portions of the surface insufficiently supported by data. The main bottleneck, on the other hand, can be found in determining consistent orientation of local planar approximations since the problem, as casted, is known to be NP-complex. Hoppe proposes a heuristical algorithm that works well for a number of examples. However, the orientability of resulting surface is not assured.

Szeliski, Tonnesen, and Terzopoulos in [27] invented particle-based models of surfaces. They define the model of a surface as a system of oriented particles behaving according to the rules of Newtonian mechanics. The particles are equipped with the internal forces comprising a long range attraction, a short range repulsion, the consistency in local orientation and principal directions, and the average spacing between the particles. The external forces attract the particles towards data points. In fact, all this is defined as a conservative system, i.e. in terms of some potentials. The surface is then found as a triangulation of the system of particles minimizing overall energy.[8] The triangulation algorithm builds on a heuristic and results in a structure which is a 3-D analogy to 2-D Gabriel graphs.

Apart form dealing with arbitrary, unknown topology, other interesting aspects can be seen in the ability to produce curvature dependent resampling of the surface since the spacing of particles can be made dependent on surface curvature. On the other hand, it is not really clear whether the post-triangulation of particles gives a simplicial approximation of the surface. It is also not clear how growing of a particle system should be properly controlled.

Here, pointing on similar aspects of previous two approaches seems in place. Both approaches define the surface through a zero set of some functions. It is the key ingredient assuring that the resulting set is a valid surface. Hoppe et

[7] They claim that the manifold structure is assured by the implicit function theorem. I do not see, at the moment, if it is true since the signed distance function is not differentiable and defined only in the vicinity of data points.

[8] Actual computation is carried out by the method equivalent to energy minimization with gradient descent.

al. define surfaces a the zero set of

$$
\begin{aligned}
d(\mathbf{p}, \mathbf{X}) &= signum(\mathbf{o}_i)\ min(distance(\mathbf{p}, \pi(\mathbf{o_i}))) \ , \\
\mathbf{o}_i &= \mathbf{o}_i(\mathbf{x}_i, k) \ ,
\end{aligned}
$$

in the vicinity of points $\mathbf{x}_i$, where k specifies the k–neighbourhood, π represents the local planar approximation at the point $\mathbf{x}_i$, and $\mathbf{p}$ stands for a surface point on. Since the signed distance function is not differentiable, its zero set can not be found analytically nor by a gradient method. A brute force (marching cubes) is applied instead. Moreover, in order to determine signum of the distance, the NP-comlete problem is faced.

Szelisky et al. on the other hand, formulates the surface reconstruction as an energy minimization. The overall energy E of the particle system attracted by input data is defined as a sum of smooth potential energies

$$E = \sum E_{ij},$$

hence it is smooth. It allows to use minimization techniques exploiting continuity. The surface is defined as a minimum level of the energy function or as a zero set resulting from $\|\nabla E\| = 0$. It assures the manifold structure by the implicit function theorem if ∇E is nonvanishing and has maximal rank everywhere where $\|\nabla E\| = 0$ [5]. The problem of finding consistent orientation does not pose itself here but, on the other hand, the particles can be trapped in a local minima of E (Is this really the case or, maybe that for certain formulations, there is just one zero set.). Is it better to start "growing" from one point risking to be easily trapped in a local minimum but staying on the manifold or to sprinkle data with particles risking that they will not all be on the same manifold but probably be sure that global minimum is achieved?

Both above approaches introduce a non-parametric, local surface representation what means that it is not so easy to impose constraints on the shape produced. For instance, it is not clear how we could force the reconstruction of certain, specific surface, e.g. torus, when we expect it.

5.4 Construction of complete surfaces from visual surfaces

Let us assume that parts of the surface are already recovered by some method and that the relative transformations between them are more or less known. The task is to exactly register the patches and merge or "zip" them together to form a valid surface. This problem seems to be the most challenging one in the context of autonomous exploration since often some partial knowledge about surface connectivity is known.

