CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT

Machine Vision

Lecture 9: Parameter estimation

What is parameter estimation?

- Last week we looked at how to determine the fundamental matrix $\boldsymbol{F}$ from image point correspondences $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x_i'}$
- To achieve this we used the fact that we know that $\boldsymbol{x_i'}^T \boldsymbol{F} \boldsymbol{x_i} = 0$, as well as that $vec[\boldsymbol{F}]^T vec[\boldsymbol{F}] = 1$ and $\det \boldsymbol{F} = 0$
- In more generic terms the inputs to this problem were:
  - A vector of observations $\boldsymbol{l}$ (e.g. the image points $\boldsymbol{x_i}, \boldsymbol{x_i}'$)
  - A vector of unknown parameters $\boldsymbol{p}$ (e.g. the elements of $\boldsymbol{F}$)
  - The model function $g[\boldsymbol{p}, \boldsymbol{l}] = \boldsymbol{0}$ (e.g. the relation $\boldsymbol{x_i'}^T \boldsymbol{F} \boldsymbol{x_i} = 0$ for all $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x_i'}$)
  - The intrinsic constraints $h[\boldsymbol{p}] = \boldsymbol{0}$ (e.g. $\det[\boldsymbol{F}] = 1$ and $\boldsymbol{f}^T \boldsymbol{f} = 1$)

What is parameter estimation?

- In more generic terms the inputs to this problem were:
  - A vector of observations $\boldsymbol{l}$ (e.g. the image points $\boldsymbol{x_i}, {\boldsymbol{x_i}}'$)
  - A vector of unknown parameters $\boldsymbol{p}$ (e.g. the elements of $\boldsymbol{F}$)
  - The model function $g[\boldsymbol{p}, \boldsymbol{l}] = \boldsymbol{0}$ (e.g. the relation ${\boldsymbol{x}'_{\boldsymbol{i}}}^T \boldsymbol{F} \boldsymbol{x_i} = 0$ for all $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x}'_{\boldsymbol{i}}$)
  - The intrinsic constraints $h[\boldsymbol{p}] = \boldsymbol{0}$ (e.g. $\det[\boldsymbol{F}] = 1$ and $\boldsymbol{f}^T \boldsymbol{f} = 1$)

- In addition to this there are two more things to consider
  - The observations are inaccurate, which means they need to be considered to be drawn from a probability distribution $\mathbf{l} \sim \mathcal{N}\left(\bar{\boldsymbol{l}}, \boldsymbol{C_{ll}}\right)$ with unknown mean $\bar{\boldsymbol{l}}$ and known covariance matrix $\boldsymbol{C_{ll}}$
  - The model function might be wrong for a subset of observations (e.g. some of the correspondence $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x}'_{\boldsymbol{i}}$ have been misidentified)

What is parameter estimation?

- In summary, the inputs to a parameter estimation procedure are
  - Observations $\boldsymbol{l}$
  - Observation covariances $\boldsymbol{C}_{\boldsymbol{ll}}$
  - Model function $g[\boldsymbol{p}, \boldsymbol{l}] = \mathbf{0}$
  - Intrinsic constraints $h[\boldsymbol{p}] = \mathbf{0}$

- The outputs are a subset of
  - Parameter estimates $\boldsymbol{p}$
  - Parameter covariances $\boldsymbol{C}_{\boldsymbol{pp}}$

- We need to consider that a subset of observations might be wrong

Direct minimal solutions

- We also looked at methods to directly compute parameters from minimal number of observations

- Examples include
  - Calculating a homography from 4 point correspondences
  $$\left(\boldsymbol{x}_{\boldsymbol{i}}^{T} \otimes \boldsymbol{S}[\boldsymbol{x}_{\boldsymbol{i}}{}']\right)vec[\boldsymbol{H}] = \boldsymbol{0}$$
  - Calculating a projection matrix from 6 point correspondences
  $$(\boldsymbol{X}^{T} \otimes \boldsymbol{S}[\boldsymbol{x}'])vec[\boldsymbol{P}] = \boldsymbol{0}$$
  - Calculating a fundamental matrix from 7 point correspondences
  $$\left(\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}} \otimes \boldsymbol{x}_{\boldsymbol{i}}'^{\boldsymbol{T}}\ \right)vec[\boldsymbol{F}] = 0$$

Direct minimal solutions

- We also looked at methods to directly compute parameters from minimal number of observations

- In more generic terms we split the observation vector into sub-vectors (e.g. a single correspondence $\boldsymbol{l_i} = (\boldsymbol{x_i}, \boldsymbol{x_i'})$)

