INSTANCE FLOW BASED ONLINE MULTIPLE OBJECT TRACKING

Sebastian Bullinger, Christoph Bodensteiner, Michael Arens

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation

ABSTRACT

We present a method to perform online Multiple Object Tracking (MOT) of known object categories in monocular video data. Current Tracking-by-Detection MOT approaches build on top of 2D bounding box detections. In contrast, we exploit state-of-the-art instance aware semantic segmentation techniques to compute 2D shape representations of target objects in each frame. We predict position and shape of segmented instances in subsequent frames by exploiting optical flow cues. We define an affinity matrix between instances of subsequent frames which reflects locality and visual similarity. The instance association is solved by applying the Hungarian method. We evaluate different configurations of our algorithm using the MOT 2D 2015 train dataset. The evaluation shows that our tracking approach is able to track objects with high relative motions. In addition, we provide results of our approach on the MOT 2D 2015 test set for comparison with previous works. We achieve a MOTA score of 32.1.

***Index Terms*—** Online Multiple Object Tracking, Instance Segmentation, Optical Flow

1. INTRODUCTION

1.1. Motivation

Tracking-by-Detection is one of the most popular approaches to tackle the problem of online Multiple Object Tracking. The pipeline consists of three stages. In the first stage, objects are independently detected in each frame. Typically, the detections are represented by a two-dimensional bounding box. In the second stage, the detections found in previous and subsequent frames are associated. One way to compute the associations of previous and subsequent objects is by determining a pairwise affinity value. The affinity value may reflect positional information and/or visual similarity. The association is typically solved by applying the Hungarian method to the affinity matrix. In the last stage, previously created tracklets are associated in order to fill the gap between missing detections and to handle occlusions.

One possibility to incorporate the position of objects in the corresponding affinity values is by predicting the object position from the current frame to subsequent frames. Typically, this is achieved by using a motion model, e.g. a Kalman Filter, or by exploiting visual cues. Motion based Tracking-by-Detection methods may struggle in scenarios, where camera and object move simultaneously. In this case, the perceived object motion is a superposition of object and camera motion. It is not always possible to describe a superposition of motions adequately using a single motion model. For example, consider the case of a car driving over a speed bump. Suddenly, the position of a person observed from the car experiences a vertical shift. In contrast, optical flow based Tracking-by-Detection does not require the definition of a motion model. Since current optical flow based methods use bounding box representations of the target objects they must deal with non-target-object surfaces contained in the bounding box. Otherwise, occlusions and background structures may influence the quality of the optical flow information.

With recently published ConvNets [1, 2] it is possible to segment the two-dimensional shape of instances of known object categories. In contrast to a simple bounding box, this representation has the advantage that it does not contain background structures or parts of other objects. We determine reliable associations of objects in subsequent frames by combining instance aware semantic segmentations and semi-dense optical flow cues. Our approach works online and uses only visual information for object association. Our flow based approach even handles objects with high relative motions, i.e. objects where the observed motion is a superposition of camera and object motion.

1.2. Contribution

To the best of our knowledge, we present the first instance aware semantic segmentation based Multiple Object Tracking approach. The segmentations allow us to track the two-dimensional shape of objects in subsequent frames on pixel level. We provide a detailed description of the basic elements of our tracking pipeline. We analyze the effectiveness of our method by providing results with different parameter configurations, e.g. different optical flow algorithms, on the MOT 2D 2015 train dataset. In addition, we compare our method with SORT on the MOT 2D 2015 test dataset using detections extracted from instance segmentations. SORT is a open source online MOT tracker, which has shown competitive results using Faster RNN detections.

1.3. Related Work

Many state-of-the-art MOT approaches like [3, 4, 5, 6] follow the Detection-by-Tracking methodology. [3, 4] incorporate optical flow algorithms [7, 8] to tackle the problem of object association. Yu et al. [6] achieve state-of-the-art results by training a person specific detector, i.e. they fine-tune the VGG ConvNet [9] using several additional training datasets. For online tracking they follow the standard Detection-by-Tracking stages. They use a Kalman filter [10] for motion prediction and the Hungarian method [11] to compute associations. Lee et al. [5] present a multi-object tracker based on a Bayesian Filtering framework. Objects are detected using an ensemble of motion detection and object detection. Xiang et al. [4] use a Markov Decision Process to perform MOT. Bounding box predictions are computed by measuring the stability of the corresponding optical flow. The quality of the optical flow result is affected by background structures and surfaces of other objects in the bounding box. Choi [3] determines a set of salient points in the input image and computes the corresponding trajectories using the algorithm of Farneback [8]. The relative motion of these trajectories w.r.t. a pair of bounding boxes is used to determine an affinity score. Schikora et al. [12] use optical flow cues to detect moving objects and to track these objects using a particle filter. Milan et al. [13] use a multi-label conditional random field to assign super pixels to object instances represented by bounding boxes. The superpixels are determined by using color information and optical flow cues. The object associations are computed offline.