Soucy and Larendeau [28] assume to have a set of exactly registered range images. They dynamically build the Venn diagram of overlapping visual surfaces, i.e. the surface is covered by a union of canonic subsets each of which corresponds to a region where different visual surfaces overlap. Two patches overlap if their points are closer than predefined threshold and if the normals at the points are similar. For each canonic subset a *canonical tangent plane* is constructed. The normal of such a plane equals to the average of view directions used to scan particular visual surface. The integration of the patches is performed independently in each canonical subset. Regular orthogonal grid – *canonical grid* – with vertices x_i, y_j is constructed in each canonical plane and a two dimensional function $\bar{z} = f(x_i, y_j)$ is computed as an averaged z value of all k overlapping patches at the point $\bar{z} = \sum_{\forall z_k} z_k(x_i, y_i)$.

The above approach decomposes the problem of surface fusion into two steps. First, the surface is broken into an atlas of local parameterizations. Second, within each parameterization, all visual surfaces are treated as functions and also combined that way.[9]

Decomposition into an atlas relies on the fact that within each canonical subset of the Venn diagram the surface can be expressed as a function over properly selected plane. The orientation of such a plane is here computed by averaging directions which particular patches were measured from and therefore are also visible. Notice, however that the average is a good choice only if all visual surfaces considered are connected. Hence we believe that all visual surfaces covered by a canonical subset of the Venn diagram must be connected although this is not stressed in [28].

Turk and Levoy in [31] proposed a bit more advanced system for object modelling from range images. They assume that rough matching of surfaces is known but precise registration is computed within their algorithm.

Creating of simple model from visual surfaces is split to two main subtasks. Surface *registration* computes rigid transformations of visual surfaces to bring their points in overlapping parts into alignment. To do this, a modification of Besl-McKay's Iterative Closest Point (ICP) algorithm [32] is used. The original ICP algorithm minimizes distances between mutualy closest points on both surfaces.[10] Turk and Levoy minimize only distances between points inside the surface that are shorter than some maximum distance. This extends the ICP algorithm for registration of partially overlapping surfaces.

To register n visual surfaces, one should ideally adopt the strategy, where each surface is registered to all its neighbouring surfaces. As far as we understand Turk's and Levoy's work, this is not done. Rather they register all surfaces with respect to one fixed, so called *anchor*, surface. The anchor surface is usually obtained by measuring the object in cylindrical coordinate system. Object is scanned when rotated so that a large part of its convex hull can be measured. The rest, concavities, the top, and the bottom of the object are then registered against anchor visual surface.

When visual surfaces are properly registered, surface *integration* follows. First, topology of the object is constructed by "zippering" overlapping patches. Zippering removes redundant triangles by "eating" them simultaneously on boundaries of both overlapping surfaces. The resulting nonoverlapping triangulations are connecetd so that boundary triangles are shared. Nonredundant triangulation is assumed to be a proper representation of object's topology. In

[9]Fusion of visual surfaces expressed as functions over their common support has also been tackled by North in [29] and Bove in [30].

[10]Each surface is regularly sampled. For each sample point, its closest point on the other surface is found. It can be proven that this algorithm converges.

the final step, *consensus* geometry is computed by averaging the vertices from all data available in the normal direction at each point of the surface.

6 Thesis proposal

The previous section has shown that reconstruction of shape from examples is a vivid field of Computer Vision. A lot of attention has been paid to various particular problems as range image acquisition, visual surface reconstruction, surface matching , registration and fusion. However, no one yet build a complete integrated system that would allow to put an object on a turn and slant table and after a few minutes or hours of measurement and processing to get a model of its shape automatically.

The goal of this work is to build a *Geometrical Model Reconstruction System* (furthermore only "the system") that scans the object under scrutiny from different viewpoints and unassistedly constructs model of its shape (geometrical model).

The asumptions that we make are the following. It is assumed that a compact object with a closed and piecewise smooth surface is measured by a triangulation based 3-D scanner with unknown but fixed calibration. The scanner provides partial surface measurements in the form of visual surfaces. It is also assumed that the object can be rotated and translated so that all its α–observable parts can be measured.

The requirements put on the system are the following. The model must be general so that surfaces of different gena might be represented uniformly. It should be also global in the sense that it will be possible to parametrize measured surface globally. The representation of the model must be limited in complexity, i.e. adding redundant data will not cause increase of complexity of the model but will improve its presision. Surface fusion must be robust so that one wrong measurement shall not flaw the whole model.

