Find your Way by Observing the Sun and Other Semantic Cues

Wei-Chiu Ma[1] Shenlong Wang[2] Marcus A. Brubaker[2] Sanja Fidler[2] Raquel Urtasun[2]

[1]Carnegie Mellon University [2]University of Toronto

Appendices

A. Solar Positioning

In this section, we detail how to compute the solar position given an approximately geo-tagged image with time-stamp. This is to automatically compute the ground truth annotations for training our Sun-CNN. Given the camera geolocation, its orientation as well as the time that the image was taken, we can estimate the position of the sun in camera coordinates. In particular, for KITTI, GPS provides geolocation, time-stamp are recorded in EXIF and camera orientation is captured from IMU. The pipeline for conducting solar positioning has two steps. First, given the geolocation and timestamp, we apply the standard solar positioning algorithm to estimate the solar position in local coordinates, centered at the GPS location with the x and y axis being north-south and west-east respectively. After that we use the camera orientation to transform the local coordinates to polar camera coordinates.

We now describe the solar positioning algorithm in more detail. We first transform the coordinates to Julian day using (Eq.. 4 - Eq. 7 in [2]). After that, we calculate the earth heliocentric position, *i.e.*the earth center's position in solar coordinate system, according to the Julian day (Eq. 8 - Eq. 12 in [2]). We can then obtain the solar geocentric longitude and latitude, *i.e.*sun's position in earth-centered coordinate system. The axis direction is chosen as Greenwich location (Eq. 13 - Eq. 14 in [2]). Note that obliquity of ecliptic and nutation need to be considered to do angle correction (Eq. 15 - Eq. 31). Given the geo-location and time of the day, we then transform the sun's geocentric position to topocentric position, *i.e.*sun's position in local coordinate (Eq. 41 - Eq. 47 in [2]). Finally, we transfer the local coordinates into camera coordinates according to the camera orientation. This gives us the sun's azimuth and zenith angles with respect to the camera coordinates. Figs. 1 and 2 depicts the overview of our algorithm.

B. Inference Details

Below we briefly summarize the algorithmic details of inference which mostly follow [**?**]. As noted in the main paper, the continuous distribution of street parameters $\mathbf{s}_t$ for each street segment u_t is represented using a Mixture of Gaussians, i.e.,

$$p(\mathbf{s}_t|u_t, \mathbf{y}_{1:t}) = \sum_{i=1}^{N_{u_t}} \omega_{u_t}^{(i)} \mathcal{N}(\mathbf{s}_t|\mu_{u_t}^{(i)}, \Sigma_{u_t}^{(i)}) \tag{1}$$

where N_{u_t} is the number of components for the mixture associated with u_t and $\mathcal{M}_{u_t}^t = \{(\omega_{u_t}^{(i)}, \mu_{u_t}^{(i)}, \Sigma_{u_t}^{(i)})\}_{i=1}^{N_{u_t}}$ are the parameters of the mixture for u_t. As observations arrive, these continuous distributions are updated along with the discrete distribution over street segments $p(u_t|\mathbf{y}_{1:t})$ as described in Alg. 1. The recursive updates for each component of each mixture model are similar to those used for Kalman filtering, and consist of a prediction or propagation step (Alg. 2) which uses the state transition model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and an update or correction step (Alg. 3) which uses the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. Due to the nonlinearity of the street node transitions, the number of mixture model components can grow in a potentially unbounded way. To deal with this, a mixture model simplification procedure (Alg. 4) is used as needed. This simplification procedure is based on removing components from the original mixture model $\mathcal{M}$ one at a time and updating the parameters of the remaining components. Components parameters are updated to minimize a variational upper bound $\hat{D}(\phi, \psi, \mathcal{M}, \hat{\mathcal{M}})$ to the KL divergence $D(\mathcal{M}\|\hat{\mathcal{M}})$ where $\hat{\mathcal{M}}$ is the reduced set of mixture components and ϕ and ψ are variational parameters. Components are removed until the (upper bound on) KL divergence of the updated mixture model with the original exceeds a threshold $\epsilon = 10^{-2} nats$. For the full derivation of the inference and simplification algorithms see [**?**].

Compared to [**?**] the main difference in the model for inference is the likelihood function

$$p(\mathbf{y}_t|\mathbf{x}_t) = p(\mathbf{o}_t|\mathbf{x}_t)p(s_t|\mathbf{x}_t)p(i_t|\mathbf{x}_t)p(r_t|\mathbf{x}_t)p(v_t|\mathbf{x}_t) \tag{2}$$

where $\mathbf{o}_t$, s_t, i_t, r_t and v_t are the odometry, sun direction, detected intersection type, detected road type and vehicle

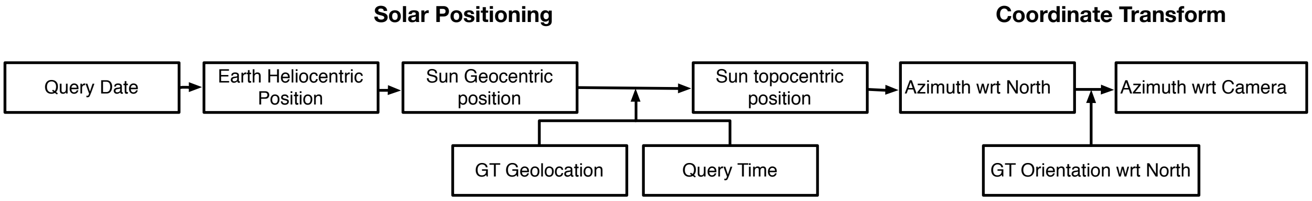

Figure 1: Pipeline for sun direction estimation from a geo-tagged image with known time-stamp and camera orientation.

