RGB-D Sensor Data Correction and Enhancement by Introduction of an Additional RGB View

Artashes Mkhitaryan and Darius Burschka

***Abstract*— RGB-D sensors are becoming more and more vital to robotics. Sensors such as the Microsoft kinect and time of flight cameras provide 3D colored point-clouds in real time, that can play a crucial role in Robot Vision. However these sensors suffer from precision issues, and often the density of the point-clouds they provide is insufficient. In this paper, we present a multi-camera system for correcting and enhancing data acquired from an RGB-D sensor. We have designed a system for two mobile robots. The main robot is equipped with an RGB-D sensor, and the auxiliary robot is equipped with a regular RGB camera. We perform the correction and the enhancement of the data acquired from the RGB-D sensor by placing the auxiliary robot in a close proximity to the target object and taking advantage of the established epipolar geometry. We have managed to reduce the relative error of the raw point-cloud from a Microsoft kinect RGB-D sensor by $74.5\,\%$ and increase its density up to 2.5 times.**

I. Introduction

The invention of RGB-D sensors such as the Microsoft kinect [20] and time of flight (ToF) cameras [21] has resulted in a tremendous leap ahead in many areas of computer/robot vision. The ability to acquire dense 3D coloured point-clouds in real time allowed significant advancements in dense stereo reconstruction, object registration, grasping, indoor navigation and motion capture among others.

For example, Holz et. al. [2] introduce an approach for mobile robots that allows them to create the obstacle map of the environment and classify the graspable objects by analysing the data acquired from an RGB-D sensor. In [1], a technique is presented in which a hierarchical, multi-view dataset of objects is created based on the RGB-D data. This is then used to identify and register objects. Schwarz et. al. [3], [4] introduce a system for human motion capture. The authors use dense 3D point-clouds in combination with the corresponding RGB images to create a graph-based representation of the target person and determine her or his pose.

Although RGB-D sensors have many advantages, it is not always possible to achieve the desirable or necessary performance without significant post-processing. Early ToF sensors had very small image resolution (204×204) and where introducing distortions in depth values around the corners of large objects due to double reflections [22]. Modern RGB-D sensors, like the Microsoft kinect, don't suffer from the resolution issue, however the average error of the depth value estimation can amount up to $4cm$ [8] depending on the distance to the target object.

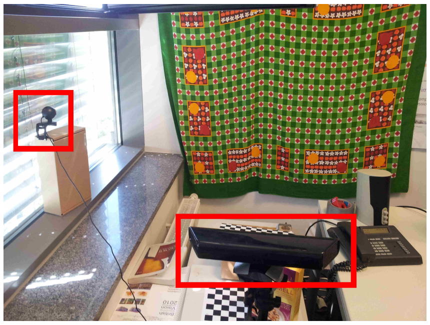


Fig. 1. Illustration of the experimental setup. The RGB-D sensor is on the bottom of the image while the RGB camera is on the left side.

In this paper, we introduce a system for enhancing and correcting the data acquired from an RGB-D sensor. The system is designed for two mobile robots, of which the main robot is equipped with an RGB-D sensor and the auxiliary robot is equipped with an RGB camera. The auxiliary robot is placed closer to the target object than the main robot and the two robots are calibrated against each other based on the data from the RGB and RGB-D sensors. In the second step we leverage the estimated epipolar geometry to perform a correction of the 3D points acquired from the RGB-D sensor. As a final step, we enhance the 3D point-cloud by taking advantage of the fact that the auxiliary robot with the RGB sensor is located closer to the target object, thus it has a more detailed view of the target than the main robot. Using this approach, we succeeded in correcting the positions of the points in the point-cloud by an average of $74.5\,\%$, and increase its density up to 2.5 times. In contrast to many temporal approaches such as kinect fusion [14], [18] our system provides the results based on only one time step.

II. Related Work

In the early RGB-D sensors (before the Microsoft kinect) correction or enhancement of the depth information was done with the help of a stereo couple. Zhu et. al. [12] introduced a technique where a ToF range sensor is used in combination with a stereo pair to compute the depth probability distribution, which in turn can be used to improve the accuracy of the point-cloud. Another approach for ToF and stereo pair fusion is introduced in [13]. Here the authors are acquiring an initial, “rough” point cloud from the ToF and enhancing it by optimising an energy function through

This work has been supported by an internal grant ”Real-Time Perception and Exploration with Collaborating Agents” of the German Aerospace Center (DLR)

A. Mkhitaryan and D. Burschka are with the Machine Vision and Perception Group, Technical University of Munich, Munich, Germany 80339 {mkhitary|burschka}@.in.tum.de

a probabilistic multi-view framework. In both of the above presented approaches, the authors use a ToF camera as an RGB-D sensor, which provides sparse inaccurate point-clouds. In the former case authors cope with the correction of the point-cloud, and in the latter they enhance the density, however neither of the approaches deal with both problems simultaneously. Gould et. al. [9] present a technique for object detection, by enhancing the 2D images. Where they use a a laser range scanner as an RGB-D sensor to acquire a sparse point cloud, which is later used with the RGB data from a video camera to achieve a high resolution, however the question of the accuracy of the point cloud is not addressed by the authors.