2. INSTANCE FLOW BASED TRACKING

2.1. Terminology

Let $\boldsymbol{I}_t$ denote the t-th image of an ordered sequence with height h and width w. Furthermore, let $\boldsymbol{I}_t(x, y)$ denote the color of pixel position (x,y) in $\boldsymbol{I}_t$ with $(x, y) \in \{1, \cdots, w\} \times \{1, \cdots, h\}$.

Instance aware segmentation systems, like [1, 2], predict for each pixel position in an input image $\boldsymbol{I}_t$ a semantic category label c and a corresponding instance index i according to equation (1)

$$\boldsymbol{S}_t(x, y) = (c, i), \tag{1}$$

where $\boldsymbol{S}_t$ denotes the instance semantic segmentation of image $\boldsymbol{I}_t$.

Optical Flow or semi-dense matching methods like [15, 14] use a pair of subsequent images, denoted as $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+o}$, to compute a two-dimensional pixel offset field for pixel positions (x,y). Here, o is the offset of the frame indices in the image sequence. An optical flow algorithm computes the optical flow of an image pair $\boldsymbol{F}_{t\to t+o}(x, y)$ for pixel positions (x, y) such that the similarity in equation (2) is maximized.

$$\begin{aligned}\boldsymbol{I}_t(x, y) \simeq \boldsymbol{I}_{t+o}(x + \boldsymbol{F}_{t\to t+o}(x, y)[0],\\ y + \boldsymbol{F}_{t\to t+o}(x, y)[1])\end{aligned} \tag{2}$$

Some optical flow algorithms estimate the optical flow only for a subset of pixels. In this case, there are pixel positions where no optical flow information is available. We will denote the set of pixel positions with valid flow information at time t with $F_t^{(v)}$.

2.2. Prediction of Segmentation Instances

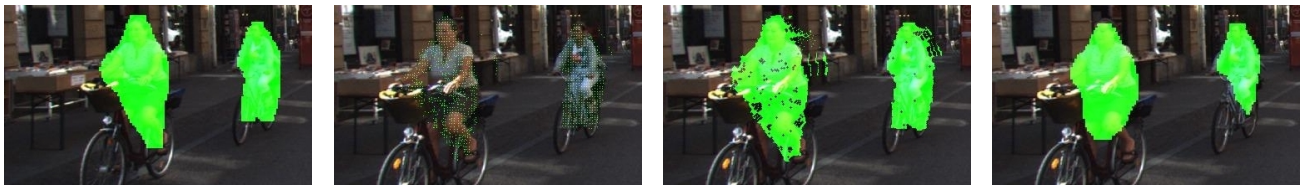
We use the ConvNet presented in [1] to compute the instance segmentation $\boldsymbol{S}_t$ for image $\boldsymbol{I}_t$. For an instance with index i of the target category c we use $\boldsymbol{S}_t$ to extract the corresponding set of occupied pixel positions $S_{t,i}$. More formally, we compute $S_{t,i}$ according to equation (3)

$$\begin{aligned}S_{t,i} = \{(x, y) | (x, y) \in \{1, \cdots, w\} \times \{1, \cdots, h\}\\ \wedge \boldsymbol{S}_t(x, y) = (c, i)\}.\end{aligned} \tag{3}$$

