Dynamic Surface Reconstruction by Recursive Fusion of Depth and Position Measurements

GERHARD KURZ
UWE D. HANEBECK

Surface estimation can be performed based on position or depth measurements. We propose a method to fuse both types of measurements. Position measurements are obtained from landmarks on the surface, i.e., they are fixed to a certain point on the surface. In contrast, depth measurements reflect the depth measured along a line emanating from a depth camera and are not fixed to a position on the surface. The proposed approach uses a mixture of Cartesian and polar or spherical coordinate to treat both measurement types accordingly. By doing so, the uncertainties associated with the different measurement types are explicitly considered. The presented method represents the surface by a spline and is applicable to both 2D and 3D applications. Surface estimation is considered as a recursive filtering problem and standard nonlinear filtering methods such as the unscented Kalman filter can be used to obtain surface estimates. We show a thorough evaluation of the proposed approach in simulations.

This is an extended version of the paper "Recursive Fusion of Noisy Depth and Position Measurements for Surface Reconstruction" [18] published at the *16th International Conference on Information Fusion (Fusion 2013)*, which received the *Jean-Pierre Le Cadre Award for Best Paper*.

Manuscript received July 31, 2013; revised December 12, 2013; released for publication December 19, 2013.

Refereeing of this contribution was handled by Peter Willett.

Authors' addresses: Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany, E-mail: (gerhard.kurz@kit.edu, uwe.hanebeck@ieee.org).

1. INTRODUCTION

Many applications require the reconstruction of surfaces based on noisy measurements. For example, in various medical applications the surface of organs needs to be reconstructed from measurements that originate from medical imaging technologies [22], [6], [29], [7]. Surface reconstruction is also a relevant topic in other areas such as robotics [23] and computer graphics [22], [11].

A practical application where the presented approach may be used is intra-operative beating heart tracking for robot-assisted coronary artery bypass graft. The idea behind this application has first been introduced by Nakamura et al. in 2001 [24] and can be summarized as follows. The surgery is carried out by a robot that is remotely controlled by a surgeon. During the surgery, the movement of the heart is observed by sensors and this information is used to control the robot in such a way that it automatically compensates for the heart motion. The surgeon is in turn presented with a stabilized image [17] of the heart and experiences the illusion of operating on the still heart, which is a significantly easier task. Because operating on the beating heart is very difficult, currently operations are usually performed on the stopped heart, which incurs significant disadvantages for the patient's health, such as a risk of anemia and cerebral microembolization [16]. In order to make robot-assisted beating heart surgery feasible, an accurate reconstruction of the moving heart surface is required.

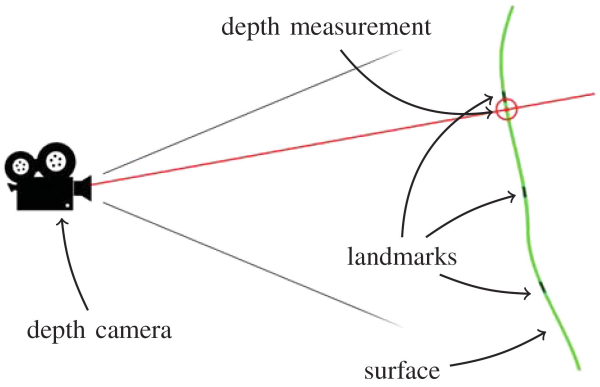

Fig. 1. The considered setting: A depth camera observes a deformable surface (light green) and obtains depth measurements (red circle) along a line emanating from the camera (red line). Additionally, some sparse landmarks (dark green) on the surface can be tracked by other means (e.g., a stereo camera system).

For reconstructing a surface, we consider two different types of measurements (see Fig. 1). First, there are position measurements originating from certain points located at a fixed positions on the surface. Position measurements are typically obtained from landmarks on the surface, for example structured regions that allow 3D reconstruction with a stereo camera system. Second, there are depth measurements that do not originate from a fixed point on the surface and only depend on the dis-

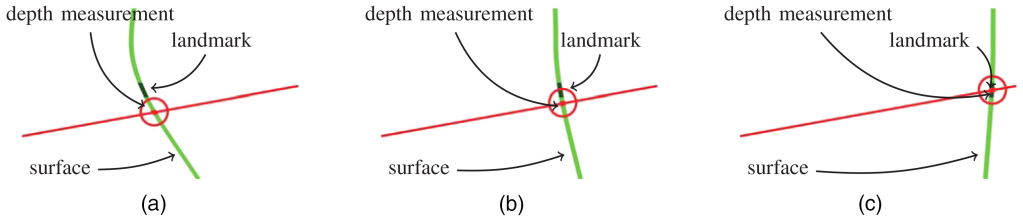

Fig. 2. Depth measurements (red circles) and position measurements of a time-varying surface. Notice how the depth measurement is obtained from different surface points as the surface deforms. (a) Beginning. (b) Middle. (c) End.

tance of a certain point in space to the surface along a given line. Depth measurements can be obtained from depth sensors such as time-of-flight (TOF) cameras or sensors based on structured light such as the Microsoft Kinect. The difference between position and depth measurements is illustrated in Fig. 2.

While position measurements are typically sparse but highly accurate, depth measurements tend to be more plentiful, but less accurate and more susceptible to noise. On the one hand, stereo camera systems may have a high resolution, but perform poorly in non-structured areas. On the other hand, TOF sensors can handle uniform surface areas, but have comparatively limited resolution and accuracy. Thus, it is beneficial to combine both types of measurements in order to achieve a more accurate and robust reconstruction of the surface compared to scenarios where only one type of measurement is used. Fusion of information from different types of sensors allows to alleviate the disadvantages of any given sensor type.

Many practical applications are not limited to a static scenario because the sensors and the surface move relatively to each other. Furthermore, the surface may deform and change shape over time. Consequently, our goal is to track surface position and shape over time and to include new information recursively as it is obtained. Prior knowledge may be included to predict the future evolution of the surface.

We now outline our main contribution. In this paper, we introduce a novel method for surface reconstruction suitable for both 2D and 3D applications. The proposed method combines depth and position measurements to recursively estimate the state of the surface while considering measurement uncertainties. It does not depend on a particular choice of sensor and can be employed in a wide area of applications. Our method is based on a spline representation of the surface whose parameters are recursively estimated using nonlinear filtering techniques. Separate measurement equations for depth and position measurements are derived in order to deal with their individual characteristics.

1.1. Structure

The paper is structured as follows. In Sec. 2, we give an overview of previous work in the area of surface reconstruction. The required prerequisites are introduced in Sec. 3. The presented method is derived for the 2D case in Sec. 4 and adapted to the 3D case in Sec. 5. We propose some further enhancements in Sec. 6 and evaluate the proposed algorithms in simulations in Sec. 7. Finally, we form a conclusion in Sec. 8.

Compared to our previous publication [18], we have significantly extended Sec. 2 to give a more complete overview of literature on the topic and Sec. 6, where we give enhancements to the proposed method that allow the incorporation of angular uncertainties and the use of approximation rather than interpolation functions. We have also performed an evaluation of the adaptive addition of nodes, which is given in Sec. 7. Furthermore, we now provide additional explanations in various places as well as a number of supplementary figures to illustrate the proposed method.

1.2. Notation

We denote vectors by underlined letters $\underline{x}$, matrices by bold letters $\mathbf{A}$, and angles by Greek letters α.

n	number of dimensions ($n = 2$ or 3)
k	time index
$\underline{p}_1, \ldots, \underline{p}_m$	points in $\mathbb{R}^n$
$f_1, \ldots, f_m$	function values in $\mathbb{R}$
$f(\cdot)$	interpolating function
$\phi(\cdot)$	radial basis function
$c_1, \ldots, c_m$	coefficients in interpolating function
interpolate$(\underline{p}_k, f_k)$	interpolation algorithm that returns interpolating function
$\underline{x}_k$	system state in $\mathbb{R}^q$
$\underline{x}_k^e, \underline{x}_k^p$	estimated and predicted state
$a_k(\cdot)$	system function
u_k	system noise with covariance $\mathbf{C}_k^u$
l	number of landmarks
$x_k^{a,b}$	position of landmark a in dimension b
$\hat{\underline{y}}_k$	position measurements
$\underline{v}_k$	position measurement noise with covariance $\mathbf{C}_k^v$
$\hat{\underline{z}}_k$	depth measurements
$\underline{w}_k$	depth measurement noise with covariance $\mathbf{C}_k^w$
$\alpha_1, \ldots, \alpha_r$	measurement angles
$s_k(\cdot)$	interpolated surface
$s_k^{\text{true}}(\cdot)$	true surface

$\phi_1, \ldots, \phi_d$	angles for additional control points in 2D
$(\phi_i, \theta_i)_{i=1,\ldots,d}$	angles for additional control points in 3D
$x_k^{1,*}, \ldots, x_k^{d,*}$	depth of additional control points
σ_{add}	initial standard deviation for additional control points
$E_k^\tau(\cdot)$	RMSE at given angle over τ time steps ending at k
$\delta_k^1, \ldots, \delta_k^r$	noise in angular domain with covariance $\mathbf{C}_k^\delta$ in 2D
$(\delta_k^i, \eta_k^i)_{i=1,\ldots,r}$	noise in angular domain with covariance $\mathbf{C}_k^{\delta,\eta}$ in 3D
λ	relaxation value for approximation
$\gamma_1, \ldots, \gamma_e$	evaluation angles
E_k	RMSE at evaluation angles

2. RELATED WORK

Traditional methods for surface reconstruction exclusively rely on position measurements. For example, Hoppe et al. presented a method to reconstruct a surface based on unorganized points [11]. As a result of the increasingly widespread use of depth cameras, algorithms exclusively based on depth measurements have been introduced, such as Kinect Fusion [13], [25].

