Extended 3D Line Segments from RGB-D Data for Pose Estimation

Anders Glent Buch, Jeppe Barsøe Jessen, Dirk Kraft,
Thiusius Rajeeth Savarimuthu, and Norbert Krüger

The Maersk Mc-Kinney Moller Institute, University of Southern Denmark,
Odense, Denmark
{anbu,jeje,kraft,trs,norbert}@mmmi.sdu.dk

Abstract. We propose a method for the extraction of complete and rich symbolic line segments in 3D based on RGB-D data. Edges are detected by combining cues from the RGB image and the aligned depth map. 3D line segments are then reconstructed by back-projecting 2D line segments and intersecting this with local surface patches computed from the 3D point cloud. Different edge types are classified using the new enriched representation and the potential of this representation for the task of pose estimation is demonstrated.

Keywords: Edge detection, pose estimation.

1 Introduction

Edges are an important aspect of most visual representations, and a large body of work has been devoted to their extraction from images [1, 2] as well as from 3D data [3–5]. The latter has in particular lately been driven by the availability of cheap commodity range sensors.

Edge structures in images can be caused by different 3D structures. 3D edges can differ in terms of their *type*, their *appearance* and their *geometry*. For the *edge type* (see Fig. 1C,i), a distinction between surface edges, orientation discontinuities, and depth discontinuities is appropriate. A *geometric representation* of 3D edge points contains the 3D orientation of the edge itself as well as the position and normals of the surface patches on either side of the edge (see Fig. 1D,i-iv).

The distinction in terms of edge type, appearance and geometry is fundamentally associated to other vision problems, such as e.g. optimal optic flow estimation, since at depth discontinuities any smoothing of optic flow along the edge will lead to artifacts. It is also important for the correct associations of edges to objects and background, also known as *border ownership* [6].

This paper presents a system in which local edges—in the following called *extended 3D line segments* (ELS)—are extracted with a *rich semantic labeling* containing information on type, appearance and geometric structure. The term *extended* is motivated by the fact that the line segments described here are an extension of the 3D line segments introduced in [7], which are extracted from stereo information. In this work, we use RGB-D data and demonstrate the potential of such an enriched description for pose estimation.

