Vis Comput (2018) 34:605–616
https://doi.org/10.1007/s00371-017-1365-x

ORIGINAL ARTICLE

PhotoSketch: a photocentric urban 3D modeling system

George Wolberg[1] · Siavash Zokai[2]

Published online: 7 April 2017

Abstract Online mapping services from Google, Apple, and Microsoft are exceedingly popular applications for exploring 3D urban cities. Their explosive growth provides impetus for photorealistic 3D modeling of urban scenes. Although classical algorithms such as multiview stereo and laser range scanners are traditional sources for detailed 3D models of existing structures, they generate heavyweight models that are not appropriate for the streaming data that these navigation applications leverage. Instead, lightweight models as produced by interactive image-based tools are better suited for this domain. The contribution of this work is that it merges the benefits of multiview geometry, an intuitive sketching interface, and dynamic texture mapping to produce lightweight photorealistic 3D models of buildings. We present experimental results from urban scenes using our PhotoSketch system.

Keywords Image-based modeling · Phototextured 3D models · Structure and motion · Multiview geometry · 3D photography · Camera calibration

1 Introduction

Reconstruction of buildings in urban scenes remains an active area of research. The production of 3D textured building models supports a myriad of applications in navigation, mapping, entertainment, virtual tourism, urban planning, and emergency management. Popular navigation and mapping tools from Google, Apple, and Microsoft have widely disseminated the benefits of urban reconstruction to the general public.

The problem of creating phototextured 3D models of existing urban structures has spawned many interactive techniques as well as automatic methods [26]. The interactive modeling processes remain cumbersome and time-consuming, while automatic reconstruction methods are prone to errors and often yield noisy or incomplete results. Automatic methods such as multiview stereo [12] are often hindered by painstaking editing necessary to fix the dense 3D models they generate, which undermines their benefit in the first place. While automatic reconstruction methods are known to omit user interaction, it is generally accepted that they do not produce satisfying results in case of erroneous or partially missing data [26]. This motivates us to design a superior interactive system that benefits from automated camera pose recovery and sparse point cloud generation, but retains a human in the loop to guide the geometry completion.

Much work in urban reconstruction begins with laser range data acquired from LiDAR cameras. Using time-of-flight principles, these cameras yield semi-dense 3D point clouds that are accurate over large distances. Early work in the use of LiDAR data for reconstruction of urban environments is presented in [34,35]. In related work, [18,36] introduced methods for reconstruction of large-scale scenes modeled from LiDAR data captured by laser range scanners and 2D color image data for the purpose of generating models of high geometric and photometric quality. Although laser range scanners are traditional sources for detailed 3D models of existing structures, they are prohibitively expensive, not

✉ George Wolberg
wolberg@cs.ccny.cuny.edu

Siavash Zokai
zokai@brainstormllc.com

[1] City College of New York, CUNY, New York, NY 10031, USA

[2] Brainstorm Technology LLC, New York, NY 10001, USA

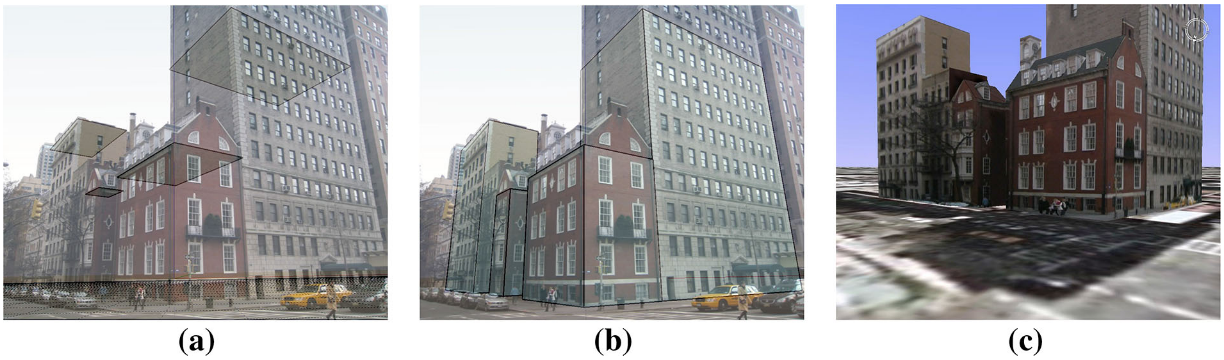


Fig. 1 **a** The image acts as a stencil upon which the user sketches building rooftops (*black boxes*) and performs (**b**) extrusion operations to generate a lightweight 3D model; **c** Final model georeferenced on Google Earth

available for mass markets, and generate heavyweight data that are often incomplete.

The subject of this paper deals with the generation of *lightweight* models from photographs, enabling this approach to reach a wide cross section of users. We propose a new system, called PhotoSketch, which is a photocentric urban 3D modeling tool that permits users to sketch directly on photographs of existing buildings. The sketches outline footprints that can be lifted into 3D shapes via a series of push–pull extrusion and taper operations. Rather than treating photographs as a postprocess that is applied *after* the model is generated, we use photographs as the starting point *before* the model is generated. Indeed, our workflow treats photographs as tracing paper upon which 2D shapes are defined prior to extruding them into 3D models. The very photographs that serve as the basis for the models automatically serve as the texture elements for them as well, thereby facilitating photorealistic visualization. This approach is characterized by users for whom the generation of approximate lightweight textured models is critical for interactive visualization.

The PhotoSketch system targets mainstream users who will feel at ease to draw upon photographs to create 3D models. The current state of the art is deficient in its efforts to *easily* produce lightweight phototextured models directly from photographs. This is the thrust that we pursue in this work. We incorporate structure from motion (SfM) to automatically recover a sparse point cloud and a set of camera poses from a set of overlapping photographs of a scene. This is essential to facilitate an intuitive user interface for building 3D models based on extrusion operations on sketches that are *drawn directly on photographs.* Although users have traditionally applied extrusion and push–pull tools in 3D environments, our application seeks to be more intuitive by applying these tools in 2D image space.

Rather than having a user fumble with the difficult process of orienting a 3D primitive into a 2D photograph of the 3D scene, the user is now able to directly draw upon the image along a recovered ground plane. In this manner, drawing can be constrained to the walls and floor of the scene to yield footprints that can then be extruded to form volumes. A model is constructed by sketching a 2D footprint on the photograph and extruding it to the proper height of the object by snapping to 3D points recovered via SfM. A push–pull graphical interface is used for this purpose. An example is given in Fig. 1.

2 Related work

An extensive survey of 3D modeling methods for urban reconstruction can be found in [26]. Our approach belongs to the category of interactive image-based modeling [10,29], which dates back to the origins of close-range photogrammetry [5,20,40]. These tools typically require a great deal of skilled user input to perform camera calibration and 3D modeling. The computer vision community has advanced image-based modeling by developing methods for automatic feature extraction and matching [6,19] and automatic camera pose recovery using multiview geometry [11,13].