However, approaches that try to combine both types of measurements are still fairly new. An early approach was published by Lindner et al. in 2007 [20]. This approach combines information from a TOF camera and a binocular camera to obtain a high-resolution colored point cloud. However, the color information is not used to obtain a more accurate estimate of the 3D shape of the observed scene.

Gudmundsson et al. [9] proposed a fusion algorithm for disparity maps obtained from stereo reconstruction and depth information obtained by TOF cameras in order to obtain higher quality disparity maps. The fusion is achieved by converting the TOF depth values to disparities in the image frames. A more sophisticated approach to the same problem has been presented by Zhu et al. in 2011 [32]. It is based on Markov Random Fields that describe the depth information and can be used to probabilistically combine measurements from a stereo camera system and a time-of-flight (TOF) camera. In the fusion process, the respective uncertainties of both sensor types are considered and their influence is weighted accordingly.

In 2008, Guan et al. [8] proposed an algorithm to combine images from several conventional cameras and a TOF camera for the purpose of 3D object reconstruction. The conventional cameras are used to provide silhouette information and allow construction of the visual hull, whereas the TOF camera is able to obtain depth information in areas where concavities occur. A probabilistic space occupancy grid, i.e., a voxel-based approach was used to obtain the object shape by calculating iso-probability surfaces with a graph-cut algorithm.

TABLE I
Comparison of fusion approaches.

Method	representation	stochastic	recursive
Lindner [20]	RGBD data	no	no
Gudmundsson [9]	disparity map	no	no
Zhu [32]	disparity map	yes	no
Guan [8]	voxel	yes	no
Groch [7]	voxel	yes	no
proposed	spline	yes	yes

Groch et al. [7] have applied a very similar approach to a medical application, but they fused stereo disparities instead of silhouette information.

There are different ways of representing the reconstructed surface. Depth sensors usually provide depth maps or point clouds as raw data, but a more sophisticated representation is desired. One common approach is to use spatial discretization and represent the surface as voxels [13], [8]. However, voxel-based approaches typically require a lot of memory and computational power if a high resolution is to be achieved. Another common approach is to use triangular meshes [22]. While triangular meshes are usually more computationally efficient than voxels, they require a large number of triangles to provide a satisfactory reconstruction of rounded shapes. An alternative is to describe the surface as a spline, which can be stored in a very compact form as it is uniquely defined by a small number of control points [19], [2], [1]. Splines are very suitable for smooth surfaces without rough edges. Unlike voxels or triangular meshes, splines can be evaluated at an arbitrary resolution and still appear smooth. Stochastic formulations of splines have previously been used by Brunn et al. [5]. Gaussian processes can also be used to model uncertain surfaces [26]. In fact, the spline based interpolation used in the proposed approach could easily be replaced with a Gaussian process regression.

Furthermore, we have to distinguish between recursive approaches that are able to include new information as it is obtained and methods that only use information from a single time step. Kinect Fusion [13] considers point clouds at consecutive time steps and tries to align them with the well-known iterative closest point algorithm [4]. The proposed algorithm is also capable of gaining more information over time because it is based on nonlinear recursive filtering methods. However, most approaches in this area are not recursive, i.e., they only consider a single time step.

An overview of the mentioned approaches is given in Table I. It should be noted that some approaches are based on stochastic foundations, e.g., Markov Random Fields, whereas others do not explicitly consider stochastic uncertainties.

3. PREREQUISITES

Before describing the proposed method for surface reconstruction, we introduce some prerequisites.

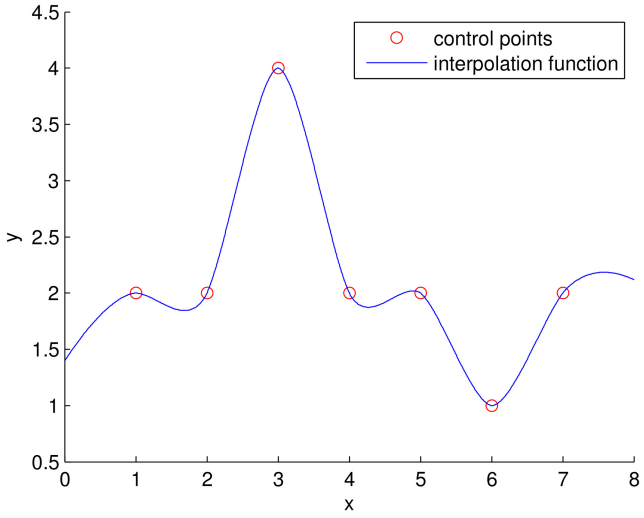

Fig. 3. Example of the interpolation achieved by the presented algorithm for $\mathbb{R} \to \mathbb{R}$ interpolation, i.e., for $n = 1$. The control points are $p_1,\ldots,p_7 = 1,2,3,4,5,6,7$ with values $f_1,\ldots,f_7 = 2,2,4,2,2,1,2$ and the basis function $\phi(x)$ is a thin plate spline.

3.1. Interpolation

Let $m \in \mathbb{N}_{>0}$, $\underline{p}_1,\ldots,\underline{p}_m \in \mathbb{R}^n$ and $f_1,\ldots,f_m \in \mathbb{R}$. The goal of interpolation is to find a function $f : \mathbb{R}^n \to \mathbb{R}$ with $f(\underline{p}_i) = f_i$ for $1 \le i \le m$ where f is smooth in some sense. There are various types of interpolation functions. An overview can be found in [1]. The proposed method does not depend on a particular choice of interpolation method. For later use, we define a function

$$\text{interpolate: } ((\mathbb{R}^n)^m \times \mathbb{R}^m) \to (\mathbb{R}^n \to \mathbb{R}),$$

$$(\underline{p}_1,\ldots,\underline{p}_m; f_1,\ldots,f_m) \mapsto f$$

that maps points $\underline{p}_1,\ldots,\underline{p}_m$ and values $f_1,\ldots,f_m$ to their interpolating function $f \in (\mathbb{R}^n \to \mathbb{R})$.

For the purpose of our experiments, we decided to use Radial Basis Functions (RBF) [6], [2] for interpolation, because they are easy to calculate by solving a system of linear equations and are applicable for any dimension n. The interpolating function f is given by

$$f(\underline{p}) = \sum_{j=1}^{m} c_j \cdot \phi(\|\underline{p} - \underline{p}_j\|),$$

where $\phi : \mathbb{R}_{\ge 0} \to \mathbb{R}$ is the basis function, $c_1,\ldots,c_m \in \mathbb{R}$ are weighting coefficients, and $\|\cdot\|$ is the Euclidean norm. A popular choice of basis function is the thin plate spline (TPS):

$$\phi(x) = \begin{cases} x^2 \log x, & x > 0 \\ 0, & x = 0 \end{cases}.$$

Because the value of $\log(0)$ is undefined, we set $\phi(0) = \lim_{x\to 0}(x^2 \log(x)) = 0$. The reason why the TPS is commonly used as a basis function, is the fact that the interpolation function is C^1 continuous and that it minimizes the energy functional

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dx\,dy,$$

which means that it provides in a certain sense the smoothest possible interpolation function.

The weighting coefficients $c_1,\ldots,c_m$ can be obtained by solving the system of m linear equations

$$f_i = \sum_{j=1}^{m} c_j \cdot \phi(\|\underline{p}_i - \underline{p}_j\|), \quad 1 \le i \le m.$$

The algorithm is given in Algorithm 1. We show an example of the interpolation produced by this algorithm in Fig. 3.