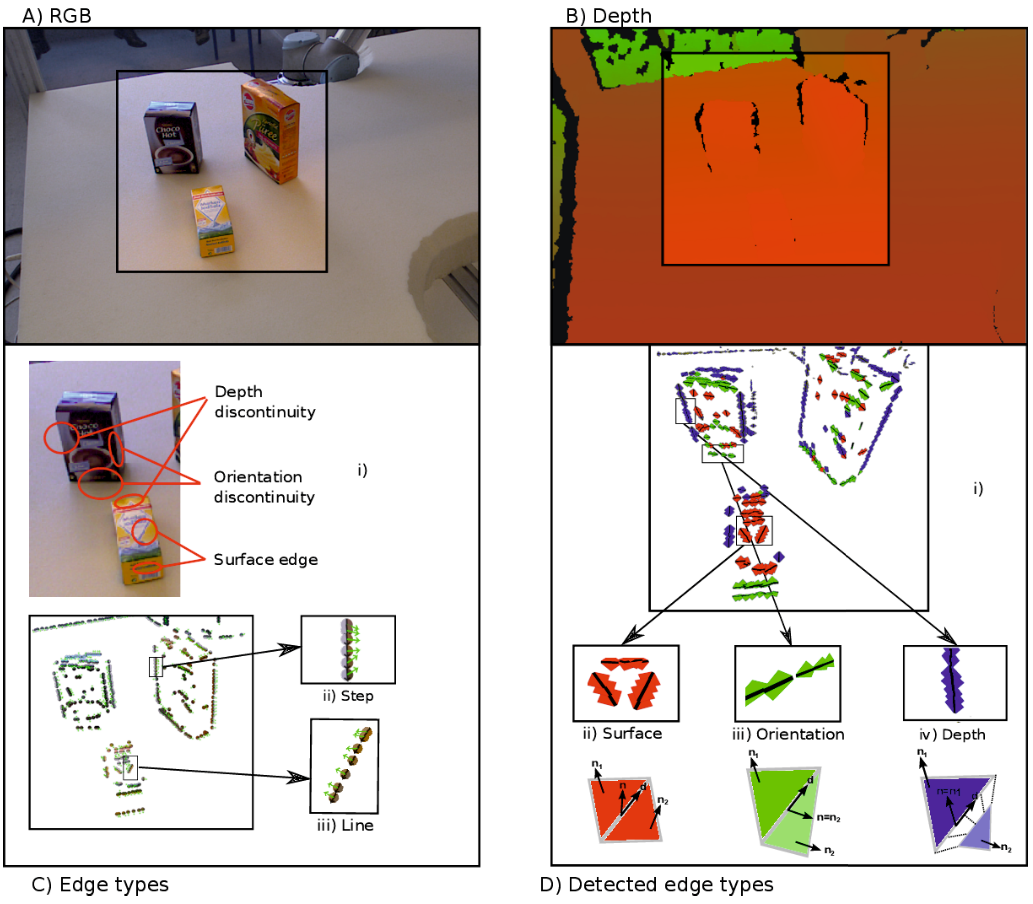

Fig. 1. Example scene with detected 2D and 3D edge types. Left: RGB image (top) with marked ground truth edge types (middle) and detected 2D edge types (bottom). Right: depth image (top), detected edge types (middle) and schematics of the associated geometries (bottom). In contrast with left part, colors are used here as indicators of the different edge types.

Edge detection in gray images or color images has been studied extensively [1, 2]. A successful algorithm for detecting edges visible in image data is the well-known Canny edge detector [8]. There are certain limitations, though. In the robotics domain, the geometric boundary edges of objects are often important; these are orientation/depth discontinuity edges, which can be nearly invisible in the image data and thus hard to detect. Typical reasons for this are color or lighting settings that result in identical appearance on each side of the edge. Edge detection based on depth information could potentially overcome this limitation.

A recent work uses a modified Canny edge detector directly on a 2D depth map to detect depth discontinuity edges in real-time [3]. The algorithm parameters are adjusted according to depth since these are correlated with the noise and uncertainty. Other approaches use local polynomial fits to 3D data directly and compute intersections or derivatives to detect orientation/depth discontinuity edges [4, 5]. In [9, 10], edge detection is performed on range data.

There have been some attempts at combining RGB and depth information to get better edge detection. The work in [11] improves the result of the Canny edge detector by dividing the RGB image according to depth layers and chooses for each layer a depth-dependent pre-smoothing frequency. This frequency is then applied in the Gaussian smoothing filter.

Our method differs from the approaches described above, since we label the edges as orientation discontinuity, depth discontinuity or surface edges, whereas [9] focus on geometric interpretations such as convex or concave edges, and [10] associates each depth discontinuity edge to an object as a border point. In contrast with [11], where 3D information is used to optimize detection, our method uses 3D structures for edge classification.

The paper is structured as follows. In Sect. 2, the different edge types are characterized, and in Sect. 3 the extraction of the ELS representation is described. Sect. 4 motivates the use of ELS for pose estimation. In Sect. 5 experimental results are presented, and finally a conclusion is given in Sect. 6.

2 A Phenomenology of 3D Edge Structures

This section discusses the semantic labeling of different edge types in more detail.

Surface Edges are characterized by a non- or smoothly changing surface normal (see Fig. 1D,ii). As shown in [12], such smooth 3D structures are often characterized by a textured or homogeneous appearance. The parameterization of a surface edge geometry consists of a 3D position and 3D edge orientation as well as a 3D surface normal (see Fig. 1C,i and Fig. 1D,ii). The appearance information is characterized by a phase value and two or three color values, depending on whether the edge structure represents a line or a step edge.

Orientation Discontinuity Edges[1] are caused by surfaces with different normals that intersect at an edge (see Fig. 1C,ii and Fig. 1D,iii). This type is not viewpoint invariant, since it can transfer into a depth discontinuity edge (see below). In the frequent case that objects have the same color on both sides of the orientation discontinuity edge (see Fig. 1C,i), the appearance difference is mainly caused by the relative orientation to the main illumination source which often leads to rather weak contrast differences.

The parametrization of the geometry of orientation discontinuity edges consists of a 3D position and 3D edge orientation as well as two 3D surface normals (see Fig. 1D,iii). The appearance information is characterized by a phase value and two or three color values as for surface edges. Note that even when the phase indicates the existence of a line structure, it is usually caused by the accidental viewing conditions and cannot be used to make any prediction about the underlying 3D structure. Hence, as a parametric description, we only associate the two colors on either side of the edge.

