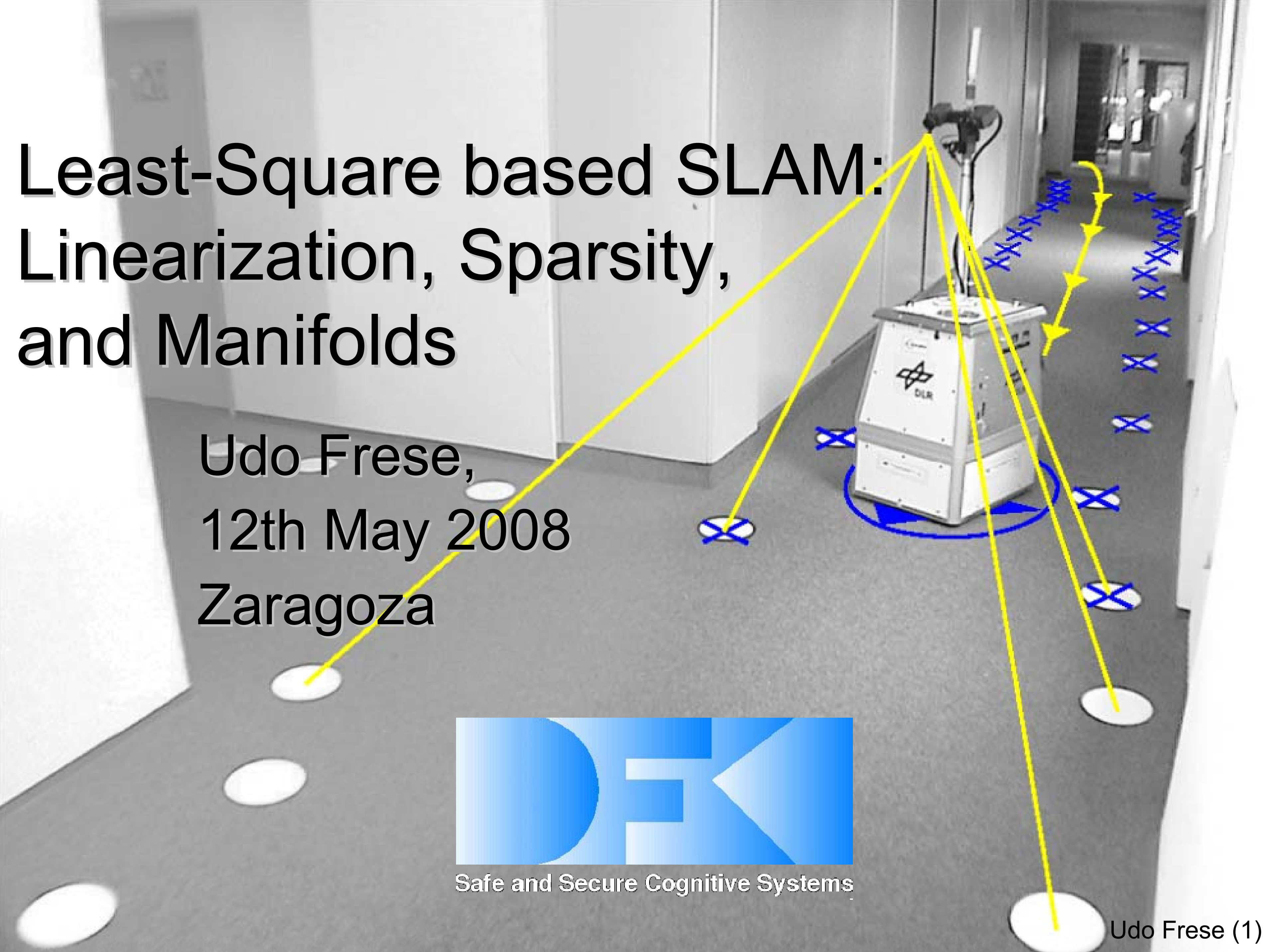


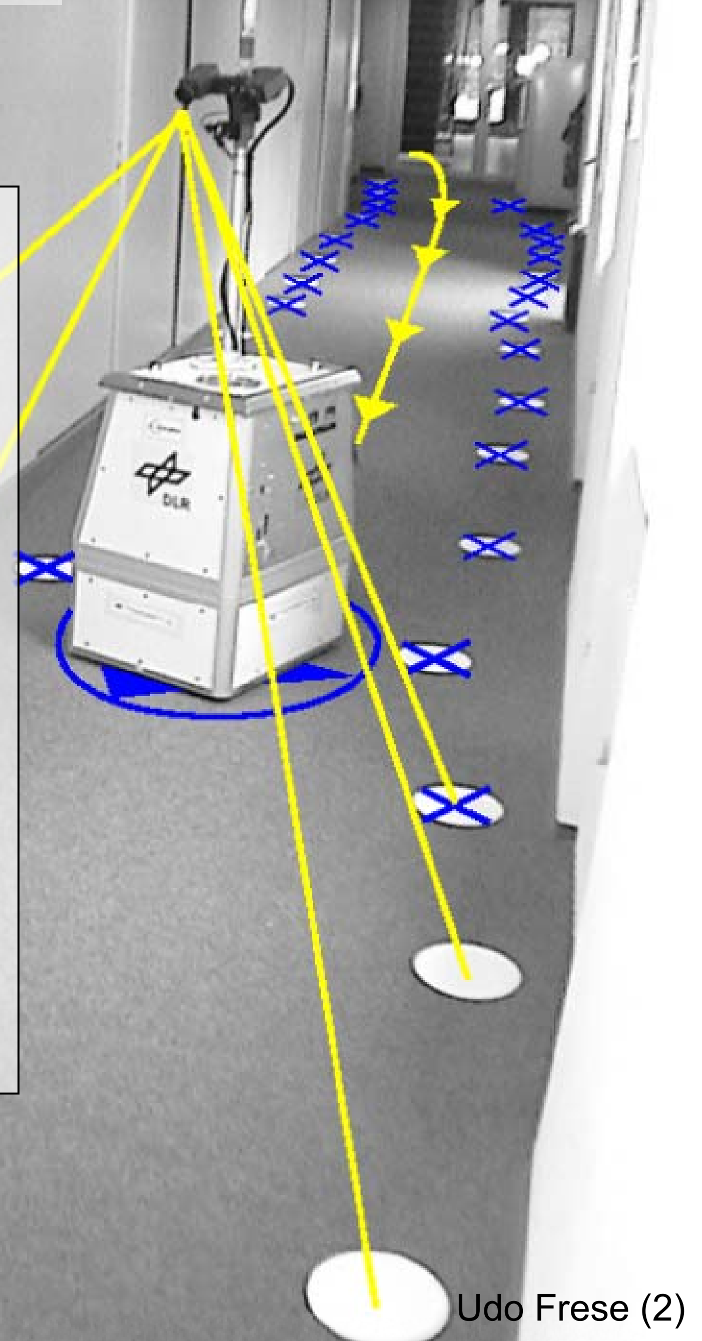
Least-Square based SLAM: Linearization, Sparsity, and Manifolds

Udo Frese,
12th May 2008
Zaragoza

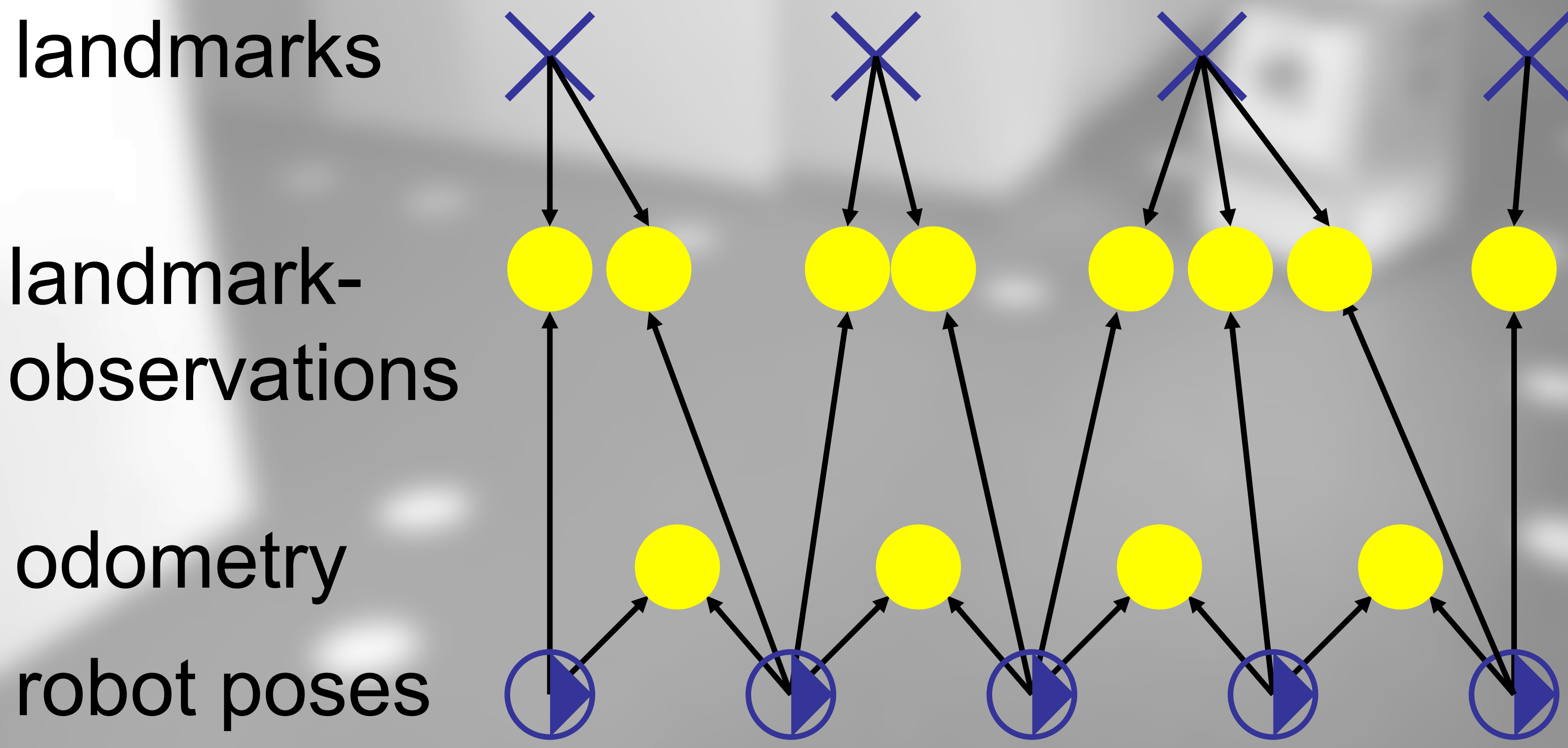
Safe and Secure Cognitive Systems

What is Least-Square based SLAM?

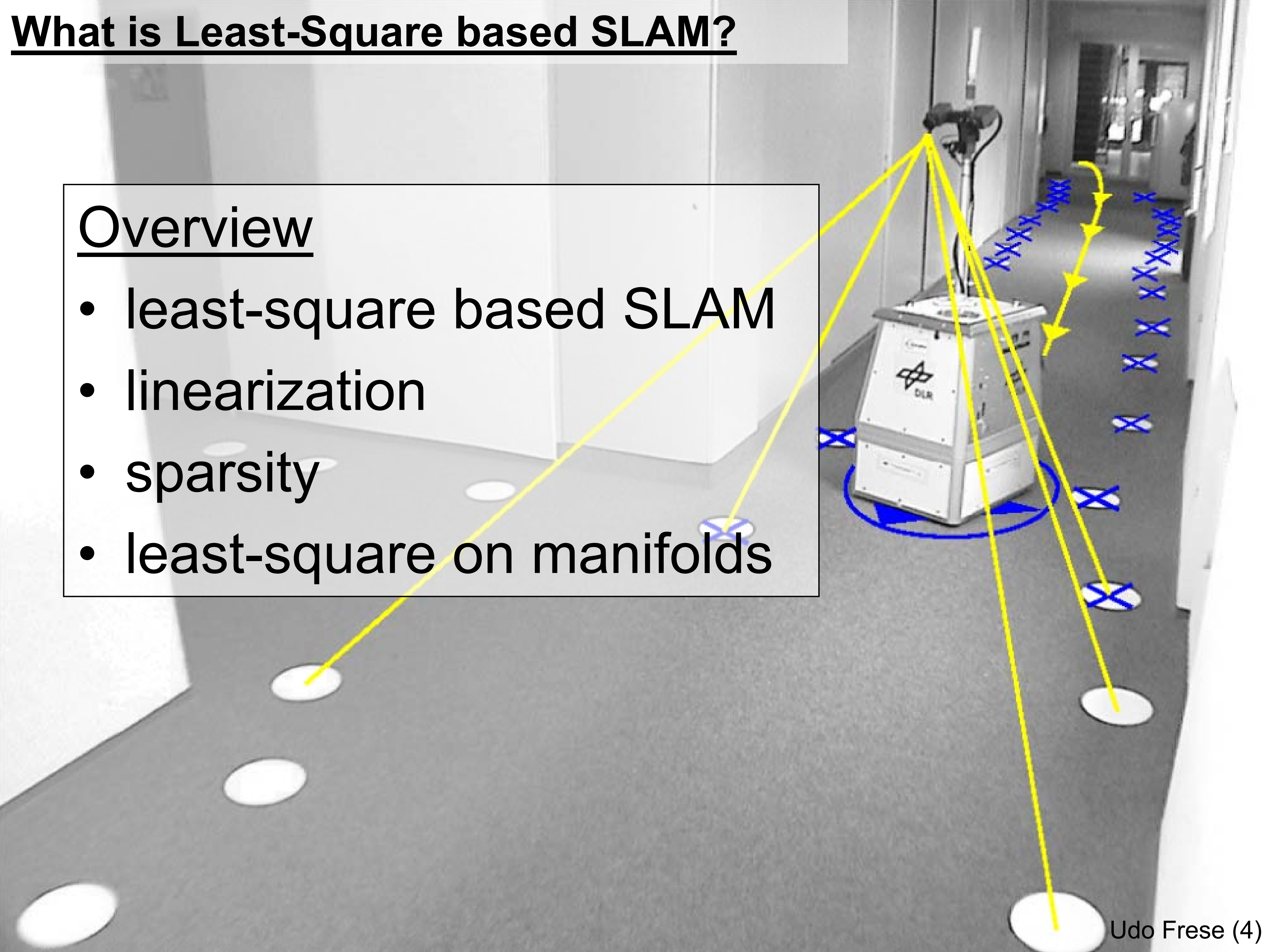
- continuously estimate a map from sensor data
- input (yellow):
  - landmark observations
  - odometry
- output (blue):
  - landmark positions
  - robot pose



