STD2P: RGBD Semantic Segmentation using Spatio-Temporal Data-Driven Pooling

Yang He[1], Wei-Chen Chiu[1], Margret Keuper[2] and Mario Fritz[1]
[1]Max Planck Institute for Informatics,
Saarland Informatics Campus, Saarbrücken, Germany
[2]University of Mannheim, Mannheim, Germany

Abstract

We propose a novel superpixel-based multi-view convolutional neural network for semantic image segmentation. The proposed network produces a high quality segmentation of a single image by leveraging information from additional views of the same scene. Particularly in indoor videos such as captured by robotic platforms or handheld and body-worn RGBD cameras, nearby video frames provide diverse viewpoints and additional context of objects and scenes. To leverage such information, we first compute region correspondences by optical flow and image boundary-based superpixels. Given these region correspondences, we propose a novel spatio-temporal pooling layer to aggregate information over space and time. We evaluate our approach on the NYU–Depth–V2 and the SUN3D datasets and compare it to various state-of-the-art single-view and multi-view approaches. Besides a general improvement over the state-of- the-art, we also show the benefits of making use of unlabeled frames during training for multi-view as well as single-view prediction.

1. Introduction

Consumer friendly and affordable combined image and depth-sensors such as *Kinect* are nowadays commercially deployed in scenarios such as gaming, personal 3D capture and robotic platforms. Interpreting this raw data in terms of a semantic segmentation is an important processing step and hence has received significant attention. The goal is typically formalized as predicting for each pixel in the image plane the corresponding semantic class.

For many of the aforementioned scenarios, an image sequence is naturally collected and provides a substantially richer source of information than a single image. Multiple images of the same scene can provide different views that change the observed context, appearance, scale and occlusion patterns. The full sequence provides a richer observation of the scene and propagating information across views has the potential to significantly improve the accuracy of semantic segmentations in more challenging views as shown in Figure 1.

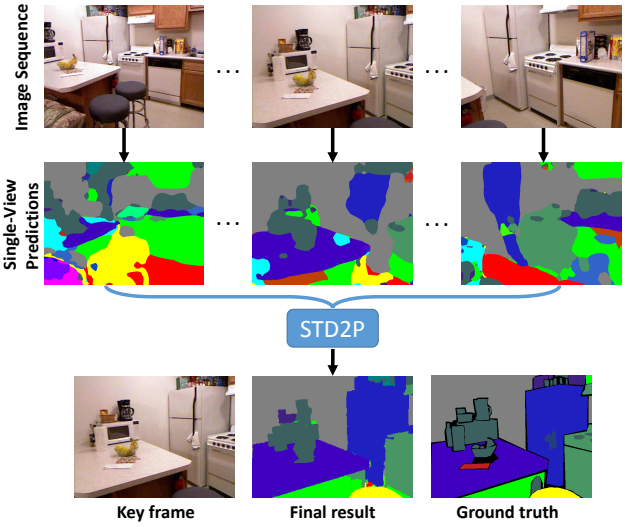

Figure 1: An image sequence can provide rich context and appearance, as well as unoccluded objects for visual recognition systems. Our *Spatio-Temopral Data-Driven Pooling (STD2P)* approach integrates the multi-view information to improve semantic image segmentation in challenging scenarios.

Hence, we propose a multi-view aggregation method by a spatio-temporal data-driven pooling (STD2P) layer which is a principled approach to incorporate multiple frames into any convolutional network architecture. In contrast to previous work on superpixel-based approaches [12, 4, 2], we compute correspondences over time which allows for knowledgeable and consistent prediction over space and time.

As dense annotation of full training sequences is time

consuming and not available in current datasets, a key feature of our approach is training from partially annotated sequences. Notably, our model leads to improved semantic segmentations in the case of multi-view observation *as well as* single-view observation at test time. The main contributions of our paper are:

- We propose a principled way to incorporate superpixels and multi-view information into state-of-the-art convolutional networks for semantic segmentation. Our method is able to exploit a variable number of frames with partial annotation in training time.
- We show that training on sequences with partial annotation improves semantic segmentation for multi-view observation *as well as* single-view observation.
- We evaluate our method on the challenging semantic segmentation datasets *NYU–Depth–V2* and *SUN3D*. There, it outperforms several baselines as well as the state-of-the-art. In particular, we improve on difficult classes not well captured by other methods.

2. Related work

2.1. Context modeling for fully convolutional networks

Fully convolutional networks (FCN) [26], built on deep classification networks [19, 34], carried their success forward to semantic segmentation networks that are end-to-end trainable. Context information plays an important role in semantic segmentation [28], so researchers tried to improve the standard FCN by modeling or providing context in the network. Liu *et al.* [24] added global context features to a feature map by global pooling. Yu *et al.* [39] proposed dilation convolutions to aggregate wider context information. In addition, graphical models are applied to model the relationship of neuron activation [5, 40, 25, 22]. Particularly, Chen *et al.* [5] combined the strengths of conditional random field (CRF) with CNN to refine the prediction, and thus achieved more accurate results. Zheng *et al.* [40] formulated CRFs as recurrent neural networks (RNN), and trained the FCN and the CRF-RNN end-to-end. Recurrent neural networks have also been used to replace graphical models in learning context dependencies [3, 33, 21], which shows benefits in complicated scenarios.

