Reconstruction of piecewise planar objects from point clouds

Martin Peternell* and Tibor Steiner†

December 2, 2002

Abstract

This article discusses the reverse engineering problem of reconstructing objects with planar faces. We will present the main geometric features of a modeling system which are the detection of planar faces and the generation of a cad model. The algorithms are applied to the problem of reconstruction of buildings from airborne laser scanner data.

Keywords:
reverse engineering, cad, computational geometry, planar faces, building reconstruction, space of planes, geometric constraints.

martin@geometrie.tuwien.ac.at, tibor.steiner@acv.ac.at

1 Introduction and Assumptions

Our main motivation to study the reconstruction of piecewise planar objects comes from the problem of *building reconstruction* while working on an industrial project. There is quite a lot of literature available for this topic, see for instance [2, 3, 4, 5, 13, 14]. More general concepts for reconstruction of geometric objects are studied in the field of *reverse engineering* of

*Institute of Geometry, Univ. of Technology Vienna, Wiedner Hauptstraße 8–10, A–1040 Wien, Austria

†Advanced Computer Vision GmbH–ACV Donau-City-Straße 1, A–1220 Wien, Austria

technical objects. We want to point to some literature, see [11, 12] and the references therein. The method described in the following is mainly designed for reconstructing buildings from laser scanner data.

The reconstruction of buildings uses very special methods on the one hand, but on the other hand there are some methods useful for the reconstruction of all objects which mainly or exclusively possess planar faces. So, we will at first describe general methods common to reconstruction of all objects with planar faces, and at second pay attention to special features of building reconstruction. Especially, the *segmentation* of an object or *detection of planar regions* is very dependent on the density of the data points.

We want to discuss the reconstruction of objects from measured point clouds. It is assumed that the points p are given by Cartesian coordinates $p = (p_1, p_2, p_3)$. The point cloud is often generated by an optical measurement device, e.g. a laser scanner. In our case, the laser scanner is positioned in an airplane and we simply say that the objects are scanned from above. Thus, our objective is different from that of street scenery reconstruction. In the latter case, buildings are scanned from the side. Approximately we are given five to ten data points per square meter. We want to point to the special literature dedicated to the measurement process, see e.g. [10].

If an object is scanned just from one side, there will be visible parts and not visible ones. In the following we assume that the objects under consideration are scanned from one viewpoint and we have the problem that not all object points are registered. Scanning buildings shows that vertical walls or very steep roof faces are not described by data points. If the used coordinate system is chosen such that the z-axis points upwards, then the data points can be considered as measurements of a function over the xy-plane $z = 0$. Unfortunately, this function is not continuous everywhere because vertical walls and height jumps on the roof cause discontinuities of the considered function. On the other hand, the function is piecewise continuous.

Architecture clearly uses more surface types than planar patches. But since many roofs mainly consist of planar faces, the reconstruction presented here uses the *hypothesis* that *the objects to be reconstructed are piecewise planar* and the roof scene can be described by a piecewise linear function over the xy-plane. We also assume that building ground plans are available. Thus, building reconstruction of a town district can be done building by building.

The article is organized in the following way: Section 2 discusses the segmentation of the object or, in other words, the detection of planar regions. In Section 3 the computation of regression planes with respect to some geometric constraints, like common slope and orthogonality is treated. The last section tells about the generation of a cad model, beginning with the definition of adjacencies between planar faces and computation of their intersections. An algorithm for generating building models is discussed. Finally a summary of the described method is given.

2 Segmentation or Detection of Planar Faces

A commonly used segmentation technique is *region growing.* One starts with a seed region and determines the parameters of the geometric object modeling the data of the seed region. The seed region is increased as long as the parameters of the model do not vary too much. If a region is segmented correctly, a new seed region is chosen as long as there are parts of the object to be modeled. This technique requires a very high density of the data points with respect to the features of the object. When scanning machine parts from near positions it is often useful to apply a data reduction to be still efficient in the segmentation process.

Another segmentation technique is called *direct segmentation* and is described in [1]. It is similar to our method, but is designed to work for more general objects to be reconstructed. The *surface normal vector* and *local quadric of regression* are estimated for each data point. This information is used to check for locally planar or spherical regions first; if this is not the case, more complex geometric surfaces are tested. Compared to the variety of surfaces occurring at machine parts, our problem is simpler, since we look for planar regions only. The difficulty of the present task lies in the low resolution of the data, the high variation of the size of the faces and features and the presence of all kinds of errors. This results in problems of finding all necessary roof faces and of forming the correct topology and adjacencies of faces.

The data we work with are very different since they are registered by a very far measurement device. The point density is rather low and does not describe all features which are present on a buildings roof. Small roof faces are not well represented by the data and cannot be detected with required accuracy. In addition, chimneys, dormer windows etc. are basically treated as outliers in the segmentation process.

A method similar to ours is described in [13]. They use a spatial extension of the *Hough transform* to determine roof faces, which is a special duality. In addition, we introduce a metric in the dual space (which is later on denoted by A^*), the parameter space of the Hough transform, which should lead to improved results determining planar faces.

We are given a point cloud P describing an object composed of planar patches. We assume, as said in Section 1 that the data points describe a function over the xy-plane $z = 0$, which needs not be continuous everywhere. The allowed discontinuities are discussed later. Usually the point cloud P contains scattered data points. For some measurement devices one obtains data points which are organized in stripes. This means that we have data points according to curves on the object which are in our case always straight line segments. Since it is not of advantage for our task we do not pay attention to this fact. Often, the data points are resampled in a way that data points over a regular grid in the xy-plane are given. By resampling the data one looses accuracy but on the other hand it is a great advantage, since the computation of the neighborhood of data points on a regular grid is very simple. Details to this fact are discussed below.

The first task is to possibly find all planar regions of the object. Since we assume that the object is scanned from above, which means against the positive z-axis of the coordinate system, we can only detect planar faces whose slope against the xy-plane is not too large. Planar faces with slope larger than 75 degree against xy are clearly not well represented by the data points. In the case the data acquisition is scanning from an airplane, the angle between the laser beams and the z-axis is usually very small (< 7 degrees).

