periodica polytechnica

Electrical Engineering
and Computer Science
57/1 (2013) 9–17
doi: 10.3311/PPee.2075
http://periodicapolytechnica.org/ee
Creative Commons Attribution

RESEARCH ARTICLE

Corner Detection and Classification of Simple Objects in Low-Depth Resolution Range Images

Viktor Kovács / *Gábor* Tevesz

Received 2013-03-31, revised 2013-06-24, accepted 2013-06-24

Abstract

This paper deals with corner detection of simple geometric objects in quantized range images. Low depth resolution and noise introduce challenges in edge and corner detection. Corner detection and classification is based on layer by layer depth data extraction and morphologic operations. Appearance based heuristics are applied to identify different corner types defined in this paper. Both computer generated and captured range images are dealt with. Synthetic range images have arbitrary range resolution while captured images are based on the sensor used. Real world data is collected using a structured light based sensor to provide dense range map.

Keywords

Range image · Corner detection · Feature extraction · Thinning

Acknowledgement

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) organized by VIKING Zrt. Balatonfüred.

This work was partially supported by the Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund (grant no.: KMR 12-1-2012-0441).

Viktor Kovács

Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Magyar tudósok krt. 2., H-1117 Budapest, Hungary
e-mail: kovacsv@aut.bme.hu

Gábor Tevesz

Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Magyar tudósok krt. 2., H-1117 Budapest, Hungary
e-mail: tevesz@aut.bme.hu

1 Introduction

Range images encode distance or depth (Z) values in each pixel instead of conventional intensity values. These images are widely used to represent spatial 3D information. Compared to other 3D model representation such as triangle meshes it has both benefits and disadvantages. Range images offer simple, ordered scan-line based data access compared to vertices and triangle indices. Range images represent data seen from a viewpoint, obstruction is always present thus data is incomplete and viewpoint dependent. Multiple range images may be used to reconstruct triangle meshes and eliminate viewpoint dependent details. Some of our examples are based on a synthetic range image consisting of simple geometric objects which may be seen in Figure 1.

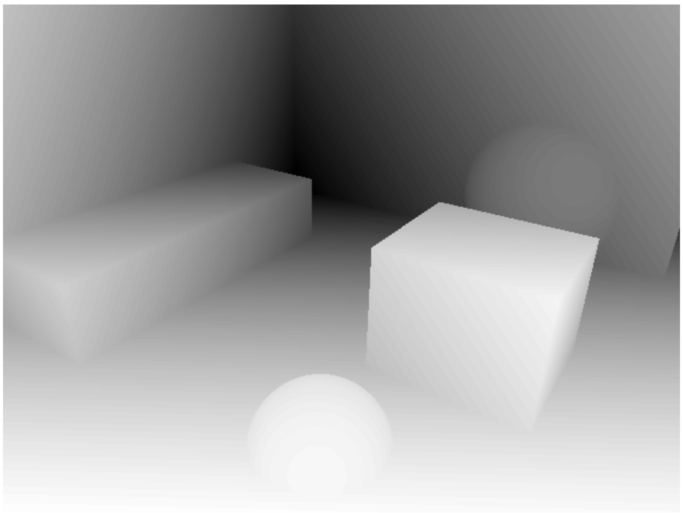

Fig. 1. Synthetic range image

A large number of range image sensors such as time-of-flight ranging devices, offer high depth resolution images together with low spatial resolution. Other implementations sacrifice depth resolution while improving spatial details. In earlier sensors such as the Odetics camera the resolution was 180x180 pixels and 127 steps in depth. Other, still widely used short range sensors provide 176x144 pixels with ± 1mm depth accuracy. When using a single ray to scan the whole image time constraints show up as bottlenecks. Other solutions incorporating a time of flight pixel array and controlled light signals offer faster readouts enabling to have even 320x240 range sensing pixels.

In case of calibrated stereoscopic images, range maps may be generated from the disparity maps using triangulation. Unfortunately it is rather difficult to exactly identify disparity in homogeneous regions thus the quality and denseness of the generated map depends on image content. However conventional stereo intensity image pairs can easily have very high resolutions. To overcome the problem of sparse disparity maps known structured light based arrangements are used. The basic idea is very similar to stereo vision, however instead of two cameras one camera and a previously known pattern is projected which behaves as a second camera. Advantages are easy matching of corresponding pixels thus dense disparity maps can be achieved even on homogeneous surfaces. Unfortunately surface properties and environment lighting may affect the results. In this paper we utilized the output of a structured light based depth sensor. The output image of the sensor has the resolution of 640x480 pixels and 12 bits of depth. Depth values are interpreted in millimeters, however not all possible values are returned, the number of different output levels are much lower. The valid depth range is approximately between 0.8-4.5m. Using intensity image sensors we are able to produce dense range maps however the quantization of the range data may still be high. In this paper we examine the possibilities to utilize low depth resolution range maps to estimate interest points. The corners of simple geometric objects serve as feature points. These points may be used later as landmark points for mapping and localization of a mobile robot in its environment.

The following part of the paper presents related work in the field. We present a related feature detection algorithm and other works and issues concerning low depth resolution images. The third section introduces our new corner detector algorithm. The following section presents results of the algorithm applied to both synthetic computer generated and both captured range images. The test images contain simple geometric objects. The captured images suffer from many artifacts such as heavy quantization and noise. The fifth section contains the discussion of the results and plans for future work.

2 Related Work

Feature detectors are used to find a set of local distinctive, unique points. One of the first and still widely used interest point detector in 2D images is the Harris corner detector [1]. It detects "corners" in the images, meaning significant changes are present in both directions around the point. A Harris score is calculated at each pixel defining a cornerness value by estimating a function of the two eigenvalues of the structure tensor. Two more recent feature detector algorithms SIFT (Scale Invariant Feature Transform) [2] and SURF [3] (Speeded-Up Robust Features) define both scale invariant interest point detection and descriptor generation. Descriptors are high dimensional vectors that are used to give identities to features thus improving matching between images to find similar or exactly the same regions. Our long-term goal is to give a robust feature detector algorithm that finds interest points in range images suffering from low depth resolution, inaccurate depth estimation.

