Improved ICP-based Pose Estimation by Distance-aware 3D Mapping

Hani Javan Hemmat, Egor Bondarev, Gijs Dubbelman and Peter H. N. de With
Eindhoven University of Technology, Eindhoven, The Netherlands

Keywords: 3D Reconstruction, Voxel-models, Camera-pose Estimation Weighting Strategy, Truncated Signed Distance Function (TSDF), Simultaneous Localization and Mapping (SLAM), Low-cost Depth Sensor.

Abstract: In this paper, we propose and evaluate various distance-aware weighting strategies to increase the accuracy of pose estimation by improving the accuracy of a voxel-based model, generated from the data obtained by low-cost depth sensors. We investigate two strategies: (a) *weight definition* to prioritize prominence of the sensed data according to the data accuracy, and (b) *model updating* to determine the influential level of the newly captured data on the existing synthetic 3D model. Specifically, we propose Distance-Aware (DA) and Distance-Aware Slow-Saturation (DASS) updating methods to intelligently integrate the depth data into the 3D model, according to the distance-sensitivity metric of a low-cost depth sensor. We validate the proposed methods by applying them to a benchmark of datasets and comparing the resulting pose trajectories to the corresponding ground-truth. The obtained improvements are measured in terms of Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) and compared against the performance of the original Kinfu. The validation shows that on the average, our most promising method called DASS, leads to a pose estimation improvement in terms of ATE and RPE by 43.40% and 48.29%, respectively. The method shows robust performance for all datasets, with best-case improvement reaching 90% of pose-error reduction.

1 INTRODUCTION

The 3D sensing and mapping of arbitrarily-shaped environments is a highly active research topic, as it comes as a pre-requisite for various currently prominent research domains, such as 3D shape acquisition and modelling, surface generation and texturing, as well as localization and robot vision. During recent years, the advent of low-cost, hand-held, and accurate 3D sensors along with the introduction of powerful general-purpose GPUs, lead to first solutions running a real-time 3D reconstruction process for relatively large and complex indoor and outdoor environments. In a 3D reconstruction process, the 3D volumetric model-generation methods play an outstanding role influencing the accuracy of obtained results, in a mutual cooperation with the camera-pose estimation methods. For localization based on 3D features, generation of an accurate 3D synthetic model containing higher details leads to a more accurate pose estimation and the associated process results in a higher quality 3D model. This correlation between 3D synthetic model and camera-pose estimation is a "chicken and egg" problem in the Simultaneous Localization and Mapping (SLAM) domain.

Various volumetric structures, for modelling 3D spaces have been introduced to represent scene geometry, including the Point Cloud structure (Rusu and Cousins, 2011), the Signed Distance Function (SDF) (Curless and Levoy, 1996), voxel-based (Newcombe et al., 2011a; Newcombe et al., 2011b), surfel-based (Chang et al., 2011; Andersen et al., 2010), and the Octree-based models (Zeng et al., 2012). The SDF model has been used to integrate the depth images into a synthetic 3D model (Kubacki et al., 2012; Ren and Reid, 2012). The RGB-D mapping, combined with visual SLAM, pose-correction, and optimization approaches, are largely investigated in recent years (Henry et al., 2012; Henry et al., 2010; Engelhard et al., 2011; Endres et al., 2012; Kümmerle et al., 2011; Hornung et al., 2013). To register new depth data with a 3D synthetic model, the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992), or similar iterative approaches (Bylow et al., 2013), have been exploited by the applications taking advantage of low-cost depth sensing devices. These applications include KinectFusion (Newcombe et al., 2011a; Izadi et al., 2011), Kintinious (Whelan et al., 2012; Whelan et al., 2013), open source KinFu (PCL, 2011), and KinFu Large Scale (Bondarev et al., 2013). All of them use a Truncated version of the SDF model (TSDF), in order to reconstruct 3D geometry of the

Javan Hemmat H., Bondarev E., Dubbelman G. and de With P..
Improved ICP-based Pose Estimation by Distance-aware 3D Mapping.
DOI: 10.5220/0004716403600367
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 360-367
ISBN: 978-989-758-009-3

environment and utilize the Kinect as the depth sensor. The TSDF model provides averaging of the complete set of the sensor data over time. While recent investigation of the Kinect technology has unveiled that it is robust to incidence angle, ambient light, and radiometric influences, the sensor is less accurate for large distance measurements (Chow et al., 2012; Khoshelham and Elberink, 2012; Khoshelham, 2011).

