Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 902039, 11 pages
http://dx.doi.org/10.1155/2014/902039

Research Article

Using Ignorance in 3D Scene Understanding

Bogdan Harasymowicz-Boggio and Barbara Siemiątkowska

Faculty of Mechatronics, Warsaw University of Technology, Ulica św. Andrzeja Boboli 8, 02-525 Warsaw, Poland

Correspondence should be addressed to Bogdan Harasymowicz-Boggio; mysticdrow@gmail.com

Received 7 March 2014; Revised 2 June 2014; Accepted 16 June 2014; Published 7 July 2014

Academic Editor: Dongbing Gu

Awareness of its own limitations is a fundamental feature of the human sight, which has been almost completely omitted in computer vision systems. In this paper we present a method of explicitly using information about perceptual limitations of a 3D vision system, such as occluded areas, limited field of view, loss of precision along with distance increase, and imperfect segmentation for a better understanding of the observed scene. The proposed mechanism integrates metric and semantic inference using Dempster-Shafer theory, which makes it possible to handle observations that have different degrees and kinds of uncertainty. The system has been implemented and tested in a real indoor environment, showing the benefits of the proposed approach.

1. Introduction

Recent years have brought a rapid advance in the area of mobile robotics and intelligent systems. At the same time, in many developed countries we observe processes such as population aging and increase of labor costs. For these reasons, since the modern society is already accustomed to the pervasiveness of advanced technology and intelligent devices, in the near future we can expect robots to become an indispensable part of our everyday life. Robots and artificial intelligence agents will be used by nonspecialists and thus will need to understand semantic concepts and instructions, which are natural in the human language. To accomplish this goal, a good understanding of the human environment by computers is an absolute necessity. Even though the field of computer vision has provided many useful tools for object recognition in the last decades, most of the current algorithms used for this task lack any kind of awareness of their own limitations and thus make it difficult to achieve a deeper understanding of the observed world. Besides robotics, the cognitive enhancement of computer vision would be of great value in such fields as automatic surveillance, augmented reality, medical imaging and expert systems, computer graphics, computer-aided geometric design, and industrial quality assurance.

Ignorance and uncertainty are natural, intuitive concepts used constantly in human cognition [1, 2]. The human mind is aware of the fact that most information regarding the surrounding world is not available to it at the moment. Such "knowledge about lack of knowledge" is used for reasoning with imaginative exploration of different possible scenarios and, if necessary, for planning actions that lead to the acquisition of additional relevant information. When processing visual stimulus, this kind of awareness allows us to quickly select the places where an object can be and where it certainly cannot.

Uncertainty also plays an important role in the process of human object recognition [3–5]. For example, we would not classify a lone stick resting against the wall as a broom. However, if it was coming out from behind a bucket, we would classify it as a possible broom, as we are aware that one of the ends is not visible and the item might be a broom. If an object is partially occluded, observed from far away, has a very small size, or the lighting conditions are bad, we are uncertain about it to some degree until we look closer. Such inference mechanisms are obvious and often performed subconsciously in our daily life. Nevertheless, the explicit use of uncertainty is rare in computer vision and robotics.

In our previous work [6] we have proposed an object recognition procedure consisting of 5 elastically designed

stages: features extraction, segmentation, hypothesis formulation, low-level inference, and high-level inference, where the usefulness of semantic relations between objects and part-based reasoning was demonstrated. However, the system was completely unaware of occlusions or any other form of perceptual limitations, which are explored in the present paper.

When data is obtained by an intelligent system in a real environment, we deal with two kinds of uncertainty: stochastic, which results from the fact that the system behaves in a random way, and epistemic, which results from lack of knowledge. In the field of robotics, when dealing with decision-making tasks under uncertain conditions, the classical probability theory is typically applied. However, it is not capable of capturing epistemic uncertainty, which in our opinion is a crucial aspect of scene understanding.

In order to integrate ignorance awareness into a holistic recognition and inference system, we rely on Dempster-Shafer theory (DST) and the generalized Markov random fields proposed in [6]. The possibility of applying DST for the simple task of isolated object classification has been previously suggested in [7, 8]. In order to use DST in general object recognition problems we must deal with several challenges such as ambiguous part and object belief integration, optimization of the belief function for a whole scene consisting of multiple known and unknown objects, and belief aggregation for large numbers of hypotheses which avoids saturation of the DST parameters.

In this paper we propose a novel, unified approach for integrating observations with knowledge about various kinds of perceptual limitations of a 3D object recognition system. To achieve this we apply Dempster-Shafer theory, which allows the explicit handling of evidence with different degrees of uncertainty. In order to improve recognition reliability we integrate this mechanism with the methods of contextual semantic inference presented in [6].

The rest of the paper is organized as follows: in Section 2 we discuss the state of the art. The background of Dempster-Shafer theory is presented in Section 3. Our method of object classification is described in Section 4. In Section 5 experimental results are presented. We draw conclusions in Section 6.

2. State of the Art

The recent development of 3D sensors and computer vision techniques has made it possible to describe the local properties of the closest environment of a mobile robot with reasonable accuracy [9, 10]. Knowledge about the robot environment is usually encoded in the form of a map. Most methods focus on the following two categories.

(i) Metric maps [11, 12] which represent some geometric features of the environment. The environment is represented as a grid of cells or as a feature-based map. This approach has two major problems: the size of the map grows with the size of the environment and the accuracy of the map largely depends on the size of a cell. Feature-based maps are attractive because of their compactness and they are very useful during the process of localization. However, the path-planning based on this kind of representation is time-consuming.

(ii) Topological maps [13] which represent relations between distinctive parts in the environment. This kind of maps has a graph structure, where nodes are used to denote some areas or places in the environment, and edges denote adjacency.

Researchers have been recently focused on topological and mixed (metric-topological) maps that are *semantic*—that is, maps that not only contain data concerning the geometry and relations between parts of the environment but also hold the meaning of various recognized, labeled elements of the observed world [14–16]. The semantic labels attached to the places and objects give information not only about their names but also about functionalities: the doorway indicates the transition between different rooms; that is, a meal can be prepared in the kitchen using ingredients from the fridge and so forth.

