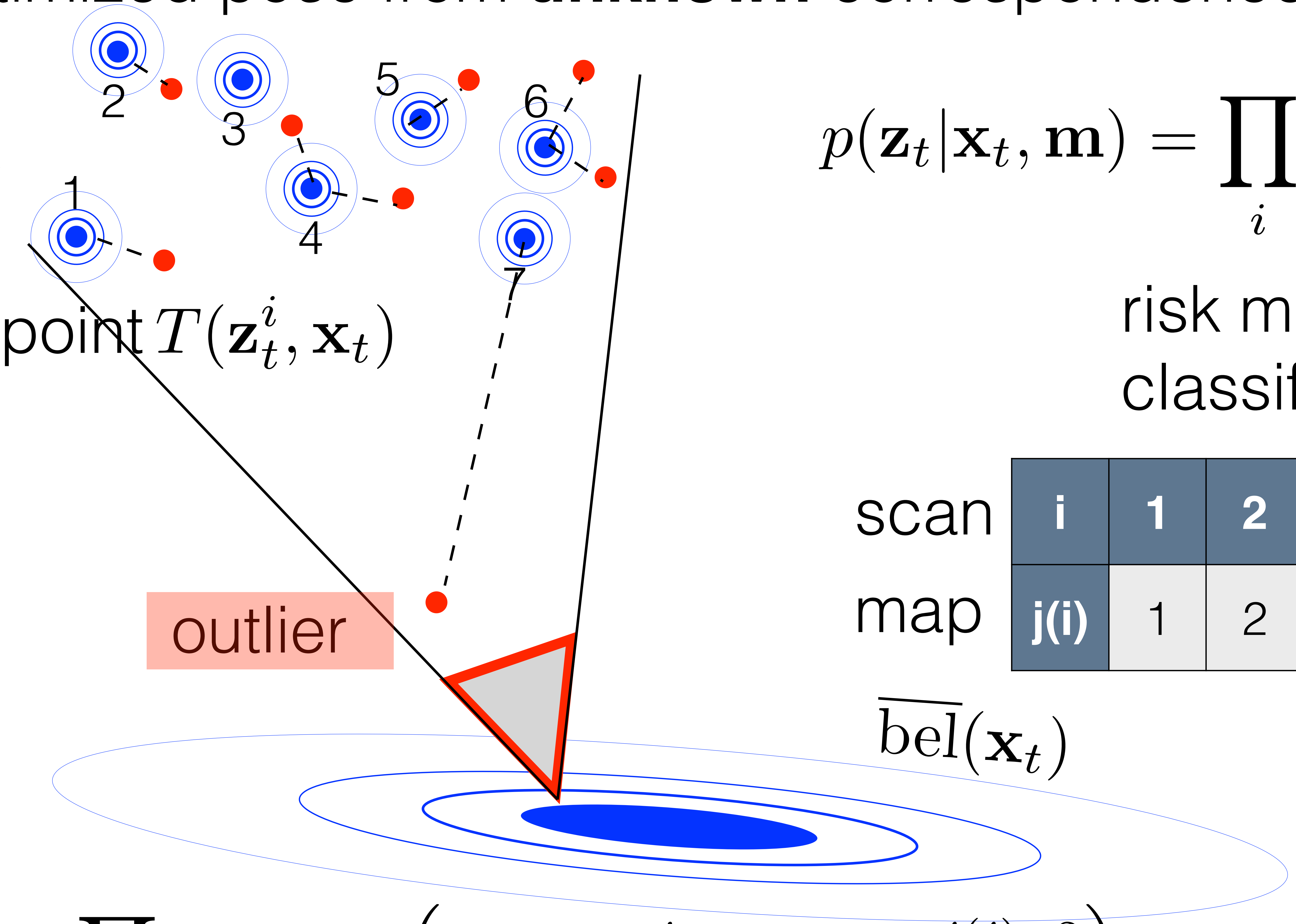
Robust regression: from ICP to RANSAC

Karel Zimmermann

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection



$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}^{j(i)})$$

risk minimizing 7-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	7

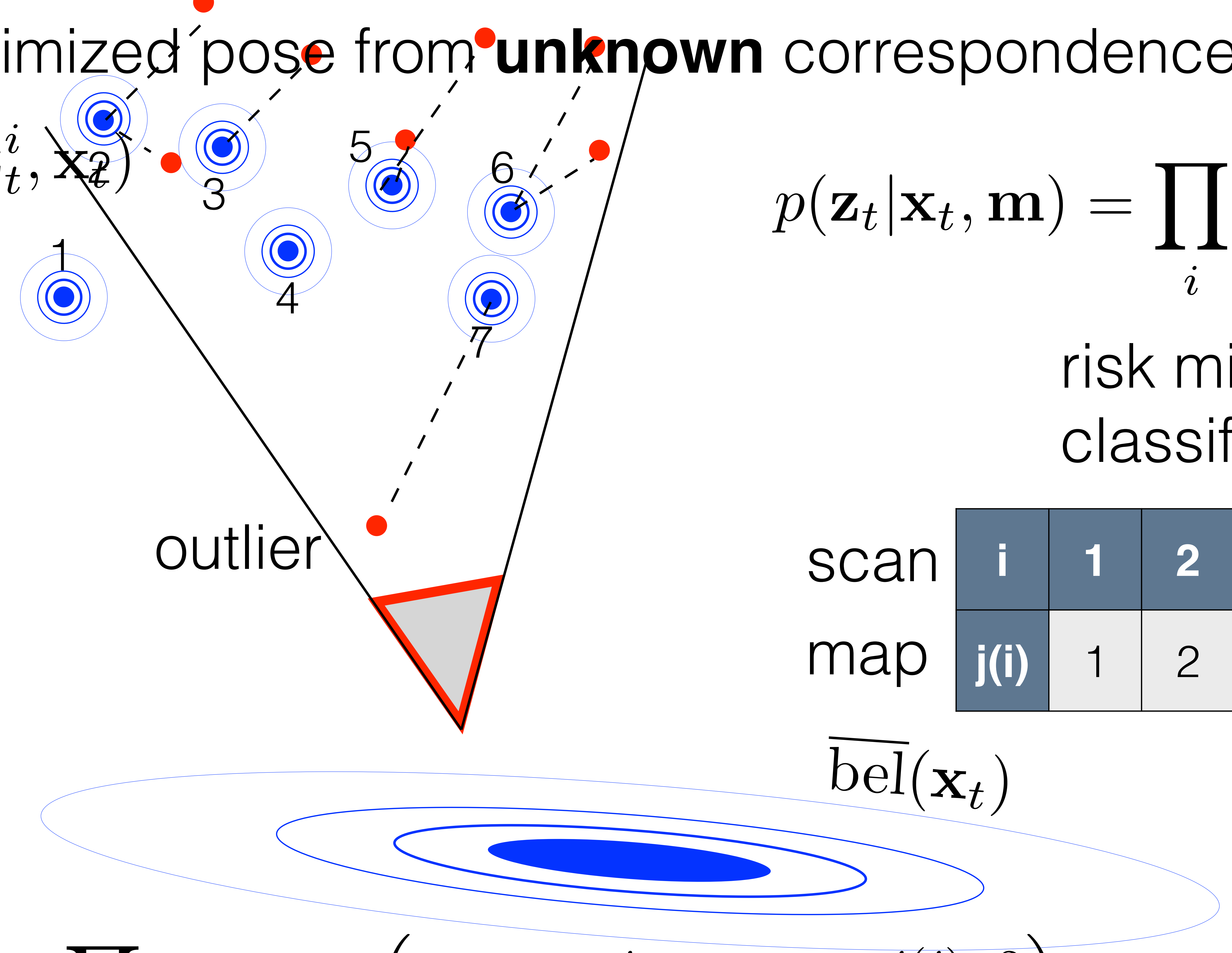
$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i K \cdot \exp\left(- \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2 \right)$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection

scan point $T(\mathbf{z}_t^i, \mathbf{x}_t)$

map point $\mathbf{m}^j$



$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}^{j(i)})$$

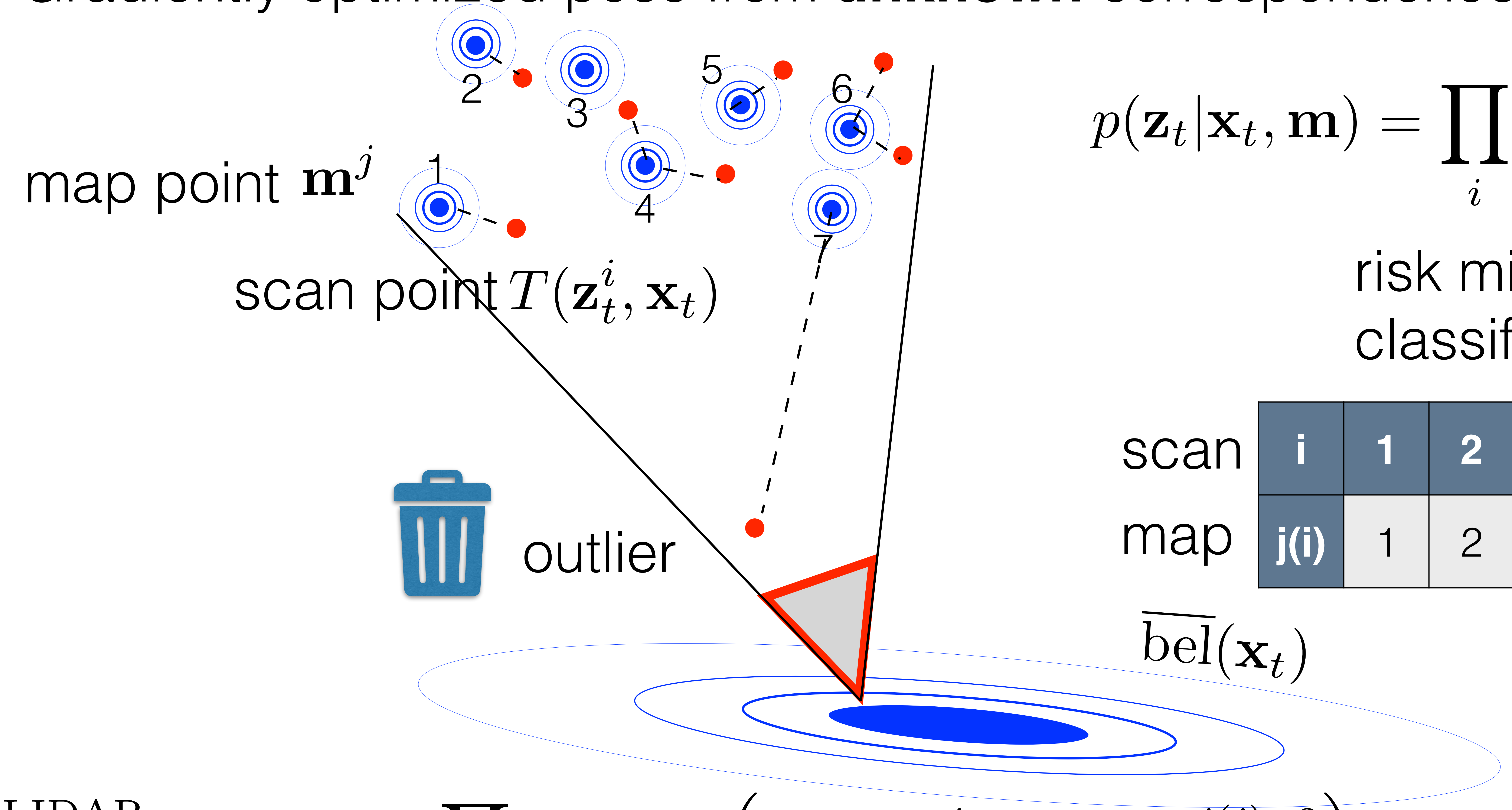
risk minimizing 7-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	7

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i K \cdot \exp\left(-\|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2 \right)$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

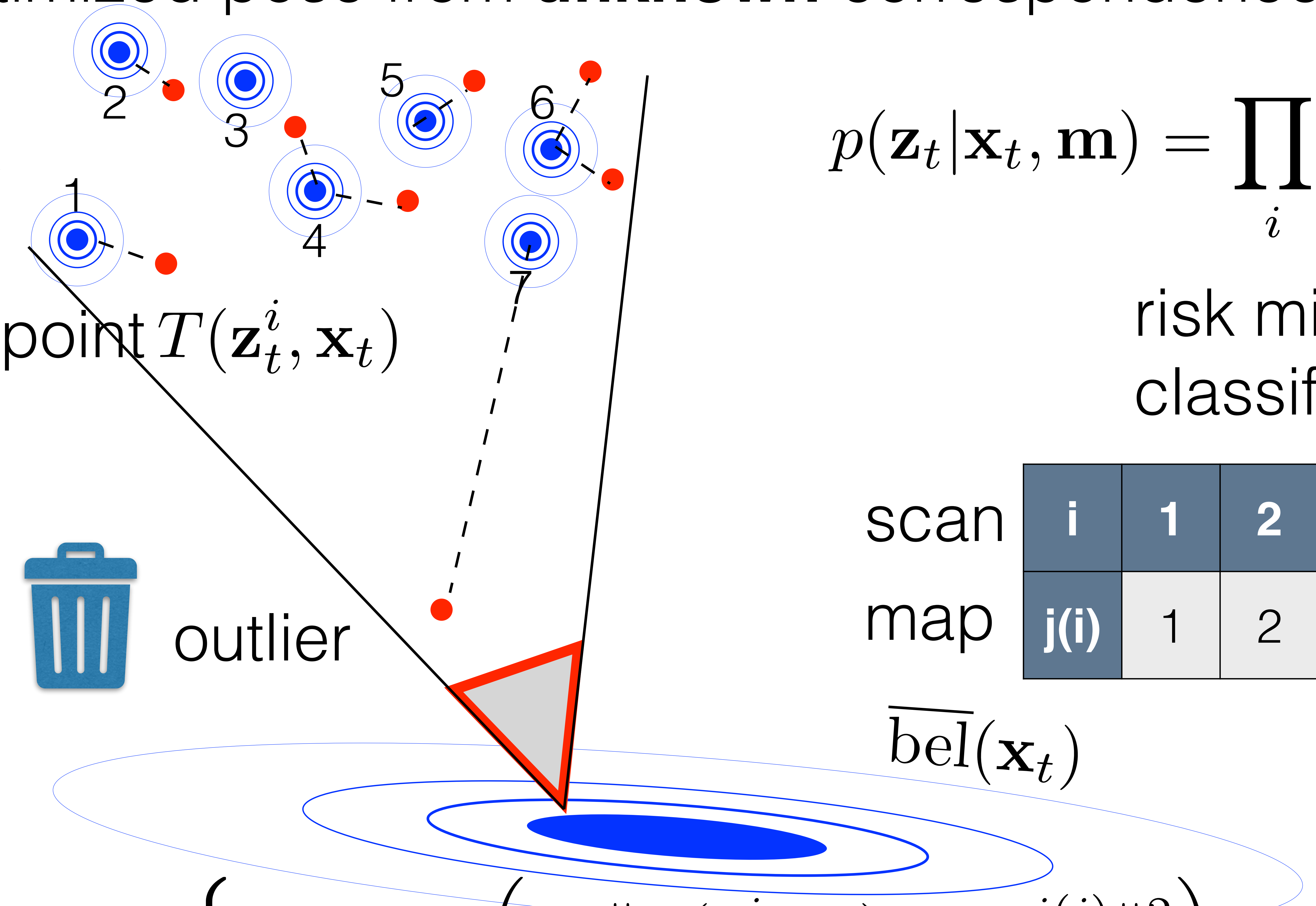
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	7

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i K \cdot \exp\left(-\|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2\right)$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

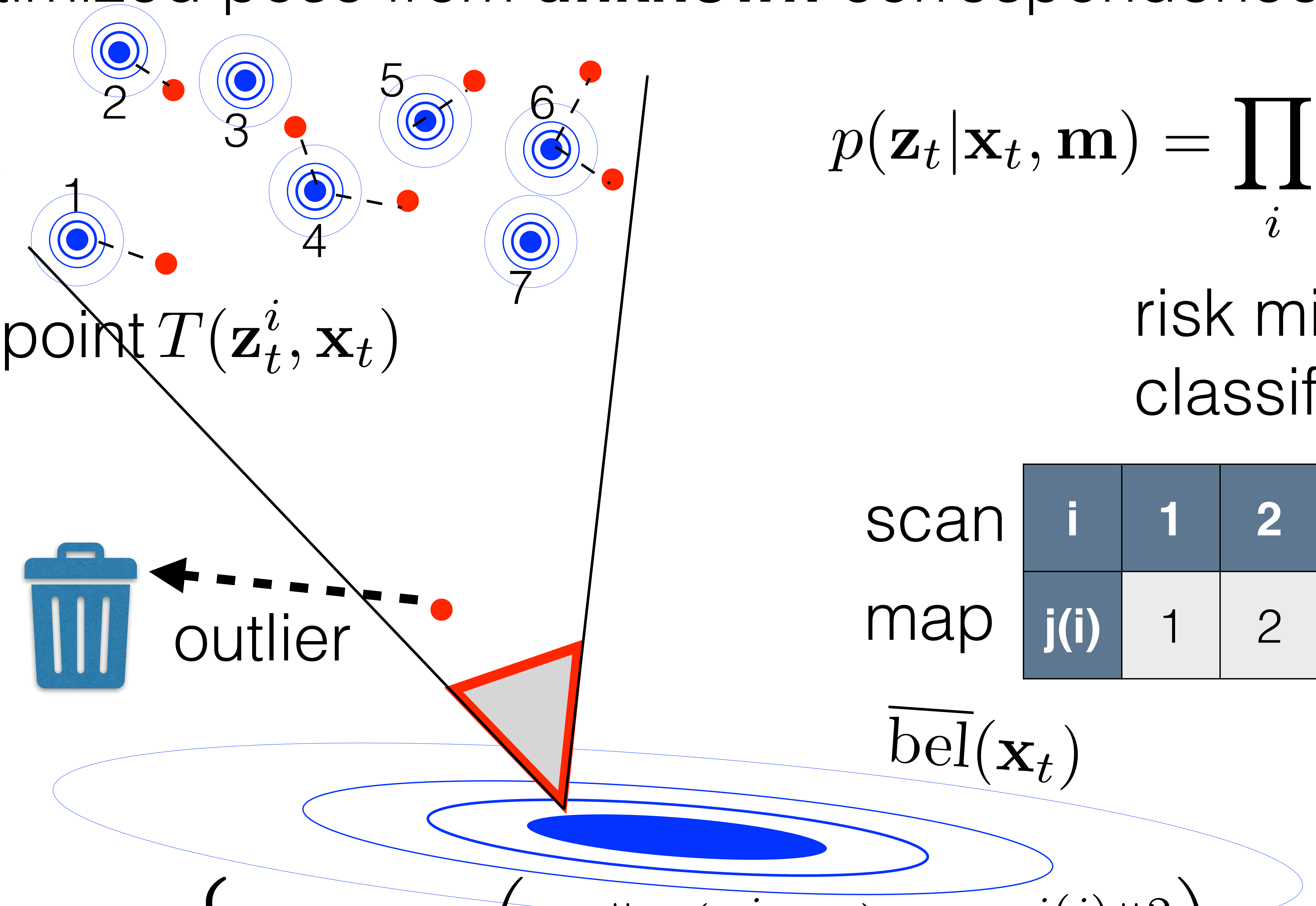
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	7

$$\mathbf{z}_t^{\mathrm{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i \begin{cases} K \cdot \exp\left(-\|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2\right) & j(i) \in [1, N] \\ p_{\mathrm{outlier}} & j(i) = N+1 \end{cases}$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

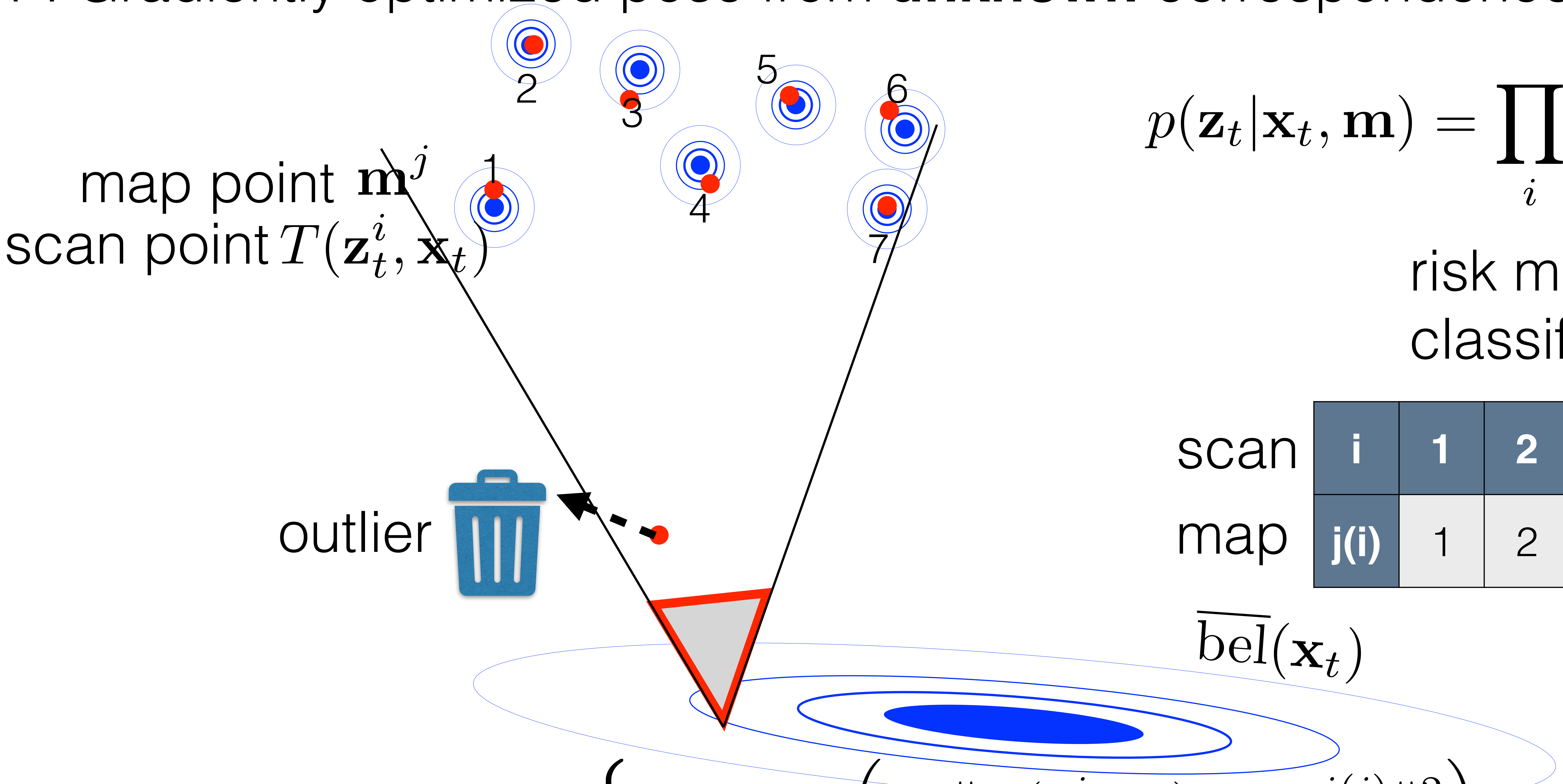
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	(trash)

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i \begin{cases} K \cdot \exp\left(- \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2 \right) & j(i) \in [1, N] \\ p_{\text{outlier}} & j(i) = N + 1 \end{cases}$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