Fig. 1a and 1d show some instance segmentation examples. For subsequent image pairs $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+o}$ we compute the optical flow $\boldsymbol{F}_{t\to t+o}$ applying one of the algorithms presented in [8, 15, 14]. This allows us to predict the pixel positions (x,y) contained in $S_{t,i}$ to the next image $\boldsymbol{I}_{t+o}$. Fig. 1b shows the prediction of two instance segmentations using [14]. We denote the set of predicted pixel positions as $P_{t\to t+o,i}$ and compute it according to equation (4)

$$\begin{aligned}P_{t\to t+o,i} = \{(x_p, y_p) | (x_p, y_p) = \boldsymbol{F}_{t\to t+o}(x, y)\\ \wedge (x, y) \in F_{t,i}^{(v)}\},\end{aligned} \tag{4}$$

where $F_{t,i}^{(v)} = S_{t,i} \cap F_t^{(v)}$ is the set of valid optical flow positions of instance i. If the optical flow algorithm does not provide flow information for each pixel we interpolate the optical flow at positions where no flow information is available. This allows us to compute dense predictions of instance segmentations. We interpolate the optical flow vectors for each instance, separately. To avoid the influence of the optical flow of background structures we use only optical flow vectors of the corresponding instance, i.e. we consider only vectors at pixel positions $F_{t,i}^{(v)}$. We use a linear interpolation of points inside the convex hull of $F_{t,i}^{(v)}$. The optical flow of points lying outside the convex hull is interpolated by using the corresponding nearest neighbor. The interpolation of optical flow vectors pointing in opposite directions generates holes and overlaps in the predicted segmentation instance. Consider the following one-dimensional example with four adjacent pixel positions and two optical flow values at the first and fourth position: $[-3, \_, \_, 3]$. The linear interpolation of the missing optical flow values yields $[-3, -1, 1, 3]$. Shifting the second and the



(a) Instance Segmentations in Image I_t

(b) Predicted Segmentation Instances in Image $I_t + 1$

(c) Closed Interpolation of the Predicted Instances in Image $I_t + 1$

(d) Detected Instances in Image $I_t + 1$

Fig. 1. Instance Flow Prediction using CPM [14]. Instance flow prediction is best visualized using objects with high velocity, like bicyclists. The offset of corresponding detections is clearly visible. The figure also highlights the importance of accurate segmentations. Otherwise we observe a distortion of the predicted instance segmentation. The figure is best shown in color.

third point according to the corresponding optical flow values, i.e. -1 and 1, moves the second pixel to the left as well as the third pixel to the right and leaves a hole in the corresponding segmentation mask. We close these holes by performing a morphological closing operation. An example of a closed interpolation of a predicted segmentation is shown in Fig. 1c.

2.3. Affinity of Objects in Subsequent Frames

To associate objects visible in image I_t with objects in frame I_{t+o} we compute an affinity score between the corresponding instance segmentations. We define the similarity of an object with index i in frame I_t and object with index j in frame I_{t+o} as the overlap of the intersection of the predicted pixel set $P_{t\rightarrow t+o,i}$ and the pixel set of instance segmentation $S_{t+o,j}$. Note that the number of segmentation instances and the order of the corresponding indices may differ. This formulation of the affinity measure reflects locality and visual similarity. Let $O_{i,j}$ denote the overlap of the prediction $P_{t\rightarrow t+o,i}$ and $S_{t+o,j}$, i.e. $O_{i,j} = \#(P_{t\rightarrow t+o,i} \cap S_{t+o,j})$. Furthermore, let n_i and n_j denote the number of segmentation instances in image I_t and I_{t+o}, respectively. We build an affinity matrix $\boldsymbol{A}$ using these pairwise overlaps according to equation (5)

$$\boldsymbol{A}_{t\rightarrow t+o} = \begin{bmatrix} O_{1,1} & \cdots & O_{1,j} & \cdots & O_{1,n_j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ O_{i,1} & \cdots & O_{i,j} & \cdots & O_{i,n_j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ O_{n_i,1} & \cdots & O_{n_i,j} & \cdots & O_{n_i,n_j} \end{bmatrix}. \quad (5)$$

