Prod. Eng. Res. Devel. (2014) 8:745–754
DOI 10.1007/s11740-014-0552-0

A new benchmark for pose estimation with ground truth from virtual reality

Christian Schlette · Anders Glent Buch · Eren Erdal Aksoy · Thomas Steil · Jérémie Papon · Thiusius Rajeeth Savarimuthu · Florentin Wörgötter · Norbert Krüger · Jürgen Roßmann

Received: 10 March 2014 / Accepted: 29 April 2014 / Published online: 15 May 2014

Abstract The development of programming paradigms for industrial assembly currently gets fresh impetus from approaches in human demonstration and programming-by-demonstration. Major low- and mid-level prerequisites for machine vision and learning in these intelligent robotic applications are pose estimation, stereo reconstruction and action recognition. As a basis for the machine vision and learning involved, pose estimation is used for deriving object positions and orientations and thus target frames for robot execution. Our contribution introduces and applies a novel benchmark for typical multi-sensor setups and algorithms in the field of demonstration-based automated assembly. The benchmark platform is equipped with a multi-sensor setup consisting of stereo cameras and depth scanning devices (see Fig. 1). The dimensions and abilities of the platform have been chosen in order to reflect typical manual assembly tasks. Following the eRobotics methodology, a simulatable 3D representation of this platform was modelled in virtual reality. Based on a detailed camera and sensor simulation, we generated a set of benchmark images and point clouds with controlled levels of noise as well as ground truth data such as object positions and time stamps. We demonstrate the application of the benchmark to evaluate our latest developments in pose estimation, stereo reconstruction and action recognition and publish the benchmark data for objective comparison of sensor setups and algorithms in industry.

Keywords Industrial assembly · Machine vision · Machine learning · Virtual reality

1 Introduction

In typical setups to implement new programming paradigms based on human demonstration and PbD [6, 33, 36, 37], multiple sensors observe the given workspace shared by the robotic system and a human operator. The human operator then demonstrates a sequence of object manipulations to the system, which—after a few trials—the robot should be able to reproduce by reasoning from the generated sensor data.

Sensors for observing robotic workspaces are stereo cameras which offer RGB image streams and allow for

C. Schlette (✉) · T. Steil · J. Roßmann
Institute for Man-Machine Interaction (MMI),
RWTH Aachen University, Aachen, Germany
e-mail: schlette@mmi.rwth-aachen.de

T. Steil
e-mail: steil@mmi.rwth-aachen.de

J. Roßmann
e-mail: rossmann@mmi.rwth-aachen.de

A. G. Buch · T. R. Savarimuthu · N. Krüger
The Maersk Mc-Kinney Moller Institute (MMMI),
University of Southern Denmark, Odense, Denmark
e-mail: anbu@mmmi.sdu.dk

T. R. Savarimuthu
e-mail: trs@mmmi.sdu.dk

N. Krüger
e-mail: norbert@mmmi.sdu.dk

E. E. Aksoy · J. Papon · F. Wörgötter
Bernstein Center for Computational Neuroscience (BCCN),
Georg-August University Göttingen, Göttingen, Germany
e-mail: eaksoye@physik3.gwdg.de

J. Papon
e-mail: jpapon@physik3.gwdg.de

F. Wörgötter
e-mail: worgott@physik3.gwdg.de

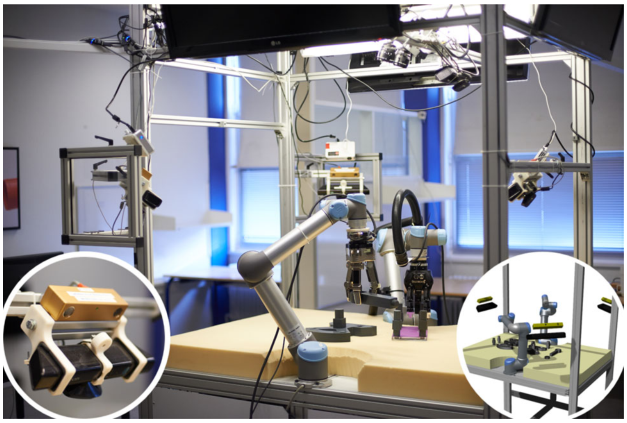

Fig. 1 Benchmark platform with two robot manipulators and three bundled pairs of RGB stereo cameras and "Microsoft Kinect" RGB-D devices with overlapping fields of view (*left*, *right* and behind the farthest robot). *Bottom left* sensor bundle in detail. *Bottom Right* virtual benchmark platform with simulated RGB stereo cameras and RGB-D devices

deriving depth information. Since depth information is crucial for evaluating 3D object poses as well as spatial relations, sensors based on scanning mechanisms to directly generate depth information are widely used: Laser scanners, PMD cameras and "Microsoft Kinect" [13, 29, 50] compile arrays of distance measurements from probing their field of view with a given sample density. Associating the measurements with (quantized) reference depths then results in clouds of points in sensor coordinates—or colored point clouds if the depth data is combined with RGB data of the same view, yielding RGB-D data. For enlarging the field of view, canceling shadows and avoiding systematic errors, often multiple sensors and combinations of different sensors are used at the same workspace. And software libraries such as "OpenCV" [7] and "Point Cloud Library" (PCL) [41] are offering quasi-standardized methods for the calibration and evaluation of such multi-sensor setups. Still, to the knowledge of the authors, there are no benchmarks available that would allow for the independent comparison of multi-sensor setups or algorithms working on them—mostly due to the fact that reliable and precise ground truth measurements are missing.

