3D Object Recognition in Range Images Using Visibility Context

Eunyoung Kim and Gerard Medioni

Abstract— **Recognizing and localizing queried objects in range images plays an important role for robotic manipulation and navigation. Even though it has been steadily studied, it is still a challenging task for scenes with occlusion and clutter.**

We present a novel approach to object recognition that boosts dissimilarity between queried objects and similar-shaped background objects in the scene by maximizing use of the visibility context. We design a new point pair feature containing discriminative description inferred from the visibility context. Also, we propose a pose estimation method that accurately localizes objects using these point pair matches. Finally, two measures of validity are suggested to discard false detections.

With 10 query objects, our approach is evaluated on depth images of cluttered office scenes captured from a real-time range sensor. The experimental results demonstrate that our method remarkably outperforms two state-of-the-art methods in terms of recognition (recall & precision) and runtime performance.

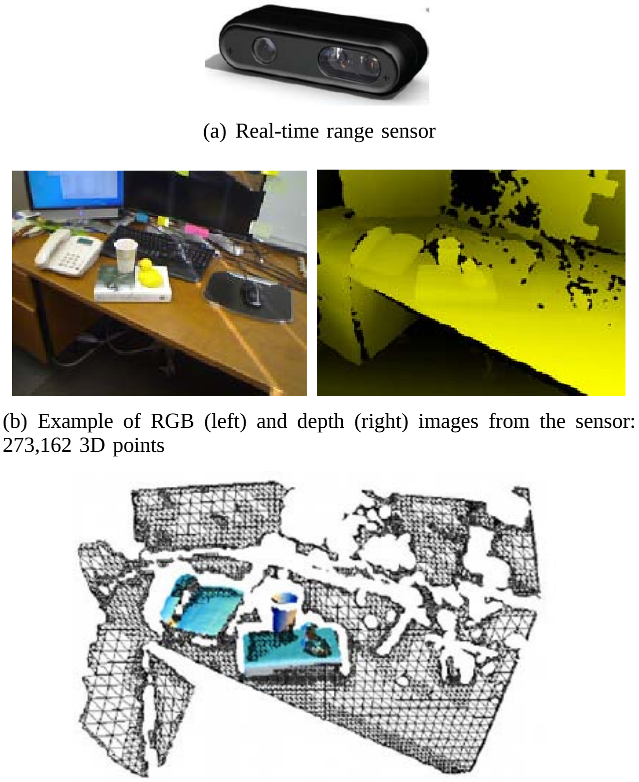
I. Introduction

Recent advanced real-time range sensors, such as the Primesensor in Fig. 1(a), enable us to obtain high resolution (640×480) depth images in real-time (up to 30 fps). These sensors began to attract the attention of many researchers in robotics, as they are small and light enough to be easily mounted on a robot and provide accurate and dense depth measurements for near objects, along with a registered RGB image.

The goal of our work is to develop a 3D object recognition system using range images acquired from the Primesensor for various applications such as SLAM and robotic manipulation. It focuses on identifying and localizing a queried free-form object by comparing its shape property to the visible surface of objects in the scene. Our method is designed to only utilize depth information because of the need for robots to work even under bad or no illumination (*e.g.* rescue or night operation).

To achieve the goal with high precision, we use a point pair feature with the *visibility context*.

Fig. 1(b) shows an example depth image from the sensor. Besides the queried objects (*e.g.* book and cup), a scene may contain a number of unknown background clutter objects (e.g. desk, monitor and keyboard) whose partial shape is very similar to the queried objects. This results in low precision rate for shape-based methods, as recall rate increases. In range image processing, however, we found that the *visibility context* can serve to discard false matches.



(a) Real-time range sensor

(b) Example of RGB (left) and depth (right) images from the sensor: 273,162 3D points

(c) Recognized objects aligned to the scene

Fig. 1. Example of the output of our method: queried objects (book, phone, cup, desk) are registered to the scene (blue mesh)

The major distinction between range images and typical 3D data is that a range image only captures the *visible* surfaces of objects *w.r.t.* viewpoints. This property makes object recognition in range images challenging, but it allows us to identify the visibility context in the given scene (*i.e.* where is visible space/surface/invisible space). In the paper, we present a new method to reduce spurious matches by encoding the additional properties of a queried object (*e.g.* dimension and convexity&concavity) inferred from the *visibility context* into feature descriptors.

A local descriptor based approach has been favored for 3D object recognition in cluttered environments [3], [13], [14]. The basic idea of these approaches is to estimate a local feature from a given point and its neighbors, and find matches having similar descriptor in the database. The matching performance of these descriptors, however, is very sensitive to descriptor scale that determines the neighborhood. It is hard to find a proper scale large enough to extract distinctive model descriptors, but also small enough to make scene

E. Kim is a Ph.D student in Computer Science Department, University of Southern California, Los Angeles, CA, USA `kimeunyo@usc.edu`

G. Medioni is a professor in Computer Science Department, University of Southern California, Los Angeles, CA, USA `medioni@usc.edu`

descriptors less distracted by clutter [6], [15]. Some methods are also affected by density of points [10], [20].

To resolve this issue, a point pair feature was introduced in [6] to model objects using all possible point pairs. Unlike other local descriptors that estimate a single feature from a given point and its neighbors, Drost *et al*. [6] represents the surface geometry around the point by the number of point pair features. It shows better and faster recognition performance against occlusion, noise and clutter than descriptor-based approaches. Also, a point pair feature is suitable to capture the visibility context.

