Mapping, Localization and Path Planning for Image-based Navigation using Visual Features and Map

Authors: Janine Thoma, Danda Pani Paudel, Ajad Chhatkuli, Thomas Probst, Luc Van Gool

In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019

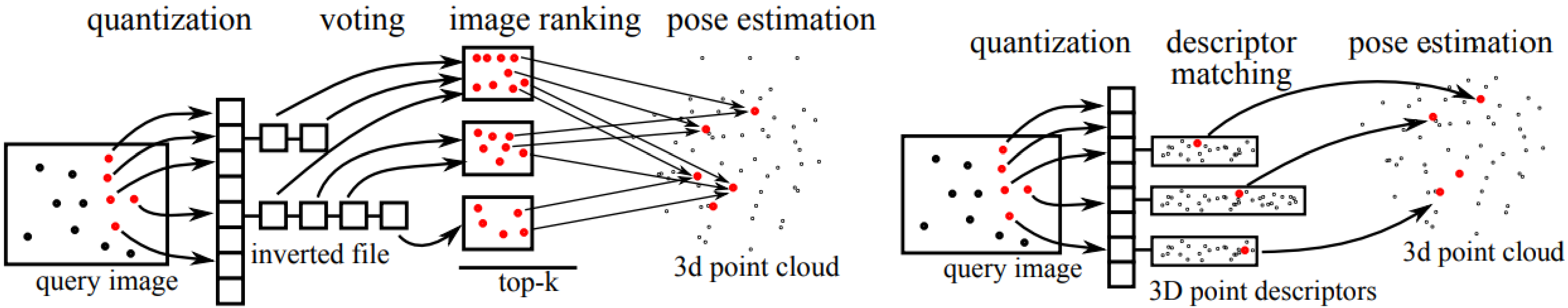
Presenter: B. Y. Huang

Outline

- Motivation
- Map representation
- Self localization
- Experimental results
- Conclusions

Motivation

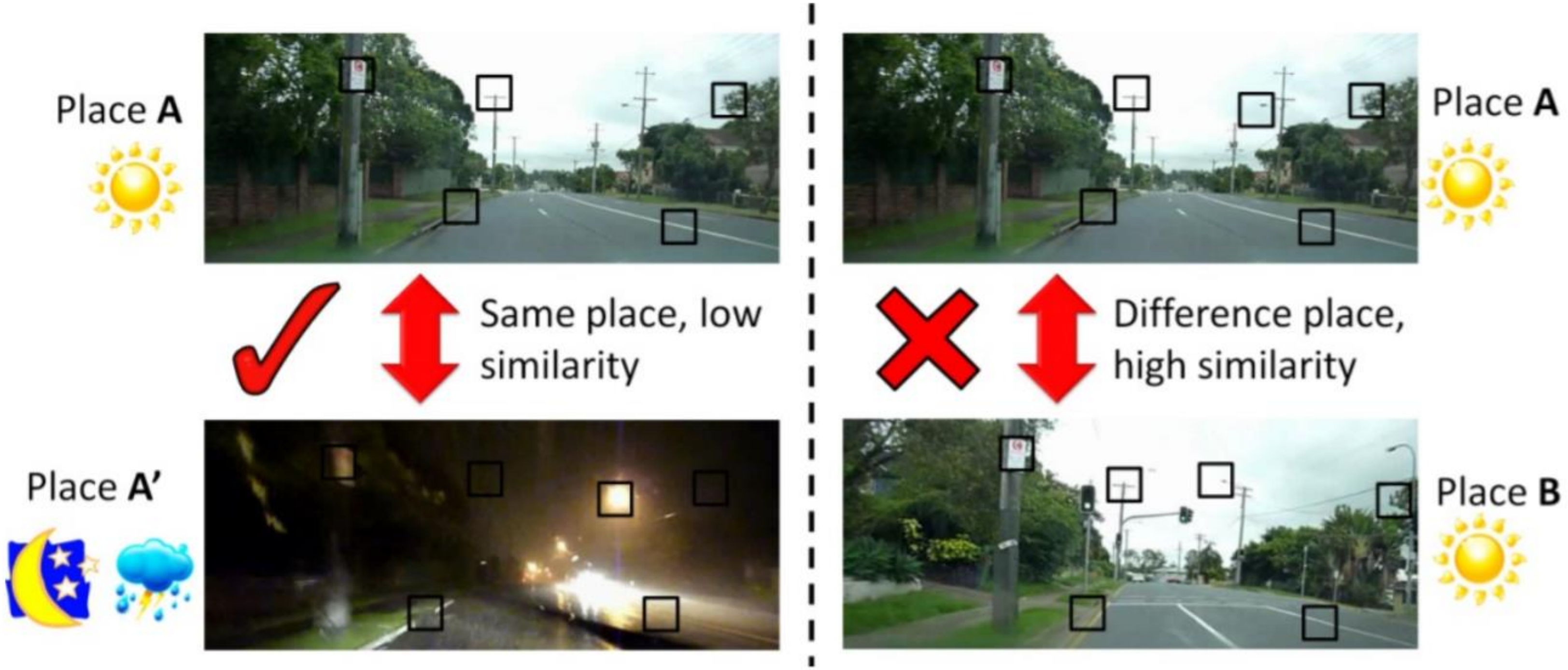
- Feature matching plays an important rule in image-based localization
- Map representation problem is not well-addressed by current methods
- The order of query image sequences is often neglected



Ref: Torsten Sattler et al., "Image Retrieval for Image-Based Localization Revisited." BMVC 2012

SeqSLAM (1/2)

- Navigation for sunny summer days and stormy winter nights
- Instead of calculating the single location most likely given a current image, calculates the best candidate matching location within every local navigation sequence.



SeqSLAM (2/2)

