

# Continuous Models for Scene and Traffic Participant Interactions in Road Scene Understanding.

Vikas Dhiman  
vdhiman@nec-labs.com

Manmohan Chandraker  
manu@nec-labs.com

## Abstract

*The long term goal of this project is to identify and represent commonly occurring potentially dangerous road situations where driver can be warned about the situations. Such kind of driver assistance requires understanding the state of various road scene elements. We broadly categorize the elements in the scene as traffic participants (TP) and as other "scene elements" (like lanes, intersections etc).*

## 1. Introduction

NEC has already developed a monocular camera based structure from motion system which is accurate for requirements of road scene understanding. So we assume that ego-motion is given as an observed variable in our graphical model. And we have following features to our disposal for estimating the 3D poses (position + orientation) of traffic participants:

**Ground plane** We assume that all the traffic participants lie on a common ground plane. This is not particularly true for the cars that are parked off the road. However, for autonomous driver assistance applications we can ignore those cars.

**Detections** We assume that 2D car detections are available with tracking informations.

**Point tracks** We assume that 2D point tracks are available.

**GPS and Map information** We assume that GPS information is available and we use openstreetmaps.org for pulling out local map for the current car's position.

**Lane Information** We assume the lane detection works well and lane information is available.

**Size prior** The distribution of size of cars follows gaussian distribution.

**Collision** A reasonable output of the system donot has any overlapping cars.

## 2. Notation

Symbol	Meaning
$\mathbf{p}^{(i)}(t)$	Position of $i$ th car at time $t$
$\omega^{(i)}(t)$	Orientation of $i$ th car at time $t$
$\mathbf{B}^i$	3D bounding box of the car (dimensions)
$\mathbf{s}^i(t)$	State of car = $\{\mathbf{p}^{(i)}(t), \omega^{(i)}(t), \mathbf{B}^i\}$
$\mathbf{p}^{(c)}(t)$	Position of camera at time $t$
$\omega^{(c)}(t)$	Orientation of camera at time $t$
$\Omega^i(t)$	Relative car pose w.r.t. camera
$\mathbf{X}_o^{(i)}$	3D points tracked on car $i$ in its own frame
$\mathbf{u}^{(i)}(t)$	Projection of $\mathbf{X}_o^{(i)}$ in camera
$\pi_{\Omega^i(t)}(\cdot)$	Projection function for pose $\Omega^i(t)$
$\mathbf{d}^i(t)$	2D bounding box of the car in image

## 3. The Model

The objective is to find the most likely traffic participant state given various evidences  $\mathbb{E} = \{\{\mathbf{u}^{(i)}(t)\}, \{\mathbf{d}^i(t)\}, \text{lane det.}, \text{map}, \text{GPS}\}$ .

Mathematically, find:

$$\{\mathbf{s}^i(t)\}^* = \arg \max P(\{\mathbf{s}^i(t)\}|\mathbb{E}) \quad (1)$$

Bayes rule

$$P(\{\mathbf{s}^i(t)\}|\mathbb{E}) = \frac{1}{Z} P(\mathbb{E}|\{\mathbf{s}^i(t)\}) P(\{\mathbf{s}^i(t)\}) \quad (2)$$

Assume conditional independence according to graphical model in 1.

$$P(\{\mathbf{s}^i(t)\}|\mathbb{E}) = \frac{1}{Z} \prod_{t=s_i}^{e_i} \prod_{i,j:i \neq j} (P(\mathbf{s}^i(t), \mathbf{s}^j(t)))$$

$$\prod_{i=1}^N \prod_{t=s_i}^{e_i} P(\mathbf{d}^i(t)|\mathbf{s}^i(t)) P(\mathbf{u}^{(i)}(t)|\mathbf{s}^i(t))$$

$$P(\mathbf{s}^i(t)|L_m(t)) P(\mathbf{s}^i(t)|\mathbf{s}^i(t-1)) P(\mathbf{s}^i(t)) \quad (3)$$