Least Square based SLAM



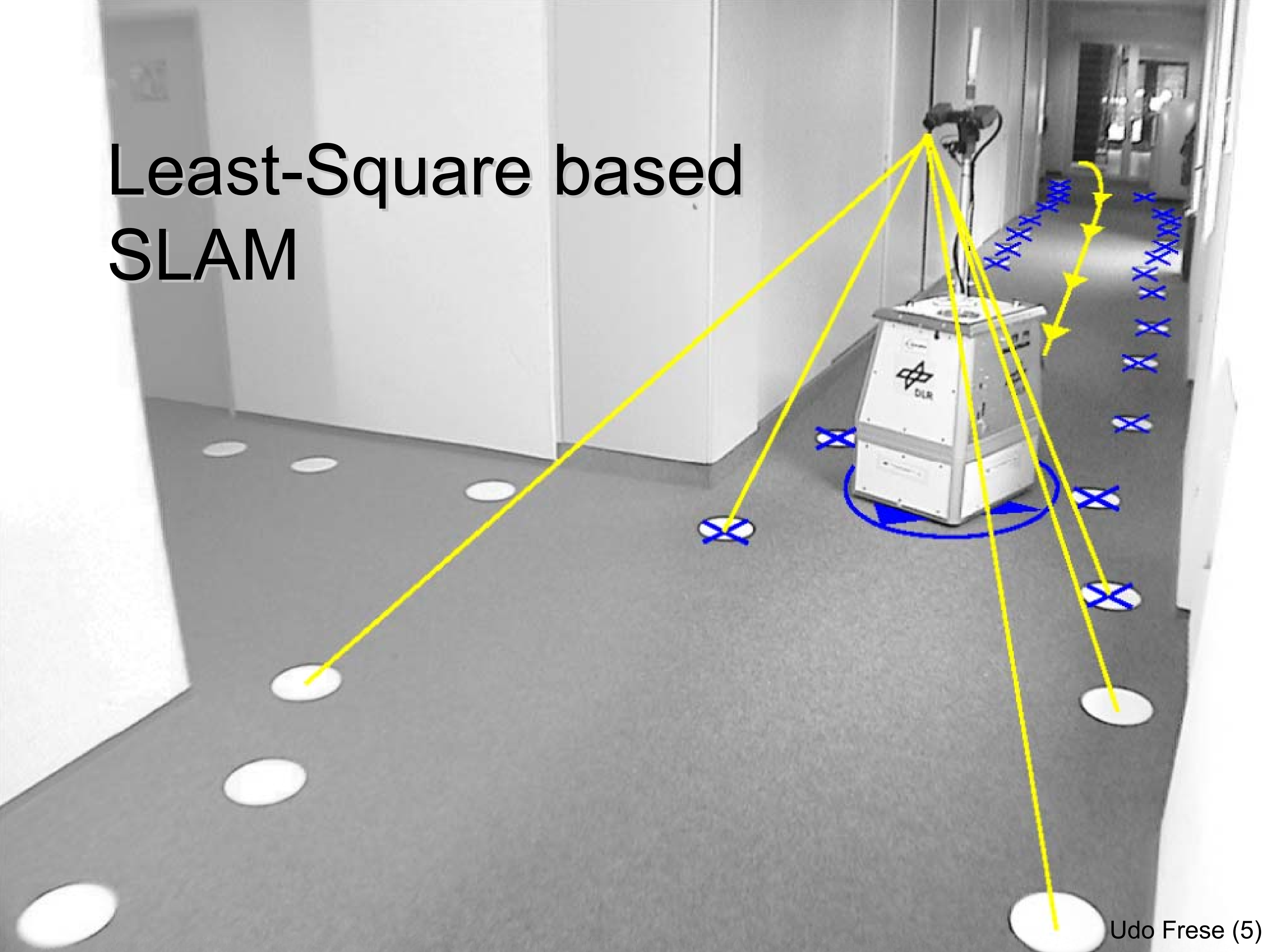
What is Least-Square based SLAM?



Overview

- least-square based SLAM
- linearization
- sparsity
- least-square on manifolds

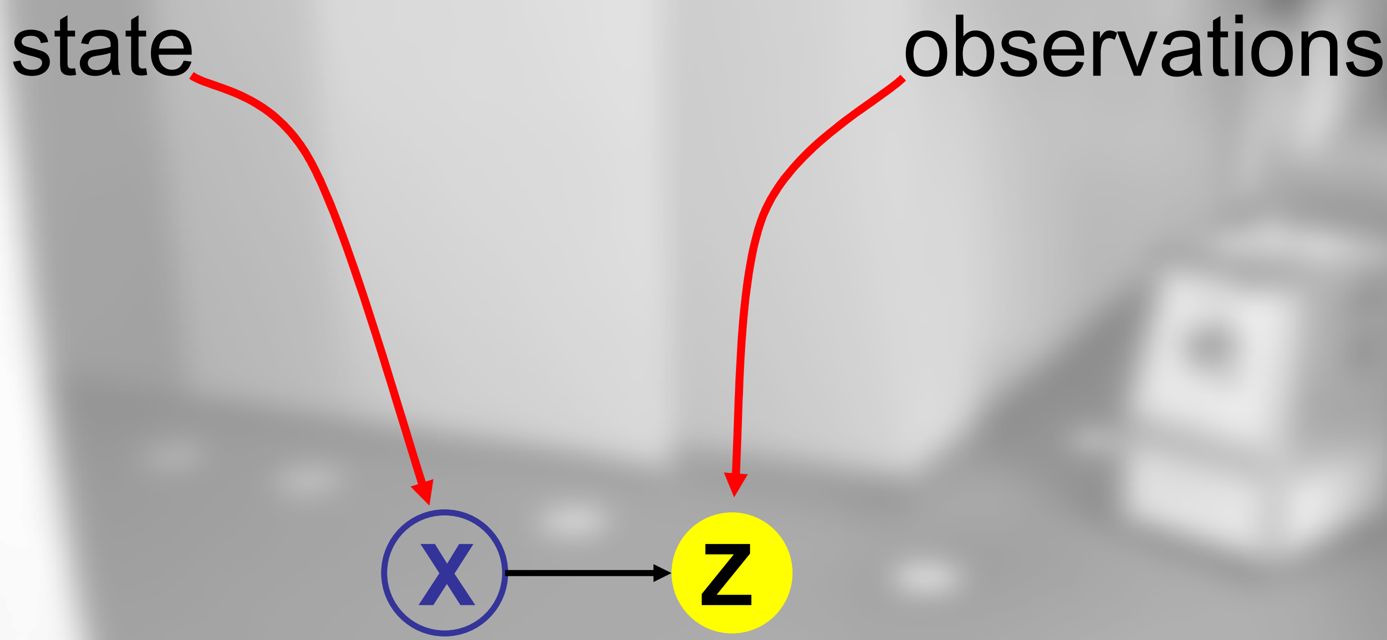
Least-Square based SLAM



Simultaneous Localization and Mapping

- invented by C.F. Gauss
  - celestial body prediction
  - surveying the kingdom of Hanover
- contribution
  - probabilistic view as maximum likelihood (Gaussian distribution)
  - reduce to linear(-ized) equation system
  - solve that (Gauss-Seidel iteration, Gaussian elimination)

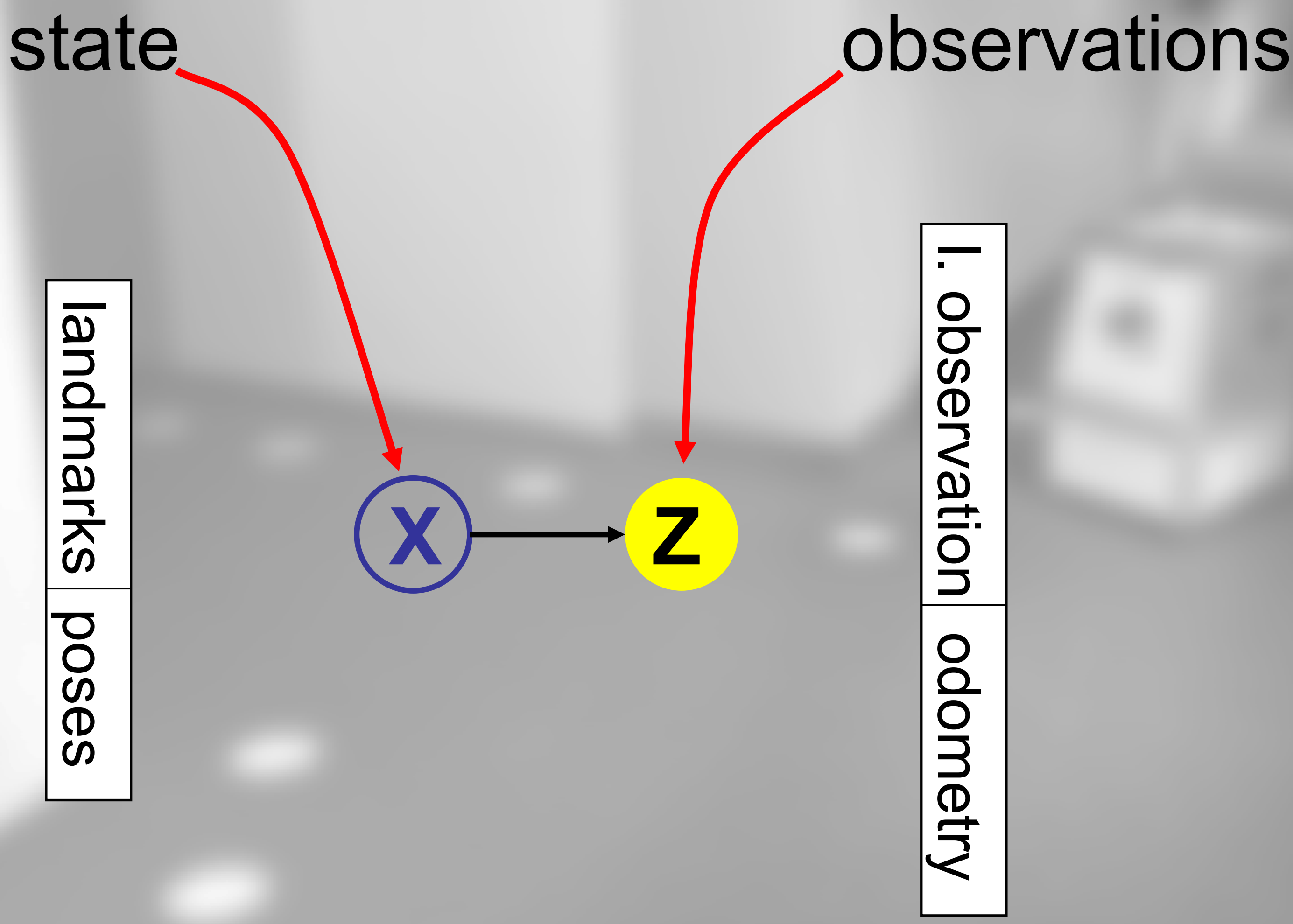
Least Square based SLAM



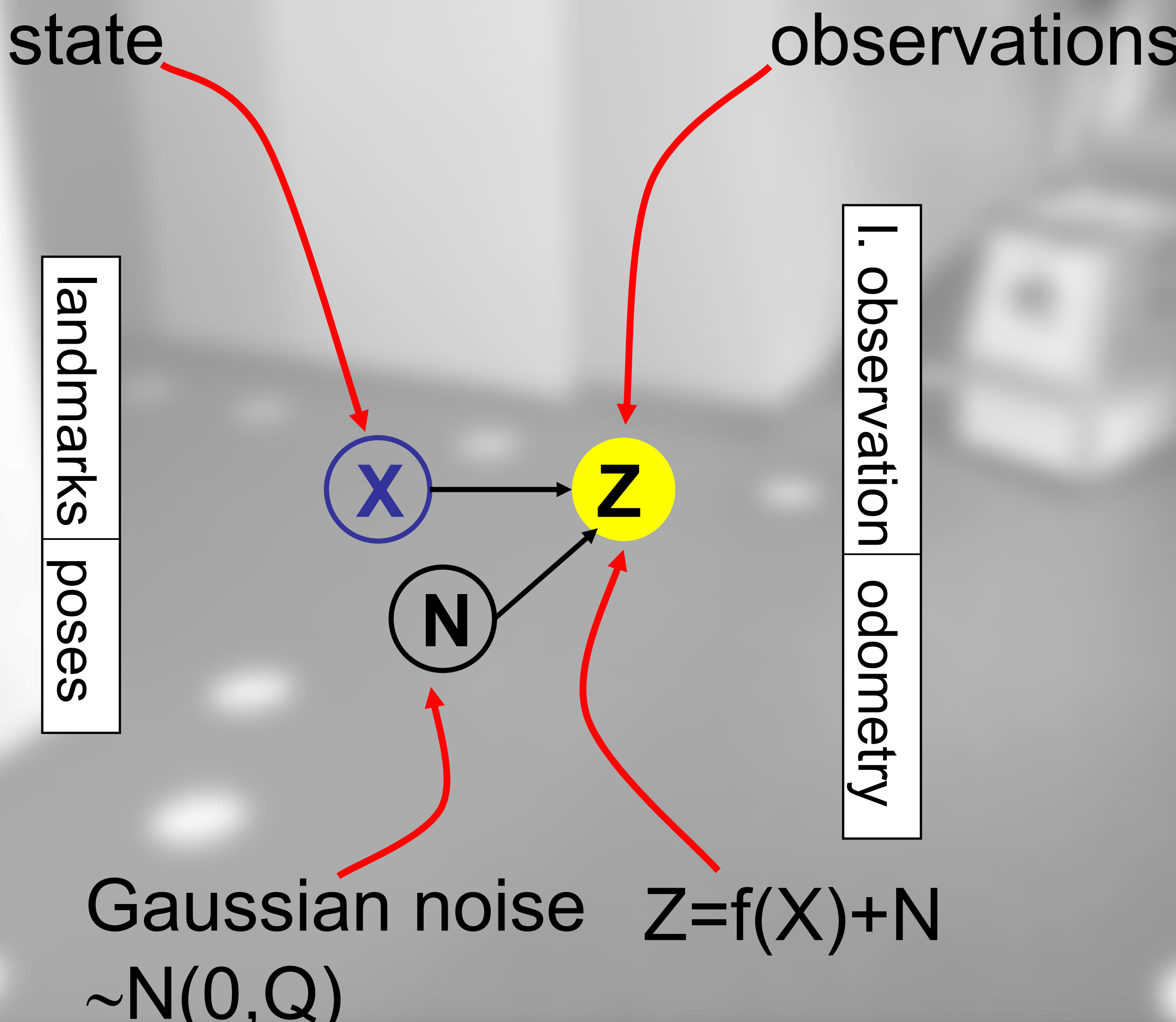
Question to the audience

- How do the vectors X and Z look like?

