When Can We Use KinectFusion for Ground Truth Acquisition?

Stephan Meister[1], Pushmeet Kohli[2],Shahram Izadi[3],
Martin Hämmerle[4], Carsten Rother[5] and Daniel Kondermann[6]

***Abstract*— KinectFusion is a method for real-time capture of dense 3D geometry of the physical environment using a depth sensor. The system allows capture of a large dataset of 3D scene reconstructions at very low cost. In this paper we discuss the properties of the generated data and evaluate in which situations the method is accurate enough to provide ground truth models for low-level image processing tasks like stereo and optical flow estimation. The results suggest that the method is suitable for the fast acquisition of medium scale scenes (a few meters across), filling a gap between structured light and LiDAR scanners. For these scenes e.g. ground truth optical flow fields with accuracies of approximately 0.1 pixel can be created. We reveal an initial, high-quality dataset consisting of 57 scenes which can be used by researchers today, as well as a new, interactive tool implementing the KinectFusion method. Such datasets can then also be used as training data, e.g. for 3D recognition and depth inpainting.**

I. Introduction and Related Work

Ground truth acquisition for performance analysis of low-level computer vision tasks such as optical flow or stereo is mainly constrained by three properties: accuracy, cost and content. Accuracy and content are limited by costs stemming from manual labor as well as measurement device prices. For example, highly accurate structured light 3D scanners are very expensive (often more than $\approx$ 50k€), labor-intensive (setup time, manual registration steps, postprocessing of data) and and are often optimized for small ($<< 1m$) to medium scale ($\approx 1m$) environments.

Computer vision algorithms have to deal with a number of competing requirements such as speed, accuracy and reliability. In real-world applications such as robotics, speed and reliability in hugely varying environments are most important. Practitioners usually cannot rely on existing benchmarks: They need to create large amounts of ground truth quickly and specifically targeted on their application. The accuracy of such a ground truth dataset needs to be one magnitude larger than the accuracy of the method to be evaluated. Hence, this paper does not focus on creating a new ground truth dataset; we examine the accuracy of a fast and cheap method to enable everyone to create his own, application-specific datasets. In particular, we focus on capturing 3D datasets, using a commercially available Microsoft Kinect®camera (cost ≈100€) and previously published 3D reconstruction system called KinectFusion [1], [2]. To this end, we obtained a binary distribution of the original authors' implementation. As discussed later, this type of rich 3D data can be used for a variety of vision-based algorithms, both as training and ground truth test data.

We offer three contributions: First, we analyze the quality of the KinectFusion 3D reconstruction method for data set capture and compare it to high-end ground truth generation techniques such as light detection and ranging (LiDAR) and discuss under which circumstances the recorded data is accurate enough to be called ground truth or training data. Second, we offer an example set of 57 scenes for public download as 3D meshes, volumetric data, and registered raw and *synthetic* depth maps, recorded in a variety of rooms, enabling researchers to train algorithms for 3D vision tasks such as object detection or depth inpainting. Third, we provide a new publicly available, interactive tool which extends KinectFusion for recording sequences and exporting 3D meshes, enabling everyone to record his own datasets.(`http://hci.iwr.uni-heidelberg.de/Benchmarks/`)

We now review the different methods employed for creating training/evaluation datasets for low-level vision problems. A straightforward way to generate training data is via Computer Graphics [3]. Early approaches for generating evaluation datasets for problems such as optical flow used short (< 14 frames) rendered sequences [4], [5], [6]. Although generation of synthetic images is easy, we need to make sure that the resulting dataset represents the data that the trained system will observe in the real world. This is an extremely challenging problem and raises the question of whether synthetic data can and should be used at all for performance analysis [7]. In contrast, the first well-known example using real data for evaluating vision algorithms is the marbled block sequence [8]. Both types of sequences are currently very limited in their number and do not represent specific application scenarios.

More recently, several real and synthetic datasets for stereo-based depth and optical flow estimation have been published on the Middlebury benchmark website [9]. There, the authors also encourage the publication of results obtained with this data. While the accuracy of the ground truth data in this benchmark is very high (about 1/60 pixel), its creation was very labor-intensive. Generally more emphasis is put on the sequences itself, not on the creation method.