Over the last years many interesting object recognition techniques have been developed [14, 17, 18]. Most of these techniques are focused on analyzing various features of the objects present on the captured scene. Even though in many cases they perform well, they do not use any kind of knowledge about the vision system's constraints, which might be regarded as a drawback.

Some works explore the effects of possible occlusions in 2D vision systems for specific tasks such as path following [19–21] and people recognition [22]. The authors of [23] take advantage of the fact that occlusions occur in order to penalize impossible alignments of known objects in the scene. Probably the most advanced use of knowledge about occlusions in 2D computer vision is presented in [24], where the authors propose to create a mask of foreground "occluders," which influences the scores of feature-based object classification on the background and [25], where the authors use reoccurring occlusion patterns learned in the training process.

3. Dempster-Shafer Theory

Reasoning under uncertainty and using inexact knowledge is frequently necessary for real-world problems [20]. Observations, which are the main source of information about the environment of a mobile robot, are uncertain, meaning that some data can be missing, unreliable, or ambiguous. The knowledge representation can be also imprecise, inconsistent, or partial. If multiple inference rules are applicable, problems of contradictions between redundant rules and problems of missing rules are not uncommon. A formalism adequate to deal with real-world information should therefore allow us to express and quantify all these aspects and provide an algorithm of integration of data from multiple sources with different degrees of uncertainty.

In order to reduce the uncertainty usually probability theory is applied—either experimental (i.e., based on the

frequency of events) or subjective (based on expert assessment). The main problem with probability is its difficulty to deal with ignorance—probabilities must be assigned even if no information is available. It requires to go beyond the rules of current interest and consider all possibilities, so it is not capable of capturing epistemic uncertainty. Also, a probability given by a single number does not provide any means to distinguish ignorance and evidence conflicts. A proper measure of ignorance could be very useful to verify whether the available knowledge is sufficient to justify a decision of the system, as it would allow us to consider the additional option of postponing the decision until enough information is gathered. In a similar way, in order to draw conclusions based on various pieces of information, it would be useful to have a quantitative measure of evidence conflict. High degrees of conflict represent situations where decisions should be made more carefully.

Dempster-Shafer theory (DST) [26–28] of evidence is a formalism for uncertain reasoning which fits the mentioned requirements. This theory was designed in order to deal with uncertainty, ignorance, and conflicts. In comparison to probability theory instead of assigning probabilities to events (hypotheses), knowledge is encoded by assigning masses m to subsets of the set T (power set) of all possible events.

Consider

$$m : 2^T \longrightarrow [0, 1]. \tag{1}$$

The masses fulfill the following requirements:

$$\sum_{a \in 2^T} m(a) = 1, \qquad m(\phi) = 0, \tag{2}$$

where ϕ denotes the empty set. A belief measure is given by the function $\text{bel} : 2^T \rightarrow [0, 1]$:

$$\text{bel}(A) = \sum_{B \subseteq A, B \neq \phi} m(B). \tag{3}$$

A plausibility measure is given by the function $\text{pl} : 2^T \rightarrow [0, 1]$:

$$\text{pl}(A) = \sum_{B \cap A \neq \phi} m(B). \tag{4}$$

The complements of *belief* and *plausibility* are called *doubt* and *disbelief,* respectively. The difference $\text{pl}(A) - \text{bel}(A)$ describes *uncertainty*. The process of data aggregation according to DST consists of the following steps.

(i) Degrees of belief for particular hypotheses are obtained based on facts, which are treated as information sources; for example, when a door handle is observed it supports the hypothesis that the robot is observing a door and denies the hypothesis that the robot is observing a fridge.

(ii) Dempster's rule is applied in order to combine degrees of belief and disbelief. Each fact (or source) can support any hypothesis with a belief degree between 0 and 1 and also deny any hypothesis with a disbelief degree (also called the belief degree for the negation of the hypothesis) between 0 and 1. Belief and disbelief in a hypothesis need not to sum to 1.

Dempster's rule of combination for two sources (1 and 2) is described as follows:

$$m_{1,2}(a) = \frac{\sum_{b \cap c = a} m_1(b) \cdot m_2(c)}{1 - \sum_{b \cap c = \emptyset} m_1(b) \cdot m_2(c)}, \tag{5}$$

where $a, b, c \subseteq T$. Based on (5) we obtain the following practical formulas:

$$\begin{aligned} m_{1,2}(h) &= \frac{m_1(h) \cdot m_2(h) + m_1(h) \cdot m_2(h_u) + m_1(h_u) \cdot m_2(h)}{1 - m_1(h) \cdot m_2(h_n) - m_1(h_n) \cdot m_2(h)}, \\ m_{1,2}(h_n) &= \frac{m_1(h_n) \cdot m_2(h_n) + m_1(h_n) \cdot m_2(h_u) + m_1(h_u) \cdot m_2(h_n)}{1 - m_1(h) \cdot m_2(h_n) - m_1(h_n) \cdot m_2(h)}, \\ m_{1,2}(h_u) &= 1 - m_{1,2}(h) - m_{1,2}(h_n), \end{aligned} \tag{6}$$

where h is the hypothesis, h_n is the negation of the hypothesis, h_u is the uncertainty of the hypothesis, and $m(*)$ is the belief (or supporting degree) function. Whereas the values $m(h)$ and $m(h_n)$ arise from evidence supporting or denying the hypothesis h and $m(h_u)$ represents the degree of uncertainty (i.e., lack of evidence altogether).

4. Inference under Uncertainty

As we have mentioned, we make use of our previous system [6]. In the first four, low-level processing stages, we integrate algorithms for extracting the uncertainty and occlusion data and we completely redesign the high-level inference stage. Before we present the details of our new approach, we briefly describe the function of the system stages.