Least Square based SLAM



Least Square based SLAM



Least Square based SLAM

$$p\left(X = x \mid Z = z\right)$$

$$= \frac{p\left(Z = z \mid X = x\right) p\left(X = x\right)}{p\left(Z = z\right)}$$

$$\propto p\left(Z = z \mid X = x\right) p\left(X = x\right)$$

$$\propto p\left(Z = z \mid X = x\right)$$

$$= p\left(N = z - f\left(X\right) \mid X = x\right)$$

$$= p\left(N = z - f\left(x\right)\right)$$

$$\propto \exp\left(-\tfrac{1}{2}\left(z - f\left(x\right)\right)^T Q^{-1}\left(z - f\left(x\right)\right)\right)$$

Least Square based SLAM

$$\hat{x} = \arg\max_x p\left(X = x \mid Z = z\right)$$

$$= \arg\min_x \left(\tfrac{1}{2}\left(z - f(x)\right)^T Q^{-1}\left(z - f(x)\right)\right)$$

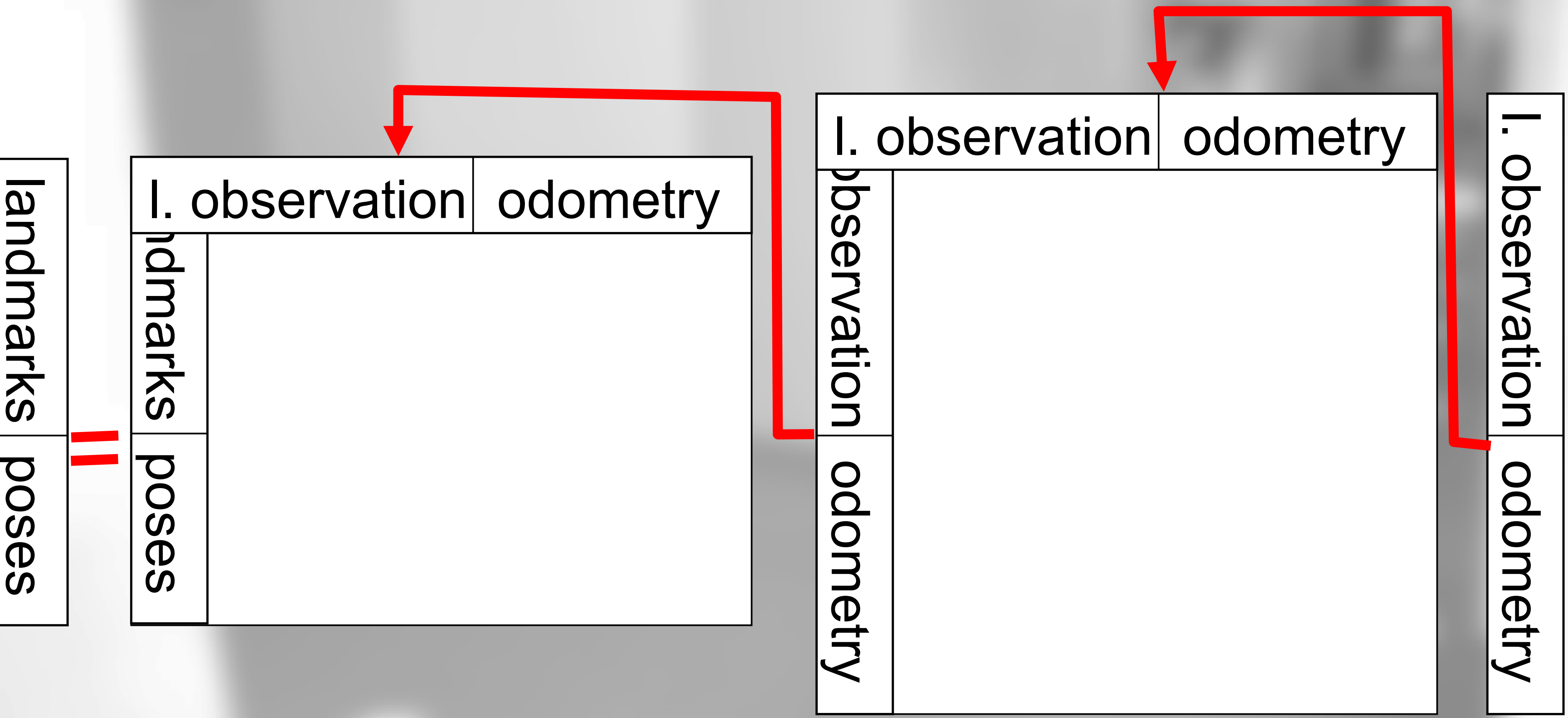
$$\Rightarrow 0 = -\left(z - f(\hat{x})\right)^T Q^{-1} \frac{df}{dx_i}(\hat{x}) \qquad \forall i$$

$$0 = -\left(z - f(\hat{x})\right)^T Q^{-1} \frac{df}{dx}(\hat{x})$$

$$0 = \frac{df}{dx}(\hat{x})^T Q^{-1}\left(z - f(\hat{x})\right)$$

gradient of f with respect to x_i

Jacobian of f



$$0 = \frac{df}{dx}(\hat{x})^T \cdot Q^{-1} \cdot \left(z - f(\hat{x})\right)$$

Least Square based SLAM

$$0 = \frac{df}{dx}(\hat{x})^T Q^{-1}\left(z - f(\hat{x})\right)$$

$$f(x) \approx f(\breve{x}) + \frac{df}{dx}(\breve{x})(x - \breve{x})$$

$$0 \approx \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) - \frac{df}{dx}(\breve{x})(\hat{x} - \breve{x})\right)$$

$$= \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) - \frac{df}{dx}(\breve{x})\hat{x} + \frac{df}{dx}(\breve{x})\breve{x}\right)$$

$$= -\frac{df}{dx}(\breve{x})^T Q^{-1}\frac{df}{dx}(\breve{x})\hat{x} + \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) + \frac{df}{dx}(\breve{x})\breve{x}\right)$$

information matrix information vector

Least Square based SLAM

$$= -\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) x + \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) + \frac{df}{dx}(\breve{x})\breve{x} \right)$$

$$\Rightarrow x =$$

$$\left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) + \frac{df}{dx}(\breve{x})\breve{x} \right)$$

$$= \breve{x} + \underbrace{\left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1}}_{C=\operatorname{cov}(x)} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x})\right)$$

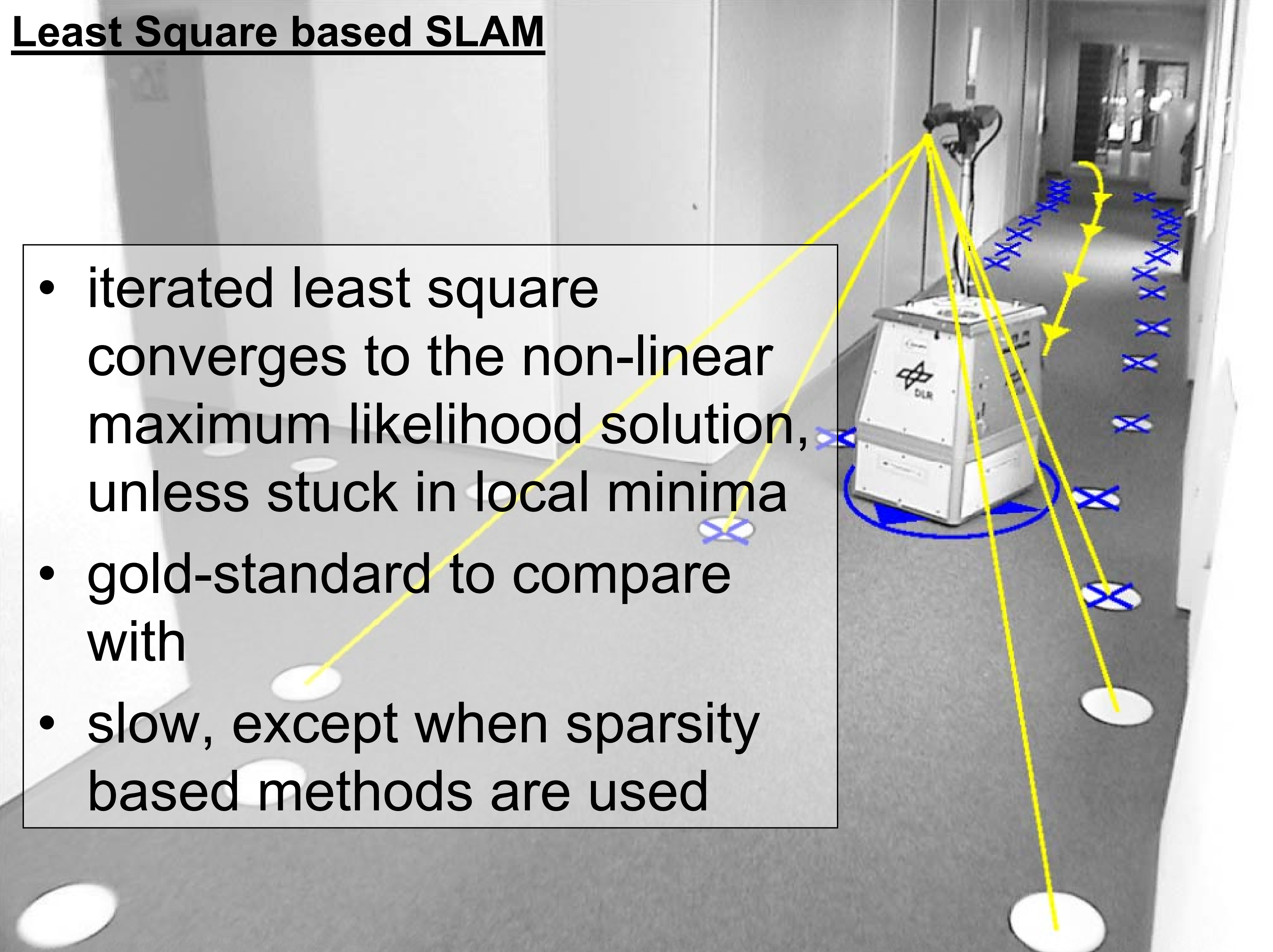
Least Square based SLAM

$$x = \breve{x} + \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x})\right)$$

- Linearized Maximum Likelihood / SLAM
  - solve the above equation
- Nonlinear Maximum Likelihood / SLAM
  - set x~ = x
  - iterate the above equation until convergence
  - non-linear minimum
  - gold-standard

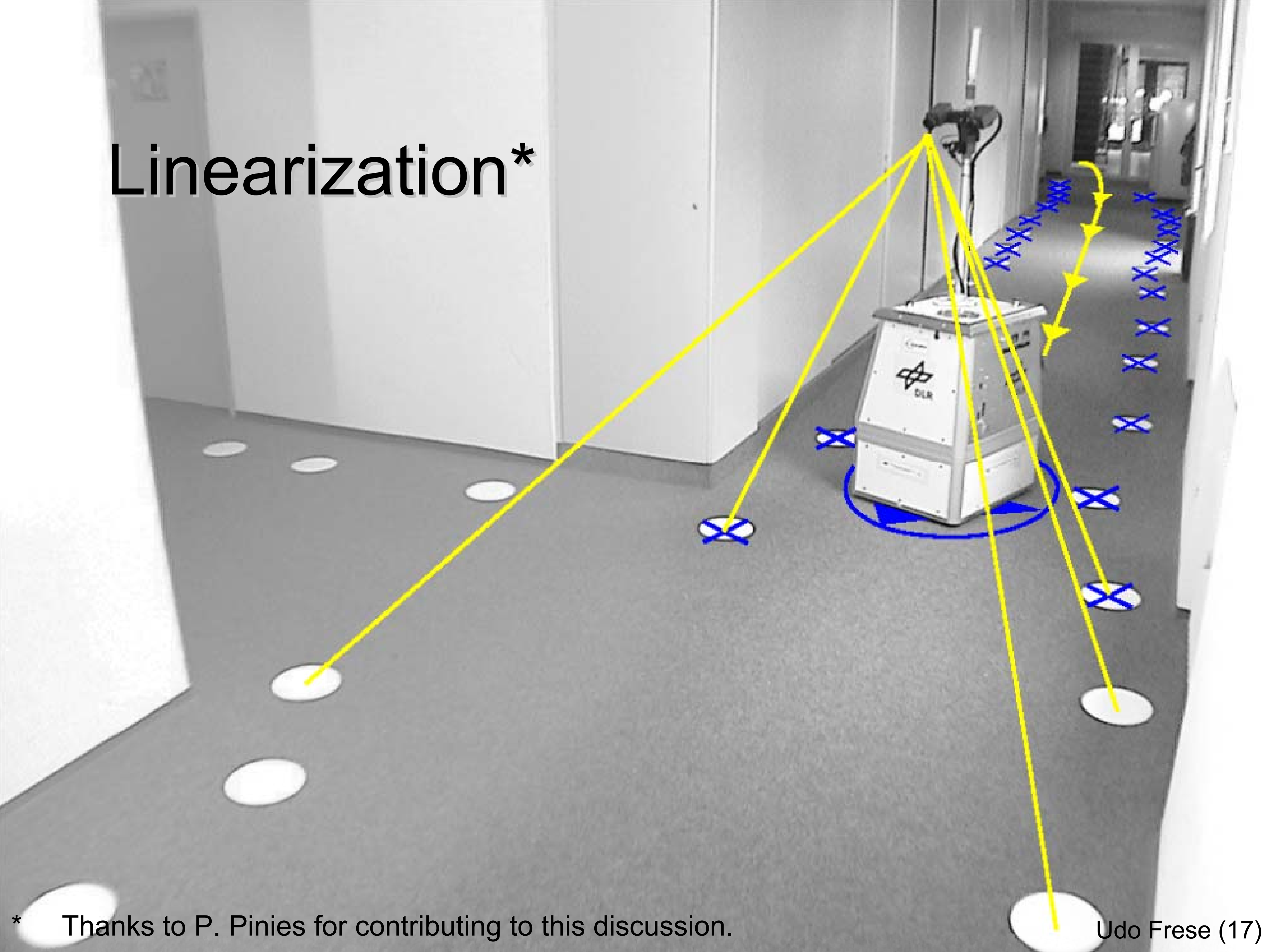
$$0 = \frac{df}{dx}(x)^T Q^{-1}\left(z - f(x)\right)$$

Least Square based SLAM



- iterated least square converges to the non-linear maximum likelihood solution, unless stuck in local minima
- gold-standard to compare with
- slow, except when sparsity based methods are used

Linearization*



* Thanks to P. Pinies for contributing to this discussion.

