Robust Real-Time 3D Modeling of Static Scenes Using Solely a Time-of-Flight Sensor

Johannes Feulner, Jochen Penne, Eva Kollorz, Joachim Hornegger
Chair of Pattern Recognition, Friedrich-Alexander-University Erlangen-Nuremberg
Email: Jochen.Penne@informatik.uni-erlangen.de

***Abstract*—An algorithm is proposed for the 3D modeling of static scenes solely based on the range and intensity data acquired by a Time-of-Flight camera during an arbitrary movement. No additional scene acquisition devices, like inertia sensor, positioning robots or intensity based cameras are incorporated. The current pose is estimated by maximizing the uncentered correlation coefficient between edges detected in the current and a preceding frame at a minimum frame rate of four fps and an average accuracy of 45 mm. The paper also describes several extensions for robust registration like multiresolution hierarchies and projection Iterative Closest Point algorithm. The basic registration algorithm and its extensions were intensively evaluated against ground truth data to validate the accuracy, robustness and real-time-capability.**

***Index Terms*—3D modeling, 3D reconstruction, ToF camera, Time-of-Flight.**

I. Introduction

MANY computer vision applications including tracking and recognition have already benefited from the robustness and speed of range and depth sensors as opposed to using regular image intensity cameras. Range imaging using ToF systems has been used in radar and Lidar applications for more than thirty years. Time-of-Flight (ToF) imaging provides a direct way for acquiring 3D surface information of objects and scenes in the current field-of-view [1]. More recently, ToF sensors are used in a wider range of applications like obstacle detection [2], gesture recognition [3][4] and automotive passenger classification [5].

For more complex application areas like map building, robot navigation or scene exploration, building a 3D model of a scene larger than the field-of-view of the applied ToF sensor imposes a 3D/3D-registration problem: Partially non-overlapping 3D surface points have to be transformed into a common coordinate system by estimating the extrinsic parameters of the range measuring sensor at each acquisition time step.

Available ToF sensors [6][7][8] provide the data at rates higher than 10 Hz. Thus, for real-time analysis it is required to perform the registration at a comparative speed and provide an on-the-fly 3D modeling of the scene observed by the ToF sensor.

ToF camera systems actively illuminate the scene with an incoherent light signal, which is modulated by a cosine-shaped signal of frequency f. Its light is usually in the non-visible part of the spectrum near the infrared spectral range. The light signal is assumed to travel with the constant speed of light in the surrounding medium $c \approx 3.00 \cdot 10^8 \frac{m}{s}$ (in vacuum) and is reflected by surfaces in the scene. By estimating the phase-shift ϕ between the emitted and the reflected light signal the distance d can be computed as follows:

$$d = \frac{c}{4\pi f} \cdot \phi. \tag{1}$$

Due to the periodicity of the cosine-shaped modulation signal, this equation is valid only if the distance to be estimated is smaller than $\frac{c}{2f}$. This upper limit for the observable distances is termed the non-ambiguity range and is approx. 7.5 m for available ToF camera systems. By modeling the ToF sensor as a pin-hole camera one can calculate the set of N 3D surface coordinates $\mathbf{P} = \{\mathbf{p}_i | \mathbf{p}_i = (x_i, y_i, z_i)^T \in \mathbb{R}^3,\ 0 \leq i \leq N-1,\ i \in \mathbb{N}_0\}$ of the field-of-view by using principles of similar triangles [9]. The necessary intrinsic ToF camera parameters can be determined by appropriate calibration routines [10][11]. Additionally, an intensity value $a_i \in \mathbb{R}_+$ for each point $\mathbf{p}_i$ is provided. It represents the amount of light reflected and is thus roughly encoding the reliability of the measured distance as more reflected light leads to a more accurate estimation of the phase-shift. 3D surface coordinates and intensity information are registered by construction. The 3D surface coordinates are given in an Euclidean coordinate system whose origin coincides with the optical center of the ToF camera. The camera coordinate system is a left handed system with the z-axis aligned with the optical axis and pointing from the scene towards the camera. The y-axis points upwards. We term the set $\mathbf{F}^j = \{(\mathbf{p}_i^j, a_i^j) | 0 \leq i \leq N-1,\ j \geq 0,\ i,j \in \mathbb{N}_0\}$ a *frame* of the ToF camera and use upper indices to distinguish multiple frames if necessary.

II. State of the Art

There is only a small number of publications on 3D modeling of static scenes using **only** ToF cameras. Most authors augment the ToF camera with either regular cameras and/or inertia sensors (Huhle *et al.* [12]), high-resolution spherical cameras (Prusak *et al.* [13]), accelerometers (Ohno *et al.* [14]) or extrinsic camera parameters provided for example by a robot arm (Fuchs *et al.* [15]). By combining the laterally low-resolution range data of the ToF camera with the high-resolution color information from regular cameras or the very accurate pose information from a robot arm, the transformation of the acquired 3D information into a common coordinate system can be done more easily as compared to computing it from ToF data alone. The authors are only aware of a single publication which purely relies on ToF camera data: Swadzba *et al.* [16] determine the relative transformation between two frames by minimizing the mean square error between corresponding 3D points using ICP (Iterative Closest Point) variants [17].

Ohno *et al.* [14] perform 3D map building using a robot mounted ToF camera enhanced by an accelerometer. They apply an improved version of the ICP algorithm and use data provided by the accelerometer as additional information for estimating an initial transformation. The authors report problems in estimating the rotation components of the camera motion.

Furthermore, none of the existing approaches can achieve real-time operations: Huhle *et al.* [12] explicitly state that their approach is not real-time-capable and report computational times of about two seconds for their complete processing chain. By Prusak *et al.* [13] frame rates of seven fps are reported but the proposed approach relies strongly on the color information of a spherical camera to perform the pose estimation and registration of the acquired data. Additionally, the authors used a ToF camera which has a small lateral resolution (64×48 pixel) even in the context of ToF cameras (which provide lateral resolutions of more than 160×120 pixel). As a result, the authors have a small amount of 3D data to process and do not report computational times for available ToF cameras with a significantly higher lateral resolution. Swadzba *et al.* [16] report computational times of nine to eleven seconds for the registration of two range data sets acquired with a PMD[vision] 19k (160×120 pixels) without using any additional information like robot pose or color information. By Ohno *et al.* [14] computational times delivering a frame rate of approx. four fps are reported.

