TIME-OF-FLIGHT BASED SCENE RECONSTRUCTION WITH A MESH PROCESSING TOOL FOR MODEL BASED CAMERA TRACKING

Svenja Kahn, Harald Wuest
Fraunhofer Institute for Computer Graphics Research, Fraunhoferstr. 5, 64283 Darmstadt, Germany
svenja.kahn@igd.fraunhofer.de, harald.wuest@igd.fraunhofer.de

Dieter W. Fellner
Interactive Graphics Systems Group, Technische Universität Darmstadt, Fraunhoferstr. 5, 64283 Darmstadt, Germany
d.fellner@gris.informatik.tu-darmstadt.de

Keywords: Markerless camera tracking, model-based tracking, 3d reconstruction, time-of-flight, sensor fusion, augmented reality

Abstract: The most challenging algorithmical task for markerless Augmented Reality applications is the robust estimation of the camera pose. With a given 3D model of a scene the camera pose can be estimated via model-based camera tracking without the need to manipulate the scene with fiducial markers. Up to now, the bottleneck of model-based camera tracking is the availability of such a 3D model. Recently time-of-flight cameras were developed which acquire depth images in real time. With a sensor fusion approach combining the color data of a 2D color camera and the 3D measurements of a time-of-flight camera we acquire a textured 3D model of a scene. We propose a semi-manual reconstruction step in which the alignment of several submeshes with a mesh processing tool is supervised by the user to ensure a correct alignment. The evaluation of our approach shows its applicability for reconstructing a 3D model which is suitable for model-based camera tracking even for objects which are difficult to measure reliably with a time-of-flight camera due to their demanding surface characteristics.

1 MOTIVATION

Markerless camera pose estimation is one of the most callenging aspects of Augmented Reality applications. Reliable and often used methods to estimate the camera pose without using markers are model-based tracking approaches (Lepetit and Fua, 2005). Up to now, the availability of a 3D model of a scene is the bottleneck of model-based approaches. For many scenes a 3D model is either not available or outdated, so a 3D model of the scene has to be reconstructed. The manual creation of a 3D model is very time-consuming. This is why there is a strong need for automatic or semi-automatic acquisition of a 3D model.

We present a semi-automatic approach to reconstruct a scene model for model-based camera tracking which is based on a sensor fusion approach of a time-of-flight camera (which captures depth images in real time) and a color camera as well as the use of a mesh processing tool. Our approach comprises three main tasks which are depicted in figure 1 and figure 2: First a colored 3D mesh is created from the data of both sensors (see section 3). Then a 3D model of the scene is constructed from one or several such 3D meshes (section 4). We propose to use a mesh processing tool for this task because the 3D measurements of a time-of-flight camera suffer from strong noise and non-systematic errors. Thus a fully automatic mesh alignment often converges to an incorrect local minimum which does not correspond to the correct alignment of the submeshes. After reconstructing a 3D model of the scene we use the 3D model for model-based markerless camera tracking (section 5). The evaluation of our approach (section 6) shows its applicability for reconstructing a 3D model which is suitable for model-based camera tracking even for objects which are difficult to measure reliably with a time-of-flight camera due to their demanding surface characteristics. In the proposed approach the scene is reconstructed in an offline preparation step and is then used for model-based camera tracking with a 2D camera. In contrast to approaches which simultaneously estimate the camera pose and the reconstruction this approach has the advantage that it is not prone

to drift in the camera pose estimation. Thus a more stable 3D reconstruction can be achieved than with online reconstruction. The user input required in the reconstruction step ensures the correctness of the reconstructed 3D model. Another advantage is that the tracking can be done with any 2D camera and no time-of-flight camera and no sensor fusion is needed for the tracking phase.