One well-known system that creates models from photographs is Façade [8]. In that work, an approximate model is built using simple 3D primitives with a hierarchical representation. The user must manually specify correspondences between 3D lines in the model and 2D lines in the photographs. The system then solves for the unknown intrinsic and extrinsic camera parameters. Once the cameras have been calibrated, textures are projected onto the model. Although compelling 3D urban area models were demonstrated in Façade, the system required laborious and time-consuming user interaction to specify correspondences in the images to solve for camera poses.

The authors in [9] have implemented ShapeCapture for close-range photogrammetry and modeling. This system also suffers from tedious manual feature tracking among images for camera calibration. As the authors have stated, they needed to manually measure and match 30 features among images for a project. After initial calibration, the system automatically finds more matches based on the epipolar geometry constraint. The modeling process is simplified by using extracted 3D points (seed points) to fit architectural primitives based on the user selection.

VideoTrace [14] is an example of an interactive modeling tool that uses structure from motion in a video sequence. Their system is simple enough for average users to create realistic models of an observed object. However, manual contour tracing and tracking are required.

Similar to our work, the interactive system in [31] operates on unordered photographs and exploits structure from motion. The user draws outlines of planar faces on 2D photographs. Vanishing point constraints are used to estimate the normal and depth of each outlined face. This modeling tool suffers when the presence of vanishing lines is not strong. Furthermore, the modeling of curved facades cannot be handled.

In [4], an interactive modeling tool was proposed based on multiview stereo (MVS) semi-dense point cloud input. The system segments the point cloud into a set of planar regions and finds polygons using the edges of segmented regions. An optimization method is used to snap the edges of adjacent polygons to automatically create a rough model. The user interactively edits, adds details, and refines the model in the *point cloud* space.

In [27], a system was developed that allows the user to create a coarse model of a street block from point clouds generated by MVS. The user defines an instant of a template (e.g., windows, doors) on the image and model. The system then automatically finds them elsewhere in the scene using template matching and places the user-drawn template in those locations to refine the model. This approach is valuable when the urban scene is replete with repetitive architectural patterns.

Recently, inverse procedural modeling (IPM) is gaining popularity with promising results [22,25,38,43]. These approaches find a procedural representation of an existing object or scene. The inputs typically are images with known poses and/or semi-dense point clouds derived from MVS. The advantages include compactness and the ability to easily vary urban scenes using the recovered grammars of the buildings. However, these methods require strong a priori knowledge about the input images, such as a requirement to have different shading on each side of a building [38] or [22] requires a priori knowledge of the building architecture.

In [16,17,39,42], algorithms are presented to create lightweight models from LiDAR or semi-dense MVS point clouds. We opt to avoid LiDAR input because they are not widely accessible to average users, and we avoid the method of [42] because they generate sweepable models that cannot represent the full class of building structures we seek to model. We also opt to avoid MVS to create lightweight models due to the strict restrictions they place on the class of buildings that may be modeled. Our attention is drawn to structures that are not limited to boxes as in [16] or to digital elevation maps (DEM) as in [39].

A sketch-based method was proposed in [30] to add 3D man-made objects onto the terrain data. They use an oblique image of the scene to model the buildings in that image. The user draws several lines to define the major axes, and the system solves for the camera pose based on the orthogonality constraint from the single view. The model faces are then projected into the image to recover textures. However, their manual modeling method is limited to symmetrical Manhattan-world buildings and does not support buildings with complex rooftops.

In [7,44], systems were developed that allow users to sketch on a single photograph to create 3D models of objects in the scene. Both methods are suitable for highly symmetrical objects. The main problem with these methods is that they only work on a single photograph. This limitation is not suitable for large urban buildings that may require several photographs to capture all viewpoints to reduce occlusion ambiguities. Furthermore, these techniques are entirely dependent on accurate edge detection to detect the outlines of their proxies as they are defined and dragged. This is subject to error when handling highly variable lighting in outdoor architectural scenes. Finally, ornate architectural details are not well handled by the cuboid approximations in [44], which is limited to modeling Manhattan-world buildings.

3 PhotoSketch workflow

In this section we describe the PhotoSketch modeling workflow and demonstrate how its design simplifies the user experience for modeling urban areas. The input to the system is a collection of unordered overlapping images. Structure from motion (SfM) is then used to track features across photographs to determine the camera pose parameters. This permits us to bring all of the photographs into a single reference frame in which we will build the 3D model.

Once camera pose recovery is complete, any user drawing made upon one of the input images will appear properly aligned in the remaining images. The rationale for having multiple overlapping images is to facilitate total coverage of the scene in the presence of occlusions. Since each image can be projected back into the scene, the texture of all 3D faces will be derived from non-occluding views.

A basic premise of PhotoSketch is that a scene image is sufficient to guide the user through a series of sketching operations and to act as a stencil upon which the user traces a footprint of the building. The system is designed in such way to simplify the user experience for modeling urban areas. This

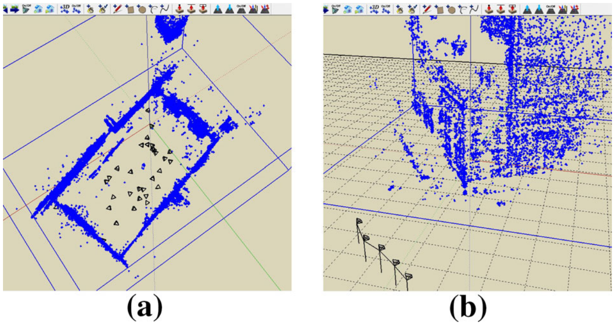


Fig. 2 **a** The recovered camera positions and the sparse reconstruction of Piazza Dante unordered dataset, **b** ordered dataset for a New York City building (on Park Ave. and 85th Street)

is achieved by providing a set of 2D sketching tools that are constrained to operate only on the ground plane and polygonal faces. These tools consist of rectangles, circles/ellipses, arcs, polylines, and splines. The ground plane serves as the sketch pad for drawing the 2D facade profile. Due to visibility issues, it is sometimes desirable to draw the footprint on a plane which does not coincide with the ground. Therefore, the user is permitted to change the offset, or height, of the sketch pad, with a zero offset referring to the ground.

The PhotoSketch workflow consists of the following steps: (1) automatic recovery of a sparse 3D point cloud and camera pose information by means of multiview geometry (Sect. 3.1); (2) alignment of the cameras with respect to the ground plane (Sect. 3.2); (3) interactive modeling based on sketching 2D footprints and applying a set of extrusion and taper operations which are guided by the photographs (Sects. 3.3, 3.4).

3.1 Structure from motion (SfM)

From a set of overlapping scene images, SfM uses automatic feature extraction and tracking to find the camera poses and reconstruct a sparse 3D point cloud [11,13,21,33,41]. The automatic recovery of camera poses is needed to accurately project the texture onto the model, and the sparse 3D point cloud is helpful to assist the user in snapping the extrusion or taper operation to the desired height. It is important to note that the recovered structure is sparse and incomplete. Although it is inadequate to fully model the object, it is useful to aid the user in building the model.

