Vision, Modeling, and Visualization (2014)
J. Bender, A. Kuijper, T. von Landesberger, and P. Urban (Eds.)

A Generalized 2D and 3D Multi-Sensor Data Integration Approach based on Signed Distance Functions for Multi-Modal Robotic Mapping

S. May[1], Ph. Koch[1], R. Koch[1], Ch. Merkl[1], Ch. Pfitzner[1] and A. Nüchter[2]

[1]Nuremberg Institute of Technology Georg Simon Ohm, Germany
[2]Julius-Maximilians-University Würzburg, Germany

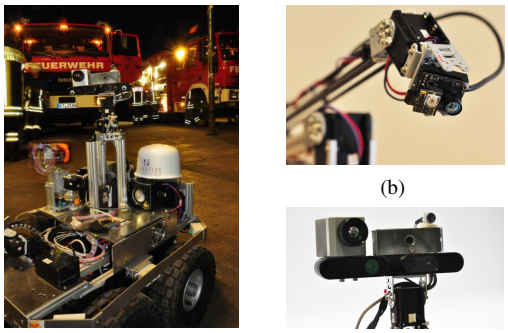
Abstract

This paper describes a data integration approach for arbitrary 2D/3D depth sensing units exploiting assets of the signed distance function. The underlying framework generalizes the KinectFusion approach with an object-oriented model respecting different sensor modalities. For instance, measurements of 2D/3D laser range finders and RGB-D cameras can be integrated into the same representation. Exemplary, an environment is reconstructed with a 3D laser range finder, while adding fine details from objects of interest by closer inspection with an RGB-D sensor. A typical application of this approach is the exploration in rescue environments, where large-scale mapping is performed on the basis of long-range laser range finders while hollows are inspected with lightweight sensors attached to a manipulator arm.

1. Introduction

Novel approaches in precise object reconstruction recently triggered several smart applications. For instance, nowadays one can register a 3D model of a real object with commodity hardware at home. Attaching a specific depth sensor and taking snapshots at different perspective views, a precise volumetric model can be reconstructed on the fly. The model can be sent to a 3D printer in order to get a replication of the real, cf. [SBKC13].

The availability of fast modeling methods are also valuable for exploration tasks. In rescue operations a rescue team needs as much information as possible from collapsed environments to assess the situation. Up to now, human rescue forces need to inspect every hollow for vital signs or hazardous substances. In doing so, human sagacity is the basis for the efficiency of search. The mission commander quickly obtains the whole picture by piecing the team's reports together. Exploring these environments with autonomous robots reduces dangerous situations to which a mission commander has to expose the team. But replacing humans with robots entails either the demand for an on-board implementation of human-like skills or a precise 3D perception concept transmitted to the control center. The later necessitates the integration of depth sensors.



(a) (b) (c)

Figure 1: *(a) Rescue robot with 3D laser scanner and Asus Xtion PRO at a fire service drill with the voluntary fire department Dettelbach, Germany. (b) Robot manipulator with CamBoard nano. (c) Sensor head with Asus Xtion PRO.*

Due to different working principles, each depth sensor has assets and drawbacks, for instance concerning the sampling rate, resolution, disposability of color or operating range. Figure 1a depicts the integration of a rescue robot equipped with 3D sensors in a fire service drill in order to assess human-robot collaboration in time-critical situations.

In this paper we show how different depth sensors are fused by generalizing the KinectFusion approach [IKH*11]. A framework that respects different sensor modalities is developed. Integrating new sensors merely requires the implementation of a concrete interface. An environmental model is reconstructed with multiple 3D sensors as depicted in Figures 1a - 1c. While covering a large environment with a 3D laser range finder, fine details from objects of interest are added by closer inspection with RGB-D or Time-of-Flight (ToF) cameras. The fusion of these sensors has two main assets: First, a rough coverage is achieved with the laser scanner that makes localization more robust in poorly structured environments as a result of 360°-field-of-view (FoV) sensing. Second, the small and lightweight sensors can be moved with a manipulator in areas that are not visible to the laser scanner, for instance in hollows with narrow entries. Environmental reconstruction is achieved on the fly while employing only a power-saving CPU.

2. Related Work

Surface reconstruction is fundamental for many robotic tasks, e.g., object detection, manipulation and environment modeling. In general, the surface of an object is to be reconstructed by merging measurements of sensors from different perspective views. Depth measurements are needed as well as the sensor's pose. If both are known, a registration procedure is dispensable and data can directly be merged.