Many aspects of range images are similar to intensity images but there are also differences. Each pixel in the depth image exactly defines a surface point in 3D. Scale invariance is therefore not an issue as in intensity images. However sampling and distance between pixels is irregular. While a homogeneous surface appears as constant values in intensity images, constant normal vectors define planes in depth images. Due to irregular sampling gradient values must be corrected.

Object edges appear in several ways in range images. Jump/step edges are discontinuities in range values. They are simple to detect and typically found at object boundaries. Crease edges appear where the local surface normal changes, for example where surfaces meet. A special case is roof edge where the gradients change signs. Smooth edges have continuity in normal changes but discontinuity in surface curvatures. Range image processing usually does not deal with smooth edges. In structured light based range image detection shadows may appear because of occlusions in the path of the projected light. These shadows have unknown depth values, which is usually indicated by special depth values. Edges separating known and unknown depth values are false edges and must be omitted.

There are several methods for range image edge detection and segmentation. Coleman et. al. [4] presented a gradient based method to characterize edge types. The procedure deals with the irregular grid sampling but assumes smooth gradient values along the image plane which is not given in our case. Bose et. al. [5] presents several methods used in range image segmentation. Both edge and region-based segmentation techniques are used. Estimating the principal curvatures for surface points are often used for the characterization of local geometry however it is difficult to estimate at high quantization errors. The main drawback of edge based techniques is that edges (besides step edges) are distorted in quantized range images thus making it difficult to estimate surface properties near any edges. Region based techniques are based on clustering similar and neighboring surface pixels to regions. However as quantized range images appear as flat surfaces perpendicular to the camera orientation having step edges between layers making it difficult to differentiate regions. Surface curvatures are also preferred because of their excellent invariant properties. Unfortunately the estimation of the curvatures mean the estimation of second order derivatives thus making it extremely sensitive to quantization noise. Usually simple (i.e. paraboloid) surfaces are fitted first.

Ideses et. al. [6] showed the redundancy of depth data in 3D visualization. They analyzed the human depth perception using stereoscopic images generated using depth images with different quantization. They showed that low depth resolution does not affect significantly the perceived 3D image. Their results present that only 20 depth levels are enough to generate 3D impression.

Having less quantization steps does not degrade the perceived visual 3D quality as much.

Viewpoint invariant patches [7] are extracted from intensity images with known 3D geometry. It is possible to create viewpoint normalized image patches from these images and apply 2D feature detector and descriptor generation algorithms to extract features from the viewpoint invariant patches. As the proposed algorithm utilized a 2D feature detector, planes had to be identified in the local 3D geometry.

A recently published 3D point cloud interest point detector is called normal aligned radial feature detector (NARF) [8]. NARF was designed to select points where the surface is stable and changes exist in the near vicinity while making explicit use of unique object boundaries in LIDAR images. The algorithm consists of the following steps: *a*, heuristics are used to find the typical 3D distance to neighboring points that are not across borders, *b*, use these distances to calculate a borderness score for the points, *c*, classify the border, *d* perform non-maximum suppression to find the exact border position. The paper also describers a descriptor generation algorithm: *a*, calculate small range image patch around the point with the observer looking along the normal, *b*, overlay a star pattern onto this patch, where adjacent endpoints correspond to a value in the final descriptor, and captures how much the pixels between the endpoints change, *c* extract a unique orientation from the descriptor, *d* finally rotate the descriptor according to the unique orientation value to make it invariant to rotation.

The algorithm explicitly takes object borders into account. Time of flight based ranging devices acquire stable border information. Stereo vision may approximate borders well if intensity information may be utilized to extract borders explicitly. Based on the utilized patterns, structured light based methods suffer from inaccurate border estimations. The paper describes that the most significant indicator is the distance between neighboring points. Unfortunately these distances show irregularities in case of quantized range values. Step edges break pattern continuities, crease edges are difficult to localize due to low depth resolution. In out algorithm we took these deficiencies into account.

3 Depth layer based range image processing

In this section we introduce a method to process coarsely quantized range images. Individual layers are extracted first as binary maps. After simple preprocessing steps a thinning algorithm is used to create skeletons of the layers. These skeletons are further processed and based on several predefined conditions we attempt to localize and classify corners of simple geometric objects.

3.1 Range layer extraction

Considering a computer generated (and quantized) ideal range image with several simple objects it is simple to detect jump edges but not obvious to find crease edges.

In case of quantized range images jump/step edge detection remains similar to conventional edge detection in intensity image processing. In case the local gradient is larger than a threshold, step edge is assumed. However crease edges also appear as discontinuities in pixel values but these must be rejected otherwise detection would lead to finding edges at each quantization level border. Segmentation is also rather difficult as most pixels in the image suggest planes perpendicular to the camera direction. Figure 2 shows a the synthetic generated image. The edges in conventional 2D image processing manner are highlighted. Both true step edges and false edges appear where neighboring layers meet. The false edges are the result of the coarse quantization.