Figure 2 depicts the camera poses and sparse reconstruction of the Piazza Dante [37] and Park Avenue / 85th Street (NYC) datasets. The frustums in Fig. 2 represent the recovered camera poses. These results were derived from our own SfM implementation. It is possible to apply other open-source solutions such as Visual SfM [41], Bundler [32], or OpenMVG [23]. The user can feed their photographs to these systems, and we can import and parse the output of these systems to get camera poses and a sparse point cloud.

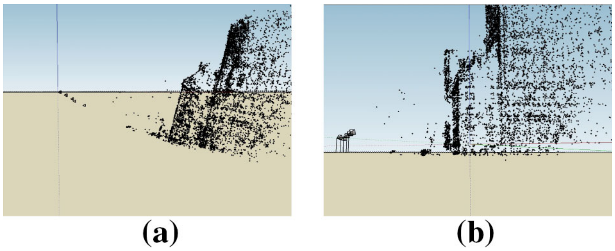


Fig. 3 Since the multiview geometry does not have knowledge of ground orientation, the structure and poses are not aligned with respect to the floor. Therefore we need a tool to properly align the ground and floor. **a** Before floor alignment, **b** after floor alignment

3.2 Alignment of the cameras with respect to the ground plane

Since the absolute position and orientation of the initial camera are unknown, we place the first camera at the origin of the world coordinate system, i.e., $K[I|0]$. Most SfM systems start with this assumption to set their frame coordinate system, unless there is additional information available from GPS and/or IMU data. With respect to this camera's coordinate system, the floor of the sparse 3D point cloud of the model now appears tilted, as shown in Fig. 3a. A ground plane alignment stage is necessary to properly rotate the camera and the sparse point cloud, as shown in Fig. 3b. This leaves the floor parallel to the ground plane.

This alignment is a crucial step for matching the 2D sketches of building footprints or rooftops across all views. In addition to the above problem, we assume that the extrusion operations are perpendicular to the floor, consistent with the facades of most buildings. We have developed an automatic method to recover the unknown rotation R_g of the first camera. This is achieved by having the user invoke a lasso tool in our 3D point selection system to collect a set of 3D points on a flat surface on the ground. We fit a plane through these points using the RANSAC method, which is robust to outliers. The normal n of the recovered plane is the direction of gravity in the SfM coordinate system. We solve for the rotation R_g that rotates normal n to align with our world coordinate system up direction $(0, 0, 1)$.

In practice, we observed that in many occasions the points on the ground are occluded with cars, trees, pedestrians and there are not enough flat 3D points to infer a plane. Also, the noisy selected 3D points are unreliable for ground plane detection in real-world situations since a few degrees of error heavily degrade the model. The user can easily observe this error when a face is pulled upward and the edges of the drawn volume do not visually appear aligned to the images. In such case, the user can activate our manual ground plane detection

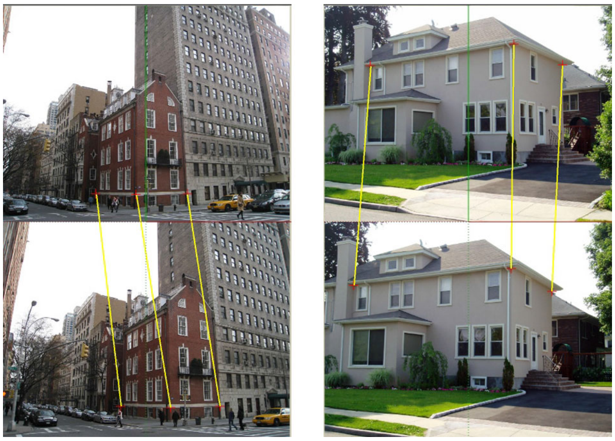


Fig. 4 Examples of correspondence points that lie parallel to the ground plane

tool by selecting at least three corresponding image points in two views that correspond to a floor or roofline in the image (Fig. 4).

The 3D position of these selected image points can be determined by triangulation since the camera poses are known. A plane is fitted to these 3D points, and the angle between the fitted plane and the ground plane of the world coordinate system determines the rotation angle necessary to rigidly rotate the 3D point cloud and the cameras. This method will also leave the floor parallel to the ground plane.

3.3 Sketching 2D profiles

After recovering the floor orientation, the user can snap the height of this plane to any 3D point in the sparse point cloud. If a point on the ground is visible, then it is best to snap to it so that the modeling may proceed from the ground up. Usually, however, the ground points are occluded and it is easier to snap to a visible point on, say, the roofline to establish a footprint. That footprint may then be extruded down toward the ground. This approach was used in the examples in this paper.

Note that if there is no 3D point to snap the ground plane to the roofline or to the floor, the user can invoke the system's manual feature tracking to establish correspondence of a visible corner among the scene images. Since the camera poses are known, we find the 3D position of the tracked features by triangulation.

After the cameras and the floor are aligned to the ground plane, the user can select images from the input set and look at the 3D scene through their respective camera frustums. The user then sketches on the ground plane. The user can select a 2D drawing tool such as a rectangle, polyline, circle/ellipse, or spline and outline the visible footprint of the building. This process only requires the user to click on the corners of the building facades. To assist the user in this process, we provide a virtual magnifying glass to help the user accurately pinpoint the corners.

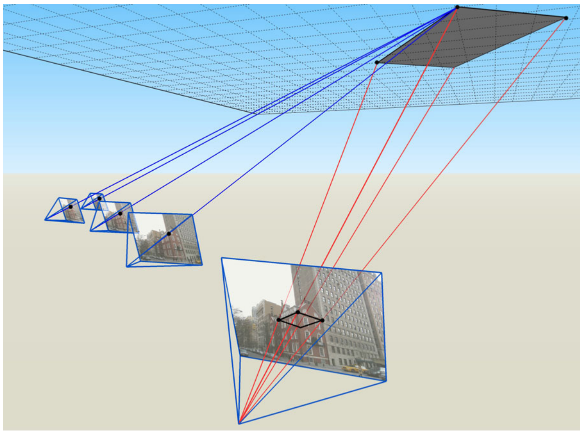


Fig. 5 The user draws a 2D footprint in one image. The 3D positions of the footprint corners are determined by shooting rays (*red*) from the camera frustum through these corners onto the ground plane, which has moved to the height of the rooftop in this example. Those 3D points are reprojected along rays to the other frustums to render their corresponding images in the other views. *Blue rays* illustrate this reprojection for a single corner point

Figure 5 shows this process in action. The user clicks on three corners of the rooftop on the rightmost image in the figure to get a parallelogram lying along the roof in the image. In order to determine the 3D points of these 2D corners, we shoot a ray from the center of projection of the camera frustum through each of the 2D points and compute the intersection onto the "ground" plane. These are shown as red lines in Fig. 5. The resulting 3D points can be reprojected to the other camera frustums, with the resulting blue rays passing through the corresponding corners in the other image views. This is all made possible by camera pose recovery, as computed using SfM. As a result, any sketch made in one image is properly projected onto all of the remaining views.