In the conventional TSDF model, a 3D array of voxels is used to form a synthetic 3D model, representing the sensed environment. The newly extracted points of each depth frame are integrated into the 3D synthetic TSDF model. The TSDF model takes advantage of a weighting strategy to discriminate newly upcoming sensed data, by assigning a higher weight to less noisy data. In the KinectFusion and KinFu, a weight value of unity is used for each valid point. We expect a conspicuous enhancement in terms of model quality by proposing a weighting strategy based on the intrinsic distance-sensitivity of the Kinect sensor. Due to the unity weight in the conventional TSDF model, the newly sensed data is integrated into the 3D model without considering its accuracy. As a subsequent result, the objects in the synthetic 3D model can be deformed by the new depth data containing information of the same objects, captured from a further distance. According to the distance sensitivity of the depth sensors, an explanation for this deformation is that the updating mechanism may overwrite the synthesized reasonably accurate data with less accurate data over time. The consequence of this model degradation has a clear impact on the quality of the camera-pose estimation algorithm. For every newly captured depth frame, the camera pose is calculated, based on the current state of the synthetic 3D model. Therefore, an inaccurate 3D model containing less details leads to a more erroneous pose estimation for each new frame. The contribution of this work is to propose a weighting strategy for the TSDF model to prevent the model deformation and, by doing so, improve the pose estimation quality accordingly. This process involves the following steps. First, a weight definition is introduced to evaluate the quality of each point of the new depth frame. Second, we present an updating method with an intelligent decision making to integrate the new depth data into the synthetic 3D model. The proposed weighting strategy has been implemented in the KinFu framework. We have evaluated the introduced methods on a group of datasets with ground-truth trajectory information. The obtained results have been compared to the ground truth in terms of Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) in both translation and rotation.

The paper is structured as follows. Section 2 introduces the weighting definitions and strategies. The experiments are presented in Section 3. Section 4 elaborates on the analysis and discussion. Section 5 concludes the paper.

2 WEIGHTING STRATEGIES

2.1 Conventional TSDF Model

In the conventional applications exploiting TSDF to model the 3D geometry (Newcombe et al., 2011a; PCL, 2011), each voxel is represented as a pair of *distance value* (D_i) and *accumulated weight* (W_i), indicating the truncated distance value to the closest surface and the weight for this value, respectively. This structure averages the captured depth data, influencing the voxel model after i frames. The voxel x of the model is updated by the corresponding pair of truncated distance value (d_{i+1}) and weight (w_{i+1}) of the *(i+1)*th depth frame, according to the following two equations:

$$D_{i+1}(x) = \frac{W_i(x)D_i(x) + w_{i+1}(x)d_{i+1}(x)}{W_i(x) + w_{i+1}(x)} \ , \quad (1)$$

$$W_{i+1}(x) = W_i(x) + w_{i+1}(x) \ . \quad (2)$$

Parameter d_i is the truncated distance value calculated for voxel x, based on the corresponding newly sensed valid depth point. Variable w_i is the weight for d_i. Equation (1) defines the integration of d_i into the corresponding voxel x. The weight for voxel x is accumulated in W_i based on Equation (2). Assigning $w_{i+1} = 1$ for each valid point found in the $(i+1)$th frame, leads to simple averaging over time. The constant unity value for the weight influences the 3D model updating process in the following way. The Kinect senses objects in close proximity more accurately (Chow et al., 2012; Khoshelham and Elberink, 2012; Khoshelham, 2011). Due to this distance sensitivity, the model is degraded by overwriting the more accurate data (closer range) by less accurate data (farther). Therefore, the objects located close to the sensor ($\leq$ 1 m) are modelled appropriately, while the objects located at a farther distance ($\geq$ 2 m) are deformed or even destroyed.

2.2 Weight Definitions and Updating

Here, we introduce more competent definitions of the weight factor to distinguish between the sensed data of close and far distances. We have evaluated various weight definitions (publicly available at http://vca.ele.tue.nl/demos/MMM14/mmm14.html). The common feature for all definitions is that a higher weight is assigned to close distances. Due to the Kinect characteristics, we have considered a valid range for depth data, defined between a maximum and a minimum distance, d_{max} and d_{min}, respectively. Besides this, the weight is bounded between 0 and a maximum weight W_{max}. The following equation defines a weight that, according to our experiments, provides the best performance among various alternatives, which is

$$weight_{depth\_point}(x) = [\frac{\frac{1}{d(x)^2} - \frac{1}{d^2_{max}}}{\frac{1}{d^2_{min}} - \frac{1}{d^2_{max}}}] * W_{max}\,. \qquad (3)$$

For each valid depth value x with distance of $d(x)$ in the range between d_{min} and d_{max}, the corresponding weight is calculated as a value between 0 and W_{max}.