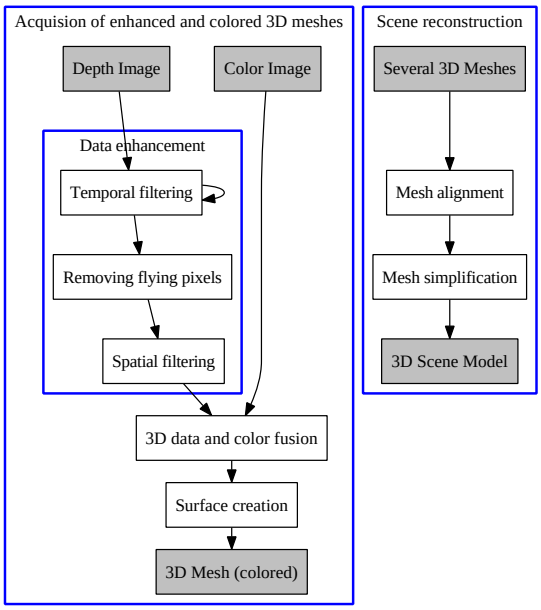

Figure 1: 3D mesh creation and scene Reconstruction. The reconstruction step is accomplished with a mesh processing tool and can thus be supervised by the user.

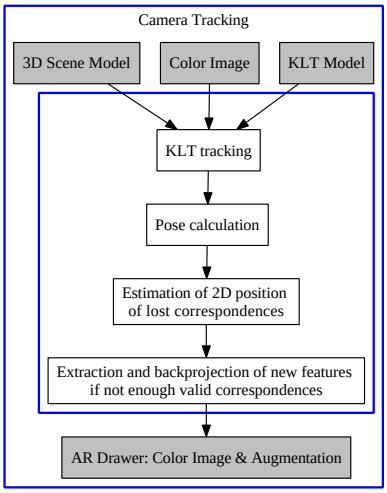

Figure 2: Camera Tracking.

Time-of-flight cameras emit near-infrared modulated light which gets reflected by the scene. The image sensor of the camera then captures the reflected light. For every pixel the distance to the scene is calculated by the phase shift of the reflected light. The distances can be transformed to cartesian coordinates, yielding a 3D measurement per pixel. We use the time-of-flight camera presented in (Oggier et al., 2006) which has a resolution of $176 * 144$ pixels. In addition to the 3D measurements the time-of-flight camera outputs an intensity image which depicts the amount of light measured by each pixel. The 3D measurements acquired by time-of-flight cameras suffer from noise and systematic errors. It is thus very important to filter and enhance the 3D measurements as described in section 3.1. With a data enhancement step based on spatial and temporal filtering we can significantly improve the quality of the 3D measurements which is important for the subsequent camera tracking step because for that purpose an as accurate as possible 3D model of the scene is needed.

2 RELATED WORK

Time-of-flight cameras are used for a wide range of tasks such as Augmented Reality occlusion calculation or user interaction. A detailed overview of the current state-of-the-art is given by (Kolb et al., 2009). In robotics time-of-flight cameras have already proven to be very useful for 3D scene reconstruction and robot pose estimation. (Prusak et al., 2008) use a time-of-flight camera for map building and pose estimation for robot navigation. They combine a time-of-flight camera with a spherical camera to create a 3D depth panorama and to estimate the position of the robot. The 3D depth panorama is created by rotating the robot. Then an occupation map is filled from which a 3D triangle mesh is finally created. (May et al., 2008) use a time-of-flight camera for simultaneous localization and mapping. In their work a merged point cloud is accumulated based on the estimated camera poses. In contrast to their approach we create the 3D mesh in an offline step and not simultaneously to the estimation of the camera pose. The reason for this is that in simultaneous localization and mapping approaches errors in the estimation of the camera pose result in errors in the 3D model reconstruction. Thus these approaches are more prone to drift than approaches in which the reconstruction phase is completed before the camera tracking is started. (Huhle et al., 2008) combine depth and color data with measurements of an inertial sensor. They describe how several colored depth images can automatically be aligned by a combined color-based and geometric registration. Most similar to our

approach is the approach presented by (Schiller et al., 2008a) who also reconstruct a 3D model for Augmented Reality camera tracking. In their work a colored depth panorama is created by mounting a time-of-flight camera and a color camera on a controllable pan-tilt unit. This is a very advantageous approach because with a controllable pan-tilt unit the camera pose of every acquired depth image is exactly known. Thus the reconstruction is very precise and no manual user interaction is needed at all. Compared to their approach our approach has the advantage that no controllable pan-tilt unit is needed and that it is not restricted to the reconstruction of 3D panoramas. By taking depth and color images from several positions we can reconstruct a full 3D model. In contrast to a 3D panorama, the camera tracking is not restricted to parts of a scene visible from a single fixed viewpoint.