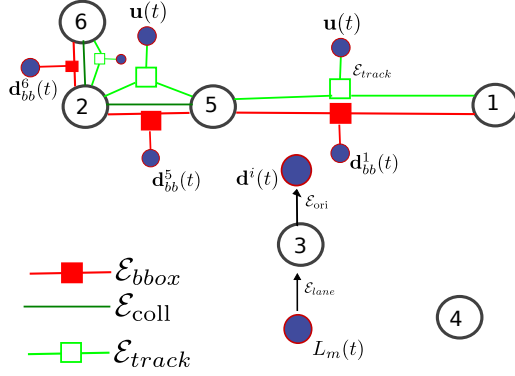


Figure 1. Graphical model. The six numbered black circles represent the unknown state variables of each car. All other nodes in the graphical model are assumed observed variables. Consider each energy in the model one by one. (1) Bounding box energy: The bounding box energy without occlusion modeling is a unary term, but with occlusion it becomes a higher order term that affects the state of occluder as well. In this graphical model we assume that the scene is being observed from left to right, hence "2" occludes "6" and "5" and "5" occludes "1". The bounding box detection is represented by  $\mathbf{d}_{bb}^i(t)$  and the statistical dependencies are represented by red lines. (2) Point tracks ( $\mathcal{E}_{track}^{it}$ ): Since occlusion is also included in modeling point tracks energy we have similar interdependencies for point tracks energy. The available point tracks are modeled by  $\mathbf{u}(t)$ . (3) Collision ( $\mathcal{E}_{coll}^{it}$ ): The collision energy mathematically is a dense graph between all the TP but here we represent collision among only those TP that are near enough to have a significant collision energy. (4) Orientation from detection ( $\mathcal{E}_{ori}^{it}$ ) (5) Orientation from lane (and map) information ( $\mathcal{E}_{lane}^{it}$ )

We can formulate similar objective function in negative log domain:<sup>1</sup>

$$\begin{aligned}
 -\log P(\{\mathbf{s}^i(t)\}|\mathbb{E}) &= Z' + \sum_{i,j:i \neq j} \sum_{t=s_i}^{e_i} \lambda_{col} \mathcal{E}_{col}^{ijt} \\
 &+ \sum_{i=1}^N \sum_{t=s_i}^{e_i} \lambda_{box} \mathcal{E}_{box}^{it} + \lambda_{track} \mathcal{E}_{track}^{it} \\
 &+ \lambda_{lane} \mathcal{E}_{lane}^{it} + \lambda_{dyn} \mathcal{E}_{dyn}^{it} + \lambda_{prior} \mathcal{E}_{prior}^{it} \quad (4)
 \end{aligned}$$

### 3.1. Collision energy

Bhattacharya coefficient  $\int_a^b \sqrt{p(x)q(x)}dx$  is a measure of similarity of two distributions  $p(x)$  and  $q(x)$ . If we represent traffic participants as gaussians in BEV, then similarity is a measure of collision. Exactly overlapping distribution results in coefficient as 1. The

<sup>1</sup>TODO:Resolve inconsistency in summation absorption in energy terms.

analytical form of Bhattacharya coefficient has been taken from <http://like.silk.to/studymemo/PropertiesOfMultivariateGaussianFunction.pdf>

$$\mathcal{E}_{col}^{ijt} = \frac{|\Sigma_i|^{\frac{1}{4}} |\Sigma_j|^{\frac{1}{4}}}{|P|^{\frac{1}{2}}} e^{-\frac{1}{8} (\mathbf{p}^{(i)}(t) - \mathbf{p}^{(j)}(t))^T P^{-1} (\mathbf{p}^{(i)}(t) - \mathbf{p}^{(j)}(t))} \quad (5)$$

where

$$P = \frac{1}{2} \Sigma_i + \frac{1}{2} \Sigma_j \quad (6)$$

$$\Sigma_i^{-1} = R_{\omega^{(i)}(t)}^T \begin{bmatrix} 2/l^i & 0 \\ 0 & 2/w^i \end{bmatrix} R_{\omega^{(i)}(t)} \quad (7)$$

