Project IP-2014-09-3155

Advanced 3D Perception for Mobile Robot Manipulators

DETECTION OF LOCAL REFERENCE FRAMES AND GEOMETRIC PRIMITIVES BASED ON PLANAR PATCHES

Technical Report

ARP3D.IR4.11-B

version 1.0

Robert Cupec

Josip Juraj Strossmayer University of Osijek

Faculty of Electrical Engineering, Computer Science and

Information Technologies Osijek

Osijek, 2018.

Ovaj rad je financirala Hrvatska zaklada za znanost projektom IP-2014-09-3155.

Mišljenja, nalazi i zaključci ili preporuke navedene u ovom materijalu odnose se na autora i ne odražavaju nužno stajališta Hrvatske zaklade za znanost.

This work has been fully supported by the Croatian Science Foundation under the project number IP-2014-09-3155.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Croatian Science Foundation.

1. Introduction

In this report, three methods designed as preprocessing steps for image segmentation, object recognition and object classification are proposed. Some object recognition approaches are based on alignment of a 3D model with a point cloud representing an object on the scene (Aldoma et al., 2012; Aldoma et al., 2013; Aldoma et al., 2016; Glover & Popovic, 2013; Papazov & Burschka, 2010). These approaches generate object hypotheses by matching features detected in the scene with the features of the same type extracted from the models of objects of interest. Each feature is assigned a *local reference frame* (LRF). By aligning a scene LRF with a model LRF, a hypothesis that an instance of this model is present in the scene in a particular pose is generated. In this report, we propose a novel method for defining LRFs based on detection of planar patches. Another method described in this report is an approach for segmenting a 3D point cloud into approximately convex surfaces. The obtained convex surfaces can be used as the basis for solving a higher-level scene understanding problem such as object recognition or object classification. Several approaches for segmentation of point clouds into convex surfaces are proposed in the literature (Attene et al., 2008; Mamou & Ghorbel, 2009; Cupec, Nyarko & Filko, 2011). The main quality of the approach proposed in this report is its computational efficiency and that it allows detection of overlapping convex surfaces. Detection of overlapping convex surfaces can be useful in solving object recognition or classification tasks in the case of shapes which cannot be uniquely segmented into convex primitives. The third method proposed in this report is a method for detection of toroidal surfaces. To the best of our knowledge, this is the only algorithm for detection of toroidal surfaces in 3D point clouds.

The approaches proposed in this paper are based on polyhedral representation of the input 3D scene, where all object surfaces in the scene are represented by polygons, referred to in this paper as *faces*, with arbitrary number of vertices. Furthermore, it is assumed that this representation is optimized w.r.t. the number of faces, which means that low curvature regions are represented by less faces, while high curvature regions are represented by a greater number of faces. Hence, the size of a face can be used as a measure of the stability of the face orientation. The representation with these properties can be obtained by the segmentation approach proposed in (Cupec, 2016). This approach, based on region growing, segments a dense 3D triangular mesh into approximately planar patches, whose boundaries are well aligned to the intersection lines between the supporting planes of adjacent patches. Hence, the obtained planar patch set can be regarded as an approximation of a polygonal representation of the input mesh. The points where three adjacent planar patches meet are referred to in this paper as *vertices*. Furthermore, the segment of a planar patch boundary connecting two vertices is referred to as an *edge*. The discussed segmentation algorithm provides the information about vertices and edges, where each vertex is assigned a list of edges connecting this vertex with other vertices. This information is used in the approaches presented in the following sections.

2. Tangent Based Local Reference Frames

In this section, an approach for defining LRFs with stable orientation, which can be used for generating object pose hypotheses is presented. A common approach for defining LRFs is to use the surface normal distribution in the local neighborhood of a selected keypoint to define the orientations of the LRF axes (Zhong Y, 2009). Stability of the LRFs defined in such a way depends on the geometric properties of the local neighborhood of the keypoint, where high curvature of the local

surface usually assures better LRF stability. In this paper, we propose a different approach, where low curvature regions are used to define stable LRFs. LRFs are defined by segmenting the object surface into planar patches and using these patches to define LRFs. A LRF defined using this approach is more stable if it is defined using larger planar patches.