6.1 System for geometrical model reconstruction

We propose the system that on the coarsest level comprises the following three components

- a 3-D scanner,
- a geometrical modeller, and
- a pan and tilt table.

Figure 12 illustrates the performance of the system. An object is placed on a *pan and tilt table*, or the scanner is mounted on robot's arm, that can be *actively controlled* by the computer. The *3-D scanner* measures one visual surface at a time and feeds data into the *geometrical modeller* running on the computer.

The modeller maintains a computer model of the object, updates it when new data are available, and plans where to look next until the model is not finished.

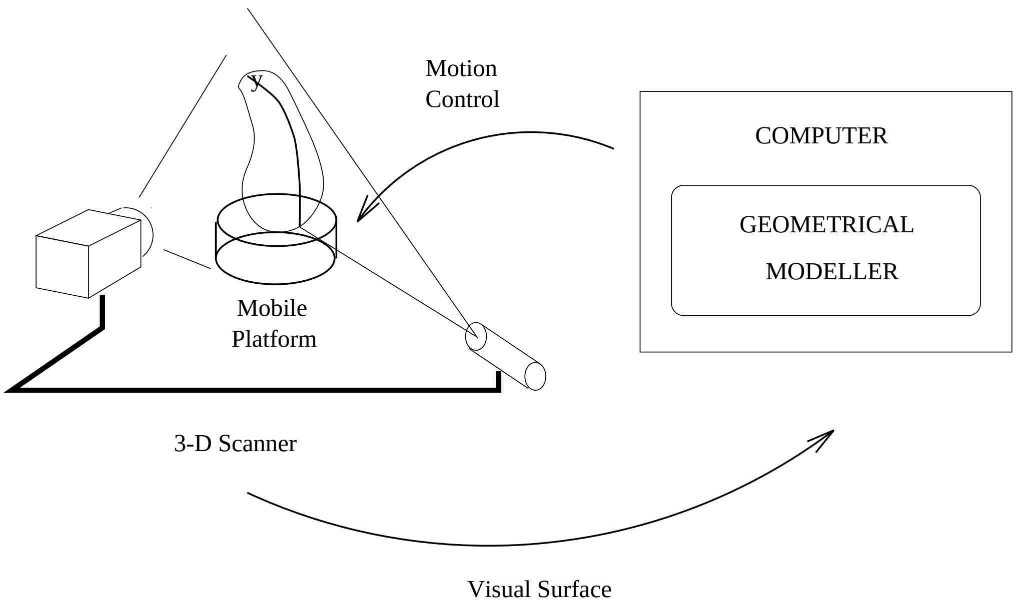

Figure 12: A system for automatic reconstruction of geometrical models.

6.2 Geometrical modeller

Geometrical modeller processes the data comming from the scanner, builds the model, and plans the next position of the sensor. It consists of a the following main modules:

- visual surface reconstructor,
- visual surface decimator,
- surface matcher,
- calibrator,
- model builder, and
- view planner.

Each module is responsible for one task that must be accomplished in order to finish the modelling of an object. The modules share current estimate of the model and an estimate of the calibration of the sensor. These are iteratively updated in the process of reconstruction.

Let us now describe one, let say the i-th, update of the model. We assume that some incomplete geometrical model, N_{i-1} is available and that there is also some estimate of the calibration H_{i-1}. The model building can be described by the following sequence of actions:

1. A 3-D scaner scans the object and computes the visual surface U_i that is stored in the form of a mesh with polygonal faces.

2. Visual surface reconstructor converts the mesh into a triangular mesh by a constrained triangulation. The edges from original mesh are preserved.

3. Visual surface decimator removes redundant triangles from the triangulated mesh. The resulting triangulation is constructed over a subset of original vertices. The goal of this step is to reduce space and time costs in next steps.

4. Surface matcher matches new visual surface U_i to the partial model N_i in two steps. First, matching based on local invariants provides an estimate of the correspondence, which is then used by thr Calibrator to update H_{i-1} to H_i. Then, the initial correspondence is further improved by the use of robust version of the ICP algorithm and updated calibration H_i.

5. Model builder updates the geometrical model.

6. View planner checks whether the model is complete and if not, it plans the position of a new view.

Let us now comment on each particular action performed in the process of model updating. Visual surface reconstruction was disscussed in detail in Section 4. Hence we start with retriangulation of the mesh obtained from the sensor.