- **Local best match**

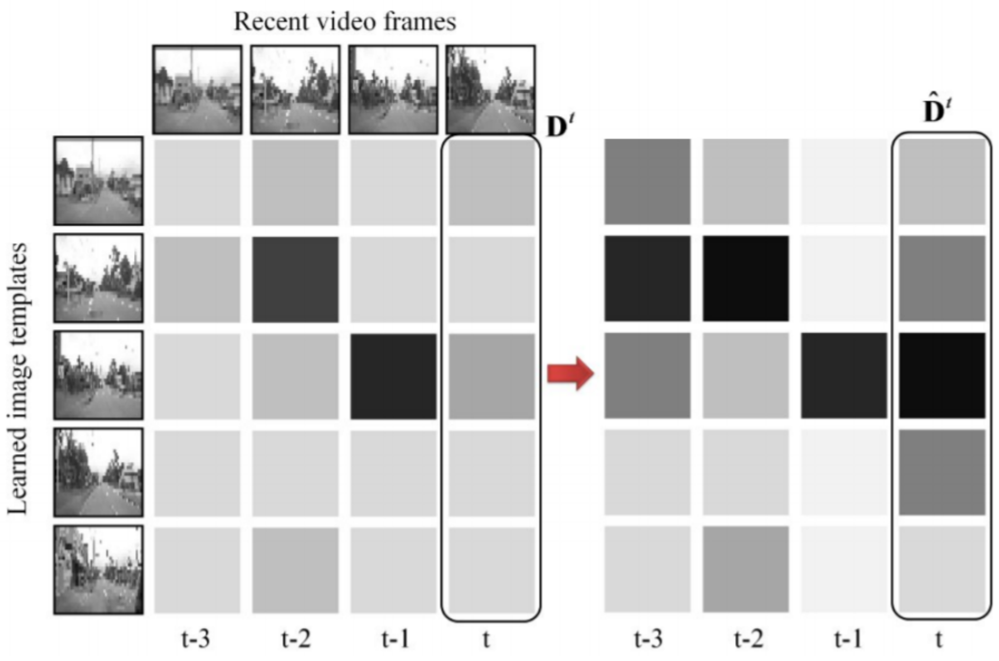
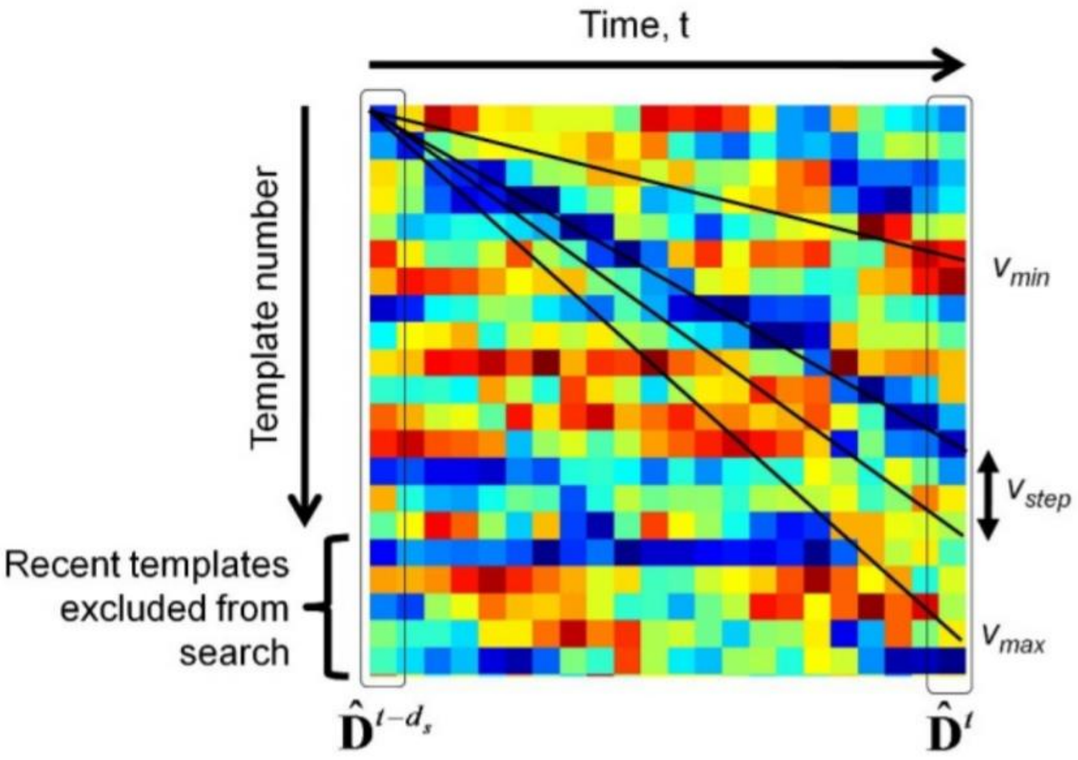


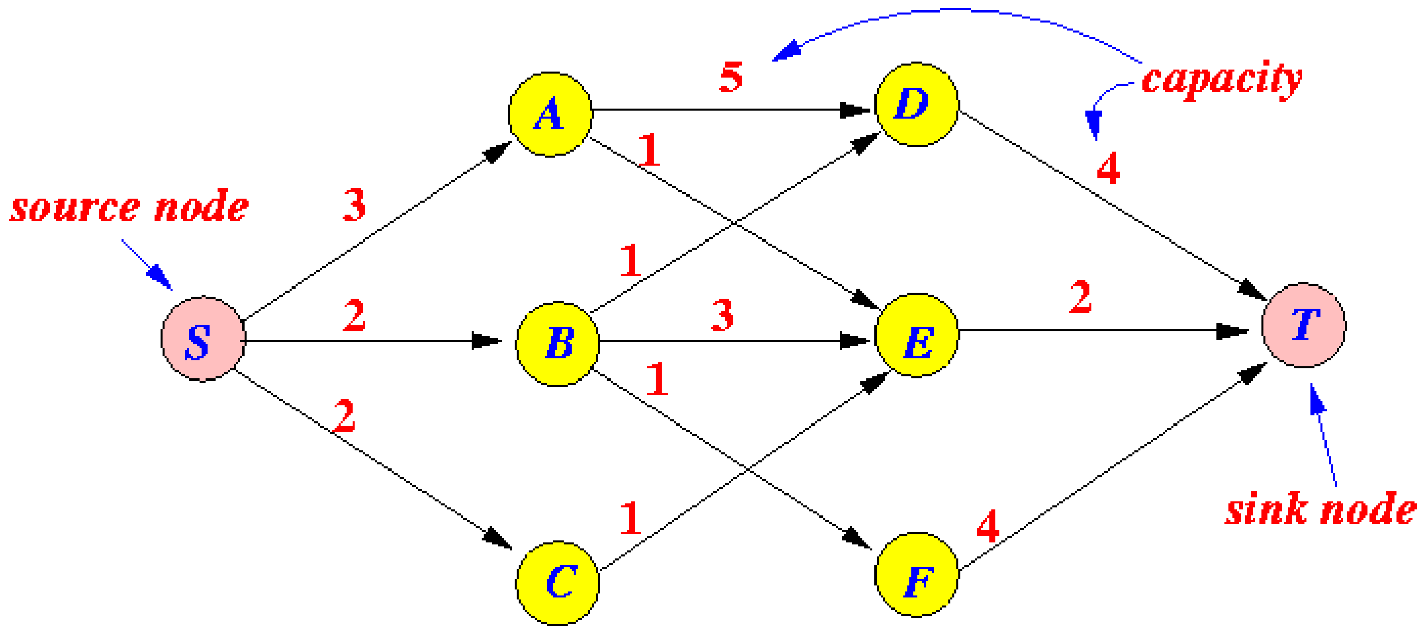
Fig. 2: Contrast enhancement of the original image difference vectors increases the number of strongly matching templates. Darker shading = smaller image difference = stronger match.