Depth Discontinuity Edges[2] (see Fig. 1C,i and Fig. 1D,iv) in general are associated to two different objects. One side of the edge belongs to the foreground object while the other side of the edge can be associated to the background.

[1] Also called *roof edges* [3] or *crease edges* [4].
[2] Also called *jump edges* [3, 4].

The parametrization of the geometry of a depth discontinuity edge consists of a 3D position and 3D edge orientation as well as two 3D surface normals corresponding to the two sides, where one is labeled as foreground and the other is labeled as background. The appearance information is characterized by a phase value and two or three color values as for surface edges. Note that even when the phase indicates the existence of a line structure, it is usually caused by the background and therefore does not provide relevant information for e.g. object recognition and pose estimation. Hence, as a parametric description, we do not use phase information and only represent the two colors associated to the two sides. Also, we distinguish between foreground and background which allows us to only make use of the relevant aspect of the depth discontinuity edge.

For correspondence search in relation with pose estimation, it is important to avoid using the appearance information associated to the background, since it does not reflect a property of the object. Even though this can be handled, there is still the problem that a depth discontinuity edge can be a rotated orientation discontinuity edge; this cannot be detected, leading to valuable information loss. Therefore, as described in Sect. 5, we omit this edge type during pose estimation to increase stability.

3 Extraction of Extended Line Segments

The extraction of ELS starts with image filtering operations, from which 2D primitives are generated [7, 13]. In this context, a *primitive* refers to a highly localized interest point which can be reliably extracted. During the extraction process, primitives can be instantiated into edge structures, textures or junctions, depending on the local properties of the filter responses. In this work, only primitives in the edge domain—also referred to as line segments—are considered. These line segments are reconstructed in 3D, and finally the local 3D surface characteristics are used to perform edge type classification in order to arrive at the extended description of a line segment.

3.1 Extraction of 2D Edge Primitives

2D edge primitives π^e represent short line segments extracted from a single image [7] and contain associated geometry G^e and appearance A^e: $\pi^e = (G^e, A^e)$. 2D edge primitives have an orientation that can be computed reliably. Their position can only be determined locally up to a one-dimensional manifold because of the aperture problem. The geometric information is thus described as $G^e = ((x, y), (o_1, o_2))$, where (o_1, o_2) describes the normalized orientation vector. Note that both $\pm(o_1, o_2)$ are valid orientations. Appearance information is subsequently used to disambiguate this information, giving a well-defined direction, as will be described in Sect. 3.4.

The appearance information of the line segment consists of two color triplets defining the color $\mathbf{c}_l$ on the left and the color $\mathbf{c}_r$ on the right side of the edge, and possibly a third color $\mathbf{c}_m$ on the edge itself for a line structure. Additionally, a phase $\varphi \in [-\pi, \pi[$ is computed, defining the grayscale transition:

$A^e = (\varphi, \mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$. The local phase describes the structure of the appearance information. It allows for the differentiation between step edges (e.g. transition from dark to bright) and line structures (e.g. bright line on darker background).

3.2 Local Surface Reconstruction

Based on RGB-D data and a proper camera calibration, a 3D point cloud with a 3D point associated to each pixel in the RGB image is computed. From this surface data, local surface parameters are derived. This is an important preprocessing step in order to achieve reliable reconstruction of the primitives.

The local point cloud is split according to the edge orientation. Given a 2D line segment position and orientation, one can determine on which side of the actual edge any point in the local 3D region is positioned. We introduce two parameters r and l to define regions within radius r separated from the actual line by a distance of at least l (see Fig. 2, left). This separation is required, since smoothing effects in the Kinect depth computation algorithm would lead to biased reconstruction for too small l. The set of points in the two regions are denoted as $\mathbb{P}_1$ and $\mathbb{P}_2$ and the total set of points inside r as $\mathbb{P}_{all}$.