ALGORITHM 1 *Interpolation based on RBFs.*

Input: radial basis function $\phi : \mathbb{R}_{\ge 0} \to \mathbb{R}$;
points $p_1,\ldots,p_m \in \mathbb{R}^n$;
values $f_1,\ldots,f_m \in \mathbb{R}$

Output: interpolation function $f : \mathbb{R}^n \to \mathbb{R}$

// prepare matrix of RBF values
$\mathbf{A} \leftarrow (m \times m$ matrix);
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow i$ **to** m **do**
    $\mathbf{A}(i,j) \leftarrow \phi(\|p_i - p_j\|)$;
    // $\mathbf{A}$ is symmetric
    $\mathbf{A}(j,i) \leftarrow \mathbf{A}(i,j)$;
  end
end
// solve $\mathbf{A} \cdot [c_1,\ldots c_m]^T = [f_1,\ldots,f_m]^T$
$[c_1,\ldots c_m]^T \leftarrow \mathbf{A}^{-1}[f_1,\ldots,f_m]^T$;
// obtain interpolation function
$f \leftarrow (p \mapsto \sum_{j=1}^{m} c_j \cdot \phi(\|p - p_j\|))$;
return f;

3.2. Polar and Spherical Coordinates

While many common approaches rely on Cartesian coordinates exclusively, we use polar coordinates (in 2D) and spherical coordinates (in 3D) to simplify certain computations similar to [3]. The transformation between Cartesian and polar coordinates is given by

$$x = r\cos(\phi),$$

$$y = r\sin(\phi),$$

and

$$r = \|(x,y)^T\| = \sqrt{x^2 + y^2},$$

$$\phi = \text{atan2}(y,x).$$

For spherical coordinates, there are several common definitions. We use the convention

$$x = r\cos(\theta)\cos(\phi),$$

$$y = r\cos(\theta)\sin(\phi),$$

$$z = r\sin(\theta),$$

and

$$r = \|(x,y,z)^T\| = \sqrt{x^2 + y^2 + z^2},$$

$$\phi = \text{atan2}(y,x),$$

$$\theta = \arcsin(z/r).$$

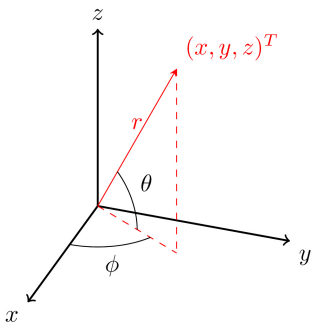

Fig. 4. Spherical coordinates as used in this paper.

An illustration of the meaning of ϕ and θ is given in Fig. 4. Be aware that we use θ as the angle between the x-y-plane and the vector $(x,y,z)^T$ whereas some common definitions use θ as the angle between the z-axis and the $(x,y,z)^T$ vector. It also deserves mentioning that some authors reverse the roles of ϕ and θ.

3.3. System and State Representation

For describing the estimate of the reconstructed surface at time step k, we use the state vector $\underline{x}_k^e \in \mathbb{R}^q$. We assume the state to be Gaussian-distributed with covariance matrix $\mathbf{C}_k^e$. A system model

$$\underline{x}_{k+1}^p = a_k(\underline{x}_k^e) + \underline{u}_k$$

with system function $a_k : \mathbb{R}^q \to \mathbb{R}^q$ and additive zero-mean Gaussian noise $\underline{u}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^u)$ can be used to describe the evolution of the state $\underline{x}_k$.

The structure of the system with the estimator is depicted in Fig. 5. If the system model is linear, the Kalman filter formulas [15] can be used to perform the prediction step. Otherwise a nonlinear filter such as the unscented Kalman filter (UKF, [14]) can be applied. Note that even if the system model is linear, we still need a nonlinear filter for the depth measurement update.

If the system does not follow any known dynamics, a random walk model may be used. In static cases, where the surface does not change over time, prediction can be omitted. As our approach for estimating the reconstructed surface is independent of the particular details of the system model, we will focus on the measurement model from now on.

4. SURFACE RECONSTRUCTION IN 2D

Let us first consider the 2D case. Although the 2D case might not seem relevant at first, there are actually a number of applications for 2D surface reconstruction. For example, LIDAR (light detection and ranging) sensors are commonly used in robotics and allow the reconstruction of obstacles as surfaces in 2D [30].

4.1. Position Measurements

We consider a set of $l \in \mathbb{N}$ landmarks on the surface. For tracking these landmarks, we define the state vector at time step k as

$$\underline{x}_k = [x_k^{1,1}, x_k^{1,2} \ldots, x_k^{l,1}, x_k^{l,2}]^T \in \mathbb{R}^{2l},$$

where $x_k^{a,b}$ represents the position of landmark $a \in \{1,\ldots,l\}$ in dimension $b \in \{1,2\}$ at time step k. In this case, the measurement model is trivially given by

$$\hat{\underline{y}}_k = \mathbf{I}_{2l\times 2l} \cdot \underline{x}_k + \underline{v}_k,$$

where $\mathbf{I}_{2l\times 2l} \in \mathbb{R}^{2l\times 2l}$ is the identity matrix, $\hat{\underline{y}}_k$ is the measurement at time step k, and $\underline{v}_k$ is additive Gaussian noise with $\underline{v}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^v)$. As the measurement equation

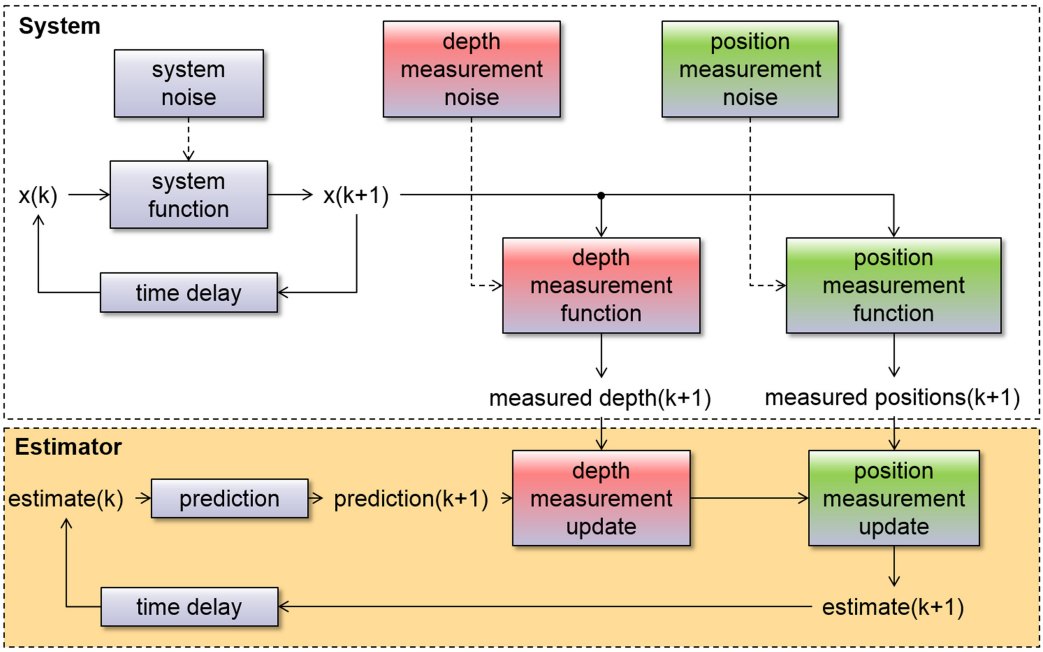

Fig. 5. The structure of the system and the recursive estimator. We propose two separate measurement updates for depth and position measurements.

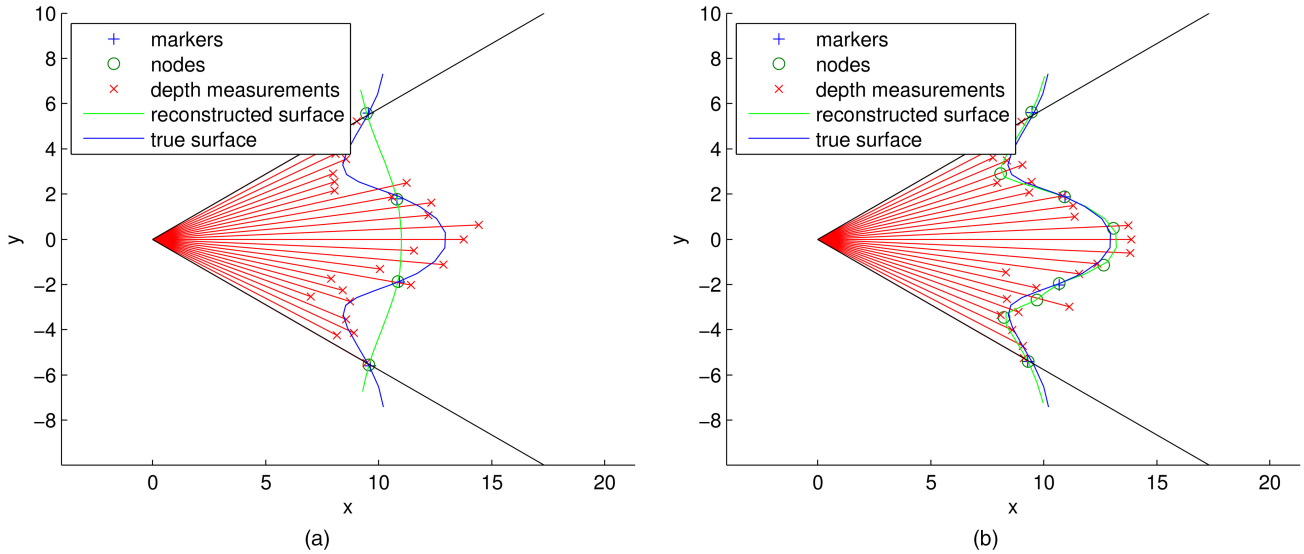

Fig. 6. Reconstructed and true surfaces in 2D. (a) Before state augmentation. (b) After state augmentation.

is linear, a Kalman filter [15] can be used to perform the measurement update. The surface s_k at time step k can be found by performing an interpolation through the currently estimated positions of the landmarks with any suitable interpolation method.