Given a range image, we first sample 3D points in multi-resolution, preserving the original structure. Then, we compute the feature descriptor for every point pair in the scene. Each scene point pair is matched with model pairs by the extracted feature, and we compute potential object poses from all the matches. Finally, the pose candidate with the largest consensus is considered as a true detection.

The contributions of our paper are summarized below:

- We underline the importance of the visibility context in object recognition, and propose a novel feature that utilizes the visibility context.
- We present a new method to compute a transformation matrix from two oriented point pairs. Our method estimates pose more accurately than existing methods.
- We propose two measurements to verify the detection.

Fig. 1(c) shows an example output of our method. Given a depth image, four queried objects are identified and registered into the scene using the estimated poses. Note that we do not use any surface registration methods like ICP to refine the estimated poses, and there is no constraint on rotation.

With extensive quantitative analysis and comparison, we demonstrate that the visibility context enhances performance of 3D free-form object recognition using range images.

The outline of the paper is as follows: Section II summarizes related work. Section III describes a preprocessing step prior to our recognition process. Section IV defines the concept of visibility context. A new point pair feature imposing visibility context is proposed in Section V. The details of our recognition process using point pair features are explained in Section VI. Finally, experimental results are demonstrated in Section VII, followed by concluding remarks.

II. Related Work

There are two main approaches to shape-based recognition: global description and local description. Global description has been popularly used for content-based 3D object retrieval [19], [22], but it is rarely used in range images, since it requires object segmentation, which is difficult in clutter. GRSD [4] and VFH [18] efficiently recognize objects using range images, but they assume that scenes have light clutter and no occlusion, and require object segmentation prior to recognition.

The most popular approach to free-form object recognition is to extract 3D local shape descriptors from the point cloud, which characterize shape properties *w.r.t.* each oriented point, and recognize objects in the scene by matching them with known objects.

Splash [21] captures the distribution of normal orientations around the reference point, and Spin image [10] represents, given a reference point p, a 2D histogram of (α, β) coordinates of its neighbors, where (α, β) coordinate spans around the orientation of the point p.

Spherical spin image [17], 3D shape context [7], normal-based signatures [13], surface patch representation [5], tensor-based representation [14], 3D Surf [11], scale-invariant recognition [3], boundary based feature NARF [20] are also proposed to improve recognition performance against occlusion and clutter. These methods have rarely been evaluated on range images containing significant background clutter. Furthermore, these local shape features are very sensitive to irregular density of 3D points.

[16] and [12] tested on range images with large background, but these also rely on intensity images.

Recently, Drost *et al*. [6] introduced a method which extracts global description from oriented point pair features in a given model, and matches the model locally. They showed that their approach outperforms other state-of-the art methods [10], [14].

Inspired by [6], we improve recognition performance by elaborating on feature description and pose estimation with the visibility context. As a result, we exhibit better recognition and runtime performance than [6] and [10].

III. Initial Step: Surface Point Refinement

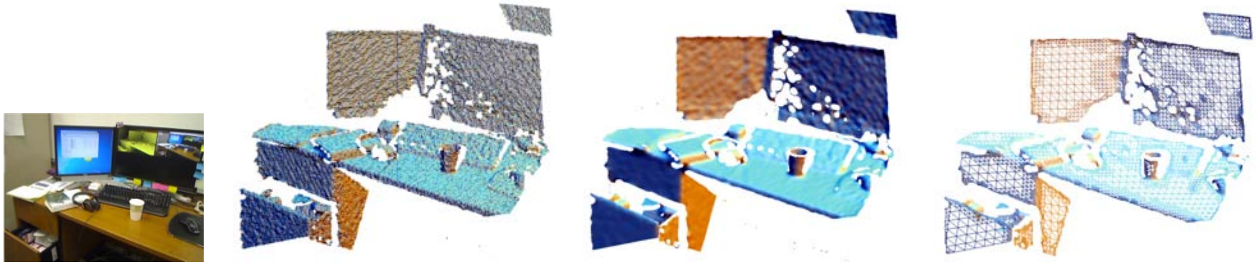
A. Surface normal estimation

For object recognition, we have to infer surface orientation for every point. Using the image coordinate corresponding to every 3D point, surface orientation inference can be easily achieved by constructing a mesh in 2D. However, range images acquired from real-time range sensors have aliasing in depth. So, in order to estimate accurate surface orientation, it is necessary to smooth the surface points. For surface smoothing, we tested some state-of-the-art methods including Algebraic Point Set Surfaces (APSS) which is based on the local fitting of algebraic spheres [8], [9], and selected Laplacian smoothing approach that calculates the average position with nearest points iteratively for each point, since it performed as well as APSS, but much faster in our experiments. Fig. 2(c) shows an example result.

B. Multi-resolution Point Sampling

As shown in Fig. 2(b), because a range image contains very dense 3D points, it is required to sample the least number of points well enough to represent the original surface characteristic for efficient processing.