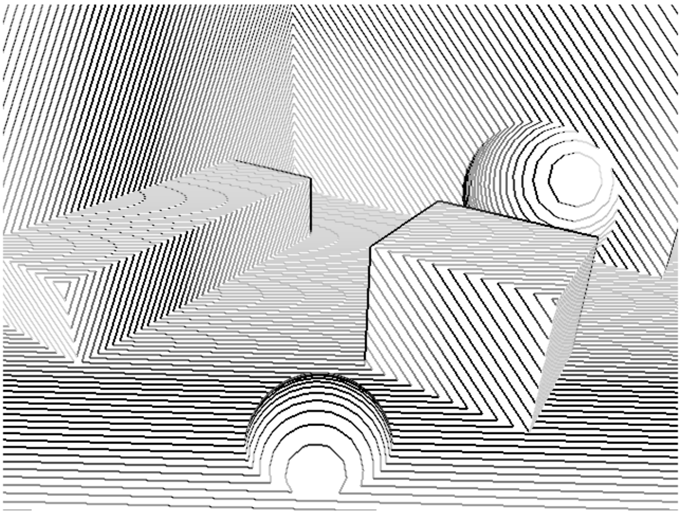

Fig. 2. The gradient operator highlights both true step edges and layer borders

Crease edge points appear where the quantization layer image contains a breakage, the layer's edges change in direction Figure 3 . In other words where corners appear in the layer. Usually it is not obvious which corner points belong to the same crease edge. Based on image complexity and noise some assumptions can be made: crease edge corner points appear along the edge on each range layer thus in case the edge is along a line the crease corner points also appear along the projection of the edge. In low complexity images where crease edge points are sparse, points close to each other on neighboring range levels may be connected to reconstruct edges. In case the image contains numerous crease edges that are line segments, Hough-transformation may be used to fit lines along the crease edge points.

Our range image processing algorithm consists of the following steps: first, skeletons are extracted from each range layer. These skeletons provide isodepth lines. Next, these lines are preprocessed to remove unwanted side branches and identify breakage points. Finally, we utilize predefined condition sets to classify interest points in the images as corner points.

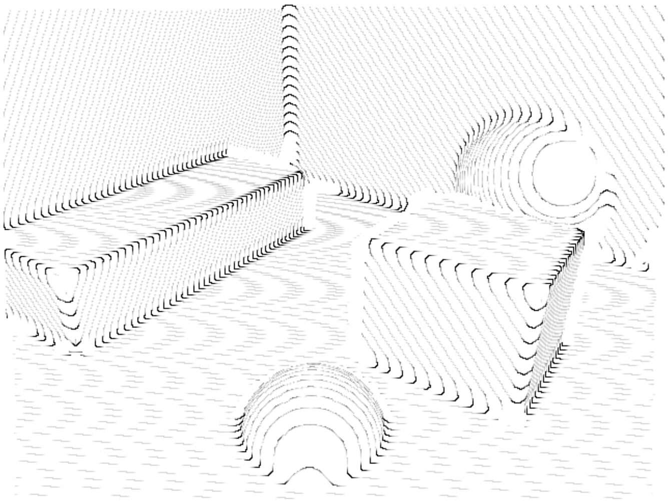

Fig. 3. Direction changes in layers highlighted

3.2 Layer skeleton formation

Based on the pixel values we extract each range layer as binary bitmaps and continue preprocessing steps individually on each layer. In synthetic computer generated images it would be possible to extract only border points of the layers. Unfortunately due to noise and steep surfaces where range layers are narrow and discontinuous it would give undesired results. Instead of utilizing simple border points binary image morphological operators are used.

First dilation operator is applied to each range layer image. Dilation is a basic morphological operation on binary images. The purpose of dilation is to increase the number of foreground pixels in the binary image around its borders. We utilize dilation as a form of noise reduction at the edge the range layers. Although fine details at the borders shall be discarded but the basic topology remains similar in the filtered image. We evaluated different structuring elements, kernels for dilation: 2n-by-2n size square, an n-radius circle and also the application of a simple 3-by-3 matrix as a structural element n times. The best results were achieved using the n-radius circle as a structuring element. The radius depends on the noise in the range image. The dilation algorithm is defined in Algorithm 1.

2D binary images are denoted as a quadruple $P = (\mathbb{Z}^2, 8, 4, B)$. $\mathbb{Z}^2$ identifies all possible points having integer coordinates on the 2D plane, (8, 4) denotes 8- and 4-adjacency for foreground and background points, B is a set of foreground points. Foreground pixels are also called black, background points may be called white points.

Algorithm 1 Pseudo code of the dilation algorithm

```
Input: picture $(\mathbb{Z}^2, 8, 4, X)$ , radius $r$
Output: picture $(\mathbb{Z}^2, 8, 4, Y)$
for all $p \in X$ do
    $N = \{q | dist(p,q) \leq r\}$
    $Y = Y \cup N$
end for
```

The next step of the binary range image layer processing is skeleton extraction. Skeletons are shape descriptors of objects in binary images. Usually they consist of continuous one pixel wide curves which carry information about the original topology of the object. There are numerous algorithms for skeleton extraction. Our algorithm is based on the SI-Q-Isthmus from Németh and Palágyi [9]. The resulting skeletons produced by this fully parallel algorithm is 4-adjacent continuous and provides few sidebranches. Depending on the range image contents most of the sidebranches are false skeleton parts in range images. In a following step of the algorithm we will attempt to eliminate more unwanted sidebranches.

3.3 Skeleton segmentation

The algorithm next creates a graph of segments for each skeleton part (Algorithm 2). Each segment has two endpoints and does not contain any junctions. A segment $s = (E_1, E_2, P)$ is described by the two endpoints (E_1, E_2) and the set of inner points (P). Segments are connected at endpoints. One range layer produces a set of segments ($S = (\{s_i\}, \{E_i\})$) and endpoints ($E = (x, y, \{s_i = (E_1, E_2, P) | E_1 = E \vee E_2 = E\})$) that connect the segments. The result of the segmentation can be seen in Figure 4. Endpoints may be categorized based on the number of segments they connect $|\{s_i\}| =$

- = 1: Simple endpoint, no other segments are connected
- = 2: Non-real endpoint, two segments are connected which may be combined
- ≥ 3: Junction point