Our system allows the user to switch from one viewpoint to another during sketching to add points from corners that are occluded in the current view. Figure 6 shows the footprints drawn in black. As a result of SfM, the camera positions and orientations are known. Therefore, a footprint drawn in one viewpoint will appear registered in the other viewpoints. Since the drawing plane height is selected by snapping to a 3D point on the edge of a roof top, each 3D position M_i of the drawn footprint corners is known. We can project M_i into view j based on the known extrinsic and intrinsic parameters derived from SfM and get its 2D projection m_i on view j using Eq. (1).

$$m_{ij} = K_j(R_j M_i + T_j) \tag{1}$$

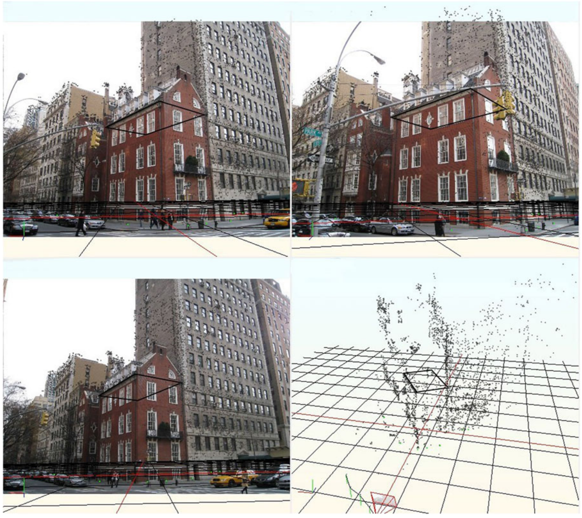
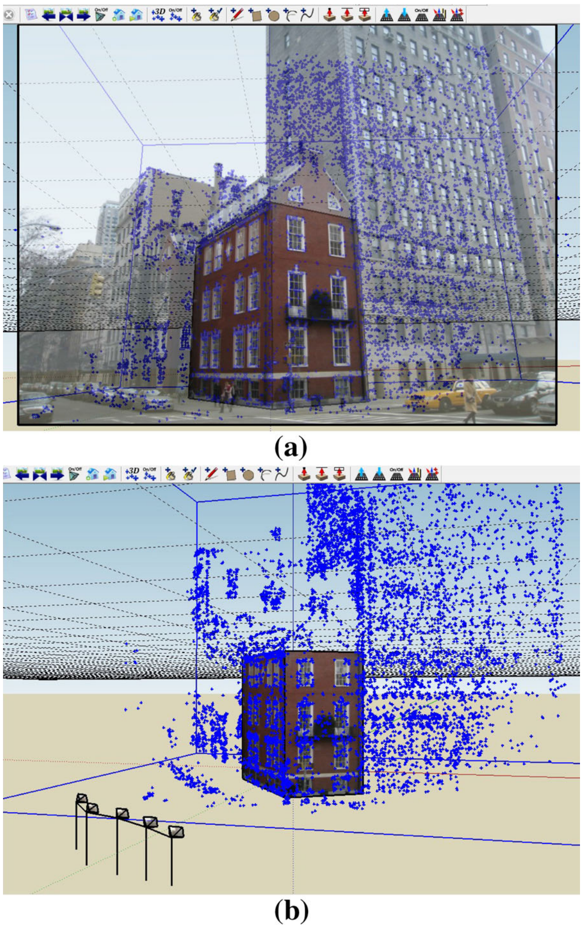


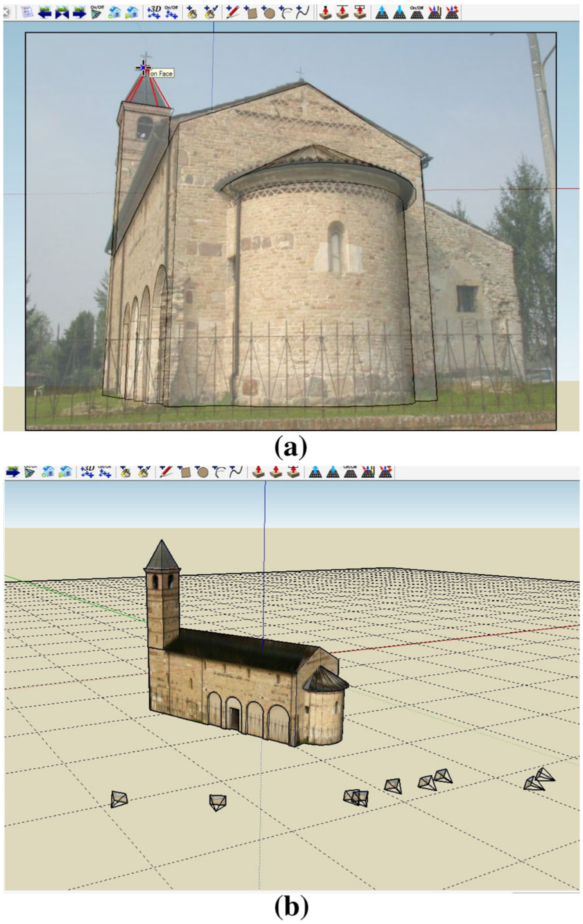
Fig. 6 The user has sketched a 2D footprint of the building on one of the images. The 2D footprint is shown in *black* in the different camera views



(a)

(b)

Fig. 7 The result of an extrusion operation in PhotoSketch. **a** Scene viewed through a camera frustum; **b** scene viewed from arbitrary vantage point behind the five camera frustums



(a)