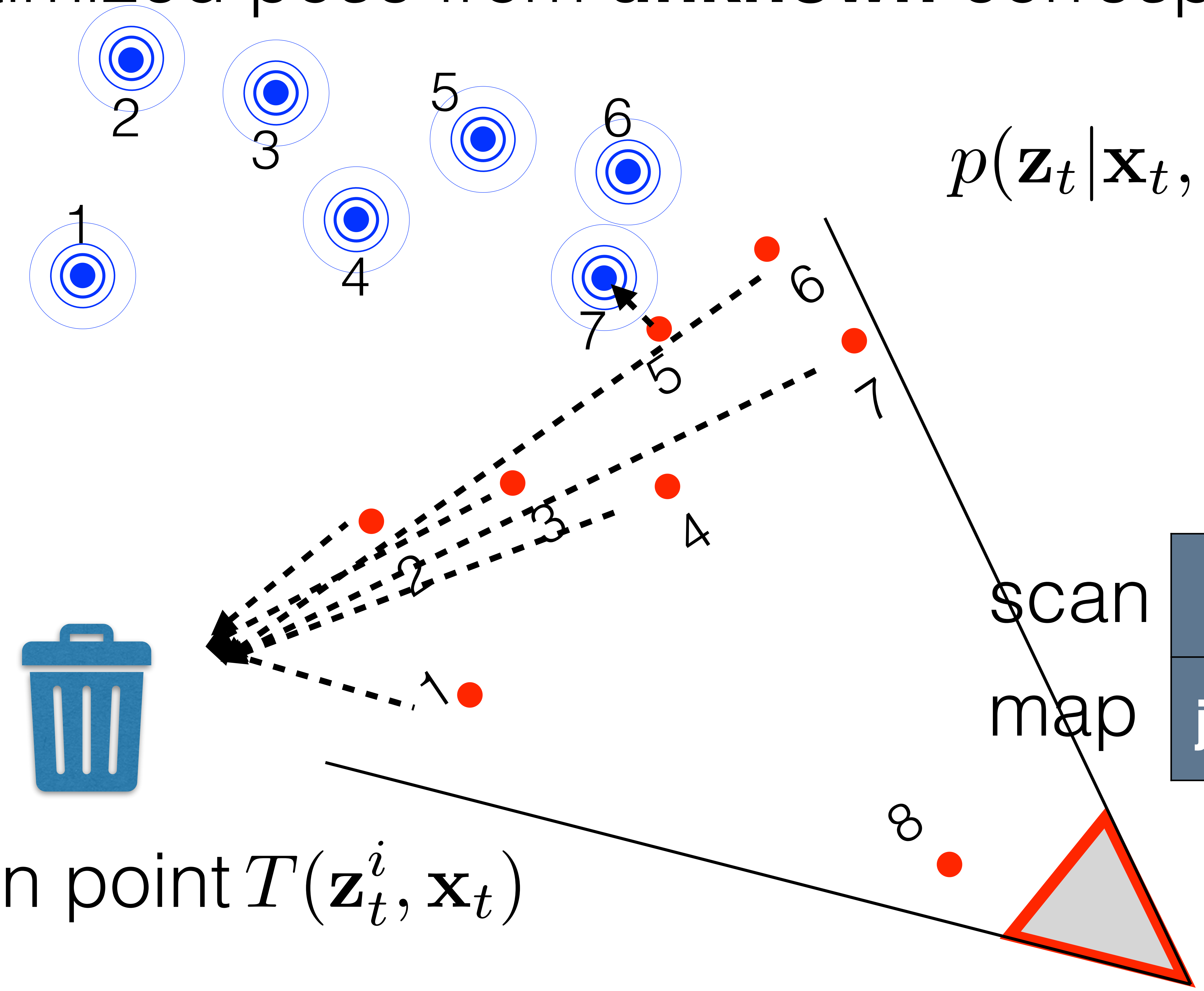
scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	4	4	5	6	6	(trash can)

$$\mathbf{z}_t^{\mathrm{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i \begin{cases} K \cdot \exp\left(- \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2 \right) & j(i) \in [1, N] \\ p_{\mathrm{outlier}} & j(i) = N+1 \end{cases}$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

ICP: Gradiently optimized pose from **unknown** correspondences with **outlier** rejection

map point $\mathbf{m}^j$



outlier

scan point $T(\mathbf{z}_t^i, \mathbf{x}_t)$

$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

scan: i	1	2	3	4	5	6	7	8
map: j(i)	🗑	🗑	🗑	🗑	7	🗑	🗑	🗑

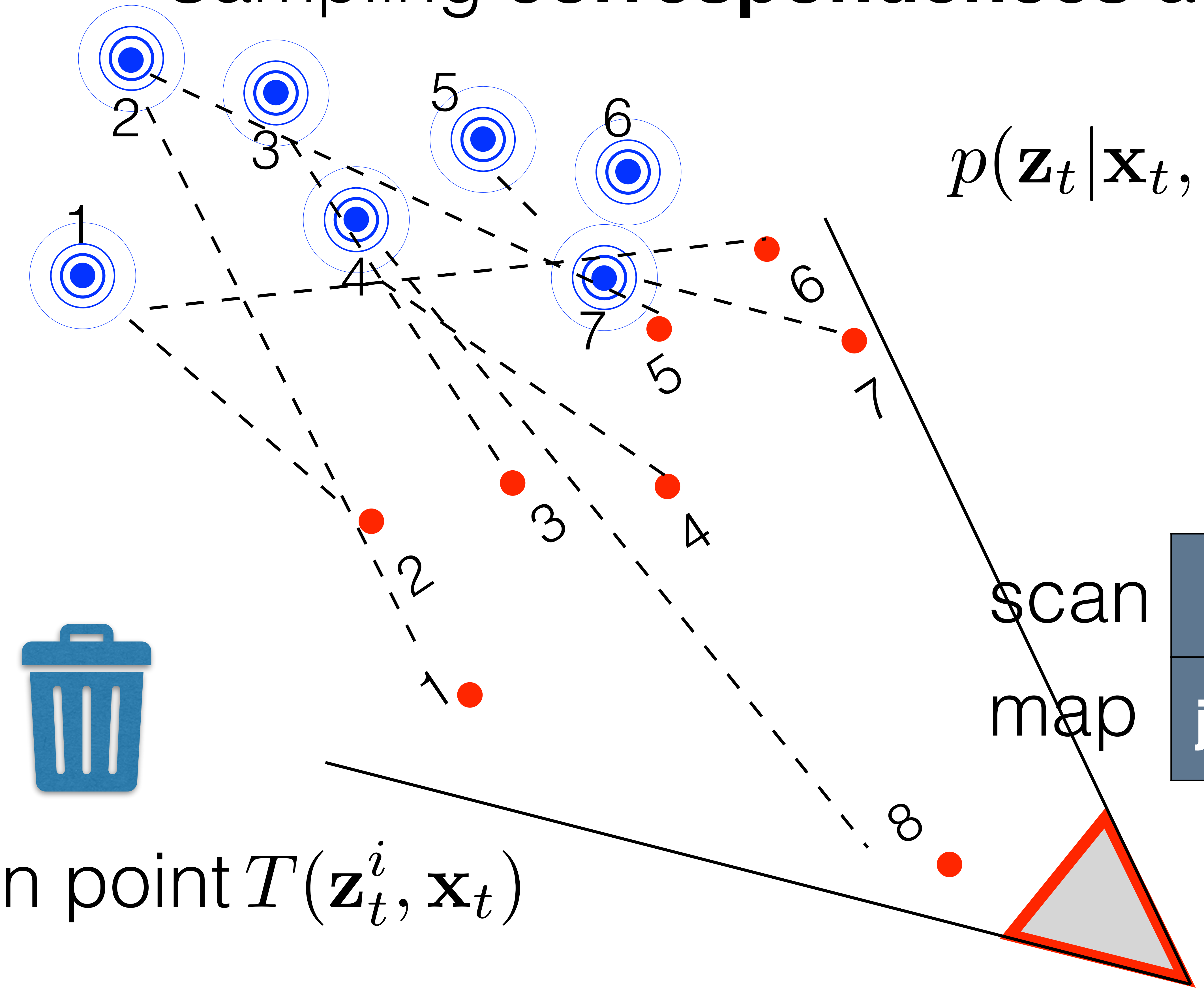
$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i \begin{cases} K \cdot \exp\left(-\|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2\right) & j(i) \in [1, N] \\ p_{\text{outlier}} & j(i) = N+1 \end{cases}$$

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem **?????????**
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Absolute orientation problem

v1.0

Sampling **correspondences** at random

map point $\mathbf{m}^j$



outlier

scan point $T(\mathbf{z}_t^i, \mathbf{x}_t)$

$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}^{j(i)})$$

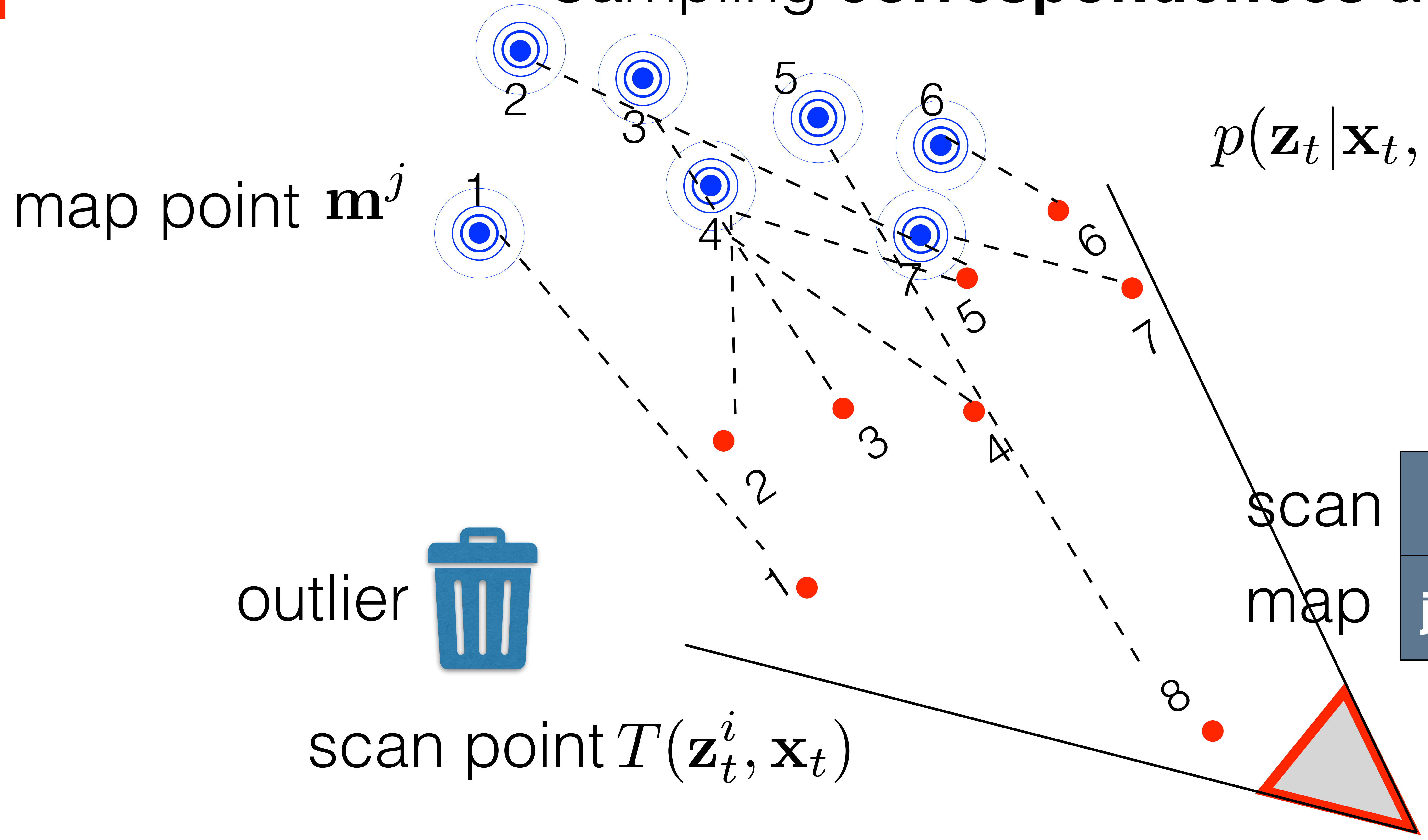
risk minimizing **8**-class classification problem

scan i	1	2	3	4	5	6	7	8
map j(i)	2	1	3	4	2	1	7	4

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t, j(i)} \prod_i \begin{cases} K \cdot \exp\left(- \| T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)} \|_2^2 \right) & j(i) \in [1, N] \\ p_{\text{outlier}} & j(i) = N + 1 \end{cases}$$

1. Sample $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_i \| T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)} \|_2^2$ … Known absolute orientation problem

Sampling **correspondences** at random



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

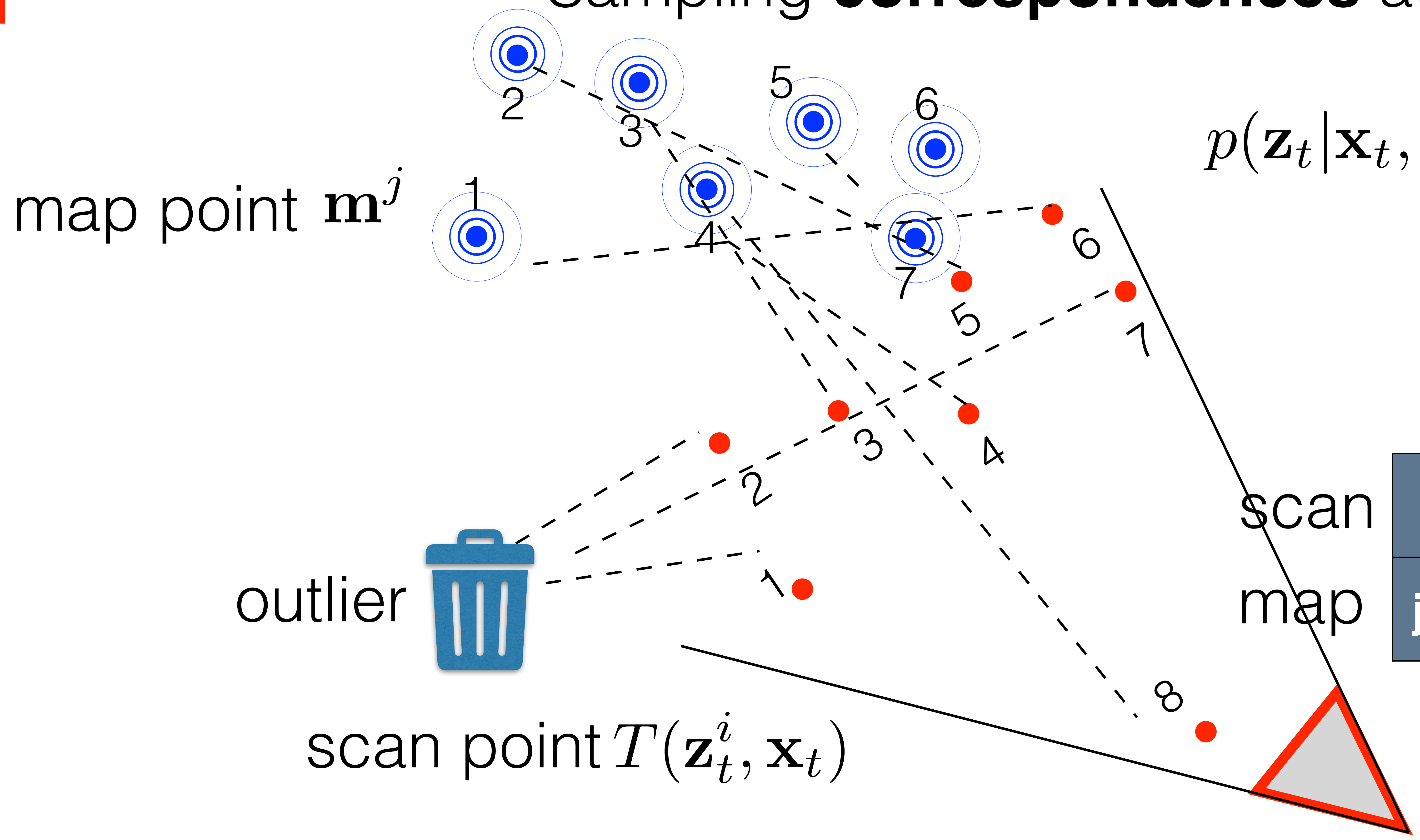
risk minimizing **8**-class classification problem

	i	1	2	3	4	5	6	7	8
scan									
map	j(i)	1	4	3	4	4	6	7	5

1. Sample $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v1.0

Sampling **correspondences** at random



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

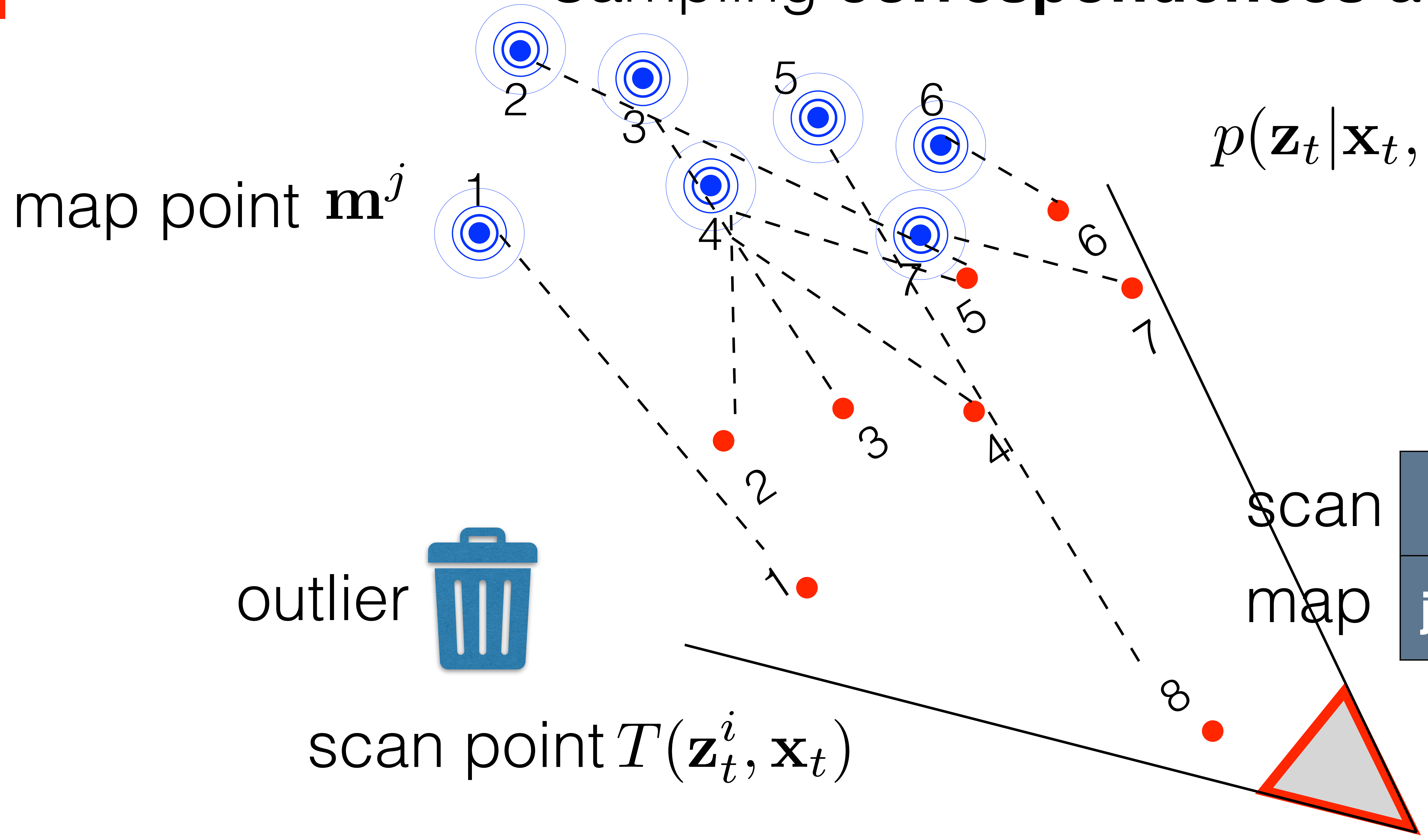
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	🗑	🗑	3	4	2	1	🗑	4

1. Sample $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v1.0

Sampling **correspondences** at random



$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}^{j(i)})$$

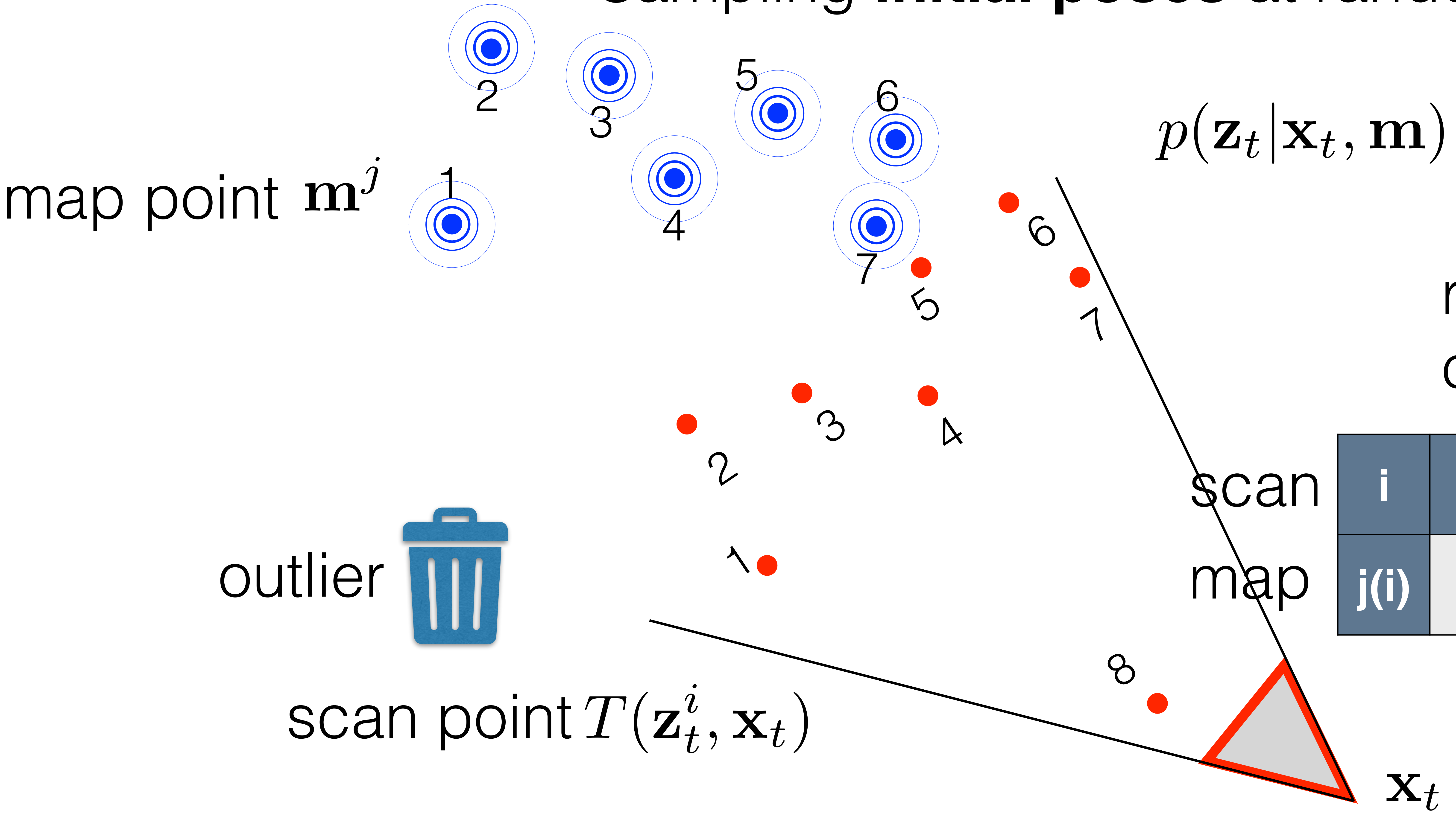
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	4	3	4	4	6	7	5

