Joint Future Semantic and Instance Segmentation Prediction

Camille Couprie, Pauline Luc, Jakob Verbeek

Joint Future Semantic and Instance Segmentation Prediction

Camille Couprie[1] Pauline Luc[1,2] Jakob Verbeek[2]

[1] Facebook AI Research, 75002 Paris, France
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France
{coupriec,paulineluc}@fb.com jakob.verbeek@inria.fr

Abstract. The ability to predict what will happen next from observing the past is a key component of intelligence. Methods that forecast future frames were recently introduced towards better machine intelligence. However, predicting directly in the image color space seems an overly complex task, and predicting higher level representations using semantic or instance segmentation approaches were shown to be more accurate. In this work, we introduce a novel prediction approach that encodes instance and semantic segmentation information in a single representation based on distance maps. Our graph-based modeling of the instance segmentation prediction problem allows us to obtain temporal tracks of the objects as an optimal solution to a watershed algorithm. Our experimental results on the Cityscapes dataset present state-of-the-art semantic segmentation predictions, and instance segmentation results outperforming a strong baseline based on optical flow.

1 Introduction

Video prediction appears as a natural objective to develop smarter strategies towards the acquisition of a visual common sense of machines. In the near future, it could help for planning and robotic applications, for instance by anticipating human behavior. Predicting future frames has known many developments in the color space [1,2,3,4,5]. Luc *et al.* [6] proposed to predict future semantic segmentations instead of color intensities. They showed that this space was more relevant, obtaining better results and directly usable high level information.

Recently, Luc *et al.* [7] introduced the more challenging task of forecasting future instance segmentation. In addition to the prediction of the semantic category of every single pixel, instance level segmentation also requires the specification of an object identifier, i.e. the delineation of every object. More specifically, [7] developed a predictive model in the space of convolutional features of the state-of-the-art Mask R-CNN segmentation approach. Although this method leads to the first instance prediction results outperforming a strong optical flow baseline, it has an extensive training time of about six days, and requires the setting of multiple hyperparameters. In addition, the predictions

* Institute of Engineering Univ. Grenoble Alpes

of this feature-based approach are not temporally consistent, i.e. there is no matching or correspondence between the object instances at time t and $t+1$.

We extend semantic segmentation forecasting by proposing a novel representation that encodes both semantic and instance information, with low training requirements and temporally consistent predictions. More specifically, from Mask R-CNN outputs and for each semantic category, we produce a map indicating the objects' presence at each spatial position, and boundaries of instances using distance transforms. An arg-max on the prediction leads to the future semantic segmentation, and the instance segmentation can be obtained by any seeded segmentation approach, such as a watershed for instance. In the following, we use "seeds" or "markers" to denote a set of pixels that mark each of the objects to be segmented. The choice to rely on seeds to obtain the final segmentation maps is a strength of our approach, allowing us to track the instance prediction in time, constituting a novel feature in comparison to [7]. In this work, we show that defining the seeds as a simple linear extrapolation of the centroids' position of past objects leads to satisfying results. Our approach is summarized in Figure 1. Our contributions are the following:

1. We introduce a simple and memory efficient representation that encodes both the semantic and the instance-level information for future video prediction.
2. We model the prediction of the final instance segmentation as a graph optimization problem that we solve with a watershed with optimality guarantees. We show that the proposed solution produces good results compared to a strong optical flow baseline, and note that the formulation allows the use of other seeded graph-based methods.
3. The use of seeds in our final instance segmentation prediction allows us to incorporate tracking of the objects in a very natural way.

2 Related work

We focus in this section on related work on instance and graph based segmentation approaches after briefly reviewing video forecasting.

2.1 Video forecasting

The video prediction task was originally proposed to efficiently model motion dynamics [1] and demonstrate a utility of the learned representation for other tasks like semi-supervised classification [2]. This self-supervised strategy was successfully employed to improve learning abilities in video games [8,9]. Many improvements were introduced to handle uncertainty such as adversarial training [3], or VAE modeling [10,11,12].

