3D-Modeling by Ortho-Image Generation from Image Sequences

Thorsten Thormählen Hans-Peter Seidel
Max Planck Institute for Computer Science*, Saarbrücken, Germany
http://www.mpi-inf.mpg.de

Figure 1: *Left to right: frames from the input video sequence; ortho-images that are automatically assembled out of a large number of different input frames; final model that is modeled manually in a 3D modeling package by using the ortho-images as blueprints.*

Abstract

A semi-automatic approach is presented that enables the generation of a high-quality 3D model of a static object from an image sequence that was taken by a moving, uncalibrated consumer camera. A bounding box is placed around the object, and orthographic projections onto the sides of the bounding box are automatically generated out of the image sequence. These ortho-images can be imported as background maps in the orthographic views (e.g., the top, side, and front view) of any modeling package. Modelers can now use these ortho-images to guide their modeling by tracing the shape of the object over the ortho-images. This greatly improves the accuracy and efficiency of the manual modeling process. An additional advantage over existing semi-automatic systems is that modelers can use the modeling package that they are trained in and can thereby increase their productivity by applying the advanced modeling features the package offers. The results presented show that accurate 3D models can even be generated for translucent or specular surfaces, and the approach is therefore still applicable in cases where today's fully automatic image-based approaches or laser scanners would fail.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shape; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Modeling packages

Keywords: Image-Based Modelling, Image-Based Rendering, Structure-from-Motion

*This work has been partially funded by the Max Planck Center for Visual Computing and Communication (BMBF-FKZ01IMC01).

1 Introduction

3D modelers are often given the task of generating a 3D model of a real object, e.g., during the production of video games or special effects in movies.

A common approach is to use a laser scanner or structured light scanner. These scanners usually give very good surface details [Levoy et al. 2000], but they are still quite expensive and have problems with reflective surfaces, translucent surfaces, dark surfaces that do not reflect the laser, or colorful surfaces that make it hard to analyze the projected light patterns.

A less expensive approach is to take a series of images or a video of the real object and try to recover the 3D information out of these images. This approach is accessible to everybody with a consumer camera. In recent years, systems have been developed that can automatically generate a 3D model from a captured image sequence. In [Niem 1999], the extrinsic and intrinsic camera parameters associated with each input image are recovered with a calibration ring, which is placed around the object. The 3D model is then estimated by a shape-from-silhouette approach. However, the images have to be taken in front of a blue-screen because otherwise the user would have to generate the silhouettes for each image manually. Structure-from-motion approaches, such as [Pollefeys et al. 2004], do not need a calibration object or blue-screen. First, feature points are tracked over the image sequence. By analyzing the feature tracks, structure-from-motion approaches can estimate the camera parameters for each input image and the 3D coordinates of each feature track. This step is called camera tracking, and several commercial and free [Thormählen 2006] camera trackers are available. After camera tracking, the 3D model is a sparse point cloud and Pollefeys et al. applied a multi-view stereo algorithm to recover a dense 3D surface model. However, multi-view stereo approaches are vulnerable to a lack of discernible features on the real surface, ambiguities in the image data, and they have problems with specular and translucent surfaces. Furthermore, automatically generated 3D models often look either noisy or overly smoothed.

Semi-automatic systems can overcome these difficulties through manual intervention. In Photomodeler [Eos Systems 2005] or the Facade system [Taylor et al. 1996], for example, the user has to manually mark corresponding points in every image of the se-

quence, which is very time-consuming. A quicker option is the Videotrace system [van den Hengel et al. 2007], which makes use of automatic camera tracking information. With Videotrace it is therefore often sufficient to sketch the shape of the object surface to be modeled only over one frame of the image sequence. A disadvantage of Videotrace, however, is that 3D modelers have to model the object in the Videotrace system with its limited tools and can no longer use their modeling package of choice. This slows down productivity because 3D modelers are usually very skilled in one specific modeling package and familiar with the tools and features offered by that package.