The (Extended) Kalman Filter from a Least-Square based Perspective

- KF implements rekursive (i.e. incremental) least square
- applies Woodbury formula for updating the inverse of a matrix to the information matrix

Linearization

$$C^+ = \left(\frac{df^+}{dx}(\breve{x})^T Q^{+-1} \frac{df^+}{dx}(\breve{x}) \right)^{-1}, Q^+ = \begin{pmatrix} Q^- & \\ & Q^m \end{pmatrix}, f^+ = \begin{pmatrix} f^- \\ f^m \end{pmatrix}$$

$$= \left(\frac{df^-}{dx}(\breve{x})^T \left(Q^-\right)^{-1} \frac{df^-}{dx}(\breve{x}) + \frac{df^m}{dx}(\breve{x})^T \left(Q^m\right)^{-1} \frac{df^m}{dx}(\breve{x}) \right)^{-1}$$

$$= \left(\left(C^-\right)^{-1} + \frac{df^m}{dx}(\breve{x})^T \left(Q^m\right)^{-1} \frac{df^m}{dx}(\breve{x}) \right)^{-1}$$

Wood-
bury

$$= C - \underbrace{C \frac{df^m}{dx}(\breve{x})^T \left(\frac{df^m}{dx}(\breve{x}) C \frac{df^m}{dx}(\breve{x})^T + Q^m \right)^{-1}}_{K} \frac{df^m}{dx}(\breve{x}) C$$

Linearization

- EKF is a KF working on the linearization…

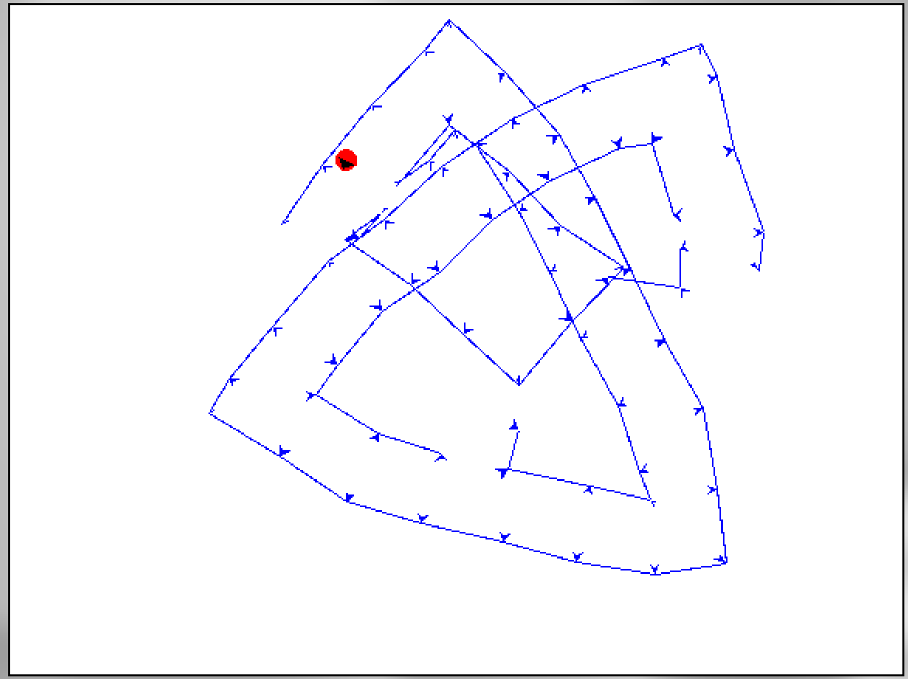
$$f(x) \approx f(\breve{x}) + \frac{df}{dx}(\breve{x})(x - \breve{x})$$

$$x^{+} = x^{-} + K\left(z - \left(f(\breve{x}) + \frac{df}{dx}(\breve{x})(x^{-} - \breve{x})\right)\right)$$

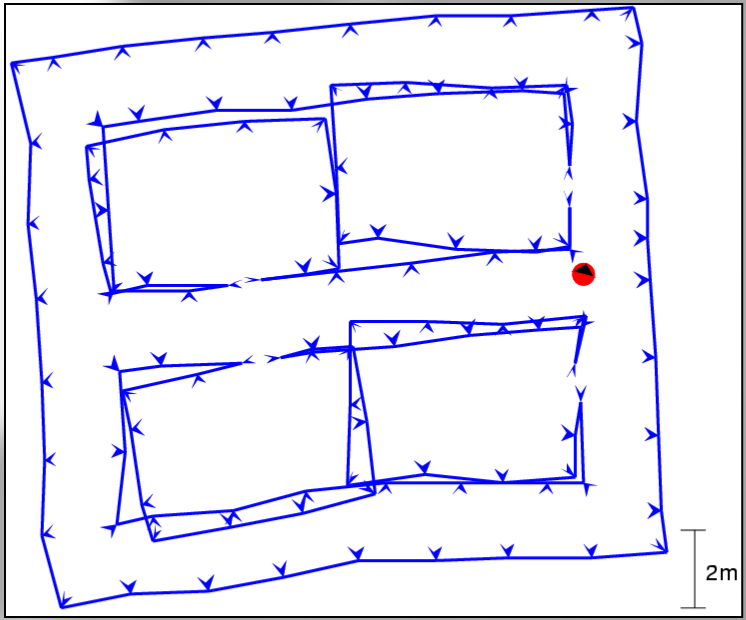
$$= x^{-} + K\left(z - f(\breve{x}) - \frac{df}{dx}(\breve{x})(x^{-} - \breve{x})\right)$$

- ..at the prior estimate
- you can't change linearization point by changing the Jacobian only
- otherwise a term as in the iEKF appears

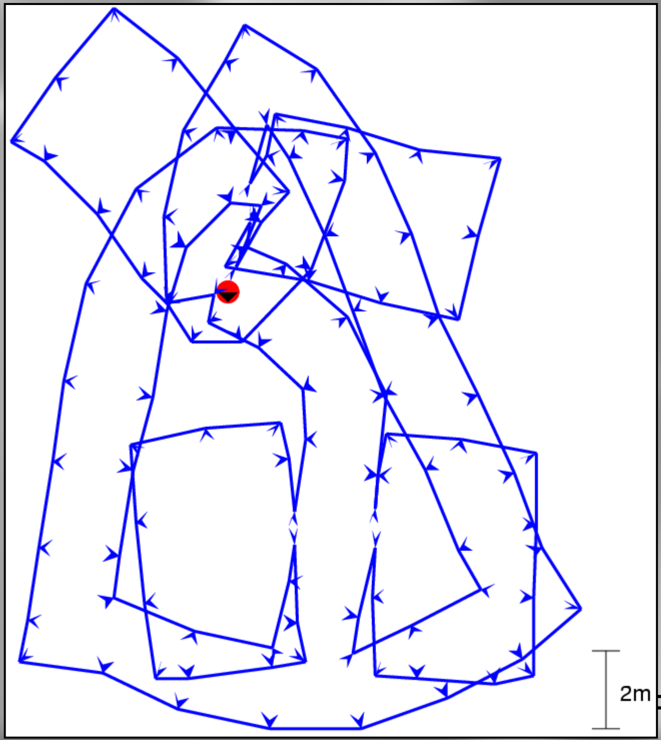
Linearization



before loop closing

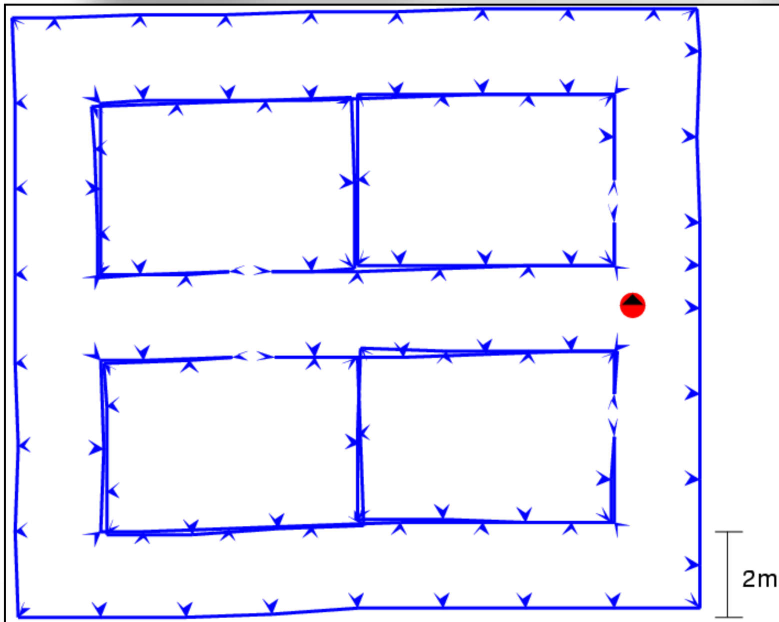


Maximum Likelihood



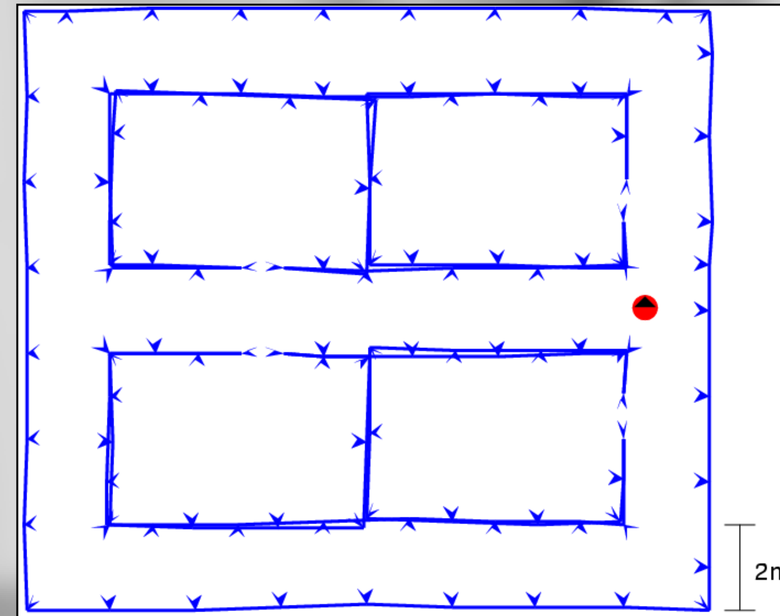
EKF

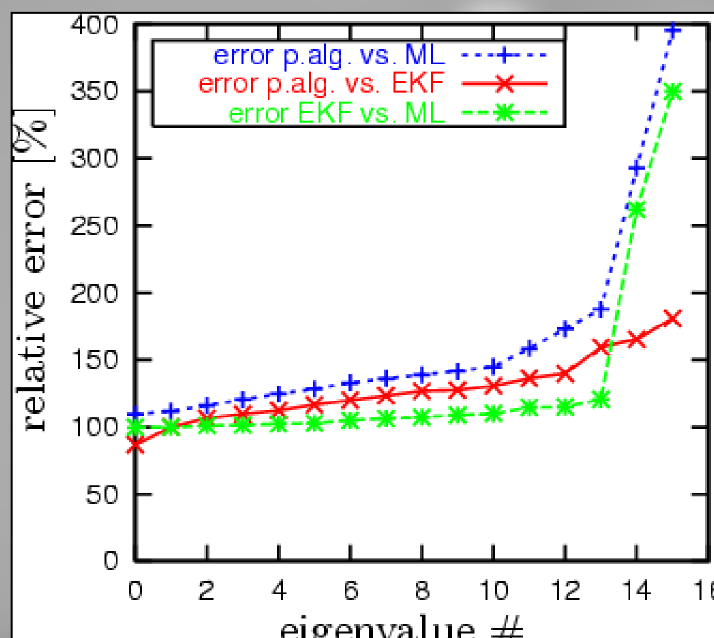
Linearization



Maximum Likelihood

EKF





Linearization

- EKF is a KF working on the linearization at the prior estimate
- iEKF is a KF working on the linearization at the posterior estimate
- ⇒when thinking about linearization
  - only the linearization points count
  - marginalization steps do not matter
  - block/sequential update does not matter, except through the linearization point

Question to the Audience: Which linearization points are used for the different observations?

	z_1	z_2	u_1	z_3	z_4	u_2	z_5	z_6
	lm a	lm b	odo	lm a	lm b	odo	lm a	lm b
Batch LS	? ?							
EKF block	? ?							
EKF single	? ?							
iEKF single	? ?							
iEKF block	? ?							
Levenb. -Margq.	? ?							

