OBJECT EXTRACTION FROM TERRESTRIAL LASER SCAN DATA FOR A DETAILED BUILDING DESCRIPTION

Katharina Jülge*, Claus Brenner

Institute of Cartography and Geoinformatics, University of Hannover, Germany
{Katharina.Juelge, Claus.Brenner}@ikg.uni-hannover.de

Working Group III/4

KEY WORDS: Feature, Extraction, Matching, Modelling, Close Range, Terrestrial, Laser scanning.

ABSTRACT

Building reconstruction has received increasing attention in the last few years. Many systems deal with reconstruction from aerial laser scans and images. As there is an increasing demand for more detail in a building's description, terrestrial data sources become more important. Preferably, extraction and reconstruction are realised as a fully automatic process. This paper describes a new approach for segmenting and describing a building's façade. Basic models representing façade parts such as windows are constructed as aggregations of geometric primitives like lines. Models are parameterized. Edges are then extracted from laser scan data of a single building. Extracted edges are preprocessed using a length filter. Subsequently, the previously defined models are fitted into the processed edge representation of the building using a constrained search approach. The goal is to find multiple occurrence of a particular shape (i.e. multiple windows), represented by an object model with a fixed parameter set, in one building. This approach works semi automatically with a view to full automation in the future.

1 INTRODUCTION

This work is part of a research project which deals with automatic derivation of 3D city models. Four data sources are used for this:

1. airborne laser scans
2. terrestrial laser scans
3. airborne images
4. terrestrial images

Objectives of the project include the fusion of multiple data sets and automatic object extraction. This paper presents one approach towards that goal. Images coming from one of the named data sources, in this case range images derived from terrestrial laser scans, are segmented. Edges are extracted as low-level primitives. These edges are to be aggregated into higher-level primitives; they represent geometric structures present in the image. Because our main concern are 3D city models, we have chosen building's façades as an example. We are looking for geometric structures which represent parts of the façades that occur multiple times, such as windows, doors or ornaments. If possible, these geometric structures are searched for in images coming from the other data sources in a way that correlations are found.

In this paper, a procedure is developed to extract those shapes. Model-based matching is used and constrained search is applied to find successful matches.

* Corresponding author

Our approach uses mostly 2D image processing tools, and searching is done using 2D features, so preprocessing has to be done to allow for distortion. To reduce the search problem to two dimensions, features are extracted in the façade using depth information and are then projected into the façade's plane. Search is carried out matching the geometry of objects found in the façades.

2 RELATED WORK

Several papers deal with the extraction of façade structures. For example, in (Wang et al., 2002) a system extracting windows from the orthoimage of a façade is described. A consensus texture façade image is calculated from several luminance-normalized façade images by weighted averaging. The resulting image is deblurred and rectangles are fitted into windows in the façade by an oriented region growing algorithm. This way, rectangles are iteratively fitted into blobs representing windows so that they grow to assume the window's size in the image. Extracted windows are then grouped using clustering algorithms.

Another example is (Werner and Zisserman, 2002). Here, the data source is an edge image with depth information calculated over multiple views. Parameterized models are fitted into façade structures such as windows or doors. Models are three-dimensional and include structures composed of straight lines such as boxes as well as more complicated models which contain arcs. To select a particular model, probabilities for models are determined using the Bayesian rule over a set of training images.

In both examples, multiple photos are used as the data source. In our case, we want to apply an algorithm falling into the same category as those two to a single range image.

In (Sester and Förstner, 1989), the concept of fitting generic models, described by a set of parameters, is presented. In this paper, emphasis is put on dealing with uncertainty.

There is also an extensive repertoire of employing search algorithms for matching problems. In (Rottensteiner, 2001), we find a classification of matching techniques which is based on (Gülch, 1994). According to this, there are three main branches of matching algorithms:

1. Raster based matching: Correspondences of images or image patches are found by comparing grey levels or function of grey levels.
2. Feature based matching: Features are extracted from images and mapping occurs between those features. This means basically finding matches between the geometric description of objects found in different images.
3. Relational matching (Vosselman, 1992): Here, topological relations of features found in images are matched. This is achieved by creating feature adjacency graphs first and then searching for matches between those graphs.

Building interpretation trees is a technique often used to establish mappings between the items in the respective search space. Depending on whether one wants to find some matches or all matches between model and data pixels, features or graph nodes, a partial or exhaustive search needs to be conducted.

In our case, we want to match geometric features doing an exhaustive search to find all relevant structures in a building's façade. To achieve this, we use a constrained search approach which is described in detail later. Several object recognition systems were successfully implemented using this technique. Earlier work includes (Grimson et al., 1990), (Flynn and Jain, 1991) and (Walker, 1999), where fuzzy rules are used to allow for uncertainty in the measurement of the constraint parameters.