Swadzba *et al.* [16] report a mean registration error of more than 29 mm when using just the data available from a PMD[vision] 19k. The biggest benefit of using additional sensors is improved accuracy. In comparison, Fuchs *et al.* [15] report a mean precision of 3 mm when using a ToF camera of comparable lateral resolution and additionally incorporating robot pose information. This is among the best reported accuracy results for ToF based registrations. Huhle *et al.* [12] describe no quantified registration error. Prusak *et al.* [13] present an initial mean error of less than 20 cm for their approach using a ToF camera combined with a spherical camera. Ohno *et al.* [14] evaluated their approach considering estimated camera motion versus real camera motion and report relative errors of 15% for translation and 17% for rotation.

A summary of the accuracy and performance of these techniques is given in Table I. Note that the authors partly used different criteria to evaluate the accuracy of their algorithms. For example, Ohno *et al.* [14] focused on the correct estimation of the camera/robot pose. Fuchs *et al.* [15] investigated how well the known 3D geometry of a cube was reconstructed. Swadzba *et al.* [16] and Prusak *et al.* [13] used arbitrarily chosen scenes with unknown 3D geometry to investigate the accuracy of their approaches by evaluating metric distances between registered 3D data sets. Huhle *et al.* [12] provide only a qualitative evaluation of reconstructed 3D scenes.

TABLE I
PUBLISHED APPROACHES FOR 3D MODELING USING TOF CAMERAS.

ToF camera[1] and author	resolution[2] (pixels)	other modalities	comput. times (s)	accuracy (mm)
PMD19k [16]	160×120	none	≥9	≥27
PMD1k-S [16]	64×16	none	≥9	≥9
PMD3k-S [13]	64×48	spherical camera	≥0.14	≤200[3]
PMD19k [12]	160×120	1600×1200px. regular camera; inertia sensor	≥2	–
SR-3000 [15]	176×144	indust. robot KUKA KR 16	–	≈3mm; ≈3°[4]
O3D100 [15]	64×50	indust. robot KUKA KR 16	–	≈3mm; ≈3°
SR-2 [14]	160×124	accelerometer	≈ 0.25	≈15%/17%[5]

[1] Abbreviations: PMD*x* for PMD[vision] *x*, SR-3000 for SwissRanger SR-3000, O3D100 for IFM O3D100 for , SR-2 for SwissRanger SR-2

[2] The lateral resolution of the ToF camera is given.

[3] This error value was reported as the initial mean error between 3D data computed using Structure-from-Motion and the 3D data available from the used ToF camera. The authors do not report any other accuracy measures.

[4] The given values refer to the error in reconstructing a 3D geometry of a cube with known dimensions from multiple acquired frames.

[5] The given values refer to the relative error in estimating the translation/rotation of the robot the data acquisition devices were mounted on.

III. BASIC REGISTRATION

Assuming a static scene, a set of M frames $\{\mathbf{F}^j | 0 \leq j \leq M-1,\, j \in \mathbb{N}_0\}$ can be transformed into a common coordinate system, when the relative transformation $(\mathbf{R}^j, \mathbf{t}^j)$ between two consecutive frames $\mathbf{F}^j$ and $\mathbf{F}^{j-1}$ (for $j \geq 1$) is known.

The proposed registration approach estimates the relative transformation by projecting edge feature vertices of frame $\mathbf{F}^{j-1}$, whose camera pose is known, into frame $\mathbf{F}^j$ with respect to the camera pose of $\mathbf{F}^j$, which has to be estimated. The original feature points of $\mathbf{F}^j$ and the projected ones are compared using the cosine of the images that are represented as vectors, which is also called uncentered correlation coefficient:

$$\mathrm{c}(\mathbf{x}, \mathbf{y}) = \begin{cases} 0 & \text{if } \|\mathbf{x}\|\|\mathbf{y}\| = 0 \\ \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} & \text{else,} \end{cases} \tag{2}$$

with $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N$. If one or both vectors have zero length, the cosine is here defined to be zero. The cosine of the angle is used as a measure for the goodness of fitting features detected in both frames to each other. The camera pose of $\mathbf{F}^j$ is found iteratively using nonlinear optimization. Fig. 1 depicts the basic registration algorithm, which is explained in detail in the following.

First, edges are extracted from the intensity data as well as their *z*-coordinates $\mathbf{Z} = \{z_i | \mathbf{p}_i = (x_i, y_i, z_i) \in \mathbf{P}\}$ of the 3D data (given in the local camera coordinate system) of each frame using the structure tensor [18]. Note that if in a certain pixel an edge has been detected, two kinds of information are available: on one hand the 2D pixel coordinates and on the other hand the 3D coordinates of the observed point in the scene. For the proposed approach it is only of interest, if a pixel contains an edge or not. The set containing the 3D