Later approaches are mainly using the Microsoft kinect as the RGB-D sensors, and are trying to correct or enhance the provided depth information, by either using an additional stereo reconstruction for the IR and RGB images, or by fusing the data acquired from the same sensor over time. Chiu et. al. [5] introduce a technique for object registration based on RGB-D data. Here the authors combine the depth map from the range sensor and the disparity map from the IR-RGB pair to increase the reliability of the depth estimate. This approach mainly focuses on reconstruction of the areas of the scene that have shiny or glossy surfaces, since the IR camera is not reliable in these cases. However the density and precision of the general point-cloud is not addressed. In [6] an approach is presented where the authors use a similar technique to increase the accuracy of the point-cloud. Henry et. al. [7] introduce a method for combining the visual features from the RGB camera and the shape-based alignment from the range sensor to construct reliable 3D maps of the environment. In [10], the authors propose the use of multiple kinect cameras to increase the reliability and robustness of image processing systems, by fusing the data acquired from each individual kinect. And last, Matyunin et. al. [11] use temporal filtering of the RGB-D sensor data to cope with occlusions and improve the temporal stability.

All of the above described approaches cope with the issue of enhancing or correcting the data from RGB-D sensors. However these approaches either deal with the correction of the point-cloud or its enhancement, and do not address both issues combined. In this paper we propose an approach that allows both the enhancement and the correction of the data acquired from an RGB-D sensor by introducing an additional RGB sensor. Our approach allows an automatic estimation of the epipolar geometry between the two sensors. It then performs the correction followed by an enhancement of the raw point-cloud data by taking an advantage of the estimated geometry.

The rest of the paper is organized as follows. First we give a detailed description of our approach in Section III, the three stages it consists of: calibration (Section III-A), correction (Section III-B) and enhancement (Section III-C). In the Section that follows we present the evaluation of our approach. Specifically in Section IV-A we describe the experimental setup, and give a quantitative (Section IV-B) and qualitative (Section IV-C) assessment. Finally, conclusions are presented in Section V.

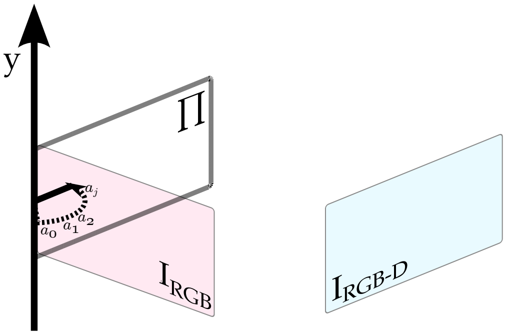


Fig. 2. Illustration of the calibration process.

III. Approach

We propose an algorithm for correction and enhancement of an RGB-D sensor (hereinafter main sensor) data, by an introduction of an auxiliary RGB camera (hereinafter auxiliary sensor). Since most of the known RGB-D sensors consist of two parts, an RGB camera, and a depth sensor, we model the main sensor similarly. We presume that the mentioned components of the main sensor are calibrated to each other, and that both intrinsic and extrinsic parameters are known. For simplicity, the colored point-cloud $\{p_i\}$ acquired from the main sensors will further on be referred to as *point-cloud* $\{p_i\}$.

The proposed algorithm can be divided into three parts: calibration, correction and enhancement. Initially, we compute the pose of the auxiliary sensor with respect to the main one, based on the images acquired from their RGB cameras and point-cloud $\{p_i\}$ (Section III-A). Further, we use the established epipolar geometry (Fig. 4, Section III-B) to correct the poses of the points in $\{p_i\}$. And finally we use the corrected point-cloud $\{p'_i\}$ and its projection to the image plane of the auxiliary sensor to increase the density of $\{p'_i\}$ (Section III-C).

A. Calibration

We establish the geometry between the sensors by solving a perspective n point problem [16]. This requires point correspondences between the point-cloud $\{p_i\}$ and the image of the auxiliary sensor. We obtain the correspondences by matching SURF [17] features from the RGB images acquired by both sensors. Since the matching algorithm for SURF features is sensitive to non-planar rotations, to insure maximum number of matches we inherit an iterative binary search like algorithm which establishes a rough rotation between the sensors along y axes. For stationary sensors the calibration step is only performed once.