The idea of the proposed approach is to identify low curvature regions on object surface, determine their tangential planes and create LRFs defined by two tangential planes with sufficiently different orientations in order to obtain stable LRF orientation. Two non-parallel planes define 5 of total of 6 degrees of freedom (DoF) of a 3D LRF. Let the normal of one of these two planes define the orientation of the z-axis of the LRF. The intersection line of these two planes is taken as its x-axis, while the origin of the considered LRF is constrained to lie on that line. Then, these two planes define 5 DoF of the LRF, since y-axis is uniquely determined by the x- and z-axis and the only remaining DoF needed to define this LRF completely is the position of the LRF's origin on the intersection line of the two planes.

We assume that the tangential plane in any point of a planar patch is identical to the supporting plane of this patch. Furthermore, we assume that a point on a common edge of two planar patches can have an infinite number of tangential planes, i.e. all planes containing this edge whose normals are defined by

$$n = \frac{sn_i + (1-s)n_j}{\left\| sn_i + (1-s)n_j \right\|}, \qquad 0 \le s \le 1.$$

where n_i and n_j are normals of the two considered planar patches.

In order to obtain a stable LRF, we require that the angle between the normals of two tangential planes defining 5DoF of a LRF is at least 45°. The proposed method starts with an initial planar patch F_i, whose supporting plane Π_i is used as the first tangential plane of a LRF. Then, the second tangential plane is searched in the neighborhood of the initial planar patch by a region growing process. The region growing procedure starts from the initial planar patch F_i being the initial region, which is then grown by adding adjacent planar patches whose normal is oriented at the angle less than 45° w.r.t. the normal n_i of F_i. If a planar patch F_j, which is added to the considered region, has a neighbor F_k whose normal n_k is oriented at the angle ≥ 45° w.r.t. n_i, then a tangential plane Π'_{ijk} is computed, which contains the common vertices of F_j and F_k and whose normal is oriented at the angle of 45° w.r.t. n_i. Note that since the angle between n_i and n_j is ≥ 45° and the angle between n_i and n_k is less than 45°, i.e.

$$\operatorname{acos}\left(n_i^T \cdot n_j\right) < \frac{\pi}{4}, \qquad \operatorname{acos}\left(n_i^T \cdot n_k\right) \ge \frac{\pi}{4},$$

there is a value s, for which

$$\operatorname{acos}\left(n_i^T \cdot \left(sn_j + (1-s)n_k\right)\right) = \frac{\pi}{4}, \tag{1}$$

which means that there is a plane with normal

$$n'_{ijk} = \frac{sn_j + (1-s)n_k}{\left\| sn_j + (1-s)n_k \right\|} \tag{2}$$

such that the angle between n_i and n'_{ijk} is 45°. Value s in (2) can be computed by solving (1).

Planes Π_i and Π'_{ijk} define a LRF, where z-axis of this LRF is parallel to n_i and x-axis is perpendicular to both n_i and n'_{ijk}. An example is given in Fig. 1, where the supporting planes of the initial planar patch F_i and the tangential plane at the edge E_{jk} between two planar patches F_j and F_k are depicted with green and blue dashed lines respectively.

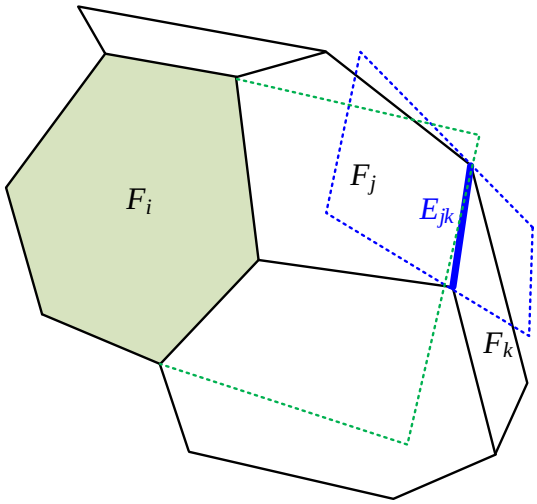


Fig. 1. A polyhedral surface with faces F_i, F_j and F_k. A LRF is defined by the supporting plane of F_i depicted by the green dashed line and a tangential plane at the edge between F_j and F_k depicted by the blue dashed line.

The region growing procedure which generates LRFs from a particular initial planar patch is constrained to the convex surface which that segment belongs to. Clustering of planar segments into convex surfaces is described in Section 3.