When both, pose and depth, are unknown, e.g., when using a hand-held monocular camera, structure from motion is applicable [Wu13]. Corresponding features in consecutive images are assigned to estimate sensor motion, e.g., based on the Scale Invariant Feature Transform (SIFT) [Low04]. Using monocular cameras, 3D models can be reconstructed up to absolute scale [SSS07].

If depth information but no pose is given, i.e., by using hand-held RGB-D cameras or laser scanners, 3D modeling is possible by simultaneously localize the sensor while creating a 3D map. Many approaches use either feature-based methods or iterative schemes like the Normal Distribution Transform (NDT) [BS03, Mag09] or the Iterative Closest Points (ICP) algorithm and variants of it [BM92, CM91, RL01, Zha92].

The interest in 3D registration with hand-held RGB-D cameras increased with the appearance of the Microsoft Kinect device on the market. For the registration of textured 3D measurements, Henry et al. utilized a joint optimization over both, the RGB and the depth channel [HKH*10]. They combined the ICP algorithm applied to depth data with SIFT feature localization in the RGB domain.

Izadi et al. applied the representation of a signed distance function (SDF) [OF02] to data streams from the Microsoft Kinect camera [IKH*11]. The group achieved real-time capability by the use of massive parallelism on GPU. The hand-held Kinect was localized by ICP registration while minimizing errors of the depth image channel through data integration. High-density 3D models of objects within a defined volume can be reconstructed setting the Kinect in motion. A high frame rate allows for the efficient search of corresponding point pairs. A projective data association – comparable to reverse calibration [BDL95]– can be used due to the small displacements between exposure poses. KinectFusion was one of the most prominent publications since 2011 in the domain of 3D reconstruction and it builds the basis for a wide variety of applications. An open-source implementation is available in the point cloud library (PCL) under the name KinFu [poi13].

Bylow et al. investigated the localization problem and found non-linear optimization on the basis of Lie algebra to be more efficient and robust [BSK*13]. Either point-to-point and point-to-plane metrics were evaluated as well as different parameter weighting and truncation strategies. The authors demonstrated the applicability of their approach with a quadrocopter carrying an RGB-D sensor. Processing was performed GPU-accelerated in real-time on a ground station.

Sturm et al. used the KinectFusion approach to reconstruct 3D models of persons from multiple views. Sending such a model to a 3D printer, one can receive a copy of the own body [SBKC13].

Zeng et al. implemented an octree-based GPU-version of KinectFusion to make the mapping of larger environments possible due to less memory consumption [ZZZL12]. As a result, scenes may be 8 times larger.

Recently, Chen et al. scaled the original KinectFusion approach using a compact volumetric representation [CBI13]. They losslessly streamed data bidirectional between the GPU and host allowing for unbounded reconstructions.

Furthermore, Whelan et al. extended the KinectFusion approach to work on large scale environments [WKLM13]. The close-range reconstruction is done classically with KinectFusion. Areas that leave the predefined volume around a sensor in motion are subsequently extracted and meshed.

During exploration in larger environments, it is important to avoid sensor heading such that RGB-D measurements are ambiguous w.r.t. the registration results, e.g., when measuring only co-planar surfaces while heading the sensor towards a plain wall. Also in the absence of objects within the measurement range, registration is impossible. That is why RGB-D SLAM is exploiting both, depth and RGB images [HKH*10]. But also a plain wall might miss enough texture to lead this approach to success.

The problem of co-planarity in depth images emerge less likely, the wider the field of view is, e.g., for a panoramic sampling scheme of rotating 3D laser range finders. Other important parameters are the sensor's working range and the

ascertainability of different materials. A combination of different sensors is advantageous in 3D exploration tasks. Concerning this aspect, we contribute to the state of the art with the generalization of SDF-based registration approaches for application to different types of range sensing units. Data integration becomes straightforward by virtue of the same representation. In contrast to above mentioned publications, our framework is not restricted on GPU-powered machines for achieving real-time applicability. It can be employed on power-saving CPUs and is even applicable on embedded platforms.