### 3.2. Continuous Point tracks energy with occlusion

We want continuous modeling of occlusion. We model occlusion as a opacity term. Based on the dimensions  $\mathbf{B}^i$ , we can compute the covariance matrix  $\Sigma_i$  as shown in Sec. 3.2.6

So far we have modeled our point tracks energy in a non-smooth way:

$$\begin{aligned}
 \mathcal{E}_{traci}^{it}(\{\Omega^i(t)\}_i, \{\Omega^i(t-1)\}_i, \{\mathbf{B}^i\}_i) = \\
 \sum_{i=1}^N \sum_{j=1}^M \int_1^\infty a_j^i(\lambda) \|\mathbf{u}_j(t) - \pi_{\Omega^i(t)}(\pi_{\Omega^i(t-1)}^{-1}(\mathbf{u}_j(t-1)))\|^2 d\lambda
 \end{aligned} \quad (8)$$

The tricky part here is inverse projection  $\pi_{\Omega^i(t-1)}^{-1}(\cdot)$ . When we project an image point back to 3D we get a ray  $\lambda K^{-1} \mathbf{u}_j(t-1)$  which is ambiguous up to a scale factor  $\lambda$ . Using our hypothesized 3D layout of the scene we can estimate this scale factor, we can compute a distribution over the  $\lambda$ .

#### 3.2.1 Occupancy function

Assuming occupancy to be a probability distribution over 3D space. For each traffic participant the occupancy is modeled as a logistic function

$$f_{occ}^i(\mathbf{x}) = L(\mathbf{x}; \mathbf{p}^{(i)}(t), \Sigma_i) \quad (9)$$

where  $L(\cdot)$  is the logistic function defined by

$$L(\mathbf{x}; \mathbf{p}^{(i)}(t-1), \Sigma_i) = \frac{1}{1 + e^{-k(1-d(\mathbf{x}, \mathbf{p}^{(i)}(t-1)))}} \quad (10)$$

where  $d(\mathbf{x}, \mathbf{p}^{(i)}(t-1)) = (\mathbf{x} - \mathbf{p}^{(i)}(t-1))^T \Sigma_i (\mathbf{x} - \mathbf{p}^{(i)}(t-1))$  and  $k = 10 \ln 49$ .  $k$  is chosen such that  $L(\cdot) = 0.98$  when  $d(\cdot) = 0.9$

We model the probability of a point being projected to a camera is dependent up two factors, (1) reflection and (2) transmission through intermediate space.

### 3.2.2 Reflection probability

For lambertian reflection we replace the surface normal with the gradient of occupancy.

$$P_{\text{reflection}} = \sum_i (\max\{0, \nabla f_{occ}^i(\mathbf{x})^\top \hat{\mathbf{r}}_j\})^2 \quad (11)$$

where  $\hat{\mathbf{r}}_j = \frac{K^{-1}\mathbf{u}_j(t-1)}{\|K^{-1}\mathbf{u}_j(t-1)\|}$  is unit vector in the direction of ray. The gradient in the direction opposite to ray yields -ve probability which needs to be clipped off. Squaring the function keep it smooth near zero.

### 3.2.3 Transmission probability

A model for transmission of light through a material of thickness  $x$ , density  $\rho$  and opacity  $k_o$  is given by Beer-Lambert law

$$I(x) = I_0 e^{-k_o \rho x} \quad (12)$$

Since both opacity and density are represented by our occupancy function  $f_{occ}^i(\mathbf{x})$ , and also the domain of our  $f_{occ}^i(\mathbf{x})$  is  $[0, 1]$  instead of  $[0, \infty]$  as in case of  $k_o$ ; we replace  $e^{-k_o \rho}$  by our transparency function  $1 - f_{occ}^i(\mathbf{x})$ . So the transmission probability over a small distance  $d\lambda$  is given by

$$P_{\text{transmission}}(\lambda + d\lambda) = P_{\text{transmission}}(\lambda)(1 - f_{occ}^i(\mathbf{x}))^{d\lambda} \quad (13)$$

