One Shot Learning for Generic Instance Segmentation in RGBD Videos

Xiao Lin[1 2], Josep R. Casas[2] and Montse Pardàs[2]
[1]*Industry and Advanced Manufacturing Department, Vicomtech, San Sebastian, Spain*
[2]*Image Processing Group (GPI), Universitat Politécnica de Catalunya, Barcelona, Spain*
xlin@vicomtech.org, {josep.ramon.casas, montse.pardas}@upc.edu

Keywords: Instance segmentation, One shot learning, Convolutional neural network.

Abstract: Hand-crafted features employed in classical generic instance segmentation methods have limited discriminative power to distinguish different objects in the scene, while Convolutional Neural Networks (CNNs) based semantic segmentation is restricted to predefined semantics and not aware of object instances. In this paper, we combine the advantages of the two methodologies and apply the combined approach to solve a generic instance segmentation problem in RGBD video sequences. In practice, a classical generic instance segmentation method is employed to initially detect object instances and build temporal correspondences, whereas instance models are trained based on the few detected instance samples via CNNs to generate robust features for instance segmentation. We exploit the idea of one shot learning to deal with the small training sample size problem when training CNNs. Experiment results illustrate the promising performance of the proposed approach.

1 INTRODUCTION

The performance of classical generic instance segmentation methods, such as (Lin et al., 2018), is usually restricted to the discriminative power of the employed hand-crafted features. Those features are not representative enough to describe and distinguish different object instances when segmenting interacting object instances in generic scenes. On the other hand, Convolutional Neural Networks (CNNs) based semantic segmentation methods introduce a good representation for the predefined semantics, which are trained to extract robust features via networks with a huge number of parameters. Although the success of applying CNNs to semantic segmentation proves the strong representation capability of CNNs can be exploited on dense prediction tasks, it also shows some drawbacks. One of the major downsides of CNNs based approaches is their hunger for training data. In semantic segmentation, training data is prepared as manually labeled segmentation masks, in which labels in the mask represent different semantics. Preparing the training data for semantic segmentation requires large efforts on manual labeling due to the big necessity of training data. Besides, the idea of semantic segmentation restricts to certain types of predefined semantics, which compromises its application to more generic scenes. From the perspective of generic segmentation, training data can hardly be prepared, since no semantics are predefined.

In video instance segmentation, methods proposed to detect/segment generic object instances, such as (Endres and Hoiem, 2010) and (Lee et al., 2011), are usually employed as an object proposal generator. An offline temporal analysis is exploited, in order to search from a pool of object proposals within a frame along a video sequence, which, in consequence, restricts them to offline applications. On the other hand, model based generic instance segmentation methods, such as (Husain et al., 2015; Koo et al., 2014), usually employ online training techniques, where instance models are trained and updated along a video sequence. These approaches introduce a way to train instance models without predefined semantics. However, the models used in these approaches are usually simple, such as Gaussian models used in (Koo et al., 2014) and quadratic functions in (Husain et al., 2015), due to the small size of the training data.

In this paper, we present a generic instance segmentation method which combines the advantages of the generic instance segmentation method introduced in (Lin et al., 2018) and those of CNNs based semantic segmentation. That is the genericity in the generic instance segmentation method and the strong object representation power in CNNs, by exploiting the idea of one shot learning. We employ the classical generic

instance segmentation method to discover object instances and build temporal correspondences based on all low level features. To represent the discovered object instances, we first train a CNN model offline for tracking generic object instances. Based on it, we fine-tune the tracking model online with the few examples of the discovered object instances, in order to obtain one CNN for each object instance to extract robust features. In that case, we can predict more accurately if a pixel belongs to the instance or not, based on the features extracted from CNNs rather than hand-crafted features used in (Lin et al., 2018). On the other hand, the genericity is also kept, since no prior information, such as initialization or predefined semantics, is introduced in the proposed approach. Furthermore, in the experiments section we also evaluate the results obtained using the generic tracking CNN model trained offline, without object specific online fine-tuning. We observe that even these generic features outperform the hand-crafted ones, with a similar run-time performance.

