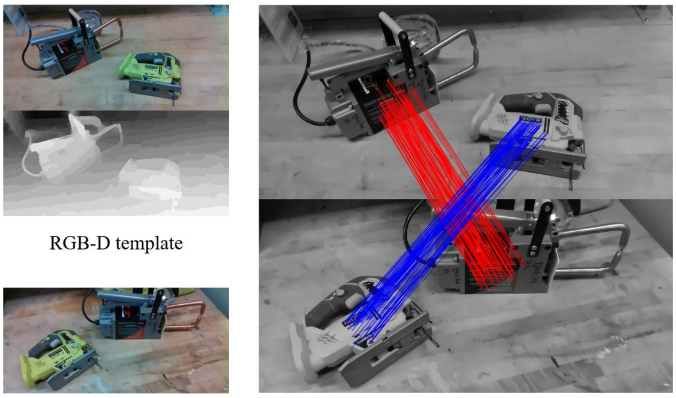
2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Learning To Find Good Correspondences Of Multiple Objects

Youye Xie*, Yingheng Tang†, Gongguo Tang*, and William Hoff‡
*Department of Electrical Engineering, Colorado School of Mines, Golden, Colorado, USA
Email: {youyexie, gtang}@mines.edu
†School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA
Email: tang96@purdue.edu
‡Department of Computer Science, Colorado School of Mines, Golden, Colorado, USA
Email: whoff@mines.edu

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9413319

***Abstract*—Given a set of 3D to 2D putative matches, labeling the correspondences as inliers or outliers plays a critical role in a wide range of computer vision applications including the Perspective-n-Point (PnP) and object recognition. In this paper, we study a more generalized problem which allows the matches to belong to multiple objects with distinct poses. We propose a deep architecture to simultaneously label the correspondences as inliers or outliers and classify the inliers into multiple objects. Specifically, we discretize the 3D rotation space into twenty convex cones based on the facets of a regular icosahedron. For each facet, a facet classifier is trained to predict the probability of a correspondence being an inlier for a pose whose rotation normal vector points towards this facet. An efficient RANSAC-based post-processing algorithm is also proposed to further process the prediction results and detect the objects. Experiments demonstrate that our method is very efficient compared to existing methods and is capable of simultaneously labeling and classifying the inliers of multiple objects with high precision.**


RGB-D template

Test image

Our method output

Fig. 1: Finding good correspondences of multiple objects. Given a set of 3D to 2D putative matches between the RGB-D template and test image, our method will label the inliers and classify them into multiple objects. In this example, two objects are detected in the test image and the predicted good correspondences of the two objects are shown in different colors respectively.

I. Introduction

A. *Finding Correspondences of Multiple Objects*

In this paper, we propose an efficient method to tackle the problem of finding reliable correspondences of multiple objects from a set of 3D to 2D putative matches. Ideally, we want the predicted, good correspondences of an object to be a subset of the ground truth inliers of that object. This problem occurs naturally in many computer vision tasks including the Perspective-n-Point (PnP) problem [1] with multiple objects and 3D object recognition [2]. After obtaining inlier correspondences, they can be applied to estimate the poses of multiple objects [3] and help the system in scene recognition and understanding [4]. An example of the process of finding good correspondences is shown in Fig. 1. Here, we used a color and depth camera (RGB-D camera) to capture a template image of objects, and then matched points from the template image to a test image. In this example, we used the scale-invariant feature transform (SIFT) descriptor [5] for feature matching. Other descriptors, such as oriented fast and rotated brief (ORB) [6], speeded up robust features (SURF) [7], and deep descriptors [8], [9] are also applicable.

Since the 3D rotation can be uniquely determined by a rotation normal vector and a rotation angle around that vector [10], our method discretizes the 3D rotation space based on the direction of the rotation vector. We put a regular icosahedron in the origin of the 3D rotation space and use the twenty facets of the regular icosahedron to define twenty convex cones, where each convex cone is constructed using three vertex vectors belonging to the same facet of the regular icosahedron [11]. All rotation vectors that point towards a facet are associated with that facet and belong to the corresponding convex cone defined by this facet. Then for each convex cone (or facet of the regular icosahedron), we train a classifier to identify inlier correspondences for poses whose rotation normal vector falls within this convex cone (or points towards this facet of the regular icosahedron). We say that an object belongs to a facet when the pose of the object is associated with a rotation vector pointing towards this facet. Therefore, if objects have distinct poses, namely, if different objects belong to different facets, each facet classifier is responsible for classifying the inliers of at most one object. We discuss how to handle the case when multiple objects belong to the same facet in Section II-D. The inlier correspondences identified by the network classifier are then post-processed to filter out any remaining outlier matches, and fit a rotation and translation for each detected object.