We develop a simple approach. Let M_σ be a mesh generated by sampling points every σ pixel. We construct four different resolution meshes, M_2, M_4, M_8, M_{16}. Then, for every face in M_{16}, if the surface orientation of every vertex composing the face coincides with one of the corresponding points in the finer mesh M_2,M_4 and M_8, respectively, the face is selected. All faces in the finer



(a) Scene (b) Original mesh (287, 692 points) (c) Smoothed surface (d) Simplified mesh (5,285 points)

Fig. 2. Surface point refinement: Surface smoothing and point sampling

meshes overlapping with the selected face are discarded. This process is iteratively applied to M_8 and M_4. Finally, the points associated with the selected faces are sampled for our recognition process. An example simplified mesh is displayed in Fig. 2(d).

IV. Visibility Context

Space can be categorized into three different types in terms of visibility: Visible space S_V, Surface S_S and Invisible space S_o. The intuition behind this is that, if a surface (S_S) is visible from a given arbitrary view point, space in front of the surface is also visible (S_V), while space behind the surface is invisible by occlusion (S_o).

The same visibility context is applied to depth images captured from real-time range sensors. Let R be a given range image. Every pixel in image R has a corresponding depth and a 3D point in the world coordinate. Given any 3D point p and a projective matrix P, the point p can be projected into the image coordinate system (u, v) with depth d_p $(d_p = p_z)$. Then, the visibility type of the point p is:

- $p \in S_V$, if $d_p - d_{R(u,v)} \leq -\tau_d$
- $p \in S_S$, if $\|d_p - d_{R(u,v)}\| \leq \tau_d$
- $p \in S_O$, if $d_p - d_{R(u,v)} \geq \tau_d$ or no depth

where $d_{R(u,v)}$ is the depth at pixel (u, v) in R. τ_d is a tolerance related to inaccuracy in depth measurements, and was set to $1\,cm$ in our experiments.

In our paper, the visibility context plays important role in increasing discriminant power in feature description (Sec. V) and discarding invalid poses (Sec. VI).

V. Point Pair Feature

Given two oriented 3D points m_1 and m_2, a point pair feature is defined as:

$$F(m_1, m_2) = (||d||, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)),$$

where $d = m_2 - m_1$ as illustrated in Fig. 3.

The value of each coordinate is mapped into an integer value and these values are used as a key to find a set of corresponding point pairs extracted from models. From every match, the potential pose of the matched model in the scene is computed and voted for final recognition.

The main drawback of this feature is computational complexity. When a range image has a huge number of 3D points and many objects share similar shapes (*e.g.* desks and books), both the number of the features extracted from the scene and the number of the matched model pairs increased. This eventually leads to expensive computation as well as many false positives.

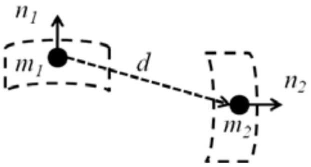


Fig. 3. Illustration of one point feature of a book model

Thus, we propose a new point pair feature that contains description inferred from the visibility context. The visibility context boosts discriminative power of each point pair feature by implicitly imposing global characteristic of a queried model, and it reduces the number of spurious matches.

A. Geometric Shape of Model

Visibility context allows us to exploit the convex/concave property associated with a given pair. In the example shown in Fig. 4, pair (p_1, p_2) and pair (p_1, p_3) are matched with the model pair (m_1, m_2) in Fig. 3, since their features $F(p_1, p_2)$, $F(p_1, p_3)$, $F(m_1, m_2)$ are identical. But, based on the visibility context that the vector d is inside the model (*i.e.* convex) and fully invisible, the match with pair (p_1, p_3) can be rejected.

The feature F thus has to include a new element V_{in}, indicating the visibility context of vector d.

$$F(m_1, m_2) = (||d||, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2), \mathbf{V_{in}}),$$

To simplify the computation, we sample points on the vector and test the visibility of these points. If the vector is short (*e.g.* $\leq 5cm$), we only use the center of the vector. Otherwise, we include two additional points near the point m_1 and m_2.

Given the sampled points $\{p_1\}$ or $\{p_1, p_2, p_3\}$, V_{in} is determined as:

$$V_{in} = \begin{cases} 2\ (\in S_V), & \text{if } p_i \in S_V,\ \exists\ i \\ 1\ (\in S_O), & \text{if } p_i \notin S_V,\ \forall i\ \&\ p_i \in S_O,\ \exists i \\ 0\ (\in S_S), & \text{otherwise} \end{cases}$$

B. Scale of Model

Scale of a queried model is a decisive feature that distinguishes the model from other background objects which

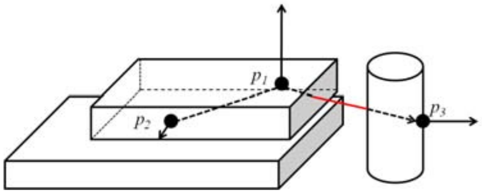


Fig. 4. Scene example

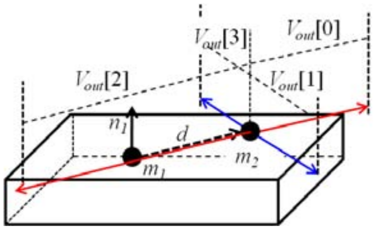


Fig. 5. Illustration of V_{out} : d and $-d$ in red, δ and $-\delta$ in blue

have similar shape. The scale information is characterized by the visibility context *w.r.t* a given pair and is imposed into feature F by adding a new descriptor V_{out}:

$$F(m_1, m_2) \\ = (||d||, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2), V_{in}, \mathbf{V_{out}}),$$

V_{out} is a four-dimension vector, and each coordinate corresponds to four vectors, d, $\delta = N_1 \perp d$, $-d$ and $-\delta$, respectively. For each vector, we measure the length from the point m_2 to the point where the vector meets the visible region as illustrated in Fig. 5, and the estimated length is used to determine the value of the corresponding coordinate in V_{out}.