2 RELATED WORK

The most challenging part of the proposed approach is how to train the CNNs based system with very limited annotations. The deep architecture of CNNs provides a complex function with a large amount of parameters so that useful representations of high dimensional data can be learned. However, this advantage of CNNs becomes an obstacle in the training process when only few annotation is provided. In this case, the learned model is strongly over-fitted due to the large number of parameters and limited training data. To tackle the problem, we employ the idea of one shot learning. The key insight of one shot learning is that, rather than learning from scratch, one can take advantage of knowledge coming from a previously learned model and solve the new learning tasks using only one or few training samples.

One shot learning is an extreme case of transfer learning. Transfer learning is widely used for training CNNs in various tasks. For instance, (Chen et al., 2016) trains a semantic segmentation network first on a image classification purpose using the large scale dataset ImageNet (Deng et al., 2009) as the training data. Then, they take this pre-trained model as an initialization for a further training with a smaller set of training data for the semantic segmentation task. In (Girshick et al., 2014), the authors also pre-train their object detection network with ImageNet on an image classification purpose.

One shot learning methods have also been developed for various tasks in the state of the art, such as image recognition (Vinyals et al., 2016; Fei-Fei et al., 2006) and gesture recognition (Konecnỳ and Hagara, 2014). More related to our approach, there are also one shot learning based approaches for video object segmentation. In (Caelles et al., 2016), the authors present one shot object segmentation on video sequences, based on a fully-convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object and segment the object in the following frames with the learned object model in the test sequence. Similarly, MaskTrack (Khoreva et al., 2016) learns to refine the detected mask of an object, by using the detections of the previous frame. The authors first synthesize the movement of an object mask between consecutive frames by performing affine transformation and non-rigid deformation to ground truth object masks in group of datasets. In this manner, the mask refinement network is generally trained off-line for generic objects in the group of datasets. Then, they fine-tune the network online for a specific object in a test sequence using only the ground truth mask provided in the first frame. One of the drawbacks of these approaches is that they require an accurate initialization for performing one shot learning on an object instance in the scene.

3 CLASSICAL GENERIC INSTANCE SEGMENTATION

In (Lin et al., 2018), the authors have introduced a classical generic instance segmentation method F, which calculates the current segmentation O_t in frame t with point cloud Ct obtained from the current RGBD frame and the previous segmentation O_{t-1}, $F(C_t, O_{t-1}) \rightarrow O_t$. O_t consists of different object instances $o_t^1, o_t^2 ... o_t^{M_o} \in O_t$, where M_o denotes the number of objects in the scene. Since the temporal correspondences between object instances are made in F, we have the observed sequence of object instances in the history for each object instance $o^i_{1...t-1}$ before the segmentation in frame t is obtained. To segment the current frame, the point cloud C_t is first divided into blobs $b_t^1, b_t^2 ... b_t^{M_b}$ by analyzing the point cloud connectivity built on a super-voxel graph $G^t(v,e)$, in which v represents super-voxels set and e represents the edge set of the adjacency of super-voxels. The current blobs are then assigned to object labels from the previous frame via an optimization process. Blobs

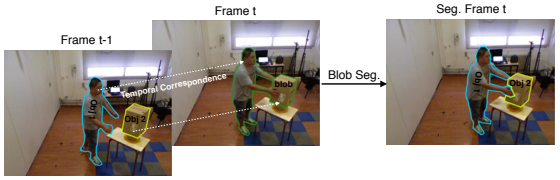

Figure 1: An example of blob segmentation in frame t considering the temporally corresponded object instances in frame $t-1$.

assigned to more than one object labels need Blob Segmentation. Fig.1 shows an example of a blob segmentation, in which a blob (the region with green boundary) in frame t is segmented with respect to the object instances detected in frame $t-1$ (the region with blue and yellow boundary) and the temporal correspondence built between these two frames.

The segmentation for the first frame is simply done by first removing the plane-like point sets in the input point cloud, then searching for connected components on the super-voxel graph built on the residual point cloud. In this manner, isolated point sets are extracted from the input point cloud, which ideally corresponds to object instances in the scene.