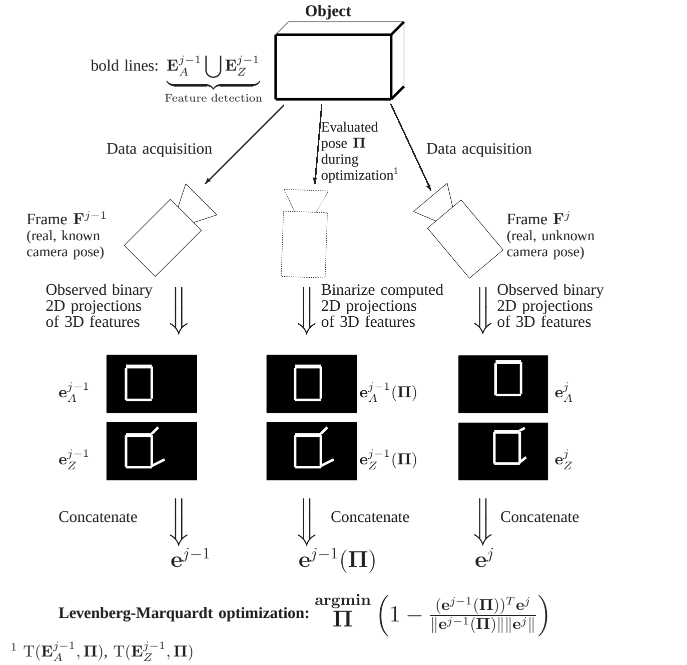

Object
bold lines: $\mathbf{E}_A^{j-1} \bigcup \mathbf{E}_Z^{j-1}$
Feature detection
Data acquisition
Evaluated pose $\mathbf{\Pi}$ during optimization[1]
Data acquisition
Frame $\mathbf{F}^{j-1}$ (real, known camera pose)
Frame $\mathbf{F}^{j}$ (real, unknown camera pose)
Observed binary 2D projections of 3D features
Binarize computed 2D projections of 3D features
Observed binary 2D projections of 3D features
$\mathbf{e}_A^{j-1}$
$\mathbf{e}_A^{j-1}(\mathbf{\Pi})$
$\mathbf{e}_A^{j}$
$\mathbf{e}_Z^{j-1}$
$\mathbf{e}_Z^{j-1}(\mathbf{\Pi})$
$\mathbf{e}_Z^{j}$
Concatenate
Concatenate
Concatenate
$\mathbf{e}^{j-1}$
$\mathbf{e}^{j-1}(\mathbf{\Pi})$
$\mathbf{e}^{j}$
Levenberg-Marquardt optimization: $\underset{\mathbf{\Pi}}{\operatorname{argmin}} \left(1 - \frac{(\mathbf{e}^{j-1}(\mathbf{\Pi}))^T \mathbf{e}^j}{\|\mathbf{e}^{j-1}(\mathbf{\Pi})\| \|\mathbf{e}^j\|}\right)$
[1] $\mathrm{T}(\mathbf{E}_A^{j-1}, \mathbf{\Pi})$, $\mathrm{T}(\mathbf{E}_Z^{j-1}, \mathbf{\Pi})$

Fig. 1. Scheme of the basic registration: Having observed 3D ($\mathbf{E}_A^{j-1}$, $\mathbf{E}_Z^{j-1}$) and corresponding binary 2D ($\mathbf{e}_A^{j-1}$, $\mathbf{e}_Z^{j-1}$) features in frame $\mathbf{F}^{j-1}$ as well as binary 2D ($\mathbf{e}_A^{j}$, $\mathbf{e}_Z^{j}$) in frame $\mathbf{F}^j$, the objective function involves the cosine of the angle between binary projected edges $\mathbf{e}^{j-1}(\mathbf{\Pi})$ of frame $\mathbf{F}^{j-1}$ and the binary edges detected in frame $\mathbf{F}^j$. $\mathbf{e}^{j-1(\mathbf{\Pi})}$ is computed by projecting $\mathbf{E}_A^{j-1}$ and $\mathbf{E}_Z^{j-1}$ to an image plane whose pose is parameterized by $\mathbf{\Pi}$.

coordinates of points corresponding to edges in the intensity data is denoted by

$$\mathbf{E}_A = \{\mathbf{p}_i | a_i \,\mathrm{is\,edge}\}, \tag{3}$$

and the set containing the 3D coordinates of points corresponding to edges in the set of *z*-coordinates is denoted by

$$\mathbf{E}_Z = \{\mathbf{p}_i = (x_i, y_i, z_i) | z_i \,\mathrm{is\,edge}\}. \tag{4}$$

The sets of observed corresponding 2D pixel coordinates of edges are denoted $\mathbf{e}_A \in \{0,1\}^N$ and $\mathbf{e}_Z \in \{0,1\}^N$. This data is binary and thus only contains information about the presence of an edge in a pixel or not:

$$\mathbf{e}_A(i) = \begin{cases} 1 & \text{if } \mathbf{p}_i \in \mathbf{E}_A \\ 0 & \text{else} \end{cases}, \; \mathbf{e}_Z(i) = \begin{cases} 1 & \text{if } \mathbf{p}_i \in \mathbf{E}_Z \\ 0 & \text{else} \end{cases}. \tag{5}$$

In (5), $\mathbf{e}_A(i)$ or respectively $\mathbf{e}_Z(i)$ denote the i-th entry of the vector. Fig. 2 exemplarily visualizes these data. Thus, for a specific frame $\mathbf{F}^j$ a vector $\mathbf{e}^j \in \{0,1\}^{2N}$ can be derived by

$$\mathbf{e}^j = (\mathbf{e}_A^j | \mathbf{e}_Z^j), \tag{6}$$

where we use upper indices to distinguish multiple frames and $|$ denotes the concatenation of vectors. Considering two consecutive frames $\mathbf{F}^j$ and $\mathbf{F}^{j-1}$ the relative translation and rotation of the camera is parameterized by $\mathbf{\Pi} = (t_x, t_y, t_z, \tau_x, \tau_y, \tau_z) \in \mathbb{R}^6$ and $\mathrm{T}(\mathbf{X}, \mathbf{\Pi})$ with $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^3\}$ denotes the perspective projection of a set of 3D points to a 2D image plane: The extrinsic parameters are given by $\mathbf{\Pi}$, the intrinsic parameters are not explicitly denoted for convenience and assumed to be constant. Using this, a vector $\mathbf{e}^{j-1}(\mathbf{\Pi}) \in \{0,1\}^{2N}$ can be derived with

$$\mathbf{e}^{j-1}(\mathbf{\Pi}) = (\underbrace{\mathrm{T}(\mathbf{E}_A^{j-1}, \mathbf{\Pi})}_{\mathbf{e}_A^{j-1}(\mathbf{\Pi})} | \underbrace{\mathrm{T}(\mathbf{E}_Z^{j-1}, \mathbf{\Pi})}_{\mathbf{e}_Z^{j-1}(\mathbf{\Pi})}), \tag{7}$$

which contains binary information about the presence of an edge in a pixel if the image plane observing the 3D edge data $\mathbf{E}_A^{j-1}$ and $\mathbf{E}_Z^{j-1}$ of frame $\mathbf{F}^{j-1}$ is moved virtually.
The registration of two frames $\mathbf{F}^j$ and $\mathbf{F}^{j-1}$ ($j \geq 1$, $j \in \mathbb{N}$) is done by minimizing

$$1 - \mathrm{c}(\mathbf{e}^{j-1}(\mathbf{\Pi}), \mathbf{e}^j) \tag{8}$$

with respect to $\mathbf{\Pi}$ using the Levenberg-Marquardt algorithm [19].

We use binary edge presence information in (5) for optimization rather than a continuous edge strength information, because in our experiments the binary information lead to a better convergence. The reason is that using continuous edge strength values derived from the structure tensor proved to lead to an alignment of the strongest edges while neglecting weaker ones.

