Query Adaptive Similarity Measure for RGB-D Object Recognition

Yanhua Cheng[1]*, Rui Cai[2], Chi Zhang[4], Zhiwei Li[2], Xin Zhao[1], Kaiqi Huang[1,3], Tieniu Tan[1,3], Yong Rui[2]

[1]CRIPAC&NLPR, CASIA [2]Microsoft Research
[3]CAS Center for Excellence in Brain Science and Intelligence Technology
[4]Sun Yat-Sen University
Email:{yh.cheng, xzhao, kaiqi.huang, tnt}@nlpr.ia.ac.cn, {ruicai, v-cz, zli, yongrui}@microsoft.com

Abstract

This paper studies the problem of improving the top-1 accuracy of RGB-D object recognition. Despite of the impressive top-5 accuracies achieved by existing methods, their top-1 accuracies are not very satisfactory. The reasons are in two-fold: (1) existing similarity measures are sensitive to object pose and scale changes, as well as intra-class variations; and (2) effectively fusing RGB and depth cues is still an open problem. To address these problems, this paper first proposes a new similarity measure based on dense matching, through which objects in comparison are warped and aligned, to better tolerate variations. Towards RGB and depth fusion, we argue that a constant and golden weight doesn't exist. The two modalities have varying contributions when comparing objects from different categories. To capture such a dynamic characteristic, a group of matchers equipped with various fusion weights is constructed, to explore the responses of dense matching under different fusion configurations. All the response scores are finally merged following a learning-to-combination way, which provides quite good generalization ability in practice. The proposed approach win the best results on several public benchmarks, e.g., achieves 92.7% top-1 test accuracy on the Washington RGB-D object dataset, with a 5.1% improvement over the state-of-the-art.

1. Introduction

RGB-D object recognition has now become an active research area with the rapid development of commodity depth cameras. These depth cameras, such as Kinect, are capable of recording synchronized color and depth data, which together provide rich multimodal information to depict an object. In the past year, a noticeable trend is that depth camera has been integrated into mobile devices like Google Tango [1] and Microsoft Hololens [2], which offer an even appealing platform for RGB-D object recognition.

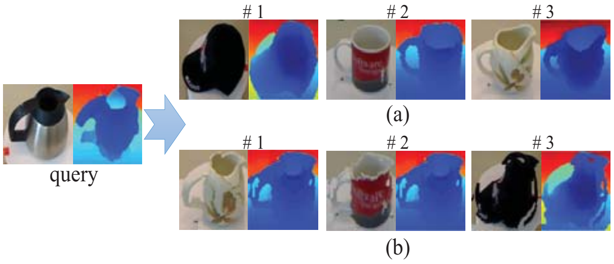


Figure 1. Given a query *pitcher*, (a) shows the top-3 categories predicted by the state-of-the-art CNN-RNN method [30], which wrongly promotes the *cap* due to its black color; and (b) shows the results of the proposed approach, in which the other *pitcher* is correctly ranked at the first position. Please be noted that in (b), both the color and depth images are warped according to the query.

Remarkable research efforts have been invested in recent years. Some of them focus on devising elaborated features for both RGB and depth channels [20, 8, 5]; while others study exploiting machine learning techniques to combine cues of the two modalities [21, 4, 7, 30]. All these methods have made significant progress to advance the state-of-the-art, *e.g.*, several approaches [5, 4, 7, 30] were observed to achieve around 98% top-5 accuracy on the very challenging Washington RGB-D object dataset [20]. Such an impressive performance indeed demonstrates the advantages of RGB-D data for object recognition, also provides a solid base for follow-up studies. Towards the demands of practical scenarios, especially of those applications on mobile platform, the major concern of existing approaches is their top-1 accuracies are not satisfactory enough. Averagely, current solutions obtain around 85% top-1 accuracy, which cannot provide a high-quality and stable user experience.

Improving top-1 accuracy of RGB-D object recognition remains a very challenging problem. The reasons are in two-fold. First, objects from the same category may appear in different poses and scales, as well as have a certain degree

[*] The work was performed at Microsoft Research.

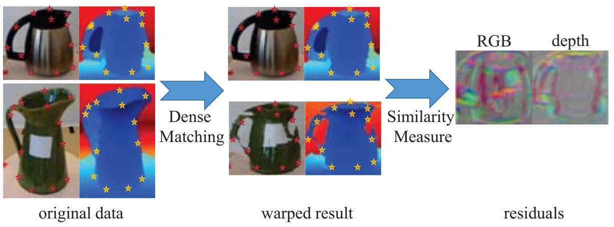


Figure 2. An illustration of the proposed dense matching-based similarity measure.

of intra-class variations. This brings troubles to most existing RGB and depth features, and tends to cause false negative decisions. Second, objects from different categories but having ambiguous texture appearances or geometric shapes usually confuse the strategies adopted to combine RGB and depth cues, and lead to false positive errors. For example, Fig. 1 (a) shows the top-3 objects considered to be the closest ones to the query *pitcher* by the state-of-the-art CNN-RNN method [30]. The *cap* ranked at the first position, which is clearly a false positive, is wrongly promoted by its black color. By contrast, the other *pitcher* ranked at the third position is demoted as its shape and appearance have some variations to the query.

