Robust and Fast Hypothesis Verification in 3D Object Recognition

Yi Lu Zheng[a,b,*], Xiao Fei Wang[a,], Peng Song[c,***], and Yin Long Xu[a,****]**

[a] *University of Science and Technology of China, HeFei, 230026 China*

[b] *Southwest University of Science and Technology, MianYang, 621010 China*

[c] *Singapore University of Technology and Design, Singapore, 487372 Singapore*

**e-mail: jimmyzyl@126.com*

***e-mail: wxf9545@mail.ustc.edu.cn*

****e-mail: peng_song@sutd.edu.sg*

*****e-mail: ylxu@ustc.edu.cn*

Abstract—3D object recognition requires generating and verifying hypothesis, where a hypothesis means a hypothesized object together with a hypothesized pose of the object in the scene. Hypothesis verification is typically achieved by transforming the hypothesized object model into the scene based on its pose, estimating the overlap between the transformed model and the scene, and accepting the hypothesis if the overlap area is sufficiently large. This paper develops a robust and fast hypothesis verification approach to improve the recognition accuracy without slowing down its speed. Our key idea is to identify corresponding points in the overlap between a hypothesized model and the scene using both point position and normal, and to speed up the identification of corresponding points by utilizing a signed distance transform map defined on the model's voxelization. Moreover, we estimate overlap area from the identified corresponding points more accurately based on an idea of *area-weighted vertices*. Experiments on publicly available datasets demonstrate the robustness and efficiency of our verification approach.

Keywords: hypothesis verification, 3D object recognition, distance transform map, area-weighted vertices

DOI: 10.1134/S1054661820040264

1. INTRODUCTION

3D object recognition from depth images gains more and more attention from the research community in recent years [1–3]. Given a scene represented as a depth image and a list of objects stored as 3D models in a database, the goal of 3D object recognition is to detect objects in the scene and localize their 6 degree-of-freedom (DOF) poses. Thanks to 3D geometrical information contained in the depth image and 3D models, 3D object recognition is less affected by variation of scale, illumination, and object texture while allowing more accurate object pose estimation than 2D object recognition.

A typical 3D object recognition framework [2] relies on local shape descriptors to find potential feature correspondences between the scene and the models. These potential feature correspondences vote for not only the existence of an object model, but also the object's pose in the scene, resulting in an object hypothesis (i.e., a hypothesize object model and its pose). However, due to nuisances in the depth image and/or limited descriptiveness of the descriptors, the feature correspondences are not guaranteed to be correct and a hypothesis voted by incorrect correspondences is likely to result in a false recognition. Thus, hypothesis verification needs to be conducted to remove these false positives to improve the recognition accuracy.

A hypothesized object and its pose can be considered as correct if the overlap between the transformed object model (based on its pose) and the scene is sufficiently large. This is because an object usually has a certain amount of visible part in the scene, and the overlap is actually the object's visible part only if the hypothesis is correct; see Figure 1. Hence, existing verification approaches reject false positives by identifying the overlap and rejecting those with small overlap area [4–6]. However, existing approaches identify corresponding points in the overlap by only measuring the pointwise distance, and estimate the overlap area by simply counting the number of corresponding points, making the overlap estimation less accurate.

This paper aims at a robust and fast hypothesis verification to improve the recognition accuracy without slowing down its speed. To achieve this, we develop a volumetric approach with two key components:

(i) Fast overlap identification. We speed up the overlap identification by preprocessing the object models in the database, i.e., voxelizing each model and building a signed distance transform map defined on the model's voxelization. During the verification, the

Received March 21, 2019; revised June 6, 2019; accepted July 7, 2020

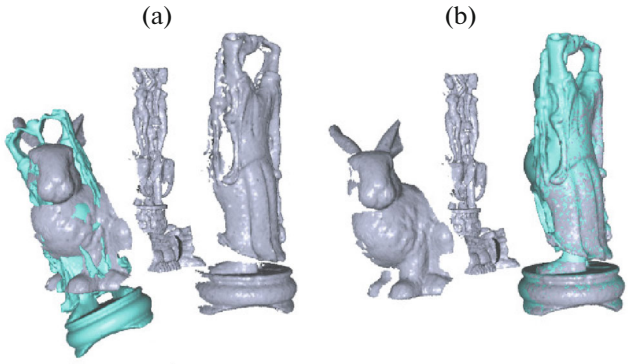


Fig. 1. Hypotheses are verified according to the overlap between the transformed model (in cyan) and the scene (in gray). (a) Small overlap indicates a false positive while; (b) sufficient overlap reports a true positive.

scene is transformed into the hypothesized model's local space, and corresponding points are quickly identified by looking up the distance of each scene point in the model's distance transform map.

(ii) Accurate overlap estimation. When identifying corresponding points, we consider not only the pointwise distance but also the similarity between the normals of the points. By this, the identified corresponding points are more likely to fall in the actual overlap region. Moreover, we estimate the overlap area more accurately by developing an idea of *area-weighted vertice*s, taking advantage of the triangular mesh of the scene or the model.

Experiments on publicly available dataset [5, 7–9] show that our verification approach is not only robust against various nuisances, but also can improve the 3Dobject recognition performance better than a baseline approach.

2. RELATED WORK

3D Object Recognition. It has been heavily investigated in the literature of computer vision and computer graphics. Many 3D object recognition frameworks have been developed. Drost et al. [10] represented 3D data globally using sparse oriented point pair features and matched the features locally using a fast voting scheme for recognition. Lam and Greenspan [3] segmented 3D data into repeatable interest segments, registered segments of the scene and models to estimate objects' poses in the scene, and clustered these poses to determine the recognition result. Pang and Neumann [11] combined machine learning procedures with 3D local features for recognition in cluttered 3D point clouds.