1.1 Benchmark platform

The benchmark platform is equipped with two 6-DOF robots for object manipulation, and a multi-sensor setup consisting of four RGB stereo cameras, three "Microsoft Kinect" RGB-D devices and two projectors for shedding structured light on the scene (see Fig. 1). The sensors are arranged with overlapping fields of view for covering a large part of the robot's workspace.[1] In order to reflect typical manual assembly tasks, the benchmark describes common situations during the assembly of the well-established "Cranfield" set [10, 27, 42, 46] (see Fig. 7). The projectors are used to project texture on objects to improve the stereo processing, since object structure improves correspondance finding. Figure 2 shows a subset of the Cranfield objects with such a projected texture. In general, the platform allows for complex manipulation of objects as well as pose estmation with high precision. It has been used in a PbD context in the IntellAct project (see Acknowledgements) in which the assembly processes of the Cranfield benchmark have been taught by human demonstration.

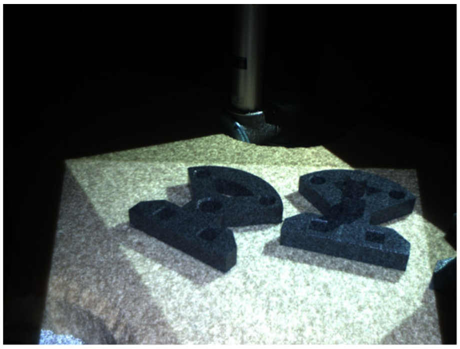

Fig. 2 A view of the real platform containing a subset of the Cranfield objects, with a texture projection for better stereo reconstructions

Overall, the platform has a functionality which is also imaginable in a future industrial setup. Multiple cameras may be required for dealing with occlusions, and two robots may be required for performing complex actions such as screw mounting.

1.2 Ground truth from VR

Our benchmark provides RGB and RGB-D data sets (stills and streams) which have been generated using a simulatable 3D representation of this benchmark platform in a VR system (see Fig. 1). Following the eRobotics methodology, the VR system features interactive 3D visualization and simulation of robot kinematics, dynamics, sensors and control in the context of the target environments [35]. Here, in particluar the detailed camera and sensor simulation allows for offering benchmark images and point clouds with controlled levels of quality, reaching from "ideal" to

[1] It is well-known that multiple Kinect sensors sharing a common field of view will cause IR interference, resulting in poor depth reconstructions. A known solution, which our platform also incorporates, is the use of vibrating motors mounted on the Kinect sensors [9]. This method has been shown to effectively blur out the noisy contributions of external sensors, while maintaining a high depth reconstruction quality.

"close to reality". Here, "close to reality" is defined by the similarity of outcomes when key factors of the real and simulated data are processed by libraries such as OpenCV and PCL, e.g. color histograms (RGB deviation, RGB saturation), edge detection, SURF feature detection and RANSAC feature similarity. In addition, assembly actions in VR have been carried out and tracked with a dataglove, thus generating accurate, objective ground truth data, e.g. exact object positions from CAD data as well as detailed information on the timing and existence of object manipulations and spatial relations between manipulated objects. The major advantage of generating ground truth from VR is the full transparency and control of data acquisition and the world model at each time step, thus providing otherwise unavailable details of the significant parameters [39]. Comparisons of real and virtual images were made in the project FastMap [40] and showed, that artificial and real images led to very similar result in the computer vision algorithms. In this work we used the standard depth error of the Kinect as our lowest noise level and compared the algorithms with less accurate testdata.

2 State of the art

This contribution is focusing on the analysis of pose estimation and action recognition based on ground truth from VR, resp. from camera and sensor simulation, as examples for similar algorithms in machine vision and learning.

2.1 Pose estimation

A number of databases for the evaluation of full 6D pose estimation algorithms exist. One of the most widely used is the object recognition dataset of Mian et al. [28], consisting of 5 models and 50 scenes, all acquired by a laser scanner. For this data, each scene contains ground truth pose information, however, this information has been obtained by running the pose estimation algorithm followed by fine registration by ICP [5]. More recently Lai et al. [26] proposed the extensive RGB-D database consisting of a large set of RGB-D views of a variety of objects and scenes. The objects are acquired from multiple views using a turntable. The ground truth information of this database consists only of the approximate turntable angle, making this database effectively unusable for benchmarking full 6D pose estimation algorithms. Finally, Aldoma et al. [3] as well as Glover and Popovic [17] have presented more challenging datasets from real scenes with clutter and occlusions. For these datasets, ground truth information is also available, but again based on a prior fine registration. In our benchmark, we provide similar information, but with guaranteed accurate ground truth information. Additionally, we provide an extension compared to previous datasets in the form of multi-view scenes.

2.2 Action recognition

Our action recognition framework relies on the concept of semantic event chains which encodes the semantics of actions from spatiotemporal relations between objects in the scene and does not use any assumption or prior knowledge in the object or action domain. Ideas to utilize spatial relations to reach semantics of actions can be found as early as in 1975 [4]. Still there are only a few approaches attempting to reach the semantics of actions for the recognition task [22, 44, 45, 49]. Although all those works to a certain extent improve the classification of manipulations and/or objects, none of them can track multiple objects in the scene and extracts key events of individual manipulations, at which richer action descriptors such as trajectory and pose information can be embedded.

Multi-Target Visual Tracking is a well-established field, which goes back over thirty years. In this work we use Sequential Bayesian Filtering (SBF), a technique which recursively estimates the time-changing posterior distribution of target states given all previous observations. We use a Sequential Monte Carlo method known as Particle Filtering to approximate the posterior, an approach which was first introduced to the vision community by Isard and Blake [23] and has been the subject of much subsequent research extending it to multiple targets [21, 48]. We adopt a filtering model which uses a separate (independent) particle filter for each target. Interactions between targets are resolved using changes to the observation and process models of the filters in order to explicitly model occlusions, similar to our previous work [32] and that of Khan et al. [24].