Finally, for fast feature matching, each value of feature F is mapped into an integer value, and then their combination is used as an index of a point pair array. This is done by dividing each distance-related descriptor ($||d||, V_{out}$) and angle-related feature ($\angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)$) by user-defined paramteres τ_g and τ_θ, respectively. τ_g and τ_θ were always set to $1.5cm$ and $12°$ in our experiments.

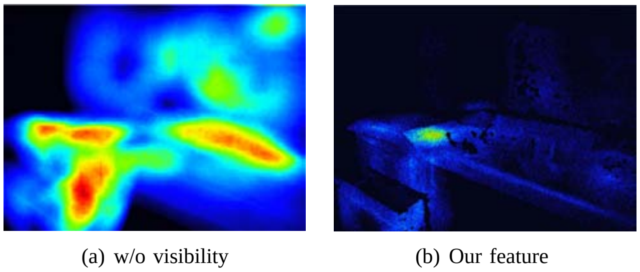
Since V_{out} is associated with visible regions, some true matches might be rejected due to occlusion and clutter. But, our new feature description remarkably reduces invalid matches as shown in Fig. 6.

This figure shows the number of votes for being the center of the book model in the scene. We infer the center of the object from every match using the pose estimation method in [6], and project it into the image coordinate using the projection matrix. Note that the color is assigned only based on relative magnitude in each image.

VI. Object Recognition

Object recognition using point pair features is straightforward. We recognize and localize queried objects by accumulating all the possible poses of the objects inferred from the point pair matches. Then, the poses which have the most support are identified as correct.

This section provides the details about our recognition process. We also introduce a novel approach to compute a transformation matrix from the matched pairs, and two measures that validate the estimated pose.



(a) w/o visibility

(b) Our feature

Fig. 6. Number of votes for potential center of the book in the scene shown in Fig. 2 (high: red, low: blue)

A. Voting Process

For every point p in the simplified range image R', $F(p, p_n)$ is computed with every neighbor p_n. The set of neighbors P_n includes 3D points ($\in R'$) in $\sigma \times \sigma$ neighborhood around the pixel (u, v) of the point p. In our experiments, σ was set to 16, and each point has at most 48 neighbors due to our sampling process with 4 or 8 or 16-pixel gap. Using the extracted feature $F(p, p_n)$, we obtain a set of model point pairs M whose feature corresponds to $F(p, p_n)$.

From every match between scene pair (p, p_n) and model pair (m_1^i, m_2^i) ($(m_1^i, m_2^i) \in M$), we are able to infer the transformation matrix that aligns the matched model with the scene.

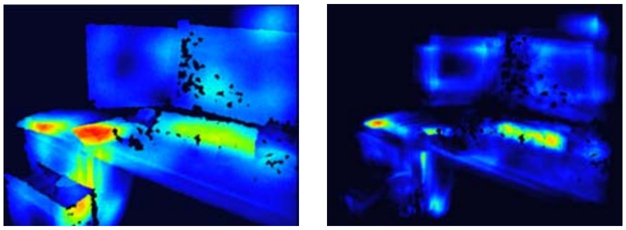
1) Pose estimation: Given two matched oriented point pairs, scene pair (p, p_n) with normal (n_p, n_{p_n}) and model pair (m_1^i, m_2^i) with normal $(n_{m_1^i}, n_{m_2^i})$, with assumption that every point p is on the surface of objects, the approach proposed in [6] infers rotation from the match by first computing T_p^x and T_m^x that align two orientation n_p and $n_{m_1^i}$ to x axis, respectively, and then the final rotation α around x axis that aligns the orientation $n_{m_2^i}$ to n_{p_n}. That is, the model pair is aligned to the scene pair by the matrix $T_x^p T_a T_m^x$. The voting process is locally performed on each point p by only voting for the angle α.

This method infers inaccurate poses as exhibited in Fig. 7(a), since it critically depends on the surface orientation of the point p, which is noisy. For clearer illustration, we only enforced the first coordinate of V_{out} to get more votes. Fig. 6(b) is the corresponding result, when all the coordinates of V_{out} are enforced.

Therefore, we present a new method that computes a transformation matrix based on the relative geometric relationship between interior points and a point pair.

Every queried model has nine arbitrary interior points I_m (*i.e.* points behind the visible surface) and a center point c_m. These points are used not only to compute a transformation matrix, but also to remove wrong poses under the visibility context that interior points are always invisible.

Fig. 8 depicts the geometric relationship between an interior point C and a point pair (P_1, P_2) with normal (N_1, N_2).