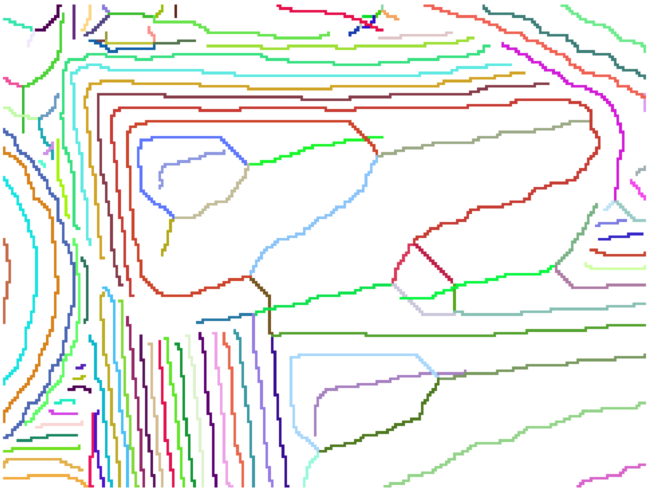

Fig. 4. Skeleton segments connect endpoints and junctions. Each segment is colored individually.

$N_8^*(p)$ ($p \in B$) denotes the 8-adjacent neighbors of point p: $N_8^*(p) = (\{q | 0 < dist(p,q) \leq \sqrt(2)\} \cap B)$. $n(p) = |N_8^*(p)|$ denotes the number of 8-neighbors of $p \in B$. Thinning algorithms often introduce unwanted sidebranches. In our application short sidebranches should be eliminated. Equation (1) gives a condition which segments should be removed. After the unwanted

Algorithm 2 Pseudo code of the skeleton segmenter algorithm

```
$Input$: skeleton $X = (\{p_i\})$
$Output$: segments $S = (\{s_i\}, \{E_i\})$
$S = \emptyset$
$P_{processed} = \emptyset$
for all $p \in (X \backslash P_{processed}) | n(p) = 1$ do
    $P = \emptyset$
    start new segment $(s_i)$, set $E_1 = p$
    $q = p$
    continue following neighbors of p: $q$ = next neighbor of $q$
    $P_{processed} = P_{processed} \cup q$
    if $n(q) = 3$ then
        Junction reached
        $E_2 = q$
        End of segment
        $S = S \cup (E_1, E_2, P)$
        Start new segments from $q$
    end if
    if $n(q) = 2$ then
        Intermediate point, continue
        $P = P \cup q$
    end if
    if $n(q) = 1$ then
        Endpoint reached
        $E_2 = q$
        End of segment
        $S = S \cup (E_1, E_2, P)$
    end if
end for
for all $p \in (X \backslash P_{processed}) | n(p) = 2$ do
    only circles left
    set first point $E_1 = p$
    continue adding points $q$ until $q \neq E_1$
    $P = P \cup q$
    set endpoint $E_2 = E_1$
    $S = S \cup (E_1, E_2, P)$
end for
```

branches are removed, connections must be reevaluated as removing a sidebranch may result in creating circles, linear segment paths etc (Algorithm 3). Results are shown in Figure 5.

$$\begin{aligned} s = (E_1, E_2, P) \in S | dist(E_1, E_2) > d_{min} \wedge \\ \wedge((n_s(E_1) = 1 \wedge n_s(E_2) \geq 3) \vee \\ \vee(n_s(E_2) = 1 \wedge n_s(E_1) \geq 3)) \end{aligned} \quad (1)$$

$n_s(E)$ denotes the number of segments connected to an endpoint: $n_s(E) = |\{s = (E_1, E_2, P) \in S | E_1 = E \vee E_2 = E\}|$.

$Connected(E, i)$ denotes the i^{th} connected segment to E, $Ep_1(s)$ is the first endpoint of segment s and $Circle(s)$ function is defined in Equation (3). The number of circle and non circle segments are connected to endpoint e if given by $n_c(e)$ and $\overline{n_c(e)}$ in Equation (2). The algorithm searches for endpoints of segments which connect to exactly two segments which of none is circle. Until such endpoints exist, the two segments are combined.

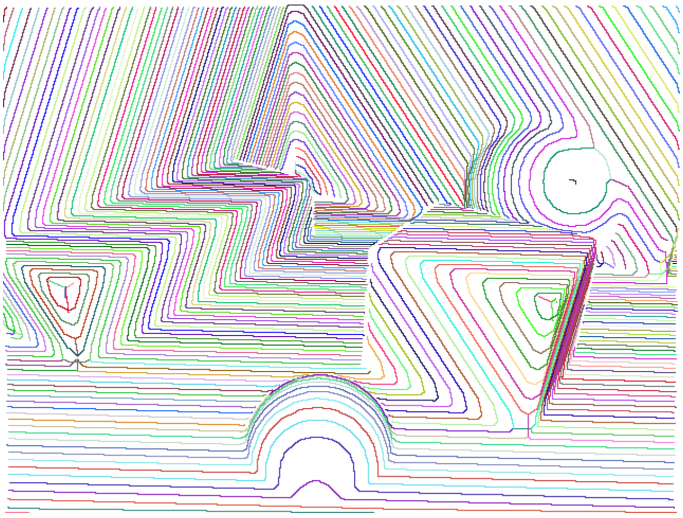

Fig. 5. Skeleton extraction reduces noise and creates iso-depth lines. Unwanted sidebranches are removed.

$$\begin{aligned} n_c(e) &= \sum_{i=1}^{n_s(e)} Circle(Connected(e, i)) \\ \overline{n_c(e)} &= \sum_{i=1}^{n_s(e)} 1 - Circle(Connected(e, i)) \end{aligned} \quad (2)$$