- **Sequence recognition**



What they want to do

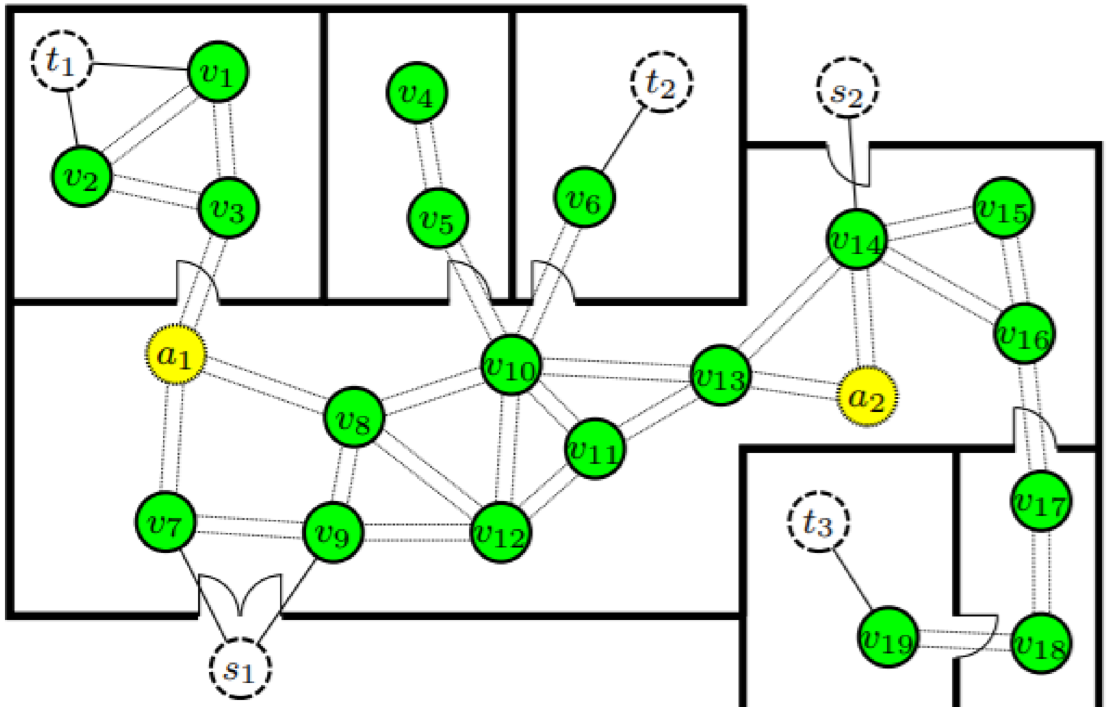
- Network flow problem (for map construction and localization)
  - maximum flow problem …



Ref: http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/NetFlow/intro.html

Outline

- Motivation
- Map representation
- Self localization
- Experimental results
- Conclusions



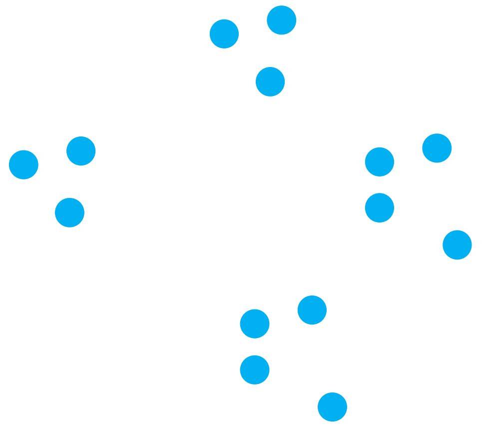
summarize the reference images as a set of “landmarks”

Map representation (1/6)

- **Construct Graph *G***
  - given database images *I* with their location coordinates *X*, and visual feature *F*

Algorithm 1 $\mathcal{V}' = \text{selectLandmarks}(\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T})$

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. Solve the flow problem (17) for sources $\mathcal{S}$ and targets $\mathcal{T}$.
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \geq \tau\}$.



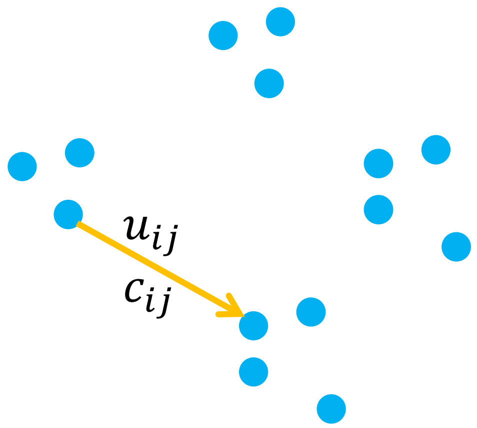
Graph obtained by *I*, *X*

Map representation (2/6)

- **Navigation assurance** – the design of flow capacity u_{ij} and flow cost rate c_{ij}
  - $u_{ij} = \lambda_x d\left(x_i, x_j\right), where\ d\left(x_i, x_j\right) \le \alpha, \forall e_{ij} \in E$
    - encourage the flow to make bigger geometric jumps
  - $c_{ij} = \lambda_f / d\left(f_i, f_j\right)$
    - encourage the distinctive consecutive features along the flow path

Algorithm 1 $\mathcal{V}' = \text{selectLandmarks}(\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T})$

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. Solve the flow problem (17) for sources $\mathcal{S}$ and targets $\mathcal{T}$.
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \ge \tau\}$.



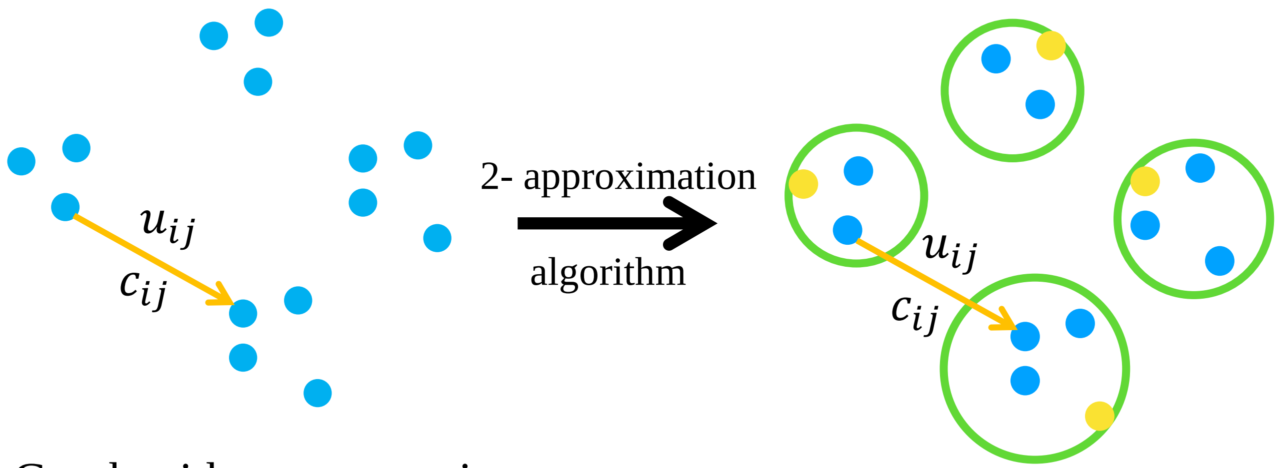
Graph with u_{ij}, c_{ij} settings

Map representation (3/6)

- **Geometric representation** – introduce anchor points
  - anchors points finding problem → k-Center problem (NP-hard)

Algorithm 1 $\mathcal{V}' = \text{selectLandmarks}(\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T})$

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. Solve the flow problem (17) for sources $\mathcal{S}$ and targets $\mathcal{T}$.
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \geq \tau\}$.



