Report X-C-TR-ContinentalHOLM-16

HOLM WP: Confidences of Traffic Signs

Tobias Weis

email: weis@ccc.cs.uni-frankfurt.de

June 6, 2017

Software Engineering
Frankfurt Institute for Advanced Studies
Johann Wolfgang Goethe-Universität
Frankfurt am Main, Germany
http://www.bfnt-frankfurt.de

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Contents

Acronyms

TS traffic sign

GT-TS ground truth traffic sign

1 Scope

The scope of this work is the development and prototyping of a method to compute confidences of traffic sign (TS) given multiple observations. A set of traffic sign observations (type, 3d position, confidence of observation) in a training area as well as accurate measurements of traffic signs of the same area (*reference data*) are given as input for the prototype. The desired output consists of an optimized set of TS with confidence values based on all observations.

Deliverables are:

- A document on the theory of how to compute optimal confidence values (this document)
- Source code for confidence computation (e.g. in a script language) that could be integrated into or re-implemented within the Road Database system

Remarks:

- Prototype can work based on GPS coordinates of TS, but the technique has to be transferable to a local coordinate system resulting from snippet merging
- The approach for confidence computation should be incremental, i.e. given new observations the set of TS positions and confidence values must be updated
- The approach should handle correctly changes in the world, i.e. when the position of signs has changed
- The reference data should be used to learn optimal confidence values. The method must work without reference data later
- Confidences should be computed regarding 1) existence, 2) type, 3) position of the TS
- It should be possible to set a threshold for selecting a sub-set of TS, e.g. "give me the optimal set of TS given their confidence has to be greater than 95%"

2 Data

This section describes the available data, necessary transformations, as well as a first analysis thereof.

2.1 Reference data

The file *Conti_A7_Kreuz_Memmingen_2016_04_29.xodr* contains accurate reference data in OpenDRIVE format [D+15].

As described in this specification, TS coordinates are encoded with their distance to the starting point of the road they belong to. Each road is encoded as a starting point and a series of geometric parts (here: polynomials). In order to extract the TS coordinates, these geometries have to be calculated.

The python script *sign_extractor.py* implements the necessary functionality and saves the computed TS coordinates to a XML file with the structure defined in section 10.2.1.

2.2 TS codes

TS codes and subtypes of the estimated data differ from the official codes defined in STVO 2013 Anlage 2 (http://www.dvr.de/betriebe_bg/daten/stvo/anlage2.htm) and VZKAT (http://www.vzkat.de/vzkat.htm). The python script *SignMapper.py* provides a mapping between both encoding schemes.

2.3 Estimated data format (current)

The latest format adopts another representation for GPS coordinates.


```
<Document version="0.1" lat="47.996180757731942" lon="10.149478010309279" height="2.6832762886597945">
  <Signs>
    <Sign id="1" type="626030" subtype="-1" conf_id="1" ts="-1">
      <ReferenceLines>
        <ReferenceLine type="start point">
          <ReferencePoint lat="47.996180757731942" lon="10.149478010309279" height="2.6832762886597945"/>
          <PointList>
            <Point id="1" x="496.15810381815038" y="-6.2692221544375144" z="1.6565999663895354"/>
          </PointList>
        </ReferenceLine>
      </ReferenceLines>
    </Sign>
    ...
```


Each sign-entry contains its type and a confidence on it, its position is encoded in terms of x/y/height-offsets in meter relative to a GPS-coordinate (lat,lon,height in m). This file currently contains 6 estimates.

Past data formats, analysis and processing efforts thereof can be found in section 10.

2.4 Perturbations and error-sources of data

Table 1 lists possible sources of perturbations/errors in a TS-detection pipeline. The developed integration- and fusion-algorithm should be able to handle them.

Table 1: Error-Sources and their consequences

Source	Consequence	Specifics
GPS-errors	positional errors	1-8m regular, might be as high as 60m
Sync-delays	positional errors	Synchronization-delays between sensors might cause positional error along the direction of driving
Sensor resolution	positional errors	Different resolutions of sensors might introduce discretization-errors
Sensor-bias	positional errors	Systemic errors (of a certain brand of sensor or computation-system inside a car) that introduces bias
Weather	Misdetection	Fog/Heavy Rain
Occlusions	Misdetection	Temporary object between sensor and sign
Algorithm	False positive	The algorithm might detect a sign although there does not exist one

Without additional knowledge about sensor- and algorithm-specifics, we can work with nested assumptions about the distributions of errors/uncertainties in the position of measurements:

1. No assumptions: Uniformly distributed errors
2. Normal assumption: Normally distributed errors (diagonal covariance, radius integration)
3. Normal assumption 2: Normally distributed errors, elongated along the street to reflect sync-delays and positional errors along the driving-direction
4. Normal assumption 2 + Sensor bias: Systematically shifted distributions

Regarding the rates of misdetections, false positives and misclassifications, the following types of errors can be assumed:

1. Uncorrelated assumption: Uniformly distributed with a certain rate ϵ
2. Type-Correlated assumption: Systemic errors in detection/classification for certain TS-types
3. Time-Correlated assumption: Systemic errors during a certain period of time (to reflect periods of bad weather)

3 Simulation

In order to create controlled data on which algorithms can be evaluated, a simulation scheme has been derived. Arbitrarily curved roads can be created on a metric grid by using polynomial equations. Evenly spaced TS are simulated on this curve. Signs can be simulated one-sided (only to the right/left side of the street-polygon with orthogonal offset), or two-sided. We consider those TS as ground truth traffic sign (GT-TS).

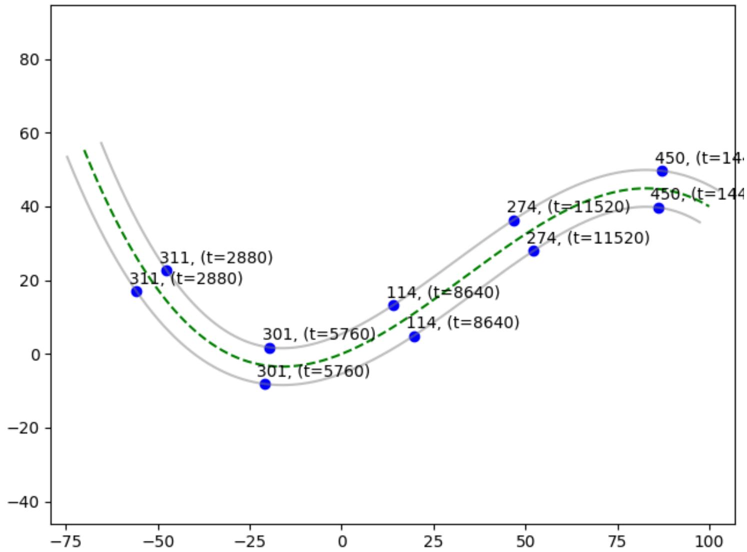

Figure 1: Simulation of a curved street with double-sided TS

Estimated TS are simulated by drawing from a parametrizable distribution. Under the assumption of normally distributed errors, the covariance of this distribution can be set according to assumed measurement uncertainties. Each estimated TS is assigned the type of the GT-TS it is sampled from, as well as its covariance matrix and a time-stamp. The simulation-order is always consecutive in terms of the order of GT-TS on the simulated road.

