

Lifting 2D object detections to 3D: A geometric approach in multiple views

Cosimo Rubino¹, Andrea Fusiello², and Alessio Del Bue¹

¹ Visual Geometry and Modelling (VGM) Lab - Istituto Italiano di Tecnologia (IIT)
Via Morego 30, Genova, Italy

² DPIA - Università di Udine - Via Delle Scienze, 208 - Udine, Italy

Abstract. We present two new methods based on Interval Analysis and Computational Geometry for estimating the 3D occupancy and position of objects from image sequences. Given a calibrated set of images, the proposed frameworks first detect objects using off-the-shelf object detectors and then match bounding boxes in multiple views. The 2D semantic information given by the bounding boxes are used to efficiently recover 3D object position and occupancy using solely geometrical constraints in multiple views. We also combine further constraints to obtain a solution even when few images are available. Experiments on three different realistic datasets show the applicability and the potentials of the approaches.

Keywords: Object localisation, object detection, interval analysis

1 Introduction

Despite strong efforts in the Computer Vision community, object detection has been mostly restricted in 2D, even if multiple exposures of the same scene are present. In this paper we are trying to tackle instead this appealing question: “If multiple images of a rigid scene are available, is it possible to recover the 3D location and occupancy of the objects only having 2D bounding boxes returned by any object detectors?”. This question is becoming impelling given the recent advancements in object detection, boosted by the advent of deep net architectures. Indeed, it is now possible to have accurate and repeatable 2D localisations of several object class instances in generic image scenes. An approach that would be able to leverage such 2D detections in 3D would make easier the geometrical interpretation of images, nowadays necessary for applications such as human robot interaction, visual question and answering, and navigation.

Object detections are, in general, represented as 2D bounding boxes containing the object image shape. This coarse representation was dictated by annotation easiness when tracing the box while labelling large datasets as in the Pascal VOC challenge [8]. Although some works leverage finer object shape annotations (e.g. [26]), only few methods can provide a detailed silhouette of the detected objects [27, 14]. Object detections in 3D have been mainly tackled using RGBD data [16] in single images. This is a direct extension of the 2D case, where annotations are directly extracted from the depth data by using ellipsoids. Gupta et al. [13] use

labelled 3D dataset as the NYUD2 dataset [20] to retrain a region-based convolutional network (R-CCN [24]), proposing candidate 2.5D regions.

Even if these works show that it is feasible to localise 3D objects just from a single RGBD image, there are less examples showing that object localisation is possible from just image sequences without any depth information. To this extent we present two new approaches that, using geometrical reasoning only, can extract the localisation of objects in a calibrated image sequence as a set of polyhedra in 3D. The first approach is inspired by [10] and it is based on a computational geometry (CG) method which has been applied to estimate, for each object, the polyhedra given by the intersections of all the pyramids, having the vertex on each camera centre and passing through the bounding boxes of the object detections. The second approach is based on Interval Analysis (IA), and following [9] solves a similar problem based on stereo triangulations. These two methods can be readily applied to any calibrated image sequence with matched bounding boxes detections. In particular, in Sec. 4 we show results on a subset of the ScanNet dataset [6] comprising more than 1250 image sequences in realistic indoor environments. To show further the flexibility of the proposed approach in different scenarios, we also show performance on two datasets (ACCV [15] and TUW [1]) with available ground truth.

2 Related work

In this review we will restrict to single or multiple views methods for 3D object localisation, to which our approach is more closely related. As the most challenging scenario, strong efforts have been devoted to the study of single image pose estimation problems. This led to the necessity to learn image to object relations in order to generalise pose estimation in 3D to several classes of objects. In many cases a training phase is performed using images of a specific category of objects from different viewpoints. Many works have exploited 3D object models to get a 3D interpretation of the scene. Zia et al. [28] used the CAD models of cars to reconstruct the scene and the objects, including additional information about the ground plane. Pepik et al. [21] reformulated the model as a 3D deformable part model by learning the part appearances according to the CAD model. Recently, Arsalan et al. [19] used two networks to regress the orientation and the dimension of cars and bikes, then applied geometrical constraints to 2D detections to obtain the 3D bounding boxes.