Diverse spaces of prediction have been considered besides the color intensities: for instance, flow fields [10], actions [13], and pose [14], or bounding boxes of objects [15]. Choosing the semantic segmentation space like in [6] allows us to significantly reduce the complexity of the predictions in contrast to RGB

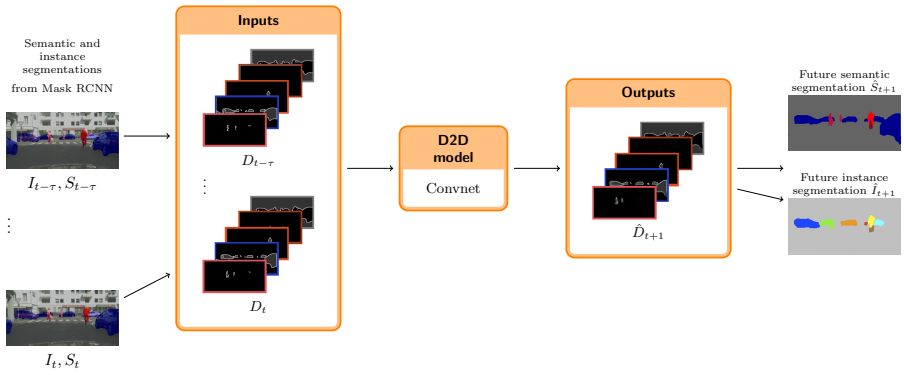

Fig. 1: Our representation enables both future semantic and instance segmentation prediction, based on distance maps from the different objects contours. For each channel of an input segmentation, corresponding to a specific class, the segmentation is decomposed into zeros for background, ones for objects and high values for contours. Then a convnet is trained to predict the future representation. Taking its argmax lets us recover the future semantic segmentation, and computing a watershed from it leads to the future instance segmentation.

values while reaching a very detailed level of spatial information about the scene. Exploiting in addition instance segmentations as in [7] is semantically richer and leads to better anticipated trajectories.

We may notice a complementarity of information arising in forecasting tasks. For instance, [16] perform joint semantic segmentation and optical flow future prediction. In an orthogonal direction, we leverage here the complementarity of instance and semantic segmentation tasks.

2.2 Instance segmentation

Among different instance segmentation methods, some are based on recurrent networks [17], others on watershed transformation [18], or on CRFs [19]. The most successful ones are based on object proposals [20,21]. In particular, the state-of-the-art for instance segmentation was recently set by the Mask R-CNN approach proposed by He *et al.*[21]. Mask R-CNN essentially extends the successful framework of Faster R-CNN for object detection [22] to instance segmentation by adding an extra branch that segments most confidently detected objects.

Distance map based representations were employed for instance segmentation in [23] but differ from our encoding. While they compute distances from object centroids, we instead compute distances from the object contours. Distance maps are also computed in [18], entirely from contours in semantically segmented objects. Again, our case is different, as we are only keeping the distance information in the contour area. Once provided future contour information, our method relies on seeded graph-based segmentation, that we review below.

2.3 Seeded Graph based segmentation

Given weighted graphs, Graph Cuts [24] aim to find a minimum cut between foreground and background seeds and were extended to multi-label segmentation in [25]. Random walker relaxes the Graph Cut problem, by considering the combinatorial Dirichlet problem [26]. Shortest Paths [27] assign each pixel to a given label if there is a shorter path from it to this label's seed than to any other label seed. After links between Graph cuts, Random walker and Shortest Paths were established by Sinop and Grady [28], and links between Graph Cuts and watershed by [29], the unified Power watershed segmentation framework was introduced [30]. It presents a novel watershed algorithm that optimizes an energy function similarly to previously cited works, while having a quasi-linear complexity and being robust to seed sizes. In this work, we take advantage of these properties (speed, accuracy, robustness to seeds size) to compute future instance maps as the solution to an optimization problem.

3 Joint future instance and segmentation prediction

In this section we detail the principle of our approach, after introducing how to infer future semantic segmentation prediction as in [7].

3.1 Background: future semantic segmentation prediction

Given a sequence of images $X_{t-\tau}$ to X_t, Luc *et al.* [7] propose a baseline for predicting future semantic segmentation that encodes the corresponding segmentations $S_{t-\tau}$ to S_t, as computed by the Mask R-CNN network [21]. Given the outputs of Mask R-CNN as lists of instance predictions, composed of a confidence score, a class k, and a binary mask, a semantic segmentation label map is created to form the inputs and targets of a convolutional network. Specifically, the encoding $S_t^{(k)}$ to feed their model, denoted $S2S$, is built as follows: If any instances have been detected in X_t, instances are sorted by order of ascending confidence. For each instance mask, if its confidence score is high enough (in practice above 0.5), the semantic segmentation spatial positions corresponding to the object are updated with label $k \in \{1, ..., K\}$. These semantic segmentation input and target maps are of resolution 128×256, i.e. downsampled by a factor 8 with respect to the original input image's resolution.