Given a set P of data points $p_i, i = 1, \ldots, N$, we want to find planar regions. For that we compute a *local plane of regression* ε_i to each data point p_i. This leads to *local estimates* for the planar faces. Further, we consider the set of local regression planes ε_i as points e_i^* in dual space A^*. All points q_k which belong to one planar face of the object, should have local regression planes ε_i which are close to each other. Thus, finding all planar faces can partially be interpreted as computing *point clusters* in the set of points e_i^* in A^*. Figure 1 shows data points of a building with underlying grid and the image points e_i^* of the local planes of regression in A^*. Now we will discuss the necessary steps for detecting planar faces in detail.

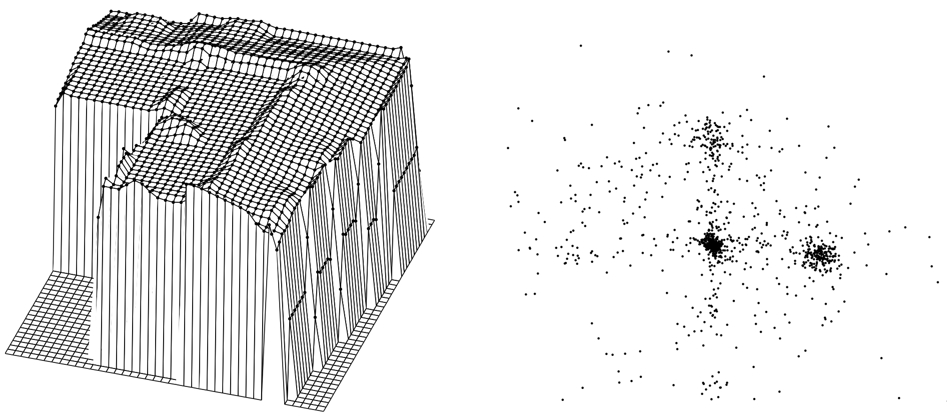
Figure 1: Left: Data points of a building(with underlying grid). Right: Point clusters in A^*

2.1 Local planes of regression

As mentioned above we compute local regression planes ε_i to all data points p_i. For that we choose data points q_{ik} in an appropriate neighborhood U_i of p_i. If p_i are grid data, we can choose q_{ik} to be the eight grid neighbors, for instance. If p_i are scattered data, we can choose U_i as circular neighborhood of the projection of p_i onto the xy-plane. The points q_{ik} are those data points of P, whose projections are in U_i. Sometimes it is an advantage to triangulate the set of data points P. Then, the neighborhood U_i is determined by the triangulation.

If we assume that the slope of the faces is not very large, a local regression plane ε can be considered to be the graph of a linear function

$$\varepsilon : z = f(x, y) = ax + by + c. \tag{1}$$

If we assume that the errors of the measurements are normally distributed, the least squares solution for ε is the right choice. So, ε is computed by minimizing

$$\sum_k (f(x_k, y_k) - z_k)^2 \to \min,$$

where (x_k, y_k, z_k) are the coordinates of the data points q_{ik}.

It is well known that if large errors are present, the minimization in the least squares sense will not lead to good local estimates. To improve this, one might first search for large errors and after having removed them from the data, compute a least squares solution. A cheap possibility for searching for large errors of the data is the minimization of the l_1 norm of the errors,

$$\sum_k \|f(x_k, y_k) - z_k\| \to \min.$$

This minimization can be formulated as a linear programming problem and is equivalent to

$$\begin{aligned} \sum_k \quad & \delta_k \to \min \\ & \text{subject to} : -\delta_k \leq f(x_k, y_k) - z_k \leq \delta_k,\ 0 \leq \delta_k. \end{aligned} \tag{2}$$

Data points corresponding to large δ_k are removed from the data set of the neighborhood U_i. After this correction, the local estimate ε is computed in the l_2-sense with respect to the reduced data set, since basically we assume normally distributed errors. Alternatively, ε can be calculated via minimization of the l_1-norm or another robust estimator.

The objects under consideration possess edges or even height jumps between the faces. Thus, the neighborhood of data points near edges or height jumps often contain data points lying on other planar faces. Those have to be treated as outliers and should be detected in the computation of the local regression plane. If the data point cloud represents a building, data points on chimneys, antennas or other things positioned at the roof faces can be considered as large errors since they are typically much too small to be represented by the data and to be reconstructed.

Finally we end up with a set of local regression planes

$$\varepsilon_i : z = a_i x + b_i y + c_i.$$

Interpreting (a_i, b_i, c_i) as affine coordinates of points e_i^* in a three dimensional space A^* we obtain a point model of all planes which can be written as graphs over the xy-plane. It is obvious that vertical planes $ax + by + c = 0$ cannot be represented as points in A^*. By the way, vertical planes correspond to points at infinity in the projective extension of A^*.

2.2 Finding Planar Faces

We mentioned earlier that data points p_k lying in a planar face of the object, possess local planes of regression ε_k which are close to one other. It is necessary to specify what *close* shall mean in this context. Since A^* is only an affine space till now, this is not well defined yet. It is necessary to introduce a metric in A^*, which measures the distance between two points e_1^*, e_2^* or two planes $\varepsilon_1, \varepsilon_2$.