Linearization

	z_1	z_2	u_1	z_3	z_4	u_2	z_5	z_6
	lm a	lm b	odo	lm a	lm b	odo	lm a	lm b
Batch LS	x|	x|	x|	x|	x|	x|	x|	x|
EKF block	x|	x|	$x\|z_{1,2}$	$x\|z_{1,2},u_1$	$x\|z_{1,2},u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..4}, u_{1,2}$	$x\|z_{1..4}, u_{1,2}$
EKF single	x|	$x\|z_1$	$x\|z_{1,2}$	$x\|z_{1,2},u_1$	$x\|z_{1..3}, u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..4}, u_{1,2}$	$x\|z_{1..5}, u_{1,2}$
iEKF single	$x\|z_1$	$x\|z_{1,2}$	$x\|z_{1,2}$	$x\|z_{1..3}, u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..5}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$
iEKF block	$x\|z_{1,2}$	$x\|z_{1,2}$	$x\|z_{1,2}$	$x\|z_{1..4}, u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..4}, u_1$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$
Levenb.-Margq.	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$	$x\|z_{1..6}, u_{1,2}$

Linearization

- still, all EKF variants use different, i.e. inconsistent linearization points for different observations, because they cannot change relinearize an observation once it is integrated.

$$x = \breve{x} + \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x})\right)$$

Linearization

- robot at (0,0,θ) observes landmark at (x,y)

$$\begin{pmatrix}1\\0\end{pmatrix} = z \approx f\begin{pmatrix}\theta\\x\\y\end{pmatrix} = \begin{pmatrix}\cos\theta\, x + \sin\theta\, y\\ -\sin\theta\, x + \cos\theta\, y\end{pmatrix}$$

- linearized at (0,1,0) and (0,2,0)

$$1 \approx 1x + \theta y \approx x \qquad 0 \approx -\theta x + 1y \approx -\theta + y$$

$$1 \approx 1x + \theta y \approx x \qquad 0 \approx -\theta x + 1y \approx -2\theta + y$$

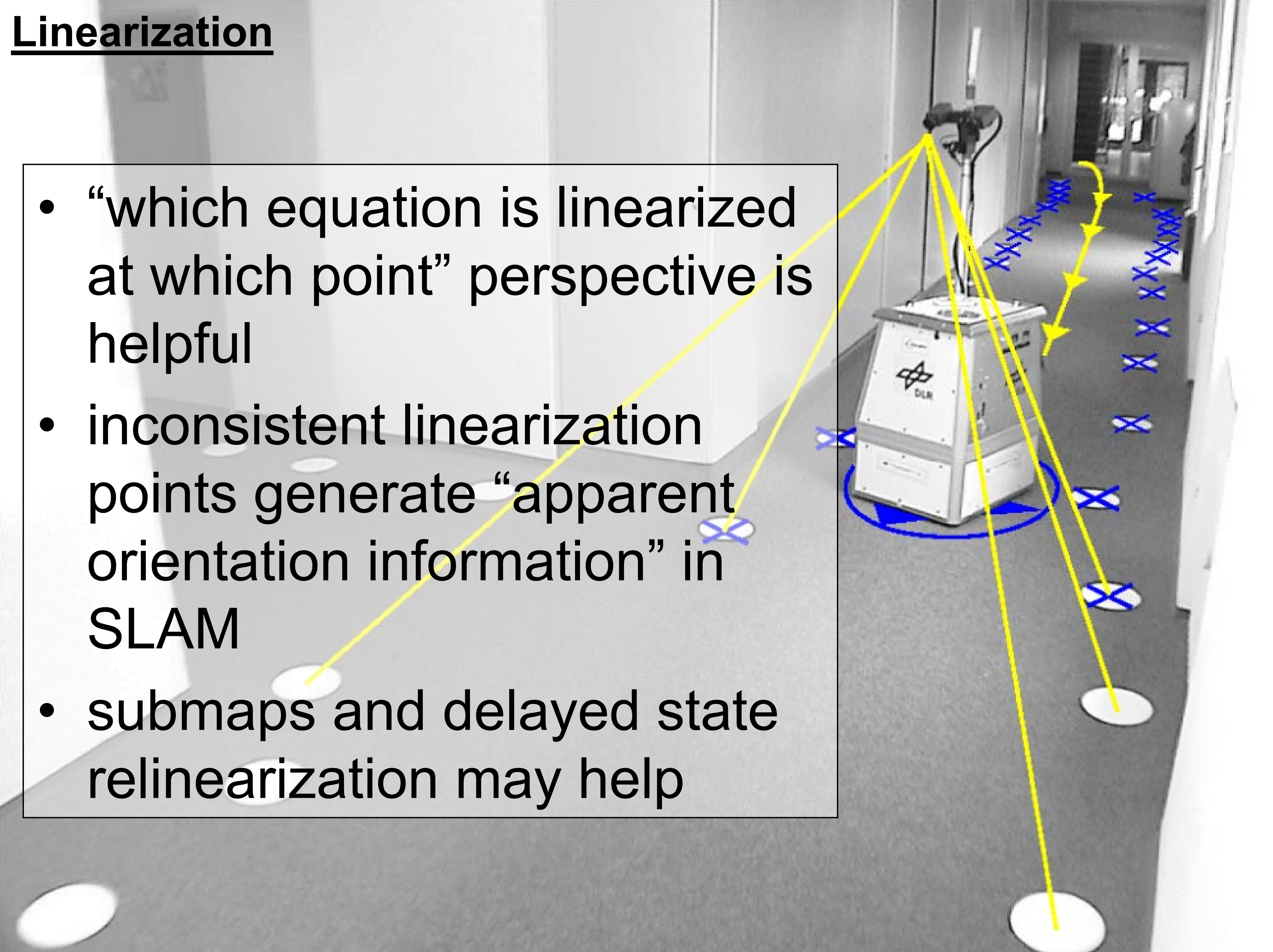
- by subtracting both right equations it can be seen, that there is "apparent" θ information

$$0 \approx \left(-\theta + y\right) - \left(-2\theta + y\right) = \theta$$

Linearization

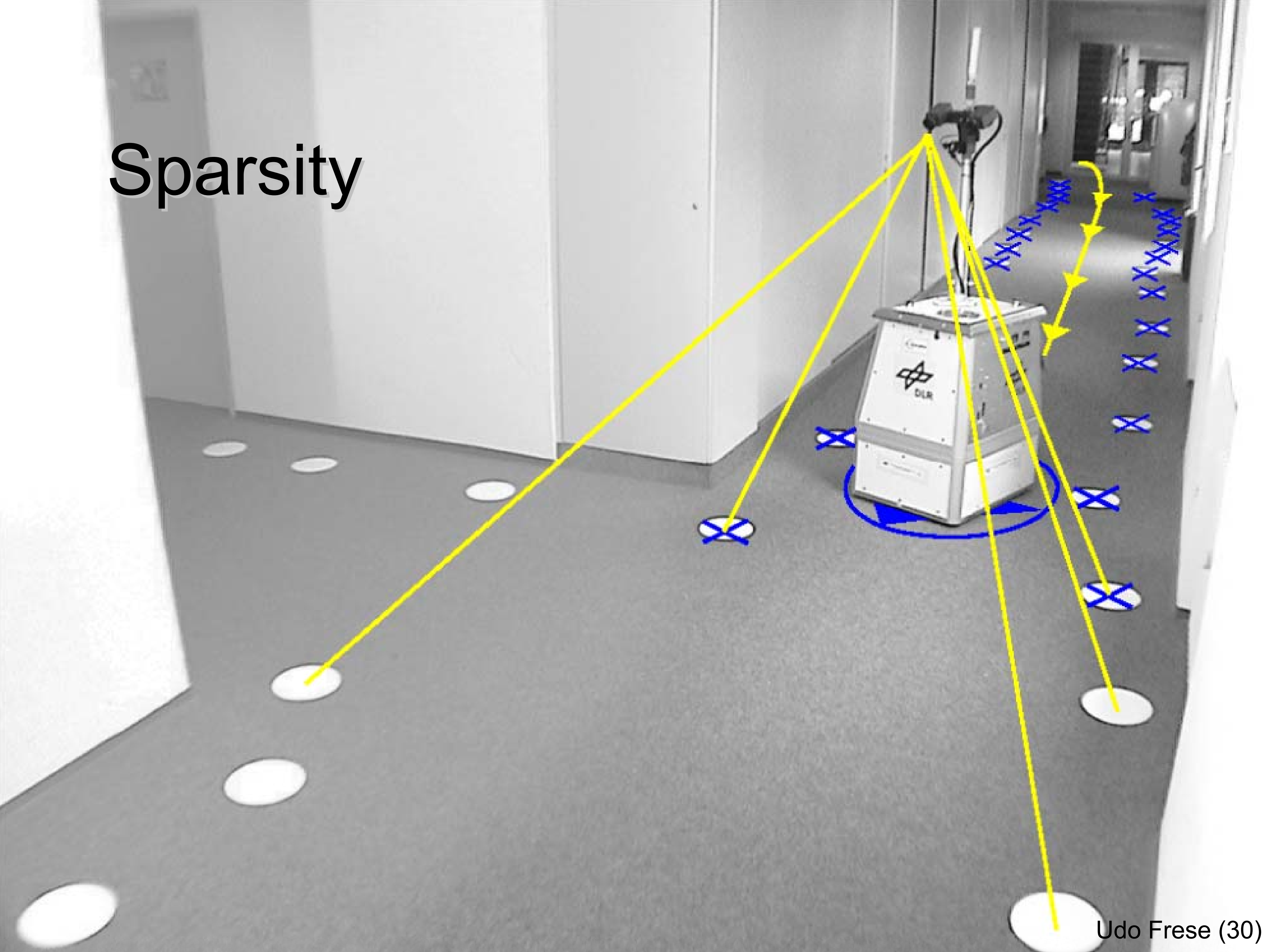
- inconsistent linearization points lead to *apparent absolute orientation* information in the covariance/information matrix
- in SLAM the real orientation information becomes smaller and smaller, hence the filter becomes inconsistent
- the problem is more about inconsistent linearization points than about wrong linearization points
- delayed state relinearization or submaps can help

<u>**Linearization**</u>



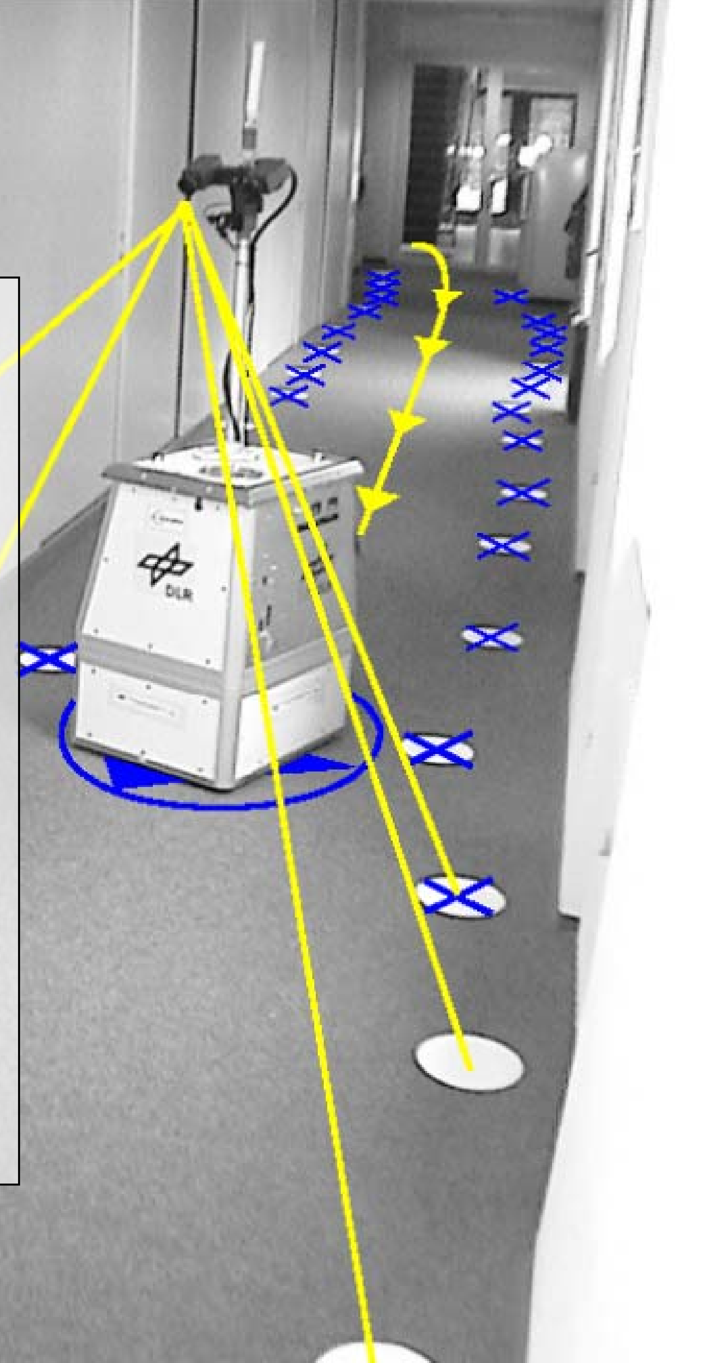
- "which equation is linearized at which point" perspective is helpful
- inconsistent linearization points generate "apparent orientation information" in SLAM
- submaps and delayed state relinearization may help

Sparsity

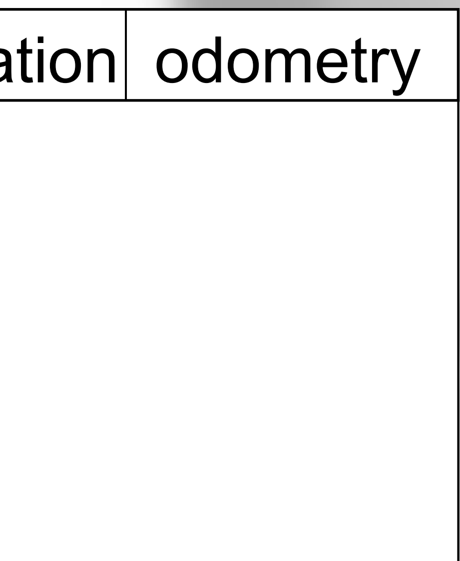


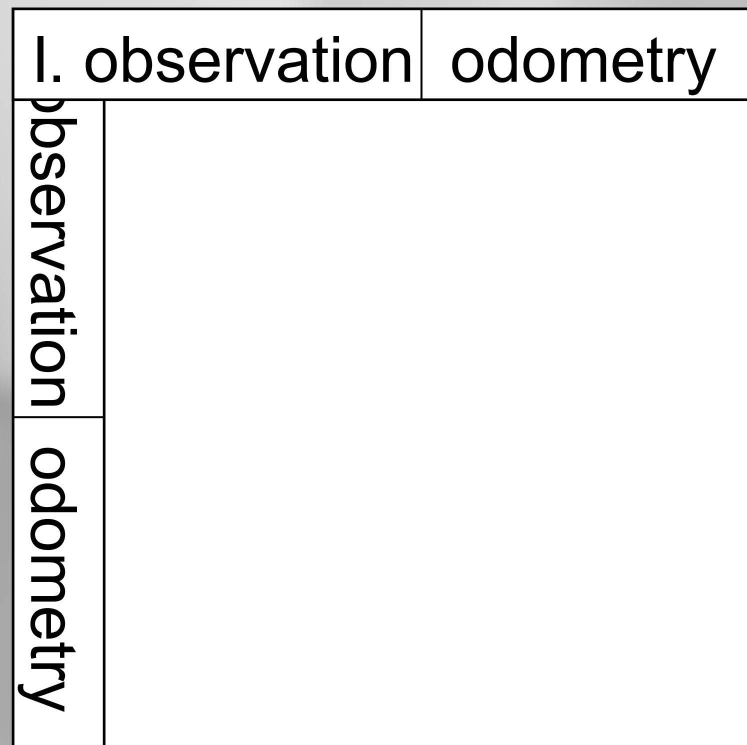
Sparsity

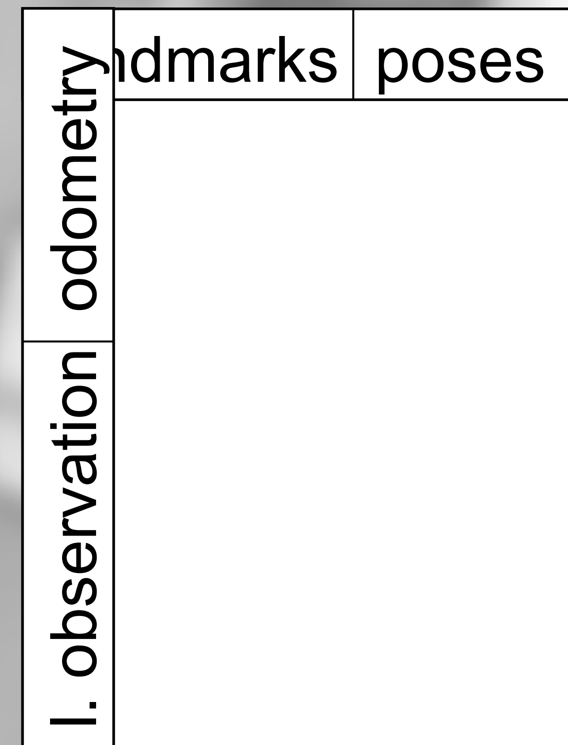
- huge matrices in LS SLAM
- how can we make computation fast enough?
- understand the block and sparsity pattern for •
- use Tim Davis's csmatrix sparse matrix package* for $()^{-1}$



* Thanks to A. Nuechter and M. Kaess for pointing this out.