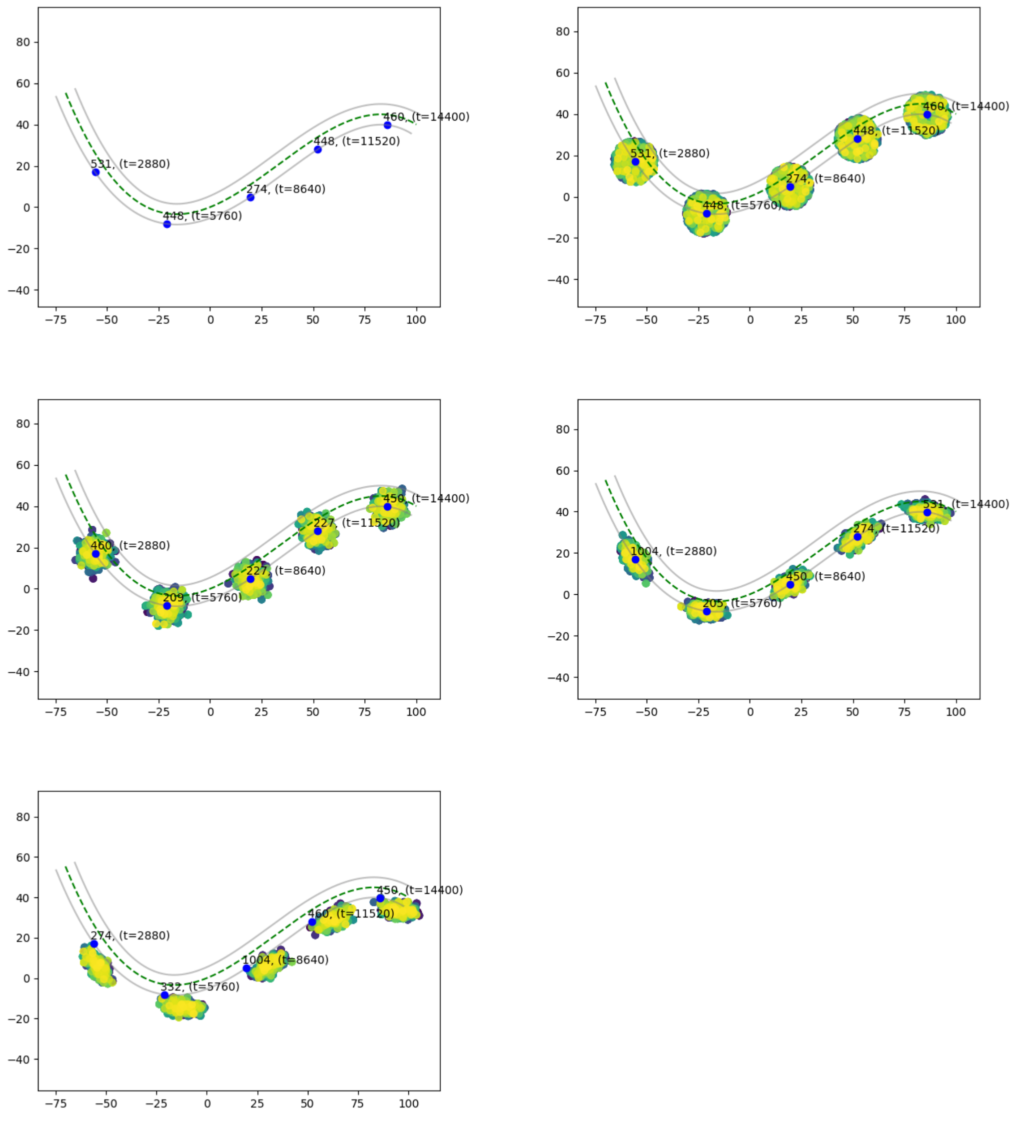

Figure 2: Simulations with different distributions of measurements. Upper-Left: Simulated GT-TS, Upper-right: Uniformly distributed estimates in a radius, Mid-left: Estimates distributed according to a normal-assumption (diagonal covariance), Mid-right: Estimates distributed according to a normal-assumption and elongated along the direction of driving, Lower-left: Estimates of mid-right with a systematic bias (relative to the cars heading direction)

4 Pipeline

It is assumed that estimated TS arrive in sets ($\mathtt{TS}_t^{\mathtt{est}}$) if they are uploaded from a specific car. TS that already exist in the database are denoted as $\mathtt{TS}_t^{\mathtt{db}}$. The overall integration pipeline without specific definitions of underlying implementations can then be split up into the following steps (as described in pseudocode in Algorithm 1):

1. Extract TS-pairs for both $\mathtt{TS}_t^{\mathtt{est}}$ and $\mathtt{TS}_t^{\mathtt{db}}$
2. For each estimated TS-pair, find TS-pairs that already exist in the database and are within a certain radius. If such a pair is found, integrate. Else add the estimated TS-pair to the database.
3. Proceed similar with single TS

The rationale behind this matching process can be explained using Figure 3.