In the first processing stage three geometric surface features are calculated for the whole point cloud. In our application we make use of an approximate, fast implementation of CAT features (convexity, anisotropy of convexity, and *theta* polar angle of the normal vectors). These are a set of intuitive, simple features that can be calculated quickly and used to describe the local properties of a surface. The convexity of a surface can be intuitively understood as a measure of how much the surface is convex nearby a selected point. A negative value of convexity would mean that the surface is concave. Anisotropy of convexity measures how much the convexity value varies depending on the direction for which it is measured. Therefore, the anisotropy of convexity for a spherical surface would be zero and for a cylindrical surface would be positive, though both surfaces have positive convexity. The *theta* angle represents the inclination of the surface relative to the gravity vector (measured with an

accelerometer). Even though this feature cannot always be applied, for most indoor objects, it is very useful.

Even though a variety of multidimensional descriptors have been developed to capture rich local surface properties [29–31], we have chosen the CAT features because of their high usefulness to distinguish simple shapes despite their low dimensionality. Due to the use of the *theta* inclination angle, these features are particularly useful in indoor environments, as they allow us to distinguish vertical and horizontal flat surfaces, which seem identical when analyzed with purely *intrinsic* shape descriptors. The benefits of applying a global z-axis reference have been shown by the authors of [32].

Let us discuss the proposed features in more detail. Consider a spherical coordinate system (R, φ, θ): radial distance R, azimuthal angle φ, and polar angle θ, beginning at a given point P which lies on the surface Π, with the axis $\theta = 0$ being the direction of the normal vector of the surface Π at the given point P. The surface Π is defined by

$$\theta = f(R, \varphi). \tag{7}$$

We define the surface convexity at P for a sphere radius R as

$$C_P(R) = \frac{1}{\pi R} \int_0^{\pi} \Psi(R, \varphi) + \Psi(R, \varphi + \pi)\, \mathrm{d}\varphi, \tag{8}$$

where

$$\Psi(R, \varphi) = \psi(R, \varphi) \cdot \delta(R, \varphi) \tag{9}$$

for $\psi(\varphi, R)$ being the θ angle of the vector $\mathbf{m}$, which is the projection of the vector normal to the Π surface at the point $[R, \varphi, f(R, \varphi)]$ onto the plane containing the axis $\theta = 0$ and the point $[R, \varphi, f(R, \varphi)]$. The δ function equals either 1 or -1. It is negative if and only if the vector $\mathbf{m}$ is pointed toward the line defined by $\theta = 0$.

Anisotropy of convexity is a measure of how much the convexity changes in different directions. For a point P on the surface Π (in a spherical coordinate system identical to the coordinate system used in (7)) anisotropy of convexity is defined for the sphere radius R by

$$\begin{aligned} A_P(R) = {} & \max_{\varphi \in [0,\pi)} \left[\Psi(R, \varphi) + \Psi(R, \varphi + \pi)\right] \\ & - \min_{\varphi \in [0,\pi)} \left[\Psi(R, \varphi) + \Psi(R, \varphi + \pi)\right]. \end{aligned} \tag{10}$$

These two features are both surface intrinsic properties. However, the objects commonly found in indoor environments have a well-defined base or set of possible base surfaces and thus lack degrees of freedom. This quality makes it possible to use the polar angle θ (i.e., inclination of the surface) as a valid feature of great discriminative power. This feature is invariant to rotation around any vertical axis.

After calculation, the CAT features are mapped into a 2D image, where the features, for convenience, are treated as 8-bit "colors" (ranging from 0 to 255)—this is done by assigning the features of each point to the corresponding pixel of the original depth image obtained from the sensor, from which the 3D coordinates were calculated. The features map is next smoothed using the mean-shift filtering algorithm described in [33] (implemented in the OpenCV library) with its spatial window radius set to 10 pixels and "color" window radius set to 20. After this operation, a canny edge detector is run and the resulting contours are connected using a simple morphological closing operation (with a square kernel of 3×3 pixels). The processed contours define a set of segments, which are further processed by rejecting the smallest ones (with an experimentally chosen area threshold of 100 pixels).

The next stage consists of formulating object parts hypotheses for the obtained segments. This is performed by matching feature histograms of the segments (i.e., the normalized histograms of CAT features for the pixels lying inside the segments) to the histograms of features calculated for "model" object parts (the model segments are obtained with the same segmentation algorithm, but using test scenes, where the objects are manually labeled). The histogram comparison is done by means of Pearson's correlation coefficient defined by

$$\mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X - \mu_X)(Y - \mu_Y)\right]}{\sigma_X \sigma_Y}. \tag{11}$$

If the correlation coefficient exceeds the required value ($\mathrm{corr}_{\min}$), an object part hypothesis is formulated. The correlation threshold is set individually for each part of the model object, depending on how unique the expected part features are. Each scene segment can have several hypotheses. A null hypothesis (hypothesis of unknown object) is additionally attached to each segment.

The low-level inference stage consists of iterative Monte Carlo matching of model point clouds provided as part of the system knowledge (described in [6]) to the scene segments whose feature histograms correlate with the known object parts. In this stage simple segments are combined into potential complex objects composed of multiple parts (these are denominated as object hypotheses or metahypotheses) by spatial model alignment. The successful alignment of several scene segments to the model usually does not provide sufficient evidence to terminate the recognition process. There are several reasons for this.

(1) Only successfully segmented patches of the scene are considered—the segmentation is imperfect, so many details are missing.

(2) Due to self and mutual occlusions, only a fraction of the full object is visible on the scene.

(3) The false-positive rate of recognition based on segment alignment is especially high for furniture regarded as a set of flat patches that can be aligned to several models.

The aim of the last processing stage (the high-level inference) is to find the best *scene theory*—the set of metahypotheses that best explains the observations taking into consideration additional, human-provided knowledge. This knowledge concerns the known objects (in our system we use boundaries for altitude above the ground level and parts importance weights) and their semantic relations with other

objects (three kinds of relations are currently used in the system, that is, *above*, *below,* and *beside*). This optimization is done using a DST belief degree (which is explained in this section) as an energy function.