For small r, the local surface is approximately planar, so we apply a RANSAC [14] plane algorithm for estimating the best fit plane parameters of $\mathbb{P}_{all}$, $\mathbb{P}_1$ and $\mathbb{P}_2$. As a result of this, three planes P_{all}, P_1 and P_2 with surface normals $\mathbf{n}_{all}$, $\mathbf{n}_1$ and $\mathbf{n}_2$ are obtained. The surface patch positions $\mathbf{p}_{all}$, $\mathbf{p}_1$ and $\mathbf{p}_2$ are computed as the centroids of the inlier subsets of $\mathbb{P}_{all}$, $\mathbb{P}_1$ and $\mathbb{P}_2$ that agree with the fitted plane models P_{all}, P_1 and P_2.

3.3 Edge Type Classification

This section describes how to identify the three basic edge types introduced earlier. These can be distinguished by the relation between the local planar patches on either side of the line as follows.

Distance Measure: If the Euclidean distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$ is large, it is likely that the edge is caused by a depth discontinuity.

Angular Measure: The difference in orientation is the dihedral angle computed as $\arccos(\mathbf{n}_1 \cdot \mathbf{n}_2)$. This parameter indicates whether the edge is part of a smooth surface or an orientation discontinuity.

The classification scheme is a simple decision tree, shown in Fig. 2, right. In this work the distance and angle thresholds are set to 0.035 m and 1 rad.

3.4 Association of Geometry and Appearance

The geometric parameters used for classification in the previous section are prone to instabilities, which one wants to avoid when associating information to the 3D primitives. For instance, the 3D position of a primitive should not be computed as the centroid of the point cloud, since it is affected by noise. Instead, the position

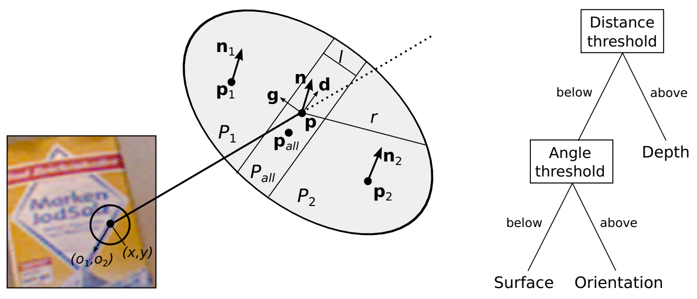

Fig. 2. Left: geometric 3D reconstruction of a 2D line segment and the parameters involved in the local surface reconstruction. A ray (solid to dashed line) is cast from the 2D pixel coordinates and intersected with either P_{all}, P_1 or P_2, depending on the edge type, in order to reconstruct the 3D position $\mathbf{p}$. For details on the parameters, see the text. Right: edge type classification scheme based on local surface patches on either side of an edge point.

is reconstructed as the intersection between the back-projected ray from the 2D primitive position and either P_{all}, P_1 or P_2, depending on the edge type. The reconstructed position, which is denoted $\mathbf{p}$, thus differs from $\mathbf{p}_{all}$. The reconstructed direction and normal of an ELS are denoted $\mathbf{d}$ and $\mathbf{n}$, and below we describe how these are calculated for the three different edge types. The parameters described in the following paragraphs are also visualized in Fig. 2, left.

Surface Edges: In this case, the edge is embedded in one surface, and the local plane orientations $\mathbf{n}_1$ and $\mathbf{n}_2$ are approximately equal. Hence, for the reconstruction of the ELS position $\mathbf{p}$, the back-projected line is intersected with P_{all}. Likewise, the intersecting line between the back-projected plane of the 2D line and P_{all} is used to calculate the 3D orientation. The normal $\mathbf{n}$ is set to $\mathbf{n}_{all}$.

In order to compute the direction $\mathbf{d}$, the mean color of all the points in both the surface patches $\mathbb{P}_1$ and $\mathbb{P}_2$, which we denote $\mathbf{c}_1$ and $\mathbf{c}_2$, is used. A *gradient direction vector* $\mathbf{g}$ pointing from the centroid of the lower intensity region to the centroid of the higher intensity region, while being orthogonal to the reconstructed orientation vector, is constructed. One then needs to check whether the reconstructed orientation together with $\mathbf{g}$ and $\mathbf{n}$ forms a right-handed coordinate system. If not, the orientation is inverted. The corrected orientation is the disambiguated 3D direction vector $\mathbf{d}$ of the ELS.