As an addition to the range layer an additional layer is created. Equation (4) describes the points added to this layer, where R is a range image: $z = R(u, v)$. This layer describes step edges and the same operations are performed as seen with range layers. The resulting points are near step edges on the surface closer to the viewer ensuring that the skeleton of the step edge layer is always on the closer side of the edges (on the foreground objects). Now all the skeleton layers are produced and shall be used afterwards.

$$Circle(s) = \left\{ \begin{array}{l} 1 | E_1 = E_2 \\ 0 | E_1 \neq E_2 \end{array} \right\} \quad (3)$$

$$\begin{aligned} (u, v) | \max(R(u + du, v + dv) - R(u, v))_{du,dv \in C} > step_{min} \\ C = \sqrt{du^2 + dv^2} < r_{max} \end{aligned} \quad (4)$$

3.4 Skeleton segment corner detection

The next step involves the localization of corners or direction changes in the skeletons. Different methods are available for this task such as line local fitting on both directions from a specified skeleton point or SUSAN-like corner detection, such as laying a circular mask around the center point and measure the longest or shortest arc that does not contain pixels from the examined skeleton. Both methods offer noise

Algorithm 3 Pseudo code of segment combiner algorithm

```
$Input$: segments $S = (\{s_i\}, \{E_i\})$
$Output$: segments $S = (\{s_i\}, \{E_i\})$

while $\left\{e \in \{E_i\} | n_c(e) = 0 \wedge \overline{n_c(e)} = 2\right\} \neq \emptyset$ do
    $s_A = Connected(e, 1) = (E_{1A}, E_{2A}, P_A)$
    $s_B = Connected(e, 2) = (E_{1B}, E_{2B}, P_B)$
    if $E_{1A} = e$ then
        $s_A = (E_{2A}, E_{1A}, P_A)$ {Reverse $E_1$ and $E_2$}
    end if
    if $E_{2B} = e$ then
        $s_B = (E_{2B}, E_{1B}, P_B)$ {Reverse $E_1$ and $E_2$}
    end if
    $E_{ij} = (u_{ij}, v_{ij}, S_{ij})$
    $P_A = P_A \cup P_B$
    $E_{2A} = E_{2B}$
    if $S_{1A}$ contains $s_B$ then
        $S_{1A} = S_{1A} \backslash s_B$
        $S_{1A} = S_{1A} \cup s_A$
    end if
    if $S_{2A}$ contains $s_B$ then
        $S_{2A} = S_{2A} \backslash s_B$
        $S_{2A} = S_{2A} \cup s_A$
    end if
    $S = S \backslash s_B$
end while
```

reduction by fitting more points or evaluating multiple circular masks with different radii. Both methods yielded good results. The fitting based algorithm must be implemented as total least squares fitting instead of simple least squares as noise is attributed to both x and y coordinates not only y. We create cornerness values from the angle between the two fitted lines (Equation (5)). To eliminate multiple detected points in a small region non-maximal suppression is used (Algorithm 4). Notice that a small amount of noise was added to the values previously to avoid having multiple equal maximal values. To evaluate the suppressed cornerness values, non-suppressed values are calculated in a specified radius around the examined point. In case the examined point returned the maximal value in the region the value is preserved otherwise set to 0.

$$c = |\pi - \max(|\alpha_i - \beta_i|)_i| + \epsilon * noise \tag{5}$$

$N_4^*(p, P)$ $(p \in P)$ denotes the 4-adjacent neighbors of point p: $N_4^*(p, P) = (\{q | 0 < dist(p, q) \leq 1\} \cap P)$.

3.5 Object corner classification

During the interpretation of the skeleton images for corner detection we use the following assumptions:

- Skeletons have corners or change direction near roof edges.
- Jump edges are near simple segment endpoints (endpoints not connecting other segments).

Algorithm 4 Pseudo code of non-maximal suppression algorithm

```
$Input$: center point $p$, segment $s(E_1, E_2, P)$, radius $r$
$Output$: non-maximal suppressed value $v$
$P_{processed} = p$
$p_l = p, p_r = p$
$v = Evaluate(p)$
$i = 0$
while $p_l \neq \emptyset \wedge p_r \neq \emptyset \wedge i \leq r$ do
    $p_l = first(N_4^*(pl, P \backslash P_{processed})$
    $P_{processed} = P_{processed} \cup p_l$
    $p_r = first(N_4^*(pr, P \backslash P_{processed})$
    $P_{processed} = P_{processed} \cup p_r$
    if $Evaluate(p_l) > v \vee Evaluate(p_r) > v$ then
        return 0
    end if
    $i = i + 1$
end while
return $v$
```

We use the previously extracted data such as segments, endpoints, skeleton corner points in the conditions of different types of corner occurrences. Figure 6 and Figure 7 present the typical corner types.

- *Type A*: these corners usually appear or disappear as detached patches in the range layers.

  ○ If circle and number of points in the segment is $n_{segment} < n_{CircleMax}$.

  ○ If linear and number of points in the segment is $n_{segment} < n_{LinearMax}$ and both endpoints are simple endpoints.

  ○ Pixels around the segment are only closer or further.

  ○ No step edge is inside a radius from the segment center.

  ○ Range layer skeleton segment corner points exist: $dist(p, p_{center}) < r_{th} \wedge |Z(p) - Z(p_{center}| < Z_{th}$. This condition is needed to exclude sphere-like objects showing up as "islands" in the range image layers

  ○ Corner point is the center of mass of the segment.

- *Type B*: these corners appear when an edge runs into a smooth surface.

  ○ Both endpoints have more than one connected segments or circular segments $(n_s(E_{1,2}) > 1 \vee Circle(Connected(E_{1,2}, 1) = 1)$.

  ○ Number of points in the segment is less than a threshold ($n_{segment} < n_{ConnectingMax}$).

  ○ Corner point is the center of mass of the segment.

- *Type C*: these corners appear as direction changes in the step edge layer skeleton segments.

  ○ Corners are where segment points have a cornerness value larger than a threshold. Non-maximal suppression is applied.