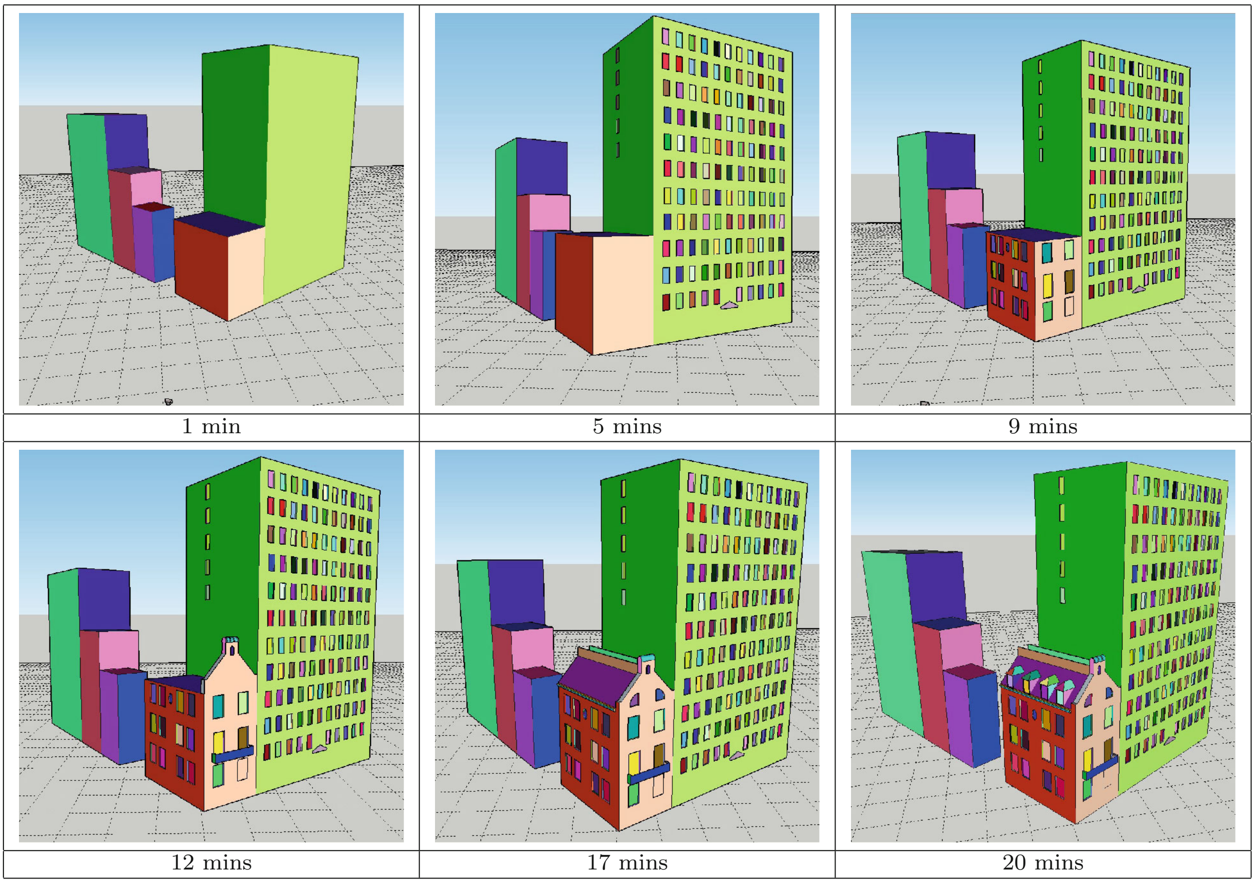
(b)

Fig. 8 The result of a taper operation on the Pozzoveggiani church [1]. **a** Scene viewed through a camera frustum as the user pulls the apex of the roof to align with the image; **b** scene viewed from arbitrary vantage point behind several camera frustums

Note that K_j is the intrinsic 3×3 matrix of the camera for view j, and the extrinsic parameters are rotation R_j and translation T_j, which define the camera pose for view j.

3.4 Extrusion, push–pull, and taper operations

The basis of our work assumes that a simple set of extrusion and taper operations is adequate to model a rich set of urban structures. This is consistent with related work in procedural modeling. [24,28] have shown that a simple set of rules is sufficient to generate an entire virtual city. However, procedural modeling focuses on creating a model from a grammar. Although this approach can automate the creation of generic urban areas, it is not appropriate for reconstructing *existing* buildings. Recent work in inverse procedural modeling [15,26,43] finds a procedural representation of an existing object or scene. This approach, however, requires strong a priori knowledge about the input images or building architecture.



1 min

5 mins

9 mins

12 mins

17 mins

20 mins

Fig. 9 Snapshots of the modeling process over time

PhotoSketch attempts to reach beyond these limitations by putting a human in the loop and establishing a simple set of rules in which the user can model buildings efficiently and rapidly from the existing photographs. The simplest available operation in our toolset is extrusion from footprints. The user only needs to drag the footprint to the desired height. This can be done either by snapping to the height of a 3D point recovered from SfM or to a visual cue on the image based on dynamic texturing. Here we want to emphasize that dynamic texturing is a key advantage of our system, assisting the user to model based on real-time texture projection. By projecting the photograph back onto the model, any modeling errors become quickly apparent in the form of misaligned texture and geometry. Real-time dynamic texturing is implemented using GPUs. Figure 7 shows the result of an extrusion operation on the footprint of Fig. 6.

A push–pull interface is available to the user to perform extrusion. Further refinement is possible by snapping the faces to sparse 3D points that represent a plane. Sketching is not limited to drawing footprints on the ground plane. The user may also draw on extruded faces and use the push–pull interface to refine the model.

The user can further edit the model by tapering to a point, line, or offset. This is often used to model rooftops. In these cases, the user can snap to a featured 3D point that represents the tapered height or dynamically adjust the height for getting an appropriate texture on the visible faces. Figure 8 shows the result of a taper operation after the extrusion operation.

4 Results

Our modeling software features simple and intuitive tools that users can leverage to create complex models in a short amount of time. Figure 9 shows snapshots of the modeling process and the elapsed time at each stage. The user can accelerate the modeling process by creating a template of a window or other architectural features and then applying copy and paste operations, individually or as a set of features. Furthermore, inference tools within our system allow for fast and accurate snapping of templates to edges and faces.

The whole process of modeling the scene in Fig. 9 was completed in 23 min. This session is broken down into three stages consisting of automatic camera pose recovery, floor alignment, and modeling, which took approximately 2, 1, and