Besides above recognition frameworks, a more commonly used framework [2] is to generate object hypotheses by matching local shape descriptors sampled from the scene and models and verify these hypotheses using geometric constraints. This recognition framework is preferable since it is robust against occlusion by employing the local shape descriptors and is more accurate due to the additional verification procedure. In the followings, we overview two key components in this framework: local shape descriptors and hypothesis verification approaches, respectively.

Local Shape Descriptors. These descriptors encode the local shape around a feature point on 3D data into a vector or a histogram. The descriptor for a feature point located on the target object in the scene should be similar to the descriptor for the corresponding point on the object model in the database. Hence, corresponding points between the scene and the model can be found by comparing their sampled local shape descriptors. Multiple pairs of corresponding points can further vote for a pose of the hypothesized object in the scene. Various local shape descriptors have been developed for 3D object recognition, e.g., spin images [12], 3D shape context [13], 3D tensor [5], and intrinsic shape signatures [14].

Very recently, researchers pay more attention to constructing local shape descriptors with relative to a local reference frame (LRF) that is uniquely defined by the local shape. By attaching a unique LRF to the descriptors, the pose of a hypothesized object in the scene can be estimated using a *single pair* of corresponding points (i.e., descriptors) by aligning the associated LRFs. Descriptors with a unique LRF include signature of histograms of orientations (SHOT) [7], rotational projection statistics (RoPS) [15], local voxelizer [16], and signature of geometric centroids (SGC) [17]. Please refer to [18] for more details about state-of-the-art local shape descriptors.

Hypothesis Verification. By matching local shape descriptors sampled from the scene and the models in the database, many object hypotheses can be generated. The next step is to verify these hypotheses to select those that are likely to be correct. Based on the number of hypotheses considered at a time, existing hypothesis verification approaches can be classified into two classes: individual based and group based approaches.

Individual based approaches verify each hypothesis independently based on the assumption that a hypothesis should be correct if the transformed object model and the scene are aligned well with a sufficient amount of overlap. Johnson and Hebert [4] spread point correspondences between the model and the scene from the initial correspondences found by matching descriptors and counted the number of correspondences after the spread for verification. Mian et al. [5] performed verification by refining the transform with iterative closest point (ICP) algorithm [19] and then counting the number of corresponding vertices between the transformed model and the scene, where a model vertex and a scene vertex are considered corresponding if their mutual distance is less than a threshold. Chen and Bhanu [20] computed the minimum root mean square (RMS) error for the distances between ran-

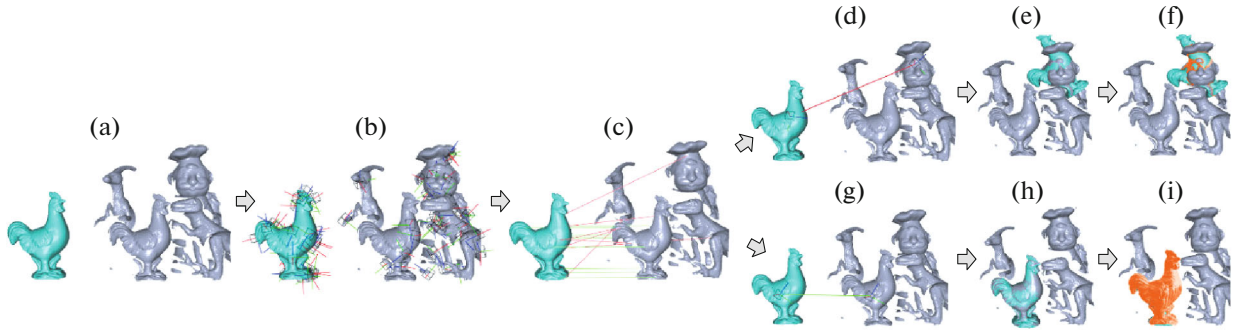


Fig. 2. 3D object recognition pipeline. (a) The input; (b) represent the input model and scene with SGC descriptors (only descriptor's LRF and bounding volume are shown); (c) find feature correspondences by comparing the SGC descriptors, where false and correct correspondences are colored in red and green respectively. Verify the hypotheses generated by (d) a false and (g) a correct feature correspondence respectively. (e, h) Transform the model into the scene based on the correspondence. (f, i) Verify each hypothesis using our approach, where the identified overlap region (colored in yellow) of the correct correspondence is much larger than that of the false correspondence.

domly selected points on the transformed model and their closest points of the scene, where a small minimum RMS error indicates that the hypothesis is likely to be correct. Bariya and Nishino [6] verified a hypothesis by computing the area of overlap between the transformed model and the scene, and employing the area of overlap divided by the total visible surface area of the transformed model within the bounding box of the scene for verification.

Compared with above works, we develop a new measure for identifying corresponding points that considers not only the point position but also its normal. Thus, our measure can get rid of point pairs that are close enough yet not form real correspondences. Moreover, we speed up the identification of corresponding points significantly by building a distance map for each object model in the database off line.

Group based approaches take into account a set of hypotheses and eliminate outliers by considering interactions among the hypotheses. Aldoma et al. [21] verified a whole set of hypotheses altogether by minimizing a global cost function that encompasses geometrical cues. Rodolà et al. [8] developed a game-theoretic framework for selecting hypotheses that satisfy global consistency constraints. Tang et al. [1] verified a group of hypotheses by measuring whether the proposed distance- and angle-preserving constraints are preserved among the hypotheses.

