Summary of Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image

Qing Cheng

November 2017

1 Motivation

Meaning of 2D and 3D vehicle analysis from monocular images in the context of self-driving cars

* currently most cars are equipped with a single camera
* it is essential for autonomously driving vehicles to understand the traffic and predict critical situations based on the information extracted from the image of the scene.
    - For the recovery of speed and direction of the surrounding cars, 3D vehicle localization and orientation jointly used with temporal description are necessary.
    - For proper traffic understanding it is important to describe surrounding vehicles in a fine way
    - For interpretation of the overall scene the characterization of the visibility of vehicle parts needs also to be obtained.

2 contribution

1. encode 3D vehicle information using characteristic points of vehicles.
    * The underlying idea: vehicles are rigid objects with well known geometry so that that 3D vehicle information can be recovered using monocular images.
    * It can find all the parts of vehicles,including the invisible by using regression instead of using a part detector to find the hidden parts.

    * The main idea of the approach is to recover the projection of these 3D points (2D shape) in the input image for each detected vehicle. Then, the best corresponding 3D model for each detection box is chosen. 2D/3D matching is performed between 2D shapes and selected 3D shapes to recover vehicle orientation and 3D location.

2. introduction of the Deep Coarse-to-fine Many-Task CNN called Deep MANTA.

    * Output: accurate 2D vehicle bounding boxes, 2D shapes, part visibility and 3D vehicle templates.
    * Originality
        - iteratively refine coarse 2D bounding boxes to provide accurate scored 2D detections.
        - the same feature vector can be used to predict many tasks at the same time: region proposal, detection, 2D box regression, part localization, part visibility and 3D template prediction

3. training dataset

    * we propose a semi-automatic annotation process using 3D models to generate labels (geometry information, visibility, etc) on real images for the Deep MANTA training.

3 Deep MANTA Model

3.1 Function

Here we propose an approach that, given a single image, provides accurate vehicle detections, vehicle part localization, vehicle part visibility, fine orientation, 3D localization and 3D template (3D dimension).

3.2 Datasets

1. **3D models:** each model corresponds to one type of vehicle (Sedan, SUV, etc). For each 3D model m, we annotate N vertices (called 3D parts). These parts correspond to relevant vehicle regions.

2. **3D shape aligned in canonical view:** $\bar{S}_m^{3d} = (p_1, p_2, .., p_N)$ with $p_k = (x_k, y_k, z_k)$ corresponding to the 3D coordinate of the k^{th} part.

3. **3D template (*i.e* 3D dimension) associated to the 3D model:** $\bar{t}_m^{3D} = (w_m, h_m, l_m)$ where w_m, h_m, l_m are the width, the height and the length of the 3D model respectively.

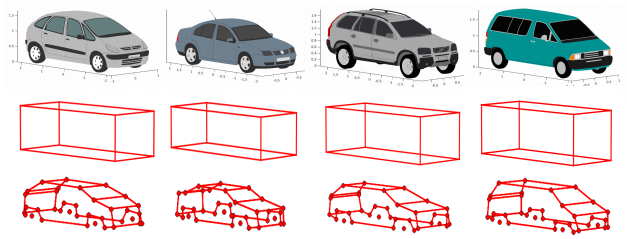

Figure 1: Some examples from the 3D template and 3D shape dataset. Each 3D model m (first line) is associated to a 3D template $\bar{t}_m^{3d}$ (second line) and a 3D shape $\bar{S}_m^{3d}$ (third line). The 3D shape corresponds to manually annotated vertices.

4. **2D/3D vehicle model:** $(B, B^{3d}, S, S^{3d}, V)$, shown in figure 2 where,

   * $B = (c_x, c_y, w, h)$ is the 2D vehicle bounding box in the image where (c_x, c_y) is the center and (w, h) represents the width and the height respectively.
   * $B^{3d} = (c_x, c_y, c_z, \theta, t)$ is the 3D bounding box characterized by its 3D center (c_x, c_y, c_z), its orientation θ and its 3D template $t = (w, h, l)$ corresponding to its 3D real size.
   * $S = \{q_k = (u_k, v_k)\}_{k \in \{1,..,N\}}$ is the vehicle 2D part coordinates in the image.
   * $S^{3d} = \{p_k = (x_k, y_k, z_k)\}_{k \in \{1,..,N\}}$ is the vehicle 3D part coordinates in the 3D real word coordinate system.
   * $V = \{v_k\}_{k \in \{1,..,N\}}$ is the part visibility vector where v_k denotes the visibility class of the k^{th} part.There are four classes: visible, occluded, self-occluded, and truncated (out of the image).