This paper aims at improving the top-1 accuracy of RGB-D object recognition. Towards the aforementioned problems, we first present a dense matching-based similarity measure to better tolerate object shape, pose, and scale variations. Then a simple yet effective learning-to-combination strategy is devised to boost the recognition accuracy based on the dense matching similarities. Fig. 1 (b) shows the result generated by the proposed approach. The other *pitcher* is now correctly ranked at the first position.

Dense Matching. Object similarities are mainly based on the distance of pre-selected RGB and depth features. Although some of these features (*e.g.*, SIFT [25], kernel [5], and convolutional descriptors [4, 30]) can tolerate a small degree of local pose, scale, and color variations, they still failed to deal with the tough *pitcher* case in Fig. 1. It is noticed that almost all of these existing features are computed solely based on the host object itself. In other words, features of an object remain the same, independently of the object it is compared to. This could be arguable as people maybe adopt a different way to distinguish objects. For example, to compare the two *pitchers* in Fig. 1, a more natural way is to first align them to the same pose and size, and then make comparisons part-by-part (*e.g.* the handle and spout parts of a *pitcher*). In this way, the description (features) of an object should be adaptive when it is compared with different objects. This is the first motivation of this paper.

One pioneer work inspiring us is the SIFT flow proposed in [24, 23]. Unlike traditional optical flow [16, 9, 10], which merely estimates the motion of the same object along time, SIFT flow is capable of building correspondences between different objects from the same category. For example, aligning the wheel parts of different models of cars. Following the idea of SIFT flow, in this paper, dense matching is proposed to transform (or warp) one object to another, to align semantically corresponding parts of the two objects. Like that shown in Fig. 2, a reference object is first warped according to the query through dense matching. After that, the similarity is measured based on the distorted object and the query. We call this *query adaptive similarity measure* because the dense matching alignment depends on the query object.

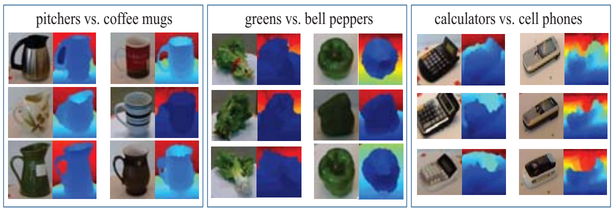


Figure 3. Example pairs of object categories being easily confused with each other. They usually share similar visual appearances or geometric shapes, which makes the choosing of fusion weight a tough problem. Please refer to the text for details.

Learning-to-Combination. In dense matching, fusing RGB and depth data is still an unavoidable problem. Particularly, we need to specify a weight to combine the agreements of local RGB and depth patches. No doubt that choosing different weights leads to entirely different object alignments. A straightforward idea is to learn a magic weight to balance the two modalities based on a training set. Here, we argue that such an ideally golden fusion weight doesn't exist at all. The truth is, the two modalities have varying contributions when comparing objects in different categories. This is the second motivation of this paper.

Fig. 3 illustrates several easily confused object categories to explain our observations. For instance, we expect to emphasize shape (depth) to boost the similarity between various *pitchers*; while at the same time we also want to emphasize texture appearance (RGB) to distinguish *pitchers* from *coffee mugs*. Also, there are situations that both RGB and depth cues have to be considered to make a right decision, such as to distinguish *calculators* from *cell phones*.

To better balance RGB and depth cues, a learning-to-combination strategy is introduced in this paper. Given a fusion weight, dense matching tries to align objects following a certain assumption of the contributions from the two modalities, and the output cost reflects how well the objects are matched under such an assumption. Through varying the weight, we construct a group of dense matchers, whose outputs together characterize the trend of similarity change under different fusion configurations. In analogy to learning-to-rank in web search, supervised learning is feasible here to find an optimal way combining responses of all the matchers, to promote query-relevant objects in recognition. Although such a learning-to-combination strategy is quite simple, it shows surprisingly good generalization ability in experiments.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 and 4 introduce details of the proposed approach. Extensive experimental results are reported in Section 5 and conclusion is drawn in Section 6.

2. Related Work

RGB-D Object Recognition. A fair amount of research efforts focused on devising RGB and depth features. Lai *et al.* [20] extracted SIFT [25], texton histograms [22] and spin images [19] over the color and depth frames individually, followed by a concatenation of all these features to depict an object. Allowing for the specificity of the depth information, depth kernel features [5] and local surface descriptors [3] were designed to represent the depth cues more effectively. Another line of work [4, 7, 30, 31] exploited very successful machine learning methods to obtain state-of-the-art performance. Browatzki *et al.* [8] exploited a multi-layer perception to fuse multiple SVMs learned on separate features. Lai *et al.* [21] extended the distance learning methods [29, 26] to define a view-to-object distance, which can effectively fuse heterogenous features. In this paper, we investigate the problem from another perspective, focusing on designing a kind of query dependent similarity measure as well as the corresponding fusing strategy. In experiments we will show that the proposed approach is a good complementary to existing solutions.