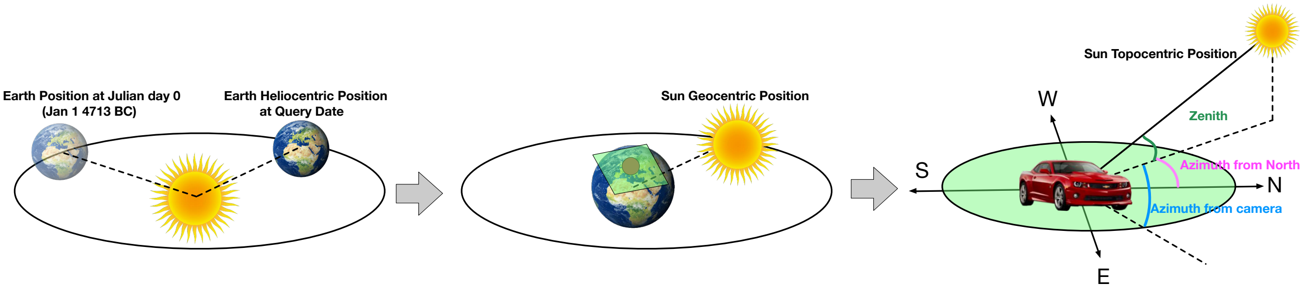

Figure 2: Illustration of how the coordinate transform is conducted. Left to right: first the earth's heliocentric position in solar coordinate is computed; it is then transform to the sun's geocentric position in earth-centered coordinates; Finally the sun's topocentric position in local coordinate and camera coordinates can be estimated.

velocity respectively. Because these terms are all either constant as a function of $\mathbf{x}_t$ (intersections, road type, velocity) or Gaussian as a function of $\mathbf{x}_t$ (odometry, sun direction) then the full likelihood function can be expressed simply as

$$p(\mathbf{y}_t|\mathbf{x}_t) = c_{obs}\mathcal{N}(\bar{\mathbf{y}}|\bar{\mathbf{M}}\mathbf{s}_t, \bar{\Sigma})$$

where $c_{obs} = p(i_t|\mathbf{x}_t)p(r_t|\mathbf{x}_t)p(v_t|\mathbf{x}_t)$, $\bar{\mathbf{y}} = (\mathbf{o}_t^T, s_t^T)^T$, $\bar{\mathbf{M}} = (\mathbf{M}_{u_t}^T, \mathbf{m}_s)^T$, and $\bar{\Sigma}) = diag(\Sigma^{\mathbf{o}}_{u_t}, \Sigma^{\mathbf{s}})$. Thus, the difference to the inference algorithm in [?] is the multiplication of the additional constant term c_{obs} and the use of the combined observation vector $\bar{\mathbf{y}}$, matrix $\bar{\mathbf{M}}$ and covariance $\bar{\Sigma}$. These changes are reflected in the update or correction step of inference which can be seen in Alg. 3.

C. Experimental Evaluation

In this section, we perform extensive quantitative evaluation in terms of computation time and the size of the potential solution space. We also show additional qualitative/quantitative results on localization.

As shown in Fig. 3, adding semantic cues may increase the computation time at the very beginning. This is because the more observations we consider, the more computation we need to perform during inference given the same number of modes (*i.e.* computing the likelihood function in the *update* step). Fortunately, with the help of semantic cues, we can drastically reduce the potential solution space after a few seconds and thus speed up the computation process. Fig. 4 shows how the potential solution space changes with time when employing different semantic cues. We can observe that when employing *all* semantics cues, our model

Algorithm 1 Filter

```
Input: Posterior at $t-1$, $\{P_u^{t-1}, \mathcal{M}_u^{t-1}\}$
Input: Observation at $t$, $\mathbf{y}_t$
Initialize mixtures, $\mathcal{M}_u^t \leftarrow \varnothing$, for all $u$
for all streets $u_{t-1}$ do
    for all streets $u_t$ reachable from $u_{t-1}$ do
        $\mathcal{M}' \leftarrow \varnothing$
        for all $(\omega, \mu, \Sigma) \in \mathcal{M}_{u_{t-1}}^{t-1}$ do
            Compute $c_{pred}\mathcal{N}(\mu_{pred}, \Sigma_{pred})$ using Alg 2
            Compute $c_{upd}\mathcal{N}(\mu_{upd}, \Sigma_{upd})$ using Alg 3
            $\mathcal{M}' \leftarrow \mathcal{M}' \bigcup \{(c_{upd}, \mu_{upd}, \Sigma_{upd})\}$
        if $u_t \neq u_{t-1}$ then
            Compute $(c, \mu, \Sigma)$ to approximate $\mathcal{M}'$
            $\mathcal{M}_{u_t}^t \leftarrow \mathcal{M}_{u_t}^t \bigcup \{(c, \mu, \Sigma)\}$
        else
            $\mathcal{M}_{u_t}^t \leftarrow \mathcal{M}_{u_t}^t \bigcup \mathcal{M}'$
for all streets $u$ do
    $P_u^t \leftarrow \sum_{(c,\mu,\Sigma)\in\mathcal{M}_u^t} c$
    $\mathcal{M}_u^t \leftarrow \{(\frac{c}{P_u^t}, \mu, \Sigma) \mid (c, \mu, \Sigma) \in \mathcal{M}_u^t\}$
    if $\frac{\ell_u}{|\mathcal{M}_u^t|} < 10$ meters then
        Simplify $\mathcal{M}_u^t$ with Algorithm 4
Normalize $P_u^t$ so that $\sum_u P_u^t = 1$.
For all $u$, if $P_u^t < 10^{-50}$ set $P_u^t \leftarrow 0$ and $\mathcal{M}_u^t \leftarrow \varnothing$
Return: Posterior at $t$, $\{P_u^t, \mathcal{M}_u^t\}$
```

Algorithm 3 Update Step

```
Input: Current mode $(c_{pred}, \mu_{pred}, \Sigma_{pred})$ on street $u_t$
Input: Observation $\mathbf{y}_t$
Construct the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t) = c_{obs}\mathcal{N}(\bar{\mathbf{y}}|\bar{\mathbf{M}}\mathbf{s}_t, \bar{\Sigma})$
$c_{obs} \leftarrow p(i_t|\mathbf{x}_t)p(r_t|\mathbf{x}_t)p(v_t|\mathbf{x}_t)$
$\bar{\mathbf{y}} \leftarrow (\mathbf{o}_t^T, s_t^T)^T$
$\bar{\mathbf{M}} \leftarrow (\mathbf{M}_{u_t}^T, \mathbf{m}_s)^T$
$\bar{\Sigma} \leftarrow diag(\Sigma_{u_t}^{\mathbf{o}}, \Sigma^{\mathbf{s}})$
Compute updated Gaussian parameters
$\Sigma_{upd} \leftarrow \left(\bar{\mathbf{M}}^T\bar{\Sigma}^{-1}\bar{\mathbf{M}} + \Sigma_{pred}^{-1}\right)^{-1}$
$\mu_{upd} \leftarrow \Sigma_{upd}\left(\bar{\mathbf{M}}^T\bar{\Sigma}^{-1}\bar{\mathbf{y}} + \Sigma_{pred}^{-1}\mu_{pred}\right)$
$\Sigma'_{upd} \leftarrow \Sigma_{u_t}^{\mathbf{y}} + \bar{\mathbf{M}}\Sigma_{pred}\bar{\mathbf{M}}^T$
$c_{upd} \leftarrow \frac{\omega c_{obs} c_{pred} |\Sigma_{upd}|^{0.5}}{|\Sigma_{pred}|^{0.5}|\bar{\Sigma}|^{0.5}} \exp\left(-\frac{1}{2}\|\bar{\mathbf{y}} - \bar{\mathbf{M}}\mu_{pred}\|_{\Sigma'_{upd}}^2\right)$
Return: Updated mode $(c_{upd}, \mu_{upd}, \Sigma_{upd})$.
```