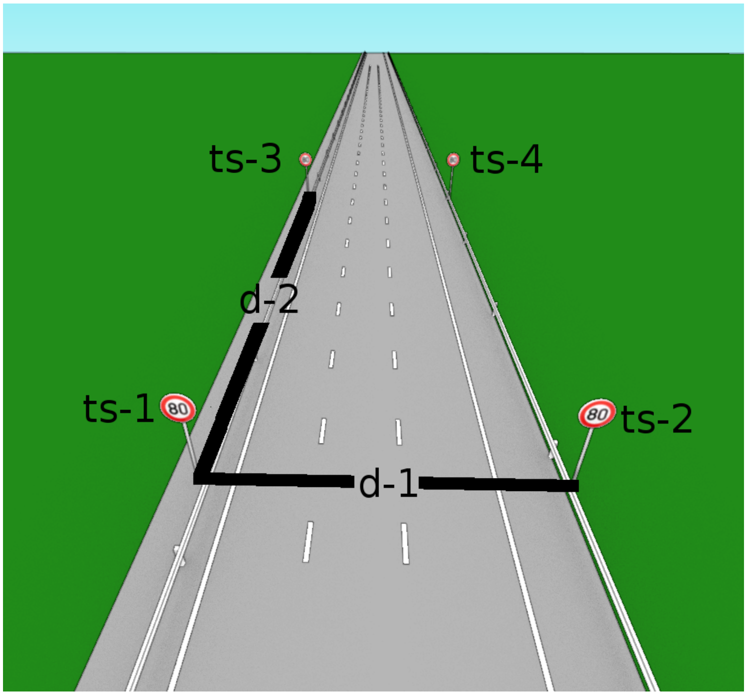

Figure 3: Signpairs

During the integration of TS, it is in our interest to set the integration-radius as high as possible in order to properly integrate matching perturbed signs. But it should still be possible to correctly distinguish between different TS that have the same typenumber (e.g. ts-1 and ts-3). One way of integration is then to set the matching-radius τ to d-2 / 2 or less. However, if we assume that d-1 (the distance between a sign-pair) is less than d-2 / 2, this would become the new limiting factor if TS are matched on a per-TS basis. If TS that are paired and belong together are encoded, pairs of TS can be matched correctly while still keeping the original integration-radius of d-2 / 2 or less. Furthermore, if a measured pair ts-1a and ts-2b (that indeed matches ts-1 and ts-2) is offset by more than the integration radius of d-1 / 2, one of the TS would be matched to a wrong partner while the other would not be matched at all (see Figure 4.

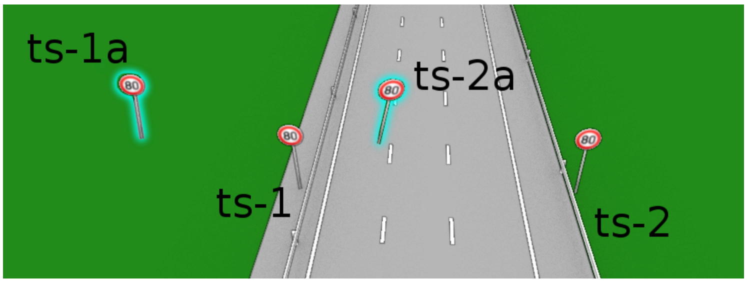

Figure 4: If matching was done on a per-sign-basis, mismatches could occur for pairs of TS. In above figure, ts-2a would be falsely matched to ts-1, whereas ts-1a would not be matched at all for integration-radii of d-1 / 2.

4.1 Integration and matching

Depending on the assumptions about error-distributions, different models can be applied for the integration of estimates:

Table 2: Mapping of error-distributions to integration schemes

Error-distribution	Integration-Method	Parameters
Uniform	Mean/Variance/Density	Bandwidth, Matching radius
Normal (diag)	Bayesian	Measurement-matrix H, State-distribution, Matching radius
	Mean/Variance/Density	Bandwidth, Matching radius
Normal (free)	Bayesian	Measurement-matrix H, State-distribution, Matching radius
	Mean/Variance/Density	Bandwidth, Matching radius
Normal + Bias	Bayesian, Density	Measurement-matrix H, State-distribution, Matching radius
	Mean/Variance/Density	Bandwidth, Matching radius, Estimated bias

Data: $\mathtt{TS}_t^{\mathtt{est}}$, $\mathtt{TS}_t^{\mathtt{db}}$
Result: $\mathtt{TS}_{t+1}^{\mathtt{db}}$
// Compute pairs in estimated TS
$\mathtt{TS}_t^{\mathtt{est,paired}} = \{\}$;
for $\forall\{\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}\} \in \mathtt{TS}_t^{\mathtt{est}}, i \neq j$ **do**
  if $distance(\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}) < \tau$ *and* $typenumber(\mathtt{ts}_{i,t}^{\mathtt{est}}) == typenumber(\mathtt{ts}_{j,t}^{\mathtt{est}})$ **then**
    $\mathtt{TS}_t^{\mathtt{est,paired}} = \mathtt{TS}_t^{\mathtt{est,paired}} \cup \{\{\mathtt{ts}_{i,t}^{\mathtt{est,paired}}, \mathtt{ts}_{j,t}^{\mathtt{est,paired}}\}\}$;
  end
end

// Compute pairs in database TS
$\mathtt{TS}_t^{\mathtt{db,paired}} = \{\}$;
for $\forall\{\mathtt{ts}_{i,t}^{\mathtt{db}}, \mathtt{ts}_{j,t}^{\mathtt{db}}\} \in \mathtt{TS}_t^{db}, i \neq j$ **do**
  if $distance(\mathtt{ts}_{i,t}^{\mathtt{db}}, \mathtt{ts}_{j,t}^{\mathtt{db}}) < \tau$ *and* $typenumber(\mathtt{ts}_{i,t}^{\mathtt{db}}) == typenumber(\mathtt{ts}_{j,t}^{\mathtt{db}})$ **then**
    $\mathtt{TS}_t^{\mathtt{db,paired}} = \mathtt{TS}_t^{\mathtt{db,paired}} \cup \{\{\mathtt{ts}_{i,t}^{\mathtt{db}}, \mathtt{ts}_{j,t}^{\mathtt{db}}\}\}$;
  end
end

// Match pairs
for $\{\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}\} \in \mathtt{TS}_t^{\mathtt{est,paired}}$ **do**
  for $\{\mathtt{ts}_{k,t}^{\mathtt{db}}, \mathtt{ts}_{l,t}^{\mathtt{db}}\} \in \mathtt{TS\_Paired}^t$ **do**
    if *pair_distance*$(\{\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}\}, \{\mathtt{ts}_{k,t}^{\mathtt{db}}, \mathtt{ts}_{l,t}^{\mathtt{db}}\}) < \tau$ **then**
      integrate_pair$(\{\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}\}, \{\mathtt{ts}_{k,t}^{\mathtt{db}}, \mathtt{ts}_{l,t}^{\mathtt{db}}\})$;
    else
      create_pair$(\{\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{est}}\})$;
    end
  end
end

// Match solos
for $\mathtt{ts}_{i,t}^{\mathtt{est}} \in \mathtt{TS}_t^{\mathtt{est}} \setminus \mathtt{TS}_t^{\mathtt{est,paired}}$ **do**
  for $\mathtt{ts}_{j,t}^{\mathtt{db}} \in \mathtt{TS}_t^{\mathtt{db}}$ **do**
    if $distance(\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{db}}) < \tau$ **then**
      integrate$(\mathtt{ts}_{i,t}^{\mathtt{est}}, \mathtt{ts}_{j,t}^{\mathtt{db}})$;
    else
      create$(\mathtt{ts}_{i,t}^{\mathtt{est}})$;
    end
  end
end

Algorithm 1: Integration algorithm

5 Integration and Confidence estimation - Parametric

As defined in the scope, the goal of this section is to compute optimal confidence values. The approach should be incremental, handle changes in the world and should work without reference data. Confidences should be computed regarding existence, position and type. It should be possible to threshold individual confidences for selecting a subset.