Graph with u_{ij}, c_{ij} settings

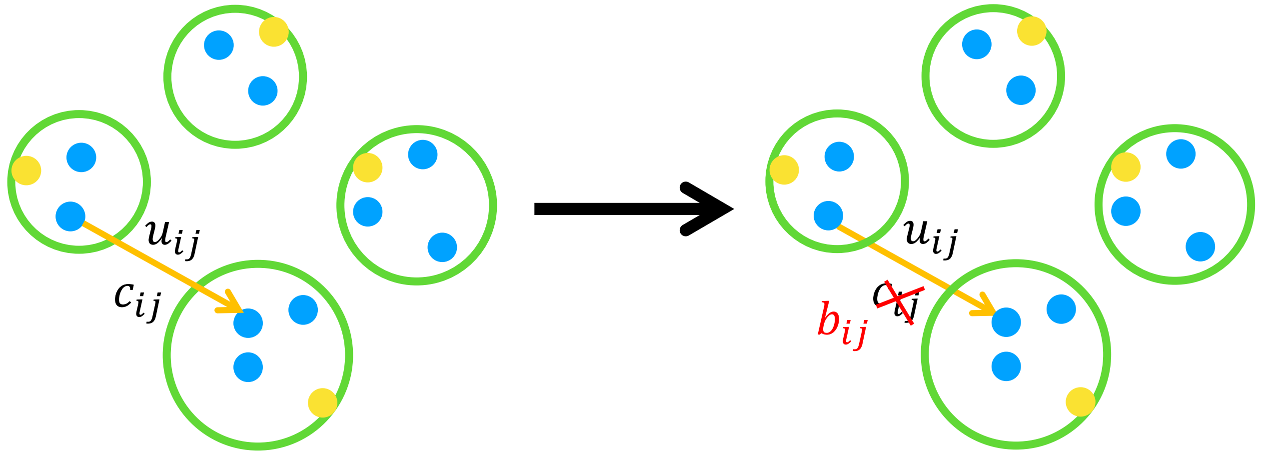
Graph with yellow anchor points

Map representation (4/6)

- **Visual representation** – introduce flow sensitivity
  - new cost rate: $b_{ij} = c_{ij} + \frac{y_{ij}}{1 - y_{ij}/u_{ij}} \times \rho_{ij}, where \;\; \rho_{ij} = 1 - \frac{d\left(f_i, f_j\right)}{\sum_{k \in N(x_i)} d\left(f_i, f_k\right)}$
    - encourage the flow spread around such that more than one landmark is selected

Algorithm 1 $\mathcal{V}' = \text{selectLandmarks}(\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T})$

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. Solve the flow problem (17) for sources $\mathcal{S}$ and targets $\mathcal{T}$.
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \geq \tau\}$.



Graph with yellow anchor points

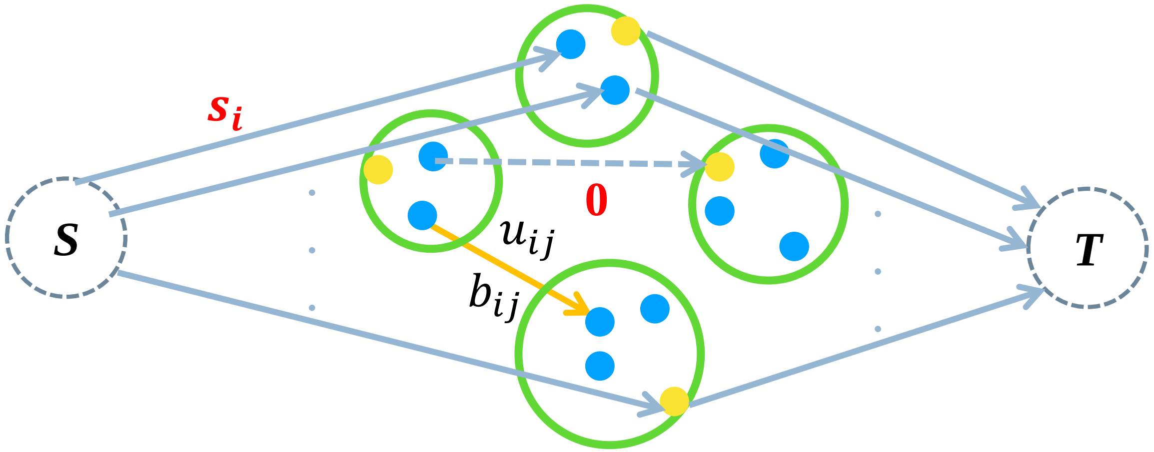
Graph with updated cost rates

Map representation (5/6)

- **Map construction**
  - solve the network flow problem
    - to transport from S to T with minimal transportation cost: $\sum_{e_{ij} \in E} y_{ij} b_{ij}$
    - each cluster should satisfy $\sum_{v_i \in N(a)} \hat{y}_i \geq \ t_g$ (neighborhood flow threshold)

Algorithm 1 $\mathcal{V}' = \text{selectLandmarks}(\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T})$

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. $\boxed{\text{Solve the flow problem (17) for sources } \mathcal{S} \text{ and targets } \mathcal{T}.}$
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \geq \tau\}$.



Graph with network flows

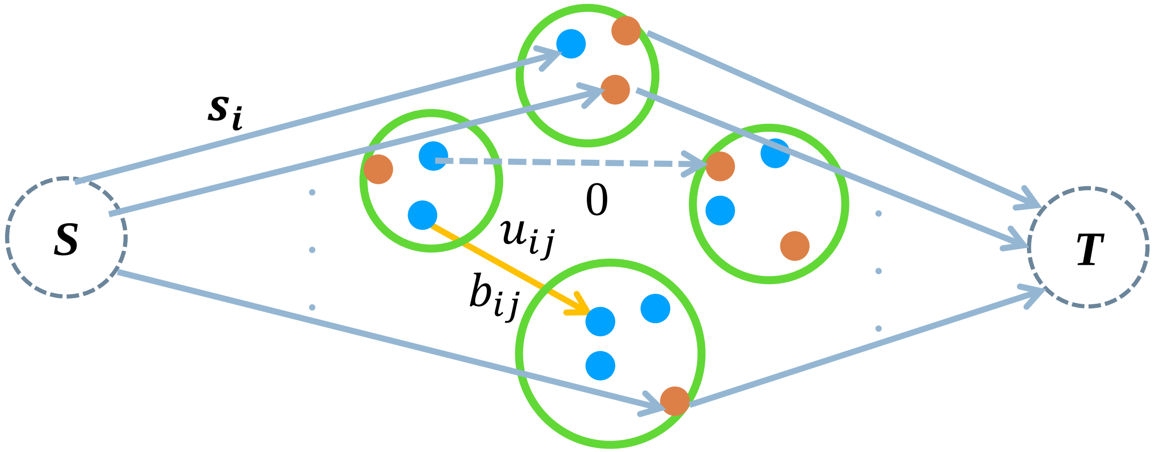
Map representation (6/6)

- **Map compactness**
  - the number of landmarks must be small, i.e. $|V'| \leq N$
    - by gradually increasing the input flow and solve the network flow problem

---

Algorithm 1 $\mathcal{V}'$ = selectLandmarks($\mathcal{I}, \mathcal{F}, \mathcal{X}, \mathcal{M}, \mathcal{S}, \mathcal{T}$)

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{X}$ (ref. Sec 4.1/(16)).
2. Compute capacity u_{ij} and rate c_{ij} for all $e_{ij} \in \mathcal{E}$, using (9).
3. Select anchor points $\mathcal{A} \subset \mathcal{M}$, by solving (10) for k-centers.
4. Compute the flow sensitivity ρ_{ij} for all $e_{ij} \in \mathcal{E}$, using (13).
5. Solve the flow problem (17) for sources $\mathcal{S}$ and targets $\mathcal{T}$.
6. Derive $\hat{y}_i$ for all $v_i \in \mathcal{V}$ from y_{ij}. Return, $\mathcal{V}' = \{v_i : \hat{y}_i \geq \tau\}$.