Given the images I_{RGB} and I_{RGB-D} acquired from the auxiliary and main sensors respectively as well as the point-cloud $\{p_i\}$ we proceed as follows. First, we project the image I_{RGB} onto a virtual 3D plane Π (Fig. 2). Following, we rotate the plain Π for a range of angles $\{\alpha_j\}$ along its y axis. Here the range $\{\alpha_j\}$ covers a broad spectrum, however

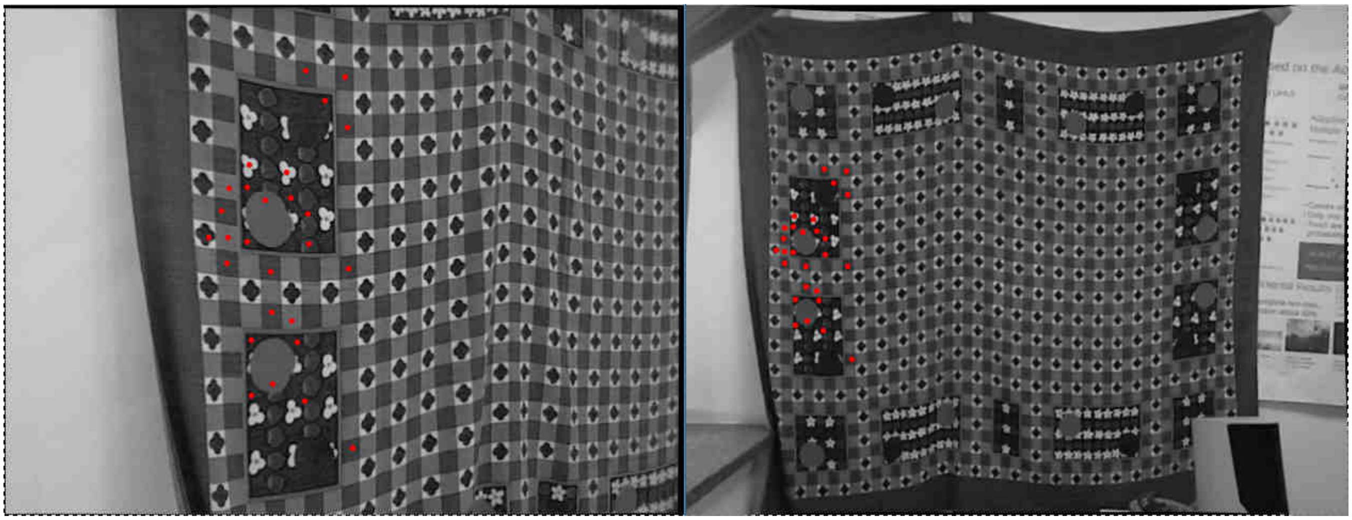


Fig. 3. Illustration of the calibration process. On the left is the image registered by the auxiliary sensor, and on the right the image acquired by the RGB camera of the main sensor. The red dots are the matching features on both images.

the step $||\alpha_j - \alpha_{j+1}||$ is large. For each α_j, we project the rotated plane Π to the image plane of the auxiliary sensor, which results in set of images $\{I^j_{RGB}\}$ and homographies $\{H^j\}$, such that:

$$H^j * I^j_{RGB}(u, v) = I_{RGB}(u, v) \qquad (1)$$

Next, for each I^j_{RGB} we extract SURF features and match them with SURF features from I_{RGB-D} (Fig. 3). Further, we define a new range of angles $\{\alpha'_j\}$ with a finer resolution (i.e. $||\alpha'_j - \alpha'_{j+i}|| < ||\alpha_j - \alpha_{j+1}||$) around the angle α_j that corresponds to the pair $\{I^j_{RGB}, I_{RGB-D}\}$ with most feature matches. We repeat this process iteratively until the difference between the largest amount of feature matches from previous and current iterations is smaller than a certain threshold.

Then we establish the rotation and translation $\{R, T\}$ between the two sensors by solving perspective-n-point problem for the $H^j * \{f(u, v)\}_j$ and $\{\hat{p_j}\} \in \{p_i\}$, where $\{f(u, v)\}_j$ are the matched features from I^j_{RGB} and $\{\hat{p_j}\}$ are the 3D points corresponding to the matched features from I_{RGB-D}.

$$H^j * \{f(u, v)\}_j = K * [R|T] * \{\hat{p_j}\} \qquad (2)$$

B. Correction

The main reason for inaccuracies in point cloud $\{p_i\}$ is the faulty depth value estimation by the RGB-D sensor [8]. We cope with this issue by taking advantage of the fact that a different view, i.e. the image plane of the RGB sensor, provides a better observability of the target object Fig. 4. Therefore, due to the depth inaccuracies the corresponding pixels to the projected points p_i would be shifted along the epipolar line Fig. 4. Note, that here the size of the shift depends on the angle ϕ between the sensors, the closer the angle to $\pi/2$ the better the observability.

