Learning Dense Correspondences via Local and Non-local Feature Fusion

Wen-Chi Chin*, Zih-Jian Jhang*†, Yan-Hao Huang†, Koichi Ito‡ Hwann Tzong Chen*,
* National Tsing Hua University, Taiwan
† Industrial Technology Research Institute, Taiwan
‡ Tohoku University, Japan

***Abstract*—We present a learning-based method for extracting distinctive features on video objects. From the extracted features, we are able to derive dense correspondences between the objects in the current video frame and in the reference template. We train a deep-learning model with non-local blocks to predict dense feature maps for long-range dependencies. A new video object correspondence dataset is introduced for training and for evaluation. Further, we propose a new feature-aggregation technique that is based on the optical flow of consecutive frames and we apply it to the integration of multiple feature maps for alleviating uncertainties. We also use the local information provided by optical flow to evaluate the reliability of feature matching. The experimental results show that our local and non-local fusion approach can reduce unreliable correspondences and thus improve the matching accuracy.**

I. Introduction

As far as robot-arm object manipulation is concerned, many methods focus on 6D object pose estimation to infer the orientation and structure of an object for manipulation [4], [5], [7], [20]. Learning-based approaches have become more popular. For example, Sundermeyer *et al*. [16] use an autoencoder to estimate 3D object orientation. The advantage is that their method does not require real annotated training data and can handle ambiguities of the object's shape. The DeepIM network proposed by Li *et al*. [11] is able to refine the pose via matching the rendered and the observed images. No hand-crafted features are needed, and refinement can be automatically learned. In contrast to previous approaches that focus on estimating 6D object poses, our work aims to estimate directly the dense correspondences between the current video frames and the reference image containing the target object. Dense correspondences are more flexible to derive the grasping points for object manipulation, particularly for non-rigid objects.

The goal of this work is to build a vision-based robot system with an RGB video camera and a robotic arm. We expect the system to effectively localize the specified grasping points of an object from the video input, by matching the learned features to derive dense correspondences between the test video frames and the reference frames. We train a model with non-local blocks to generate feature maps that contain long-range dependencies of local features. We present a new video object correspondence dataset for training and evaluation. The ground-truth correspondences are automatically constructed via solving SLAM using an RGBD camera. Furthermore, we present a new feature-aggregation method to estimate the confidence levels of correspondences using the optical flow from consecutive video frames. The local information provided by the optical flow can be used to reduce false correspondences, and when integrated with the non-local features, can jointly further improve the matching accuracy.

II. Related Work

Early methods for finding correspondences focus mainly on matching sparse local hand-crafted features, such as SIFT [14] and HOG [1]. In contrast, recent methods rely more on learning the features through deep neural networks to characterize the high-level information [9], [18]. Long *et al*. [13] propose to estimate the semantic correspondences between different images with CNNs. The idea is similar to SIFT Flow [12], except they learn the features using CNN models trained on ImageNet. It is shown that CNN features are more representative than those hand-crafted features. Besides, Sundermeyer *et al*. [16] develop an augmented autoencoder to estimate 3D orientation. They propose an implicit representation of object orientation that is characterized by the samples in latent space. Its advantage is that no real annotated training data are required, and it can inherently deal with symmetries of objects. Li *et al*. propose the DeepIM network [11], which can refine the pose via matching the rendered and the observed images. Their method does not need hand-crafted features and can automatically learn to perform refinement. Tekin *et al*. [17] introduce a single-shot approach to the prediction of an object's 6D pose from an RGB image.

Some recent approaches [2], [15] contribute to learning dense feature descriptors for specific object instances through self-supervised training from RGBD images. Schmidt *et al*. [15] present a new method to learn the visual descriptors for estimating dense correspondences. They use a 3D generative model to automatically label correspondences in RGBD video data. Florence *et al*. [2] also adopt the notion of self-supervision and propose the Dense Object Nets, with a ResNet architecture to learn consistent dense visual representations of objects from RGBD data for robotic manipulation.

III. Method

During the training phase, we input a sequence of RGB video frames into our model, which combines a ResNet-34, two non-local blocks [19], and an upsampling block. The