$$\boldsymbol{l} = (\boldsymbol{l_1} \quad \cdots \quad \boldsymbol{l_n})^T$$

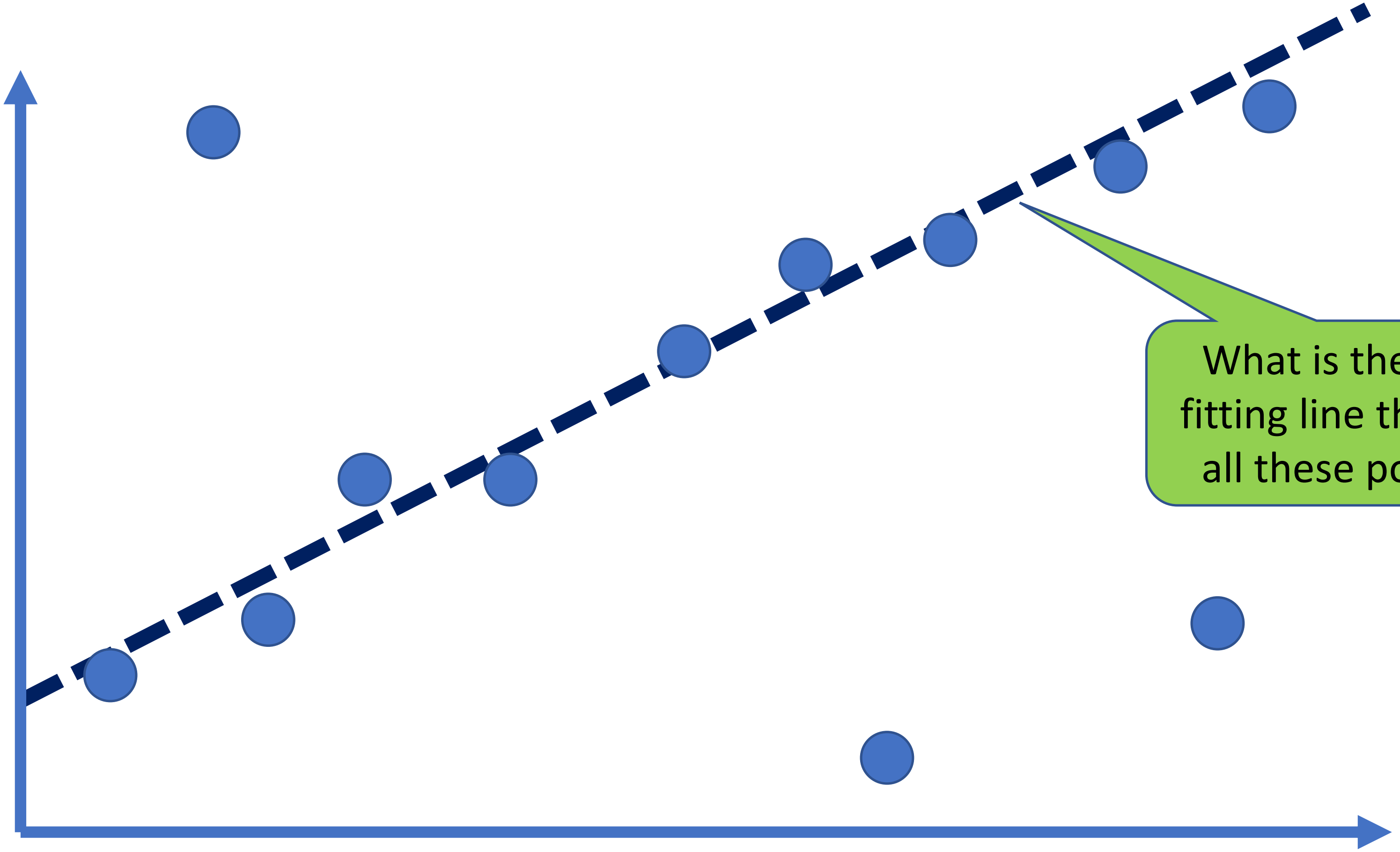
- We then choose a minimal subset $\{i_1, \dots, i_m\}$ and select the corresponding observation vector

$$\boldsymbol{l'} = (\boldsymbol{l_{i_1}} \quad \cdots \quad \boldsymbol{l_{i_m}})^{\boldsymbol{T}}$$

- We then apply the algorithm $\mathcal{A}$ (e.g. DLT) to directly calculate the parameter vector from the observations vector

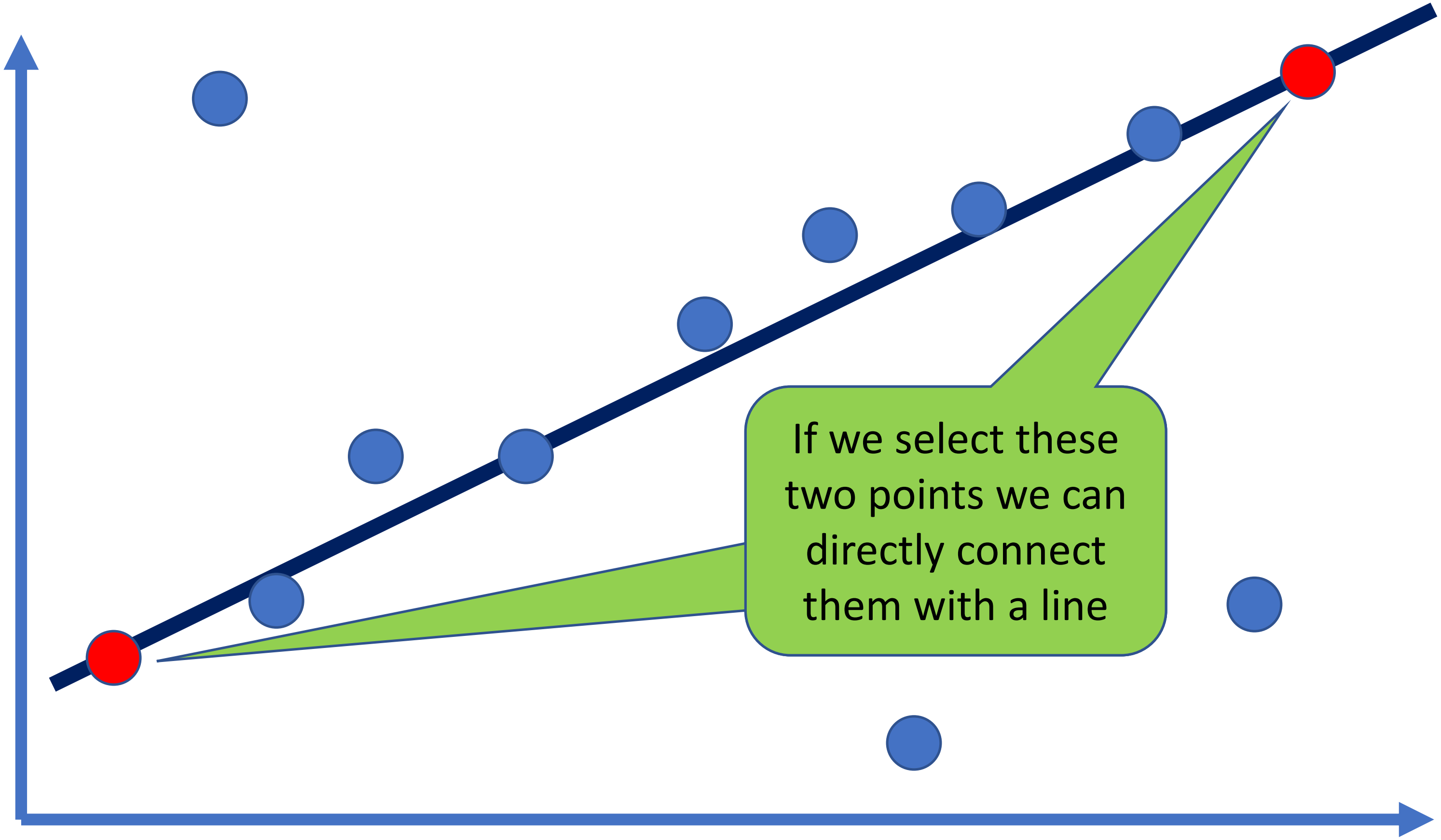
$$\widetilde{\boldsymbol{p}} = \mathcal{A}[\boldsymbol{l'}]$$