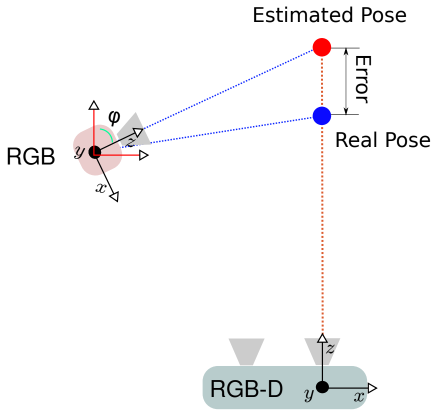


Fig. 4. Illustration of the faulty depth value estimation. Here the blue dot represents the real pose of the point, and the red one the estimated pose by the depth sensor. Note that from a different view, i.e. on the image plane of the auxiliary sensor (here depicted in red) the faulty estimate would be shifted along the epipolar line.

We project the point-cloud $\{p_i\}$ on the image plane of the RGB sensor:

$$I_P = K * (R * \{p_i\} + T) \qquad (3)$$

where K is the intrinsic matrix of the auxiliary sensor and I_P is the resulting image. Following, we compute the matches between all the pixels of I_P and I_{RGB} using Farneback's [15] algorithm. Next, for each matching pair of pixels $\{I_p(u_j, v_j), I_{RGB}(u'_j, v'_j)\}$ we compute the displacement values (du_j, dv_j):

$$\begin{aligned} du_j &= u_j - u'_j \\ dv_j &= v_j - v'_j \end{aligned} \qquad (4)$$

and for each $I_P(u_j + du_j, v_j + dv_j)$ pixel we compute $\hat{v}^j_P$ and $\hat{v}^j_d$ unit vectors:

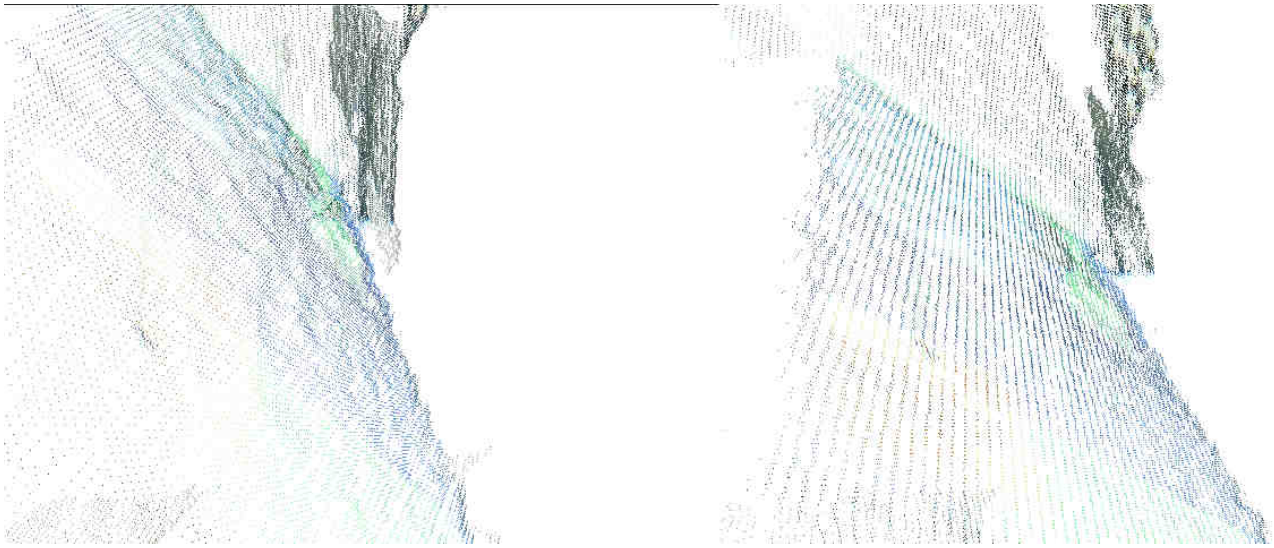


Fig. 5. Illustration of the point-cloud correction. Here the right side represents the raw point-cloud acquired from the RGB-d sensor, and the left side the same point-cloud after correction step. Note, that for the visualisation purposes the example is brought on a flat surface, and that the placement of the virtual camera has been done by hand, thus viewing angle is not exactly the same in both of the images.

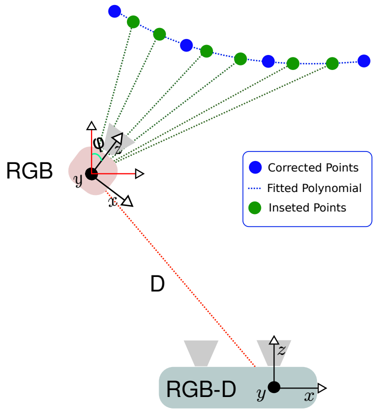


Fig. 6. Illustration of the point-cloud enhancement, here the blue dots represent the corrected points of the point-cloud, the green dashed line is the fitted polynomial, and the green dots are the new interpolated points of the enhanced point-cloud

