Kinect based 3D Scene Reconstruction

Niclas Zeller
HS Karlsruhe
Moltkestrasse 30,
76133 Karlsruhe,
Germany
niclas.zeller@hs-karlsruhe.de

Franz Quint
HS Karlsruhe
Moltkestrasse 30,
76133 Karlsruhe,
Germany
franz.quint@hs-karlsruhe.de

Ling Guan
Ryerson University
350 Victoria Street,
Toronto, M5B 2K3,
Canada
lguan@ee.ryerson.ca

ABSTRACT

This paper presents a novel system for 3D scene reconstruction and obstacle detection for visually impaired people, which is based on Microsoft Kinect. From the depth image of Kinect a 3D point cloud is calculated. By using both, the depth image and the point cloud a gradient and RANSAC based plane segmentation algorithm is applied. After the segmentation the planes are combined to objects based on their intersecting edges. For each object a cuboid shaped bounding box is calculated. Based on experiments the accuracy of the presented system is evaluated. The achieved accuracy is in the range of few centimeters and thus sufficient for obstacle detection. Besides, the paper gives an overview about already existing navigation aids for visually impaired people and the presented system is compared to a state of the art system.

Keywords

3D scene reconstruction, electronic travel aid, Microsoft Kinect, obstacle detection, RANSAC plane segmentation, visually impaired

1 INTRODUCTION

For safe traveling in known and unknown environments blind as well as visually impaired people depend on travel assistance devices. Until now the white cane and the guide dog are the most popular ones. Nevertheless, both the white cane and the dog are not able to detect suspended objects, which are hanging at head height in front of the visually impaired person. Besides, the range, especially of a white cane, is limited to one to two meters and a dog often cannot be used indoor. Alternatively, or supportive to these conventional devices, electronic navigation devices can be used. In this paper we are especially interested in so called electronic travel aids (ETAs), which are devices that do not require any additional infrastructure like GPS and man-made landmarks or prior knowledge about the environment. Other systems can be categorized in electronic orientation aids (EOAs) and position locator devices (PLDs). EOAs provide information to reach a certain destination. This can be offline information (e.g. the floor plan of a building) as well as online information (e.g. a continuous tracking of the position within a given map). PLDs are used for positioning (e.g. GPS) and thus often are applied within an EOA.

The main goal of an ETA is to warn a visually impaired person about upcoming obstacles and impart perception of her or his surroundings. Even though there exist plenty of assistance devices, most of them are not well accepted by the community of the blind and visually impaired. Most of the existing ETAs warn the user only about obstacles right in front of her or him but do not give any further perception about the person's environment. Other camera based systems try to impart perception of the scene but tend to overwhelm the user's senses by conveying redundant information. Section 2 gives a short overview about existing systems and research projects.

The idea of this paper is to develop a system which records the blind person's environment and reduces the recording to a minimum amount of significant information. To record the environment Microsoft Kinect is used. The advantages of Kinect compared to other camera systems is its low price as well as the extensive software development kit (SDK). Thus, Kinect gives the ability to develop a low priced prototype system without a long familiarization period, making it ideal for rapid prototyping.

Based on the depth image of Kinect, an algorithm is developed which detects obstacles in a scene and models them by a small number of cuboid objects. The infor-

 ISBN 978-80-86943-71-8

mation needed to describe these cuboids easily can be modulated onto stereo audio signals for example.

Although the algorithm is developed based on the depth image captured by Kinect, it can be applied to any other depth images. Thus, the presented system is ideal to verify the feasibility of the proposed idea and the presented algorithm easily can be combined with other depth camera systems, e.g. a stereo camera system or a plenoptic camera included in glasses.

The presented algorithm first performs a plane segmentation. This segmentation is a combination of a gradient based 2D image segmentation and a random sample consensus (RANSAC) based plane segmentation [Fis81a], which is applied to the 3D point cloud. The plane segmentation will be described in Section 3. The second step of the algorithm is the object modeling and thus the reconstruction of the recorded scene. This approach is based on finding intersecting edges between neighboring plane segments. The object modeling will be presented in Section 4. In Section 5 the algorithm is analyzed based on its accuracy and Section 6 shows the results of the system applied in test environments. Section 7 compares our system to a state of the art system [Rod12a] and Section 8 draws conclusions.