3. 3D OBJECT RECOGNITION PIPELINE

3D object recognition takes a set of object models $\{M_i\}$ and a scene scan as inputs. Each model M_i is represented as a triangular mesh while the scene S can be represented as a triangular mesh or a point cloud. Our 3D object recognition pipeline consists two phases: hypothesis generation (Figs. 2a–2c) and hypothesis verification (Figs. 2d–2i).

3.1. Hypothesis Generation

Local Shape Descriptor. We employ the signature of geometric centroids (SGC) descriptor [17] to find corresponding features between S and $\{M_i\}$. SGC descriptor takes as input a point cloud P and a basis point p, and describes the regional shape around p using the distribution of points in a cubical support surrounding p. The support is discretized into axis-aligned bins (i.e., voxels) with relative to a unique LRF, and the number of points falling within each bin is counted. The descriptor vector is formed by concatenating the number of points in each bin.

Model and Scene Representation. For each M_i and S, we sample N feature points from its triangular mesh or point cloud. To better represent the shape, the feature points should cover the whole shape and should not be too close to one another. Thus, we randomly sample the triangular mesh or the point cloud and enforce minimal separation distance between the samples to obtain N feature points. For each feature point, an SGC descriptor is constructed, and stored in a library. By this, each M_i and S are represented as a set of SGC descriptors; see Fig. 2b.

Hypothesis Generation. We compare each descriptor of S with those of M_i, and compute the descriptor similarity scores. A feature point on S and a feature point on M_i are considered as a match if the descriptor similarity score is larger than a threshold. We sort all matches based on their similarity scores and select the top K matches as the feature correspondences, see Fig. 2c. Note that these feature correspondences are not guaranteed to be correct, e.g., due to less salient, incomplete, or symmetric local shapes. Each feature correspondence leads to a model hypothesis, where the pose of the hypothesized model (i.e., model-to-scene transform) is calculated by aligning the LRFs of associated descriptors.

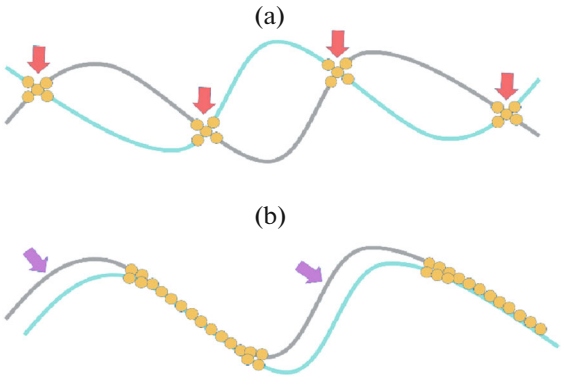


Fig. 3. Identify corresponding points (in orange) between two shapes (in gray and cyan respectively) using only pointwise distance. (a) Point pairs with short distance yet different normals are incorrectly identified as corresponding points (see red arrows); (b) point pairs with similar normals yet moderate distance are incorrectly identified as non-corresponding points (see purple arrows).

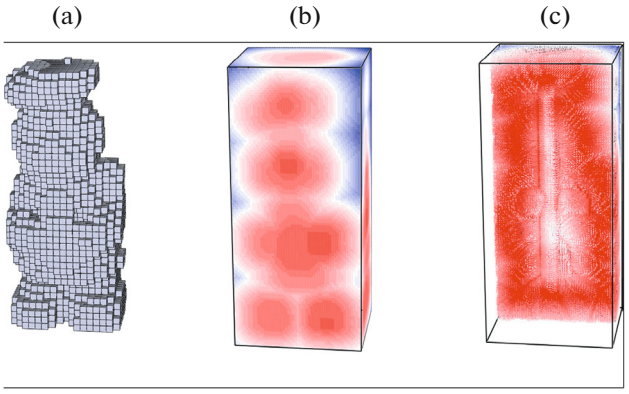


Fig. 4. (a) Voxelize the CHEF model with 110 × 79 × 44 voxels, where exterior voxels are not shown; (b) build a distance transform map on the voxelization, where small and large distances are colored in blue and red, respectively; and (c) compute a gradient vector for each voxel, and visualize them as small arrows. Note that we remove the frontal half of the gradient map for visualization.

3.2. Hypothesis Verification

Given a scene S and a model hypothesis denoted as (M_i, T_i), where M_i is the hypothesized object, T_i is the model-to-scene transform, and $\bar{M}_i$ is the transformed M_i with T_i. Our goal is to verify whether M_i really appears in S with pose T_i; see Fig. 2 (e&h).

Existing approaches [4–6] identify the overlap region based on finding all corresponding points using the pointwise distance, where a model point and a scene point are considered corresponding (i.e., falling in the overlap region) if their mutual distance is less than a threshold. The area of overlap region is further estimated by counting the number of corresponding points.

However, these approaches have three limitations. First, identifying corresponding points solely based on the pointwise distance may not be sufficient since a pair of non-corresponding points can be close to each other physically yet have very different normals; see Fig. 3 for 2D examples. Second, identifying corresponding points between two 3D shapes is a time-consuming process. The time complexity remains high even after speeding up the search process with a k-d tree. Third, the value of estimated overlap area is heavily affected by the point density of input data, making it difficult to set a consistent threshold for different input datasets.

4. HYPOTHESIS VERIFICATION APPROACH

This section presents a volumetric approach to address the limitations of existing verification approaches, consisting of three steps: 1) preprocessing object models in the database; 2) identifying overlap region for a hypothesis; and 3) estimating overlap area to verify the hypothesis.