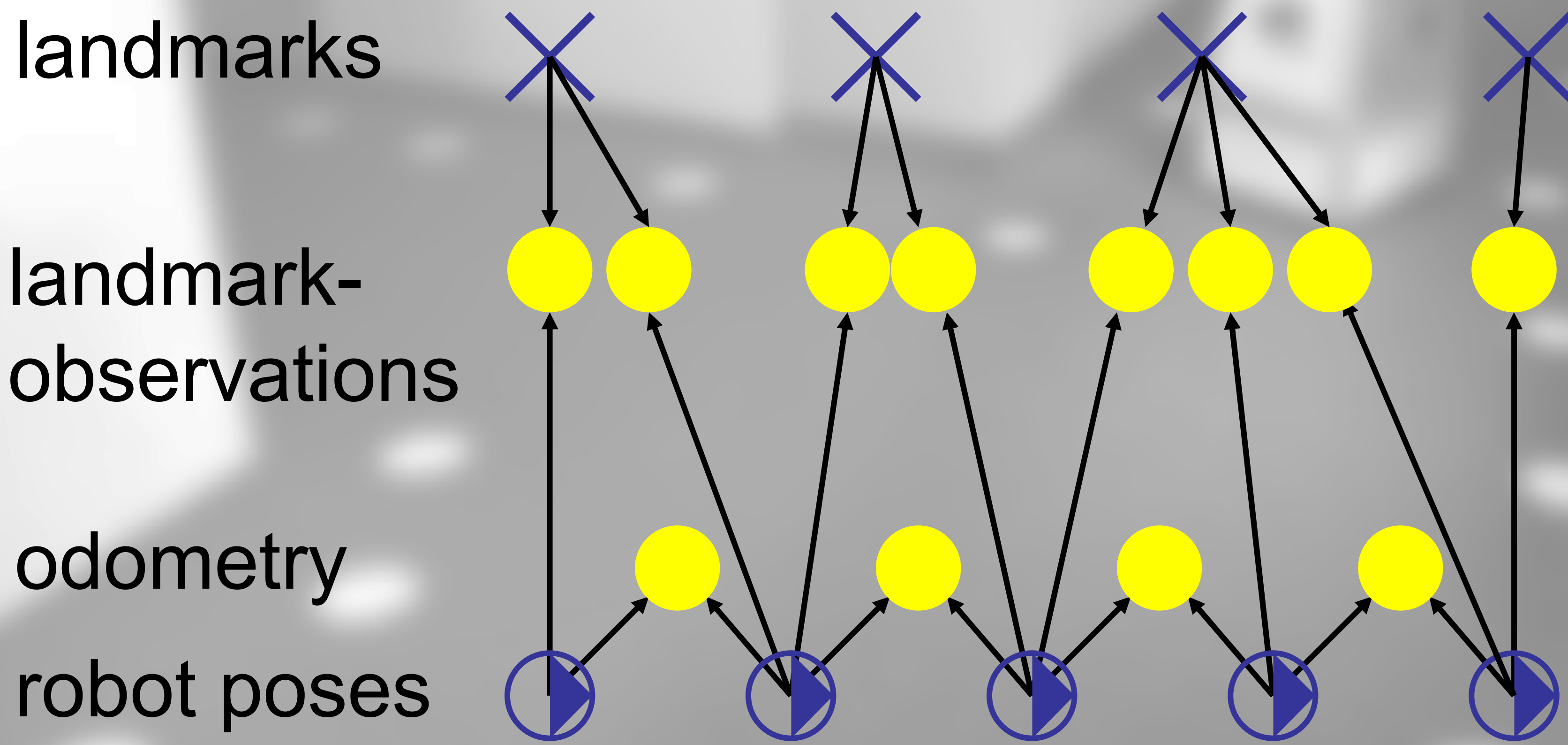


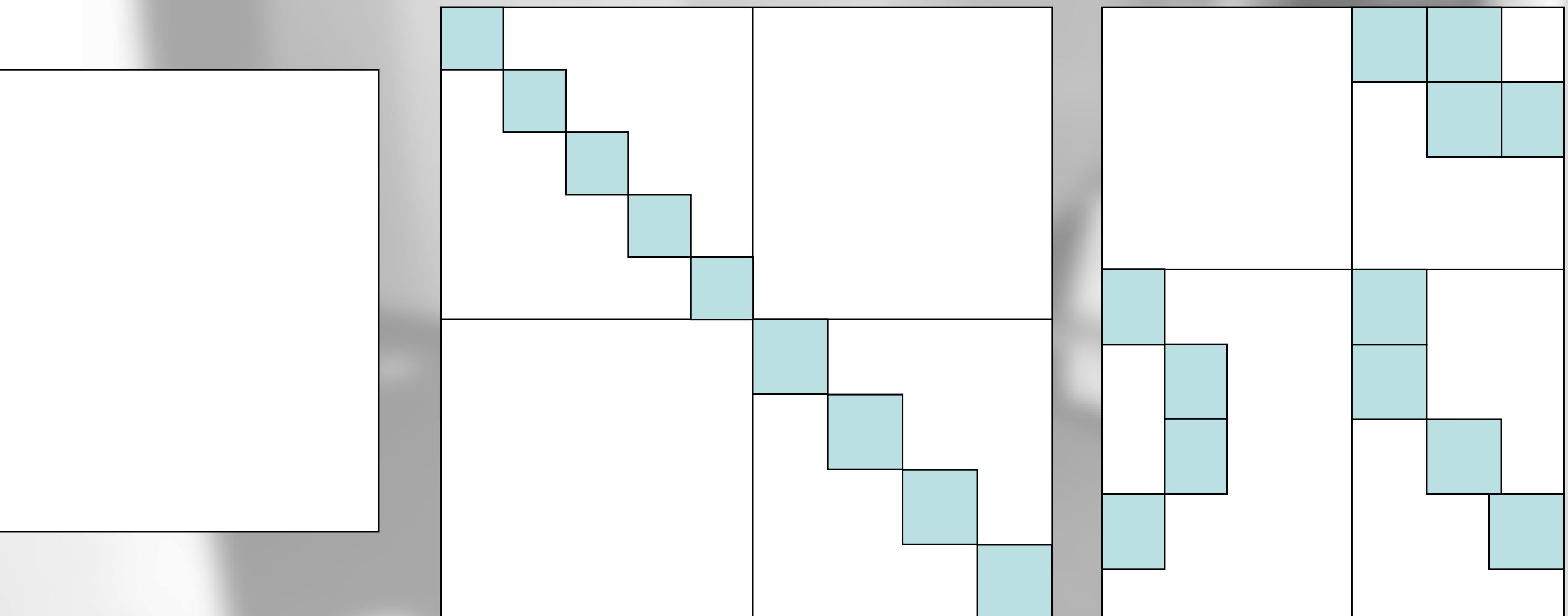
$$\frac{df}{dx}(\breve{x})^T \quad \cdot \quad Q^{-1} \quad \cdot \quad \frac{df}{dx}(\breve{x})$$

Least Square based SLAM

Question to the audience:

- What information on Q and df/dX is coded in this Bayes net?





$$\frac{df}{dx}(\breve{x})^T \quad \cdot \quad Q^{-1} \quad \cdot \quad \frac{df}{dx}(\breve{x})$$

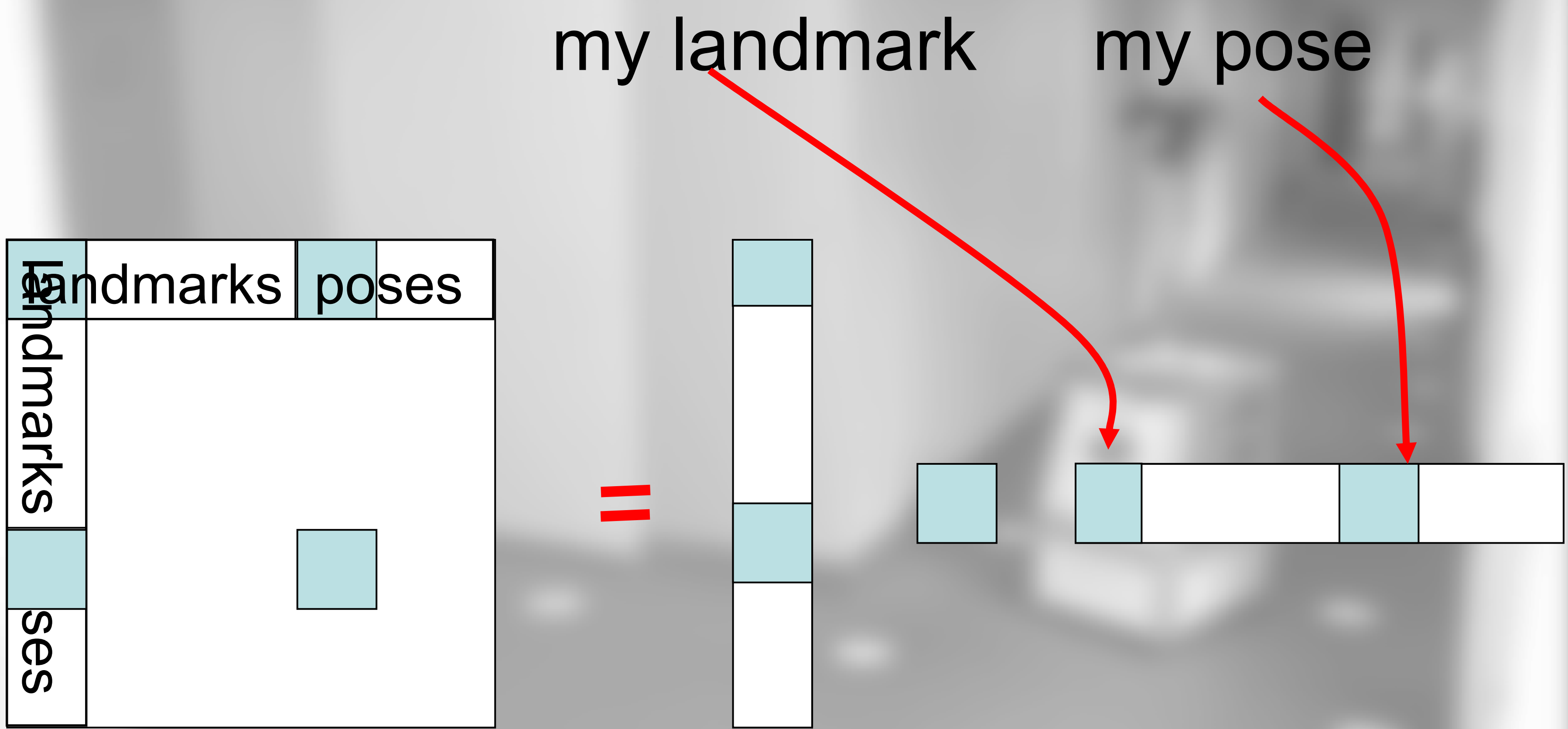
Least Square based SLAM

- exploit that C is block diagonal, i.e. measurements are independent

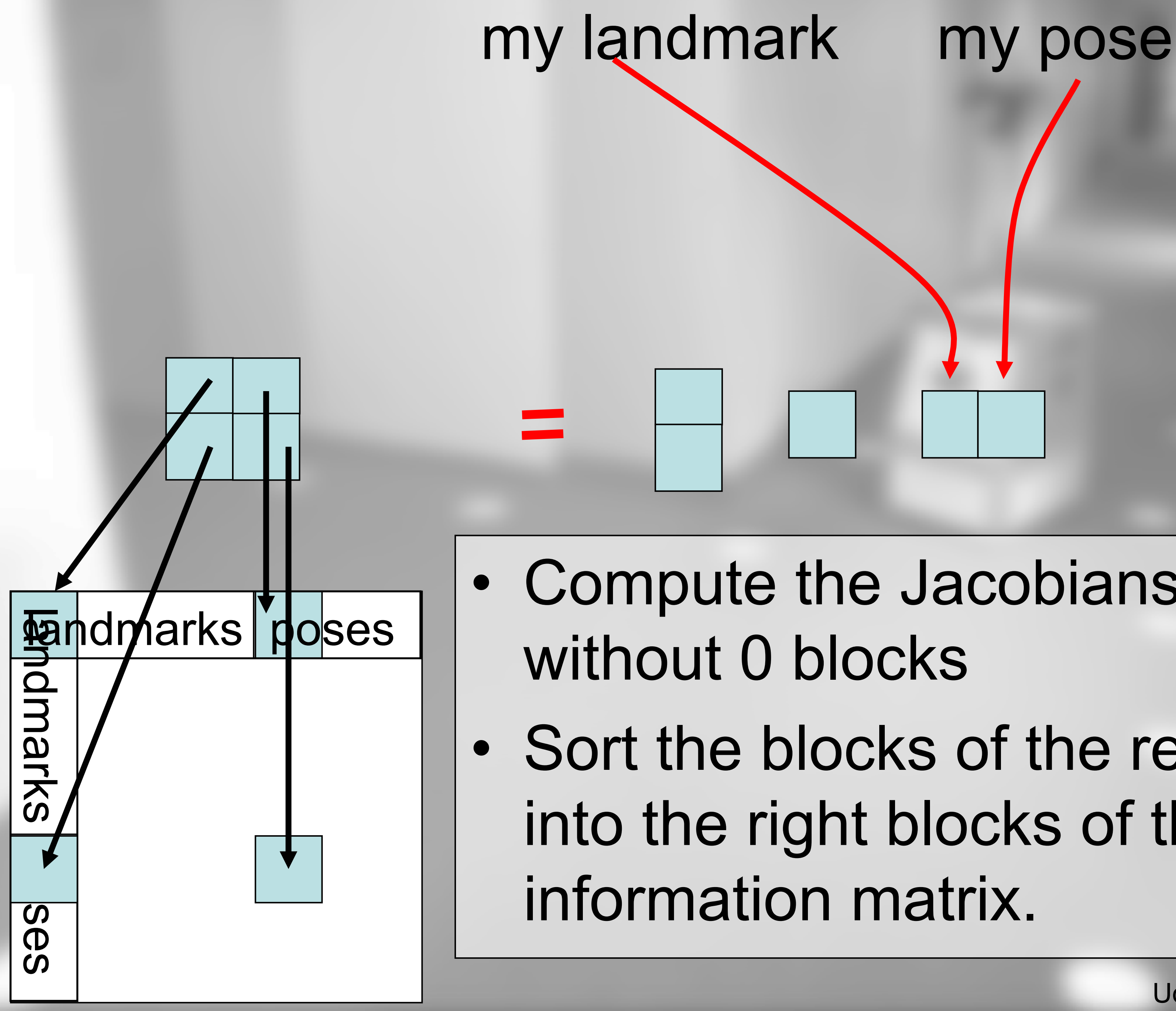
$$\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x})$$

$$= \sum_i \frac{df_i}{dx}(\breve{x})^T Q_i^{-1} \frac{df_i}{dx}(\breve{x})$$

- information “adds up” in the information matrix



$$\frac{df_i}{dx}(\breve{x})^T \cdot Q_i^{-1} \cdot \frac{df_i}{dx}(\breve{x})$$



- Compute the Jacobians without 0 blocks
- Sort the blocks of the result into the right blocks of the information matrix.