As explained at the beginning of this section, the proposed approach for defining LRFs is designed for generating pose hypotheses. Since, the number of features used in hypothesis generation process determines the complexity of this process, a desirable property of a hypothesis generation method is that it generates a relatively small set of hypotheses, which contains the true hypothesis with a high probability. In order to reduce the number of features, the proposed approach considers only dominant planar patches. For a particular convex surface, the largest planar patch F_i is identified and only planar patches whose size is $\geq \tau_{TRF,1} \cdot |F_i|$ are considered as candidates for initialization of the LRF generation process, where $\tau_{TRF,1}$ is a user defined parameter and $|F_i|$ is the size of the largest planar patch of the considered convex surface. Furthermore, the distance λ_{ijk} between the orthogonal projections of the common vertices of F_j and F_k onto the intersection line of Π_i and Π'_{ijk} is used as a measure of saliency of the tangential plane Π'_{ijk}. All tangential planes Π'_{ijk} generated for a given segment F_i are sorted according to this measure and only those planes Π'_{ijk} having $\lambda_{ijk} \geq \tau_{TRF,2} \cdot \lambda_{i,max}$ are used to generate LRFs, where $\tau_{TRF,2}$ is a user defined parameter and $\lambda_{i,max}$ is the maximum value λ_{ijk} for that particular patch F_i. Furthermore, in order to avoid creation of similar LRFs, a LRF is

created from a plane Π'_{ijk} only if there isn't another plane Π'_{ipq} such that the angle between x-axes of TRFs created from Π'_{ijk} and Π'_{ipq} is < $\tau_{TRF,3}$ and $\lambda_{ipq} > \lambda_{ijk}$.

3. Segmentation into Convex Surfaces

In this section, an algorithm for segmentation of 3D point clouds into convex surfaces based on aggregation of planar patches is presented. The input to the algorithm is a triangular mesh $\mathcal{M}$ obtained from a 3D point cloud and the output is a set $\mathcal{C}$ of convex surfaces C_i. First, $\mathcal{M}$ is segmented into planar patches using the method described in (Cupec, 2016). A benefit of this representation is that the convex hull of the vertices of a set of planar patches represents a good approximation of the convex hull of the union of all points of this set, which allows fast aggregation of planar patches into convex segments, since only a relatively small number of vertices must be considered.

Aggregation of planar patches into approximately convex surfaces is based on a region growing process explained by Algorithm 1. The process is initialized by selecting the largest planar patch F_i which represents the initial convex surface C_k from set $\mathcal{B}$ containing of all planar patches which are not already assigned to a convex surface. This surface is then grown by adding adjacent segments F_j which satisfy convexity constraint with the convex surface C_k grown so far. In addition to planar patches, a convex surface C_k is assigned a set of vertices V_{C_k}. The convexity criterion a planar patch F_j must satisfy in order to be joined to a convex surface C_k is that the percentage of the vertices from the set V_{F_i} which belong to the convex surface is $\geq \eta_{convex}$, where η_{convex} is a user defined parameter. A vertex whose position is defined by vector p is considered to belong to the convex surface C_k if the following condition is satisfied

$$n_p^T \cdot p - d_p \leq \varepsilon_{convex} \tag{3}$$

for all $F_p \in C_k$, where n_p is the normal of F_p, d_p is the distance of the supporting plane of F_p to the origin of the mesh RF and ε_{convex} is a user defined tolerance. If a segment F_j satisfies the convexity criterion, it is added to the convex surface C_k and all of its vertices which satisfy (3) are added to V_{C_k}.

Since a new planar segment considered for addition to a grown convex surface must be consistent with all the segments already grouped in this convex surface according to the convexity criterion, the final shape of the convex surface is heavily dependent on the order in which planar segments are added to it. In the approach applied in our research, selection of the next segment for inclusion in the grown convex surface is made according to the similarity of the orientation of its normal to the mean convex surface normal. The mean normal of a convex surface C_k is defined by

$$\bar{n}_k = \frac{\sum_{F_i \in C_k} |F_i| \cdot n_i}{\left\| \sum_{F_i \in C_k} |F_i| \cdot n_i \right\|},$$

where $|X|$ denotes the cardinality of a set X. In each iteration of the region growing process, the list Q of candidates for inclusion in the grown convex surface C_k is updated and, at the beginning of each iteration, the candidate whose normal has most similar orientation to $\overline{\boldsymbol{n}}_k$ is selected. After a region growing process is completed, the planar patches aggregated in the new surface C_k are removed from set $\mathcal{B}$. Then, another region growing process is started by selecting the largest planar patch in $\mathcal{B}$.

