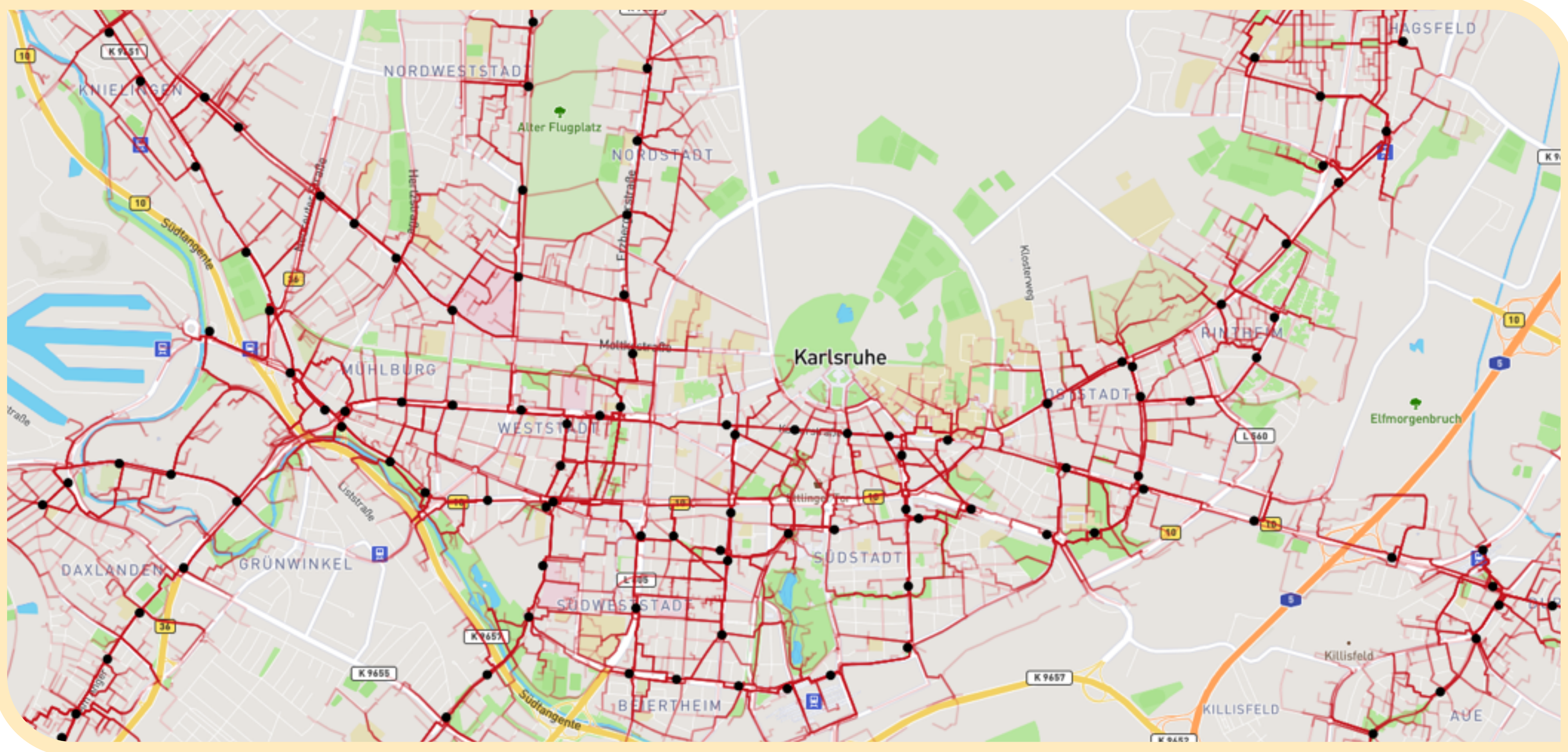
Mind the Gap: Virtual Shorelines for Blind and Partially Sighted People

Daniel Koester, Maximilian Awiszus, Rainer Stiefelhagen
Karlsruhe Institute of Technology