4.1. Preprocess Object Models

To facilitate the identification of corresponding points in the overlap region, we preprocess object models in the database by voxelizing each M_i and building a signed distance transform map defined on the voxelization. Note that the preprocess is performed on $\{M_i\}$ rather than S since $\{M_i\}$ remain unchanged for different recognition tasks.

To voxelize each M_i, we compute a bounding box of M_i, partition the bounding box into $W \times H \times D$ voxels, cast parallel rays through the model, and use parity count to classify voxels as interior or exterior [22]; see Fig. 4a. Next, we compute a distance transform map on the voxelization [23], which assigns a signed distance value for each voxel. This distance value indicates the shortest (discrete) distance between the voxel and the model surface, where interior (exterior) voxels have negative (positive) values; see Fig. 4b. We further compute a gradient vector for each voxel using the distance values of its 5 × 5 × 5 neighborhood, to encode the local shape orientation information of M_i into the voxel; see Fig. 4c.

4.2. Identify Overlap Region

To identify overlap region between M_i and S, we first transform S into the local object space of M_i using T_i^{-1}; see Fig. 5a. For each point p with normal $\vec{n}$ on the transformed S (denoted as S'), we first locate the voxel of M_i that p resides in, say V_k. Therefore, the distance value of V_k (stored in the distance transform map), say d is the shortest distance between p and M_i. The gradient vector of V_k, say $\overrightarrow{n_m}$ which is considered as approximated normal of p's corresponding point on M_i. Note that d and $\overrightarrow{n_m}$ are obtained by looking up the distance

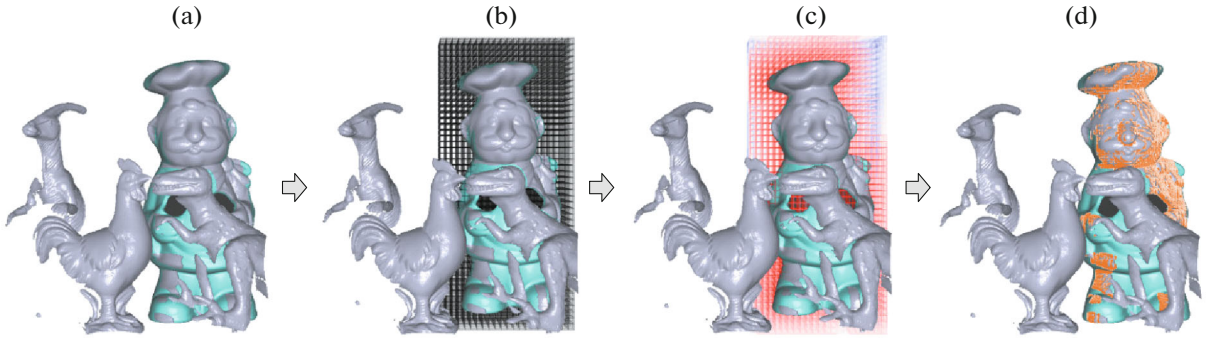


Fig. 5. Our hypothesis verification approach. (a) A hypothesis to be verified, where the model (in cyan) has been transformed into the scene (in gray); (b) voxelize the model in its local object space; (c) create a distance map for the model within the voxelization; (d) identify the overlap region by utilizing the voxelization and the distance map. Note that we remove the frontal half of the voxelization in (b) and the distance map in (c) for visualization.

transform map, costing almost no time; see Figs. 5b, 5c. We measure the confidence that p falls in the overlap region using

$$c(p) = \omega_1 d + \omega_2 \cos^{-1}(\vec{n} \cdot \overrightarrow{n_m}), \quad (1)$$

where ω_1 and ω_2 are the weights, and $\omega_1 = 5.0$ and $\omega_2 = 1.0$ in our experiments. The first term measures the pointwise distance. The second term measures the similarity between the normals of the points, which is not been considered in previous approaches. Once $c(p)$ is computed, we compare it with a predefined threshold c_{thres}, which is set as 4.1 in our experiments. We consider that p falls in the overlap region if $c(p)$ is smaller than c_{thres}. Figure 5d shows the identified overlap region using our approach.

4.3. Estimate Overlap Area

Rather than simply using the number of corresponding points to approximate the overlap area as in previous approaches, we estimate the overlap area more accurately by taking advantage of the triangular mesh of S or M_i.

Denote the set of points (vertices) on S that fall in the overlap region as P_0; see Fig. 6a. One straightforward way is to identify triangles whose vertices all belong to P_0 (see dark orange triangles in Fig. 6b and take these (full) triangles as the whole overlap region. However, this simple approach ignores (partial) triangles that have one or two vertices fall in the overlap region.

To better estimate the overlap region, we develop a new approach based on *area-weighted vertices*. The key idea is to conceptually divide each triangle into three parts corresponding to each vertex and to assign the part area to its corresponding vertex.The area assigned to an arbitrary vertex, say A in ΔABC, is calculated as $(\angle ABC/180)S_{\Delta ABC}$, where $S_{\Delta ABC}$ is the area of ΔABC. For each vertex in the mesh, every triangle that shares the vertex will assign the associated part area to the vertex. By summarizing these assigned areas, we obtain the total area assigned to the vertex. By further summarizing the total area assigned to every vertex (point) in P_0, we obtain an estimated area for the whole overlap region; see Fig. 6b for an illustration.