An important contribution of our method is that we do not require any costly iterations to identify inlier correspondences, unlike traditional methods. Instead, inliers are identified by a single pass through a network, followed by a short post-processing step. The post-processing step does use an iterative algorithm, but the number of iterations are very small. As a result, our method is much faster than competing state of the art methods. Also, we can handle the case where multiple objects are present in the scene. In Section III, we show experimental results on synthetic data as well as a publicly available dataset.

B. Related Work

Given a set of putative matches, many methods have been proposed to detect inlier correspondences and fit a model, among which RANSAC [12] is the de facto standard in practice [13]. Some extensions of RANSAC include MLESAC [14] which chooses the solution maximazing the likelihood, PROSAC [15] which explores hypotheses from a gradually increasing subset of matches, and USAC [16] which combines multiple RANSAC improving techniques into a unified framework. Some approaches [17], [18] extend RANSAC to incorporate multiple objects. However, since these approaches rely on sampling a small subset of matches to estimate the hypothesis, as the portion of outliers or noise level increases, the required number of iterations for hypothesis estimation increases significantly.

In contrast, learning-based methods have attracted much interest due to their non-iterative end-to-end processing approaches [19], [20], [21]. Most learning-based methods for pose estimation take raw images as the input [22], [23], [24], [25]. However, [13] shows that this approach is not suitable for scenes with occlusion and large baselines. For outlier rejection, [26] proposes a learning-based differentiable counterpart of RANSAC called DSAC, which tries to mimic RANSAC.

Recently, some approaches have been proposed to use a network to find inliers among point correspondences. [13] proposes a network to directly predict inlier probabilities for 2D to 2D correspondences. [27] applies the network of [13] to the case of 3D to 2D correspondences and achieves promising results for the Perspective-n-Point (PnP) problem. Our work is closely related to [13] and [27]. Nonetheless, [13] and [27] assume there exists only one model or object among the correspondences, whereas our work allows multiple objects.

The rest of this paper is organized as follows. In Section II, we propose the learning-based facet network and post-processing algorithm. Several numerical simulations and an experiment on real data are reported in Section III. Finally, we conclude this paper in Section IV.

II. The Proposed Method

A. Learning-based Facet Network

As introduced in Section I, we discretize 3D rotation space into twenty convex cones according to the twenty facets of a regular icosahedron. For each facet, as shown in Fig. 2, a facet classifier is trained to identify correspondences that are compatible with a pose whose rotation normal vector points towards this facet. Thus, there is a bijective relationship between the 20 facet classifiers and the 20 convex cones defined by the 20 facets of the regular icosahedron.

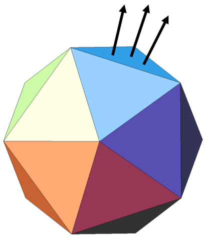
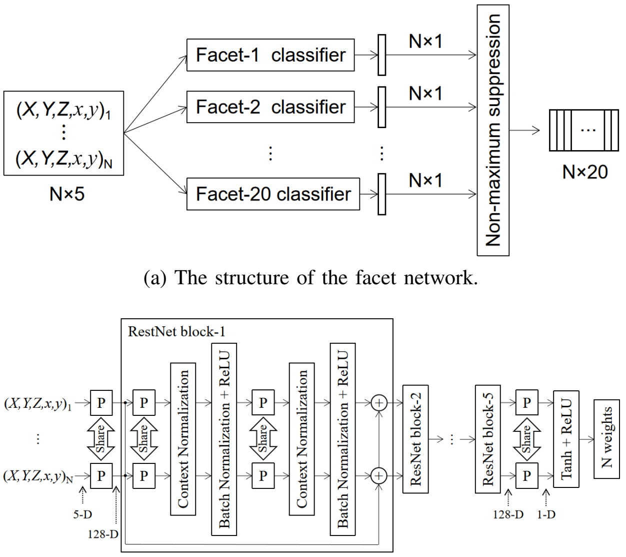


Fig. 2: The regular icosahedron and three vectors pointing towards the same facet.

Since all the 3D to 2D point correspondences are interchangeable, the order of the input correspondences should not affect the prediction result. Therefore, we adopt the ResNet block structure proposed in [13], which shares weights between correspondences and allows different number of matches as input, to build our facet classifiers as shown in Fig. 3. Specifically, the facet network consists of 20 facet classifiers of the same structure but different weights. If we have N putative matches, the input of the facet network is of size $N \times 5$ where each row stores a 3D to 2D match. Each match consists of the 3D point from the RGB-D template and its corresponding normalized 2D point in the test image. A multilayer perceptron with shared weights is applied to each match individually and context normalization [13], which implements normalization on each neuron using information among all matches, is responsible for embedding global information. The output is of size $N \times 20$ where the (i, j)-th entry stores the inlier probability (from 0 to 1) of the i-th match for facet-j. The outputs of the facet classifiers are passed through a non-maximum suppression block. This ensures that each row has at most one non-zero entry, since we assume that each inlier match can only belong to one facet.