In contrast to other well-known data sets, our new benchmark set captures manipulation actions where multiple objects are interacting with each other in a given assembly task to address the occlusion, tracking and recognition problems. Actions in the proposed dataset are recorded from multiple views with static RGB-D devices since we are interested in understanding the spatiotemporal interactions between the manipulated objects. The conventional data sets, however, employ the entire human body configurations and movements as main features and therefore either do not involve hand-tool features [18, 25, 43] or are not rich to provide enough recordings required for benchmarking [45, 49].

2.3 Camera simulation

The VR system features a camera and sensor simulation [14, 38, 40] which is able to resemble the technical specifications and output data signatures (in particular noise and errors) of the visual sensors in question. To achieve real-time simulation, we utilize rasterization techniques that can be implemented in modern shader-driven GPUs for hardware accelerated real-time rendering. At first, the camera parameters are measured according to [19] or obtained by the documentation of the manufacturer. The camera simulation then provides a real-time simulation of various optical and electronic effects. It exceeds the standard camera simulation of other simulation software as V-Rep [11], Gazebo [31] or Microsoft Robotics Developer Studio [30] as it allows for simulating various optical and electronic effects in real-time. The modular and flexible approach allows the extension of further effects as they are needed. A simplified schematic global description of the rendering process for different optical and electronic effects is shown in Fig. 3. The input data consists of the geometric description of the scene and lighting conditions (direction, color and lighting model) which are combined in an appropriate lighting shader to ensure real-time visualization.

The sequential arrangement of different optical effects is not interchangeable and needs to be computed in the right order, e.g. lens distortion has to be added before various noise effects are rendered. Therefore a shaderstack combines the different shaders in the right order and processes the rendered images. The different parameters of optics and sensor are given to the shaderstack that combines these values to a subsequent chain of different shading programs. The resulting programs are completely GPU based and allow for high performance simulation of different optical and electronic effects, real-time rendering and interactive adjustment of camera parameters. The various optical effects have a significant impact on computer vision algorithms and thus are important for a realistic visualization of a scenario. Radial symmetric distortion appears if the magnification of a lens increases or decreases with the distance from the optical center, e.g. a decrease of magnification leads to barrel distortion (see Fig. 4).

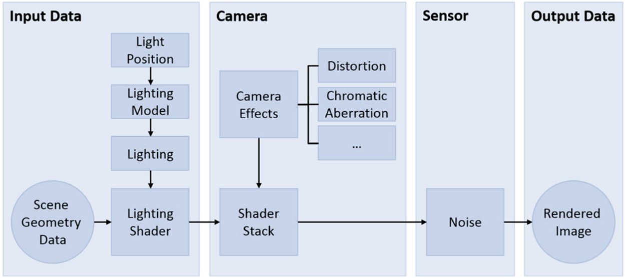

Fig. 3 Concept of rendering the virtual scene with effects

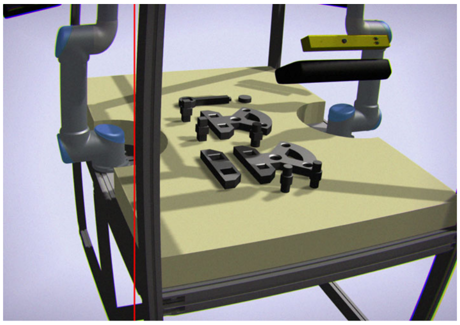

Fig. 4 Barrel distortion with chromatic aberration. A vertical line is inserted to demonstrate the effect

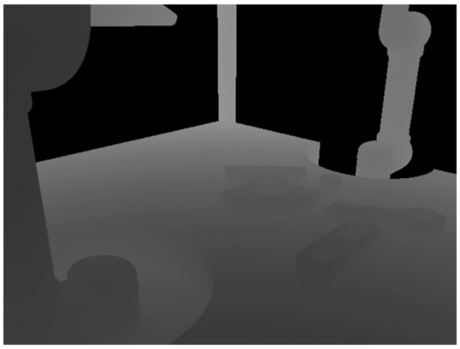

Fig. 5 Depth image as used for point cloud generation

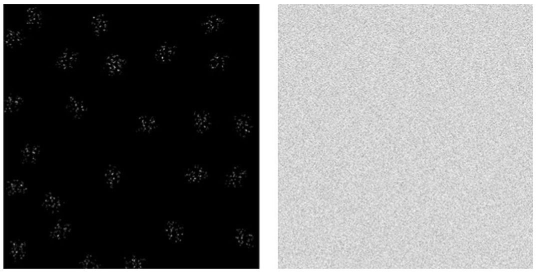

Fig. 6 Noise textures. *Left* hotpixel noise. *Right* chromatic gauss-distributed noise

Depth of field is supported by our simulation but neglected here, since all objects are in sufficient distance from the camera and the simulated image sensors are small. Instead, the depth pass of our renderer is used for the generation of point clouds as depicted in Fig. 5. Such depth images are also the basis of adding noise to the point clouds by adding or subtracting random values to the grey-scale image depending on the size of the scene and the chosen error. Optical CMOS or CCD sensors produce a varying amount of noise [12], depending on lighting conditions, temperature and pixel size. Our simulation allows for multiple noise functions, such as hotpixel noise, color noise and monochrome noise as shown in Fig. 6. The amount of noise is taken from real images or are delivered by the sensor manufacturers, e.g. by taking images in complete darkness in order to obtain hotpixel noise. The resulting noise textures are added to the rendered image in a post processing step. The reproducibility of highly dynamic noise effects [15] can be accomplished through the active simulation time as seeds for distribution of semi-random noise values.