2 STATE OF THE ART

This section gives an overview about already existing ETA systems. Beside some commercially available tools, there exists a variety of research activities focusing on ETAs. In [Dak10a] Dakopoulos and Bourbakis give a good overview about already existing ETAs, as well as research activities in this field. Manduchi and Coughlan [Man12a] present a more general overview about electronic assisting devices for blind people. Most of the commercially available systems rely on distance sensors for detecting obstacles within the surroundings of a blind person. K-Sonar [Zab06a] for example is a tool which uses ultrasonic sensors to perceive obstacles in front of the user. The received information is conveyed to the user by stereo headphones. K-Sonar either can be used as hand held device or can be mounted on a long cane. A similar system as K-Sonar is the Laser Long Cane [Rit01a, Rit02a] from Vistac GmbH [Vis13a]. In this system laser distance sensors are attached to a long cane. The sensors monitor the area in front of the user on upper body height. The system warns the user by vibrations in the sensor device about obstacles.

Beside these commercially available products, there exist various research projects, which are focusing on distance sensor based ETAs. Manduchi and Yuan are developing a system for detecting steps using laser distance sensors [Yua04a, Yua05a]. Dunai et al. work on a system called CASBliP [Dun12a]. This tool is based on a time-of-flight (TOF) line scan camera. The camera scans a horizontal plane in front of the blind user and transforms it into stereo audio signals. One big advantage of this system is the long range of about 15 m.

Other ultrasonic based systems are NavBelt and GuideCane developed by Shoval et al. [Sho03a]. Both systems are based on the same ultrasonic sensors, which scan the surroundings. In the NavBelt system the sensors are attached to a belt, which is carried by the user. GuideCane is a small vehicle at the end of a long cane on which the sensors are arranged. Based on the sensor data, signal processing algorithms calculate a safety path in traveling direction. In the NavBelt system, the user is informed about the safety path by audio signals, while GuideCane steers into the direction of this path. One big disadvantage of GuideCane is that the system is not able to pass or detect steps.

Other ultrasonic systems, like the project of Cardin et al. [Car05a] use tactile feedback to transmit the sensor's information.

More sophisticated than systems based on distance sensors are camera based systems. These systems try to convey perception of the users environment instead of just cautioning against upcoming obstacles. One commercially available system is vOICe [vOI13a]. Here the image of a camera, which is arranged in glasses, is transformed into spatial audio signals. These signals are presented to the user by headphones. Gonzalez-Mora et al. focus on a similar idea. In [Gon06a] a system is described which uses Head Related Transfer Functions (HRTFs) to modulate audio signals with the image information. Even though these systems show promising results, interpreting those signals involves a long lasting training period. Besides, the often very sensitive aural sense of blinds cannot be used for other tasks. Thus, other systems try to reduce the camera data to a small amount of significant information before it is transmitted to the user. Dakopoulos for example describes in his PhD-thesis a prototype system [Dak09a] that uses a binocular stereo camera to receive a depth image of the scene in front of the user. The depth image is reduced to a resolution of $4\,\text{pixel} \times 4\,\text{pixel}$. This depth information is presented to the user by a 4×4 vibration motor array, which is attached to her or his stomach.

Saez Martinez and Escolano Ruiz also work on a stereo camera based system for obstacle detection [Sae08a]. Here, algorithms are used to combine sequences of depth images to a 3D-map. For obstacle detection the user's motion is estimated based on the image sequences and thus obstacles in travel direction are detected. However, the system only warns the user about upcoming obstacles and does not use the image information for scene percipience.

Rodriguez et al. describe another stereo camera based system [Rod12a]. A short description of this approach is given during the comparison in Section 7.

Most of the camera based systems developed so far mainly focus on detecting obstacles with more or less high spatial resolution. Thus, they can prevent the user from being overwhelmed by a huge amount of data. But thereby they also reduce very much the conveyed information. We in contrast are interested in remodeling the recorded scene. This later will give us the opportunity to classify the remodeled objects. For example by classifying a number of steps as a stairway, which otherwise would be considered as an insuperable obstacle. Thus, we can provide to the user a high amount of information with limited amount of data.

3 DEPTH IMAGE PLANE SEGMENTATION

A pixel in the depth image is defined by its 2D image coordinates x_I and y_I and will be denoted by the vector $\mathbf{X}_I = (x_I, y_I)^T$ in this paper. Each pixel contains a depth value d, which represents the distance to the corresponding object point. In the following the depth value for a pixel $\mathbf{X}_I$ will be denoted by $d(\mathbf{X}_I)$ or $d(x_I, y_I)$.

The 3D world coordinate system is defined based on the orientation and position of Kinect. A point in the world coordinate system is defined by the coordinates x_W, y_W, and z_W and will be denoted by the vector $\mathbf{X}_W = (x_W, y_W, z_W)^T$. All three coordinates (x_W, y_W, and z_W) have the unit centimeter.