A convolutional model is then trained with 4 inputs S_{t-3} to S_t to predict S_{t+1}. This model S_2S constitutes a strong baseline for our work. However, this encoding does not take advantage of the instance information.

3.2 Predicting distance map based representations

Architecture For the previously described baseline S_2S and our proposed D_2D model, we adopt the convolutional network architecture proposed in [6]. It is a single scale convnet composed of 7 layers of convolutions, three of them dilated,

and each of them followed by a ReLU, except for the last one. We use the same feature map scale parameter $q = 1.25$ that allows an efficient training. For the prediction of multiple frames, the single frame prediction model is applied auto-regressively, using its prediction for the previous time step as input to predict the next time step, and so on.

Distance based encoding We now introduce a new method for representing the instance and semantic information together. As illustrated in Figure 1, our method defines a new encoding of the semantic and instance representation at time t called D_t. Our convolutional network will be trained with inputs D_{t-3} to D_t to output the future representation $\hat{D}_{t+1}$. The algorithm to obtain our representation $D_t^{(k)}$ for class k at time t is defined as follows.

We denote each boolean array forming a segmentation mask of instance m in image X_t as $I_t^{(m)}$. The instance segmentation predictions are given by Mask R-CNN outputs, and are downsampled by a factor 8 with respect to the original input image's resolution, similarly to the previously described baseline.

Let us denote the size of a mask $I_t^{(m)}$ by $n \times p$, and (x, y) the integer coordinates of the image pixels. For each instance m of class k, we compute a truncated Euclidean distance map $d_t^{(k,m)}(x, y)$ to the background pixels as described in [31].

More formally,

$$d_t^{(k,m)}(x,y) = \min_{i,j:0\le i<n \text{ and } 0\le j<p \text{ and } I_t^{(m)}(i,j)=0} \lfloor \left((x-i)^2 + (y-j)^2\right)^{\frac{1}{2}} \rfloor. \quad (1)$$

The distance maps of all instances of same class k are merged in $d_t^{(k)}$:

$$d_t^{(k)} = \max_m d_t^{(k,m)}. \quad (2)$$

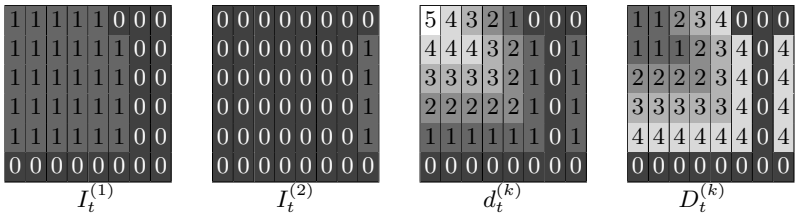

Fig. 2: Illustration of different steps building our distance map based encoding $D_t^{(k)}$ at time t given a class k from individual instance segmentations $I^{(1)}$ and $I^{(2)}$ for class k.

An illustration of this step is shown in Figure 2. For the special case of the background class, the distance is computed relatively to the set of all instances.

We are mostly interested in keeping the contour information in our representation, the distance map in the center of the objects is irrelevant for our task. However, the distance information in a close neighborhood of the objects contours may be useful to introduce some flexibility in the penalization of the prediction errors that are frequent in contours area. The smoothness introduced by the distance information in the contour area allows small mistakes without too much penalization. Therefore, to eliminate the unnecessary distance values of object centers, we bound $d_t^{(k)}$ to θ to flatten the distance values located in the centers of the instances. In practice, we set $\theta = 4$.

As we also want to encode the semantic segmentation information in a way to obtain it from an argmax operation, we transform $d_t^{(k)}$ to indicate objects by ones, and background by zeros: our final action on $d_t^{(k)}$ is therefore to invert its value by multiplying by -1 and adding $(\theta+1)$ in the areas of objects. In summary, from the merged distance map $d_t^{(k)}$ of Equation 2, our encoding is defined as

$$D_t^{(k)} = -\min\left(d_t^{(k)}, \theta\right) + \mathbb{1}(d_t^{(k)})(\theta + 1), \tag{3}$$

where $\mathbb{1}()$ is the indicator function, equal to 1 when $d_t^{(k)} > 0$ and 0 otherwise.