Robust estimation



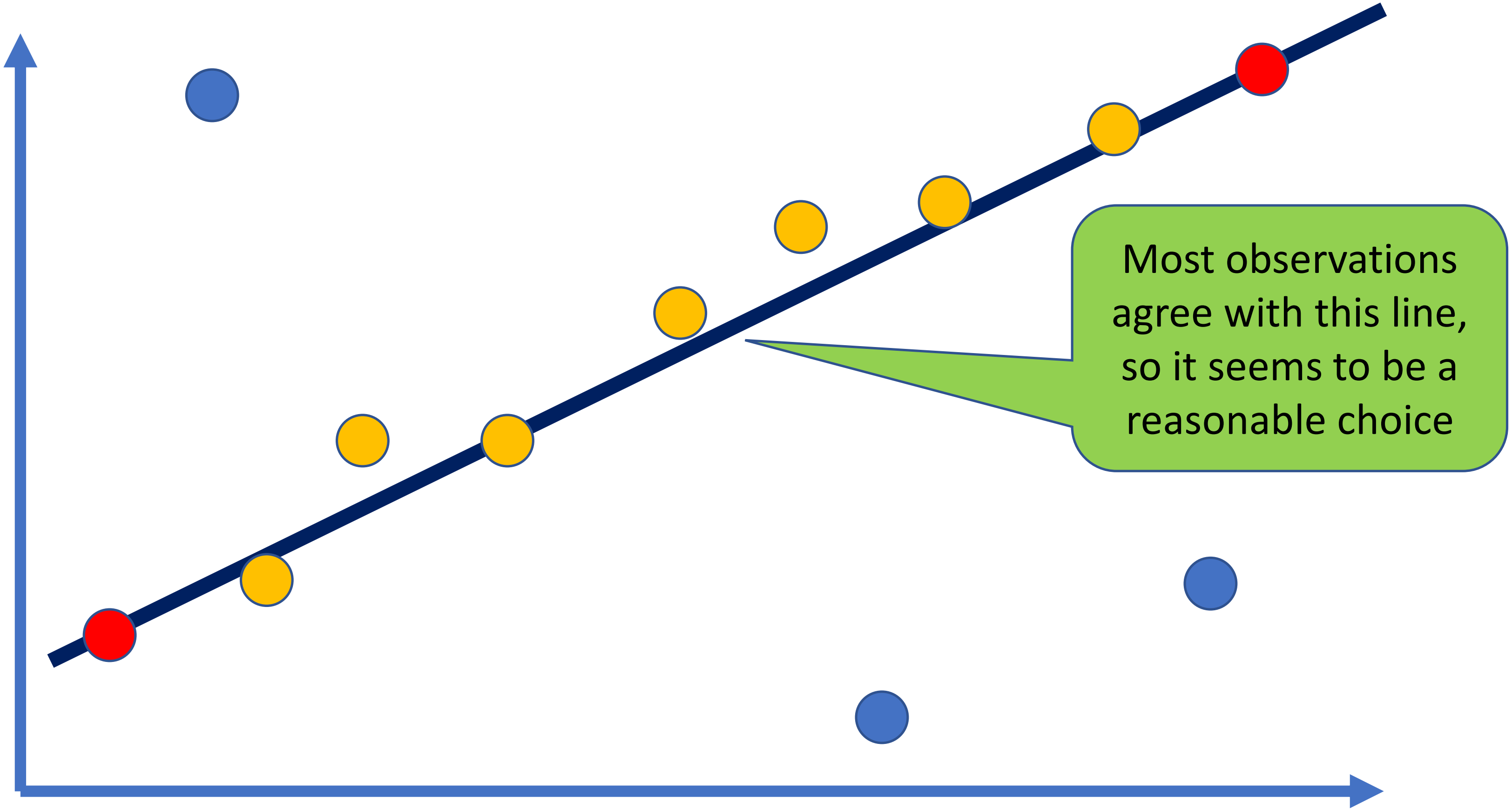
What is the best fitting line through all these points?

