LEARNING DENSE VOXEL WISE GEOMETRIC DESCRIPTOR USING SINGLE RGB-D IMAGE

Junhao Wang
Data Science Institute
Columbia University
New York, NY 10027
jw3668@columbia.edu

Shuran Song
Computer Science
Columbia University
New York, NY 10027
Research Projcet Advisor
shurans@cs.columbia.edu

ABSTRACT

Geometric matching is the process of matching local geometry of an object to its "correspondence" in the same scene where the "correspondence" can be defined differently based on the application of the algorithm. In an robotic assembly task, one may be interested in finding the aligned position of object in its package in the same scene in which case a meaningful correspondence can be defined as the matching between the local geometry of the object and its position inside a package. In this research project, I proposed a geometric descriptor learning algorithm based on RGB-D images of objects, Dense 3D Matching Net, that operates on structured 3D representation to learn a dense voxel-wise descriptor and retrieve correspondence in a voxel level between object and its package in a scene. The proposed algorithm can correctly retrieve the correspondence on average 87% of time when 1 l_2 distance match tolerance is allowed. In addition, it exhibits certain degree of understanding of the underlying local geometric of the voxel space.

***K*eywords** Visual Descriptor Learning · Computer Graphics · 3D Vision

1 Introduction and Motivation

How to put an object correctly into its package in an robotic assembly line? This is a well studied question and in fact, the transformation can be estimated with low level handcrafted geometric feature descriptor. However, in the real world situation, the product in an assembly line can alter rapidly within a matter of weeks. A good model to tackle this problem should be able to trained really fast, have higher accuracy and is capable of generalizing to unseen geometry. Tradition methods involve using histogram based geometric feature descriptor[1, 2, 3] which is susceptible to object occlusion and possess minimal level of geometric understanding.In this research project, I proposed and demonstrated the viability of voxel-wise geometric descriptor learned through Dense 3D Matching Net. Instead of directly using RGB-D image, I instead used a truncated signed distance function[4] to first project the RGB-D into another structured volumetric representation. The Dense 3D Match net was trained on tsdf volume. Although, the generalizability of the Dense 3D Matching Net was not specifically tested due to time constraints and lack of computation resources, it was,yet,empirically proofed to be capable of predicting matching between objects and its corresponding position in the packages with high accuracy. In addition, it also acquired low level understanding of local geometry to some extent.

Research Report Organization. In this report, I will first present my approach on the data acquisition process which includes the details from the software that I used, the set up of synthetic data acquiring, information being recorded, and derivation of mathematical formulation wherever is needed. Then I will describe the details of the Dense 3D Matching Net as well as the loss function that is used during training. At the end, I will present qualitative and quantitative results of the learned dense voxel descriptor.

2 Related Work

My work is most related to 3D match[5]. This research project is the direct extension of 3D match under Shuran's guidance. One major difference that distinguishes the approach presented in 3D match from mine is that in the previous paper, a local 3D patch around an interest point is selected and the match training is between local geometry centered around interest point and its correspondence in different image frame. However, in Dense 3D Net,the training is done on every voxel inside the object and their correspondence inside the package in the same scene. In addition, truncated signed distance function(TSDF) is used in this approach in compare to truncated distance function(TDF) that is used in 3D match. This work is also inspired by the Dense Object Nets[6] where a dense visual descriptor is trained. However, his approach directly operates on RGB images whereas my methodology utilizes the richness of information that a depth image can carry.