Dense Matching. The nature of dense matching is to insure some kind of agreement between matched pixels, as well as enforce smoothness in their local neighborhoods. Such a formulation has been extensively studied in optical flow [16, 9, 10] to align temporally adjacent frames via brightness constancy. Recently, the additional depth data was also exploited in RGB-D flow [15, 28] to estimate motion field in 3D space. Both optical flow and RGB-D flow are limited to track the same object along time (and in space). SIFT flow [24, 23] relaxes such a limitation, and is able to match similar objects in different scenes. This paper extends the framework of SIFT flow to align objects, taking the agreements of both RGB and depth modalities into consideration.

3. Our Approach

3.1. Dense Matching for Similarity Measure

To match a query object I_q and a reference object I_r in database, I_r is first warped according to I_q by dense matching. The goal is to reduce the distractions of possible variations, and provide more robust inputs to further similarity measure. Essentially, dense matching is to construct a mapping relation $\mathcal{M}$: $I_r \rightarrow I_q$, in which each pixel in I_r is associated with one pixel in I_q, by

$$(x', y') = (x, y) + (d_x, d_y), \tag{1}$$

where (x, y) is the coordinate of a pixel p_i in I_r, and (x', y') is the coordinate of the matched pixel p'_i in I_q. (d_x, d_y) is the displacement of p_i, and is denoted by the vector $\mathbf{t}_i = (d_x, d_y)$ for the sake of simplicity.

To optimize the displacements of all the pixels as a whole, following the formulation of typical optical flow [16, 9, 10] and SIFT flow methods [24, 23], here in dense matching we consider three constraints including agreements of local RGB and depth patches (data term), strength of displacement (range term), and neighborhood smoothness (smoothing term). The total cost is defined as

$$E(\mathbf{t}_i) = \underbrace{\sum_i D_i(\mathbf{t}_i)}_{\text{data term}} + \alpha \underbrace{\sum_i \|\mathbf{t}_i\|_1}_{\text{range term}} + \beta \underbrace{\sum_{i,j \in \mathcal{N}} \min(\|\mathbf{t}_i - \mathbf{t}_j\|_1, \lambda)}_{\text{smoothing term}}. \tag{2}$$

Here the range term, expressed as the L_1 norm of the displacement vector, penalizes unexpected large deformations. The smoothing term constrains the displacement vectors of adjacent pixels to be similar, where $\mathcal{N}$ is a 2×2 spatial neighborhood, and the L_1 norm truncated by λ is adopted to account for the discontinuities of local object boundaries. In addition, α and β are two constant coefficients to balance different terms. The settings of α, β, and λ follow the SIFT flow work, more details please refer to [23].

The data term in (2) differs from those adopted in optical flow and SIFT flow. Besides visual agreement, depth agreement also contributes to this constraint. Together we have

$$\begin{aligned} D_i(\mathbf{t}_i) = \theta \cdot [f_{rgb}(p_i|I_r) - f_{rgb}(p_i + \mathbf{t}_i|I_q)] + \\ (1-\theta) \cdot [f_{depth}(p_i|I_r) - f_{depth}(p_i + \mathbf{t}_i|I_q)]. \end{aligned} \tag{3}$$

Here, $f_{rgb}(p|I)$ and $f_{depth}(p|I)$ are respectively the RGB and depth feature descriptors extracted at the point p from the object I, and $\theta \in [0, 1]$ is a specified fusion weight to combine RGB and depth cues.

Optimization. Following the computational framework proposed in [23], we exploit a dual-layer loopy belief propagation to minimize the discrete energy function. To speed up the algorithm, a coarse-to-fine matching scheme is employed as well. For both modalities (depth and RGB), maps of their features are down-sampled to construct a multi-resolution pyramid. The mapping relation $\mathcal{M}$ is roughly estimated at the coarsest resolution, and is then propagated and refined layer-by-layer, until reaches to the finest resolution. Such a scheme significantly reduces the computation time, also generates more robust result.

Conditional Similarity. Given the optimized mapping relation $\mathcal{M}$ from I_r to I_q, we define the conditional similarity as

$$s_\theta(I_r|I_q) = \exp(-\gamma^{-1} \sum_i D_i(\mathbf{t}_i)), \tag{4}$$

where γ is the scale parameter, which is empirically set as the average data term cost over the training set. The score explicitly evaluates the local appearance and shape agreements between the two objects, pixel-by-pixel, adopting a certain fusion weight θ to balance the two cues. It is worth noting that the conditional similarity is asymmetric, since the dense matching is query dependent. In general, the matching between objects from the same category usually converges with a relatively small cost, and generates a high similarity score according to (4). By contrast, the cost of matching objects from different categories (*e.g.* aligning the *cap* to the query *pitcher* in Fig. 1 (b)) is expensive, as a consequence, results in a poor similarity score.

3.2. Learning-to-Combine Dense Matchers

Although the proposed conditional similarity is quite simple (basically it is directly based on the sum of pixel-wise feature distances), as will be shown in Section 5, a single dense matcher is capable of providing top-1 test accuracy ranged from about 83% to 88%, adopting different fusion weight θ. Such a performance is already comparable with those of most existing solutions, and proves the rationality of the dense matching idea. However, just like we explained in Section 1, it is hard to further improve the performance as a constant fusion weight cannot deal with all the complicated situations when comparing objects from various categories.