Robust estimation



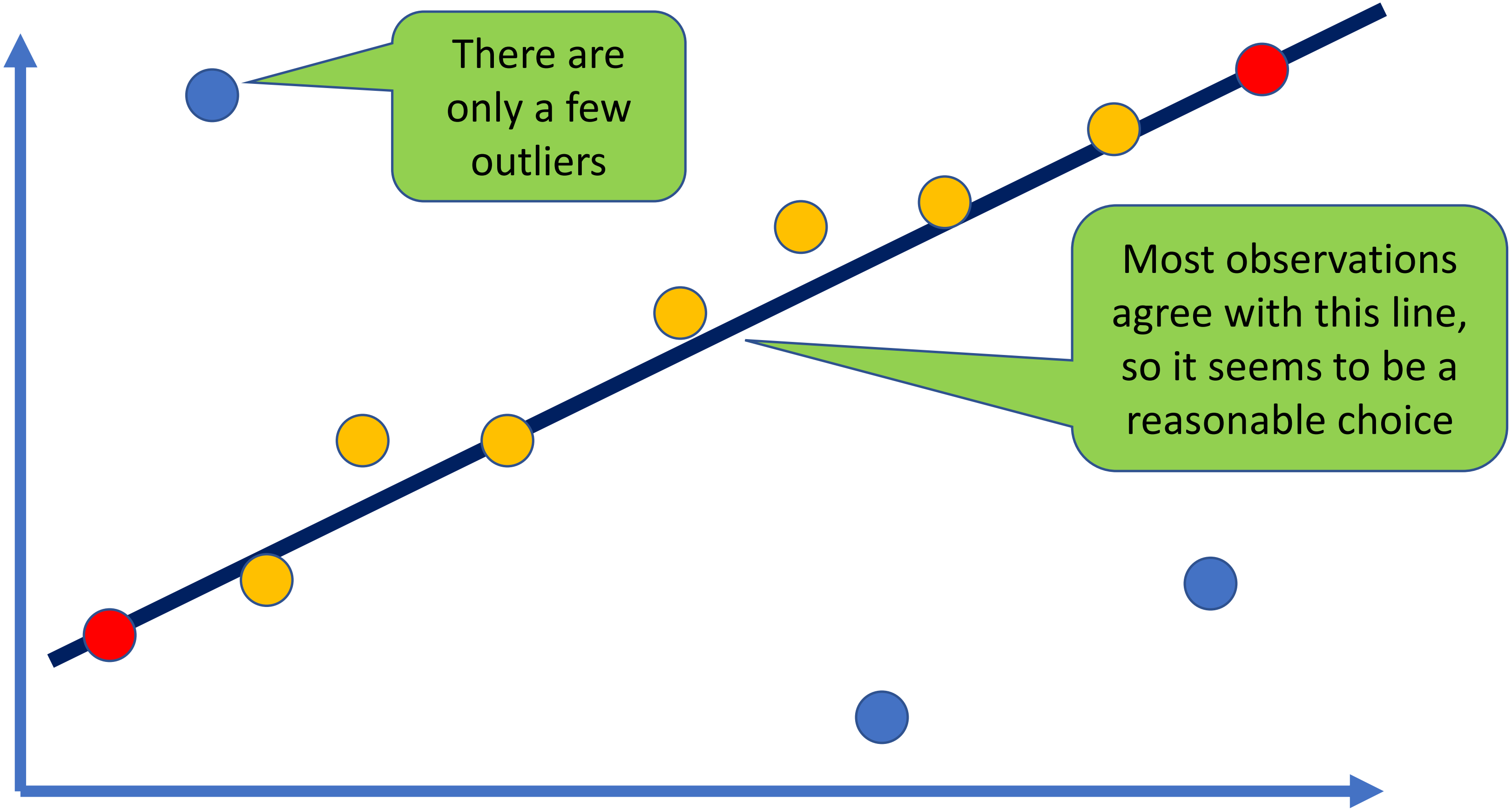
If we select these two points we can directly connect them with a line