In (Lin et al., 2018), blob segmentation is achieved by labeling nodes on the graph of the blob with assigned object labels via a Fully Connected Conditional Random Field (FC-CRF) model. FC-CRF introduces an unary energy describing the degree of confidence that a super-voxel belongs to an object instance and a pairwise energy representing the degree of confidence that two super-voxels belong to the same object instance. Optimizing the energy function with the two energy terms provides the best labeling of the graph, which implicitly represents the segmentation of the blob. The unary energy for each node on the graph is defined based on low level features, such as 3D distance and color. As in (Lin et al., 2016), we define the unary energy for labelling node v_i with object label o_j as the mean distance between node v_i in the current frame and the k-nearest nodes labeled by o_j in the previous frame. This mean distance is computed comparing feature vectors which concatenate 3 components: color feature (color histogram in LAB color space), shape feature (local surface normal) and 3D position (3D coordinates of the node centroid). Details can be found in (Lin et al., 2016). These low level features are not always discriminative enough for well distinguishing/segmenting different object instances in a blob, which produces segmentation errors.

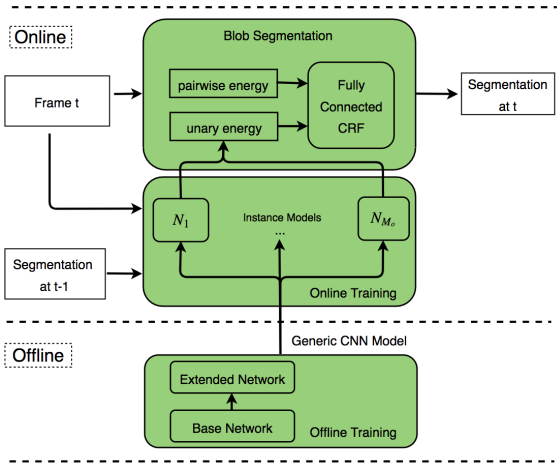

Figure 2: The schema of proposed approach.

4 CNNS BASED UNARY ENERGY LEARNING

To tackle the above mentioned problem, we propose to exploit CNNs to extract robust features for defining the unary energy in the blob segmentation task. In practice, we train one CNN model N_i for each object instance based only on the few observations of that object instance in the history. The CNN N_i extracts feature maps from the input data and outputs a 2 classes probability map via a softmax layer at the end of the CNN N_i. The probability map consists of probabilities that each pixel belongs to instance i or not. For a super-voxel v_i, the probability is computed as the mean probability of pixels in v_i. Then, we simply employ the probabilities of the super-voxels obtained from the CNN models of different object instances as the unary energy. However, training CNNs with millions of parameters from scratch usually requires a large number of annotated data, in order to optimize the parameters for extracting robust representation of the input data. In our case, we only have few object instance observations in the history $o^i_{1\ldots t-1}$ in frame t, which can be employed as training data. With limited number of annotated data, it is difficult to follow the training-from-scratch process. Thus, we follow the method proposed in (Khoreva et al., 2016) to perform one shot learning using the object instance observations in the history.

Given the segmentation of an object o^i_{t-1} in frame $t-1$ and the input color image I_t in frame t, our aim is to train a CNN $N_i\left(I_t, o^i_{t-1}\right) \rightarrow P^t_i$, where P^t_i represents a probability map for object instance o^i at time t. $P^t_i(x,y)$ stands for the output probability that the pixel (x,y) on the input image I_t belongs to object

o_t^i or not ($P(x,y) = \left[P_{o_t^i}(x,y), P_{\bar{o_t^i}}(x,y)\right]$). The CNN model generates the current object instance segmentation by refining the object instance segmentation o_{t-1}^i in frame $t-1$ with respect to the current color image I_t. We formulate the CNN based unary energy in Eq. 1 as:

$$\mu_{v_j}(i) = \frac{1}{M_{v_j}} \sum_{\forall (x,y) \in v_j} P_i(x,y) \quad (1)$$

where M_{v_j} stands for the number of pixel contained in super-voxel v_j. Note that, in Eq. 1, we omit the notation t for conciseness.

We employ two steps to achieve the training process: the offline training and online training step. In the offline training step, a base network is first employed to learn the generic attributes in an image classification task. Then, we extend the base network to learn a generic notion of how to segment an object instance taking a color image and a mask in the previous frame as the input. In the online training step, we specify the extended network to a specific object instance by fine-tuning the the generic model obtained in the previous step, using only the few observations of the object instance in a sequence. Fig.2 shows the schema of the proposed approach.

4.1 Offline Training

A VGG network (Simonyan and Zisserman, 2014) is used as our base network and is pre-trained on ImageNet (Deng et al., 2009) for an image classification task, which has proven to be a very good initialization in other tasks (Chen et al., 2016; Girshick et al., 2014). Although the network is not capable of performing image segmentation, it provides generic attributes in the network, which can be further specified to tackle other tasks.