To achieve this, we remember that the planes under consideration are graphs of linear functions over the xy-plane. What we are actually interested in are the distances between corresponding points of two planes $\varepsilon_1, \varepsilon_2$. The simplest but sufficient way to define a correspondence between two planes

$$\varepsilon_1 : z = a_1x + b_1y + c_1 \text{ and } \varepsilon_2 : z = a_2x + b_2y + c_2$$

is the use of z-parallel lines, see figure 2. The orthogonal projection of the objects to be reconstructed is a domain of interest D in the xy-plane. Therefore, the squared distance between two planes, containing faces of the object, can be defined in the following way:

$$d(\varepsilon_1, \varepsilon_2)^2 = \frac{1}{\text{area(D)}} \int_D ((a_2 - a_1)x + (b_2 - b_1)y + (c_2 - c_1))^2 dxdy.$$

The squared distance function is a positive definite quadratic form in the coordinates a_i, b_i, c_i. Thus, the corresponding metric $d(\varepsilon_1, \varepsilon_2)$ is a Euclidean metric and A^*, equipped with this metric, becomes a Euclidean space. The scalar product matrix is not the unity matrix as it is for the canonical Euclidean metric, but the matrix notation of the squared distance function reads as

$$d_D(\varepsilon_1, \varepsilon_2)^2 = (c_2 - c_1, a_2 - a_1, b_2 - b_1) \cdot \begin{pmatrix} \int 1 & \int x & \int y \\ \int x & \int x^2 & \int xy \\ \int y & \int xy & \int y^2 \end{pmatrix} \cdot \begin{pmatrix} c_2 - c_1 \\ a_2 - a_1 \\ b_2 - b_1 \end{pmatrix}. \tag{3}$$

As shown in [8] it is possible to apply a coordinate transformation in A^* such that the scalar product matrix becomes the unity matrix. This has some advantages, in particular for the computation of distances between a large number of data points. The transformation can be determined by computing the eigenvalues and eigenvectors of the scalar product matrix from (3).

Remark: The introduced metric depends on the chosen *coordinate system.* In case of building reconstruction from airborne laser scanner data the z-axis of the coordinate system coincides approximately with the direction of the laser beams.

In general it might be necessary to use *different coordinate systems* in order to fully cover the space of planes appropriately. This results in different local mappings of the space of planes to affine 3-spaces A^* and in different Euclidean metrics.

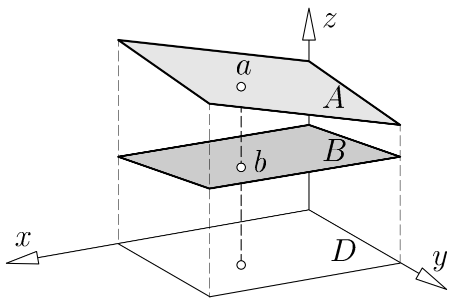

Figure 2: Definition of the correspondence and distance between two planes.

The method which is applied to find the point clusters in A^* essentially depends on the number and the density of data points representing the object. For the task of finding roof faces of buildings and for the nowadays present density of the data points (1-8 points per square meter) we basically apply the following method:

We look for spherical neighborhoods U_i^* (with respect to the metric defined by 3) of image points e_i^* in A^* which contain sufficiently many points e_j^*. The number of points corresponds to the size of the roof faces to be reconstructed. Typically, planar regions with less than 10 data points are not of interest, moreover mostly not well represented by the data points. The radius r of the spherical neighborhood U_i^* corresponds to the estimated standard deviation σ of the z-coordinates of the measurement points p_i, which depends on the measurement device. Usual accuracies of the z-coordinates of airborne laser scanner data p_i are 0.1-0.15 cm.

A useful choice is $r = \sigma$. In addition we want to mention that the variation of points e_i^* is rather large, since the local estimates for the roof planes are very inexact in points of the boundary of the building. The right subfigure of figure 1 shows only a subset of the points e_i^* obtained by calculating all regression planes. We have to zoom into the total set of points e_i^*, to make the clusters visible.

Each cluster in A^* corresponds to a plane ε containing one or possibly more faces of the object. If the object to be reconstructed is convex, the segmentation is nearly complete. We just have to find all data points p_k, lying in the plane ε. These can be slightly more data points than those corresponding to the local regression planes forming the cluster. Finally, the plane carrying the face is the plane of regression to the data points p_k.

If the object B is not convex and this is the case for many buildings, especially in urban areas, the plane ε corresponding to a cluster will contain not only the data points of a face of B but will intersect B in other faces and contain data not lying in the considered face. It is even possible that there exist planes containing more than one face of B. Thus, the data points of a planar face have to be a connected set of points within the plane ε defined by the cluster. According to this fact, a planar region is defined to be the *largest connected component* of the data points p_k which lie close to ε.

To define a useful topology in the set of scattered data points P one uses for instance a triangulation of the points p_i. If p_i are grid data the neighborhoods of points are already defined by the underlying grid.

If we have processed this for all clusters in A^* we obtain a segmentation of the object B. The segmentation consists of planar regions $R_1, \ldots, R_m$ and usually one has some data points left, which do not lie in any of the computed regions. If those are just few, we may interpret them as errors; if many points are not covered by planar regions it might be a contradiction to the hypothesis that B is piecewise planar.

We summarize the algorithm to find planar regions:

1. Compute local planes of regression (local planar fits) ε_i for all data points p_i.

2. Determine the plane ε (point e^* in A^*) which possesses a maximal number of neighbors (planes) in a spherical neighborhood in A^*. This set of neighboring planes defines the maximal cluster of local regression planes.

3. Compute a plane of regression φ with respect to all data points p_k determined by the maximal cluster (defined by local planes of regression ε_k).