Orientation Discontinuities: For this type, the local plane orientation on each side of the line is different, but ideally the two planes P_1 and P_2 and the reconstructed orientation intersect in one line. Hence, for the reconstruction of the line segment, two planes from which one can choose to compute the geometry are available. To achieve the most stable results, the back-projected ray of the 2D position is intersected with both P_1 and P_2, and the mean is selected as $\mathbf{p}$.

The ELS normal vector $\mathbf{n}$ is now chosen as either $\mathbf{n}_1$ or $\mathbf{n}_2$, depending on which surface patch has the highest intensity. This is done to avoid taking the mean of two orientations, which is unstable. As for surface edges, the direction $\mathbf{d}$ is disambiguated using the generated gradient direction vector $\mathbf{g}$.

Depth Discontinuities: In this case, only the surface patch closest to the sensor according to $\mathbf{p}_1$ and $\mathbf{p}_2$ is considered. The gradient direction vector $\mathbf{g}$ is now constructed such that it points "inwards", i.e. from the reconstructed position $\mathbf{p}$ towards the centroid of the patch closest to the sensor. This is used to disambiguate the direction $\mathbf{d}$. Accordingly, the normal $\mathbf{n}$ is chosen as the fitted normal of the closest patch.

Associated Coordinate Systems: Fig. 3 shows an example of the disambiguated edge directions (red), gradient directions (green) and chosen normals (blue) for the ELS representation of a textured object under two different viewpoints. The figure shows some orientation discontinuity primitives at the creases, where the normal vector is calculated based on the brightest region, in this case white areas, as opposed to orange or shaded areas. Notice how some of these disappear when the crease rotates towards the sensor. We stress that the three vectors $\mathbf{d}$, $\mathbf{g}$ and $\mathbf{n}$ by construction form an orthogonal frame in $SO(3)$, which is very useful for later processing. The object shown here is taken from the RGB-D database [15], which contains example scenes and a set of objects captured with a turntable. This database is revisited in the experiments.

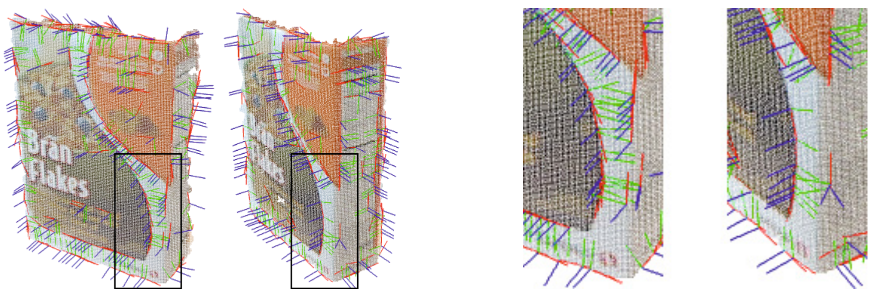

Fig. 3. Left: visualization of the calculated edge direction (red), gradient direction (green) and chosen normal (blue) vectors of a textured object seen from two views, separated by approximately 30°. Right: zoom of regions marked in leftmost part.

Representation: To summarize the preceding sections, we parameterize an ELS as follows: $\Pi^E = \left(\xi^E, G^E, A^E, f^E\right)$, with geometry $G^E = (\mathbf{p}, \mathbf{d}, \mathbf{g}, \mathbf{n})$ and appearance $A^E = (\varphi, \mathbf{c}_m, \mathbf{c}_1, \mathbf{c}_2)$. We have also introduced the type parameter $\xi^E \in \{S, O, D\}$ and the foreground indicator $f^E \in \{1, 2\}$. The type parameter is assigned the value S for surface edges, O for orientation discontinuities and D for depth discontinuities. The foreground indicator is used for D types only and indicates whether $\mathbf{c}_1$ belong to the foreground ($f^E = 1$) or the background ($f^E = 2$).

4 Pose Estimation

This section describes how the presented ELS representation can be used for pose estimation. Since the extended line segments are highly localized, they are augmented with a more descriptive context descriptor to facilitate correspondence search in the estimation process. During pose estimation, the context descriptor of each model ELS is matched with the scene and the nearest matching ELS in the scene is found. The result of context descriptor matching is a set of putative model-scene correspondences, for each of which we can compute a relative transformation as a pose candidate. Since outliers exist in correspondence space, a robust method must be applied. A popular approach to this problem is RANSAC, although Hough-like voting-based methods have also proven efficient [16]. Our method proceeds as follows:

1. **Context Descriptor Matching:** Compute the ELS representation for both the model and the scene, calculate context descriptors for each ELS and perform nearest feature matching to get a set of one to one point correspondences between the model and the scene.
2. **Pose Candidate Calculation:** For each correspondence, use the associated ELS geometry to compute a full relative $SE(3)$ transformation between the model and the scene, giving a single pose candidate.
3. **Voting-Based Estimation:** Accumulate all pose candidates in a continuous voting space. Correct correspondences should all vote for the same pose, resulting in a detectable mode in the voting space. This mode is returned as the result.