In this paper, we present a novel approach that allows fast interactive generation of 3D models from image sequences and allows modelers to stick to their modeling package of choice. Figure 2 shows an overview of the workflow of the novel approach. First, the camera parameters for each input image are estimated by automatic camera tracking. Afterwards, approaches from image-based rendering are used to generate an orthographic projection on a bounding box that is placed around the object. These ortho-images can be imported as background maps in the orthographic views of any modeling package (e.g., the top, side, and front view). Now, modelers can use the ortho-images to guide their modeling with the familiar tools of their modeling package. Thereby, they can use all the advanced features that the modeling package has to offer, such as spline modeling or subdivision methods. Because of the orthographic projection, the 3D information is directly given by the 2D user interactions in the orthographic views. This greatly improves the accuracy and efficiency of the manual modeling process. The approach is capable of handling not only diffuse surfaces but even translucent or specular surfaces and is therefore still applicable where today's laser scanners or fully automatic image-based approaches would generate inaccurate results.

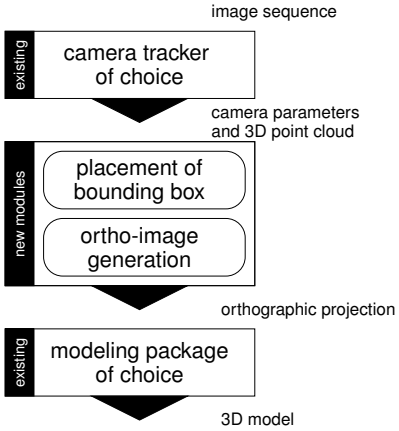

Figure 2: *Workflow overview*

As can be seen in Fig. 2, the two new modules in the workflow are the placement of the bounding box and the ortho-image generation. These two modules are described in the following two sections. Section 4 presents results, and the paper ends with a conclusion.

2 Placement of the Bounding Box

The bounding box, which will be later used for ortho-image generation, can usually not be chosen arbitrarily. In cases where there are symmetries in the object, a correct bounding box can save a lot of work in the modeling step. Therefore, the bounding box is placed interactively using the camera parameters, which were estimated in the previous automatic camera tracking step.

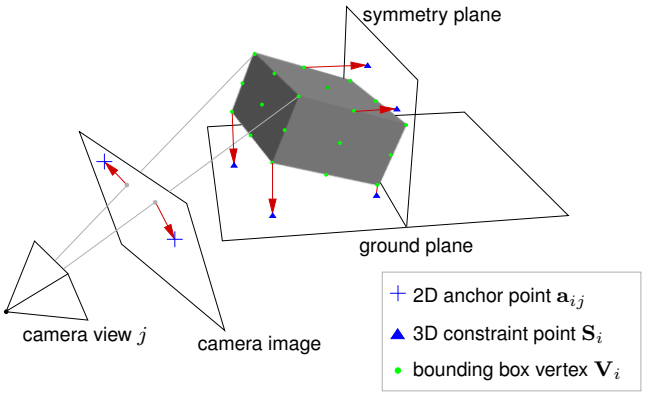

Figure 3: *Interactive placement of a bounding box with 2D anchor points in the camera image and additional 3D constraints. In this example, two 3D constraints are given, namely that the bounding box should be located on top of the ground plane and that the symmetry plane should cut exactly through the middle of the bounding box.*

As shown in Fig. 3, the bounding box is represented by 3D vertices $\mathbf{V}_i$ on its 6 surfaces. Similar to the approach described in [Gibson et al. 2003], the position, orientation, and anisotropic size of the bounding box is estimated by a non-linear optimization algorithm, which attempts to match user-specified 2D anchors $\mathbf{a}_{ij}$ for the projection of a vertex $\mathbf{V}_i$ in the camera view j. The user can drag the 2D anchors with the mouse over the image plane while the optimization is performed at interactive speed. However, for this application, the approach by Gibson et al. must be extended so that additional 3D constraints can be optimized. These 3D constraints can, for instance, enforce that a vertex $\mathbf{V}_i$ lies on a ground plane and/or on one or multiple symmetry planes.