Karlsruhe Institute of Technology

ICCV 17 ACVR 2017



- We propose a novel routing system for blind and partially sighted people on a **shoreline level of detail**.
- We rely on **openly available geolocation data**.
- The routing considers **actual white-cane based movement** along inner & outer shorelines.
- We evaluate on 1870 routes between public transit stations and common destinations in an urban area.
- The algorithm creates **safer routes**: avoid informal crossings, prefer accessible pedestrian signals and integrate available shorelines.
- Our system **improves the users' understanding** of the upcoming route, the environment lying ahead and its impediments.

```
 1: d := {0, ∞, ..., ∞}
 2: prev := {0, −1, ..., −1}
 3: q := {(p_{r=0}, 0)}
 4: while q ≠ ∅ do
 5:     p_u := q.pop()
 6:     for all l_i ∈ (S'_R ∪ R) do
 7:         p_v := f_near(p_u, l_i)
 8:         if d[u] + δ_{p_u p_v l_I} < d[v] then
 9:             d[v] := d[u] + δ_{p_u p_v l_I}
10:             prev[v] := u
11:             q.push(p_v, d[v])
12:         end if
13:     end for
14: end while
```

Routing Algorithm:

(1-3) initialize cumulative node {***d***}istance, {***prev***}ious node for shortest connection and distance sorted priority {***q***}ueue

(4-5) while ***q*** not empty, take closest node

(6) check all reachable shoreline or OSM route segments

(7) closest façade point for $\boldsymbol{p_u}$ along $\boldsymbol{l_i}$

(8-11) if distance to new node < *current*, store and add new node to ***q***

Modified Cost Function:

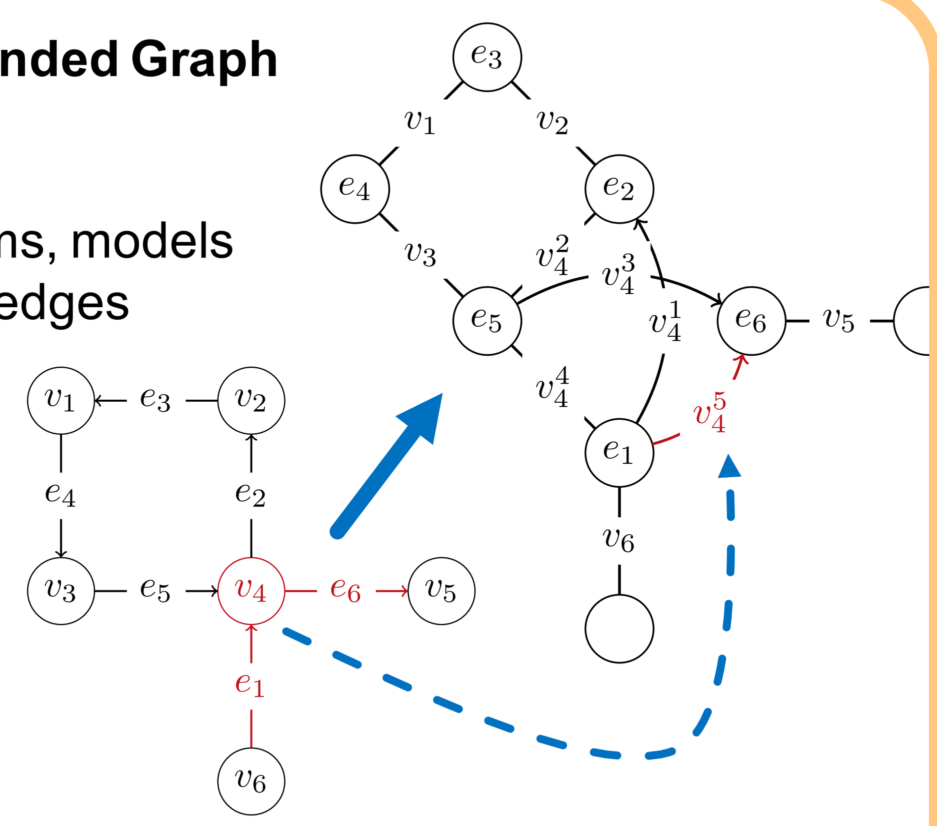
$$\delta_{p_u p_v l_I} := \begin{cases} \mathcal{W}_S \cdot \|p_u - p_v\|_2\,, & (\delta_{p_u l_I} = 0) \wedge (l_i \in \mathcal{S}) \\ \mathcal{W}_R(l_i) \cdot \|p_u - p_v\|_2\,, & (\delta_{p_u l_I} = 0) \wedge (l_i \in \mathcal{R}) \\ \delta_{p_u l_I}\,, & otherwise \end{cases}$$