> **Definition:** A *traffic sign (TS)* denotes a structure that includes a 3D position in latitude, longitude (WGS84) as well as height (m). It furthermore includes a covariance-matrix encoding the uncertainties of said coordinates. Its semantic type is encoded as numeric identifier, usually a 3-digit code for main type (i.e. speed-limit equals 274) and a subtype with at most 3 digits (i.e. speed limit 80: 274-80).

5.1 Considerations

As we currently have at most three measurements per sign (without covariance matrix) and limited knowledge about the involved factors of the data-generating model, assumptions about bias/variance of measurements have to be made. Further considerations on potential sources of perturbations/errors can be found in the appendix (section 8).

5.2 Bayesian minimum-variance-estimator

As per definition of the scope, TS estimates will be encoded as their 3D positions and according covariance matrices representing the uncertainties. We propose a bayesian minimum-variance-estimator in form of a modified Kalman filter. The proposed workflow for map-creation is the following: Anytime a measurement is detected, its distance to signs with the same type/subtype combination is calculated. If no match is found within a predefined radius [1], append it to our database with the covariance of the measurement. If a match is found, integrate the TS by updating pose and covariance of the sign in the database. Note that in contrast to the classic Kalman filter, we have no "time-update" ($x_k = Ax_{k-1} + Bu_k + w_{k-1}$), as there do not exist control inputs or state-transitions.

The measurement update involves the following equations:

$$y_k = Hx_k + v_k, \tag{1}$$

where H represents the measurement equation that relates the current state x_k (in our case, this is a single TS existent in the database) to a measurement y_k (an estimated TS) under measurement noise v_k that is assumed to be normal distributed with mean zero and variance R: $p(v) \sim N(0, R)$. Here, there is no special relation

[1] this radius can be motivated by the average distance between two TS of the same type, as well as the uncertainties in the measurements

between the current state x_k and a measurement y_k, therefore H is set to the identity matrix $H = I$.

Now the process of updating the estimated state of a TS, as well as its covariance-matrix can be formulated as follows:

A distance vector between the current state estimate x_k^- and the measurement y_k is calculated:

$$v_k = y_k - H_k x_k^- \tag{2}$$

We calculate an intermediate covariance matrix S_k by integrating the current covariance matrix of the state P_k^- and the covariance matrix of the measurement R_k:

$$S_k = H_k P_k^- H_k^T + R_k \tag{3}$$

and compute the Kalman gain. This gain determines how strong the current estimate will be moved towards the measurement, as well as how much the covariance matrix of the state estimate will be rescaled depending on the covariance matrices of both estimates:

$$K_k = P_k^- H_k^T S_k^{-1}. \tag{4}$$

The updated state estimate based on this gain is then computed as

$$x_k = x_k^- + K_k v_k, \tag{5}$$

and the state covariance matrix as

$$P_k = P_k^- - K_k S_k K_k^T \tag{6}$$

5.2.1 Estimation of confidences

Confidences about the position of a TS are naturally encoded in the covariance matrix of each TS by above approach.

We can select or represent a confidence value by inspecting the area defined by the ellipse of the covariance matrix. If we assume a covariance

$$P_k = \begin{pmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{pmatrix}, \tag{7}$$

we can integrate over a desired range and use the resulting value as a confidence estimate (how much of the distributions mass is within this radius), see figure 5. The range can be represented as an integration radius and represents a parameter of the system. It enables queries like selecting a subset of TS from the database, for which 99% of the probability distribution are within a certain radius.

5.3 Evaluation on simulated data

We evaluate the behavior of proposed method on estimated TS simulated according to section 3. Visualizations of sampled TS, the resulting map after integrating all measurements, as well as the behavior of their distance to the simulated GT-TS are visualized in figure 6. It can be observed that even if measurements originate from a distribution with large covariances, the error (distance to GT-sign) rapidly decreases below 1m.

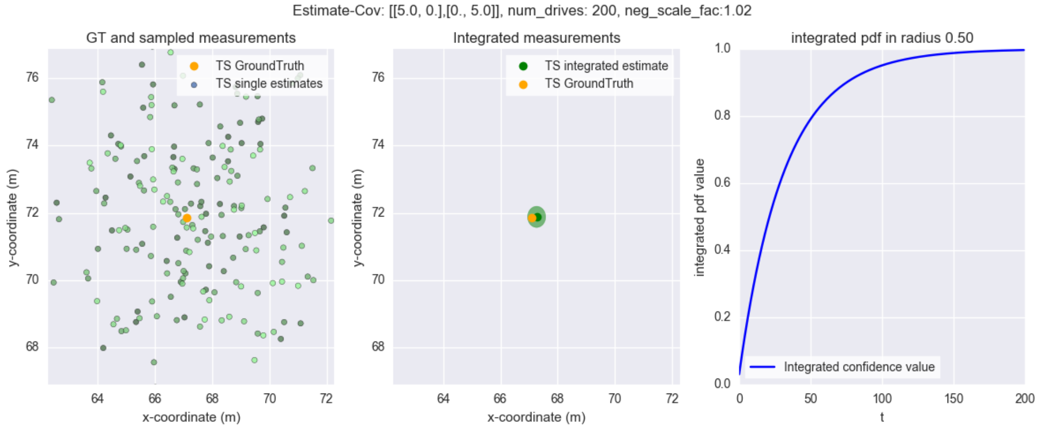

Figure 5: Assuming 200 measurements of the same TS, sampled from a multivariate normal distribution with diagonal covariance ([5,0],[0,5]). Each TS measurement gets assigend the same covariance matrix. The middle plot shows the final integrated measurement with the covariance ellipse of the estimate. The right plot shows the integration of the pdf within a radius of 0.5m.

5.4 Integration of negative measurements

> **Definition:** *Negative measurements* are TS measurements that have been predicted (exist in the database), but not observed.

Several causes for negative TS information exist:

- permanent removal (by authorities)
- temporary removal
- occlusion by static objects (plants, dirt)
- occlusion by dynamic objects (trucks)
- sensor error

An optimal choice on how to integrate negative measurements depends on whether the above factors can be distinguished by the system.

In [TV07], several methods of integrating missing but predicted observations are proposed: 1) reduction of sensor's detection probability, 2) increase of covariance of process-noise, 3) reducing the existence-probability (a probability-measure unrelated to the filter with arbitrarily chosen parameters/values.