(a) Vector alignment approach (b) Our approach

Fig. 7. Comparison on pose estimation: Vector alignment approach [6] vs. our approach

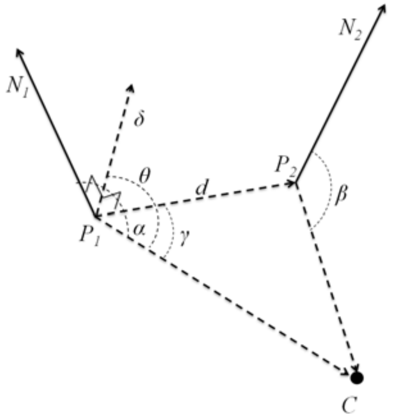


Fig. 8. Geometric relationship between a point C and a point pair (P_1, P_2) with orientation (N_1, N_2)

It is formulated as:

$$\begin{aligned} ||C-P_1|| \cdot cos(\alpha) &= N_1 \cdot (C-P_1) \\ ||C-P_2|| \cdot cos(\beta) &= N_2 \cdot (C-P_2) \\ ||C-P_1|| \cdot cos(\gamma) &= d' \cdot (C-P_1) \\ ||C-P_1|| \cdot cos(\theta) &= \delta \cdot (C-P_1), \end{aligned} \tag{1}$$

where $d' = \frac{d}{||d||}$ and $\delta = N_1 \perp d'$.

The relationship is reformed to Eq. 2.

$$M_N \cdot C = M_D, \tag{2}$$

$$M_N = \begin{bmatrix} N_{1x} & N_{1y} & N_{1z} \\ N_{2x} & N_{2y} & N_{2z} \\ d'_x & d'_y & d'_z \\ \delta_x & \delta_y & \delta_z \end{bmatrix}, \quad C = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}$$

$$M_D = \begin{bmatrix} N_1 \cdot P_1 \\ N_2 \cdot P_2 \\ d' \cdot P_1 \\ \delta \cdot P_1 \end{bmatrix} + \begin{bmatrix} ||C-P_1|| \cdot cos(\alpha) \\ ||C-P_2|| \cdot cos(\beta) \\ ||C-P_1|| \cdot cos(\gamma) \\ ||C-P_1|| \cdot cos(\delta) \end{bmatrix} = M_A + M_B$$

For our recognition process, every model pair has a set of M_B inferred from a center point c_m and three interior points. During the recognition process, when two pairs are matched, we infer M_N and M_A from the scene pair based on Eq 2, and compute the position of the center (c_s) and the three interior points (I_{s3}) of the model in the scene using the least square method.

Finally, the transformation matrix T_s is determined by inferring translation from the estimated center c_s and rotation from the three interior point I_{s3}, respectively. Fig. 7 compares our pose estimation results with the approach in [6]. When all the coordinates of V_{out} are enforced, our final voting result is given in Fig. 9. Compared to Fig. 6(b), the center estimated by our method is more compact.

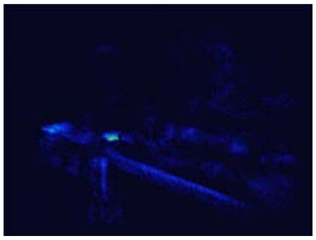


Fig. 9. Number of votes for the book center estimated by our approach

2) Voting process: The next step is to validate the pose T_s by checking that original visibility context of the center point c_m and the interior points I_m is preserved. First, the visibility context of the point c_s should coincide with one of the point c_m. For example, if $c_m \in S_s$, c_s also should be in S_s. Also, all interior points should be in S_{IV} after aligning them into the scene. When both conditions are satisfied, the pose T_s is validated.

When the pose is validated, we vote for it as the potential pose of the matched object. Unlike the approach in [6] that locally accumulates all the potential poses only *w.r.t.* the point p, we vote for every estimated pose in 6D (3D for translation and 3D for rotation) to localize objects globally by accumulating the results obtained from all the matches.

In order to achieve this, we divide the space into uniform voxels with scale of $1cm$. In our experiments, we construct the voxels in range of $(-1m$, $+1m)$ in x and y coordinates and $(0.5m, 1.5m)$ in z coordinate.

As illustrated in Fig. 10, the rotation angles extracted from the matrix T_s is indicated by an index θ and we increase the support for the index θ in the voxel (v_x, v_y, v_z) where the center point c_s is mapped into. That is, the voxel coordinate determines the translation of the pose and the angle with the highest votes in the voxel determines the rotation.

We apply the same voting process to every point pair and accumulate the votes to infer the final pose of the matched objects in the scene.

B. Pose Verification

After voting, each voxel contains a set of potential poses of the corresponding objects with the number of supports. Even though our approach discards many spurious poses, there may still be some false-positive detections. For example, the handset of a phone is similar to a cup in terms of shape and scale. Therefore, we define new measures to verify a selected pose using surface property in terms of two aspects: 1) How well the object is aligned into the scene by the estimated pose (surface alignment score) and 2) the corresponding surface should be separable from the scene (surface separability score).