The steps are described in more detail below.

Context Descriptor Matching: In order to apply the ELS to a pose estimation scenario, each ELS is augmented with a more descriptive context descriptor based on the characteristics of the underlying model surface point cloud in the local neighborhood, constrained by an Euclidean radius. We adopt the recently proposed context descriptor described in [17], where appearance relations are calculated using simple RGB differences, and geometric relations are calculated based on surface normals. The context radius is set to 0.025 m.

Pose Candidate Calculation: The relative pose between two matched primitives is used as a single vote during estimation. This is easily calculated using the associated geometry. A rotation matrix $\mathbf{R}$ constructed by setting the tuple $(\mathbf{d}, \mathbf{g}, \mathbf{n})$ as columns, since these by construction are orthogonal and form a $SO(3)$ frame. The position directly gives the translation of the ELS relative to the intrinsic coordinate system of the model. For a single edge primitive, the model-relative $SE(3)$ transformation is thus given as the following homogeneous transformation matrix:

$$\mathbf{T}^E_{model} = \begin{bmatrix} \multicolumn{3}{c}{\mathbf{R}^E_{model}} & \mathbf{p}^E_{model} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where subscripting is used to indicate the reference frame in which the primitive is described. Likewise, when calculating primitives for a scene, a set of scene-relative primitives with local frames $\mathbf{T}^E_{scene}$ is computed. Pose estimation is now the process of finding the transformation that correctly aligns the model with the scene. For a matched model-scene primitive pair $\left(\Pi^{E_i}, \Pi^{E_j}\right)$, a candidate model-scene pose is thus calculated as:

$$\hat{\mathbf{T}}_{scene}^{model} = \mathbf{T}_{scene}^{E_j} \cdot \left(\mathbf{T}_{model}^{E_i}\right)^{-1} \tag{2}$$

Voting-Based Estimation: A voting scheme based on nonparametric kernel density estimation (KDE) is applied. In very short terms, pose space $SE(3)$ is modeled statistically using the product of a trivariate triangular kernel for translation and a pair of antipodal von Mises-Fischer distributions for rotation, parameterized by quaternions. For more details, we refer to [18]. In contrast with other voting methods, a single vote is now modeled as a continuous function or a kernel in $SE(3)$, centered around the sampled pose, with a user-specified bandwidth. These are set to 0.01 m and 0.1 rad for position and orientation, respectively. Voting is performed by calculating the weighted sum of a set of kernels, and the optimal pose is given by the mode of the resulting kernel density. This has the advantage over discretized voting spaces that boundary effects are implicitly handled. The use of this method is exemplified in the following section.

5 Experiments

Two experiments are presented to validate the described representation. The first experiment quantifies the strength of the context descriptor calculated for each ELS and makes a quantitative statement about the use of depth discontinuity edges versus surface and orientation discontinuity edges. In the second experiment, pose estimation results from a real setup are shown.

Descriptor Test. In order to validate the strength of the context descriptor attached to each ELS, a controlled experiment for evaluating the number of true correspondences between two displaced models is performed. A related study has been performed in [17], and that approach is briefly outlined in the following.

Ten randomly chosen objects from the RGB-D database [15] are considered. For each of the considered objects, the first and the 15th frame are chosen, these are displaced by approximately 30°. The ELS context descriptors for the two views are calculated, and one to one matching is performed to get a set of candidate point matches between the two views of the model. Since the two views are close, the iterative closest point [19, 20] algorithm can then be used to find the alignment transformation between them. Following this, we verify if matched points are now closely aligned by applying a distance threshold of 0.01 m. The remaining point correspondences are considered to be true. This approach can be seen as a 3D equivalent of the well-known evaluation method used for 2D descriptors [21, 22], with the difference that the complete set of true correspondences are derived here using geometric constraints only. The principle behind the test is visualized in Fig. 4.