Through studying the behaviors of a single dense matcher, it is found the trend of matching cost varying along with the fusion weight θ also reveals some sort of information. For example, occasionally the cost may be relatively stable within a certain range of θ, but changes dramatically out of that interval. In view of similar observations, a natural thought is to construct a group of dense matchers equipped with various θ, and explore the change of matching scores, expecting to provide more clues to boost the recognition accuracy. In particular, θ is gradually varied from depth-dominated ($\theta = 0$) to RGB-dominated ($\theta = 1$), and all the responding similarity scores are merged in a linear way. That is, the combined similarity score between the query I_q and the reference I_r can be written as

$$s_{sum}(I_r|I_q) = \sum_\theta w_\theta \times s_\theta(I_r|I_q) + b = \mathbf{w}^\top \mathbf{\Phi}_{I_r|I_q}, \tag{5}$$

where the fusion weight θ is uniformly sampled from the interval $[0.0, 1.0]$ with a stride δ_θ (*i.e.*, $\theta = [0.0 : \delta_\theta : 1.0]$), $\mathbf{\Phi}_{I_r|I_q} = \{s_{0.0}(I_r|I_q), \cdots, s_{1.0}(I_r|I_q), 1\}$ is the vector of dense matcher scores, and $\mathbf{w} = \{w_{0.0}, \cdots, w_{1.0}, b\}$ is the vector of combining weights. A good choice of the combining weights $\mathbf{w}$ should satisfy $s_{sum}(I_q^+|I_q) > s_{sum}(I_q^-|I_q)$, where I_q^+ denotes objects from the same category of q and I_q^- indicates objects from other categories.

Training data collection. It is critical to collect a set of high quality training data to learn an optimal $\mathbf{w}$. Randomly selecting positive pairs (two objects from the same category) and negative pairs (two objects from different categories) is a possible way, but may not be beneficial to improve the accuracy. This is because random sampling cannot collect enough tough cases as most object categories are not difficult to be distinguished from each other. A more convenient way to collect easily confused samples is exploiting the power of exiting methods [5, 4, 7, 30]. Specifically, the training set is constructed as follows. First, a held-out validation subset is used to query the remaining dataset (those test samples to be adopted for experimental evaluation have been excluded before this step), using one of those off-the-shelf approaches. For each query I_i, there are usually both positive I_i^+ and negative I_i^- samples among its retrieved neighboring objects. As a result, a set of triplets $\Omega = \{(I_i, {I_i}^+, {I_i}^-)\}_{i=1}^N$ is constructed for learning the combining weights $\mathbf{w}$, which should promote $s_{sum}(I_i^+|I_i)$ and demote $s_{sum}(I_i^-|I_i)$.

Ranking SVM. In analogy to the formulation of structured learning-to-rank [17], $\mathbf{w}$ can be optimized through minimizing the following loss function

$$\begin{aligned} &\min \tfrac{1}{2} \|\mathbf{w}\|_2^2 + C \textstyle\sum \xi_{i,i^+,i^-} \\ &s.t. \forall (I_i, {I_i}^+, {I_i}^-) \in \Omega, \quad \xi_{i,i^+,i^-} \geq 0, \\ &\quad \mathbf{w}^\top \mathbf{\Phi}_{I_i^+|I_i} - \mathbf{w}^\top \mathbf{\Phi}_{I_i^-|I_i} > 1 - \xi_{i,i^+,i^-}. \end{aligned} \tag{6}$$

As with those parameters in classical SVM, here C is a non-negative tuning parameter and $\{\xi_{i,i^+,i^-}\}_{i=1}^N$ are slack variables to tolerate some degrees of ranking error. Although this is not a convex or differentiable problem, the cutting plane algorithm can efficiently minimize the upper bound of the loss function [17].

Ideally, a separate weight vector $\mathbf{w}$ should be optimized for each object category (it is straightforward to extend (6) to do this). While in practice, unfortunately, it was found that over-fitting is inevitable due to the limited training samples from each category available in those public RGB-D datasets [20, 8]. As a compromise, we learn a global $\mathbf{w}$ to adapt all the categories, which shows very promising generalization ability in experiments.

Recognition. Finally, object recognition is carried out through voting. Similar to the steps for collecting training data, for a test query I_t, a pre-selected off-the-shelf approach is first adopted to identify T candidate categories, and retrieve the K nearest objects for every candidate category. Then, the score of I_t belonging to the category

$c_i, 1 \leq i \leq T$ is defined as

$$s_{vote}(I_t; c_i) = \frac{1}{K} \sum_{I_r \in c_i} s_{sum}(I_r | I_t), \tag{7}$$

where I_r is one of the retrieved top-K nearest examples of category c_i. At last, I_t is recognized as an object of the category associated with the highest voting score, as

$$category(I_t) = \underset{c_i}{argmax}\, s_{vote}(I_t; c_i). \tag{8}$$

4. Implementation Details

Dense Matching. To characterize the local patch structures, for each modality, two kinds of descriptors are extracted for every pixel. One is the 128-dimensional convolutional descriptor based on a 9×9 patch centered by a pixel [14]; the other is a 200-dimensional kernel descriptor, consisting of the gradient kernel and LBP kernel proposed in [5]. The two descriptors are concatenated and compressed through PCA dimension reduction, finally resulting in a 256-dimensional feature vector for each pixel of each modality.