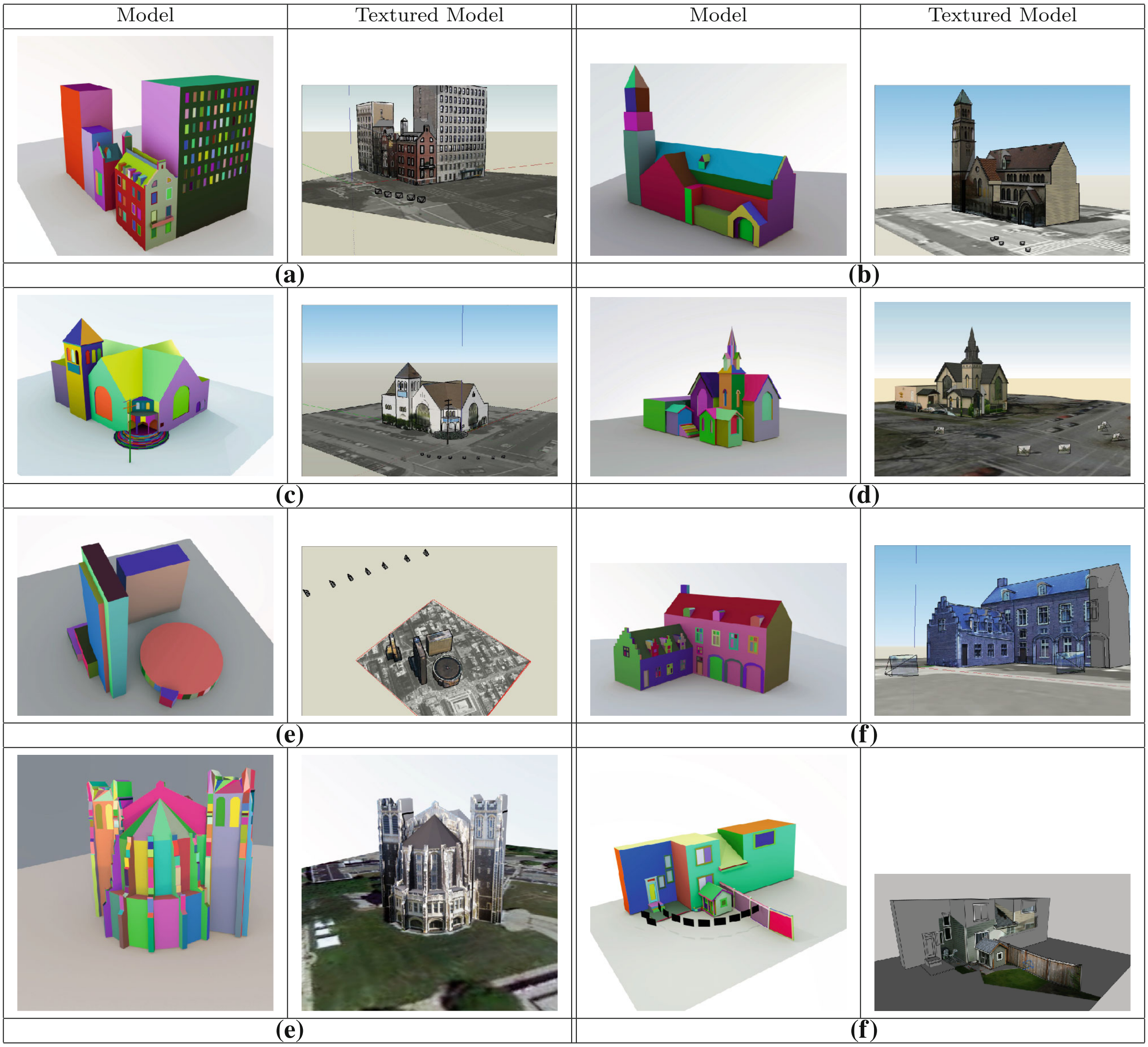


Fig. 10 Models uploaded on Google Earth. Notice that these lightweight models are represented with less than 100 polygons each. **a** Park Ave and 85th Street (NYC), **b** 99th Street and Amsterdam Ave. (NYC), **c** Hollywood Spanish SDA Church (Los Angeles), **d** Knox Presbyterian Church (Vancouver), **e** Madison Square Garden (NYC), **f** Leuven Castle [2], **e** Shepard Hall (CCNY), **f** Playhouse [3]

20 min, respectively. The user in this experiment is familiar with the software and its user interface. During modeling, the user records the elapsed time at each stage and captures a screen shot.

The resulting files are very compact, and the models, on average, have 50–100 polygons. The user can georeference a model by aligning the model footprint with the georeferenced satellite imagery from Google Earth. Figure 10 shows the result of uploaded models on Google Earth. The reconstructed buildings in Fig. 10a, b consist of only 108 and 82 polygons, respectively, and were modeled using extrusions, 2D offsets, and a few taper to point/line operations.

Our system can model buildings with non-planar facades. The user draws 2D profiles using arc, spline, line tools and then extrudes them to proper heights. Figure 11 shows a model of the Guggenheim Museum (NYC). The inputs consist of only three images. Our SfM module was able to find camera poses based on the features from the *neighboring* buildings. However, the Guggenheim Museum itself has no texture and therefore no semi-dense point cloud, as used in [4], could be extracted from this building. Furthermore, interactive modeling systems that depend on vanishing points and lines [31] will fail to model this building because few such features can be extracted reliably. Finally, the target building has a curved profile so the interactive modeling tools that are tuned to extract large planes or vanishing lines [4,31] would fail as well.

We have compared our reconstruction results with fully automatic commercial urban reconstruction products such as Agisoft PhotoScan and Pix4D Mapper. Those systems use input photographs without any user interaction to generate a dense mesh based on SfM and dense multiview stereo. Figure 12 shows the reconstructed models of the Hollywood Spanish SDA Church (Los Angeles). Notice that the meshes are very noisy and of poor quality. This is in contrast to the clean lightweight model produced by our semi-automatic PhotoSketch system, as shown in Fig. 12c. The automatic methods produced their results in 5 min, while the user spent 30 min to produce the model using PhotoSketch. However, given the poor quality of the automatic results, considerable time would need to be added to generate a lightweight watertight crisp model as shown in Fig. 12c.

While camera pose recovery and sparse point cloud generation are computed automatically in our system, the user interacts with our push–pull system to create models that are volumetric and watertight. However, the results of automatic systems are only thin shell meshes with a lot of holes. To achieve a watertight model with automatic systems, they would require a large number of photographs from every angle to cover all views of the target building. An additional problem of fully automatic commercial products is the unwanted modeled objects that are not part of the target building, such as vegetation, street signs, and cars. They are all connected as a single mesh with holes, and even part of the sky has leaked into the modeled mesh. Therefore, fully automatic methods to create a model from these meshes require a great deal of cumbersome editing, sophisticated segmentation, and hole-filling operations to clean and simplify the mesh.

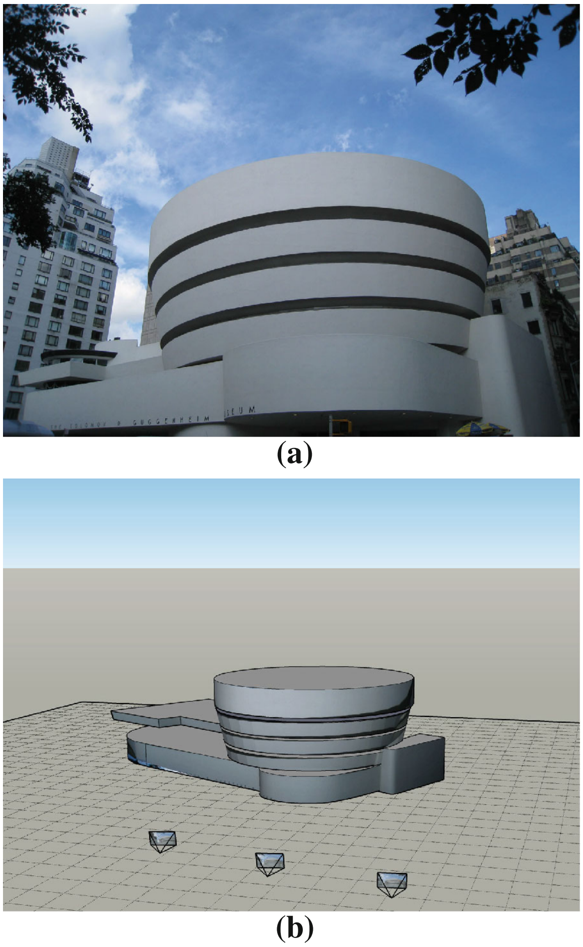


Fig. 11 Our system can model buildings with non-planar facades. **a** The Guggenheim Museum and **b** the resulting model with three displayed camera frustums

5 Conclusion

We have developed an easy-to-use photocentric 3D modeling tool for urban areas. The contribution of this work is that it merges the benefits of automatic feature extraction, multiview geometry, an intuitive sketching interface, and dynamic texture mapping to produce lightweight photorealistic 3D models of buildings. Users can sketch directly upon a set of overlapping photographs of existing buildings, outlining their footprints and lifting them into 3D shapes via a series of push–pull extrusion and taper operations. Once camera pose recovery is complete, any user drawing made upon one of the input images will appear properly aligned in the remaining images. This permits modeling operations to be applied and visualized from the viewpoint of any input image over a wide coverage of the scene, even in the presence of occlusions. By using dynamic texture mapping to project the photographs back onto the model in real time, any modeling errors become quickly apparent in the form of misaligned texture and geometry.