3 SEGMENTATION

We are looking for structures in façades, like windows, doors and ornaments, that have the following properties:

1. They occur repeatedly and are arranged in a certain way, for example in rows and columns or other regular fashions.
2. They consist of features that represent discontinuities, i. e. edges.
3. They have the same size.
4. They have identical geometric properties, like angles and distances between edges.

Starting from these prerequisites, we construct models for the façade structures. For the moment, we concentrate on windows. These models consist so far of straight lines. The simplest model is a rectangle with variable aspect ratio. More complex models have a rectangle as the outline and also contain interior structure like grids which are often found in windows. Figure 1 shows those generic models used.

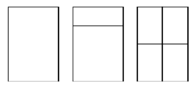

Figure 1: Models for windows.

To find instances of these models, we have to decide on a suitable representation of the laser scan data and extract straight lines. Those straight lines are then matched to one of the models using search, as described in chapter 4.

The range image of a terrestrial laser scan is used. A subset of the point cloud containing a building's façade is clipped. The range image is modified so that the façade appears parallel to the image plane, i.e. points having the same distance from the façade are assigned the same range value (see figure 2).

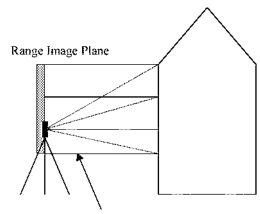

Figure 2: Modified range image.

Burns' line extracting algorithm (Burns et al., 1986) is then used to segment the range image. The result is a table of straight segments representing breaks in depth of the range image. Segments are filtered for length as very short segments are usually insignificant. We now give a short description of how Burns' algorithm works. Our implementation for the application of this algorithm on range images is identical to the one for ordinary images, no adaption is necessary.

For the line extraction, gradient images in x and y direction are calculated for the range image. Pixels are then sorted into overlapping buckets according to their gradient. Pixels in the same bucket are grouped using region labelling. For each region, a plane is fitted which represents the gradient slope in that region. This plane is then intersected with a plane representing the average gradient in every region. This way, straight lines are obtained. The lines are clipped by the boundaries of their support regions.

Those 2D segments can be transformed into 3D by finding points in the point cloud corresponding to the endpoints of the extracted lines. Then, a plane representing the building's façade is found. All 3D segments are projected into that plane. This way, distortion of façade structures can be eliminated.

4 CONSTRAINED SEARCH

4.1 Principle

Constrained Search is a technique for matching models to data devised by (Grimson et al., 1990). The principle is to build an interpretation tree that associates model features with data features (see figure 3). In order to avoid search explosion by testing every possible pairing of model features and data features, constraints are used to prune the interpretation tree. The goal is to use these constraints to rule out inconsistent matches at an early stage of the search. Every subnode of a pruned node will not be visited during the search process, therefore complexity of the tree is reduced.

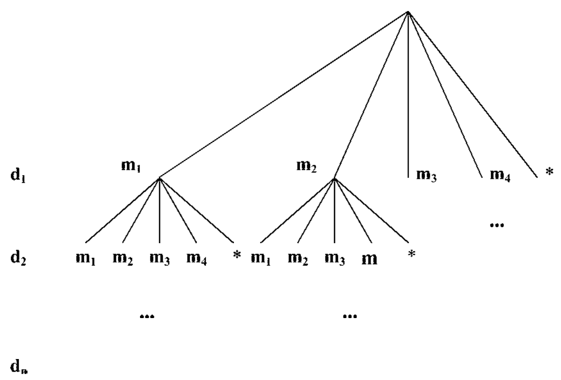

Figure 3: Part of interpretation tree.

Constraints can be unary or binary. Unary constraints define consistency of pairings between one model and one data feature, whereas binary constraints define consistency between pairings of two model features and two data features. Definitions for the constraints used in our system will be given later.

Overall consistency of the match is verified by finding a transformation that transforms the model features into data feature space (pose estimation) and calculating the deviation. If the deviation is below a set threshold, the match is considered consistent and therefore accepted.

The depth of the tree is defined by the number of data features, whereas the number of children of each node corresponds to the number of model features. Null associations are used to account for data features which are not associated with a model feature.

4.2 Application

In our system, model segments are associated with data segments. We need to determine constraints which are meaningful in the context of this task. As stated in chapter 3, the structures we are looking for have certain geometric properties. These can be interpreted as constraints. In particular, we are going to make use of orthogonality and parallelism. These can be phrased in the form of an angle constraint and a distance constraint. Because we don't use absolute values for angles and distances in the range image, constraints are binary and check that relations between model edges hold for relations between data edges. As the models described earlier contain the right angles and parallel lines that can be found when examining façade structures, these are exactly the conditions that hold true for the data edges which these models are matched to.