Algorithm 1 Mesh Segmentation into Convex Surfaces

Input: $\mathcal{M}$, ε_{convex}, η_{convex}

Output: $\mathcal{C} = \{C_1, C_2, \dots\}$

1 : Segmentation of $\mathcal{M}$ into planar surface segments. The result is a set of planar segments $\mathcal{F} = \{F_1, F_2, \dots\}$
2 : $\mathcal{B} \leftarrow \mathcal{F}$
3 : $k \leftarrow 1$
4 : **Repeat**
5 : $F_i \leftarrow$ the largest segment in $\mathcal{B}$
6 : $C_k \leftarrow F_i$
7 : $\overline{\boldsymbol{n}}_k \leftarrow n_i$
8 : $Q \leftarrow F_i$
9 : $V_{C_k} \leftarrow \varnothing$
10 : **Repeat**
11 : $F_j \leftarrow$ segment from Q whose normal has the most similar orientation to $\overline{\boldsymbol{n}}_k$
12 : Remove F_j from Q.
13 : $V_{F_j} \leftarrow$ set of all vertices on the boundary of F_j
14 : $\overline{V}_{F_j} \leftarrow$ subset of V_{F_j} containing only vertices which satisfy (3) for all $F_l \in C_k$
15 : **If** $|\overline{V}_{F_j}| / |V_{F_j}| \geq \eta_{convex}$ **then**
16 : $V_{C_k} \leftarrow V_{C_k} \cup \overline{V}_{F_j}$
17 : $\overline{\boldsymbol{n}}_k \leftarrow \dfrac{|C_k| \cdot \overline{\boldsymbol{n}}_k + |F_j| \cdot n_j}{\left\| |C_k| \cdot \overline{\boldsymbol{n}}_k + |F_j| \cdot n_j \right\|}$
18 : $C_k \leftarrow C_k \cup F_j$
19 : Remove from Q all segments which don't satisfy (3) for all vertices from $\overline{V}_{F_j}$.
20 : Add to Q all segments from $\mathcal{B}$ adjacent to F_j which satisfy (3) for all vertices from V_{C_k} and which are not in Q already.
21 : **end if**
22 : **until** Q is empty.
23 : Add C_k to $\mathcal{C}$.
24 : Remove from $\mathcal{B}$ all segments contained in C_k.
25 : $k \leftarrow k + 1$
26 : **until** $\mathcal{B}$ is empty.
27 : **return** $\mathcal{C}$

As an example, the result obtained by applying the proposed mesh segmentation algorithm to a depth image and a 3D object model from the Kinect dataset (Aldoma et al., 2012) is shown in Fig. 1, where every convex segment is displayed in different color.

During the region growing process, the candidates for inclusion into the grown convex surface are searched among the elements of $\mathcal{B}$, i.e. among planar patches which are not already assigned to a convex surface (See line 20 of Algorithm 1). Thereby, convex surfaces representing disjoint point sets are obtained. If the condition for inclusion of adjacent planar patches into the grown convex surface C_k in line 20 of Algorithm 1 is modified to allow addition of all planar patches $F_l \in \mathcal{F} \setminus C_k$ adjacent to F_j instead of constraining the selection to $\mathcal{B}$, another variant of the discussed algorithm is obtained, which produces overlapping convex surfaces.