3 CREATION OF AN ENHANCED AND COLORED 3D MESH

For the creation of a 3D mesh the quality of the 3D measurements needs to be enhanced. Then the enhanced 3D points get fused with the colors from the color camera. Finally a triangle mesh surface is created based on a distance criterion which is needed to prevent a surface creation over empty parts of a scene.

3.1 Preprocessing and Filtering

Due to noise and systematic errors in the depth images the depth data needs to be preprocessed before it can be used for geometric reconstruction. Depending on the surface properties, the reflectivity and the distance of the scene the measured distances can differ from the real distances by several millimeters or centimeters. The measurement accuracy depends strongly on the material of the objects in the scene. Measurements are more accurate for objects with materials that reflect light in a diffuse manner than for objects with specular surface properties. Measurements of materials that reflect only few of the light emitted by the time-of-flight camera are less accurate than 3D measurements of materials with a high reflectivity.

Temporal filtering We use a camera tripod to take several images of a scene from the same viewpoint and merge several measurements to get more reliable measurements. In our setup ten depth images are acquired from each viewpoint. The mean of these measurements is calculated for every pixel. This average value per pixel reduces the noise which is inherent in the measurements of single depth images.

Removing flying pixels Flying pixels occur at depth discontinuities, where the near-infrared light emitted by the time-of-flight camera gets reflected in part by an object in the foreground and in part by an object in the background. This effect can be seen in the left column of figure 3. In this figure the 3D measurements of a time-of-flight camera are colored according to their distance to the camera. Red 3D points are close to the camera whereas green 3D points have a medium distance and blue points are far away. The right column is the intensity image of the time-of-flight camera. The second row shows a picture detail. The flying pixels in the left column are erroneous measurements (there is no object at the position of these measurements) and must be removed. To remove flying pixels and isolated 3D measurements we apply a filter which examines the eight 3D points corresponding to the neighbours of a pixel in the 2D image and rejects pixels if less than n of their neighbours have a distance below a fixed threshold. In our setup we chose n=4 and a maximal euclidean distance of 8cm. The result of applying this filter can be seen in the second column of figure 3.

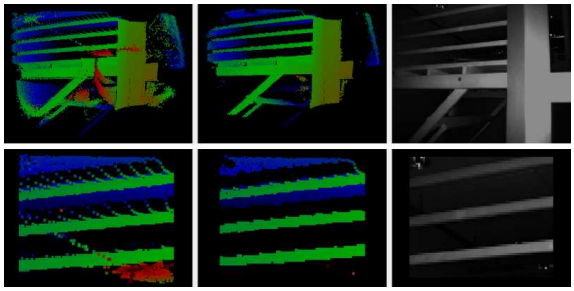

Figure 3: Removing flying pixels.

Spatial filtering To remove random measurement noise in the images while preserving sharp edges we apply a bilateral gaussian filter on the depth values. (Durand and Dorsey, 2002) describe a bilateral filter for image intensities. We adapted this filter for depth images. In contrast to a non-bilateral gaussian filter (which smoothes surfaces but does not preserve sharp edges and corners) the bilateral filter uses an additional weighting factor g for each value in the filter mask. Elements with a similar function value (in this case 3D points with a similar depth value) get a higher weighting factor g than 3D measurements whose depth values differ much. This prevents the smoothing of edges and corners. Equation 1 represents a bilateral gaussian filter for a pixel s where k(s) is a normalization term, G(p) is the gaussian kernel and g(d,n) is the additional weighting factor which

depends on the distance d between the z values of the 3D points at the pixels s and p. The 3D points (x,y,z) are in the camera coordinate system and the camera points along the z-axis.