PhotoSketch is superior to other state-of-the-art interactive modeling systems, such as those proposed by Arikan et al. [4] and Sinha et al. [31]. Although we share the same input as [31], they require the automatic extraction of vanishing points and lines to guide the user in modeling planar facades. In addition to being error prone, the computation of vanishing lines makes their approach unsuitable for modeling buildings with any curved profiles such as those shown in Figs. 8 and 11, which we can readily handle.

In the work of [4], the proposed method requires the generation of a dense 3D point cloud from multiview stereo, akin to the fully automatic techniques shown in Fig. 12. However, they perform planar segmentation on this point cloud and then snap the resulting planes using optimization techniques to generate a rough model. The user must then interactively

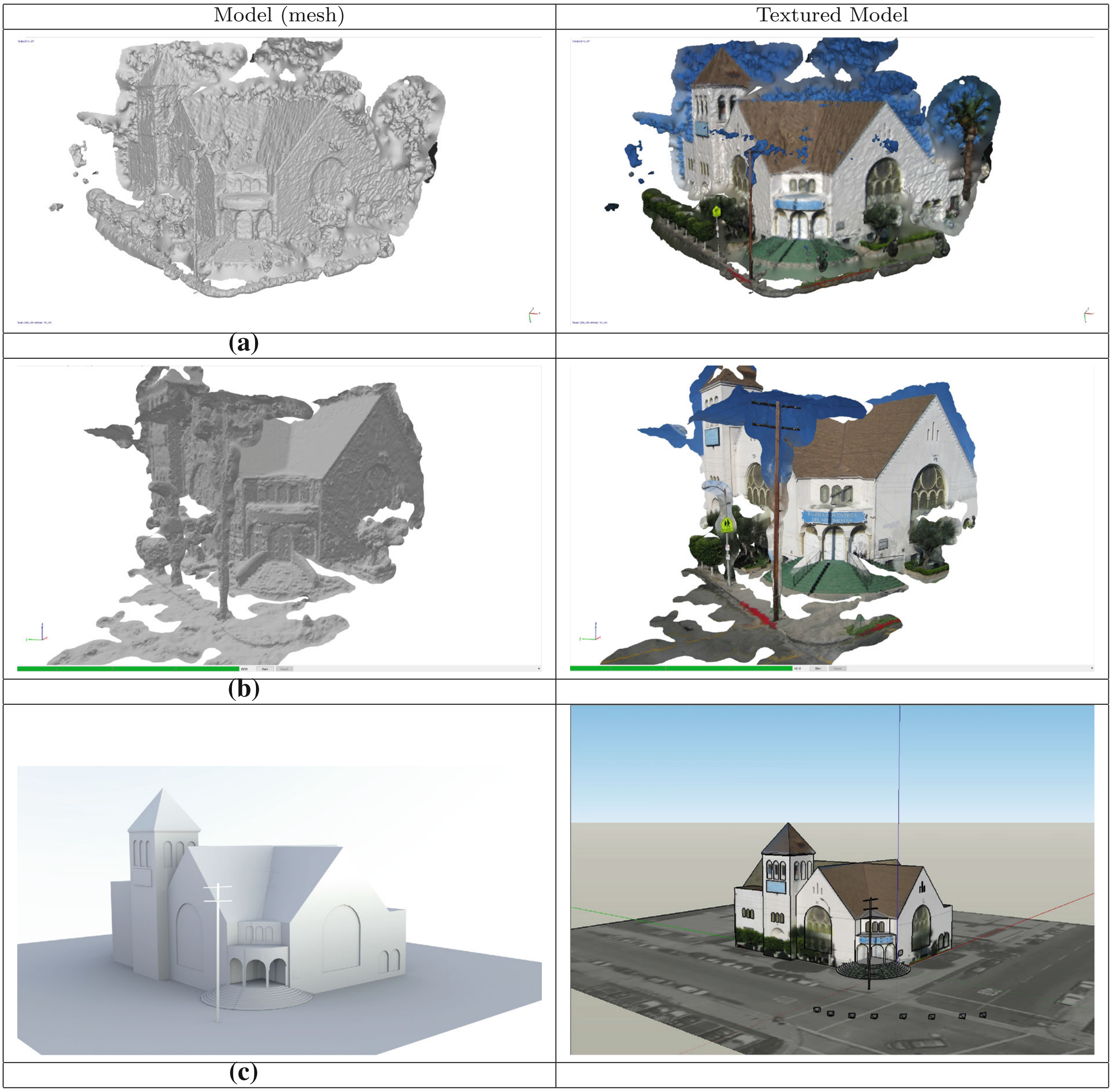


Fig. 12 The output of fully automatic commercial systems such as **a** Agisoft PhotoScan and **b** Pix4D Mapper. **c** Lightweight output of our interactive PhotoSketch system

refine and edit this model in 3D, which is more tedious and complex than the simple push–pull interface of PhotoSketch. Furthermore, their use of superimposed dense 3D point clouds to aid the interactive modeling process presents substantial visual clutter to the user. Instead, PhotoSketch is a photocentric urban 3D modeling tool that treats photographs as tracing paper upon which 2D shapes are sketched prior to extruding them into 3D models. The very photographs that aid the modeling process automatically serve as the texture elements for them as well, thereby facilitating photorealistic visualization.

Our modeling system does not handle contemporary architectural buildings that do not follow traditional bottom-up design. This includes buildings such as the Sydney Opera House or Frank Gehry buildings such as the Walt Disney Concert Hall in Los Angeles. These types of buildings, however, constitute a tiny fraction of the urban landscape. Instead, we focus on the vast majority of buildings, including those that are targeted by procedural modeling techniques.

Acknowledgements This work was supported by a grant from the US Department of Energy (DE-NA0002492).