3 Data Acquisition

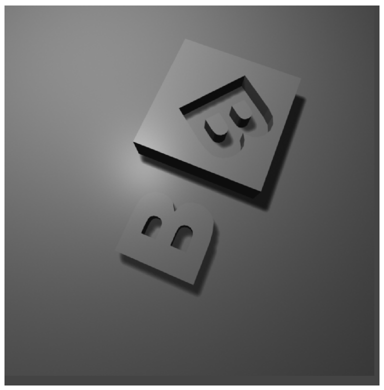
Figure 1: Object and Package

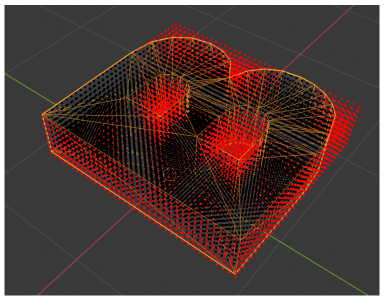
Figure 2: Points Inside the Package

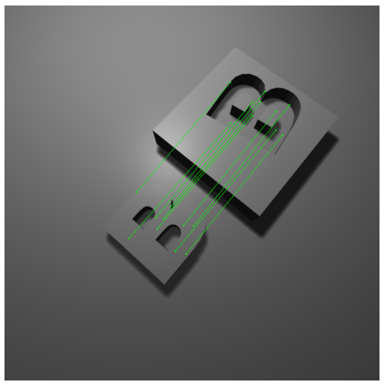
Figure 3: Correspondence

3.1 Data Generation Pipeline

Due to the lack of relevant data set in object and package RGB-D images with dense ground truth correspondence, I generated synthetic object and package mesh for this research project to collect dense ground truth correspondence and RGB-D images. I used an open source 3D modeling software blender to acquire the ground truth matching and RGB-D images of the object and its position inside the package at different camera views and package pose. A mesh with the shape of letter 'B' is used as the object, and complementary space of a rectangular mesh which its corner completely contain the bounding box of letter 'B' is used as the package as shown in **Figure 1**.The mesh letter 'B' is placed at the world origin $(0, 0, 0)^T$. To obtain the dense ground truth matching between letter 'B' and its package, it is necessary to acquire all the point inside a mesh.A dense point cloud over the letter "B" is created and each of point inside the point cloud is evaluated using **Algorithm 1** to determine whether if it is inside the mesh. In the **Figure 2**,one shall observe that all the results of the algorithm where all the red points represent the voxel outside the mesh and the black points represent the voxel inside the mesh. To create the package of different orientations, a new rectangle that fully contain the letter 'B' is created and centered at $(0, 0, 3)^T$. Each time, the letter 'B' is translated along x-axis(to the right) 3 meters and $z - axis$ 0.4 meters and rotated randomly along z-axis so that the upper surface of letter "B" will align to the upper surface of the rectangle mesh. Then a geometry Boolean operation is performed and the resulting mesh will be considered as package. The object translation T and rotation R defines the object's pose inside the package which will be used to generate the correspondence as shown in **Figure 3**. At the same time, the camera is placed at certain height along the z-axis and randomly located along a circular motion of radius r. The camera will always look at the middle point, $(0, 0, 1.5)^T$, between letter 'B' and its package. Each time, the camera record the package and image at random combination of camera pose and package pose. Camera's focal length is carefully tuned such that the RGB-D image will fully contain the content of both object and package.

Algorithm 1: Algorithm to Determine Whether a Point is Inside the Mesh

```
Input  : $P$– dense point cloud grid
Output : $L$ – list of Boolean indicating whether certain point is inside mesh
initialize array L;
for $p_i$ in $P$ do
    $p_{\text{nearest}_i} \leftarrow$ compute nearest point on mesh surface;
    $n_i \leftarrow$ compute the face normal at $p_{\text{nearest}_i}$;
    $p_{\text{direction}_i} \leftarrow p_i - p_{\text{nearest}_i}$;
    if $p_{direction_i} \bullet n_i \leq 0$ then
        $L.append(True)$;
    end
    $L.append(False)$;
end
```

3.2 Information Recorded

- Camera Intrinsic Matrix
- Camera Extrinsic Matrix
- Package Pose
- Grid Points that is inside letter 'B'
- RGB-D Image of object and package

3.3 Truncated Singed Distance Function

In this research project, instead of using RGB and Depth Image, a 2D representation, I projected RGB-D Image into structured volumetric representation using truncated signed distance function. Analogous to pixel, a structured 2D representation that encodes color information, each voxel in truncated signed distance volume encodes the geometric relationship with respect to the prospective projected surface. Given the recorded information, tsdf volume can be computed using **Algorithm 2**. There are three important parameters in TSDF volume computation, voxel grid, voxel size,and margin. For this research project, the bounding box contain the letter 'B' range from -1 meter to 2 meter, -2 meter to 2 meter, 0 meter to 1 meter along x-axis, y-axis and z-axis respectively in the world space. The voxel size used in this research project is 0.1 meter. The margin is set to 0.5.