3.1 Gradient based depth image segmentation

The gradient based depth image segmentation is used to perform a rough preprocessing, which reduces false segmentation during the RANSAC algorithm. Besides, the gradient based algorithm is very sensitive in detecting small steps, which could be disregarded by the RANSAC algorithm. To perform the algorithm, out of the depth image $d(\mathbf{X}_I)$, its gradient vector $\vec{g}(\mathbf{X}_I)$ is calculated. The gradient vector $\vec{g}(\mathbf{X}_I)$ can be calculated based on any common gradient filter (e.g. Sobel or Canny operator). For the experiments presented in Sections 5 and 6 the gradient vector $\vec{g}(\mathbf{X}_I)$ is defined as given in eq. (1). This gradient definition is very sensitive on small edges and steps since no low pass filtering is applied to the depth image. Compared to the Sobel and Canny operator the computation time of the filter in eq. (1) is very low. Besides, the calculated gradient does not have to be very accurate since after the gradient based segmentation the RANSAC algorithm performs an accurate plane segmentation.

$$\vec{g}(\mathbf{X}_I) = \begin{pmatrix} d(x_I+1, y_I) - d(x_I, y_I) \\ d(x_I, y_I+1) - d(x_I, y_I) \end{pmatrix} \tag{1}$$

After calculating the gradient vector $\vec{g}(\mathbf{X}_I)$, it is filtered by a median filter. The median filter reduces the number of outlying values caused by interpolation artifacts and quantization errors.

For segmentation each pixel is compared with each of its four neighboring pixels. Two neighboring pixels $\mathbf{X}_I^i$ and $\mathbf{X}_I^j$ are assigned to the same segment if the following two conditions are satisfied:

1. The norm of the difference vector Δg between $\vec{g}(\mathbf{X}_I^i)$ and $\vec{g}(\mathbf{X}_I^j)$ has to be below a threshold T_g.

$$\Delta g = \left\| \vec{g}\left(\mathbf{X}_I^i\right) - \vec{g}\left(\mathbf{X}_I^j\right) \right\| \tag{2}$$

2. The difference Δd between the real depth value $d(\mathbf{X}_I^j)$ and the estimated depth value $\hat{d}(\mathbf{X}_I^j)$ at the position $\mathbf{X}_I^j$ has to be below a threshold T_d.

$$\Delta d = \left| d\left(\mathbf{X}_I^j\right) - \hat{d}\left(\mathbf{X}_I^j\right) \right| \tag{3}$$

$$\hat{d}\left(\mathbf{X}_I^j\right) = d\left(\mathbf{X}_I^i\right) + \vec{g}\left(\mathbf{X}_I^i\right)^T \cdot \begin{pmatrix} x_I^j - x_I^i \\ y_I^j - y_I^i \end{pmatrix} \tag{4}$$

3.2 Depth image to point cloud transformation

As already mentioned above, the RANSAC algorithm will be performed based on a set of 3D points in world coordinates. Thus, each pixel in the depth image $\mathbf{X}_I$ has to be transformed into a 3D point in world coordinates $\mathbf{X}_W$. This is done based on the transformation matrix $\mathbf{A}$, which is a 4×4 matrix defined in eq. (5). The transformation defined by $\mathbf{A}$ is a combination of a rotation, a translation and the central projection performed by the camera. To describe the non-linear central projection by a system of linear equations the homogeneous component k has to be introduced as a fourth dimension. For the given definition it is considered that the unit vector of the depth component $\vec{e}_d$ is orthogonal to the image plane but independent of the homogenous component k as given in eq. (5).

$$\begin{pmatrix} k \cdot x_I \\ k \cdot y_I \\ d \\ k \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} \tag{5}$$

From the system of equations given in eq. (5) the equations (6) and (7) are received by inserting the fourth row into the first and second and rearranging them afterwards. Eq. (8) is received as the third row in eq. (5).

 ISBN 978-80-86943-71-8

Since all three equations are linear in the coefficients of **A** and image as well as world coordinates can be measured by calibration points, all 15 coefficients of **A** can be estimated by linear regression.

$$x_I = a_{11} \cdot x_W + a_{12} \cdot y_W + a_{13} \cdot z_W + a_{14} - a_{41} \cdot x_W x_I - a_{42} \cdot y_W x_I - a_{43} \cdot z_W x_I \quad (6)$$

$$y_I = a_{21} \cdot x_W + a_{22} \cdot y_W + a_{23} \cdot z_W + a_{24} - a_{41} \cdot x_W y_I - a_{42} \cdot y_W y_I - a_{43} \cdot z_W y_I \quad (7)$$

$$d = a_{31} \cdot x_W + a_{32} \cdot y_W + a_{33} \cdot z_W + a_{34} \quad (8)$$

Equivalently to the image points $\mathbf{X}_I$ in the 2D segmentation the 3D points $\mathbf{X}_W$ can be aligned in segments.

