Deep Network Flow for Multi-Object Tracking cvpr17

2019/01/22 12:08 PM

Network flow graph construction:

- each detection is are presented by to nodes
- The 2nd node of each detection is connected to the 1st node of every other detection which belongs to a later frame and lies within a finite temporal window
- nonconsecutive frame detections are connected to deal with occlusions and false negatives
- the source ignored is connected to the 1st node of each detection and the 2nd node fees detection is connected to the sink and node to represent the start and end of trajectories;
- Edges between boxes that spatially far apart are also removed under the smooth motion assumption

2019/01/26 1:50 PM

The link between the 2 nodes representing the same detection are used in the formulation of the flow constraint where the total flow coming in would be equal to the flow going across this link which in turn would also be equal to the total flow going out;

the flow conservation constraints are represented by a single linear constraint in the LP formulation;

The box constraint obtained by relaxing the integer constraint on the flow variables is actually nondifferentiable and needs to be approximated using something called log barriers

The equality constraint representing flow conservation is also removed by the performing a change of basis so that the lower-level problem actually ends up becoming an unconstrained and differentiable optimization problem

The binary detection links are divided into 3 different types some of which are in turn divided into multiple types in a hierarchical fashion, all to incorporate some intuitions which in a sense shifts the engineering from the costs to the loss functions

The loss function itself is a weighted function of the unity and binary link costs where the weights again are chosen to incorporate even more intuitions;

the sequence is processed in a piecewise fashion where the LP problem is solved for all frames in a temporal window and then the window is moved by some number of frames which is less than the window size so that some overlap remains;

Trajectories from nearby solutions are then matched using bipartite matching where the cost between 2 trajectories is inversely proportional to the number of detections they share

In the 1st experiments, very simple neural networks are used – a single layer model with 64 neurons and a 2 layer model with 32 neurons in each
input to the neural networks is not the input image but a handcrafted feature involving the difference between the bounding boxes, their detection confidences, normalized time difference and the intersection over union;

In the 2nd experiments, motion and appearance features are also incorporated

Not much improvement in performance was demonstrated over using just the bounding box features versus including the motion features as well;

Appearance features are constructed by extracting the patch underneath the bounding box, resizing it to a fixed size, using ResNet-50 to extract its features and concatenating everything as input to the Siamese network;
some more feature concatenation is then done using the output of the MLP which is all not quite clear;

Incorporating appearance information shows more significant improvement over using just box information but the results are only given for MOT15 data set while the others are given for kitti data set which might indicate that appearance features simply did not work for the latter;

In spite of all of the sophistication, they did not manage to improve upon the recall of the detector which highlights the importance of the end to end training of the detector;