[1]Heidelberg Collaboratory for Image Processing, University of Heidelberg and Intel Visual Computing Institute (IVCI)`stephan.meister@iwr.uni-heidelberg.de`
[2]Microsoft Research, Cambridge, UK, `pkohli@microsoft.com`
[3]Microsoft Research, Cambridge, UK, `shahrami@microsoft.com`
[4]LiDAR Research Group - GIScience; Institute of Geography,University of Heidelberg, `M.Haemmerle@stud.uni-heidelberg.de`
[5]Microsoft Research, Cambridge, UK, `carrot@microsoft.com`
[6]Heidelberg Collaboratory for Image Processing, University of Heidelberg, `daniel.kondermann@iwr.uni-heidelberg.de`

Fig. 1. Some representative rendered depth maps obtained from the 3D model generated using KinectFusion.These images give a general impression about the scenes in the dataset

For the creation of ground truth depth maps from multiple views, usually photogrammetric techniques of higher accuracy such as LiDAR and high-precision structured light scanning methods are employed. Well-known datasets have e.g. been published in [10].

Finally, in cases where ground truth is far too expensive or difficult to obtain, reference datasets containing difficult scenes can be recorded. The authors of such datasets assume that experts are able to qualitatively evaluate the results. For automotive scenarios, three large representative datasets have been published, two of them with partial ground truth [11], [12] and one without [13].

Our approach is closely related to multi-view 3D reconstruction. The fundamental difference to these previous approaches is that we want to enable everyone to create large sets of 3D *surface* reconstructions in *real-time* with the KinectFusion system, using a low cost Kinect sensor as capturing device. (An evaluation of the Kinect sensor accuracy itself has been performed by [14].)

II. Capturing 3D Models With KinectFusion

In the KinectFusion system [2] depth data from a consumer Kinect depth camera (and possibly other depth cameras) is integrated into a regular voxel grid structure stored on the graphics card (GPU) to produce a 3D volumetric reconstruction of the scene. Surface data is encoded *implicitly* into voxels as signed distances, truncated to a predefined region around the surface, with new values integrated using a weighted running average. The global pose of the moving depth camera is predicted using a point-plane iterative closest point (ICP) algorithm while drift is mitigated by aligning the current raw depth map with the accumulated model. For evaluation, we obtained a binary distribution of the original authors' implementation. As an extension of our system we have added capabilities to extract a geometric isosurface from the volumetric data using a GPU-based implementation of the marching cubes algorithm [15]. For each voxel, the signed distance value at its eight corners is computed. The algorithm uses these computed signed distances as a lookup (into a table stored as a 1D texture on the GPU) to produce the correct polygon at the specific voxel. This results in an exported mesh in a common format that can be used in 3D modeling applications such as MeshLab[1].

To deal with large scale capture of 3D datasets, we have created a simple data recorder and player which is made available as download. The application works as follows: We first capture the 3D scene using a process similar to the standard KinectFusion reconstruction process. Once the user has achieved a high quality of reconstruction the application saves the voxel volume and marching cubes mesh. A synchronized sequence of raw depth maps, synthetically generated depth maps (via raycasting) and 6 degrees-of-freedom camera poses (containing a 3x3 rotation and 3x1 translation vector) are then written to disk.

To test the effectiveness of the KinectFusion approach, we collected a dataset comprising each depth and color sequences in a variety of different locations such as offices, living rooms, kitchens, bedrooms, study rooms etc. Representative examples of all the sequences are shown in Figure II. During acquisition we explicitly avoided moving objects or people in the depth maps. 55 sequences consist of 900 frames each; two have 500 frames.

The entirety of these datasets, consisting of 3D meshes, voxel volumes, synthetic and raw depth maps, RGB images as well as camera poses (location + orientation) can be used for various vision-based tasks: First of all, the 3D models with known accuracy can be used to evaluate other reconstruction algorithms such as multiple view techniques based on color images. The high quality synthetic and raw depth maps can be compared, e.g. to evaluate denoising as well as depth inpainting algorithms. Together with the acquired RGB information and camera poses, each real color image can be augmented with synthetic depth ground truth. With each two of such color-depth-pairs and based on the known camera transformation, optical flow fields (as defined in [9]) can be generated by projecting the resulting 3D motion vectors into image space. Furthermore, to circumvent the limited accuracy of any real measurement device, fully synthetic sequences with ground truth can be rendered for scenes utilizing the known, realistic geometrical complexity. In this context, experiments with different lighting and materials can be carried out. All of these various datasets can also be used in machine learning based approaches to train a system to automatically enhance depth data or 3D models based on application-specific knowledge.