3.3 RANSAC plane segmentation

After all pixels $\mathbf{X}_I$ are projected into 3D points $\mathbf{X}_W$, to the 3D points of each segment the RANSAC algorithm [Fis81a] is applied.

RANSAC is an iterative method to robustly estimate a certain model from a number of measurements. In our application RANSAC is used to fit planes to the 3D point cloud $\mathbf{X}_W$.

In each iteration step the algorithm randomly picks three sample points ($\mathbf{X}_W^1$, $\mathbf{X}_W^2$, and $\mathbf{X}_W^3$) out of the set of input points and defines a plane Π fitting these three points. For each point $\mathbf{X}_W$ in the set the distance d_Π to the plane Π is calculated. It is checked whether the distance d_Π is underneath a threshold T_{RAN} or not. Points with a distance d_Π smaller than T_{RAN} are considered to be part of the estimated plane. All other points are considered to be outliers. This procedure is repeated N_{trials} times to find the best fitting plane. Best fit is defined in the sense of lowest number of outliers.

The number N_{trials} of iteration steps which are needed to reach convergence can be calculated as follows: If we consider that an iteration step results in n_{IN} inliers by a total number of n_{PTS} points in the input set, the probability that all three independently picked samples ($\mathbf{X}_W^1$, $\mathbf{X}_W^2$, and $\mathbf{X}_W^3$) are inliers can be estimated as given in eq. (9).

$$P(\text{all 3 samples are inliers}) \approx \left(\frac{n_{IN}}{n_{PTS}} \right)^3 \quad (9)$$

From eq. (9) the probability that at least one of the three samples is an outlier is given in eq. (10).

$$P(\text{at least 1 sample is an outlier}) = 1 - P(\text{all 3 samples are inliers}) \quad (10)$$

We define the constraint that with a certain probability, which is denoted by p in eq. (11), there must be at least one trial without any sample being an outlier. With this constraint the number of trials needed can be calculated as follows.

$$N_{trails} = \left\lceil \frac{\log(1-p)}{\log(P(\text{at least 1 sample is an outlier}))} \right\rceil \quad (11)$$

For the experiments presented in Sections 5 and 6 the probability, that at least one trial occurs where none of the three samples is an outlier, was set to $p = 0.99$.

The number of inliers n_{IN} in eq. (9) does not result from the best fitting plane Π_n but from a randomly chosen plane Π. Thus the minimum needed number of trails N_{trails} is always overestimated except for the case that the plane Π is already the best fitting plane.

The RANSAC algorithm results in an estimated plane Π_n, a set of inliers, which builds a new segment and a set of outliers. To the set of outliers again the RANSAC algorithm is applied until the number of outliers is underneath a defined minimum segment size.

4 OBJECT MODELING

Based on the plane segments, geometric objects are modeled. This is done by combining planes to objects. At the current state of the algorithm only cuboid objects are modeled. Most scenarios where people move are dominated by man-made objects, which can be described more or less by cuboids. In further development steps it will be considered to include also different geometric shapes (e.g. cylinders or spheres). However, for obstacle warning, cuboid objects are sufficient. The object modeling is divided in several steps. In the first step intersecting edges between neighboring planes are calculated. Then the floor plane is detected and extracted. In the third step the cuboid objects are modeled out of the plane segments and the intersecting edges.

4.1 Plane intersection

Since, by definition, a plane has infinite extent, there always exists an intersecting edge between any two planes which are not parallel to each other. The challenge in the intersecting edge retrieval is to consider only those edges as existent which do exist in the recorded scene.

To solve this problem a neighborhood graph is calculated. Even though all plane segments are defined already in 3D world coordinates, the neighborhood graph is built based on image coordinates. This is because the computational effort for finding neighbors in a 2D pixel grid is enormously reduced compared to the case of a 3D point cloud. Due to the known transformation between image and world coordinates, after establishing the neighborhoods the graph easily can be translated to the 3D points.

In this graph each segment (plane) defines a vertex. All vertices, for which the segments i and j are direct neighbors, are connected by an edge e_{ij}. Out of the marginal 3D points between two segments a straight line $\hat{L}_c^{ij}$ is estimated by linear regression. Besides, the real intersecting edge between the planes Π_i and Π_j L_c^{ij} is calculated analytically. Within the range of adjacent points between both planes the maximum distance between the estimated and the analytically calculated edge is determined. If the maximum distance between both edges lies below a defined threshold, the edge is considered to exist.