Given a detection result (matched object M_i and the corresponding pose T_i), we define

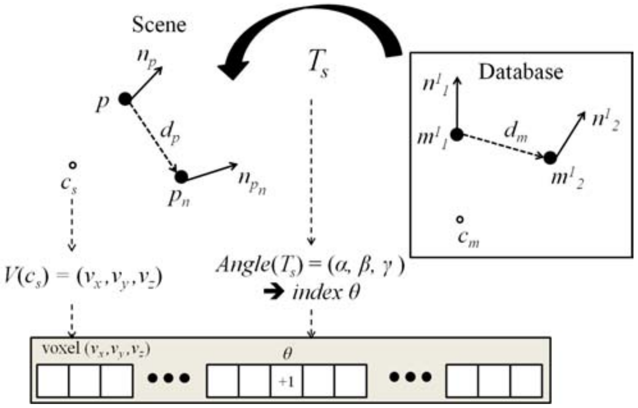


Fig. 10. Overview of voting process

Surface alignment: Motivated by observation that the registered model should be included in the surface or invisible region (due to occlusion), after transforming the model points (*i.e.* $M'_i = T_i M_i$), our alignment score is defined as:

$$= 1 - \frac{|\{x | x \in S_v, x \in M'_i\}|}{|M'_i|} \tag{3}$$

Surface separability: The scene surface S_m corresponding to the registered model M'_i should have less surface consistency with its neighboring surface. To measure surface separability, the surface S_m is grouped with its neighbors whose surface orientation is very similar ($\leq 15°$). Then, the separability score is:

$$= 1 - \frac{\|\mathcal{D}(S_m) - \mathcal{D}(M'_i)\|}{\mathcal{D}(M'_i)}, \tag{4}$$

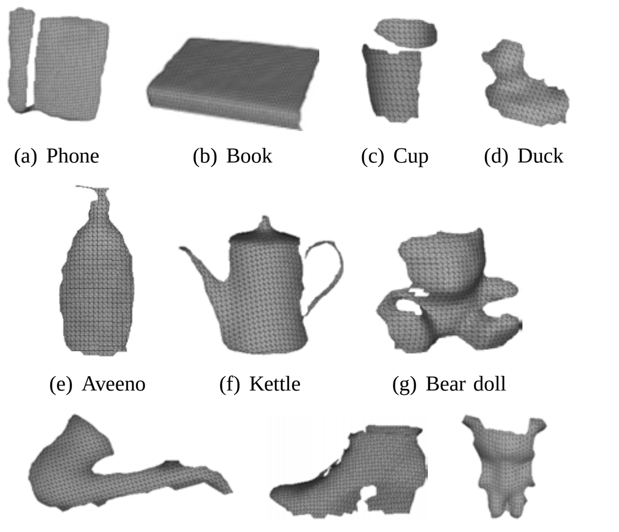
where $\mathcal{D}(i)$ is the dimension of the surface i. When both costs are larger than a user-defined threshold (*e.g.* 0.65 in our experiments), the estimated pose with the highest votes is recognized as a correct detection.

VII. Experimental Results

Our approach was extensively evaluated on large number of range images acquired from a real-time range sensor, PrimeSensor[1]. The PrimeSensor (Fig. 1(a)) produces accurate depth measurements for a 640×480 resolution image in real time (30 fps) by triangulation, for objects between $80\,cm$ and $4\,m$. The setup for our experiments is as follows:

Queried objects: Our database has 10 different query objects as displayed in Fig. 11. To validate our system in terms of variance in views, each object is modeled by a single depth map captured from the sensor.

Ground truth: It is very difficult to collect ground truth of 3D object poses in an uncontrolled environment. Thus, to build ground truth for our test datasets, we manually marked all the regions occupied by the true instances in every image. Given the recognition results, we first align the corresponding model to the scene using the estimated pose. Then, we project the points into the image space and count the number of points falling into the ground truth region. If the count is high ($\geq$ 80% of the total number of points in the model), it is considered as a true positive detection. Otherwise, it is counted as a false detection. In case of the book, it was set to 40% due to occlusion by the other objects.



(a) Phone (b) Book (c) Cup (d) Duck
(e) Aveeno (f) Kettle (g) Bear doll
(h) Toy horn (i) Shoe (j) Doll 2

Fig. 11. Queried objects : We only use a single range image for each class

Comparative analysis: For comparison, we used two state-of-the art methods, [6] and Spin image [10].

We implemented [6] with the parameter values described in the paper. For the Spin image method, we used the source code available at [1]. Note that this module only returns the best pose for each object, while ours detects all matches in the scene. For its best recognition performance, we made resolution of the queried objects and the test range images uniform. Moreover, recognition without compression was processed with the recommended parameters.

We did not use open source implementations in Point Cloud Library [2], since it had no explicit recognition modules using local features such as NARF [20], and GRSD [4] and VFH [18] require object segmentation which is very challenging in our scenario.

A. Test with non-distinctive shaped objects

This test aims at evaluating precision performance of the system. That is, we queried four objects (book, cup, phone, duck) whose partial shape has high similarity with background objects.

Our test dataset (Dataset 1) is obtained from a very cluttered office environment, moving the camera around the environment. It contains 498 depth images with 1813 true instances. The true instances are taken at various views with occlusion and some of them partially appear in the images. Some examples are shown in Fig. 14(a). It is worth noting that we recognize and localize the queried objects in every image independently.