4.2. Depth Measurements

In addition to position measurements, we now want to include depth measurements into the estimation procedure. For the moment, we assume a single depth camera. Without loss of generality, we define it to be located at the origin of the coordinate system and facing towards $[1,0]^T$. We further assume that the depth camera can obtain $r \in \mathbb{N}$ depth measurements $\hat{z}_k^1,\dots,\hat{z}_k^r$ at r different angles $\alpha_1,\dots,\alpha_r$. These angles are typically evenly spread across the depth camera's field of view. Consequently, the measurement equation has to calculate the intersections of the lines at angles $\alpha_1,\dots,\alpha_r$ with the surface. Depending on the surface representation, calculating this intersection can be very difficult. One of the key ideas of our approach is to use polar coordinates, which nicely circumvents this problem. If we parameterize the surface as a function $s_k : \mathbb{R} \to \mathbb{R}$ which maps angles α to distances $s_k(\alpha)$, the intersection for the lines at angles $\alpha_1,\dots,\alpha_r$ are trivially calculated as $s_k(\alpha_1),\dots,s_k(\alpha_r)$.

This yields the measurement equation

$$\underline{\hat{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1) \\ \vdots \\ s_k(\alpha_r) \end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha) = \text{interpolate } (p_k; f_k)(\alpha),$$

$$p_k = (\text{atan2}(x_k^{1,2}, x_k^{1,1}),\dots,\text{atan2}(x_k^{l,2}, x_k^{l,1})),$$

$$f_k = \left(\left\| \begin{bmatrix} x_k^{1,1} \\ x_k^{1,2} \end{bmatrix} \right\|, \dots, \left\| \begin{bmatrix} x_k^{l,1} \\ x_k^{l,2} \end{bmatrix} \right\| \right)$$

with measurements $\underline{\hat{z}}^k$ and Gaussian noise $\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$. Intuitively, the landmark coordinates $(x_k^{i,1}, x_k^{i,2})_{1 \le i \le l}$ are converted to polar coordinates with angular coordinates p_k and radial coordinates f_k. From these polar coordinates, the interpolation function $s_k(\cdot)$ is obtained. By evaluating the interpolation function $s_k(\cdot)$ at the measurement angles $\alpha_1,\dots,\alpha_r$, we calculate the measurement, which is finally disturbed by noise $\underline{w}_k$.

Be aware that w_k not only contains the stochastic error from the noise of the depth camera but also a modeling error as a result of the spline interpolation. The equations for p_k and f_k follow from the conversion of Cartesian into polar coordinates as described in Sec. 3.2. The surface function s_k is derived from $\underline{x}_k$ by interpolation, which is in general nonlinear in $\underline{x}_k$. Thus, it is necessary to use a nonlinear filter to perform the depth measurement update. For example, the UKF [14], the S^{2}KF (Smart Sampling Kalman Filter) [28], or the Gaussian filter introduced in [12] may be used.

4.3. State Augmentation

While it is possible to use depth measurements as described previously, the achievable accuracy is still strongly limited by the number of position measurements. The reason for this issue is the fact that the number of degrees of freedom of the reconstructed surface is determined by the number of position measurements. An example of this limitation is depicted in Fig. 6. Before state augmentation, all four markers are estimated correctly, but the estimated surface is very different from the true surface. After augmenting the state, additional nodes give the surface more degrees of freedom and the true surface can be approximated much more closely.

To improve accuracy, we augment the state by additional control points that do not correspond to landmarks. One may be tempted to augment the state by the Cartesian coordinates of points in $\mathbb{R}^2$, which lie some-

where on the surface, and to try to estimate their position. However, as these points do not produce measurements originating from a fixed position on the surface, their location cannot be uniquely determined from the measurements (see Fig. 2). Any position in space that leads to the same interpolated surface s_k is just as reasonable an estimate as any other. Consequently, the problem is underdetermined and the state is not observable.

The key idea is to introduce additional nodes not as arbitrary points in $\mathbb{R}^2$ but in polar coordinates as $d \in \mathbb{N}$ depths at certain fixed angles $\varphi_1,\dots,\varphi_d$. This yields an augmented state

$$\underline{x}_k = (\underbrace{x_k^{1,1}, x_k^{1,2}, \dots, x_k^{l,1}, x_k^{l,2}}_{\text{landmarks}}, \quad \underbrace{x_k^{1,*}, \dots, x_k^{d,*}}_{\text{additional control points}} \quad)^T \in \mathbb{R}^{2l+d},$$

where $x_k^{1,*} \dots x_k^{d,*}$ are the depths at angles $\varphi_1,\dots,\varphi_d$. These angles are not part of the state, as they are not estimated but chosen as fixed values. Consequently the depths $x_k^{1,*} \dots x_k^{d,*}$ are uniquely determined by the shape of the surface and, hence, the state is observable.

This poses the question how the angles $\varphi_1,\dots,\varphi_d$ should be chosen. Simple approaches include random angles inside the camera's field of view or angles that lie on an evenly spaced grid. More sophisticated ways to choose appropriate angles are discussed in Sec. 6.1.

The measurement equation for positions becomes

$$\hat{\underline{y}}_k = [\mathbf{I}_{2l\times 2l} \mathbf{0}_{2l\times d}] \underline{x}_k + \underline{v}_k,$$

with identity matrix $\mathbf{I}_{2l\times 2l} \in \mathbb{R}^{2l\times 2l}$ and zero matrix $\mathbf{0}_{2l\times d} \in \mathbb{R}^{2l\times d}$, which just ignores the additional control points. The measurement equation for depth becomes

$$\hat{\underline{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1) \\ \vdots \\ s_k(\alpha_r) \end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha) = \text{interpolate } (p_k; f_k)(\alpha),$$

$$p_k = (\text{atan2}(x_k^{1,2}, x_k^{1,1}), \dots, \text{atan2}(x_k^{l,2}, x_k^{l,1}), \varphi_1, \dots, \varphi_d),$$

$$f_k = \left(\left\| \begin{bmatrix} x_k^{1,1} \\ x_k^{1,2} \end{bmatrix} \right\|, \dots, \left\| \begin{bmatrix} x_k^{l,1} \\ x_k^{l,2} \end{bmatrix} \right\|, x_k^{1,*}, \dots, x_k^{d,*} \right),$$

which now includes the additional control points in the interpolation process. Once again, the landmark coordinates $(x_k^{i,1}, x_k^{i,2})_{1\le i\le l}$ are converted to polar coordinates. Together with the additional control points, whose polar coordinates are $(\phi_i, x_k^{i,*})_{1\le i\le d}$, they form a list of points with angular coordinates p_k and radial coordinates f_k. The surface $s_k(\cdot)$ is calculated by interpolation through all of these points. When augmenting the state, the covariance matrix is augmented as well according to

$$\mathbf{C}_k^e \leftarrow \begin{bmatrix} \mathbf{C}_k^e & \mathbf{0}_{2l\times d} \\ \mathbf{0}_{d\times 2l} & \sigma_{\text{add}}^2 \cdot \mathbf{I}_{d\times d} \end{bmatrix}$$

with uncertainty $\sigma_{\text{add}} > 0$.

Augmenting the state vector also involves augmenting the system model. For this purpose, we define an augmented system function $a_k : \mathbb{R}^{2l+d} \to \mathbb{R}^{2l+d}$ that maps the augmented state at time step k to the augmented state at time step $k+1$. The system noise $\underline{u}_k$ is augmented to $2d+l$ dimensions as well. The details of this augmentation depend on the system model that is used for the given application. In case of a random walk model, the additional dimensions can be assumed to conform to a random walk model as well.

5. SURFACE RECONSTRUCTION IN 3D

For many applications that are relevant in practice, 3D surface reconstruction is required. Fortunately, the presented methods can easily be applied to a 3D setting as well.

To accommodate for the third dimension, a few changes are required. Positions in $\mathbb{R}^2$ are replaced with positions in $\mathbb{R}^3$ and polar coordinates are replaced with spherical coordinates. We also change the surface representation to a function $s_k : \mathbb{R}^2 \to \mathbb{R}$ that maps pairs of angles (α,β) to distances $s_k(\alpha,\beta)$. Once again we assume a single depth camera. Without loss of generality, it is located in the origin, its orientation is aligned with the coordinate system, and it is facing towards $[1,0,0]^T$.