References

1. http://profs.sci.univr.it/~fusiello/demo/samantha/
2. http://www.cs.unc.edu/~marc/
3. http://research.microsoft.com/en-us/um/redmond/groups/ivm/PlanarStereo/supplementary/playhouse.html
4. Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S.: O-snap: optimization-based snapping for modeling architecture. ACM Trans. Graph. **32**(1), 6:1–6:15 (2013)
5. Atkinson, K.B.: Close Range Photogrammetry and Machine Vision. Whittles Publishing, Dunbeath, UK (2003)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**, 346–359 (2008)
7. Chen, T., Zhu, Z., Shamir, A., Hu, S.M., Cohen-Or, D.: 3-sweep: extracting editable objects from a single photo. ACM Trans. Graph. **32**(6), 195:1–195:10 (2013)
8. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. Comput. Graph. Proc. SIGGRAPH '96 **30**, 11–20 (1996)
9. El-Hakim, S., Whiting, E., Gonzo, L.: 3D modeling with reusable and integrated building blocks. In: Conference on Optical 3D Measurement Techniques (2005)
10. Faugeras, O., Laveau, S., Robert, L., Csurka, G., Zeller, C.: 3D reconstruction of urban scenes from sequences of images. In: Gruen, A., Kuebler, O., Agouris, P. (eds.) Automatic Extraction of Man-Made Objects from Aerial and Space Images. Birkhauser, Basel, Switzerland (1995)
11. Faugeras, O., Luong, Q.T., Papadopoulou, T.: The Geometry of Multiple Images. MIT Press, Cambridge, MA (2001)
12. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1–8 (2007)
13. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
14. van den Hengel, A., Dick, A.R., Thormählen, T., Ward, B., Torr, P.H.: Videotrace: rapid interactive scene modelling from video. ACM Trans. Graph. Proc. SIGGRAPH '07 **26**(3), 86 (2007)
15. Hou, F., Qin, H., Qi, Y.: Procedure-based component and architecture modeling from a single image. Vis. Comput. **32**(2), 151–166 (2016)
16. Li, M., Nan, L., Liu, S.: Fitting boxes to Manhattan scenes using linear integer programming. Int. J. Digit. Earth **9**, 806–817 (2016)
17. Li, W., Wolberg, G., Zokai, S.: Lightweight 3d modeling of urban buildings from range data. In: 3DIMPVT, pp. 124–131 (2011)
18. Liu, L., Stamos, I., Yu, G., Wolberg, G., Zokai, S.: Multiview geometry for texture mapping 2D images onto 3D range data. IEEE Conference Computer Vision and Pattern Recognition (CVPR) pp. 2293–2300 (2006)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
20. Luhmann, T., Robson, S., Kyle, S., Harley, I.: Close Range Photogrammetry: Principles, Techniques and Applications. Wiley, Hoboken (2006)
21. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3-D Vision: From Images to Geometric Models. Springer, Berlin (2004)
22. Mathias, M., Martinović, A., Weissenberg, J., Van Gool, L.: Procedural 3d building reconstruction using shape grammars and detectors. In: 3DIMPVT, pp. 304–311 (2011)
23. Moulon, P., Monasse, P., Marlet, R., Others: Openmvg. an open multiple view geometry library. https://github.com/openMVG/openMVG
24. Müller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V.: Procedural modeling of buildings. ACM Trans. Graph. Proc. SIGGRAPH '06 **25**(3), 614–623 (2006)
25. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM Trans. Graph. **26**(3), 85 (2007)
26. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., van Gool, L., Purgathofer, W.: A survey of urban reconstruction. Comput. Graph. Forum **32**(6), 146–177 (2013)
27. Nan, L., Jiang, C., Ghanem, B., Wonka, P.: Template assembly for detailed urban reconstruction. Comput. Graph. Forum **34**, 217–228 (2015)
28. Parish, Y.I.H., Müller, P.: Procedural modeling of cities. Comput. Graph. (Proc. SIGGRAPH '01) pp. 301–308 (2001)
29. Remondino, F., El-Hakim, S.: Image-based 3d modelling : a review. Photogramm. Rec. **21**, 269–291 (2006)
30. Samavati, F., Runions, A.: Interactive 3D content modeling for digital earth. Vis. Comput. **32**(10), 1293–1309 (2016)
31. Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3D architectural modeling from unordered photo collections. In: SIGGRAPH Asia '08, pp. 159:1–159:10 (2008)
32. Snavely, N.: http://www.cs.cornell.edu/~snavely/bundler/
33. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. ACM Trans. Graph. Proc. SIGGRAPH '06 **25**(3), 835–846 (2006)
34. Stamos, I., Allen, P.K.: Automatic registration of 2D with 3D imagery in urban environments. In: Proceedings of International Conference On Computer Vision (ICCV) pp. 731–737 (2001)
35. Stamos, I., Allen, P.K.: Geometry and texture recovery of scenes of large scale. Comput. Vis. Image Underst. **88**(2), 94–118 (2002)
36. Stamos, I., Liu, L., Chen, C., Wolberg, G., Yu, G., Zokai, S.: Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. Int. J. Comput. Vis. **78**(2), 237–260 (2007)
37. Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A.: Samantha: Structure-and-motion pipeline on a hierarchical cluster tree. http://www.diegm.uniud.it/fusiello/demo/samantha/
38. Vanegas, C.A., Aliaga, D.G., Beneš, B.: Building reconstruction using Manhattan-world grammars. Comput. Vis. Pattern Recognit. **0**, 358–365 (2010)
39. Verdie, Y., Lafarge, F., Alliez, P.: Lod generation for urban scenes. ACM Trans. Graph. **34**(3), 30:1–30:14 (2015)
40. Werner, T., Schaffalitzky, F., Zisserman, A.: Automated architecture reconstruction from close-range photogrammetry. In: Proceedings of CIPA 2001 International Symposium
41. Wu, C.: Visualsfm. http://ccwu.me/vsfm
42. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Schematic surface reconstruction. In: Proceedings of IEEE CVPR pp. 1498–1505 (2012)
43. Wu, F., Yan, D.M., Dong, W., Zhang, X., Wonka, P.: Inverse procedural modeling of facade layouts. ACM Trans. Graph. **33**(4), 121:1–121:10 (2014)
44. Zheng, Y., Chen, X., Cheng, M.M., Zhou, K., Hu, S.M., Mitra, N.J.: Interactive images: cuboid proxies for smart image manipulation. ACM Trans. Graph. **31**(4), 99:1–99:11 (2012)

George Wolberg is a Professor of Computer Science at the City College of New York. He received his B.S. and M.S. degrees in Electrical Engineering from Cooper Union in 1985 and his Ph.D. degree in Computer Science from Columbia University in 1990. He was an early pioneer of image morphing and has conducted research on warping, interpolation, registration, 3D reconstruction, and structure from motion. Prof. Wolberg is the recipient of an NSF Presidential Young Investigator Award, CCNY Outstanding Teaching Award, and NYC Mayor's Award for Excellence in Science and Technology. He is the author of Digital Image Warping, the first comprehensive monograph on image warping and morphing.

Siavash Zokai received the B.S. degree in Electrical Engineering from Sharif University of Technology in 1990, the M.S. degree in Computer Science from the City College of New York/CUNY in 1997, and the Ph.D. degree in Computer Science from the City University of New York in 2004. During 2002 and 2003, he was an intern and consultant in the Imaging and Visualization Department at Siemens Corporate Research in Princeton, New Jersey. Dr. Zokai is currently a Research Scientist at Brainstorm Technology LLC in New York City. His research interests include image registration, 3D photography, and augmented reality.