Fig.12 shows ROC curves on the recognition results. The curve on our results (red) demonstrates that our approach significantly outperforms [6] (blue). Also, our recognition results without pose verification (only based on the number

[1] `http://www.primesense.com/files/FMF_2.PDF`

TABLE I

SPIN IMAGE PERFORMANCE ON DATASET 1

recall rate	precision rate	time
0.20	0.19	1.49 hrs

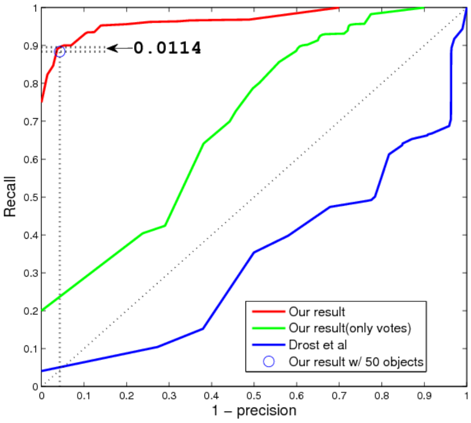


Fig. 12. Recall vs.(1-precision) curves

of the votes received) (green) shows better performance than [6] (blue). Some recognition results of our method are shown in 14(a). The full recognition results of ours and [6]'s method on this dataset are given in the attached supplementary video.

Our approach surpasses [6] in runtime performance as well. Our method took 8.65 s/image, whereas [6] took 54.5 s/image. The system was tested on dual quad-core 3.0 GHz CPUs with 3GB of RAM. Both our voting process and [6] were used on 8 CPUs using OpenMP. On average, our point refinement took 790 ms.

Table I gives the recognition and runtime performance of the Spin image method on Dataset 1. Compared to our result, the Spin image method shows much lower recall rate at the same precision. It returned many false matches, since it finds the best match only based on local shape similarity without consideration of global characteristic such as actual scale of an object. Even though it was processed on the virtual machine (it requires old linux system) on the same PC, it is very slow. It took more than 4 weeks to process Dataset 1.

B. Test with number of queried objects

To validate the scalability of our method, we queried 50 different objects (*e.g.* bottles as displayed in Fig. 13) including our original objects to every image in our dataset. The result is marked by a circle in the chart in Fig.12. At the same precision rate (0.956), the detection rate only drops 0.0114 (0.895 → 0.8836) from our original result. The average computation time with 50 objects is 52.3s/image.

The computational complexity of our approach is O(NM), where N is the number of 3D points in the scene and M is the number of neighboring points of each point (In our experiments, M was at most 48). Practically, it is proportional to the number of point pair matches.

When queried objects have common shape like a book, many model pairs are redundantly matched with scene pairs and this leads to expensive computation, whereas objects having distinctive features may require less computation.

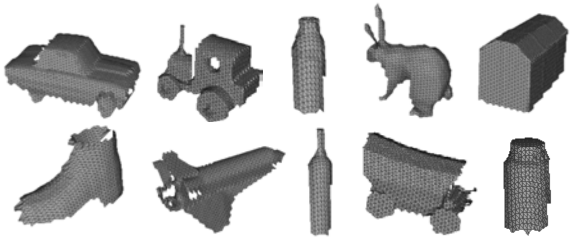


Fig. 13. Example of queried objects

TABLE II

PERFORMANCE COMPARISON ON DATASET 2

	recall rate	precision rate	time
Ours	0.88	0.91	26 s
Spin Image	0.34	0.22	9.53 hrs

C. Test with clutter (Dataset2)

We also ran our module with 10 queried objects on 100 range images which contain the objects compactly with various poses and view points as displayed in Fig 14(b). For performance comparison, the Spin image method was use on the images (only 15 images due to heavy computation time) as well. Table II clearly shows that our method performs better than the Spin Image method.

VIII. CONCLUSION

In this paper, we have proposed a recognition method that utilizes the visibility context, rarely used for object recognition. The visibility context enables us to infer distinctive pattern for each point pair feature so that we can discard many spurious matches. We also elaborate on accurate pose estimation. Lastly, the extensive comparison exhibits the remarkable recognition performance of our method. Our future work includes the use of spatial-temporal context in video scenes, recognition with a large number of query objects and improvement in runtime performance.

REFERENCES

[1] Mesh toolbox. `http://www.cs.cmu.edu/~vmr/software/meshtoolbox/introduction.html`.

[2] Point cloud library. `http://pointclouds.org/`.

[3] Prabin Bariya and Ko Nishno. Scale-hierarchical 3D object recognition in cluttered scenes. In *CVPR*, 2010.