In conventional implementations of the TSDF model, the updating process is straightforwardly performed with constant unity weight. The weight definition from Equation (3) enables us to discriminate between closer and farther distances. Therefore, such weight definition enables intelligent updating of the TSDF model via prevention of more accurate values being overwritten by less accurate data. In the remainder of this section, we introduce two updating algorithms to guarantee that the synthetic 3D model is updated with the most accurate data available during the updating process.

Distance-Aware (DA) Updating Method

In this method, we exploit the following rule to update each voxel value in the synthetic 3D model: *"if a voxel has already been updated by a truncated distance value with a higher weight, never update it with a truncated distance value of lower weight"*. The DA updating method is formulated as:

$$Flag(v,x) = weight_{new}(x) \geq r\% \times weight_{LMU}(v), \qquad (4)$$

$$Update(v,x) = \begin{cases} \text{Integrate } x \text{ into } v & \textit{if}\,(Flag(v,x)), \\ \text{Discard } x, \text{ keep } v & \text{otherwise.} \end{cases} \qquad (5)$$

The last maximum updated weight $weight_{LMU}(v)$ contains the value for the maximum weight that voxel v has ever been updated with. To enhance the robustness to noise, we have proposed a tolerance range r, with $0 \leq r \leq 100$. This leads to the integration of the distance values close to the $weight_{LMU}(v)$ affected by noise.

The $Update(v,x)$ function *conditionally* updates the 3D model according to Equation (1). Throughout the updating process, the accumulated weight value for each voxel $W_i(x)$ is collected, based on Equation (2). The weight of each new voxel value w_i is equal to the weight of the corresponding distance value (Equation (3)). Therefore, $0 \leq w_i \leq W_{max}$. In comparison with the conventional TSDF implementation with $w_i = 1$, the introduced method grows faster with the *accumulated weight value* for each voxel. The accumulation of the weight values rapidly exceeds the 1-Byte specification used for the conventional algorithm with $w_{i+1} = 1$. Using a 2-Byte word for W_i can circumvent this, but it leads to a larger memory requirement, so that we introduce the DASS method below.

Distance-Aware Slow-Saturation (DASS) Updating Method

The DASS method is an alternative to maintain the framework of the conventional TSDF implementation and avoid the rapid saturation of the accumulated weight in the DA method. The DASS method is similar to the DA method, except for the way it accumulates the weight for each voxel. To this end, the DASS method utilizes the weight definition of Equation (3) for the $Update(v,x)$ function to conditionally update the 3D model, similar to the DA method. However, in contrast with the DA method, in DASS the new weight w_{i+1} is set to unity to calculate the accumulated weight value W_{i+1}. Assigning $w_{i+1} = 1$ in the DASS method suppresses the quick saturation of the accumulated weight value, while the $Update(v,x)$ function guarantees the intelligent updating.

It should be mentioned that both proposed methods have the benefit of the distance-based updating feature compared to the conventional algorithms. The DA and DASS algorithms still enable efficient memory management, while employing this advantageous feature.

3 EXPERIMENTS

3.1 Implementation

We have used the original framework of the open source KinFu implementation from the Point Clouds Library (PCL, 2011) to implement the proposed methods. We have exploited the original structure and introduced the new weight definitions and updating algorithms as discussed above. We have utilized a previously allocated but unused byte in the original structure to store the $weight_{LMU}$ part of the W_i.

Table 1: Average performance of the proposed methods applied to the benchmark 17 datasets.