In the following Section we compare a few test datasets to high-accuracy, high-cost scans to evaluate the absolute quality of this dataset, with special emphasis being put on the geometric accuracies.

III. Quality Analysis

To analyze the accuracy of the KinectFusion method in different scales we created three test scenes with highly accurate ground truth. (In this section, we use the term *ground truth* for the expensive, slow 3D scan with accuracies typically at least one order of magnitude higher than the

[1]Meshlab software, `http://meshlab.sourceforge.net/`

Fig. 2. Photos of the three test scenes: statue, targetbox and office.

kinect.) Although the KinectFusion system is able to work with different depth data sources, we limited the experiments to the original Kinect sensor.

For each scene we aligned the mesh generated by KinectFusion to the ground truth data using a standard ICP implementation (Meshlab). We then computed several error measures to quantify the differences between the datasets: First, for each vertex of the KinectFusion generated point cloud we computed the minimal distance to the next face of the ground truth mesh. (In the case of the office point cloud scene where no mesh was available we computed the distance to the KinectFusion mesh for each 3D point in the ground truth.) We call this the *per vertex euclidean error*.

Second, for each vertex of the KinectFusion point cloud we calculated the difference between its normal and the normal of the closest vertex in the ground truth point cloud. This we call the *per vertex angle error*, which is more sensitive to corners and depth discontinuities and allows evaluation of sections which are critical to some image processing algorithms.

If not mentioned otherwise, the values in all images are linearly scaled according to the displayed colorbar. Minimum(blue) and maximum(red) are each mentioned in the figure captions.

Statue Scene:: Most depth cameras and 3D scanners have optics with a fixed focal length as well as a minimal and maximal acquisition depth. This is a limiting factor for the size and resolution of the scenes or objects one wants to scan using these devices. The first scene is composed of an approximately 40cm high wooden statue. Our aim with this statue is to evaluate the lower limit of resolution KinectFusion can provide (cf. Figures 2, 3). Ground truth for this scene was generated by scanning the statue with a high precision structured light-scanner[2].

To achieve maximum KinectFusion accuracy, we chose the

[2]Breuckmann smartSCAN-HE, resolution of down to 10 microns depending on field of view

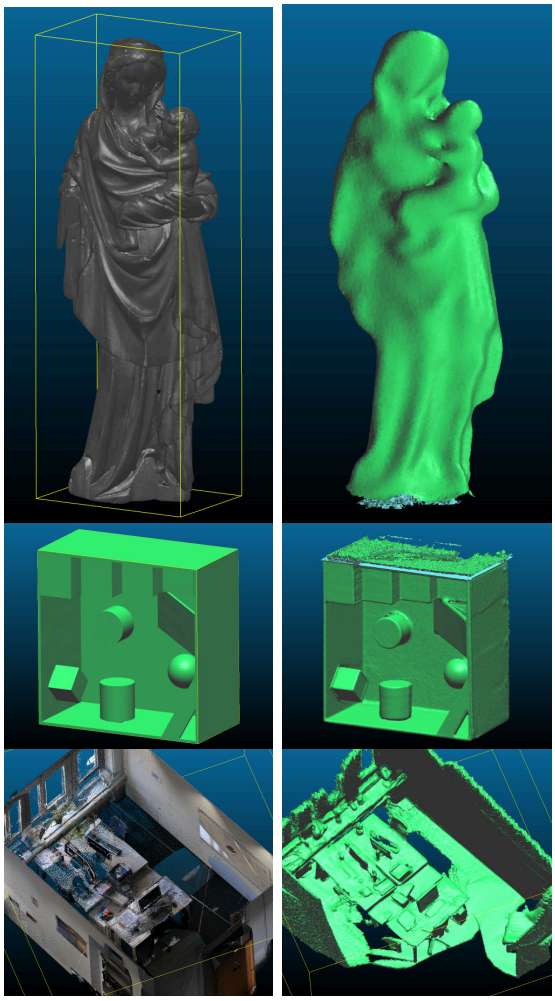

Fig. 3. Left: ground truth renderings. Right: mesh generated by KinectFusion.

implicit voxel volume to be as small as possible ($(0.8m)^3$ in this case). The resolution of 512^3 voxels is close to the maximum (600^3) our graphic card[3] could handle and accounts for voxel side lengths of $\approx 1.6mm$. (Memory requirements do scale with the third power of the volume resolution). The camera/object distance was approximately 1 meter in this case.