All retrieved intersecting edges are defined as directed straight lines. This means the direction vector $\vec{d}_{ij}$ of L_c^{ij} has to be defined with a certain direction. By definition the segment i is on the left and the segment j on the right side of the projection of L_c^{ij} onto the depth image plane in pointing direction.

After defining intersecting edges there will be pixels that are assigned to the wrong segment. This mean pixels of the segment i which are on the right side of L_c^{ij} and pixels of j which are on the left side of L_c^{ij}. These pixels are assigned to the respectively other segment and the 3D points $\mathbf{X}_W$ are projected onto the corresponding plane.

4.2 Floor plane extraction

Before objects can be built, the floor plane has to be extracted. This is done based on two parameters. The plane orientation, which is defined by a reference normal vector $\vec{n}_{ref}$ and the distance between the floor plane and the optical camera center $\vec{c}$, which is defined by d_{ref}.

To classify a plane as floor plane the angle between the plane normal vector and the reference $\Delta\phi$ has to be underneath a threshold T_ϕ and the distance to the camera center Δd has to be lower than a threshold T_{dist}.

All plane segments, which are lying underneath the floor plane, are erased. This of course does not conform all scenes but simplifies the processing. Planes underneath the floor plane result, for instance, from stairs going downwards. In future development these planes as well as gaps within the floor plane will be considered because detecting downward stairs and gaps in the floor is an absolute must for a reliable system.

4.3 Object reconstruction

To build objects out of the plane segments each intersecting edge L_c^{ij} is classified to be either a convex or a concave edge from Kinect's perspective.

To classify the intersecting edges the normal vectors $\vec{n}$ of the plane segments have to point to the direction of the optical camera center $\vec{c}$. This can be realized since

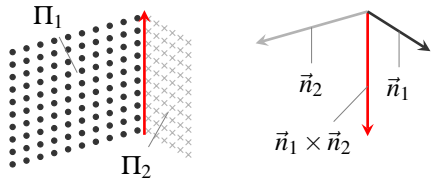


Figure 1: Classification of a concave crossing edges. (a) Two planes Π_1 and Π_2 connected by a concave intersecting edge L_c^{12} (red line). (b) Cross product of the normal vectors $\vec{n}_1$ and $\vec{n}_2$.

$-\vec{n}$ and $\vec{n}$ define the same plane and thus normal vectors, which are pointing away from the optical center, just can be flipped. Since the intersecting edge L_c^{ij} between the planes Π_i and Π_j has a certain direction, such that Π_i is left and Π_j is right of the edge in pointing direction, this property is used to classify the edges. If the cross product of $\vec{n}_i$ and $\vec{n}_j$ points to the same direction as the direction vector $\vec{d}_{ij}$ of L_c^{ij}, the edge is classified as convex. Otherwise the edge is classified as concave (see Fig. 1).

$$L_c^{ij} \widehat{=} \begin{cases} \text{convex} & \text{if} \quad \frac{\vec{n}_i}{\|\vec{n}_i\|} \times \frac{\vec{n}_j}{\|\vec{n}_j\|} = \frac{\vec{d}_{ij}}{\|\vec{d}_{ij}\|} \\ \text{concave} & \text{else.} \end{cases} \tag{12}$$

After classifying all edges into convex or concave objects are built. All planes connected by a convex edge are combined to one object.

For each object a cuboid shaped bounding box is calculated. To avoid underestimation of the object dimensions the boundaries are chosen such that all 3D points assigned to the object are included within the cuboid bounding box. This mostly results in an overestimation of the object size but nevertheless potential obstacles never will be neglected.

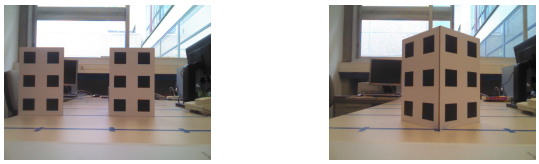
5 ACCURACY ANALYSIS

In this section two different experiments are shown, which evaluate the accuracy of the described algorithms applied to images gathered with Microsoft Kinect.

5.1 Plane accuracy

In the first experiment planar panels (size $19\,\text{cm} \times 29\,\text{cm}$) are placed parallel to the depth image plane of Kinect. Each panel is placed on a defined position within the world coordinate system (see Fig. 2(a)). The scene is recorded by the depth camera of Kinect and the algorithm is applied. Each panel results in a rectangular object in the reconstructed scene, defined by its four corner points. Based on the position of the reconstructed object, the accuracy of the system can be measured. In this experiment only the z-component of an object is evaluated since

 ISBN 978-80-86943-71-8



(a) (b)

Figure 2: Setups for accuracy analysis.