Following the instructions in [23], we assign $\alpha = 0.05$, $\beta = 2$, and $\lambda = 20$ for the energy function (2) of dense matching. It takes approximately 9 seconds to compute the dense matching between an pair of objects, using a single thread of a 2.4GHz Intel processor. Of course in recognition, the matching process can be easily speeded up through multi-threading parallelization on multiple cores or machines.

Learning-to-Combination. In the learning step, we adopted the CNN-RNN method [30] to collect training triplets, and utilized SVM^{rank} [18] to learn the combination weights. For recognition, we investigated the impacts of choosing different values for the parameters δ_θ, T, and K, and report the experimental results in the next section. Also, we will show that the proposed approach can be integrated with various existing solutions and generally improve their performance.

5. Experiments

5.1. Experiment Setup

Datasets. We evaluated the proposed approach on two public available benchmark datasets, the Washington RGB-D object dataset [20] and the 2D3D object dataset [8].

The Washington RGB-D object dataset is a large-scale and multi-view dataset collected by Microsoft Kinect. It consists of 300 household objects grouped into 51 categories. Each object was captured from 3 vertical angles as well as multiple horizontal angles, resulting roughly 600 images per object. Following the instruction of [20], the 10 trials provided by the dataset were adopted to evaluate the average accuracy.

Table 1. Top-1 test accuracies of existing methods and the proposed approach on the Washington RGB-D object database.

Methods	Top-1 accuracy (%)
Linear SVM [20]	81.9 ± 2.8
Kernel SVM [20]	83.8 ± 3.5
Random Forest [20]	79.6 ± 4.0
IDL [21]	85.4 ± 3.2
KDES [5]	86.2 ± 2.1
CKM [4]	86.4 ± 2.3
HMP [6]	82.1 ± 3.3
SP-HMP [7]	87.5 ± 2.9
CNN-RNN [30]	87.6 ± 2.0
CRNN+CT [12]	87.2 ± 1.1
Our Method	**92.7** ± 1.0

Table 2. Top-1 test accuracies of existing methods and the proposed approach on the 2D3D object database.

Methods	Top-1 accuracy (%)
MLP [8]	82.8
KDES [5]	92.8 ± 1.5
CKM [4]	88.7 ± 2.3
SP-HMP [7]	91.0
CNN-RNN [30]	92.5 ± 1.2
Our Method	**93.6** ± 0.7

The 2D3D object dataset has 156 objects organized into 14 categories, which are popular in typical household or office environments. Each object was recorded every 10° around the vertical axis on a turntable, yielding 36 views per object. Following the same setting of [8], we randomly split the dataset into a training set (82 objects) and a testing set (74 objects). The evaluation also repeated for 10 times to measure the average performance.

Methods. We collected as much as possible performance reported by most well-known methods on the two datasets. Several representative methods [5, 4, 30] only have results for the Washington dataset in literature, to make a more comprehensive comparison, we also implemented these methods and verified them on the 2D3D dataset. For the proposed approach, we chose the CNN-RNN model [30] as the base method to help identify $T = 5$ candidate object categories, each of which contains $K = 10$ nearest exemplars for voting. The learning-to-combination works on 11 dense matchers whose fusion weights θ ranged from $[0, 1]$ with the stride $\delta_\theta = 0.1$. More detailed analysis for selecting these parameters, as well as base method, will be discussed in Section 5.3.

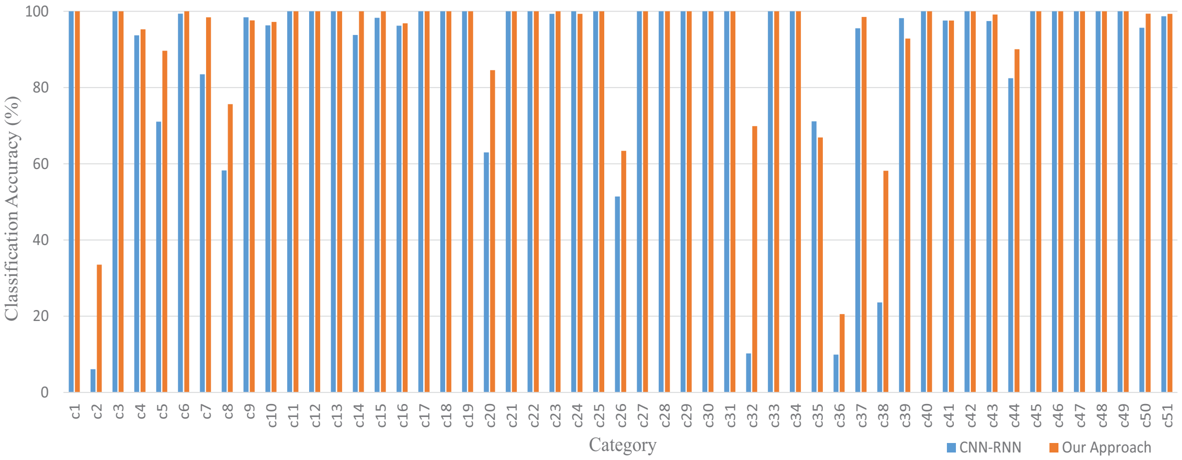


Figure 4. Separate accuracy for each of the 51 categories in the Washington dataset. Our approach clearly improve the performance of several tough categories (*e.g.*, c32 and c38).