4.1. Parts Present on the Scene. In the proposed system each part of a hypothetical object (whether attached to a particular scene segment or not) is regarded as an evidence source. For the found object parts (associated with an observed segment) the DST parameters are calculated with the equations presented below. For any given hypothesis let us define a few parameters:

$$q = 2 \cdot \frac{\operatorname{corr}(H_S, H_M)}{(1 + \operatorname{corr}_{\min})}. \tag{12}$$

This parameter (q) contains the correlation of the segment feature histogram (H_S) with the model part histogram (H_M) relative to the minimal correlation required to formulate a hypothesis (i.e., the correlation threshold $\operatorname{corr}_{\min}$). q is truncated to fit into the range $[0; 1]$. Next, for any segment hypothesis of index j we define

$$s_j = q_j \cdot \frac{\sum_{i=0}^{N} q_i - q_j}{\sum_{i=0}^{N} q_i}, \tag{13}$$

where $\sum_{i=0}^{N} q_i$ is the sum of the q values for all the candidate hypotheses of the given segment (including j). The s value is equivalent to accumulated plausibility of the alternative part hypotheses of the scene segment. The normalization factor of this parameter is required (as will be shown through the rest of this section) to ensure that hypotheses of object parts that were actually found influence positively the belief in their combined object hypotheses (i.e., $m(h) > m(h_n)$).

Consider

$$u_{\text{area}} = 1 - \frac{|A_M - A_S|}{A_M} \tag{14}$$

This parameter (u_{area}) captures the uncertainty derived from deviations of the hypothetical part's surface area (A_S) compared to the model part's area (A_M). Such deviations are allowed to a certain point due to limited precision of the segmentation algorithm and partial occlusions.

Consider

$$u_{\text{dist}} = 1 - \frac{d}{d_{\max}}, \tag{15}$$

where d is the mean distance from the sensor to the segment points and $d_{\max}$ is the maximum sensor range. This value (u_{dist}) represents the uncertainty which comes from loss of precision and resolution for further objects. It should be noted that the linear precision loss model we used is a simplification and does not describe accurately the Kinect sensor. However, it reflects the optical linear relation of the size of the objects perceived as a 2D depth map to the sensor distance, which directly affects the segmentation algorithm. Both, u_{area} and u_{dist} are truncated to fit in the range $[0, 1]$.

We can now define the DST parameters of h—the hypothesis of presence of an object part—arising from a present (i.e., found) segment as follows:

$$\begin{aligned} m(h_u) &= 1 - u_{\text{area}} \cdot u_{\text{dist}}, \\ m(h) &= q\frac{1 - m(h_u)}{q + s}, \\ m(h_n) &= s\frac{1 - m(h_u)}{q + s} = 1 - m(h) - m(h_u). \end{aligned} \tag{16}$$

Therefore, the hypothesis uncertainty sums its *area* and *distance* components. The mass is based on the available evidence—the similarity of segment features to a model segment (q parameter). The disbelief is based on the accumulated evidence supporting the alternative possibilities (s parameter). As stated before, for found parts the equations are normalized to ensure $m(h) > m(h_n)$.

4.2. Absent Parts. The object parts which have not been found on the scene are also valuable sources of information. Their absence can provide strong evidence against a particular object hypothesis. However, the fact that a part has not been found does not necessarily lead to the conclusion that it is missing. The two main factors that can limit the part's visibility are occlusions and the camera frame boundaries. One of the novel features of the presented system is the ability to detect these factors and quantify them as uncertainty by using a basic form of spatial imagination described in this section.

To calculate the parts visibilities, the model object point cloud concerning a given hypothesis is aligned to the scene point cloud—this is performed using the transformation found during the low-level inference stage (for the reasons mentioned while discussing that stage, the segment-based alignment, is only a guess, not a sufficient recognition method). The model object is then projected into the flat camera coordinate system (knowing the intrinsic camera parameters). The depth of the projected model points is subtracted from the input scene depth at the corresponding points, calculating how many object points that are "closer" to the view point than the observed scene points. An example of such projection is presented in Figure 1.

Let us denote the depth map (2D pixel array) of the scene point cloud as a matrix $\mathbf{K} = [k_{ij}]$. The projection of the transformed model point cloud is a set of N points, consisting of x and y pixel coordinates and depths: $P = \{(x_i, y_i, d_i)\}$. The number of model points which should be visible can be calculated as

$$N_{\text{VMP}} = \sum_{i=0}^{N} T\left(k_{y_i x_i} - d_i - \varepsilon\right), \tag{17}$$

where ε is a small distance margin left in order to compensate alignment imperfections. T is a thresholding function, which assigns 1 to positive and 0 to nonpositive arguments. If some of the model object points fall out of the map boundaries, they are naturally excluded from this sum. The hypothetical

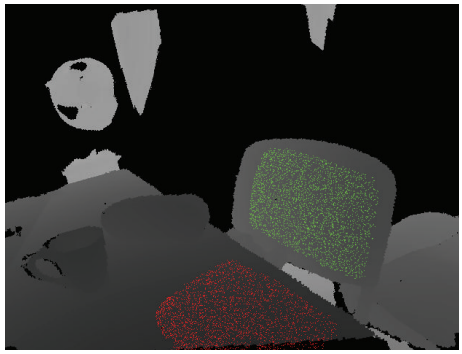

Figure 1: Assumed visible (green) and occluded (red) regions of a hypothetical object (a chair) obtained by spatial reasoning (output on the depth map).

visibility of a given part which has not been found is calculated simply as

$$v = \frac{N_{\mathrm{VMP}}}{N_{\mathrm{TMP}}}, \tag{18}$$

where N_{TMP} is the total number of model part points. We propose to calculate the DST parameters of h—the hypothesis of presence of an object part—arising from the fact that it was not found as

$$\begin{aligned} m(h) &= 0, \\ m(h_u) &= 2(1 - v), \\ m(h_n) &= 1 - m(h_u), \end{aligned} \tag{19}$$

where $m(h_u)$ is truncated to fit into the $[0, 1]$ range. Therefore, the disbelief of the hypothesis is based on the evidence pointing that the space where the missing part should be is empty (i.e., not occluded). The mass of the hypothesis is obviously zero. As we can see, a part whose visibility is less than 0.5 will have $m(h_u) = 1$ and thus no effect on the further calculations. This formula has been chosen in order to compensate for the imprecision of the segmentation algorithm near the borders between overlapping parts—if we expect most of a part to be occluded, it is likely that it has been omitted in the segmentation, so its absence is not regarded as relevant evidence. However, if a part is missing and it should be (based on the alignment results) in an area that is fully visible (i.e., empty area) with $v = 1$, it will strongly affect the hypothesis, as it will have $m(h_n) = 1$, meaning complete disbelief in the object part hypothesis.