Figure 3 shows that the general shape of the statue could be retrieved by KinectFusion but finer surface detail is lost. The histogram in Figure 4 shows that at least half of all estimated surface points are closer than $5mm$ to the correct value. Additional 75% of all points have an error smaller than $10mm$. Hence, the system can be used for tasks where $10mm$ resolution in the absolute world coordinates is sufficient. The error of the surface normals is widely distributed, mainly due to concave sections such as the folds in the garment. From this result we conclude that highly curved and concave details below the scale of around $10mm$ cannot be resolved well with the current Kinect system, although the voxels are small enough.

[3]nvidia GTX 480 with 1.5 GB Ram

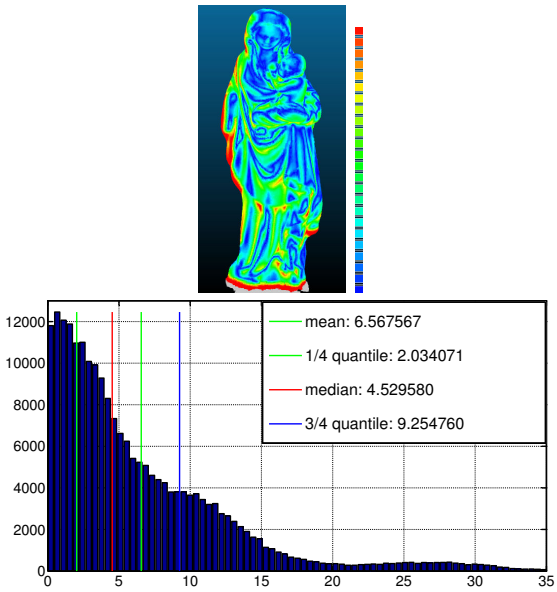

Fig. 4. Statue. euclidean error (0-8mm, $> 8mm$ is gray), histogram of euclidean error.

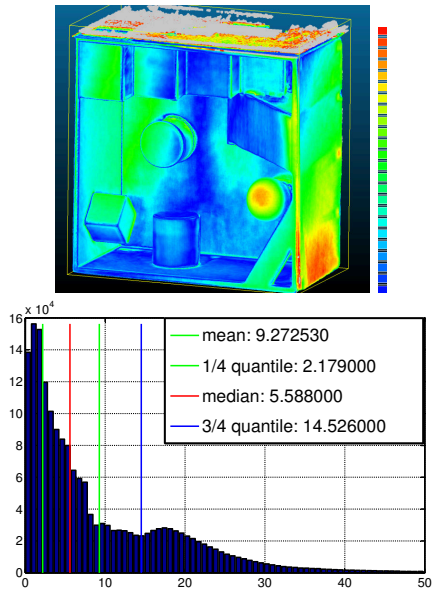

Fig. 5. Targetbox: euclidean error (0-15mm, higher errors in gray), histogram of euclidean errors.

Targetbox Scene:: The second test object is a target box especially designed for the evaluation of depth cameras. It is 1 x 1 x 0.5 meter in size and contains several geometric objects made of styrofoam, as well as many regions with slanted surfaces, curvature or sharp 90 degree corners which are typically problematic for any depth acquisition system. Therefore, we manually measured the box with an accuracy higher than $1mm$.

Fitting the box size, we set the implicit voxel volume to $(1.6m)^3$ with 600^3 voxels, yielding a voxel side length of $\approx 2.7mm$. We only scanned the interior of the box and therefore ignore its outside in the evaluation, visualized in gray as can be seen in Figure 5 (scan distance was again ca. 1 meter).

Angle errors $> 90°$ are mostly caused by vertices whose nearest neighbor was matched to one vertex on the other side of the surface (e.g. the inside and outside walls). Such errors should be either ignored or handled as if they were flipped by $180°$ (marked gray in Figure 5). Generally, surfaces which are flat or have high curvature radii (like the styrofoam sphere or cylinder) are reconstructed well with minimal angular error. Sharp corners on the other hand are partly smoothed out. We found slightly higher histogram densities for $45°$ angle errors which suggests that for a sharp $90°$ edge at least one additional face with $45°$ is generated by the marching cubes algorithm. The euclidean errors are generally low and in the same range (5-10mm) as in the previous statue scene. Only the higher error on the styrofoam sphere suggests that the algorithm underestimates the volume of curved regions. We can conclude that the voxel size of $\approx 2.7mm$ was sufficiently small for this experiment and that the accuracy of around $10mm$ is also valid for such a medium scale scene.