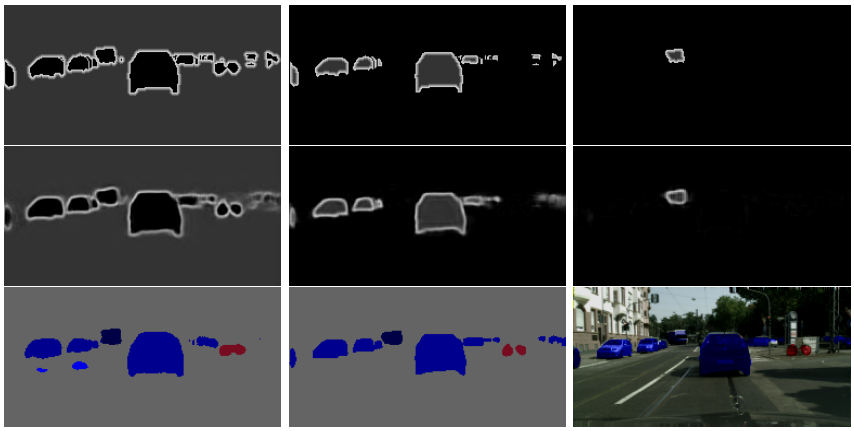

Fig. 3: Distance map inputs for one image. First line: Last input distance maps for classes background, car and truck. Second line: short term predictions for the same classes. Last line: prediction of the baseline $S2S$, distance-based prediction, and distance-based prediction superimposed with future RGB frame.

Examples of such a representation D_t are displayed in Figures 2 and 3. Given inputs $D_{t-\tau}, ..., D_t$, the convolutional network described in the previous section is trained to predict the future D_{t+1}. We denote its output by $\hat{D}_{t+1}$. The final segmentation $\hat{S}_{t+1}$ is then retrieved by computing the argmax over the different

classes:

$$\hat{S}_{t+1} = \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}} \hat{D}^{(k)}_{t+1}. \tag{4}$$

The map of maximum elements may then be exploited to lead to individual object instance segmentations as presented in the next section.

3.3 Forecasting instance segmentation

The obtained map of maxima of our distance based representation contains object contour information as high values, resembling image gradient. As the background class map also contains meaningful object contour information, we add it to the map of maxima to straighten the contour map. We note this contour map

$$W = \max_{k\in\{1,\dots,K\}} \hat{D}^{(k)}_{t+1} + \hat{D}^{(background)}_{t+1}. \tag{5}$$

By construction, its minima form seeds to object instances and background. It is therefore very natural to apply a watershed algorithm on the obtained map. Seeing the map as a topological relief, this method simulates water growing from minima, and builds a watershed line every time different water basins merge.

As studied in [29,30], the watershed transform [32] may be seen as part of a family of graph based optimization methods that includes Graph Cuts, Shortest Paths, Random Walker. The Power watershed algorithm [30] is an optimization algorithm for seeded segmentation that arose from these findings, gathering nice properties: the exact optimization of a graph-based objective, robustness to small seeds and a quasi-linear complexity. These reasons justify the use of the Power watershed approach. In our experiments, we present results using minima as seeds, but also propose a better strategy that allows us to track each object instance. To that end, we identify object tracks from the two preceding instance segmentations and linearly extrapolate their centroid positions, to obtain our object seed.

We now describe the two steps of our instance segmentation method. The first one consists in the extraction of seeds, and the second in graph-based optimization given these seeds. The two steps are illustrated in Figure 4.

Object trajectory forecasting for seed selection Specifically, the creation of our list of seed coordinates z involves:

– Building a graph for the two preceding frames t and $t-1$, where the nodes are the objects centroids, linked by an edge when they are of the same semantic class. Each edge is weighted by a similarity coefficient w depending on the sizes s and average RGB intensities, denoted $c^{(1)}, c^{(2)}, c^{(3)}$ of its nodes:

$$w_{t,t-1} = \frac{|s_t - s_{t-1}|}{\max(s_t, s_{t-1})} + \frac{\sum_{i=1}^{3} \log(\|c_t^{(i)} - c_{t-1}^{(i)}\|^2 + 1)}{3\log(255^2)}. \tag{6}$$

Objects of similar appearance are therefore linked by an edge of small weight.