1. Sample $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v2.0

Sampling **initial poses** at random



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

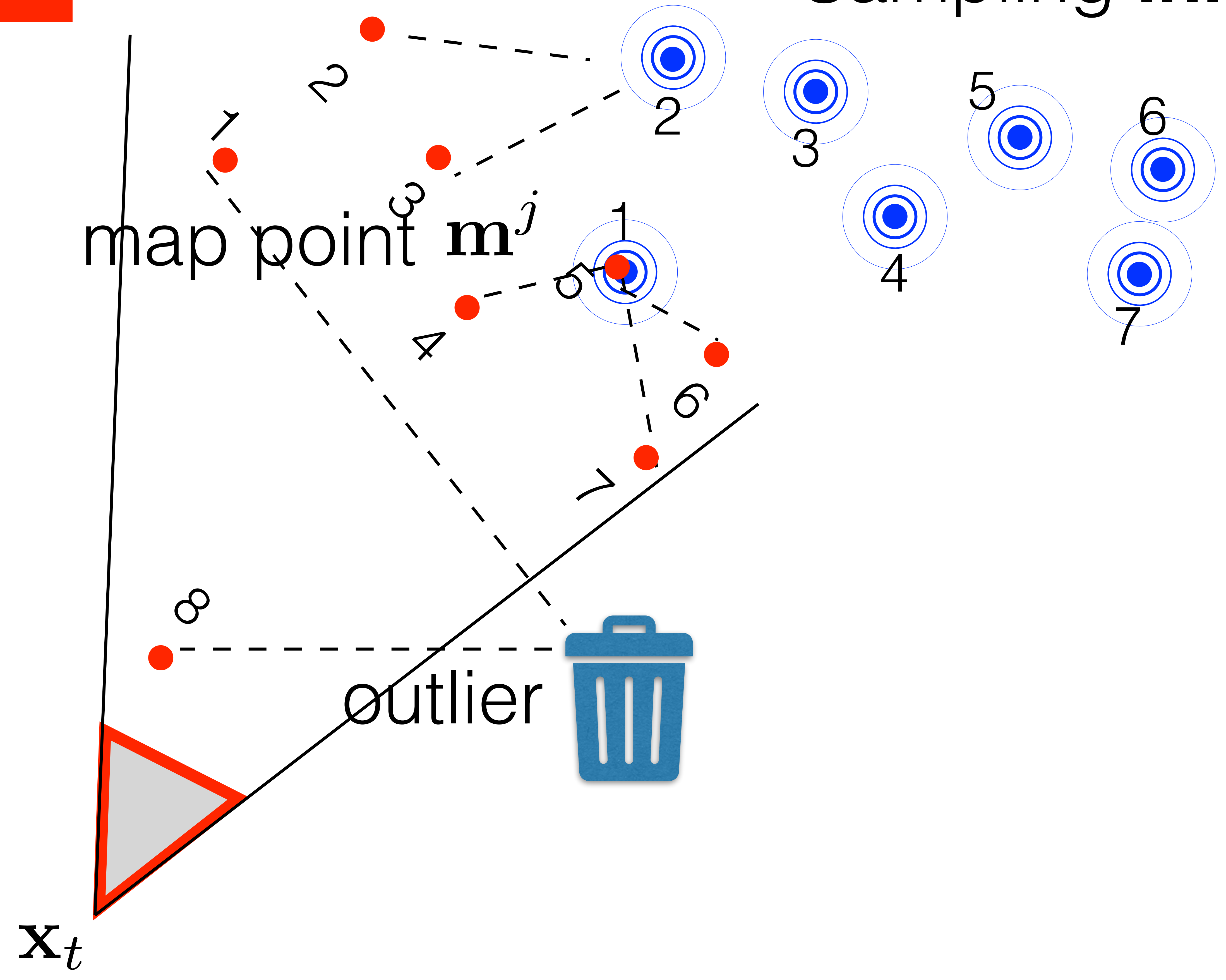
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	4	3	4	4	6	7	5

0. Sample initial pose $\mathbf{x}_t$ at random.
1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v2.0

Sampling **initial poses** at random



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	4	3	4	4	6	7	5

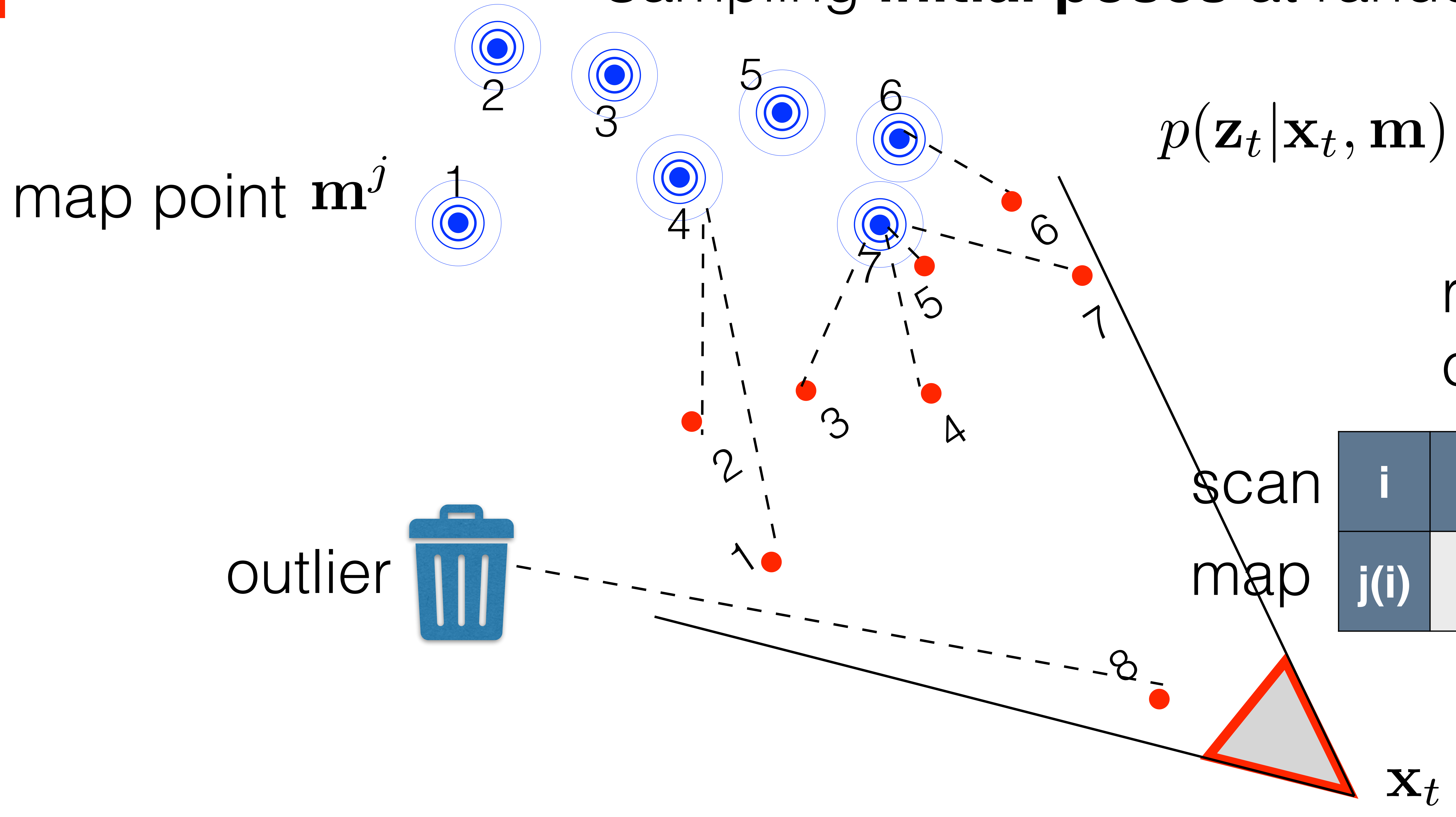
0. Sample initial pose $\mathbf{x}_t$ at random.

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i,\mathbf{x}_t)-\mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem
2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i,\mathbf{x}_t)-\mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v2.0

Sampling **initial poses** at random

$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$



risk minimizing **8**-class classification problem

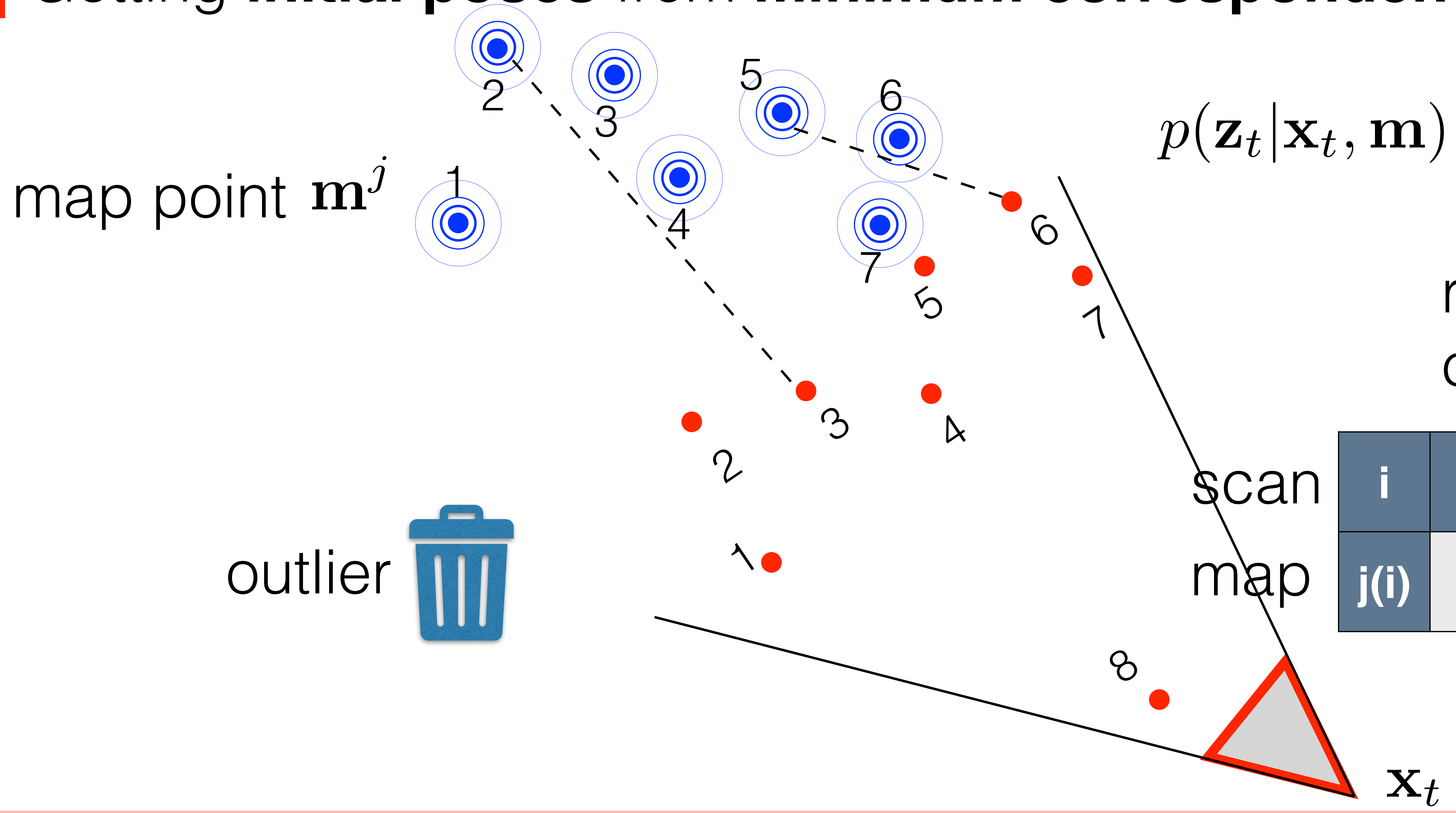
scan i	1	2	3	4	5	6	7	8
map j(i)	1	4	3	4	4	6	7	5

0. Sample initial pose $\mathbf{x}_t$ at random.

1. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem

2. $\arg\min_{\mathbf{x}_t} \sum_i \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem

v3.0 Getting **initial poses** from **minimum correspondences** sampled at random



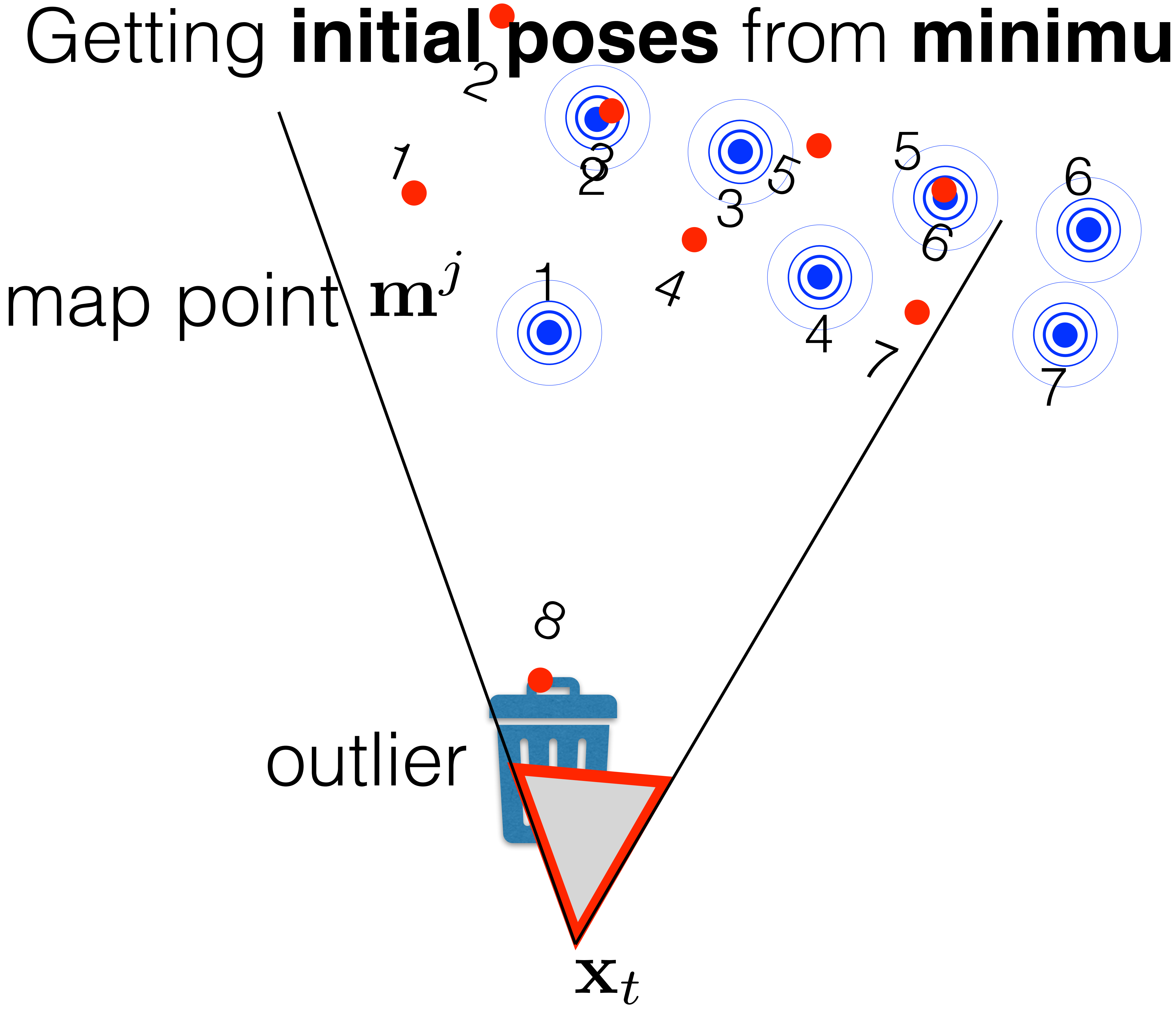
$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**			2			5		

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem for all
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem for all

v3.0 Getting **initial poses** from **minimum correspondences** sampled at random



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

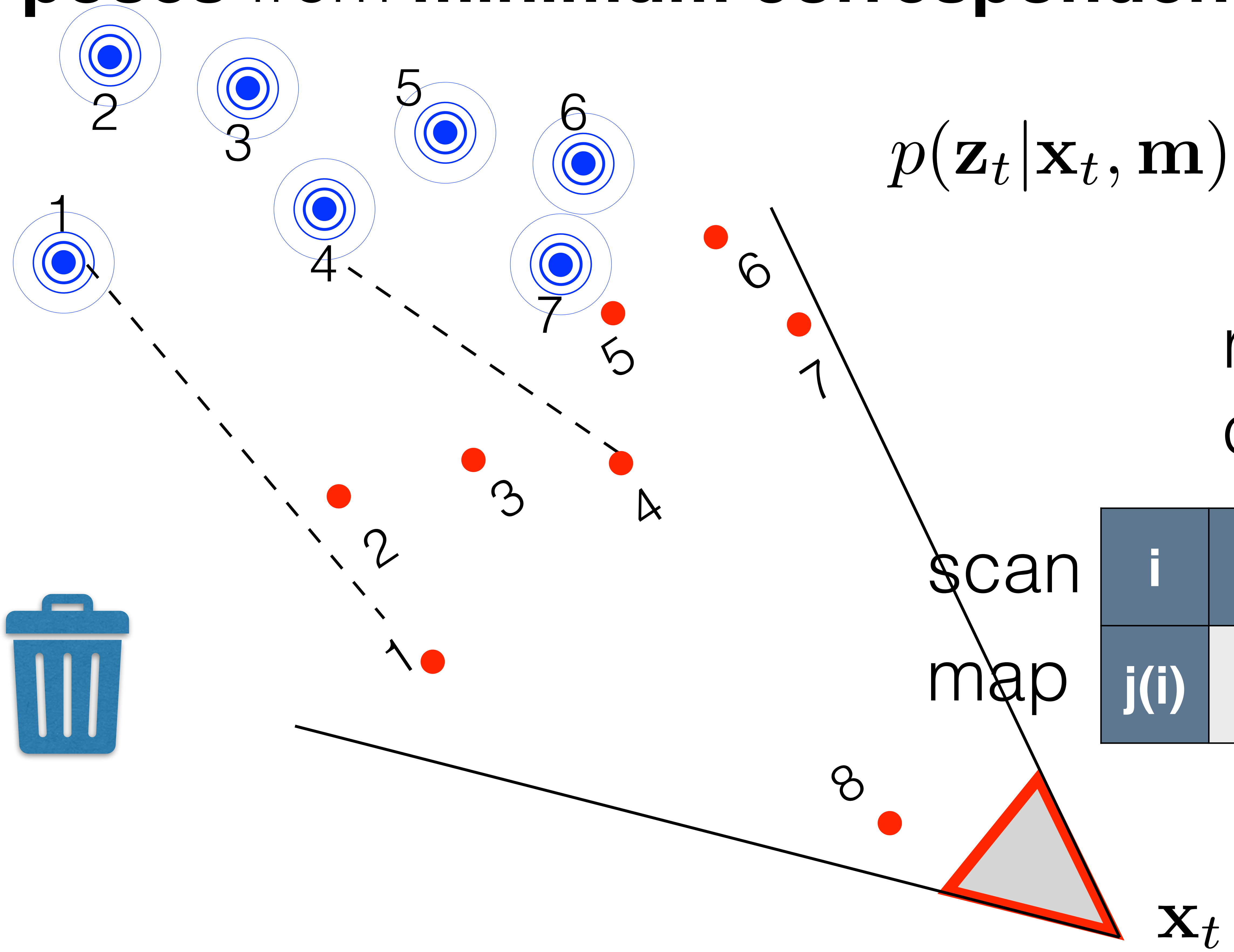
	i	1	2	3	4	5	6	7	8
scan									
map	j(i)			2			5		

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i\in S} \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem for all
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem for all

v3.0 Getting **initial poses** from **minimum correspondences** sampled at random

map point $\mathbf{m}^j$

outlier



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

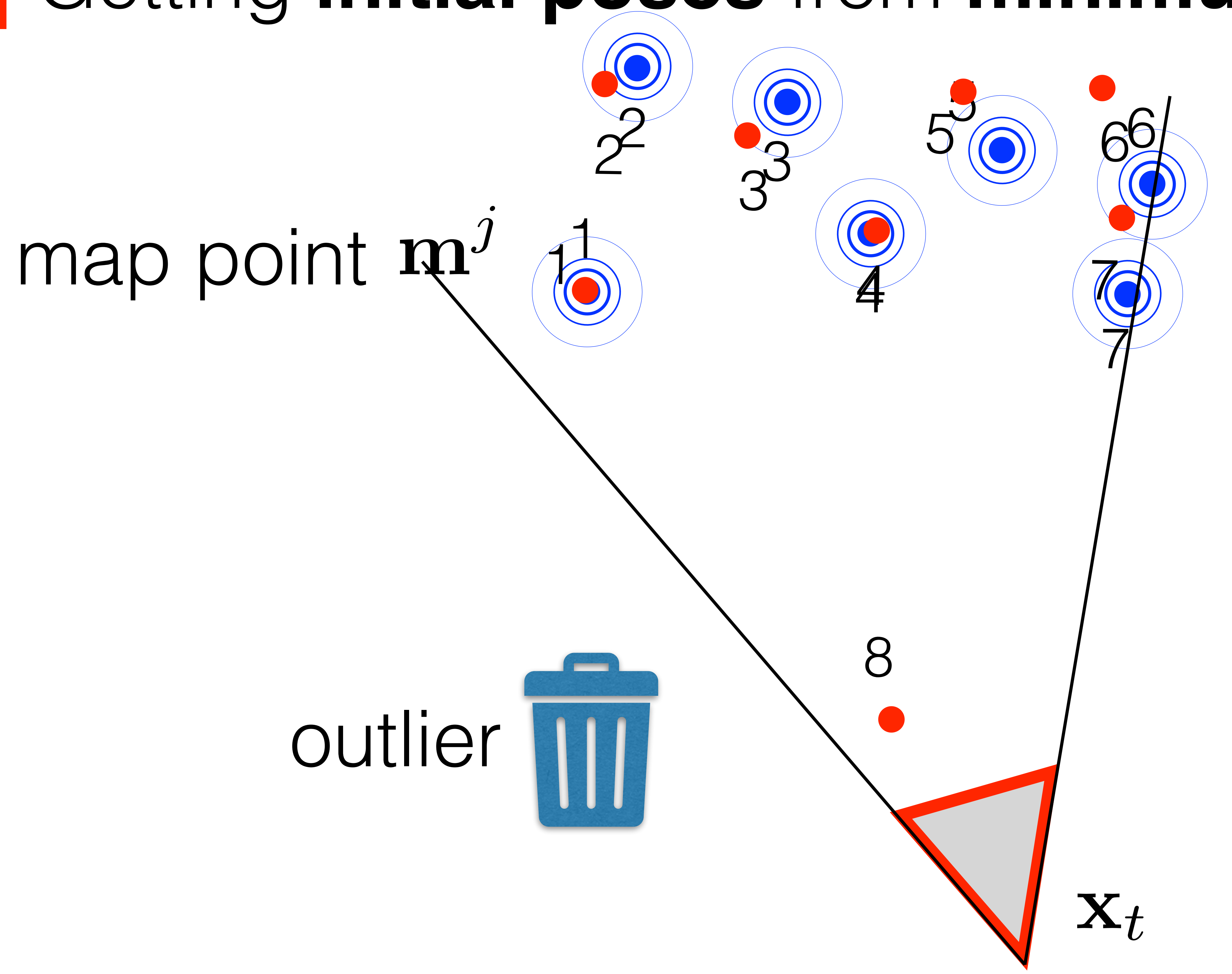
risk minimizing **8**-class classification problem

scan: **i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map: **j(i)**			2			5		

clean sample

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem for all
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem for all

v3.0 Getting **initial poses** from **minimum correspondences** sampled at random



$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}^{j(i)})$$

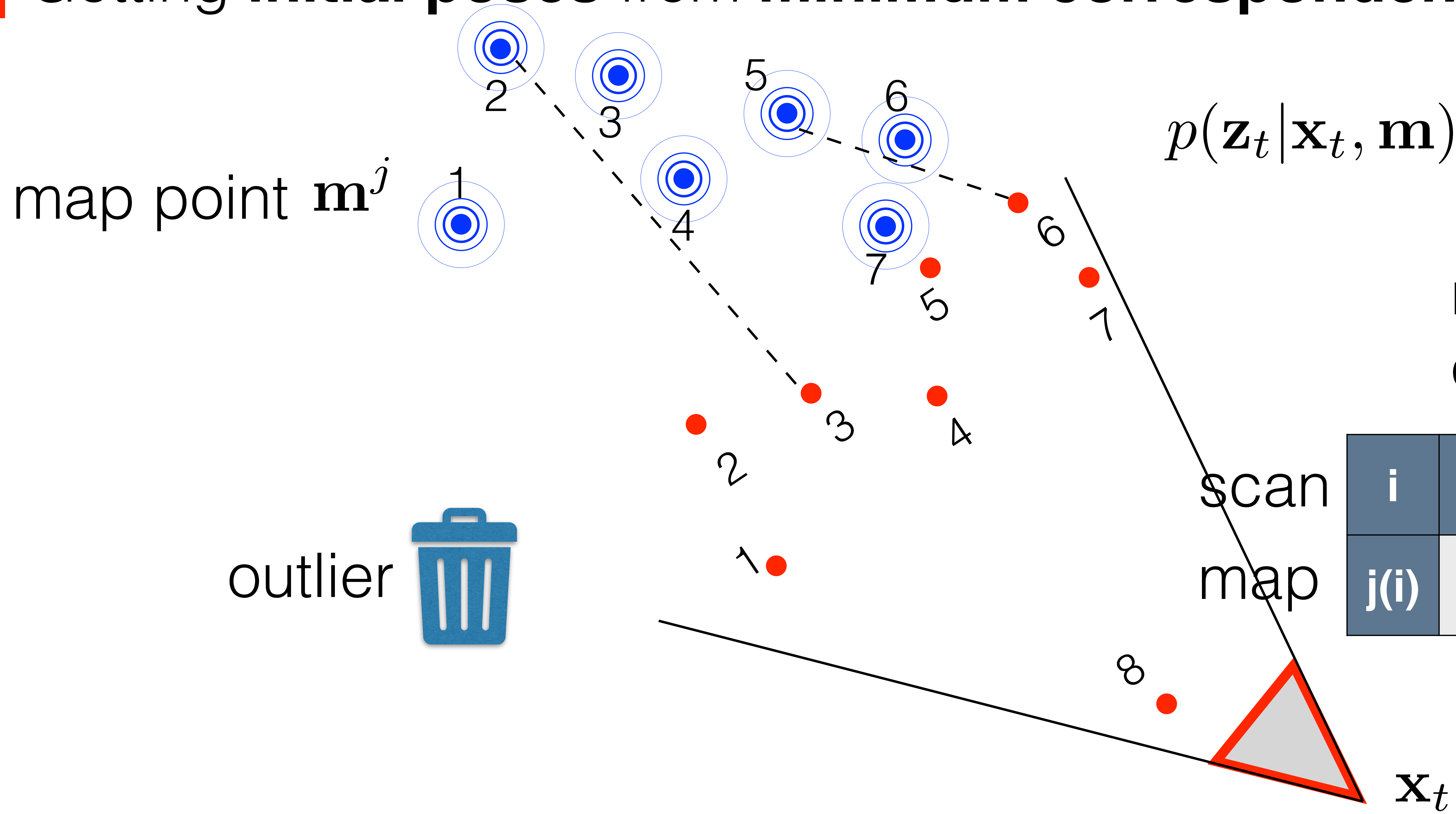
risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	4	3	4	4	6	7	5

clean sample

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem for all
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem for all

v3.0 Getting **initial poses** from **minimum correspondences** sampled at random