- *Type D*: these corners appear where a crease edge point is near a step edge. This suggests the two edges intersecting resulting in a corner. Due to the special view angle the step edge does not produce a significant change in direction.

◦ Function (6) is evaluated at each step edge point and non-maximal suppression with thresholding is applied. Crease edge points are taken into consideration only if the depth difference is less than a threshold. This results in points on the step edge that have crease edge points closest.

$$\begin{aligned} d_{sc}(p_{step}) &= \max(dist(p_{step}, p_{crease})^{-1})| \\ &\left|R(p_{step}) - R(p_{crease})\right| < Z_{th} \qquad (6) \\ \text{where } p_{crease} &\in \{\forall \text{crease edge points}\} \end{aligned}$$

- *Type E*: these corners appear where a step edge breaks runs into a surface connecting to a crease edge. Due to the skeletonization and depth difference thresholding when identifying step edges, the endpoints of step edges are misplaced.

  ◦ Range layer segment breakpoints and endpoints are identified around step edge skeleton endpoints. The furthest breakpoint is found which is closer than the step edge segment endpoint. A range layer segment endpoint is identified which is closest to the layer that is n_{offset} quantization steps further than the breakpoint. n_{offset} is an arbitrary number based on the quantization levels, step edge threshold, etc. This point is identifies the corner point.

- *Type F*: these corners are (perhaps distorted) cone type corners. The detection of these corners are omitted in this paper and are subject of further research. Sphere-like objects must be omitted as these objects do not contain any real corner points.

Points that are close to the image borders should be omitted.

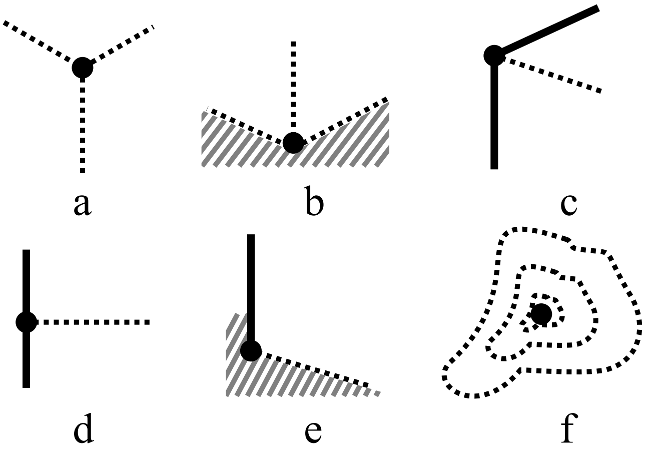

Fig. 6. Corner types: bold lines indicate jump edges, dashed lines indicate crease edges.

4 Evaluation

In this section we evaluate the performance of the algorithm. Both synthetic and real world captured data are used. Data was captured based on a structured light based sensor.

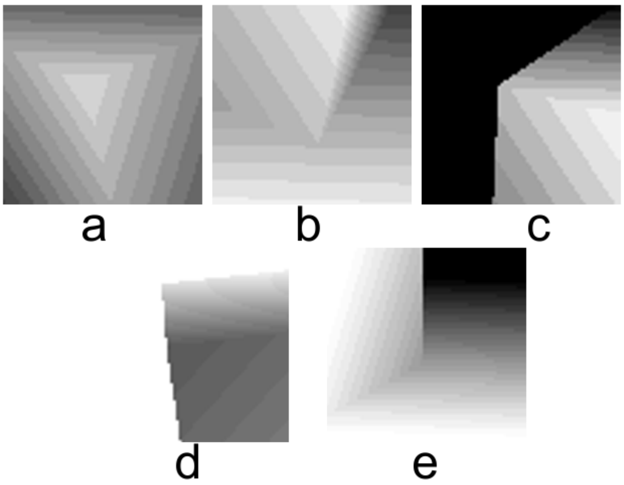

Fig. 7. Corner types: examples from range images.

4.1 Synthetic range image

The previously described algorithm was applied to test images. Results from the example image (with 93 range levels) is shown in Figure 8. Out of 11 real corners all were found. Additional 6 points were detected as corners. From these 6 points 4 were due to the sphere objects in the image. Spheres do not contain corners, they were added to the test image for evaluation purposes. Some points were detected multiple times. A post processing step could be introduced to eliminate such cases.

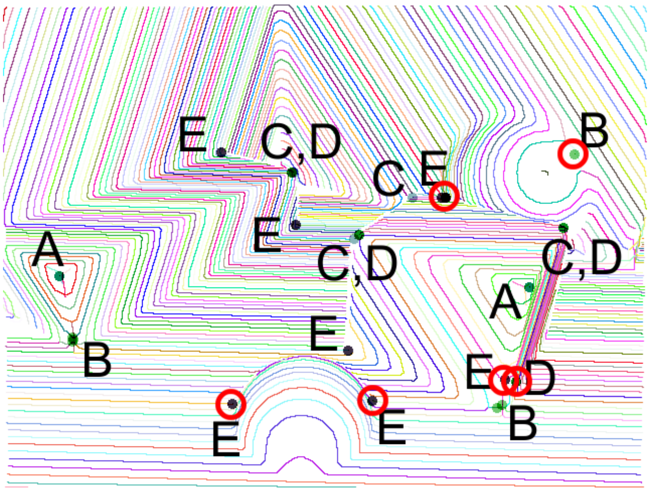

Fig. 8. All true corners were found. False true detections are circled.

The same image was also tested at even lower depth levels (35 layers). Still all the corners were found. Two false positive corners appeared around spherical objects.