3. Data Integration Framework

The framework's basis, the signed distance function, represents the distance of a given point to a surface. The space from which a map is to be reconstructed, is divided in fine elements, i.e., cells in the case of 2D mapping and voxels for a 3D representation. Let $\vec{v}$ be the center of an arbitrary element, $\vec{p}$ the sensor's position and m the distance measurement determined in the direction of the given element, the signed distance function reads:

$$d(\vec{v}) = m - ||\vec{p} - \vec{v}|| \tag{1}$$

The penetration depths ρ and ε respect sensor noise and define the finest granularity to be distinguished. Negative values of the SDF correspond to elements that are not visible to the sensor due to occlusions. Therefore, signed distances are truncated or respectively multiplied with a weight – resulting in a truncated signed distance function (TSDF). Weights are calculated with an exponential model according to Bylow et al. with the following function [BSK*13]:

$$f(d,\varepsilon,\rho) = \begin{cases} 1 & \text{if } d \geq -\varepsilon \\ e^{-\sigma(d-\varepsilon)^2} & \text{if } d < -\varepsilon \text{ and } d > -\rho \\ 0 & \text{if } d < -\rho \end{cases} \tag{2}$$

The representation of data as TSDF can be considered as superior for the Kinect over other representations w.r.t. measurement noise and multiple-view data integration. In addition, surface normals can be extracted conveniently. For more details, the interested reader is pointed to related publications of KinectFusion [IKH*11, BSK*13].

Any sensor needs a specific model respecting the sampling scheme and measurement precision. Measurement samples belong to certain lines of sight. Assignments of TSD elements to measurements and vice versa are the basis of the data integration approach. This formulates two different problem statements:

First, it is necessary to extract synthetic measurements from the TSD space for any sensor model. This addresses the question of how measurements would look like for a specific sensor at an arbitrary pose. Those synthetic point clouds can be used for drift-free sensor localization as it is done with KinectFusion.

Second, one needs to know how new measurements are to be integrated into the TSD space. For this, each element is projected back with the specific sensor model to determine which measurement ray comes closest to it.

Algorithm 1 sketches the registration and integration process of new measurement data.

Algorithm 1 Registration and integration of new measurement data.

```
1:  procedure ONSENSORDATAREVEIVE(sensor)
2:      model ← RAYCAST(sensor)
3:      scene ← get data from sensor
4:      T_icp ← icp registration of model and scene
5:      T_sensor ← T_icp T_sensor                ▷ update sensor pose
6:      if (T_sensor T_last_push^-1 > thresh) then
7:          T_last_push ← T_sensor
8:          PUSH(sensor)
9:      end if
10: end procedure
```

The model is obtained by intersecting the TSD space with measurement rays of the sensor model by calling method RAYCAST, cf. Algorithm 2. The result represents a synthetic point cloud comprising information from all previously pushed range measurements. The scene, i.e., the most recent sensor data, is aligned with the model by ICP registration. The pose change represented as transformation matrix $\mathbf{T}_{icp}$ is applied to update the current sensor pose $\mathbf{T}_{sensor}$. Subsequently, the scene is integrated into the TSD space by calling method PUSH, if a significant movement has been performed since the last integration, cf. Algorithm 3.

3.1. Sensor Modeling

The generalization of TSD integration needs the definition of a sensor interface. The specific modalities of every sensor has to be transparent for the processing chain in order to reside generic. All sensors have in common

- a pose represented as transformation matrix,
- the disposability of point-wise distance measurements along certain lines of sight,
- and probably information about the reflection of emitted or ambient light, i.e., grayscale or color measurements.

This generic interface is designed as abstract *Sensor* class, cf. Figure 2. From this class the specific model is inherited. Two methods need to be implemented:

- One method for providing the measurement rays respecting the current pose of sensor (*getRayMap*),
- and one method for the back projection to the measurement matrix, i.e., the assignment of an arbitrary element in the TSD space to a measurement ray (*backProject*).

In the following two different sensor models are discussed.