Recently, incorporating superpixels in convolutional networks has received much attention. Superpixels are able to not only provide precise boundaries, but also to provide adaptive receptive fields. For example, Dai *et al.* [8] designed a convolutional feature masking layer for semantic segmentation, which allows networks to extract features in unstructured regions with the help of superpixels. Gadde *et al.* [12] improved the semantic segmentation using superpixel convolutional networks with bilateral inception, which can merge initial superpixels by parameters and generate different levels of regions. Caesar *et al.* [4] proposed a novel network with free-form ROI pooling which leverages superpixels to generate adaptive pooling regions. Arnab *et al.* [2] modeled a CRF with superpixels as higher order potentials, and achieved better results than previous CRF based methods [5, 40]. Both methods showed the merit of providing superpixels to networks, which can generate more accurate segmentations. Different from prior works [12, 4], we introduce superpixels at the end of convolutional networks instead of in the intermediate layers and also integrate the response from multiple views with average pooling, which has been used to replace the fully connected layers in classification [23] and localization [41] tasks successfully.

2.2. Semantic segmentation with videos

The aim of multi-view semantic segmentation is to employ the potentially richer information from diverse views to improve over segmentations from a single view. Couprie *et al.* [7] performed single image semantic segmentation with learned features with color and depth information, and applied a temporal smoothing in test time to improve the performance of frame-by-frame estimations. Hermans *et al.* [16] used the Bayesian update strategy to fuse new classification results and a CRF model in 3D space to smooth the segmentation. Stückler *et al.* [35] used random forests to predict single view segmentations, and fused all views to the final output by a simultaneous localization and mapping (SLAM) system. Kundu *et al.* [20] built a dense 3D CRF model with correspondences from optical flow to refine semantic segmentation from video. Recently, McCormac *et al.* [27] proposed a CNN based semantic 3D mapping system for indoor scenes. They applied a SLAM system to build correspondences, and mapped semantic labels predicted from CNN to 3D point cloud data. Mustikovela *et al.* [29] proposed to generate pseudo ground truth annotations for auxiliary data with a CRF based framework. With the auxiliary data and their generated annotations, they achieved a clear improvement. In contrast to the above methods, instead of integrating multi-view information by using graphical models, we utilize optical flow and image superpixels to establish region correspondences, and design a superpixel based multi-view network for semantic segmentation.

3. Fully convolutional multi-view segmentation with region correspondences

Our goal is a multi-view semantic segmentation scheme, that integrates seamlessly into exciting deep architectures and produces highly accurate semantic segmentation of a single view. We further aim at facilitating training from partially annotated input sequences, so that existing datasets

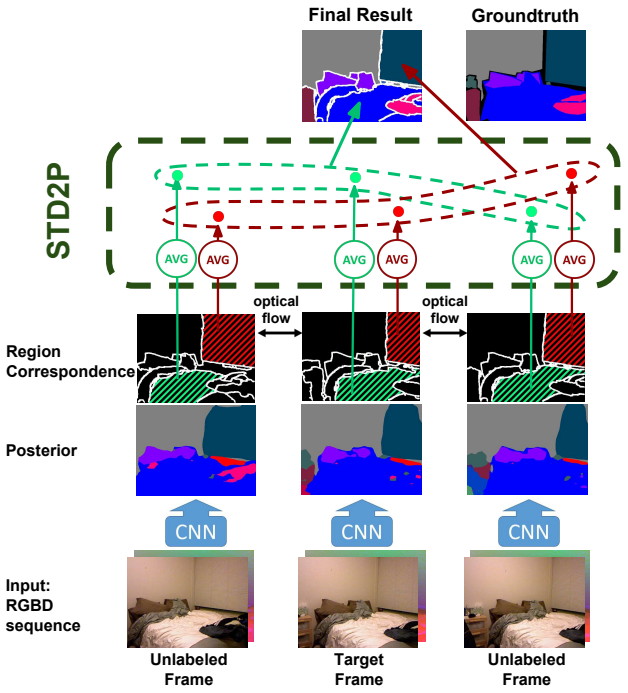

Figure 2: Pipeline of the proposed method. Our multi-view semantic segmentation network is built on top of a CNN. It takes a RGBD sequence as input and computes the semantic segmentation of a target frame with the help of unlabeled frames. We use superpixels and optical flow to establish region correspondences, and fuse the posterior from multiple views with the proposed Spatio-Temporal Data-Driven Pooling (STD2P).

can be used and the annotation effort stays moderate for new datasets. To this end, we draw on prior work on high quality non-semantic image segmentation and optical flow which is input to our proposed Spatio-Temporal Data-Driven Pooling (STD2P) layer.

Overview. As illustrated in Figure 2, our method starts from an image sequence. We are interested in providing an accurate semantic segmentation of one view in the sequence, called *target frame*, which can be located at any position in the image sequence. The two components that distinguish our approach from a standard fully convolutional architecture for semantic segmentation are, first, the computation of region correspondences and, second, the novel spatio-temporal pooling layer that is based on these correspondences.

We first compute the superpixel segmentation of each frame and establish region correspondences using optical flow. Then, the proposed data-driven pooling allows to aggregate information first within superpixels and then along their correspondences inside a CNN architecture. Thus, we achieve a tight integration of the superpixel segmentation and multi-view aggregation into a deep learning framework for semantic segmentation.