Algorithm 4 GMM Simplification

```
Input: Mixture model parameters $\mathcal{M} = \{(\omega_j, \mu_j, \Sigma_j)\}$ and approximation threshold $\epsilon$
Initialize $\mathcal{M}' = \mathcal{M}$
loop
    Select a component to remove $\hat{b} = \arg\min_{b'}\omega_{b'}$
    $\hat{\mathcal{M}} \leftarrow \mathcal{M}' \setminus \{(\omega_{\hat{b}}, \mu_{\hat{b}}, \Sigma_{\hat{b}})\}$
    Initialize the variational parameters $\phi$, $\psi$ and $\hat{D} \leftarrow \infty$
    while $\hat{D} \geq \epsilon$ and not converged do
        Minimize $\hat{D}(\phi, \psi, \mathcal{M}, \hat{\mathcal{M}})$ with respect to $\phi$, $\psi$ and $\hat{\mathcal{M}} = \{(\hat{\omega}_i, \hat{\mu}_i, \hat{\Sigma}_i)\}$
        for all mixture components $i$ in $\hat{\mathcal{M}}$ do
            for all mixture components $j$ in $\mathcal{M}$ do
                Compute Gaussian KL divergences
                $D_{j,i} \leftarrow D(\mathcal{N}(\mu_j, \Sigma_j)\|\mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i))$
            for all mixture components $j$ in $\mathcal{M}$ do
                Update variational parameters
                $\psi_{j,i} \leftarrow \hat{\omega}_i \frac{\phi_{j,i}}{\sum_{j'} \phi_{j',b}}$
                $\phi_{j,i} \leftarrow \omega_j \frac{\psi_{j,i}\exp(-D_{j,i})}{\sum_{i'} \psi_{j,i'}\exp(-D_{j,i})}$
            Update components of $\hat{\mathcal{M}}$
            $\hat{\omega}_i \leftarrow \sum_j \phi_{j,i}$
            $\hat{\mu}_i \leftarrow \frac{\sum_j \phi_{j,i}\mu_j}{\sum_j \phi_{j,i}}$
            $\hat{\Sigma}_i \leftarrow \frac{\sum_j \phi_{j,i}\left(\Sigma_j + (\mu_j - \hat{\mu}_i)(\mu_j - \hat{\mu}_i)^T\right)}{\sum_j \phi_{j,i}}$
        $\hat{D} \leftarrow \sum_{i,j}\left(\log\frac{\phi_{j,i}}{\psi_{j,i}} + D_{j,i}\right)$
    if $\hat{D}(\phi, \psi, \mathcal{M}, \hat{\mathcal{M}}) \geq \epsilon$ then
        Return: $\mathcal{M}'$
    else
        $\mathcal{M}' \leftarrow \hat{\mathcal{M}}$
```

reduces the solution space much faster and leads to a shorter amount of computation time than considering only odometry [1] on *all* sequences. One should also note that adding a single semantic cue may sometimes result in a longer computation time, which implies that the semantic cue is not helpful in such scenario and cannot effectively reduce the search space. Take sequence 07 for example, the vehicle is driving in an urban area where there are no highways

Algorithm 2 Prediction Step

```
Input: Parameters of current mode $\mu, \Sigma$
Input: Street nodes $u_t, u_{t-1}$
if $\|\frac{d}{d\mu} g(\mu, \Sigma)\| < \eta$ then
    Analytically approximate $c_{pred}\mathcal{N}(\mu_{pred}, \Sigma_{pred})$
    $c_{pred} \leftarrow p(u_t | u_{t-1}, \mathbf{s}_{t-1} = \mu)$
    $\mu_{pred} \leftarrow \mathbf{A}_{u_t, u_{t-1}}\mu + \mathbf{b}_{u_t, u_{t-1}}$
    $\Sigma_{pred} \leftarrow \Sigma^{\mathbf{s}}_{u_t} + \mathbf{A}_{u_t, u_{t-1}}\Sigma\mathbf{A}^T_{u_t, u_{t-1}}$
else
    Sample to compute $c_{pred}\mathcal{N}(\mu_{pred}, \Sigma_{pred})$
    for $j = 1, \ldots, M$ do
        $\mathbf{s}^{(j)}_{t-1} \sim \mathcal{N}(\mu, \Sigma)$
        $\mathbf{s}^{(j)}_{t} \leftarrow \mathbf{A}_{u_t, u_{t-1}}\mathbf{s}^{(j)}_{t-1} + \mathbf{b}_{u_t, u_{t-1}}$
        $w^{(j)} \leftarrow p(u_t | u_{t-1}, \mathbf{s}_{t-1} = \mathbf{s}^{(j)}_{t-1})$
    $c_{pred} \leftarrow M^{-1} \sum_{j=1}^{M} w^{(j)}$
    $\mu_{pred} \leftarrow (Mc_{pred})^{-1} \sum_{j=1}^{M} w^{(j)}\mathbf{s}^{(j)}_{t}$
    $\Sigma_{pred} \leftarrow \Sigma^{\mathbf{s}}_{u_t} + \sum_{j=1}^{M} w^{(j)} \frac{(\mathbf{s}^{(j)}_{t} - \mu_{pred})(\mathbf{s}^{(j)}_{t} - \mu_{pred})^T}{Mc_{pred}}$
Return: Predicted mode $(c_{pred}, \mu_{pred}, \Sigma_{pred})$
```

nearby, therefore adding *road type* not only doesn't help (since the mis-classification of our Road-Type-CNN will even increase the uncertainty) but also increase the computation time (as there are two likelihoods to compute per frame). As shown in Tab. 1, while some semantic cues may occasionally be problematic, when all cues are combined, our inference algorithm is able to cope with these errors and improves significantly. Even when we do not use the sun cue (assuming sun is invisible due to weather/time constraint), we still outperform [1]. The comparison between our full model (considering *all* cues) and [1] can be found in Fig. 5. Tab. 1 shows detailed ablation studies.