6.2.1 Constrained triangulation

Constrained triangulation is included mainly to set a fixed interface between 3-D scanners and the modeller. Different 3-D scanners can provide different meshes. The constrained triangulation can be performed in many ways but we shall prefer constrained Delaunay triangulation which is fast and produces the most flat surface as mentioned in Section 4.3. The triangulation is performed on a planar projection of the mesh, e.g. in the image,. This triangulation preserves edges of the original mesh.

6.2.2 Decimation of visual surfaces

The distribution of sampling density on a visual surface greatly varies at each point of the surface. It depends on the angles between surface normal and rays that project the point into the cameras. Dense sampling is obtained where both angles between the rays and the normal are small. On the other hand, coarser data are obtaing near occlusion boundaries. In any case, the sampling has no relation to the structure of the surface.

We wish to reduce the number of triangles representing the visual surface in areas where its curvature is low. We formulate this task as looking for the best approximation of a triangulated surface by another triangulated surface that is created over a subset of vertices of the original mesh. For instance, we might look for the closest triangulated surface with maximally n triangles, or we might want to simultaneously minimize n and a residual error to get a consensus between the precision and space costs usin Minimun Desciption Length principle [33].

Let us mention that certain data reduction is necessary. Otherwise a few surfaces would overflow memory of any computer.

6.2.3 Surface matching and registration

It is assumed that two partially overlapping surfaces U_α and U_β related by a global transformation are available. In our case, all thought transfromation are subgroups of a projective group in $\mathcal{P}^3$.

Surface matching aims at finding the pointwise correspondences $\Psi_{\alpha\beta}$, Figure 13, betwen partially overlapping surfaces U_α U_β. Matching does not impose any constraints on $\Psi_{\alpha\beta}$.

Surface registration, in contrary, looks for the best transformation T that overlays U_α and U_β. In other words T is found by minimization of

$$e = min_T \; \rho(U_\alpha, T(U_\beta)),$$

where ρ is a function evaluating the quality of the overlap. In Euclidean geometry it might be a distance between the points on a surface. In equi-affine volume between the surfaces will do the work. In our work we will to register surfaces only in Euclidean geometry where the definition of an invariant ρ is easy. Moreover we will assume that initial estimate of T is available through the use of matching or that it is known a priory.

The iterative closest point algorithm (ICP) developed by Besl and McKay in [32] solves registration provided that a good initial estimate of T is available. We have extended this algorithm to register partially overlapping curves [34]. The curves are represented as a strings of points. The algorithm is the following:

1. Find the initial transformation by applying a curve matching algorithm.
2. Refine the correspondence between the curves by finding pairs of closet points on curves.
3. Compute the optimal Euclidean motion between the corresponding points in the Least squares sense.
4. Terminate if the mean square distance of the corresponding points drops below a preset threshold or if the maximal number of iterations is exceeded, otherwise go to 2.

Matching is in general very difficult task. It emerges if T is completely unknown. This can happen if either we do not know how the sensor moves or if its calibration is completely unknown.

Matching task is easier if T is a member of some parametric group with small number of parameters like Euclidean group in $\mathbb{R}^3$ which has 6 independent parameters. The invariants to that group can be computed at each point of the surfaces. Since invarints does not change, they have to be same for both

U_α and U_β at the corresponding points. Good invariants for 3-D curves are, for instance, curvature and torsion. For surfaces, principal curvatures will probably suffice if the surface has rich differential structure. All points with the same values of invariants on different surfaces can be taken as a sets of possible matches. These have to be verified so that a one-to-one relation emerges. See for instance [34, 35] for an algorithm that matches the curves related by an unknown Euclidean transformation.

Since the surface matching is in its full generality very difficult, if not impossible, we admit to replace a posteriori estimation of the transformations by a priory controlling the motion of an object and/or the scanner. However, the automatic matching and registration of partially overlapping surfaces remains one of the key research topics within this work.

6.2.4 Model representation and construction

As we have shown *visual surfaces* can be reconstructed from the points scanned in particular views and stored as triangulated surface patches. We assume that the geometries of visual surfaces can differ by the transformation T given in Equation (10). Each visual surface is homeomorphic to a plane and thus can have its own local coordinate system.