When multiple images are available, recent works have tried to include geometrical reasoning to explicitly use constraints given by the multiple views. Bao et al. [3] tried to deduce both the viewpoint motion between multiple images and the pose of the objects using a part-based object detector. To reach the same goal a monocular SLAM approach was used by Dame et al. [7], combining it with shape priors-based 3D tracking and 3D reconstruction approaches, while Findler et al. [11] reduced all the objects to 3D bounding boxes with each side being a planar approximation of the object.

Differently from these methods that use strong semantics and heuristics, our approaches are based exclusively on geometrical reasoning, using directly the 2D

bounding boxes to define a polyhedra reconstruction problem, indicating where the objects are located in 3D. Unlike the Visual Hull of Laurentini [17] where the siluettes of the objects are used, we used bounding box detections as 2D input. An approach to infer the location of the objects was presented by Crocco et al. [5], estimating the occupancy of the objects through a quadric reconstruction problem. Differently to our work, they apply the simpler orthographic camera model. Furthermore, our approach is resilient if some of the detections are missing, since [5] solves the problem using the factorisation of a complete matrix containing the ellipses parametrisation for every object at every frame.

3 Lifting 2D bounding boxes to 3D

Our approach first extracts object detections from every frame of a generic image sequence. Given all the detections in each frame, we use a modified tracking-by-detection method [12] to associate the bounding boxes among different frames. This algorithm computes a distance matrix using patch appearance and associate detections using the Hungarian method for bipartite matching. We relaxed the part associated to the smoothness of the object trajectory because we might not have consistent camera motion among consecutive frames thus causing the corresponding consecutive bounding boxes to be far apart. Notice that, it is common that bounding boxes might not be precisely aligned with the true object centre and often they include a portion of background.

We then assume that the object is bounded by a rectangular region \mathcal{B}_i in image i . In 3D space, each region \mathcal{B}_i defines a semi-infinite pyramid \mathcal{Q}_i with its apex in the camera center (see Figure 1), which bounds the possible locus of the object. In the case of two views, assuming that the object's projections are bounded by rectangles \mathcal{B}_1 and \mathcal{B}_2 in the images respectively, the object in space must lie within a polyhedron \mathcal{D} as in Fig. 1. Geometrically, \mathcal{D} is obtained by intersecting the two semi-infinite pyramids defined by the two rectangles \mathcal{B}_1 and \mathcal{B}_2 and the respective centres of projection C_1 and C_2 .

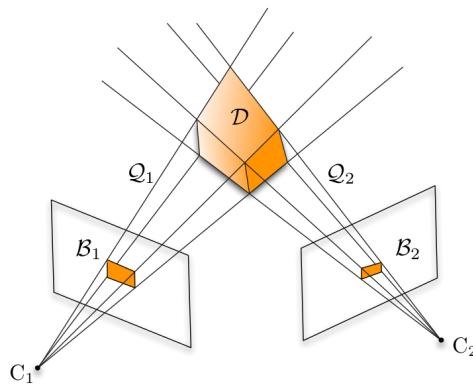


Fig. 1. Bounding the object in 3D from 2D detections. Here a graphical example with two images, where the semi-infinite pyramid is defined from the centre of projection and the bound \mathcal{B}_i .

In the general case of n views, the object is localised inside the polyhedron formed by the intersection of the n semi-infinite pyramids generated by the rectangles $\mathcal{B}_1, \dots, \mathcal{B}_n$:

$$\mathcal{D} = \mathcal{Q}_1 \cap \mathcal{Q}_2 \cdots \cap \mathcal{Q}_n. \quad (1)$$

Analytically, the polyhedron \mathcal{D} is defined as the following set:

$$\mathcal{D} = \{X \in \mathbb{R}^3 : \exists x_i \in \mathcal{B}_i, i = 1 \dots n \text{ s.t. } \forall i : x_i = \Pi_i(X)\} \quad (2)$$

where Π is the known perspective projection onto the i -th image.