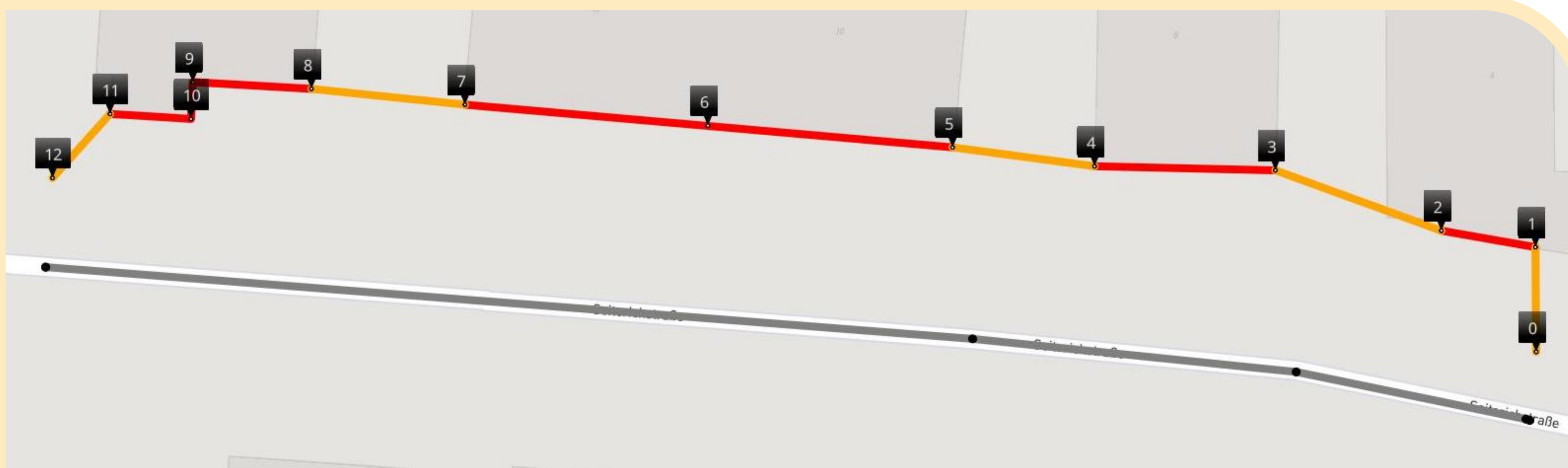
$$\delta_{p_u l_I} := (1 + |\mathcal{C}_{p_u p_v}| \cdot \mathcal{W}_R) \cdot d(p_u, l_i) \qquad \mathcal{W}_R(l_i) := \begin{cases} \mathcal{W}_C, & informalCrossing(l_i) \\ \mathcal{W}_{PS} \cdot \mathcal{W}_R, & PedestrianSignal(l_i) \\ \mathcal{W}_{APS} \cdot \mathcal{W}_R, & APS(l_i) \\ \mathcal{W}_{APS_p} \cdot \mathcal{W}_R, & PilotTone(l_i) \\ \mathcal{W}_R, & otherwise \end{cases}$$

Pre-Defined Weight Constraints: $\mathcal{W}_C > \mathcal{W}_{PS} > \mathcal{W}_{APS} > \mathcal{W}_{APS_p} > \mathcal{W}_R > \mathcal{W}_S > 1$

Directed Graph to Edge Expanded Graph Transformation:

- ***DG*** used for routing algorithms, models distances between nodes in edges
- Dual ***EEG*** allows to model ***Turn-Restrictions***
- Here: disallow right turn from e_1 to e_6 at intersection v_4
- Allows us to model different ways to cross the same intersection