2.4. Online Multiple Object Tracking

The state of the presented instance flow tracker T_t at time t consists of a set segmentation instances $S_{t,k}$ with unique identifiers $id_{t,k}$ and a counter for the number of missed detections $m_{t,k}$, i.e. $T_t = \{(S_{t,k}, id_{t,k}, m_{t,k}) | k \in \{1, \cdots, n_t\}\}$, where n_t is the number of tracks at time t. We initialize this state with the segmentation instances in the first frame (if any). For subsequent frames the tracker state segmentations $S_{t,k}$ are predicted using equation (4). Let $\boldsymbol{I}_t$ denote the previous image and $\boldsymbol{I}_{t+1}$ the current image. In order to solve the association of segmentation instances in the tracker state $S_{t,k}$ and segmentations instances $S_{t+1,j}$ found in current image we use the steps described in section 2.2 and 2.3 to compute the affinity matrix $\boldsymbol{A}_{t\rightarrow t+1}$. We apply the Hungarian Method [11] on $\boldsymbol{A}_{t\rightarrow t+1}$, which results in a set of matching index pairs P_t. We ensure the validity of an index pair $(k, j) \in P$ by verifying that $\boldsymbol{A}_{t\rightarrow t+1}(k, j) > 0$.

For all valid index pairs $(k, j) \in P_t$ we update the segmentation instances maintained by the tracker, i.e. we set $S_{t,k} = S_{t+1,j}$, but keep the unique tracklet identifier $id_{t,k}$. We add all non-matching segmentation instances found in image $\boldsymbol{I}_{t+1}$ with a new unique identifier to the set of segmentation instances maintained by the tracker. In addition, we remove all non-matching segmentation instances contained in the tracker state, if $m_{t,k} > md$, where md is the number of allowed missing detections. Otherwise, we replace the instance segmentation with a dense prediction of the corresponding pixel positions as described in section 2.2.

3. EVALUATION

We evaluate our Instance Flow based online Multiple Object Tracking approach on the popular MOT dataset [16] using instance aware semantic segmentations computed by [1] and the optical flow / matching algorithms presented in [8], [15] and [14]. We also analyze the effect of varying the value of md. We compare our approach with SORT [17], an open source online MOT tracker, which showed competitive results using Faster RNN [18] detections. SORT follows the Tracking-by-Detection pipeline, i.e. Bounding Box detections, a Kalman filter for motion prediction and the Hungarian method for object association. The performance of Tracking-By-Detection approaches is strongly dependent on the quality of detections. By applying SORT on detections derived from instance segmentations we compare the tracking performance without the influence of different detector performances.

We use the following combinations in our evaluation: *Faster-RNN +SORT* combines Faster RNN bounding box detections [18] and SORT [17] tracking. *MNC+SORT* integrates detections extracted from MNC instance segmentations [1] instead. *MNC+CPM*, *MNC+DeepMatch* and *MNC+PolyExp* use the

Table 1. MOT 2D 2015 Benchmark Test Set Evaluation.

Method	md	**MOTA** ↑	MOTP ↑	FAF ↓	MT ↑	ML ↓	FP ↓	FN ↓	ID sw ↓	Frag ↓
FasterRNN+SORT	-	**33.4**	72.1	1.3	11.7%	30.9%	7,318	32,615	1,001	1,764
MNC+SORT	-	**27.5**	70.5	0.5	7.5%	50.9%	2,972	40,924	661	1,292
MNC+CPM (ours)	0	**30.6**	71.3	0.8	10.5%	34.0%	4,863	35,325	2,459	2,953
MNC+CPM (ours)	1	**32.1**	70.9	1.1	13.2%	30.1%	6,551	33,473	1,687	2,471

Table 2. MOT 2015 Benchmark KITTI-13 Evaluation.

Method	md	Rcll	Prcn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	**MOTA**	MOTP	MOTAL
MNC+SORT	-	18.8	77.3	0.12	42	0	14	28	42	619	3	6	**12.9**	65.2	13.2
MNC+CPM (ours)	0	38.7	69.7	0.38	42	0	32	10	128	467	25	38	**18.6**	67.2	21.7
MNC+CPM (ours)	1	43.8	65.7	0.51	42	4	32	6	174	428	14	30	**19.2**	66.7	20.8
MNC+DeepMatch (ours)	0	38.7	69.7	0.38	42	0	32	10	128	467	25	38	**18.6**	67.2	21.7
MNC+DeepMatch (ours)	1	43.8	63.8	0.55	42	3	31	8	188	431	14	29	**16.9**	66.8	18.6
MNC+PolyExp (ours)	0	38.7	69.7	0.38	42	0	32	10	128	467	39	40	**16.8**	67.3	21.7
MNC+PolyExp (ours)	1	42.7	61.2	0.61	42	3	31	8	206	437	30	33	**11.7**	66.8	15.4

MNC instance segmentations of [1] as well as the optical flow of [14], the deep matching algorithm of [15] and the optical flow of [8], respectively.