In (8) the usage of the cosine of the angle between the two vectors $\mathbf{e}^{j-1}(\mathbf{\Pi})$ (projected edges with respect to extrinsic camera parameters) and $\mathbf{e}^j$ (currently observed edges) is motivated by the consideration of the following two cases:

1) $\mathbf{e}^{j-1}(\mathbf{\Pi})$ contains only zero-entries, i.e. from the camera pose $\mathbf{\Pi}$ no edges are visible.
2) $\mathbf{e}^{j-1}(\mathbf{\Pi})$ contains non-zero-entries at positions where $\mathbf{e}^j$ contains zero-entries, i.e. the current pose $\mathbf{\Pi}$ represents a bad estimation of the real camera pose and the edges of the previous frame are visible but completely non-overlapping with the edges in the current frame.

When applying the sum of squared distances (SSD) or sum of absolute distances (SAD) as distance measure, 1) will yield a smaller distance value than 2). When computing the cosine of the angle between the two image vectors, it will in both cases yield zero, and only increase for an increasing overlap of non-zero-entries. Thus, when using the objective function proposed in (8) the constellation described in 1) does not yield an additional local minimum, in contrast to the case when the SSD or SAD is utilized. The occurrence of the case described in 1) was frequently observed during our experiments and the proposed objective function proved to enable a robust convergence of the optimization.
The last three components of $\mathbf{\Pi} = (t_x, t_y, t_z, \tau_x, \tau_y, \tau_z)$ are the Euler-angle parameterization of the rotation component of the relative pose $\mathbf{\Pi}$ between frames $\mathbf{F}^{j-1}$ and $\mathbf{F}^j$. Because rotations do not form a vector space, they cannot be optimized using normal gradient decent or LM optimization. Note that this is a fundamental problem and does not depend on the parameterization of the rotation. However, small rotations locally do approximately form a vector space. The reason is that a small rotation has almost the same effect as a translation. For example, if the earth was a perfect sphere, moving on its surface 20 km to the north would actually be a rotation by some tiny angle about the center, but the

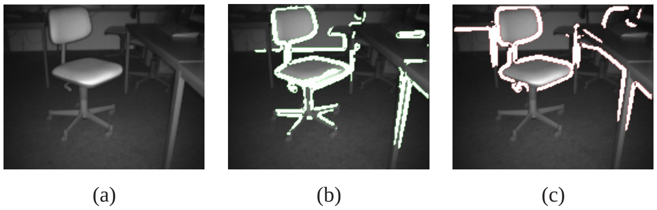


Fig. 2. Feature detection: Original intensity data (2(a)), overlayed with detected features in the amplitude data (2(b); features depicted white) and overlayed with features detected in the z-coordinates (2(c); features depicted white). White pixels in Fig. 2(b) correspond to entries of $\mathbf{e}_A$ equal to 1. White pixels in Fig. 2(c) correspond to entries of $\mathbf{e}_Z$ equal to 1 (see (5)).

effect is very similar to a translation. Here the optimization is performed in a coordinate system coinciding with the local camera coordinate system, which is determined by the camera pose at frame $\mathbf{F}^{j-1}$. Thus, the rotation components of $\mathbf{\Pi}$ can be assumed to be small numbers and can be treated as a vector space, which permits the use of normal gradient decent. This imposes the requirement that the camera was not rotated far between two consecutively acquired frames. This implies that the registration shall not take too much time. The reader is referred to section V to see how this requirement was fulfilled by our experiments.

The LM optimization estimates the gradient of the objective function given in (8) using forward differences. When optimizing a multivariate function that has a highly different sensitivity on different parameters, it is crucial to properly scale the parameters before computing the gradient. Otherwise, the components of the gradient corresponding to the parameters with low influence will usually be near to zero. This problem occurs in our case, because the translation parameters are measured in millimeter and the rotation parameters in radian. A translation by one mm causes only a small change, but a rotation by one radian makes a huge difference. Furthermore, the influence of the translation parameters also depends on the distance of the camera from the scene, whereas the influence of the rotation parameters remains constant. To cope with this, the parameters are scaled once for an optimization run so that a unit step in one parameter results in a shift of the border pixels of the camera plane by approximately 0.5 pixel. The gradient is computed by making a unit step for each parameter. Further details on scale problems when optimizing translation and rotation using gradient decent can be found in [20].

To cope with systematic errors, which may be caused by inaccurate intrinsic camera parameters leading to a drift in the registration, a frame $\mathbf{F}^j$ is only integrated into the 3D model if the camera moved significantly since the last frame $\mathbf{F}^{j-k}$, with $0 \leq k \leq j \text{ and } j, k \in \mathbb{N}_0$, that was integrated into the model. Otherwise, the model is not updated, but the registration procedure of the following frame $\mathbf{F}^{j+1}$ will use $\mathbf{\Pi}^j$ as initial parameters. A new frame is always registered with the last frame that was integrated into the scene.

IV. Extensions of the basic registration algorithm

Different extensions of the basic registration approach based on (8) were investigated to prevent the non-linear optimization from getting stuck in a local optimum. We use upper round brackets (like $\mathbf{\Pi}^{(icp)}$ or $\mathbf{\Pi}^{(0)}$) to distinguish parameterization obtained by different extensions, while keeping the convention of using upper indices with no round brackets (like $\mathbf{\Pi}^j$) to address parameterizations corresponding to a certain frame.

A. Camera Motion Prediction

Given the poses of the past three frames $\mathbf{\Pi}^{j-1}$, $\mathbf{\Pi}^{j-2}$ and $\mathbf{\Pi}^{j-3}$, the current pose, which is to be estimated, is predicted using different assumptions on the camera motion:

- assuming constant position, i.e. the camera did not move since the last frame: the predicted pose is denoted $\mathbf{\Pi}^{(0)}$ and is equal to $\mathbf{\Pi}^{j-1}$
- assuming constant velocity, i.e. the velocity observed between poses $\mathbf{\Pi}^{j-1}$ and $\mathbf{\Pi}^{j-2}$ is assumed to be constant: the predicted pose is denoted $\mathbf{\Pi}^{(1)}$
- assuming constant acceleration, i.e. the acceleration observed between poses $\mathbf{\Pi}^{j-1}$ and $\mathbf{\Pi}^{j-2}$ is assumed to be constant: the predicted pose is denoted $\mathbf{\Pi}^{(2)}$

Note that doing all computations in the camera coordinate system of frame $\mathbf{F}^{j-1}$ satisfies the assumption of only small rotation angles and thus the six-dimensional pose parameterization can be assumed to form a vector space, which is utilized for estimating the velocity and the acceleration of the rotation of the camera.