3 Benchmark experiments

Using the benchmark data from VR, we analyzed and evaluated our latest developments in pose estimation and action recognition as well as stereo reconstruction as an important intermediate link between the two.

3.1 Pose estimation

We have generated multiple different virtual RGB-D scenes for benchmarking pose estimation. All in all, we have generated six scenarios, one scenario referring to a random placement of a random subset of the available objects on the table. The four objects involved in the experiments are shown in Fig. 7. For each scenario, we provide three views, one for each simulated RGB-D device. Additionally, each scenario is recorded at six noise levels, ranging from 0 mm (ideal) to 10 mm. Note that the largest noise level is much higher than the expected noise of the real “Microsoft Kinect”, as pointed out in recent studies [20]. For each scenario, we thus get 18 single-view test scenes, and consequently 6 three-view scenes. In total, over all six scenarios, we thus get 108 single-view scenes and 36 three-view scenes.

For benchmarking, we apply our existing pose estimation method proposed in [8] which implements an optimized RANSAC [16], achieved by a prerejection step based on low-level geometric consistency of the point pairs sampled during each iteration. Contrary to the original work, we apply the SHOT feature [47] here, which represents state of the art in shape matching, for obtaining feature correspondences for the estimation routine. Finally, to obtain a high accuracy, we refine the poses using ICP [5]. An example scenario with pose estimation results is shown in Fig. 8.

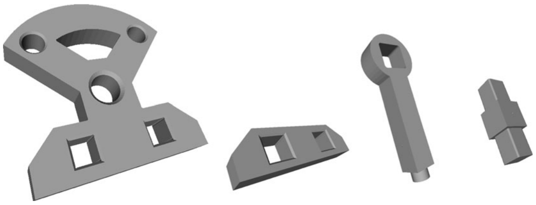

Fig. 7 The four object models used in the pose estimation experiments, here represented CAD models. For the experiments, these models have been resampled to point clouds to allow for feature extraction. The objects are named from *left* to *right*: *Faceplate*, *Separator*, *Pendulum* and *Bolt*

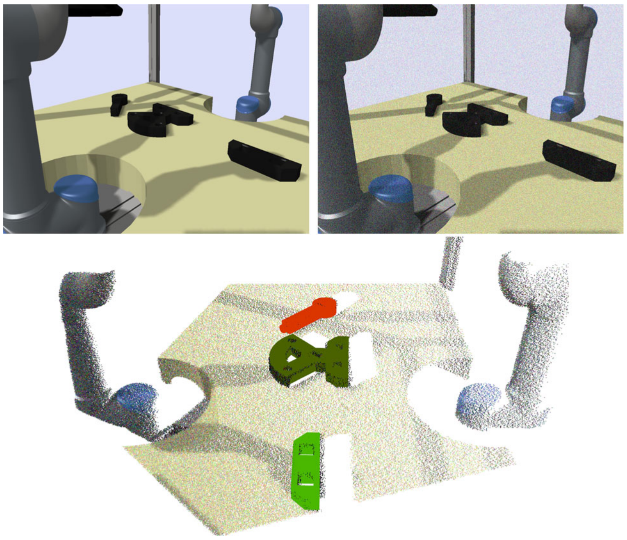

Fig. 8 Qualitative pose estimation result for a single view scene. *Top left* ideal input scene. *Top right* same scene at the highest noise level. *Bottom* pose estimation result of three objects for the top right scene, viewpoint slightly different, revealing the depth noise which has a standard deviation of 10 mm. The aligned object models are overlaid with random colors (color figure online)

We evaluate the single-view scenes and the three-view scenes separately. Translation errors are evaluated as the Euclidean distance between the estimated translation vector and the ground truth translation. For evaluation of the rotation error, we take the geodesic manifold distance between the estimated rotation R_{Est} and the ground truth rotation R_{GT}:

$$\left|\log\left(R_{Est}^T \cdot R_{GT}\right)\right| = \arccos\left(\frac{\mathrm{trace}\left(\mathrm{R}_{\mathrm{Est}}^{\mathrm{T}} \cdot \mathrm{R}_{\mathrm{GT}}\right) - 1}{2}\right) \tag{1}$$

where the logarithmic map, when applied to a rotation matrix in $SO(3)$, gives the Lie algebra $so(3)$, also known as the angle-axis representation. The norm of this vector represents the minimal angle required to align the two rotation frames, and lies in the interval $[0, 180]$ deg.

For all objects, with the exception of the small bolt, there is a two-fold symmetry around the main axis, which is near vertical in Fig. 7. For the bolt object (rightmost in Fig. 7), there are four symmetries around this axis, plus an additional symmetry around the near horizontal axis, giving eight possible rotations for this object. All symmetry rotations produce a correct alignment of the objects. Thus, when performing evaluation, we must handle these cases by applying the symmetry rotations to each candidate pose, and evaluating the ground truth rotation against the nearest matching rotation.