$$z_s^* = \frac{1}{k(s)} \sum_{p \in \Omega} G(p) g(d,n) z_p \quad (1)$$

$$k(s) = \sum_{p \in \Omega} G(p) g(d,n) \quad (2)$$

We chose the function represented by equation 3 as the weighting factor g. The factor n ($n \geq 1$, in our setup n=10) sets how fast g declines if the z-values are dissimilar. If n=1 the bilateral filter corresponds to a standard, non-bilateral gaussian filter.

$$g(d,n) = \frac{1}{n^{(1+d)^n}}, \quad d = |z_p - z_s| \quad (3)$$

$$d = |z_p - z_s| \quad (4)$$

The gaussian kernel G is the standard kernel of a non-bilateral two-dimensional gaussian filter. We use a 7x7 kernel with $\sigma = 2.0$.

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (5)$$

3.2 Sensor Fusion

To fuse the depth data with color information we use a camera rig which rigidly couples the time-of-flight camera and the color camera. For the data fusion the intrinsic camera calibration matrix K of the color camera and the relative transformation $(\Delta R, \Delta t)$ between the cameras needs to be known. We use the analysis-by-synthesis approach of (Schiller et al., 2008b) to calculate the intrinsics as well as the relative transformation between the cameras.

The time-of-flight camera measures a 3D point p_t per pixel in its camera coordinate system. To combine the 3D data with the data from the color camera we need to find the corresponding color pixel p_c for every 3D measurement by the following transformations:

$$p_c = K(\Delta R p_t + \Delta t) \quad (6)$$

The 3D point is transformed from the time-of-flight camera coordinate system to the color camera coordinate system and is then projected to the image coordinate system of the color camera with the given camera calibration matrix K. To compute the correct pixel coordinates, the homogeneous image coordinates are distorted with the radial distortion parameters of the color camera. The field of view of both cameras overlaps only partially so for some 3D measurements at the margins there is no color information available. We keep uncolored parts because they are useful for the alignment step of the 3D reconstruction: Two meshes can only be aligned if they have overlapping parts. After the alignment uncolored parts can be removed where color information is available from other overlapping meshes.

3.3 Surface Generation from the Point Set

To create a surface from the measured 3D points we construct a triangle mesh whose triangles connect neighbouring points whose depth values are similar. We do not create a surface of points whose depth values differ by more than a threshold (in our setup 8cm). If there is an object in the foreground and another object in the background no triangles should be created in the space between these objects. With the threshold we also prevent a surface generation connecting measured 3D points with outliers. In the unenhanced mesh in figure 5 many holes are visible due to 3D measurements whose depth values differ too much. This is significantly improved in the enhanced mesh.

4 SCENE RECONSTRUCTION

For some tracking scenarios a single enhanced and colored 3D mesh created as described in the previous section can already suffice. If there is a need to track a larger environment several 3D meshes need to be combined to a larger 3D model of the scene. We propose to use a mesh processing tool for this task so that the user can supervise a correct alignment of the submeshes.

Figure 4: 3D model of a room reconstructed from several colored time-of-flight images. The surface of 3D measurements for which no color is available is colored in blue.

4.1 Alignment of Depth Images

For the alignment of several colored 3D meshes we use MeshLab (Cignoni et al., 2008), a powerful mesh

processing tool. Two point sets can initially be aligned manually by choosing corresponding points in the two point sets. Then the Iterative Closest Point (ICP) (Besl and McKay, 1992) algorithm is used to iteratively improve the alignment of the two meshes. A manual control of the alignment helps to ensure that it converges to the right alignment. This is particularly important for noisy and at some points still erroneous data. It is also recommandable to supervise the alignment manually because the ICP algorithm does not always deliver correct results. For example if we want to merge two views of a partially overlapping wall the ICP algorithm will slide the two parts more and more onto another. This should be supervised and eventually be corrected by the user.