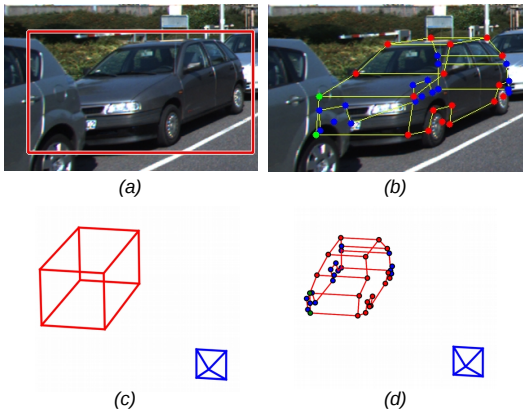

Figure 2: Example of one 2D/3D vehicle model. *(a)* the bounding box B, *(b)* 2D part coordinates S and part visibility V: visible parts (red), occluded parts (green) and self-occluded parts (blue). *(c)* the 3D bounding box B^{3d} and *(d)* the associated 3D shape S^{3d}.

*

-

3.3 Deep MANTA Network Architecture

Our system has two main steps.

1. First, the input image is passed through the Deep MANTA network that outputs 2D scored bounding boxes, associated vehicle geometry (vehicle part coordinates, 3D template similarity) and part visibility properties. Detail in section 3.3.1

2. The second step is the inference which uses Deep MANTA outputs and a 3D vehicle dataset to recover 3D orientations and locations. Detail in section 3.3.2

3.3.1 Deep MANTA Network

As illustrated in Figure 3 Phase 1, the Network takes the images as an input and outputs a final bounding box set $B_3 = \{B_{i,3}\}_{i \in \{1,..,K\}}$,as well as the MANTA network also returns all 2D vehicle part coordinates S_i, part visibility V_i and 3D template similarity T_i associated with each bounding box $B_{i,3}$.

To get the final bounding box set $B_3 = \{B_{i,3}\}_{i \in \{1,..,K\}}$, the paper applies a coarse-to-fine forward architecture. Step 1: Given an entire input image, the network (Conv layers RPN) returns a first set of K object proposals $B_1 = \{B_{i,1}\}_{i \in \{1,..,K\}}$. Step 2: The outputs from Step 1 are then extracted from

a feature map, pooled to a fixed size using ROI Pooling, and then forwarded to the second-level network (sharing some weights with the first level) with the offset transformations to get the second set of K objects $B_2 = \{B_{i,2}\}_{i \in \{1,..,K\}}$. Step 3: repeat the step 2 to get the final set of K objects $B_3 = \{B_{i,3}\}_{i \in \{1,..,K\}}$. Afterwards, non-maximum suppression is applied to remove the redundant detections to get a new set.

The template similarity vector T_i is defined as $T_i = \{r_m\}_{m \in \{1,..,M\}}$. $r_m = (r_x, r_y, r_z)$ corresponds to the three scaling factors to apply on the 3D template $\bar{t}_m^{3d}$ to fit the real 3D template of the detected vehicle i.

3.3.2 Deep MANTA Inference

As illustrated in Figure 3 Phase 2, the Inference takes the Deep MANTA network outputs, the 3D shape dataset $\{\bar{S}_m^{3d}\}_{m \in \{1,..,M\}}$ and the 3D template dataset $\{\bar{t}_m^{3d}\}_{m \in \{1,..,M\}}$ as inputs and outputs the 3D bounding box B_j^{3d} and the 3D part coordinates S_j^{3d}.