Algorithm 2: TSDF Integration Using Single Depth Image

```
Input  : $I$ – Depth Image
         $K$ – Camera Intrinsic Matrix
         $RT$ – Camera Extrinsic Matrix
Output : Truncated Signed Distance Function Volume
initialize world grid point $P_{\text{world}}$ ;
for each voxel $p_i$ in $P_{world}$ do
    $p_{camera-view_i} \leftarrow RT \bullet \mathrm{P}_i$;
    $p_{pixel_i} \leftarrow K \bullet P_{\text{camera-view}}$;
    if $P_{pixel_i}$ in the view frustum then
        $sdf_i \leftarrow I_{pixel_i} - Depth(P_{pixel_i})$;
        if $I_{pixel_i} \geq 0$ and $sdf_i \geq -margin$ then
            $tsdf_i \leftarrow \min(1, \frac{sdf_i}{margin})$;
            store $tsdf_i$ at voxel $p_i$
```

3.4 Correspondence

After obtaining the tsdf volume for the object, for each voxel that is inside the object, the corresponding voxel in the package will be calculated using the transformation matrix T that is saved during data generation pipeline,that

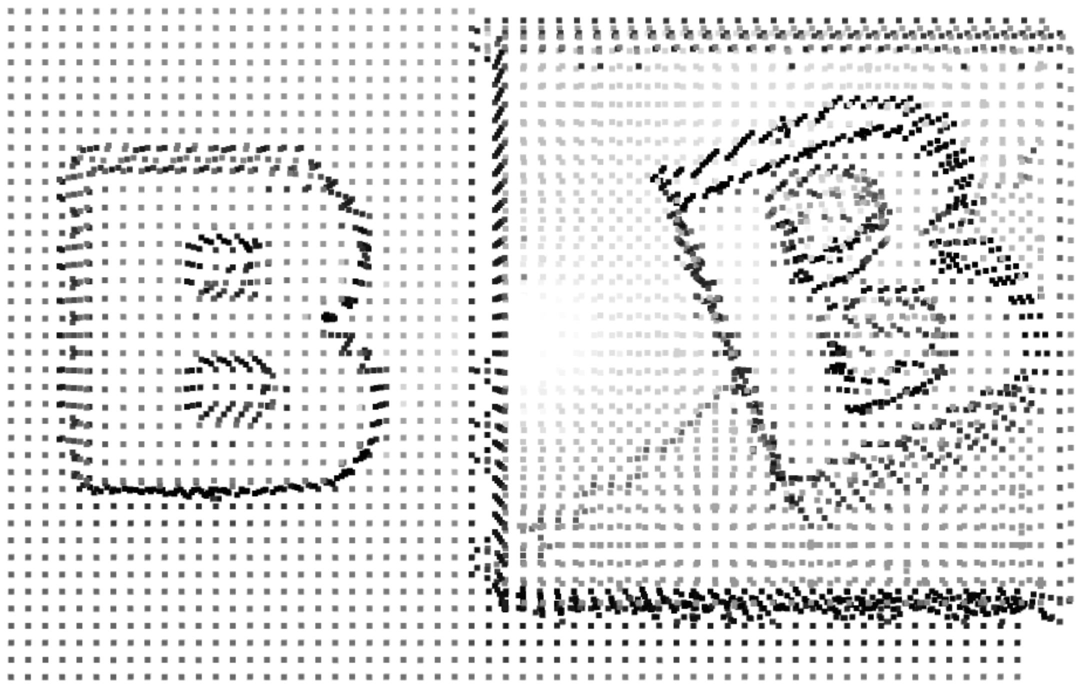

Figure 4: TSDF Volume

is $p_{package} = T \bullet P_{object}$, and $p_{package}$ will be rounded to closest integer. In such way, we can obtain the ground truth in the 3D space despite the occlusion of object and package in the RGB-D image space. For each non-matching correspondence, I randomly sample the $p_{non-match-package}$ in the entire voxel space of package, meanwhile making sure the non-matching voxel point is at least certain threshold away from the ground truth matching voxel point to compensate the inaccuracy that is incurred due to single RGB-D projective truncated signed distance function integration. **Figure 4** shows the point cloud of reconstructed 3D scene of object and package using tsdf.