For our given ray  $\mathbf{x}_j = \lambda \hat{\mathbf{r}}_j$ , the probability that the point  $\mathbf{u}_j(t-1)$  is reflected from a distance  $\lambda$  is given by

$$f_\lambda(\mathbf{u}_j(t-1), \lambda) = P_{\text{reflection}} \prod_0^\lambda (1 - f_{occ}(\lambda \hat{\mathbf{r}}_j))^{d\lambda} \quad (14)$$

where  $\prod_0^\lambda$  represents the *product integral* from 0 to  $\lambda$ . A product integral is a simple integral in log domain

$$\prod_0^\lambda (1 - f_{occ}(\lambda \hat{\mathbf{r}}_j))^{d\lambda} = e^{\int_0^\lambda \ln(1 - f_{occ}(\lambda \hat{\mathbf{r}}_j)) d\lambda} \quad (15)$$

### 3.2.4 Reprojection error

$$f_{\text{reproj}}^i(\mathbf{u}_j(t-1), \lambda) = \pi_{\Omega^i(t)}(\pi_{\Omega^i(t-1)}^{-1}(\mathbf{u}_j(t-1))) \quad (16)$$

$$= \frac{p_{1:2}\lambda + q_{1:2}}{p_3\lambda + q_3} \quad (17)$$

where  $p_{1:3} = KR_t^i(R_{t-1}^i)^\top \hat{\mathbf{r}}_j$  and  $q_{1:3} = KR_t^i(R_{t-1}^i)^\top T_{t-1} + KT_t$

Now that we have an association from TP  $i$  to point track  $j$  through  $\lambda$ , we can come up with an association probability

$$a_{j,t-1}^i(\lambda) = \frac{(\max\{0, \nabla f_{occ}^i(\mathbf{x})^\top \hat{\mathbf{r}}_j\})^2}{P_{\text{reflection}}} f_\lambda(\mathbf{u}_j(t-1), \lambda) \quad (18)$$

$$= (\max\{0, \nabla f_{occ}^i(\mathbf{x})^\top \hat{\mathbf{r}}_j\})^2 \prod_0^\lambda (1 - f_{occ}(\lambda \hat{\mathbf{r}}_j))^{d\lambda} \quad (19)$$

Now coming back to our point track energy, the distribution over  $\lambda$  becomes our association function which instead of associating each point track with a traffic participant, associates it with a depth ( $\lambda$ ). Hence our summation over  $N$  traffic participants is replaced by an integral over the entire ray.

$$\begin{aligned} \mathcal{E}_{\text{track}}^i(\{\Omega^i(t)\}_i, \{\Omega^i(t-1)\}_i, \{\mathbf{B}^i\}_i) = \\ \sum_{i=1}^N \sum_{j=1}^M \int_1^\infty a_{j,t-1}^i(\lambda) \|\mathbf{u}_j(t) - f_{\text{reproj}}^i(\mathbf{u}_j(t-1), \lambda)\|^2 d\lambda \end{aligned} \quad (20)$$

### 3.2.5 Approximations

Reflection probability of  $i$ th traffic participant is easy to compute analytically

$$\begin{aligned} P_{\text{reflection}}^i &= (\max\{0, \nabla f_{occ}^i(\mathbf{x})^\top \hat{\mathbf{r}}_j\})^2 \\ &= (\max\{0, \nabla L(\mathbf{x}; \mathbf{p}^{(i)}(t-1), \Sigma_i)^\top \hat{\mathbf{r}}_j\})^2 \end{aligned} \quad (21)$$

where

$$\begin{aligned} \nabla L(\mathbf{x}; \mathbf{p}^{(i)}(t-1), \Sigma_i)^\top \hat{\mathbf{r}}_j \\ = \nabla k d_i(\mathbf{x}) \operatorname{sech}^2\left(\frac{k}{2} d_i(\mathbf{x})\right) \end{aligned} \quad (22)$$

where  $d_i(\mathbf{x}) = 1 - d(\mathbf{x}, \mathbf{p}^{(i)}(t-1))$  is a signed distance measure from the contour of the ellipsoid where  $d(\mathbf{x}, \mathbf{p}^{(i)}(t-1))$  is 1.