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

scan i	1	2	3	4	5	6	7	8
map j(i)			2			5		

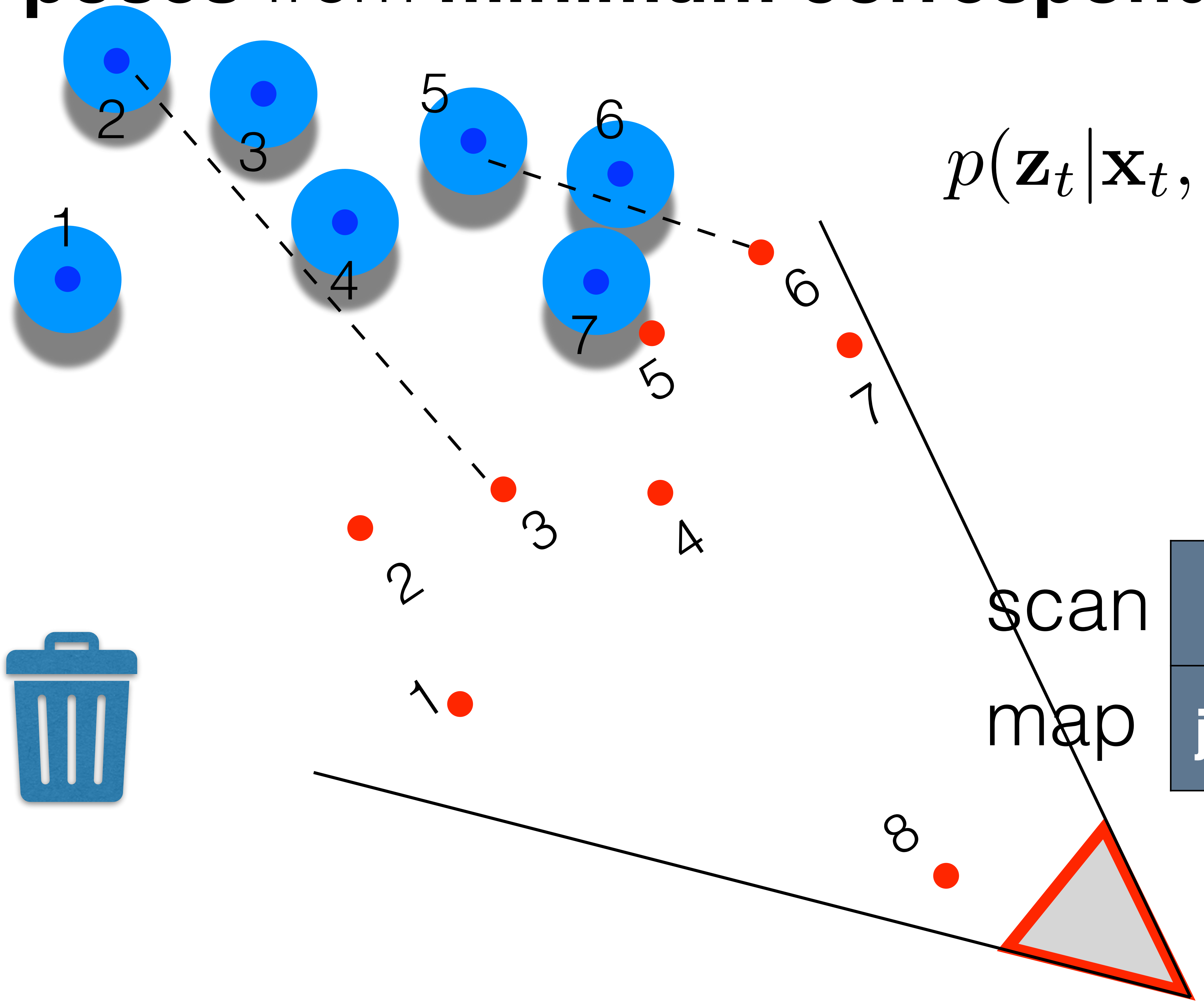
speed up

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i\in S} \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. $\arg\min_{j(i)} \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Nearest neighbour problem for all
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i,\mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Known absolute orientation problem for all

SLOW SLOW SLOW SLOW SLOW

v4.0 Getting **initial poses** from **minimum correspondences** sampled at random

map point $\mathbf{m}^j$



outlier

$\mathbf{x}_t$

$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

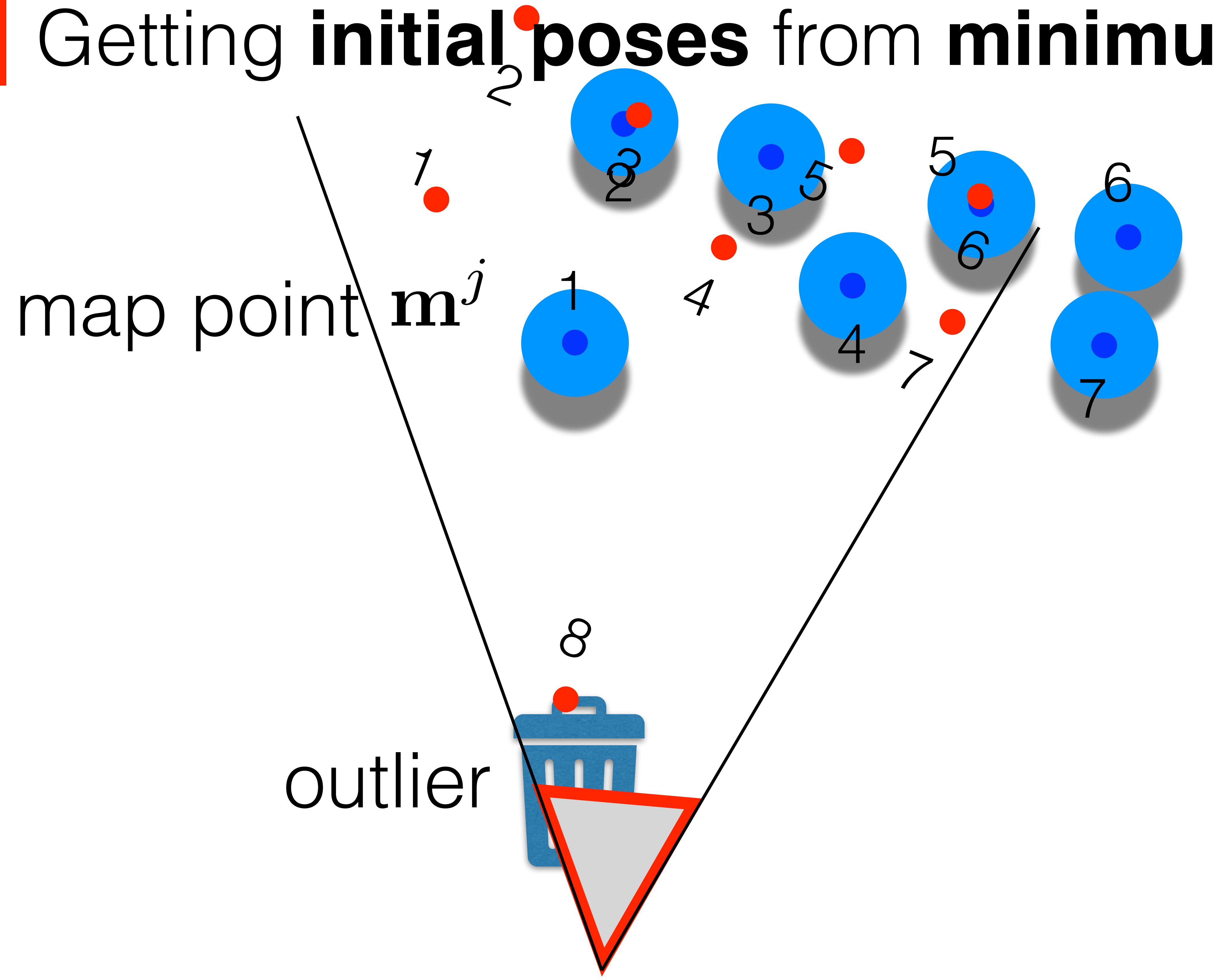
risk minimizing **8**-class classification problem

	i	1	2	3	4	5	6	7	8
scan									
map	j(i)			2			5		

simplified probability distr.

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.

v4.0 Getting **initial poses** from **minimum correspondences** sampled at random



$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

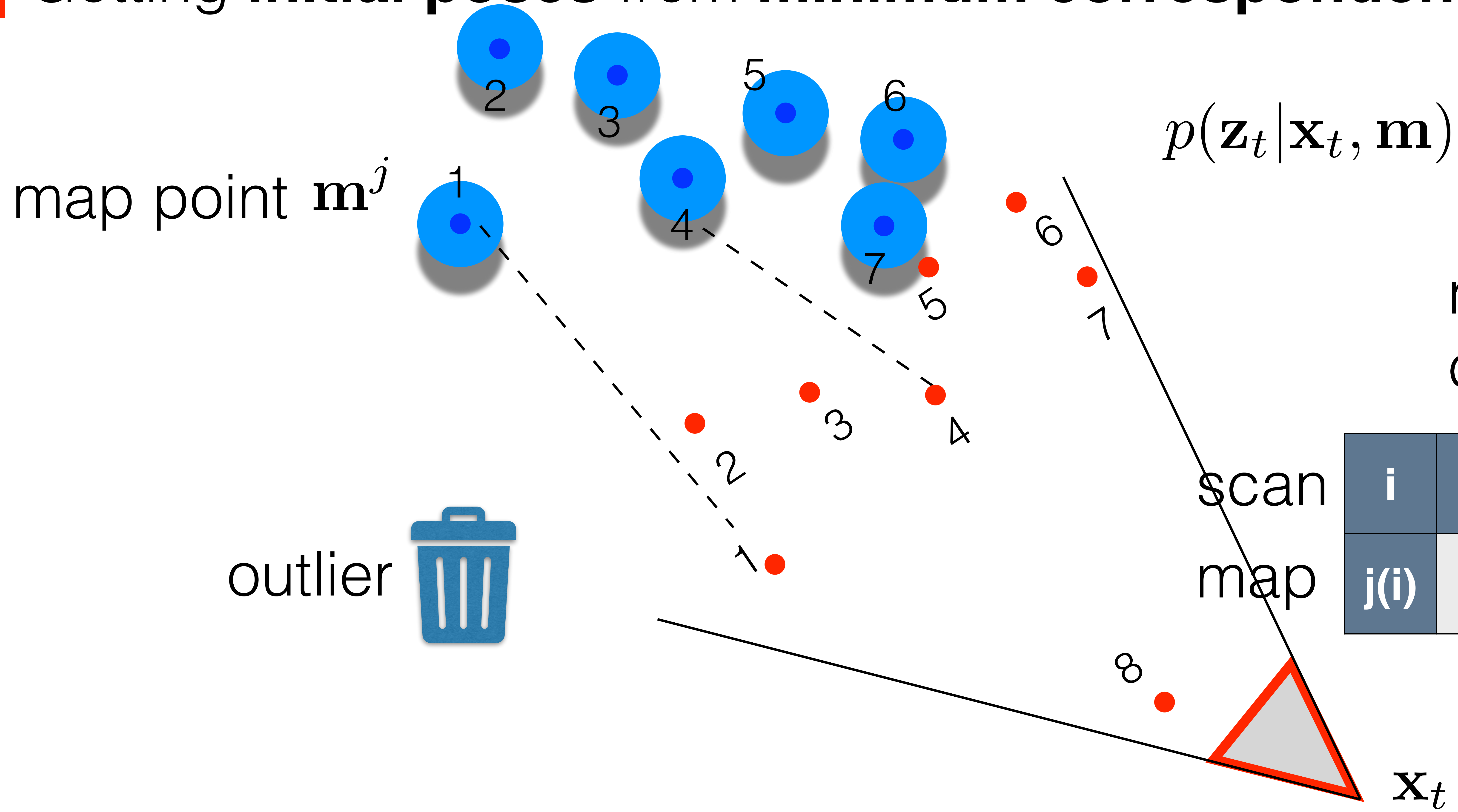
risk minimizing **8**-class classification problem

scan	i	1	2	3	4	5	6	7	8
map	j(i)			2			5		

$\mathbf{x}_t$

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. Prob. of pose being correct ~ number of **scan** points within **map point neigh** **=2**

v4.0 Getting **initial poses** from **minimum correspondences** sampled at random



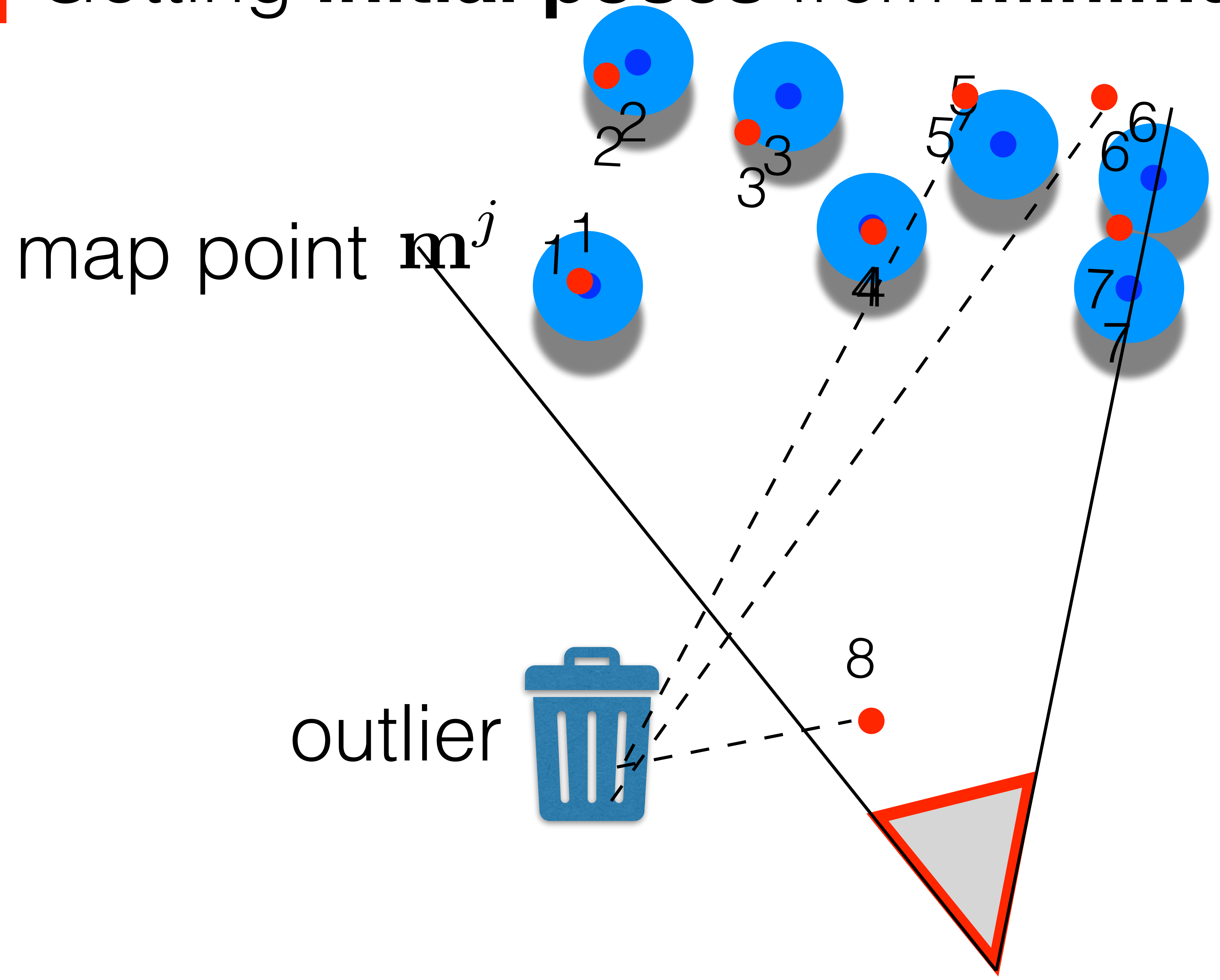
$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t, \mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

	i	1	2	3	4	5	6	7	8
scan									
map	j(i)	1			4				

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. Prob. of pose being correct ~ number of **scan** points within **map point neigh**

v4.0 Getting **initial poses** from **minimum correspondences** sampled at random



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

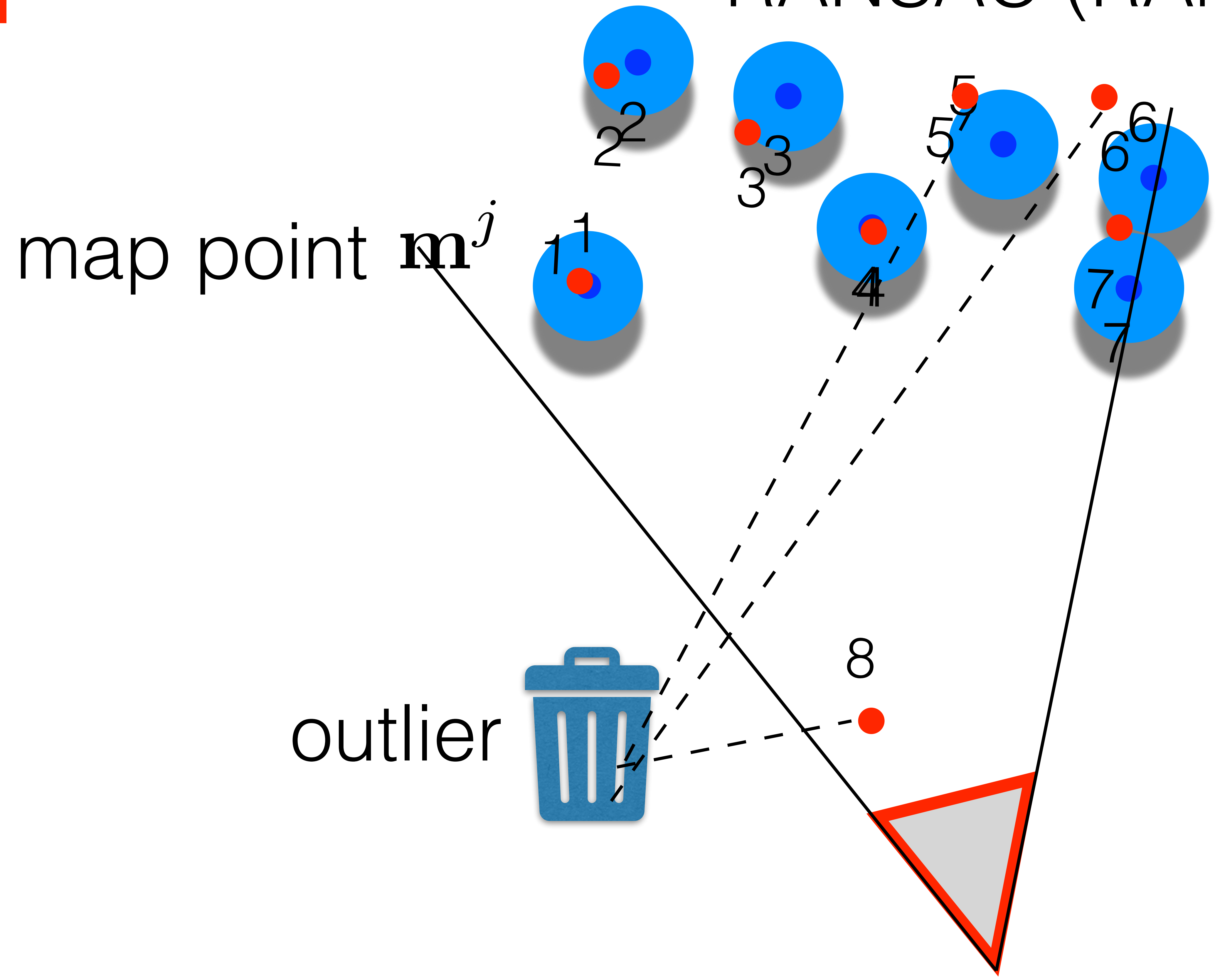
scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1			4				

$\mathbf{x}_t$

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. Prob. of pose being correct ~ number of **scan** points within **map point neigh** **=5**

v4.0

RANSAC (RAndom SAmple Consensus)



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

risk minimizing **8**-class classification problem

scan	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
map	**j(i)**	1	2	3	4	🗑	🗑	7	🗑

$\mathbf{x}_t$

1. Sample minimum subset S of correspondences $j(i)$ at random.
2. $\arg\min_{\mathbf{x}_t} \sum_{i \in S} \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Get pose hypothesis $\mathbf{x}_t$ from S.
3. Prob. of pose being correct ~ number of **scan** points within **map point neigh** **=5**
4. $\arg\min_{\mathbf{x}_t} \sum \|T(\mathbf{z}_t^i, \mathbf{x}_t) - \mathbf{m}^{j(i)}\|_2^2$ … Solve absolute orientation from inliers

v4.0

RANSAC (RAndom SAmple Consensus)

K … number of trials/iterations

p … probability, that we have selected a clean sample at least once out of K trials.