Benchmark	ATE (mm)			RPE (mm)			Improvement (%)	
Frames	KinFu	DA	DASS	KinFu	DA	DASS	DA	DASS
1162.71	229.16	191.64	129.71	295.42	218.44	152.78	21.22	45.84

a	f1_desk
b	f1_floor
c	f1_room
d	f1_teddy
e	f1_xyz
f	f2_desk
g	f2_xyz
h	f3_household_p1
I	f3_household_p2
j	f3_notexture_far
k	f3_notexture_near
l	f3_texture_far
m	f3_texture_near
n	f1_desk2
o	f1_plant
p	f2_flower
q	f3_teddy

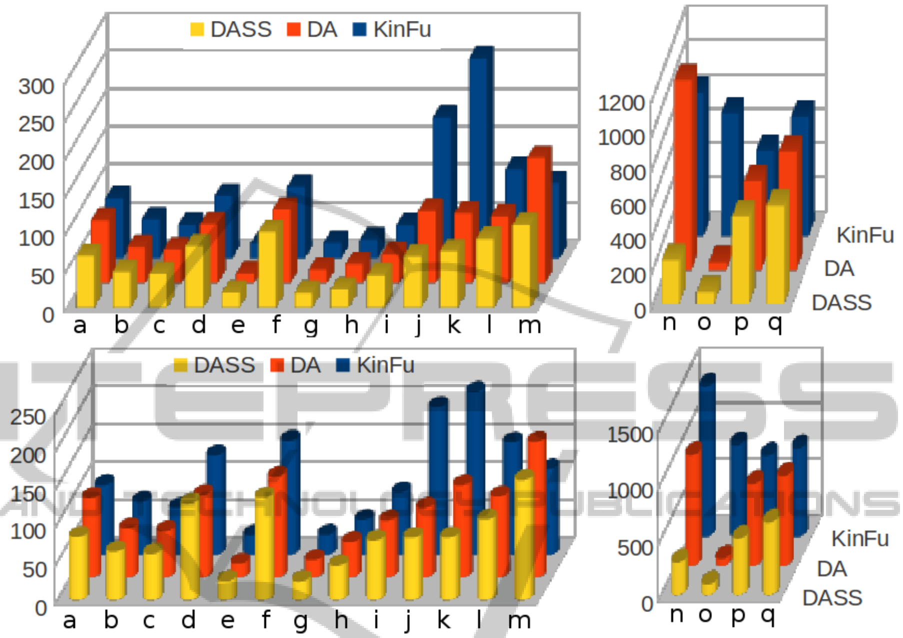

Figure 1: Detailed ATE and RPE values for the DA, DASS, and original KinFu methods applied to 17 datasets of the TUM benchmark (mm). First row depicts the ATE metric and the RPE is illustrated in second row.

3.2 Dataset

We have evaluated the proposed algorithms on the TUM RGB-D benchmark (Sturm et al., 2012). For each dataset of the benchmark, we have compared the resulting pose trajectories of the DA, DASS, and original KinFu methods against the corresponding ground-truth trajectory. For the r parameter in Equation (4), we have explored a range of various settings and selected a value that leads to the highest quality ($r = 80$).

3.3 Evaluation Approach

We have used two prominent methods to compare the estimated trajectory obtained by the DA, DASS, and KinFu methods to the ground truth: the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE). The ATE measures the difference between points of the true and the estimated trajectories. The RPE is the error in the relative motion between two pairs of consecutive poses.

4 ANALYSIS AND DISCUSSION

4.1 Quantitative Evaluation

As depicted in Figure 1, for most of the datasets, the DASS and/or DA methods outperform the original weighting strategy of the KinFu for both the ATE and RPE. Due to Table 1, in comparison with the original KinFu, the DA method reduces on the average the ATE and RPE by 16.37% and 26.06%, respectively. Even better than this, the DASS method improves the pose estimation by reducing the ATE and RPE on the average by 43.40% and 48.29%, respectively.

An interesting finding is that the DA and DASS methods improve the KinFu pose-estimation performance more in terms of RPE rather than ATE. One reason for this is due to the different nature of the ATE and RPE. The ATE is an appropriate metric to measure the performance of visual SLAM systems, while the RPE is well-suited for measuring the drift of a visual odometry system. Since there is no loop-closure detection in the original KinFu, the ATE suffers from this, whereas locally improved quality and accuracy of the 3D model, leads to a better performance of the ICP as a pose estimator. Hence, the drift error in the poses reduced so that the RTE reduction is more significant than the ATE.

Table 2 focuses on details of the ATE metric. Comparing the DA method to the original KinFu, the ATE is reduced by 17.85% and 25.07% in terms of mean and median, respectively. Besides this, the standard deviation is improved by 10.62%. For the DASS method, the improvement is even higher. In comparison with the KinFu, the DASS method improves the ATE by 47.82%, 54.80%, and 30.26% in terms of mean, median, and standard deviation, respectively.

Table 2: Detailed ATE values for the DA, DASS, and original KinFu methods applied to 17 datasets of the TUM benchmark (mm).