3.1. Region correspondences

Motivated by the recent success of superpixel based approaches in deep learning architectures [12, 4, 1, 9] and the reduced computational load, we decide for a region-based approach. In the following, we motivate and detail our approach on establishing robust correspondences.

Motivation. One key idea of our approach is to map information from potentially unlabeled frames to the target frame, as diverse view points can provide additional context and resolve challenges in appearance and occlusion as illustrated in Figure 1. Hence, we do not want to assume visibility or correspondence of objects across all frames (e.g. the nightstand in the target frame as shown in Figure 2). Therefore, video supervoxel methods such as [13] that force interframe correspondences and do not offer any confidence measure are not suitable. Instead, we establish the required correspondences on a frame-wise region level.

Superpixels & optical flow. We compute RGBD superpixels [15] in each frame to partition a RGBD image into regions, and apply Epic flow [31] between each pair of consecutive frames to link these regions. To take advantage of the depth information, we utilize the RGBD version of the structured edge detection [10] to generate boundary estimates. Then, Epic flow is computed in forward and backward directions.

Robust spatio-temporal matching. Given the precomputed regions in the target frame and all unlabeled frames as well as the optical flow between those frames, our goal is to find highly reliable region correspondences. For any two regions R_t in the target frame f_t and R_u in an unlabeled frame f_u, we compute their matching score from their intersection over union (IoU). Let us assume w.l.o.g. that $u < t$. Then, we warp R_u from f_u to $R_u^{'}$ in f_t using forward optical flow. The IoU between R_t and $R_u^{'}$ is denoted by $\overrightarrow{IoU}_{tu}$. Similarly, we compute $\overleftarrow{IoU}_{tu}$ with backward optical flow. We regard R_t and R_u as a successful match if their matching score meets $\min(\overleftarrow{IoU}_{tu}, \overrightarrow{IoU}_{tu}) > \tau$. We keep the one with the highest matching score if R_t has several successful matches. We show the statistics of region correspondences on the NYUDv2 dataset in Figure 3.

It shows that 87.17% of the regions are relatively small (less than 2000 pixels) The plot on the right shows that those small regions generally only find less than 10 matches in a whole video. Contrariwise, even slightly bigger regions

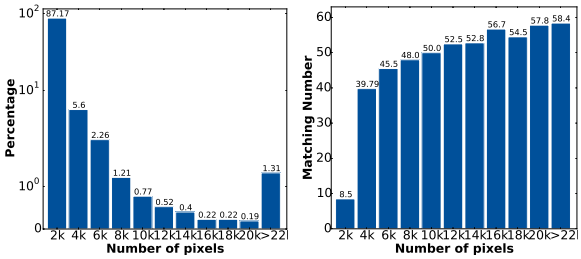

Figure 3: Statistics of region correspondences on the NYUDv2 dataset. (left) Distribution of region sizes; (right) Histogram of the average number of matches over region sizes.

can be matched more easily and they cover large portions of images. They usually have more than 40 matches in a whole video, and thus provide adequate information for our multi-view network.

3.2. Spatio-Temporal Data-Driven Pooling (STD2P)

Here, we describe our Spatio-Temporal Data-Driven Pooling (STD2P) model that uses the spatio-temporal structure of the computed region correspondences to aggregate information across views as illustrated in Figure 2. While the proposed method is highly compatible with recent CNN and FCN models, we build on a per frame model using [26]. In more detail, we refine the output of the deconvolution layer with superpixels and aggregate the information from multiple views by three layers: a spatial pooling layer, a temporal pooling layer and a region-to-pixel layer.

Spatial pooling layer. The input to the spatial pooling layer is a feature map $I_s \in R^{N\times C\times H\times W}$ for N frames, C channels with size $H \times W$ and a superpixel map $S \in R^{N\times H\times W}$ encoded with the region index. It generates the output $O_s \in R^{N\times C\times P}$, where P is the maximum number of superpixels. The superpixel map S guides the forward and backward propagation of the layer. Here, $\Omega_{ij} = \{(x,y)|S(i,x,y) = j\}$ denotes a superpixel in the i-th frame with region index j. Then, the forward propagation of spatial average pooling can be formulated as

$$O_s(i,c,j) = \frac{1}{|\Omega_{ij}|}\sum_{(x,y)\in\Omega_{ij}} I_s(i,c,x,y) \tag{1}$$

for each channel index c of the i-th frame and region index j. We train our model using stochastic gradient descent. The gradient of the input $I_s(i,c,x,y)$, where $(x,y) \in \Omega_{ij}$, in our spatial pooling is calculated by back propagation [32],

$$\begin{aligned}\frac{\partial L}{\partial I_s(i,c,x,y)} &= \frac{\partial L}{\partial O_s(i,c,j)}\frac{\partial O_s(i,c,j)}{\partial I_s(i,c,x,y)} \\ &= \frac{1}{|\Omega_{ij}|}\frac{\partial L}{\partial O_s(i,c,j)}.\end{aligned} \tag{2}$$