These modifications yield the state representation

$$\underline{x}_k = [\underbrace{x_k^{1,1}, x_k^{1,2}, x_k^{1,3}, \dots, x_k^{l,1}, x_k^{l,2}, x_k^{l,3}}_{\text{landmarks}}, \underbrace{x_k^{1,*}, \dots, x_k^{d,*}}_{\text{additional control points}} \quad]^T \in \mathbb{R}^{3l+d},$$

where $x_k^{a,b}$ represents the position of landmark a in dimension $b \in \{1,2,3\}$ at time step k. The angles of the additional control points $(\varphi_1,\theta_1),\dots,(\varphi_d,\theta_d)$ are once again fixed and not part of the state. The measurement equation for positions is now

$$\hat{\underline{y}}_k = [\mathbf{I}_{3l\times 3l} \mathbf{0}_{3l\times d}] \underline{x}_k + \underline{v}_k,$$

and the measurement model for depth is

$$\hat{\underline{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1,\beta_1) \\ \vdots \\ s_k(\alpha_r,\beta_r) \end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha,\beta) = \text{interpolate } (p_k; f_k)(\alpha,\beta),$$

$$p_k = \left(\begin{bmatrix} \text{atan2}(x_k^{1,2}, x_k^{1,1}) \\ \arcsin\left(\dfrac{x_k^{1,3}}{\|(x_k^{1,1}, x_k^{1,2}, x_k^{1,3})^T\|} \right) \end{bmatrix}, \dots, \begin{bmatrix} \text{atan2}(x_k^{l,2}, x_k^{l,1}) \\ \arcsin\left(\dfrac{x_k^{l,3}}{\|(x_k^{l,1}, x_k^{l,2}, x_k^{l,3})^T\|} \right) \end{bmatrix}, \begin{bmatrix} \varphi_1 \\ \theta_1 \end{bmatrix}, \dots, \begin{bmatrix} \varphi_d \\ \theta_d \end{bmatrix} \right),$$

$$f_k = \left(\left\| \begin{bmatrix} x_k^{1,1} \\ x_k^{1,2} \\ x_k^{1,3} \end{bmatrix} \right\|, \ldots, \left\| \begin{bmatrix} x_k^{l,1} \\ x_k^{l,2} \\ x_k^{l,3} \end{bmatrix} \right\|, x_k^{1,*}, \ldots, x_k^{d,*} \right),$$

where $(\alpha_1, \beta_1), \ldots, (\alpha_r, \beta_r)$ are the angles at which depth measurements are obtained. The expressions for p_k and f_k follow from the conversion of Cartesian into spherical coordinates as introduced in Sec. 3.2.

6. ENHANCEMENTS OF THE PROPOSED METHOD

In this section, we present several enhancements to the proposed method. These enhancements may not be absolutely necessary to apply the proposed methods but improve their performance and allow their use in a wider variety of applications.

6.1. Adaptive Addition of Nodes

When augmenting the state by adding control points, one has to choose the angles where the additional nodes should be located. In the 2D case, we need to determine the angles $\varphi_1, \ldots, \varphi_d$ and in the 3D case the pairs of angles $(\varphi_1, \theta_1), \ldots, (\varphi_d, \theta_d)$. Simple approaches may involve picking these angles at random within the view of the depth camera or choosing angles that lie on a grid.

However, these choices are usually not optimal. Additional control points should be added adaptively in areas where the error is large or where the expected gain in accuracy is high. One approach for the 2D case is to calculate the RMSE (root mean square error) between estimate and measurement

$$E_k^\tau(\alpha_i) = \sqrt{\frac{1}{\tau} \sum_{j=0}^{\tau-1} (s_{k-j}(\alpha_i) - \hat{z}_{k-j}^i)^2}$$

at time step k within a sliding window of length $\tau \in \mathbb{N}$ for each angle $\alpha_1, \ldots, \alpha_r$. A large RMSE suggests a control point at this position may be desirable and thus one should choose

$$\varphi = \underset{\alpha_i, 1 \le i \le r}{\arg\max} (E_k^\tau(\alpha_i))$$

as the angle of the new control point. This approach can be generalized to the 3D case by using the error function

$$E_k^\tau(\alpha_i, \beta_i) = \sqrt{\frac{1}{\tau} \sum_{j=0}^{\tau-1} (s_{k-j}(\alpha_i, \beta_i) - \hat{z}_{k-j}^i)^2}$$

and obtaining the new control point according to

$$(\varphi, \theta) = \underset{(\alpha_i, \beta_i), 1 \le i \le r}{\arg\max} (E_k^\tau(\alpha_i, \beta_i)).$$

In our experiments, we found that adding nodes successively tends to give better results than adding several nodes at once.

6.2. Handling Missing Measurements

In a practical setting, both position and depth measurements may be missing, for example when a tracked landmark is occluded or when the depth sensor is unable to provide a valid depth measurement at a certain angle. It is possible to handle these cases with slight modifications to the proposed method. The measurement models for both position and depth can simply omit the entries of $\hat{\underline{y}}_k$ and $\hat{\underline{z}}_k$ that could not be measured at time step k. Consequently, even a surface that is never visible as a whole at any given time step can be reconstructed over time.

6.3. Angular Uncertainty

The approach, as introduced before, assumes that the measurement angles $\alpha_1, \ldots, \alpha_r$ are exactly known and that only the depth values at these angles are affected by noise. This is a realistic assumption for certain depth sensors like TOF-cameras, where depth measurements are noisy but the measurement angles are fixed by the optical properties of the camera. However, depth sensors based on other measurement principles such as sensors based on structured light may not fulfill this assumption.

It is possible to extend the presented approach to include uncertainty in the measurement angles in order to more accurately model depth sensors whose measurement angles are affected by non-negligible noise. For this purpose, the measurement equation for depth has to be modified to include non-additive noise terms. In the 2D case, this yields

$$\hat{\underline{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1 + \delta_k^1) \\ \vdots \\ s_k(\alpha_r + \delta_k^r) \end{bmatrix} + \underline{w}_k,$$

where $[\delta_k^1, \ldots, \delta_k^r]^T \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^\delta)$ is zero-mean Gaussian noise with covariance $\mathbf{C}_k^\delta$. Nonlinear filters such as the UKF can handle this kind of noise, typically by means of augmenting the state with the non-additive noise components. This method is also applicable to the 3D case by changing the measurement equation to

$$\hat{\underline{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1 + \delta_k^1, \beta_1 + \eta_k^1) \\ \vdots \\ s_k(\alpha_r + \delta_k^r, \beta_r + \eta_k^r) \end{bmatrix} + \underline{w}_k,$$

where $[\delta_k^1, \ldots, \delta_k^r, \eta_k^1, \ldots, \eta_k^r]^T \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^{\delta,\eta})$ is zero-mean Gaussian noise with covariance $\mathbf{C}_k^{\delta,\eta}$.

6.4. More Than One Depth Camera

We previously assumed that our surface was observed by just a single depth camera. This assumption can be dropped by selecting a reference camera and transforming the depth measurements of additional cameras into the coordinate system of the reference camera. The relation between the coordinate systems

and the associated uncertainties can be obtained with standard camera calibration procedures [31], [10].

If the cameras are observing the surface from sufficiently different angles, angular uncertainty may be introduced by the transformation into a different coordinate system even if there was only uncertainty in depth initially. If necessary, the angular uncertainty can be handled as described in the previous section.

6.5. Choice of Interpolation or Approximation Function

In Sec. 3.1, we introduced the commonly used interpolation method based on thin plate splines. However, it should be noted that the proposed approach is not tied to a particular interpolation method. Other interpolation schemes can easily be used by replacing the interpolate(p_k; f_k) function with a different algorithm. Overviews of possible interpolations methods can be found in [1], [33], and [21]. Depending on the application, the true shape of the surface may be more closely approximated by a certain type of interpolation. Furthermore, the choice of interpolation method influences the scalability of the algorithm for a large number of points m.

Furthermore, it is possible to consider a relaxed interpolation problem and to require approximation only. This means that $f(\underline{p}_i) = f_i$ does not need to hold exactly, but we only require that $f(\underline{p}_i) \approx f_i$. In the case of thin plate splines this can easily be achieved by setting $A(i,i) = \phi(0) + \lambda$ for $i = 1,\dots,m$ in Algorithm 1, where $\lambda > 0$ controls how strongly the problem is relaxed [27]. In our experiments, using an approximation rather than an interpolation has proven to be more stable, because outliers that result from poorly estimated points no longer affect the shape of the surface as much.