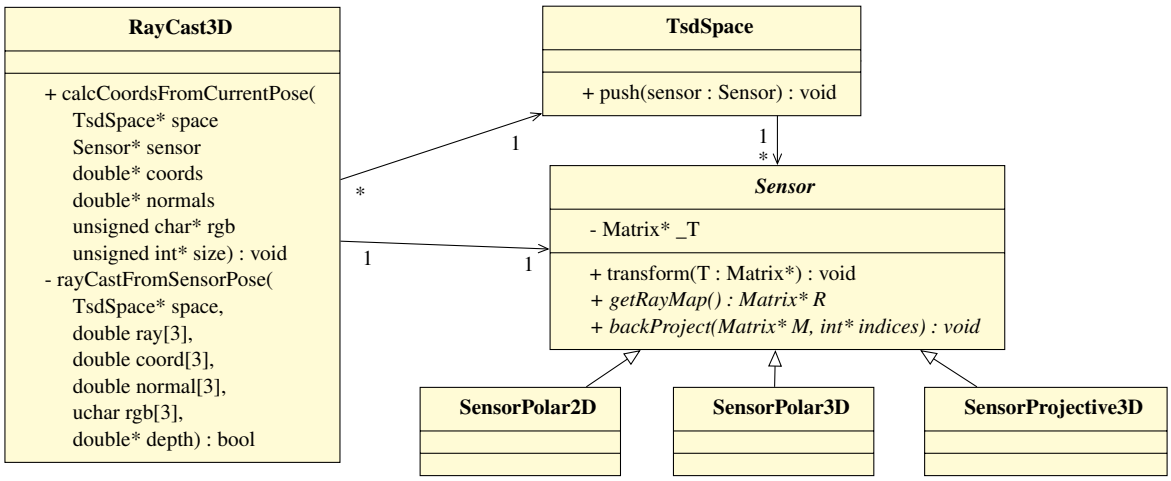


Figure 2: *Specification of a generic interface for TSD data integration and back projection in UML notation.*

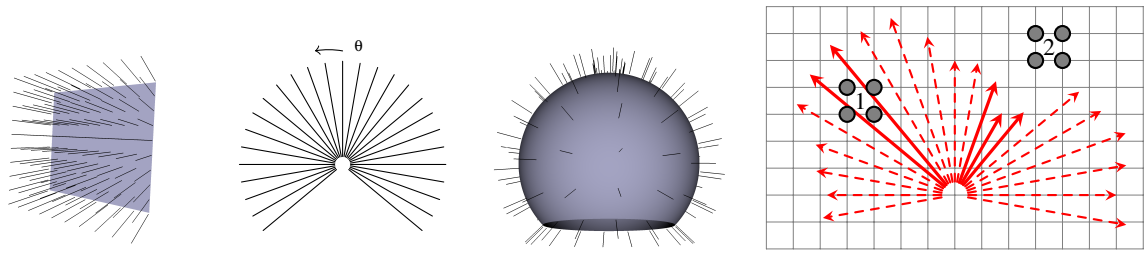


Figure 3: *(a) Raycasting model for RGB-D and ToF sensors. (b) Raycasting model for 2D laser range finder. (c) Raycasting model for 3D laser range finder. (d) Acceleration scheme: Elements remaining empty or being not in the area of sight can be skipped during push execution. Partition edges are used to determine the set of measurement values to which the partition's elements would be assigned (solid lines).*

Pinhole Camera Model

The basis for ToF and RGB-D sensors is the pinhole camera model, i.e., the projection of homogenous coordinates $\vec{\xi}$ to an image plane using the sensor's 3×4 projection matrix as follows.

$$\mathbf{P} = \begin{pmatrix} f_u & 0 & t_u & 0 \\ 0 & f_v & t_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3}$$

$$\mathbf{P}\vec{\xi} = (su, sv, s)^{\mathrm{T}} \rightarrow (u, v)^{\mathrm{T}}, \tag{4}$$

where f_u and f_v represent scaling parameters and t_u and t_v the coordinates of the principal point. The parameter s respects the fact that all points along a ray of sight are projected to the same pixel coordinates $(u, v)^{\mathrm{T}}$. This ambiguity is resolved with the measured distance.

The ray casting module employing equation (3) and (4) determines a pixel-dependent line of sight by inversion. For this trivial case it reads:

$$x = \frac{1}{f_u} \cdot u - t_u \qquad y = \frac{1}{f_v} \cdot v - t_v \qquad z = 1 \tag{5}$$

These definitions allow for assignment of arbitrary coordinates to the measurement matrix and vice versa. Figure 3a depicts this specific sensor model.