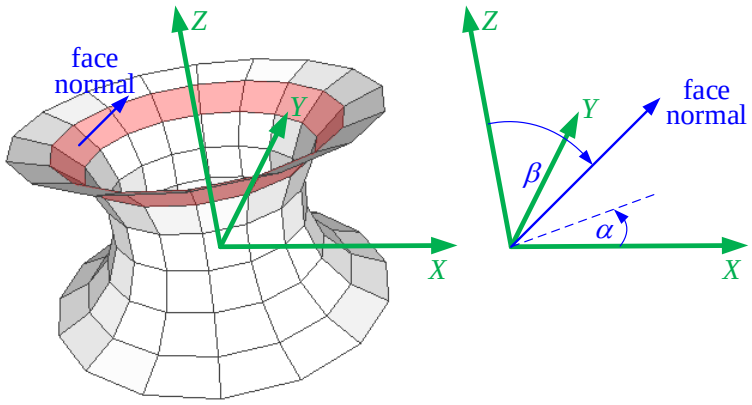
(a) (b)

Fig. 1. Segmentation of a 3D object model (a) and a depth image (b) into approximately convex surfaces. Convex surfaces are painted in different colors.

4. Detection of Toroidal Surfaces

In this section, an algorithm for detection of toroidal surfaces in 3D meshes is proposed. Toroidal surfaces considered in this paper can be used to model e.g. holes in objects or bended objects like bended pipes or cables. The input to the algorithm is a triangular mesh $\mathcal{M}$ obtained from a 3D point cloud and the output is one or multiple instances or the torus model described in this section.

The model of toroidal surfaces used in this work is shown in Fig. 2. Toroidal surfaces are modelled by polyhedrons composed of multiple faces. A toroidal surface is defined w.r.t. a reference frame (RF), depicted in Fig. 2 in green color. The orientation of each face normal w.r.t. this RF is defined by two angles explained in Fig. 2. Angle α defines the orientation of the orthogonal projection of a face normal onto the xy-plane, while β is the angle between the face normal and the z-axis. The faces are grouped in *cones*, where all faces of one cone have the same angle β, referred to in this paper as the *cone angle*. One of these cones is denoted in Fig. 2 by red color.



(a) (b)

Fig. 2. Model of toroidal surfaces. The dashed blue line represents the orthogonal projection of the considered face normal onto the xy-plane of the reference frame.

The ith cone is defined by cone angle β_i, where $i = 1, \ldots, n_\beta$. Values β_i are user defined. By defining this angle sequence, a user can choose a desired resolution of the torus model. In our research, we use the angle sequence defined by

$$\beta_i = (i-4)\pi/8, \quad i = 1,\ldots,7.$$

Analogously, angles α of the torus model are defined by a sequence α_1, α_2, …, α_{n_α}. The same sequence is used for all cones. In our research, we use the angle sequence defined by

$$\alpha_j = (j-1)\pi/8, \quad j = 1,\ldots,16.$$

Let F_{ij} be the jth face of the ith cone of a torus model. The normal of this face is defined by

$$n_{ij} = \left[\cos\alpha_j \sin\beta_i, \quad \sin\alpha_j \sin\beta_i, \quad \cos\beta_i\right]^T. \tag{4}$$

The supporting plane of face F_{ij} is defined by the following equation

$$n_{ij}^T p - d_{ij} = 0,$$

where p is the coordinate vector of a point on the considered plane w.r.t. the torus model RF and d_{ij} is the distance of this plane w.r.t. the same RF. Each plane defines a half-space. In this paper, we use the following convention. If

$$n_{ij}^T p - d_{ij} \le 0, \tag{5}$$

then point p belongs to the half-space defined by plane (n_{ij}, d_{ij}). The ith cone of a torus model can be regarded as the point set obtained by union of half-spaces defined by planes (n_{ij}, d_{ij}), j = 1, …, n_α. Hence, a point p belongs to the ith cone if at least one of equations (5), where j = 1, …, n_α, is satisfied, i.e. if

$$\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right) \le 0. \tag{6}$$

The whole torus represents the intersection of point sets defined by (6) for i = 1, …, n_β. Hence, a point p belongs to the torus if all equations (6), where i = 1, …, n_β, are satisfied, i.e. if

$$\max_{1\le i\le n_\beta}\left(\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right)\right) \le 0. \tag{7}$$

A point p lies on the torus surface if

$$\max_{1\le i\le n_\beta}\left(\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right)\right) = 0. \tag{8}$$

The set of points belonging to a torus can be defined by

$$V_T = \left\{ p \in \mathbb{R}^3 \,\middle|\, \max_{1\le i\le n_\beta}\left(\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right)\right) \le 0 \right\}.$$