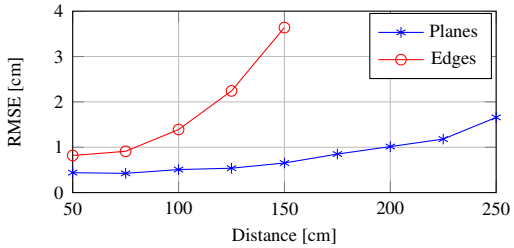


Figure 3: Accuracy analysis. Blue asterisks: RMSE of retrieved planes. Red circles: RMSE of retrieved intersecting edges.

the algorithm will overestimate the boundaries in the directions x_W and y_W as described in Section 4.3.

Fig. 3 presents the root mean square error (RMSE) of the retrieved planes for different distances to Kinect (blue asterisks). For each corner of a plane the error e_i of the z-component between retrieved and given value is calculated, resulting in four error values for each panel. Four panels were recoded in each distance resulting in a total number of $N = 16$ error values per distance. Out of these error values the RMSE is calculated as given in eq. (13).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \tag{13}$$

As one can see, the RMSE rises approximately quadratically with the distance. Nevertheless, even in a distance of 2.5 m still an RMSE of 1.7 cm is reached.

5.2 Intersecting edge accuracy

In the second experiment the panels are arranged as shown in Fig. 2(b). The intersecting edge between the two panels lies on a defined position in the world coordinate system. The scene is recorded by Kinect and the algorithm is applied. The algorithm calculates an intersecting edge between the two planes resulting from the panels. The error between the real and the calculated edge e_i is defined as the distance between both edges at the two marginal points of the edge. Thus, for each recorded edge two error values are received. For each distance six objects were recorded, resulting in a total number of $N = 12$ error values. For this setup the RMSE is also calculated as given in eq. (13).

Fig. 3 shows the RMSE for different distances to the Kinect sensor (red circles). As one can see, only distances below 1.5 m are evaluated. The reason for that is the threshold of the RANSAC algorithm, which is increased linearly, dependent on the depth value d. With rising depth value d, the range of the quantization steps also rises and thus the accuracy decays. To avoid false segmentation resulting in many small planes, the segmentation thresholds (T_d, T_g, and T_{RAN}) are adjusted. Thus, the two panels will be segmented as one consecutive plane at far distances. Nevertheless, for close distances up to about 1.5 m experiments show very accurate results.

6 APPLICATION

In this section the results of the algorithm for two different scenes are presented to get a qualitative assessment of the system.

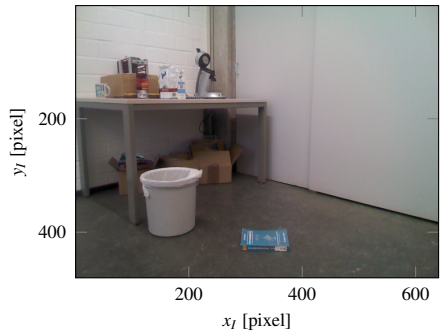
Fig. 4(a) shows the RGB image of the first recorded scene. The scene includes several items, which are supposed to be detected and modeled as objects. The table in the back of the scene represents a large obstacle, which should be modeled as one object. Also the trash bin and the book lying on the floor are supposed to be detected. These two objects represent dangerous obstacles a person might trip on. Fig. 5(a) shows the RGB image of the second scene. In this scene a stairway is recoded. The main goal for this scene is to model the single steps as objects. Additionally, in both scenes the floor plane must be classified to reconstruct the scene correctly.

Fig. 4(b) shows the output of the algorithm corresponding to the scene in Fig. 4(a) and Fig. 5(b) the one corresponding to Fig. 5(a), respectively. In both figures an object is represented by its bounding box in 3D world coordinates. Besides, the floor plane as well as the boundaries of the field of view are plotted in the figure. As one can see, in both scenes the floor plane was detected correctly.