The results of these experiments are shown in Table 1 for the individual camera cases and in Table 2 for the three

Table 1 Ground truth median pose errors over all single-view RGB-D scenes for all noise levels

Noise level (mm)	None	1	2.5	5	7.5	10
Faceplate						
Trans. error (mm)	0.050	3.7	3.8	4.7	6.6	8.3
Rot. error (deg)	0.0	0.35	0.34	0.58	0.53	0.83
Separator						
Trans. error (mm)	0.21	5.2	5.4	6.2	7.7	8.7
Rot. error (deg)	1.2	1.6	1.6	1.1	1.2	0.94
Pendulum						
Trans. error (mm)	0.080	2.8	2.9	4.3	6.1	9.2
Rot. error (deg)	0.77	0.75	0.93	1.8	3.5	2.4
Bolt						
Trans. error (mm)	0.26	5.2	5.2	5.5	8.9	10
Rot. error (deg)	1.8	2.4	2.3	1.2	1.0	2.3
Average						
Trans. error (mm)	0.15	4.2	4.3	5.2	7.3	9.1
Rot. error (deg)	0.94	1.3	1.3	1.2	1.6	1.6

For the small bolt object, rotation errors are not available due to multiple symmetries. In few cases for the separator and the pendulum, the orientation is completely misestimated, causing a high rotation error

Table 2 Ground truth median pose errors over all three-view RGB-D scenes for all noise levels

Noise level (mm)	None	1	2.5	5	7.5	10
Faceplate						
Trans. error (mm)	0.030	3.4	3.5	4.3	5.9	6.9
Rot. error (deg)	0.18	0.23	0.23	0.37	0.95	0.87
Separator						
Trans. error (mm)	0.020	3.5	3.8	4.1	4.8	5.4
Rot. error (deg)	0.65	0.76	0.75	0.85	1.2	1.6
Pendulum						
Trans. error (mm)	0.3	4.1	4.7	5.2	5.9	8.6
Rot. error (deg)	1.1	0.77	0.39	2.0	1.2	1.4
Bolt						
Trans. error (mm)	0.24	3.4	3.6	6.2	5.5	11
Rot. error (deg)	1.5	0.89	2.2	0.66	1.2	3.8
Average						
Trans. error (mm)	0.15	3.6	3.9	5.0	5.5	8.0
Rot. error (deg)	0.86	0.66	0.89	0.97	1.1	1.9

camera cases. We observe that the translation errors are very small for the noise-free, ideal case, which is expected. The reasons for that these numbers are not zero are (1) that the single-view scenes only show partial views of the objects, (2) that the pose estimation routine uses a subsampled point cloud of the objects for speed (the voxel resolution is set to 5 mm), and finally (3) that the ICP refinement is limited to 10 iterations, also because of speed considerations. We see a systematic increase in the errors with increasing noise level, both in translation and rotation. More importantly, we observe that the three-view case generally provides a higher level of robustness against noise. Indeed, the results in Table 2 reveal reduced errors for the high noise levels. This is also an expected result, since the full representation provided by the three-view scenes allows for ICP to converge to a pose closer to the centroid using observations on all sides of the object surfaces. In a few instances, the rotation was misestimated. This occurred e.g. for the pendulum in the single-view dataset at a noise level of 7.5 mm, causing a high mean rotation error in these cases. Curiously, for this object the three-view case resulted in higher translation and rotation errors for all but the two highest noise levels. Again, these errors are a result of the poor rotation estimates. Likewise, the bolt, being very small, shows quite unstable results, primarily due to poor rotation estimates. All row errors have the same weight in the calculation of the average errors (bottom rows), which allows the large errors in the bolt rotation estimates to cause a higher average rotation error in one three-view case (highest noise level). Apart from this, the general picture remains, namely that the use of three views allows for increased robustness towards noise.

3.2 Stereo reconstruction

As an additional application of the demonstrated simulation platform, we have constructed an example scene with an added simulated texture projector as also used in the real setup described in Sect. 1.1. This allows for a better reconstruction of point cloud data from an RGB image pair. The stereo cameras used in the setup are the "Point Grey Bumblebee2" model [34], which have a resolution of 1024x768 px. As in the real platform, we place the texture projector above the stereo camera, capture the virtual scene at different noise level, and run the OpenCV block matching algorithm to reconstruct point cloud data. The scene, which contains three of the known objects and one cluttering object, is shown with increasing pixel noise in Fig. 9.

As expected, we get a strong dependency between the amount of pixel noise, and the ability of the block matching algorithm to compute reliable stereo correspondences. Qualitative results are shown in Fig. 10.

Finally, we have executed our pose estimation algorithm on these point clouds. In Fig. 11, we show estimation results for the ideal case and the lowest noise level. For the noisy scene, a false positive occurs for the bolt. This happens because the reconstruction noise accidentally creates local structures similar to those of the bolt. For the last four noise levels, the local shape features computed on

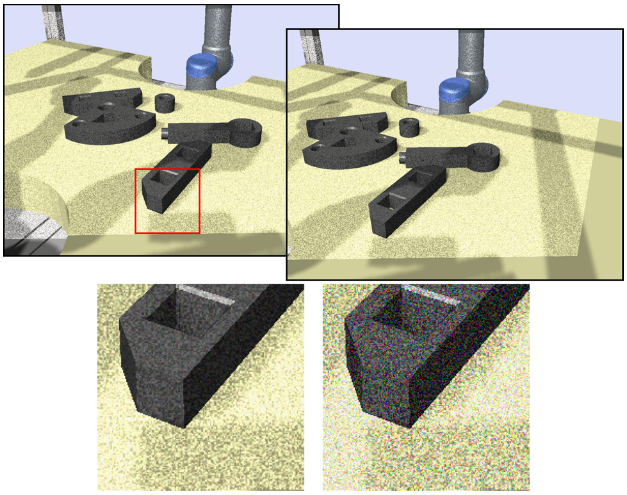

Fig. 9 An example scene with a simulated texture projector, suitable for stereo reconstruction. *Top* noise-free image pair for the simulated stereo camera, showing the projected texture used during stereo matching. *Bottom* zoom of the region marked in the *top image* for the ideal case (leftmost) and at the highest pixel noise level (rightmost). We show only a small region of the noisy stereo images for visibility purposes, and this picture is best viewed in the electronic version