If the user has specified N 2D anchors and M 3D constraints, the non-linear objective function is given by

$$\min_{\mathtt{H}} \sum_{N} d(\mathbf{a}_{ij}, \mathtt{P}_j \, \mathtt{H} \, \mathbf{V}_i) + \lambda \sum_{M} D(\mathbf{S}_i, \mathtt{H} \, \mathbf{V}_i) \quad , \qquad (1)$$

whereby the 2D anchors $\mathbf{a}_{ij} = (x, y, 1)^\top$ and 3D vertices $\mathbf{V_i} = (X, Y, Z, 1)^\top$ are given in homogeneous coordinates; the 4×4 matrix $\mathtt{H}$ describes the anisotropic similarity transformation, which transforms the 3D vertices $\mathbf{V_i}$ from the local coordinate system of the bounding box to the global coordinate system; $\mathtt{P}_j$ is the 3×4 camera matrix of camera view j; λ is the weighting factor of the 3D constraints; and $d(\ldots)$ and $D(\ldots)$ are the Euclidean distances of homogeneous points in 2D and 3D, respectively. The 3D points $\mathbf{S}_i$ are given by the 3D constraints, e.g., $\mathbf{S}_i$ can be the closest perpendicular point to $\mathbf{V}_i$ on a ground plane or on a symmetry plane, as shown in Fig. 3. The anisotropic similarity transformation $\mathtt{H}$ has 9 parameters, 3 each for position, rotation, and anisotropic scaling. These parameters are optimized in real-time by a Levenberg-Marquardt optimizer, which minimizes Eq. (1).

Alternatively, the user could select points that belong to the object of interest out of the 3D point cloud, which was generated in the automatic camera tracking step. An optimization algorithm could then automatically calculate the minimal bounding box for the selected 3D points under the additional 3D constraints. However, our experiments showed that the presented method of interactively moving 2D anchors is equally fast and more robust in practice, especially in cases where there are no or few 3D points on some parts of the objects.

3 Ortho-Image Generation

Once the bounding box is defined, an ortho-image is generated for each of the six sides of the bounding box with techniques from image-based rendering [Shum and Kang 2000]. The term 'ortho-image' or 'ortho-photo' is often used in photogrammetry literature, e.g., [Habib et al. 2007; Karras et al. 2007], where ortho-images are especially useful as texture maps for 3D terrain.

3.1 Ortho-images without geometry

If many camera views are available, ortho-images can be rendered without the knowledge of the object's 3D geometry. This is especially useful for transparent or translucent objects, where it is difficult to estimate the 3D geometry from the image sequence.

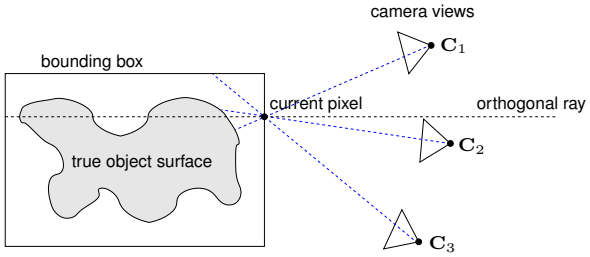

Figure 4: *Generation of an ortho-image without knowledge of the true 3D geometry.*

To determine the color intensity for a pixel of the ortho-image, the orthogonal ray through that pixel is calculated (see Fig. 4). Then, for each camera view j, the angle $\gamma_{\mathbf{C}_j}$ between the orthogonal ray and the ray from the camera center $\mathbf{C}_j$ to the pixel of the ortho-image is calculated. The camera view corresponding to the smallest of these angels, i.e. $\min\{\gamma_{\mathbf{C}_j}\}$, is used to determine the color intensity for that pixel. As can be seen in Fig. 4, a large $\min\{\gamma_{\mathbf{C}_j}\}$ can cause an incorrect color intensity. Therefore, pixels are processed only if $\min\{\gamma_{\mathbf{C}_j}\} < \tau$. We set τ to 2° in our experiments.

3.2 Ortho-images with approximate geometry

If less camera views are available, the smallest angle, $\min\{\gamma_{\mathbf{C}_j}\}$, for a pixel is often larger than the threshold τ. However, if the approximate geometry of the object is known, the correct color intensity for that pixel can be retrieved. As before, the camera view with the smallest angle is selected, but now, as illustrated in Fig. 5, the color intensity of the pixel is determined by projecting a ray from the camera center $\mathbf{C}_j$ to the point where the approximate surface reconstruction and the orthogonal ray intersect.