B. Projection ICP algorithm

The predicted pose $\mathbf{\Pi}^{(2)}$ is iteratively refined using an ICP variant, called projection ICP [21]. Only the pose which was predicted assuming constant acceleration is refined as it is the most dynamic prediction and most sensitive to erroneously estimated acceleration. The such estimated pose is denoted $\mathbf{\Pi}^{(icp)}$.

C. Multiresolution Hierarchy

Multiresolution approaches aim at smoothing the objective function by starting the minimization at a rather coarse level and then switch to finer resolutions. In the proposed algorithm the multiresolution is achieved by convolving the binary edge data $\mathbf{e}^j$ (considered as a binary 2D image) with an isotropic 2D gauss kernel of standard deviation σ. An initial value of $\sigma = 2.0$ pixel was chosen for the first level of the hierarchy. Heuristically, for each level of the hierarchy the used standard deviation is doubled.

D. Random Search

In order to further improve robustness of registration, S randomly chosen parameterizations $\mathbf{s}^{(i)} \in \mathbb{R}^6$ close to an estimated parameterization $\hat{\mathbf{\Pi}}$ are considered:

$$\mathbf{s}^{(i)} = \hat{\mathbf{\Pi}} + \mathbf{n}^{(i)},\ i = 1, ..., S,\ i \in \mathbb{N}_0 \quad (9)$$

where $\mathbf{n}^{(i)} \in \mathbf{R}^6$ is drawn from a zero-mean multivariate Gaussian distribution with statistically independent components. For the standard deviations of the rotation and translation parameters, values of 0.04 radian and 50 mm were chosen. Applying the addition of vectors in (9) for rotation and translation components is justified by performing all calculations in the coordinate system of the current frame and thus reasonably assuming rotation components to be small and consequently forming a vector space.

E. Extended Registration Algorithm

The above mentioned approaches constitute the extension of the basic registration algorithm described in section III.

1) Predict poses $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$ and $\mathbf{\Pi}^{(2)}$ according to IV-A.
2) Starting with $\mathbf{\Pi}^{(2)}$ iteratively estimate a pose $\mathbf{\Pi}^{(icp)}$ according to IV-B.
3) For each pose $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$, $\mathbf{\Pi}^{(2)}$ and $\mathbf{\Pi}^{(icp)}$ compute random poses according to IV-D. For each pose $S = 10$ random samples are computed. The poses are denoted $\mathbf{\Pi}^{(0,n_i)}$, $\mathbf{\Pi}^{(1,n_i)}$, $\mathbf{\Pi}^{(2,n_i)}$ and $\mathbf{\Pi}^{(icp,n_i)}$ with $1 \leq i \leq S$, $i \in \mathbb{N}$.
4) Evaluate the objective function given in (8) for $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$, $\mathbf{\Pi}^{(2)}$, $\mathbf{\Pi}^{(icp)}$ and each $\mathbf{\Pi}^{(0,n_i)}$, $\mathbf{\Pi}^{(1,n_i)}$, $\mathbf{\Pi}^{(2,n_i)}$, $\mathbf{\Pi}^{(icp,n_i)}$ with $1 \leq i \leq S$, $i \in \mathbb{N}$. The pose yielding the smallest objective function value is denoted $\overline{\mathbf{\Pi}}$.
5) $\overline{\mathbf{\Pi}}$ is used as an initial solution for a multiresolution LM-optimization according to IV-C. Three resolution levels are used. The final computed parameterization is denoted $\mathbf{\Pi}$.

V. EXPERIMENTS

To evaluate the proposed algorithm several indoor office scenes containing chairs, desks, computers etc. were investigated. The camera used was a CSEM SR-3100 [7] with a lateral resolution of 144×176 pixels. Camera motion was induced by either a pan-tilt-unit or a board-ruler-instrument allowing precise movement (rotation and translation). Table II shows a summary of the acquired scenes. A short description of the camera movement and the number of frames acquired is given. Each scene was observed with a non-moving camera. From ten consecutive frames the standard deviation of the z-components of each point was computed. The average of these values is a rough indicator of the amount of noise by which the computed 3D coordinates are corrupted. Values in the range of 17-72 mm were observed for the scenes investigated in the experiments.

A. Reconstruction Accuracy

To evaluate the accuracy of the reconstruction, the point set $\mathbf{P} = \{\mathbf{p}_i | 0 \leq i \leq N-1, i \in \mathbb{N}_0\}$ of a frame $\mathbf{F}$ was transformed with a ground-truth rigid transformation $(\mathbf{R}^r, \mathbf{t}^r)$ determined by the parameters $\mathbf{\Pi}^r$ given by the pan-tilt-unit or the board-ruler-instrument and an estimated transformation $(\mathbf{R}^e, \mathbf{t}^e)$ determined by the computed parameters $\mathbf{\Pi}^e$, resulting in the ground truth transformed point set $\mathbf{P}^r$ and the estimated point set $\mathbf{P}^e$

$$\begin{aligned} \mathbf{P}^r &= \{\mathbf{p}_i^r = \mathbf{R}^r \mathbf{p}_i + \mathbf{t}^r\} && (10) \\ \mathbf{P}^e &= \{\mathbf{p}_i^e = \mathbf{R}^e \mathbf{p}_i + \mathbf{t}^e\}. && (11) \end{aligned}$$

TABLE II
THE VIDEO SEQUENCES USED FOR EVALUATION

Video Name[1]	Description	Frames[2]	Noise (mm)[3]
tiltOnly	short tilt motion	11	17
panOnly	pan motion, direction of view is orthogonal to rotation axis	36	15
longPanSlow	long slow pan motion with a downward tilted camera	50	43
longPanFast	same as longPanSlow but faster	26	43
longPanFastShaky	same as longPanFast but camera is moved shaky	26	25
tiltPan	short tilt followed by a long pan motion	53	25
longPan	long pan motion by approx. 180° with a downward tilted camera	67	72
zigzag	alternating pan and tilt motions	45	40
transSlow	slow translation motion	28	54
transFast	fast translation motion	14	54
backward	translation backward	10	40
panTransOpp	pan motion combined with translation into the opposite direction	9	42
panTransSame	pan motion combined with translation into the same direction	17	22

[1] This name will be used furthermore as identifier for the acquired data. The second column gives detailed information about the camera motion which was performed.

[2] To validate the feasibility of the proposed algorithms data sets containing different numbers of frames were acquired and used for evaluation.