However, the transmission probability needs to be approximated. So based on intuition, we approximate the  $P_{\text{transmission}}$  by following function

$$P_{\text{transmission}} = \prod_i L_u(\mathbf{u}, \mu_u^i, \Sigma_u^i) L_\lambda(\lambda; \mu_d^i) \quad (23)$$

$$L_u(\mathbf{u}, \mu_u^i, \Sigma_u^i) = \frac{1}{1 + e^{-k_u(1 - (\mathbf{u} - \mu_u^i)^\top \Sigma_u^i (\mathbf{u} - \mu_u^i))}} \quad (24)$$

$$L_\lambda(\mathbf{u}, \lambda; \mu_d^i) = \frac{1}{1 + e^{-k_d(\lambda - \mu_d^i)}} \quad (25)$$

where

$$\mu_u^i = \pi_{\Omega^i(t)}(\mathbf{p}^{(i)}(t-1)) \quad (26)$$

$$\Sigma_u^i = \pi_{\Omega^i(t)}(\Sigma_i) \quad (27)$$

$$\mu_d^i(\mathbf{u}) = \Omega^i(t) \quad (28)$$

$$k_u = 10 \log(49) \quad (29)$$

$$k_d = \frac{\log(49)}{\sqrt{h^2 + l^2 + w^2}} \quad (30)$$

is the distance of the centre of the traffic participant from the camera.

The association probability becomes

$$a_{j,t-1}^i(\lambda) = \text{sech}^4\left(\frac{k}{2}d_i(\mathbf{x})\right) (\max\{0, \nabla k d_i(\mathbf{x})^\top \hat{\mathbf{r}}_j\})^2 \prod_i L_u(\mathbf{u}, \mu_u^i, \Sigma_u^i) L_\lambda(\mathbf{u}, \lambda; \mu_d^i) \quad (31)$$

So the energy becomes

$$\mathcal{E}_{\text{track}}^{it}(\cdot) = \sum_{i=1}^N \sum_{j=1}^M \int_1^\infty a_{j,t-1}^i(\lambda) E_{\text{reproj}}^i(\lambda) d\lambda \quad (32)$$

where  $x = \lambda \hat{\mathbf{r}}_j$  and  $E_{\text{reproj}}^i(\lambda) = \|\mathbf{u}_j(t) - f_{\text{reproj}}^i(\mathbf{u}_j(t-1), \lambda)\|^2$  is reprojection error which is a quadratic in  $\lambda$

The integral in the above expression is computed numerically.

### 3.2.6 Sigma of ellipsoid

$$\mu_t^i = \begin{bmatrix} 0 & 0 & \frac{h}{2} \end{bmatrix}^\top \quad (33)$$

$$\Sigma_t^i = \begin{bmatrix} \frac{4}{l^2} & 0 & 0 \\ 0 & \frac{4}{w^2} & 0 \\ 0 & 0 & \frac{4}{h^2} \end{bmatrix} \quad (34)$$

In tracklet coordinates the equation of ellipsoid is

$$(\mathbf{x}_t - \mu_t^i)^\top \Sigma_t^i (\mathbf{x}_t - \mu_t^i) = 1 \quad (35)$$

Moving to camera coordinates

$$(R\mathbf{x}_c + t - \mu_t^i)^\top \Sigma_t^i (R\mathbf{x}_c + t - \mu_t^i) = 1 \quad (36)$$

Let  $t' = t - \mu_t^i$

$$(R\mathbf{x}_c + t')^\top \Sigma_t^i (R\mathbf{x}_c + t') = 1 \quad (37)$$

$$\mathbf{x}_c^\top R^\top \Sigma_t^i R \mathbf{x}_c + 2(R^\top t')^\top R^\top \Sigma_t^i R \mathbf{x}_c + t'^\top \Sigma_t^i t' = 1 \quad (38)$$