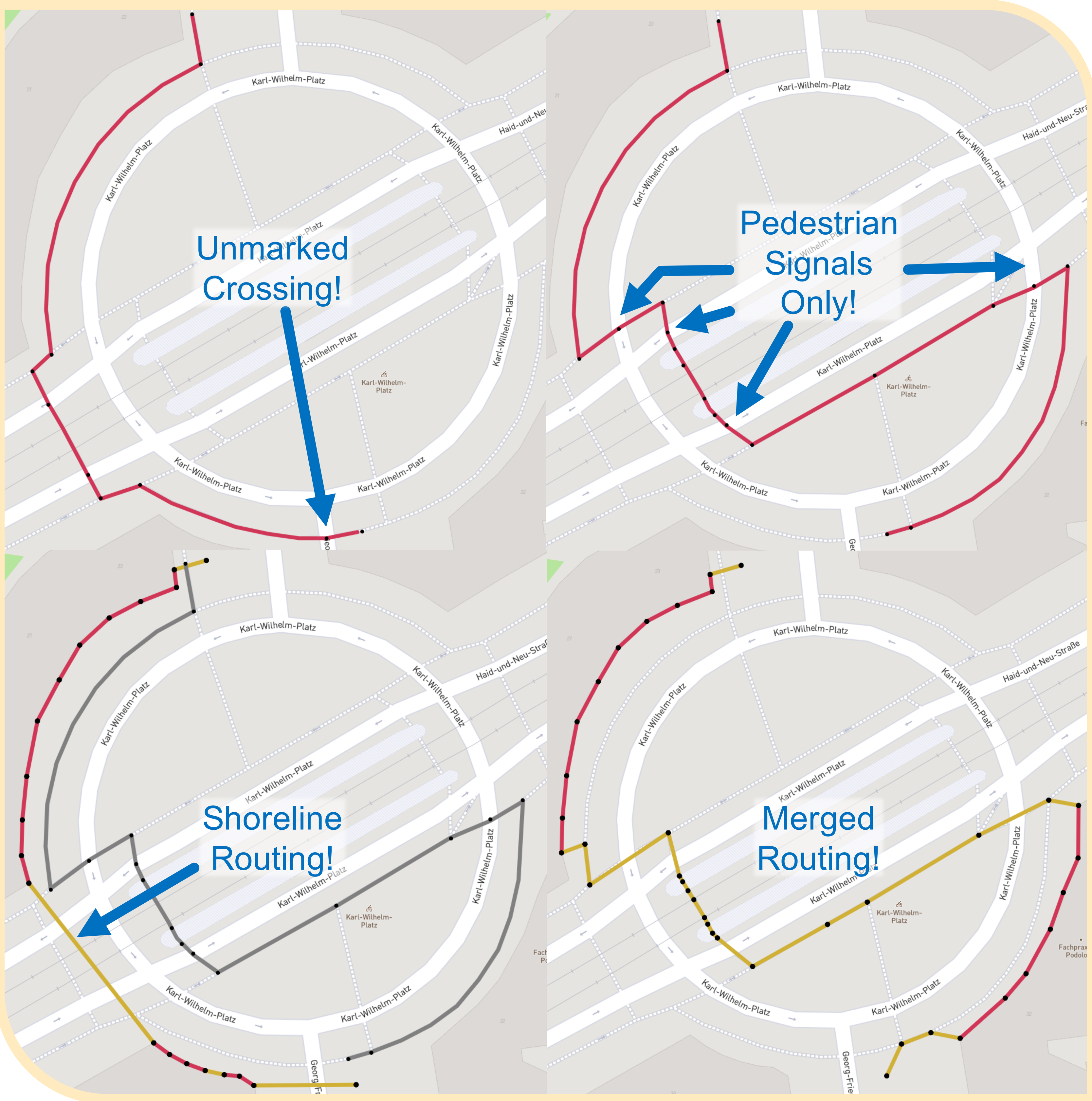




0) *"Please turn north until you reach a façade."*
1) *"Follow the façade to the left for 8m."*
2) *"Continue for 18m at 1 o'clock to cross a driveway."*
3) *"Follow the façade for 16m."*
4) *"Continue for 12m straight to cross a driveway."*

…

10) *"Turn right and follow the façade for 6m."*
11) "Continue for 6m at 10 o'clock across the sidewalk."
12) *"You have reached your destination."*



	$\bar{d}$	$\bar{r}_w$	$\overline{ps}$	$\overline{ps}_a$	$\overline{ps}_p$	$\bar{c}$
R_{OSRM}	**621**	26,4	2.327	0.205	0.019	0.702
$R_{Walkway}$	654	**46,4**	**4.819**	**0.338**	0.070	1.501
R_{APS}	655	45,5	4.709	0.320	**0.814**	**0.615**

Public transit station based route evaluation:
distance ($\bar{d}$), % pedestrian walkway ($\bar{r}_w$), # pedestrian signal ($\overline{ps}$), haptic/aural/pilot-tone *APS* ($\overline{ps}_a/\overline{ps}_p$), % real/virtual shorelines ($\bar{s}_r/\bar{s}_v$) and # informal crossings ($\bar{c}$).

	$\bar{d}$	$\bar{r}_w$	$\bar{s}_r$	$\bar{s}_v$	$\bar{c}$
$R_{Shorelines}$	**178**	00.0	**31.4**	7.9	.056
R_{final}	198	**22.0**	26.0	**8.2**	**.035**