Let's consider a case where a torus V_T is used to model a hole in an object. Let O be the object without the hole and O' the same object, but with hole modelled by the torus. Then, the following equation holds

$$O' = O \cap V_T .$$

Given a RF and a 3D mesh $\mathcal{M}$, the approach proposed in this section detects toroidal surfaces by detecting cones and grouping the detected cones into toroidal surfaces. First, $\mathcal{M}$ is segmented into planar patches using the method described in (Cupec, 2016). As the result, mesh $\mathcal{M}$ is represented by a set of planar patches, with vertices and edges defined in Section 1. Then, cones are searched by considering the tangents in all edges as candidates for cone faces. An example of a tangential plane at an edge between two planar faces is shown in Fig. 1. As already noted in Section 2, infinite number of tangential planes exists at an edge between two faces. Let $\mathcal{T}(E)$ be the set of all tangential planes in edge E. For a particular β_i, cones are detected by searching for connected sets $\mathcal{E}_c$ of mesh edges with the property that for every $E \in \mathcal{E}_c$ there is a tangential plane $(n_{E,i}, d_{E,i}) \in \mathcal{T}(E)$ such that

$$\angle\left(n_{E,i}, z\right) = \beta_i ,$$

where symbol $\angle$ denotes the angle between two vectors and z is the unit vector representing the z-axis of the torus RF. A set of edges $\mathcal{E}_c$ is *connected* if for any two edges E, $E' \in \mathcal{E}_c$ there exists a sequence E_1, E_2, .., E_n, such that $E_1 = E$, $E_n = E'$ and each two consecutive edges E_i and E_{i+1} share a common endpoint. Furthermore, we require that each edge $E \in \mathcal{E}_c$ defines one cone face. Hence, an edge $E \in \mathcal{E}_c$ should not be contained in the half-space defined by the plane $(n_{E',i}, d_{E',i})$ of another edge $E' \in \mathcal{E}_c$, i.e. the following condition must be satisfied by all edge pairs E, $E' \in \mathcal{E}_c$

$$n_{E',i}^T p_{E,k} - d_{E',i} > 0, \quad k = 1,2 , \tag{9}$$

where $p_{E,1}$ and $p_{E,2}$ are the endpoints of E. In order to make the algorithm robust to the measurement noise, instead of (9), the following relaxed condition is used

$$n_{E',i}^T p_{E,k} - d_{E',i} \geq -\varepsilon, \quad k = 1,2 , \tag{10}$$

where ε is an experimentally determined tolerance. From every edge set $\mathcal{E}_c$ with the properties described above, a cone is created. For every $j = 1, \ldots, n_\alpha$, d_{ij} is computed by

$$d_{ij} = \min_{\substack{E \in \mathcal{E} \\ k=1,2}} \left(n_{ij}^T p_{E,k} \right). \tag{11}$$

The obtained parameters d_{ij} together with normals n_{ij} define a complete cone. Note that each d_{ij} is defined by normal n_{ij} and one of the endpoints of edges $E \in \mathcal{E}_c$, for which the dot product $n_{ij}^T p_{E,k}$ is minimum. The endpoint which defines parameter d_{ij} is referred to in this paper as the *reference point* of the plane (n_{ij}, d_{ij}) and denoted by p_{ij}.

The detected cones are then combined in toruses. A torus is created from a sequence of cones, where each of these cones corresponds to one angle β_i, $i = 1, \ldots, n_\beta$. The criterion which is used to decide whether two cones belong to the same torus is that all reference points of these cones must lie on the

surface of the torus consisting of these two cones. This means that all reference points of two cones, where the angle of the first cone is β_i and the angle of the other cone is β_{i+1}, must satisfy the following equation

$$\max_{i,i+1}\left(\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right)\right) = 0. \tag{12}$$

Equation (12) is a special case of equation (8) for a torus consisting of two consecutive cones. Since d_{ij} is computed by (11), all reference points of the ith cone satisfy

$$\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right) = 0.$$

Furthermore, if $p_{i+1,j}$ is the reference point of the plane ($n_{i+1,j}$, $d_{i+1,j}$) and

$$n_{ij}^T p_{i+1,j} - d_{ij} \le 0\,, \tag{13}$$

then

$$\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right) \le 0.$$

Consequently, if (13) is satisfied for all reference points of two consecutive cones, then (12) is satisfied for all reference points of these two cones. In order to make the algorithm robust to the measurement noise, instead of (13), the following relaxed condition is used

$$n_{ij}^T p_{i+1,j} - d_{ij} \le \varepsilon\,. \tag{14}$$

Finally, a torus is created for every sequence of cones, where each two consecutive elements of the sequence satisfy the described criterion.