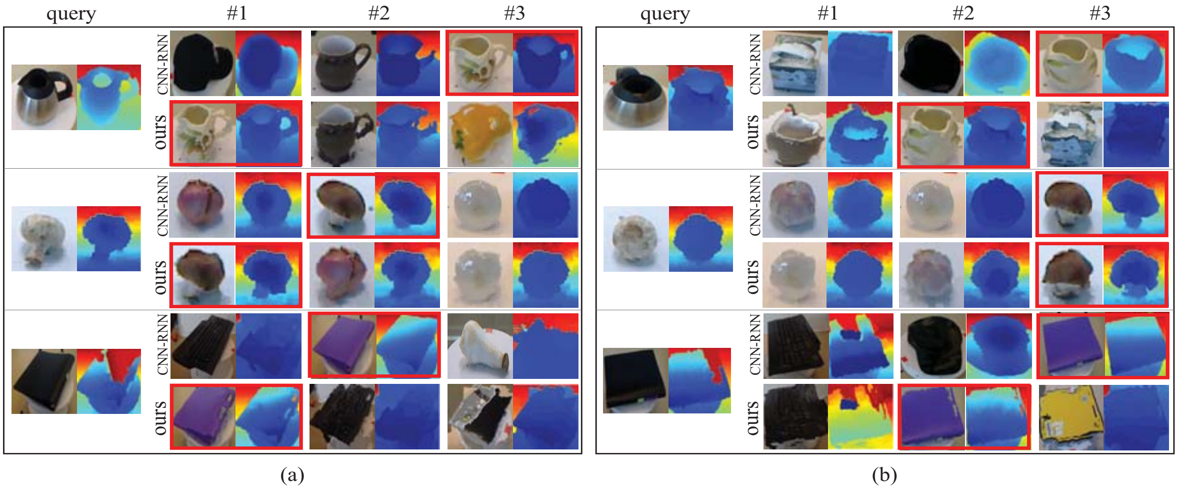


Figure 5. (a) shows some difficult examples that are misclassified by CNN-RNN but correctly recognize by our method. (b) shows some very confused examples which are misclassified by both CNN-RNN and our method. It is noticed that our method ranks the candidate categories more reasonably. (Red rectangle denotes the true positive for the query.)

5.2. Overall Performance

Table 1 and Table 2 respectively present the top-1 recognition accuracies of various methods on the Washington RGB-D object dataset and the 2D3D dataset. All of these methods employ both RGB and depth modalities. On both of the two datasets, the proposed approach achieves the best results. Specially, on the challenging Washington dataset, our approach clearly outperforms existing methods, exceeding the state-of-the-art CNN-RNN method[1] with a 5.1% improvement. On the 2D3D dataset, our approach also improves the top-1 accuracy by around 0.8% over the current best. In addition, our approach has the most stable performance (*i.e.*, the smallest standard deviation) over different trials. In general, the overall performance indeed demonstrates the effectiveness of the proposed solution for boosting the top-1 accuracy of RGB-D object recognition.

To clearly show the performance gains of our approach, a detailed comparison of the separate accuracy for every category on the Washington RGB-D dataset is given in Fig. 4. Referring to the existing state-of-the-art CNN-RNN method as baseline, we observe that our approach can significantly improve the performance of several tough categories, while simultaneously preserving high accuracies for others. Some specific examples are given in Fig. 5 (a). The query *pitcher*, *mushroom* and *binder* are misclassified as *cap*, *garlic* and *keyboard* respectively by CNN-RNN method, due to the

[1] [30] reported its result as 86.8 ± 3.3 with the softmax classifier. For a fair comparison, we report its result with the linear SVM classifier like other methods [5, 6, 4, 7, 12]. Note the results have already been updated by the latest work [13, 11]

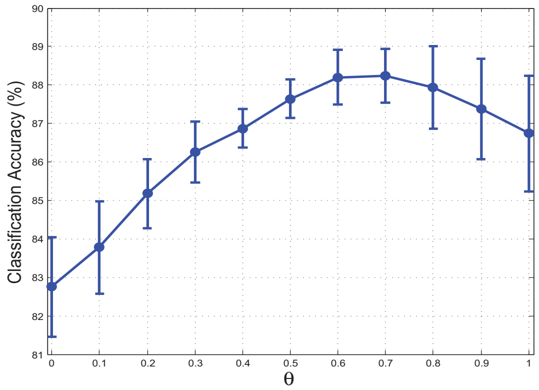


Figure 6. The top-1 accuracy of object recognition based on a single dense matcher equipped with fusion weight $\theta \in [0, 1]$.

confused color and appearance. However, through warping by dense matching, the retrieved objects can be aligned to the query with a similar pose and scale in our similarity measure. As a consequence, the stable local structures of objects can be found for the query, such as the handle and spout of *pitcher*, the stipe and pileus of *mushroom* and the interlayer of *binder*, which are used in our approach to significantly boost the similarity of intra-class objects and lead a correct recognition.