[3] The value given is the average standard deviation of the z-coordinate of 3D points in 10 consecutive frames acquired from a static scene with a non-moving camera.

The mean Euclidean distance

$$e = \frac{1}{|\mathbf{P}|} \sum_i \|\mathbf{p}_i^e - \mathbf{p}_i^r\| \tag{12}$$

between theses transformed points serves as a measure for the quality of the estimated transformation.

Two different rigid transformations were used to measure

- the absolute accuracy of the reconstruction and
- the accuracy of the registration of two frames.

The first transformation maps the point set $\mathbf{P}$ from the camera coordinate system into the global world coordinate system, which coincides with the camera pose at frame $\mathbf{F}^0$. The corresponding error is called accumulated registration error e_{acc} since it depends on the quality of the camera pose determination of earlier frames. The second transformation maps $\mathbf{P}$ into the camera coordinate system of the frame the point set was registered with. The resulting error e_{rel} is called relative registration error. In contrast to e_{acc}, it does not accumulate over time.

The error measures e_{acc} and e_{rel} were averaged over a sequence $\mathbf{F}^j, j = 0, 1, \ldots N-1$, of ToF frames to obtain the mean accumulated registration error and the mean relative registration error:

$$\overline{e}_{acc} = \frac{1}{N-1} \sum_{j=1}^{N-1} e_{acc}^j; \;\; \overline{e}_{rel} = \frac{1}{N-1} \sum_{j=1}^{N-1} e_{rel}^j \tag{13}$$

TABLE III

MEAN ACCUMULATED REGISTRATION ERROR $\overline{e}_{acc}$ IN MILLIMETER FOR ALL VIDEOS OF TABLE II AND DIFFERENT RECONSTRUCTION METHODS (FOR DETAILS SEE SECTION V-A).

Video name	Basic	Extended1	Extended2	Extended3
tiltOnly	260	108	**43**	46
panOnly	1022	260	**205**	498
longPanSlow	641	472	**422**	483
longPanFast	1749	571	431	**233**
longPanFastShaky	1765	673	433	**412**
tiltPan	1003	614	**203**	306
longPan	1082	597	**416**	511
zigzag	511	**248**	316	420
transSlow	556	**377**	820	637
transFast	711	647	**198**	219
backward	190	224	80	**48**
panTransOpp	241	222	**215**	288
panTransSame	695	**134**	135	266

TABLE IV

MEAN RELATIVE REGISTRATION ERROR $\overline{e}_{rel}$ AND ITS STANDARD DEVIATION IN MILLIMETER.

Video name	Basic	Extended1	Extended2	Extended3
tiltOnly	103 ± 55	73 ± 69	**17** ± 3	19 ± 4
panOnly	333 ± 345	31 ± 13	**25** ± 12	25 ± 11
longPanSlow	145 ± 141	35 ± 14	33 ± 11	**31** ± 11
longPanFast	353 ± 217	91 ± 67	39 ± 28	**26** ± 9
longPanFastShaky	406 ± 256	83 ± 46	64 ± 34	**52** ± 18
tiltPan	317 ± 310	126 ± 185	25 ± 10	**24** ± 8
longPan	177 ± 138	36 ± 23	27 ± 16	**25** ± 12
zigzag	169 ± 121	43 ± 42	43 ± 32	**40** ± 18
transSlow	285 ± 205	167 ± 137	**86** ± 52	92 ± 55
transFast	711 ± 369	383 ± 200	**43** ± 18	59 ± 24
backward	111 ± 53	225 ± 132	28 ± 25	**25** ± 15
panTransOpp	141 ± 62	145 ± 64	**122** ± 80	136 ± 106
panTransSame	490 ± 318	40 ± 21	**23** ± 3	43 ± 16
Average	287.8	113.7	44.2	45.9
Median	231	78	30.5	28.5
Correlation with noise	-0.03	0.32	0.27	0.26

Table III shows the mean accumulated registration error $\overline{e}_{acc}$ for different videos and specific variants of the registration algorithm, and table IV shows the mean relative registration error together with its standard deviation. Several combinations of extensions of the basic registration algorithm were evaluated, but the tables only report the results for the three most effective combinations of extensions and the basic registration algorithm:

- **Basic** denotes the algorithm described in section III.
- **Extended1** denotes the basic algorithm extended by the multiresolution hierarchy described in section IV-C.
- **Extended2** denotes the extended algorithm (IV-E) with out the ICP extension of section IV-B.
- **Extended3** is the algorithm enhanced by all extensions (see section IV-E).

The approach proposed in [16] is the only one using solely ToF range data, and there also the mean relative registration error $\overline{e}_{rel}$ is used to evaluate the quality of the registration. The ToF camera used by [16] was a PMD[vision] 19k with 160×120 pixels, which is comparable to the SwissRanger SR-3100 used for our evaluation. [16] reports a mean relative registration error of 27±22 mm and 85±107 mm for two scenes which were evaluated. For our approach 13 scenes were evaluated.

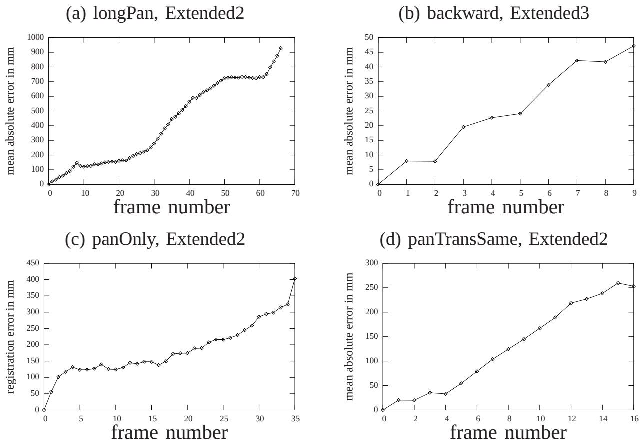


Fig. 3. Examples of error accumulation over time. For different videos and registration methods, the accumulated registration error e^{j}_{acc} is shown for each frame j (see section V-A). It is the average Euclidean distance between point pairs which were on one hand transformed with the ground truth camera parameters and on the other transformed with the estimated camera parameters into the camera coordinate system of frame $\mathbf{F}^0$. As erroneously estimated camera parameters in frame $\mathbf{F}^j$ effect the estimation of the camera parameters for frame $\mathbf{F}^{j+1}$ the error accumulates.