When S is represented as a point cloud, the above approach will not work since a triangular mesh is missing. For such case, we use the mesh of M_i instead, rather than reconstructing a new mesh from the point cloud of S to avoid introducing unnecessary errors and uncertainty. In detail, we project each point in P_0 onto M_i along its normal direction, obtaining a new set of points on M_i. For each projected point on M_i, we split the corresponding triangle of M_i into three smaller triangles by using partitioning lines from the point to each vertex of the triangle. By this, we can follow the above approach to estimate the overlap area using the processed triangular mesh of M_i.

Based on the estimated overlap area, the score of hypothesis (M_i, T_i) is calculated as:

$$S(M_i, T_i) = \frac{estimate\ overlap\ area}{M_i's\ total\ surface\ area'} \quad (2)$$

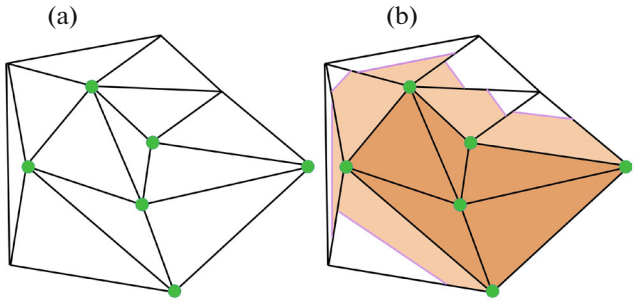


Fig. 6. 2D illustration of our approach to estimate overlap area. (a) A scene/model mesh and points (in green) identified in the overlap region; (b) full triangles in the overlap region are colored in dark orange while partial triangles in the overlap region are colored in light orange. Note that the boundary (in purple) of the overlap region are only for illustration as our goal is to estimate the overlap area rather than computing the overlap boundary.

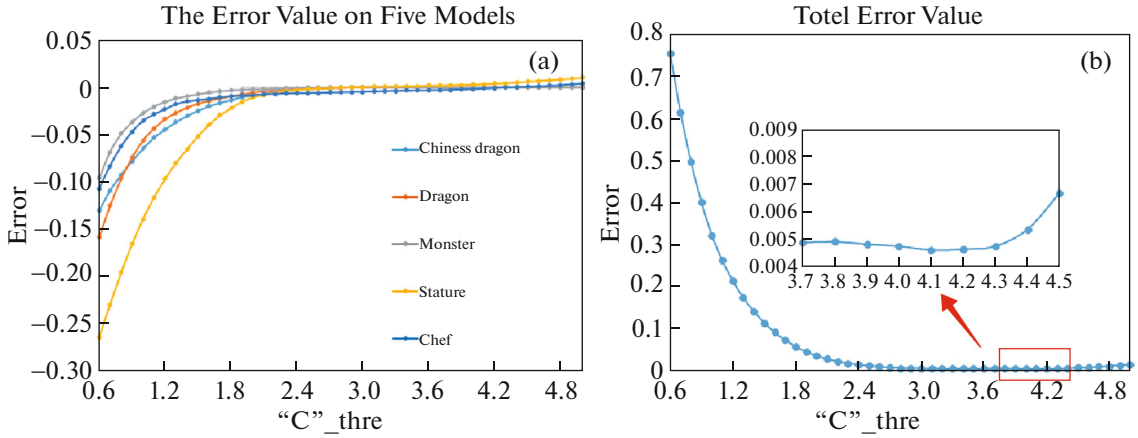


Fig. 7. Parameters selection experiment. (a) Shows the error between the groundtruth and our method with different c_{tres}. (b) Shows the total error of our method on these models.

We accept the hypothesis if the score is larger than a threshold (typically set as 13% in our experiments).

5. RESULT

In this section, we conduct experiments on publicly available datasets to evaluate the robustness of our verification approach with respect to the model pose, point density, and object clutter, as well as to evaluate its performance to identify incorrect hypotheses and to improve 3D object recognition accuracy.

5.1. Parameter Selection

Because the discrete distance d is obtained from the distance map, the points belonging to the overlapping area should satisfy $d = 0.0$. We can exclude the points with a non-zero d by setting ω_1 as a big constant.

Due to the condition $d = 0.0$, we found Eq.(1) a homogeneous equation about ω_2 and c_{thres}. We assume $\omega_2 = 1.0$ and repeat the experiment fifty five times with different c_{thres} on each model. By arranging the error between out result and the groundtruth we get Fig. 7. From Fig. 7a,b we come to the conclusion that the best choice for c_{thres} is 4.1. In conclusion, we choose $\omega_1 = 5.0$, $\omega_2 = 1.0$, $c_{\text{thres}} = 4.1$ for our method.

5.2. Robustness to the Model Pose

To evaluate robustness of our verification approach to the estimated model pose, we compare our approach with the following three approaches:

• *Baseline approach*. The score of a hypothesis is calculated as the number of corresponding points (based on pointwise distance only) divided by the total number of model vertices. Note that we have sped up the identification of corresponding points by building a k-d tree for the model.

• *Distance-only approach*. This is a variation of our approach, in which we enable the distance term while disabling the normal term in Eq. (1).

• *Normal-only approach*. This is the other variation of our approach, in which we enable the normal term while disabling the distance term in Eq. (1)

Figure 8 shows the comparison results on the Bologna dataset [7] under two cases, i.e., the object model is well aligned or roughly aligned with the scene. Yet, our approach is much faster than the baseline since the distance map of object models built during the preprocessing step enables looking up the shortest distance rather than computing it online. Moreover, the score of our approach drops a bit (i.e., from 36.0 to 36.2%, see Fig. 8a) while the score of other approaches drops significantly (see Figs. 8b–8d) when the model pose becomes less accurately estimated, demonstrating the robustness of our approach to the model pose.