objective of training is to minimize the pixelwise contrastive loss with respect to the output dense feature maps. We add the non-local blocks to characterize the feature correlations at different positions in the feature map. More specifically, the non-local blocks capture long-range dependencies by modeling interactions between any two positions, regardless of their spatial distance. During the inference phase, we extract the feature maps from the input frame I_t and its adjacent frames I_{t-1} and I_{t+1}. We further combine the feature maps using a flow-driven feature-aggregation scheme. Our approach also includes a mechanism to avoid wrong correspondences based on the proposed measurement of unreliability. The pipeline of the inference phase illustrated in Fig. 1 comprises two parts: *i)* the feature-aggregation scheme that combines feature maps of adjacent frames using optical flow, and *ii)* the generation of the unreliability map for filtering out unreliable correspondences. Note that, the input of the model during the inference phase is a series of video frames $\{I_t\}$. Feature maps $\{f_t\}$ of these frames are generated using the trained model. These feature maps contain non-local dependencies that characterize local features. FlowNet 2.0 [6] is used to compute the optical flow between consecutive frames of the input video. With the optical flow at hand, we combine consecutive feature maps into one aggregated feature map $\widehat{f}_t$ for frame I_t. The aggregated feature map that integrates local and non-local information could be used to find corresponding points between the test frame I_t and a reference frame I_r simply by associating the closest pairs in the feature space. Since multiple frames are used, the derived correspondences are more accurate than those obtained by a single image. In our method, we could also estimate an unreliability map by comparing the difference between adjacent feature maps based on optical flow and use this unreliability map to predict and filter out bad correspondences. The details of deriving dense feature maps, aggregating consecutive feature maps into one, and filtering out bad correspondence points using the unreliability map are described as follows.

A. Deriving the Dense Feature Map

Our model extracts the feature map of each input video frame based on the architecture of Dense Object Nets [2], which contains a 34-layer ResNet [3] as the backbone, and the output of ResNet is bilinearly upsampled to the original input size. To encode long-range correlations among features, we insert two additional non-local blocks [19] after ResNet-34 to capture higher-level dependencies. At any given position, the non-local blocks evaluate the weighted average of feature similarities to all positions in the input feature maps. The contrastive loss $\mathcal{L}$ of two input frames I_t and I_r in our model is defined by

$$\mathcal{L}(I_t, I_r) = \mathcal{L}_{\text{match}}(I_t, I_r) + \mathcal{L}_{\text{non-match}}(I_t, I_r)\,, \quad (1)$$

$$\mathcal{L}_{\text{match}}(I_t, I_r) = \frac{1}{|P|} \sum_{(\mathbf{u}_t, \mathbf{v}_r) \in P} \|f_t(\mathbf{u}_t) - f_r(\mathbf{v}_r)\|^2\,, \quad (2)$$

$$\mathcal{L}_{\text{non-match}}(I_t, I_r) = \frac{1}{|Q|} \sum_{(\mathbf{u}'_t, \mathbf{v}'_r) \in Q} \max\left\{0, M - \|f_t(\mathbf{u}'_t) - f_r(\mathbf{v}'_r)\|^2\right\}\,, \quad (3)$$

where f_t and f_r are the feature maps of size $R^{W\times H\times D}$ derived from input frames I_t and I_r of size $R^{W\times H\times 3}$. P and Q are match and non-match point pairs found in I_t and I_r, respectively. For each match point pair $(\mathbf{u}_t, \mathbf{v}_r) \in P$, the feature distance between them in feature space should be as small as possible. For each non-match point pair $(\mathbf{u}'_t, \mathbf{v}'_r) \in Q$, the feature distance between them in feature space should be larger than a pre-defined constant M. Therefore, the training data regarding a pair of I_t and I_r consist of two subsets: match point pairs P and non-match point pairs Q.

In order to exploit more global information for finding dense correspondences, non-local operations are employed after ResNet-34. Inspired by Non-local Neural Networks[19], the non-local operation in our model is defined as

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j), \quad (4)$$

where i is the position on the output feature map whose response is to be computed and j is the coordinate of all possible positions in the input feature map. x is the input feature map and y is the output signal of the same size as x. The pairwise function f computes the relationship between i and all j and g is the unary function of the input signal at the position j. The output values are normalized by a factor $C(x)$. The unary function g is computed as

$$g(x_j) = W_g x_j, \quad (5)$$

where W_g is the learned weight matrix. We implement the function g as 1×1 convolution.

As mentioned in [19], there are many different choices for the pairwise function f . We adopt the embedded Gaussian function as our function f and it is defined as follows

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}, \quad (6)$$

where $\theta(x_i) = W_\theta x_i$ and $\phi(x_j) = W_\phi x_j$ are implemented with two-dimensional convolutions. Besides, $C(x)$ is calculated as

$$C(x) = \sum_{\forall j} f(x_i, x_j), \quad (7)$$

for a given position i, $\frac{1}{C(x)} f(x_i, x_j)$ becomes the $softmax$ computation along the dimension j.

The non-local block used in our model is depicted in Fig. 2. Through non-local blocks, our model can capture global dependencies in the feature map.