MNC+CPM, *MNC+DeepMatch* and *MNC+PolyExp* achieve similar results on the MOT 2015 training set. A reason for this is the slow motion of camera and pedestrians in most MOT 2015 sequences. In these cases, the quality of object associations is mainly dependent on the segmentation quality. The results of *MNC+CPM* for the test set is shown in table 1. The biggest difference of the evaluated algorithms in the train dataset is observed in the KITTI-13 sequence, which is the only video captured from a driving platform. In this case, the positions of the objects in image coordinates are strongly affected by the motion of the vehicle, i.e object positions show remarkable shifts between subsequent images. The corresponding results are shown in table 2. In terms of MOTA *MNC+CPM* (with $md = 1$) outperforms *MNC+DeepMatch* as well as *MNC+PolyExp*. This shows the importance of the quality, e.g. density and reliability, of the selected optical flow / matching algorithm. *MNC+DeepMatch* is very sparse and *MNC+PolyExp* and can not handle big object shifts as shown in Fig. 2. All optical flow approaches show a higher MOTA score than *MNC+SORT*. This demonstrates the strength of optical flow based approaches in videos with high relative motions of objects. It also shows the difficulty to describe a superposition of motions with a single motion model. We observe, that the number of id switches (IDs) of *MNC+SORT* is significantly lower than the ones of the evaluated optical flow based approaches. This confirms our impression that the used semantic instance segmentation [1] is unstable. However, we are able to decrease the number of id switches by using dense predictions as instance segmentations in the subsequent frame (e.g. $md = 1$).

(a) Prediction using CPM. (b) Prediction using PolyExp.

Fig. 2. Importance of the quality of the optical flow algorithm. The prediction using PolyExp is not correctly shifted.

4. CONCLUSION

We presented an online Multiple Object Tracking approach exploiting semantic instance segmentations and optical flow cues. The algorithm is able to track the two-dimensional shape of objects in subsequent frames. We evaluated our approach in the domain of pedestrians. The algorithm shows its benefits while tracking objects with high relative motions. Currently, the tracker only supports basic tracking functionality. In future work, we want to combine our approach with tracker management algorithms to increase its performance, for example by handling occlusions. We demonstrated that semantic instance segmentations are an interesting alternative to conventional bounding box detections.

5. REFERENCES

[1] Jifeng Dai, Kaiming He, and Jian Sun, "Instance-aware semantic segmentation via multi-task network cascades," *CoRR*, vol. abs/1512.04412, 2015.

[2] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei, "Fully convolutional instance-aware semantic segmentation," *CoRR*, vol. abs/1611.07709, 2016.

[3] Wongun Choi, "Near-online multi-target tracking with aggregated local flow descriptor," *CoRR*, vol. abs/1504.02340, 2015.

[4] Yu Xiang, Alexandre Alahi, and Silvio Savarese, "Learning to track: Online multi-object tracking by decision making," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 4705–4713.

[5] Byungjae Lee, Enkhbayar Erdenee, SongGuo Jin, and Phill-Kyu Rhee, "Multi-class multi-object tracking using changing point detection," *CoRR*, vol. abs/1608.08434, 2016.

[6] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan, "POI: multiple object tracking with high performance detection and appearance feature," *CoRR*, vol. abs/1610.06136, 2016.

[7] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, San Francisco, CA, USA, 1981, IJCAI'81, pp. 674–679, Morgan Kaufmann Publishers Inc.

[8] Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, Berlin, Heidelberg, 2003, SCIA'03, pp. 363–370, Springer-Verlag.

[9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[10] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[11] Harold W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[12] M. Schikora, W. Koch, and D. Cremers, "Multi-object tracking via high accuracy optical flowand finite set statistics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 1409–1412.

[13] Anton Milan, Laura Leal-Taix, Konrad Schindler, and Ian Reid, "Joint tracking and segmentation of multiple targets," in *CVPR*, 2015.

[14] Yinlin Hu, Rui Song, and Yunsong Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid, "Deepmatching: Hierarchical deformable dense matching," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323, 2016.

[16] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015.

[17] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft, "Simple online and realtime tracking," *CoRR*, vol. abs/1602.00763, 2016.

[18] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.