5.3. Robustness to the Point Density

To evaluate robustness to the scene point density, we synthesize three scenes with different point density and conduct experiments on them using our approach and the baseline; see Fig. 9. Thanks to *the area-weighted vertices* idea, our estimated overlap area is independent of input point cloud density. Thus, our computed scores on the three scenes have much smaller variation than those of the baseline, and are more close to the ground truth score, which is 32.1%; see Figs. 9b, 9c.

5.4. Robustness to the Occlusion

To evaluate the robustness to the clutter, we conduct experiments on three cluttered scenes from [8] and two real word scenes from [9] using our approach.

Figure 10 shows the verification results, from which we can see that our approach can verify the hypotheses correctly when objects in the scenes are very close to

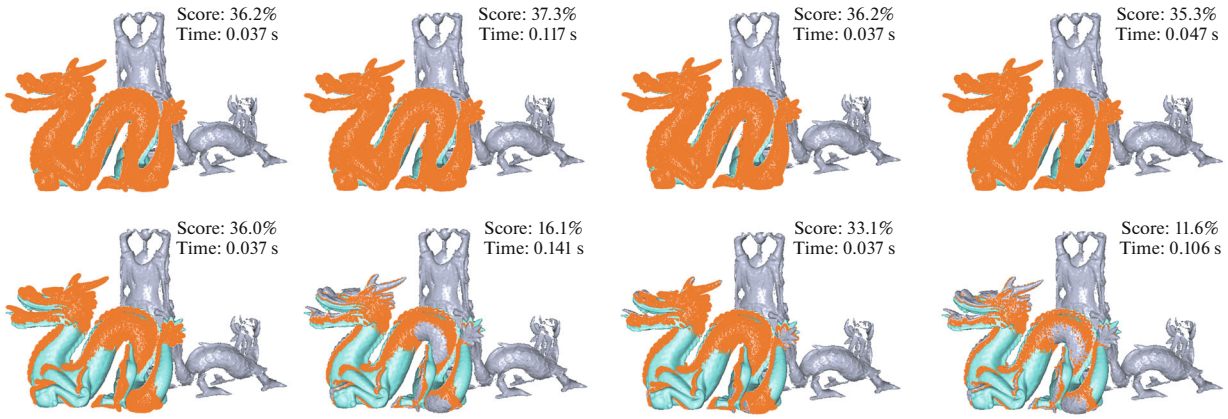


Fig. 8. Verify hypotheses using (a) our, (b) baseline, (c) distance-only, and (d) normal-only approaches for the object model is well aligned (top) and roughly aligned (bottom). The identified overlap regions are colored in orange. The score and computation time are shown on the top left corner of each result.

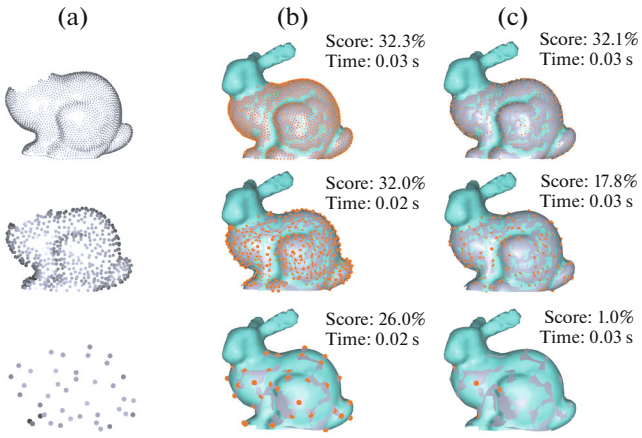


Fig. 9. For (a) synthesized scenes with different point density, we verify hypotheses using (b) our approach and (c) the baseline, where identified corresponding points are colored in orange and are rendered in different sizes for visualization. From top to bottom, the point cloud of the scene consists of 2019, 541 and 41 points respectively.

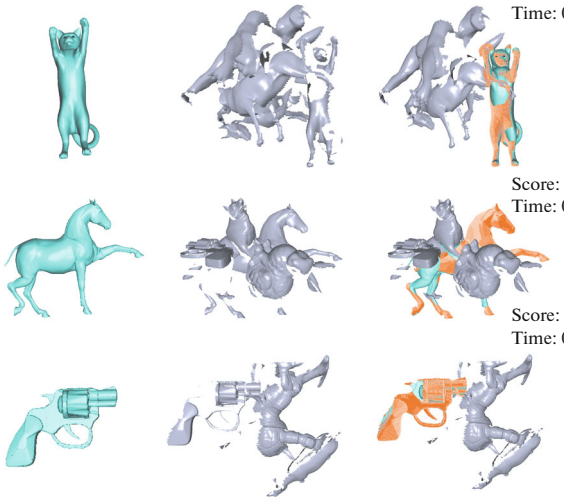


Fig. 10. Evaluate our approach on three cluttered scenes. (a) Target object model; (b) the scene; and (c) our verification result and the corresponding score.

one another even some objects are occluded by its neighbors; then Fig. 11 indicates that our approach can correctly recognize the overlap area as the model and scene being real scan data.

5.5. Robustness to the False Positives

To evaluate the ability to identify incorrect hypotheses (i e., false positives), we take two false recognition results on the UWA dataset [5] as inputs and conduct verification experiments on them using our approach and the baseline; see Fig. 12. From the results, we can see that our approach reports a much smaller score than the baseline, leading to a higher chance to reject these incorrect hypotheses. This is because our approach considers normals for identifying corresponding points such that point pairs with small distance but very different normals are excluded from the overlap region.