Office Scene:: The third scene is a small office room (6 x 4 x 2.5m) and represents one additional example of the dataset described in Section II. Ground truth data was acquired by terrestrial LiDAR using a Riegl VZ-400 time-of-flight scanner. Its accuracy is stated with $5mm$. The manufacturer also documents a precision of $3mm$. Inside the office overall six scan positions were necessary for a sufficient coverage. This equals to about one day of labor for acquisition and postprocessing.

In order to fit the whole room into the 512^3-voxel volume we had to choose a voxel side length of about $\approx 13.7mm$ while keeping a scan distance of 1 to 2 meters. This means that the actual accuracy of the Kinect system of about 5-10mm can no longer be fully exploited. Given current graphics hardware, the office scene represents the maximum size which can be scanned by the KinectFusion system.

ICP alignment of the KinectFusion mesh and the ground truth mesh are here not perfectly accurate as a small scaling along the object axes was necessary. This is caused by three reasons: first, the scene is heavily cluttered containing many regions were any 3D scanning devices fails. Second, the increased voxel sizes create a coarser mesh which is more difficult to align to the LiDAR results. Third, the LiDAR scan itself is more inaccurate in regions with small scale detail and contains some holes and regions of low point cloud density. The euclidean error is therefore about one magnitude larger than for the other scenes. Yet, most vertices with errors $> 100mm$ are actually on the outside of the room as the marching cubes algorithm produces walls which are not flat faces but have a certain volume. As Figure 6 shows, the error is well below $80mm$ for most vertices. These high errors are caused by regions were both methods fail. Future work should focus on detecting such regions of low certainty in order to mask them out in the resulting benchmark datasets. To get an idea of the accuracy in more confident regions, a robust statistical measure such

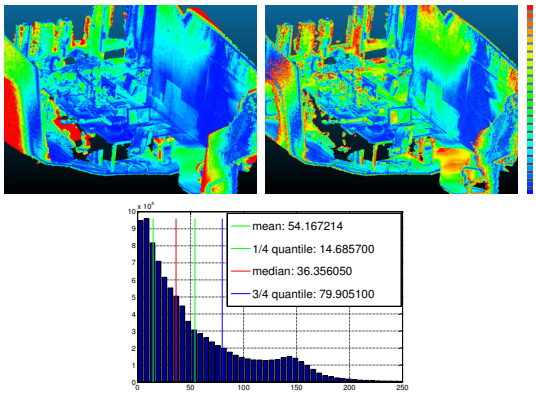

Fig. 6. Office: euclidean error (0-100mm, higher errors are also red), euclidean error(0-50mm, higher errors are transparent), histogram of euclidean error.

as the median error can be used whose value is just below three voxel sizes ($36mm$). This indicates that even if the number of voxels were increased, the measurement volume of the current KinectFusion system (using Kinect depths as input) should not be much larger than $7x7x7m$ to achieve maximum accuracy. Yet, more accurate depth sensors and larger amounts of graphics card memory might soon alleviate this limit. For now we conclude that very careful acquisition of all concave regions in the office is very challenging with both 3D scanning methods.

A. *Ground Truth Accuracy for Optical Flow and Stereo*

In optical flow, real frames at two successive time steps of a video can be augmented with synthetic depth maps based on known camera poses. As our KinectFusion scenes are static, the depth maps can be reprojected to flow fields using the camera transformation between both views. Although optical flow is only induced by camera movement, challenging flow fields can be created (similar to the yosemite, grove and urban scenes in [9]). For stereo, the second color image should be aligned with respect to the epipoles. In order to use real color images, for a given single view, a nearby view can be found which is then rectified based on the known camera poses and previously measured internal camera parameters.

In realtime computer vision applications a sufficient accuracy often is about one pixel in motion or disparities. Hence, to achieve ground truth quality, the KinectFusion system should record data which is one order of magnitude more accurate. We synthesized a stereo disparity map and an optical flow field from two virtual views of the targetbox scene (camera distance $\approx 1.3m$, field of view $40°$, maximum flow magnitude ≈ 25 pixel). We then compared the flows and disparities for both the high-accuracy scans as well as the KinectFusion scans. Figure 7 shows the per pixel endpoint error, a widely used error measure for optical flow evaluation [9]. The mean endpoint error for this scene was 0.06 pixel with a median of 0.02 pixel. Most errors occurred on depth discontinuities. To evaluate stereo disparity accuracies we transformed the depths to disparity values (focal length 1100 pixel, $7.5cm$ eye separation). The mean disparity error was 0.25 pixel with a median of 0.11 pixel.