Algorithm 2 represents an implementation of the described method. The input to this algorithm are the set of vertices $\mathcal{V}$ and set of edges $\mathcal{E}$ of planar patches obtained by the method described in (Cupec, 2016), where vertices are represented by their coordinates w.r.t. the given RF. Furthermore, angles α_j, $j = 1, \ldots, n_\alpha$ and β_i, $i = 1, \ldots, n_\beta$, tolerance ε and the minimum number of edges in a cone $n_{mincone}$ are also provided by the user. The discussed algorithm allows detection of toruses consisting of an arbitrary number of cones between 1 and n_β. In general, a torus can include cone angles from any set β_i, $i_{min} \le i \le i_{max}$, where $1 \le i_{min} \le i_{max} \le n_\beta$. The algorithm returns a set of toruses $\mathcal{D}$ represented by pairs (i_{min}, d), where i_{min} is the index of the angle β_i of the first cone and d is a vector composed of parameters d_{ij} of all torus faces. Vector d represents a concatenation of vectors d_i defining the faces of particular cones, i.e. the first n_α elements of d are parameters $d_{i_{min},j}$, $j = 1, \ldots, n_\alpha$, the second n_α elements of are parameters $d_{i_{min}+1,j}$, $j = 1, \ldots, n_\alpha$, etc. Hence, vector d has total of $n_\alpha \cdot (i_{max} - i_{min} + 1)$ elements. This vector together with angles α_j and β_i and the torus RF completely defines a torus. The torus surface is defined by the following equation

$$\max_{i_{min}\le i\le i_{max}}\left(\min_{1\le j\le n_\alpha}\left(n_{ij}^T p - d_{ij}\right)\right) = 0,$$

where normals n_{ij} are defined by (4).

Algorithm 2 Detection of toroidal surfaces

Input: $\mathcal{V}$, $\mathcal{E}$, α_j, $j = 1, \ldots, n_\alpha$, β_i, $i = 1, \ldots, n_\beta$, ε, $n_{mincone}$

Output: $\mathcal{D}$

1 : Compute $\mathcal{T}(E)$ for all edges $E \in \mathcal{E}$.
2 : Create empty graph $\mathcal{G}$.
3 : $\mathcal{D} \leftarrow \emptyset$
4 : **For** $i = 1$ **to** n_β
5 : $\mathcal{C}_i \leftarrow \emptyset$
6 : Determine the set $\mathcal{E}_i$ of edges $E \in \mathcal{E}$ for which a tangential plane $(n_{E,i}, d_{E,i}) \in \mathcal{T}(E)$ exists.
7 : $\mathcal{E}_{assigned} \leftarrow \emptyset$
8 : **For** every edge $E \in \mathcal{E}_i$
9 : **If** $E \notin \mathcal{E}_{assigned}$ **then**
10 : $\mathcal{E}_{connected} \leftarrow$ the connected set of edges $E' \in \mathcal{E}_i \setminus \mathcal{E}_{assigned}$ containing E
11 : $\mathcal{E}_{assigned} \leftarrow \mathcal{E}_{assigned} \cup \mathcal{E}_{connected}$
12 : **If** $|\mathcal{E}_{connected}| \geq n_{mincone}$ **then**
13 : Determine $\mathcal{E}_c \subseteq \mathcal{E}_{connected}$ containing edges which satisfy (10) by greedy search.
14 : **If** $|\mathcal{E}_c| \geq n_{mincone}$ **then**
15 : Compute d_{ij} for every $j = 1, \ldots, n_\alpha$ by applying (11) to all edges $E \in \mathcal{E}_c$.
16 : Create vector d whose components are values d_{ij}.
17 : Create matrix P, where the jth column of P is the coordinate vector p_{ij} of the reference point of the plane (n_{ij}, d_{ij}).
18 : Add $C = (d, P)$ to $\mathcal{C}_i$.
19 : Insert C as a node into $\mathcal{G}$.
20 : **If** $i > 1$ **then**
21 : **For** every $C' \in \mathcal{C}_{i-1}$
22 : **If** all reference points of C and C' satisfy (14), **then** Connect nodes C and C' by a directed edge, where C is the source node and C' is the target node.
23 : **end for**
24 : **end if**
25 : **end if**
26 : **end if**
27 : **end if**
28 : **end for**
29 : **end for**
30 : Detect all sequences of connected nodes in graph $\mathcal{G}$ starting with a node, which is not a target node of any directed edge in $\mathcal{G}$, and terminating in a node, which is not a source node of any directed edge in $\mathcal{G}$.
31 : **For** every node sequence $C_i = (d_i, P_i)$, $i = i_{min}, \ldots, i_{max}$, detected in line 30, where $C_i \in \mathcal{C}_i$.
32 : Create a torus from the node sequence. A torus is represented by the index i_{min} and vector $d = \left[d_{i_{min}}^T, \ \ldots, \ d_{i_{max}}^T \right]^T$.
33 : Insert pair (i_{min}, d) into $\mathcal{D}$.
34 : **end for**
35 : **return**