The network is then extended to cope with the segmentation task. We follow Deeplab-ASPP (Chen et al., 2016), which replaces the fully connected layers in VGG network with atrous upsampling layers to achieve dense classification in a semantic segmentation task. Deeplab-ASPP is selected due to its outstanding performance in semantic segmentation. Then, we extend the network to allow an extra mask channel in the input. The extra mask channel is meant to provide an estimation of the visible area of the object in the current frame, its approximate location and shape. We can then train the extended network to output an accurate segmentation of the object instance, given as input the current image and a rough estimate of the object mask. To simulate the noise of the previous frame output, during offline training, we generate input masks by deforming the annotations using affine transformation as well as non-rigid deformations via thin-plate splines(Bookstein, 1989), followed by a coarsening step (dilation morphological operation) to remove details of the object contour. We apply this data generation procedure over a dataset of 10^4 images containing diverse object instances. The affine transformations and non-rigid deformations aim at modelling the expected motion of an object between two frames. The coarsening permits us to generate training samples that resemble the test time data, simulating the blobby shape of the output mask given from the previous frame by the extended network. These two ingredients make the estimation more robust to noisy segmentation estimates while helping to avoid accumulation of errors from the preceding frames.

4.2 Online Training

The offline training provides the extended network the ability to refine a roughly estimated mask of a generic object instance (e.g. the instance mask in the previous frame) to a segmentation of the object instance. In the case of a particular sequence, we fine-tune the extended network, in order to adapt it to the specific object instance based on the few observation of this object instance in the history.

Given the observations of an object instance $o_{1...t-1}^i, i \in \{1...M_o\}$ and the images $I_{1...t-1}$, we obtain $t-2$ training data, each of which contains $\left\langle o_{j-1}^i, I_j, o_j^i \right\rangle, j \in \{1...t-1\}$. Apart from this, we also perform data augmentation for the $t-1$ observations following the data generation method introduced in Section 4.1, in which we randomly generate o_{j-1}^i for $\left\langle I_j, o_j^i \right\rangle$ by applying affine transform and non-rigid deformation. The extended model is fine-tuned based on these training data, in order to learn the appearance of a specific object instance and segment it in the current frame.

4.3 Training Details

Following the descriptions in previous subsections, we provide the training details of our network regarding the offline and online training strategies.

4.3.1 Network Architecture

The base network follows the architecture of VGG network (Simonyan and Zisserman, 2014). VGG network employs 5 groups of convolutional layers with kernel size $3*3$ to extract robust features from an input image. Following each group of convolutional layers, a max pooling layer is provided to downsample the internal feature maps, so that the features can

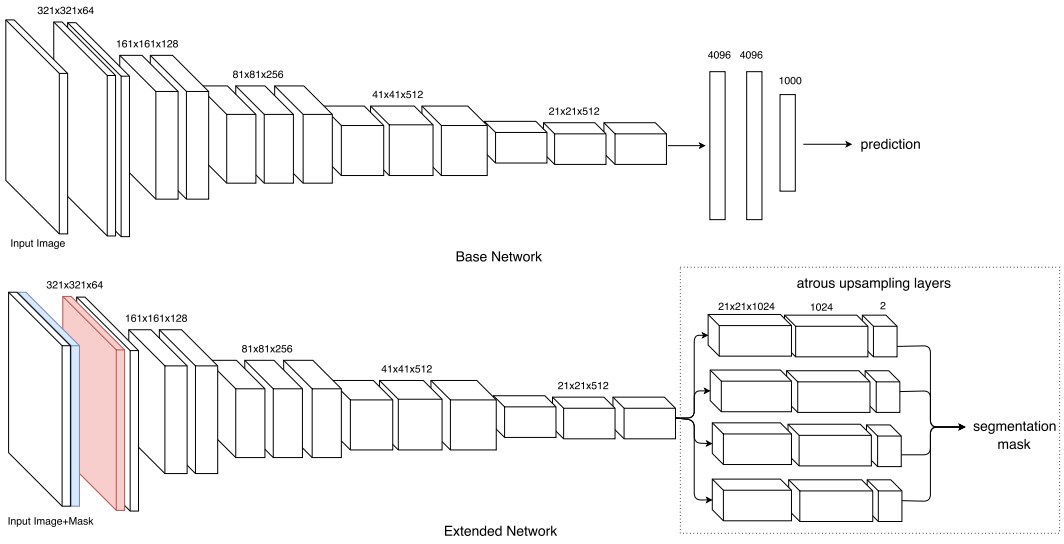

Figure 3: The architectures of the base network and extended network.