3.1 Vertex enumeration solution

The semi-infinite pyramid \mathcal{Q}_i can be written as the intersection of the four negative half-spaces $\mathcal{H}_1^i, \mathcal{H}_2^i, \mathcal{H}_3^i, \mathcal{H}_4^i$ defined by its supporting planes. Thus, the solution set D can be expressed as the intersection of $4n$ negative half-spaces:

$$\mathcal{D} = \bigcap_{\substack{i=1 \dots n \\ \ell=1 \dots 4}} \mathcal{H}_\ell^i. \quad (3)$$

Implicitly these equations represent the polyhedron \mathcal{D} , and indeed this is also called the H-representation of \mathcal{D} . However, we aim at an explicit description of \mathcal{D} in terms of vertices and edges, also called a V-representation. The problem of producing a V-representation from an H-representation is called the VERTEXENUMERATION problem, in Computational Geometry. The vertices and the faces of \mathcal{D} can be enumerated in $O(n \log n)$ time, being n the number of cameras [22]. In particular we used the implementation of the reverse search vertex enumeration algorithm described in [2] and available on the web³.

In the following, this approach based on Computational Geometry (proposed in [10]) will be referred to as the “CG approach”. In the next section, following [9], we shall describe how the solution set can be enclosed with an axis-aligned box using an approach based on *Interval Analysis*, henceforth dubbed “IA approach”.

3.2 Bounded Computational Geometry method

The polyhedron generated by the CG approach can approximate effectively the 3D volume occupied by a detected object if several images of the object with a large baseline between cameras are available. Otherwise, when there are few images with a narrow baseline between cameras, the computed polyhedron can easily overestimate the occupancy volume. To reduce this effect, we bounded the estimated volume by including a prior over its maximum elongation. This is done by first finding the centroid of the object using triangulation between the centres of the bounding boxes in different views [4]. Then, the final polyhedron is obtained by cutting the pyramid, generated by CG with two planes, with a distance before and after the object 3D centroid equal to half of the maximum size of the object⁴, and with the normal aligned to the optical axis of the camera. We will henceforth refer to this variation as the CG_b method.

³ <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>

⁴ An upper bound for the size of several object classes has been extracted from the ShapeNet dataset: <https://www.shapenet.org/>.

3.3 Interval Analysis

Interval Analysis [18] is an arithmetic defined on intervals, rather than on real numbers. It was firstly introduced for bounding the measurement errors of physical quantities for which no statistical distribution was known. In the sequel of this section we shall denote intervals with boldface. Underscores and overscores will represent respectively lower and upper bounds of intervals. \mathbb{IR} stands for the set of real intervals. If $f(x)$ is a function defined over an interval \mathbf{x} then $\text{range}(f, \mathbf{x})$ denotes the range of $f(x)$ over \mathbf{x} .

If $\mathbf{x} = [\underline{x}, \bar{x}]$ and $\mathbf{y} = [\underline{y}, \bar{y}]$, a binary operation between \mathbf{x} and \mathbf{y} is defined in interval arithmetic as:

$$\mathbf{x} \circ \mathbf{y} = \{x \circ y \mid x \in \mathbf{x} \wedge y \in \mathbf{y}\}, \forall \circ \in \{+, -, \times, \div\}.$$

Operationally, interval operations are defined by the min-max formula:

$$\mathbf{x} \circ \mathbf{y} = [\min \{\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y}\}, \max \{\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y}\}] \quad (4)$$

Interval division \mathbf{x}/\mathbf{y} is undefined when $0 \in \mathbf{y}$.

In general, for arbitrary functions, interval computation cannot produce the exact range, but only approximate it.

Definition 1 (Interval extension [23]). A function $\mathbf{f} : \mathbb{IR} \rightarrow \mathbb{IR}$ is said to be an interval extension of $f : \mathbb{R} \rightarrow \mathbb{R}$ provided that $\text{range}(f, \mathbf{x}) \subseteq \mathbf{f}(\mathbf{x})$ for all intervals $\mathbf{x} \subset \mathbb{IR}$ within the domain of f .