$$\hat{v}_P^j = \frac{K*\begin{pmatrix} u_j + du_j \\ v_j + dv_j \\ 1 \end{pmatrix}}{\|K*\begin{pmatrix} u_j + du_j \\ v_j + dv_j \\ 1 \end{pmatrix}\|} \tag{5}$$

$$\hat{v}_D^j = \frac{p_j}{\|p_j\|}$$

where p_j is the 3D point that corresponds to $I_P(u_j, v_j)$ pixel, i.e. $I_P(u_j, v_j) = K*(R*p_j + T)$. The new corrected point-cloud $\{p'_j\}$ (see Fig. 5) is computed by triangulating the $\{\hat{v}_P^j\}, \{\hat{v}_D^j\}$ sets. For triangulation we use the algorithm proposed by Hartley et. al. in [19] (Chapter 12.5).

C. Enhancement

In the final stage of our algorithm we enhance point-cloud $\{p'_i\}$, i.e. we increase its density. Fig. 7 shows the difference between the image registered by the auxiliary sensor and the one obtained by projecting $\{p'_i\}$ on to it's image plane. Here the black dots represent perfect color matches between the two images. Note, that they are distributed sparsely in the image plane. This is due to the fact that the auxiliary sensor is located closer to the target object than the main sensor Fig. 9. Here, for each of the non-black pixels the only information missing for the 3D reconstruction of its pose is the depth value. Latter we interpolate based on the pose of the point p'_i that correspond to the neighboring black pixels. Note, that this kind of interpolation has an advantage over simple sub-sampling techniques of range images such as interpolation of range values, in that it provides a better preservation of the original shape, due to the fact that here the interpolation is done only for one degree of freedom (Fig. 6).

The interpolation is performed as follows. Around each of the non-black pixels we define a window. Further, we analyze the depth discontinuities within that window. This is done by comparing the absolute distances between points p'_i corresponding to nearest black-pixel neighbors, to a certain threshold. If there are no depth discontinues within the window, we interpolate the depth value of the non-black pixel by fitting a quadratic polynomial within the window. The latter is based on depth values of the 3D points corresponding to neighboring black pixels located within the window Fig. 6.

$$D(u, v) = q_1*u^2 + q_2*v^2 + q_3*u*v + q_4*u + q_5*v + p_6 \tag{6}$$

Where $D(u, v)$ -is the depth value of the colored pixel, (u, v) are its pixel coordinates, and q_i are the coefficients of the polynomial. An example of the final result is depicted in Fig. 8. Here, the borders of the target object are left

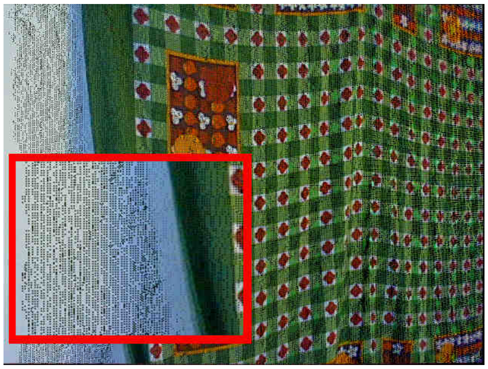


Fig. 7. Projection of the coloured 3D point-cloud to the image plane of the auxiliary sensor. The difference between the registered image on the RGB sensor and the projected, corrected coloured point cloud is evident. The black dots represent one-to-one colour matches. Note, that since the auxiliary sensor is located closer to the target object than the main sensor, the black dots are sparse.

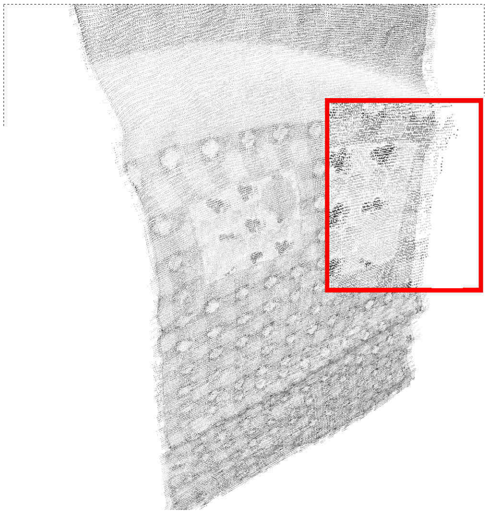


Fig. 8. Final result after the correction and the enhancement steps have been performed on Fig. 7. for the visualization purposes the content of the red rectangle has been zoomed and the borders of the object has not been processed to show the difference between the raw and processed data.

unprocessed to illustrate the difference between the row and processed data. To insure better visibility the content of the red rectangle has been zoomed.

IV. Experiments and Results

In this Section, we first give an overview of the experimental setup used to evaluate our approach, followed by a description of the conducted experiments. Finally, we present results for qualitative and quantitative assessment.

