EG UK Theory and Practice of Computer Graphics (2012)
Hamish Carr and Silvester Czanner (Editors)

Simplified User Interface for Architectural Reconstruction

Fabian Wanner, Fabrizio Pece and Jan Kautz

University College London, UK

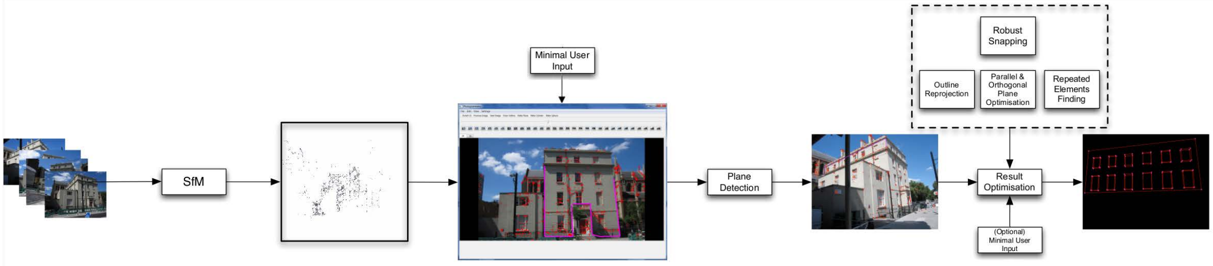

SfM
Minimal User Input
Plane Detection
Robust Snapping
Outline Reprojection
Parallel & Orthogonal Plane Optimisation
Repeated Elements Finding
Result Optimisation
(Optional) Minimal User Input

Figure 1: *System overview.*

Abstract

We present a user-driven reconstruction system for the creation of 3D models of buildings from photographs. The structural properties of buildings, such as parallel and repeated elements, are used to allow the user to create efficiently an accurate 3D structure of different building types. An intuitive interface guides the user through the reconstruction process, which uses a set of input images and a 3D point cloud. The system aims to minimise the user input by recognising imprecise interaction and ensuring photo consistency.

1. Introduction

We present a hybrid framework (Figure 1) for reconstructing façades and buildings. It combines features from rule-based and semi-automatic modelling that exploit the advantages of such approaches while overcoming their limitations. We use the strengths of these systems and combine them with a more user-friendly interface that adjusts user input to create results more quickly. The goal of this work is to design a system that will not force the user to compromise on features, but that can be intuitive, fast and automated enough to let the user concentrate on the artistic aspect of modelling.

The initial input to our system is a set of photographs of the target object, which are used to obtain a 3D point cloud and the camera parameters from structure from motion. Different computer vision algorithms are applied to create 3D models quickly and efficiently with limited user input. The user draws the outline of a plane in one view from which the 3D location of the plane and its boundaries are estimated. User input is supported by fixing inaccurate interactions and finding repeated elements. Afterwards, the reconstruction process is executed that fits planes while taking into account the relationships between them.

2. Related Work

Architectural modelling is a challenging task whose success depends on the quality of the resulting model and the ease of its creation. Its approaches can be divided broadly into two main categories: automatic and semi-automatic.

Automatic Model Reconstruction To perform automatic reconstruction a point cloud and the camera intrinsic and extrinsic camera properties are essential. The most common method to obtain this information is Structure from Motion (SfM) [SK93]. Geometric information like planes can be fit to the output of SfM to construct a rough model. Sinha *et al.* [SSS09] use the scene's vanishing points and directions to automate the scene modelling and 3D line reconstruction, while Gallup *et al.* [GFP10] try to fit local best planes at several locations in the scene, and then merge them into a single global plane. Li *et al.* (GlobFit [LWC*11]) show a successful approach on analysing the global properties of an object to facilitate its 3D reconstruction.

Interactive Model Reconstruction A common problem of automatic reconstruction is that the object to be modelled