Method	rsme	mean	median	std
KinFu	229.16	199.26	173.76	109.41
DA	191.64	163.69	130.20	97.79
DASS	129.71	103.98	78.53	76.31

One observation is that on the average, the DASS method improves the pose estimation approximately two times more than the DA method. This performance difference is explained by the number of frames that is used for the accumulation of the weight metric. If the number of frames involved in the accumulation grows, then the intrinsic noise component of the depth sensor is more averaged and its influence decreases, thereby improving the quality.

We investigate the RPE improvement with respect to translation and rotation (see Tables 3 and 4). Due to Table 3, the translation error for the DA method is reduced by 22.30%, 45.14%, and 17.36% in terms of rsme, median, and standard deviation, respectively. The translation error reduction is higher for the DASS method by 46.76%, 56.26%, and 44.46% for the same parameters.

Table 3: Detailed RPE values expressed as translation error, for the DA, DASS, and original KinFu methods applied to 17 datasets of the TUM benchmark (mm).

Method	rsme	mean	median	std
KinFu	386.32	295.42	254.22	245.64
DA	300.17	218.44	139.47	203.01
DASS	205.67	152.78	111.20	136.42

Regarding the improvement of the rotation error shown in Table 4, the DA method reduces the error by 22.72%, 24.07%, 7.43%, and 21.48% in terms of rsme, mean, median, and standard deviation, respectively. The DASS method improves the rotation error by 29.58%, 35.35%, 35.82%, and 22.22% for the same parameters.

An interesting observation is that according to the rsme, mean, and standard deviation metrics for the RPE, there is a higher improvement in terms of translation rather than rotation by a factor of 1.72. For the median, this is opposite by a factor of 3.87. Besides this, there is a huge difference between the mean and the median for rotation error: the mean value is 68 times larger than the median. We explain these discrepancies by relatively large outliers which influence other metrics than the median.

Table 4: Detailed RPE expressed as rotation error, for the DA, DASS, and original KinFu methods applied to 17 datasets of the TUM benchmark (degree).

Method	rsme	mean	median	std
KinFu	17.36	13.67	0.19	10.64
DA	13.42	10.38	0.17	8.36
DASS	12.23	8.84	0.12	8.28

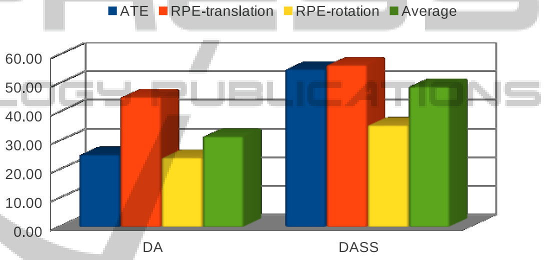

Figure 2: Percentage of error reduction for the DA and DASS methods compared to the original KinFu based on the median metric.

Another finding is that for both methods, the highest improvements result from using the median metric. The reason for this is that the mean, compared to the median, is not a robust computation technique, since it is sensitive to extreme scores and largely influenced by outliers. In the exploited benchmark, as illustrated in Figure 1, there are some extreme error values for the datasets of f1_desk, f2_flower, and f3_teddy, which as outliers influence the mean metric. Figure 2 depicts the improvements when using the median metric. Using the average median, the DA and DASS methods elevate the pose-estimation process by 31.43% and 48.80%, respectively.

4.2 Correspondence between Pose Estimation and Model Accuracy

The pose-estimation improvement achieved by the proposed methods is indirectly enabled by the accuracy enhancement of the synthetic 3D model. A more accurate 3D model can elevate the result of the pose-estimation algorithm by providing more details of the 3D geometry. In turn, a more precise estimation of pose increases quality and accuracy of the 3D model. In the following paragraphs, we illustrate this mutual dependence between the 3D model quality and pose-estimation accuracy.

As an extreme case, in the f1_plant dataset the pose trajectories and the snapshots of the corresponding meshes obtained by the DA, DASS, and the original KinFu methods are depicted in Figures 3 and 4, respectively. The pose trajectories for the DA and DASS methods are relatively close, while the former performs better. The synthetic 3D model obtained by the DA is more accurate and contains more details compared to one obtained by the DASS method. For the KinFu method, the destruction of the 3D model is caused by the appearance of large errors in pose estimation.