4. The planar region is the maximal connected component of all data points lying close to the plane φ.

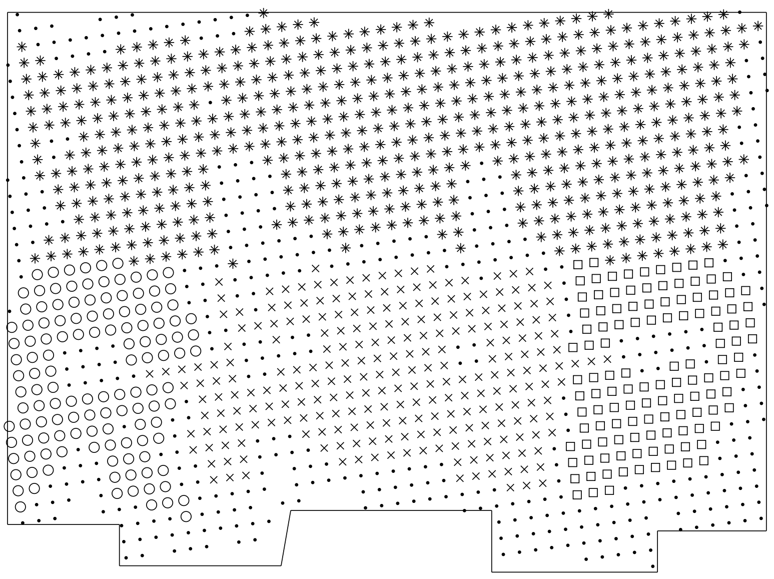

Figure 3: Projection of the segmentation of an object: Data points belonging to regions are displayed as x, stars, circles and squares; remaining data points are displayed as simple dots. The solid lines represent the ground plan of the building to be reconstructed.

By repeating this procedure we will find all planes containing the main roof faces sequentially. Small faces are ignored.

2.3 Accuracy of Finding Planar Faces

In general, after removing outliers, we may assume normally distributed errors of the measurements, in particular the z-coordinates of the data points $p_i = (x_i, y_i, z_i)$, $i = 1, \dots, m$. Further we assume that the measurements z_i are uncorrelated which could be questionable sometimes. For simplicity, x_i, y_i are assumed to be exact. The variation of the data is computed as follows (if we assume that the data are normally distributed): Let $A \cdot u = f$ the matrix notation of the regression problem with A as $m \times 3$ coefficient matrix, f as right hand side and $u = (a, b, c)$ as vector of unknowns. Let $v = A \cdot u - f$, where u is the already computed least squares solution. The variation of the data is

$$\sigma_0^2 = \frac{1}{m-3} v^t \cdot v.$$

It is the squared mean vertical distance of the data points p_i from the plane of regression $z = ax + by + c$.

To determine the accuracy of the coefficients a, b, c one might proceed as follows: The singular value decomposition of A is

$$A = UWV^t$$

where V is a 3×3 orthogonal matrix, W is a 3×3 diagonal matrix, containing the singular values $w_k, k = 1, 2, 3$ and U is $m \times 3$ with orthogonal column vectors. The least squares solution is obtained by $u = VW^{-1}U^t f$. Since the measurements f possess equal variance and assuming that they are uncorrelated, $E(f \cdot f^t) = I$ holds. Taking additionally $U^tU = I$ into account, the covariance matrix of the solution $u = (a, b, c)$ is

$$\operatorname{cov}(u) = E(u \cdot u^t) = V \cdot W^{-2}V^t.$$

The standard deviations of the coordinates a, b, c are the square roots of the diagonal elements of $\operatorname{cov}(u)$. From this one can derive the accuracy of the planar faces.

3 Computation of Regression Planes under Geometric Constraints

Many human made objects possess certain regularities like symmetries, constant angles between faces, constant slope of the faces with respect to a reference direction, orthogonality or parallelity of face normals. In particular at machine parts several of these properties may occur. We want to point to relatively new articles dealing with this topic, see [6, 7]. Segmentation under geometric constraints is sometimes called *model beautification.*

In case of buildings *common slope of roof faces* with respect to a reference direction and *parallelity* or *orthogonality* of the projections of the face normals onto the xy-plane occur quite often. Thus it is desirable to respect these properties in the modeling procedure if possible. Since many buildings are oriented with respect to two orthogonal main directions we start with the determination of these directions. We do not want to go into details here but sketch the most important features with respect to building reconstruction.

3.1 Main directions of a building

Let R_j be the detected planar regions of the roof and let n_j be the projections of the normals of these regions. Let T be the given ground plan polygon. Its segments are denoted by t_i. First we determine two orthogonal directions e, f such that as many segments t_i and normals n_j are nearly parallel or orthogonal to e and f. If all segments t_i and normals n_j are nearly parallel or orthogonal to each other, it is possible to compute e, f as solutions of a least squares adjustment. But in general this property will not hold, and our experience tells us that a solution in the l_2-sense is too sensitive with respect to outliers. Thus, we determine the vectors that form angles larger than a threshold $(5-10°)$ with all others. These vectors are classified as outliers with respect to the main directions. If the remaining vectors can be partitioned into two sets and these sets contain sufficiently many vectors (compared to the input vectors) we compute e, f as solutions of a minimization in the l_2-sense.

3.2 Regression Planes with Respect to Main Directions

Let e, f be the main directions of the object. Consider a planar region R whose projection n of the normal is nearly parallel to e or f. The plane of regression ε carrying the face R should be computed such that the projection of its normal is parallel to e or f. This results in a regression problem with only two free parameters left. We set

$$\varepsilon : z = \gamma\alpha x + \gamma\beta y + c,$$

where (α, β) are the coordinates of either e or f, depending on if n is nearly parallel to e or f. Thus, γ and c are the free parameters to be determined.

It is even possible to compute planes of regression for more than one planar region simultaneously, under the assumption that the projections of their normals are parallel to e or f.

We do not want to go into details here but mention that it is also possible to compute planes of regression for regions whose slope is nearly equal in a similar way. All planes can be computed simultaneously whose regions possess similar slopes.

Remark: In general, satisfying geometric constraints in an optimization leads to complicated non linear systems of equations. We have omitted this by computing the solution step by step. At first we compute estimates of the planes carrying the planar regions. At second we determine the main directions and finally we calculate the regression planes under the mentioned geometric constraints, but by the use of the previous steps. This strategy seems to be helpful since the computation is simple and each step results in a linear problem.

4 Main Steps of Modeling Piecewise Planar Objects

Now we want to describe the main features and procedures of a modeling system which is able to generate cad-models from measured scattered data points. We assume that the segmentation of the point cloud is performed as described in Section 2. Since the presented examples all show models of buildings, we focus on the reconstruction of these objects. However, most of the methods work similarly for arbitrary piecewise planar objects which can be considered as graphs over a planar domain.