Temporal pooling layer. Similarly, we formulate our temporal pooling which fuses the information from N frames $I_t \in R^{N\times C\times P}$, which is the output of spatial pooling layer, to one frame $O_t \in R^{C\times P}$. This layer also needs superpixel information Ω_{ij}, which is the superpixel with index j of the i-th input frame. If $\Omega_{ij} \neq \varnothing$, there exists correspondence. The forward propagation can be expressed as

$$O_t(c,j) = \frac{1}{K}\sum_{\Omega_{ij}\neq\varnothing} I_t(i,c,j) \tag{3}$$

for channel index c and region index j, where $K = |\{i|\Omega_{ij} \neq \varnothing, 1 \leq i \leq N\}|$, which is the number of matched frames for j-th region. The gradient is calculated by

$$\begin{aligned}\frac{\partial L}{\partial I_t(i,c,j)} &= \frac{\partial L}{\partial O_t(c,j)}\frac{\partial O_t(c,j)}{\partial I_t(i,c,j)} \\ &= \frac{1}{K}\frac{\partial L}{\partial O_t(c,j)}.\end{aligned} \tag{4}$$

Region-to-pixel layer. To directly optimize a semantic segmentation model with dense annotations, we map the region based feature map $I_r \in R^{C\times P}$ to a dense pixel-level prediction $O_r \in R^{C\times H\times W}$. This layer needs a superpixel map on the target frame $S_{\text{target}} \in R^{H\times W}$ to perform forward and backward propagation. The forward propagation is expressed as

$$O_r(c,x,y) = I_r(c,j), \quad S_{\text{target}}(x,y) = j. \tag{5}$$

The gradient is computed by

$$\begin{aligned}\frac{\partial L}{\partial I_r(c,j)} &= \sum_{S_{\text{target}}(x,y)=j}\frac{\partial L}{\partial O_r(c,x,y)}\frac{\partial O_r(c,x,y)}{\partial I_r(c,j)} \\ &= \sum_{S_{\text{target}}(x,y)=j}\frac{\partial L}{\partial O_r(c,x,y)}.\end{aligned} \tag{6}$$

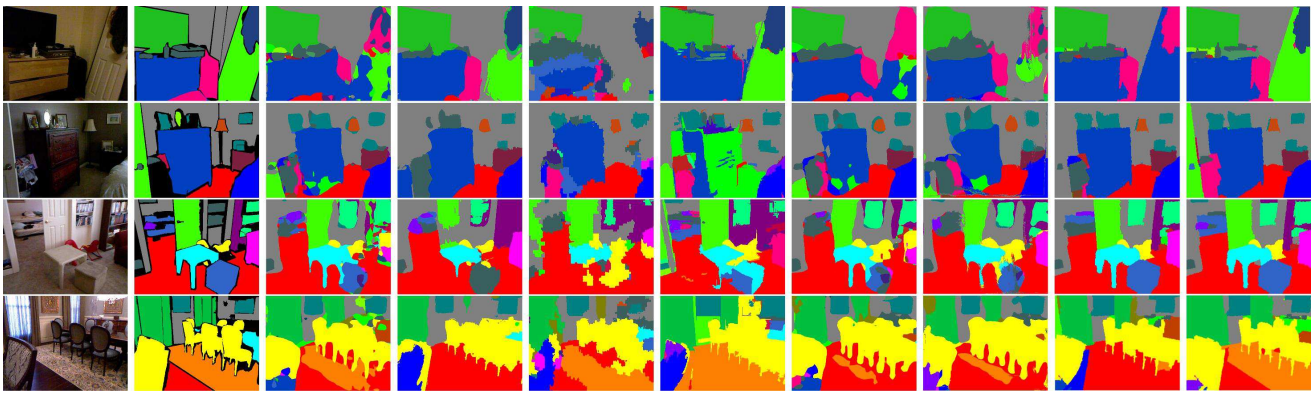

Figure 4: Visualization examples of the semantic segmentation on NYUDv2. Column 1 shows the RGB images and column 2 shows the ground truth (black represents the unlabeled pixels). Columns 3 to 6 show the results from CRF-RNN [40], DeepLab-LFOV [6], BI(3000) [12] and E2S2 [4], respectively. Columns 7 to 9 show the results from FCN [26], single-view superpixel and multi-view pixel baselines. The results from our whole system are shown in column 10. Best viewed in color.

Implementation details. We regard the frames with groundtruth annotations as target frames. For each target frame, we equidistantly sample up to 100 frames around it with the static interval of 3 frames. Next, we compute the superpixels [15] and Epic flow [31] with the default settings provided in the corresponding source codes. The threshold τ for the computation of region correspondence is 0.4 (cf. section 3.1). Finally, for each RGBD sequence, we randomly sample 11 frames including the target frame together with their correspondence maps as the input for our network. We use RGB images and HHA representations of depth [15] and train the network by stochastic gradient descent with momentum term. Due to the memory limitation, we first run FCN and cache the output *pool4_rgb* and *pool4_hha*. Then, we finetune the layers after *pool4* with a new network which is the copy of the higher layers in FCN. We use a minibatch size of 10, momentum 0.9, weight decay 0.0005 and fixed learning rate 10^{-14}. We finetune our model by using cross entropy loss with 1000 iterations for all our models in the experiments. We implement the proposed network using the *Caffe* framework [17], and the source code is available at `https://github.com/SSAW14/STD2P`.

4. Experiments and analysis

We evaluate our approach on the 4-class [30], 13-class [7], and 40-class [14] tasks of the *NYU–Depth–V2* (NYUDv2) dataset [30], and 33-class task of the SUN3D dataset [38].