Landmarks are marked in orange color

Outline

- Motivation
- Map representation
- Self localization
- Experimental results
- Conclusions

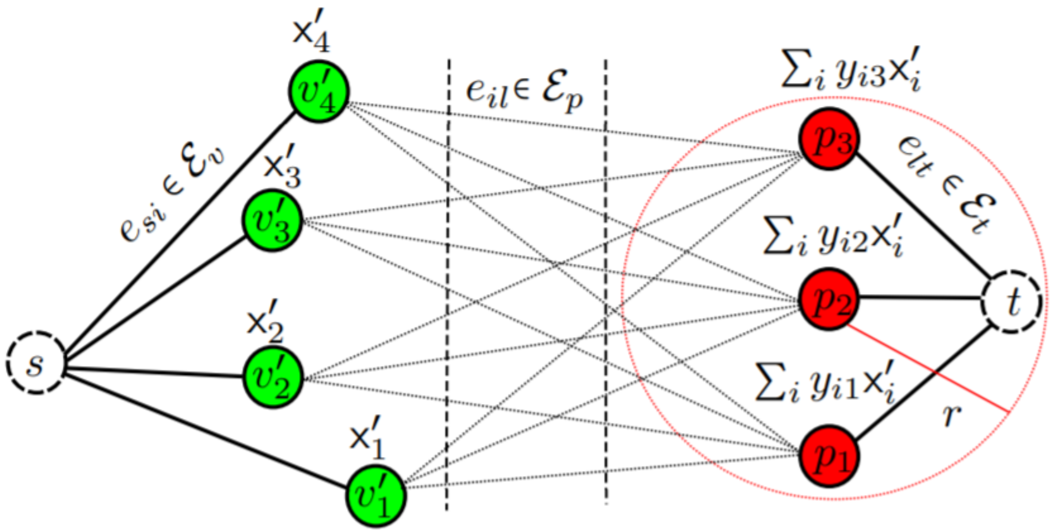
With such a good map representation,
how to do localization ?

Self localization (1/3)

- **Graph matching problem**
  - graph setup (vertices V, edges $\mathcal{E}$ w.r.t. geometric and feature measures)

Algorithm 2 $\mathcal{L}$ = selfLocalization($\mathcal{V}'$, $\mathcal{I}_p$)

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{V}'$ and $\mathcal{I}_p$ (ref. Fig. 2).
2. Compute rates $\{c_{ij}\}$ and capacities $\{u_{ij}\}$, using (18)–(19).
3. Obtain flows $\{y_{ij}\}$ by solving the flow problem (22).
4. Compute the location x_l using (20). Return, $\mathcal{L} = \{\mathrm{x}_l\}_{l=1}^{q}$.



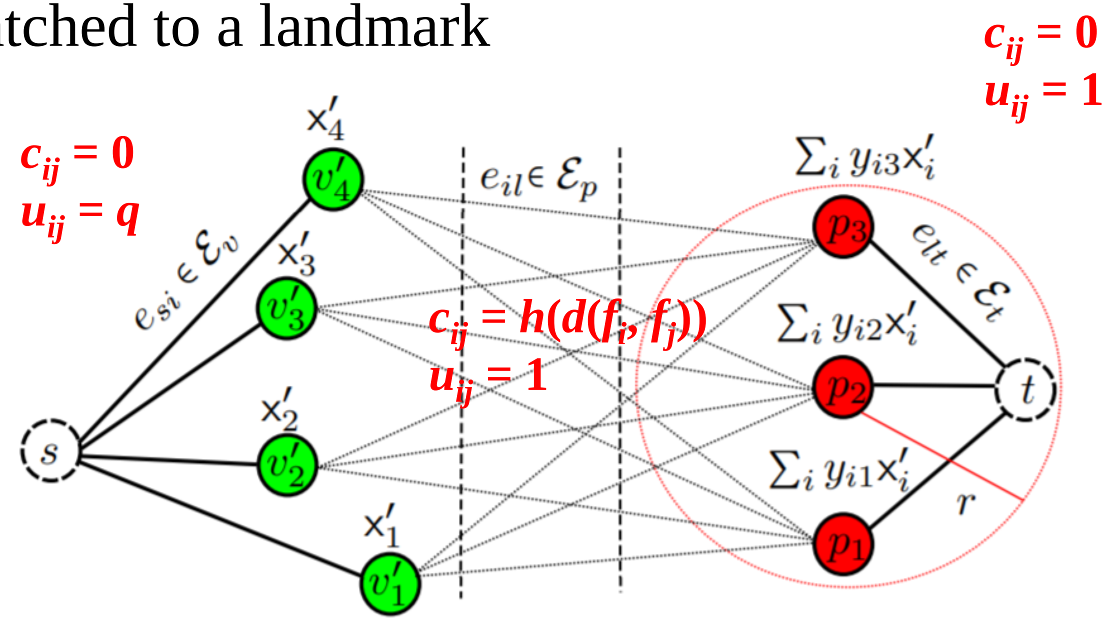
Self localization (2/3)

- **Visual matching**
  - flow cost rate: $c_{ij} = h\left(d\left(f_i, f_j\right)\right), \forall e_{ij} \in \mathcal{E}_p; c_{ij} = 0, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_p$
  - flow capacity: $u_{ij} = q, \forall e_{ij} \in \mathcal{E}_v; u_{ij} = 1, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_v$
    - ensure a query image cannot be matched to more than one landmark
    - while allow many query images to be matched to a landmark

Algorithm 2 $\mathcal{L}$ = selfLocalization($\mathcal{V}', \mathcal{I}_p$)

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{V}'$ and $\mathcal{I}_p$ (ref. Fig. 2).
2. Compute rates $\{c_{ij}\}$ and capacities $\{u_{ij}\}$, using (18)–(19).
3. Obtain flows $\{y_{ij}\}$ by solving the flow problem (22).
4. Compute the location x_l using (20). Return, $\mathcal{L} = \{\mathrm{x}_l\}_{l=1}^{q}$.

* solved by Second Order Cone Programming (SOCP)

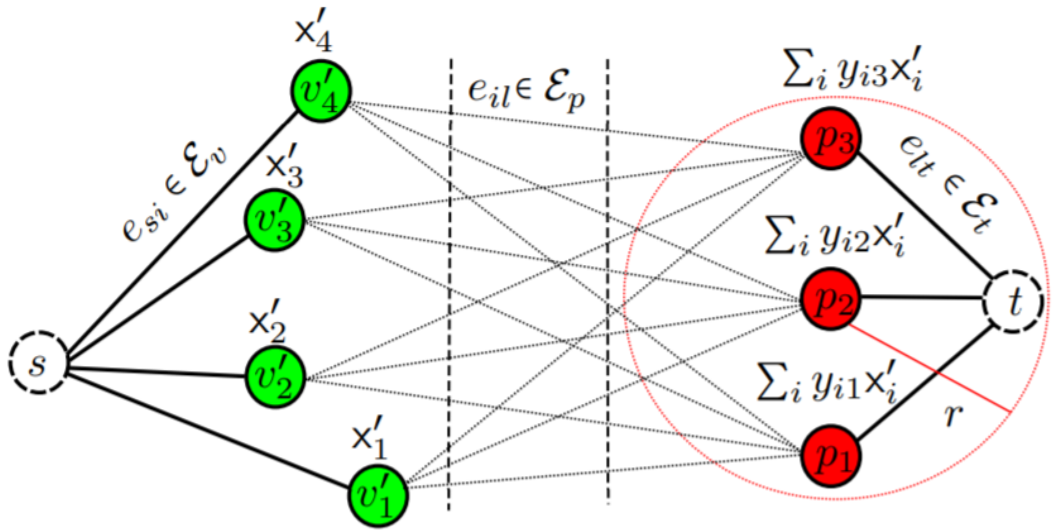


Self localization (3/3)

- **Geometric matching**
  - infer the geometric locations of the query images
  - location: $x_l = \sum_{v_i \in v'} x_i y_{il} \;\; for\ p_l \in P, where \sum_i y_{il} = 1$
  - $\left|\sum_{v_i \in v'} x_i y_{i(l+1)} - \sum_{v_i \in v'} x_i y_{il}\right| \leq r, \forall p_l, p_{l+1} \in P$

Algorithm 2 $\mathcal{L}$ = selfLocalization($\mathcal{V}', \mathcal{I}_p$)

1. Construct $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\mathcal{V}'$ and $\mathcal{I}_p$ (ref. Fig. 2).
2. Compute rates $\{c_{ij}\}$ and capacities $\{u_{ij}\}$, using (18)–(19).
3. Obtain flows $\{y_{ij}\}$ by solving the flow problem (22).
4. Compute the location x_l using (20). Return, $\mathcal{L} = \{\mathrm{x}_l\}_{l=1}^{q}$.