4.2 Mesh Simplification

The size of the 3D model increases rapidly with every added submesh because every submesh has up to $176 * 144 = 25344$ measured 3D points. To ensure a fast processing of the reconstructed model in the tracking phase the size of the reconstructed 3D model is significantly reduced with quadric edge collapse decimation, a mesh reduction algorithm which is part of (Cignoni et al., 2008). On the right of figure 6 a simplified version of the 3D model of figure 4 can be seen. It was created by iteratively applying an edge collapsing step. The size of the 3D model was reduced by the factor 8. Whereas there was an obvious fall off in color quality (due to the fact that our reconstructed model stores the color per vertex), the geometry of the mesh was well preserved.

5 CAMERA TRACKING

The scene model reconstructed as described in the previous sections is used to estimate the camera pose with a model-based point tracker. Tracking can be done with a custom 2D camera, no additional time-of-flight camera is needed. We build on the modular system described in (Becker et al., 2007) and use a model based Lucas Kanade feature tracker (KLT). Our KLT model is similar to (Bleser et al., 2006). The tracking is initialized manually. For the initialization we select corresponding 3D points from the reconstructed scene model and 2D points from the first camera image. The initial camera pose is calculated from these correspondences. With the known initial pose features can be extracted from the current image via Shi Tomasi corner detection (Shi and Tomasi, 1994). The new features are stored in the feature map. The 3D coordinate of each feature is calculated via backprojection onto the given 3D model of the scene as described in (Bleser et al., 2006). After this first initialization of the feature map the camera can be moved and the following steps are executed for each frame (see figure 2): First the KLT features of the feature map are tracked in the current image with an optical flow tracker to get the 2D positions of the features. Then the camera pose is estimated from the 2D/3D correspondences in the feature map. The 2D coordinates of lost features are estimated by projecting their 3D coordinates into the image coordinate system with the current pose. If the number of successfully tracked features in the current frame is too low, new features are extracted and backprojected onto the 3D model of the scene.

6 RESULTS

6.1 Technical room and office

To evaluate our approach we reconstructed 3D models of a technical room and an office and used these 3D models for model-based camera tracking. The technical room was the more demanding scenario because it has a lot of surfaces which are hard to acquire reliably with a time-of-flight camera. Metallic surfaces do not have diffuse surface properties but reflect the light emitted by the time-of-flight camera in a specular way. This is why the 3D values of metallic surfaces measured by a time-of-flight camera suffer from strong errors and are thus very challenging.

Figure 5: Smoother surfaces with less gaps due to data enhancement.

In both scenarios the data enhancement step proved to be indispensable as it considerably improves the quality of the measured 3D points. Figure 5 illustrates these enhancements. The temporal and spatial smoothing significantly reduces the number of gaps of the reconstructed surface. This is important for the model based tracking because 2D features can only be tracked where 3D data is available. The reconstructed surfaces are also smoother than without the data enhancement step whereas edges are well preserved.

We overlayed the color image sequences with the reconstructed 3D model of the scene to evaluate whether the reconstructed 3D models can be successfully used for markerless camera tracking (see figure 6). The camera pose could be successfully tracked in both rooms with the reconstructed 3D models and our model-based camera tracking approach.

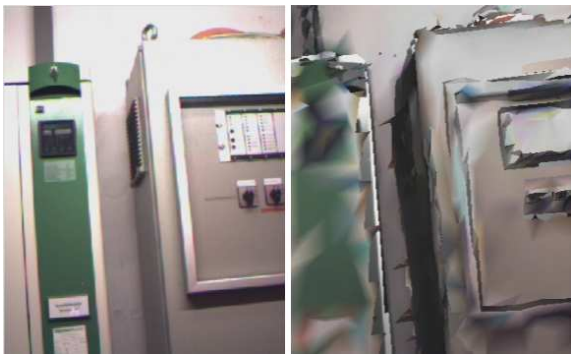

Figure 6: Left: Color Image, Right: Color Image augmented with reconstructed 3D model (strongly simplified mesh).