7. EVALUATION

In order to evaluate the proposed algorithm, we have performed several simulations. All simulations use the UKF [14] for nonlinear filtering. We use the following constants:

- initial estimate $\underline{x}_0^e$: uniformly random between $\underline{0}$ and $\underline{1}$
- initial covariance: $\mathbf{C}_0^e = 10 \cdot \mathbf{I}_{q\times q}$
- initial variance for additional nodes: $\sigma_{\text{add}}^2 = 10$
- noise covariance for position: $\mathbf{C}_k^v = 0.01 \cdot \mathbf{I}_{q\times q}$
- noise covariance for depth: $\mathbf{C}_k^w = 1 \cdot \mathbf{I}_{r\times r}$

For interpolation, we apply the RBF algorithm depicted in Algorithm 1 and use a scaled version of the thin plate spline as the RBF:

$$\phi(x) = \begin{cases} (x/1000)^2 \log(x/1000), & x > 0 \\ 0, & x = 0 \end{cases}.$$

7.1. Simulations in 2D

As a performance measure, we want to determine how similar the reconstructed surface is to the true surface. For this purpose, we choose $e \in \mathbb{N}$ evaluation angles $\gamma_1,\dots,\gamma_e$ and define the RMSE E_k of the estimated surface at time step k as

$$E_k = \sqrt{\frac{1}{e}\sum_{i=1}^{e}(s_k(\gamma_i) - s_k^{\text{true}}(\gamma_i))^2},$$

where $s_k^{\text{true}}(\cdot)$ is the true surface in polar coordinates. This can be interpreted as the error in depth, measured from the camera towards the surface.

We consider a depth camera with a viewing angle of 60° and a resolution of $r = 25$ measurements at equidistant measurement angles. There are $e = 26$ evaluation angles, which are equidistant in a 72° angle around the camera center, so we evaluate the extrapolation capability of the algorithm as well.

7.1.1 Static Case:

The true surface that we try to estimate is given by

$$s_k^{\text{true}}(\gamma) = 11 + 2\cos(9 \cdot \gamma)$$

and does not change over time. Thus, we omit the prediction step. This surface is the same as depicted in Fig. 6. We start with $l = 4$ landmarks and no additional nodes. From time step $k = 10$ to time step $k = 20$ we add one node at each time step, so we have $d = 11$ additional nodes afterwards. The angles $\varphi_1,\dots,\varphi_d$ are chosen deterministically and are evenly distributed across the camera's view.

The simulation was carried out repeatedly and the median and mean RMSE of the results of 100 Monte Carlo runs are shown in Fig. 7a. As can clearly be seen, the error is very high until time step $k = 10$, because the surface description does not have a sufficient number of degrees of freedom. After all additional nodes have been inserted at $k = 20$, the estimate quickly converges to a point where it has a consistently low error.

7.1.2 Dynamic Case:

We consider the same situation as in the static case except for the fact that the surface is now time-variant. The moving surface is given by

$$s_k^{\text{true}}(\gamma) = 11 + 2\cos(9 \cdot \gamma) + \sin(0.1 \cdot k)$$

and the system model is assumed to be unknown. Consequently, we use a random walk model for prediction. The system noise is modeled by the covariance matrix $\mathbf{C}_k^{\xi} = \text{diag}(0.1,\dots,0.1)$.

Once again, we performed 100 Monte Carlo runs and calculated the mean and median RMSE. The results are depicted in Fig. 7b. Overall the results look similar to the static case, but the RMSE is generally higher as is to be expected. The deviation between mean and median shows that there are a few outliers, so estimation is not quite as robust as in the static case.

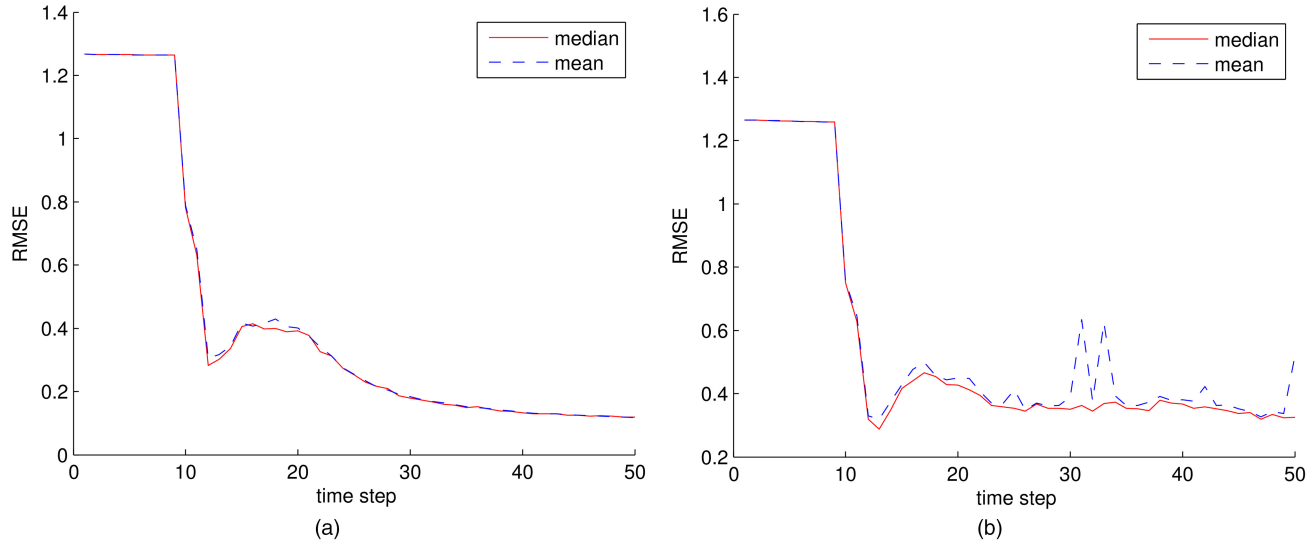

Fig. 7. Median and mean RMSE for each time step in the 2D case. Additional control points are inserted from time step $k = 10$ until $k = 20$. (a) Static surface. (b) Dynamic surface.

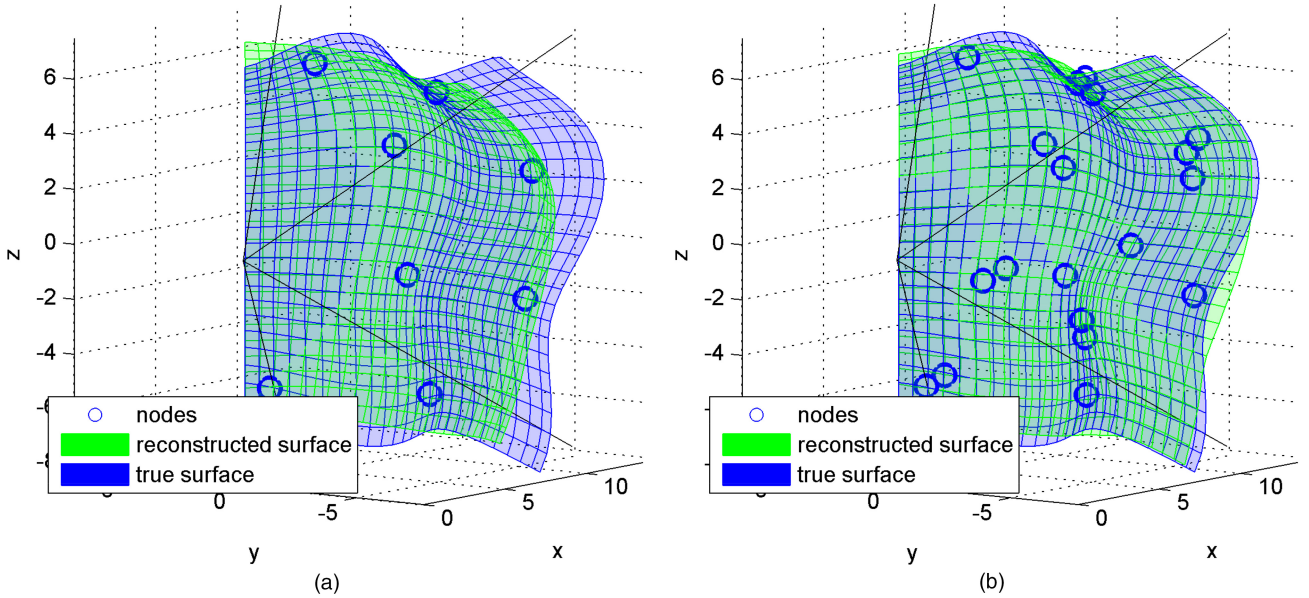

Fig. 8. Reconstructed and true surfaces in 3D (static case). (a) Time step $k = 9$, without additional control points. (b) Time step $k = 50$, with additional control points.

7.2. Simulations in 3D

Similar to the 2D case, we choose pairs of evaluation angles $(\gamma_1,\delta_1),\ldots,(\gamma_e,\delta_e)$ and define the RMSE E_k at time step k as

$$E_k = \sqrt{\frac{1}{e}\sum_{i=1}^{e}(s_k(\gamma_i,\delta_i) - s_k^{\text{true}}(\gamma_i,\delta_i))^2},$$

where $s_k^{\text{true}}(\cdot,\cdot)$ is the true surface in spherical coordinates.

We assume a depth camera with a horizontal and vertical viewing angle of 60° and a resolution of 25×25, so $r = 25^2 = 625$. The measurement angles are located on an equidistant 25×25 grid. For evaluation, we use vertical and horizontal angle of 72° and 26×26 equidistant evaluation angles.