4 Dense 3D Match Net

4.1 Architecture

To learn a voexl wise dense geometric descriptor, the function f need to map input tsdf volume ,$G \in R^{L \times W \times H \times 1}$, where L indicates length, W indicates width, H indicates height to $f(G) \in R^{L \times W \times H \times F}$ where F indicates feature space. A 3D U-net encoder decoder model is used as the mapping function[7]. 3D U-net is a state of art encoder decoder network that has been broadly used in many representation learning task. The model consists of three down sampling modules, and three up sampling modules. The down sampling modules each is composed of two 3D convolution each followed by a batch normalization and relu activation and followed by a 3D max-pooling. After the each down-sample module, the current tensor is also recorded and will be concatenated with tensor input in the up sampling module at corresponding level. The up sampling module will have similar architecture except the last operation is 3D up pulling. The structure of the model is shown in **Figure 5**.

4.2 Training

The model is trained as the way in Siamese Fashion network[8]. Two parallel 3D U-Nets each will take a batch of tsdf volume of objects and packages respectively. In addition, the two parallel 3D U-Nets also share the model weights. After both 3D U-Nets map each tsdf volume to its corresponding feature space, a contrastive loss is used to minimize

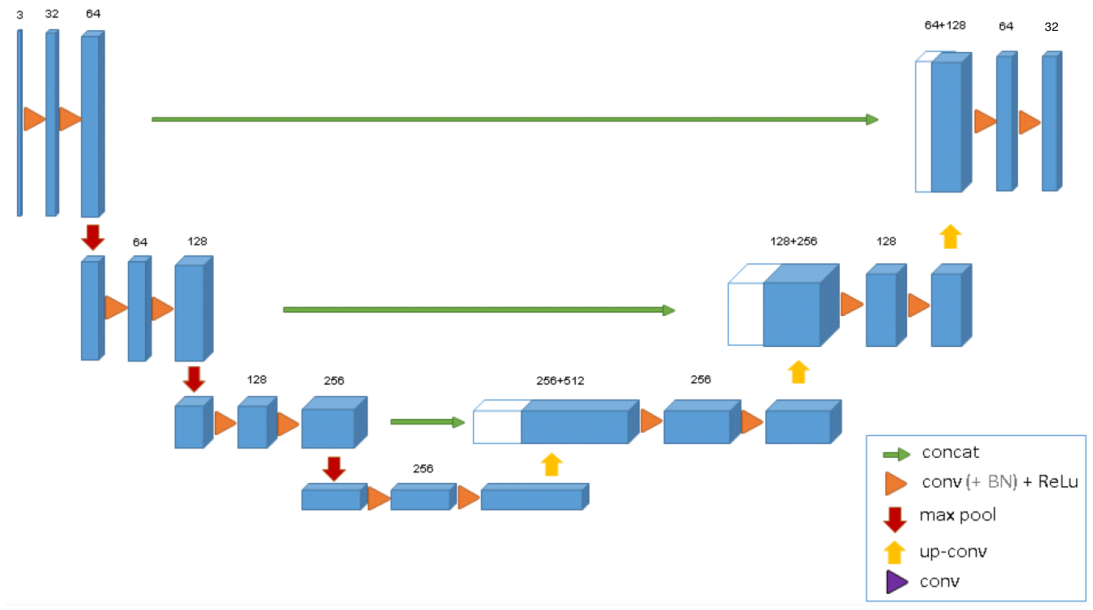

Figure 5: 3D U-Net Architecture

the l_2 norm between geometric descriptors of true correspondences,$D_a, D_{a_{\text{match}}}$, and maximize the l_2 norm between non-matching geometric descriptors, $D_a, D_{a_{\text{non-match}}}$. The purpose of contrastive loss is to not only train the model with correct ground truth, but also explicitly expose the model to non-matching descriptors such that during the testing the model will not make as many false positives. The loss function is described below

$$\mathcal{L}_{matches}\left(D_a, D_{a_{match}}\right) = \frac{1}{N_{\text{matches}}} \sum_{N_{\text{matches}}} \|D_a - D_{a_{\text{match}}}\|_2$$

$$\mathcal{L}_{\text{non-matches}}\left(D_a, D_{a_{\text{non-match}}}\right) = \frac{1}{N_{\text{non-matches}}} \sum_{N_{\text{non-matches}}} \max(0, M - ||D_a - D_{a_{\text{non-match}}}||_2)$$

$$\mathcal{L}\left(I_a, I_b\right) = \mathcal{L}_{\text{matches}}\left(D_a, D_{a_{\text{match}}}\right) + \mathcal{L}_{\text{non-matches}}\left(D_a, D_{a_{\text{non-match}}}\right)$$