6.2 Industrial Object

To evaluate whether our approach can also be used to reconstruct and track a 3D model of an object whose 3D acquisition by a time-of-flight camera is very challenging we reconstructed the industrial object of figure 7. Time-of-flight data and color images taken from five different viewpoints were used to reconstruct the object. The main challenges of this object are the metallic surfaces on the front side and the open slots with metallic side walls inside the object by which the light emitted by the time-of-flight camera gets multiply reflected, resulting in unreliable and erroneous 3D measurements (see bottom of figure 7). With an accurate reference 3D model of the object (center of figure 7) we calculated the error in the 3D measurements acquired by the time-of-flight camera. The reference model and the reconstructed model were aligned with the Iterative Closest Point algorithm and then both rendered into the z-buffer of the graphics card with the same virtual extrinsic and intrinsic camera parameters. By unprojecting the values of the z-buffer we got a 3D measurement for each pixel of the two virtual images which we compared to calculate the euclidean differences between the measurements of the time-of-flight camera and the real 3D values.

The mean euclidean distance of a view of the front of the industrial object is 4.3cm. However, the differences of the reference model and the reconstructed

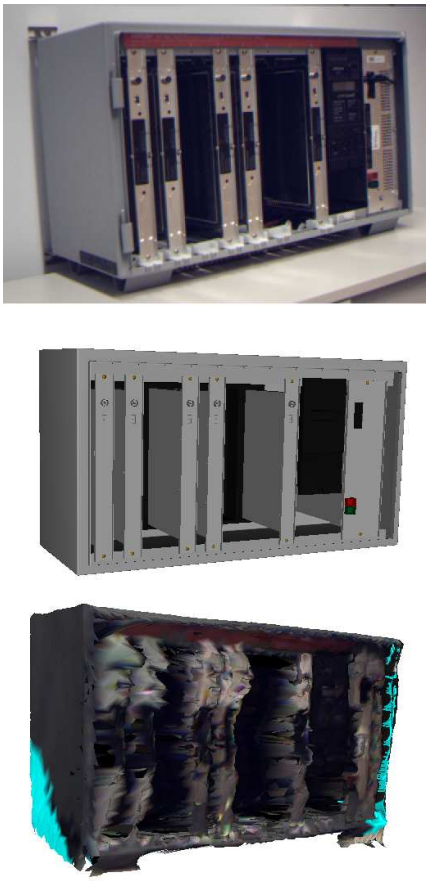

Figure 7: Top: Industrial object with metallic surfaces which are difficult to measure reliably with a time-of-flight camera. Center: Reference 3D Model. Bottom: Reconstructed 3D Model.

model differ a lot depending on their location. The red channel of the image in figure 8 shows the euclidean distances between the reference 3D model and the reconstructed 3D model. The euclidean distance in centimeters was multiplied with the factor 20 and the red value of each pixel was interpolated linearly between 0 (black, this means that at this pixel there was no difference between the reference 3D model and the reconstructed 3D model) and 255 (bright red, representing a difference of more than $255/20 = 12.75$cm). The differences between the reference model and the reconstructed model are much higher inside the object than at the front part of the object. The pixels colored in blue are measurements where there is a surface of the reconstructed model but not of the reference model (the base of the object was not modelled in the reference model). The yellow pixels show parts where no surface was created in the reconstructed 3D model due to too big differences between the distance

values of neighboured 3D measurements.

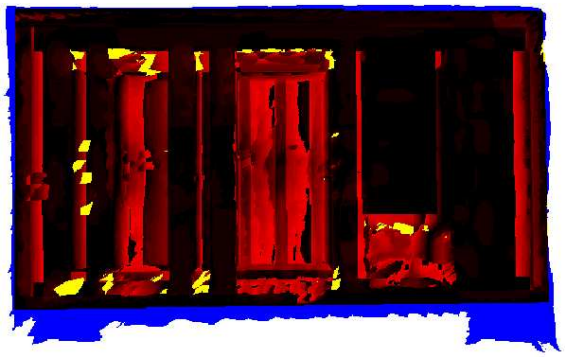

Figure 8: Color encoded differences of the reference 3D model and the reconstructed 3D model.