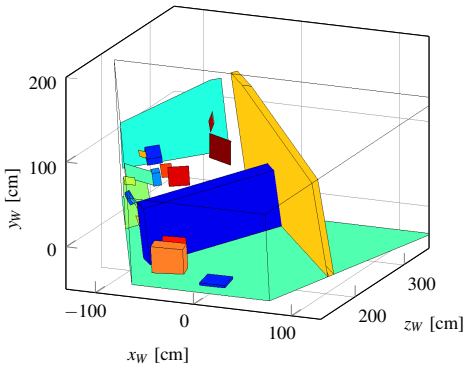
In the first scene (Fig. 4) the objects of main interests, the book, the trash bin and the table, were detected and modeled as objects by the algorithm. Nevertheless, the result of the scene reconstruction is not perfect. For example the left edge of the table is separated into several small object. This comes from the high quantization of the depth information d for large values. The quantization causes planes standing in a steep angle to the image plane to result in a stepped gradient, instead of a homogenous depth gradient. Thus, the single steps are segmented into single planes by the algorithm instead of one large plane. The reconstructed scene also includes wrongly modeled objects for the items standing on the table. Nevertheless, those items are not of major interest for the task of obstacle detection.

In Fig. 5(b) one can see that this scene was reconstructed very well, too. The lowest five steps of the

 ISBN 978-80-86943-71-8

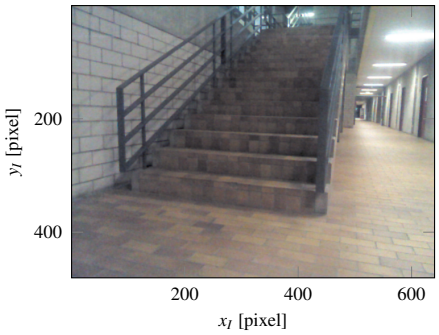


(a) Kinect RGB image

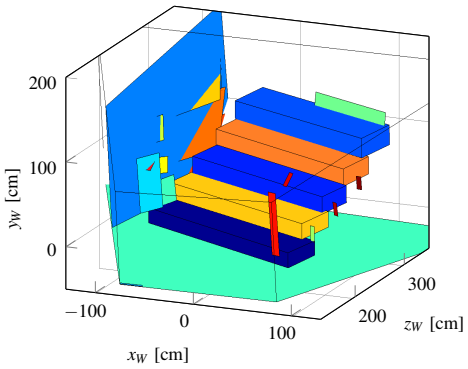


(b) 3D scene reconstruction

Figure 4: Result of the obstacle detection system for test scene 1.



(a) Kinect RGB image



(b) 3D scene reconstruction

Figure 5: Result of the obstacle detection system for test scene 2.

stairway are modeled as separate object. All further steps are out of range for the depth image system. The banisters are not modeled very well since they have very small and reflective surfaces. The left sidewall also results in several objects since it is separated by the left banister. This problem has to be minded in the further development process.

Both scenes show good results of the algorithm. However, there are scenarios the system cannot handle properly, especially when there exist lots of undefined areas within the depth image. These undefined areas result when items shadow each other but also at the presents of direct solar irradiation [Ell12a]. Thus, Kinect is inappropriate for outdoor scenarios.

7 COMPARISON TO A STATE OF THE ART SYSTEM

In this Section the algorithm presented in this paper is compared to the system presented by Rodriguez et al. in 2012 [Rod12a]. This comparison shows the novel contribution of our approach to the research field of ETAs. [Rod12a] is quite comparable to our approach since it also focuses on simplifying the recorded 3D point cloud. Besides, the paper contains a very extensive experimental part with feedback from visually impaired persons, who were testing their system. Based on this feedback we can foresee the demand on our system and on what we have to focus in further developments.

The system presented in [Rod12a] is a stereo camera based system. From the stereo images a disparity map and thereby a 3D point cloud is calculated. By using RANSAC for plane fitting, the floor plane is extracted from the point cloud. All 3D points, which are not part of the floor plane, are divided into bins. This is done by projecting all points on the floor plane and defining a polar grid, which forms the bin margins. Each bin, which contains a sufficient high amount of points, is considers as obstacle. The system already has an audio feedback included, which is based on bone conduction technology and thus, does not block the user's ears. Even though [Rod12a] is not based on Microsoft Kinect, it is quite comparable to the approach presented here. Our approach easily can be adapted to a stereo camera system just as [Rod12a] can be adapted to Microsoft Kinect.

At the current state of development, there are two big advantages of the system presented by Rodriguez et al. compared to our approach. Firstly, it is already running in real time and secondly it has already audio feedback included.

In terms of conveying information about the user's surrounding our system is much more precise than the one in [Rod12a]. While Rodriguez et al. have 12 rigid po-

 ISBN 978-80-86943-71-8

sitions in front of the user where objects can occur, in our approach objects are placed independently from any grid in the 3D space. The necessity of placing the obstacles independently from any grid is confirmed by the feedback of testing persons in [Rod12a]. The visually impaired, who where testing the system demanded for more resolution in the depth domain since they were not able to estimate the relevance of an obstacle.