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG12/037-040

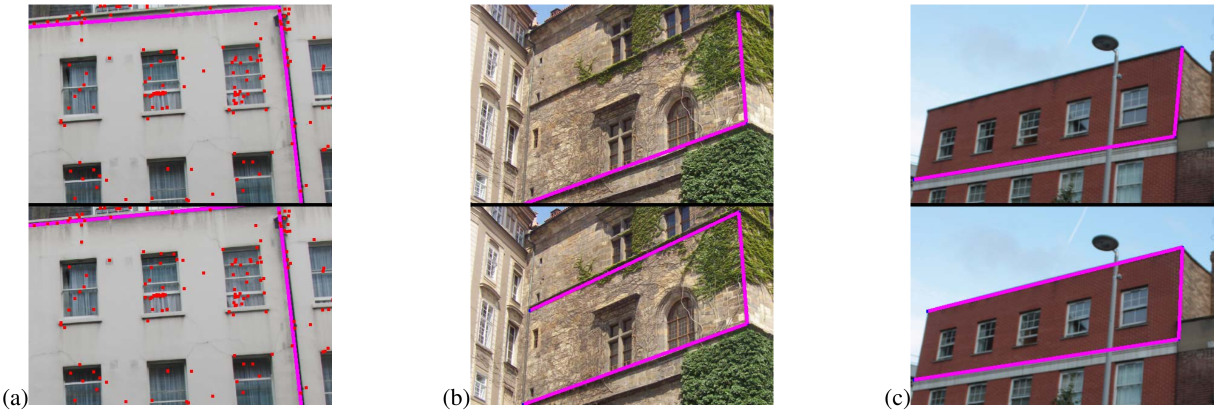


Figure 2: *Robust snapping. The first row shows the original clicks, the second row the results (only a partial image is shown to show details of the snapping location). The algorithm correctly handles intersections that lie outside the window (a), obfuscated edges (b) and occlusion (c).*

can only be partially reconstructed from the input. Procedural modelling [MWH*06] overcomes such limitations by employing a set of rules to model objects. Recently, techniques have been developed where the grammars can either fit to existing data [MW07] or be edited interactively [LWW08]. While rule-based modelling can generate big scenes efficiently, replicating and modifying existing buildings is difficult. Therefore, semi-automatic approaches that overcome some problems like ambiguities [XFT*08], object replacement [NSZ*10] or repetition detection ([WFP10]) have been introduced. Finally, Sinha *et al.* [SSS*08] introduced different types of interaction, such as plane welding, the creation of fillet planes and extrusion, to simplify the creation of models considerably

3. Preprocessing

The system obtains a 3D point cloud and the camera parameters using the freely available software Bundler [SSS06]. The density and accuracy of the point cloud depends on the quality and number of the input images. For Bundler to succeed, the individual views have to overlap each other so that matching feature points can be found across multiple views. We circumvent errors introduced by lens distortion by running a rectification step on the input images to obtain undistorted data. Bundler produces a sparse point cloud that is the backbone of our system, and usually is sufficient. If the number of 3D points is too small, dense Multi-view Stereo algorithms can be used to create a denser point cloud at the cost of longer execution times.

4. User Input Support

In this system, we describe how we make the user's task as simple as possible, while he is creating planes by outlining them. This simplification is achieved by attempting to increase the accuracy of the user input by predicting its intended location. We also discover repeated elements in the images to reduce the amount of interaction required.

User Interaction Prediction The task of outlining a plane can be very tedious and time consuming due to the high accuracy of user input needed. Our approach reduces the required accuracy.

Whenever an edge is created by the user, the system selects an area A around the edge points to find the optimal edge from the image. The size of A depends on the length of the edge, as short edges are more likely to contain big errors. A Canny edge detector finds the dominant edges in A and then we apply the Hough Transform to find the longest edge. To reduce the computational time, the Hough Transform's search area is limited to the area A. Furthermore, the optimal edge and the edge marked by the user are assumed to be similar. This allows us to reduce the gradient search space, especially for long edges. These two speed ups make the delay for finding the optimal edge only noticeable with very long edges. The corners are then calculated by intersecting the individual optimal edges. This algorithm copes with partial occlusions and also edges that partially lie outside the image, as seen in Figure 2.

Repeated Elements Many buildings contain repeated elements and marking all of these can be a tedious task for the user. For that reason, we incorporated a simple repetitive element detection algorithm, where the emphasis lies on finding most of the repeated elements rather than avoiding false positives. Our algorithm works by using each of the corners of a previously marked plane P and running Normalised Cross-Correlation on the entire image for each of them. Figure 4(b) gives an example of the corner candidates CC. The number