Robust estimation



Most observations agree with this line, so it seems to be a reasonable choice

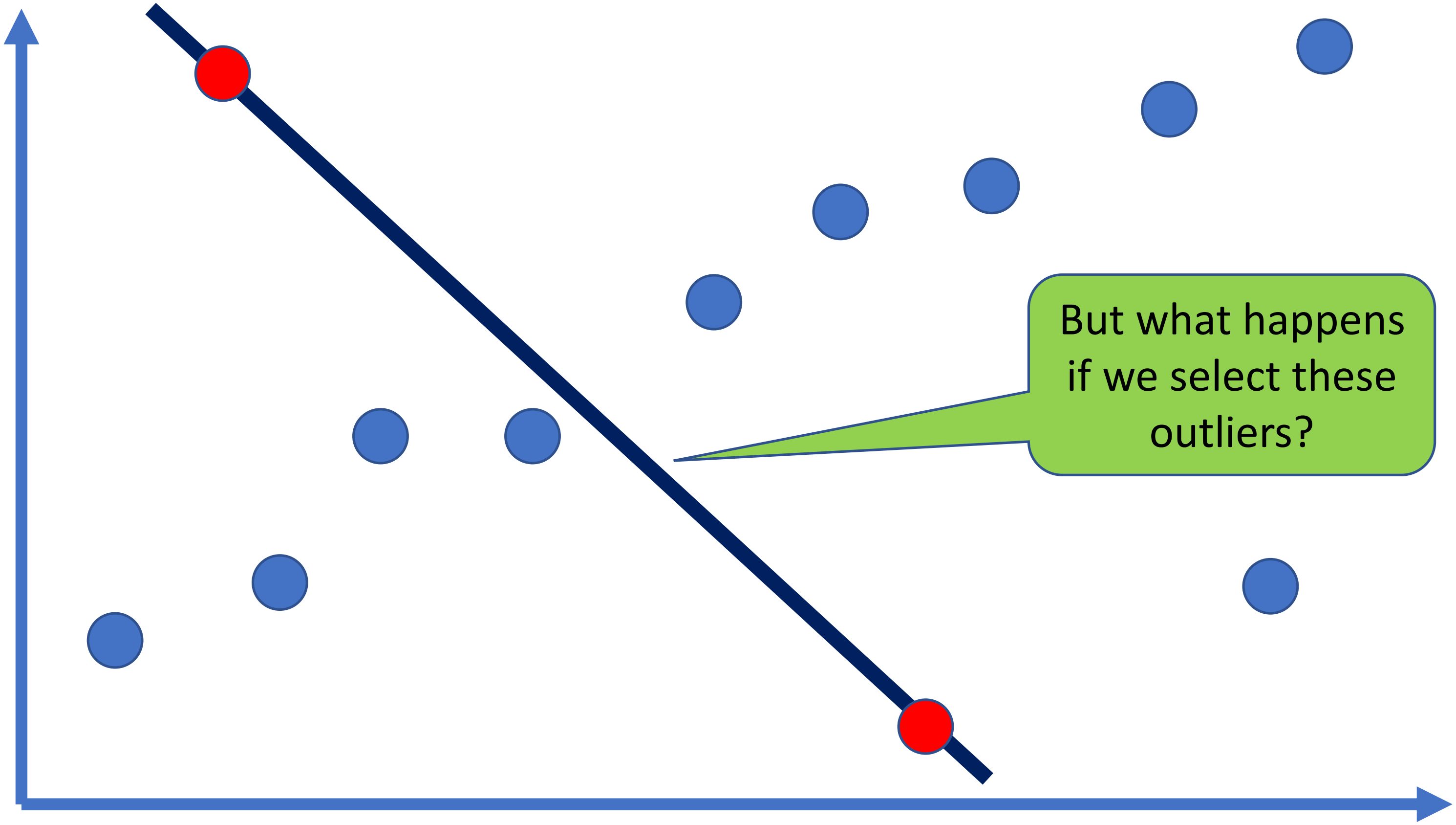
Robust estimation



There are only a few outliers

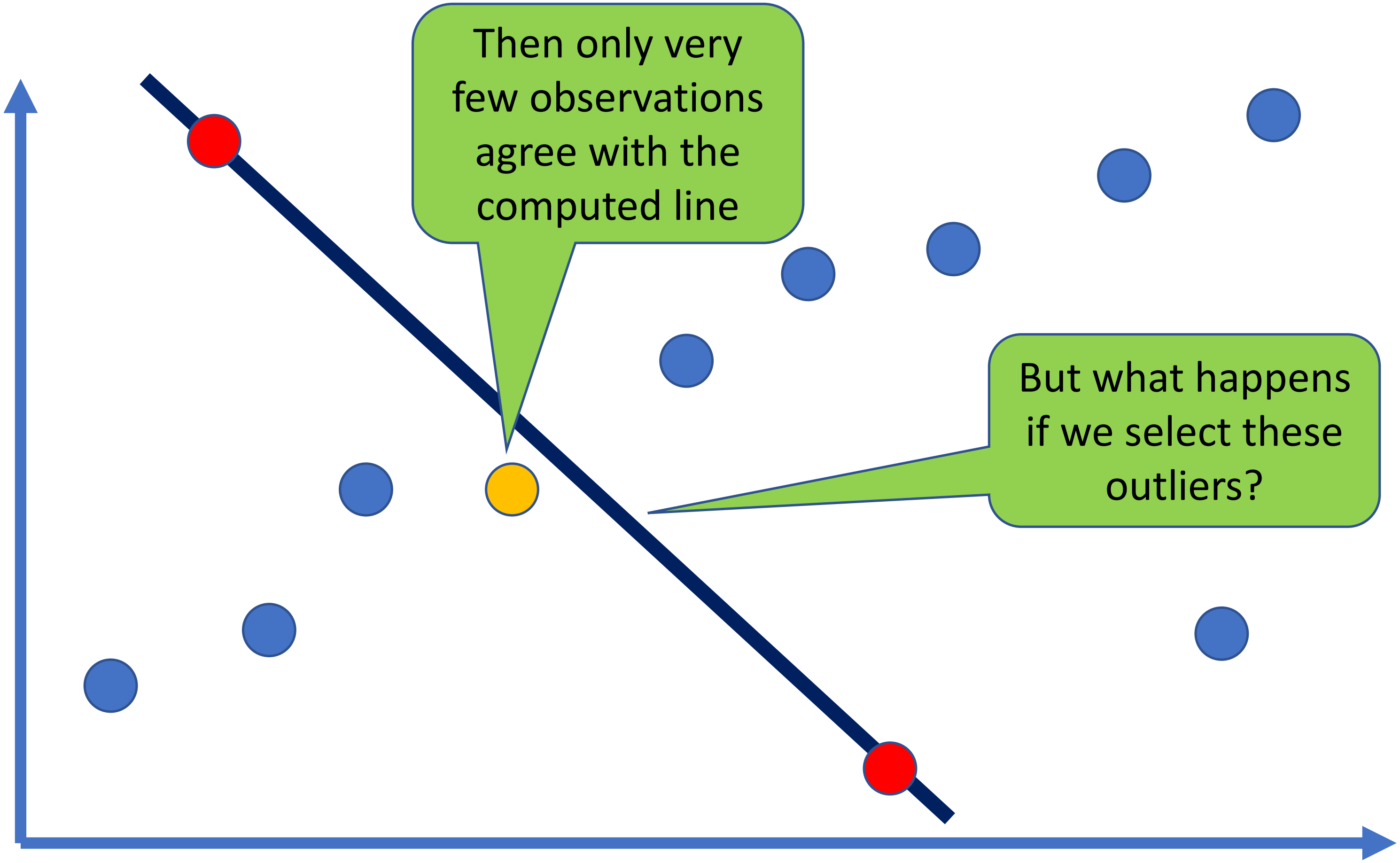
Most observations agree with this line, so it seems to be a reasonable choice