Fig. 7. Left: rendering with optical flow as hsv color overlay; Middle: optical flow endpoint error (0-0.2 pixel, higher errors are white) between ground truth and KinectFusion based scene; Right: stereo disparity error (0-1 pixel, higher errors are white) between ground truth and KinectFusion based scene.

We conclude that KinectFusion based geometry data can indeed be used to generate ground truth optical flow and stereo information in case the application requires accuracies in the order of magnitude of around one pixel. Optical Flow evaluation is hereby limited to static scenes but still useful e.g. for simultaneous location and mapping (SLAM) problems. With these results, we would like to encourage practitioners to create their own ground truth datasets with content specifically designed to sample the space of challenges within a given application.

IV. Conclusion and Future Research

We have compared 3D reconstructions produced by the KinectFusion algorithm with ground truth data obtained from high-precision 3D scanners. The Kinect sensor has several advantages over such systems: The setup is fast as no calibration is needed, scanning is fast, meshed results are available within minutes and in contrast to LiDAR or structured light scanners, no extensive manual postprocessing is needed. The Kinect sensor also is also more portable and small compared to other devices, facilitating the acquisition of additional viewpoints in highly complex scenes. Finally, the effective field of measurement is quite large, closing the gap between portable structured light scanners which are typically restricted to volumes $< (1m)^3$ and LiDAR equipment for larger outdoor scenes. We offer an exemplary set of sequences in this scale range for download. (`http://hci.iwr.uni-heidelberg.de/Benchmarks/`)

We found that the system can resolve object details with a minimum size of approximately $10mm$. This also represents the minimum radius of curvature for slanted or curved surfaces which can be reconstructed reliably. Sharp (depth) edges or highly concave scenes are as problematic for KinectFusion as for many other 3D scanning technologies. For indoor scenes with a volume of $(7m)^3$ this accuracy drops to $\approx 80mm$ with GPU memory and the Kinects minimum object distance as the limiting factors. Optical flow and stereo ground truth can be created with average accuracies in the range of better than 0.1 pixel.

Future work will focus on the quantification and detection of missing or incorrect geometry. Furthermore, we are going to investigate other cheap depth sensors for more accurate KinectFusion input data.

V. Acknowledgements

This work has been partially funded by the Intel Visual Computing Institute, Saarbrücken (IVCI) and is part of the Project "Algorithms for Low Cost Depth Imaging". We thank Prof. Susanne Krömker, Anja Schäfer and Julia Freudenreich of the Visualization and Numerical Geometry Group (Interdisciplinary Center for Scientific Computing, University of Heidelberg) for carrying out the structured light scans and for additional advice. We also thank Markus Forbriger, Larissa Müller and Fabian Schütt of the LiDAR Research Group (University of Heidelberg) for their help in acquiring the LiDAR scans.

References

[1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, vol. 7, 2011, pp. 127–136.

[2] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion : Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11, 2011, pp. 559–568.

[3] O. Mac Aodha, G. Brostow, and M. Pollefeys, "Segmenting video into classes of algorithm-suitability," in *CVPR2010*, 2010, pp. 1054–1061.

[4] D. Heeger, "Model for the extraction of image flow," *Journal of the Optical Society of America*, vol. 4, no. 8, pp. 1455–1471, 1987.

[5] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.

[6] J. L. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *IJCV*, vol. 12, no. 1, pp. 43–77, 1994.

[7] S. Meister and D. Kondermann, "Real versus realistically rendered scenes for optical flow evaluation." in *Proceedings of 14th ITG Conference on Electronic Media Technology*, 2011.

[8] M. Otte and H. Nagel, "Optical flow estimation: advances and comparisons," in *ECCV*, 1994, pp. 51–60.

[9] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *IJCV*, vol. 92, no. 1, pp. 1–31, 2011.

[10] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR2006*, vol. 1, 2006, pp. 519–528.

[11] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *Proc. of 23rd International Conference on Image and Vision Computing New Zealand*, 2008, pp. 1–6.

[12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR2012*, Providence, USA, June 2012.

[13] S. Meister, B. Jähne, and D. Kondermann, "Outdoor stereo camera system for the generation of real-world benchmark data sets," *Optical Engineering*, vol. 51, 2012.

[14] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS Workshop Laser Scanning*, vol. 38, 2011, p. 1.

[15] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.