TABLE V

TIME FOR OPTIMIZATION/TOTAL TIME FOR REGISTRATION PER FRAME IN MILLISECONDS[1].

Video name	Basic	Extended1	Extended2	Extended3
tiltOnly	47/203	128/276	212/335	240/470
panOnly	38/103	149/235	228/315	238/389
longPanSlow	56/132	156/232	219/315	245/343
longPanFast	48/121	168/305	222/335	243/402
longPanFastShaky	44/131	52/341	245/386	270/464
tiltPan	50/106	147/223	222/302	235/355
longPan	54/115	161/246	232/311	249/363
zigzag	43/114	149/236	238/338	255/410
transSlow	40/94	117/185	226/332	238/352
transFast	39/106	108/188	206/368	237/425
backward	42/144	107/194	205/366	231/537
panTransOpp	42/147	113/235	186/304	211/378
panTransSame	36/114	145/260	194/335	221/394
Average	44.5/ 125.4	130.8/ 242.8	218.1/ 333.2	239.5/ 406.3

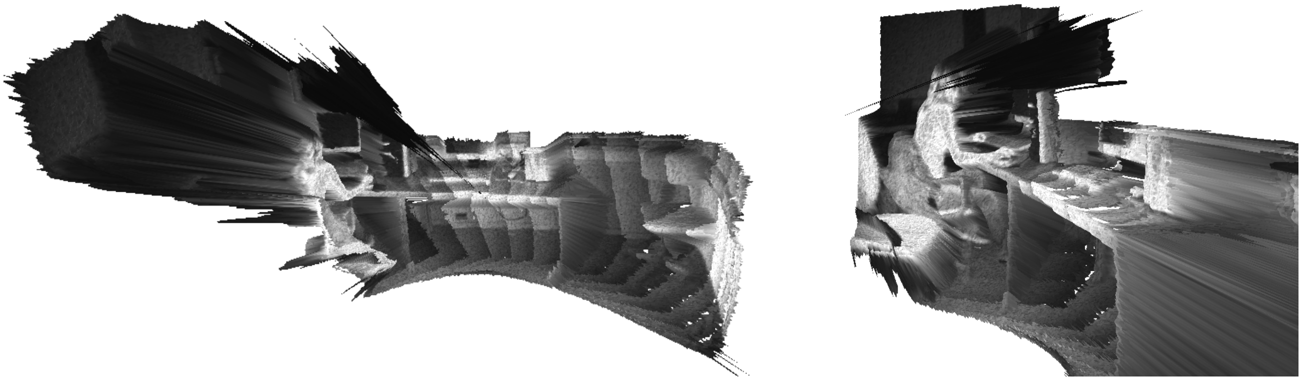
[1] For given values *x/y* *x* denotes the number of milliseconds used for performing the optimization. The time difference $y - x$ is spent for feature extraction, visualization routines or inserting triangles in the Oct-tree if the projection ICP algorithm was used (algorithm **Extended3**).

B. Performance

Performance evaluation for all the processing steps was done on a system with a Intel Pentium M 725A processor (1.6 GHz) and 1GB of RAM. Table V shows the overall time for each of the specific investigated algorithms and the time which the optimization procedure took as well.

VI. RESULTS AND DISCUSSION

The obtained results are discussed with regard to accuracy and performance, on the one hand, and on the other hand robustness of the proposed approaches. Finally, the feasibility of the registration algorithm is addressed with emphasis on the observed benefits and pitfalls.



(a)

(b)

Fig. 4. Reconstruction result for video `tiltPan`. 4(a) shows 3D model observed from viewpoint approximately to real position of the camera. 4(b) shows the same 3D model observed from a different virtual viewpoint approximately right to the desk.

A. Accuracy and Performance

The results of accuracy evaluation show that robustness could clearly be improved by the methods described in section IV. The extension that had the greatest influence on robustness by itself was the multiresolution registration. The method that worked best overall was the combination of motion prediction under the assumption of constant acceleration, random search and blurred multiresolution registration (**Extended2**).

The experiments showed that the described extensions do significantly improve the robustness which is documented by the corresponding reduced values for $\overline{e}_{acc}$ in Table III.

Table IV shows the mean relative registration error $\overline{e}_{rel}$ between two frames: It can be observed that the accuracy of the registration is competitive to a purely ICP based registration as the one used in [16]. There, a mean relative registration error between 27±22 mm and 85±107 mm is reported for the registration of **two** frames from a camera similar to the one used in our experiments. This is comparable to the overall average registration error of approx. 45 mm which was achieved with the approach proposed in this paper (see Table IV). With regard to the computed standard deviations we furthermore conclude on the reproducibility of the quality of the registration results independent from the camera motion. From the mean relative registration error and its standard deviation no preference of the proposed algorithm for certain types of camera motion can be observed. [16] report computational times of nine to eleven seconds for the registration of two frames. With our approach, registration and model construction can be achieved at approx. 334 ms on average per frame, yielding a frame rate of three fps.

B. Robustness

Table IV also displays the correlation coefficient of the mean relative registration error achieved with a certain variant of the registration algorithm and the noise present in a specific scene (see Table II). For the **Basic** variant of the registration no linear dependence of the achieved accuracy and the noise can be found. For the extended variants, the computed correlation coefficients were approx. 0.3. These correlation values are small. Thus, we conclude that our registration approach is robust to noise in the 3D point coordinates. Although the absolute correlation values are small, the increasing value of the correlation coefficient for the extended variants of the registration algorithm (**Extended1**, **Extended2**, **Extended3**) validates the following hypothesis: Due to the improved accuracy of the registration the computed mean relative registration error $\overline{e}_{rel}$ is mainly due to scene noise compared to the basic registration (**Basic**) whose errors seem to stem primarily from mis-registration.

Regarding the approaches which incorporate other modalities (see Table I) we do not reach the reported accuracies, as we do not use additional acquisition devices.

C. Feasibility of 3D Modeling

The error accumulation displayed in Table III can be assumed to increase roughly linearly. While error accumulation is of minor importance for short sequences of less than ten frames, it can reach relatively high values for long sequences with for instance 60 frames (see examples given in Fig. 3). In spite of partially large error values for each frame, the reconstructions are still informative and the overall scene structure is well retrievable. To illustrate the effect, we display 3D reconstruction results for the scene `tiltPan` which contained 53 frames: The mean relative registration error $\overline{e}_{rel}$ using algorithm variant **Extended2** was 25 mm, while the mean accumulated registration error was 199 mm with a maximum value of approx. 500 mm computed for the last frame. But Fig. 4(a) and Fig. 4(b) show that the 53 frames were registered properly. It is worth noting that the reconstruction worked even though the window pane to the right of the person led to very noisy 3D coordinates in this part of the scene. The reason is that the used features are not very sensitive to noise.