So, the following binary geometric constraints are used for pruning the interpretation tree:

1. Angle Constraint: This constraint compares the angle between two models segments to the angle between two data segments. A user-defined deviation is allowed. Formally it looks like this:

$$\theta_{i,j}$$

be the angle between the edge normals of the data edges i and j,

$$\Theta_{pq}$$

be the angle between the edge normals of the model edges p and q. Then

$$\begin{aligned} &angle\_constraint(i,j,p,q) = true \\ \textit{iff} \quad &\theta_{i,j} \in [\Theta_{pq} - \epsilon_a, \Theta_{pq} + \epsilon_a] \end{aligned} \tag{1}$$

2. Distance Constraint: This constraint compares the distance between two model segments to the distance between two data segments. Obviously, this constraint is not scale invariant. Maximum and minimum distance between the endpoints of one edge to the line through the other edge are calculated for both edges.

$$d_{l,ij},\, d_{h,ij}$$

be the smallest distance and the biggest distance between data edges, whereas

$$D_{l,pq},\, D_{h,pq}$$

be the smallest and the biggest distance between model edges. A certain deviation is allowed.

$$\begin{aligned} &distance\_constraint(i,j,p,q) = true \\ \textit{iff} \quad &[d_{l,ij}, d_{h,ij}] \subseteq [D_{l,pq} - \epsilon_d, D_{h,pq} + \epsilon_d] \end{aligned} \tag{2}$$

The distance constraint is of particular importance because it provides the most effective pruning of our interpretation tree without discarding correct solutions.

Unary constraints are not used because consistency checks can be made only by relative comparisons between model and data segments, not absolute comparisons. Constraints that consider length are not used because generally the length of a data segment is not very meaningful. In our range images, edges are usually extracted in a fragmented way and can be much shorter or much longer than an edge in the real world.

Because we want to find all possible matches of models in our data set, we do an exhaustive search of the interpretation tree and store every consistent solution found.

4.3 Verification of Hypotheses

Photogrammetric methods are used to determine the transformation that transforms our model into data space. Two transformations in are of particular importance: a Helmert transformation with 4 free parameters and an affine transformation with 6 free parameters in two-dimensional space.

Because known approaches usually work for estimating parameters by transforming points, significant points are used for each edge instead of transforming lines. There are several possibilities for this and certain problems associated with each possibility:

1. Middle points of edges: Because the length of an edge is of limited significance (real edges are usually fragmented into several edges in the edge image), the middle point of an edge is somewhat arbitrary. Apart from this, information is lost because a line contains more information than a single point.
2. Endpoints: Here, the same problem with the length and exact position of an edge occurs. In fact, the endpoints of an extracted edge usually don't coincide with the endpoints of a real edge. Besides, using endpoints means that edges are given an orientation. To correctly find all possible solutions, each edge would have to be considered twice, once in each direction. This doubles the depth of the interpretation tree.
3. Intersection points: These probably give the most useful information. Intersection points are calculated by treating edges as lines, so points not directly lying on an edge are also found. In fact, those intersection points usually give the best approximation for the endpoints of the real edges of a building's facade. The only problem is that a set of edges can have many intersection points, some of them too far away from the extracted edge to be meaningful. Solutions containing such points have to be rejected.

The general form of a 2D affine transformation is as follows:

$$X = a + cx - dy \tag{3}$$
$$Y = b + ex + fy \tag{4}$$

In a Helmert transformation, it holds that

$$e = d \tag{5}$$

and

$$f = c \tag{6}$$

so the equations simplify to

$$X = a + cx - dy \tag{7}$$
$$Y = b + dx + cy \tag{8}$$

The Helmert transformation has proven particularly useful in our example. It provides rotation, transformation and uniform scaling in both directions.