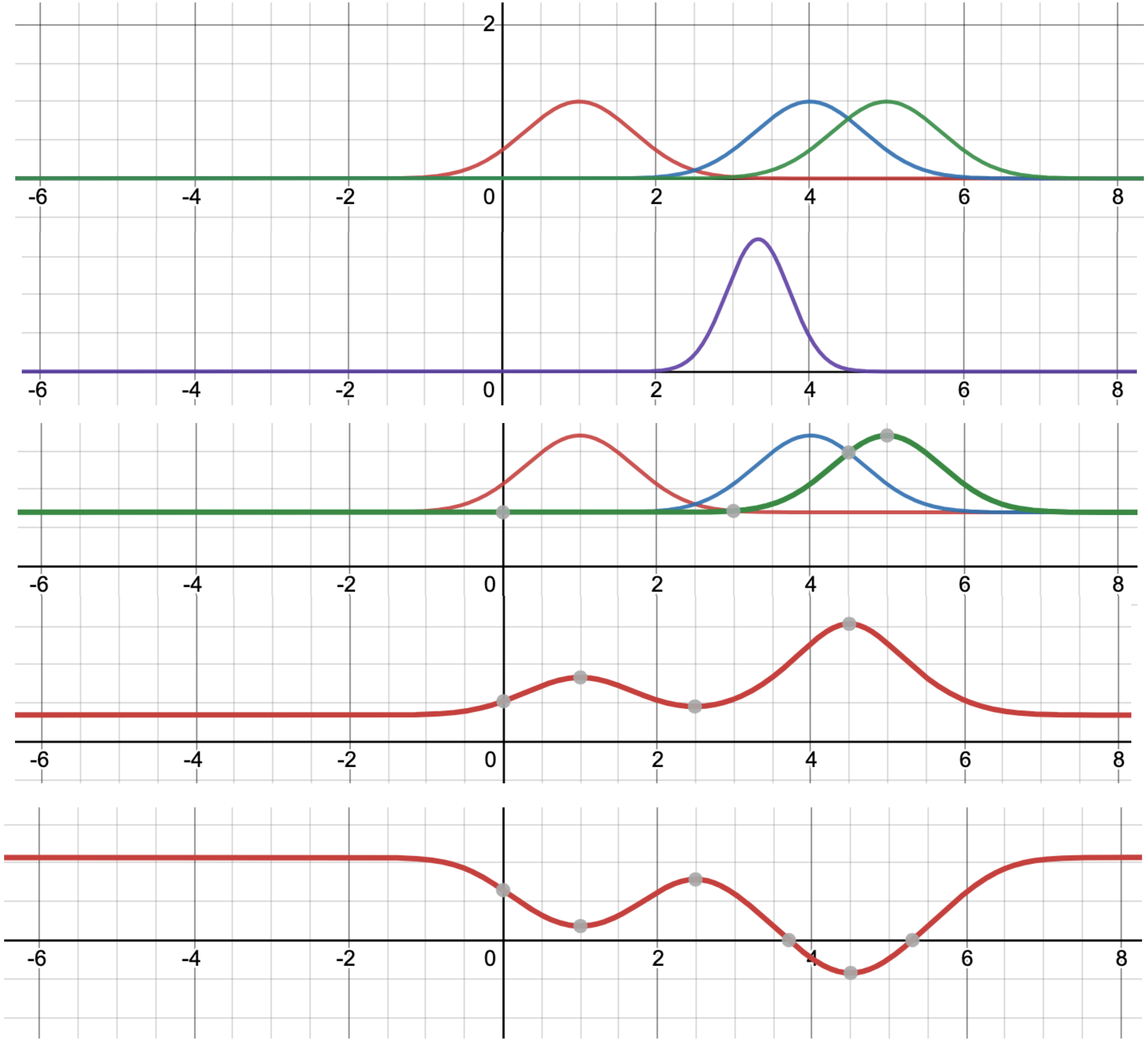
N … total number of correspondences (N=5)

w … fraction of inliers (w = 3/5 = 0.6)

s … size of |S| (s=2)

$$K = \frac{\log(1-p)}{\log(1-w^s)}$$

Use all possible correspondences simultaneously? $p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$



$$K \cdot \exp\left(- \|\mathbf{x}_t - \mathbf{m}^j\|_2^2 \right) \quad \forall_j$$

$$\prod_j K \cdot \exp\left(- \|\mathbf{x}_t - \mathbf{m}^j\|_2^2 \right)$$

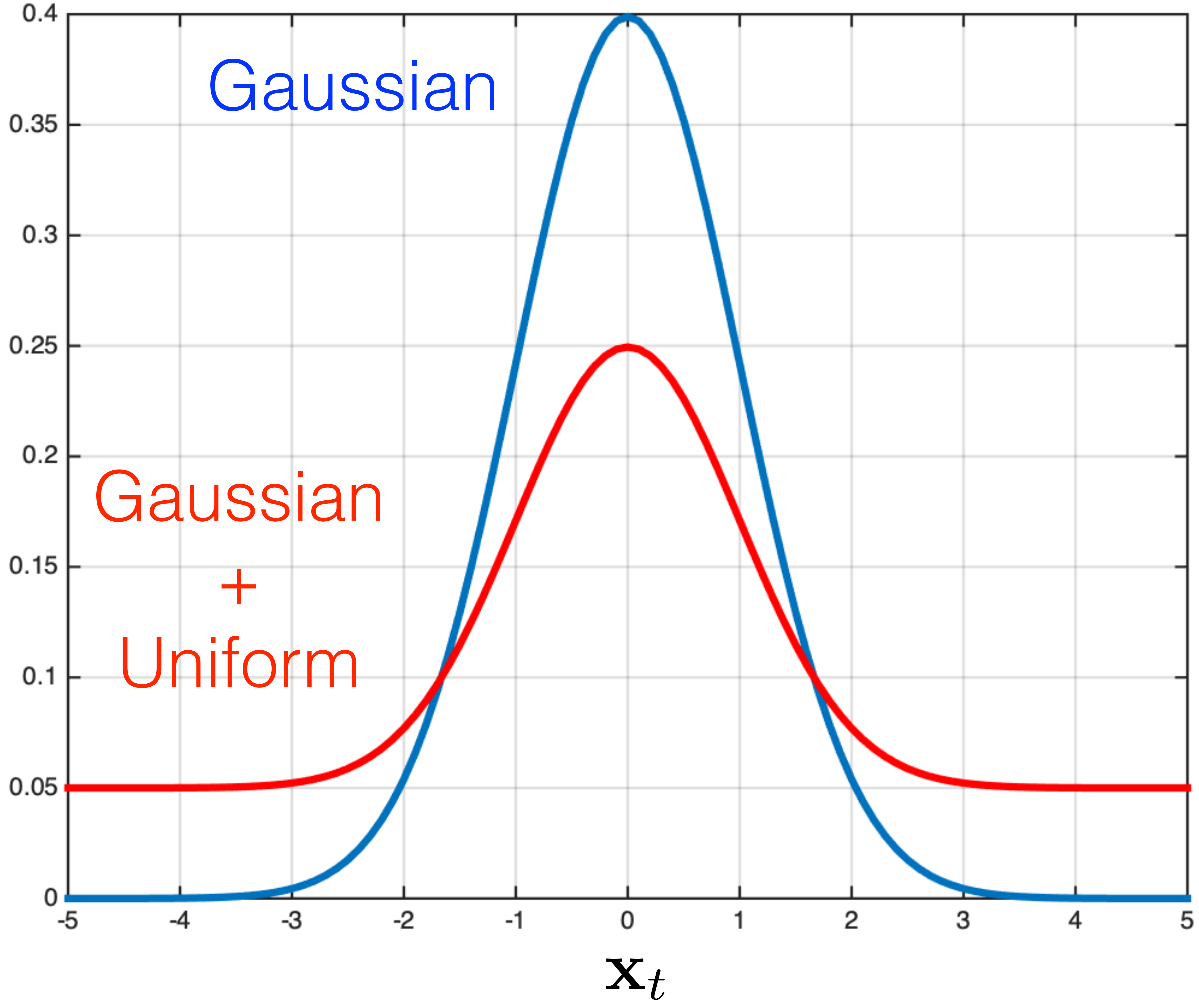
$$U + K' \cdot \exp\left(- \|\mathbf{x}_t - \mathbf{m}^j\|_2^2 \right) \quad \forall_j$$

$$\prod_j \left(U + K' \cdot \exp\left(- \|\mathbf{x}_t - \mathbf{m}^j\|_2^2 \right) \right)$$

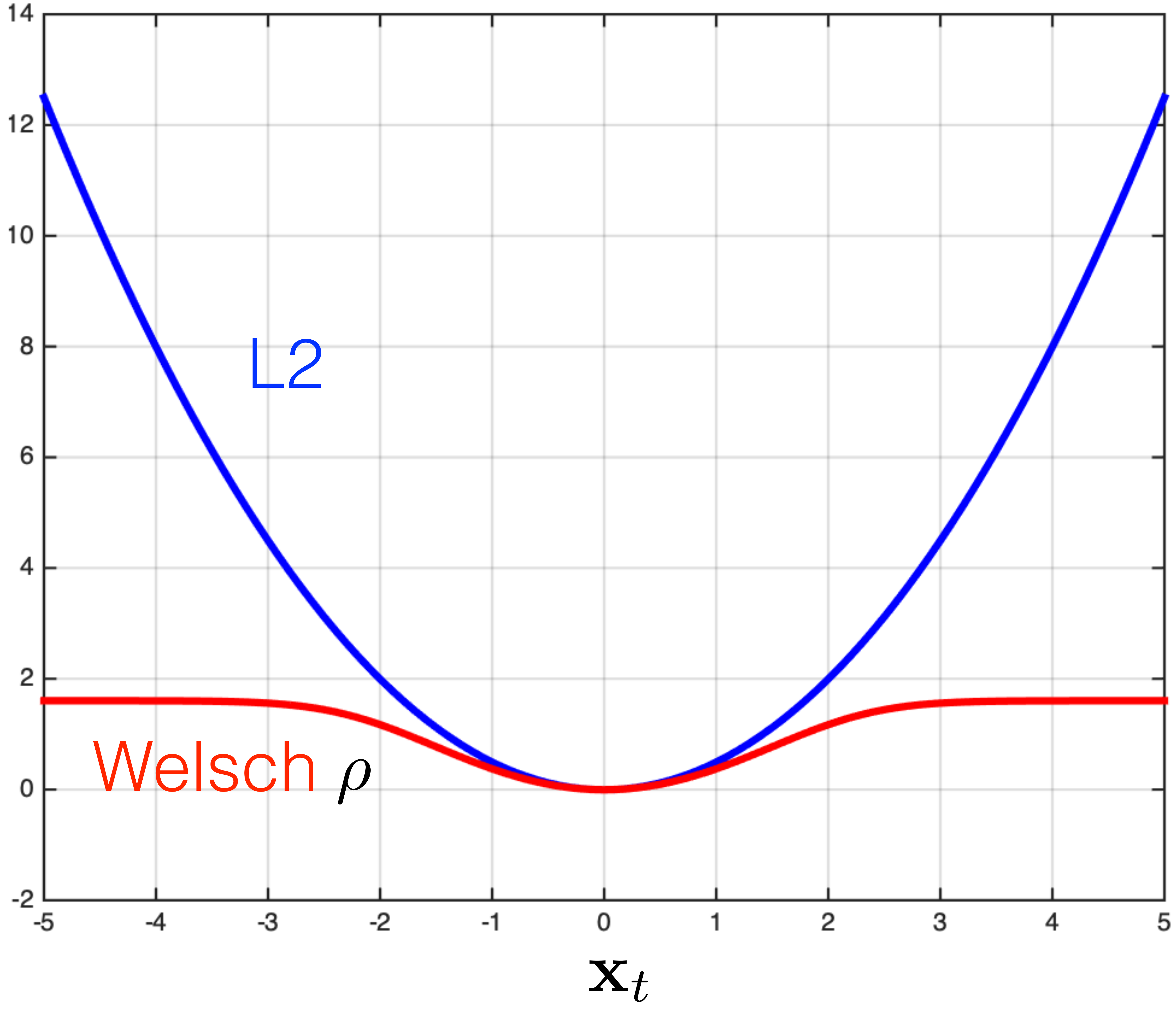
$$\sum_j -\log\left(U + K' \cdot \exp\left(- \|\mathbf{x}_t - \mathbf{m}^j\|_2^2 \right) \right)$$

RANSAC vs robust regression

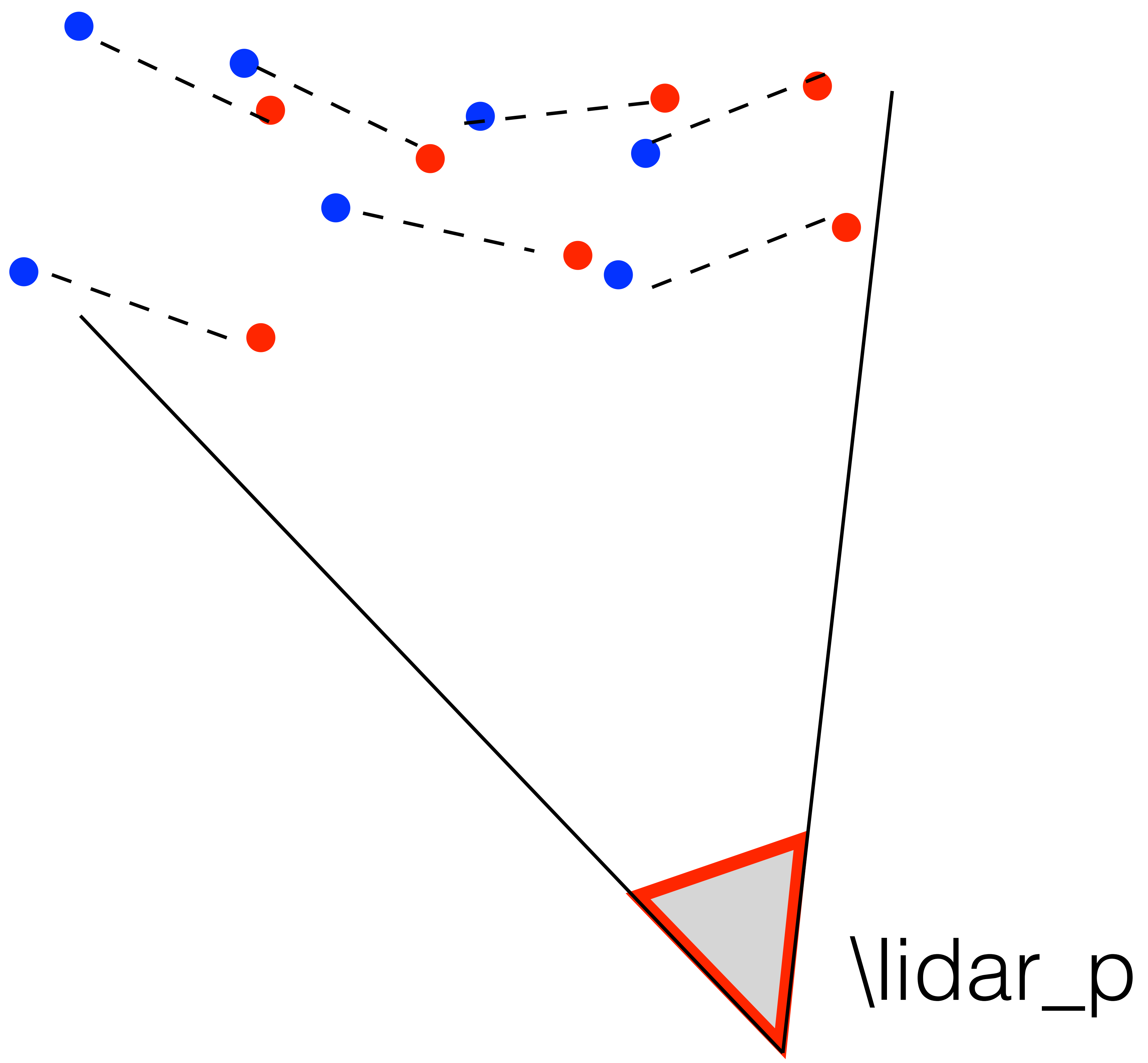
$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}^j)$



Corresponding losses



Alignment of two pointclouds



$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

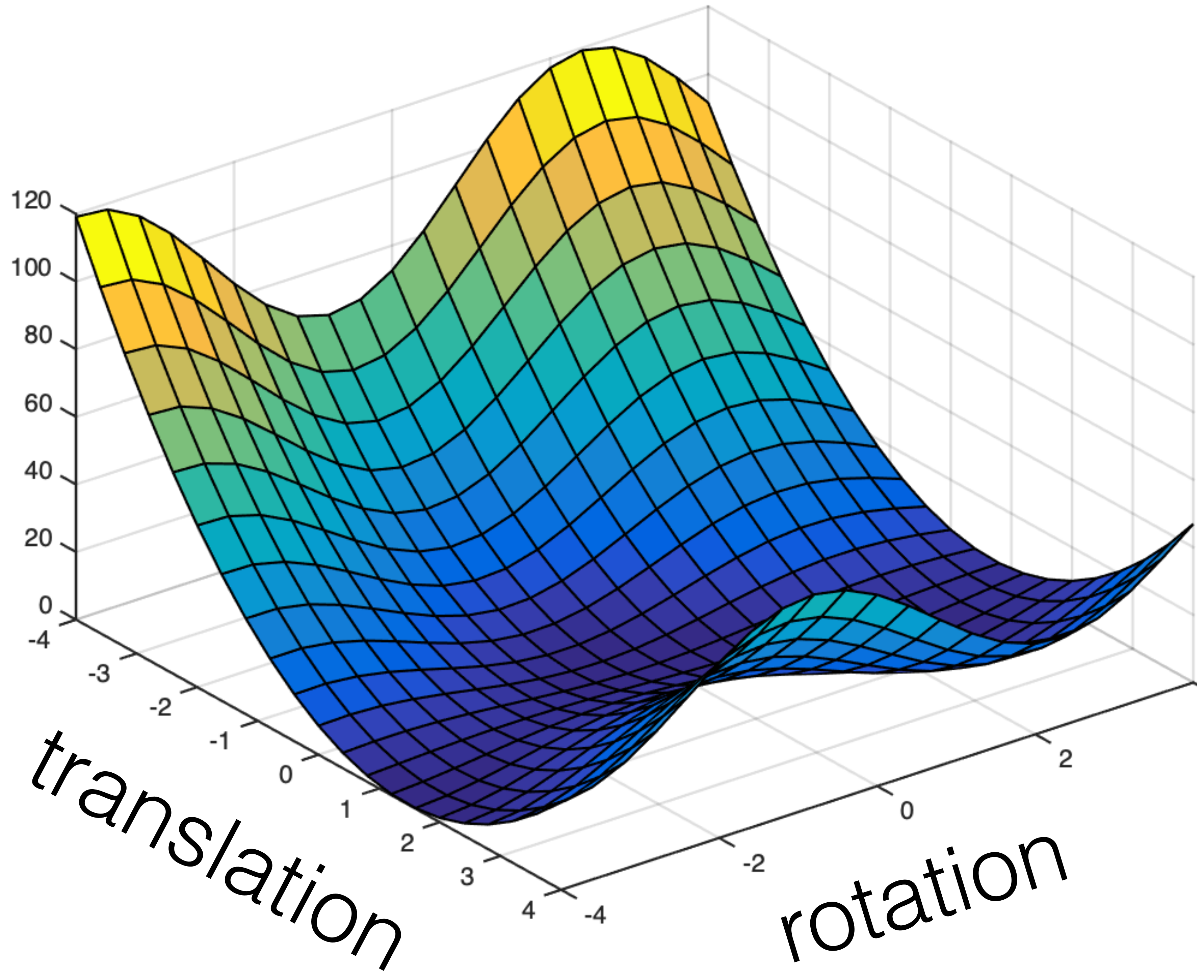
~~**L2 regression**~~

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \boldsymbol{\rho}(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)$$

Robust regression

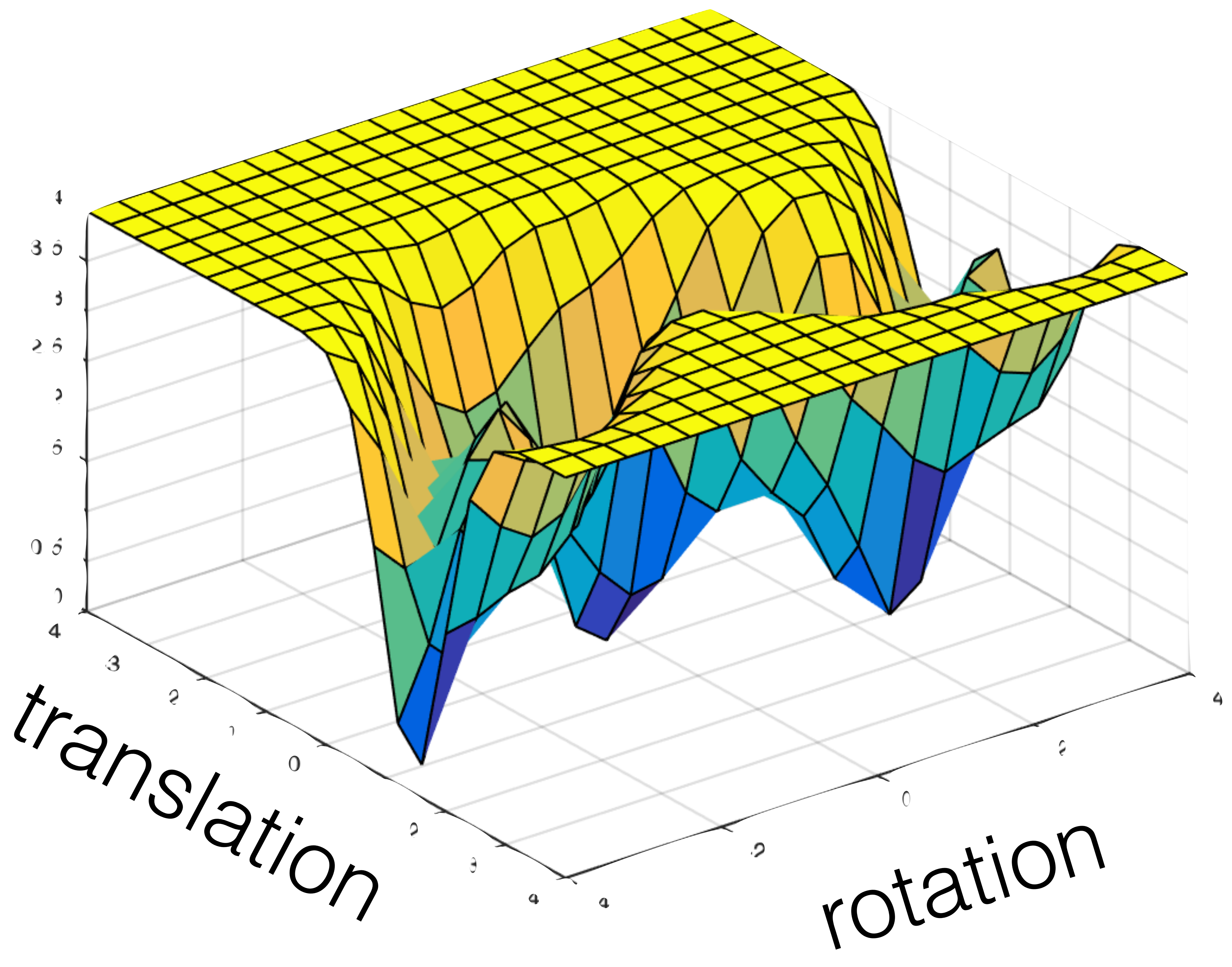
Gradient optimisation of robust loss

L2 landscape



- Convex in translation space
- Non-convex but smooth in SO3

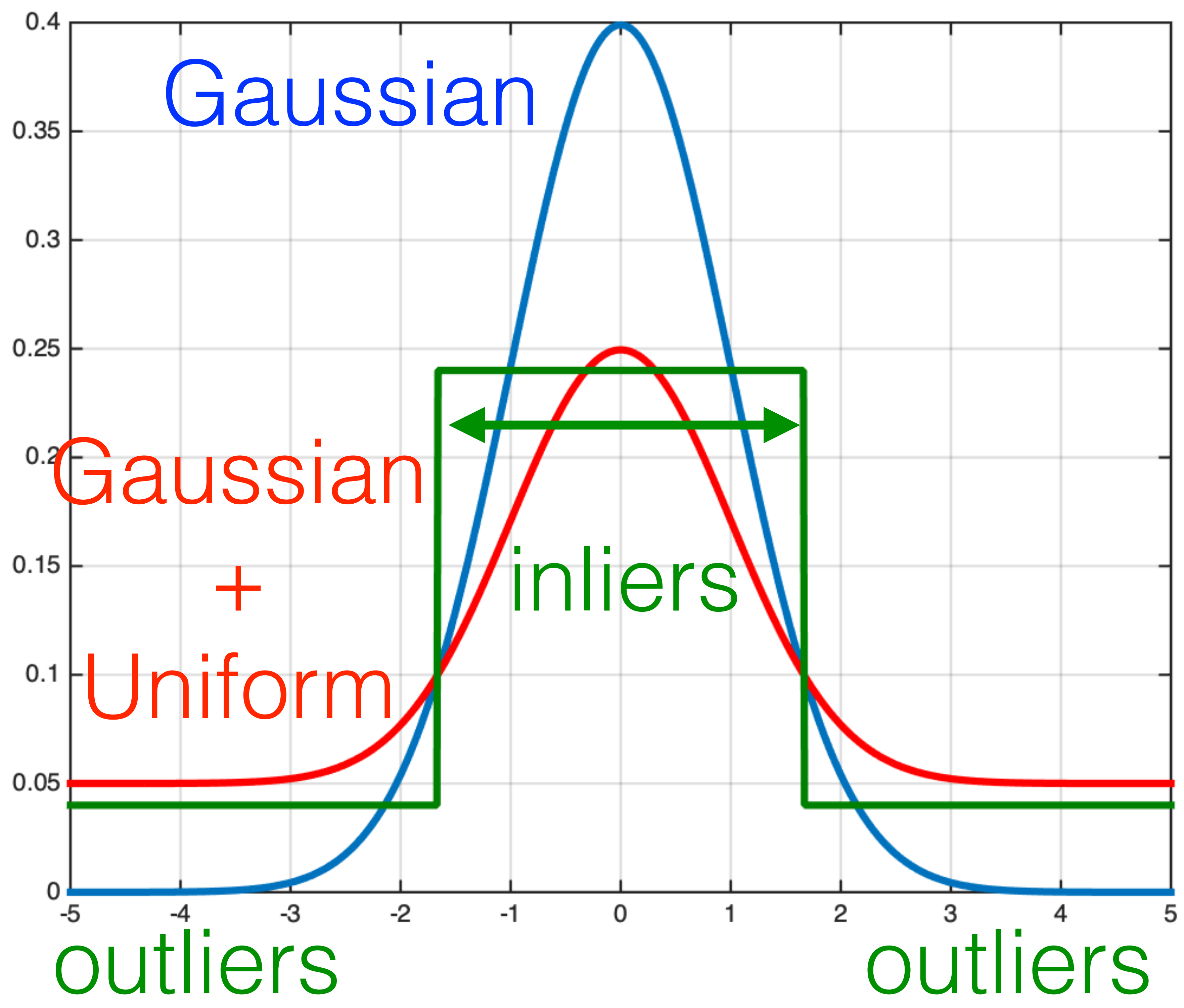
Welsch landscape



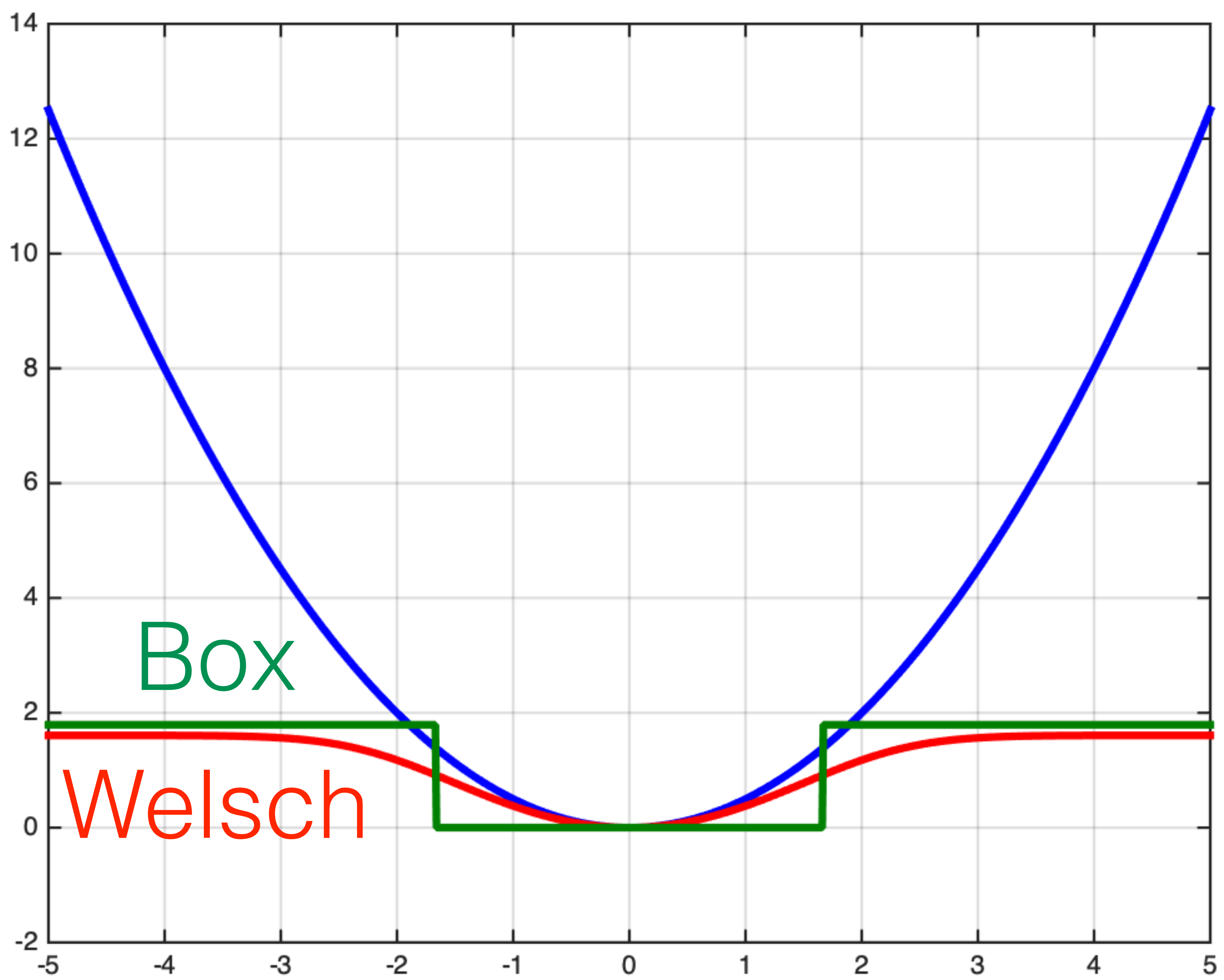
- Non-convex: Large narrow plateaus with zero gradient
- Good initialization required