Let  $\Sigma_c^i = R^\top \Sigma_t^i R$  and  $\mu_c^i = -R^\top t'$

$$(\mathbf{x}_c - \mu_c^i)^\top \Sigma_c^i (\mathbf{x}_c - \mu_c^i) - \mu_c^{i\top} \Sigma_c^i \mu_c^i + t'^\top \Sigma_t^i t' = 1 \quad (39)$$

$$\text{Let } \Sigma_c^i = \frac{\Sigma_c^i}{1 - t'^\top \Sigma_t^i t' + \mu_c^{i\top} \Sigma_c^i \mu_c^i}$$

$$(\mathbf{x}_c - \mu_c^i)^\top \Sigma_c^i (\mathbf{x}_c - \mu_c^i) = 1 \quad (40)$$

Hence, we have following expression for mean and sigma of ellipsoid:

$$\mu_c^i = -R^\top t' \quad (41)$$

$$\Sigma_c^i = \frac{\Sigma_c^i}{1 - t'^\top \Sigma_t^i t' + \mu_c^{i\top} \Sigma_c^i \mu_c^i} \quad (42)$$

## 4. Continuous object detection score energy with occlusion

To start with we have hypothesized 3D bounding boxes  $\{\mathbf{B}^i\}_i$  over all traffic participants and we have object detection scores modeled as mixture of Gaussians over four dimensional space of bounding box boundaries. Let the detection score function over the detection  $\mathbf{d}^i(t)$  be

$$S(\mathbf{d}^i(t)) = \sum_j A_j \exp((\mathbf{d}^i(t) - \mu_j^{(d)})^\top \Sigma_j^{(d)-1} (\mathbf{d}^i(t) - \mu_j^{(d)})) \quad (43)$$

### 4.1. Soft occlusion regions

Let the projections of hypothesized soft ellipsoids be  $\pi_{\Omega^i(t)}(\mathbf{B}^i)$  with some mean  $\mu_i$  and variance  $\Sigma_i$  (assuming that projection of ellipsoids can be approximated by ellipses on image plane). We take the occlusion (intersection of two ellipses) to be modelled by product of the respective Gaussians.

$$O_{ij}(x; \mu_{Oij}, \Sigma_{Oij}) = N(x; \mu_i, \Sigma_i) N(x; \mu_j, \Sigma_j) \quad (44)$$

The product of Gaussian can be computed analytically.

### 4.2. Merging occlusion regions with detection score

But what does detection score function represents? It is a measure of correlation of trained object features and the features in the candidate bounding box. If the detection scores are indeed Gaussians around the mean bounding box  $\mu_j$ , the  $\Sigma_j^{-1}$  represents the variance (and covariance) around the mean bounding box. But when the candidate objects seem to be occluded we have additional information available. In such a case, we can say that the occluded object might have unexpected effect on the detection scores. However, the uncertainty in the detection score in loop sided. The mean detection bounding box must have picked up correct boundaries on the un-occluded sides but we cannot say the same

about occluded boundaries. The effect on detection score depends a lot on the features of the occluder. Given this understanding we want to decrease our confidence on the mean detection boundaries around the occluded boundaries. One of the simplest ways will be to scale the appropriate diagonal element of  $\Sigma_j$  by an appropriate scaling factor.

We observe that the soft occlusion regions are 2D while the detection scores are 4D. But that should not be a problem because the detection scores are modeled as  $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]^\top$ , we can stack the vectors in the soft occlusion region domain as the regions are independent of min or max variation. The parameters of 4D occlusion distribution are given by

$$\mu'_{Oij} = [\mu_{Oij}^\top \mu_{Oij}^\top]^\top \Sigma'_{Oij} = \begin{bmatrix} \Sigma_{Oij} & \Sigma_{Oij} \\ \Sigma_{Oij} & \Sigma_{Oij} \end{bmatrix} \quad (45)$$