To collect our training data automatically, we use an RGBD camera to capture videos from different viewing angles and use RTAB-Map [10] to recover relative camera poses between frames. Depth maps are used to find correspondence pairs in 3D. They could be transformed to a unified coordinate system using the pose information, and the ground-truth correspondences between two depth maps could be obtained by

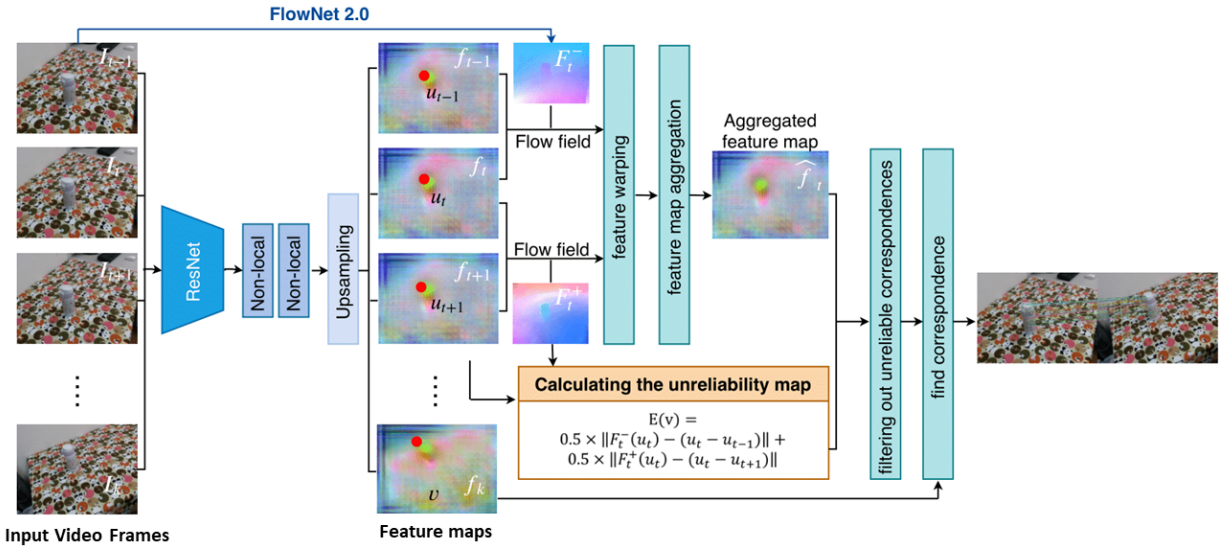


Fig. 1. An overview of our approach. For each input frame I_t, the feature map f_t is generated from a ResNet followed by two non-local blocks [19]. With the non-local operations, feature maps can contain long-range dependencies. Besides, we present a new feature-aggregation method to combine feature maps weighted by the pixelwise confidence levels using optical flow estimated by FlowNet 2.0 [6]. We also compute an unreliability map to avoid false correspondences.

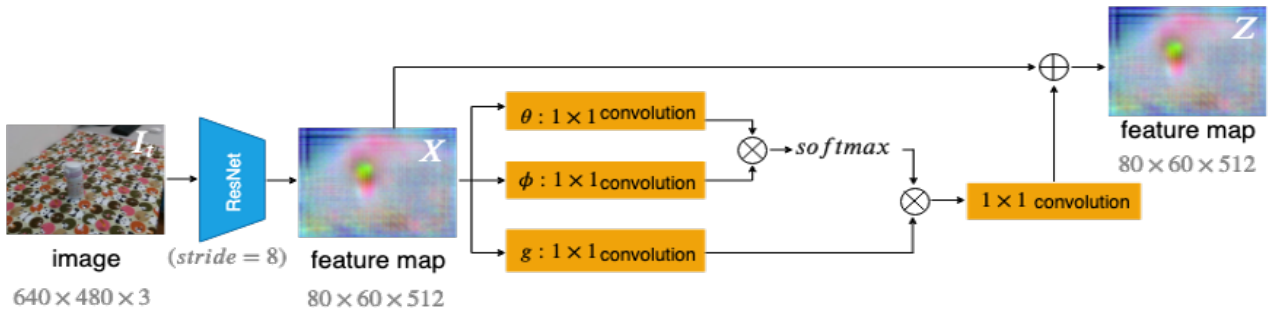


Fig. 2. The overview of **non-local block.** The input feature map X is fed into the block as the size of $80 \times 60 \times 512$ which is the same as the output Z. "$\otimes$" denotes matrix multiplication, and "$\oplus$" denotes element-wise sum and the softmax function is performed.

associating the closest pairs in the unified coordinate system. These correspondences could be transformed back to RGB frames using camera extrinsic parameters. For measuring the matching error of Dense Object Nets [2], object masks are needed in the training process. According to [2], the matching errors of points on the object are more important than the matching errors of points in the background. We use a simple background subtraction technique to generate the object masks for constructing the ground-truth training data. Note that, the RGBD input and the procedure of SLAM are only needed for collecting training and evaluation data. Our method for estimating dense correspondences does not require any depth information or camera parameters.