We also evaluated the sensitivity of the detected corners by changing the number of layers. Figures show the real number of corners in the image (ground truth), the number of true positive detections (corners may be identified several times improving reliability), the number of different true positive corners (all corners counted once), the number of false positive and false negative detections. Results show that increasing the number of layers over a point only marginally benefits the number of detected different corners. The total number of detected corners increases gradually (as the same corners are detected several times). This

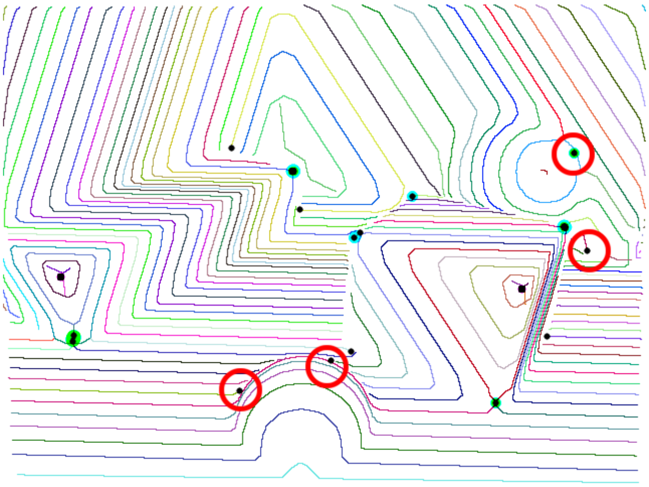

Fig. 9. Corners detected in image having even less range levels

effect may be used to improve credibility of corners detected several times in a small radius.

Figure 10 shows a favorable situation: the total number of detected corners are increasing with using more and more layers. So does the number of different corners until it reaches a maximal value slightly lower than the ground truth. The number of improper detections (false positive and negative) both remain low independently from the number of layers.

Figure 11 presents a much less favorable situation. This image contained not only box type of objects but spheres as well. As the number of layers increases, details are show up. Continuous large layers break into several parts not suitable for this method. The spherical objects also mislead the detector.

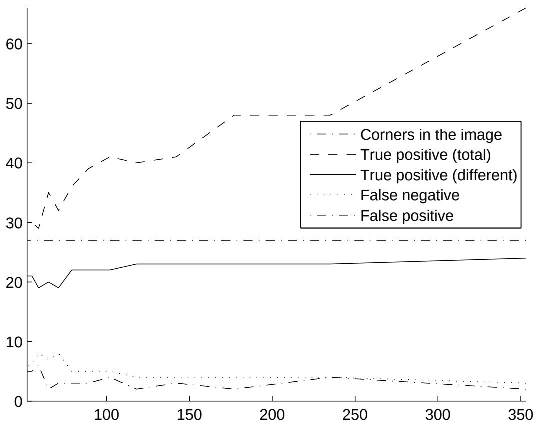

Fig. 10. Impact of changing the number of layers to the corner detection (a)

4.2 Captured range image

The range images were captured using a structured light based sensor. The produced image contained only approximately 50 different layers. The raw image suffered from significant noise which was successfully reduced due to the dilation-thinning procedures. Most of the true corners were found but due to the shadows introduced by the sensor (due to the principal of the operation) led to inaccurate step edge detection. These false edges appeared around the shadows interfering with the conditions. To reduce the effects of the shadows, simple inpainting was used. Pixels in the shadow areas where no distance data was available were filled with the distance data of the first non-shadow or missing pixel's data to the right in the row. The direction was based on the principal of operation. This simple operation eliminated the false step edges around shadow areas while it also kept real step edges if present. This row by row operation distorted layers, however in case of small missing patches or shadows areas, these were filtered successfully. Dilation was used as well to reduce noise. Compared to synthetic images, a slightly larger filter radius was used for captured images.

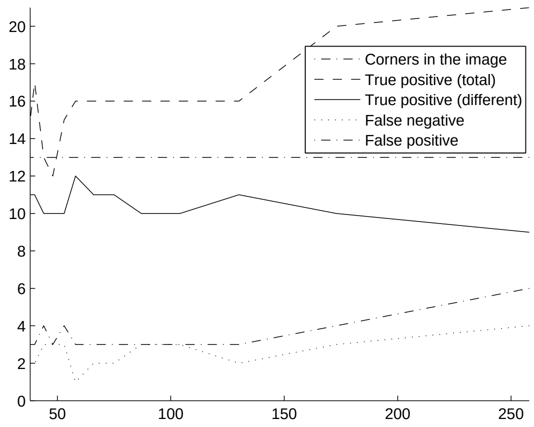

Fig. 11. Impact of changing the number of layers to the corner detection (b)

Here we present the results on three test images. Figure 12 shows two boxes and the detected corners. The image consists of only 45 layers. Out of 15 corners 12 were identified successfully. The furthest points on the top of the boxes were missed due to the small breakage angles of jump edges. The rest of corners were identified successfully sometimes even multiple times. Depending on the error threshold 5-6 false positive corners were found (1 type B, 1 type D, 3 type E).

Figure 13-14 show the results of test image (b) and (c). Test image (b) consists of 42 range layers. Out of 14 corners 11 were identified. The number of false positive corners is 7. In test image (c) the method found 9 out of 11 corners. Also 4 false positive detections also appeared.

5 Conclusions and future work

The algorithm utilizes many parameters which must be set up in advance based on the sensor properties (such as measurement distance, quantization etc.). It is subject of further research to optimize these values based on image contents to minimize user defined parameters. The algorithm performed very well in synthetic conditions. Some special cases should be handled in the corner detection condition sets as well. Ideal images showed that increasing the number of layer beyond a point does not provide any benefits, it is reasonable to compress range images before applying this algorithm.