5.6. Improve Recognition Performance

Finally, we conduct 3D object recognition experiments on four different datasets, including the Bologna dataset [7], UWA dataset [5, 24], Cvlab dataset [9] and the clutter dataset [8], where we verify three different models with respect to all the scenes in each dataset. The hypotheses generation phrase of the recognition is achieved following the approach in [17] with the SG-C descriptor while the hypotheses verification phrase is performed with our, baseline, distance-only, and normal-only approaches respectively; see Fig. 13. Note that the threshold on the verification score to accept a hypothesis is set the same value for all the four approaches. Figure 13b shows that a number of correct hypotheses are verified as false recognition by the baseline, probably due to the inaccurate estimation of the overlap. In sharp contrast, all correct hypotheses (true positives) and all incorrect hypotheses (false

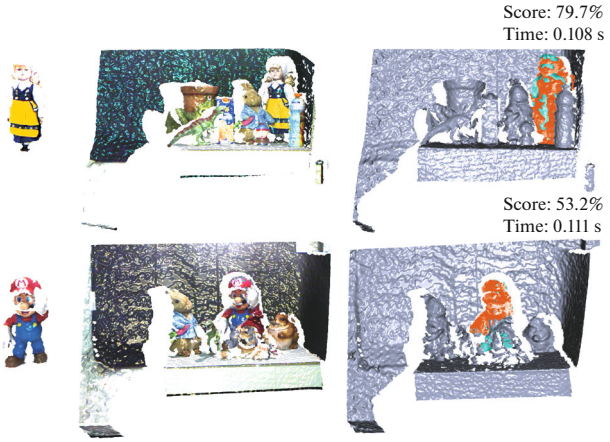


Fig. 11. Evaluate our approach on real scan. (a) The target scan; (b) the scene; and (c) our verification result and the corresponding score.

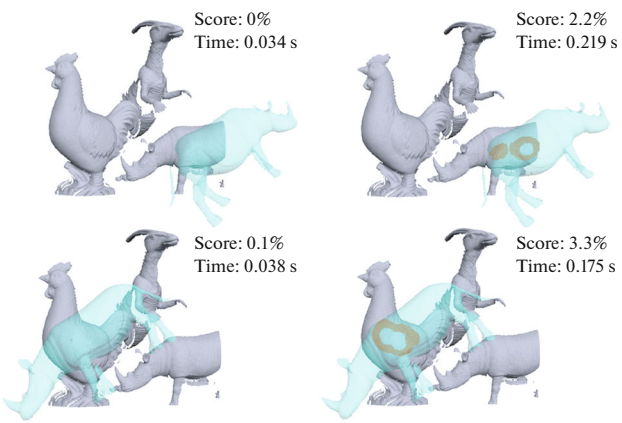


Fig. 12. Evaluate (a) our approach and (b) the baseline to identify incorrect hypotheses, where the hypothesized object either has a wrong pose (top) or forms a false recognition (bottom).

(a)

(b)

	Our		Baseline		Normal-only		Dystance-only		Ground Truth	
Bologna dataset	56	8	49	8	47	8	50	3	56	8
UWA dataset	66	12	60	12	57	12	62	2	68	12
Cvlab dataset	51	6	45	6	43	6	45	3	51	6
Clutter dataset	60	10	54	10	50	10	55	2	60	10

Fig. 13. (a) An example recognition result on the Bologna dataset with the Armadillo model. (b) Number of true positives (left item) and false positives (right item) that are successfully verified by our, baseline, distance-only, and normal only approaches. The ground truth of the recognition results are shown on the rightmost column.

positives) are successfully verified by our approach, making the recognition rate improve from 90.8% before the verification to 100% after the verification.

6. CONCLUSIONS

This paper presents a volumetric approach that can improve the accuracy and speed for verifying hypotheses in 3D object recognition. The improvement on accuracy is achieved by a new measure to identify corresponding points in the overlap region and the weighted vertices approach to estimate the overlap area. The improvement on speed is achieved by building a distance transform map on each object model offline such that corresponding points can be identified simply by looking up distance values in the map. A wide variety of experimental results demonstrate the benefit of our verification approach.

ETHICS DECLARATIONS

Conflict of interest. The authors declare that they have no conflicts of interest.

Ethical approval. This article does not contain any studies with human participants or animals performed by any of the authors.

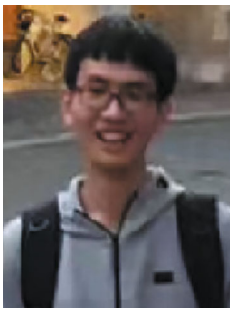
Disclosures and disclaimer. All publicly available datasets in the paper have been noted as required.