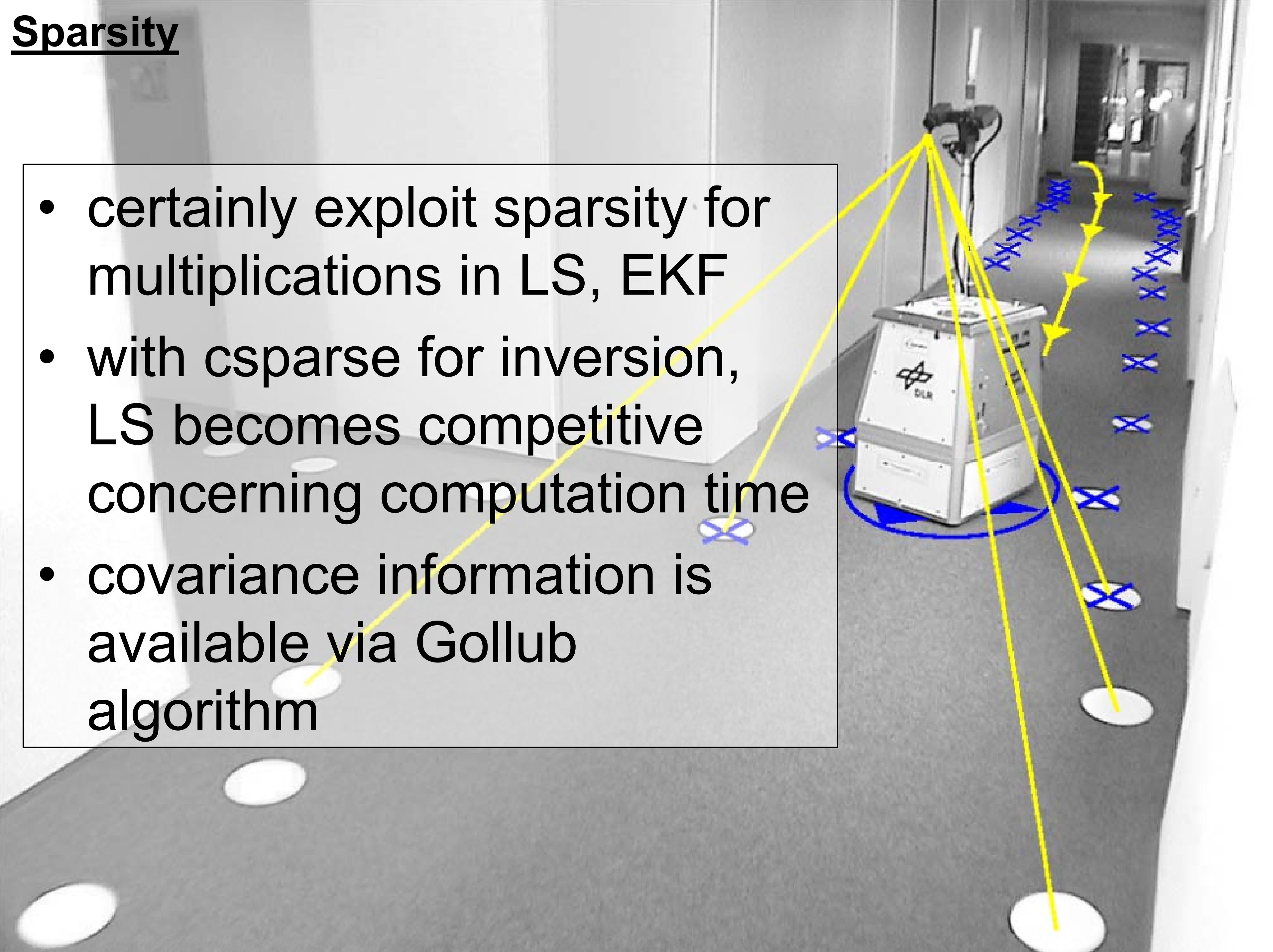
Least Square based SLAM

$$x = \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x})\right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x}) + \frac{df}{dx}(\breve{x})\breve{x}\right)$$

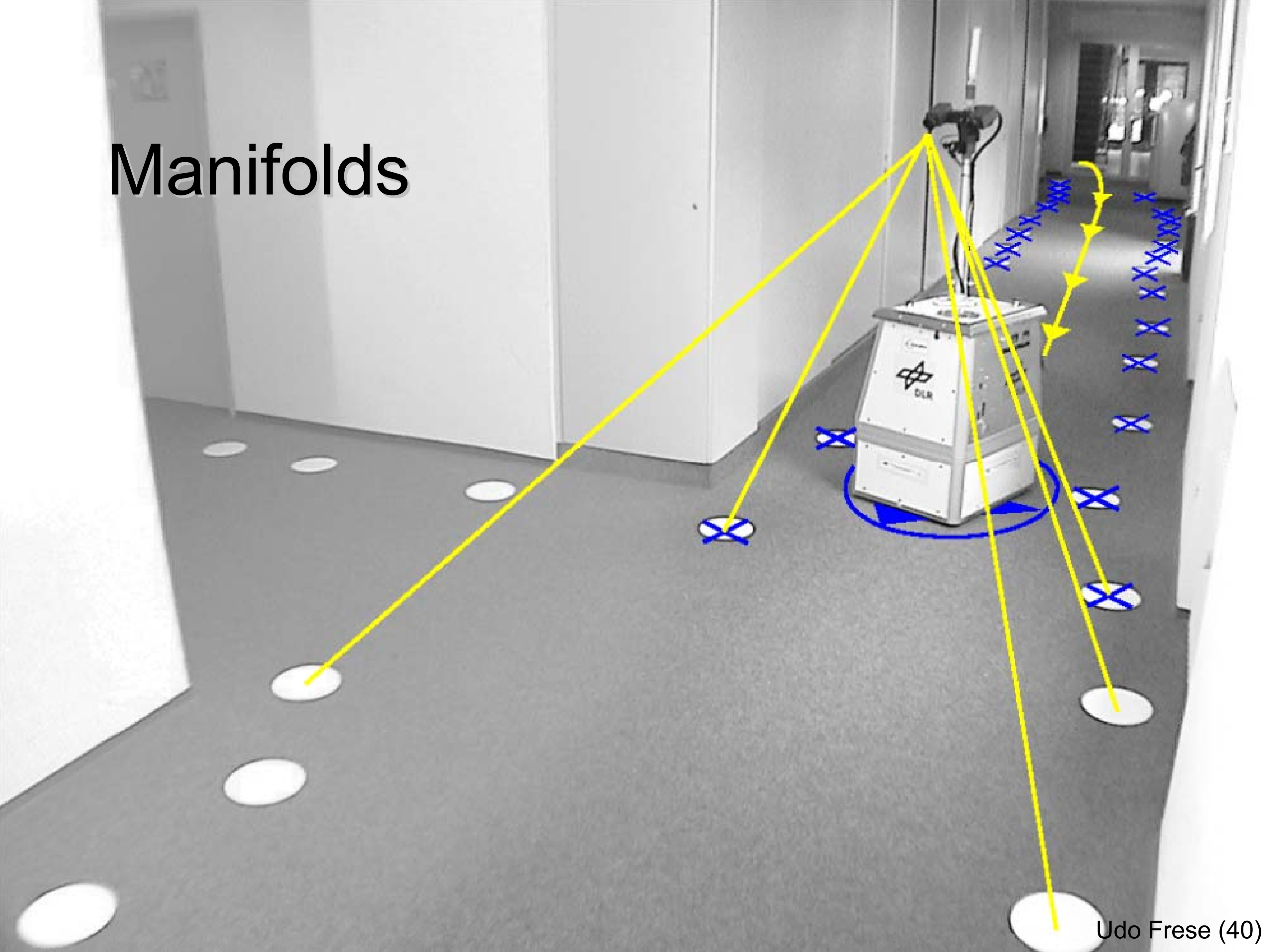
- how to do the inversion?
- solve an equation instead (MATLAB \)
- use Tim Davis' csparse package
- available for C++ or MATLAB
- selected parts of the inverse can be computed by the Gollub algorithm

Sparsity

- certainly exploit sparsity for multiplications in LS, EKF
- with csparse for inversion, LS becomes competitive concerning computation time
- covariance information is available via Gollub algorithm



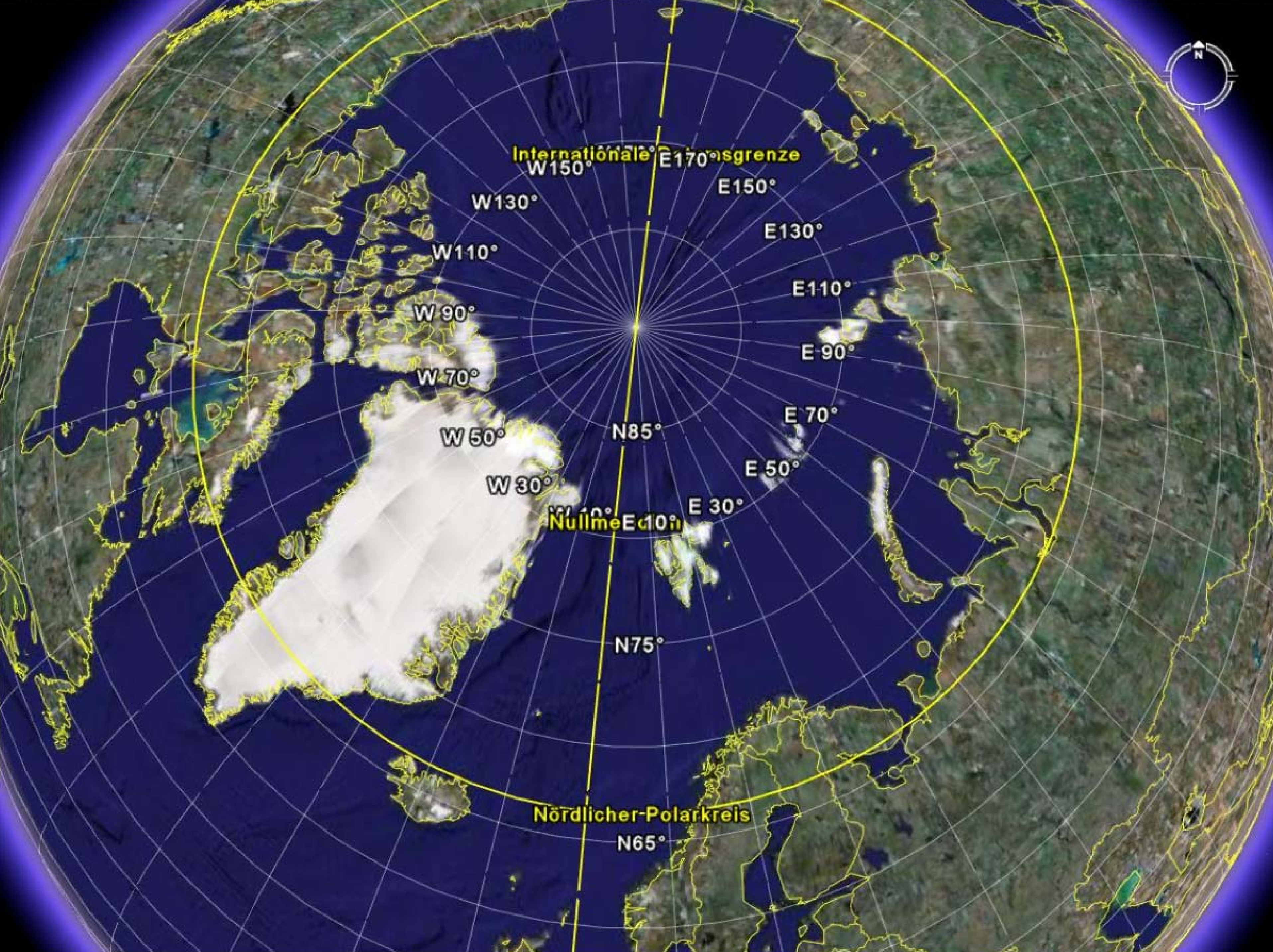
Manifolds



Manifolds

problem: some states are not vectors

- 2D orientation has 2π periodicity
- 3D orientation has 3-DOF, represented as
  - 3 Euler angles with singularity
  - unit quarternion q, with $|q|=1$
  - quarternion $q\neq 0$, where $|q|$ does not matter
  - 3×3 matrix, Q with $Q^TQ=I$
- 3D direction ($\rightarrow$inverse depth) has 2-DOF,
  - 2 angles with singularity
  - unit vector v, with $|v|=1$
  - vector $v\neq 0$, where $|v|$ does not matter


Internationale
W150°
E170°
E150°
W130°
E130°
W110°
E110°
W 90°
E 90°
W 70°
E 70°
W 50°
N85°
E 50°
W 30°
E 30°
N75°
Nördlicher-Polarkreis
N65°
N

Manifolds

- all these states need special treatment
- look locally like R^n, but globally different
- they are called *manifolds* in mathematics

Manifolds

- observation / dynamic functions view the state S as
  - structured, such as an object oriented class
  - with components that have a specific name, type, and meaning
- generic algorithms (e.g. EKF update equation, LS, etc.) view the state S as
  - a flat vector
  - with as many numbers as DOF
  - without anything additional to consider

Idea

- treat S as a encapsulated black-box data-type and use an operator $[+]: S \times R^n \rightarrow S$ to provide flat vector access for the generic algorithm
- [+] applies a local perturbation parameterized by a flat vector to the state
- n corresponds to DOF of the state
- encapsulation as in OO-design
- axiomatization as in mathematics