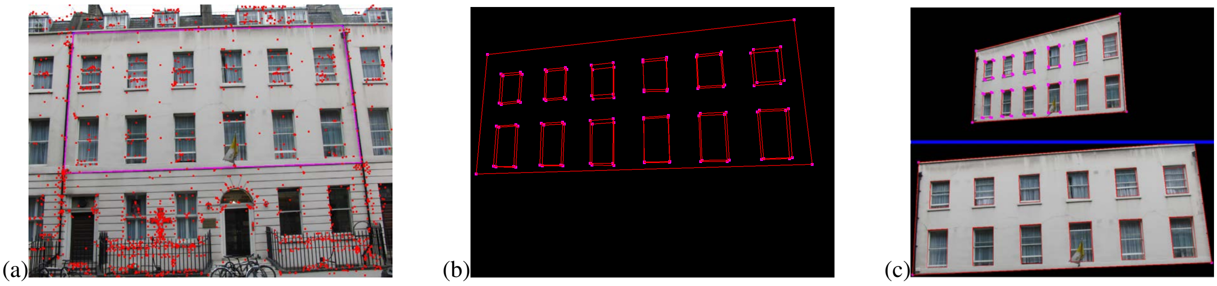


Figure 3: *System results: (a) original view, (b) untextured results, (c) textured results (different view points).*

of false positives and missing corners is large. P also provides us with information about the distances between the corners. Searching through all the potential corners, we detect repeated element candidates by comparing distances between P and all the CCs. If we find a set of points where at least 50% plus one of the corners lie within 15% of the distance of the previously calculated corners we accept the result as a repeated element. These thresholds proved to give the best compromise between recall and precision. The results can be seen in 4(c). Missing or incorrect detections can be fixed easily by the user through drag and drop or candidate removal respectively.

5. Plane Creation and Optimisation

The improved user input from 4 is used to create, modify and fit planes in the scene. These points as well as the relationship between planes are discussed here.

Plane Creation and Modification After preprocessing is completed, the emphasis of our system lies on minimising the required user input. Instant feedback allows the user to guide the reconstruction process and immediately correct mistakes by the system. To create a plane it is sufficient to draw its outline in a single image, as the point cloud and camera properties from SfM provide sufficient information to find the plane's 3D location.

From the outline, point cloud and camera parameters, a modifiable plane is generated using RANSAC and least squares plane fitting. Different types of interaction can be performed on existing planes. Merging points allows treating these points as a single point with reduced degrees of freedom to ensure coplanarity of the individual planes. To improve the model's quality, interactions with entire planes, in a similar fashion as in Sinha *et al.* [SSS*08], are also possible: planes can be moved along their normal or 'welded' together, removing gaps and overlaps between them. A fillet planes without 3D points due to lack of visibility in the images or texture, can be created by selecting two planes that have a common point. Extrusions, a common feature in buildings, can be performed by marking the area to be extruded. Any such user input is used to adjust the reconstruction using the newly added information.

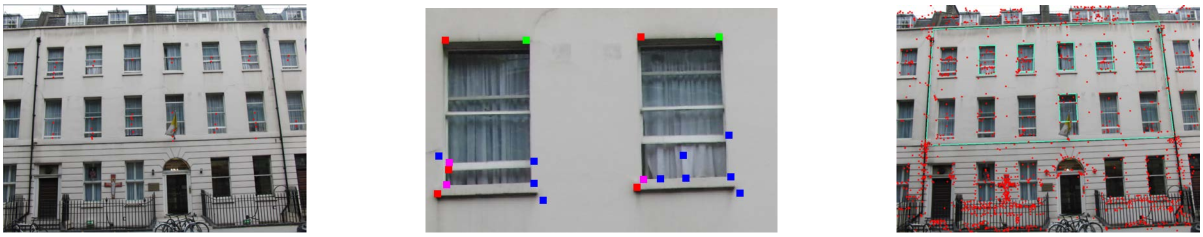
Object Relationships Each plane is a separate entity, changing a set of equivalent objects requires transforming each of them individually. In our system, the relationships between objects (as shown in [LWC*11]) are used to apply a transform to multiple entities simultaneously. For example, planes marked as parallel will remain parallel at all times.

To make use of such relationships, the user can specify them directly or indirectly through actions that create relationships between elements. Transitive relationships are obeyed and propagated to any subsequent actions. These relationships simplify the task of interacting with the scene as multiple extrusions can be handled concurrently or many planes moved around at once without breaking the already existing model and its properties.