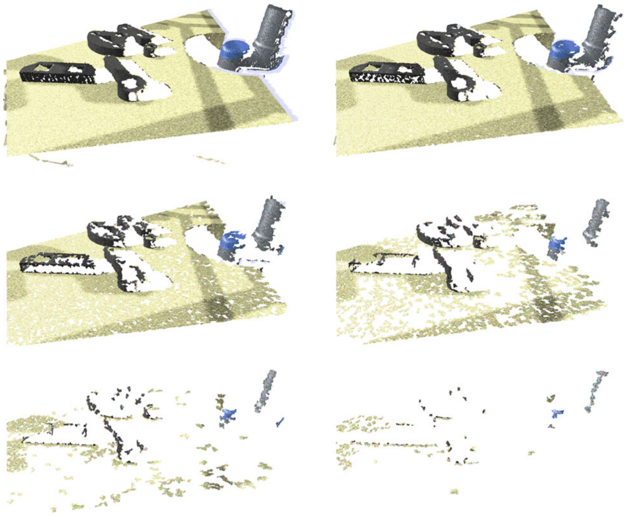

Fig. 10 Stereo reconstruction results for the image pairs visualized in Fig. 9. Pixel noise of the input stereo images increase from *left* to *right* and *top* to *bottom*

the point cloud are severely distorted, leading to failure during matching.

3.3 Action recognition

We have recorded the Cranfield assembly task with 3 virtual RGB-D devices for evaluating the action recognition stage. Our recognition framework utilizes action semantics which are bootstrapped from consistently tracked objects in the scene. We applied the framework introduced in [32] to the fused point clouds to separately track all objects present in the scene. Figure 12 illustrates the tracked 3D positions of Cranfield parts (*Bolt*, *Faceplate*, *Pendulum*, and *Separator*) at each frame compared to the ground truth.

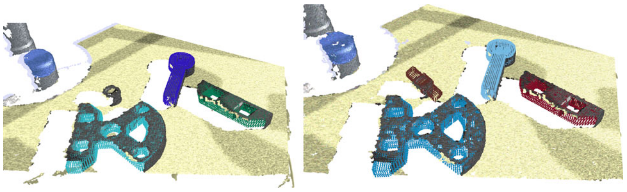

Fig. 11 Pose estimation results for the two top point clouds in Fig. 10, viewpoint set to other side of the table. For the rightmost case, a false positive occurs for the bolt

In the assembly task all consistently tracked objects are represented by graphs in which nodes represent manipulated objects and edges indicate whether two objects touch each other or not. By using an exact graph matching technique, the framework discretizes the entire graph sequence into decisive main graphs. A new main graph is identified whenever a new node or edge is formed or an existing edge or node is deleted. Thus, each main graph represents a key frame in the manipulation sequence. Sequences of all extracted key frames are employed for measuring the semantic similarities, yielding action recognition as described in [1, 2]. Figure 12 depicts sample key frames with tracked objects (each is indicated with a different color) and corresponding graphs, at which a new action is recognized.

4 Benchmark data

The benchmark presented in our contribution is publicly available and contains stills and action sequences from multiple setups at various levels of noise. The benchmark data is available at http://www.mmi.rwth-aachen.de/exchange/data/pesi2014/benchmark.htm.

For stills, the number and positions of objects vary, while camera and lighting positions remain constant. Each dataset contains a RGB and PCL folder, where the corresponding files can be found. The filename is defined by the camera, e.g. *Kinect_1_RGB.png* in folder *Set_003\0.2 Noise RGBData* is an image taken from the perspective of the "Microsoft Kinect" device no. 1 with 20 % chromatic noise. The definitions of noise levels are described in *NoiseInfo.txt* files for each set. In addition, each set of stills contains a *Ground Truth.txt* and a *CameraPositions.txt* file, where ground truth positions and orientations of objects and cameras are given.

Action sequences where recorded using an "Intersense IS 9000" tracking system with an "Immersion Cyber