When we talk about cad-models of piecewise planar objects, we consider a boundary representation of the object (a representation of the object in terms of faces, edges and vertices). The equations of the faces given, it is our goal to compute the edges and vertices of each face. The edges are clearly the intersection segments between adjacent faces and the vertices are the intersection points of three or more adjacent faces.

Given a set P of data points p_i, the above mentioned segmentation results in planar regions $R_1, \ldots, R_m$. All these regions contain sufficiently many data points. Small regions (with respect to the resolution of the data) are ignored since they are not well represented by the data. In particular we are only interested in the main geometric features of the object because small and possibly not exactly determined regions are more cumbersome than helpful for the reconstruction process. There may be data points left, which are not covered by any region. For modeling the object, it is necessary to determine the adjacency relations between the planar regions R_j. Since the object can possess height discontinuities, it is possible that we have to introduce height jumps between some regions. The basic steps of the implemented modeling method are the following:

1. Determine adjacencies between planar regions.

2. Calculate the intersection segments between intersecting adjacent regions. Find the height jumps of non-intersecting regions R_j, R_k whose projections R'_j, R'_k are adjacent.

3. Determine the adjacencies of planar regions and walls, which are determined by the given ground plans.

4. Calculate the intersection segments between planar regions and adjacent walls.

5. Compute closed polygons for all roof faces.

6. Generate a cad model.

In the following sections we will describe these steps in more detail.

4.1 Determine Adjacencies between Planar Regions

Since the data acquisition is a scanning process with direction mainly against the z-axis, it is nearly no loss of information and thus sufficient to study the projections R'_j of the regions R_j onto the xy-plane for defining adjacency relations between the regions R_j. There are several possibilities to define adjacencies between planar regions R'_j. If the data points p_i possess an underlying grid, we can use this grid for the definition of adjacencies. If we work with scattered data, we can use a (planar) triangulation of the projections of the data points (TIN - triangular irregular network). Whatever, it is basically simple to determine neighboring regions. We increase the regions R'_j forming outer offset regions $\bar{R}'_j$. The offset distance depends on the resolution of the data. The intersection of two regions $\bar{R}'_j, \bar{R}'_k$ shall be denoted by S'_{jk}. We call it simply *intersection region.* In case that the data possess an underlying grid, the offset operations as well as filling of possibly occurring holes in the regions R'_j can be done with mathematical morphological operations, see e.g. [9].

Let R_j and R_k be two regions, whose projections R'_j and R'_k are neighbors. We assume that the regions are adjacent in a segment-like region of a length, which is relevant for the object.

If S'_{jk} is a segment-like region, we consider R_j and R_k as neighbors. If S_{jk} consists of just one or very few points, the adjacency of regions R_j and R_k is ignored. Let R_j, R_k be neighbors, then there are mainly two possibilities:

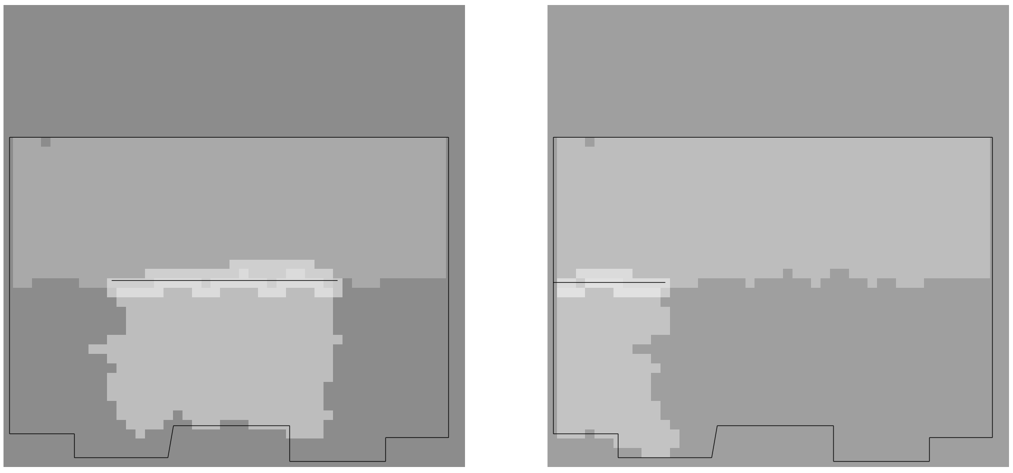

Figure 4: Intersection mask between regions and estimated intersection segment.

1. The regions R_j, R_k intersect each other in a segment s_{jk} whose projection s'_{jk} is at least partially contained in S'_{jk}.
2. The regions R_j, R_k do not intersect or the projection of the intersection line does not meet the intersection region S'_{jk}. This property indicates height jumps of the object. An auxiliary vertical face has to be introduced to connect the faces R_j and R_k. The position of the vertical face is approximated by the intersection region S'_{jk}.

4.2 Estimates of the Intersection Segments

Let R_j and R_k be intersecting regions. The unbounded line of intersection l_{jk} is easily computed from the equations of the planes ε_j and ε_k corresponding to the regions R_j, R_k. We are just interested in the intersection segment s_{jk} which is really present at the object. Since in general it is not possible to calculate the vertices at s_{jk} immediately, we compute estimates for the two boundary points on the true segment. This estimation is denoted by $\tilde{s}_{jk}$. Its carrier line is clearly l_{jk}, but the vertices are numerically not exact.

The segment $\tilde{s}_{jk}$ is bounded by two auxiliary vertical planes $h_i : a_i x + b_i y + c_i = 0$. These planes are chosen to be orthogonal to l_{jk} and to pass through the extremal points of the intersection region S'_{jk} in the direction of l_{jk}, see figure 5. So, the segment $\tilde{s}_{jk}$ depends on the quality of the intersection region S'_{jk}.