The NYUDv2 dataset contains 518 RGBD videos, which have more than 400,000 images. Among them, there are 1449 densely labeled frames, which are split into 795 training images and 654 testing images. We follow the experimental settings of [9] to test on 65 labeled frames. We compare our models of different settings to previous state-of-the-art multi-view methods as well as single-view methods, which are summarized in Table 1. We report the results on the labeled frames, using the same evaluation protocol and metrics as [26], pixel accuracy (*Pixel Acc.*), mean accuracy (*Mean Acc.*), region intersection over union (*Mean IoU*), and frequency weighted intersection over union (*f.w. IoU*).

Table 1: Configurations of competing methods

	RGB	RGBD
Single-View	[11, 18]	[4, 5, 6, 9, 12, 15, 26, 36, 37, 40]
Multi-View	/	[7, 16, 35, 27]

4.1. Results on NYUDv2 40-class task

Table 2 evaluates performance of our method on NYUDv2 40-class task and compares to state-of-the-art methods and related approaches [26, 9, 15, 18, 11, 40, 5, 6, 12, 4] [1]. We include 3 versions of our approach:

Our *superpixel* model is trained on single frames without additional unlabeled data, and tested using a single target frame. It improves the baseline FCN on all four metrics by at least 2 percentage points (*pp*), and it achieves in particular

[1]For [26, 9, 15, 18, 11], we copy the performance from their paper. For [40, 5, 6, 12, 4], we run the code provided by the authors with RGB+HHA images. Specifically, for [12], we also increase the maximum number of superpixels from 1000 to 3000. The original coarse version and the fine version are abbreviated as BI(1000) and BI(3000).

Table 2: Performance of the 40-class semantic segmentation task on NYUDv2. We compare our method to various state-of-the-art methods: [26, 15, 18, 11] are also based on convolutional networks, [5, 40, 6] are the models based on convolutional networks and CRF, and [12, 4, 9] are region labeling methods, and thus related to ours. We mark the best performance in all methods with **BOLD** font, and the second best one is written with UNDERLINE.

Methods	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	blinds	desk	shelves
Mutex Constraints [9]	65.6	79.2	51.9	66.7	41.0	55.7	36.5	20.3	33.2	32.6	44.6	53.6	49.1	10.8	9.1
RGBD R-CNN [15]	68.0	81.3	44.9	65.0	47.9	47.9	29.9	20.3	32.6	18.1	40.3	51.3	42.0	11.3	3.5
Bayesian SegNet [18]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Multi-Scale CNN [11]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CRF-RNN [40]	70.3	81.5	49.6	64.6	51.4	50.6	35.9	24.6	38.1	36.0	48.8	52.6	47.6	13.2	7.6
DeepLab [5]	67.9	83.0	53.1	66.8	57.8	57.8	43.4	19.4	45.5	41.5	49.3	58.3	47.8	15.5	7.3
DeepLab-LFOV [6]	70.2	85.2	55.3	68.9	**60.5**	59.8	**44.5**	25.4	**47.8**	**42.6**	47.9	57.7	**52.4**	**20.7**	9.1
BI (1000) [12]	62.8	66.8	44.2	47.7	35.8	35.9	10.9	18.3	21.5	35.9	41.5	30.9	47.4	12.8	8.5
BI (3000) [12]	61.7	68.1	45.2	50.6	38.9	40.3	26.2	20.9	36.0	34.4	40.8	31.6	48.3	9.3	7.9
E2S2 [4]	56.9	67.8	50.0	59.5	43.8	44.3	31.3	24.6	37.9	32.7	46.1	45.0	51.8	15.8	9.1
FCN [26]	69.9	79.4	50.3	66.0	47.5	53.2	32.8	22.1	39.0	36.1	50.5	54.2	45.8	11.9	8.6
Ours (*superpixel*)	70.9	83.4	52.6	68.5	54.1	56.0	40.4	25.5	38.4	40.9	51.5	54.8	47.3	11.3	7.5
Ours (*superpixel+*)	72.4	84.3	52.0	71.5	54.3	58.8	37.9	28.2	41.9	38.5	52.3	58.2	49.7	14.3	8.1
Ours (*full model*)	**72.7**	**85.7**	**55.4**	**73.6**	58.5	**60.1**	42.7	**30.2**	42.1	41.9	**52.9**	**59.7**	46.7	13.5	**9.4**