4.3. Weights. The particular parts observations are not of equal importance—for example, a patch of flat surface can belong to many objects, whereas an ellipsoidal patch is uncommon and can provide strong evidence for the presence of a particular object such as a lamp. In the presented inference system, such part properties can be human-defined. In order to make it possible to include such kind of knowledge in the inference mechanism, we propose a "weighted" modification of the regular DST rule of combination. This

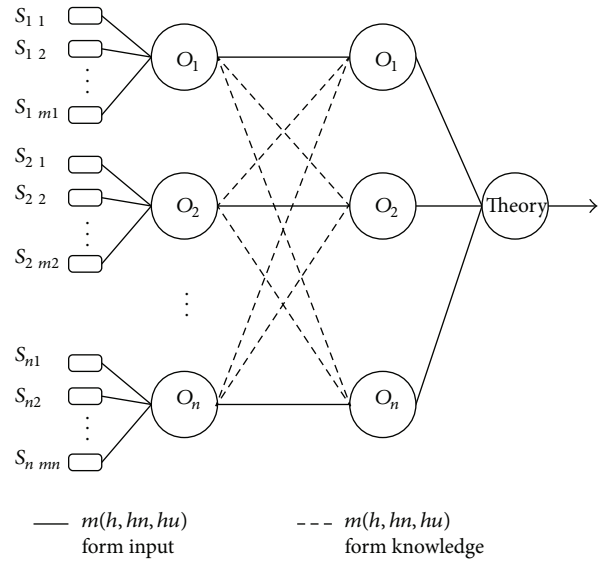

Figure 2: DST theory quality calculation scheme. Data flows from left (object parts DST parameters S_{nm}) to right (DST parameters for the whole scene theory).

modification consists of applying a weight w to the DST parameters of a hypothesis just before combination according to the following equations:

$$\begin{aligned} m_w(h) &= \frac{w \cdot m(h)}{w \cdot m(h) + w \cdot m(h_n) + m(h_u)}, \\ m_w(h_n) &= \frac{w \cdot m(h_n)}{w \cdot m(h) + w \cdot m(h_n) + m(h_u)}, \\ m_w(h_u) &= 1 - m_w(h) - m_w(h_n). \end{aligned} \tag{20}$$

The provided weights can be different for found parts (which we denominate as *positive weights*) and absent parts (which we call *negative weights*). The importance of this feature is best explained giving the example of a door. Detecting the vertical board is weaker evidence for the door presence than detecting the handle, but not detecting the board is stronger evidence against the door hypothesis than not detecting the handle.

4.4. Fusion. The high-level inference mechanism introduced in this paper uses a three-stage Dempster-Shafer fusion algorithm presented in Figure 2. The rectangles represent the DST parameters calculated for the parts of each object included in a scene theory (classification of all the scene segments) for both present and absent parts.

These parameters are integrated in the first stage of the fusion method (the metric stage) for each object in the weighted modification of Dempster's rule of combination described before. The resulting parameters $(m(h), m(h_n), m(h_u))$ for each object hypothesis are passed to the next stage (the semantic stage) and fused with a priori parameters that arise from the detected semantic relations with other objects present in the theory. These a priori parameters, fixed for each defined relation, are part

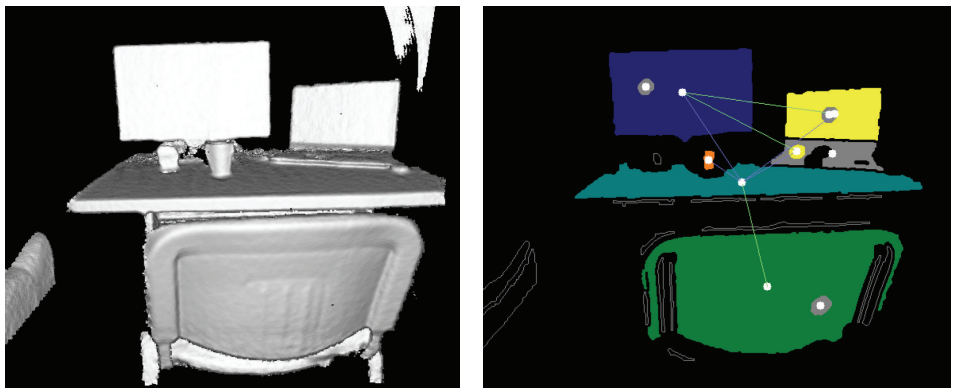

Figure 3: Recognition results for a simple scene (*monitor*—blue, *laptop*—yellow, *cup*—orange, *table*—turquoise, *chair*—green, and *unknown*—gray), including semantic relations: *beside*—green and *above*—blue.

of the human-provided knowledge of the system and can define either positive ($m(h) > 0, m(h_n) = 0$) or negative ($m(h) = 0, m(h_n) > 0$) relations. Obviously, for impossible or very unusual relations (such as *chair above monitor*) providing $m(h_n) = 1$ guarantees that the result of the fusion procedure will also be $m(h_n) = 1$, meaning certain rejection of such theory. Finally, the output parameters calculated for all the objects are combined together, resulting in a single set of DST parameters for the whole theory. Due to the tendency of Dempster's rule to converge to 1 for large quantities of plausible elements, the authors decided to apply a scaling factor for the $m(h)$ and $m(h_n)$ parameters resulting from the second processing stage (this factor has been experimentally set to 0.1). However, the scaling factor is applied only if the $m(h_n)$ parameter of a given object is less than 1, which prevents accepting impossible theories (contradictory to human-provided categorical rules).