Note that there is a trade off between the number of classifiers and resolution in the 3D rotation space. Increasing the number of classifiers by discretizing the 3D rotation space into more exclusive convex cones would lead to higher rotation space resolution but requires more training effort. In addition, an alternative approach is to train a multi-class classifier instead of several binary classifiers as in this paper. Nevertheless, having several binary classifiers that can be trained separately and individually provides much more flexibility to the model. Specifically, if some of the weights are missing or corrupted, we only need to retrain the specific classifiers with corrupted weights. Moreover, if we are only interested in the objects with certain range of rotation or we have the prior knowledge on the range of rotation for the objects of interest, we don't need to apply all classifiers and only the classifiers for the rotation of interest are sufficient for the object detection.



(a) The structure of the facet network.

(b) The structure of each facet classifier.

Fig. 3: The structure of facet network consists of twenty facet classifiers of the same structure. (a) The facet network. (b) The facet classifier where P denotes multilayer perceptron. We apply the ResNet block structure proposed in [13]. $(X, Y, Z)_i$ is the 3D point in the RGB-D template and $(x, y)_i$ is the corresponding, normalized x and y coordinates of the i-th match in the test image.

B. Network Training

Since each facet classifier is responsible for only one facet (or convex cone), namely, it identifies whether a 3D to 2D match is compatible with a pose whose rotation vector lives in the specific convex cone, the matches which are inliers for one classifier are outliers for the rest of the classifiers. Therefore, the 20 facet classifiers are trained separately using the binary cross entropy loss function.

$$L = -\left[\frac{\alpha_1}{N_{in}}\sum_{i=1}^{N}\mathbb{1}_i\log(p_i) + \frac{\alpha_2}{N_{out}}\sum_{i=1}^{N}(1-\mathbb{1}_i)\log(1-p_i)\right] \tag{1}$$

where N_{in} and N_{out} are the total number of inlier and outlier matches. $N_{in} + N_{out} = N$. $\mathbb{1}_i$ is the indicator function which is 1 when the i-th match is the inlier for the classifier under training and 0 otherwise. p_i is the estimated inlier probability of the i-th match. $\alpha_1 = 1$ and $\alpha_2 = 2$ are the weights.

For each facet classifier, its training dataset contains 32000 examples where each example consists of 200 3D to 2D matches. The validation set is of size 320. Each example contains the matches of multiple objects whose number is uniformly selected in $\{1, 2, 3\}$. At most 1 of them is the inlier object of the current facet, which provides robustness to the classifier against outlier objects interference. The matches belonging to the same object follow the same 3D transformation. To create the inlier object of a specific facet, we randomly sampled its rotation vector within the convex cone associated with this facet using the three vertex vectors [11]. The network is trained using Adam optimization algorithm [28] with 0.0001 initial learning rate and 32 batch size for 200 epochs. The learning rate would decrease by half if the loss on the validation set does not decrease for 7 consecutive epochs. The detailed 3D to 2D matches generation process and noise information for our experiment are described in Section III Experiments.

C. Post-processing And Object Detection

After receiving the inlier probabilities, denoted as $\mathbf{W_{in}} \in \mathbb{R}^{N\times 20}$, from the facet network, a RANSAC-based post-processing component is implemented to detect the objects in the test image and return the correspondences for each of the detected objects. Specifically, the post-processing component contains two steps. The first step is adaptive thresholding. If we assume there are at most k objects in the test image, we threshold each entry of $\mathbf{W_{in}}$ to either 0 (outlier) or 1 (inlier), starting with a threshold of 0.9 and then gradually decreasing the threshold value with a step size of 0.05. This process will stop when we have k columns of $\mathbf{W_{in}}$ that have at least n_1 non-zero entries, or when the threshold value reaches T_1.