However,it was also finds out during training the number of non-matches will quickly drop below 1%. Followed procedure in dense object net [6], I also used hard negative scaling in the contrastive loss function:

$$N_{\text{hard-negatives}} = \sum_{N_{\text{non-matches}}} \mathbb{1}(M - ||D_a, D_{a_{\text{non-match}}}||_2) > 0)$$

$$\mathcal{L}_{\text{non-matches}}\left(D_a, D_{a_{\text{non-match}}}\right) = \frac{1}{N_{\text{hard-negatives}}} \sum_{N_{\text{non-match}}} \max(0, M - ||D_a - D_{a_{\text{non-match}}}||_2)$$

During training, I found out using very dense matching and non-matching helps in terms of learning the geometric voxel descriptor. Thus each time, a million of matching voxel and a million of non-matching voxel were sampled from object and packages. The margin M was set to 0.5 and Adam optimizer was being used. The training was done using google colab GPU. With a batch size of 8, on average it took 40 seconds to train a single step. The model was trained for 30 epoches and results will be discussed in the following section.

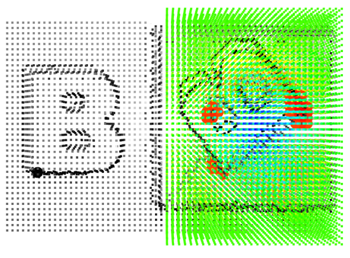
Figure 6: Heatmap Visualization (a)

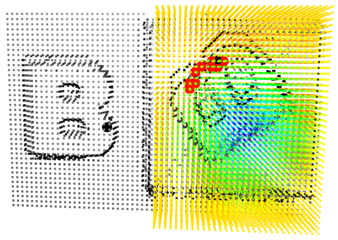
Figure 7: Heatmap Visualization (b)

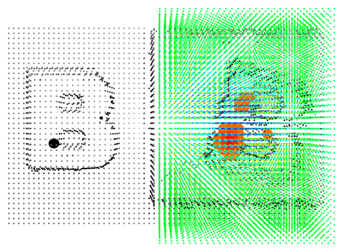
Figure 8: Heatmap Visualization (c)

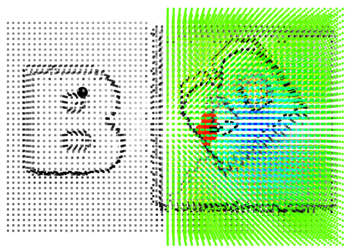
Figure 9: Heatmap Visualization (d)

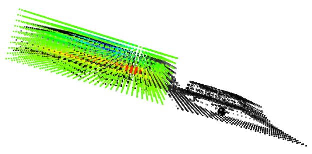
Figure 10: Heatmap Visualization (e)

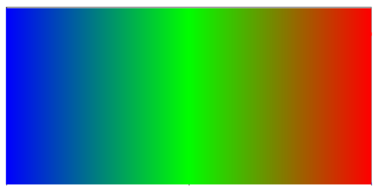
Figure 11: Color Range

5 Results

5.1 Qualitative Assessment

In the **Figure 6**,**Figure 7**, **Figure 8**,**Figure 9**, and **Figure 10**, on the left and right are object and package respectively as reconstructed point cloud using truncated singed distance function. The black dot in the object and package represents the true correspondence in the voxel space. A dense 3D voxel-wise heat-map is placed on top of the package to visualize the l_2 norm between learned descriptor of interest point, black dot in object, and every voxel in the bounding box that contain the package. The color palette is shown in **Figure 11** range from blue to red where blue indicates the highest difference an red indicates the smallest l_2 difference between two geometric descriptor. The red dot in the package voxel space indicates the top 30 matches based on descriptor l_2 distance difference. From **Figure 6**,**Figure 7**, **Figure 8**, one can observe that the geometric voxel descriptor is able reasoning about the local geometry of correspondence between object and its position in the package given that it tends to match corner of the object to several corner positions and curved surface to several curved surface positions. **Figure 9**,**Figure 10** are the tsdf reconstruction of the same testing pair from different angles. The learned voxel descriptor is able to inference about the height, z-axis, in a sense that the level that contain the ground truth correspondence have reddish color which indicates smaller l_2 distance.