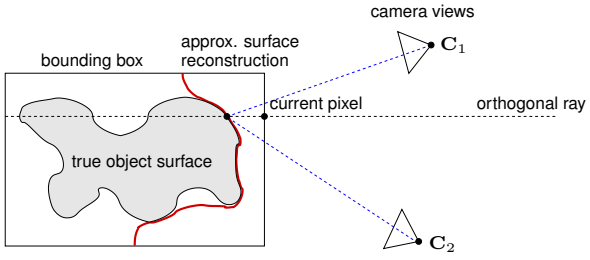

Figure 5: *Generation of an ortho-image with knowledge of the approximate 3D geometry.*

In general, the approximate surface reconstruction is not known and must be estimated from the image sequence. The following describes how the graph-cut approach for multi-view stereo in [Vogiatzis et al. 2005] is adapted and extended.

First, appropriate camera views must be selected. A large parallax between the camera views improves the accuracy of the stereo algorithm. On the other hand, a large parallax causes more occlusions and more perspective distortions between the images, which makes matching of image content harder. Out of all available camera views, a subset of views is selected for which the entire currently generated ortho-image lies within their viewing frustum, and the angle between the viewing direction and the surface normal of the ortho-image is less than $\alpha = 30^\circ$. Out of this subset, those four camera views that have a maximal parallax between themselves are chosen for stereo matching. By selecting only very few camera views, we speed up the depth computation, and a larger number of views does not usually improve the result significantly.

The space of the bounding box is quantized into voxels on a regular grid. Typically, a total of $128 \times 128 \times 512$ voxels are used, whereby 512 voxels are used in the normal direction of the currently generated ortho-image. For each of the voxels, a photo-consistency score is calculated with a window size of 7×7 pixels. The window positions in the four selected camera images are determined by projecting the center of the voxel into the four camera views.

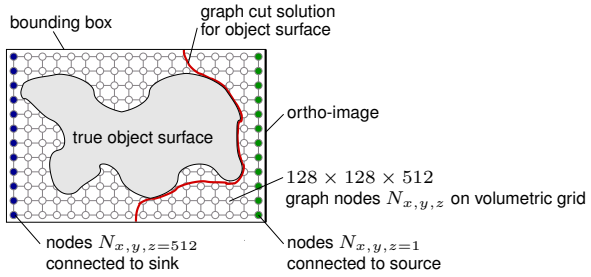

Figure 6: *Generation of a 3D surface reconstruction using a graph-cut approach for multi-view stereo*

A graph is built, whereby for each voxel, a node $N_{x,y,z}$ is generated that is connected by six edges to its six direct neighbor nodes in the voxel grid. The edge weights are chosen according to the photo-consistency score, whereby a high photo-consistency corresponds to a low edge weight (see [Vogiatzis et al. 2005] for details).

As shown in Fig. 6, all nodes $N_{x,y,z=1}$ are connected to the source, and all nodes $N_{x,y,z=512}$ are connected to the sink. Now, a globally optimal solution for the 3D object surface can be found in polynomial computing time using graph-cuts. The graph-cut algorithm finds a cut through the edges of the graph so that the nodes connected to the source are separated from those connected to the sink and the sum of cut edge weights is minimal. Therefore, the graph-cut solution corresponds to the 3D surface reconstruction with the highest photo-consistency under a smoothness constraint. To control the smoothness of the 3D surface reconstruction, we can multiply all edge weights in the normal direction of the ortho-image by a factor. A larger factor results in a smoother 3D surface reconstruction because it makes cuts through edges in normal direction more costly.

The 3D point cloud that is produced by the camera tracking step (see Fig. 2) is generated from 2D feature points that were tracked consistently over multiple images of the sequence. It is important to incorporate this reliable 3D information given by the 3D point cloud into the graph-cut surface estimation process. Only those 3D points are used that are visible in the current ortho-image. Each node $N_{x=p_x,y=p_y,z=p_z}$ in the graph that has a 3D point $(p_x, p_y, p_z)^\top$ in its voxel volume is connected to the source. The neighboring node