Remark: If the object is truly piecewise planar with no additional features present in the faces and all adjacencies determined correctly, it is possible to find the vertices bounding the intersection segments directly. To achieve this goal we have to find three planar regions R_i, R_j, R_k which share the property that the following relations hold:

$$R_i \text{ intersects } R_j \text{ intersects } R_k \text{ intersects } R_i.$$

Any triple of regions with this property defines a vertex which is common to the segments s_{ij}, s_{jk}, s_{ki}. If there is a height jump between two of these regions, this property is no longer true. Also, the vertices which are intersections of segments with walls, cannot be computed in this way. Therefore we prefer to compute estimates $\tilde{s}$ of the segments first and compute the exact vertices afterwards. Moreover, if adjacency relations are missing or parts of the objects are not planar in a way that we are not able to compute planar regions, it is not possible to find such triples of planar regions immediately.

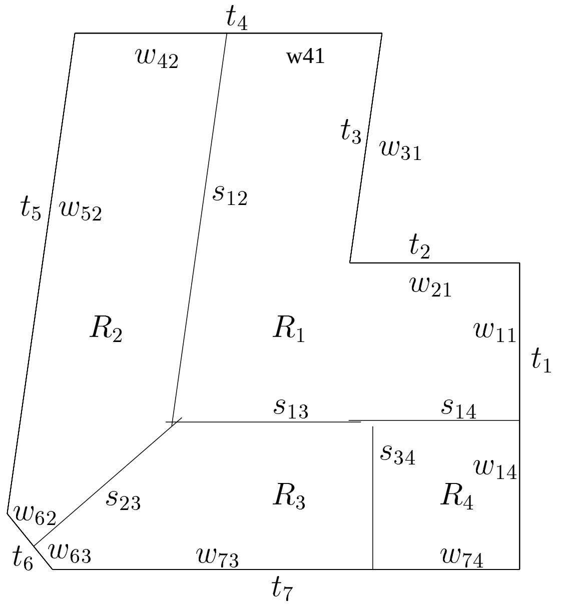

Figure 5: Estimation of intersection segments.

A note on *non–trihedral vertices*: When studying the simplified geometry of buildings or other piecewise planar objects, one recognizes that there are trihedral and non-trihedral vertices present. There are researchers and publications which pay attention to this problem and geometric constraints are formulated to model non–trihedral vertices. We have decided to omit this and generate cad-models with trihedral vertices only. This leads to the disadvantage that the finally generated models sometimes possess some vertices being close together.

4.3 Determining Height Jumps

For all pairs of regions R_j, R_k whose projections of their intersection lines l_{jk} do not cross the intersection regions S_{jk} in sufficiently many points, we have to introduce vertical auxiliary faces. It will be discussed now, how these faces and their position can be determined.

Let R_j and R_k be two planar regions whose top views are neighbors but which do not intersect. Further we have already computed the region S'_{jk}. Considering a roof as a piecewise linear function $f(x, y)$, the region S'_{jk} estimates a region of discontinuity. The discontinuity can be present along a simple edge but also along a polygon (curve). We have to determine the type of discontinuity.

At first, S'_{jk} approximates the region of discontinuity. At second, we may consider a piecewise linear function $f(x, y)$ over T with the property that the restrictions of f to the domains R'_j, R'_k are exactly the values determined by the carrier planes $\varepsilon_j, \varepsilon_k$ of these faces. By calculating the norm of the gradient ∇f and searching for large values of $||\nabla f||$, we also find an approximation for the region where the height jump between R_j and R_k takes place.

In practice, we may proceed as follows: If the height jump takes place along a simple segment s'_{jk} we try to find a segment contained in the region S'_{jk} which is 'perpendicular' to ∇f at points of S'_{jk}. This can only be true, if these mentioned vectors ∇f are nearly parallel.

If the height jump takes place along a curve or polygon, the gradient ∇f varies significantly at points of the region S'_{jk}. Ideally, the direction of the gradient ∇f is perpendicular to the curve, forming the discontinuity of f. The determination is a challenging task because of the low resolution of the data and the sensitivity of ∇f with respect to noise.

In case that the object possesses main directions we try to choose $\tilde{s}'_{jk}$ as a straight line segment parallel to the main directions. If S'_{jk} is curve-like or polygonal, the problem is difficult. We can try to find a polygon s'_{jk} containing as few as possible vertices such that the approximation of the medial axis of S'_{jk} by the polygon s'_{jk} is sufficiently good. The segments of the polygon are chosen to be parallel to the main directions, if possible.

We note that the currently implemented algorithm can handle height jumps along edges only.

4.4 Intersections of Roof Faces with Walls

This section does not possess any generalization to arbitrary piecewise planar objects since now the given ground plans which define the walls bounding the building play a very important role. What has to be done here is to determine all adjacency relations between planar regions at the roof and the given vertical planes, representing the walls. The determination is similar to the determination of adjacencies of roof faces. Let the ground plan polygon be denoted by T and its single segments be denoted by t_i. For any segment t_i we determine adjacent planar regions R'_j. If p_i are data points with an underlying grid we define adjacencies using the grid. If p_i are scattered vector data we will use a *constrained triangulation* of p_i, which additionally contains the segments t_i of the ground plane as edges. The procedure of intersecting the planar regions with the given walls consists of two steps:

1. Determine the adjacencies between segments t_i and regions R'_j.
2. Compute the necessary intersection segments w_{ij} between vertical planes W_i passing through t_i and planes ε_j which carry faces R_j.

In practice it is again not always easy to find the exact vertices bounding the intersection segments w_{ij}, thus we compute estimates for these vertices first. The estimated segments are denoted by $\tilde{w}_{ij}$. In particular, if two or more roof faces are adjacent to a vertical wall and the roof faces do not intersect each other, this estimation is necessary.

On the other hand, if the segment t of the ground plan polygon has only one neighboring face R and the investigation of the adjacency indicates that the end points of t are the projections of the end points of the intersection segment w_{ij}, we have found the true segment.