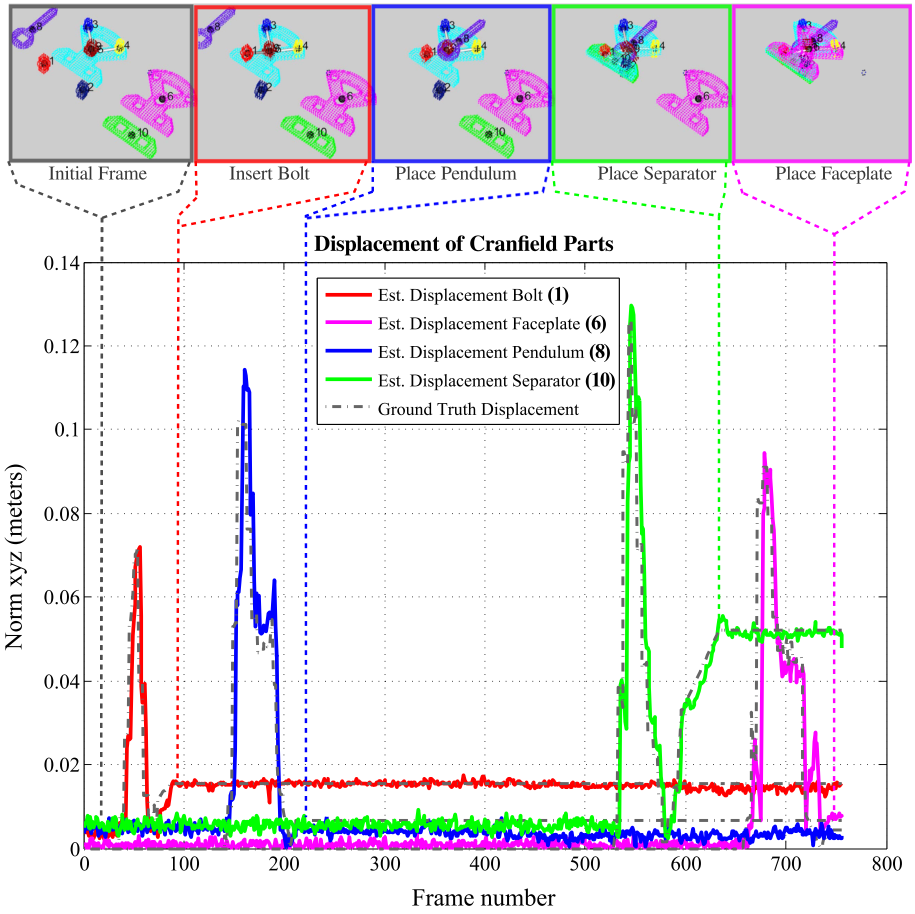

Fig. 12 Tracked 3D positions of Cranfield parts *Bolt*, *Faceplate*, *Pendulum*, and *Separator* compared to the ground truth. Extracted key frames indicated on the *top* represent the semantic changes in the scene and are employed to recognize actions. Each object is assigned with a unique graph node and color. White edges indicate touching relations between objects. Recognized action labels are given below each key frame (color figure online)

Glove 2" for tracking the user interaction when assembling the virtual Cranfield set. The data was recorded and played back in simulation, where images and point clouds were generated from the simulation data with selected levels of noise at a rate of 33 ms (30 frames per second). The naming scheme of action sequences is similar to stills, but the simulation time is added to the filename e.g. *simtime-05880ms_Kinect_2_PointCloud.pcd* corresponds to the point cloud of "Microsoft Kinect" device no. 2 at simulation time $t = 5.88$ s. For action sequences, the file *GroundTruthPositionsLog.txt* lists the frame of objects at timestamp t and can be used to evaluate the algorithms.

5 Conclusion

In this work, we have simulated an industrial setup equipped with multiple RGB stereo cameras and RGB-D devices. We have provided realistic signal information in terms of images with various noise levels partially resembling the characteristics of the image and depth sensors involved. In particular, this allowed us to provide ground truth information for pose estimation, stereo reconstruction and action recognition as examples of typical algorithms in machine vision and learning.

This has been a problem so far, since ground truth for such algorithms is very hard to define in real setups due to the problem of estimating object poses with higher certainty than cameras would allow. Moreover, even if more precise sensors would be used, there would still be the problem of assigning coordinate system to objects without requiring additional sensors with sufficient accuracy. VR on the other hand can provide suitable ground truth data—but generally faces the problem that images produced in VR are too ideal due to a insufficient modeling of noise and other effects. We could overcome this problem by a camera and sensor simulation which supports the generation of ideal images as well as images that closely resemble the characteristics of current RGB and RGB-D devices.

By means of this data, we were able to benchmark the precision of a pose estimation algorithm on images "close to reality". Besides benchmarking pose estimation, we have also used the data to investigate stereo reconstruction from multiple cameras as well as action recognition from sequences captured from tracking a dataglove in VR. In future work, we want to use the framework used here to investigate other vision algorithms relevant in the context of industrial platforms such as tracking under occlusions as well as the required number of cameras and camera positions to reach a certain amount of pose certainty. Nowadays such optimization processes usually are performed manually and require a lot of resources which make robot installations expensive. Our approach can help to reduce such cost by extensive prior optimization in VR. In addition, the VR-based approach allows for evaluating and furthering the application of algorithms from the fields of machine vision and learning for industrial assembly.

Acknowledgments The research leading to these results has received funding from the European Communities Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under Grant Agreement No. 269959, IntellAct.