Methods	curtain	dresser	pillow	mirror	floormat	clothes	ceiling	books	fridge	tv	paper	towel	showercurtain	box	whiteboard
Mutex Constraints [9]	**47.6**	27.6	**42.5**	30.2	32.7	12.6	56.7	8.9	21.6	19.2	**28.0**	28.6	22.9	1.6	1.0
RGBD R-CNN [15]	29.1	34.8	34.4	16.4	28.0	4.7	60.5	6.4	14.5	31.0	14.3	16.3	4.2	2.1	14.2
Bayesian SegNet [18]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Multi-Scale CNN [11]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CRF-RNN [40]	34.8	33.2	34.7	20.8	24.0	18.7	**60.9**	29.5	31.2	41.1	18.2	25.6	23.0	7.4	13.9
DeepLab [5]	32.9	34.3	40.2	23.7	15.0	20.2	55.1	22.1	30.6	49.4	21.8	32.1	6.4	5.8	14.8
DeepLab-LFOV [6]	36.0	36.9	41.4	32.5	16.0	17.8	58.4	20.5	**45.1**	48.0	21.0	**41.5**	9.4	**8.0**	14.3
BI (1000) [12]	29.3	20.3	21.7	13.0	18.2	14.1	44.7	10.9	21.5	30.4	18.8	22.3	17.7	5.5	12.4
BI (3000) [12]	30.8	22.9	19.5	13.9	16.1	13.7	42.5	21.3	16.6	30.9	14.9	23.3	17.8	3.3	9.9
E2S2 [4]	38.0	34.8	31.5	31.7	25.3	14.2	39.7	26.7	27.1	35.2	17.8	21.0	19.9	7.4	**36.9**
FCN [26]	32.5	31.0	37.5	22.4	13.6	18.3	59.1	27.3	27.0	41.9	15.9	26.1	14.1	6.5	12.9
Ours (*superpixel*)	34.5	41.6	37.7	20.1	15.9	20.1	56.8	28.8	23.8	51.8	19.1	26.6	**29.3**	6.8	4.7
Ours (*superpixel+*)	42.9	35.9	40.8	27.7	31.9	19.3	55.6	28.2	38.3	46.9	17.6	31.2	11.0	6.5	28.2
Ours (*full model*)	40.7	**44.1**	42.0	**34.5**	**35.6**	**22.2**	55.9	**29.8**	41.7	**52.5**	21.1	34.4	15.5	7.8	29.2

Methods	person	nightstand	toilet	sink	lamp	bathtub	bag	other struct	other furni	other props	Pixel Acc.	Mean Acc.	Mean IoU	f.w. IoU
Mutex Constraints [9]	9.6	30.6	48.4	41.8	28.1	27.6	0	9.8	7.6	24.5	63.8	-	31.5	48.5
RGBD R-CNN [15]	0.2	27.2	55.1	37.5	34.8	38.2	0.2	7.1	6.1	23.1	60.3	-	28.6	47.0
Bayesian SegNet [18]	-	-	-	-	-	-	-	-	-	-	68.0	45.8	32.4	-
Multi-Scale CNN [11]	-	-	-	-	-	-	-	-	-	-	65.6	45.1	34.1	51.4
CRF-RNN [40]	57.9	31.4	57.2	45.4	36.9	39.1	4.9	14.6	9.5	29.5	66.3	48.9	35.4	51.0
DeepLab [5]	55.3	37.7	57.9	47.7	40.0	44.7	6.6	18.0	12.9	**33.8**	68.7	46.9	36.8	52.5
DeepLab-LFOV [6]	**67.0**	41.8	**69.7**	46.8	**40.1**	**45.1**	2.1	**20.7**	12.4	33.5	**70.3**	49.6	39.4	54.7
BI (1000) [12]	45.9	15.8	56.5	32.2	24.7	17.1	0.1	12.2	6.7	21.9	57.7	37.8	27.1	41.9
BI (3000) [12]	44.7	15.8	53.8	32.1	22.8	19.0	0.1	12.3	5.3	23.2	58.9	39.3	27.7	43.0
E2S2 [4]	35.0	17.6	31.8	36.3	14.8	26.0	**9.9**	14.5	9.3	20.9	58.1	52.9	31.0	44.2
FCN [26]	57.6	30.1	61.3	44.8	32.1	39.2	4.8	15.2	7.7	30.0	65.4	46.1	34.0	49.5
Ours (*superpixel*)	66.1	37.4	56.1	46.3	34.5	26.7	5.8	12.7	12.3	30.6	68.5	48.7	36.0	52.9
Ours (*superpixel+*)	66.7	34.1	62.8	**47.8**	35.1	26.4	8.8	19.3	10.9	29.2	68.4	52.1	38.1	54.0
Ours (*full model*)	60.7	**42.2**	62.7	47.4	38.6	28.5	7.3	18.8	**15.1**	31.4	70.1	**53.8**	**40.1**	**55.7**

better performance than recently proposed methods based on superpixels and CNN[12, 4].

Our *superpixel+* model leverages additional unlabeled data in the training while it only uses the target frame for test. It obtains 3.4*pp*, 2.1*pp*, 1.1*pp* improvements over the *superpixel* model on *Mean Acc.*, *Mean IoU* and *f.w. IoU*, leading to more favorable performance than many state-of-the-art methods [9, 15, 18, 11, 40, 5, 12, 4]. This highlights the benefits of leveraging unlabeled data.

Table 3: Comparison of average and max spatio-temporal data-driven pooling.

Spatial/Temporal	Pixel Acc.	Mean Acc.	Mean IoU	f.w. IoU
AVG / AVG	**70.1**	**53.8**	**40.1**	**55.7**
AVG / MAX	69.4	51.0	38.0	54.4
MAX / AVG	66.4	45.4	33.8	49.6
MAX / MAX	64.9	44.5	32.1	47.9

Our *full model* leverages additional unlabeled data both in the training and test. It achieves a consistent improvement over the *superpixel+* model and outperforms all competitors in *Mean Acc.*, *Mean IoU* and *f.w. IoU* by $0.9pp, 0.7pp, 1.0pp$ respectively. Particularly strong improvements are observed on challenging object classes such as dresser(+$7.2pp$), door(+$4.8pp$), bed(+$4.7pp$) and TV(+$3.1pp$).