Finding the best theory is a nontrivial optimization problem, which in our implementation is solved using a genetic algorithm, which aims to find the scene theory with the highest mass $m(h)$ (calculated according to the presented fusion mechanism) among all possible scene theories. The genetic algorithm initializes a population of theories (each classified scene segment is regarded as a gene), which then evolves for a fixed number of generations. Besides cross-over and mutation, each individual performs local optimization by single-gene substitutions (as described in [6]). The theory mass is used as the algorithm's fitness function. An example scene theory is shown in Figure 3.

4.5. Computational Complexity. In order to find the computational complexity for the worst-case scenario of the presented algorithms, we take into consideration the number of known object parts—N, which is roughly proportional to the number of known objects. The effort to calculate the proposed features and perform segmentation is proportional to number of points (P) present on the scene. Assuming that the scene resolution and size are fixed, for the first two processing stages, we get the order of complexity $O(P) = O(1)$.

The hypothesis formulation and low-level inference algorithms compare and align (for complex objects) each of the M scene segments to each model part. These two stages share the worst-case scenario order of complexity, which is $O(MN)$, as the number of maximum iterations of the Monte Carlo alignment method is fixed.

The last processing stage, as is currently implemented, has the highest worst-case order of complexity (even though in practice it has been observed to require about 1/3 of 4th stage's effort to complete). The used genetic algorithm requires just a $O(M)$-complex fitness function for the evaluation of each individual. However, the most complex part resides in the local optimization performed for each individual. If all the scene segments were assigned all the possible hypotheses and the local optimization went through all the possible scene theories, the order of complexity would be that of a brute-force search—that is, $O(M \cdot M^N) = O(M^{N+1})$. Even though there is no guarantee for a smaller complexity, the genetic algorithm proves much faster than a brute-force search. It would be possible to limit the maximum complexity by simplifying the local optimization to merely recalculating an individual's fitness for each single-gene substitution without further iterations. This would lead to a maximum order of complexity of $O(M^2N)$ but would not necessarily reduce the real amount of computation, as the genetic algorithm would probably need more generations to reach the best theory.

5. Experiments

The system described in the previous section has been implemented and tested using point clouds obtained using the Kinect Fusion algorithm [34] based on input from a Kinect sensor. The Kinect Fusion is a SLAM-based point cloud integration and filtering algorithm which has been used by the authors in order to enhance the input precision and to register reflective (e.g., a monitor) and dark surfaces which are very poorly captured in single Kinect frames (we used the open source version implemented in the Point Cloud Library by Anatoly Baskeheev). Scene views can be obtained with Kinect Fusion in real time, which requires only a few seconds of observation to achieve point cloud qualities significantly

superior to single frame point clouds. Such observations can be easily made by a mobile robot equipped with a Kinect-like sensor and a graphics processing unit (GPU).

Three sets of scenes (mostly related to an office environment) have been registered for the experiment as follows:

(1) a training set of 18 scenes from the office environment, which contain mainly fully visible objects observed from a short distance, used to extract the model point clouds and verify the human-provided parameters of the system (only some of these scenes were used to create the model objects);

(2) a test set of 14 realistic scenes of the office environment with known and unknown objects in different positions, captured from different distances; Occlusions occur but the environment presented in these scenes was relatively tidy and thus the set was denominated as the *easy* test set;

(3) a test set of 12 scenes with numerous, prevailing unknown objects and/or heavy occlusions. Some of these scenes are completely unrelated to the office environment; this set has been denominated as the *hard* test set; an example scene from this set is presented in Figure 4.

The system has been provided with 20 model views and 21 semantic relations, as well as the positive and negative part weights for complex objects. These parameters have been chosen using feedback from the test set only. In the experiment we used 9 known object classes: *chair, clock, cup, lamp, laptop, monitor, mouse, pencil case,* and *table*, of which 5 are treated as complex objects (i.e., composed of more than one part). The recognition performance of the system has been compared using the 26 test scenes for the following versions of the system:

(1) the system without any knowledge about its own perceptual limitations (i.e., limiting its belief on the formulated hypotheses based only on alternative hypotheses strength and provided part weights. However, hypotheses are not formulated for segments whose size is too big to possibly correspond to a given object part),

(2) the system aware of occlusions and limited field of view (i.e., using DST parameters for *absent* object parts as described in the previous section) but unaware of its limited precision,

(3) the system aware of its limited precision (i.e., using DST parameters for *present* object parts as described before) but not aware of occlusions and limited field of view,

(4) the *full* system aware of both kinds of limitations (using the DST parameters for all parts).

The performance was measured using 4 parameters:

(1) correct object recognitions relative to the number of known objects present on the scene;

(2) incorrect positive recognitions relative to the number of known scene objects (whether the incorrect decisions concern a known or unknown object);

(3) the $E1$ total error rate defined as

$$E1 = \frac{\mathrm{FN} + \mathrm{FP}}{\mathrm{NO}}, \tag{21}$$

where FN is the number of unrecognized known objects present on the scene (false negative), FP is the number of incorrect recognitions (false positive), and NO is the number of visible known objects;

(4) the $E2$ weighted error rate, calculated similarly to the $E1$ parameter, but using a weighting factor

$$E2 = \frac{0.2 \cdot \mathrm{FN} + \mathrm{FP}}{\mathrm{NO}}. \tag{22}$$

This error rate takes into consideration the fact that usually the cost of an incorrect positive recognition (false positive) for a mobile robot is higher than the cost of missing a known object (false negative), as the first case is more likely to cause counterproductive or even dangerous actions.