5.2 quantitative assessment

During the testing, I test on three different scenarios based on the difference of voxel descriptor between point of interest in the object and voxel in the package: interest point voxel matches exactly to the groud truth correspondence voxel in the package, interest point voxel matches to a voxel that is within one l_2 distance from ground truth correspondence voxel in the package,interest point voxel matches to a voxel that is within two l_2 distance from ground truth correspondence voxel in the package. Every voxel inside the object mesh is tested with above procedure and 66 object package pairs from different camera pose and package pose to provide the distribution of matching count. There are total of 7414 voxel inside every object. **Figure 12** displays the matching distribution over the 66 testing pairs. Although using exact matching metric, the model is only able to correctly on average less than half of the object voxel, when relax the metric with 1 l_2 distance tolerance, the matching performance improves significantly with on average around 87% correct matching. In case of 2 distance tolerance, the model is able to correctly match on average 98% voxel in the object to package voxel based on learned geometric dense descriptor. In **Table 1**, a detailed matching accuracy percentile is displayed.

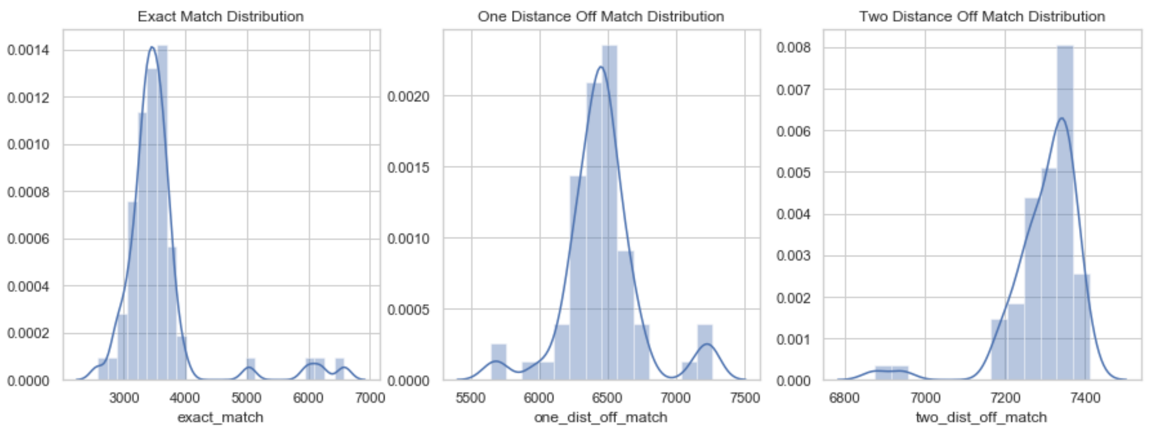

Figure 12: Match Count Distribution

Percentile	Exact Match Accuracy	1 Distance Off Match Accuracy	2 Distance Off Match Accuracy
10	0.428%	0.839%	0.973%
20	0.443%	0.850%	0.978%
30	0.449%	0.860%	0.981%
40	0.460%	0.865%	0.984%
50	0.466%	0.870%	0.988%
60	0.477%	0.875%	0.989%
70	0.485%	0.881%	0.991%
80	0.494%	0.889%	0.992%
90	0.482%	0.902%	0.994%

Table 1: Match Accuracy Percentile Table Over 66 Testing Object Package Pair

6 Feature Work

Generalizability Although dense 3D Match Net demonstrates its feasibility to understand low level geometric relationship and match correspondence with promising accuracy,one important question yet to be answered is its ability to generalize. Since for object, only a single geometry is used, it is not clearly on how well will the learned descriptor perform in unseen mesh with potentially very different low level geometry. This questions can be further addressed by including multiple geometry mesh each with very unique shape, and designed a meta learning training scheme and perform few shot matching task.

Muti-Modality So far the Model only using single feature space, tsdf volume which only use depth Image, to train the Dense 3D Matching Net. However, it can be improved to include additional modalities to facilite the signal recevied by the model. In future work, RGB color channel will be included in the tsdf volume in which case, the gradients of the color encoded in the geometric voxel space may provide additional information for the model to understand the low level geometry of object and package.

Integration in Other Vision-Based Tasks With the learned descriptor, it can be used as a matching module to integrate into other model pipeline to address low level geometric matching. A direct example would be integrate this matching module into robotic manipulation tasks. For example, during a robotic package assembly pipeline,the learned descriptor as matching module can provide geometric matching of interests points. Such points can not only be used for pose estimation between object and package,but also it can give guidance on the formulation of the package assembly task.

References

[1] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. 2004.

[2] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[3] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. 1996.

[5] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[6] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[8] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification.