LEARNING DENSE CORRESPONDENCES FOR VIDEO OBJECTS

Wen-Chi Chin[1]*, Zih-Jian Jhang*[2]*, Hwann-Tzong Chen*[1]*, and Koichi Ito*[3]

[1]Department of Computer Science, National Tsing-Hua University, Taiwan.
[2]Industrial Technology Research Institute (ITRI), Taiwan.
[3]Graduate School of Information Sciences, Tohoku University, Japan.

ABSTRACT

We introduce a learning based method for extracting distinctive features on video objects. Based on the extracted features, we are able to derive dense correspondences between the object in the current video frame and the reference template, and then use the correspondences to identify the grasping points on the object. We train a deep-learning model to predict dense feature maps using the training data collected via solving *simultaneous localization and mapping* (SLAM). Further, a new feature-aggregation technique based on the optical flow of consecutive frames is applied to the integration of multiple feature maps for alleviating uncertainties. We also use the optical flow information to assess the reliability of feature matching. The experimental results show that our approach effectively reduces unreliable correspondences and thus improves the matching accuracy.

***Index Terms*—** dense correspondence, visual descriptor, optical flow, feature map aggregation

1. INTRODUCTION

Many methods for robot-arm object manipulation have been focusing on 6D object pose estimation to infer the orientation and structure of an object in an RGB or RGBD image [1, 2, 3, 4]. More recent methods often adopt a learning-based approach. Sundermeyer et al. [5] develop an augmented autoencoder to estimate 3D orientation. They propose an implicit representation of object orientation characterized by samples in a latent space. The representation is advantageous in that no real annotated training data are required and it can inherently deal with symmetries of objects. Li et al. propose the DeepIM network [6], which is able to refine the pose via matching the rendered and the observed images. Their method does not require hand-crafted features and can automatically learn to do refinement. Tekin et al. [7] introduce a single-shot approach to the prediction of an object's 6D pose from an RGB image. Unlike the aforementioned approaches that focus on estimating 6D object poses, our work aims to estimate directly the dense correspondences between current video frames and the reference video frames containing the target object. Dense correspondences are more flexible fro deriving the grasping points, in particular for non-rigid objects.

Some recent approaches [8, 9] contribute to learning dense feature descriptors for specific object instances through self-supervised training from RGBD images. More detailed reviews of related work on visual descriptor learning can be found in [8, 9]. Schmidt et al. [8] present a new method to learn the visual descriptors for dense correspondence estimation. They use a 3D generative model to automatically label correspondences in RGBD video data. Florence et al. [9] also adopt self-supervision and propose the Dense Object Nets, which train on ResNet architectures using RGBD data to learn consistent dense visual representations of objects for robotic manipulation.

The goal of this work is to build a visual-learning robot system with an RGB video camera and a robotic arm. The robot system can automatically learn to localize the specified grasping points of an object from the video input, by matching the learned features to derive the correspondences between the test video frames and the reference video frames. We train a deep model to generate feature maps for predicting the dense correspondences. The training data with ground-truth correspondences are automatically collected by solving SLAM using an RGBD sensor. Furthermore, we present a new feature-aggregation method to estimate the confidence levels of correspondences using optical flow from consecutive video frames. We analyze the difference between the optical flow computed from adjacent frames and the displacement predicted by the correspondence network. The difference could be used to reduce false correspondences and thus could improve the matching accuracy of specified points.

2. METHOD

Fig. 1 illustrates the inference pipeline of our model. The input of the model is a sequence of video frames captured by an RGB camera. A ResNet with residual connections is used to extract a feature map from each input frame. We also use FlowNet2.0 [10] to compute the optical flow from consecutive frames of the input video. Based on the optical flow, our model derives an aggregated feature map from neighboring frames. By combining multiple observations from different

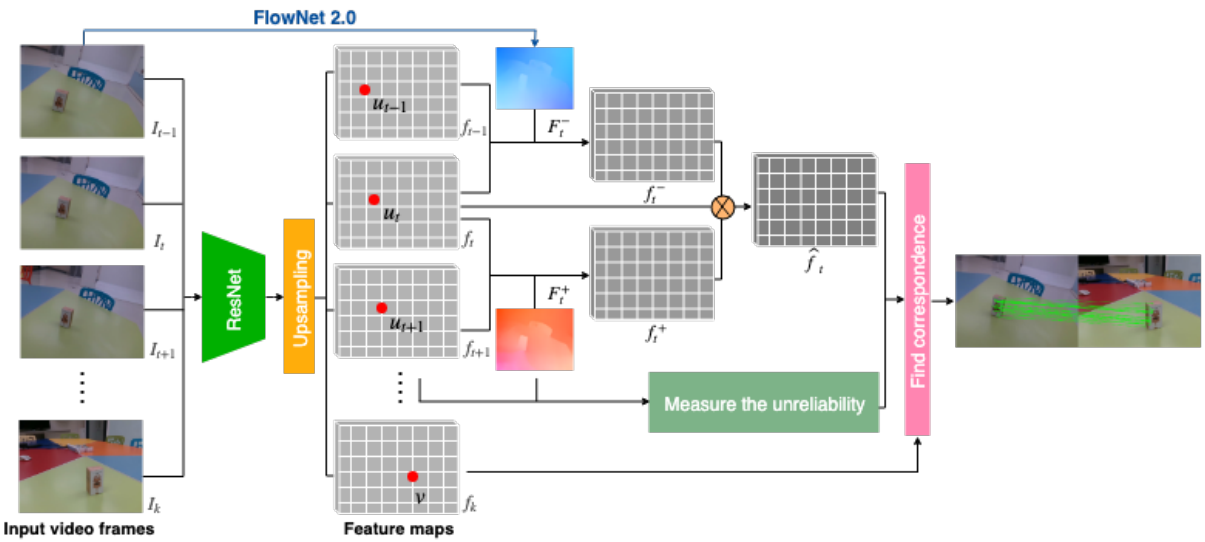


Fig. 1. An overview of our approach. For each input frame, the feature map is generated by Dense Object Nets [9]. We use FlowNet 2.0 [10] to estimate optical flow, and we warp the adjacent feature maps based on the flow. We present a new feature-aggregation method to combine the feature maps weighted by the pixelwise confidence levels. We also compute an unreliability map from the flow information and set a threshold to filter out unreliable correspondences.

frames (different views), we can mitigate the uncertainties in feature extraction. Our model also estimates an unreliability map to filter out unreliable areas. The detailed settings and mechanisms of the components in the pipeline are described as follows.

2.1. Deriving the Dense Feature Map

We use Dense Object Nets [9], which contains a 34-layer ResNet [11] as the backbone, to extract the feature map for each input video frame. We train this model using the pixelwise contrastive loss. It aims to minimize the distance between the matched keypoints under the constraint that the distance between non-matched keypoints should be at least M, where M is a margin parameter. The loss L of two input frames I_a and I_b is defined by

$$L(I_a, I_b) = L_{\text{matched}}(I_a, I_b) + L_{\text{non-matched}}(I_a, I_b), \quad (1)$$

$$L_{\text{matched}}(I_a, I_b) = \frac{1}{|P|} \sum_{(u_a, u_b) \in P} \|f_a(u_a) - f_b(u_b)\|^2, \quad (2)$$

$$L_{\text{non-matched}}(I_a, I_b) = \frac{1}{|Q|} \sum_{(u_a, u_b) \in Q} \max\left\{0, M - \|f_a(u_a) - f_b(u_b)\|^2\right\}, \quad (3)$$

where f_a and f_b are the feature maps of size $R^{W \times H \times D}$ derived from input frames I_a and I_b of size $R^{W \times H \times 3}$. The training dataset regarding the pair I_a and I_b consists of two subsets P and Q, where P contains pairs of matched keypoints and Q contains pairs of non-matched keypoints.

2.2. Aggregating Feature Maps

Feature warping. We can extract the feature map of an input frame from the model trained by Dense Object Nets. We then compute the optical flow between adjacent input frames using FlowNet 2.0 and warp their feature maps according to the flow. The warping of feature maps is done by

$$f_t^- = W(f_{t-1}; F_t^-), \quad (4)$$

$$f_t^+ = W(f_{t+1}; F_t^+), \quad (5)$$

where $W(\cdot; F)$ performs bilinear warping with respect to the flow F, and f_t^- and f_t^+ mean that the feature maps are warped from frame I_{t-1} to frame I_t and from frame I_{t+1} to frame I_t. 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. A mechanism to enhance the features is to aggregate the adjacent frames by their flow information. After feature warping, we get multiple feature maps aligned to the same time step. We then define a confidence map ω of the same size as the input frame, and use it to give pixelwise weights for integrating the feature maps. The confidence level of a point u in the confidence map is computed by one of the following two equations:

$$\omega_t^-(u) = \exp(-\lambda \|I_t(u) - I_t^-(u)\|), \quad (6)$$

$$\omega_t^+(u) = \exp(-\lambda \|I_t(u) - I_t^+(u)\|), \quad (7)$$

where $\lambda = 0.5$ is used to adjust the speed of convergence, and I_t^- and I_t^+ denote the warped results of frames I_{t-1} and I_{t+1}

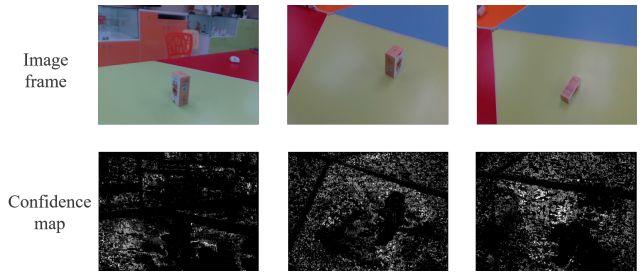


Fig. 2. We compute the confidence maps as the weights to combine adjacent feature maps. Brighter pixels mean higher confidence levels.

to time t. The confidence value is between 0 and 1, depending on the correctness of flow estimation. A higher confidence value that is closer to 1 means that the point is more reliable. The aggregated feature map is beneficial for resolving uncertainties since we combine multiple observations. Some examples of confidence maps are shown in Fig. 2.

The semi-aggregated feature maps at time t are

$$\widehat{f}_t^- = \omega_t^- \odot f_t^- + (1 - \omega_t^-) \odot f_t \,, \tag{8}$$

$$\widehat{f}_t^+ = \omega_t^+ \odot f_t^+ + (1 - \omega_t^+) \odot f_t \,, \tag{9}$$

where f_t^- denotes the feature map aggregated from the neighboring frames at $t-1$ and t, and f_t^+ is defined likewise, as in (4) and (5). The operator $\odot$ means element-wise product between two maps. Finally, we calculate the average of the two semi-aggregated feature maps $\widehat{f}_t^-$ and $\widehat{f}_t^+$, and get the final aggregated feature map $\widehat{f}_t$. The aggregated feature map contains flow information and is more representative.

2.3. Filtering out Unreliable Correspondences

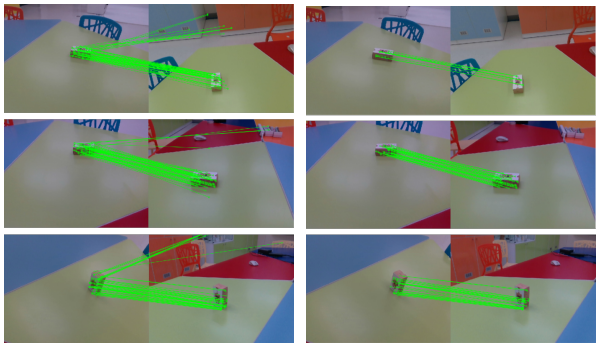
We can find the corresponding point $u \in I_t$ of a reference point $v \in I_r$ as the closest point in the feature space. However, not all points are distinctive and reliable for feature matching. We present a new mechanism to measure the reliability of feature correspondences using the flow information.

Approximating the matching reliability. To measure the matching reliability, we calculate the matching error using ground-truth correspondences. For any image pair with known relative poses, the ground truth of matched 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.

During inference we do not have no ground-truth correspondences, so we propose to use optical flow as side information to approximate the matching error. Suppose that u_{t-1}, u_t, and u_{t+1} in three adjacent frames I_{t-1}, I_t, and I_{t+1} are the corresponding points of v in a reference frame according to the feature maps extracted by the trained model. We could use these points to compute the displacements $u_t - u_{t-1}$ and $u_{t+1} - u_t$. Besides, we also obtain the flow motion F estimated by FlowNet 2.0. Then we can measure the unreliability E based on the difference between these two sources of displacements of adjacent frames

$$E(v) = (\|F_t^-(u_t) - (u_t - u_{t-1})\| + \|F_t^+(u_t) - (u_t - u_{t+1})\|)/2 \,, \tag{10}$$

where $F_t^-(u_t)$ means the motion vector from the forward flow field $F_t^- = F(I_{t-1}, I_t)$ at point u_t. If the displacement of the corresponding points found by our trained model is inconsistent with the displacement estimated by the flow, it implies that the correspondence at that point is probably unreliable and we should avoid using it. We use the median of E measured in training phase to get the threshold. During inference, we can filter out unreliable points whose unreliability E is larger than the threshold. With this filtering mechanism, we are able to increase the matching accuracy without using predefined object masks.



(a) Dense Object Nets [9] (b) Our method

Fig. 3. Some qualitative results. Our method can get better correspondences than Dense Object Nets.

3. EXPERIMENTS

3.1. Data Collection

We use an RGBD sensor to capture images from different viewing angles and run RTAB-Map [12] with RGB images and the accompanied depth maps to obtain ground-truth 3D dense correspondences for training. RTAB-Map estimates the transformation between frames via SLAM. Depth images can be transformed to a unified coordinate system using the pose information, and the ground-truth correspondences between two images are obtained under the unified coordinate. We nee object masks to measure the matching error of Dense Object Nets [9]. 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. We collect, in total, eight videos for training and two videos for evaluation. The length of each video is forty seconds.

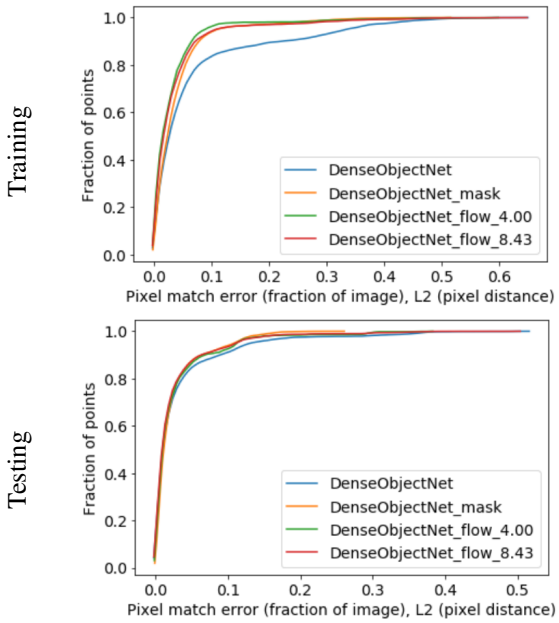


Fig. 4. The cumulative distribution function of pixel matching error for Dense Object Net, Dense Object Net with the object mask, and our method under different reliability thresholds.

3.2. Training

Our experiments are performed on an NVIDIA GTX Titan X GPU. The network architecture is implemented in PyTorch. We fine-tune a ResNet model, which is pretrained on ImageNet, using ADAM optimizer [13] with a learning rate of 10^{-4} for 5000 epochs. The image pair is randomly chosen from eight training videos at each epoch. It takes about two hours to train the model.

3.3. Experimental Results

We use a model of Dense Object Nets trained on our data to extract the initial feature maps from the test videos. Then, we apply the proposed feature-aggregation mechanism to enrich the feature maps with flow information and obtain the aggregated feature maps $\widehat{f}$. We also filter out some bad matching points using the unreliability map E. Some qualitative results are shown in Fig. 3. The matching points found by Dense object Nets [9] are shown in Fig. 3(a) and it can be seen that some object points are matched to the background even with the predefined mask in the current video frame. In comparison, after filtering out some matching points with an unreliability value larger than the threshold, which is the median unreliability value measured during the training phase, we can find that the results shown in Fig. 3(b) contain fewer false matching points. Furthermore, Fig. 4 depicts the cumulative distribution function of pixel matching error for the original Dense Object Nets and our method. The reliability threshold of our method (the median of E measured during the training phase) is 8.43. Besides, we also set another reliability threshold to 4.00 manually for comparison. As can be seen, whether we use the measured reliability threshold (8.43) or the manual setting threshold (4.00), our method can effectively filter out unreliable points. In particular, the reliability threshold derived from the median of E works better. Our method outperforms the original Dense Object Nets and improves the matching accuracy without the need of any predefined object masks. We also compute the number of pixels in the target frame that are closer than the ground-truth correspondence of reference frame in feature space. Fig. 5 shows that our method is effective in filtering out false matching points.

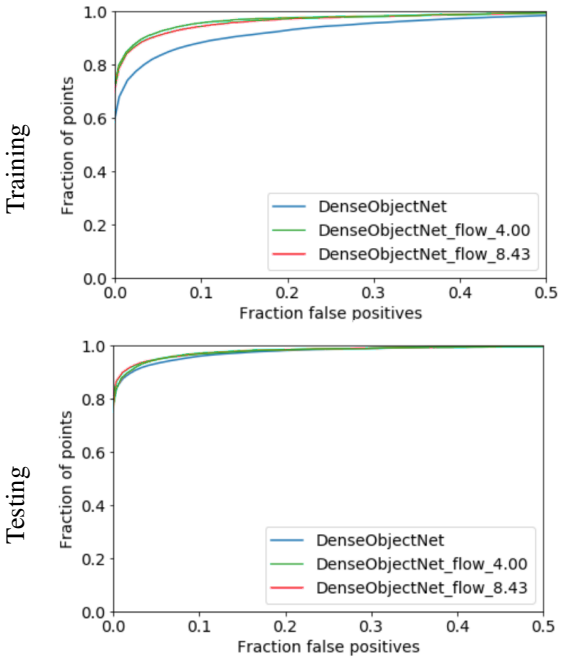


Fig. 5. The cumulative distribution function of false positive rate for Dense Object Net and our method with different reliability thresholds.

4. CONCLUSION

We present a self-supervised pipeline for learning representative features on video objects. The learned features can be used to find dense correspondences between video frames of different viewing angles. Our method incorporates the flow information into the process of feature extraction and matching. The proposed feature-aggregation technique reduces uncertainties by combining multiple observations. Our method also employs the flow consistency to filter out unreliable correspondences and thus improves matching accuracy.

Acknowledgement. This work is supported in part by MOST grant 106-2221-E-007-080-MY3 and BioProA+ program.

5. REFERENCES

[1] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige, "Going further with point pair features," in *Computer Vision - ECCV - 14th European Conference*, 2016, pp. 834–848.

[2] Tomas Hodan, Pavel Haluza, Stepán Obdrzálek, Jiri Matas, Manolis I. A. Lourakis, and Xenophon Zabulis, "T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects," in *IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 880–888.

[3] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab, "SSD-6D: making rgb-based 3d detection and 6d pose estimation great again," in *IEEE International Conference on Computer Vision, ICCV*, 2017, pp. 1530–1538.

[4] Paul Wohlhart and Vincent Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015, pp. 3109–3118.

[5] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel, "Implicit 3d orientation learning for 6d object detection from RGB images," in *Computer Vision - ECCV - 15th European Conference*, 2018, pp. 712–729.

[6] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Computer Vision - ECCV - 15th European Conference*, 2018, pp. 695–711.

[7] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua, "Real-time seamless single shot 6d object pose prediction," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 292–301.

[8] Tanner Schmidt, Richard A. Newcombe, and Dieter Fox, "Self-supervised visual descriptor learning for dense correspondence," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 420–427, 2017.

[9] Peter R. Florence, Lucas Manuelli, and Russ Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *2nd Annual Conference on Robot Learning*, 2018, pp. 373–385.

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 1647–1655.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, 2016, pp. 770–778.

[12] Mathieu Labbé and François Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2661–2666.

[13] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2014.