Figure 4 demonstrates that our method is able to produce smooth predictions with accurate boundaries. We present the most related methods, which either apply CRF [40, 6] or incorporate superpixels [12, 4], in the columns 3 to 6 of this figure. According to the qualitative comparison to these approaches, we can see the benefit of our method. It captures small objects like chair legs, as well as large areas like floormat and door. In addition, we also present FCN and the *superpixel* model at the 7-th and 8-th column of Figure 4. The FCN is boosted by introducing superpixels but not as precise as our *full model* using unlabeled data.

Average vs. max spatio-temporal data-driven pooling. Our data-driven pooling aggregates the local information from multiple observations within a segment and across multiple views. Average pooling and max pooling are canonical choices used in many deep neural network architectures. Here we test average pooling and max pooling both in the spatial and temporal pooling layer, and show the results in Table 3. All the models are trained with multiple frames, and tested on multiple frames. Average pooling turns out to perform best for spatial and temporal pooling. This result confirms our design choice.

Region vs. pixel correspondences. We compare our *full model*, which is built on the region correspondences, to the model with pixel correspondences. It only uses the per-pixel correspondences by optical flow and applies average pooling to fuse the information from multiple view. The visualization results of this baseline are presented in column 9 of Figure 4. Obtaining accurate pixel correspondences is challenging because the optical flow is not perfect and the error can accumulate over time. Consequently, the model with pixel correspondences only improves slightly over the FCN baseline, as it is also reflected in the numbers in Table 4. Establishing region correspondences with the proposed rejection strategy described in section 3.1 seems indeed to be favorable over pixel correspondences. Our *full model* shows a significant improvement over the pixel-correspondence baseline and FCN in all 4 measures.

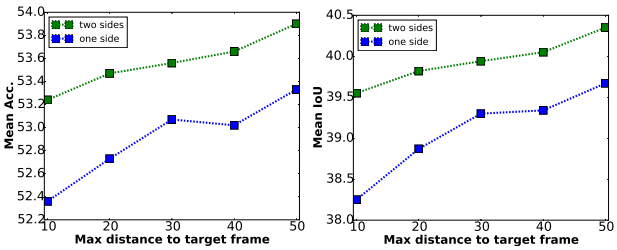

Figure 5: The performance of multi-view prediction with varying maximum distance. Green lines show the results of using future and past views. Blue lines show the results of only using past views.

Table 4: Comparison results with baselines on NYUDv2 40-class task

Methods	Pixel Acc.	Mean Acc.	Mean IoU	f.w. IoU
FCN [26]	65.4	46.1	34.0	49.5
Pixel Correspondence	66.2	45.9	34.6	50.2
Superpixel Correspondence	**70.1**	**53.8**	**40.1**	**55.7**

Analysis of multi-view prediction. In our multi-view model, we subsample frames from a whole video for computational considerations. There is a trade-off between close-by and distant frames to be made. If we select frames far away from the target frames, they can provide more diverse views of an object, while matching is more challenging and potentially less accurate than for close-by frames. Hence, we analyze the influence of the distance of selected frames to target frames, and report the *Mean Acc.* and *Mean IoU* in Figure 5. In results, providing wider views is helpful, as the performance is improved with the increase of max distance. And selecting the data in the future, which is another way to provide wider views, also contributes to the improvements of performance.

4.2. Results on NYUDv2 4-class and 13-class tasks

To show the effectiveness of our multi-view semantic segmentation approach, we compare our method to previous state-of-the-art multi-view semantic segmentation methods [7, 16, 35, 27] on the 4-class and 13-class tasks of NYUDv2 as shown in Table 5. Besides, we also present previous state-of-the-art single-view methods [11, 37, 36]. We observe that our *superpixel+* model already outperforms all the multi-view competitors, and the proposed temporal pooling scheme further boosts *Pixel Acc.* and *Mean Acc.* by more than $1pp$ and then outperforms the state-of-the-art

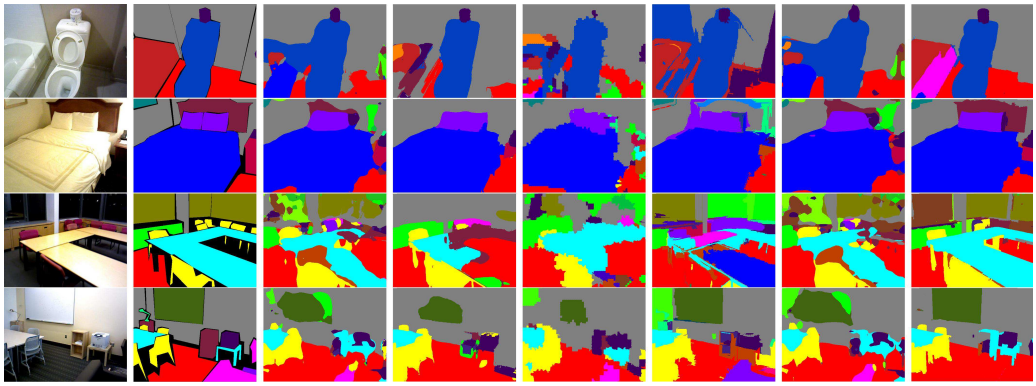

Figure 6: Qualitative results of the SUN3D dataset. For each example, the images are arranged from top to bottom, from left to right as color image, groundtruth, CRF-RNN [40], DeepLab-LFOV [6], BI [12], E2S2 [4], FCN [26] and ours.

Table 5: Performance of the 4-class (left) and 13-class (right) semantic segmentation tasks on NYUDv2.