With our discussion so far, the function  $O_{ij}(\cdot)$  gives an additional measure of uncertainty associated with each point in the space. To finally re-model our detection scores scaled by continuous occlusion we sample  $O_{ij}(\cdot)$  at the mean detection boundaries from GMM  $S(\cdot)$  and we scale the detection boundary variance by  $\mathcal{P}_{ij} = \rho_{ij} \rho_{ij}^\top$  where  $\rho_{ij} = \mathbf{1} + O_{ij}(\mu_j^{(d)}; \mu'_{Oij}, \Sigma'_{Oij})$ . The new variance in detection score is given by

$$\Sigma_j^{(d)} = \prod_{k \in \text{occluders}} \mathcal{P}_{kj} \Sigma_j^{(d)} \prod_{k \in \text{occluders}} \mathcal{P}_{kj}^\top \quad (46)$$

The detection scores GMM with occlusion is given by replacing the covariance matrix

$$S'(\mathbf{d}^i(t)) = \sum_j A_j \exp((\mathbf{d}^i(t) - \mu_j^{(d)})^\top \Sigma_j'^{(d)-1} (\mathbf{d}^i(t) - \mu_j^{(d)})) \quad (47)$$

### 4.3. Lane energy

The lanes be are modeled as splines. The lanes geometry information can be obtained from camera input as well as from the available maps localized by GPS. We need to combine information in a different framework to lead to a more accurate solution by using both the sources. In a probabilistic framework it is always useful to compute uncertainty along the estimated values. Here we assume that the confidence in lane detection is inversely proportional to the square of distance of the specific point on the lane to the camera  $\Sigma_{L_m}(x_{L_m}) = \frac{1}{\|\mathbf{x}_{L_m} - \mathbf{p}^{(c)}(t)\|^2}$ .

A lane is modeled by four control points of a Bézier curve  $L_j = \{l_0, l_1, l_2, l_3\}$ . The parametric curve is given by  $L_j(k) = \sum_{i=0}^3 C_i (1-t)^{3-i} t^i l_i$ . Bézier are double differentiable so one can find tangents ( $L'_j(k)$ ) and normals

$R_{\frac{\pi}{2}} L'_j(k)$  where  $R_{\frac{\pi}{2}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$  is 90° rotation matrix.

Now, there are some algorithms [2, 1] to approximate the closest point on a Bézier curve  $k_x = \pi_{L_m(k)}(x)$ . Consequently one can get solutions for shortest distance  $s = \text{DIST}(L_j(k), x)$  and orientation at a closest point on the lane  $\mathbf{t} = \text{TAN}(L_j(k), x)$ .

#### 4.3.1 Orientation within lane

- We can find the tangent to a lane
- The energy is given by the dot product between car orientation and tangent to the lane at that point.

$$\mathcal{E}_{\text{lane}}^{it} = \sum_{m \in M_{\text{close}}} (1 - \omega^{(i)}(t) \cdot \text{TAN}(L_m(k), \mathbf{p}^{(i)}(t))) \Sigma_{L_m}(\mathbf{p}^{(i)}(t)) \quad (48)$$

where  $M_{\text{close}} = \{m : \text{DIST}(L_m(k), \mathbf{p}^{(i)}(t)) < 50\}$  is the set of nearby lanes and

$$\Sigma_{L_m}(\mathbf{p}^{(i)}(t)) = \frac{1}{1 + \exp(-q(w_{\text{road}} - \text{DIST}(L_m(k), \mathbf{p}^{(i)}(t))))} \quad (49)$$

for some constant  $w_{\text{road}}$  that represents the width of the road.

### 4.4. Transition probability

Dynamics constraints should not only enforce smooth trajectories, but also the holonomic constraints.