Figure 5 summarizes the experiment results. As we can see, the use of ignorance awareness slightly decreased the average correct recognition rate. However, it dramatically reduced the incorrect recognition rates as well as the error rates for both test sets, thus improving the system reliability. The results suggest that the use of *present* part uncertainty has higher impact on the system performance than the use of *absent* part uncertainty. However, combining both kinds of limitation awareness leads to the best overall results. As expected, the advantage of processing both kinds of ignorance was greater for the *hard* test set. For the *easy* test set, even though the incorrect recognition rate was lowest for the *full* system, the $E1$ total error rate was even slightly higher for the *full* system compared to the version with only *present* part uncertainty. The cause of this is the fewer numbers of unknown objects with misleading similarities to known object parts in this set. However, when taking into consideration the unequal cost of both kinds of errors (i.e., false positive and false negative), the *full* system performed best for this set as well.

The overall, relative reduction of error rates for the *full* system was of 12.5% and 35.9% compared to the version without *absent* part uncertainty: 58.8% and 79.0% compared to the version without *present* part uncertainty and 68.4% and 85.3% compared to the version without any kind of uncertainty for the $E1$ and $E2$ error rates, respectively.

The presented algorithms have been implemented in C++ CPU code, only partially exploiting multicore possibilities (at the low-level inference processing stage). The full processing time for a single scene depends on its complexity and ranges up to one minute.

6. Conclusions

In this work we have proposed a method to obtain and use information about the perceptual limitations of a 3D object

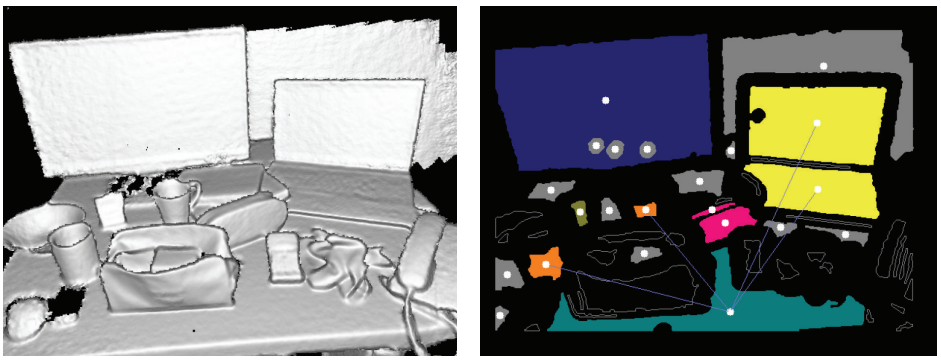

Figure 4: One of the test point clouds enhanced with kinect fusion, along with the recognition results (*monitor*—blue, *laptop*—yellow, *cup*—orange, *table*—turquoise, *clock*—olive, *pencil case*—pink, and *unknown*—gray).

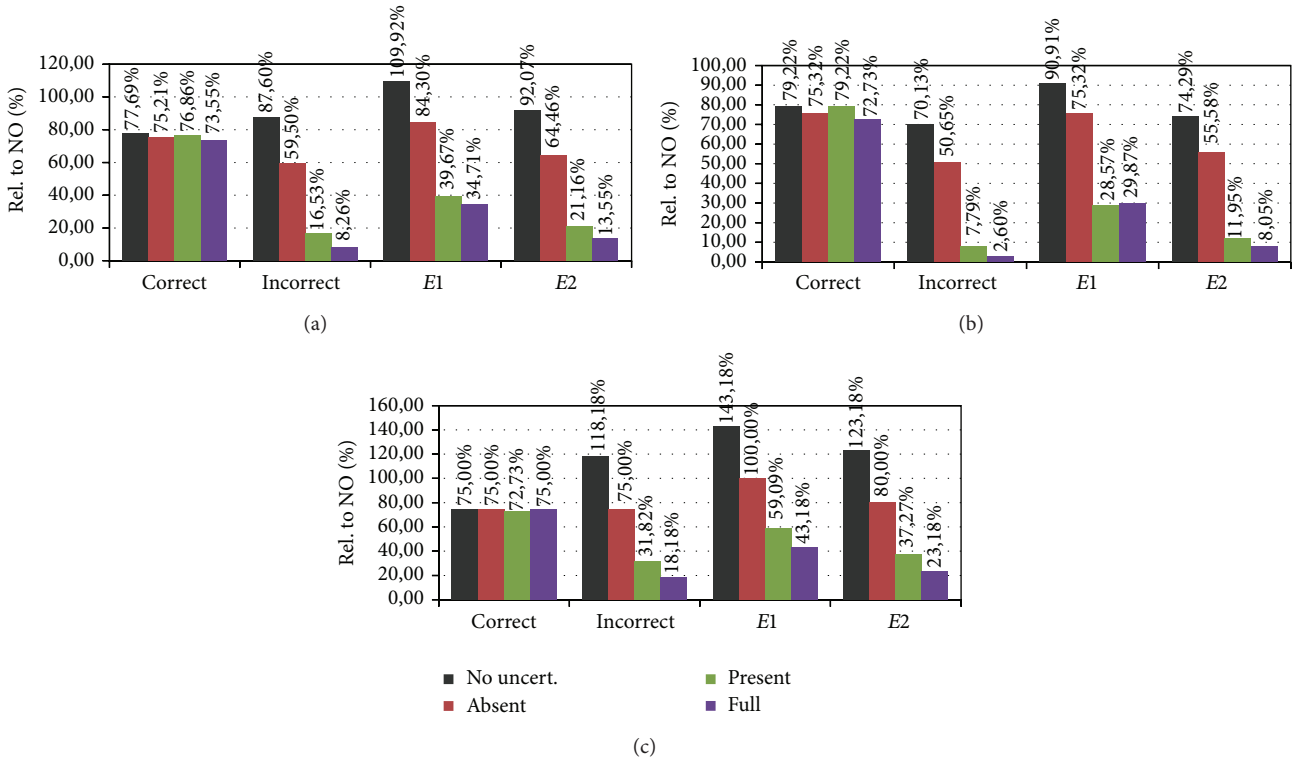

Figure 5: Experiment results statistics for the *combined* (a), *easy* (b) and *hard* (c) test sets. The charts compare the performance parameters (relative to NO—the number of visible known objects) of the system without ignorance awareness (*No uncert.*), with the systems using *absent* part and *present* part ignorance awareness (the first coming from occlusions and field of view and the latter from limited sensor and segmentation precision) as well as the *full* system using both.