Manifolds

- motivated by "symmetries and perturbations"[1]
- some prior work but without the framework view[2]
- related to Lie-algebras and manifolds
- but thorough mathematical structure is still unclear to me

1 J.A. Castellanos, J.M.M. Montiel, J. Neira, J.D. Tardos The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building, IEEE Transactions on Robotics and Automation, 1999
2 E. Kraft. A quaternion-based unscented kalman filter for orientation tracking, 2003

Example: 3-D Orientations SO(3)

- Rot(v) is a rotation around v by an angle of |v|
- quarternions: $$Rot(v) = \left(\cos\left(\frac{|v|}{2}\right), v\frac{\sin(|v|/2)}{|v|} \right)$$
- matrices:

$$Rot(v) = \cos(|v|)I + \frac{\sin(|v|)}{|v|}[v]_\times + \frac{(1-\cos(|v|))}{|v|^2}vv^T$$

Question to the audience: Is there a singularity at v=0? Or anywhere else?

$$Rot(v) = \left(\cos\left(\frac{|v|}{2}\right), v\frac{\sin(|v|/2)}{|v|} \right)$$

Question to the audience: Is there a singularity at v=0? Or anywhere else?

$$Rot(v) = \left(\cos\left(\frac{|v|}{2}\right), v\frac{\sin(|v|/2)}{|v|} \right)$$

- not at 0, since sinc(0)=1, and sinc'(0)=0, so Rot(v) $\approx$(1,v) at v=0
- however, singularity at |v|=2π, since changing the direction of v has no effect then

Example: 3-D Orientations SO(3)

- Rot(v) is a rotation around v by an angle of |v|
- quarternions: $$Rot(v) = \left(\cos\left(\frac{|v|}{2}\right), v \frac{\sin(|v|/2)}{|v|} \right)$$
- matrices:

$$Rot(v) = \cos(|v|) I + \frac{\sin(|v|)}{|v|} [v]_\times + \frac{(1-\cos(|v|))}{|v|^2} v v^T$$

$$q [+] v = q \, Rot(v), \qquad q_2 [-] q_1 = aRot\left(q_1^{-1} q_2\right)$$

"Axioms"

(I): s[+]_ must be local diffeomorphism for all s

(II): [+] must be locally a "linear approximation"

(III): [-] must be the inverse of [+]

$$[+]: S \times R^n \to S$$

$$(II) \qquad s[+](v_1 + v_2) \approx (s[+]v_1)[+]v_2$$

$$[-]: S \times S \to R^n$$

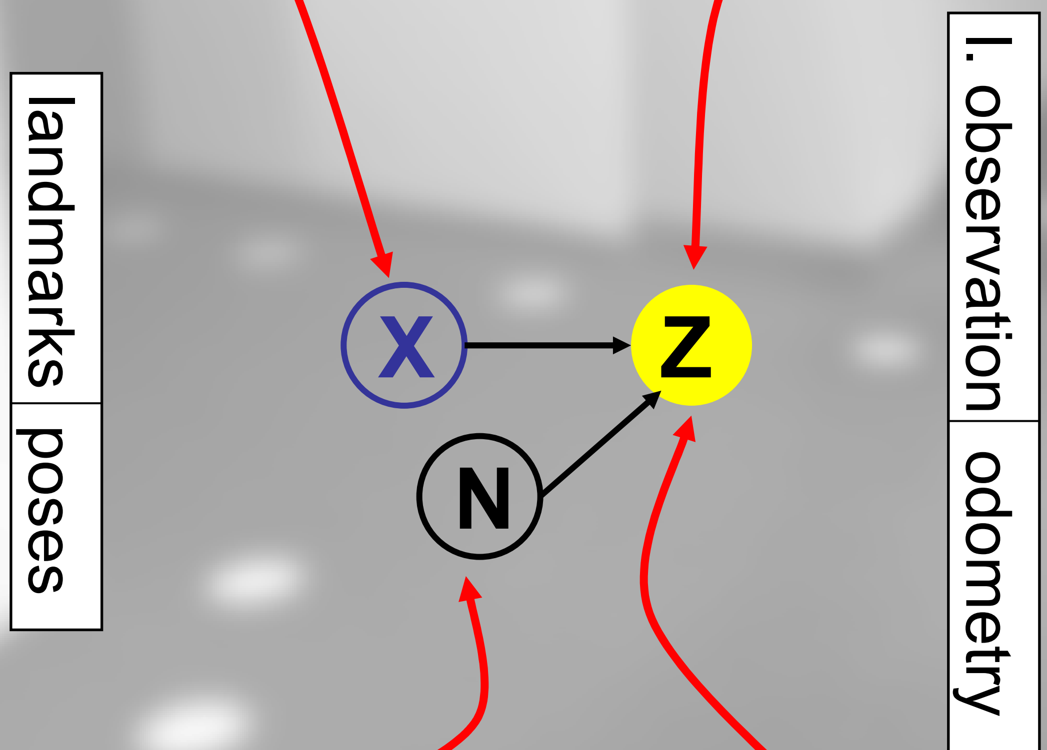
$$(III) \qquad s_1[+](s_2[-]s_1) = s_2$$

Manifolds

equipped with [+]

state

observations



Gaussian noise ~N(0,Q)

Z=f(X) [+] N

Manifolds

$$p(X = x \mid Z = z)$$

$$= \frac{p(Z = z \mid X = x)p(X = x)}{p(Z = z)}$$

$$\propto p(Z = z \mid X = x)p(X = x)$$

$$\propto p(Z = z \mid X = x)$$

$$= p(N = z[-]f(X) \mid X = x)$$

$$= p(N = z[-]f(x))$$

$$\propto \exp\left(-\tfrac{1}{2}(z[-]f(x))^T Q^{-1}(z[-]f(x))\right)$$

Manifolds

$$\hat{x} = \arg\max_x p(X = x \mid Z = z)$$

$$= \arg\min_x \left(\tfrac{1}{2}\left(z[-]f(x)\right)^T Q^{-1}\left(z[-]f(x)\right)\right)$$

$$= \arg\min_\delta \left(\tfrac{1}{2}\left(z[-]f(\breve{x}[+]\delta)\right)^T Q^{-1}\left(z[-]f(\breve{x}[+]\delta)\right)\right)$$

$$\Rightarrow 0 = -\left(z[-]f\left(\breve{x}[+]\hat{\delta}\right)\right)^T Q^{-1} \frac{d\left([z]-f(\breve{x}[+]\delta)\right)}{d\delta_i}\left(\hat{\delta}\right) \qquad \forall i$$

$$0 = -\left(z[-]f(\breve{x}[+]\delta)\right)^T Q^{-1} \frac{d\left([z]-f(\breve{x}[+]\delta)\right)}{d\delta}\left(\hat{\delta}\right)$$

$$0 = \frac{d\left([z]-f(\breve{x}[+]\delta)\right)^T}{d\delta}\left(\hat{\delta}\right) Q^{-1}\left(z[-]f\left(\breve{x}[+]\hat{\delta}\right)\right)$$

Manifolds

$$0 = \frac{df\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}\left(\hat{\delta}\right)^T Q^{-1}\left(z[-]f\left(\breve{x}[+]\hat{\delta}\right)\right)$$

$$z[-]f\left(\breve{x}[+]\delta\right) \approx z[-]f\left(\breve{x}\right) + \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)\delta$$

$$0 = \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z[-]f\left(\breve{x}\right) + \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)\delta\right)$$

$$= \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)\delta$$

$$+ \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z[-]f\left(\breve{x}\right)\right)$$

information matrix   information vector

Manifolds

$$\delta = -\left(\frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1} \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0) \right)^{-1}$$

$$\frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z[-]f\left(\breve{x}\right)\right)$$

$$\hat{x} = \breve{x}[+]-\left(\frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1} \frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0) \right)^{-1}$$

$$\frac{d\left(z[-]f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z[-]f\left(\breve{x}\right)\right)$$

Comparison

- vectorspace LS

$$\hat{x} = \breve{x} + \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1} \left(z - f(\breve{x}) \right)$$

- manifolds LS

$$\hat{x} = \breve{x} [+] - \left(\frac{d\left(z[-]f(\breve{x}[+]\delta)\right)}{d\delta}(0)^T Q^{-1} \frac{d\left(z[-]f(\breve{x}[+]\delta)\right)}{d\delta}(0) \right)^{-1}$$

$$\frac{d\left(z[-]f(\breve{x}[+]\delta)\right)}{d\delta}(0)^T Q^{-1} \left(z[-]f(\breve{x}) \right)$$

- viewing it as a mapping of perturbations in x to perturbations in f(x)

Comparison simplified (Z is vectorspace)

- vectorspace LS

$$\hat{x} = \breve{x} + \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}\left(z - f(\breve{x})\right)$$

- manifolds LS

$$\hat{x} = \breve{x} [+] \left(\frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1} \frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0) \right)^{-1}$$

$$\frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z - f(\breve{x})\right)$$

Manifolds

Question to the audience: Where is the difference to treating $x_0[+]\delta$ as a parameterization for x and applying VS-LS to δ?

- vectorspace LS

$$\hat{x} = \breve{x} + \left(\frac{df}{dx}(\breve{x})^T Q^{-1} \frac{df}{dx}(\breve{x}) \right)^{-1} \frac{df}{dx}(\breve{x})^T Q^{-1}(z - f(\breve{x}))$$

- manifolds LS

$$\hat{x} = \breve{x}[+]\left(\frac{d(f(\breve{x}[+]\delta))}{d\delta}(0)^T Q^{-1} \frac{d(f(\breve{x}[+]\delta))}{d\delta}(0) \right)^{-1} \frac{d(f(\breve{x}[+]\delta))}{d\delta}(0)^T Q^{-1}(z - f(\breve{x}))$$

Question to the audience:

- vectorspace LS $\hat{\delta} = \breve{\delta} + \left(\frac{df\left(x_0[+]\delta\right)}{\delta}\left(\breve{\delta}\right)^T Q^{-1} \frac{df\left(x_0[+]\delta\right)}{\delta}\left(\breve{\delta}\right) \right)^{-1}$

$$\frac{df\left(x_0[+]\delta\right)}{\delta}\left(\breve{\delta}\right)^T Q^{-1}\left(z - f\left(x_0[+]\breve{\delta}\right)\right)$$

- manifolds LS $\hat{x} = \breve{x}[+]\left(\frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1} \frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0) \right)^{-1}$

$$\frac{d\left(f\left(\breve{x}[+]\delta\right)\right)}{d\delta}(0)^T Q^{-1}\left(z - f\left(\breve{x}\right)\right)$$

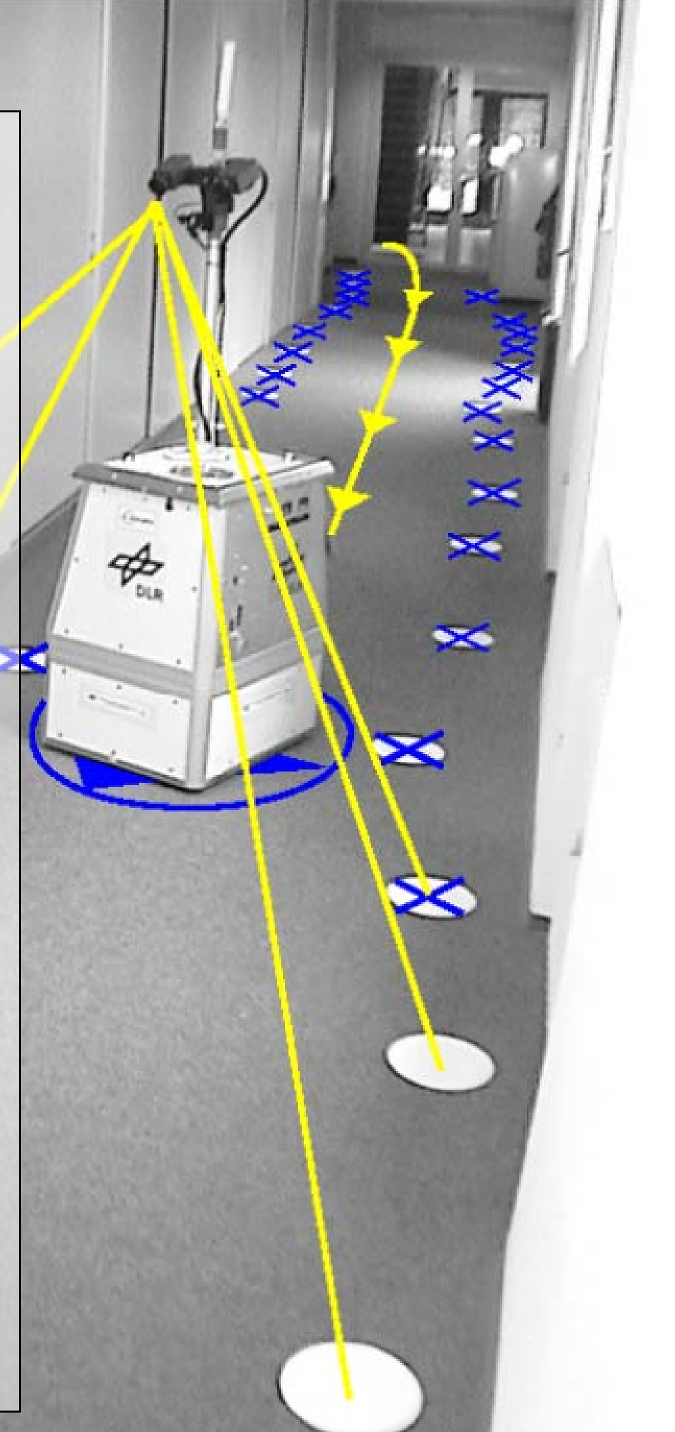
- VS-LS would accumulate in δ and might run into singularities, M-LS accumulates in x and only parameterizes each small step

Manifolds

- how to get the Jacobian?
- numerically by evaluating $z[-]f(x[+]\delta)$ for small unit vectors $\delta=\pm\varepsilon I_i$
- or by evaluating on σ points, such as UKF
- whole UKF can be directly used on manifolds by replacing – with [-] and + with [+]

Manifolds

- 3-D orientations, 3-D directions, …, pose parameterization problems
- encapsulate the structure of manifolds by defining perturbation operators [+], [-]
- mostly existing formulas work by replacing + with [+] and – with [-] with common sense applied
- iterations are accumulated in the state



Summary

- least square (LS) is the gold-standard approach
- linearization problems come mainly from inconsistent linearization points
- LS can be made efficient by exploiting sparsity
- singularity problems of rotations, directions can be encapsulated by a perturbation operator [+],[-]