B. Aggregating Feature Maps across Time

Feature warping. With our trained model that combines ResNet-34 and non-local blocks, we can extract the feature map with non-local information of an input frame at the inference phase. To aggregate feature maps of adjacent frames I_{t-1}, I_t, I_{t+1} in the video, we compute the optical flow between adjacent input frames, and warp the feature maps according to the flow. The warping of the feature map is done

by

$$f_t^- = W(f_{t-1}; F_t^-), \tag{8}$$

$$f_t^+ = W(f_{t+1}; F_t^+), \tag{9}$$

where $W(\cdot; F)$ performs bilinear warping with respect to the flow F, and f_t^- and f_t^+ are the feature maps warped from frame I_{t-1} to frame I_t and from frame I_{t+1} to frame I_t, respectively. Similarly, we define $F_t^- = F(I_{t-1}, I_t)$ and $F_t^+ = F(I_{t+1}, I_t)$ as the forward flow from I_{t-1} to I_t and the backward flow from I_{t+1} to I_t.

Feature map aggregation. After feature warping, we get multiple feature maps aligned to the same time step t. To combine those feature maps into one, we calculate a confidence map ω of the same size as each feature map and use it as the pixelwise weighting for integrating the feature maps. The following two equations compute the confidence level of a point $\mathbf{u}$ in the confidence map:

$$\omega_t^-(\mathbf{u}) = \exp(-\lambda \| I_t(\mathbf{u}) - I_t^-(\mathbf{u}) \|), \tag{10}$$

$$\omega_t^+(\mathbf{u}) = \exp(-\lambda \| I_t(\mathbf{u}) - I_t^+(\mathbf{u}) \|), \tag{11}$$

where $\lambda = 0.5$, and I_t^- and I_t^+ denote the warped images of frames I_{t-1} and I_{t+1} to time t. The confidence value is within 0 and 1, depending on the quality of the flow estimation at each pixel. A smaller difference between I_t and the warped image yields a higher confidence value. It implies that the flow estimation is more reliable at that point and thus we can assign a larger weight on the warped feature map. Two semi-aggregated feature maps at time t are then obtained as

$$\widehat{f_t^-} = \omega_t^- \odot f_t^- + (1 - \omega_t^-) \odot f_t \,, \tag{12}$$

$$\widehat{f_t^+} = \omega_t^+ \odot f_t^+ + (1 - \omega_t^+) \odot f_t \,, \tag{13}$$

where $\widehat{f_t^-}$ denotes the semi-aggregated feature map from the neighboring frames at $t-1$ and t, and $\widehat{f_t^+}$ is defined likewise. The operator $\odot$ means element-wise product between two maps. Finally, we calculate the average of $\widehat{f_t^-}$ and $\widehat{f_t^+}$, and get the final aggregated feature map $\widehat{f_t}$. The aggregated feature map can be used to find the dense correspondences of objects. Some examples of confidence maps are shown in Fig. 3.

Fig. 3. We compute the confidence maps as the weights for combining adjacent feature maps. Top: input video frames. Bottom: confidence maps that are derived from the original frames and the warped frames. Brighter pixels mean higher confidence levels.

C. Filtering out Unreliable Correspondences

The corresponding point $\mathbf{u}$ in frame I_t of a reference point $\mathbf{v}$ in frame I_r is defined as the closest point in the feature space. However, not all points are distinctive and reliable for feature matching. We present a new mechanism to measure the reliability of feature correspondences also based on the flow information.

Approximating the matching reliability. When the ground-truth correspondences are known, we can measure the matching error of correspondences. For any image pair with known relative poses, the ground truth of matching points can be found by transforming 3D points into a unified coordinate system using pose information, and the matching error can be computed as the coordinate difference. On the other hand, when the ground-truth correspondences are unknown, as is the case during testing, we propose to take into account the neighboring frames $\{I_{t-1}, I_t, I_{t+1}\}$ and use optical flow as side information to predict the matching error. Given a point $\mathbf{u}_t$ in frame I_t, we could get the corresponding point $\mathbf{u}_{t-1}$ in frame I_{t-1} by finding the closest point in the feature space according to the dense correspondences between I_t and I_{t-1}. The corresponding point $\mathbf{u}_{t+1}$ in frame I_{t+1} could be identified in a similar way. The displacement between $\mathbf{u}_{t-1}$ and $\mathbf{u}_t$ may be viewed as a special type of flow inferred from the dense feature maps. Under the assumption that the optical flow predicted by FlowNet 2.0 is locally more reliable for consecutive frames, if the displacement computed from the feature correspondence is inconsistent with the optical flow, it may imply that the correspondence found on feature maps is probably unreliable. As a result, we measure the unreliability E of dense feature maps based on the difference between these two sources of displacement/flow estimation

$$E(\mathbf{u}_t) = 0.5 \| F_t^-(\mathbf{u}_t) - (\mathbf{u}_t - \mathbf{u}_{t-1}) \| + 0.5 \| F_t^+(\mathbf{u}_t) - (\mathbf{u}_t - \mathbf{u}_{t+1}) \| \,, \tag{14}$$

where $F_t^-(\mathbf{u}_t)$ is the motion vector of the forward flow field $F_t^- = F(I_{t-1}, I_t)$ at point $\mathbf{u}_t$. If $E(\mathbf{u}_t)$ is large, we should avoid using that point. To determine the threshold, we use the median of E as a threshold estimated at the training phase. During the inference phase, we could filter out those match points whose unreliability value is higher than the threshold. In Dense Object Nets [2], the matching accuracy could be improved if we constrain the matching points to be inside the object mask. With our filtering mechanism, we can improve the matching accuracy without additional object masks.

IV. Experiments

A. Data Collection

This work introduces a new object correspondence dataset. We use both an RGBD camera and an RGB camera to capture videos from different viewing angles. As shown in Fig. 4, we collect a variety of objects, including book, bottle, cup, earphones, plush toy, slipper, and stapler. We use the RGBD camera to collect data for training and evaluation. Besides, we also use the RGB camera to collect some test videos. For the RGBD camera, each object has three original

videos taken in the same environment. The length of each video is about ten seconds. Two of the videos of each object will be used for evaluation. To increase the diversity of our training dataset, we augment the data by separating the foreground object and replacing the background. In addition to the original background, each object is augmented with 15 different synthetic scenes. As a result, we collect 16 training videos for each object with 16 different background scenes. We find that the additional training videos with synthetic backgrounds are helpful for improving the distinctiveness of the learned features. As mentioned earlier, the other two original videos are used as the evaluation data. We use RTAB-Map [10] to find relative camera poses between frames and use the accompanying depth maps to obtain ground-truth 3D dense correspondences for training. RTAB-Map estimates the transformation between frames via solving SLAM. With the estimated camera poses, we can transform the depth maps of different views into the same coordinate system, and find the dense correspondences for the observable areas in the video frames. On the other hand, we also use RGB camera to capture some test videos. We collect 10 videos for each object with 5 different scenes. So we use the RGB camera to collect 70 videos whose length is about twenty seconds as test data. With these test data, we can more easily present visual results.

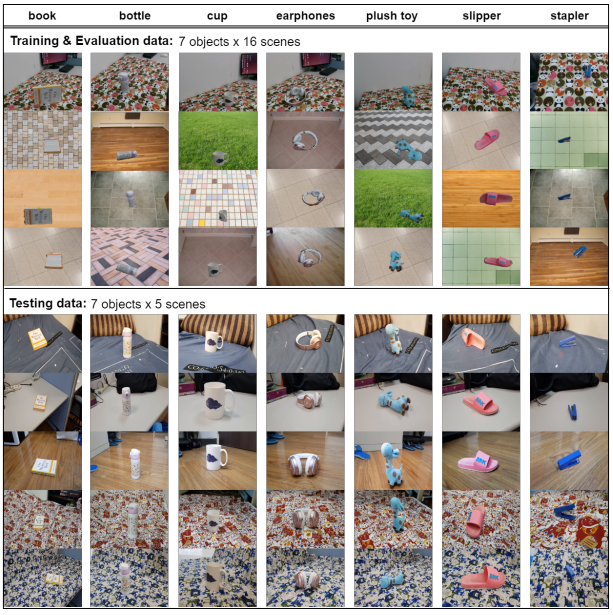


Fig. 4. A new object correspondence dataset: We use both an RGBD camera and an RGB camera to collect videos. The dataset contains seven objects: book, bottle, cup, earphones, plush toy, slipper, and stapler. We augment the data via synthesis with sixteen different background scenes for training data and evaluation data. Besides, we also use the RGB camera to collect seventy videos with five different scenes.

B. Training

The experiments are performed on an NVIDIA GTX Titan X GPU. The network architecture is implemented in PyTorch. We fine-tune a ResNet-34 model with stride-8, which is pretrained on ImageNet, using ADAM optimizer [8] with the learning rate of 10^{-4} for 5000 epochs. In our network, we insert two non-local blocks [19] after ResNet-34 to learn the correlations between features. At each epoch, two frames, not necessarily belonging to the same video, are randomly chosen from the 16 training videos to form a pair. For each pair, there are 10^4 match points at most, and the number of non-match points is 150 times the number of match points. It takes about two hours to train the model.