The second step is a RANSAC-based clustering step. We sort the columns of $\mathbf{W_{in}}$ based on the number of non-zero entries in each column. Then starting from the column with the largest number of predicted inliers, we first fit a rotation and translation and then verify this transformation using predicted inliers from all columns. Predicted inliers in other columns that agree with this transformation will be assigned to the current examining column. In addition, those confirmed inliers will be excluded in the following transformation verification for other columns. This process will repeat until all columns with at least n_2 number of predicted inliers are examined. The RANSAC-based clustering step can be viewed as RANSAC with a restricted subset of matches for hypothesis estimation. Because the inlier portion in each subset is very high after network prediction and thresholding, the post-processing component is extremely efficient, requiring very few iterations (this is verified in the experiments in Section III). The reason that we verify the transformation using predicted inliers in other columns is that, due to noise, some ground-truth inliers belonging to the same object may spread to several facets. This can happen, for example, if this object's rotation normal vector is pointing close to the facet boundary. Any remaining predicted matches after the clustering step will be discarded.

Thus, the post-processed output denoted as $\mathbf{W_{out}} \in \mathbb{R}^{N\times 20}$ has many zero columns and, ideally, only k columns with a large number of non-zero entries. Then a simple thresholding with threshold value T_2 on the normalized number of predicted inliers for each column can be applied to detect the objects. Here, the normalized number of predicted inliers is defined as the number of predicted inliers in a facet divided by the total number of predicted inliers. For the example object shown in Fig. 1, we show the results of processing in Fig. 4. This figure shows raw matches, and the normalized number of predicted inliers for different facets after thresholding and after clustering respectively.

The hyper parameters of the post-processing should be set accordingly based on the estimated statistics and noise

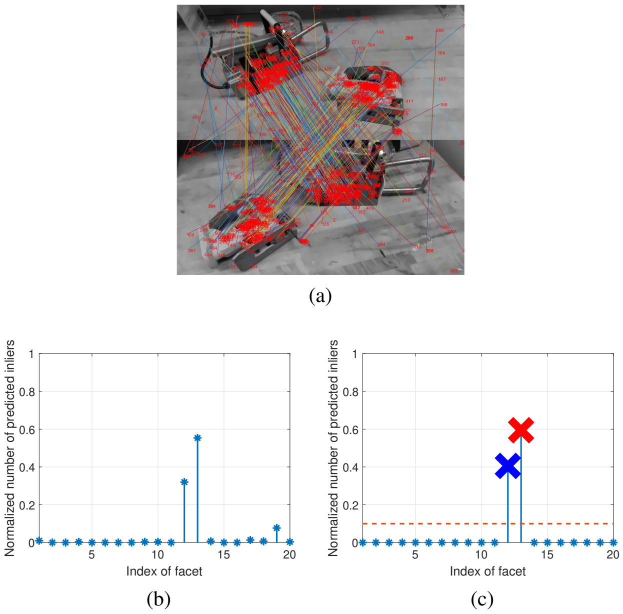


Fig. 4: Post-processing and object detection. (a) Raw matches using SIFT descriptor. (b) The normalized number of predicted inliers for different facets after adaptive thresholding. (c) The normalized number of predicted inliers for different facets after post-processing. Yellow dotted line shows the threshold T_2 for object detection. We set $k = 3$, $T_1 = 60\%$, $T_2 = 0.1$, $n_1 = 20$, and $n_2 = 10$.

level of the data. Specifically, k should be set to be the estimated, largest number of objects among the matches and n_2 represents the minimal number of matches expected for each object. n_1 should be set slightly greater than n_2 to allow for some contaminated inliers prior to the post-processing step. Note that due to the false negative prediction caused by the noise and network error, n_2 is normally smaller than the ground-truth statistics of the data. T_1 represents the desired minimal probability of each predicted match being an inlier, which should be set higher when the noise level is low. T_2 should be set slightly less than the estimated, mininal normalized number of inliers of the objects.

D. Discussion

In this paper, we assume objects to have distinct poses; namely, different objects have their rotation normal vectors pointing towards different facets, so that each peak in Fig. 4 corresponds to one object in the test image. If there exists several objects with normal vectors that point towards the same facet, we find that one potential solution is to train another angle network which consists of multiple angle classifiers. Specifically, each angle classifier is responsible for detecting correspondences for poses that have the rotation angle around the normal vector falling in a specific angle range. Then by combining the results from the facet and angle networks, one can classify several objects belong to the same facet in a non-iterative manner. Alternatively, one can implement RANSAC sequentially on the predicted inliers belonging to the same facet, and set the stop criterion based on a pre-set minimum number of inliers for each object.