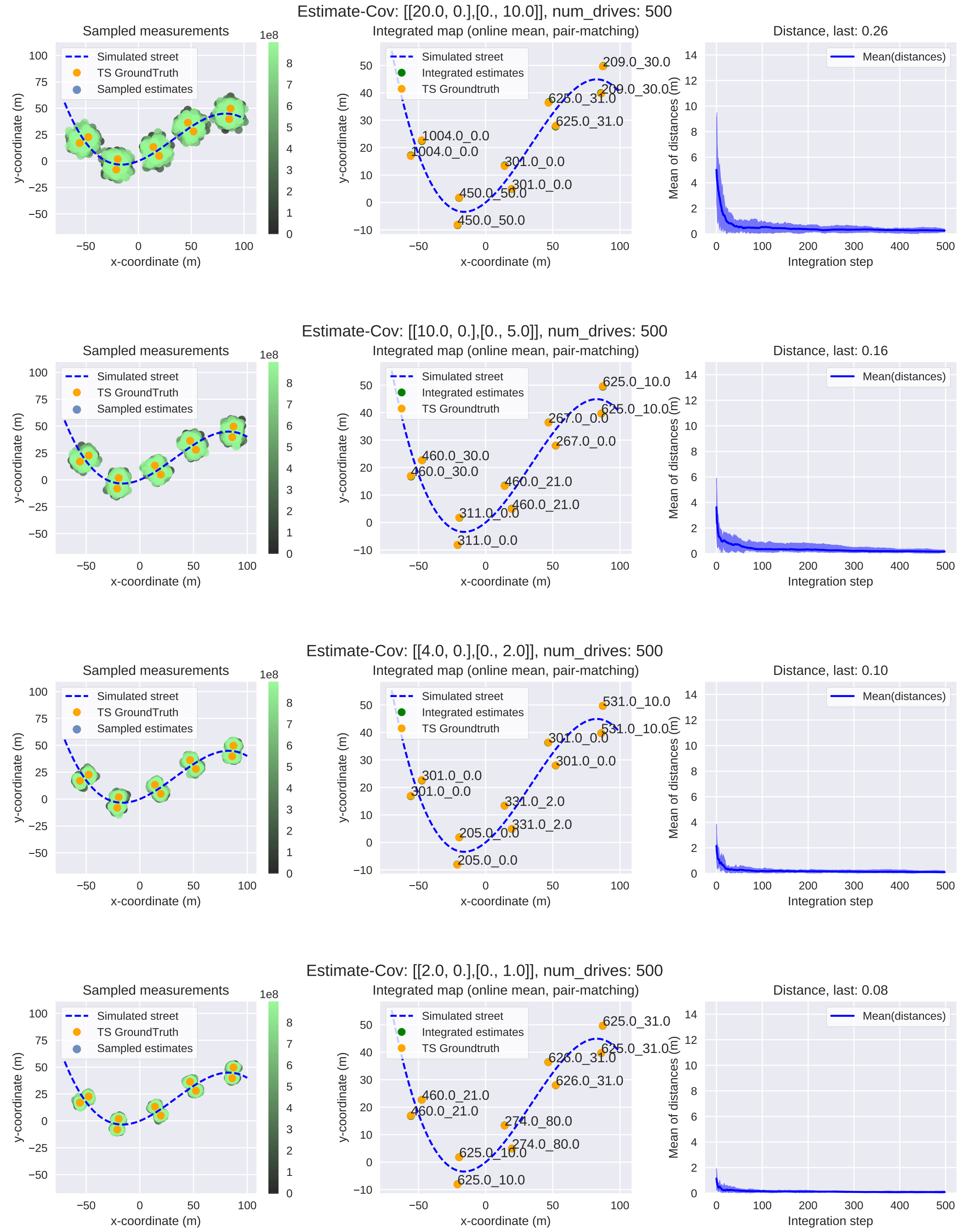

Figure 6: Resulting estimates and their error with respect to GT-TS for different covariances of measurements. In all cases the position of integrated estimated TS converged to the positions of GT-TS with very low uncertainty (middle). The shaded regions in the right plot visualize the maximum/minimum distances of estimated TS to their matching GT-TS.

5.4.1 Approach 1: Multiplicative factor on TS covariance

Integrating a negative measurement by multiplying the covariance matrix of a TS by a factor ϕ:

$$P_k = P_k^- . * \phi \tag{8}$$

Given information about detection probabilities/accuracies of the involved algorithms and probabilities of the causes of negative measurements listed above, as well as the costs of a) predicting a sign that is gone and b) not predicting an existing sign, this factor could be chosen to minimize the involved costs. Sample plots for different values of ϕ can be seen in figure 7.

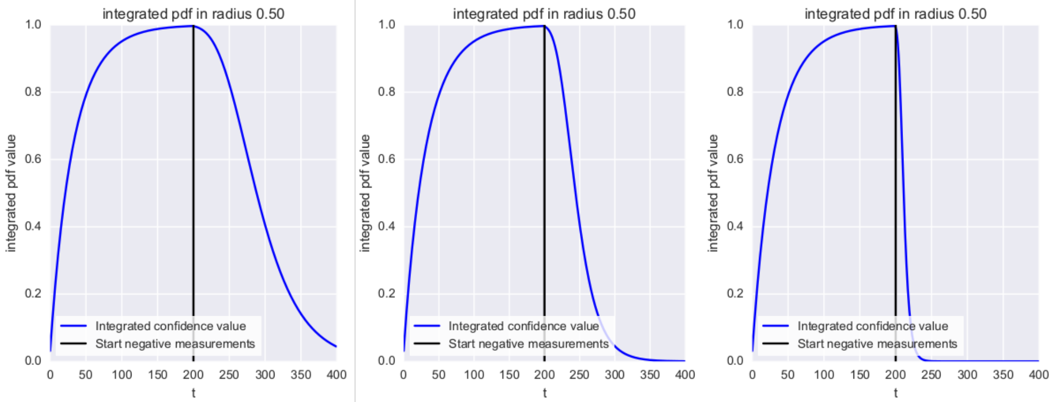

Figure 7: The parameter ϕ can be used to steer the integration of negative information. Large factors enable rapid ”forgetting”, whereas small factors increase the duration for which a sign is maintained despite negative measurements. Simulated values for ϕ are: 1.025 (left), 1.05 (middle), 1.2 (right).

6 Integration and Confidence estimation - Non-parametric

This chapter describes and evaluates methods that can be applied if no covariances or error-models can be derived and input-data is limited.

As before, we assume that input data stems from a distribution. However, we assume that the underlying distribution and error-model is not known.

6.1 Mean integration

Once single and paired TS are identified (as described in Algorithm 1), a possibility of nonparametric integration is computing the new mean for both saved signs and the newly matched estimated sign:

$$\mathtt{ts}_{t+1}^{\mathtt{db}} = \frac{(\sum_{i=0}^{N} \mathtt{ts}_i^{\mathtt{db}}) + \mathtt{ts}_{t+1}^{\mathtt{est}}}{N+1} \tag{9}$$

The variance of the measurements can also be computed and can be used as a measure of confidence:

$$Var(\mathtt{ts}_{\mathtt{t}}^{\mathtt{db}}) = \frac{1}{N}\sum_{i=0}^{N}(\mathtt{ts}_i^{\mathtt{db}} - \mathtt{ts}_t^{\mathtt{db}})^2 \tag{10}$$

6.2 Kernel Density Estimation

A non-parametric way to estimate the probability density function of a random variable.