Outline

- Motivation
- Map representation
- Self localization
- Experimental results
- Conclusion

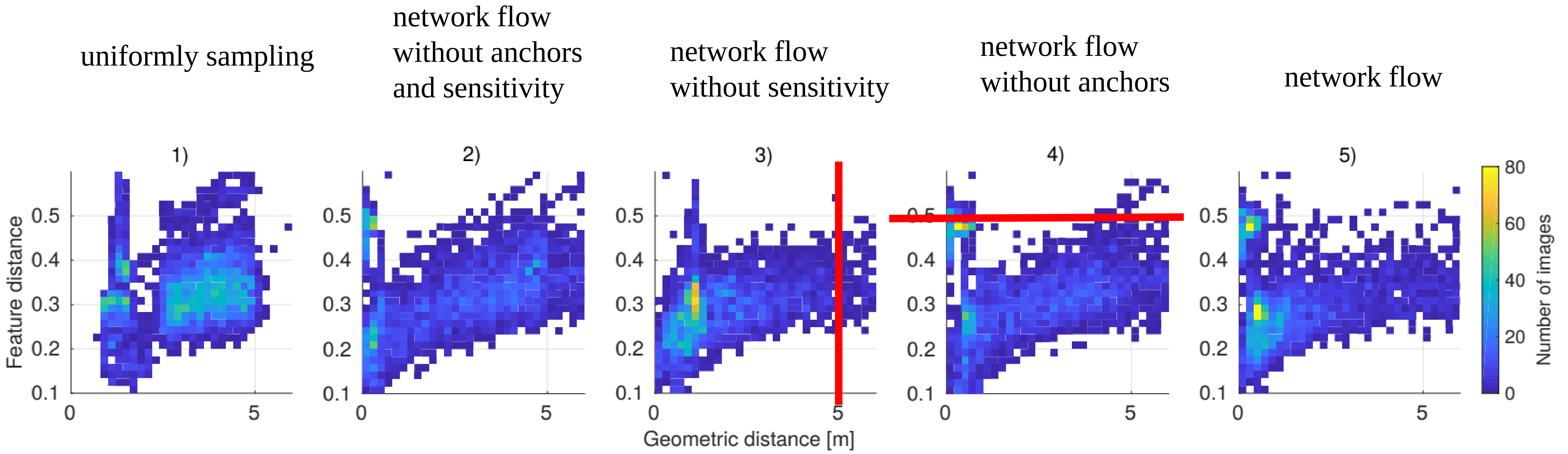
Dataset & visual features

- **Datasets**
  - Indoor: the COLD-Freiburg database
  - Outdoor: the Oxford Robotcar database

- **Visual features**
  - Obtained by NetVLAD and VGG16 FC3

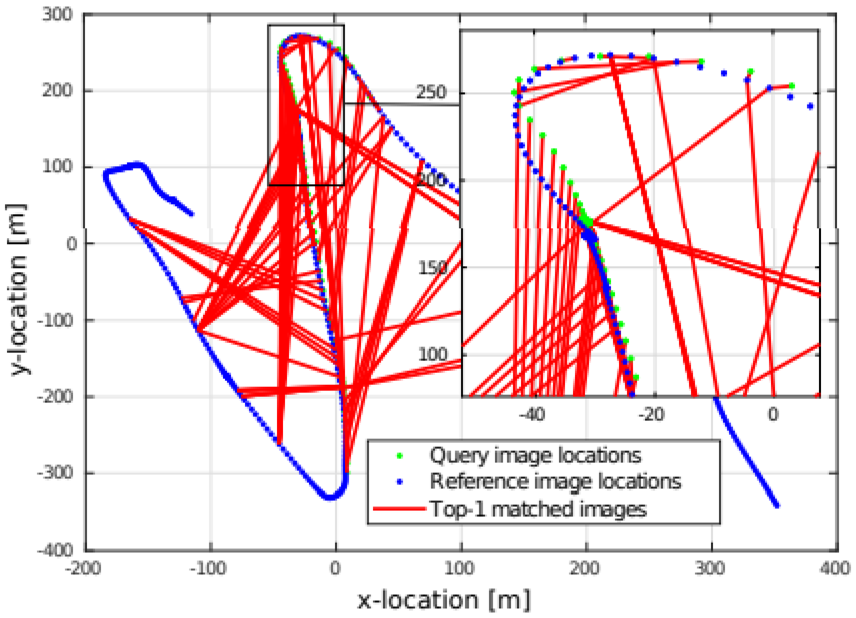
Representation comparison

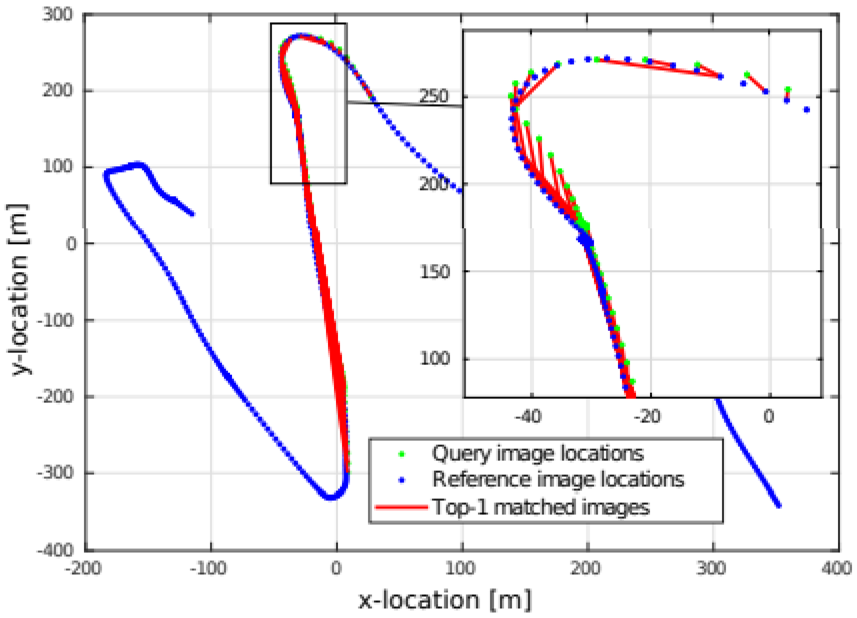
- Distribution of normalized feature and geometric distances between reference set V and geometric nearest neighbors in the summarized reference set V' ($|V| = 4833$ $|V'| = 250$, rainy Oxford Robotcar sequence 2015-10-29 12:18:17)