We also demonstrate the effectiveness of our full model by performing inference on the full city map, which contains over 4500km of roads. From 6, we can observe that the localization results are comparable to those using sub-region maps, implying that the semantic cues we adopted are simple yet highly discriminative. Tab. 1 shows quantitative results when considering different combination of semantic cues. One should note that the reported localization time/computation time may be different from [1] since the map we used is much more complete than [1]. While [1] pruned dirt roads and alleyways during the preprocessing stage based on human prior knowledge, we preserve all drivable roads in the map. This makes the localization problem more difficult. The comparison between two maps is shown in Fig. 7 for a sub-region. We can observe that the map we employed has a larger region and contains more road. In particular, we use double the amount of roads.

References

[1] M. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *CVPR*, pages 3057–3064. IEEE, 2013.

[2] I. Reda and A. Andreas. Solar position algorithm for solar radiation applications. *Solar energy*, 76(5):577–589, 2004.

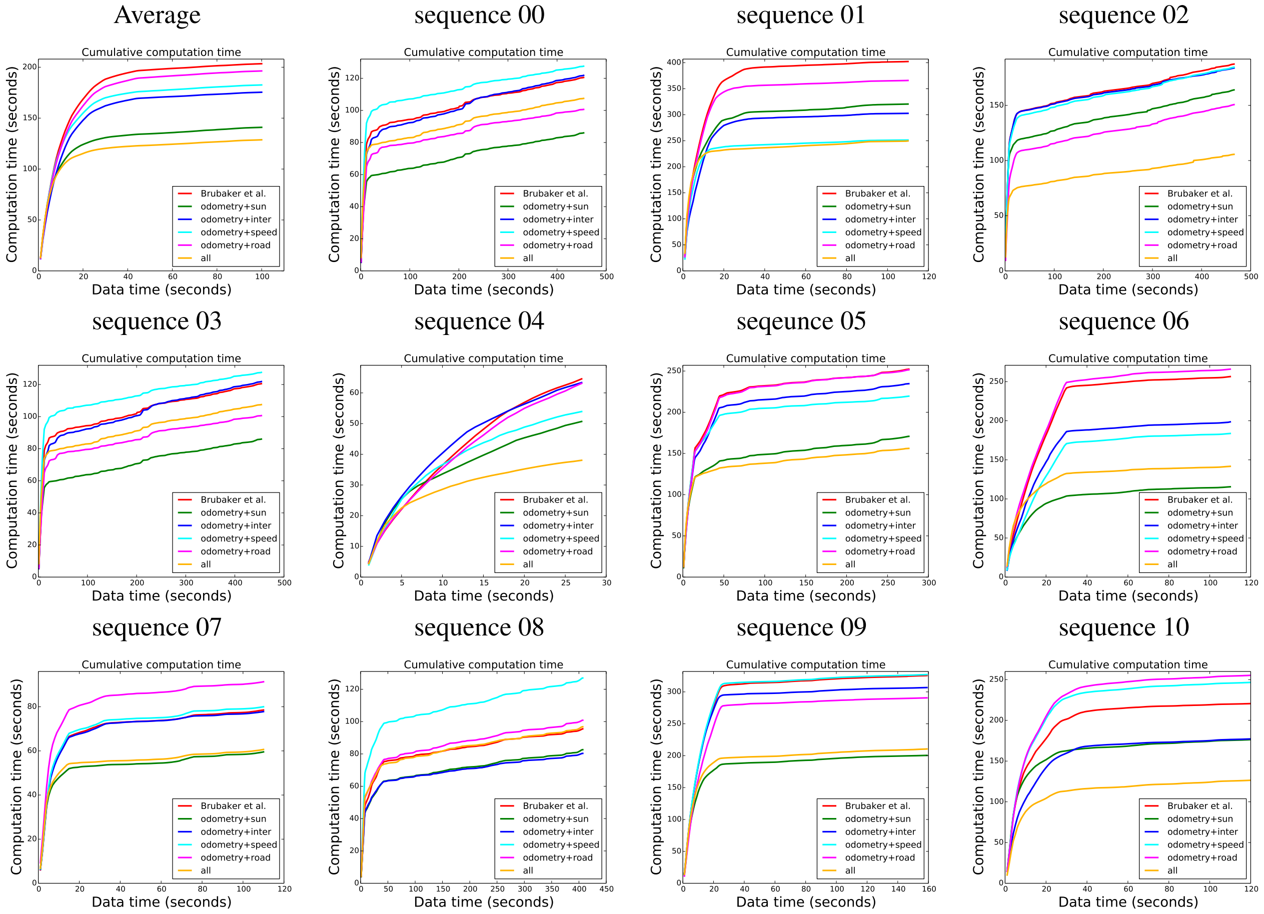
Average
sequence 00
sequence 01
sequence 02
sequence 03
sequence 04
seqeunce 05
sequence 06
sequence 07
sequence 08
sequence 09
sequence 10

Figure 3: **Cumulative computation time:** We show the computation time required when employing different semantic cues, where red lines consider only visual odometry; green sun+odometry, blue intersection+odometry, cyan velocity+odometry, magenta road type+odometry, and orange exploit all semantic cues. With the help of semantic cues, we can drastically speed up the computational process. One should note that employing *all* semantics cues leads to a shorter amount of computation time than considering only odometry [1] on *all* sequences. We run our program on 16 cores with a basic Python implementation. Although our inference is slower than real-time in some sequences, we argue that we can reduce it by employing more computational resource (*i.e.* using more cores). We compute the average by using only the sequences that all methods localize.