Such a function is also called an *inclusion function*. So, given a function f and a domain \mathbf{x} , the inclusion function yields a rigorous bound (or enclosure) on $\text{range}(f, \mathbf{x})$. This property is particularly suited for error propagation: If \mathbf{x} bounds the input error on the variable x , $\mathbf{f}(\mathbf{x})$ bounds the output error. Therefore, if the exact value is contained in interval data, the exact value will be contained in the interval result.

Definition 2 (Natural interval extension [23]). Let us consider a function f computable as an arithmetic expression \mathbf{f} , composed of a finite sequence of operations applied to constants, argument variables or intermediate results. A natural interval extension of such a function, denoted by $\mathbf{f}(\mathbf{x})$, is obtained by replacing variables with intervals and executing all arithmetic operations according to the rules above.

Please note how different expressions for the same function yield different natural interval extensions. For instance, $\mathbf{f}_1(\mathbf{x}) = \mathbf{x}^2 - \mathbf{x}$, and $\mathbf{f}_2(\mathbf{x}) = \mathbf{x}(\mathbf{x} - 1)$ are both natural interval extensions of the same function. For example, consider the expression $f(x) = x - x$ which is equivalent to 0. However evaluating the expression with the interval $[1, 2]$, gives $\mathbf{f}([1, 2]) = [1, 2] - [1, 2] = [-1, 1]$, because the piece of information that the two intervals represent the same variable is lost. In general, although the ranges of interval arithmetic operations are exact, this is not so if operations are composed. For example, if $\mathbf{x} = [0, 1]$ we have $\mathbf{f}_2(\mathbf{x}) = [0, 1]([0, 1] - 1) = [0, 1][-1, 0] = [-1, 0]$, which strictly includes $\text{range}(f, [0, 1]) = [-1/4, 0]$.

It is well-known that Interval Analysis systematically overestimates the bound on the results of a computation: this is the price to pay for its simplicity.

3.4 Interval-based triangulation

Let us assume that we can write a closed form expression that relates the 3D point X to its projections $x_1 = \Pi_1(X)$ and $x_2 = \Pi_2(X)$ in two images (see [9]):

$$X = f(x_1, x_2) \quad (5)$$

If we let x_1 and x_2 in Eq. (5) vary in \mathcal{B}_1 and \mathcal{B}_2 respectively, then $\text{range}(f, \mathcal{B}_1 \times \mathcal{B}_2)$ describes the polyhedron \mathcal{D} that contains the object. Interval Analysis gives us a way to compute an axis-aligned bounding box containing \mathcal{D} by simply evaluating $f(\mathbf{x}_1, \mathbf{x}_2)$, the natural interval extension of f , with $\mathcal{B}_1 = \mathbf{x}_1$ and $\mathcal{B}_2 = \mathbf{x}_2$.

The 3D interval $f(\mathbf{x}_1, \mathbf{x}_2)$ encloses the polyhedron \mathcal{D} , and, in general, it is an overestimate. In fact, intervals can model only axis-aligned rectangular boxes; moreover, as seen in the examples, interval evaluation inevitably introduces overestimation.

The approach is easily extensible to the general n -views case. As defined in Sec. 3, the sought polyhedron \mathcal{D} is formed by the intersection of the semi-infinite pyramids generated by back-projecting in space the sets $\mathcal{B}_1, \dots, \mathcal{B}_n$. Thanks to the associativity of intersection, \mathcal{D} can be obtained by first intersecting pairs of such pyramids and then intersecting the results. Let $\mathcal{D}_{i,j}^2$ be the solution set of the triangulation between view i and view j . Then:

$$\mathcal{D} = \bigcap_{\substack{i=1, \dots, n \\ j=i+1, \dots, n}} \mathcal{D}_{i,j}^2. \quad (6)$$

An enclosure of the solution set \mathcal{D} is obtained by intersecting the $n(n-1)/2$ enclosures of $\mathcal{D}_{i,j}^2$ computed with the IA method described above. Since each enclosure contains the respective solution set $\mathcal{D}_{i,j}^2$, their intersection contains \mathcal{D} . In summary, the IA approach yields a rectangular axis-aligned bounding box $f(\mathbf{x}_1, \mathbf{x}_2)$ that contains the polyhedron \mathcal{D} . This method is faster and easier to implement (basing on an interval arithmetic library, such as INTLAB [25]) than the CG one, but the enclosure is – in general – an overestimate.