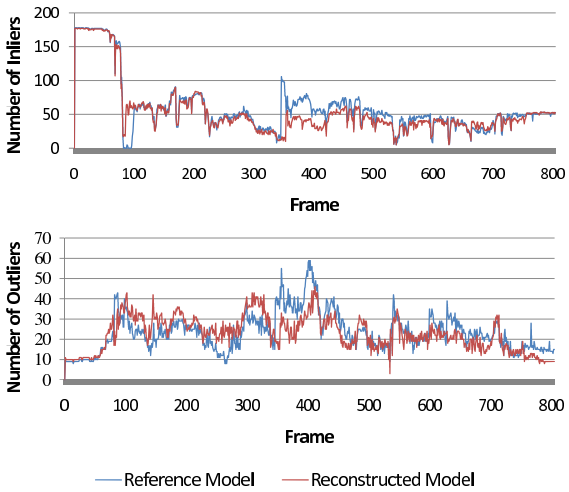

Figure 9: Number of tracking inliers and outliers when the refence model and the reconstructed model are used for tracking.

Both the reference 3D model and the reconstructed 3D model were used for model-based markerless camera tracking as described in section 5. For this purpose we used a 2D image sequence consisting of 800 frames. The (handheld) color camera was moved from a side view of the object around the object and back. Figure 9 plots the number of inliers of the tracked 2D-3D correspondences of each frame as well as the number of outliers. There are less inliers in the model based tracking with the reconstructed model than with the perfect reference model but the difference is rather small. The industrial object was successfully tracked with the reference model as well as with the reconstructed model through the whole image sequence (see figure 10). With both models there were several frames in which the augmentation of the 3D model on the image was not accurate but with both 3D models the correct tracking was recovered automatically after some frames. There were also several frames in which the tracking based on the reconstructed 3D model was more accurate than the tracking based on the reference 3D model (figure 11).

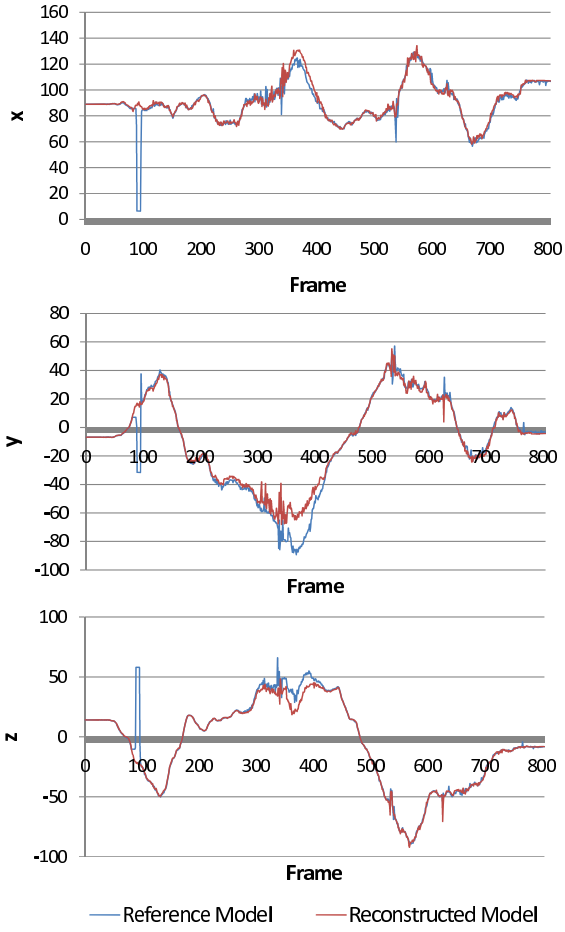

Figure 10: The camera positions calculated with the reference model and the reconstructed model.

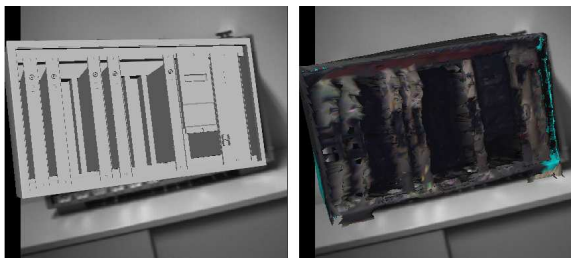

Figure 11: Frame 343: The reconstructed 3D model is more accurately tracked than the reference model.