There are some very tough examples misclassified by both CNN-RNN and our method, as showed in Fig. 5 (b). Actually, humans can be confused to tell such query *mushroom* and *binder* since both the appearance and shape of the two queries are not distinguished enough. Although the top ranked object is a false positive, our approach can provide a more reasonable ranking of categories, where the ranking position of the true positive can be promoted by our approach. See the query *pitcher* for instance. The results prove the effectiveness of our approach as well.

5.3. Detailed Analysis

5.3.1 Effectiveness of Dense Matching

In this section, we study the effectiveness of dense matching, through applying a single dense matcher, without the learning-to-combination strategy for object recognition. Following the same setting of the experiment setup, we evaluate the performance of a single dense matcher with different fusion weight θ on the Washington RGB-D dataset. The results are showed in Fig. 6. When $\theta = 0.7$, the single dense matcher can obtain the best result of 88.2% top-1 accuracy, which is competitive to the existing state-of-the-art (CNN-RNN, 87.6%). This proves the rationality of the dense matching idea for object recognition.

However, the performance based on a single dense matcher can be sensitive to θ. For instance, when $\theta = 0$, the performance can decrease to below 83%. The probable reason is that objects from the same category can appear

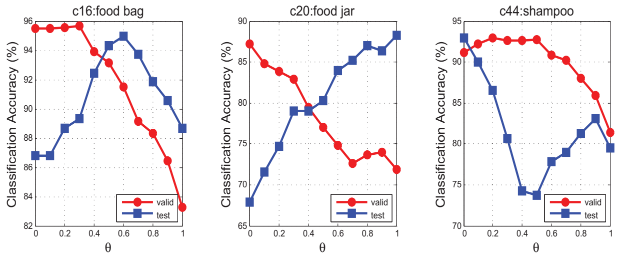


Figure 7. Accuracies of three example categories on both validation and testing sets. Clearly that optimal fusion weights on one set lead to bad performance on the other set. This is an evidence of intra-class variations, which is also one of the bottlenecks of a single dense matcher.

in different poses and scales, as well as have some degree of intra-class variations. Such a single dense matcher cannot deal with all the complicated situations. An evidence is illustrated in Fig. 7. Even belonging to the same class, the object instances can depend on very different fusion weights to distinguish themselves with other categories. The results also demonstrate that devising multiple dense matchers equipped with different fusion weights for object recognition is necessary.

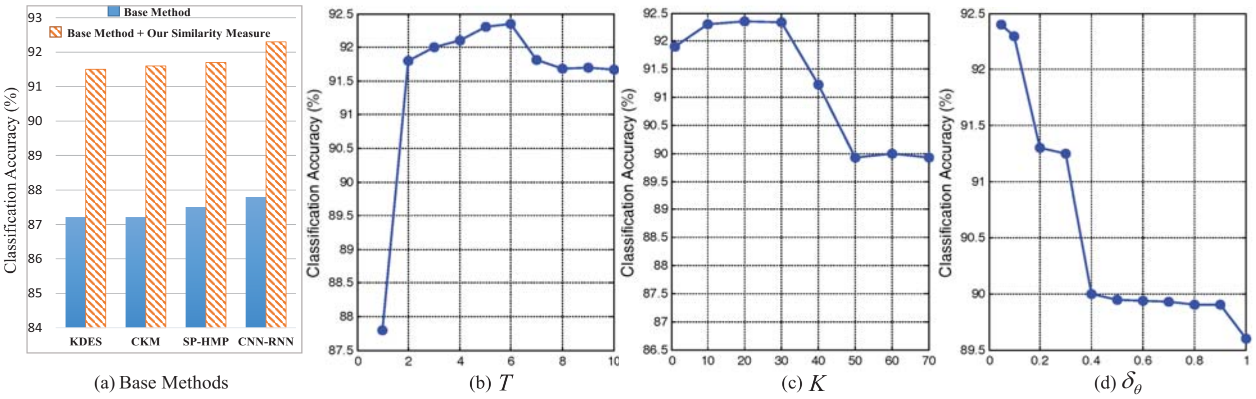
5.3.2 Effectiveness of Learning-to-Combination

In this section, we study the effects of the parameter setting to the performance of learning-to-combination strategy. The similarity measure based on learning-to-combination strategy is closely related to (1) the base method; (2) T; (3) K and (4) δ_θ. The default setting of all the parameters in our experiments is: CNN-RNN based, $T = 5$, $K = 10$ and $\delta_\theta = 0.1$. Now we analyze each parameter of them by fixing others as the default setting. For clarity, all the experimental results below are carried out on the Washington RGB-D dataset over the second train/test split provided by [20] (We empirically find that the result over this split is approximate to the average result.).