Figure 13 shows the process of visual surface acquisition and model construction in terms of manifold definitions. Real object N is scaned by a 3-D scanner thus providing visual surfaces U_α and U_β. Maps χ_α and χ_β are completely known if the sensor is calibrated or they are only known up to some projective mapping H if the calibration is unknown. However, they are known in pointwise manner, i.e. we know a one-to-one correspondence between smapled points on N and their projections to V_α and V_β respectively. The map $\Psi_{\alpha\beta}$ is obtained by surface matching and registration. $\varphi_{\alpha\beta}$ is then obtained by the composition.

We wish to construct a parametric a model of N and represent it as a support domain $\tilde{N}$ and a map ϕ. We think that $\tilde{N}$ might be well represented as a C^0 continuous manifold. In computer it would be realized as a triangular mesh. ϕ then might be from the polynomial family. The simplest possible choice is planar triangulation in which case ϕ can be uniquely defined just by storing the coordinates of measured points. The question is, how to construct ξ_α. It is also not clear how to implement building of this model. We feel that model construction could be a two-step process. In the first step, the modeller would only try to cover the whole object with visual surfaces and to infer the topology of the manifold. The model for this task need not be in parametric form. Once topology was known, one could construct $\tilde{N}$ in some canonic form, i.e. as a shere with n handles. Again it is not clear how ξ_α would be constructed.

Alternatively, one could represent $\tilde{N}$ implicitly by a function F sampled on 3-D grid. Grid could be irregular or it even need not be a grid but, for instance, a Delaunay tetrahedralization of some space. For instance it would suffice to thetraheralize the convex hull of all surface points after surface registration. The trahegralization would be, of course, constrained so that the faces of all visual surfaces would be preserved. On such a tetrahedralization one could then define function F (how?) and use its zero level set as a model of N.