A. Experimental Setup

In the conducted experiments we used a Microsoft kinect [20] as the main sensor, and a colored web-cam with $640 \times 480px$ resolution as the auxiliary sensor. The sensors were placed as depicted in Figs. 1 and 9. We have conducted the experiments on a variety of target objects, such as a planar wall and a carton box with flat, planar sides to evaluate the performance of the correction stage, and a large carpet with a curvy surface to evaluate the performance of the enhancement step.

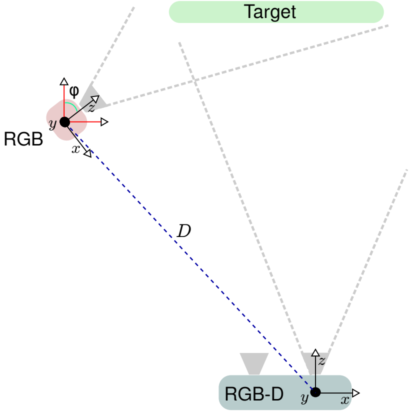


Fig. 9. System setup

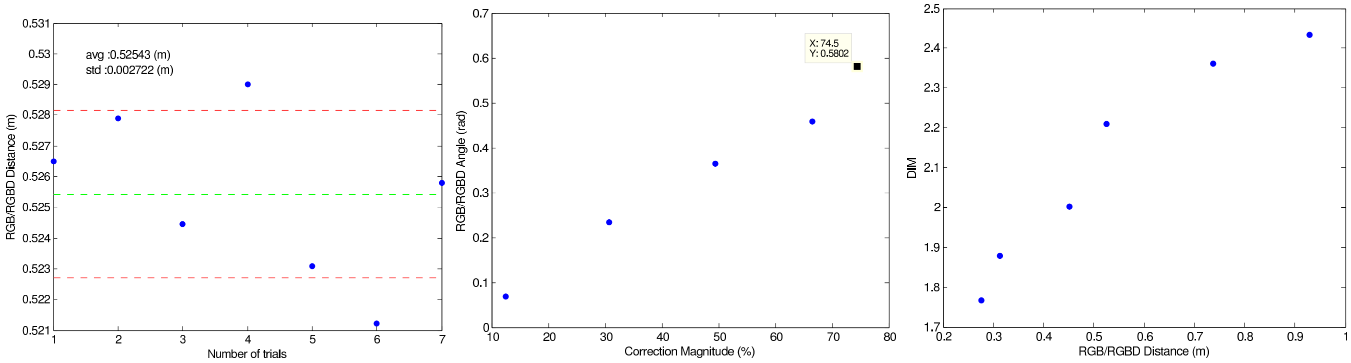
B. Quantitative Assessment

As mentioned in Section III, our approach can be divided into three logical stages, calibration, correction, and enhancement. Further we provide the qualitative assessment for each of the mentioned logical stages.

1) Calibration of Sensors: To evaluate the calibration of our system we ran the calibration stage several times with static placement of the sensors. For each iteration we have computed the distance $\|R' * T\|$ between the two sensors. The results are presented in Fig. 10(a). We have used a checker bored calibration pattern to obtain the true distance of $0.524\,78\,\mathrm{m}$ between the sensors. On average the calibration stage performed with an absolute error of $\pm 0.0022\,\mathrm{m}$ which is 0.5% of the absolute distance.

2) Evaluation of point-cloud correction: As is mentioned in Section III-B, the average distance between the raw and corrected corresponding points depends on the displacement of the pixel corresponding to point p_i along the epipolar line in the image plane of the auxiliary sensor. Since the observability of the error by the auxiliary camera mainly depends on the angle ϕ (Fig. 9), the evaluation of the correction stage is done in reference to it. As a target object, we chose an object that has a planar flat surface Fig. 5. Since the mathematical description of a planar surfaces is well known, it is possible to compute the ground truth analytically. The cameras where placed as is depicted in Fig. 9. Where the distance between the target object and the RGB-D sensor was found to be approximately $1.35\,\mathrm{m}$.

We perform the evaluation of the correction stage of our approach in reference to the angle ϕ and average correction magnitude. The latter is defined as the difference between the average displacement values Q and Q' of the processed and raw point-clouds from the planar surface of the target object. We have processed the point-cloud corresponding to the target object for a range of angles ϕ, and for each iteration we have computed the average correction magnitude. Each measurement was done 20 times, and for each time the average was computed over 3000 points. The results of our