III. Experiments

In this section, we report the results of several numerical simulations and an experiment on the GMU kitchen dataset [29]. To train our network[1], we generate a synthetic training dataset of 32000 examples and a validation dataset of 320 with outliers and noise as described in Section II-B. We follow the data generation procedure described in [27] for PnP and extend it to the case of multiple objects. Specifically, each example comprises 200 3D to 2D matches. The number of objects in each example and the inlier portion of each object is uniformly selected in $\{1, 2, 3\}$ and between $[0.2, 0.3]$ respectively. We first generate 3D points in camera coordinates whose X, Y, Z are uniformly sampled from the ranges of $[-1, 1]$, $[-1, 1]$, and $[4, 8]$ respectively. Then using the intrinsic parameters $f_x = f_y = 800$, $x_c = 320$ and $y_c = 240$, we project the 3D points onto the 2D image and add Gaussian noise with 5 pixels standard deviation. For those matches belonging to the same object, we set their ground-truth translation of the camera pose as their centorid and randomly set the rotation. Matches that do not belong to any objects have random translation and rotation.

Metrics. For simulations, we generate a testing dataset of 1000 examples. For each example, we calculate the inlier detection precision and recall, and record the average number of RANSAC iterations in the post-processing step and the average time consumption (unit: second) using a GTX 1080 GPU for network inference and an i7-6700 CPU for post-processing. Inlier detection precision is defined as the number of detected ground-truth inliers divided by the number of predicted inliers. Recall is defined as the number of detected ground-truth inliers divided by the number of all ground-truth inliers.

Compared methods. We implement sequential RANSAC [18], [30] which applies RANSAC to detect each object sequentially, and removes the inliers from the dataset as each transformation is detected. If the number of objects is one, sequential RANSAC is equivalent to classical RANSAC. In addition, we train the inlier prediction network proposed in [13]. Since they assume there is only one object in each example, we retrain their network so that it predicts the inliers without classifying them into different objects. Then a sequential RANSAC post-processing is performed to fit the transformations of multiple objects. Moreover, since their network is deeper than our facet network, we train it with a training dataset of 64000 examples. And if not explicitly stated, we set $k = 3$, $T_1 = 60\%$, $T_2 = 0.1$, $n_1 = 20$, and $n_2 = 10$ for our approach. Note that both sequential RANSAC as well as the network of Yi et al. [13] followed by sequential RANSAC require knowledge of the ground-truth number of objects, which controls the number of transformations they want to fit. For fairness and to study the performance of inlier prediction and object detection of our method individually, in Section III-A and III-B, we directly pick the n largest

[1] Code is available at https://github.com/youyexie/Learning-To-Find-Good-Correspondences-Of-Multiple-Objects

peaks from the post-processed normalized number of predicted inliers for different facets to calculate the metrics, where n is the ground-truth number of objects. In Section III-C, we study the object detection performance of our method individually. For the real data, the ground-truth number of objects is not provided.

A. Finding Correspondences Of One Object

We first study the simple case where there is only one object with 30% inlier portion in each example of the testing dataset, with 2 pixels standard deviation Gaussian noise. Since the network of Yi et al. [13] predicts weights for each match, an inlier detection threshold is needed. We adjust this threshold so that they achieve similar recall to our method. Since in the one object case the standard deviation of the noise is not very large, we set $T_1 = 70\%$. The result is recorded in Table I, from which we can observe that all methods achieve over 99% precision and over 75% ground-truth inliers are detected. However, our method requires a much smaller average number of iterations compared to others.

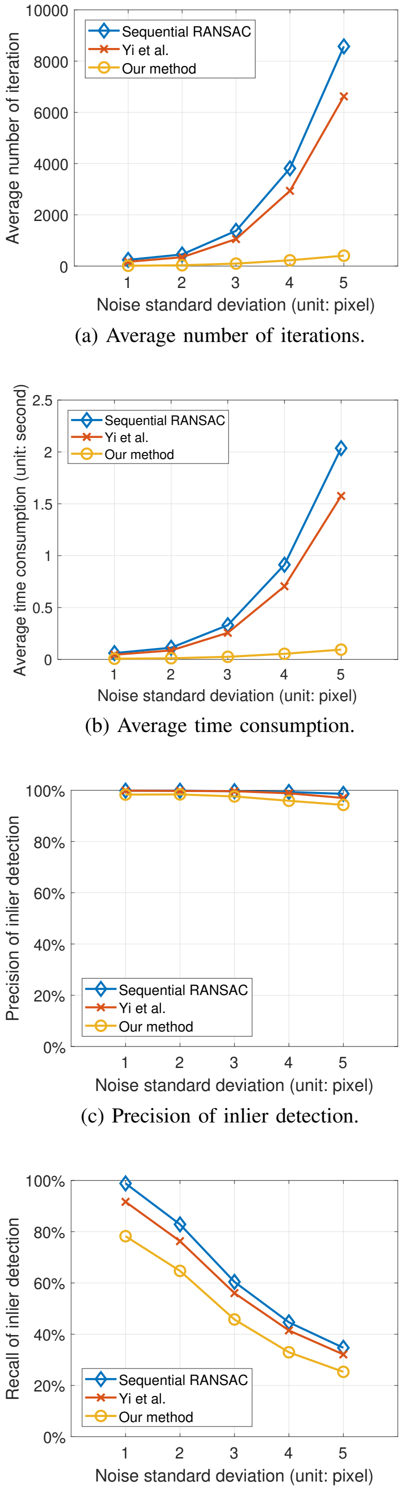
TABLE I: Finding correspondences of one object.