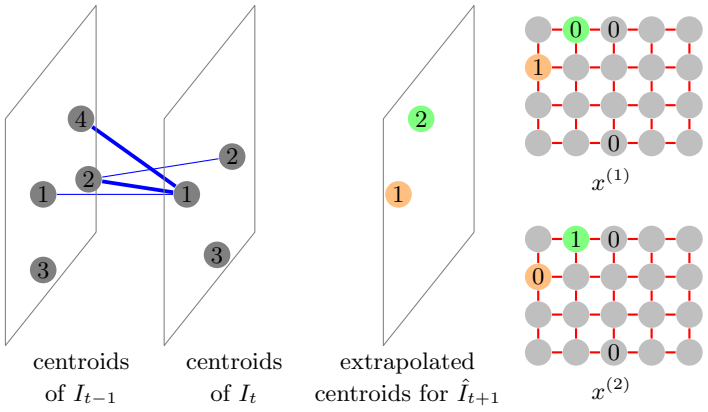

Fig. 4: Computing the future instances. (a) First step: compute the coordinates of future instance centroids positions by selection of shortest paths in the blue graph weighted by w (Equation 6). (b) Second step: Computing the solution $x^{(1)}, x^{(2)}$ to two sub-problems defined in the red graph by Power-watershed optimization (Equation 7), corresponding to continuous labelings of instances 1 and 2. After the computation of the background labeling, given by $x^{(3)} = 1 - (x^{(1)} + x^{(2)})$, the final future instance prediction is given by $\hat{I}_{t+1} = \text{argmax}\,(x^{(1)}, x^{(2)}, x^{(3)})$.

- For each object of frame t : compute the shortest edge to objects of frame $t-1$ when possible. Store the matched centroids trajectory. Remove the edge and its nodes from the graph, and repeat.
- Linear extrapolation of future centroids' coordinates.

This procedure is illustrated in Figure 4a.

Final segmentation step via seeded graph based optimization The Power watershed algorithm is then used with its default parameters ($q = 2, p \to \infty$) to compute an optimal watershed segmentation map.

Formally, a new graph (V, E) is built where the set of nodes V corresponds to instance pixel labels to discover, and edges from E are linking neighboring nodes in a 4 connected setting. The weights W are given by maxima of the network prediction computed from Equation 5. Given a set of L identified instance centroids whose node positions are stored in a vector z, L labelings $x^{(l)}$ on the graph are computed as the solution of

$$\underset{x^{(l)}}{\text{argmin}} \lim_{p \to \infty} \sum_{e_{ij} \in E} (W_i + W_j)^p (x_i^{(l)} - x_j^{(l)})^2, \tag{7}$$

subject to $x_{z_i}^{(l)} = 1$ if $i = l$ and $x_{z_i}^{(l)} = 0$ for all $i \neq l$. For the background segmentation, we define background seeds by a set of two points placed at the middle of the top and bottom halves of the frame. These seeds positions are added to the vector z and therefore enforced in the computation of the $x^{(l)}$. A solution $x^{(l+1)}$ is computed as $x^{(l+1)} = 1 - \sum_{l=1}^{L} x^{(l)}$. An illustration is provided in Figure 4b. The labeling of the graph leads us to our map of future predictions $\hat{I}_{t+1}$ at each pixel i given by

$$\hat{I}_{t+1}(i) = \delta \operatorname*{argmax}_{l} x_i^{(l)}, \tag{8}$$

where δ is the index of the most common class in the corresponding values of $\hat{S}_{t+1}$. For the prediction of future semantic segmentations at multiple time steps, because the network is trained on discrete inputs, we need to adjust the inputs when predicting autoregressively. Instead of applying the model again on the outputs, we discretize the output D_{t+1} by rounding its elements and projecting back the values between 0 and θ. This helps reducing error propagation.

4 Experiments

We now demonstrate that we are able to predict instance and semantic segmentation with an increase in performance for the latter task.

Our experiments are performed on the Cityscapes dataset [33], that contains 2975 videos for training, 500 for validation and 1525 for testing. As only the 20th frame of video contains annotations, and to be consistent with previous work, we aim to predict this frame in two settings. Short term predictions consist in predicting frame 20 using frames $8, 11, 14, 17$ and mid term, computing frames $14, 17, 20$ from 2, 5, 8, 11. The mid term prediction setting is therefore more challenging, as it aims to forecast a 0.5 seconds future. As in [6,7], our models are validated using the IoU SEG metric on the validation set, which corresponds to the mean Intersection over Union computed between the predictions and the segmentation obtained via Mask R-CNN. As Mask R-CNN is an object-based segmentation method, it only outputs segmentations for the 8 classes that correspond to moving object instances: person, rider, car, truck, bus, train, motorcycle, and bicycle. We also report results of the same copy and flow baseline. The copy approach simply provides the last input as future segmentation. The flow baseline is based on pixel warping using optical flow computed between the last two frames. *D2D* was trained using stochastic gradient descent with a momentum of 0.9, and a learning rate of 0.02.