Polar Model

The model for a 3D laser scanner uses conversion between polar and Cartesian coordinates. The line of sight in the 2D scanning plane of a Hokuyo UTM-30LX device $(x'z')$ is determined by

$$x' = \sin\theta \qquad z' = \cos\theta, \tag{6}$$

where θ is the angle of the rotating mirror deflecting the laser beam, cf. Figure 3b. The rotation of the 2D scanner around the scanner's center axis through the angle ϕ yields the 3D components of the resulting line of sight.

$$x = \cos\phi \cdot x' \qquad y = \sin\phi \cdot x' \qquad z = z' \tag{7}$$

The back projection converts an arbitrary point $\vec{p} = (x\,y\,z)$ in polar representation as follows.

$$\phi = \arctan\frac{y}{x} - \pi,\ \phi \in [-\pi;\pi] \tag{8}$$

$$\theta = \arccos\frac{z}{||\vec{p}||} \tag{9}$$

These definitions provide the mutual assignment of arbitrary coordinates and laser beams. Figure 3c depicts the sensor model for the 3D laser range finder. A model for 2D localization is also covered by setting $\phi = 0$.

3.2. Raycasting

Providing a specific model for each sensor allows for a universal ray casting module. The ray caster queries the set of rays from the instantiated sensor module and walks along them through the TSD space. If a sign change is determined, coordinates and normals are calculated by trilinear interpolation, cf. [IKH*11]. Algorithm 2 sketches the implementation of the universal ray casting module. Coordinates and normals are returned, if a sign change of the TSDF between neighboring elements are detected.

Algorithm 2 Raycasting through TsdSpace.

```
1:  procedure RAYCAST(sensor)
2:      R ← get rays from sensor
3:      for each ray r ∈ R do
4:          V ← first element in TsdSpace along r
5:          tsd_prev ← NAN
6:          while V is inside TsdSpace do
7:              tsd ← tsd_V                    ▷ assign property of V
8:              if tsd ≤ 0 ∧ tsd_prev > 0 then
9:                  c ← extract coordinates f. TsdSpace
10:                 n ← extract normals f. TsdSpace
11:                 break
12:             end if
13:             tsd_prev ← tsd
14:             V ← next element in TsdSpace along ray
15:         end while
16:     end for
17: end procedure
```

3.3. Data Integration

Back projection is necessary in order to assign an element from the TSD space to the corresponding measurement ray. That means, the element could only have been seen from the sensor pose along this ray. If the measurement value is significantly smaller than the elements's distance, objects are located in between. The designation whether an element is close to a surface considers the penetration depths ρ and ε. In contrast, the measurement value might be significantly larger. This means that the element is empty – there is no object near by.

Algorithm 3 TSD-Integration of generic sensors.

```
1:  procedure PUSH(sensor)
2:      p⃗ ← get current position from sensor
3:      data ← get data from sensor
4:      mask ← get mask from sensor
5:      for each element V in TsdSpace do
6:          v⃗ ← obtain coordinates of V
7:          idx ← project v⃗ back to measurement index
8:          if (mask[idx]) then
9:              distance ← ||p⃗ - v⃗||
10:             d ← data[idx] − distance
11:             if d ≥ −ρ then
12:                 tsd ← min(d/ρ, 1.0)
13:                 w ← f(d, ρ, ε)
14:                 tsd_V ← (tsd_V·w_V + tsd·w)/(w_V + w)
15:                 w_V ← w_V + w
16:             end if
17:         end if
18:     end for
19: end procedure
```

Algorithm 3 outlines the integration of new measurements into the TSD space. A measurement mask is used to indicate their validity, i.e., to exclude invalid values due to low reflectivity or exceedance of measurement range. Due to the employment of a generic interface, the concrete modalities of the sensing unit is transparent to the algorithm.

3.4. Acceleration scheme

The employment of a TSD representation demands a huge amount of memory and computing power. Currently available implementations rest upon massive parallelism on GPUs. Most of the space is wasted since allocated elements stay empty, i.e., there is no measurement assigned to it. For that, elements are grouped into partitions, such that a partition has a high probability of containing empty or unseen elements. Empty elements are those cells/voxels, which were in the area of sight but too far from a sampled surface. Unseen elements are hidden due to occlusions. We choose cell partitions of $16 \times 16\,\mathrm{cm}^2$ and voxel partitions of $16 \times 16 \times 16\,\mathrm{cm}^3$ in office-like environments. The edges of each partition can be used to verify quickly, if any element inside the partition needs to be considered during the push execution. Figure 3d depicts the approach. Partition 1 is crossed by two laser beams. Measurements to the related surface are larger than the truncation radius. All cells remain empty. There is merely need for fixing $tsd = 1$ and increasing the weight w_V for the entire partition. Partition 2 is not in the area of sight. Thus, all elements inside keep being untouched and do not need to be instantiated.