ICP SLAM - outlier detection procedure

- Assume that inliers are uniformly distributed within small range (rest are outliers)
- Search for R,t which minimize number of outliers



Probability distributions



Corresponding losses

ICP SLAM - gradient optimisation of robust loss

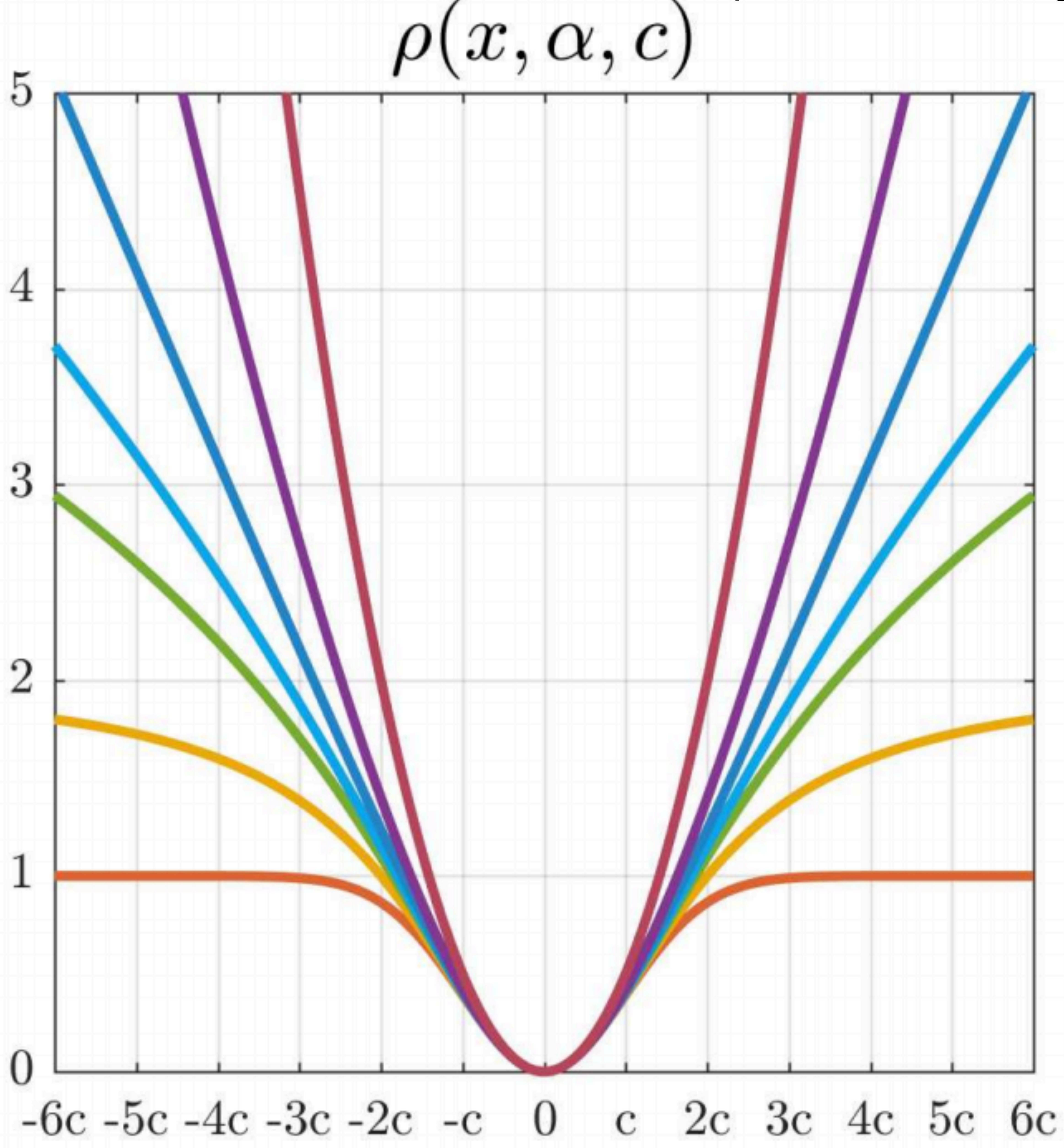
- Convex in translation space
- Non-convex but smooth in SO3

- Non-convex+Large narrow plateaus with zero gradient
- Any gradient optimization requires good initialization

- Zero gradients
- Combinatorial optimization

Shape of robust regression functions [Barron CVPR 2019]

https://arxiv.org/abs/1701.03077

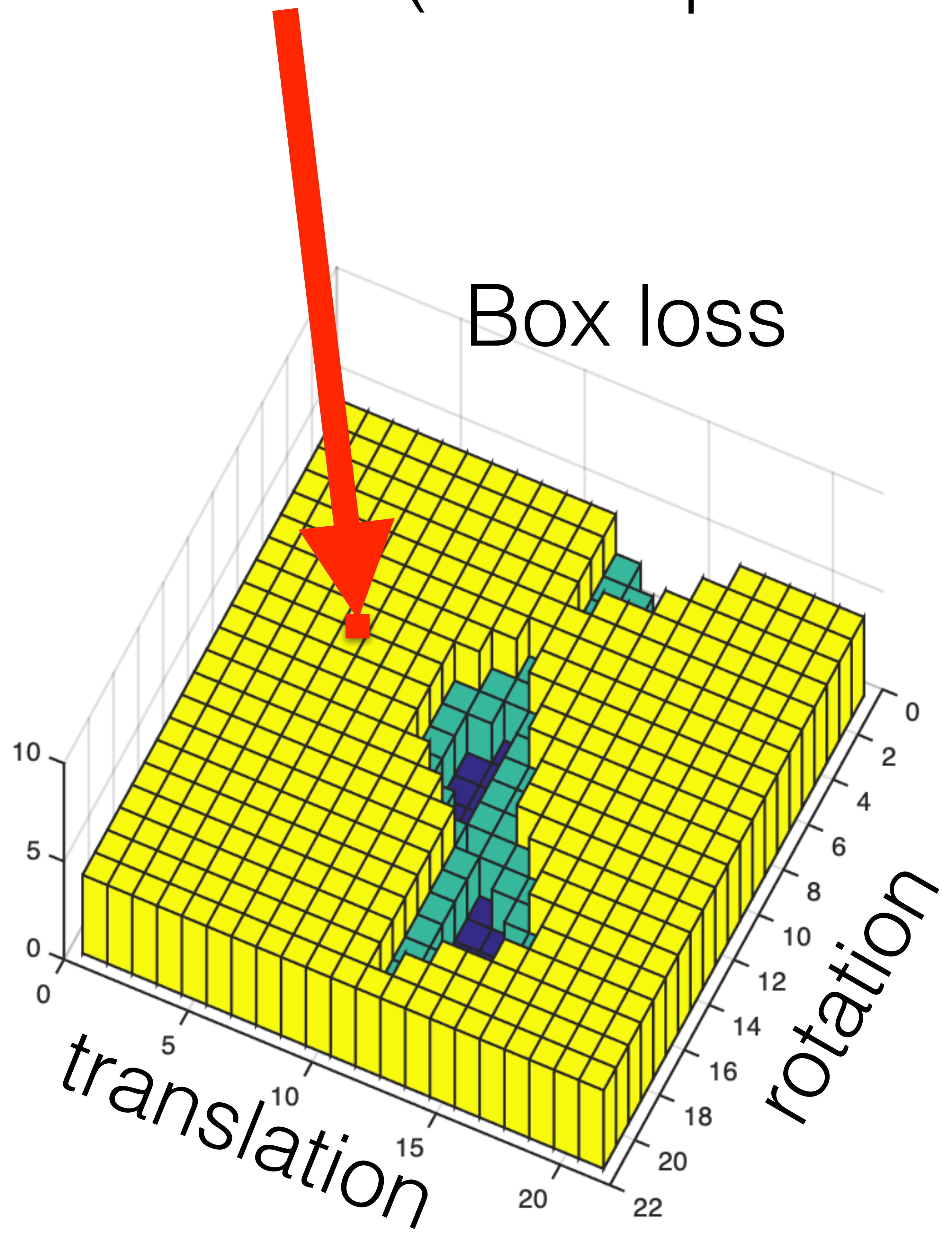


$$\rho(x, \alpha, c) = \frac{|\alpha - 2|}{\alpha} \left(\left(\frac{\left(x/c\right)^2}{|\alpha - 2|} + 1 \right)^{\alpha/2} - 1 \right)$$

Optimizing box-loss

Naive optimization algorithm:

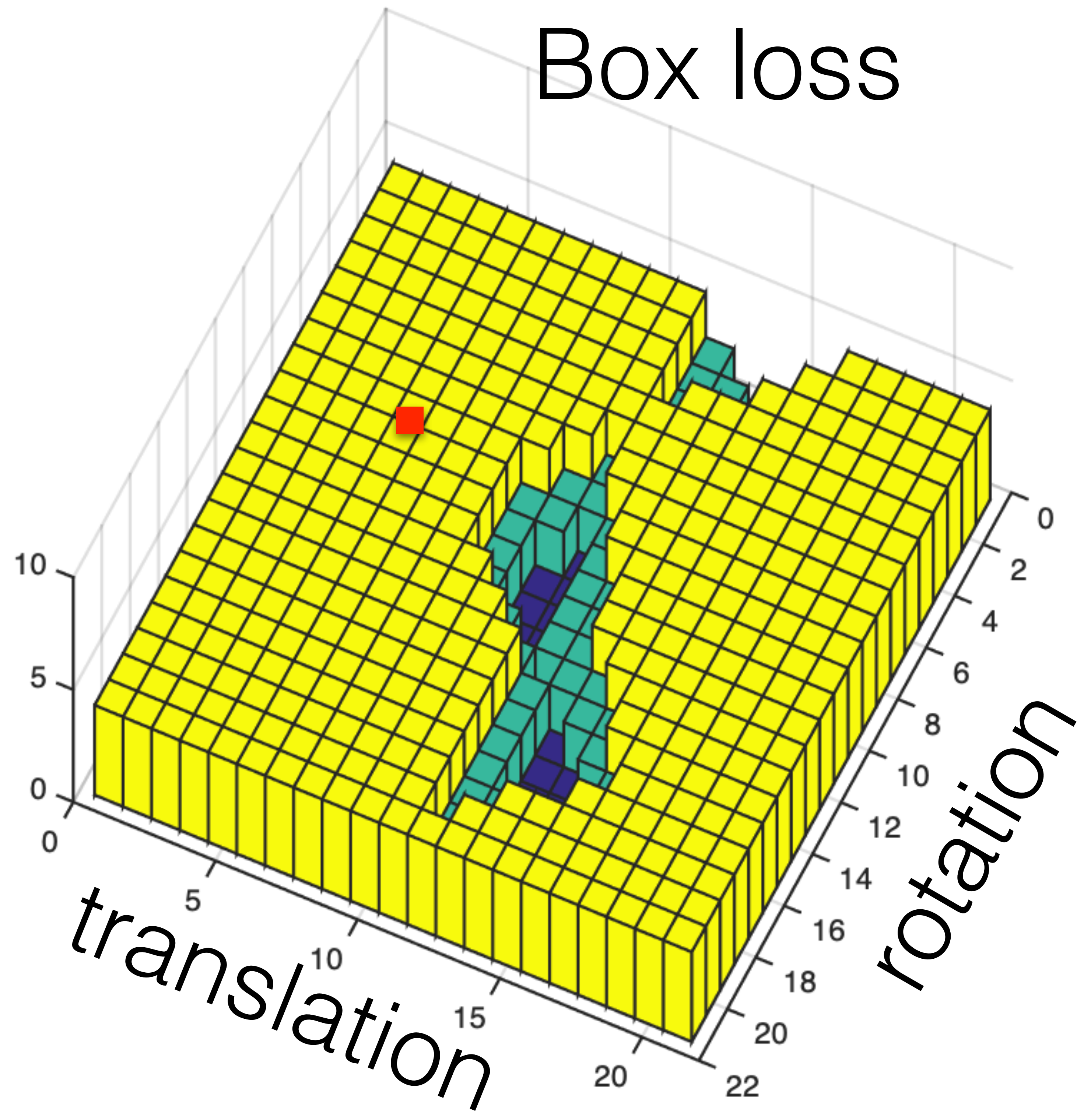
1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)



Optimizing box-loss

Naive optimization algorithm:

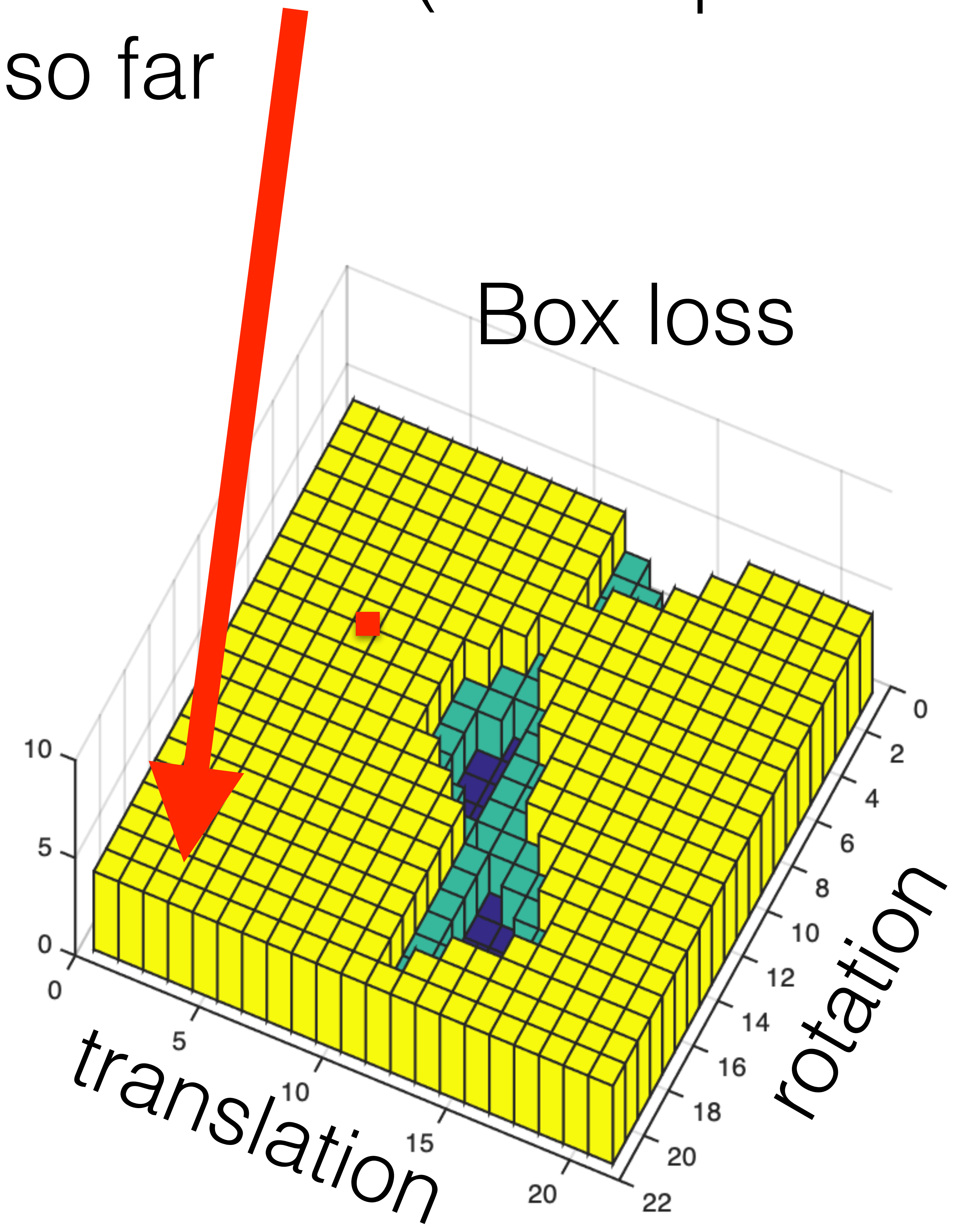
1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times



Optimizing box-loss

Naive optimization algorithm:

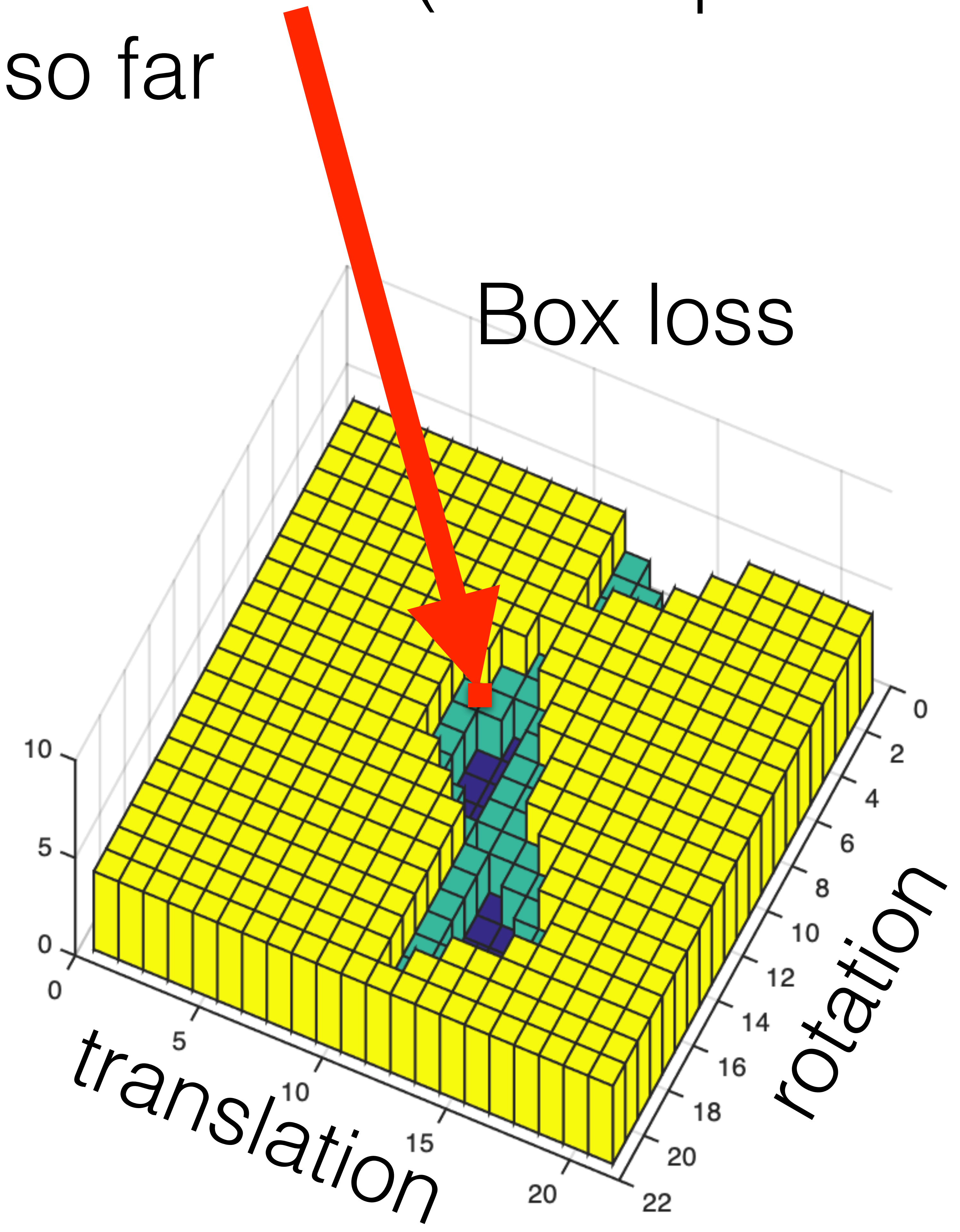
1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times



Optimizing box-loss

Naive optimization algorithm:

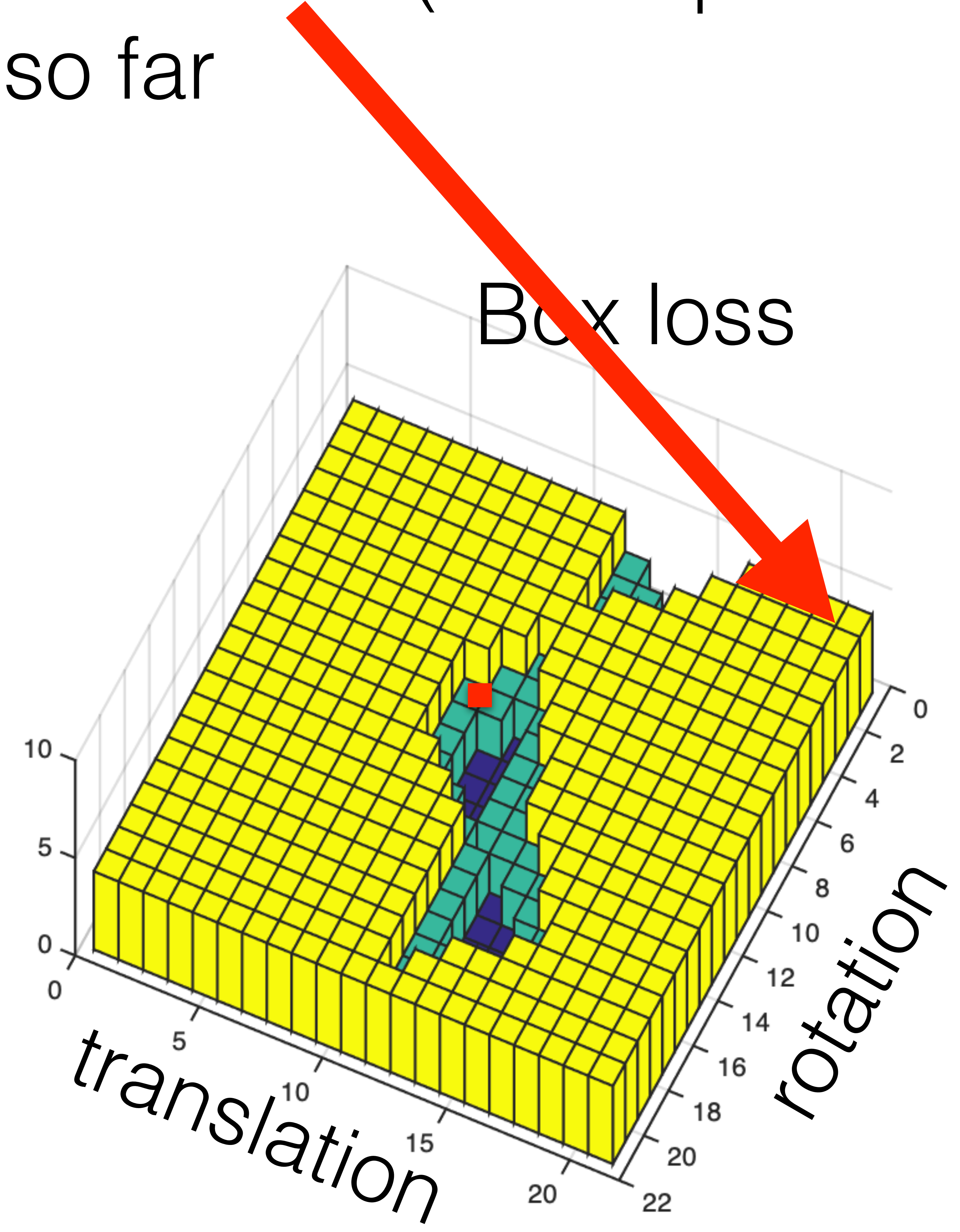
1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times



Optimizing box-loss

Naive optimization algorithm:

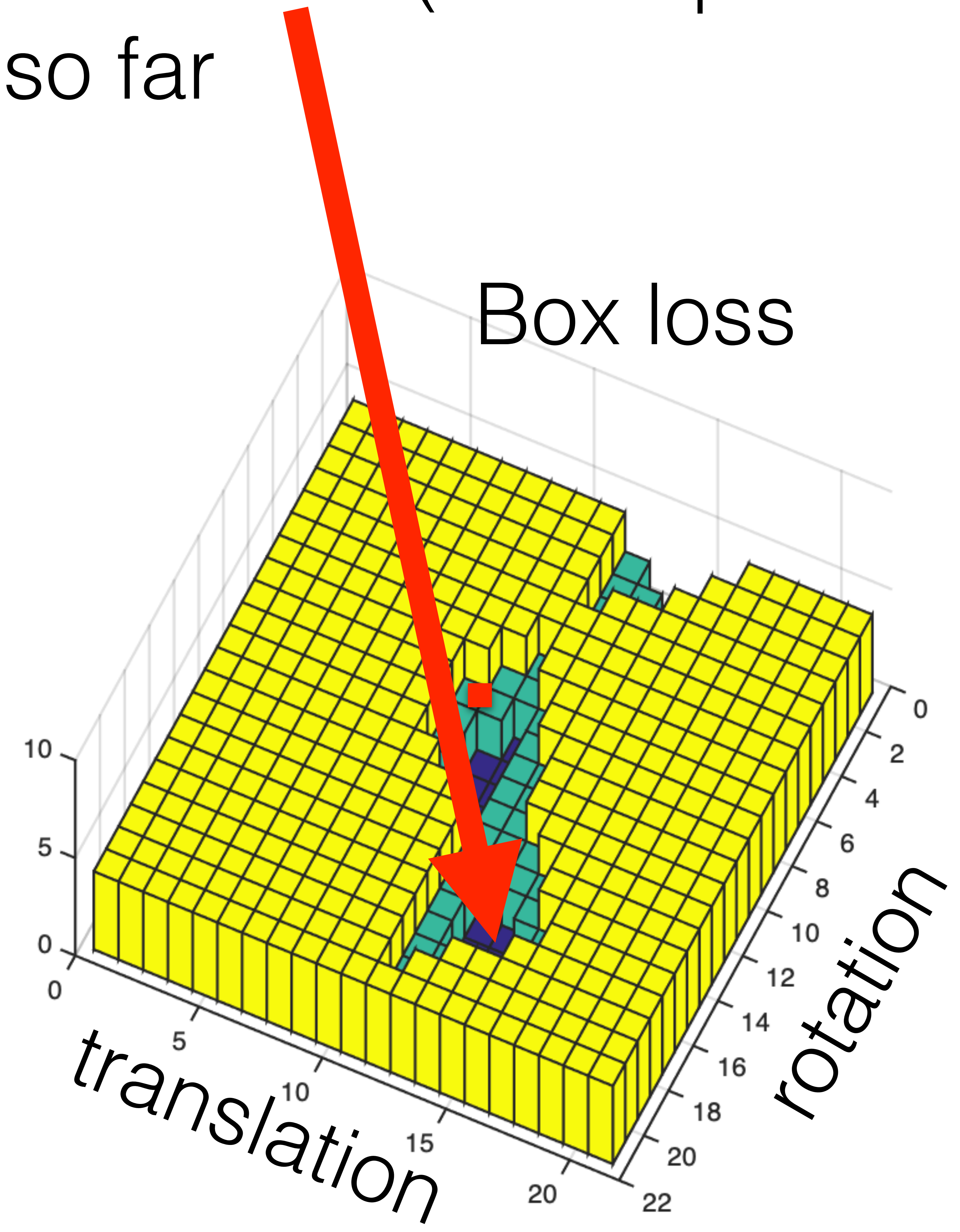
1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times



Optimizing box-loss

Naive optimization algorithm:

1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times

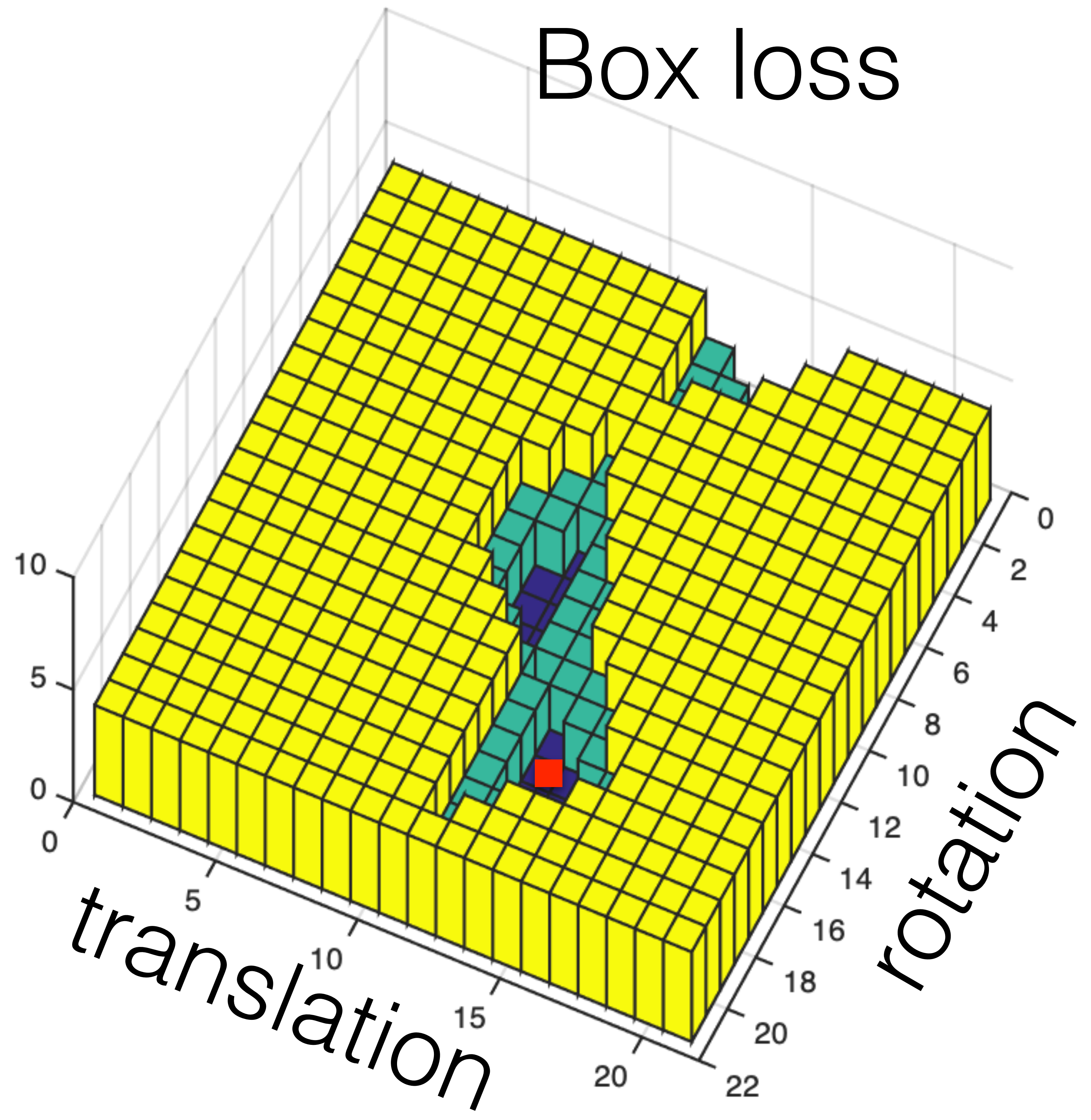


Optimizing box-loss

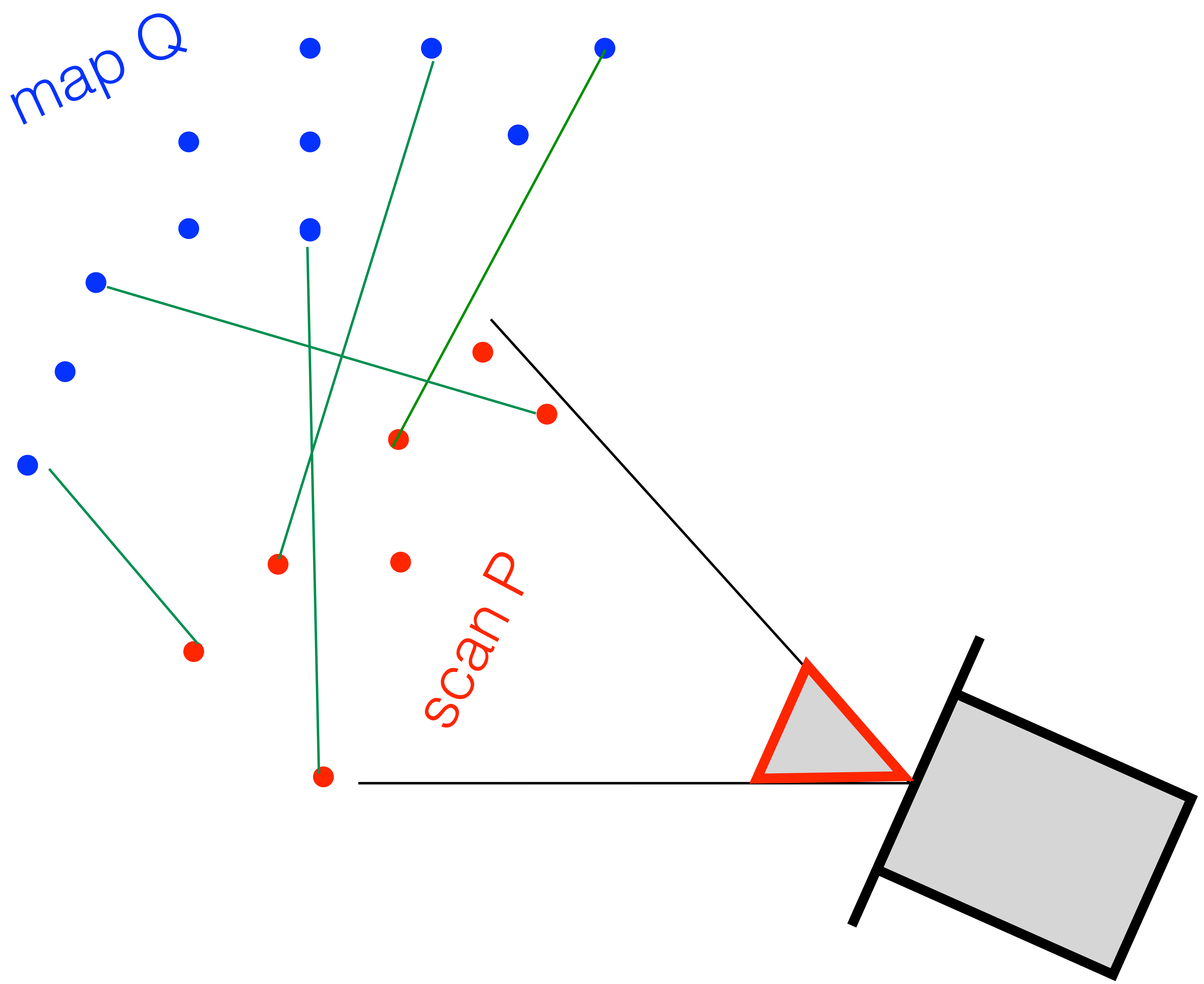
Naive optimization algorithm:

1. Sample hypothesis (R,t) at random
2. Evaluate value of the box-loss function (at this point R,t)
3. Remember the lowest value so far
4. repeat K times

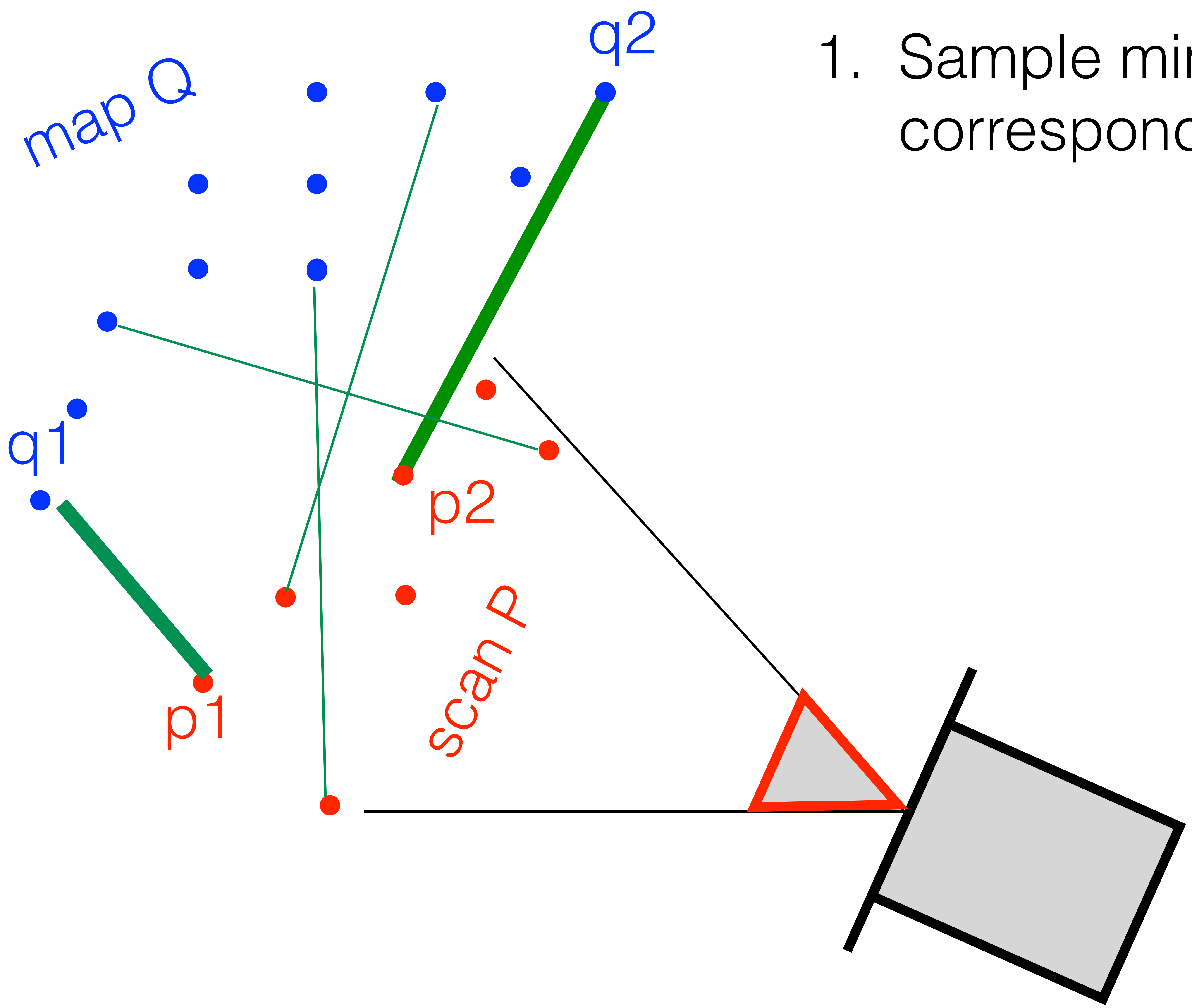
if K is huge and you are lucky



RANSAC (RAndom SAmple Consensus)

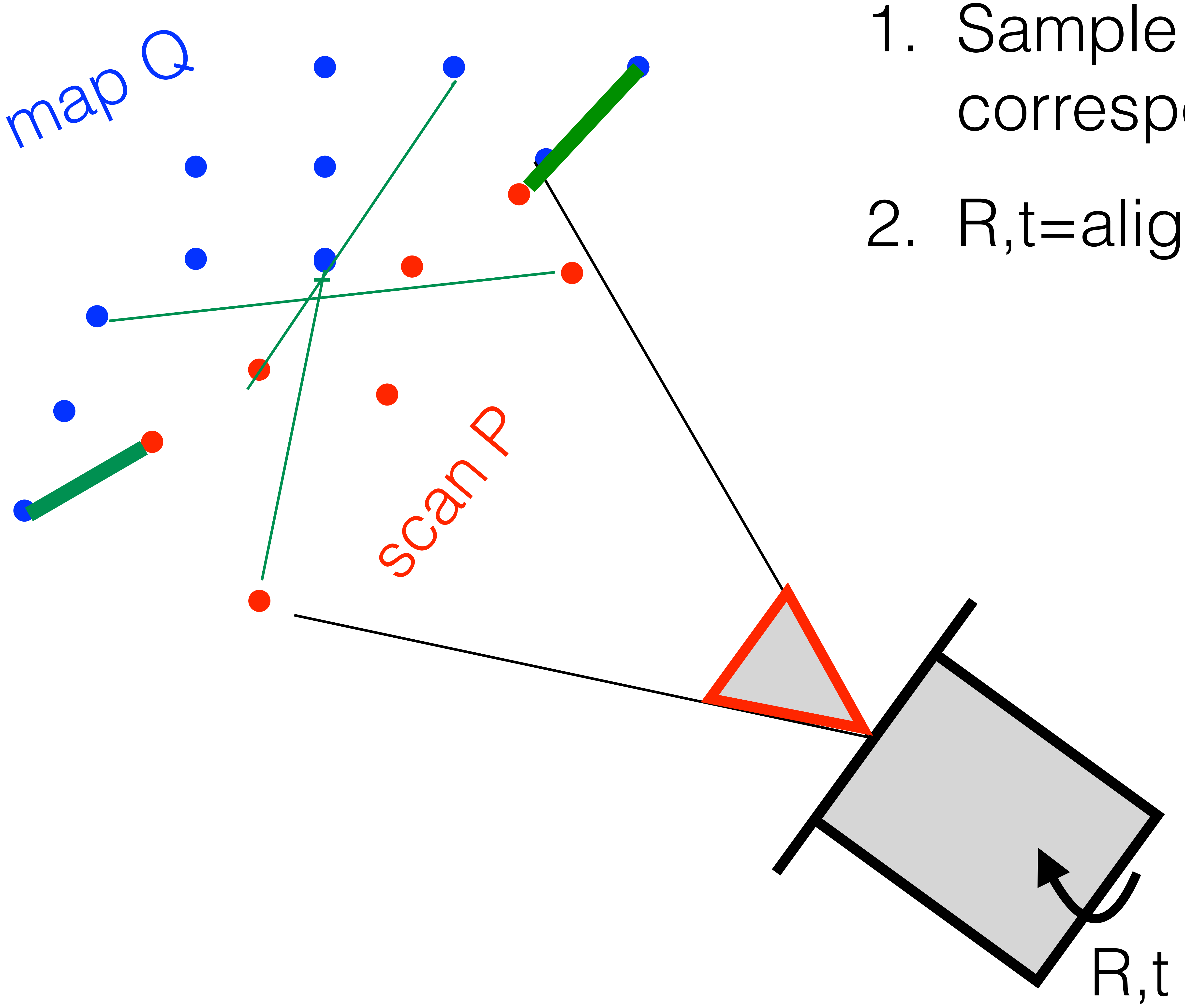


RANSAC (RAndom SAmple Consensus)



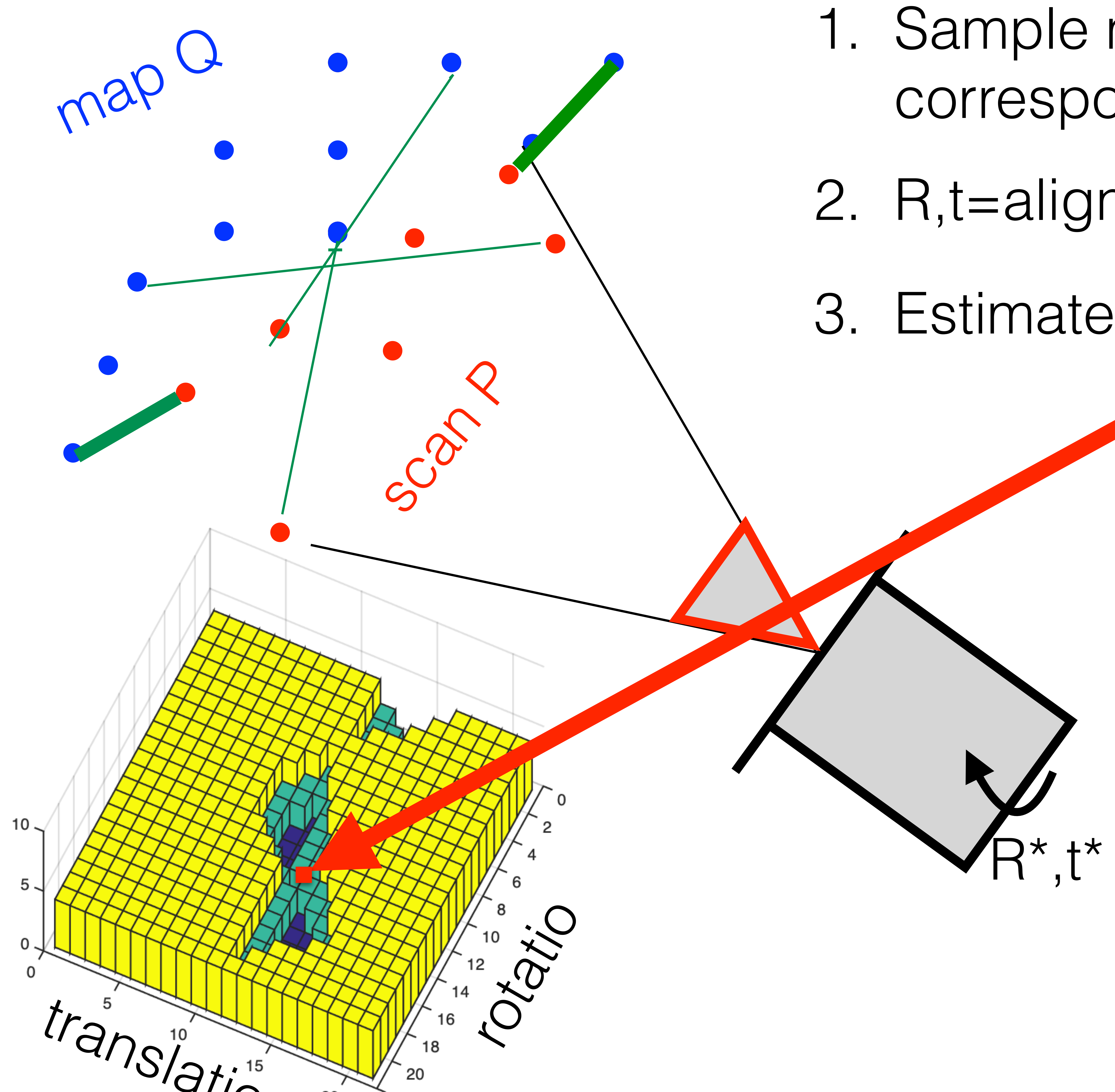
1. Sample minimal subset of correspondences (p, q).