7.2.1 Static Case:

Similar to the static 2D case, we consider a time-invariant surface. In spherical coordinates, it is given by

$$s_k^{\text{true}}(\gamma,\delta) = 12 + \sin(7\cdot\gamma) + \sin(7\cdot\delta).$$

Because the 3D surface has more degrees of freedom, we start with $l = 8$ landmarks. Once again, we introduce $d = 11$ additional control points from time step $k = 10$ to $k = 20$. The pairs of angles $(\varphi_1,\theta_1),\ldots,(\varphi_d,\theta_d)$ are evenly distributed across the field of view of the camera in a deterministic way. The surface before and after introducing additional nodes is depicted in Fig. 8. The

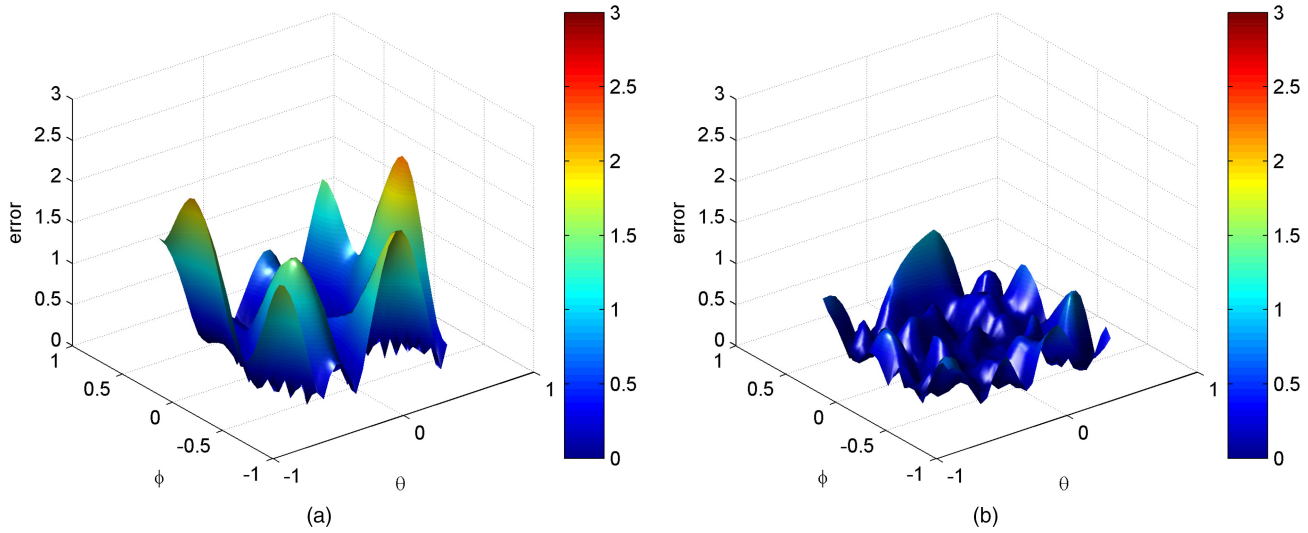

Fig. 9. Absolute error in 3D (static case). (a) Time step $k = 9$, without additional control points. (b) Time step $k = 50$, with additional control points.

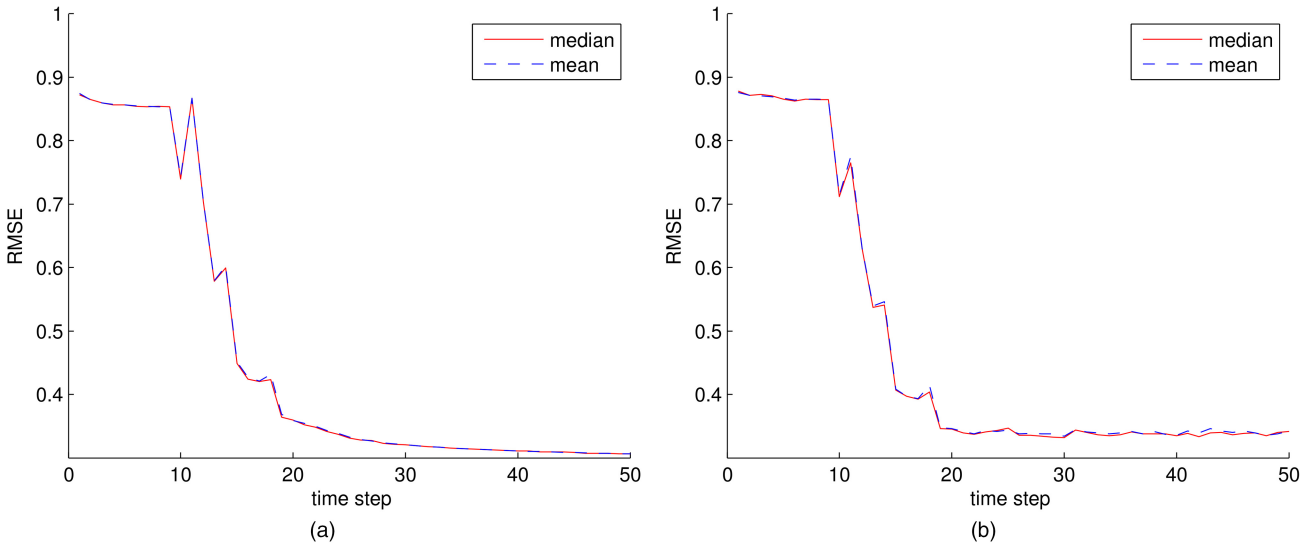

Fig. 10. Median and mean RMSE for each time step in the 3D case. Additional control points are inserted from time step $k = 10$ until $k = 20$. (a) Static case. (b) Dynamic case.

error between true and reconstructed surface can be seen in Fig. 9.

Our results from 100 Monte Carlo runs are shown in Fig. 10a. A comparison with Fig. 7a shows little difference to the 2D case, although the range of values is different because a different surface is reconstructed.

7.2.2 Dynamic Case:

For the dynamic 3D case we consider the time-variant surface

$$s_k^{\text{true}}(\gamma, \delta) = 12 + \sin(7 \cdot \gamma) + \sin(7 \cdot \delta) + \sin(0.1 \cdot k).$$

The system model is a random walk model with system noise $\mathbf{C}_k^c = \text{diag}(0.1, \ldots, 0.1)$. Fig. 10b shows the results from 100 Monte Carlo runs. This simulation demonstrates that our methods works almost as well in a dynamic as in a static setting.

7.3. Adaptive Addition of Nodes

In this section, we evaluate the approach proposed in Sec. 6.1 to add nodes adaptively in areas where the modeling error is large. We consider the same scenario as in the 2D static case (see Sec. 7.1.1). The sliding window has a length of $\tau = 9$. Three additional nodes are inserted at time steps $k = 10$, $k = 20$ and $k = 30$. For this simulation, we used an approximation with relaxation value $\lambda = 10^{-8}$ rather than an interpolation.

We performed 1000 Monte Carlo runs. The mean and median RMSE is depicted in Fig. 11a. The increase in accuracy is clearly visible each time a node is added. Fig. 11b shows the angles, at which the additional nodes are inserted. Obviously, these angles are not identical in each run, but it is clearly visible that the three additional nodes are mostly inserted at similar positions (approximately $-20°$, $0°$, and $20°$).

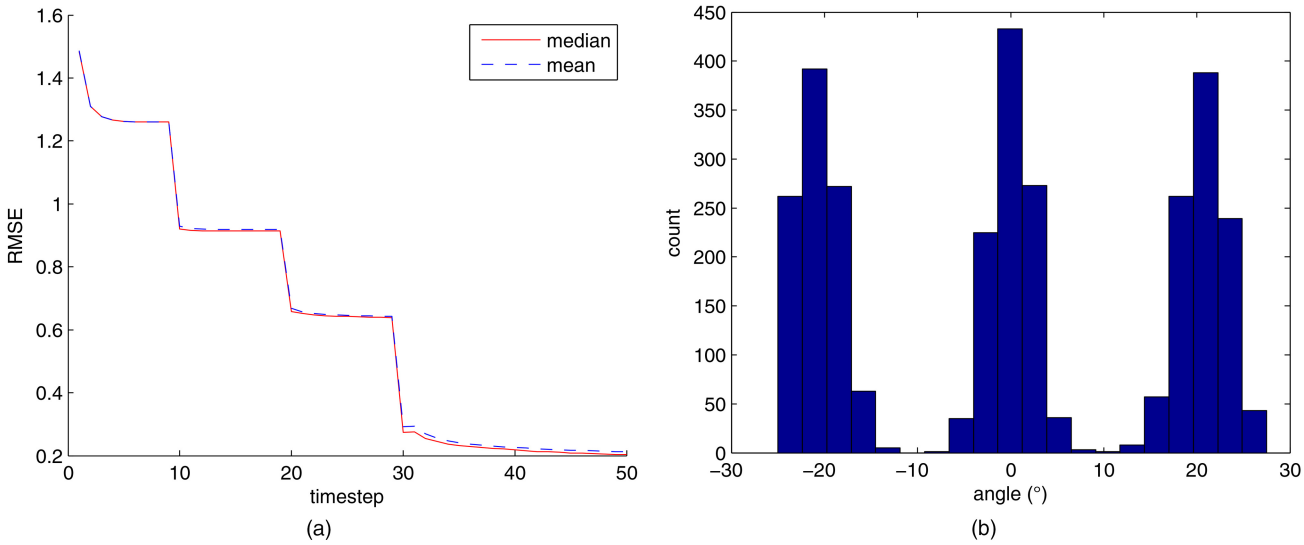

Fig. 11. Adaptive addition of nodes. Nodes are added at time steps $k = 10$, $k = 20$, and $k = 30$. (a) RMSE over time. (b) Angles where nodes are added.