The semantic segmentation accuracy is computed via the mean intersection over union with the ground truth. The instance segmentation accuracy is provided by computing the AP and AP-50. As our instance predictions are not associated with classifiers scores, we set the confidence equal to 1 everywhere. Mask R-CNN, *F2F* [7], and the optical flow baseline all produce a list of instance maps that may overlap with each other. As argued in [19], the AP measures favor this category of methods to the detriment of approaches that output a unique

	Short term (0.17 seconds)	Mid term (0.50 seconds)
	IoU	IoU
Oracle [6]	64.7	64.7
S2S [6]	55.3	40.8
Oracle [7]	73.3	73.3
Copy [7]	45.7	29.1
Flow [7]	58.8	41.4
S2S [7]	55.4	42.4
F2F [7]	**61.2**	41.2
D2D	56.0	**43.0**

Table 1: Short and mid term semantic segmentation of moving objects (8 classes) performance on the Cityscapes validation dataset.

answer at each spatial position. Since the former methods in fact eventually threshold their results at the confidence parameter 0.5 for visualization purposes, we compute AP and AP-50 on the segments formed by a non-overlapping segmentation map.

Specifically, for each method, we compute a superimposition of instance segments by filling a map with segments ranked by ascending confidence. In the AP and AP-50 computations, there is a step where segment proposals are matched with ground truth segments. For each proposal segment, if less than half of their pixels overlap with any object of the superimposed map, this segment is discarded in the evaluation. Then we compute AP and AP-50 scores on ground truth segments and remaining segments. We note the obtained scores "Non Overlapping AP": NO-AP and NO-AP-50. In the particular case of our $D2D$ results, AP and NO-AP are equivalent.

Our future semantic segmentation performance is reported in Table 1. While the $F2F$ and flow baseline results lead to high mean IoU in the short term, their performance are lower than $D2D$ in the mid term term setting. $D2D$ also slightly improves over the $S2S$ baseline that was state-of-the-art for future semantic segmentation prediction.

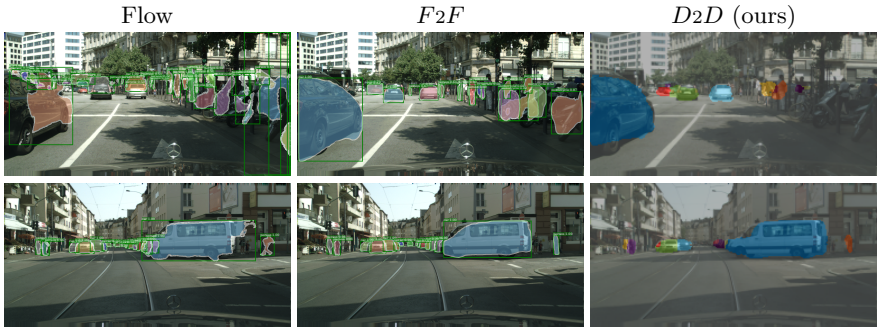

Fig. 5: Mid term future instance segmentation results. The flow baseline produces large distortions of objects subject to large displacements.