4 Experiments

We tested our methods on three datasets: ACCV [15], TUW [1] and ScanNet[6]. These datasets present different imaging conditions related to camera motion, number of frames for each sequence, number of objects and their distance from the camera. In total we tested 1240 different image sequences with an overall number of 42,000 frames.

All the datasets provide the camera parameters and the annotated ground truth (GT) point clouds of the objects inside the scene. For each object, we evaluated the GT 3D bounding box by enclosing the given 3D point clouds. For each frame and each object we also generated a set of 2D bounding boxes to simulate the output of an object detector. This is done by fitting with a box the 2D reprojections of the labelled point clouds associated to each object. Additionally, we have also evaluated oriented bounding boxes, by aligning the box with respect to the orientation of the objects onto the 2D image frames. The alignment is performed by

considering the orientation of an image mask associated to the reprojected points, returned by the function *regionprops* in MATLAB.

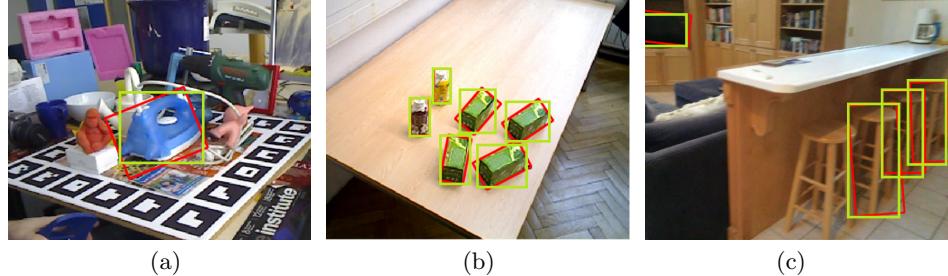


Fig. 2. A frame with oriented bounding boxes (red) and bounding boxes aligned to the axes (green) of Seq. “Iron” of the ACCV dataset, Seq. 7 of the TUW dataset and scene0000 of the ScanNet dataset.

Results have been evaluated by computing the 3D Intersection over Union (IoU) between the bounding boxes associated to the GT and to the reconstruction. Our methods perform very well for the ACCV dataset since the sequences have a high number of images taken from a camera that performs a large rotation around the objects. Differently, the TUW and ScanNet datasets have a reduced number of frames and a limited motion of the camera, thus reducing drastically the performance of the proposed methods. The computational costs of both methods can be deduced by [10] and [9].

4.1 ACCV dataset evaluation

The ACCV dataset [15] contains 15 sequences, each of them depicting a single object laying on a table at different camera viewpoints (from 100 to 1000 per sequence). We used 9 sequences for which the 3D point cloud of the object is provided, and limited the number of views to 100 for each sequence. The number of views and the motion of the camera affects positively the CG approach, as shown in the left image of Fig. 3: The larger the angle spanned by the viewpoints around the object, the better the performances of the method. As shown in Tab. 1, the results are remarkable for CG, with an average IoU of 0.85. Unlike the CG one, the IA approach does not reach high results in term of IoU (average IoU: 0.37) because of its tendency to overestimate the volume, as can be seen in Fig. 3 and as already explained in Sec. 3.3. Tab. 2 shows results using oriented bounding boxes with an average IoU of all the sequences similar to the average IoU given by bounding boxes aligned to the image axis. By analysing each sequence, there is a net improvement in the “Driller” and “Can” because the oriented bounding boxes can describe better objects with an anisotropic shape.