8. CONCLUSION

We have presented an algorithm for recursively combining depth and position measurements for surface reconstruction considering uncertainties. Surface representation as a spline allows for a compact state representation. The measurement equation for position is trivial but deriving the measurement equation is challenging if Cartesian coordinates are used. In order to solve this issue, we have presented a way to use polar or spherical coordinates, which simplifies the problem significantly.

Through evaluation by means of simulations we have shown the viability of our approach in both 2D and 3D settings for static as well as dynamic surfaces. Our experiments clearly demonstrate the benefits of adding additional control points in order to better incorporate depth measurements. We also demonstrate the viability of the proposed method to adaptively choosing the location of additional nodes. The considered examples show how a much more accurate surface estimate can be obtained by the combined use of position as well as depth measurements.

Future research may include more sophisticated ways to insert additional control points. A practical application of the presented algorithm in a medical setting is planned.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) within the Research Training Groups RTG 1126 "*Intelligent Surgery—Development of new computer-based methods for the future working environment in visceral surgery*."

REFERENCES

[1] Amidror, I.
Scattered data interpolation methods for electronic imaging systems: A survey.
Journal of Electronic Imaging 11, 2 (Apr. 2002), 157–176.

[2] Arad, N., and Reisfeld, D.
Image warping using few anchor points and radial funcions.
Computer Graphics Forum 14 (Feb. 1995), 35–46.

[3] Baum, M., and Hanebeck, U. D.
Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs.
In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)* (Chicago, Illinois, USA, July 2011).

[4] Besl, P., and McKay, N. D.
A method for registration of 3-D shapes.
IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 2 (1992), 239–256.

[5] Brunn, D., and Hanebeck, U. D.
A Model-Based Framework for Optimal Measurements in Machine Tool Calibration.
In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)* (Barcelona, Spain, Apr. 2005).

[6] Carr, J. C., Fright, W. R., and Beatson, R. K.
Surface interpolation with radial basis functions for medical imaging.
IEEE Transactions on Medical Imaging 16 (1997), 96–107.

[7] Groch, A., Haase, S., Wagner, M., Kilgus, T., Kenngott, H., Schlemmer, H., Hornegger, J., Meinzer, H., and Maier-Hein, L.
A probabilistic approach to fusion of time of flight and multiple view based 3d surface reconstruction for laparoscopic interventions.
Int. J. CARS 7 (Suppl. 1) (2012), 397–398.

[8] Guan, L., Franco, J.-S., and Pollefeys, M.
3d object reconstruction with heterogeneous sensor data.
In *International Symposium on 3D Data Processing, Visualization and Transmission* (Atlanta, États-Unis, 2008).

[9] Gudmundsson, S. A., Aanaes, H., and Larsen, R.
Fusion of stereo vision and time-of-flight imaging for improved 3d estimation.
International Journal of Intelligent Systems Technologies and Applications 5, 3 (2008), 425–433.

[10] Hartley, R. I., and Zisserman, A.
Multiple View Geometry in Computer Vision, second ed.
Cambridge University Press, 2004.

[11] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W.
Surface reconstruction from unorganized points.
SIGGRAPH Comput. Graph. 26, 2 (July 1992), 71–78.

[12] Huber, M. F., and Hanebeck, U. D.
Gaussian Filter based on Deterministic Sampling for High Quality Nonlinear Estimation.
In *Proceedings of the 17th IFAC World Congress (IFAC 2008)* (Seoul, Republic of Korea, July 2008), vol. 17.

[13] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A.
Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera.
In *ACM Symposium on User Interface Software and Technology* (Oct. 2011), pp. 559–568.

[14] Julier, S. J., and Uhlmann, J. K.
Unscented filtering and nonlinear estimation.
Proceedings of the IEEE 92, 3 (Mar. 2004), 401–422.

[15] Kalman, R. E.
A new approach to linear filtering and prediction problems.
Transactions of the ASME Journal of Basic Engineering 82 (1960), 35–45.

[16] Keeling, W. B., Williams, M. L., Slaughter, M. S., Zhao, Y., and Puskas, J. D.
Off-pump and on-pump coronary revascularization in patients with low ejection fraction: a report from the society of thoracic surgeons national database.
The Annals of Thoracic Surgery 96, 1 (July 2013), 83–89.

[17] Kurz, G., and Hanebeck, U. D.
Image Stabilization with Model-Based Tracking for Beating Heart Surgery.
In *11. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC12)* (Düsseldorf, Germany, Nov. 2012).

[18] Kurz, G., and Hanebeck, U. D.
Recursive Fusion of Noisy Depth and Position Measurements for Surface Reconstruction.
In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)* (Istanbul, Turkey, July 2013).

[19] Lee, S., Wolberg, G., and Shin, S. Y.
Scattered data interpolation with multilevel B-splines.
IEEE Transactions on Visualization and Computer Graphics 3, 3 (1997), 228–244.

[20] Lindner, M., Kolb, A., and Hartmann, K.
Data-fusion of pmd-based distance-information and high-resolution rgb-images.
In *International Symposium on Signals, Circuits and Systems, 2007.* (2007), vol. 1.

[21] Liu, W., and Ribeiro, E.
A survey on image-based continuum-body motion estimation.
Image and Vision Computing 29, 8 (2011), 509–523.

[22] Lorensen, W. E., and Cline, H. E.
Marching cubes: A high resolution 3d surface construction algorithm.
SIGGRAPH Comput. Graph. 21, 4 (Aug. 1987), 163–169.

[23] Marton, Z. C., Rusu, R. B., and Beetz, M.
On fast surface reconstruction methods for large and noisy point clouds.
In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)* (May 2009), pp. 3218–3223.

[24] Nakamura, Y., Kishi, K., and Kawakami, H.
Heartbeat synchronization for robotic cardiac surgery.
In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)* (Seoul, Korea, May 2001), pp. 2014–2019.

[25] Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., and Fitzgibbon, A.
Kinectfusion: Real-time dense surface mapping and tracking.
In *10th IEEE international symposium on mixed and augmented reality (ISMAR)* (Oct. 2011), IEEE, pp. 127–136.

[26] Rasmussen, C. E., and Williams, C. K. I.
Gaussian Processes for Machine Learning.
The MIT Press, 2006.

[27] Sprengel, R., Rohr, K., and Stiehl, H. S.
Thin-plate spline approximation for image registration.
In *Proceedings of the 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (1996), vol. 3, IEEE, pp. 1190–1191.

[28] Steinbring, J., and Hanebeck, U. D.
S2KF: The Smart Sampling Kalman Filter.
In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)* (Istanbul, Turkey, July 2013).

[29] Stoyanov, D., Mylonas, G. P., Deligianni, F., Darzi, A., and Yang, G. Z.
Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures.
In *Proceedings of the Medical Image Computing and Computer Assisted Interventions (MICCAI 2005)* (2005), vol. 2, pp. 139–146.

[30] Ye, C., and Borenstein, J.
Characterization of a 2D laser scanner for mobile robot obstacle negotiation.
In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)* (2002), vol. 3, pp. 2512–2518.

[31] Zhang, Z.
A flexible new technique for camera calibration.
IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 11 (2000), 1330–1334.

[32] Zhu, J., Wang, L., Yang, R., Davis, J., and Pan, Z.
Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps.
IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 7 (July 2011), 1400–1414.

[33] Zitová, B., and Flusser, J.
Image registration methods: a survey.
Image and Vision Computing 21 (2003), 977–1000.

Gerhard Kurz received his Dipl.-Inform. in computer science from the the Karlsruhe Institute of Technology (KIT), Germany, in 2012. Currently, he is working towards a Ph.D. degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of medical data fusion, nonlinear estimation, and directional filtering.

Uwe D. Hanebeck received his Dipl.-Ing. in Electrical Engineering, with first-class honors, from the Ruhr-Universitaet Bochum, Germany, in 1991. He obtained his Dr.-Ing. (Ph.D.) degree in Electrical Engineering from the Technische Universitaet Muenchen, Germany, in 1997. In 2003 he was appointed as professor at the Universitaet Karlsruhe (TH). He is the head of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe Institute of Technology (KIT). His research interests are sensor-actuator-networks, human-centered robotics, and telepresence with emphasis on theory-driven approaches based on stochastic estimation theory and uncertainty models. Research results are applied to various fields like human-robot-cooperation, robotic surgery systems, personal assistance systems, and extended-range telepresence.