An approach for estimating parameters based on least squares adjustment for equally weighted observations is used (Niemeier, 2001). The coefficients of the transformation equations for all points are written in matrix form:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & x_1 & -y_1 \\ 0 & 1 & y_1 & x_1 \\ 1 & 0 & x_2 & -y_2 \\ 0 & 1 & y_2 & x_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_p & -y_p \\ 0 & 1 & y_p & x_p \end{pmatrix} \tag{9}$$

The points in the target co-ordinate system are written as

$$\mathbf{l} = \begin{pmatrix} X_1 \\ Y_1 \\ \vdots \\ X_p \\ Y_p \end{pmatrix} \tag{10}$$

The estimate for the transformation parameters is calculated as

$$\hat{x} = (A^T A)^{-1} A^T l \tag{11}$$

5 SEARCH STRATEGY

5.1 Initialization

First, an initial model needs to be fitted. Our model so far is a rectangle with variable aspect ratio which is automatically estimated. For this, the user of our system is required to select a structure in the edge image by enclosing it with a bounding box. A generic rectangle is then fitted into the edges inside the bounding box. For the initial fitting, an interpretation tree is built for matching four model segments to the data segments present. No constraints are used. At leaf level of the tree, an estimate for the aspect ratio of the rectangle is made. The model rectangle is stretched accordingly. Then, a Helmert transformation is calculated to transform the rectangle into data space.

Typically, if there are more than four data segments present in the bounding box, more than one suitable rectangle is found. In this case, the rectangle matching most data segments and providing the best fit is used. Alternaltively, the user can select one of several solutions. This way, a custom-made model is found for the structure which we wish to find in our edge image.

5.2 Search

The model derived during initialization is now searched for in the whole edge image or a user-defined subsection thereof. It can be freely rotated or translated, but the aspect ratio is kept constant, and only a limited amount of scaling is allowed, as we will see later.

An edge image can consist of several hundred edges, so pruning of the interpretation tree is inevitable. The angle and distance constraint defined earlier are used for that. The distance constraint also rules out solutions which would require scaling of the model beyond the bounds of the constraint, which accounts for the limited amount of scaling.

Because there are usually many edges in the façade of a building and the number of edges determines the depth of the tree, careful application of constraints is necessary to avoid search explosion. It is possible to divide the search space and search several small trees instead of one big one. For this, the edge image is tiled after initialization. Tiles are twice as big as the initialized model, and tiling takes place in a way that tiles overlap halfway so every correct solution is contained in at least one complete tile, although it can also be contained partially in several other tiles.

Effectively, tiling means enforcing the distance constraint in a way that two data segments can't be part of the same match if their distance is more than twice the size of the model. For the search inside the individual tiles, the distance constraint can now be relaxed without causing a search explosion because the number of edges inside a tile is usually small by comparison to the total number of edges in a building façade.

It is also possible to filter edges after initialization and before further search. An angle criterion can be used because after the first fit it is known which angles edges have to be suitable candidates for more matches.

Another way to reduce complexity during search and also eliminate finding the same solution several times is to delete data edges from the search space after matching them successfully. If a real edge consists of several segmented edges, more than one match could be found which essentially represents the same building structure. In fact, only one solution is of interest, so the rest can be safely omitted. Apart from this, due to the rotational symmetry of models, a solution is found multiple times, once for each orientation of the model. If data edges are deleted after finding the first solution, this can't happen anymore.

6 EXPERIMENTAL RESULTS

6.1 Tests

We testet our procedure on various single laser scans of buildings. This section presents the results for the Opera House in Hannover. First, the range image is calculated and line extraction is applied. The edge image is shown in figure 4. A section of the edge image showing a single window was selected and used for initialization of a rectangle (figure 5).

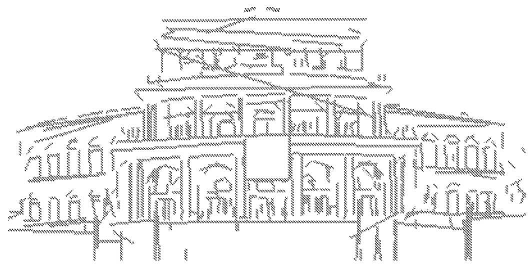

Figure 4: Edge image.

A small frame is defined by the user for an initial estimate of one window. A generic rectangle is fitted and stretched according to the window's proportions (see figure 5). No constraints are used.

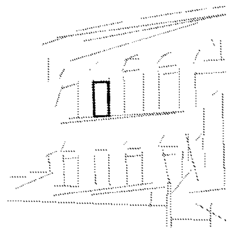

Figure 5: Initialized model.

After successful initialization, multiple occurrences of this window are found by matching the rectangle to the edges inside a user-defined bigger frame, which can contain the whole façade or a particularly interesting part thereof (see figure 6). Full automation for this step is aimed for, so far the user also needs to define deviations for the constraints so a meaningful fit is achieved.

In figure 6, it can be seen that windows of similar size and shape as the initial window are successfully found. Some of the windows are found multiple times. This is due to the fact that there are many small edges for each window and the initial rectangle can be successfully fitted into several different subsets of them. To remedy this problem, one could apply a more complex window model, or otherwise use a filtering postprocessing step which eliminates overlapping matches. No meaningful matches are found in the central part of the building.

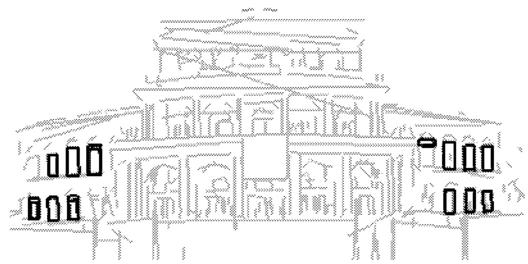

Figure 6: Correspondences found.

6.2 Discussion and Future Work

In the example presented, initialization for one window leads to successful matching of similar windows once they are properly extracted. So far, a pose estimate is only carried out for matches which yield at least 4 intersection points. That means that windows are found where there is at least one edge per side actually extracted from the range image. It is desirable to allow for a limited amount of uncertainty so two or three edges per window can be used to instantiate a model. This will be subject to further investigations.

The extracted windows can be used to propose a pattern in which windows are arranged along a façade. Windows found by the same model with the same constraints parameters are used for that. This makes it possible to predict the presence of windows also for spots where no structures are found: The hypotheses can then be used to direct another search step.

Another way of improving the fit is to use weighted estimation of parameters. Possible candidates for weights are the following:

1. Intersections of edges that are actually present in the edge image could be assigned a higher weight than intersection points that are calculated by prolongations of edges.
2. The length of the extracted edges could be used in some way.

7 CONCLUSION AND OUTLOOK

A semi-automatic method for finding multiple occurrence of a shape in a building's façade has been proposed. In this paper, we have described how we applied segmentation of laser scans to produce an edge image and constrained search to match structures in the edge image.

In the future, there are several applications for this procedure in our research project. We will apply the algorithm to photos of a building as well in order to find correspondences between the laser scan and the photo. The objective is to automatically apply textures to 3D models of a building derived from a laser scan. Models for shapes so far consist only of straight lines. It is planned to extend the model library so that models contain parameterized curves as well.

Once structures are found, properties describing a building's façade can be defined. For example, one can count the number of windows which are arranged horizontallay or vertically. It is even possible to conclude the number of storeys that a building has. It is also possible to determine the relative size and position of windows or other structures by comparison to the building's size and geometry and identify a particular building amongst others in images coming from a different data source.

It is also possible to use this algorithm for registration of different terrestrial laser scans. From structures found in every single scan, one could estimate the relative pose of these scans to each other and calculate a transformation. This way, structures found in buildings by our algorithm can replace tie points which are generally used for this task.

REFERENCES

Burns, J. B., Hanson, A. R. and Riseman, E. M., 1986. Extracting straight lines. IEEE Transactions on Pattern Analysis and Machine Intelligence 4, pp. 425–455.

Flynn, P. J. and Jain, A. K., 1991. BONSAI: 3-D Object Recognition Using Constrained Search. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, pp. 1066–1075.

Grimson, W. E. L., Lozano-Pérez, T. and Huttenlocher, D. P., 1990. Object Recognition by Computer. The MIT Press.

Gülch, E., 1994. Erzeugung digitaler Geländemodelle durch automatische Bildzuordnung. PhD thesis, University of Stuttgart, Institute of Photogrammetry.

Niemeier, W., 2001. Ausgleichungsrechnung. Walter de Gruyter.

Rottensteiner, F., 2001. Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS. PhD thesis, Vienna University of Technology, Faculty of Science and Informatics.

Sester, M. and Förstner, W., 1989. Object location based on uncertain models. In: Proc. 11. DAGM Symposium, Hamburg, Informatik Fachberichte, Vol. 219, Springer Verlag, pp. 457–464.

Vosselman, G., 1992. Relational Matching. Springer-Verlag.

Walker, E. L., 1999. Combining geometric invariants with fuzzy clustering for object recognition. In: Proc. of the Annual Conference of the North American Fuzzy Information Processing Society, pp. 571–574.

Wang, X., Totaro, S., Taillandier, F., Hanson, A. R. and Teller, S., 2002. Recovering façade texture and microstructure from real-world images. In: Proc. of the ISPRS Commission III Symposium: Photogrammetric Computer Vision, pp. A–381 ff.

Werner, T. and Zisserman, A., 2002. Model selection for automated reconstruction from multiple views. In: Proc. of the British Machine Vision Conference, pp. 53–62.

ACKNOWLEDGEMENT

The presented work has been done within in the scope of the junior research group "Automatic methods for the fusion, reduction and consistent combination of complex, heterogeneous geoinformation". The project is funded by the VolkswagenStiftung.