This experiment is performed with our context descriptors as well as with state of the art descriptors in the image domain (SIFT [23] and SURF [24]) and in the shape domain (Spin images [25] and FPFH [26]). SIFT and SURF use their own keypoint detectors for locating interest points in the images. The shape descriptors are calculated for all 3D points in the model, and a spatial

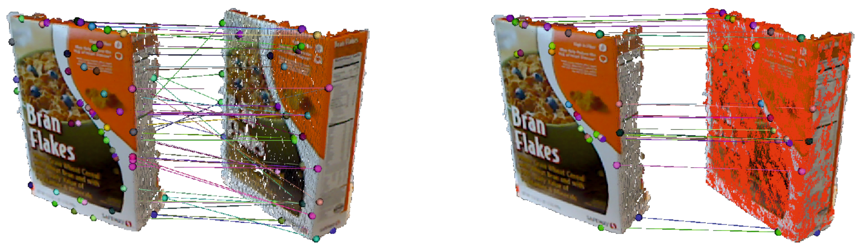

Fig. 4. Principle behind the descriptor test based on two different views of the same model. Left: initial correspondences calculated using context descriptors around ELS features. Right: alignment of the leftmost view onto the rightmost overlaid in red and remaining correspondences after Euclidean thresholding of initial correspondences.

radius of 0.025 m is used, as for our own context descriptor. The results of this experiment is shown in Fig. 5, in which we have split our representation into border points (depth discontinuity edges D) and non-border points (surface and orientation discontinuity edges S and O). The numbers in Fig. 5 verify that the use of both appearance and geometric information by our context descriptors provides superior discrimination properties when compared to other descriptors. However, it is also clear that the missing surface information for D primitives has a negative effect on the matching. Thus, these should be omitted, if possible, when matching two ELS representations.

Real World Experiment. We have also applied the pose estimation method in our own robotic setup, where we use the object pose estimation for testing different grasping techniques. The stored object ELS context models are generated from a segmented frontal view of the objects. During estimation, we calculate the ELS context descriptors for the scene and then perform KDE-based alignment. Based on the results in the previous section, only surface and orientation discontinuity edges are considered. Since the representation is very sparse, the voting is very fast, keeping estimation time below 0.05 s. The result is shown in Fig. 6.

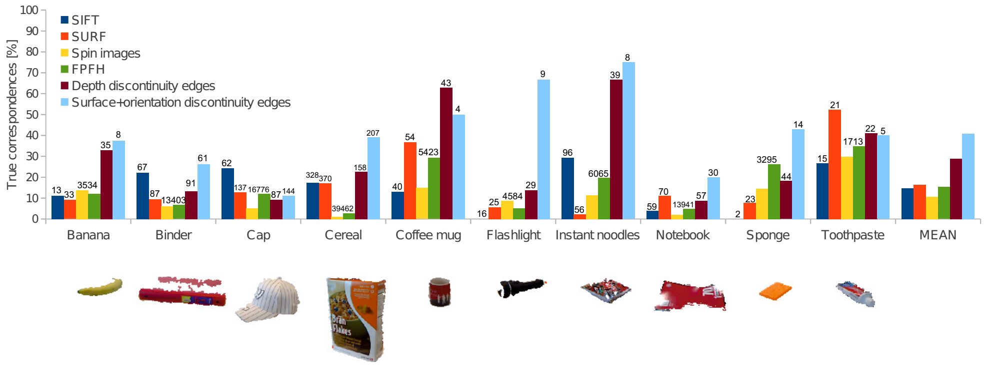

Fig. 5. True correspondence fractions for each of the individual ten RGB-D objects considered and the mean over all objects rightmost. The numbers above the bars show the number of points used in the representation, which for the two shape descriptors (Spin images and FPFH) is equal to the complete number of points in the point cloud.