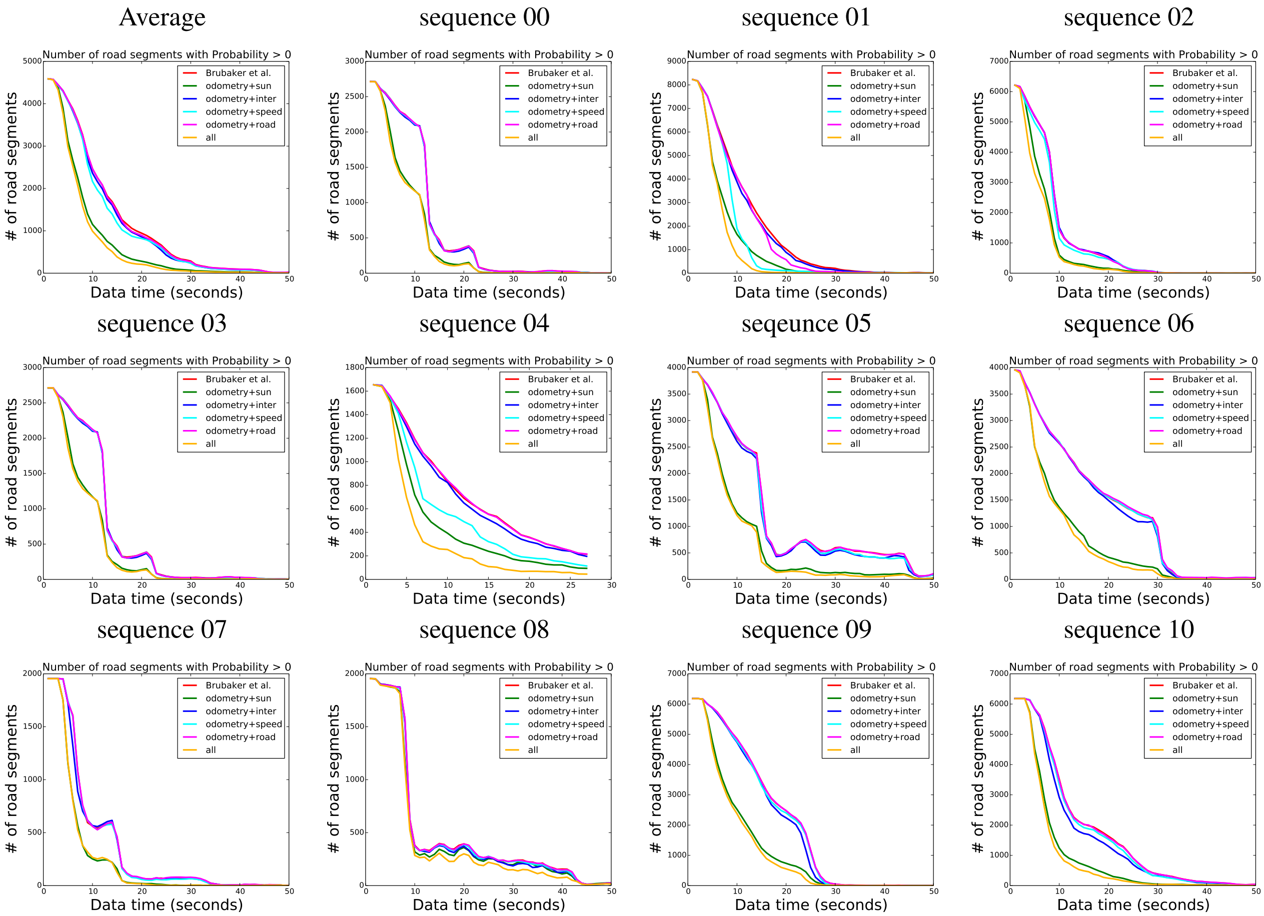

Figure 4: **Road segments with probability larger than 0.** We show the number of road segments (3m) with probability larger than 0 when employing different semantic cues, where red lines consider only visual odometry; green sun+odometry, blue intersection+odometry, cyan velocity+odometry, magenta road type+odometry, and orange exploit all semantic cues. With the help of semantic cues, we can drastically reduce the potential solution space. One should note that employing *all* semantics cues reduces the uncertainty much faster than considering only odometry [1] on *all* sequences.