Raycasting benefits from the outlined acceleration scheme too. If a ray crosses an empty or unseen partition, element testing, i.e., bilinear interpolation for calculating the TSDF, is not needed.

3.5. Movement detection for 3D laser range finder

Mapping and localization with a 3D laser range finder needs to concern the recording time of a single scan. The custom 3D laser range finder, that is used for the experiments in this paper, is based on a Hokuyo UTM-30LX device. For reasonable resolutions the framerate is limited to 1 Hz. While taking a sensor frame, the pose of the laser range finder has to be fixed, otherwise the scan appears skewed. If motion reconstruction is not possible, i.e., if no inertial measurement unit is available or the computing power is too low to deskew the data take by means of relaxation algorithms, the 3D scan is unsuitable for registration.

For the reconstruction of larger environments with a 3D scanner, it is sufficient to register data at greater intervals. The typical movement scheme of a rescue robot is step by step to areas that need closer inspection. This scheme is also typical for teleoperation. An operator needs to stop the robot from time to time to orient himself and to plan the next movement. When the robot stops, the next 3D scan is registered and pushed into the TSD representation.

In order to decouple the reconstruction approach from external signals, two subsequent 3D scans are evaluated for absence of motion. The similarity of the obtained transformation matrix $\mathbf{T}_{4\times4}$ and identity $\mathbf{I}$ is checked by the eigenvalues $\vec{\lambda}$. Ideally the transformation between two scans from the same pose is the identity matrix $\mathbf{I}$ with all eigenvalues $\lambda_i = 1$. If any of the eigenvalues λ_i differs to the ideal value 1 by a given threshold λ_{th}, the scanner is rated to be in motion between the last two scans. If no movement was detected, the last received point cloud is used for data integration into the TSD volume. With this approach the 3D scanner can even be guided by hand while the environment is reconstructed on the fly. Algorithm 4 shows data integration on the basis of a 3D laser range finder with movement detection.

Algorithm 4 Data integration of 3D laser scans.

```
1:  procedure DATAINTEGRATION(scan)
2:      scene ← scan
3:      T_icp ← ICP(scene_prev, scene)
4:      λ⃗ ← calculate eigenvalues for T_icp
5:      if all eigenvalues ||λ_i − 1|| < λ_th then
6:          T_icp ← ICP(model, scene)
7:          T_pose = T_icp T_pose
8:          model ← scan
9:      end if
10:     scene_prev ← scan
11: end procedure
```

4. Experiments and Results

The data integration framework has been applied to two different mapping tasks. By the use of a 2D laser range finder, a large indoor environment can be reconstructed. In an extended experiment data from a 3D laser range finder, an RGB-D camera and a ToF camera are integrated into the framework.

4.1. 2D Mapping

The data integration framework is applied to a SICK LMS100 laser range finder in a larger indoor environment, cf. Figure 6a. The map is defined for an area of $128 \times 128\,\mathrm{m}^2$ with a granularity of 1.5 cm. The approach is applied on the fly during mission. A framerate between 7 and 10 Hz could be achieved on a power-saving Core i7 CPU (45W TDP). Figure 4 depicts the timing of the separate operations in the execution chain of the TSD approach. This seconds the result of Holz et al. that ICP-based mapping algorithms can perform similarly well as Rao-Blackwellized Particle Filter implementations [HB10].

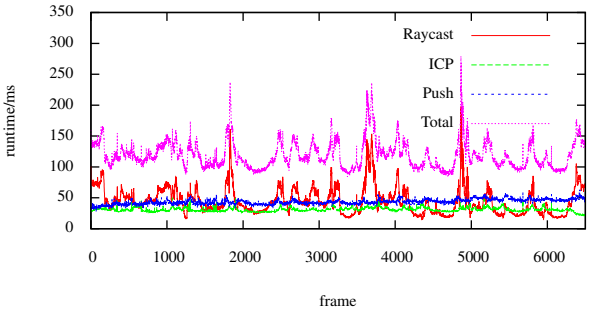


Figure 4: *Timing results for the 2D mapping with the SICK LMS100 in an area of* $120 \times 120\,\mathrm{m}^2$ *with* 1.5 cm *resolution.*

4.2. 3D Mapping

For 3D mapping three different sensors are tested: a custom 3D laser range finder based on a Hokuyo UTM-30LX scanner, a CamBoard nano Time-of-Flight (ToF) camera and an Asus Xtion PRO device. The sensor model for the laser scanner considers a resolution of 0.25° in the 2D scanning plane, i.e., around the mirror axis. These scanning planes are rotated with 10 rpm around the scanner's center axis resulting in a horizontal resolution of 3°, cf. Figure 7a. A half rotation provides a 360°-FoV sampling as depicted in Figure 3c.