Methods	Pixel Acc.	Mean Acc.	Pixel Acc.	Mean Acc.
Couprie *et al.* [7]	64.5	63.5	52.4	36.2
Hermans *et al.* [16]	69.0	68.1	54.2	48.0
Stückler *et al.* [35]	70.6	66.8	-	-
McCormac *et al.* [27]	-	-	69.9	63.6
Wang *et al.* [36]	-	65.3	-	42.2
Wang *et al.* [37]	-	74.7	-	52.7
Eigen *et al.* [11]	83.2	82.0	75.4	66.9
Ours (*superpixel+*)	82.7	81.3	74.8	67.0
Ours (*full model*)	**83.6**	**82.5**	**75.8**	**68.4**

Table 6: Performance of the 33-class semantic segmentation task on SUN3D. All 65 images are used as the test set.

Methods	Pixel Acc.	Mean Acc.	Mean IoU	f.w. IoU
Mutex Constraints [9]	**65.7**	-	28.2	51.0
CRF-RNN [40]	59.8	-	25.5	43.3
DeepLab [5]	60.9	30.7	24.0	44.1
DeepLab-LFOV [6]	62.3	35.3	28.2	46.2
BI (1000) [12]	53.8	31.1	20.8	37.1
BI (3000) [12]	53.9	31.6	21.1	37.4
E2S2 [4]	56.7	**47.7**	27.2	43.3
FCN [26]	58.8	38.5	26.1	43.9
Ours (*superpixel+*)	62.5	40.8	29.4	47.8
Ours (*full model*)	65.5	41.2	**32.9**	**51.5**

[11]. In particular, the recent proposed method by McCormac *et al.* [27] is also built on CNN, however, their performance on 13-class task is about $5pp$ worse than ours.

4.3. Results on SUN3D 33-class task

Table 6 shows the results of our method and baselines on the SUN3D dataset. We follow the experimental settings of [9] to test all the methods [9, 40, 5, 6, 12, 4, 26] on all 65 labeled frames in SUN3D, which are trained with the NYUDv2 40-class annotations. After computing the 40-class prediction, we map 7 unseen semantic classes into 33 classes. Specifically, *floormat* is merged to *floor*, *dresser* is merged to *other furni* and five other classes are merged to *other props*. Among all the methods, we achieve the best *Mean IoU* score that our *superpixel+* and *full model* are $1.2pp$ and $4.7pp$ better than [9] and [6] . For *Pixel Acc.*, our method is comparable to the previous state of the art [9]. In addition, we observe that our *superpixel+* model boosts the baseline FCN by $3.7pp$, $2.3pp$, $3.3pp$, $3.9pp$ on the four metrics, and applying multi-view information further improves $3.0pp$, $0.4pp$, $3.5pp$, $3.7pp$, respectively. Besides, we achieve much better performance than DeepLab-LFOV, which is comparable to our model on the NYUDv2 40-class task. This illustrates the generalization capability of our model, even without finetuning on the new domain or dataset.

5. Conclusion

We have presented a novel semantic segmentation approach using image sequences. We design a superpixel-based multi-view semantic segmentation network with spatio-temporal data-driven pooling which can receive multiple images and their correspondence as input. We propagate the information from multiple views to the target frame, and significantly improve the semantic segmentation performance on the target frame. Besides, our method can leverage large scale unlabeled images for training and test, and we show that using unlabeled data also benefits single image semantic segmentation.

Acknowledgments

This research was supported by the German Research Foundation (DFG CRC 1223) and the ERC Starting Grant VideoLearn.

References

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[2] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.

[3] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, 2015.

[4] H. Caesar, J. Uijlings, and V. Ferrari. Region-based semantic segmentation with end-to-end training. In *ECCV*, 2016.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2014.

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

[7] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013.

[8] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015.

[9] Z. Deng, S. Todorovic, and L. Jan Latecki. Semantic segmentation of rgbd images with mutex constraints. In *CVPR*, 2015.

[10] P. Dollár and C. Zitnick. Structured forests for fast edge detection. In *CVPR*, 2013.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[12] R. Gadde, V. Jampani, M. Kiefel, and P. V. Gehler. Superpixel convolutional networks using bilateral inceptions. In *ECCV*, 2016.

[13] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. In *CVPR*, 2010.

[14] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013.

[15] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*. 2014.

[16] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *ICRA*, 2014.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.

[18] A. Kendall, B. Vijay, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016.

[21] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, 2016.

[22] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2015.

[23] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[24] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[25] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[27] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. *arXiv preprint arXiv:1609.05130*, 2016.

[28] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[29] S. K. Mustikovela, M. Y. Yang, and C. Rother. Can ground truth label propagation from video help semantic segmentation? *ECCV Workshop on Video Segmentation*, 2016.

[30] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[31] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[33] B. Shuai, Z. Zuo, G. Wang, and B. Wang. Dag-recurrent neural networks for scene labeling. In *CVPR*, 2016.

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

[35] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke. Dense real-time mapping of object-class semantics from rgb-d video. *Journal of Real-Time Image Processing*, 10(4), 2015.

[36] A. Wang, J. Lu, W. Gang, J. Cai, and T.-J. Cham. Multi-modal unsupervised feature learning for rgb-d scene labeling. In *ECCV*, 2014.

[37] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang. Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks. In *ECCV*, 2016.

[38] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013.

[39] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[41] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.