Let $\mathtt{TS}_t^{\mathtt{est}} = \{\mathtt{ts}_0^{\mathtt{est}}, \mathtt{ts}_1^{\mathtt{est}}, ...\mathtt{ts}_t^{\mathtt{est}}\}$ be the set of all estimated TS until time t, and assume those are samples from an unknown density f. We can estimate the shape of f at every point x using the *kernel density estimator*, where K is a kernel with bandwidth h:

$$\hat{f}_h(x) = \frac{1}{N}\sum_{i=0}^{N} K_h(x - \mathtt{ts}_i^{\mathtt{est}}) \tag{11}$$

Possible kernels are the Gausskernel

$$k(t) := \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2}t^2\right), \tag{12}$$

or the Cauchy-Kernel

$$k(t) := \frac{1}{\pi(1+t^2)} \tag{13}$$

among others (Picard-, Epanechnikov- kernel).

The bandwidth of the kernels is a free parameter that has strong influence on the resulting estimate. Large bandwidths may oversmooth the data (we might aggregate too many signs into one estimated mode), whereas too small bandwidths may undersmooth the data (we might split estimates into different modes, whereas they belong to the same GT-TS.

7 Parameter estimation/optimization

Every parameter ultimately relates back to the uncertainties in the processes of measuring the TS. In order to capture these uncertainties and optimize integration parameters, the displacements of TS to their respective GT-TS can be investigated.

Given a set of estimated traffic signs, $\mathtt{TS}_t^{\mathtt{est}} = \{\mathtt{ts}_0^{\mathtt{est}}, \mathtt{ts}_2^{\mathtt{est}}, ..., \mathtt{ts}_t^{\mathtt{est}}\}$ and the respective reference traffic sign (GT-TS) $\mathtt{ts}^{\mathtt{gt}}$, we can estimate the following quantities:

Without any assumed prior knowledge

1. Groundtruth-facts to determine the integration radius: Minimum distance of two GT-TS of the same type that are not paired
2. Distances of paired GT-TS to determine radius for pair-matching
3. The average error
$$e = \frac{\sum_{i=0}^{N}(\mathtt{ts}_i^{\mathtt{est}} - \mathtt{ts}^{\mathtt{gt}})}{N} \tag{14}$$
which can help in determining an integration radius, selecting the bandwidth-parameter of a kernel-density-estimator as presented in section 6.
4. The distribution of estimated TS by using distribution fitting techniques like *maximum likelihood methods*, the *method of moments* or the *method of L-moments*. If the estimated data is distributed according to a matching distribution, this information can be used to select appropriate parameters for the parametric integration and confidence estimation presented in section 5.
5. Given a subset of estimates that can be distinguished due to sensor-characteristics and -brands or system-variants, their specific error-characteristics (as listed in Table 1: gps,sync,sensor-specific characteristica, weather) can be calculated and used to compensate for biases in future integrations. An example of systematic bias can be found in Figure 8. Given the cars heading direction at the time of measurement, a systematic bias can be calculated by analyzing the mean offset to the reference TS.

(Normally) distributed measurements

1. Estimate the covariance and shape of the underlying distribution profile using (weighted) least squares algorithms. These are the parameters needed to perform an optimal Bayesian integration and confidence estimation as presented in section 5.

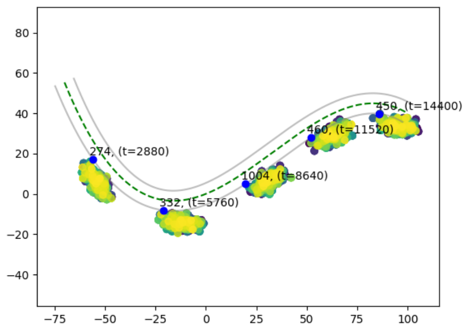

Figure 8: Biased measurements

8 Appendix - Error sources

GPS uncertainties([Lan99], [Gl]):

Position errors are caused mainly by GPS- and measurement uncertainty. Common perturbations of a GPS position are:

- Atmospheric effects (molecules in trophosphere affect electromagnetic signal transmission)
- Multipath effects (reflection of signal, difference in pathlength causes signals to interfere with each other and contribute to errors in the pseudorange observables)
- Satellite geometry (dilution of precision, modulates other errors)
- Measurement noise (obstacles, satellite blocks)
- Ephemeris data (list of a i single satellites positions (in comparison to almanac that is a rough prediction of all satellites orbits) as a function of time, contains locations computed from orbit measurements, along with corrections. Orbital position errors may be present in ephemeris data, causing errors in calculated positions.)
- Satellite clock drift

The only source of error that can reliably be calculated is the DOP (dilution of precision). This can be split up into the following factors: PDOP (position dilution of precision) is a unitless measure that refers to the quality of horizontal (HDOP) and vertical (VDOP) measurements. A low PDOP indicates a higher probability of position accuracy.

$$PDOP^2 = HDOP^2 + VDOP^2 \tag{15}$$

US government accuracy specification: 7.8 meters (95%), but the satellite geometry (DOP) can magnify or reduce the effects of other GPS errors.

8.0.1 GPS/Camera synchronization

If the framerate of the camera (and therefore the visual detections) and the GPS-updates are not synchronized, this will introduce an additional error.

9 Appendix: Past experiments

9.1 Kernel density estimation

We can use a kernel density estimation to calculate confidence values for the locations/existence of sings: As we are working with geospatial data in the WGS84 coordinate system, we choose haversine as metric and consequently the ball_tree algorithm. However, it remains open how to find an optimal bandwidth-parameter for this technique to both separate single signs in pairs and still integrate measurements that are further apart. Global scaling of resulting probabilities influences confidences of single signs. KDE has therefore not been considered futher.

Figure 9: The result of the kernel density estimation on estimated traffic signs

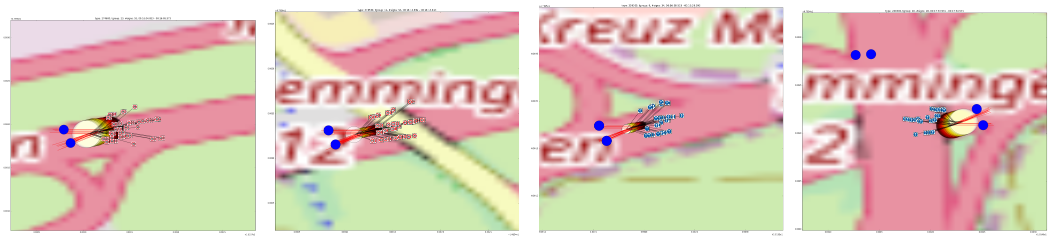

Figure 10: Initial data-integration (icons and black lines) and corrected version (red) after heading-integration. Blue circles denote the ground-truth-position, the other ellipses visualize the GPS-uncertainty as well as the timestamp (black is older, white is newer. In the old integration, the measurements may point "backwards" considering the driving direction, which is fixed by the proposed approach.