This phase consists of two steps. Step 1: use the template similarity $T_j = \{r_m\}_{m \in \{1,..,M\}}$ returned by the network to find the closest 3D template $c \in \{1, .., M\}$ in the 3D template dataset $\{\bar{t}_m^{3d}\}_{m \in \{1,..,M\}}$. This is done by applying the scaling transformation r_m to each 3D template to get a new 3D template set $\{t_m^{3d}\}_{m \in \{1,..,M\}}$ and then choosing the one that minimize the distance between t_m^{3d} and $\bar{t}_m^{3d}$. Step 2: 3D shape $\bar{S}_c^{3d}$ is rescaled to fit the 3D template $t_j = t_c^{3d}$. Then, a pose estimation algorithm is performed to match the rescaled 3D shape $\bar{S}_c^{3d}$ with the 2D shape S_j using a standard 2D/3D matching to generate the 3D bounding box B_j^{3d} and the 3D part coordinates S_j^{3d}.

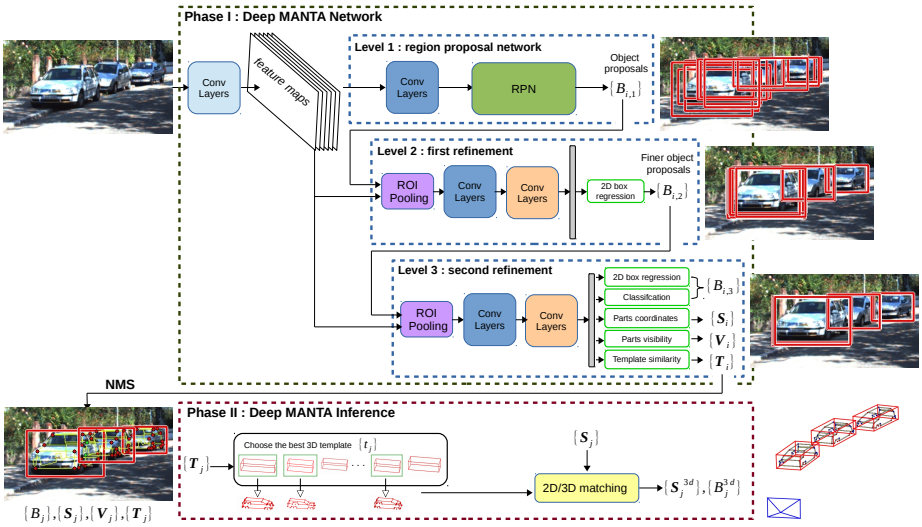

Figure 3: Overview of the Deep MANTA approach. The entire input image is forwarded inside the Deep MANTA network. Conv layers with the same color share the same weights. Moreover, these three convolutional blocks correspond to the split of existing CNN architecture. The network provides object proposals $\{B_{i,1}\}$ which are iteratively refined ($\{B_{i,2}\}$ and then the final detection set $\{B_{i,3}\}$). 2D part coordinates $\{S_i\}$, part visibility $\{V_i\}$ and template similarity $\{T_i\}$ are associated to the final set of detected vehicle $\{B_{i,3}\}$. A non-maximum suppression (NMS) is then performed. It removes redundant detections and provides the new set $\{B_j, S_j, V_j, T_j\}$. Using these outputs, the inference step allows to choose the best corresponding 3D template using template similarity T_j and then performs 2D/3D pose computation using the associated 3D shape.

4 Training - The Loss Functions

This section defines all the tasks of the MANTA network and the associated loss functions. In the following, each object proposal at each level of refinement l, is indexed by i and it is represented by its box $B_{i,l} = (c_{x_{i,l}}, c_{y_{i,l}}, w_{i,l}, h_{i,l})$. The closest ground-truth vehicle box B to $B_{i,l}$ is selected. Associated ground-truth parts S, ground-truth visibility V and ground-truth template t are also selected. We denote the standard log softmax loss as P and the robust SmoothL1 loss defined in [1] as R.

RPN loss $\mathcal{L}_{rpn}$: defined in [2]

Detection loss $\mathcal{L}_{det}$ is the detection loss function focusing on discriminating vehicle and background bounding box as well as regressing bounding boxes.