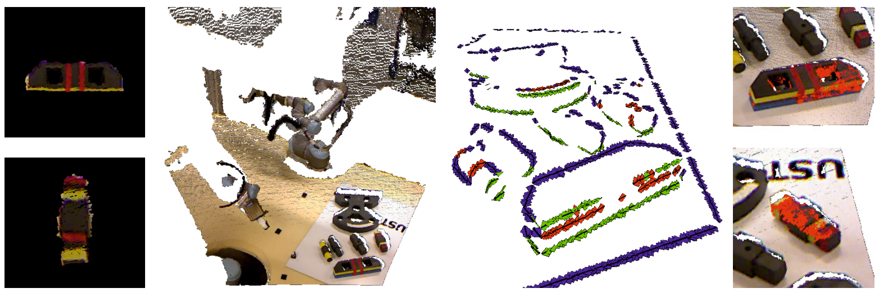

Fig. 6. Left: two captured frontal object views, stored as the object representations, and from which the ELS context representations are created. Middle: input Kinect scene (leftmost) and a zoom (rightmost) of the generated ELS features for the objects on the table, edge types indicated by the same colors as in Fig. 1. Right: pose estimation results for the two stored object views, with the alignment of the captured views overlaid in red color. Estimation time is below 0.05 s per object.

6 Conclusion

We have presented a novel system for the detection and classification of different types of edge primitives using both image data and 3D surface data. Our system is founded on image processing for the initial extraction of edges. These are reconstructed in 3D and then classified into three fundamental edge types: surface, orientation discontinuity and depth discontinuity edges. This is achieved by an edge classification scheme which uses local surface characteristics. Our edge descriptor provides valuable low-level symbolic and geometric information for a range of applications. We have successfully demonstrated the use of our representation for pose estimation in a real setup.

Acknowledgments. This work has been supported by the EC project IntellAct (FP7-ICT-269959).

References

1. Basu, M.: Gaussian-based edge-detection methods-a survey. IEEE Transactions on Systems, Man and Cybernetics 32(3), 252–260 (2002)
2. Oskoei, M.A., Hu, H.: A survey on edge detection methods. Technical Report CES-506, University of Essex (2010)
3. Lejeune, A., Piérard, S., Van Droogenbroeck, M., Verly, J.: A new jump edge detection method for 3D cameras. In: IC3D, Liège, Belgium (2011)
4. Jiang, X., Bunke, H.: Edge detection in range images based on scan line approximation. CVIU 73(2), 183–199 (1999)
5. Parvin, B., Medioni, G.: Adaptive multiscale feature extraction from range data. Computer Vision, Graphics and Image Processing, 346–356 (1989)
6. Fang, F., Boyaci, H., Kersten, D.: Border ownership selectivity in human early visual cortex and its modulation by attention. The Journal of Neuroscience 29(2), 460–465 (2009)

7. Pugeault, N., Wörgötter, F., Krüger, N.: Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision) 7(3), 379–405 (2010)
8. Canny, J.: A computational approach to edge detection. TPAMI 8(6), 679–698 (1986)
9. Al-hujazi, E., Sood, A.: Range image segmentation with applications to robot bin-picking using vacuum gripper. In: IEEE International Conference on Systems, Man, and Cybernetics (1990)
10. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Point feature extraction on 3d range scans taking into account object boundaries. In: ICRA, pp. 2601–2608 (2011)
11. Economopoulos, A., Martakos, D.: Depth-assisted edge detection via layered scale-based smoothing. In: ISPA, pp. 149–154 (2001)
12. Kalkan, S., Wörgötter, F., Krüger, N.: Statistical analysis of local 3d structure in 2d images. In: CVPR, pp. 1114–1121 (2006)
13. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. Image and Vision Computing 27, 628–636 (2009)
14. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
15. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: ICRA, pp. 1817–1824 (2011)
16. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. In: CVPR, pp. 998–1005 (2010)
17. Buch, A.G., Kraft, D., Kämäräinen, J.K., Petersen, H.G., Krüger, N.: Pose estimation using local structure-specific shape and appearance context. In: ICRA (accepted, 2013)
18. Detry, R., Piater, J.: Continuous surface-point distributions for 3D object pose estimation and recognition. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 572–585. Springer, Heidelberg (2011)
19. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. TPAMI 14(2), 239–256 (1992)
20. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. IJCV 13(2), 119–152 (1994)
21. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. TPAMI 27(10), 1615–1630 (2005)
22. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV 65(1-2), 43–72 (2005)
23. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV, vol. 2, pp. 1150–1157 (1999)
24. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
25. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. TPAMI 21(5), 433–449 (1999)
26. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: ICRA, pp. 3212–3217 (2009)