	Precision	Recall	Average number of iterations
RANSAC	99.9%	80.4%	374.3
Yi et al. [13]	99.8%	75.4%	24.4
Our method	99.2%	75.7%	17.5

B. Finding Correspondences Of Multiple Objects

Now we turn to the case of multiple objects, where each example of the testing dataset contains 3 objects with distinct poses and the same inlier portion of 30%. Thus, the outlier portion is 10% and the pseudo-outlier [31] portion, which is defined as the outlier portion to each object, is 70%. We set the inlier detection threshold as 0.5 for Yi et al. [13]. We vary the standard deviation of the Gaussian noise and record the results in Fig. 5. Under severe noise, some inliers are heavily contaminated and that explains the significant drop of recall as the noise standard deviation increases.

From the results we can observe that although our method is slightly inferior to sequential RANSAC and Yi et al. [13] in terms of inlier detection precision and recall, our method nevertheless achieves over 94.2% precision under severe noise and large pseudo-outlier interference. More importantly, our method is around 15× faster than Yi et al. [13] and 20× faster than sequential RANSAC in terms of the average number of iterations and average time consumption when the standard deviation of noise is 5 pixels. In addition, the average time consumption of our method is below 0.1 second per example consisting of 200 3D to 2D matches. This is due to the fact that the feed-forward classifier network is very efficient and effective in predicting inliers, thus reducing the number of iterations required by the RANSAC-based post-processing step and total processing time. To further verify the efficiency of our method, a similar experiment fixing the Gaussian noise standard deviation to 2 pixels and varying the inlier portion of each object is also implemented, and the average number of iterations and time consumption are recorded in Fig. 6. These results confirm that our method is substantially faster than the other two methods, in terms of the number of iterations and average time consumption.



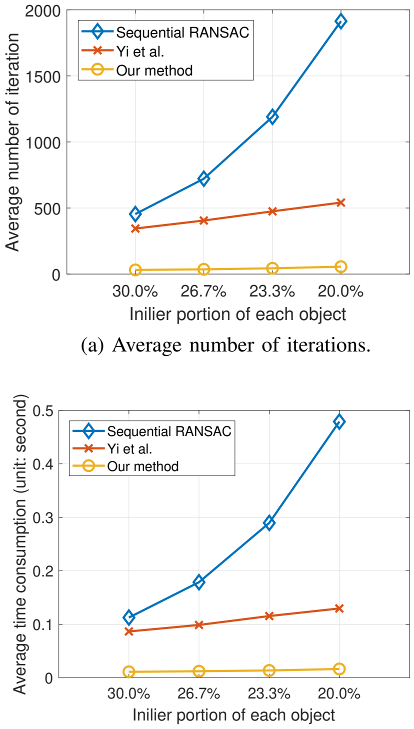
(a) Average number of iterations.

(b) Average time consumption.

(c) Precision of inlier detection.

(d) Recall of inlier detection.

Fig. 5: Finding correspondences of multiple objects with varying standard deviation of the additive noise.



(a) Average number of iterations.

(b) Average time consumption.

Fig. 6: The effect of the inlier portion of each object on the average number of iteration and time consumption.

C. *Object Detection Performance*

Besides being very efficient, our method can detect multiple objects among correspondences automatically and we examine the object detection performance in this section. Each example of the testing dataset has the ground-truth number of objects uniformly sampled from $\{1, 2, 3\}$ and all objects have the same inlier portion. When 20% of the inliers of an object are detected, we count it as a success detection and we define the object detection accuracy as the number of detected objects divided by the total number of objects. The object detection accuracy under different noise level and inlier protion of each object is recorded in Fig. 7, which shows that our method can detect the objects very accurately.