7 CONCLUSION

We presented a scene reconstruction approach in which a time-of-flight camera and a color camera are combined in a sensor fusion approach to create textured 3D models of a scene. The 3D data quality is significantly enhanced by temporal and spatial filtering. Several such colored submeshes are aligned with a mesh processing tool. The use of a mesh processing tool has the important advantage that the correct alignment (which is difficult for 3D measurements acquired by time-of-flight cameras due to the random and systematic errors in the 3D data) can be supervised by a user. Compared to a manual creation of a 3D model the needed workload is considerably reduced. Thus the 3D model can easily be created and also easily be updated when some parts of the reconstructed scene change.

The evaluation of our approach shows that the reconstructed 3D models created with our approach can successfully be used for model based camera tracking. This is even the case for industrial objects with a metallic surface which are difficult to measure reliably with a time-of-flight camera.

So far the color information is already very useful for the reconstruction and for the initialisation of the camera tracking because it is much easier for the user to supervise a correct alignment of the submeshes if the texture of the meshes is displayed. In future work we will extend our approach by using the color information of the reconstructed 3D model not only for the reconstruction step and the camera pose initialization but also for the frame-to-frame camera tracking.

8 ACKNOWLEDGEMENTS

This work has been performed within the AvilusPlus research project granted by the German Ministry of Education and Research (BMBF). The authors wish to thank the MeshLab developers and the members of the Multimedia Information Processing Group of the University of Kiel for making their MultiCameraCalibration tool and MeshLab publicly available.

REFERENCES

Becker, M., Bleser, G., Pagani, A., Stricker, D., and Wuest, H. (2007). An architecture for prototyping and application development of visual tracking systems. In *Capture, Transmission and Display of 3D Video (Proceedings of 3DTV-CON 07 [CD-ROM])*.

Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14(2), pages 239–256.

Bleser, G., Wuest, H., and Stricker, D. (2006). Online camera pose estimation in partially known and dynamic scenes. In *ISMAR 2006: Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 56–65.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). Meshlab: an open-source mesh processing tool. In *Sixth Eurographics Italian Chapter Conference*, pages 129–136.

Durand, F. and Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 257–266.

Huhle, B., Jenke, P., and Straßer, W. (2008). On-the-fly scene acquisition with a handy multi-sensor system. *International Journal of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):255–263.

Kolb, A., Barth, E., Koch, R., and Larsen, R. (2009). Time-of-Flight Sensors in Computer Graphics. In *Proc. Eurographics (State-of-the-Art Report)*.

Lepetit, V. and Fua, P. (2005). Monocular model-based 3d tracking of rigid objects: A survey. In *Foundations and Trends in Computer Graphics and Vision*, volume 1, pages 1–89.

May, S., Droeschel, D., Holz, D., Wiesen, C., and Fuchs, S. (2008). 3d pose estimation and mapping with time-of-flight cameras. In *IEEE/RS International Conference on Intelligent Robots and Systems(IROS), Workshop on 3D-Mapping*.

Oggier, T., Lustenberger, F., and Blanc, N. (2006). Miniature 3d tof camera for real-time imaging. In *Perception and Interactive Technologies*, pages 212–216.

Prusak, A., Melnychuk, O., Roth, H., Schiller, I., and Koch, R. (2008). Pose estimation and map building with a time-of-flight camera for robot navigation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):334–364.

Schiller, I., Bartczak, B., Kellner, F., and Koch, R. (2008a). Increasing realism and supporting content planning for dynamic scenes in a mixed reality system incorporating a time-of-flight camera. In *Proceedings of the European Conference on Visual Media Production, CVMP*, volume 5.

Schiller, I., Beder, C., and Koch, R. (2008b). Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXI. ISPRS Congress, pages 297–302.

Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600.