Robust estimation



But what happens if we select these outliers?

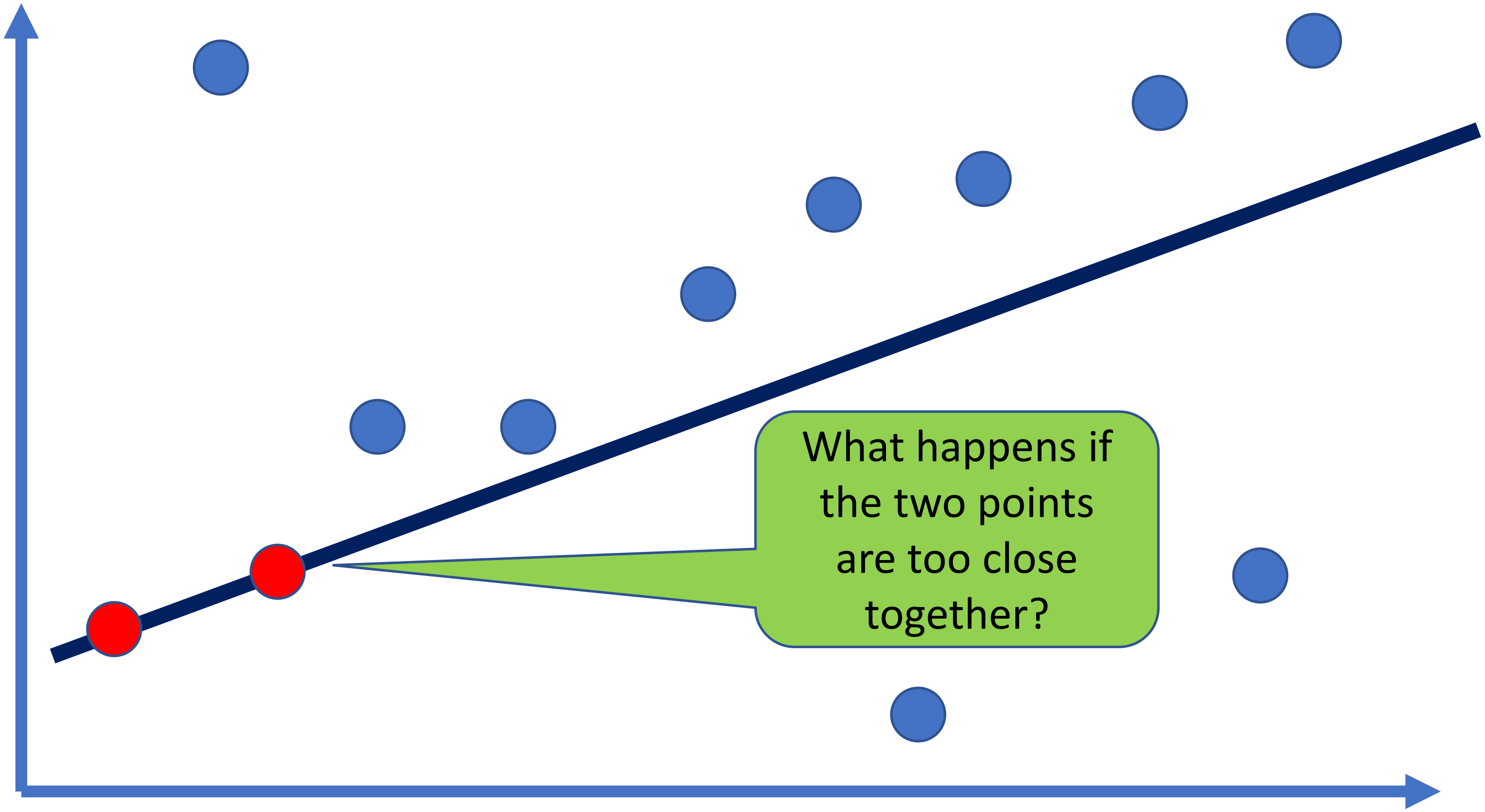
Robust estimation



Then only very few observations agree with the computed line

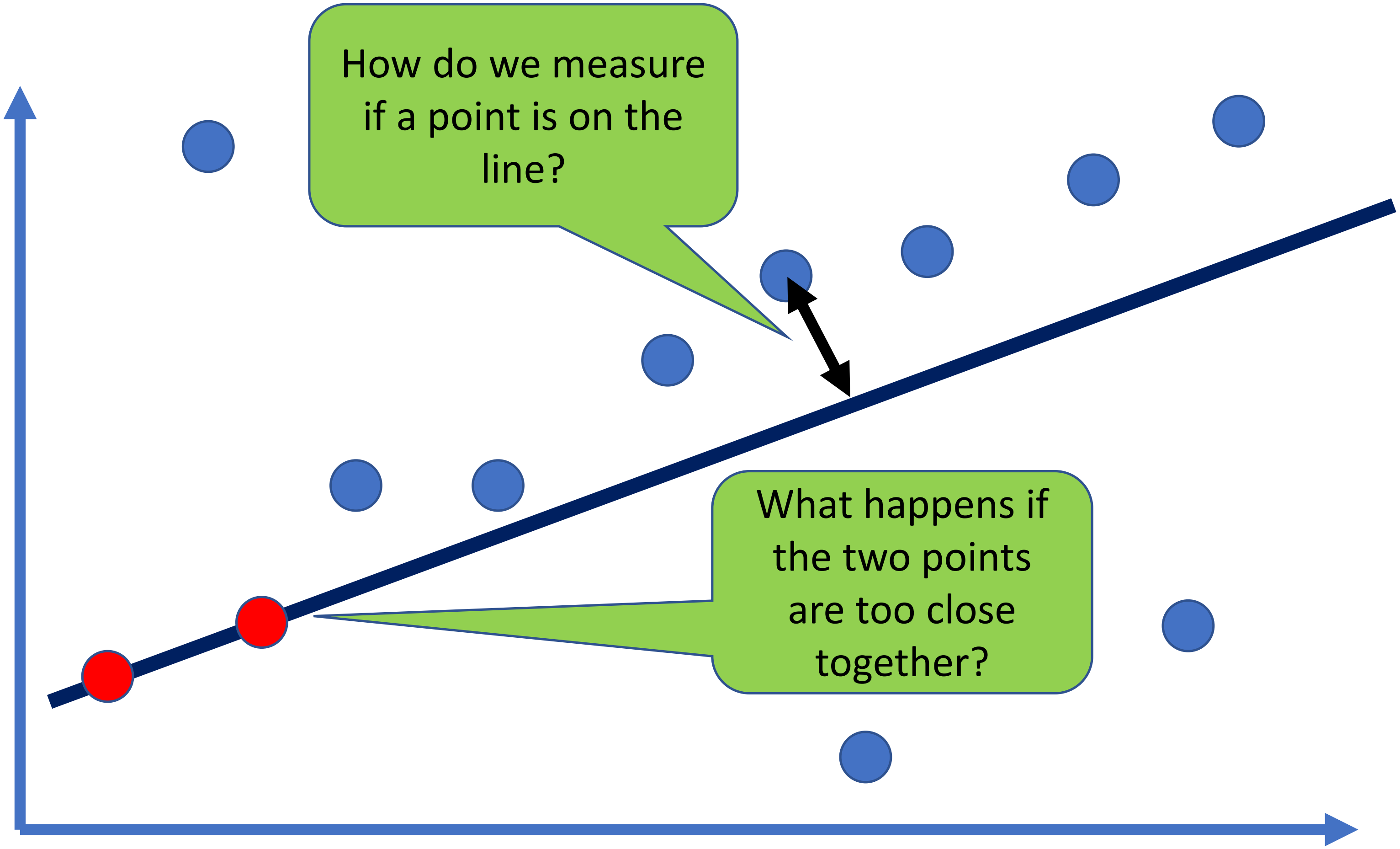
But what happens if we select these outliers?

Robust estimation



What happens if the two points are too close together?

Robust estimation



How do we measure if a point is on the line?

What happens if the two points are too close together?

Robust estimation

- The basic idea behind the **RANdom Sampling Consesus (RANSAC)** algorithm is to

  - Select a minimal subset of observations at random
  $$\boldsymbol{l}' = (\boldsymbol{l}_{\boldsymbol{i}_{\boldsymbol{1}}} \quad \cdots \quad \boldsymbol{l}_{\boldsymbol{i}_{\boldsymbol{m}}})^T$$
  - Directly calculate the parameter vector $\widetilde{\boldsymbol{p}} = \mathcal{A}[\boldsymbol{l}']$
  - Use the remaining observations $\boldsymbol{l}'' = (\boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{1}}} \quad \cdots \quad \boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{n-m}}})^T$
  - Then use the model function to determine if they also agree with the model $\widetilde{\boldsymbol{p}}$, i.e. test for all $\boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}} \in \boldsymbol{l}''$ if $g[\widetilde{\boldsymbol{p}}, \boldsymbol{l}_{j_k}]$ is close enough to zero
  - Iterate this process and choose the model $\widetilde{\boldsymbol{p}}$ with maximum support consensus

Error propagation

- So how do we determine if observations agree with a model?