Parallel Plane Optimisation After the plane creation, the user can set some planes to be parallel enforcing parallelism between all marked planes. Our initial attempt of running RANSAC on each of the planes and then adapting the rotation of the planes worked when both of the planes contained similar numbers of 3D points, but problems occurred when the point numbers varied a lot. To overcome this, we moved all points of the planes so that their centroid lies at the origin, as the centroid is always part of the optimal plane. Running RANSAC to remove outliers, then repositioning the points so that the centroid of the remaining points is at the origin and finally performing a least squares plane fitting operation allowed us to find optimal parallel planes, which then were moved back to their original position.

6. Results

As shown in Figure 2 the prediction of the users clicks is robust to many situations, for example occlusions and corners outside an image. The optimisations run at interactive speed as the search area in terms of space and angles is kept as small as possible. Figure 4(c) shows a sample result from the repeated element analysis, in which most of the repeated structures have been found. Despite the detection of many



(a) Running NCC for the whole template area. Note the large amount of false positives.

(b) Using NCC on all corners.

(c) Our repeated element analysis. 3rd window in top row was marked and others were found automatically.

Figure 4: *Repeated element analysis results.*

false positive corners (Figure 4(b)) from NCC, our algorithm is able to keep the false positive repetitive elements to a minimum. Incorrect or missing provides can be fixed easily through the tools provided.

The user interaction in our system proves to be easy to use, as shown in the video. The process for the creation of the model in Figure 3 had the following process. The user selected a view in which the entire part of the object to be reconstructed can be seen. Then the main plane is selected, followed by the selection of one window. During repetition detection one repeated element was missed and another one was incorrectly detected requiring only two interactions by the user to fix this. With our emphasis on improving the user interactive side of the interface, creating a simple but accurate 3D model from a small number of inaccurate user interactions is a good result.

7. Conclusion and Future Work

We presented a novel system to create 3D model of human-built objects that requires very little user interaction. The system gets as input a set of images, and outputs a 3D model. To minimise the user input, the system takes advantage of several computer vision algorithms and exploits structural properties of objects such as parallelism and repeated elements. The user is mainly required to outline planes in the model and to support the identification of relationships between objects. This allows the system to take away the burden of repetitive and trivial tasks from the artist while leaving to him controls of the reconstruction process. With the help of the user input, we keep the computation times to a minimum, as typically the system responds within one second to any user input. The models created are simple but accurate, in particular with respect to the relationships between the individual planes.

In the future, we will improve the results by finding alignment correspondences between edges to ensure that the window boundaries are parallel in 3D. At the moment, we are detecting optimal edges from only a single image. In the future, we are planning to expand this, so that it takes into account all the images provided. The same can be done for the repeated element analysis. Using more advanced texturing techniques would improve the quality of the model.

References

[GFP10] Gallup D., Frahm J.-M., Pollefeys M.: Piecewise planar and non-planar stereo for urban scene reconstruction. In *IEEE Computer Vision and Pattern Recognition* (2010).

[LWC*11] Li Y., Wu X., Chrysanthou Y., Sharf A., Cohen-Or D., Mitra N. J.: Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics 30*, 4 (2011).

[LWW08] Lipp M., Wonka P., Wimmer M.: Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1.

[MW07] Müller P., Wonka P.: Image-based Procedural Modeling of Facades. *Most 26*, 3 (2007).

[MWH*06] Müller P., Wonka P., Haegler S., Ulmer A., Van Gool L.: Procedural modeling of buildings. *ACM Transactions on Graphics 25*, 3 (2006).

[NSZ*10] Nan L., Sharf A., Zhang H., Cohen-Or D., Chen B.: Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (2010).

[SK93] Szeliski R., Kang S. B.: Recovering 3D shape and motion from image streams using nonlinear least squares. In *IEEE Computer Vision and Pattern Recognition* (1993), pp. 752–753.

[SSS06] Snavely N., Seitz S. M., Szeliski R.: Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH* (2006), pp. 835–846.

[SSS*08] Sinha S. N., Steedly D., Szeliski R., Pollefeys M., Agrawala M.: Interactive 3d architectural modeling from unordered photo collections. In *ACM SIGGRAPH Asia* (2008).

[SSS09] Sinha S., Steedly D., Szeliski R.: Piecewise planar stereo for image-based rendering. In *ICCV09* (2009), pp. 1881–1888.

[WFP10] Wu C., Frahm J.-M., Pollefeys M.: Detecting large repetitive structures with salient boundaries. In *Proc. of European Conference on Computer Vision* (2010), pp. 142–155.

[XFT*08] Xiao J., Fang T., Tan P., Zhao P., Ofek E., Quan L.: Image-based façade modeling. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1.