Difference of top-1 matches

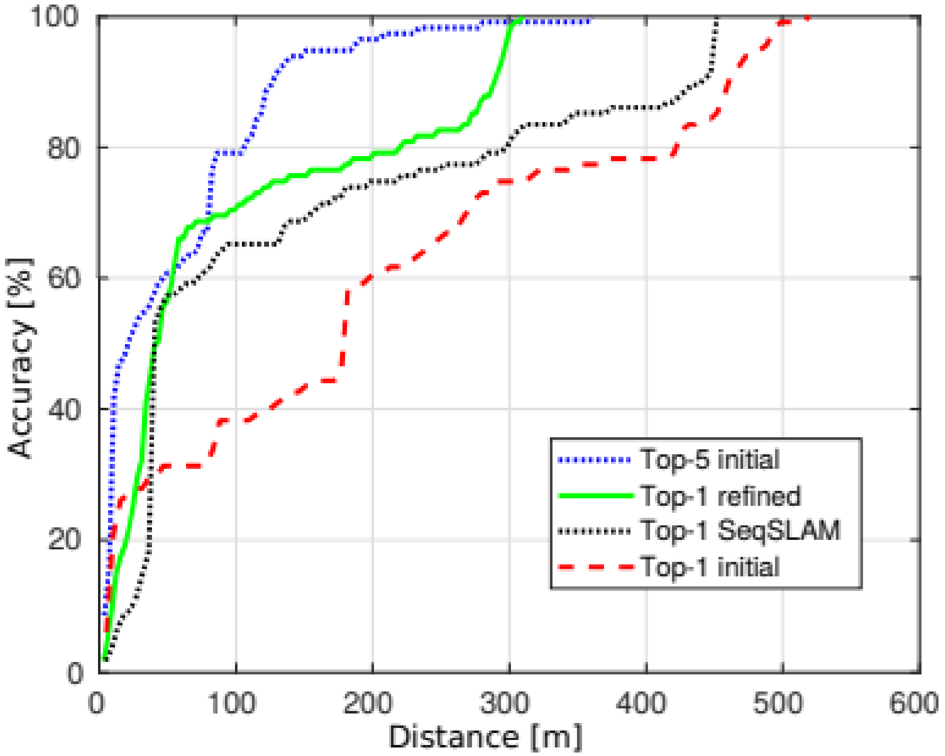
- (Left): original top-1 matches; (Right): refined top-1 matches
  between a rainy sampled reference and
  an overcast query sequence of Oxford Robotcar dataset





Compare to SeqSLAM

- Tolerance of 80m, SeqSLAM reaches 60.9% while our method reaches 68.7%
- Top 1initial and Top 5 initial are the localization without sequence information



Accuracy vs. Distance (1/2)