The coarse horizontal resolution is a tradeoff to keep the scanning time low. When performing a typical, natural stop-and-go exploration, undistorted scans are taken, otherwise they need to be motion compensated. Those undistorted 3D scans are registered and pushed to the TSD space with 2 cm granularity. Figure 7b depicts a resulting 3D environment map after pushing a few 3D scans. In spite of the coarse resolution of the scanner, the mapping approach achieves a dense representation. The time for registration and integrating data into the TSD space is negligible compared to the scanning time.

Areas of interest are inspected separately. The RGB-D camera is used to fill a TSD space of fine resolution (5 mm).

The higher density of measurement points and the disposability of coloring is clearly advantageous.

The fusion of both TSD spaces are straightforward due to the same representation. Both are to be aligned by registration. Figure 7c depicts the reconstruction of the area of interest labeled in Figure 7b. On the Core i7 the approach achieves a frame rate of 2 Hz for 640×480 resolution and 8 Hz for 320×240 resolution in an environment of $2.5 \times 2.5 \times 2.5\,\mathrm{m}^3$ with 1 cm granularity.

For the ToF camera a high frame rate is achieved. Hand-operated mapping of a $1.28 \times 1.28 \times 1.28\,\mathrm{m}^3$ TSD volume of 5 mm granularity is applicable. Figure 5 shows the runtime of the separate operations in the execution chain. In average a frequency of 10 Hz is achieved. The reconstruction of the area of interest is shown in Figure 7d;

The framework has been evaluated against the ground-truth benchmark of Sturm et al. [SEE*12]. The accuracy lies in a comparable range to the GPU implementation of KinFu.

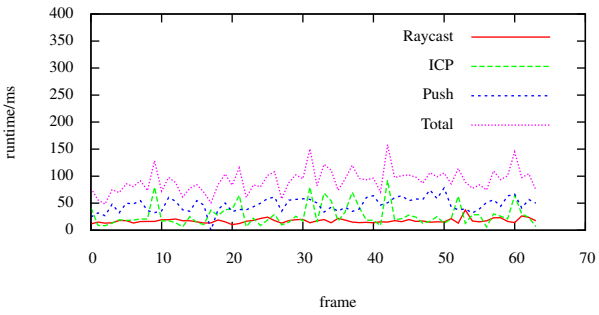


Figure 5: *Timing results for an hand-held CamBoard nano applied to an area of* $1.28 \times 1.28 \times 1.28\,\mathrm{m}^3$ *with* 5 mm *granularity.*

5. Conclusions and Future Works

This paper presented a data integration approach of different depth sensors exploiting assets of the signed distance function. It contributes to the state of the art concerning three main aspects.

First, it generalizes the KinectFusion approach to combine different depth sensing units. Only a specific sensor interface is needed to be implemented in order to integrate a certain sensor. Second, the registration step on the basis of the ICP algorithm is exchangeable. The framework can be integrated in any 2D/3D mapping approach. Finally, the generic representation makes 2D/3D sensor data integration straightforward. The approach can deal with coarse resolutions, when one needs to be aware of processing time or frame rate respectively. Environmental reconstruction is achieved on the fly while employing only a power-saving CPU.

The software framework is made available as open-source at `http://github.com/autonohm/obviously`. Future work will focus on full-automatic 3D mapping independent of sensor configuration. Optimization of the registration step offers a high benefit.

5.1. Acknowledgement

This research has been funded by STAEDTLER Stiftung (foundation) within the project *Robot Assistance for Exploration in Rescue Missions (02/13 - 04/14)*. The foundation's support is gratefully acknowledged. Furthermore, we thank the voluntary fire department of Dettelbach for supporting this research.