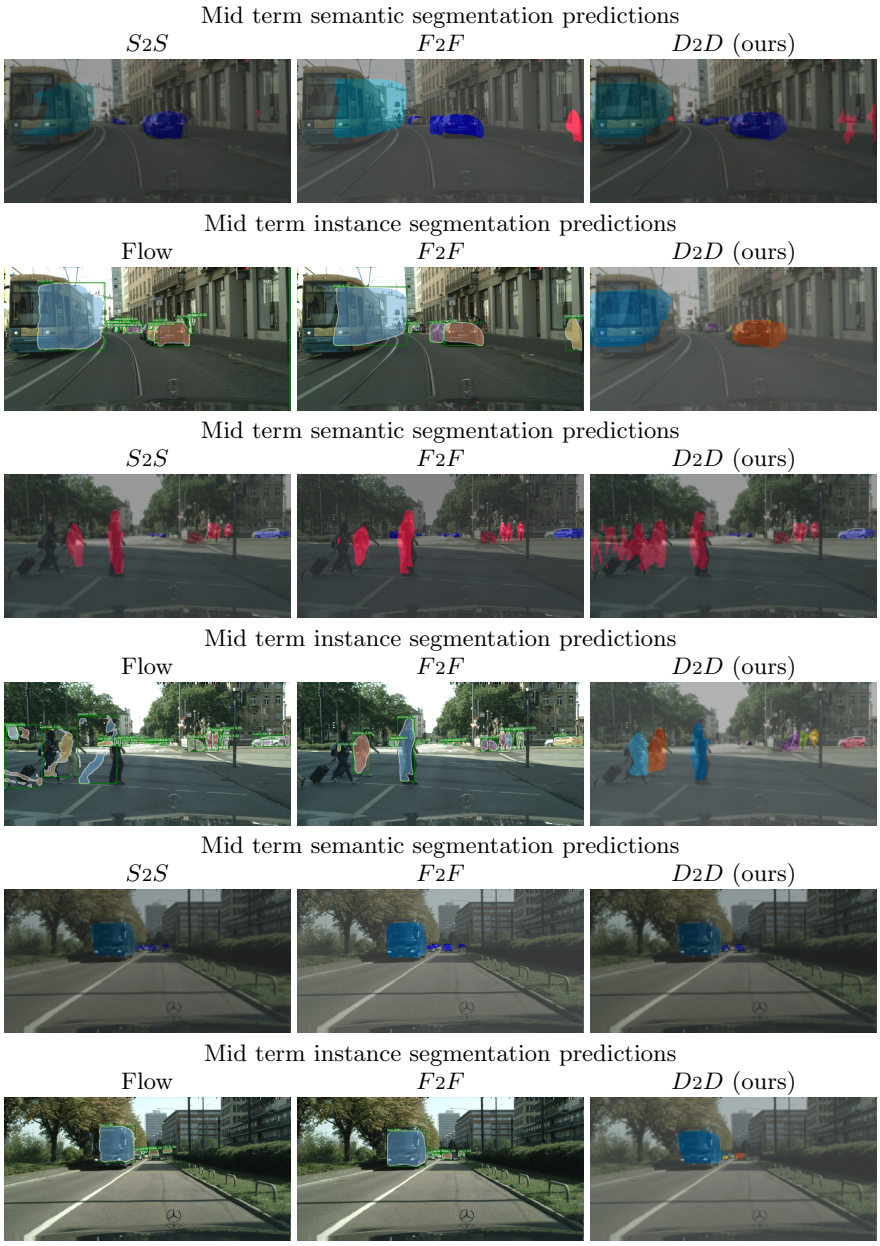

Fig. 6: Additional mid term comparative results. The flow baseline produces large distortions of small objects, in particular pedestrians. The $F2F$ predictions may

	Short term		Mid term	
	NO-AP-50	NO-AP	NO-AP-50	NO-AP
Mask R-CNN oracle	57.7	33.0	57.7	33.0
Copy last input	19.8	8.9	5.8	1.5
Optical flow baseline	30.8	14.1	9.5	3.6
F2F	**30.8**	**15.5**	**16.1**	**6.6**
D2D future centroids (oracle)	18.9	8.8	11.7	4.4
D2D pred. (seeds: minima)	14.2	6.5	7.1	2.9
D2D linear extrapolation	14.9	6.7	10.2	3.7

Table 2: Instance segmentation performance on the Cityscapes validation dataset, in terms of "Non-overlapping" AP measures. To avoid the bias of the standard AP and AP-50 measure in favor of proposal based methods, where several overlapping solutions are evaluated, we propose the "Non-overlapping" AP metrics, which consists in AP with a specific change in the matching step with ground truth segments. We first create a superimposed map for each proposal based methods (here the oracle, copy, F2F, optical flow baselines). For each proposal segment, if less than half of their pixels overlap with any object of the superimposed map, this segment is discarded in the evaluation. Then we compute AP and AP-50 scores on ground truth segments and remaining segments.

We compare our results with the $S2S$ baseline, optical flow baseline and the $F2F$ approach in Figures 6 and 5. We observe that our method is fairly accurate in a number of situations where $F2F$ and the Flow baseline meet difficulties for mid term predictions.

Table 2 provides quantitative results of instance segmentation accuracies of proposed methods. Our $D2D$ approach does not compare favorably with the other baselines for short term predictions. However, for mid term ones, it clearly outperforms the copy baseline, and performs slighly better than the flow baseline. We experiment using three different sets of seeds for model $D2D$: minima of the predictions $\hat{S}$, extrapolated object centroids, and centroids of the oracle future segmentation, to provide an upper bound for our method's performance. We observe that $F2F$ does lead to superior results, but at a much higher training cost. Learning in the pyramidal feature space of Mask R-CNN requires indeed to train and then finetune four networks, fixing each time an adequate learning rate. As summarized in Table 3, our approach is much lighter with less than 1M parameters, leads to superior semantic segmentation results, and comprises a built-in object tracking mechanism.

Figure 7 presents mid term segmentation results that illustrate the effectiveness of the proposed built-in tracking strategy of instances.