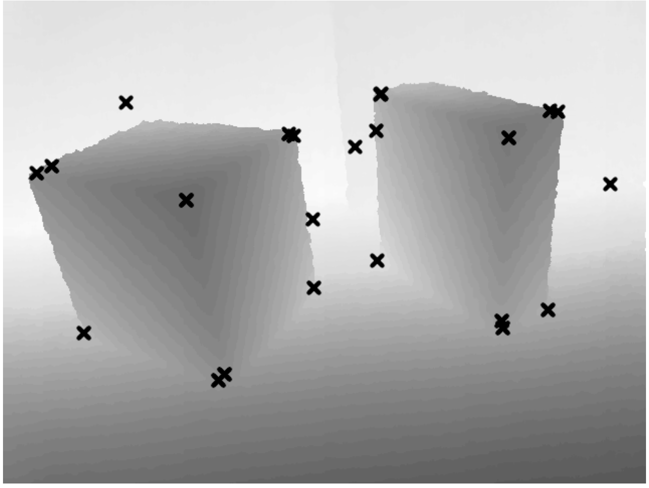

Fig. 12. Test image (a) (real)

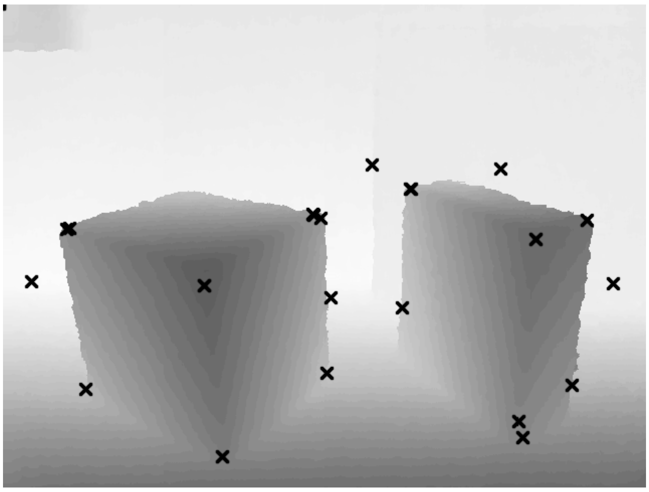

Fig. 13. Test image (b) (real)

Real data suffers from very low number of range levels, significant noise and other distortions which misled the corner detector. Due to the principal of operation shadows may appear near object borders introducing false step edges. Further research must be carried out to improve robustness in such conditions. Also parameter values required by the algorithm should be adaptively based on image content.

Future work also aims to give a descriptor generation method and utilize color information during corner detection. In many cases both information sources (range and intensity) are needed to detect and verify corner points. We look forward to utilize such an algorithm for SLAM (simultaneous localization and mapping) applications in mobile robotics.

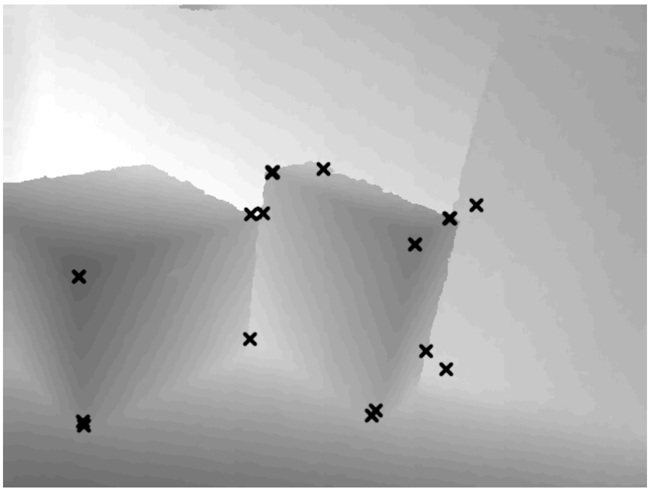

Fig. 14. Test image (c) (real)

References

1 **Harris C, Stephens M**, *A Combined Corner and Edge Detector*, In: Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147–151, `http://www.bibsonomy.org/bibtex/22d2048f92453ed8d1426782cfe774b62/zap`.

2 **Lowe DG**, *Distinctive Imag e Features from Scale-Invariant Keypoints*, Int. J. Comput. Vision, **60**(2), (2004), 91–110, DOI 10.1023/B:VISI.0000029664.99615.94.

3 **Bay H, Ess A, Tuytelaars T, Van Gool L**, *Speeded-Up Robust Features (SURF)*, Comput. Vis. Image Underst., **110**(3), (2008), 346–359, `http://dx.doi.org/10.1016/j.cviu.2007.09.014`, DOI 10.1016/j.cviu.2007.09.014.

4 **Coleman S, Suganthan S, Scotney B**, *Characterising Range Image Features via Gradient Operators*, In: Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on, 2007, pp. 629-634, DOI 10.1109/ICIAP.2007.4362847.

5 **Bose SK, Biswas KK, Gupta SK**, *An integrated approach for range image segmentation and representation*, Artificial Intelligence in Engineering, **10**(3), (1996), 243–252, `http://dx.doi.org/10.1016/0954-1810(95)00035-6`, DOI 10.1016/0954-1810(95)00035-6.

6 **Ideses IA, Yaroslavsky LP, Fishbain B**, *Depth maps: faster, higher and stronger?*, In: Proc. SPIE 6805, Three-Dimensional Image Capture and Applications, 2008, pp. 68050B-68050B-12, DOI 10.1117/12.765536.

7 **Changchang W, Clipp B, Xiaowei L, Frahm J-M, Pollefeys M**, *3D model matching with Viewpoint-Invariant Patches (VIP)*, In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1-8, DOI 10.1109/CVPR.2008.4587501.

8 **Steder B, Rusu RB, Konolige K, Burgard W**, *Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries*, In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2011.

9 **Németh G, Palágyi K**, *2D Parallel Thinning Algorithms Based on Isthmus-Preservation*, In: ISPA 2011: 7th International Symposium on Image and Signal Processing and Analysis: Dubrovnik, Croatia, 4 - 6 September 2011. IEEE, 2011, pp. 585-590.