References

[BDL95] BLAIS G., D. LEVINE M.: Registering multiview range data to create 3d computer objects. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8 (1995), 820–824. 2

[BM92] BESL P., MCKAY N.: A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (February 1992), 239 – 256. 2

[BS03] BIBER P., STRASSER W.: The normal distributions transform: a new approach to laser scan matching. In *Proceedings of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2003), pp. 2743–2748. 2

[BSK*13] BYLOW E., STURM J., KERL C., KAHL F., CREMERS D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems Conference (RSS)* (June 2013). 2, 3

[CBI13] CHEN J., BAUTEMBACH D., , IZADI S.: Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph. 32*, 4 (July 2013), 113:1–113:6. 2

[CM91] CHEN Y., MEDIONI G.: Object modeling by registration of multiple range images. In *In Proceedings of the IEEE Conference on Robotics and Automation (ICRA)* (Sacramento, CA, USA, 1991), pp. 2724–2729. 2

[HB10] HOLZ D., BEHNKE S.: Sancta simplicitas – on the efficiency and achievable results of slam using icp-based incremental registration. In *In Proceedings of the IEEE Conference on Robotics and Automation (ICRA)* (2010), pp. 1380–1387. 6

[HKH*10] HENRY P., KRAININ M., HERBST E., REN X., FOX D.: Rgbd mapping: Using depth cameras for dense 3D modeling of indoor environments. In *RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS* (2010). 2

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (2011). 2, 3, 5

[Low04] LOWE D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. 2

[Mag09] MAGNUSSON M.: *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Örebro University, 2009. Örebro Studies in Technology 36. 2

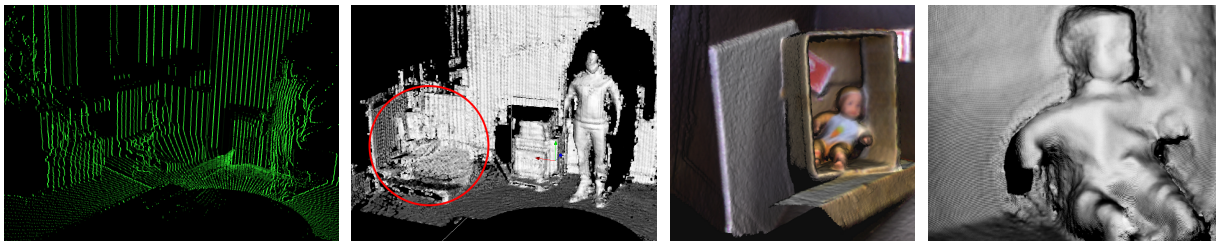
[OF02] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces (Applied Mathematical Sciences)*, 2003 ed. Springer, Nov. 2002. 2

[poi13] Point cloud library (PCL). http://pointclouds.org, 2013. Accessed on 13/10/2013. 2

(a) (b)

(c)

Figure 6: *Results with 2D mapping approach. (a) Bird's-eye view on a University building at Kesslerplatz, Nuremberg (Source: Google Earth). (b) Ground view along the corridor in the University's building. (c) Resulting 2D map with TSD approach.*



(a) (b) (c) (d)

Figure 7: *Application of TSD approach to different sensors. (a) Raw 3D laser scan taken at* 10 rpm. *(b) 3D reconstruction with laser data. The red label highlights an area of interest. (c) 3D reconstruction of inspection area from an RGB-D camera. (d) 3D reconstruction obtained with the CamBoard nano.*

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modellling (3DIM)* (Quebec City, Canada, 2001). 2

[SBKC13] STURM J., BYLOW E., KAHL F., CREMERS D.: CopyMe3D: Scanning and printing persons in 3D. In *German Conference on Pattern Recognition (GCPR)* (Saarbrücken, Germany, September 2013). 1, 2

[SEE*12] STURM J., ENGELHARD N., ENDRES F., BURGARD W., CREMERS D.: A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)* (Oct. 2012). 7

[SSS07] SNAVELY N., SEITZ S. M., SZELISKI R.: Modeling the world from internet photo collections. *International Journal of Computer Vision* (2007). 2

[WKLM13] WHELAN T., KAESS M., LEONARD J., MCDONALD J.: Deformation-based loop closure for large scale dense RGB-D SLAM. In *Proceedings of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2013). 2

[Wu13] WU C.: Towards linear-time incremental structure from motion. In *Proceedings of the International Conference on 3D Vision (3DV)* (2013). 2

[Zha92] ZHANG Z.: *Iterative Point Matching for Registration of Free-Form Curves*. Tech. Rep. RR-1658, INRIA Sophia Antipolis, Valbonne Cedex, France, 1992. 2

[ZZZL12] ZENG M., ZHAO F., ZHENG J., LIU X.: A memory-efficient kinectfusion using octree. In *Proceedings of the First International Conference on Computational Visual Media (CVM)* (2012), pp. 234–241. 2