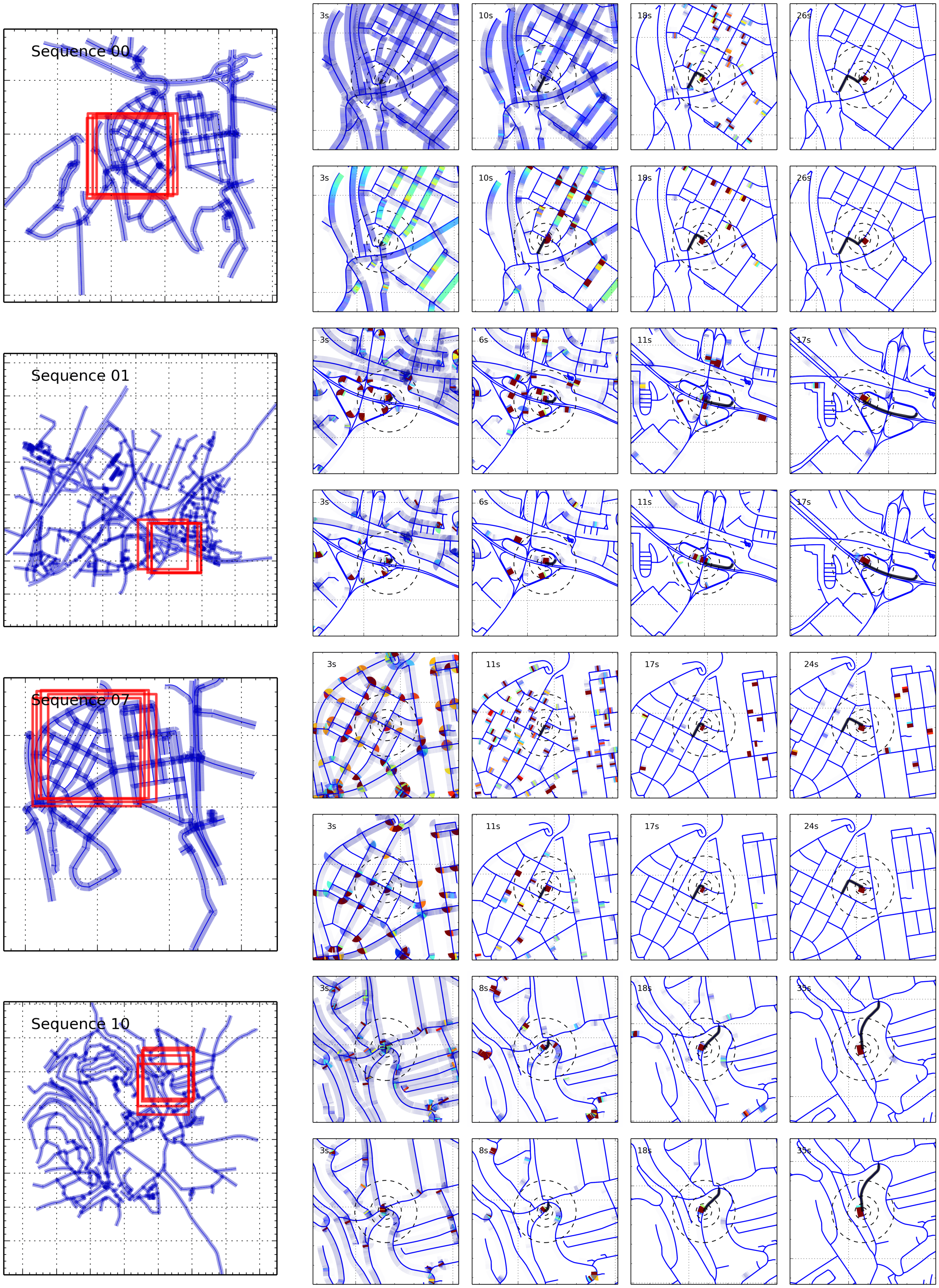

Figure 5: **Qualitative localization results compared to [1]:** The left most column shows the sub-map region for each sequence, followed by zoomed in sections of the map showing the posterior distribution over time. The upper row is the result of [1], and the lower row is ours. The black line is the GPS trajectory and the concentric circles indicate the current GPS position. Grid lines are every **500m**. Red regions indicate high probability, while blue regions indicate low probability.

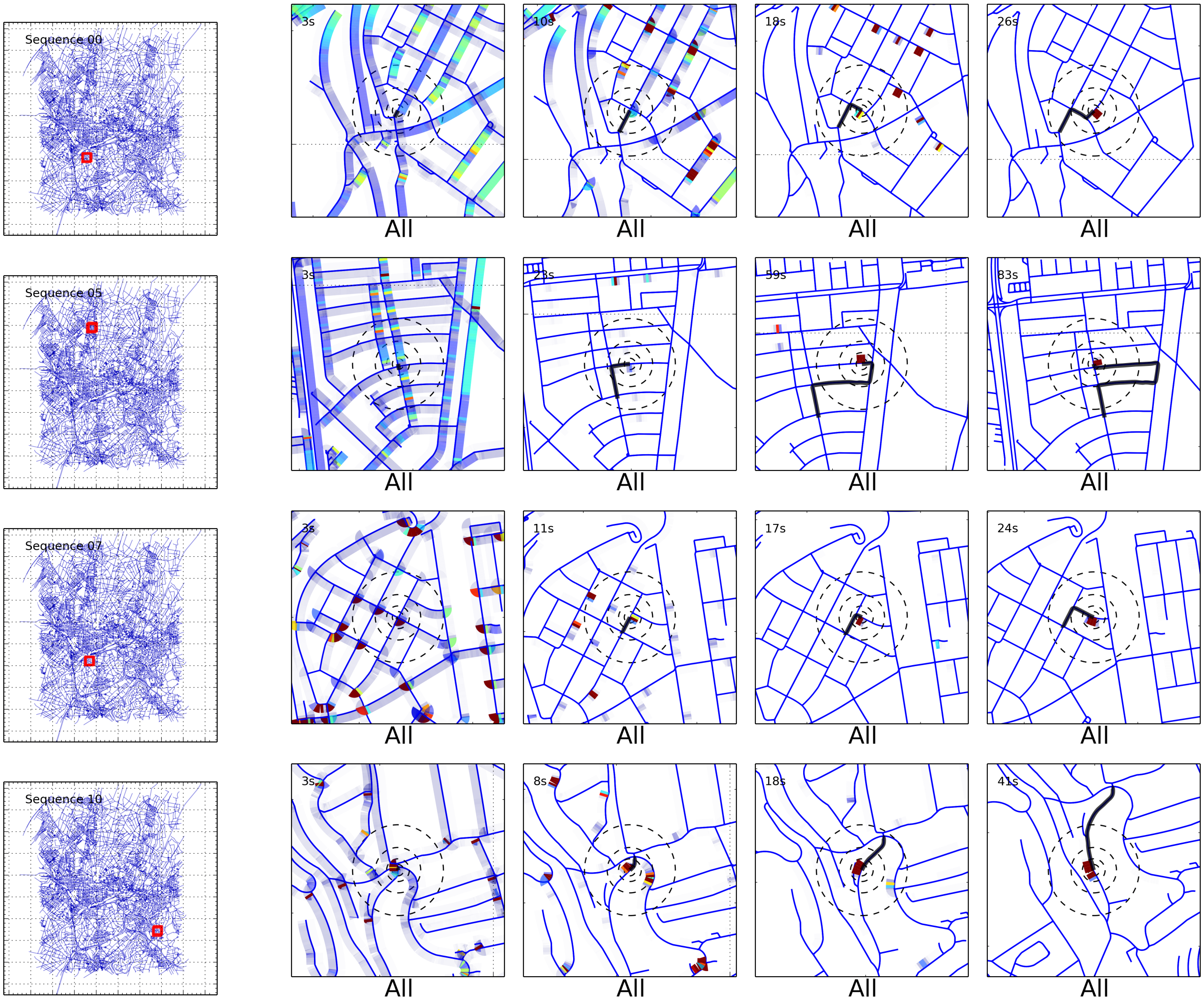

Figure 6: **Qualitative Results on Full Map:** The left most column shows the full map for each sequence, followed by zoomed in sections of the map showing the posterior distribution over time. The black line is the GPS trajectory and the concentric circles indicate the current GPS position. Grid lines are every **2km**. Red regions indicate high probability, while blue regions indicate low probability. The results show that our full model still achieve comparable performance even using the full map.