Note that the registration error also depends on the distance of the camera to the scene, which was in the range of approximately one to four meters. A mis-registration of one degree will cause a four times higher accumulated registration error for the current frame when the scene is four times farther away. There are also other factors like the typically noisy background of ToF images which limit the registration accuracy.

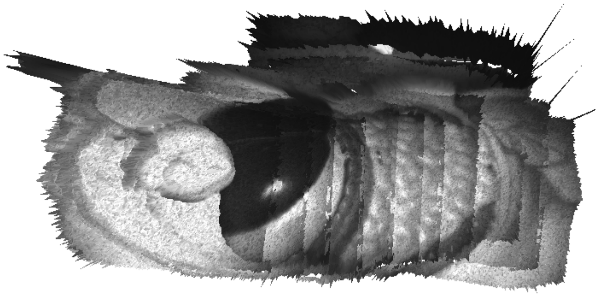


Fig. 5. Reconstruction of a medical dummy without abdominal wall. Observable organs from left to right: heart (white), liver (black) and colon (gray).

To show the feasibility of the purely ToF data driven 3D modeling of static scenes we provide a visualization of the 3D model computed from twelve frames acquired from a medical dummy whose abdominal wall was removed (Fig. 5): Note that the small anatomical structures of the colon are accurately registered and organs like the liver are correctly stitched together. The vertical stripes which can be observed in the region of the colon are not caused by mis-registration but by different observed intensity values due to changing illumination conditions when moving the camera around the medical dummy.

VII. Conclusion

An algorithm for 3D modeling of static scenes based solely on data acquired by a moving ToF camera was proposed. The rigid transformation of the camera between consecutively acquired frames is estimated by maximizing the uncentered correlation coefficient between features detected in the current and a preceding frame. Besides the basic algorithm three extended variants were examined which incorporated different combinations of a multiresolution hierarchy, camera motion prediction, random search and projection ICP algorithm.

The experimental analysis showed that the variant using multiresolution hierarchy, motion prediction and random search clearly increased robustness of the registration (variant **Extended2**). Incorporation of the projection ICP algorithm led to improvements only for certain scenes. Registration of two consecutive frames can be done at frame rates of about four fps. Thus, occlusion effects between consecutively acquired frames which may prevent finding matching features can be reasonably ignored.

We could show that in terms of the reconstruction accuracy our approach is clearly competitive to other proposed approaches. The magnitude of the mean relative registration error (see Table IV) can be explained by the distance measurement noise whichwas evaluated for each investigated scene (see Table II). The benefit of our approach becomes even more clear when the computational times are considered: To our knowledge there has been no proposed algorithm which can perform the registration of consecutive ToF frames purely data-driven in real-time or even approximately real-time.

References

[1] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck, “Smart Pixel – Photometric Mixer Device (PMD) / New System Concept of a 3D-Imaging-on-a-Chip,” in 5^{th} *International Conference on Mechatronics and Machine Vision in Practice*, 1998, pp. 259–264.

[2] T. Schamm, S. Vacek, J. Schröder, M. Zöllner, and R. Dillmann, “ Obstacle detection with a PMD-camera in autonomous vehicles ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 70 – 77.

[3] M. Holte, T. Moeslund, and P. Fihl, “ View Invariant Gesture Recognition using the CSEM SwissRanges SR-2 Camera ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 53 – 60.

[4] E. Kollorz and J. Hornegger, “ Gesture recognition with a time-of-flight camera ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 86 – 93.

[5] P. Devarakota, M. Castillo-Franco, R. Ginhoux, B. Mirbach, and B. Ottersten, “Occupant Classification Using Range Images,” *IEEE Transactions On Vehicular Technology*, vol. 56, no. 4, July 2007.

[6] PMDTec GmbH, 2007. [Online]. Available: www.pmdtec.com

[7] MESA Imaging AG, 2007. [Online]. Available: www.swissranger.ch

[8] ifm electronic gmbh, 2007. [Online]. Available: www.ifm.de

[9] A. G. Linarth, J. Penne, B. Liu, O. Jesorsky, and R. Kompe, “Fast Fusion of Range and Video Sensor Data,” in *Advanced Microsystems for Automotive Applications 2007*, J. Valldorf and W. Gessner, Eds., Berlin, 2007, pp. 119–134.

[10] Z. Zhang, “ Flexible Camera Calibration by Viewing a Plane from Unknown Orientations,” in *Proceedings of the International Conference on Computer Vision* , 1999, pp. 666 – 673.

[11] M. Linder and A. Kolb, “ Lateral and depth calibration of PMD-distance sensors,” in *International Symposium on Visual Computing (ISVC06)* , vol. 2. Springer, 2006, pp. 524 – 533.

[12] B. Huhle, P. Jenke, and W. Straßer, “ On-the-Fly Scene Acquisition with a Handy Multisensor-System ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 17 – 25.

[13] A. Prusak, O. Melnychuk, and H. Roth, “ Pose Estimation and Map Building with a PMD-Camera for Robot Navigation ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 104 – 112.

[14] K. Ohno, T. Nomura, and S. Tadokoro, “ Real-Time Robot Trajectory Estimation And 3D Map Construction using 3D Camera ,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* . IEEE, 2006.

[15] S. Fuchs and S. May, “ Calibration and Registration for Precise Surface Reconstruction with TOF Cameras ,” in *Proceedings of the Workshop Dynamic 3D Imaging in conjunction with DAGM'07* . DAGM e.V., 2007, pp. 35 – 43.

[16] A. Swadzba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe, “A Comprehensive System for 3D Modeling from Range Images Acquired from a 3D ToF Sensor,” in *Proc. of International Conference on Computer Vision Systems*. Bielefeld University, Bielefeld, Germany: University Library of Bielefeld, 2007.

[17] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[18] U. Köthe, “Edge and Junction Detection with an Improved Structure Tensor,” *Lecture Notes in Computer Science*, vol. 2781, pp. 25–32, 2003.

[19] D. W. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[20] M. Wheeler and K. Ikeuchi, “Iterative Estimation of Rotation and Translation using the Quaternion,” Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-95-215, 1995.

[21] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.