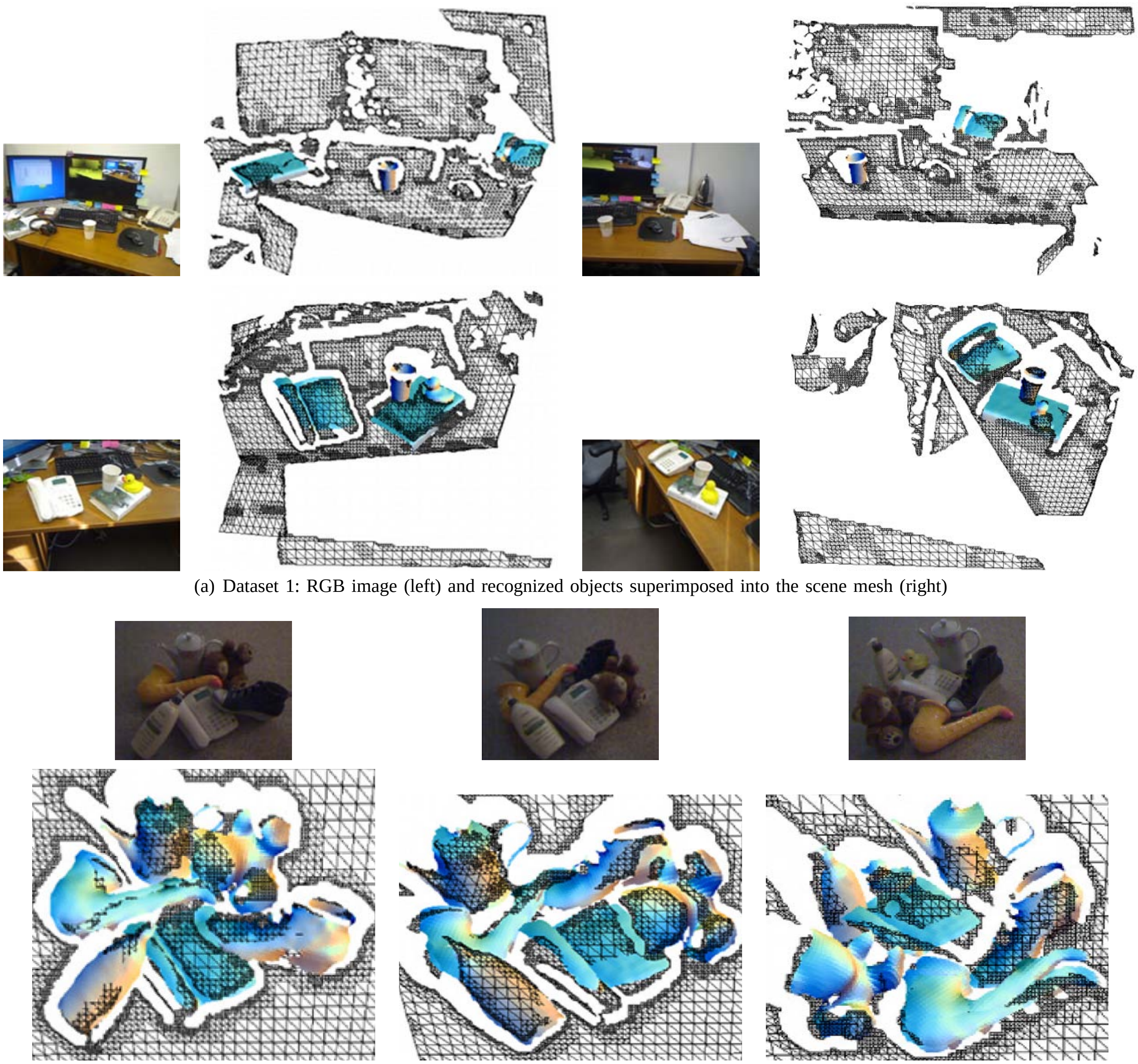
[4] Michael Beetz, Zoltan Csaba Marton, Dejan Pangercic, Radu Bogda Rusu, and Adreas Holzbach. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In *IEEE-RAS International Conference on Humanoid Robots*, 2010.

[5] Hui Chen and Bir Bhanu. 3D free form object recognition in range images usingi local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[6] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *CVPR*, 2010.

[7] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[8] Gael Guennebaud, Marcel Germann, and Markus Gross. Dynamic sampling and rendering of apss. In *Eurographics*, 2008.



(a) Dataset 1: RGB image (left) and recognized objects superimposed into the scene mesh (right)

(b) Dataset 2: RGB image (top) and recognized objects (bottom)

Fig. 14. Qualitative results on both datasets

[9] Gael Guennebaud and Markus Gross. Algebraic point set surfaces. In *Siggraph*, 2007.

[10] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *TPAMI*, 21(5):433–449, 1999.

[11] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough transforms and 3D surf for robust three dimensional classification. In *ECCV*, 2010.

[12] Sukhan Lee, Eunyoung Kim, and Yeonchul Park. 3D object recognition using multiple features for robot manipulation. In *ICRA*, 2006.

[13] Xinju Li and Igor Guskov. 3D object recognition from range images using pyramid matching. In *ICCV*, 2007.

[14] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. 3-D model-based object recognition and segmentation in cluttered scenes. *TPAMI*, 28(10):1584–1601, 2006.

[15] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *IJCV*, 89(2-3):348–361, 2010.

[16] Morgan Quigley, Siddarth Batra, Stephen Gould, Ellen Klingbeil, Quoc Le, Ashley Wellman, , and Andrew Y. Ng. High accuracy 3D sensing for mobile manipulation: Improving object detection and door opening. In *ICRA*, 2009.

[17] Salvador Ruiz-Correa, Linda G. Shapiro, and Marina Meila. A new signature-based method for efficient 3D object recognition. In *CVPR*, 2001.

[18] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, 2010.

[19] Philp Shilane and Thomas Funkhouser. Distinctive regions of 3D surfaces. *ACM TOG*, 26(2):Article 7, 2007.

[20] Bastian Steder, Radu Bogdan, Rusu Kurt Konolige, and Wolfram Burgard. Point feature extraction on 3d range scans taking into account object boundaries. In *ICRA*, 2011.

[21] Fridtjof Stein and Gerard Medion. Structural indexing: efficient 3-D object recognition. *PAMI*, 28(10):125–145, 1992.

[22] Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Shape Modeling Applications*, 2004.