C. Experimental Results

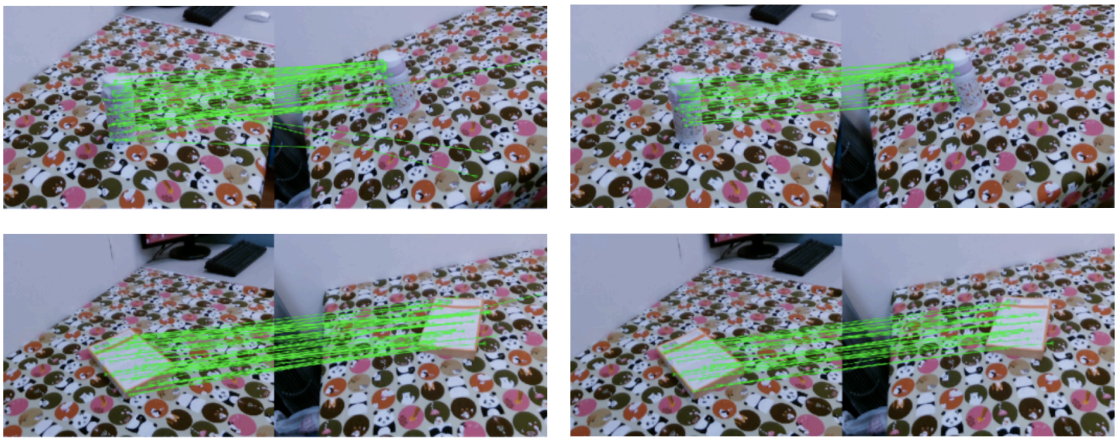
We extract the initial feature maps from the test video frames using our model. We use the proposed feature-aggregation scheme to obtain the aggregated feature maps $\hat{f}$ based on optical flow. We also use the flow information to compute the unreliability map E and to filter out bad correspondences. Fig. 5 shows some qualitative results on our evaluation data. Our method is compared with the state-of-the-art method, Dense Object Nets [2]. As shown in Fig. 5(a), when the background is complicated, some points on the object are matched to the background when using Dense Object Nets. In comparison, our method can generate more representative feature maps through non-local blocks and feature-aggregation mechanism. Besides, our method can filter out bad matching points based on the measurement of unreliability, as shown in Fig. 5(b). Our method can effectively avoid the false matching points. More importantly, our method does not need to rely on any predefined object masks.

Fig. 6 shows more qualitative results between source and target images on our test dataset, which are collected by the RGB camera. We randomly select two different frames as the source image and the target image in the twenty-second videos. Then we find the corresponding points through our method. We show the top 50 matches chosen according to the unreliability value E. We can see that since our training dataset contains many different scenes, the model can make the learned features on the object more representative. In addition, we can learn both local and non-local features through our approach. Because of the above advantages, we can achieve good results without any predefined masks.

To measure the correspondence performance, we compute the pixel matching error, the false positive rate, and the 3D matching error. We compare our learned features with the state-of-the-art Dense Object Nets [2], the SIFT [14] descriptor, and the SURF descriptor on the task of finding matching points between the reference frame and the target frame on the evaluation data of our dataset.

Fig. 7 depicts the cumulative distribution function of the pixel matching error for the state-of-the-art Dense Object Nets, SIFT, SURF, and our method. Our method (the green line) performs better than the Dense Object Nets and improves the matching accuracy without the pedefined object masks on all object categories.

Besides, Fig. 8 shows a quantitative comparison on the false positive rate. We compute the number of pixels in the target



(a) Dense Object Nets [2] (b) Our method

Fig. 5. Qualitative results on our dataset. (a) and (b) indicate the correspondences found by Dense Object Nets and our method, respectively. Our method can effectively filter out bad matching points and get better correspondences.

frame that are closer than the ground-truth corresponding points of the reference frame in feature space. The low false positive rate means there are fewer wrong matching candidates. In all object categories, our method (the green line) is better than other methods in terms of the false positive rate. This means that our method can enhance the distinctiveness of feature maps and successfully filter out unreliable points.

As reported in Table. I, we also evaluate the correspondence performance by measuring the average 3D matching error. To calculate the error between predicted corresponding points and the ground truth, we project the points into the 3D world coordinate system with respect to the depth map and the camera parameters. The error is measured in centimeter. The numbers in boldface in the table indicate the best performance. Our method does not need any predefined object masks and can even perform better than the masked version of Dense Object Nets. We have tried two different training settings, **Our-nonlocal2-flow** and **Our-flow**. The first one is to insert two non-local blocks after ResNet-34 in the training phase, and the other is not. Both of them include the feature-aggregation mechanism and filter out unreliable points using flow information. For most categories, **Our-nonlocal2-flow** performs significantly better than other methods.

D. Ablation Study

We show the ablation analysis with different components in our model. We measure the average 3D matching error in centimeter and report the results on the evaluation videos of our object correspondence dataset. Table. II compares the performance of our model with different components in terms of the average 3D matching error. The baseline models in the first three rows compute the matching error of not using a non-local block but using a combination of applying feature map aggregation and/or filtering out unreliable points by flow information during the inference phase. We can see that applying feature map aggregation and using flow information to filter out some unreliable correspondences to the baseline model both can improve the performance. The results in the next three rows are obtained using a single non-local block. With only one non-local block, the model degrades performance slightly, since it is easily influenced by some cluttered background. The last three rows show that adding two non-local blocks are more useful. The two non-local blocks compute the feature correlations in the feature map more adequately and credibly, increasing the performance of all baseline models significantly. In particular, we can see that using all components performs the best in terms of the average 3D matching error in centimeter.