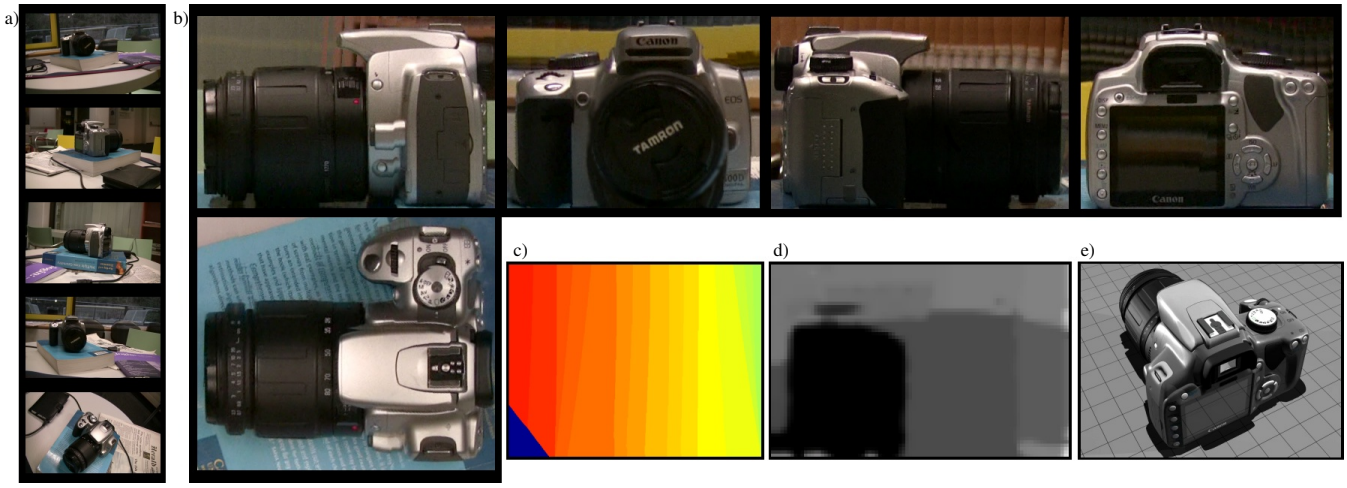

Figure 7: *Reconstruction of a Canon EOS 400D camera: a) frames from the input video sequence; b) generated ortho-images for the right, front, left, back, and top view; c) color-coded index of the video frame that is used to generate that pixel of the front view; d) grayscale-coded depth map for the left view; e) final 3D model that is modeled manually in a 3D modeling package by using the ortho-images as blueprints*[1].

$N_{x=p_x,y=p_y,z=p_z+1}$ in the normal direction of the ortho-image is connected to the sink. Additionally, all nodes $N_{x=p_x,y=p_y,z<p_z}$ are connected to the source, and all nodes $N_{x=p_x,y=p_y,z>p_z+1}$ are connected to the sink, and their edge weights are set to a very high value. This enforces that the graph-cut solution must cut through the edge between the two nodes $N_{x=p_x,y=p_y,z=p_z}$ and $N_{x=p_x,y=p_y,z=p_z+1}$ and thereby must fulfill the additional constraint given by the position of the 3D point.

4 Results

The presented approach has been applied to a variety of image sequences with different challenges. In the following, three examples are presented. These examples are also shown in the video provided with this paper.

The first example, the reconstruction of a Canon EOS 400D camera, is shown in Fig. 7. The input sequence is taken with a consumer HDV video camera and has a total length of 693 frames. The camera parameters and 3D point cloud are estimated with the 'Voodoo Camera Tracker' software [Thormählen 2006]. A few 3D points that lie on the surface of the depicted book are selected, and a plane is estimated through the selected 3D points. In an analogous manner, another plane is estimated that represents the back plane of the Canon EOS 400D camera. Now, the bounding box is placed interactively as described in section 2, whereby the two estimated planes are applied as 3D constraints for the location of the bounding box. Since the HDV video camera is moving only once around and over the object, there are not enough camera views available to apply the method of subsection 3.1. Instead, the method of subsection 3.2 is used to estimate an approximate surface reconstruction (an example of a depth map can be seen in Fig. 7), and the ortho-images for the right, back, left, front, and top view are automatically generated. The ortho-images can be easily imported as background maps into any modeling package of choice (see Fig. 8). Because the modelers can use the ortho-images with the modeling package taht they are trained in, the manual creation of a high-quality 3D model is fast, efficient, and convenient. A comparison with a laser scan in Fig. 9 shows that the created 3D model of the Canon EOS 400D camera is accurate. The laser scan and the generated 3D model can be aligned with a RMSE of 1.287 mm. Upon close inspection of the ortho-images in Fig. 7, it is possible to see that each is assembled out of a large number of input images and that there are a few image parts where wrong intensity values are assigned to pixels because of inaccurate estimation of 3D geometry. A few wrongly assigned intensity values do not usually affect the applicability of the ortho-images as blueprints for modeling. 3D modelers can usually level out these small errors by applying their high-level knowledge about how that object should appear.

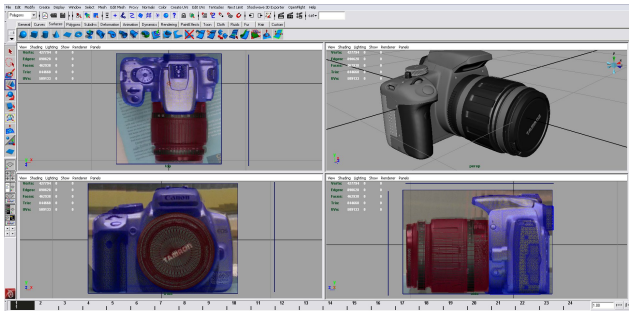

Figure 8: *The ortho-images can be imported as background maps in the orthographic views of any modeling package, here Autodesk Maya.*

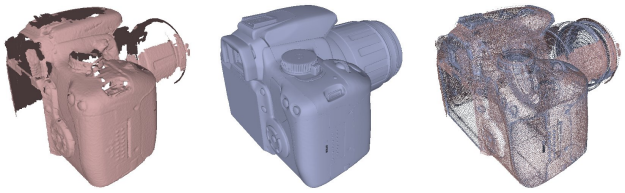

Figure 9: *Left to right: laser scan; final 3D model that is generated manually from the ortho-images; comparison of the laser scan (red points) with the final 3D model (blue points).*

As a second example, the reconstruction of a Holden Astra Twin-Top car is presented in Fig. 1. The input sequence is again taken with a consumer HDV video camera and has a total length of 597 frames. This sequence is quite different because the camera is actually not moving. The camera is mounted on a tripod and the car is moving on a large turntable. However, if the background of the

[1] Special thanks to Benjamin Waschk (http://www.janundben.de)

scene is ignored, the geometric relationship between an object on a turntable and a static camera can also be interpreted as if the object is static and the camera is orbiting around the object. Therefore, the presented approach can be applied in the same way as in the previous example. The sequence is especially challenging because of the specular paintwork and the lack of discernible features on the body of the car. This makes it extremely hard for multi-view stereo algorithms to estimate the 3D geometry correctly. Nevertheless, the approximate geometry, which is generated with the graph-cut method of subsection 3.2, is good enough to produce the ortho-images as shown in Fig. 1.

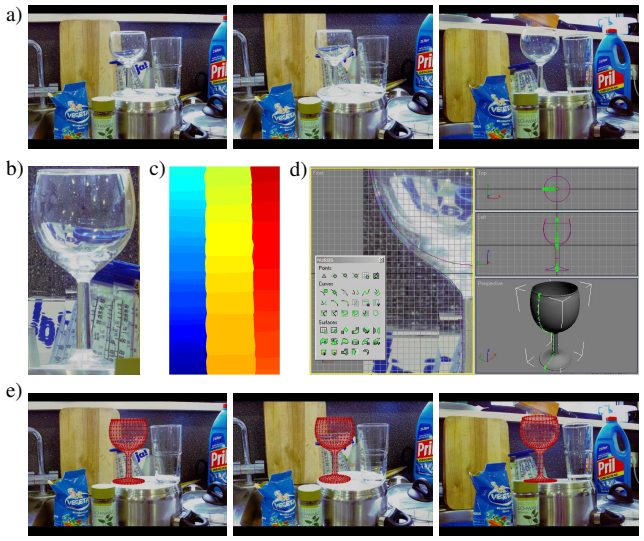

Figure 10: *Reconstructing a wine glass from an image sequence: a) original image sequence; b) ortho-image of the wine glass; c) color-coded index of the image that is used to generate that pixel of the ortho-image; d) modeling of the glass using the ortho-image and NURBS tools in Autodesk 3ds Max; e) rendering of the reconstructed wine glass into the original image sequence.*