recognition system. By applying Dempster-Shafer theory, the uncertainty information has been integrated into a metric-semantic inference system. The sources of uncertainty have been separated into two categories: *absent* object parts, for which the system's belief in their real absence is obtained using a form of occlusion-aware spatial imagination, and *present* object parts, for which the uncertainty is connected to distance-related precision loss and segment size deviations. The impact of both forms of ignorance awareness has been estimated in an experiment involving realistic scenes of an indoor environment with variated difficulty levels. The experiment shows that the use of each kind of uncertainty effectively increases the system reliability by diminishing the false-positive error rates.

The presented inference mechanism is apt to be expanded by adding new kinds of uncertainty, as well as new kinds of evidence concerning the observed scene. In future works we intend to use a wider range of features and focus on real-time performance by exploiting GPU processing power.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been partially supported by the National Science Center (Grant 2011/01/B/ST6/07385).

References

[1] P. G. Armour, "The five orders of ignorance," *Communications of the ACM*, vol. 43, no. 10, pp. 17–20, 2000.

[2] A. Varouxaki, N. H. Freeman, D. Peters, and C. Lewis, "Inference neglect and ignorance denial," *British Journal of Developmental Psychology*, vol. 17, no. 4, pp. 483–499, 1999.

[3] J. T. Enns, *The Thinking Eye, The Seeing Brain: Explorations in Visual Cognition*, W. W. Norton & Company, New York, NY, USA, 2004.

[4] G. W. Humphreys, C. J. Price, and M. J. Riddoch, "From objects to names: a cognitive neuroscience approach," *Psychological Research*, vol. 62, no. 2-3, pp. 118–130, 1999.

[5] G. Humphreys and M. Riddoch, "Chapter 15: the cognitive neuropsychology of object recognition and action," in *Handbook of Cognition*, K. Lamberts and R. L. Goldstone, Eds., Psychology Press, Hove, UK, 2001.

[6] B. Harasymowicz-Boggio and B. Siemitkowska, "Object classification with metric and semantic inference," in *Proceedings of the IEEE 6th European Conference on Mobile Robots (ECMR '13)*, pp. 186–191, 2013.

[7] A. P. Dempster and W. F. Chiu, "Dempster-Shafer models for object recognition and classification," *International Journal of Intelligent Systems*, vol. 21, no. 3, pp. 283–297, 2006.

[8] B. Harasymowicz-Boggio and B. Siemi-atkowska, "Object classification using Dempster-Shafer theory," in *Mechatronics 2013: Recent Technological and Scientific Advances*, pp. 559–565, Springer, 2013.

[9] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden Markov models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 2026–2031, April 2005.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, Mass, USA, 2005.

[11] R. Triebel, B. Frank, J. Meyer, and W. Burgard, "First steps towards a robotic system for exible volumetric mapping of indoor environments," in *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04)*, Lisbon, Portugal, 2004.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, Cambridge, Mass, USA, 2005.

[13] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artificial Intelligence*, vol. 152, no. 1, pp. 47–104, 2004.

[14] B. Siemiatkowska, J. Szklarski, M. Gnatowski, and A. Borkowski, "Towards se-mantic navigation system," in *Recent Advances in Intelligent Information Systems*, M. Klopotek, A. Przepiorkowski, S. Wierzchon, and K. Trojanowski, Eds., pp. 711–720, 2009.

[15] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[16] Ó. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[17] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. I/900–I/903, September 2002.

[18] H. Zender, O. Mozos, P. Jensfelt, M. Geert-Jan, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems (RAS)*, vol. 56, no. 6, pp. 493–502, 2008.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3354–3361, Providence, RI, USA, June 2012.

[20] A. Teichman and S. Thrun, "Practical object recognition in autonomous driving and beyond," in *Proceedings of the IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO '11)*, pp. 35–38, Half-Moon Bay, Calif, USA, October 2011.

[21] T. Goedem, T. Tuytelaars, and L. V. Gool, "Omnidirectional sparse visual path following with occlusion-robust feature tracking," in *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS '05), Conjunction with (ICCV '05)*, 2005, pp. 1806–1811.

[22] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Improved multi-person tracking with active occlusion handling," in *Proceedings of the ICRA Workshop on People Detection and Tracking*, vol. 2, 2009.

[23] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Proceedings of the 10th Asian Conference on Computer Vision*, pp. 135–148, Springer, 2010.

[24] M. Z. Zia, M. Stark, and K. Schindler, "Explicit o cclusion modeling for 3D object class representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3326–3333, June 2013.

[25] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, June 2013.

[26] P. L. Bogler, "Shafer-Dempster reasoning with a pplications to multisensor target identification systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17, no. 6, pp. 968–977, 1987.

[27] B. Chokr and V. Kreinovich, "How far are we from complete knowledge, complexity of knowledge acquisition in the Dempster-Shafer approach," in *Advances in the Dempster-Shafer Theory of Evidence*, pp. 555–576, John Wiley & Sons, 1994.

[28] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, USA, 1976.

[29] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European Conference on Computer Vision: Part III (ECCV '10)*, pp. 356–369, Berlin, Germany, 2010.

[30] T. Federico, S. Samuele, and L. D. Stefano, "Unique shape context for 3d data description," in *Proceedings of the ACM Workshop on 3D Object Retrieval (3DOR '10)*, pp. 57–62, New York, NY, USA, 2010.

[31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA '09), pp. 3212–3217, 2009.

[32] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3d laser range data in urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 4391–4398, 2012.

[33] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[34] S. Izadi, D. Kim, O. Hilliges et al., "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*, pp. 559–568, New York, NY, USA, October 2011.