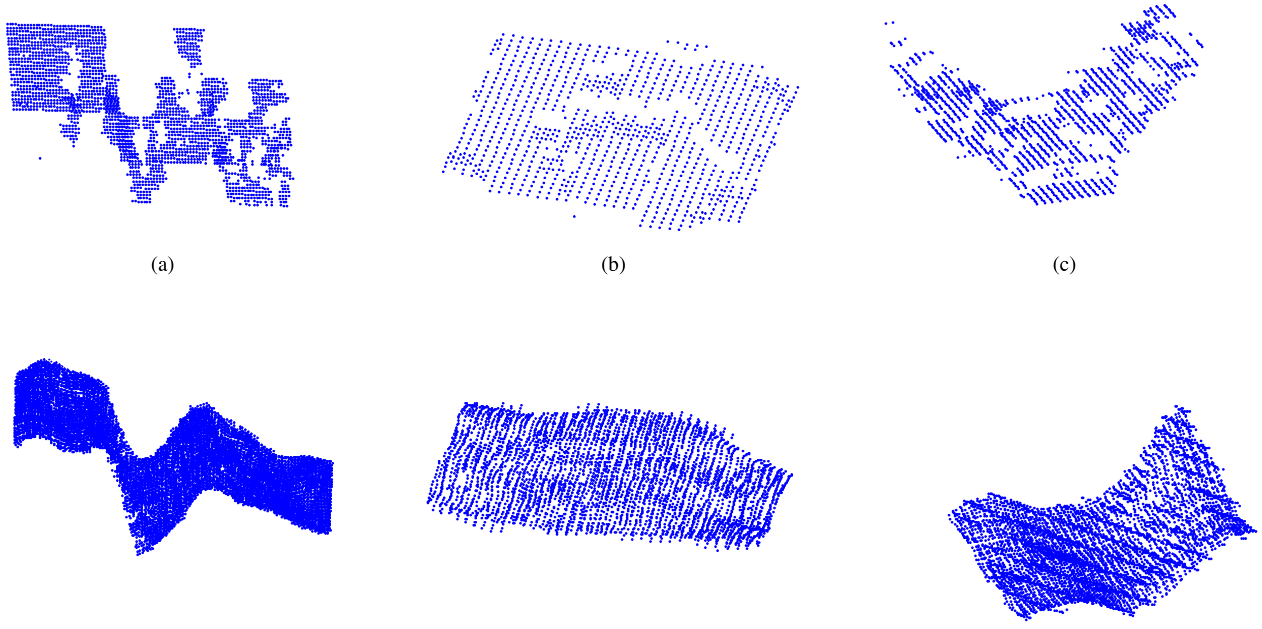


(a) Assessment of the accuracy of the calibration stage. Here, the absolute distance between the cameras is 0.524 78 m, the standard deviation of the estimated values is around 2.7 mm and worst case scenario error is around 3.5 mm

(b) Relative correction magnitude relation to the angle ϕ (Fig. 9) between the cameras.

(c) Relation of the density increase multiplier (DIM) to the distance D (Fig. 9) between the cameras.

Fig. 10.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 11. Depiction of the final result for qualitative assessment. Here Figs. 11(a) to 11(c) illustrate the raw data as acquired by the RGB-D sensor while Figs. 11(d) to 11(f) are the same patches after the correction and enhancement steps have been applied. The patches represent parts of the carpet illustrated in Fig. 8. Note, that as a result of processing some of the gaps from the row data have been filled and the patches obtained smoother and more realistic curvatures.

experiments are depicted in Fig. 10(b). We have achieved a relative average correction magnitude of around 74.5 %. Here, the absolute average displacement Q of the raw point-cloud was around 4 cm, and the absolute value of Q' was around 1 cm. Note that the average correction magnitude also depends on the distance of the auxiliary sensor to the target object, however the corresponding experiments has shown that this dependency is week and neglectful in comparison to dependency to the angle ϕ, thus the results of those experiments are omitted.

3) Evaluation of Enhancement: To evaluate the enhancement step, we first introduce a metric called density increase multiplier (DIM). The latter is defined as the ratio between the amount of points after enhancement to the amount of points before. As was mentioned the increase in density after the enhancement stage is due to the fact that the auxiliary sensor is located closer to the target object than the main sensor. Thus, we evaluate the enhancement stage in reference to the distance D between the sensors and the DIM. This is realized by keeping the main sensor static and moving the auxiliary sensor towards the target object. For each distance we performed 20 iterations to estimate the average DIM. The results of the experiments are illustrated in the Fig. 10(c). The original density of the patch was found to be around 3600 points. In the best case scenario, where the cameras were 1 m apart from each other, we achieved up to 2.5 times increase in the density of the point-cloud.

C. Qualitative Assessment

1) Correction Step: For qualitative assessment of the correction step we use a carton box with planar flat sides. This is done so it is easier for a naked eye to notice the changes. The results are illustrated in Fig. 5. The left side of the image depicts the raw point-cloud as acquired by the RGB-D sensor. The right side is the same point-cloud after a correction stage has been performed (the colors of the illustration are inverted, to insure a better visibility on printed formats). Note, that in both cases the viewing angle was adjusted by hand, thus a reasonable displacement of the scene is to be assumed.