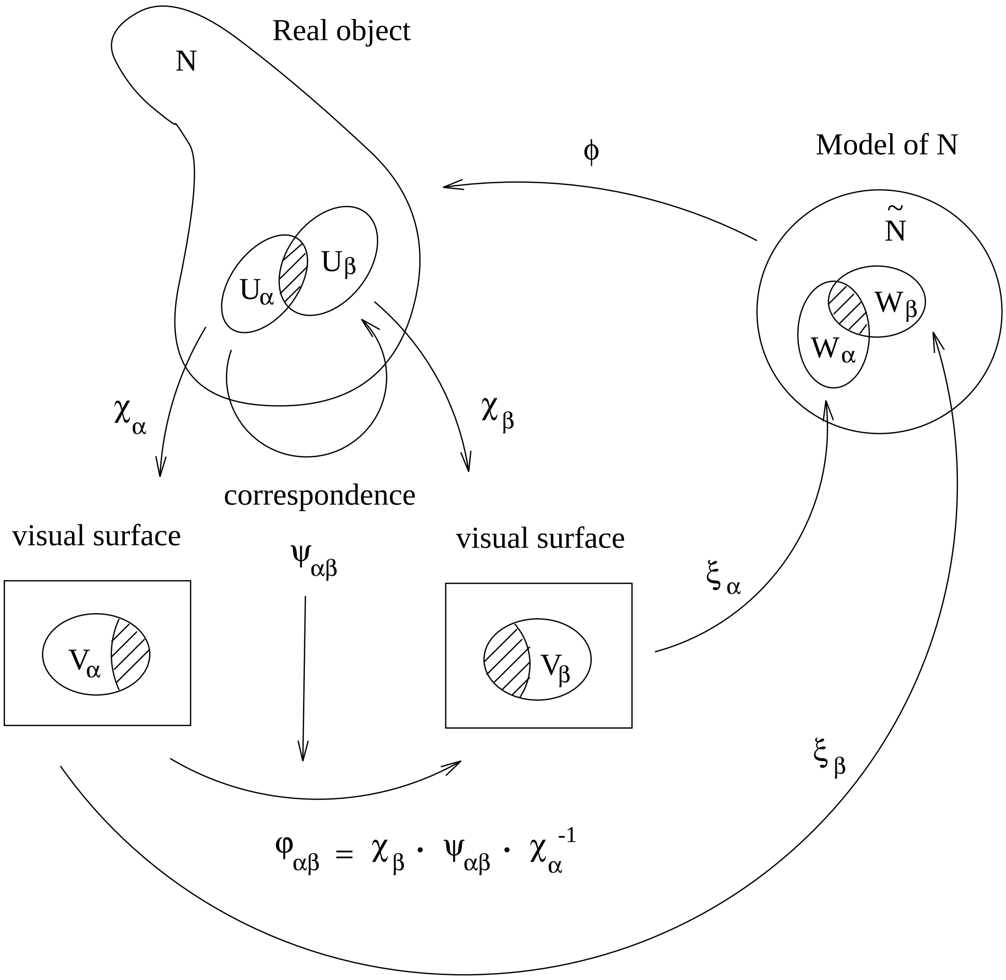

Figure 13: Model construction by defining manifolds and maps between them.

It is clear that the choice of the manifold representation is key issue in the modeller and determines almost completely the way how visual surfaces will be integrated.

6.2.5 Measurement planning

Views have to be planned automatically so that the system arrives at a complete reconstruction in some finite time or finds that the object under scrutiny is not observable. Then, it should terminate as soon as observable portion of the surface is measured with desired precision. The autonomous exploration should be driven by the lack of data and the level of uncertainty in the model. The level of data support and its uncertainty can be evaluated locally on the surface recovered and the object and/or the scanner will positioned so that maximum information can be obtained from each view.

Decision where to look next or when to stop model construction could also use additional information about the sensor, constraints imposed on its placement, and so on. Such a knowledge is not parametrical rather it can be efficiently represented as rules. We can forseen some groups of rules that certainly should be considered like:

- general knowledege like *surfaces are smooth* and α*–observable*,
- true values or estimates of parameters like α,
- constraints on sensor placement,
- visual surfaces should partially but not completely overlap,
- termination criteria,
- quality evaluation criteria,

and many others. The representation of this information as well as the inference of the best position of the sensor must be intensively studied.

7 Conclusion

Automatic modelling from examples is a difficult task. We have shown that there are already reasonable sensors providing dense measurements from surfaces. We have analyzed geometry of triangulation based sensors and shown how both geometry and topology can be reconstructed. We have also discussed the point reconstruction in the case when the calibration of the sensor is unknown and found that Euclidean reconstruction can be obtained if overlapping surfaces measured from different viewpoints can be matched. The notion of object and its shape was specified and their mathematical model, manifold, was mentioned. The analysis of related literature shows that there is currently no complete integrated system that would deliver geometrical models from the examples but there are is a lot of interest to build one.

The problem of closed surface reconstruction is clearly stated and a possible solution is proposed on a general level. Even though we already have experiences in particular tasks like visual surface reconstruction [14, 13], curve matching [34, 35], calibration of the camera systems [36], and a view selection [37], lot of work have to be still done. In particular, right representation of geometrical model must be chosen and reconstruction planning have to be thoroughly studied.

References

[1] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, 1993.

[2] Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *European Conference on Computer Vision*, pages 563–578, 1992.

[3] Olivier D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–333, 1992.

[4] Jan J. Koenderink. *Solid Shape.* Massachusetts Institute of Technology, 1990.

[5] Peter J. Olver. *Applications of Lie Groups to Differential Equations.* Springer Verlag, New York, 1986.

[6] S. Osher and S. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton–Jacobi formalism. *Journal of Computational Physics*, 79:12–49, 1988.

[7] R.C. Veltkamp. 3-D computational morphology. In R.J. Hubbold and R. Juan, editors, *EUROGRAPHICS '93*, volume 12, pages C–115–C–127, 1993.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[9] R. Jarvis. Range sensing for computer vision. In A.K. Jain and P.J. Flyn, editors, *Three Dimensional Object Recognition Systems*, volume 1 of *Advances in Image Communication*, pages 17–56. Elsvier Science Publishers B.V., 1993.

[10] M. Maruyama and S. Abe. Range sensing by projecting multiple slits with random cuts. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):647–651, June 1993.

[11] M. Rioux. Laser range finder based on synchronzed scanners. *Applied Optics*, 23(21):3837–3844, November 1984.