The final example is the 3D reconstruction of a transparent wine glass. The input is now a sequence of 140 still images taken with a digital SLR camera. Because the glass has rotational symmetry, a 3D reconstruction can be generated from a single ortho-image. Without knowledge of approximate 3D geometry, the 140 camera views are sufficiently dense to generate this single ortho-image by applying the method described in section 3.1. As exhibited in Fig. 10, the ortho-image has artifacts in the center of the glass because objects and light sources from the environment are reflected differently into the camera images. However, the contour of the wine glass, which is important in the 3D modeling step, is almost perfectly preserved. After loading the ortho-image into a 3D modeling package, it takes less than a minute to generate a convincing 3D reconstruction.

5 Discussion and Conclusion

Obviously, it is possible in some cases to skip the first three workflow steps described in this paper because an ortho-image can also be approximated by taking a single image of the object with a very long tele lens. However, especially for large objects, this direct approach is not easy to apply. Between the camera and the car, a distance of $4.5\,\mathrm{m} \cdot \cot 2^\circ = 128.86\,\mathrm{m}$ is needed to generate ortho-images for a car with a side length of 4.5 meters and with a maximal error for the direction of the orthogonal ray of 2 degrees. In practice, it would also take some effort to place the camera so that neighboring ortho-images (e.g. side and front views) are taken from exactly perpendicular directions. Clearly, these difficulties can be avoided by using the workflow presented in this paper.

The presented results show that the novel semi-automatic approach described in this paper can generate high-quality and accurate 3D models from image sequences. The workflow is especially easy to apply because of its straightforward interaction with existing modeling packages. In comparison with state-of-the-art, fully automatic, image-based approaches, the presented approach allows 3D reconstruction in more situations (e.g., specular or transparent objects) and produces a reconstruction of higher quality. This higher quality is achieved at the expense of more manual intervention. However, this manual intervention is kept to a minimum and the presented approach comprises more automatic steps than other existing semi-automatic systems.

References

EOS SYSTEMS, 2005. Photomodeler: A commercial photogrammetry product http://www.photomodeler.com.

GIBSON, S., HUBBOLD, R. J., COOK, J., AND HOWARD, T. L. J. 2003. Interactive reconstruction of virtual environments from video sequences. *Computers and Graphics 27*, 293–301.

HABIB, A. F., KIM, E. M., AND KIM, C. J. 2007. New methodologies for true orthophoto generation. *Photogrammetric Engineering and Remote Sensing 73*, 1, 25–36.

KARRAS, G., GRAMMATIKOPOULOS, L., KAISPERAKIS, I., AND PETSA, E. 2007. Generation of orthoimages and perspective views with automatic visibility checking and texture blending. *Photogrammetric Engineering and Remote Sensing 73*, 4, 403–411.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project. In *Proc. ACM SIGGraph*, K. Akeley, Ed., 131–144.

NIEM, W. 1999. Automatic reconstruction of 3d objects using a mobile camera. *Image Vision Comput. 17*, 2, 125–134.

POLLEFEYS, M., GOOL, L. V., VERGAUWEN, M., VERBIEST, F., CORNELIS, K., TOPS, J., AND KOCH, R. 2004. Visual modeling with a hand-held camera. *Int. J. Comput. Vision 59*, 3, 207–232.

SHUM, H.-Y., AND KANG, S. B. 2000. A review of image-based rendering techniques. In *Proc. Visual Communications and Image Processing*, 2–13.

TAYLOR, C., DEBEVEC, P., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. ACM SIGGraph*, 11–20.

THORMÄHLEN, T. 2006. *Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge*. PhD thesis, University of Hannover, related software 'Voodoo Camera Tracker' can be downloaded from http://www.digilab.uni-hannover.de.

VAN DEN HENGEL, A., DICK, A., THORMÄHLEN, T., WARD, B., AND TORR, P. H. S. 2007. Videotrace: rapid interactive scene modelling from video. In *Proc. ACM SIGGraph*, 86–90.

VOGIATZIS, G., TORR, P. H. S., AND CIPOLLA, R. 2005. Multi-view stereo via volumetric graph-cuts. In *Proc. IEEE Computer Vision and Pattern Recognition*.