be extracted more globally in the following convolutional layers. A Rectified Linear Unit (ReLU) is used as the activation function for each convolutional layer. Similarly, the extended network follows the architecture of Deeplab-ASPP which shares the same features network than VGG network (Simonyan and Zisserman, 2014) and substitutes the fully connected layers in the VGG network with atrous upsampling layers. These atrous upsampling layers perform Atrous Spatial Pyramid Pooling (ASPP) on the feature maps to achieve the dense classification task in semantic segmentation. Following (Khoreva et al., 2016), we extended Deeplab-ASPP to allow an extra mask channel in an input (denoted blue input channel in the extended network in Fig.3) by adding another channel in the filters of the first convolutional layer. Fig.3 shows an illustration of the architecture of the base network and extended. Note that the pooling layers are not shown in the figure for conciseness.

4.3.2 Offline Training

The extended network is initialized from a base network pre-trained on ImageNet for an image classification task. For the added channel in filters of the first convolutional layer (see the red layer in the extended network in Fig.3) and atrous upsampling layers, we use Gaussian initialization. The training data used in the offline training process is generated from serveral datasets (Cheng et al., 2015; Li et al., 2014; Movahedi and Elder, 2010; Shi et al., 2016) by performing affine transformation and thin-plate splines (Bookstein, 1989). That is to say, for each object mask o on image I, we generate transformed and deformed masks of o, which forms several offline training samples. For affine transformation, we consider random scaling ($\pm$5% of object size), translation ($\pm$10% shift) and rotation ($\pm 10^{\circ}$). For deformation, we use 5 control points and randomly shift them within $\pm$10% margin of the original object mask. Next, the mask is coarsened using dilation operation with 5 pixel radius. This mask deformation procedure is applied over all object instances in the training set. For each image two different masks are generated.

We use Stochastic Gradient Descent (SGD) with mini-batches of 10 images and a polynomial learning policy with initial learning rate of 0.001. The momentum and weight decay are set to 0.9 and 0.0005, respectively. The network is trained for 20k iterations.

4.3.3 Online Training

For online adaptation, we fine-tune the model previously trained offline for 200 iterations with training samples generated from the few observations in the history. We augment the few observations by image flipping and rotations as well as by deforming the annotated masks for an extra channel via affine and non-rigid deformations with the same parameters as for the offline training. This results in an augmented set of 10^3 training images. The network is trained with the same learning parameters as for offline training, fine-tuning all convolutional layers.

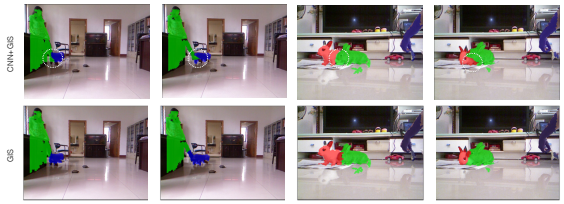

Figure 4: Examples of qualitative results from CNN+GIS in the first row and GIS in the second row.

5 EXPERIMENTS

In this section, we report the experiment results in the RGBD video foreground segmentation dataset (Fu et al., 2017) comparing with the classical generic instance segmentation (GIS) method introduced in (Lin et al., 2018). The RGBD video foreground segmentation dataset (Fu et al., 2017) contains 12 RGBD sequences captured in 7 different types of scenes with multiple objects. Since blob segmentation is needed only when objects interact with each other (physically attached), we perform both the CNN based generic instance segmentation (CNN+GIS) and the classical GIS on all the sequences, but evaluation is only made in frames which involve object interactions. We keep the evaluation metrics used by (Lin et al., 2018) in the experiment as mean Intersection over Union (mIoU). Fig.4 shows some comparison results, in which results from CNN+GIS are shown in the first row and results from GIS in the second row. CNN+GIS obtains clearly improved segmentation results than GIS due to the better defined unary energy (see the better object boundaries obtained in CNN+GIS). A quantitative comparison is also made on this dataset, shown in Table 1. Apart from GIS and CNN+GIS, we introduce a comparison to CNN+GIS without performing online training (CNN+GIS-OT). CNN+GIS obtains around 6% higher mIoU than GIS, whereas CNN+GIS-OT also outperforms GIS with around 2% higher mIoU. To fully exploit the RGBD data, we have also explored the possibility to incorporate the depth map as an extra input channel in CNN+GIS, however no improvement is observed, while the complexity is increased.