4.2 TUW dataset evaluation

The TUW dataset [1] contains 15 annotated sequences showing a table with different sets of objects deployed on it. The number of frames per sequence ranges from 6 to 20, therefore fewer frames are available with respect to the ACCV dataset. Moreover, the objects in the images are not centred in the 3D scene as in the previous case.

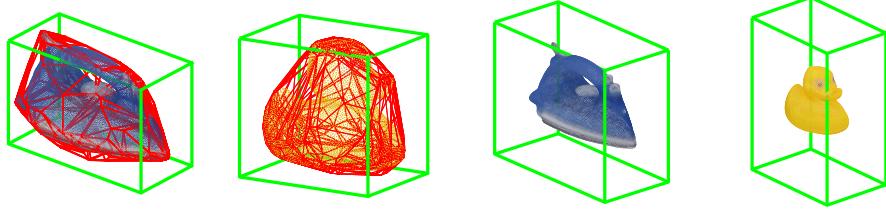


Fig. 3. Results for 2 ACCV sequences. In the figures we show the GT point clouds of the objects and in green the estimated 3D bounding box. On the left are displayed the results by using the CG, where the red wire-frame represents the estimated polyhedron. On the right are shown the results of the IA method.

Table 1. Estimated IoU for 9 sequences from ACCV dataset, by using CG and IA with bounding boxes aligned to the image axis.

	Iron	Duck	Ape	Can	Driller	Vise	Glue	Cat	Lamp	Avg.
IA	0.34	0.14	0.27	0.39	0.33	0.63	0.50	0.18	0.53	0.37
CG	0.81	0.85	0.89	0.87	0.77	0.85	0.90	0.75	0.87	0.84

Table 2. Estimated IoU for 9 sequences from ACCV dataset, by using CG with bounding boxes aligned to the point cloud reprojections.

	Iron	Duck	Ape	Can	Driller	Vise	Glue	Cat	Lamp	Avg.
CG	0.80	0.84	0.73	0.92	0.90	0.82	0.93	0.70	0.87	0.83

We used both the CG and IA on these sequences, and the results are displayed in Tab. 4. In this case, it is clear a drop of performance for the CG approach, on average 0.27, while the IA approach fails to provide usable localisations by overestimating the volume when there are few frames available.

We also performed an evaluation by considering the 2D bounding boxes aligned with the objects and we also evaluated the performance of the CG_b method. As expected, the results (reported in Tab. 4) outperform the original CG method in terms of IoU, reaching an average precision of 0.40. Indeed, if few frames are present, the constrain on the volume of the polyhedra is fundamental for not obtaining excessively overestimated volumes.

4.3 ScanNet dataset

ScanNet is a RGB-D dataset of real-world indoor environments proposed by [6] and it is the most challenging tested dataset. ScanNet main advantage is the high number of annotated sequences, 1513 in total. This dataset provides, for each sequence, all the camera parameters and a dense 3D reconstruction of the environment. Several objects and regions in the 3D point cloud are labelled, thereby providing ground truth for object localisation and occupancy estimation. We se-

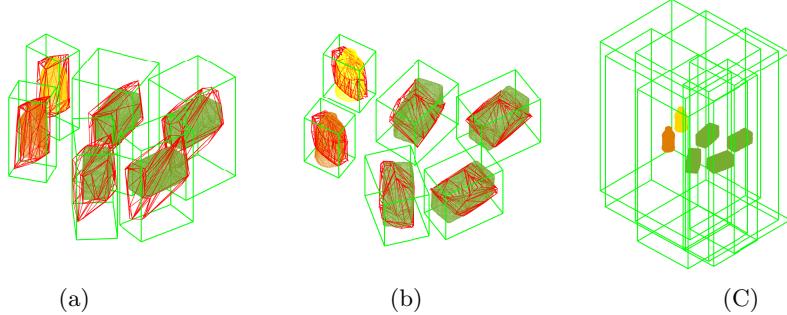


Fig. 4. Results for Seq. 7 of the TUW dataset. In the figures we show the GT point clouds of the objects and in green the estimated 3D bounding box. In Fig. (a) is displayed the result by using the CG, where the red wire-frame represents the estimated polyhedron. In Fig. (b) is shown the results of the CG_b with oriented bounding boxes, while in Fig. (c) the estimation performed by using the IA [9] method.