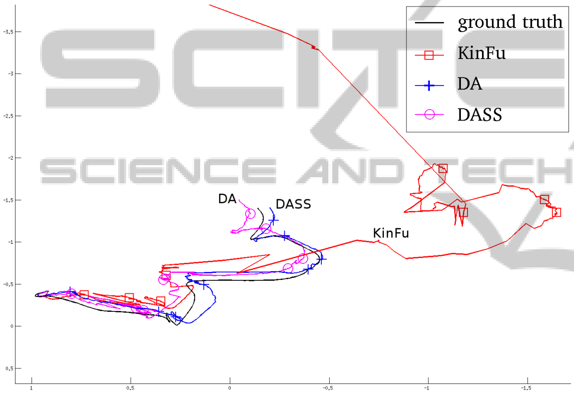

Figure 3: Camera pose trajectories for the DA, DASS, and original KinFu compared to the ground truth (f1_plant data).

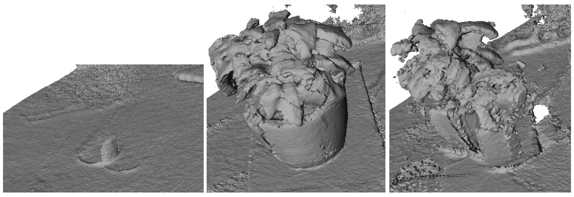

Figure 4: From left to right: snapshots of the final 3D synthetic meshes for the f1_plant dataset obtained by the original KinFu, DA, and DASS methods, respectively.

Regarding the moderate case of the f3_notex_near dataset as shown in Figures 5 and 6, there are no significant deformations in the corresponding meshes of the DA and DASS methods, while for the KinFu, appearance of a false object significantly deforms the 3D model. This is reflected as a huge drift in the corresponding pose trajectories of the original KinFu.

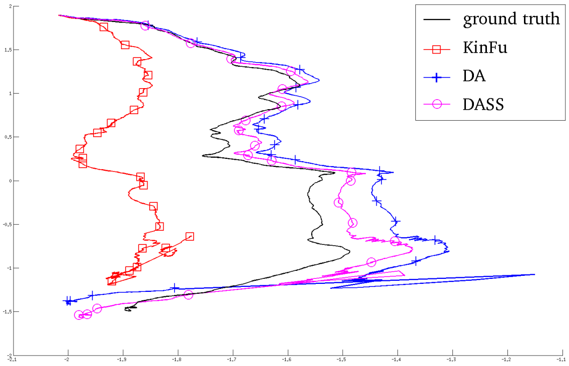

Figure 5: Camera pose trajectories for the DA, DASS, and original KinFu compared to the ground truth for the f3_notex_near dataset.

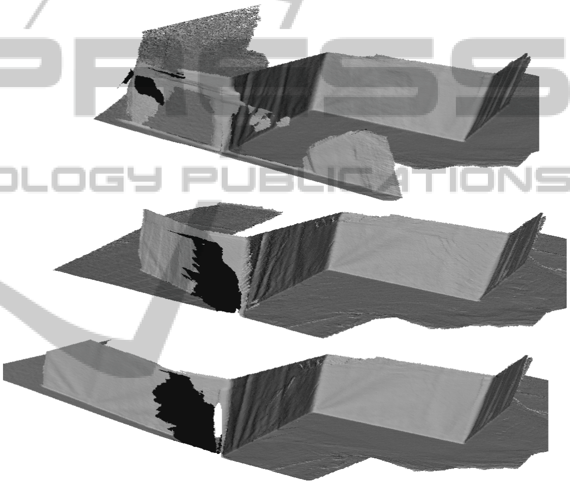

Figure 6: From top to bottom: snapshots of the final 3D synthetic meshes for the f3_notex_near dataset obtained by the original KinFu, DA, and DASS methods, respectively.

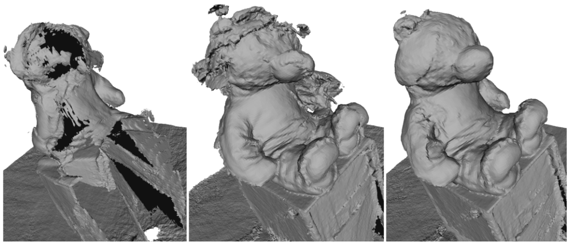

Figure 7: From left to right: snapshots of the final 3D synthetic meshes for the f3_teddy dataset obtained by the original KinFu, DA, and DASS methods, respectively.