Another advantage of our approach compared to [Rod12a] and basically all other existing systems is that our system tries to preserve the approximate shape of a recorded object. Thus, in further development stairs as well as other objects can be classified based on their characteristic shape. Nevertheless, still an appropriate interface to the user has to be developed.

8 CONCLUSIONS

The overview of state of the art systems in Section 2 shows that until now there is no operating ETA which satisfies all demands of blind people. Besides, the feedback of visually impaired people documented in [Rod12a] states a demand of high sophisticated ETAs. This demand is very encouraging for our system to be developed further.

The accuracy analysis presented in this paper shows that the developed system works accurately in the range of a few meters in front of the user. In addition the two applications presented in Section 6 show the capability of the system. Scenes can be reconstructed in detail by primitive cuboid objects and even small objects can be detected.

In further development our system has to be miniaturized and has to be combined with a feedback system. Besides, experiments with testing persons have to be organized.

9 REFERENCES

[Car05a] Cardin S., Thalmann, D. and Vexo F. Wearable obstacle detection system for visually impaired people. VR Workshop on Haptic and Tactile Perception of Deformable, pp.50-55, 2005.

[Dak09a] Dakopoulos D. TYFLOS: a wearable navigation prototype for blind & visually impaired; design, modelling and experimental results. PhD-Thesis, Wright State University, 2009.

[Dak10a] Dakopoulos D. and Bourbakis N.G. Wearable obstacle avoidance electronic travel aids for blind: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.40, no.1, pp.25-35, January 2010.

[Dun12a] Dunai L., Garcia B.D., Lengua I. and Peris-Fajarnes G. 3D CMOS sensor based acoustic object detection and navigation system for blind people. Annual Conference on IEEE Industrial Electronics Society, IECON, vol.38, pp.4208-4215, 2012.

[Ell12a] El-laithy R.A., Huang J. and Yeh M. Study on the use of Microsoft Kinect for robotics applications. Position Location and Navigation Symposium (PLANS), IEEE/ION, pp.1280-1288, April 2012.

[Fis81a] Fischler M.A. and Bolles R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol.24, no.6, pp.381-395, July 1981.

[Gon06a] Gonzalez-Mora J.L., Rodriguez-Hernandez A.F., Burunat E., Martin F. and Castellano, M.A. Seeing the world by hearing: Virtual Acoustic Space (VAS) a new space perception system for blind people. Information and Communication Technologies (ICTTA '06), vol.1, pp.837-842, 2006.

[Man12a] Manduchi R. and Coughlan J. (Computer) vision without sight. Communications of the ACM, vol.55, no.1, pp.96-104, 2012.

[Rit01a] Ritz M., Koenig L. and Woeste L. Orientation aid for the blind and the visually disabled. US Patent US 6298010 B1, October 2001.

[Rit02a] Ritz M., Koenig L. and Woeste L. Orientation aid for the blind and the visually disabled. US Patent US 6489605 B1, December 2002.

[Rod12a] Rodriguez A., Yebes J.J., Alcantarilla P.F., Bergasa L.M., AlmazÃ¡n J. and Cela A. Assisting the visually impaired: obstacle detection and warning system by acoustic feedback. Sensors, vol.12, no.12, pp17476-17496, 2012.

[Sae08a] Saez Martinez J.M. and Escolano Ruiz F. Stereo-based aerial obstacle detection for the visually impaired. Workshop on Computer Vision Applications for the Visually Impaired, 2008.

[Sho03a] Shoval S., Ulrich I. and Borenstein J. NavBelt and the Guide-Cane - Robotic-based obstacle-avoidance systems for the blind and visually impaired. IEEE Robotics & Automation Magazine, vol.10, no.1, pp.9-20, 2003.

[Vis13a] Vistac GmbH. Retrieved February 2014. from http://www.vistac.de.

[vOI13a] vOICe. Augmented Reality for the Totally Blind. 1996. Retrieved February 2014 from http://www.seeingwithsound.com/.

[Yua04a] Yuan D. and Manduchi R. A Tool for Range Sensing and Environment Discovery for the Blind. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04), pp.39-47, 2004.

[Yua05a] Yuan D. and Manduchi R. Dynamic envi-

ronment exploration using a virtual white cane. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), pp.243-249, 2005.

[Zab06a] Zabonne Ltd. K-Sonar. 2006. Retrieved February 2014 from http://zabonne.co.nz/.

 ISBN 978-80-86943-71-8