2) Assessment of the Entire Algorithm: Fig. 11 illustrates the improvements achieved in different patches of the point-cloud. These patches were obtained using the carpet depicted in Fig. 8 as a target object. The latter was chosen due to its curvy structure, which makes it easier to observe the improvements of the entire approach. Figs. 11(a) to 11(c) illustrate the unprocessed patches, while Figs. 11(d) to 11(f) are the respective patches after the correction and enhancement. Note, that the processed patches have smoother and more realistic representation of the curves, as well as that some of the gaps notable in the raw patches has been filled.

V. Conclusion

Currently, RGB-D sensors play a major role in such robotics applications as navigation, grasp planning, object recognition and perception. In most of these applications the precision of the point-clouds is a decisive factor in overall system performance. In this paper we presented a system for correcting and enhancing the data acquired from an RGB-D sensor by introducing an auxiliary RGB sensor. We have managed to correct the accuracy of the point-cloud from RGB-D sensor by an average magnitude of 74.5 %, and increase its density up to 2.5 times. Note that the Microsoft kinect sensor, is one of the most used RGB-D sensors to our knowledge, and has an average absolute error of 4 cm in depth estimation [8].

References

[1] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox “A Large-Scale Hierarchical Multi-View RGB-D Object Dataset”, IEEE ICRA, May 9-13, 2011, Shanghai, China

[2] Dirk Holz , Stefan Holzer , Radu Bogdan Rusu , and Sven Behnke “Real-Time Plane Segmentation using RGB-D Cameras”, In Proceedings of 15th RoboCup International Symposium, 2011, Springer, pp. 306-317, 2012

[3] Loren Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab “Estimating human 3d pose from time-of-flight images based on geodesic distances and optical flow”, IEEE Face and Gesture Recognition (FG), March, 2011, Santa Barbara, USA.

[4] Loren Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab “Human skeleton tracking from depth data using geodesic distances and optical flow”, In Image and Vision Computing,Elsevier, 2012.

[5] Wei-Chen Chiu, Ulf Blanke and Mario Fritz “Improving the Kinect by Cross-Modal Stereo”, 22nd British Machine Vision Conference (BMVC), September, 2011, Dundee, UK.

[6] Ke Xu, Lei Qin, and Lin Yang “RGB-D Fusion Toward Accurate 3D Mapping”, IEEE International Conference on Robotics and Biomimetics, December 7-11, 2011, Phuket, Thailand.

[7] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren and Dieter Fox “RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments”, International Symposium on Experimental Robotics ISER, 2010.

[8] K. Khoshelham “Accuracy Analysis of Kinect Data”, ISPRS workshop laser scanning, August 29-31, 2011, Calgary, Canada.

[9] Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng and Daphne Koller “Integrating Visual and Range Data for Robotic Object Detection”, Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications M2SFA2, 2008, Marseille, France.

[10] Lazar Sumar and Andrew Bainbridge-Smith “Feasibility of Fast Image Processing Using Multiple Kinect Cameras on a Portable Platform”, Department of Electrical and Computer Engineering, Univ. Canterbury, New Zealand.

[11] Sergey Matyunin, Dmitriy Vatolin, Yury Berdnikov and Maxim Smirnov “Temporal Filtering for Depth Maps Generated by Depth Camera”, The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011.

[12] Jiejie Zhu, Liang Wang, Ruigang Yang and James Davis “Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps”, Computer Vision and Pattern Recognition CVPR, June 23-28, 2008.

[13] Young Min Kim, Christian Theobalt, James Diebel, Jana Kosecka, Branislav Miscusik and Sebastian Thrun “Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction”, IEEE Computer Vision Workshops ICCV Workshop, 2009.

[14] David Molyneaux “KinectFusion rapid 3D reconstruction and interaction with Microsoft Kinect” International Conference on the Foundations of Digital Games Pages 3-3, 2012, NY, USA.

[15] Gunnar Farneback “Two-frame motion estimation based on polynomial expansion” Lecture Notes in Computer Science, 2003, (2749), 363-370.

[16] Vincent Lepetit, Francesc Moreno-Noguer and Pascal Fua “EPnP: An Accurate O(n) Solution to the PnP Problem” International Journal of Computer Vision Volume 81, Number 2 (2009), 155-166,

[17] Herbert Bay, Tinne Tuytelaars and Luc Van Gool “SURF: Speeded Up Robust Features” Lecture Notes in Computer Science ECCV, 2006, Volume 3951/2006, 404-417,

[18] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera” ACM Symposium on User Interface Software and Technology, October 2011

[19] Richard Hartley and Andrew Zisserman “Multiple View Geometry in computer vision” Cambridge University Press, 2008

[20] Microsoft kinect: “http://www.xbox.com/en-US/KINECT”

[21] Time of Flight Sensor: “http://www.mesa-imaging.ch/”

[22] ToF Manual: “http://www.mesa-imaging.ch/dlm.php?fname=customer/Customer_CD/SR4000_Manual.pdf”