Figure 7 depicts the corresponding meshes for the f3_teddy dataset, where the original weighting strategy of the KinFu outperforms the DA method in terms of RPE. According to the slightly more improved ATE, the corresponding 3D model of the DA method is less deformed compared to the one obtained by the original KinFu method. As shown, the DASS method leads to the best 3D model and the smallest ATE and RPE errors.

Another extreme case occurs in Figure 8, which illustrates the snapshots of the obtained meshes for the f2_desk dataset, in which the original weighting strat-

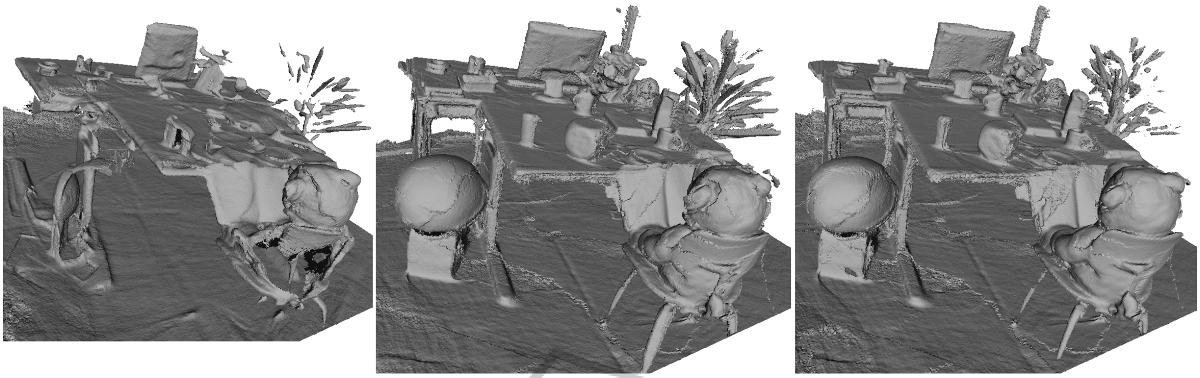

Figure 8: From left to right: snapshots of the final 3D synthetic meshes for the f2_desk dataset obtained by the original KinFu, DA, and DASS methods, respectively. (Note the differences between the objects on the table and the ball object at the left.)

egy of the KinFu outperforms the other methods in terms of ATE. The minor 2.99% and 5.39% increase of the ATE is compensated by a much larger reduction of the RPE with 11.65% and 10.40% for the DA and DASS methods, respectively. As a consequence, the 3D model accuracy is elevated by the DA and DASS methods as visualized.

5 CONCLUSIONS

We have proposed intelligent distance-aware weighting strategies for the Truncated Signed Distance Function (TSDF) voxel-based model to enhance 3D reconstruction model quality. The increased model quality leads to an improvement of the pose-estimation algorithm by providing more accurate data. In conventional TSDF, every newly sensed depth value is directly integrated into the 3D model, so that, when using low-cost depth sensors, less accurate depth data can overwrite more accurate data. For distance-aware weighting, we have considered weight definition and model updating to be essential aspects. These aspects are combined into our new proposed weighting strategies, Distance-Aware (DA) and Distance-Aware Slow-Saturation (DASS) methods, to intelligently integrate the depth data into the synthetic 3D model, according to the distance-sensitivity metric of the sensor. Both the DA and DASS methods prevent the already-fused data to be overwritten by less accurate data.

We have compared the resulting pose trajectories of the DA, DASS, and the original KinFu methods to the corresponding ground-truth trajectory in terms of Absolute Pose Error (ATE) and Relative Pose Error (RPE). Based on the quantitative results on 17 datasets, we have found that on the average, the DA and DASS methods compared to the original KinFu, reduce the pose estimation error in terms of the ATE by 16.37% and 43.40%, respectively. In terms of the RPE, the achieved improvements for the DA and DASS methods are 26.06% and 48.29%, respectively. In extreme cases the improvements in ATE and RPE can grow even up to 93.12% and 92.26% for DA and 90.16% and 88.81% for DASS, respectively.

An interesting observation is that the DA and DASS methods reduce the RPE more efficiently than the ATE, which is explained by the ATE being an appropriate metric to express the performance of visual SLAM systems, whereas the RPE measures the drift of a visual odometry system. Since there is no loop-closure detection in the original KinFu, the ATE suffers from this, whereas locally improved quality and accuracy of the 3D model, leads to a better performance of the ICP as a pose estimator. Hence, the drift error in the poses reduced so that the RTE reduction is more significant than the ATE. Using all datasets on the average, the DA and DASS methods improve the pose-estimation process of the original KinFu by 21.22% and 45.84%, respectively.