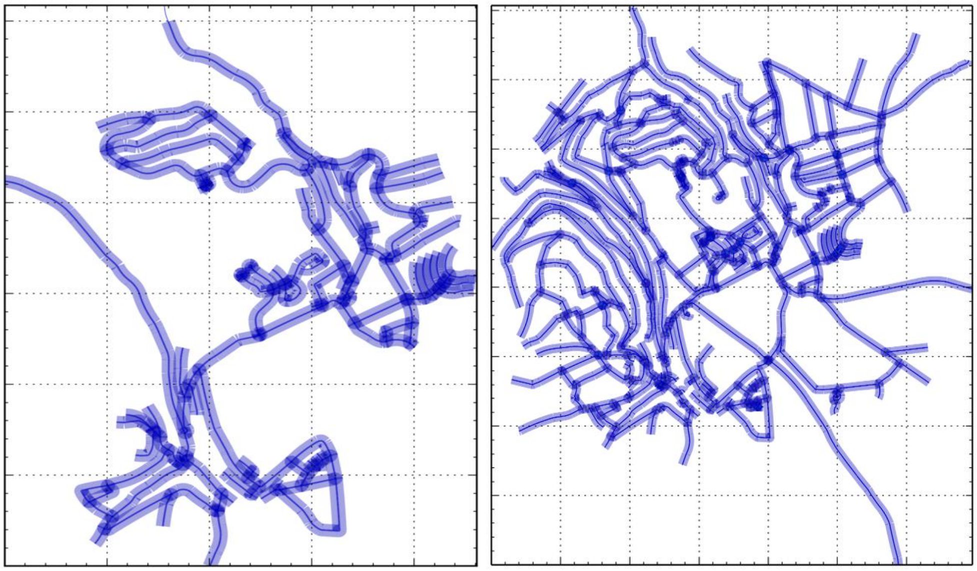

Figure 7: **Map Comparison:** We show two maps of the same sub-region. The map of [1] is on the left, while ours is on the right. Unlike [1] who pruned roughly half of the roads during preprocessing stage based on human prior knowledge, we preserve all drivable roads in the map. The map used in this work has a larger region and contains more road. In particular, our map has double the amount of roads. Grid lines are every 500m.