The following equation adds a penalty if the change in position is not in direction of previous orientation.

$$\mathcal{E}_{\text{dyn-hol}}^{it} = 1 - \omega^{(i)}(t-1) \cdot (\mathbf{p}^{(i)}(t) - \mathbf{p}^{(i)}(t-1)) \quad (50)$$

The following equation adds a penalty for change in position and orientation but a penalty for change in velocity is much better approximation. However, in a Markovian setting that would mean extending the state space of the car to include velocity.

$$\mathcal{E}_{\text{dyn-ori}}^{it} = \|\omega^{(i)}(t) - \omega^{(i)}(t-1)\|^2 \quad (51)$$

$$\mathcal{E}_{\text{dyn-vel}}^{it} = \|(\mathbf{p}^{(i)}(t) - 2\mathbf{p}^{(i)}(t-1)) + \mathbf{p}^{(i)}(t-2)\|^2 \quad (52)$$

As a result the dynamics are modeled by weighted combination of holonomic constraint and smoothness constraints.

$$\lambda_{\text{dyn}} \mathcal{E}_{\text{dyn}}^{it} = \lambda_{\text{dyn-hol}} \mathcal{E}_{\text{dyn-hol}}^{it} + \lambda_{\text{dyn-ori}} \mathcal{E}_{\text{dyn-ori}}^{it} + \lambda_{\text{dyn-vel}} \mathcal{E}_{\text{dyn-vel}}^{it} \quad (53)$$

#### 4.5. Size Prior

Prior can include among many other things the size prior on the car.

$$\mathcal{E}_{\text{prior}}^{it} = (\mathbf{B}^i - \hat{\mathbf{B}})^\top \Sigma_{\hat{\mathbf{B}}}^{-1} (\mathbf{B}^i - \hat{\mathbf{B}}) \quad (54)$$

where  $\hat{\mathbf{B}}$  is the mean traffic participant dimensions and  $\Sigma_{\hat{\mathbf{B}}}$  is the correspondence covariance matrix.

#### References

- [1] X.-D. Chen, Y. Zhou, Z. Shu, H. Su, and J.-C. Paul. Improved algebraic algorithm on point projection for beziercurves. In *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pages 158–163. IEEE, 2007.
- [2] Y. L. Ma and W. T. Hewitt. Point inversion and projection for nurbs curve and surface: control polygon approach. *Computer Aided Geometric Design*, 20(2):79–99, 2003.

func	equation	ms per 6 frames	comment
totalBBoxEnergy	Sec ??	268	The equation summed up for all 6 tracklets
laneEnergy	Sec 4.3	163	
laneOrientationEnergy	Eq (48)	155	
collisionEnergyHellingerDistance	Eq (5)	93	
totalPosTransitionEnergy	Eq (50)	15	
totalSizeEnergy	Eq (54)	0.9	
occlusionMaskFromOptVector	Sec ??	123	
bboxEnergyFromOccMask	Eq (??)	140	
vis_area_from_triangle	$\ \triangle_{\Omega^i(t)}^s(\mathbf{B}^i) \setminus o(\Omega_z^i(t))\ $	76	
distance2curve	$\text{DIST}(L_j(k), x)$	74	Called 2 times per tracklet
filterMapWaysByDistance	$M_{\text{close}} = \{m : \text{DIST}(\cdot) < 50\}$	52	called 4 times per tracklet
area_of_triangle	$\ \triangle_{\Omega^i(t)}^s(\mathbf{B}^i)\ $	48	
occlusionMask	Step 3 of Sec ??	46	
func	equation	ms per 6 frames	comment
totalContPtTracksEnergy	Sec 3.2	25000	Called 8 times per edge
integrand	Eq (32)	23000	
repmat		5000	
assocCoeffEval	Eq (31)	4000	
evalPreflection	Eq (21)	3000	
gradfocci	$\nabla f_{occ}^i(\mathbf{x})^\top \hat{\mathbf{r}}_j$	2040	
associationCoefficientInit	Eq (30)(41)	1706	
projectToImage	$\frac{Ku}{Ku(3)}$	4000	
evalCumulativePtrans	(23)	1178	
ptransmissionApproxInit	(30)	1000	
ellipsoidCentreDist	$(x - \mu)^\top \Sigma (x - \mu)$	1000	
squeeze		951	
trackletToCamTransform	$R_{\omega^{(i)}(t)}, t$	786	
ellipsoidMeanSigma	Eq (41)	812	