References

1. Aksoy EE, Abramov A, Wörgötter F, Dellen B (2010) Categorizing object-action relations from semantic scene graphs. In: IEEE international conference on robotics and automation (ICRA), pp 398–405
2. Aksoy EE, Abramov A, Dörr J, Ning K, Dellen B, Wörgötter F (2011) Learning the semantics of object-action relations by observation. Int J Rob Res 30(10):1229–1249
3. Aldoma A, Tombari F, Di Stefano L, Vincze M (2012) A global hypotheses verification method for 3d object recognition. In: European conference on computer vision (ECCV), Springer, pp 511–524
4. Badler N (1975) Temporal scene analysis: conceptual descriptions of object movements. PhD thesis, University of Toronto, Canada
5. Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256
6. Billard A, Calinon S, Guenter F (2006) Discriminative and adaptive imitation in uni-manual and bi-manual tasks. Rob Auton Syst 54(5):370–384
7. Bradski G, Kaehler A (2008) Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media
8. Buch AG, Kraft D, Kamarainen JK, Petersen HG, Krüger N (2013) Pose estimation using local structure-specific shape and appearance context. In: IEEE international conference on robotics and automation (ICRA), pp 2080–2087
9. Butler DA, Izadi S, Hilliges O, Molyneaux D, Hodges S, Kim D (2012) Shake'n'sense: reducing interference for overlapping structured light depth cameras. In: Proceedings of the 2012 ACM annual conference on human factors in computing systems, ACM, pp 1933–1936
10. Collins K, Palmer AJ, Rathmill K (1985) The development of a European benchmark for the comparison of assembly robot programming systems. In: Robot technology and applications (Robotics Europe Conference), pp 187–199
11. Coppelia Robotics (2014) V-REP. http://www.coppeliarobotics.com/
12. Cos S, Uwaerts D, Hermans L (2006) Evaluation of STAR250 and STAR1000 CMOS image sensors. In: ESA international conference of guidance, navigation and control systems, pp 1–6
13. El-Laithy RA, Huang J, Yeh M (2012) Study on the use of microsoft kinect for robotics applications. In: IEEE symposium on position location and navigation (PLANS), pp 1280–1288
14. Emde M, Rossmann J (2013) Validating a simulation of a single ray based laser scanner used in mobile robot applications. In: International symposium on robotic and sensors environments (ROSE), pp 55–60
15. Farrell K, Okincha M, Parmar M (2008) Sensor calibration and simulation. In: SPIE 6817, Digital Photography IV, pp 1–9
16. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm ACM 24(6):381–395
17. Glover J, Popovic S (2013) Bingham procrustean alignment for object detection in clutter. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 2158–2165
18. Gupta A, Davis LS (2007) Objects in action: An approach for combining action understanding and object perception. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
19. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
20. Herrera C, Kannala J (2012) Joint depth and color camera calibration with distortion correction. IEEE Trans Pattern Anal Mach Intell 34(10):2058–2064
21. Hue C, Le Cadre JP, Perez P (2002) Tracking multiple objects with particle filtering. IEEE Trans Aerosp Electron Syst 38(3):791–812
22. Ikeuchi K, Suehiro T (1994) Toward an assembly plan from observation, part I: task recognition with polyhedral objects. IEEE Trans Rob Autom 10(3):368–385
23. Isard M, Blake A (1998) CONDENSATION: conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28
24. Khan Z, Balch T, Dellaert F (2005) MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Trans Pattern Anal Mach Intell 27(11):1805–1819
25. Koppula HS, Gupta R, Saxena A (2013) Learning human activities and object affordances from RGB-D videos. Int J Rob Res 32(8):951–970
26. Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset. In: IEEE international conference on robotics and automation (ICRA), pp 1817–1824
27. Martinez D, Alenya G, Jimenez P, Torras C, Rossmann J, Wantia N, Aksoy EE, Haller S, Piater J (2014) Active learning of manipulation sequences. In: IEEE international conference on robotics and automation (ICRA)
28. Mian AS, Bennamoun M, Owens R (2006) Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans Pattern Anal Mach Intell 28(10): 1584–1601
29. Microsoft (2012) Kinect for windows SDK version 1.5. http://www.kinectforwindows.org/
30. Microsoft (2014) Microsoft robotics developer studio 4. http://www.microsoft.com/en-us/download/details.aspx?id=29081
31. Open Source Robotics Foundation (2014) GazeboSim. http://gazebosim.org/
32. Papon J, Kulvicius T, Aksoy EE, Wörgötter F (2013) Point cloud video object segmentation using a persistent supervoxel world-model. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3712–3718
33. Pardowitz M, Knoop S, Dillmann R, Zöllner RD (2007) Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. IEEE Trans Syst Man Cybern B Cybern 37(2):322–332
34. Point Grey (2011) Bumblebee2 stereo camera. http://ww2.ptgrey.com/stereo-vision/bumblebee-2
35. Roßmann J (2012) eRobotics: the symbiosis of advanced robotics and virtual reality technologies. In: ASME international design in engineering technical Conference and computers and information in engineering (IDETC/CIE), vol 2, pp 1395–1402

36. Roßmann J, Ruf H, Schlette C (2009) Model-based programming ’by demonstration’— fast setup of robot systems. In: Advances in robotics research: Theory, implementation, application. Springer, pp 159–168
37. Roßmann J, Schlette C, Ruf H (2010) A tool kit of new model-based methods for programming industrial robots. In: IASTED international conference on robotics and applications (RA), pp 379–385
38. Roßmann J, Hempe N, Emde M, Steil T (2012a) A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications. In: German conference in robotics (ROBOTIK), pp 1–6
39. Roßmann J, Schlette C, Wantia N (2012b) Virtual reality providing ground truth for machine learning and programming by demonstration. In: ASME international design in engineering technical conference computers and information in engineering (IDETC/CIE), pp 1501–1508
40. Roßmann J, Steil T, Springer M (2012c) Validating the camera and light simulation of a virtual space robotics testbed by means of physical mockup data. In: International symposium on artificial intelligence, robotics and automation in space (i-SAIRAS), pp 1–6
41. Rusu RB, Cousins S (2011) 3d is here: Point Cloud Library (PCL). In: IEEE international conference on robotics and automation (ICRA)
42. Schou C, Carøe CF, Hvilshøj M, Damgaard JS, Bøgh S, Madsen O (2012) Human assisted instructing of autonomous industrial mobile manipulator and its qualitative assessment. In: AAU workshop on human-centered robotics, pp 22–28
43. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: International conference on pattern recognition (ICPR), vol 3, pp 32–36
44. Sridhar M, Cohn GA, Hogg D (2008) Learning functional object-categories from a relational spatio-temporal representation. In: European conference on artificial intelligence (ECAI), pp 606–610
45. Summers-Stay D, Teo CL, Yang Y, Fermüller C, Aloimonos Y (2012) Using a minimal action grammar for activity understanding in the real world. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4104–4111
46. Thomas U, Wahl FM (2001) A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)), pp 1458–1464
47. Tombari F, Salti S, Di Stefano L (2010) Unique signatures of histograms for local surface description. In: European conference on computer vision (ECCV), Springer, pp 356–369
48. Vermaak J, Godsill S, Perez P (2005) Monte Carlo filtering for multi target tracking and data association. IEEE Trans Aerosp Electron Syst 41(1):309–332
49. Yang Y, Fermüller C, Aloimonos Y (2013) Detection of manipulation action consequences (MAC). In: International conference on computer vision and pattern recognition (CVPR), pp 2563–2570
50. Zhang Z (2012) Microsoft kinect sensor and its effect. IEEE MultiMed 19(2):4–10