Table 2 shows average time spent for building the unary energy in GIS, CNN+GIS and CNN+GIS-OT in one blob segmentation respectively. Although CNN+GIS outperforms GIS in mIoU, the computational complexity is higher than GIS. With a trade-off in accuracy, the computation complexity of CNN+GIS can be decreased by eliminating online training process or reducing the online training samples to obtain the expected run-time performance in the applications.

	mIoU
GIS	67.2
CNN+GIS	73.5
CNN+GIS-OT	69.1

Table 1: Quanitative comparision between GIS, CNN+GIS and CNN+GIS without performing online training

	time
GIS	0.07s
CNN+GIS	12s
CNN+GIS-OT	0.09s

Table 2: Run-time performance of building the unary energy in GIS, CNN+GIS and CNN+GIS without performing online training

6 CONCLUSION

In this paper, we have presented a method which combines the strong object representation power in CNN based semantic segmentation methods and the genericity in the generic instance segmentation method introduced in (Lin et al., 2018), and applied the combined approach to solve an instance segmentation problem. We verify the feasibility of employing one-shot learning method to model object instances with very few examples discovered by the generic object instance segmentation (GIS) method. The experiment results illustrate that an improved segmentation performance can be obtained by combining those two methods. On the other hand, instance independent learned features for tracking obtain a better result than hand-crafted features based on color, shape and 3D distance, with just a slight increase of the computational time. Features fine-tuned to the instance that is being tracked achieve the best results, but with a much higher run-time performance.

7 ACKNOWLEDGEMENT

This work has been developed in the framework of project TEC2016-75976-R and TEC2013-43935-R, financed by the Spanish Ministerio de Economia y Competitividad and the European Regional Development Fund (ERDF).

REFERENCES

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585.

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Van Gool, L. (2016). One-shot video object segmentation. *arXiv preprint arXiv:1611.05198*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H., and Hu, S.-M. (2015). Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Endres, I. and Hoiem, D. (2010). Category independent object proposals. In *European Conference on Computer Vision*, pages 575–588. Springer.

Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Fu, H., Xu, D., and Lin, S. (2017). Object-based multiple foreground segmentation in rgbd video. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Husain, F., Dellen, B., and Torras, C. (2015). Consistent depth video segmentation using adaptive surface models. *Cybernetics, IEEE Transactions on*, 45(2):266–278.

Khoreva, A., Perazzi, F., Benenson, R., Schiele, B., and Sorkine-Hornung, A. (2016). Learning video object segmentation from static images. *arXiv preprint arXiv:1612.02646*.

Konecnỳ, J. and Hagara, M. (2014). One-shot-learning gesture recognition using hog-hof. *Journal of Machine Learning Research*, 15:2513–2532.

Koo, S., Lee, D., and Kwon, D.-S. (2014). Incremental object learning and robust tracking of multiple objects from rgb-d point set data. *Journal of Visual Communication and Image Representation*, 25(1):108–121.

Lee, Y. J., Kim, J., and Grauman, K. (2011). Key-segments for video object segmentation. In *2011 International Conference on Computer Vision*, pages 1995–2002. IEEE.

Li, Y., Hou, X., Koch, C., Rehg, J. M., and Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287.

Lin, X., Casas, J., and Pardàs, M. (2016). 3d point cloud segmentation oriented to the analysis of interactions. In *The 24th European Signal Processing Conference (EUSIPCO 2016)*. Eurasip.

Lin, X., Casas, J. R., and Pardàs, M. (2018). Temporally coherent 3d point cloud video segmentation in generic scenes. *IEEE Transactions on Image Processing*.

Movahedi, V. and Elder, J. H. (2010). Design and perceptual validation of performance measures for salient object segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56. IEEE.

Shi, J., Yan, Q., Xu, L., and Jia, J. (2016). Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):717–729.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638.