In general, the output of this procedure are estimates $\tilde{w}_{ij}$ of the intersection segments w_{ij} of all roof faces R_j with adjacent walls W_i. Figure 5 shows regions and approximated intersection segments of roof faces and walls for a building. For simplicity, the tilde symbols are omitted in the displayed notations.

4.5 Computation of Faces Bounded by Closed Polygons and Generation of the Cad-Model

So far, we have computed all intersection segments s_{jk} and w_{ij} between the segmented roof faces R_j, R_k and between the walls W_i and the roof faces R_j, Our intention is to calculate closed boundary polygons r_j for all roof faces R_j. We use the notation from above, where T denotes the polygon of the ground plan, t_i denotes its segments, $\tilde{s}_{jk}$ denotes the estimated intersection segment between the roof faces R_j and R_k and $\tilde{w}_{ij}$ is the estimated intersection segment of the wall W_i and the roof face R_j. The polygons r_j to be computed, shall possess the following properties:

1. The projection r' of each polygon r bounding the roof face R is entirely contained in the polygon T.
2. Polygons r'_j and r'_k corresponding to adjacent regions R'_j, R'_k touch along s'_{jk}. This is the projection of the intersection segment s_{jk} or the projection of a vertical auxiliary face between R_j, R_k.
3. The polygons r_j do not intersect each other.

Finally, we need an additional property: *The domains determined by the polygons r_j should form a complete and valid partition of the domain bounded by T.* That means, an arbitrary point q of the interior of T is either a vertex of one or more polygons, or it is contained in an edge of a polygon or it is an interior point of a unique polygon r_j.

The computation of closed polygons r_j for each roof face R_j uses the additional assumption, that the projection R'_j of every roof face R_j is a simply connected domain. This guarantees the boundary of R'_j to consist of one polygon r_j only. It is clear, this assumption will not hold in any case. A simple counter example is the roof of a small elevator room on the (flat) roof of a building. All small domains which are islands in other roof faces are ignored for the computation of the 'outer' boundary r_j of a domain R'_j. These extra features have to be treated as extra buildings and have to be modeled in this way.

But that is in general a crucial point: Modeling of those extra features is dependent on the detection of height jumps between regions/faces. If height jumps happen along polygons or curves the modeling of auxiliary vertical faces is most difficult.

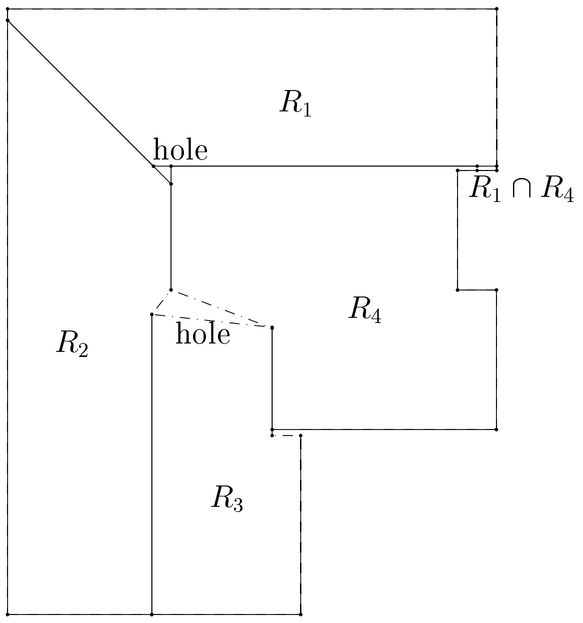

Figure 6: Closed polygons as projections of roof faces within the ground plan.

The computation of closed polygons is basically easy if all segments $\tilde{s}_{jk}$ and $\tilde{w}_{ij}$ are present and are good estimates of the true segments. To obtain a simpler notation we formally unify the families of segments $\tilde{s}_{jk}$ and $\tilde{w}_{ij}$ but denote the resulting set of segments again by $\tilde{s}_{jk}$. The two steps to be done for each family of segments $\tilde{s}_{jk}$, where j is fixed (index of the roof face) and $k = 1 \dots m$ are the following:

1. Sort the segments $\tilde{s}_{jk}$ in the way that the sorted segments $g_1, \dots, g_m$ have the property that the intersection $g_i \cap g_{i+1}$ is a vertex of the desired polygon.

2. Intersect consecutive segments or insert an artificial segment if consecutive segments do not intersect.

3. Guarantee that the final polygon r_j is entirely contained in T and does not intersect any other polygon r_k.

That leads to closed polygons r_j for all roof faces R_j. In practice it might happen that the collection of all polygons r_j do not form a valid partition of the domain T, but there exist holes which are not contained in any polygon.

It is necessary to introduce extra faces to fill the holes. Figure 6 shows holes but also intersecting roof faces. The planes containing these faces are estimated from data points lying near these holes.

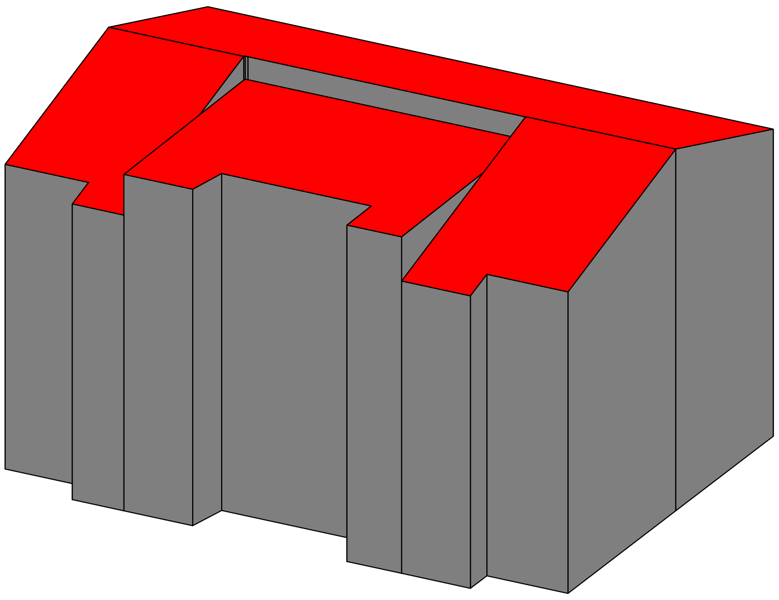

Figure 7: Cad-model of a building.