10 Appendix - Past and intermediate data, preprocessing and analysis

10.1 Estimated data format II (legacy)

Format of estimated signs ([2]) stayed the same as in 10.2, but additional data of corresponding GPS-data ([3]) as well as raw sign-measurements (unfused single-frame measurements) ([4]) have been supplied.

Thanks to the GPS-data, it has been possible to detect a flaw in the conversion of measurements from the camera-coordinate-system to a global coordinate system (the heading of the car did not seem to be considered in this transformation). Figure 10 displays some examples of the expression of this as well as the proposed corrections.

10.2 Estimated data format I (legacy)

Estimated data is supplied by another group in the XML-format specified in section 10.2.1. Coordinates are already in WGS84 ([5]).

10.2.1 XML

The XML structure for TS is as follows:

```
<Signs>
  <Sign id="000" subtype="000" lat="0.000000" lon="0.000000" height="0.0000" conf_id="0.00" ts="0" />
...
</Signs>
```

[2] Seq_2016_11_23_12h53m24s_Filt.xml
[3] Seq_2016_11_23_12h53m24s_gpsCAN.xml
[4] Seq_2016_11_23_12h53m24s_Raw.xml
[5] cap_20160602102549_Signs_2.xml

The entries have the following semantics:

Name	**Semantic**
id	TS code
subtype	Subtype of TS code (e.g. value of speedlimit)
lat	Latitude in WGS84
lon	Longitude in WGS84
height	Height above ground plane in meters
conf_id	Confidence in the type of the sign
ts	Timestamp of estimate in Microseconds since capture start

10.3 Pre-processing

As a first step, associated consecutive measurements (estimate clusters) of the same TS need to be identified and grouped.

A single estimate has the following structure:

$$e(TS) = \begin{bmatrix} lat \\ lon \\ height \\ c \\ sc \\ ts \end{bmatrix} \tag{16}$$

We first divide the set of all TS estimates ($e(TS)$) into the respective sign-type clusters (SC) based on their code (c) and sub-code (sc) (see Figure 14, bottom): To identify estimates that belong together (consecutive in time, belonging to the same TS), we further identify temporal groups in these sign-type clusters by grouping estimates based on their timestamp. Based on assumptions about ego-speed of the capturing vehicle, the capturing framerate and the spacing of traffic-signs, it can be assumed that estimates of which the temporal difference is bigger than 2 seconds belong to different TS. The resulting set of estimate clusters (EC) contains temporally grouped estimates of the same type and subtype (see Figure 11).

$$EC_{c,sc,t} = \{e(TS)_{c,sc,t} | c^i == c^j, sc^i == sc^j, t^i - t^j < 2000000 \ \forall i, j \in |\{e(TS)\}|\} \tag{17}$$

Some signs appear in pairs (left and right side of the road). The DBSCAN (ball-tree algorithm with haversine distance metric) algorithm can be used to further spatially cluster estimates (eps has to be set to value smaller than road-width and larger than the spacing between estimates of the same TS, here: 7m).

To get a final estimate of each cluster, we can use the mean of the coordinates:

$$\mathtt{fe}_{c,sc,t} = \frac{\sum e(TS) \in E_{c,sc,t}}{|E_{c,sc,t}|} \tag{18}$$

Figure 11: Clustering pipeline: 1) all estimates, 2) estimates with same code and subcode, 3) temporal estimate clusters 4) spatial groups

10.4 Matching

Goal of this chapter is to identify

1. if reference data for a final estimate `fe` exists, and if this is the case,
2. calculate the distance between both

Reference data has the following structure:

$$r(TS) = \begin{bmatrix} lat \\ lon \\ height \\ c \\ sc \end{bmatrix} \tag{19}$$

The goal is now to identify a reference TS ($r(TS)$) with same code and subcode and minimal distance to each final estimation `fe`:

$$min(haversine(r(TS)_{c,sc}, fe(TS)_{c,sc,t}))\forall c, sc, t \tag{20}$$

In the current implementation, we can search the set of all reference TS with same code and subcode, later on this can be implemented using geospatial algorithms directly on a database. The result of all final estimates can be seen in Figure 12. Detailed visualizations can be seen in Abstract I.

10.5 Measurement errors

Out of 99 final estimates, the following 61 signs have a matching reference TS with a distance below 100m:

Description	Count
Arrow right (white on blue)	9 times
Speed lim 80	28 times
Speed lim 100	6 times
Arrow straight (white on blue)	11 times

Figure 13 shows the histogram of minimal distances of the final estimates to the nearest reference TS.

10.5.1 Analysis

(Figure 14, top) shows an overlay of the given reference data on a map of OpenStreetMap, whereas (Figure 14, middle) displays all estimates and (Figure 14, bottom) shows the estimates for one specific TS type and subtype (Speed limit:80).

Regarding the timestamps of the data, it becomes obvious that

- several estimates exist for a single TS
- the estimates exhibit a spread along the driving direction, indicating that there are errors either in distance-computation, ego-speed- or GPS-compensation.