- Rainy sequence for reference
- benefit from map? ☹
  - uniform map vs. our map

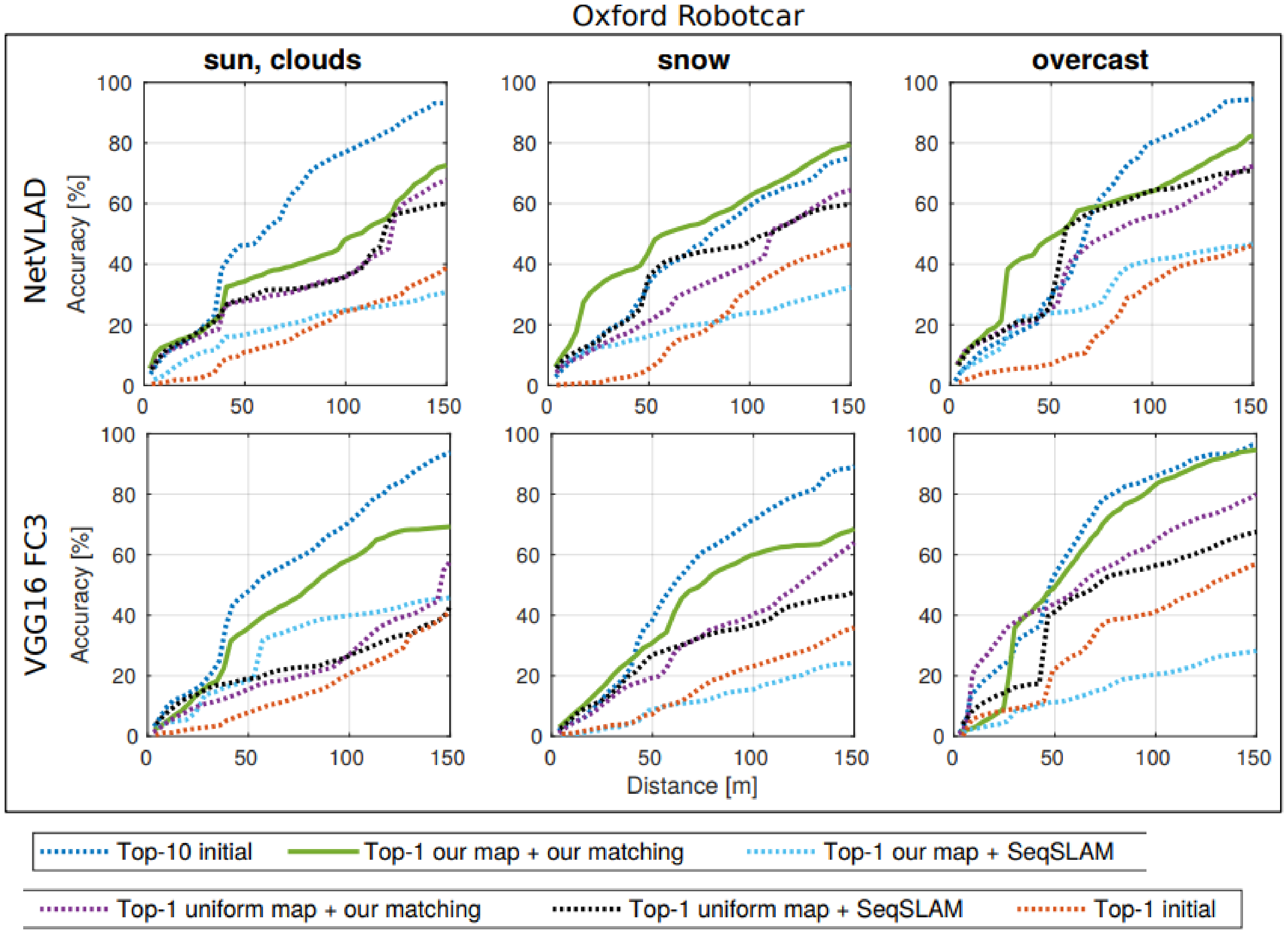
Top-1 our map + SeqSLAM

Top-1 uniform map + SeqSLAM

- benefit from matching? ☺
  - our matching vs. SeqSLAM

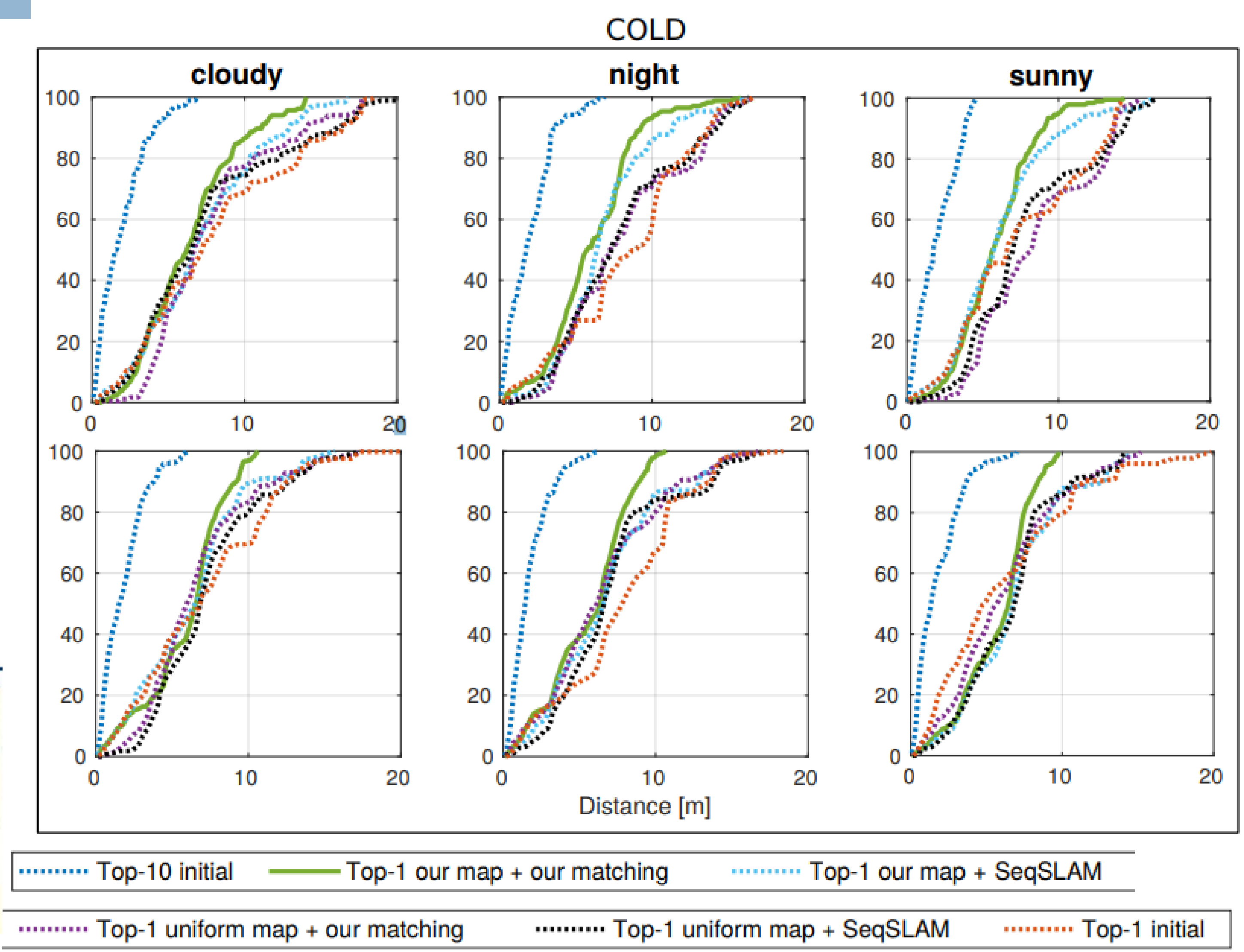
Top-1 uniform map + our matching

Top-1 uniform map + SeqSLAM



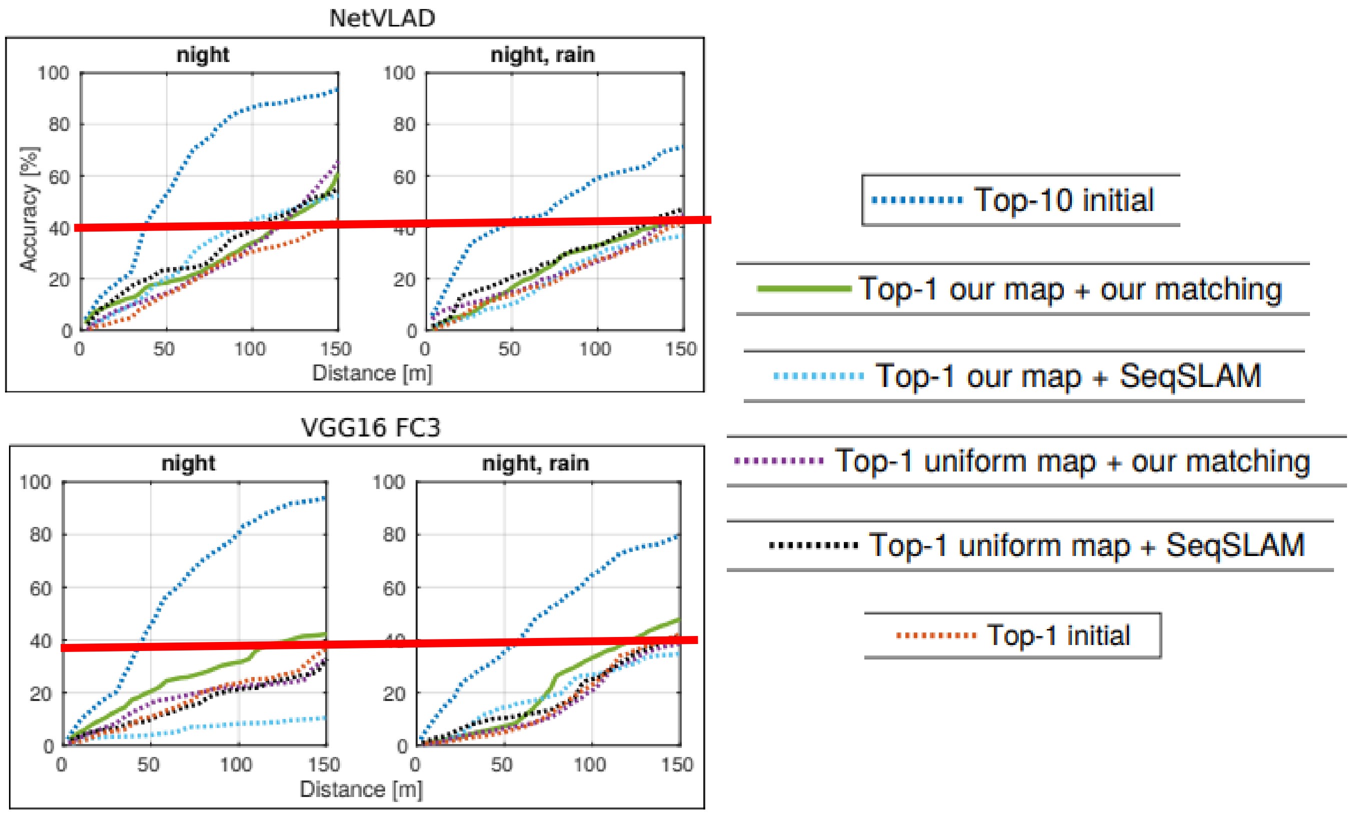
Accuracy vs. Distance (2/2)

- Second sunny sequence for reference
- First cloudy, night, sunny sequences for query



Some failure cases (ours and SeqSLAM)

- fails on sequences with non-distinctive image features, such as the outdoor night sequences in the Oxford Robotcar dataset



Outline

- Motivation
- Map representation
- Self localization
- Experimental results
- Conclusions

Conclusions

- Formulated requirements for map building, and self localization
- Based on these requirements, experiments show the benefit during localization