Table 3. Estimated IoU for 15 sequences from TUW dataset, by using CG and IA with bounding boxes aligned to the image axis.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Avg
IA	0.00	0.00	0.03	0.02	0.00	0.01	0.01	0.00	0.00	0.02	0.01	0.01	0.01	0.01	0.03	0.01
CG	0.17	0.05	0.53	0.25	0.32	0.29	0.23	0.25	0.12	0.41	0.24	0.23	0.23	0.38	0.33	0.27

Table 4. Estimated IoU for 15 sequences from TUW dataset, by using CG_b with bounding boxes oriented to the point cloud reprojections.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Avg
CG_b	0.45	0.09	0.34	0.37	0.39	0.33	0.42	0.57	0.28	0.57	0.43	0.45	0.39	0.33	0.52	0.40

lected a subset of 1215 image sequences that have a minimum of 3 frames. We also did not consider all the sequences with a poor estimation of the motion of the camera, which can heavily affect objects localisation.

In this case the average results for the CG method is 0.04, while IA fails the reconstruction. The reason of this poor performance is mainly due the short baseline and small camera rotation. We also applied the CG_b to the ScanNet dataset by considering as input the oriented 2D bounding boxes. since the inclusion of two extra planes in the CG_b helps to limit the volume of the reconstruction as can be seen in Fig.5(b), especially when motion of the camera is reduced and the polyhedron computed by the CG is unlimited, as in Fig.5(a). In Fig. 5 we included some statistical information about the estimations, like the distribution among the sequences of the IoU results by using both the CG 6(a) and with the CG_b 6(b) approaches.

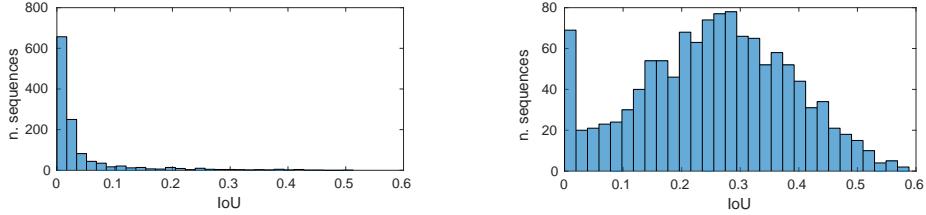


Fig. 5. Distribution of the IoU results for the 1215 selected sequences of the ScanNet dataset with the CG (left) and with the CG_b (right) approaches.

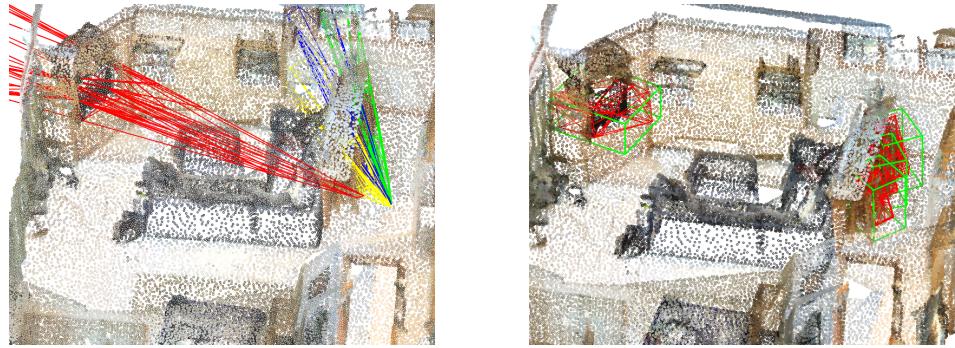


Fig. 6. Results for scene0000 of the ScanNet dataset. On the left is displayed the reconstruction by using only the CG approach, with the polyhedrons coloured differently to distinguish each reconstructed objects; on the right the estimation by using oriented bounding boxes and the CG_b approach, with the estimated polyhedrons in red and the associated bounding box in green.