REFERENCES

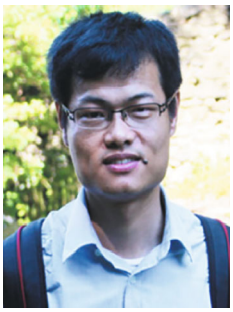
1. K. Tang, P. Song, and X. Chen, "3D object recognition in cluttered scenes with robust shape description and correspondence selection," IEEE Access **5** (99), 1833–1845 (2017).
2. Y. Guo, M. Bennamoun, F. Sohel, et al., "3D object recognition in cluttered scenes with local surface features: A survey," IEEE Trans. Pattern Anal. Mach. Intell. **36** (11), 2270–2287 (2014).
3. J. Lam and M. Greenspan, "3D object recognition by surface registration of interest segments," in *International Conference on 3d Vision. IEEE Computer Society* (2013), 199–206.
4. A. E. Johnson and M. Hebert, "Surface matching for object recognition in complex three-dimensional scenes," Image Vision Comput. **16** (9–10), 635–651 (1998).
5. A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," Int. J. Comput. Vision **89** (2–3), 348–361 (2010).
6. P. Bariya and K. Nishino, "Scale-hierarchical 3D object recognition in cluttered scenes," in *Computer Vision and Pattern Recognition* (2010), pp. 1657–1664.

7. F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision Conference on Computer Vision* (Springer-Verlag, 2010), pp. 356–369.
8. E. Rodolà, A. Albarelli, F. Bergamasco, et al., "A scale independent selection process for 3D object recognition in cluttered scenes," Int. J. Comput. Vision **102** (1–3), 129–145 (2013).
9. F. Tombari and L. D. Stefano, "Hough voting for 3D object recognition under occlusion and clutter," IPSJ Trans. Comput. **4**, 20–29 (2012).
10. B. Drost, M. Ulrich, N. Navab, et al., "Model globally, match locally: Efficient and robust 3D object recognition," in *Computer Vision and Pattern Recognition* (2010), pp. 998–1005.
11. G. Pang and U. Neumann, "Training-based object recognition in cluttered 3D point clouds," in *International Conference on 3DTV* (2013), pp. 87–94.
12. A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," IEEE Trans. Pattern Anal. Mach. Intell. **21** (5), 433–449 (2002).
13. A. Frome, D. Huber, R. Kolluri, et al., "Recognizing objects in range data using regional point descriptors," Lect. Notes Comput. Sci. **3023**, 224–237 (2004).
14. Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *IEEE International Conference on Computer Vision Workshops* (2010), pp. 689–696.
15. Y. Guo, F. Sohel, M. Bennamoun, et al., "Rotational projection statistics for 3D local surface description and object recognition," Int. J. Comput. Vision **105** (1), 63–86 (2013).
16. P. Song and X. Chen, "Pairwise surface registration using local voxelizer," in *Pacific Graphics Short Papers* (Eurographics Assoc., 2015).
17. K. Tang, P. Song, and X. Chen, "Signature of geometric centroids for 3D local shape description and partial shape matching," Lect. Notes Comput. Sci. **10115** (2017).
18. Y. Guo, M. Bennamoun, F. Sohel, et al., "A comprehensive performance evaluation of 3D local feature descriptors," Int. J. Comput. Vision **116** (1), 66–89 (2016).
19. P. J. Besl and N. D. Mckay, "A method for registration of 3-D shapes," IEEE Trans. Pattern Anal. Mach. Intell. **14** (2), 239–256 (2002).
20. H. Chen and B. Bhanu, "3D free-form object recognition in range images using local surface patches," Pattern Recognit. Lett. **28** (10), 1252–1262 (2007).
21. A. Aldoma, F. Tombari, L. D. Stefano, et al., "A global hypotheses verification method for 3D object recognition," in *European Conference on Computer Vision* (Berlin–Heidelberg, 2012), pp. 511–524.
22. F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," IEEE Trans. Vision Comput. Graphics **9** (2), 191–205 (2003).
23. G. J. Grevera, "The 'Dead reckoning' signed distance transform," Comput. Vision Image Understanding **95** (3), 317–333 (2004).
24. A. Mian, M. Bennamoun, and R. Owens, "3D model-based object recognition and segmentation in cluttered scenes," IEEE Trans. Pattern Anal. Mach. Intell. **28** (10), 1584–1601 (2006).

Yillu Zheng, being born in 1987, is a PhD at the University of Science and Technology of China since September 2016. From 2013 to 2017, she worked as a teacher in Southwest University of Science and Technology. She received her B.S. and M.S. from Southwest University of Science and Technology in 2006 and 2010.



Xiao Fei Wang, being born in 1995, received this BSc degree in 2017 from University of Science and Technology of China. And he is now a master degree candidate at University of Science and Technology of China, supervised by Prof. Ligang Liu.



Peng Song, being born in 1986, is an Assistant Professor at Singapore University of Technology and Design (SUTD) since 2019. Prior to joining SUTD, Peng was a research scientist at EPFL, Switzerland, an associate researcher at University of Science and Technology of China, and a research fellow at Nanyang Technological University, Singapore. He received his PhD from Nanyang Technological University in 2013, his master and bachelor degrees both from Harbin Institute of Technology in 2009 and 2007 respectively. is an Assistant Professor at Singapore University of Technology and Design (SUTD) since 2019. Prior to joining SUTD, Peng was a research scientist at EPFL, Switzerland, an associate researcher at University of Science and Technology of China, and a research fellow at Nanyang Technological University, Singapore. He received his PhD from Nanyang Technological University in 2013, his master and bachelor degrees both from Harbin Institute of Technology in 2009 and 2007 respectively.

Yin Long Xu, being born in 1963, received the BS degree in mathematics from Peking University, in 1983, and the MS and PhD degrees in computer science from the University from China (USTC), in 1989 and 2004, respectively. He is currently a professor in School of Computer Science and Technology, USTC, and is leading a research group in doing some networking and high performance computing research. His research interests include network coding, storage systems, combinatorial optimization, design and analysis of parallel algorithms, parallel programming tools, etc. He received the Excellent PhD Advisor Award of Chinese Academy of Science in 2006.