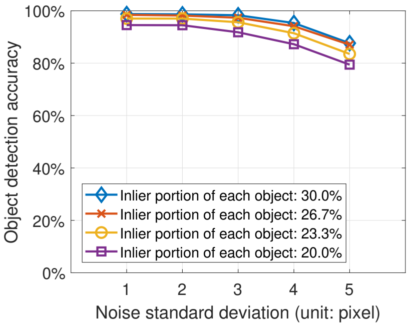
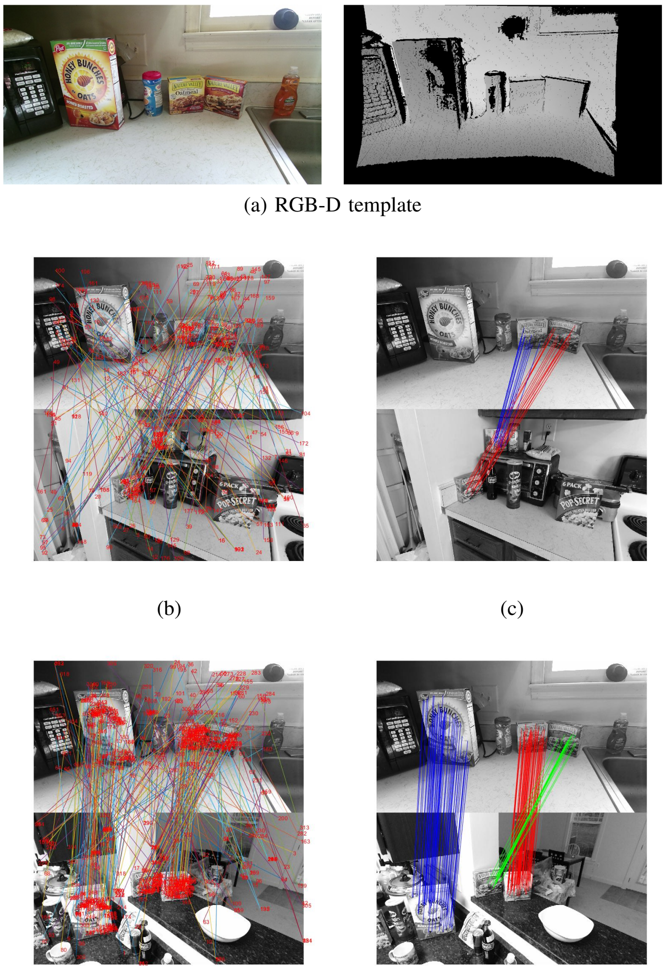


Fig. 7: The object detection accuracy with different Gaussian noise standard deviation and inlier portion of each object.

D. *Performance on GMU Kitchen Dataset [29]*

In the last experiment, we implement our method on the GMU kitchen dataset [29] which consists of multiple kitchen scenes. Specifically, we take two images from scenes 1 and 7 as the RGB-D templates and several images from the rest of the scenes as the test images. Then based on the distribution of the 3D points in the templates and the provided intrinsic matrix, we generate a synthetic training dataset to train our facet network. SIFT descriptors [5] are applied for the feature matching. The inlier prediction results of multiple objects on GMU Kitchen Dataset are shown in Fig. 8 and 9, in which different colors of correspondences indicates different detected objects. The promising results imply that by slightly adjusting the synthetic training dataset, our method is capable of simultaneously finding the good correspondences and classifying them into multiple objects on real data.



(a) RGB-D template

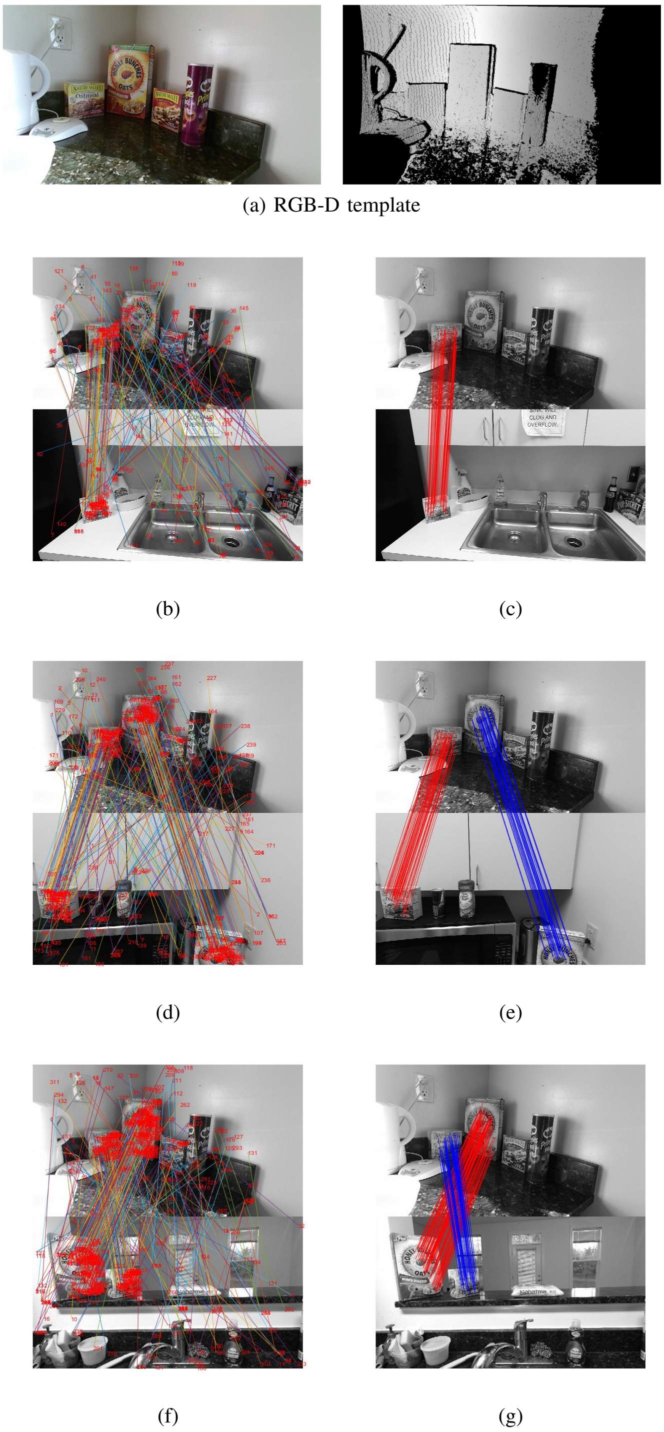
(b)