The object proposal i at the refinement level l is assigned to a class label $C_{i,l}$, 1 for a vehicle and 0 otherwise. The classification criteria is the overlap

between the box $B_{i,l}$ and the ground-truth box B. The predicted class returned by Deep MANTA network for the proposal is $C^*_{i,l}$. A target box regression vector $\Delta_{i,l} = (\delta_x, \delta_y, \delta_w, \delta_h)$ is also defined as follows:

$$\delta_x = (c_{x_{i,l}} - c_x)/w \qquad \delta_w = log(w_{i,l}/w)$$
$$\delta_y = (c_{y_{i,l}} - c_y)/h \qquad \delta_h = log(h_{i,l}/h)$$

The predicted regression vector returned by Deep MANTA network is $\Delta^*_{i,l}$. The detection loss function is defined by:

$$\mathcal{L}^l_{det}(i) = \lambda_{cls} P(C^*_{i,l}, C_{i,l}) + \lambda_{reg} C_{i,l} R(\Delta^*_{i,l} - \Delta_{i,l})$$

with λ_{cls} and λ_{reg} the regularization parameters of box classification and box regression respectively.

Part loss $\mathcal{L}_{parts}$ is the loss corresponding to vehicle part localization.

Using the ground-truth parts $S = (q_1, .., q_N)$ and the box $B_{i,l}$ associated to the object proposal i at level l, normalized vehicle parts $S_{i,l} = (\bar{q}_1, .., \bar{q}_N)$ are computed as follows:

$$\bar{q}_k = (\frac{u_k - c_{x_{i,l}}}{w_{i,l}}, \frac{v_k - c_{y_{i,l}}}{h_{i,l}}).$$

The predicted normalized parts are $S^*_{i,l}$. The part loss function is defined as:

$$\mathcal{L}^l_{parts}(i) = \lambda_{parts} C_{i,l} R(S^*_{i,l} - S_{i,l})$$

with λ_{parts} the regularization parameter of part loss.

Visibility loss $\mathcal{L}_{vis}$ is the loss related to part visibility.

This loss is only optimized on the final level of refinement $l = 3$. The ground-truth visibility vector $V_i = V$ is assigned to the object proposal i. The predicted visibility vector is V^*_i. The visibility loss function is defined as:

$$\mathcal{L}_{vis}(i) = \lambda_{vis} C_{i,3} P(V^*_i, V_i)$$

with λ_{vis} the regularization parameter of visibility loss.

Template similarity loss $\mathcal{L}_{temp}$ is the loss related to template similarity

This loss is only optimized on the final level of refinement $l = 3$. Instead of directly optimizing the three dimensions of the 3D template t, we encode it as a vector T using the 3D template dataset. For training, the *log* function is applied to each element of T for better normalization (similarity values are thus in $[-1, 1]$). The ground-truth template similarity vector vector $T_i = T$ is assigned to the object proposal i. The predicted template similarity vector is T^*_i. The template similarity loss function is defined as:

$$\mathcal{L}_{temp}(i) = \lambda_{temp} C_{i,3} R(T^*_i - T_i)$$

with λ_{temp} the regularization parameter of template similarity loss.

The joint loss L We use the Faster-RCNN framework [2] based on RPN to learn the end-to-end MANTA model. Given an input image, the network joint optimization minimizes the global function:

$$\mathcal{L} = \mathcal{L}^1 + \mathcal{L}^2 + \mathcal{L}^3$$

with

$$\mathcal{L}^1 = \mathcal{L}_{rpn},$$

$$\mathcal{L}^2 = \sum_i \mathcal{L}^2_{det}(i) + \mathcal{L}^2_{parts}(i),$$

$$\mathcal{L}^3 = \sum_i \mathcal{L}^3_{det}(i) + \mathcal{L}^3_{parts}(i) + \mathcal{L}_{vis}(i) + \mathcal{L}_{temp}(i),$$

where i is the index of a proposal object. These three losses correspond to the three levels of refinement of the Deep MANTA architecture: finer is the level, bigger is the amount of information learned.

5 Experiment

6 Conclusion

"I always thought something was fundamentally wrong with the universe" [**?**]

References

[1] R. Girshick. *Fast r-cnn.* ICCV, 2015.

[2] He K. Girshick R. B. Ren, S. and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks.* NIPS, 2015.