[12] P. Vuylsteke and A. Oosterlinck. Range image acquisition with single binary-encoded light pattern. *IEEE Pattern Analysis and Machine Intelligence*, 12(2):143–164, 1990.

[13] Tomáš Pajdla. Laser plane range finder - the implementation at the cvl. Technical Report Nr. K335-95-98, Czech Technical University, Prague, October 1995.

[14] Tomáš Pajdla. BCRF - Binary Illumination Coded Range Finder: Reimplementation. ESAT MI2 Technical Report Nr. KUL/ESAT/MI2/9502, Katholieke Universiteit Leuven, Belgium, April 1995.

[15] O. Faugeras and B. Mourrain. On geometry and algebra of hte point and line correspondences between n images. In E. Grimson, editor, *ICCV '95*, volume 1, pages C–115–C–127, 1995.

[16] Abdrew Zisserman, Paul A. Beardsley, and Reid. Ian D. Metric calibratior of a stereo rig. In *IEEE Compure Society Workshop: Representation of Visual Scenes*, MIT Cambridge, MA, June 1995.

[17] B.K. Choi, H.Y. Shin, Y.I. Yoon, and J.W. Lee. Triangulation of scattered data in 3-D space. *Computer Aided Design*, 20(5):239–248, June 1988.

[18] J.D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transaction on Graphics*, 3(4):266–286, October 1984.

[19] S.D. Rippa. Minimal roughness property of the Dealunay triangulation. *Computer Aided Geometrical Design*, 7:489–497, 1990.

[20] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, July 1988.

[21] R.L. Stevenson and E.J. Delp. Viewpoint invariant recovery of visual surfaces from sparse data. *IEEE Pattern Analysis and Machine Intelligence*, 14(9):897–909, September 1992.

[22] Andrew Blake and Andrew Zisserman. *Visual Reconstruction.* The MIT Press, 1987.

[23] S. Sinha and B.G. Schunck. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Pattern Analysis and Machine Intelligence*, 14(1):37–55, January 1992.

[24] R. Szeliski. Estimating motion from sparse range data without correspondence. *IEEE-CH266-1/88*, pages 207–216, 1988.

[25] H. Ferguson, A. Rockwood, and J. Cox. Topological design of sculptured surfaces. *Computer Graphics*, 26(2):149–156, July 1992.

[26] P.A. Firby and C.F. Gardiner. *Surface Topology.* John Wiley & Sons, New York, 1982.

[27] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Curvature and continuity control in particle-based surface model. In *SPIE: Geometric Methods in Computer Vision*, San Diego, CA, July 1993.

[28] Soucy M. and Laurendeau D. Surface modeling from dynamic integration of multiple range views. In *Proceedings of the 11th International Conference on Pattern Recognition*, volume I, pages 449–452, The Hague, The Netherlands, September 1992. IEEE Computer Society Press, Volume I.

[29] Peter R.J. North. Optimal surface fusion. In *Proceedings of British Machine Vision Conference*, September 1991.

[30] Jr. Bove, V. Michael. Probabilistic method for integrating multiple sources of range data. *Journal of Optical Society of America*, 7(12), December 1990.

[31] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proc. SIGGRAPH'94, Annual Conference Series In Computer Graphics*, pages 311–318, Orlando, Florida, 1994.

[32] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[33] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Co., Series in Computer Science, IBM Almaden Research Center, San Jose, California U.S.A, 1989.

[34] Tomáš Pajdla and Luc Van Gool. Matching of 3-D curves using semi-differential invariants. In *5th International Conference on Computer Vision*, Cambridge, MA, June 1995. (to appear).

[35] Tomáš Pajdla and Luc Van Gool. Efficient matching of space curves. In *6-th International Conference on Computer Analysis of Images and Patterns*, pages 25–32, Prague, Czech Republic, September 1995. Czech Pattern Recognition Society, IAPR, Lecture note in Computer Science, Springer Verlag.

[36] Tomáš Svoboda and Tomáš Pajdla. Efficient motion analysis. Technical Report Nr. K335-95-95, Czech Technical University, Prague, October 1995.

[37] Václav Hlaváč, Aleš. Leonardis, and Tomáš Werner. Automatic selection of reference views for image-based scene representations. In *6-th European Conference on Computer Vision*, April 1996. submitted.