(c)

(d)

(e)

Fig. 8: Finding good correspondences on GMU kitchen dataset. (a) shows the RGB-D template. (b) and (d) are the raw matches using SIFT descriptor and the results are shown in (c) and (e) respectively.



(a) RGB-D template

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 9: Finding good correspondences on GMU kitchen dataset. (a) shows the RGB-D template. (b), (d), and (f) are the raw matches using SIFT descriptor and the results are in (c), (e), and (g) respectively.

IV. Conclusion

In this paper, we propose an efficient method consisting of a learning-based facet network and a RANSAC-based post-processing step to accurately find good correspondences of multiple objects with distinct poses, given a set of 3D to 2D putative matches. We discretize the 3D rotation space using a regular icosahedron, and for each facet of the icosahedron, a classifier is trained to identify inlier correspondences for poses that have a rotation normal vector pointing towards the facet. According to our experiments, the proposed method is extremely efficient compared to existing methods and is able to simultaneously identify inliers and detect objects accurately.

References

[1] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

[2] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, F. Wu, and Y. Rui, "Efficient 2d-to-3d correspondence filtering for scalable 3d object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 899–906.

[3] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[8] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1183–1192.

[9] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks." in *Bmvc*, vol. 1, no. 2, 2016, p. 3.

[10] J. C. John *et al.*, "Introduction to robotics: mechanics and control," *Reading: Addison-Wesley*, 1989.

[11] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2666–2674.

[14] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[15] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 220–226.

[16] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "Usac: a universal framework for random sample consensus," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 2022–2038, 2012.

[17] W. Zhang and J. Kǒsecká, "Nonparametric estimation of multiple structures with outliers," in *Dynamical Vision*. Springer, 2006, pp. 60–74.

[18] R. Toldo and A. Fusiello, "Robust multiple structures estimation with j-linkage," in *European conference on computer vision*. Springer, 2008, pp. 537–547.

[19] Y. Xie, G. Tang, and W. Hoff, "Chess piece recognition using oriented chamfer matching with a comparison to cnn," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 2001–2009.

[20] W. Pei, Y. Xie, and G. Tang, "Spammer detection via combined neural network," in *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2018, pp. 350–364.

[21] Y. Xie, Z. Wang, W. Pei, and G. Tang, "Fast approximation of non-negative sparse recovery via deep learning," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2921–2925.

[22] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[23] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[24] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese, "Generic 3d representation via pose estimation and matching," in *European Conference on Computer Vision*. Springer, 2016, pp. 535–553.

[25] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[26] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[27] Z. Dang, K. Moo Yi, Y. Hu, F. Wang, P. Fua, and M. Salzmann, "Eigendecomposition-free training of deep networks with zero eigenvalue-based losses," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 768–783.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] G. Georgakis, M. A. Reza, A. Mousavian, P.-H. Le, and J. Košecká, "Multiview rgb-d dataset for object instance detection," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 426–434.

[30] M. Zuliani, C. S. Kenney, and B. Manjunath, "The multiransac algorithm and its application to detect planar homographies," in *IEEE International Conference on Image Processing 2005*, vol. 3. IEEE, 2005, pp. III–153.

[31] C. V. Stewart, "Bias in robust estimation caused by discontinuities and multiple structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 818–833, 1997.