		00	01	02	03	05	06	07	08	09	10	Average
Localization Time	O[1]	22s	90s	27s	36s	52s	*	27s	**79**s	30s	43s	46 ± 24s
	OS	**20**s	16s	22s	27s	43s	69s	**13**s	**79**s	26s	33s	28 ± 22s
	OI	22s	23s	25s	30s	51s	68s	34s	80s	18s	41s	33 ± 21s
	OR	23s	23s	27s	36s	52s	*	27s	**79**s	30s	42s	38 ± 18s
	OV	22s	90s	28s	34s	53s	*	27s	**79**s	36s	42s	47 ± 24s
	OSIR	**20**s	13s	**21**s	24s	**42**s	**31**s	**13**s	80s	21s	16s	29 ± 20s
	OSIV	**20**s	**12**s	**21**s	**21**s	43s	**31**s	**13**s	80s	**13**s	17s	28 ± 21s
	OSRV	**20**s	**12**s	23s	24s	44s	68s	**13**s	**79**s	29s	36s	36 ± 22s
	OIRV	22s	90s	23s	30s	52s	*	28s	80s	18s	41s	44 ± 26s
	OSIRV	**20**s	**12**s	**21**s	**21**s	43s	**31**s	**13**s	80s	**13**s	**15**s	**25 ± 21**s
Computation Time	O[1]	121s	280s	175s	48s	198s	194s	60s	94s	224s	190s	171s
	OS	86s	222s	112s	24s	147s	**118**s	42s	81s	163s	117s	121s
	OI	122s	287s	122s	34s	180s	170s	55s	81s	246s	210s	164s
	OR	101s	253s	147s	48s	230s	259s	81s	120s	261s	194s	183s
	OV	128s	234s	180s	50s	171s	184s	51s	94s	231s	204s	164s
	OSIR	**82**s	285s	**104**s	20s	**137**s	164s	40s	105s	176s	135s	137s
	OSIV	**82**s	175s	124s	28s	144s	155s	**38**s	80s	**152**s	118s	119s
	OSRV	84s	211s	123s	29s	118s	144s	**38**s	114s	170s	153s	128s
	OIRV	110s	207s	139s	33s	203s	168s	70s	82s	266s	164s	156s
	OSIRV	93s	**174**s	106s	**19**s	141s	124s	39s	**78**s	190s	**106**s	**117**s
Computation Time Per Frame	O[1]	0.27s	2.55s	0.37s	0.60s	0.72s	1.76s	0.55s	0.23s	1.41s	1.58s	0.69s
	OS	0.19s	2.02s	0.24s	0.30s	0.53s	**1.07**s	0.39s	0.20s	1.02s	0.98s	0.49s
	OI	0.27s	2.61s	0.26s	0.43s	0.65s	1.55s	0.50s	0.20s	1.55s	1.75s	0.67s
	OR	0.22s	2.30s	0.32s	0.60s	0.84s	2.36s	0.73s	0.30s	1.64s	1.62s	0.74s
	OV	0.28s	2.13s	0.39s	0.63s	0.62s	1.67s	0.47s	0.23s	1.46s	1.70s	0.67s
	OSIR	**0.18**s	2.59s	**0.22**s	0.25s	**0.50**s	1.49s	0.37s	0.26s	1.11s	1.13s	0.56s
	OSIV	**0.18**s	1.59s	0.27s	0.35s	0.52s	1.41s	**0.35**s	0.20s	**0.96**s	0.98s	0.48s
	OSRV	0.19s	1.92s	0.26s	0.36s	0.43s	1.31s	**0.35**s	0.28s	1.07s	1.28s	0.52s
	OIRV	0.24s	1.88s	0.30s	0.41s	0.73s	1.53s	0.64s	0.20s	1.67s	1.37s	0.64s
	OSIRV	0.20s	**1.59**s	0.23s	**0.24**s	0.51s	1.13s	0.36s	**0.19**s	1.20s	**0.88**s	**0.48**s
Position	O[1]	2.3 ± 1.6m	6.7 ± 3.2m	4.1 ± 3.0m	4.8 ± 2.0m	3.1 ± 1.6m	*	1.9 ± 1.0m	2.9 ± 1.7m	4.4 ± 3.4m	3.9 ± 1.8m	3.2 ± 2.4m
	OS	**2.3 ± 1.5**m	6.8 ± 4.2m	4.2 ± 3.0m	4.3 ± 1.9m	3.1 ± 1.6m	2.9 ± 2.6m	1.9 ± 1.0m	3.0 ± 1.7m	**4.3 ± 3.3**m	3.8 ± 2.3m	3.3 ± 2.4m
	OI	2.5 ± 1.8m	5.6 ± 3.4m	3.4 ± 2.3m	4.4 ± 1.6m	4.1 ± 3.2m	8.1 ± 3.5m	1.7 ± 1.0m	3.1 ± 1.9m	4.7 ± 3.1m	5.3 ± 3.7m	3.4 ± 2.7m
	OR	2.3 ± 1.6m	3.7 ± 3.1m	4.3 ± 3.2m	4.8 ± 2.1m	**3.0 ± 1.5**m	*	2.0 ± 1.1m	3.0 ± 1.8m	4.4 ± 3.3m	**3.7 ± 1.8**m	3.3 ± 2.4m
	OV	2.3 ± 1.6m	**2.6 ± 1.6**m	4.0 ± 3.0m	4.7 ± 2.0m	3.0 ± 1.6m	*	1.9 ± 1.0m	**2.9 ± 1.7**m	4.4 ± 3.3m	4.1 ± 2.3m	**3.2 ± 2.3**m
	OSIR	2.5 ± 1.8m	5.5 ± 3.8m	3.5 ± 2.5m	3.9 ± 1.8m	4.0 ± 3.2m	8.0 ± 3.3m	1.6 ± 0.9m	3.1 ± 1.8m	4.9 ± 3.3m	5.3 ± 4.0m	3.5 ± 2.7m
	OSIV	2.5 ± 1.9m	5.6 ± 3.8m	3.5 ± 2.5m	3.9 ± 1.9m	4.0 ± 3.1m	8.0 ± 3.3m	1.7 ± 0.9m	3.0 ± 1.8m	4.8 ± 3.4m	5.2 ± 3.7m	3.4 ± 2.7m
	OSRV	**2.3 ± 1.5**m	3.2 ± 1.5m	4.2 ± 3.0m	4.0 ± 2.0m	3.1 ± 1.6m	**2.9 ± 1.8**m	2.0 ± 1.0m	2.9 ± 1.7m	4.4 ± 3.3m	3.9 ± 2.2m	**3.2 ± 2.3**m
	OIRV	2.5 ± 1.9m	3.8 ± 3.1m	**3.3 ± 2.2**m	4.3 ± 1.8m	3.9 ± 3.0m	*	1.8 ± 0.9m	3.2 ± 2.0m	4.8 ± 3.2m	5.3 ± 4.0m	3.3 ± 2.5m
	OSIRV	2.5 ± 1.9m	5.8 ± 3.5m	3.4 ± 2.4m	**3.7 ± 2.0**m	4.0 ± 3.2m	7.7 ± 3.3m	**1.7 ± 0.9**m	3.1 ± 1.8m	4.9 ± 3.2m	5.2 ± 4.0m	3.4 ± 2.7m
Heading	O[1]	1.3 ± 1.2°	5.2 ± 1.7°	**1.2 ± 1.1**°	1.6 ± 1.2°	**1.3 ± 1.0**°	*	1.8 ± 1.1°	1.3 ± 1.4°	**1.1 ± 1.1**°	1.4 ± 1.2°	1.3 ± 1.3°
	OS	1.5 ± 1.3°	4.5 ± 3.5°	1.4 ± 1.1°	1.8 ± 1.2°	1.9 ± 1.4°	1.3 ± 2.1°	**1.4 ± 1.0**°	1.3 ± 1.4°	1.6 ± 1.0°	2.0 ± 1.1°	1.5 ± 1.4°
	OI	1.1 ± 1.2°	2.9 ± 2.9°	1.3 ± 1.2°	1.7 ± 1.1°	1.4 ± 1.2°	1.8 ± 1.5°	2.4 ± 1.3°	**1.2 ± 1.4**°	1.2 ± 1.1°	1.5 ± 1.1°	1.3 ± 1.3°
	OR	1.4 ± 1.4°	1.6 ± 1.3°	**1.2 ± 1.1**°	1.6 ± 1.2°	1.4 ± 1.0°	*	1.9 ± 1.1°	1.3 ± 1.5°	1.3 ± 1.2°	1.4 ± 1.1°	1.3 ± 1.3°
	OV	**1.2 ± 1.2**°	**1.2 ± 0.7**°	**1.2 ± 1.1**°	1.6 ± 1.2°	1.3 ± 1.1°	*	2.0 ± 1.5°	**1.2 ± 1.4**°	1.2 ± 1.1°	1.4 ± 1.1°	**1.3 ± 1.2**°
	OSIR	1.5 ± 1.3°	3.4 ± 2.7°	1.4 ± 1.2°	1.6 ± 1.2°	1.8 ± 1.3°	1.9 ± 1.9°	1.5 ± 1.2°	1.3 ± 1.5°	1.7 ± 1.3°	2.2 ± 1.1°	1.6 ± 1.4°
	OSIV	1.5 ± 1.4°	3.4 ± 3.1°	1.4 ± 1.2°	1.7 ± 1.2°	1.9 ± 1.4°	2.2 ± 2.0°	1.5 ± 1.0°	1.3 ± 1.4°	1.7 ± 1.2°	2.1 ± 1.1°	1.6 ± 1.4°
	OSRV	1.5 ± 1.4°	1.8 ± 1.1°	1.5 ± 1.2°	1.6 ± 1.2°	1.9 ± 1.4°	1.7 ± 2.1°	1.5 ± 1.1°	1.3 ± 1.4°	1.7 ± 1.2°	2.1 ± 1.2°	1.6 ± 1.3°
	OIRV	1.3 ± 1.3°	2.2 ± 1.7°	1.3 ± 1.1°	**1.6 ± 1.0**°	1.4 ± 1.1°	*	2.5 ± 1.3°	**1.2 ± 1.4**°	1.2 ± 1.2°	**1.4 ± 0.8**°	1.4 ± 1.3°
	OSIRV	1.4 ± 1.4°	3.3 ± 3.1°	1.4 ± 1.1°	1.7 ± 1.3°	2.0 ± 1.5°	**1.3 ± 1.9**°	1.4 ± 1.1°	1.3 ± 1.3°	1.6 ± 1.3°	1.9 ± 1.1°	1.5 ± 1.4°

Table 1: **Quantitative Evaluation:** ”O”, ”S”, ”I”, ”R”, ”V” represent the observation types that were used during inference, i.e., visual odometry, sun direction, intersection type, road type and velocity. The average localization time, position and heading error are computed with sequences that localizes. Sequences that did not localize are indicated with a ”*”. No approach localize Sequence 04. When employing all semantics our approach localizes faster and requires less computation time per frame.