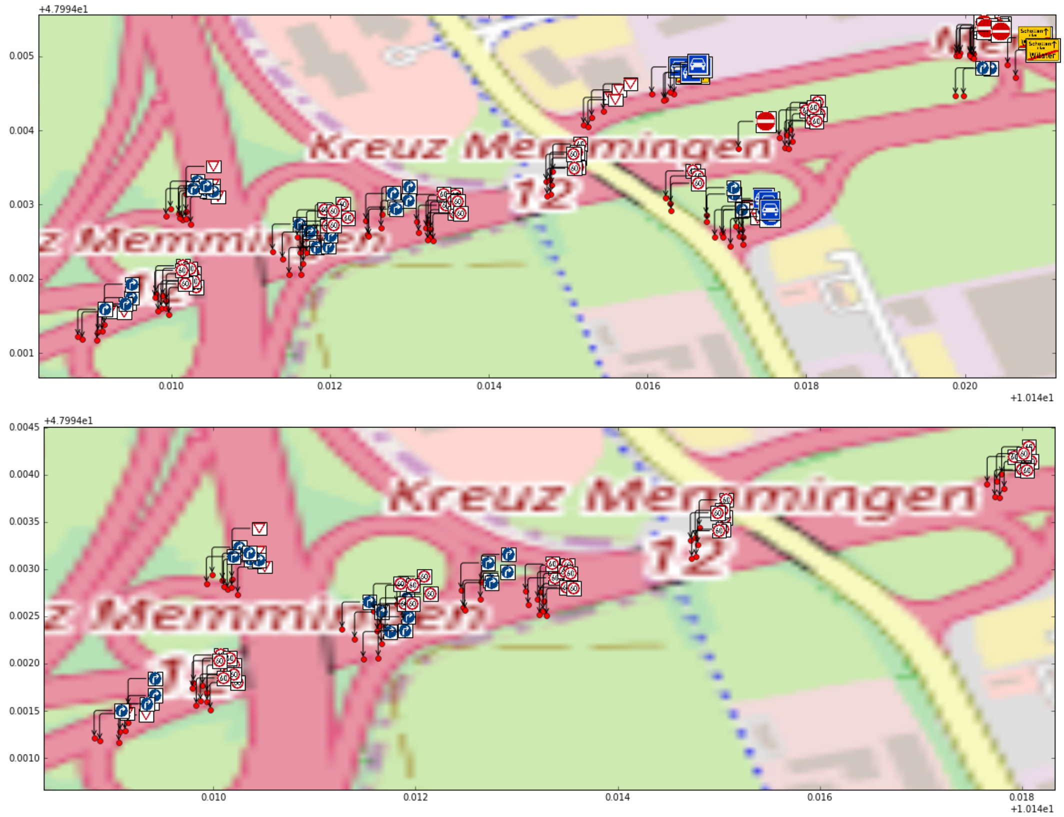

Figure 12: Final estimates, all (top) and with (bottom) matching reference TS

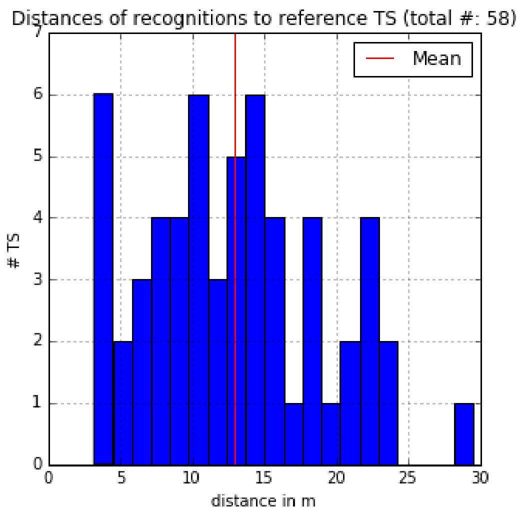

Figure 13: Histogram of the minimal distances of final estimates to nearest reference TS

- some tracks have been driven several times, resulting in multiple estimate-groups per TS

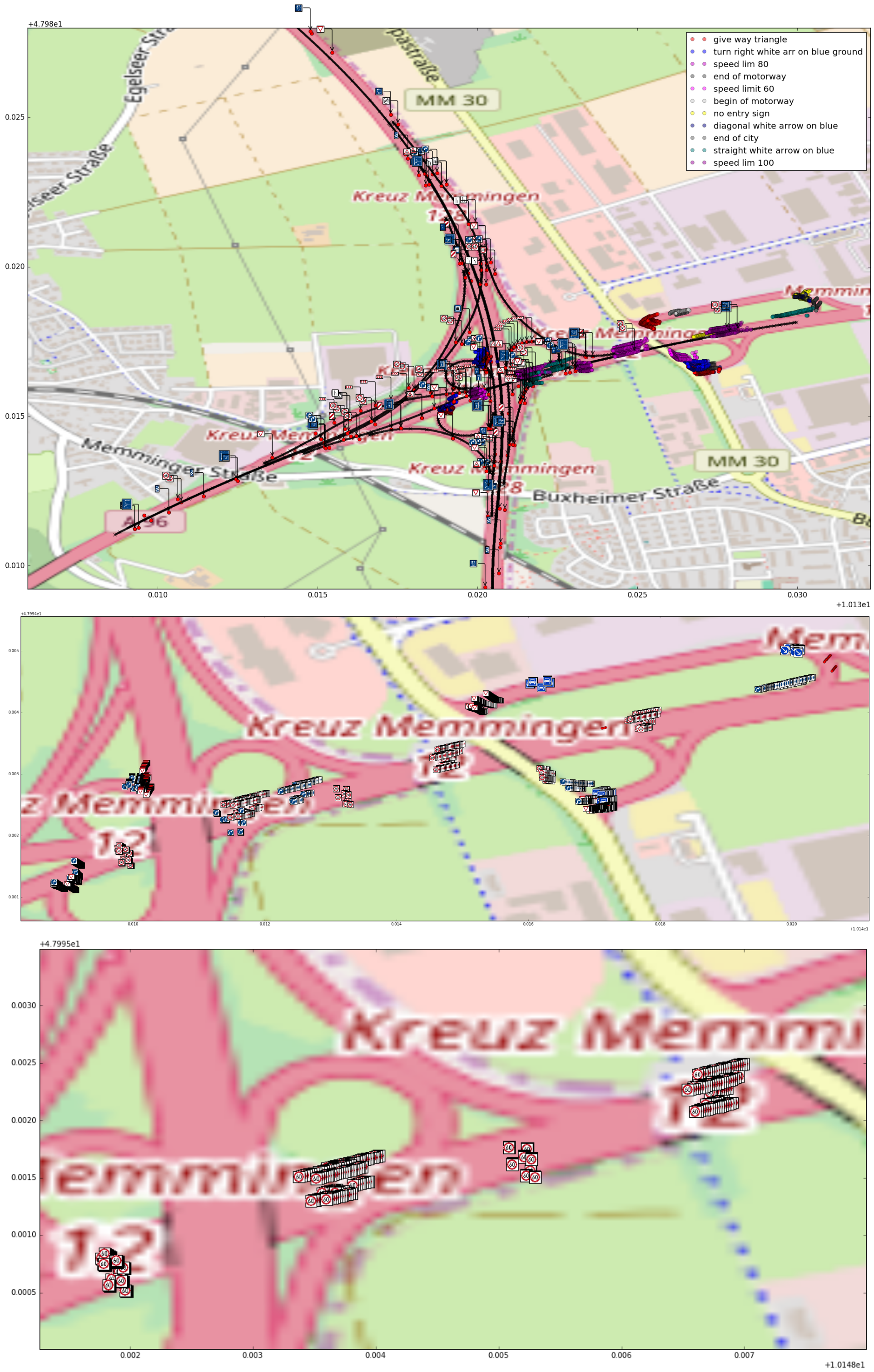

Figure 14: Top: Reference data overlayed on OSM map, middle: Estimated data overlayed on OSM map, bottom: Estimates for TS ”Speed Limit: 80”

References

[D+15] Dupuis, M u. a.: OpenDRIVE Format Specification. In: *VIRES Simulationstechnologie GmbH* (2015)

[Gl] GPS laboratory, Montana State U.: Global Positioning System. http://www.montana.edu/gps/slides/2GPSAccuracy.pdf

[Lan99] Langley, Richard B.: Dilution of precision. In: *GPS world* 10 (1999), Nr. 5, S. 52–59

[TV07] Tischler, Karin ; Vogt, Heike S.: A sensor data fusion approach for the integration of negative information. In: *Information Fusion, 2007 10th International Conference on* IEEE, 2007, S. 1–7