V. Conclusion

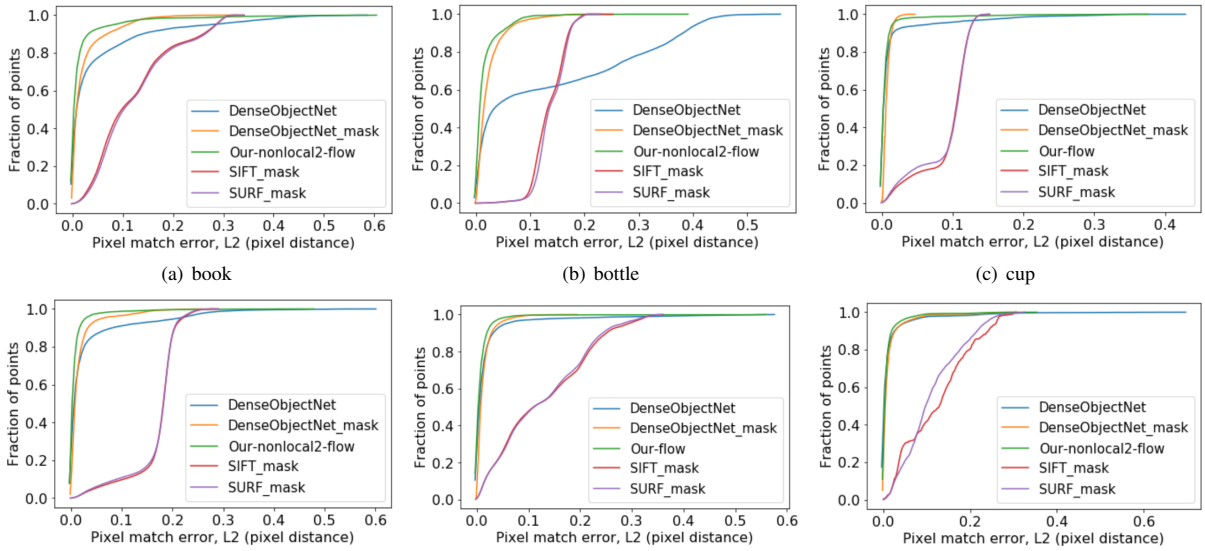
We have presented an effective pipeline for the learning of distinctive features on video objects. Besides, we also introduce a new video object correspondence dataset for training and evaluating feature descriptors. The learned features by our method can be used to find dense correspondences between video frames. On the one hand, our method can characterize non-local correlations of features, and on the other hand, it incorporates the flow information into the process of feature extraction and matching. The proposed feature-aggregation scheme alleviates uncertainties by combining multiple observations. Our method also employs the flow consistency to filter out unreliable correspondences and thus can improve matching accuracy. We conduct qualitative and quantitative experiments using the new video dataset. The results show that our method, which benefits from the fusion of local and non-local information, performs better than previous state-of-the-art methods.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, pages 886–893, 2005.

(a) Source image (b) Target image (c) Top 50 matches

Fig. 6. More qualitative results between source and target images on our dataset. We visualize top 50 matches according to the unreliability value.



(a) book (b) bottle (c) cup

(d) earphones (e) plush toy (f) slipper

Fig. 7. The cumulative distribution function of pixel matching error for *i*) Dense Object Nets, *ii*) Dense Object Nets with object masks, *iii*) Our method, *iv*) SIFT with object masks, and *v*) SURF with object masks.

TABLE I

QUANTITATIVE COMPARISON WITH SIFT-MASK, SURF-MASK, AND DENSE OBJECT NETS, IN TERMS OF THE AVERAGE 3D MATCHING ERROR (CM). WE MEASURE IT BY TRANSFORMING CORRESPONDING POINTS INTO THE WORLD COORDINATE SYSTEM USING THE DEPTH INFORMATION AND THE CAMERA PARAMETERS. THE NUMBERS IN BOLDFACE INDICATE THE BEST PERFORMANCE.

	SIFT-mask	SURF-mask	DenseObjectNets	DenseObjectNets-mask	Our-nonlocal2-flow	Our-flow
book	34.53	34.63	6.15	2.85	**2.45**	2.63
bottle	29.42	30.62	14.25	3.28	**2.86**	6.49
cup	26.32	28.36	2.33	1.46	1.42	**1.23**
earphones	20.09	19.69	3.01	1.81	**1.16**	1.65
plush toy	21.76	21.83	2.13	1.52	1.45	**1.34**
slipper	13.41	12.54	2.04	1.70	**1.17**	1.39
stapler	5.26	5.56	3.55	1.95	**1.46**	1.61

[2] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *2nd Annual Conference on Robot Learning*, pages 373–385, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 770–778, 2016.

[4] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige. Going further with point pair features. In *Computer Vision - ECCV - 14th European Conference*, pages 834–848, 2016.

[5] T. Hodan, P. Haluza, S. Obdrzálek, J. Matas, M. I. A. Lourakis, and X. Zabulis. T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects. In *IEEE Winter Conference on Applications of Computer Vision*, pages 880–888, 2017.

[6] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1647–1655, 2017.