Selection of base method. As showed in Fig. 8 (a), the proposed similarity measure can significantly improve the top-1 accuracies of RGB-D object recognition for all the four methods. Note that we only learn the combining weight vector $\mathbf{w}$ based on CNN-RNN model, and directly apply $\mathbf{w}$ for other methods, which shows the good generalization ability for the learning-to-combination strategy.

Selection of T. As showed in Fig. 8 (b), when $T > 2$, our approach can obtain a relatively stable and high accuracy (around 92.0%) of object recognition. Note that when $T = 1$, it actually denotes the result of CNN-RNN model. The results demonstrate our method is insensitive to T.

Selection of K. As showed in Fig. 8 (c), a wide range of K ($K < 30$) can guarantee a stable and high performance of our approach. While a big K ($K > 30$) can de-



(a) Base Methods (b) T (c) K (d) δ_θ

Figure 8. Performance analysis of the learning-to-combination strategy: (a) the improvements achieved by using different base methods to propose candidate categories; (b) the accuracies when working with different T number of candidate categories; (c) the accuracies when keeping different K reference objects for each candidate category; and (d) performance changes with different stride δ_θ.

cease the performance a little, since some dissimilar examples are included for each category, and they can interfere with the similarity measure of the corresponding category using the voting (7). When K exceeds a threshold 50, those last added examples have very small similarity scores and can be negligible for the similarity measure of each category, resulting in a stable result finally.

Selection of δ_θ. Fig. 8 (d) demonstrates a small δ_θ (*i.e.*, more dense matchers) can lead more powerful similarity measure. The main reason is that the added dense matchers can provide more cues between the pairwise objects and contribute to the combined similarity measure. However, a too small stride, *e.g.*, $\delta_\theta < 0.1$, can improve the performance a little, but at the price of increased computational complexity. With a trade-off of the accuracy and efficiency, we assign $\delta_\theta = 0.1$ in our experiments.

6. Conclusion and Future Work

This paper proposes a new similarity measure based on dense matching, which can significantly improves the top-1 accuracy of RGB-D object recognition. Our similarity measure has two advantages: (1) through dense matching, the two objects in comparison can be transformed and aligned with each other, of which the similarity measure is more meaningful; (2) with a learning-to-combination strategy, the measure can explore a dynamic fusion way to combine RGB and depth cues effectively, which can offers surprisingly good generalization to apply the proposed measure for object recognition. Experiments on two public RGB-D object datasets demonstrate the effectiveness of our method.

In the future, beyond RGB-D object recognition, we will study the more difficult RGB-D object detection by applying the proposed similarity measure. An alternative way is to exploit the framework of exemplar-svm [27] while using our measure to evaluate the similarity between the pairwise objects. The main drawback is that computing the dense matching between the testing bounding boxes and the training exemplars is highly time consuming since a testing image can have thousands of bounding boxes with sliding windows. To speed up our algorithm, we will focus on more efficient algorithm to optimize the dense matching function, as well as benefiting from the hardware such as GPU.

Acknowledgment

This work is funded by the National Basic Research Program of China (Grant No. 2012CB316302), National Natural Science Foundation of China (Grant No. 61322209 and Grant No. 61175007), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06040102).

References

[1] `https://www.google.com/atap/project-tango/`.

[2] `https://www.microsoft.com/microsoft-hololens/en-us/`.

[3] A. Albarelli, E. Rodola, F. Bergamasco, and A. Torsello. A non-cooperative game for 3d object recognition in cluttered scenes. In *3DIMPVT*, 2011.

[4] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. In *ICRA*, 2012.

[5] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, 2011.

[6] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: architecture and fast algorithms. In *NIPS*, 2011.

[7] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. *ISER, June*, 2012.

[8] B. Browatzki, J. Fischer, B. Graf, H. Bulthoff, and C. Wallraven. Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset. In *ICCV Workshops*, 2011.

[9] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[10] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005.

[11] Y. Cheng, R. Cai, X. Zhao, and K. Huang. Convolutional fisher kernels for rgb-d object recognition. In *3DV*, 2015.

[12] Y. Cheng, X. Zhao, K. Huang, and T. Tan. Semi-supervised learning for rgb-d object recognition. In *ICPR*, 2014.

[13] Y. Cheng, X. Zhao, K. Huang, and T. Tan. Semi-supervised learning and feature evaluation for rgb-d object recognition. *Computer Vision and Image Understanding*, 2015.

[14] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[15] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *ICRA*, 2013.

[16] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[18] T. Joachims. Training linear svms in linear time. In *KDD*, 2006.

[19] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5):433–449, 1999.

[20] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011.

[21] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *ICRA*, 2011.

[22] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[23] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

[24] C. Liu, J. Yuen, A. Torralba, and J. Sivic. Sift flow: dense correspondence across different scenes. In *ECCV*, 2008.

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[26] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008.

[27] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[28] J. Quiroga, T. Brox, F. Devernay, and J. Crowley. Dense semi-rigid scene flow estimation from rgbd images. In *ECCV*, 2014.

[29] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[30] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, 2012.

[31] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV*, 2014.