Quality enhancement of reconstructed 3D models using coplanarity and constraints

H. Cantzler[1], R.B. Fisher[1] and M. Devy[2]

[1] Division of Informatics, University of Edinburgh,
Edinburgh, EH1 2QL, UK
{helmutc,rbf}@dai.ed.ac.uk
[2] LAAS-CNRS, 31077 Toulouse, France
michel@laas.fr

Abstract

We present a process to improve the structural quality of automatically acquired architectural 3D models. Common architectural features like orientations of walls are exploited. The location of these features is extracted by using a probabilistic technique (RANSAC). The relationships among the features are automatically obtained by labelling them using a semantic net of an architectural scene. An evolutionary algorithm is used to optimise the orientations of the planes. Small irregularities in the planes are removed by projecting the triangulation vertices onto the planes. Planes in the resulting model are aligned to each other. The technique produces models with improved appearance. It is validated on synthetic and real data.

Keywords

Surface geometry, Shape, Scene analysis, Constrained architectural reconstruction

1 Introduction

The process of 3D reconstruction is often affected by noise in the measurements. Furthermore, inaccuracies are created by view merging, segmentation and surface fitting. One way to improve the reconstruction is to use more sophisticated methods like photogrammetry techniques. Another way is to exploit properties of the scene. Architectural scenes are particularly suitable for the application of constraints since the geometry is typically very structured. Architectural constraints can be used for 3D reconstruction from single [15, 7] or multiple [4, 1] intensity images. Features used for architectural constraints are typically straight lines, large coplanar regions and the parallelism and orthogonality of lines or planes. These kinds of features can be easily found in architecture scenes. In [3] research is described that improves architectural 3D models by automatically straightening edges. The work presented in this paper concentrates on extracting planar regions and applying coplanar, parallelism and orthogonality constraints more comprehensive then in previous work to the full 3D model. We apply the constraints to the data following meshing. Zabrodsky concluded in [16] that corrections following meshing generally give a greater improvement. Our method is independent of the calculation of the 3D structure unlike the work presented in [15, 7, 4, 1] where constraints are used in combination with reconstruction from intensity images.

This work consists of three steps. First, architectural features are extracted from already triangulated 3D models (Section 2). We use a RANSAC technique [5] to find planes in the model (similar to [2]). The next step is the automatic extraction of the constraints out of the scene. Few papers have dealt with the automatic extraction leaving it to the user to specify them [11, 14]. The interpretation of the scene is formalised as a constraint satisfaction problem [13]. Liedtke used a semantic net for interpretation of architectural scenes [8]. His interpretation is hypothesis driven. Hypotheses are verified or falsified by matching the 3D objects against the image. In our work we match the planes against a semantic net of a house by using a backtracking tree search (Section 3). The semantic net concentrates on the definition of the 3D objects and its relations. We check the interpretations only by verifying the relationships between the 3D objects. Constraints are assigned to almost-regularities like parallel or orthogonal walls. The last and final step consists of applying the constraints to the model (Section 4). The original model is fitted to the new constrained model. Optimising the model can be done in a number of ways (*e.g.* numerically [2, 14] or evolutionary [11]). We use an evolutionary approach. The model and the constraints are passed to the GenoCop 5 algorithm, proposed by Michalewicz [9]. The vertices are projected onto the planes after finding the optimal parameters. The result is a model with fewer irregularities (*e.g.* edges on walls) and aligned walls.

2 Feature detection

At all stages of the process, the model is a mesh consisting of vertices $V = \{(x, y, z)'\}$ and triangles $T = \{(v_1, v_2, v_3)\}$. The first step is to extract planes from the raw triangulated model. Before starting the extraction the model is normalised. It is mapped into an unit sphere at the origin. A robust RANSAC algorithm [5] is then used to obtain a set of planes. The algorithm generates a number of random plane hypothesis from the points in V. The distance of a triangle centroid to the hypothetical plane is calculated by computing the difference between the distance of the plane to the origin D and the dot product between the triangle centroid $C = (c_x, c_y, c_z)'$ and the unit plane normal $N = (n_x, n_y, n_z)'$. Triangles that satisfy the following inequality belong to the hypothetical plane.

$$|C \cdot N - D| < tolerance \tag{1}$$

The size of a hypothetical plane is calculated by adding up its triangle sizes. The hypothesis that creates the largest plane is selected. The exact number of planes in a model is not known. So, we repeat the RANSAC algorithm until the size of the resulting plane falls under a certain threshold. (An EM algorithm could instead have been used to select the number of planes and fit them, but we chose a simpler technique to focus on the reconstruction issues.)

This technique gives reasonable results. However, it sometimes produces a plane that consists of small disconnected patches distributed over the scene. An architectural plane (*e.g.* a wall) is not usually separated by a large gap. However small gaps frequently occur for example due to the presence of pipes or decorations. Therefore, the planes are analysed by single linkage clustering [6] to ensure that the triangles of a plane are closely connected. The cluster technique starts with the individual triangles and groups them together to form larger and larger clusters (hierarchical clustering). The distance between two clusters is defined as the minimal Euclidean distance of any two triangles belonging to different clusters (nearest neighbor method). The clustering terminates after reaching a certain distance. This distance specifies how far apart parts of the plane can be.

3 Scene interpretation

We interpret the scene using the features (planes) found previously. A model of an architectural scene is described in a semantic net (see figure 1). The model entities are represented as nodes in the net. The nodes are connected via different types of relationships. A semantically meaningful description is assigned to the scene features by matching them to the semantic net. A backtracking tree search is used to find the best match. The algorithm takes as input a set of features F, a set of possible model labels L and a set of binary model relationships R which limits the possible labelling. The tree search starts with the first feature from F and assigns all labels from L. A second feature is fetched from F and all labels are assigned. At this level some of the labels might be ruled out because they violate the given relationships. This process continues until all features have been labelled. A consistent labelling then exists if each feature is assigned a valid label that is also arc consistent with adjacent nodes. The relationships between features are used to select appropriate geometrical constraints for enforcing parallelism or orthogonality later in the optimisation step.

The model-entities (labels) and the relationships among the entities represent the knowledge of a typical architectural scene. Possible labels are L = {*Side Wall, End Wall, Base Plane, Ceiling/Floor, Roof, No Feature*}. The binary relationship functions check if the architectural relationship between two features and their labels is valid (*e.g.* horizontal and vertical walls are almost perpendicular). Angle relationships between two features are checked with a certain tolerance (3 degrees). The "Above" relationship is satisfied if 99% of the vertices of one plane are above a second plane defined by surface normal and distance. *No Feature* does not have any relation with a normal feature and can therefore be assigned everywhere. The final labelling is obtained by finding the solution that maximises the number of architectural labels.

The semantic net models a reasonable subset of all houses. It includes the interior and exterior structure of houses. The model can include an arbitrary number of walls. They can be on the same level or on different ones (then seperated by a *Floor/Ceiling*). The base plane is below all other parts of the building. It represents the ground on which the house stands. The roof is modelled as a typical sharp roof. Errors in the scene description are resolved by labelling them as *No Feature*. The semantic net can be easily extended with features like windows and doors. These features can be modelled as parallel and close to the actual walls. However, the previous plane detection concentrates on finding big planes. So, modelling windows and doors is not necessary at this step.

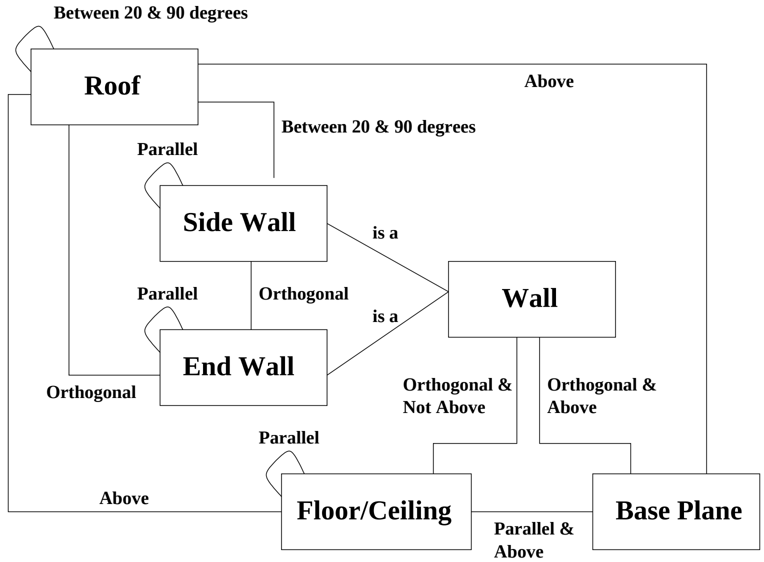

Fig. 1. The model of the architectural scene is represented by a semantic net. Nodes represent the model entities and are linked by architecturally meaningful relationships.

4 Model optimisation

Optimising the model by enforcing the constraints found previously is formulated as a nonlinear programming problem. There are many algorithms which are designed to search spaces for an optimum solution. Some of them become ill-conditioned and fail with nonlinear problems. We use the GenoCop 5 algorithm developed by Michalewicz [9]. It is a genetic algorithm (GA) which uses real-value genes and includes methods to deal with linear, non-linear, inequality and domain constraints.

The GA uses the parameter vector $\boldsymbol{p}$ which concatenates all the parameters for the individual planes as the chromosome. The evaluation function consists of the squared residuals of the vertices and the constraint functions. The squared residual is the squared geometric distance from the mesh vertices $\{x_{i,j}\}$ to their planes $\{P_i\}$. The residual of every plane is normalised with its number of vertices N_i. Thus, model size does not affect results. Every constraint is represented by a constraint function c(). The values of these functions correspond to the degree that the constraints are satisfied. The constraint functions can be seen as a penalty functions. λ is a weight factor which scales the constraints to the residuals.

$$\sum_i \frac{1}{N_i} \sum_j dist(P_i(\boldsymbol{p}), x_{i,j})^2 + \lambda \sum_i c^{(i)}(\boldsymbol{p}) \qquad (2)$$

Additionally, constraints are used to narrow the search space of the evolutionary algorithm. Domain constraints are applied to individual components of the surface normals and the distances. Each of the parameters can never be outside the range [-1,+1] since the 3D model is mapped into a normal sphere at the origin. Furthermore, unity constraints are applied to the surface normals N.

So far we have obtained the optimised model parameters. We now project the vertices of the planes onto their planes. We calculate the new coordinates $V_p = (x_p, y_p, z_p)'$ of the vertex with the original vertex $V = (x, y, z)'$, the unit surface normal of the plane $N = (n_x, n_y, n_z)'$ and the distance D of the plane to the origin as:

$$V_p = V - tN \tag{3}$$

where

$$t = \frac{V \cdot N - D}{N \cdot N} \tag{4}$$

5 Experimental results

The proposed technique described above is general. It is independent of the way the 3D model was created (*i.e.* from range or intensity data) and of model properties like variance of the triangle size. It has been applied to several triangulated models. We will here present results for a synthetic model and for two reconstructed real models.

First, we applied the described technique to the synthetic model. The model consists of a perfect mesh of three walls at 90 degrees (1323 vertices & 2400 triangles). Two walls are parallel. A varying amount of Gaussian distributed 3D noise is added to the vertices. The first graph shows the angle error from plane extraction (top curve), improving the plane fit (middle curve) and application of constraints (bottom curve, near noise level axis). Improving the plane fit is done without using any constraints in the evaluation function. The angle error from plane extraction is a result of the random nature of RANSAC. Improving the fit using all data points from the planes gives much better results. Finally, using the constraints gives an angle error very close to zero. The second graph shows the mean squared residual after plane extraction (top curve), improving the fit (dashed curve) and constraining the model (solid curve). The parameters obtained from RANSAC show the biggest error. The mean residuals from improving the fit and from applying the constraints are fairly similar and are both significantly below the the RANSAC curve. The two graphs show that applying constraints improves the orientation of the walls without worsening the fit.

We show an experiment with the reconstructed model of Arenberg castle (in Belgium) reconstructed by the Catholic University of Leuven [10]. The model was reconstructed from an image sequence of 20 images (6292 vertices & 12263 triangles). The walls and the ground on the original solid model show clearly a lot of small irregularities (see figure 4). 5 planes are extracted (3 walls, 1 floor and 1 roof). The planes are constrained by 7 constraints. The angles between the planes vary from the optimum by 1.5 degrees on average before optimisation. After optimisation they differ less than 0.01 degrees. The result shows the model with removed irregularities and constrained planes. The average disparity of the moved vertices as a fraction of the model diameter is 0.33%. The optimisation step took 54 seconds on an Intel Celeron with 400MHz.

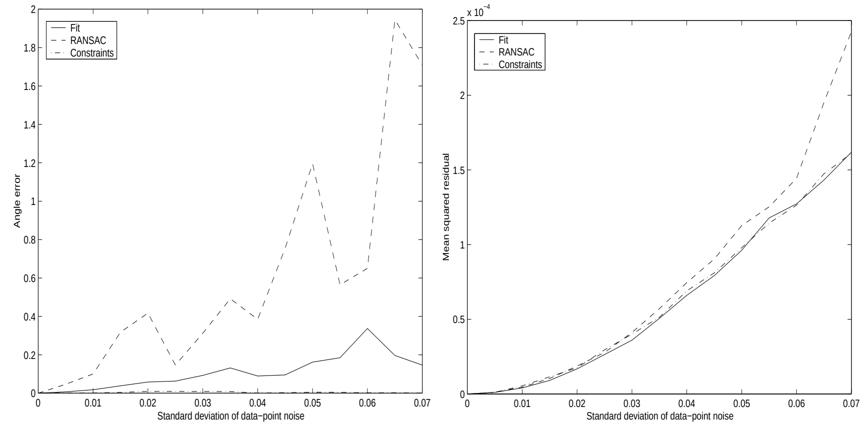

Fig. 2. Results for the synthetic model. The left graph shows the angle error in degrees versus the noise level. The graph on the right shows the mean squared residual versus the noise.

Next, we briefly describe results for a Bavarian farmhouse reconstructed by the European Commission Joint Research Centre (JRC) [12]. It was reconstructed from multiple range data scans (12504 vertices & 16589 triangles). This is a full 3D model. The plane extraction finds 4 walls and two planes for the roof. The orientations of the walls are already fairly good. The angles between the planes differ on average by 0.5 degrees in the original model. After optimisation they differ less than 0.01 degrees. The original solid model shows small edges on the walls. The result has these edges projected onto the wall (see figure 3 for a close view of a wall).

Fig. 3. A close view of a wall of the farmhouse. On the left is the unconstrained model. Surface ripples are most easily seen in the circled areas. On the right is the optimised model.

6 Conclusion and future work

Previous work used architectural constraints mainly for scene reconstruction from intensity images. This work shows how architectural constraints can be used for improving the reconstruction of full 3D models independent of the sensor data. Only 3D information is used. The constraints make architectural features more regular in terms of their architectural properties. We exploit common architectural features like walls and their relationships to each other.

Initially, a RANSAC technique obtains a set of planes from the 3D data. We automatically discover the graph of constraints between the planes by using a tree search strategy. Even conservatively loose thresholds on angles and position lead to a correct labelling of the planes in the scene. The model parameters are optimised with a robust evolutionary algorithm. A numerical normalisation of the model beforehand leads to domain constraints on the parameters which speeds up the search algorithm. The experimental results show how imperfections like small irregularities on planes and the orientations of walls are corrected. The visual appearance of the model is enhanced.

Future work aims at incorporating edges into the process of model optimisation. This includes extraction of edges in the model, straightening of edges and the use of parallelism or orthogonality constraints where applicable.

References

1. C. Baillard and A. Zisserman. A plane-sweep strategy for the 3d reconstruction of buildings from multiple images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 33(B2):56–62, 2000.
2. A. Bartoli. Piecewise planar segmentation for automatic scene modeling. *Conference on Computer Vision and Pattern Recognition, Hawaii*, pages 283–289, 2001.
3. P. Dias, V. Sequeira, J.G.M. Goncalves, and F. Vaz. Combining intensity and range images for 3d architectural modelling. *International Symposium on Virtual and Augmented Architecture, Dublin*, pages 139–145, 2001.
4. A. Dick, P. Torr, and R. Cipolla. Automatic 3d modelling of architecture. *British Machine Vision Conference, Bristol*, pages 372–381, 2000.
5. M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
6. A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
7. D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. *Eurographics*, 18(3):39–50, 1999.
8. C.-E. Liedtke, O. Grau, and S. Growe. Use of explicit knowledge for the reconstruction of 3d object geometry. *International Conference on Computer Analysis of Images and Patterns, Prague*, pages 580–587, 1995.
9. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.
10. M. Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, University of Loewen, 1999.
11. C. Robertson, R.B. Fisher, N. Werghi, and A. Ashbrook. Fitting of constrained feature models to poor 3d data. *Proceedings Adaptive Computing in Design and Manufacture, Plymouth*, pages 149–160, 2000.

12. V. Sequeira, K. Ng, E. Wolfart, J.G.M. Goncalves, and D.C. Hogg. Automated reconstruction of 3d models from real environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:1–22, 1999.
13. D.L. Waltz. *Generating Semantic Descriptions from Drawings of Scenes with Shadows*. PhD thesis, AI Lab, MIT, 1972.
14. N. Werghi, R.B. Fisher, A. Ashbrook, and C. Robertson. Shape reconstruction incorporating multiple non-linear geometric constraints. *Computer-Aided Design*, 31(6):363–399, 1999.
15. M. Wilczkowiak, E. Boyer, and P.F. Sturm. Camera calibration and 3d reconstruction from single images using parallelepipeds. *International Conference on Computer Vision, Vancouver*, pages 142–148, 2001.
16. H. Zabrodsky and D. Weinshall. Using bilateral symmetry to improve 3d reconstruction from image sequences. *Computer Vision and Image Understanding*, 67(1):48–57, 1997.

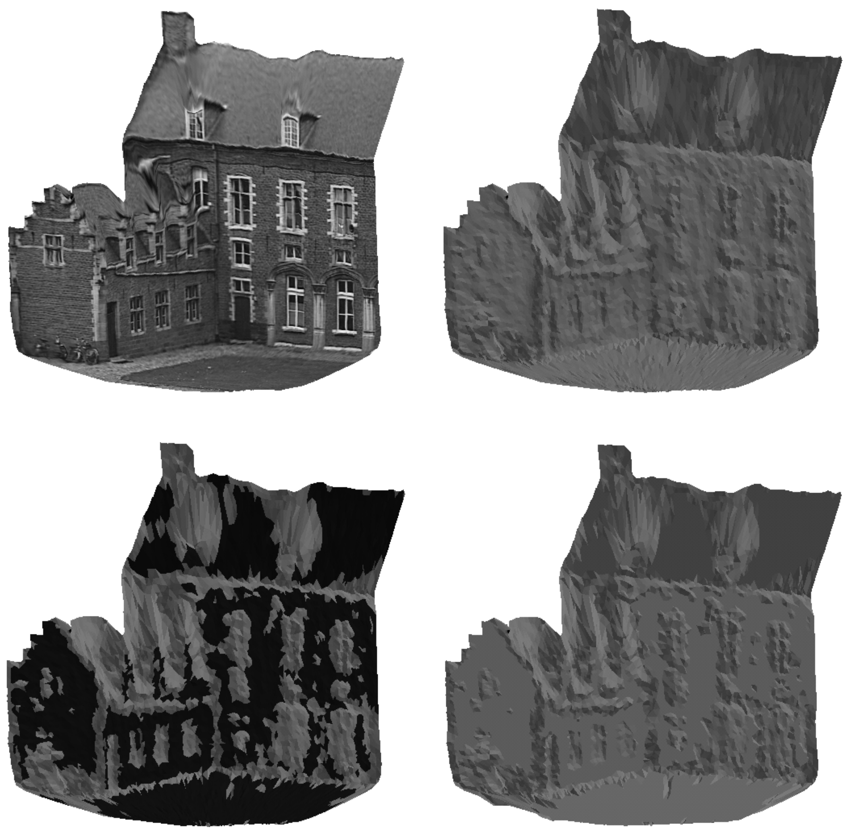

Fig. 4. The textured model (top/left), the original solid model (top/right), the extracted planes (bottom/left) and the resulting model after optimisation (bottom/right) from the castle. The extracted planes are displayed a bit darker than in the solid model.