RANSAC (RAndom SAmple Consensus)



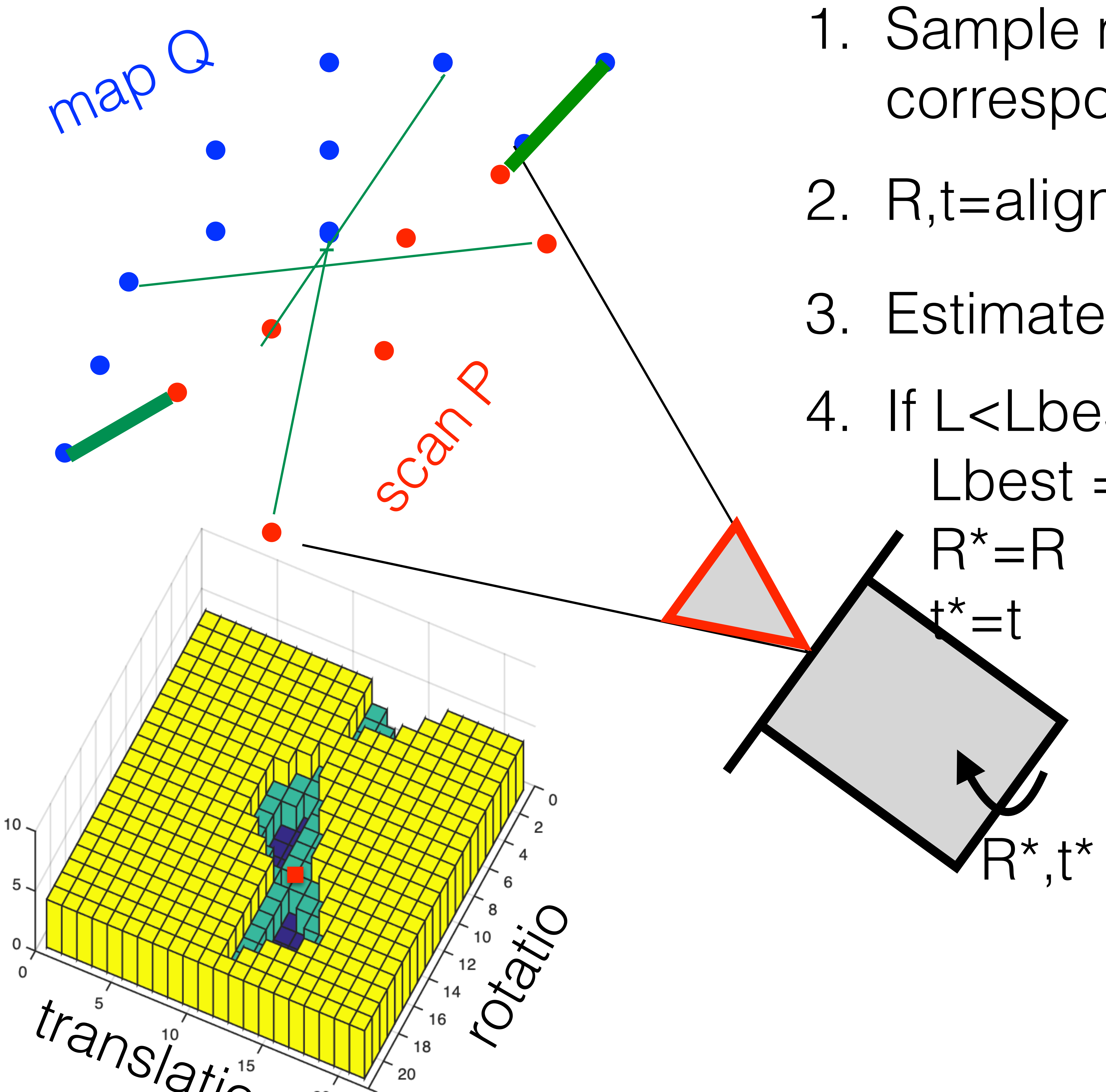
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)

RANSAC (RAndom SAmple Consensus)



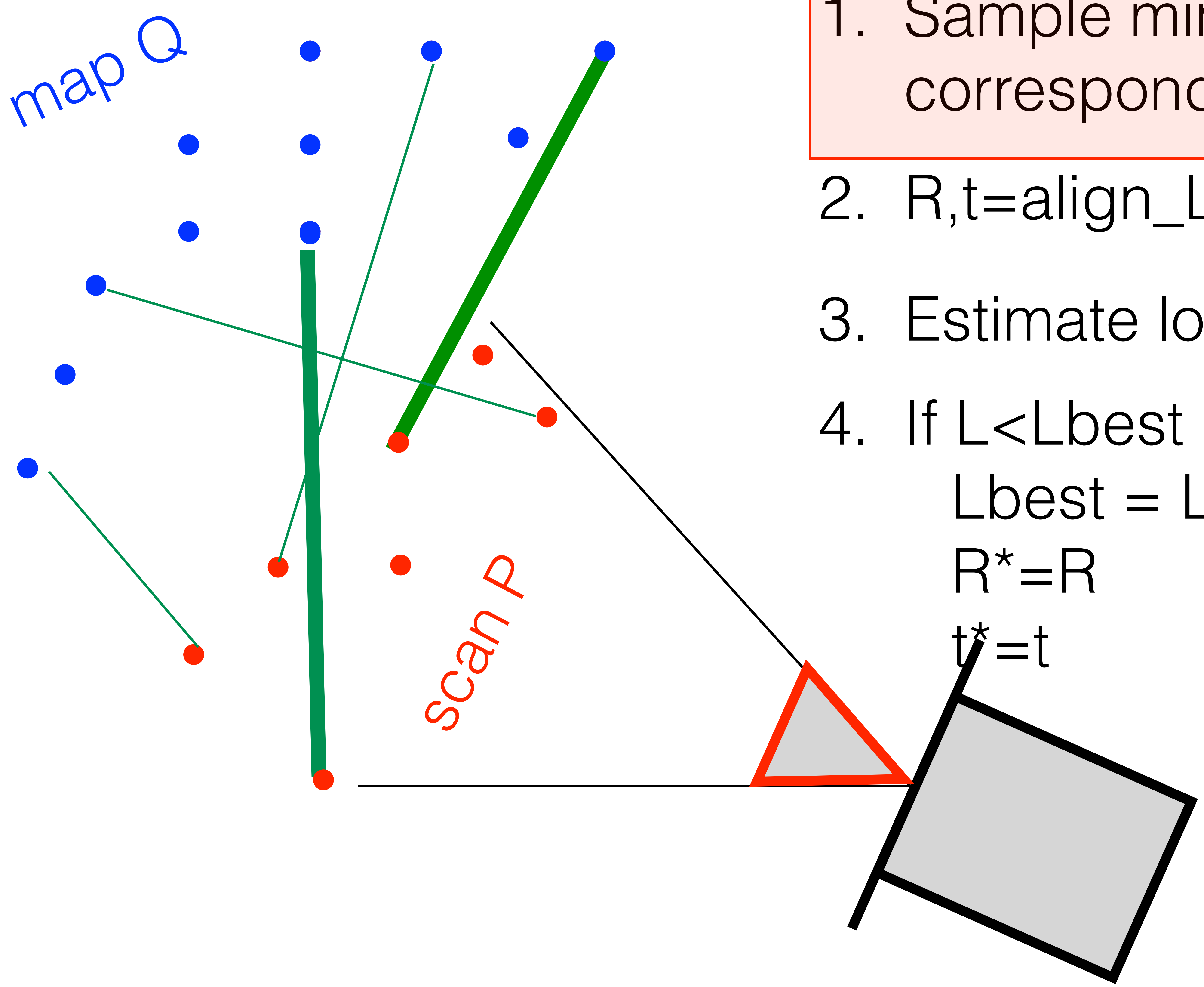
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss **L=3**

RANSAC (RAndom SAmple Consensus)



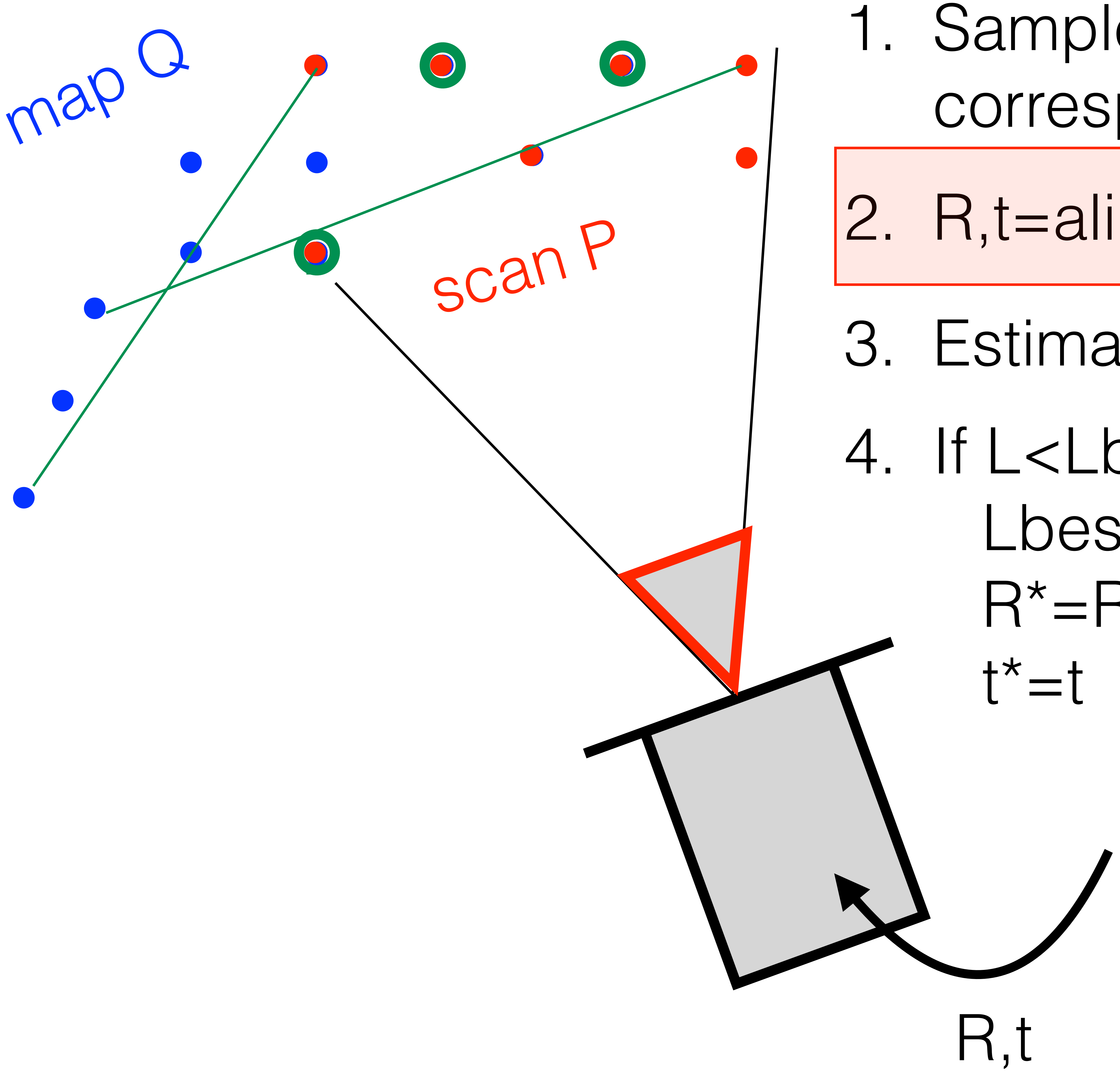
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss **L=3**
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

RANSAC



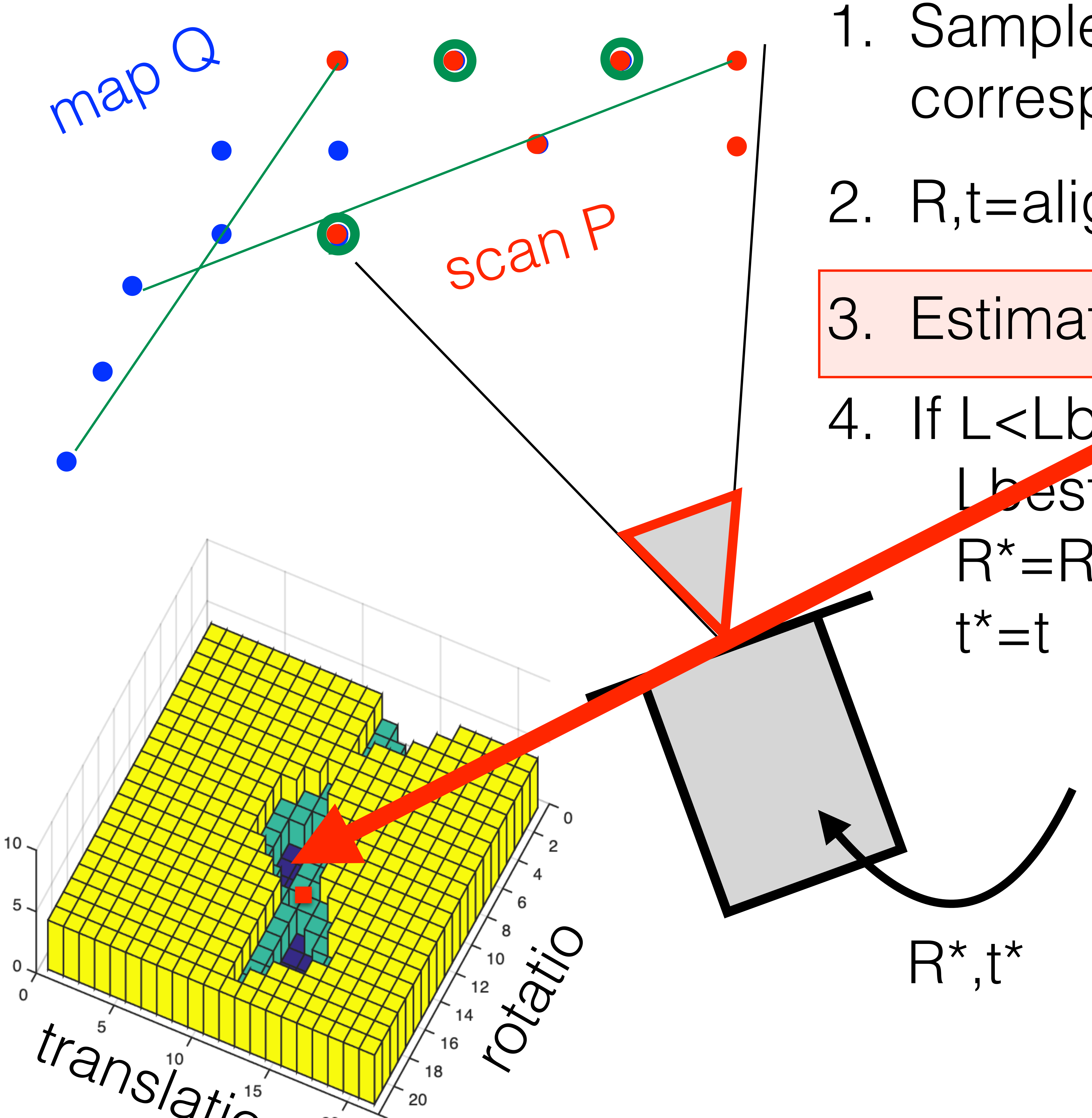
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

RANSAC (RAndom SAmple Consensus)



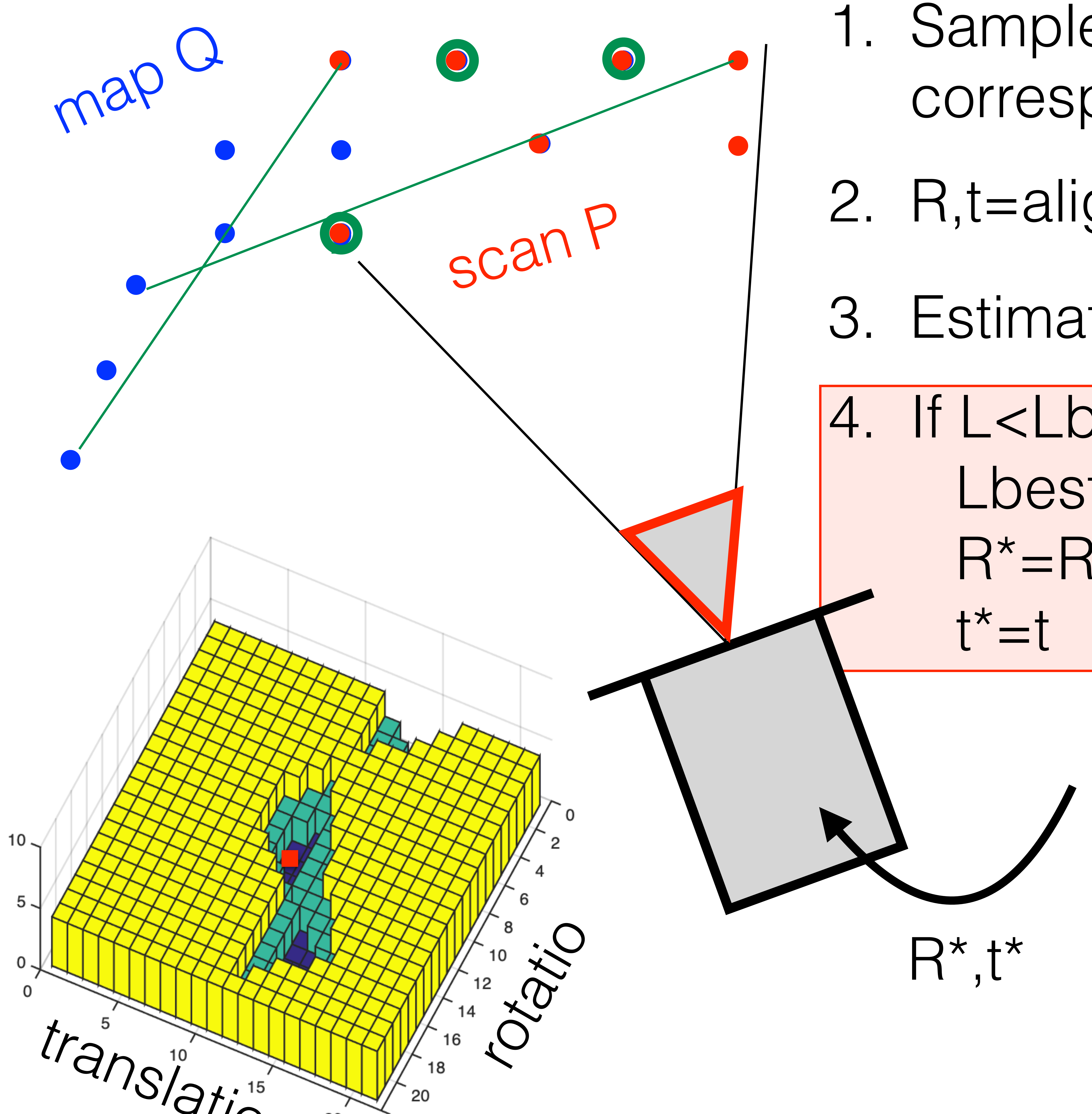
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

RANSAC (RAndom SAmple Consensus)



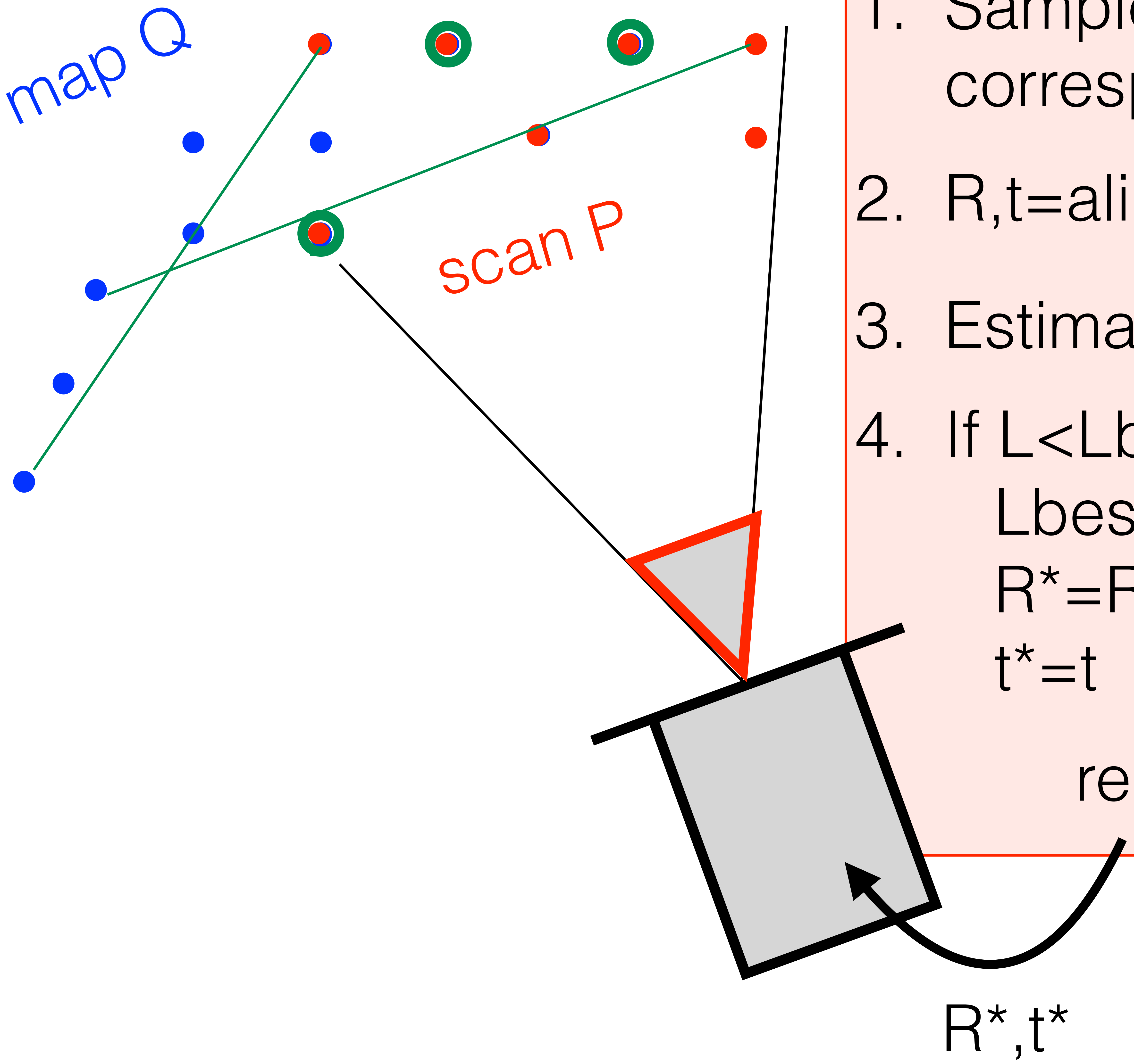
1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss **L=2**
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

RANSAC (RAndom SAmple Consensus)



1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss **L=2**
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

RANSAC (RAndom SAmple Consensus)



1. Sample minimal subset of correspondences (p, q).
2. R,t=align_L2(p, q)
3. Estimate loss **L=2**
4. If L<Lbest
   Lbest = L
   R*=R
   t*=t

repeat K-times

Summary

- Minimizing L2-loss on unclean correspondences (with outliers) yields biased pose estimate (and pointcloud alignment).
- Minimizing robust norms (Welsch) is complicated due to large plateaus with almost zero gradients.
- It can be replaced by inlier detection method (RANSAC), which randomly sample reasonable hypothesis (R,t).
- RANSAC is often used for 2D-2D correspondences and large motion (e.g. reconstruction of 3D world from collection of unordered RGB images).
- When motion between successive frames is sufficiently small and RGBD data are available (self-driving cars), gradient minimization of a robust loss by a Levenberg-Marquardt is quite OK.
- **Takehome message:** When designing the loss function always think about:
  A. Underlying probability distribution
  B. Optimization of the resulting landscape

Summary 2

- SLAM implementations:
  - Nvidia Issac SLAM: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam
  - ORB SLAM: https://github.com/alsora/ros2-ORB_SLAM2
- Datasets, benchmarks and challenges:
  - Waymo https://waymo.com/intl/en_us/dataset-download-terms/
  - Kitti http://www.cvlibs.net/datasets/kitti/