We have shown visually that in some cases, which the original KinFu destroys the final synthetic 3D model or deforms it, the DA and DASS methods are sufficiently robust to preserve model reconstruction. We have found that there is a clear mutual dependence between the pose-estimation accuracy and the quality of the 3D model. This can be exploited to enhance either the pose estimation or the 3D model quality, depending on the application or interest.

In the near future, this research work can be further improved in several ways. First, a more efficient model implementation in memory, such as Octree-based structures, can help to suppress the fast saturation of the accumulated weight in the DA method. Second, the research on weighting strategies can be enhanced with angle-aware and texture-aware methods, besides distance-awareness. For example, when exploring the texture-aware strategy, we may exploit the RGB information along with the depth data to elevate the 3D model quality.

ACKNOWLEDGEMENTS

This research has been performed within the PANORAMA project, co-funded by grants from Belgium, Italy, France, the Netherlands, the United Kingdom, and the ENIAC Joint Undertaking.

REFERENCES

Andersen, V., Aans, H., and Brentzen, J. A. (2010). Surfel based geometry reconstruction. In Collomosse, J. P. and Grimstead, I. J., editors, *Theory and Practice of Computer Graphics, Sheffield, United Kingdom, 2010. Proceedings*, pages 39–44.

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

Bondarev, E., Heredia, F., Favier, R., Ma, L., and de With, P. H. N. (2013). On photo-realistic 3D reconstruction of large-scale and arbitrary-shaped environments. In *CCNC*, pages 621–624.

Bylow, E., Sturm, J., Kerl, C., Kahl, F., and Cremers, D. (2013). Real-time camera tracking and 3D reconstruction using signed distance functions. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany.

Chang, J. Y., Park, H., Park, I. K., Lee, K. M., and Lee, S. U. (2011). Gpu-friendly multi-view stereo reconstruction using surfel representation and graph cuts. *Comput. Vis. Image Underst.*, 115(5):620–634.

Chow, J., Ang, K., Lichti, D., and Teskey, W. (2012). Performance analysis of a low-cost triangulation-based 3d camera: Microsoft kinect system. In *ISPRS12*, pages XXXIX–B5:175–180.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *ACM SIGGRAPH Conference Proceedings*. ACM.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. (2012). An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE Int. Conf. on*, pages 1691–1696. IEEE.

Engelhard, N., Endres, F., Hess, J., Sturm, J., and Burgard, W. (2011). Real-time 3D visual slam with a hand-held camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vasteras, Sweden.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3D modeling of indoor environments. *the 12th Int. Symposium on Experimental Robotics (ISER)*.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. Journ. Robotics Research*, 31(5):647–663.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. 24th annual ACM Symp. User interface software and technology*, UIST '11, pages 559–568, New York, NY, USA. ACM.

Khoshelham, K. (2011). Accuracy analysis of kinect depth data. *ISPRS Workshop Laser Scanning*, 38:1.

Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454.

Kubacki, D. B., Bui, H. Q., Babacan, S. D., and Do, M. N. (2012). Registration and integration of multiple depth images using signed distance function.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), IEEE Int. Conf. on*, pages 3607–3613. IEEE.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011a). Kinectfusion: Real-time dense surface mapping and tracking. In *Proceeding of 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA. IEEE Computer Society.

Newcombe, R. A., Lovegrove, S., and Davison, A. J. (2011b). Dtam: Dense tracking and mapping in real-time. In Metaxas, D. N., Quan, L., Sanfeliu, A., and Gool, L. J. V., editors, *ICCV*, pages 2320–2327. IEEE.

PCL (2011). Kinectfusion implementation in the PCL. http://svn.pointclouds.org/pcl/trunk/.

Ren, C. Y. and Reid, I. (2012). A unified energy minimization framework for model fitting in depth. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 72–82. Springer.

Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), IEEE/RSJ Int. Conf. on*, pages 573–580.

Whelan, T., Johannsson, H., Kaess, M., Leonard, J. J., and McDonald, J. (2013). Robust real-time visual odometry for dense rgb-d mapping. In *IEEE International Conference on Robotics and Automation, ICRA*.

Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia.

Zeng, M., Zhao, F., Zheng, J., and Liu, X. (2012). A memory-efficient kinectfusion using octree. In *Proceedings of the First international conference on Computational Visual Media*, CVM'12, pages 234–241, Berlin, Heidelberg. Springer-Verlag.