[7] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. In *IEEE International Conference on Computer Vision, ICCV*, pages 1530–1538, 2017.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.

[10] M. Labbé and F. Michaud. Online global loop closure detection for large-scale multi-session graph-based SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666, 2014.

[11] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Computer Vision - ECCV - 15th European Conference*, pages 695–711, 2018.

[12] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.

[13] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 1601–1609, 2014.

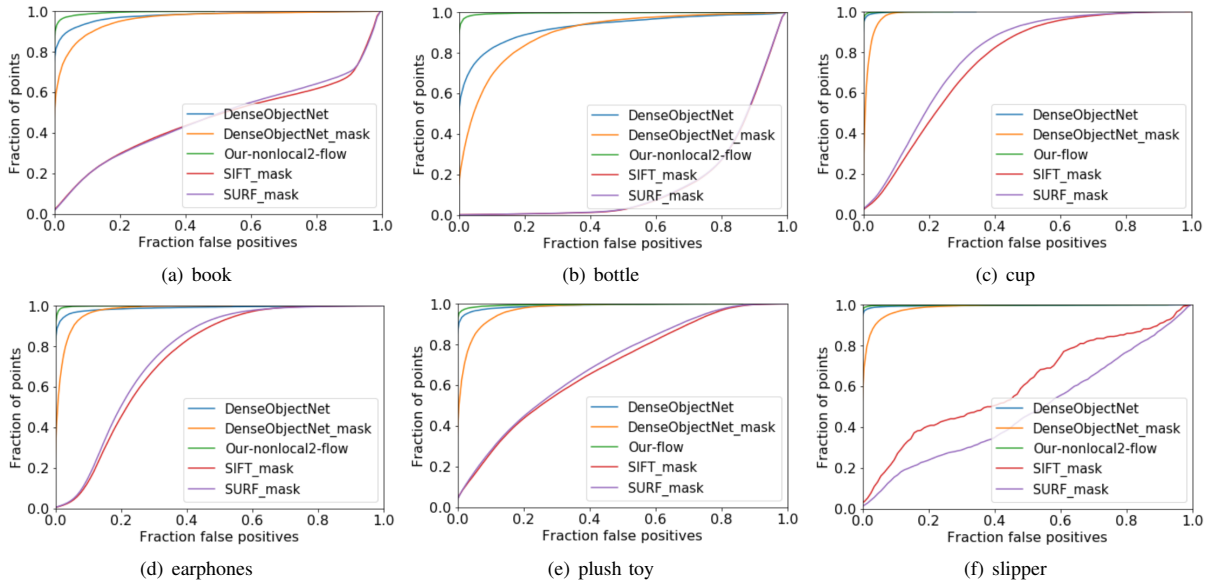
[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] T. Schmidt, R. A. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017.

[16] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from RGB images. In *Computer Vision - ECCV - 15th European Conference*, pages 712–729, 2018.

[17] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 292–301, 2018.

[18] A. Verma, H. Qassim, and D. Feinzimer. Residual squeeze CNDS deep learning CNN model for very large scale places image recognition. In *8th IEEE Annual Ubiquitous Computing, Electronics and Mobile*



(a) book (b) bottle (c) cup

(d) earphones (e) plush toy (f) slipper

Fig. 8. The cumulative distribution function of the false positive rate. We compare our method with Dense Object Nets, SIFT, and SURF on different objects.

TABLE II

AVERAGE 3D MATCHING ERROR COMPARISON OF DIFFERENT COMPONENTS. WE DENOTE BY "1" AND "2" ADDING ONE NON-LOCAL BLOCK AND ADDING TWO NON-LOCAL BLOCKS. NUMBERS IN BOLD INDICATE THE BEST PERFORMANCE AND UNDERLINED ONES ARE THE SECOND BEST.

Non-local block	Feature map aggregation	Filter by flow	3D matching error (cm)						
			book	bottle	cup	earphones	plush toy	slipper	stapler
✗	✗	✓	2.67	6.53	1.38	1.71	1.47	1.45	1.72
✗	✓	✗	2.71	6.62	1.43	1.75	1.51	1.58	1.83
✗	✓	✓	2.63	6.49	**1.23**	1.65	**1.34**	1.39	1.61
1	✗	✓	9.50	3.00	1.59	1.57	1.65	1.27	3.76
1	✓	✗	2.89	2.97	1.46	1.62	1.69	1.32	3.79
1	✓	✓	2.81	2.95	1.44	1.49	1.62	1.20	3.77
2	✗	✓	2.49	2.90	1.44	1.24	1.52	1.19	1.49
2	✓	✗	2.51	2.93	1.47	1.31	1.56	1.23	1.52
2	✓	✓	**2.45**	**2.86**	1.42	**1.16**	1.45	**1.17**	**1.46**

Communication Conference, UEMCON, pages 463–469, 2017.

[19] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 373–385, 2018.

[20] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3109–3118, 2015.