4.5.1 Generation of a cad-model

So far we have computed closed planar 3D-polygons for each roof face and we know which faces intersect and which faces have to be connected by auxiliary vertical faces. The last step is to finalize the cad-model. What is missing is the computation of the auxiliary vertical faces. By analyzing all interior edges s'_{jk}, it is easy to check if a face has to be introduced. If there is a height jump between regions R_j and R_k, two z-values are present for each vertex of the segment s'_{jk}. The vertical face which is introduced is a vertical trapezoid, with two vertical edges and two edges v_j, v_k corresponding to s'_{jk}; v_j lies in the face R_j and v_k is a segment in R_k. If v_j and v_k do not intersect in an interior point, we are done. If the segments v_j, v_k intersect in a boundary point, the trapezoid becomes a vertical triangle. If the two segments v_j, v_k intersect, the invalid trapezoid must be splitted into two triangles. This completes the generation of a cad model, see Figures 7 and 8.

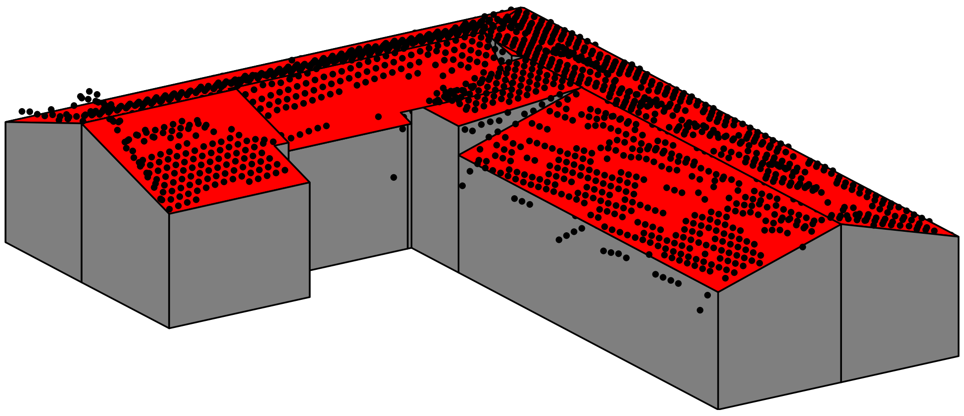

Figure 8: Point cloud and cad-model of a building.

4.6 Summary

We have described a method to generate building models from laser scanner data. It mainly consists of these steps:

1. Detect planar regions.
2. Consider additional geometric constraints.
3. Generate a cad model.

A prototype of what is described here is implemented in Matlab. Many features of the technique are also useful for reverse engineering of other technical objects which are piecewise planar. The details clearly depend on the resolution of the data and the special geometry of the objects to be reconstructed.

Acknowledgement

This work has been carried out within the K plus Competence Center *Advanced Computer Vision* and has been funded from the K plus program.

References

[1] Benkő, P., Martin, R.R., Várady, T.: Algorithms for reverse engineering boundary representation models, *Computer-Aided Design* 33, 839–851, 2001.

[2] Brenner, C.: Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen, Dissertation, Uni. Stuttgart 2000.

[3] Brenner, C.: Towards fully automatic Generation of City Models, IAPRS, Vol. 33, Part B3, Amsterdam, 2000.

[4] Haala, N.: Gebäuderekonstruktion durch Kombination von Bild- und Höhendaten, Dissertation, Stuttgart, 1996.

[5] Maas, H.G.: The suitability of airborne laser scanner data for automatic 3D object reconstruction, Third international Workshop on Automatic Extraction of man-made objects from aerial and space images, Switzerland, 2001.

[6] Langbein, F.C., Mills, B.I., Marschall, A.D. and Martin, R.R.: Finding Approximate Shape Regularities in Reverse Engineering Solid Models Bounded by Simple Surfaces, *Sixth ACM Symposium on Solid Modelling and Applications*, Ann Arbor, Michigan, 2001.

[7] Mills, B.I., Langbein, F.C., Marshall, A.D. and Martin, R.R.: Approximate Symmetry Detection for Reverse Engineering, *Sixth ACM Symposium on Solid Modelling and Applications*, Ann Arbor, Michigan, 2001.

[8] Pottmann, H., Peternell, M.: On approximation in spaces of geometric objects, In: The Mathematics of Surfaces IX (R. Cipolla, R. Martin, eds), Springer, 2000, 438–458.

[9] Serra, J.: *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[10] Steinle, E. and Vögtle, T.: Effects of different laser scanning modes on the results of building recognition and reconstruction, Intern. Archives

of the ISPRS, Vol. XXXIII, Part B3, Proc. ISPRS Congress, Amsterdam, July, 2000, pp. 858-865.

[11] Várady, T., Benkő, P., Kós, G.: Reverse engineering regular objects: simple segmentation and surface fitting procedures, Int. J. Shape Modeling **4** (1998), 127–141.

[12] Várady, T., Martin, R. R., Cox, J.: Reverse engineering of geometric models — an introduction. Comp. Aided Design **29** (1997) 255–268.

[13] Vosselman, G.: Building reconstruction using planar faces in very high density height data, IAPRS, Vol. 32, Part 3-2W5, *Automatic Extraction of GIS Objects from Digital Imagery*, München, 1999.

[14] Vosselman, G., Dijkman, S.: 3D building model reconstruction from point clouds and ground plans, Intl. Archives of Photogrammetry and Remote Sensing, Vol. XXXIV-3/W4, Annapolis, MD, 2001, pp. 37–43.

[15] Weidner, U.: Gebäudeerfassung aus digitalen Oberflächenmodellen, Dissertation, München, 1997.