5 Conclusion

We have presented two approaches based on two already existing methods to perform the localisation (position and occupancy) of detected objects by using as input 2D bounding boxes associated to the objects and the camera parameters. Extensive experiments on real datasets confirm that the problem of estimating 3D localisation and occupancy from 2D bounding boxes is solvable. Between the two proposed approaches, IA tends to overestimate the enclosure with respect to CG. It is also clear that higher performance are obtained with higher number of frames and camera motion. Further improvements can still be obtained by including more data-driven priors about the surrounding environment and on the objects sizes and appearance. In particular, the ScanNet dataset performance can be further improved, representing a new challenge for the community.

References

1. Aldoma, A., Faulhammer, T., Vincze, M.: Automation of ground truth annotation for multi-view rgb-d object instance recognition datasets. In: IROS (2014)
2. Avis, D.: A revised implementation of the reverse search vertex enumeration algorithm. In: Polytopes — Combinatorics and Computation, pp. 177–198 (2000)

3. Bao, S.Y., Xiang, Y., Savarese, S.: Object co-detection. In: ECCV. Springer (2012)
4. Byr  d, M., Josephson, K.,   str  m, K.: A column-pivoting based strategy for monomial ordering in numerical gr  bner basis calculations. In: ECCV (2008)
5. Crocco, M., Rubino, C., Del Bue, A.: Structure from motion with objects. In: CVPR. pp. 4141–4149. IEEE (2016)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nie  ner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. arXiv (2017)
7. Dame, A., Prisacariu, V.A., Ren, C.Y., Reid, I.: Dense reconstruction using 3d object shape priors. In: CVPR. pp. 1288–1295. IEEE (2013)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (Jun 2010)
9. Farenzena, M., Fusello, A., Dovier, A.: Reconstruction with interval constraints propagation. In: CVPR. pp. 1185–1190 (2006)
10. Farenzena, M., Fusello, A.: Stabilizing 3D modeling with geometric constraints propagation. Computer Vision and Image Understanding 113(11), 1147–1157 (2009)
11. Fidler, S., Dickinson, S., Urtasun, R.: 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In: NIPS. pp. 611–619 (2012)
12. Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R.: 3d traffic scene understanding from movable platforms. PAMI 36(5), 1012–1025 (2014)
13. Gupta, S., Girshick, R., Arbel  ez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: ECCV. pp. 345–360 (2014)
14. He, K., Gkioxari, G., Doll  r, P., Girshick, R.: Mask r-cnn. arXiv (2017)
15. Hintersto  sser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: ACCV, pp. 548–562. Springer (2012)
16. Kim, B.s., Xu, S., Savarese, S.: Accurate localization of 3d objects from rgb-d data using segmentation hypotheses. In: CVPR. pp. 3182–3189 (2013)
17. Laurentini, A.: The visual hull concept for silhouette-based image understanding. TPAMI 16(2), 150–162 (1994)
18. Moore, R.E.: Interval Analysis. Prentice-Hall (1966)
19. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. arXiv (2016)
20. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
21. Pepik, B., Gehler, P., Stark, M., Schiele, B.: 3d2pm–3d deformable part models. In: ECCV, pp. 356–370. Springer (2012)
22. Preparata, F.P., Shamos, M.I.: Computational Geometry. An Introduction, chap. 2, pp. 72–77 (1985)
23. R.B.Kearfott: Rigorous Global Search: Continuos Problems. Kluwer (1996)
24. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
25. Rump, S.: INTLAB - INTerval LABoratory. In: Developments in Reliable Computing, pp. 77–104. Kluwer Academic Publishers (1999)
26. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV 77(1), 157–173 (2008)
27. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: ICCV (2015)
28. Zia, M.Z., Stark, M., Schindler, K.: Towards scene understanding with detailed 3d object representations. IJCV 112(2), 188–203 (2015)