The greedy search in line 13 is implemented as follows. First, compatibility of each edge pair (E', E'') is determined by evaluating (10). Then, each edge $E' \in \mathcal{E}_{connected}$ is assigned a cost representing the sum of the lengths of all edges which are not compatible with E' minus the length of E'. Then, a list of edges sorted according to this cost in ascending order is created. Finally, the edges from this list and are added to $\mathcal{E}_c$ one by one starting from the one with the lowest cost, where the edges, which are not compatible with all edges already in $\mathcal{E}_c$, are not added. Thereby, the obtained set $\mathcal{E}_c$ contains only edges which are mutually compatible according to the criterion (10). The greedy search tends to maximize the total sum of edge lengths in $\mathcal{E}_c$.

The proposed approach detects toroidal surfaces aligned with a given RF. Note that only the z-axis of the torus RF is important for successful torus detection, since it defines the torus axis, while the orientation of the other two axes doesn't influence significantly the result. If no a priori knowledge about the torus axis is available, the proposed algorithm can be repeatedly applied for a set of different axis orientations. For example, a set of unit vectors obtained by uniform sampling of the unit sphere can be used as z-axes of the torus RF, while the other two axes can be chosen arbitrarily.

References

Aldoma A, Tombari F, Di Stefano L and Vincze M (2012) A Global Hypothesis Verification Method for 3D Object Recognition, *European Conference on Computer Vision (ECCV)*, pp. 511–524.

Aldoma A, Tombari F, Prankl J, Richtsfeld A, Di Stefano L and Vincze M (2013) Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DoF pose estimation, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2104–2111.

Aldoma A, Tombari F, Di Stefano L and Vincze M (2016) A Global Hypothesis Verification Framework for 3D Object Recognition in Clutter, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1383–1396.

Attene M, Mortara M, Spagnuolo M and Falcidieno B (2008) Hierarchical Convex Approximation of 3D Shapes for Fast Region Selection, *Computer Graphics Forum*, vol. 27, no. 5, pp. 1323–1332.

Cupec R (2016) 3D Mesh Segmentation into Planar Segments Based on Region Growing and Plane Intersection, Croatian Science Foundation, project IP-2014-09-3155, Technical Report ARP3D.TR3, J. J. Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technologies Osijek, Croatia [http://ferit.hr/ARP3D/TR3_v2.1.pdf]

Cupec R, Nyarko EK and Filko D (2011) Fast 2.5D Mesh Segmentation to Approximately Convex Surfaces, *Proceedings of the European Conference on Mobile Robots (ECMR)*, Örebro, Sweden pp. 127–132

Glover J and Popovic S (2013) Bingham Procrustean Alignment for Object Detection in Clutter, *IEEE/RSJ Int. Conf. Intell. Robots & Systems (IROS)*, Tokyo, pp. 2158–2165.

Mamou K and Ghorbel F (2009) A simple and efficient approach for 3D mesh approximate convex decomposition, *Int. Conf. on Image Processing (ICIP)*, Cairo, Egypt.

Papazov C and Burschka D (2010) An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes, *Asian Conference on Computer Vision (ACCV)*, Part I, pp. 135–148

Zhong Y (2009) Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition, IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops), pp. 689–696.