	Feature based *F2F*	Optical Flow baseline	Distance based *D2D* (ours)
Mid term sem. segm (IoU)	41.2	41.4	**43.0**
Mid term inst. segm (NO-AP-50)	**16.1**	9.5	10.2
Tracking included	no	**yes**	**yes**
Training time	6 days	-	**1 day**
Network size	65M	-	0.8 M
Training hyperparam. to tune	8	-	2
Inference time	**some sec.**	2 min	**some sec.**
Post-processing	threshold	hole filling, thresh.	optimization

Table 3: Comparative overview of future segmentation methods based on Mask R-CNN. Our *D2D* approach cumulates state-of-the-art semantic segmentation performance, inference and training speed, and temporally consistent results.

Our prediction of frame 14 Our prediction of frame 17 Our prediction of frame 20

Fig. 7: Mid term instance segmentation results produced by our *D2D* model. Most forecasted instances are consistent in a 0.5 seconds future.

5 Conclusion

We introduced a novel approach for predicting both future instance and semantic segmentation. Our distance map based encoding allows us to recover both information by a simple argmax or a graph-based optimization algorithm.

We improve in term of mean IoU over the state-of-the-art method for future semantic segmentation prediction while also allowing future instance prediction efficiently. While obtaining a lower performance in terms of instance segmentation performance compared to feature level prediction, we improve over a strong

optical flow baseline. Furthermore, relying on seeded segmentation allows us to incorporate tracking into our results and obtain an optimal solution.

Ultimately, we hope to employ our representation as a light, simple and effective building block to develop more sophisticated and better performing forecasting methods.

Acknowledgment. This work has been partially supported by the grant ANR-16-CE23-0006 "Deep in France" and LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01). We thank Piotr Dollárd and anonymous reviewers for their precious comments.

References

1. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv 1412.6604 (2014)
2. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMs. In: ICML. (2015)
3. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR. (2016)
4. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: NIPS. (2016)
5. Denton, E., Birodkar, V.: Unsupervised learning of disentangled representations from video. NIPS (2017)
6. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV. (2017)
7. Luc, P., Couprie, C., Verbeek, J., LeCun, Y.: Predictive learning in feature space for future instance segmentation. In: ECCV. (2018)
8. Oh, J., Guo, X., Lee, H., Lewis, R.L., Singh, S.P.: Action-conditional video prediction using deep networks in atari games. arXiv 1507.08750 (2015)
9. Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. In: ICLR. (2017)
10. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from static images using variational autoencoders. In: ECCV. (2016)
11. Denton, E., Fergus, R.: Stochastic video generation with a learned prior. In: ICML. (2018)
12. Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic variational video prediction. ICLR (2018)
13. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating the future by watching unlabeled video. In: CVPR. (2016)
14. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: ICCV. (2017)
15. Bhattacharyya, A., Fritz, M., Schiele, B.: Long-term on-board prediction of people in traffic scenes under uncertainty. CVPR (2018)
16. Jin, X., Xiao, H., Shen, X., Yang, J., Lin, Z., Chen, Y., Jie, Z., Feng, J., Yan, S.: Predicting scene parsing and motion dynamics in the future. NIPS (2017)
17. Romera-Paredes, B., Torr, P.: Recurrent instance segmentation. In: ECCV. (2016)
18. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR. (2017)

19. Arnab, A., Torr, P.H.S.: Pixelwise instance segmentation with a dynamically instantiated network. CVPR (2017)
20. Pinheiro, P., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV. (2016)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV. (2017)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. (2015)
23. Watanabe, T., Wolf, D.: Distance to center of mass encoding for instance segmentation. arXiv 1711.09060 (2017)
24. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. ICCV (2001)
25. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI **23** (2001)
26. Grady, L.: Random walks for image segmentation. PAMI **28(11)** (2006) 1768–1783
27. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. ICCV (2007)
28. Grady, L., Sinop, A.K.: Fast approximate random walker segmentation using eigenvector precomputation. CVPR (2008)
29. Allène, C., Audibert, J.Y., Couprie, M., Keriven, R.: Some links between extremum spanning forests, watersheds and min-cuts. Image and Vision Computing (2009)
30. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watershed: A unifying graph-based optimization framework. PAMI **33**(7) (2011) 1384–1399
31. Meijster, A., Roerdink, J.B.T.M., Hesselink, W.H. In: A General Algorithm for Computing Distance Transforms in Linear Time. Springer US (2000) 331–340
32. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. PAMI **13**(6) (1991) 583–598
33. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)