- Remember, all our observations are assumed to be drawn from a probability distribution with mean $\boldsymbol{l}$ and covariance $\boldsymbol{C_{ll}}$
- When calculating a vector $\boldsymbol{y} = f[\boldsymbol{l'}]$ from observations, the uncertainty of the observation can be propagated according to

$$\boldsymbol{C_{yy}} = \left(\frac{\partial f}{\partial \boldsymbol{l'}}\right) \boldsymbol{C_{l'l'}} \left(\frac{\partial f}{\partial \boldsymbol{l'}}\right)^T$$

- In particular, if $\mathbf{y} = \boldsymbol{Al'} + \boldsymbol{a}$ is linear, then $\boldsymbol{C_{yy}} = \boldsymbol{AC_{l'l'}A^T}$

Uncertainty of homogeneous entities

- The uncertainty is typically defined in Euclidean space, where it can be characterised by a covariance matrix
- For example a 2d point has a 2 × 2 covariance matrix

$$\boldsymbol{C}_{xx} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

- When we create a homogeneous point we simply put a “1” at the end of the vector
- This 1 does not have any uncertainty, so what is the covariance matrix of the homogeneous 3 vector?
- Under most circumstances it is possible to use a singular covariance matrix where the homogeneous component has zero variance, i.e.

$$\boldsymbol{C}_{xx} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- This can also be applied to 3d points, and the rules of error propagation can be used to define covariance matrices of other entities
- Error propagation must also be applied when normalising homogeneous enities

Hypothesis testing

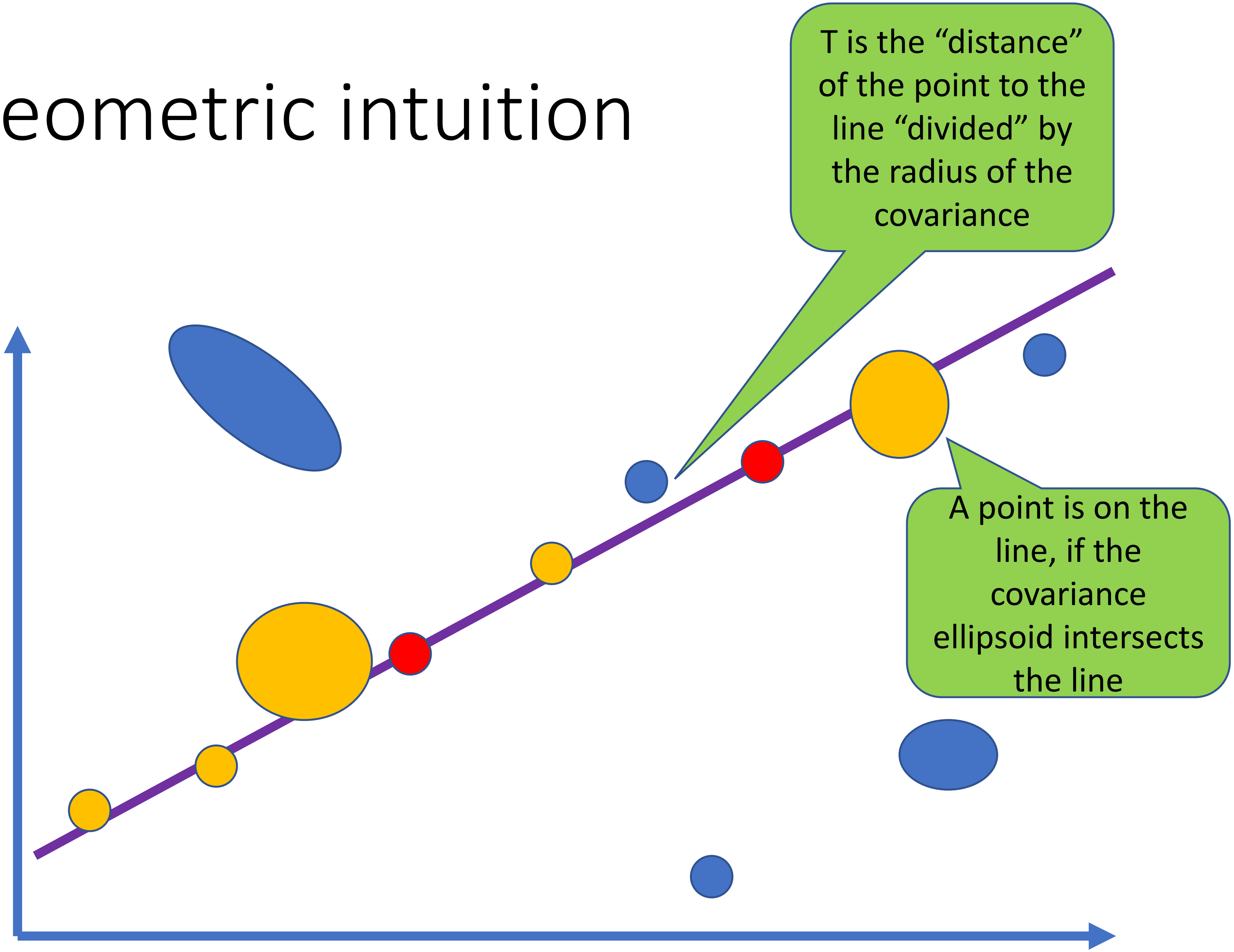
- So how do we determine if observations agree with a model?

- First, we calculate the value of the model function $\mathbf{y} = g\left[\widetilde{\boldsymbol{p}}, \boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}}\right]$ that we want to test
- Then we calculate the Jacobian $\boldsymbol{A} = \frac{\partial g\left[\widetilde{\boldsymbol{p}}, \boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}}\right]}{\partial \boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}}}$ and propagate the uncertainty $\boldsymbol{C}_{\boldsymbol{yy}} = \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}}\boldsymbol{l}_{\boldsymbol{j}_{\boldsymbol{k}}}}\boldsymbol{A}^{\boldsymbol{T}}$
- Normalising the squared $\boldsymbol{y}$ with its uncertainty yields the χ^2 test statistic

$$T = \boldsymbol{y}^T \boldsymbol{C}_{\boldsymbol{yy}}^{-\mathbf{1}} \boldsymbol{y}$$

- This can be threshold to determine if $\boldsymbol{y}$ is “close enough” to zero, i.e. if it is an inlier or an outlier

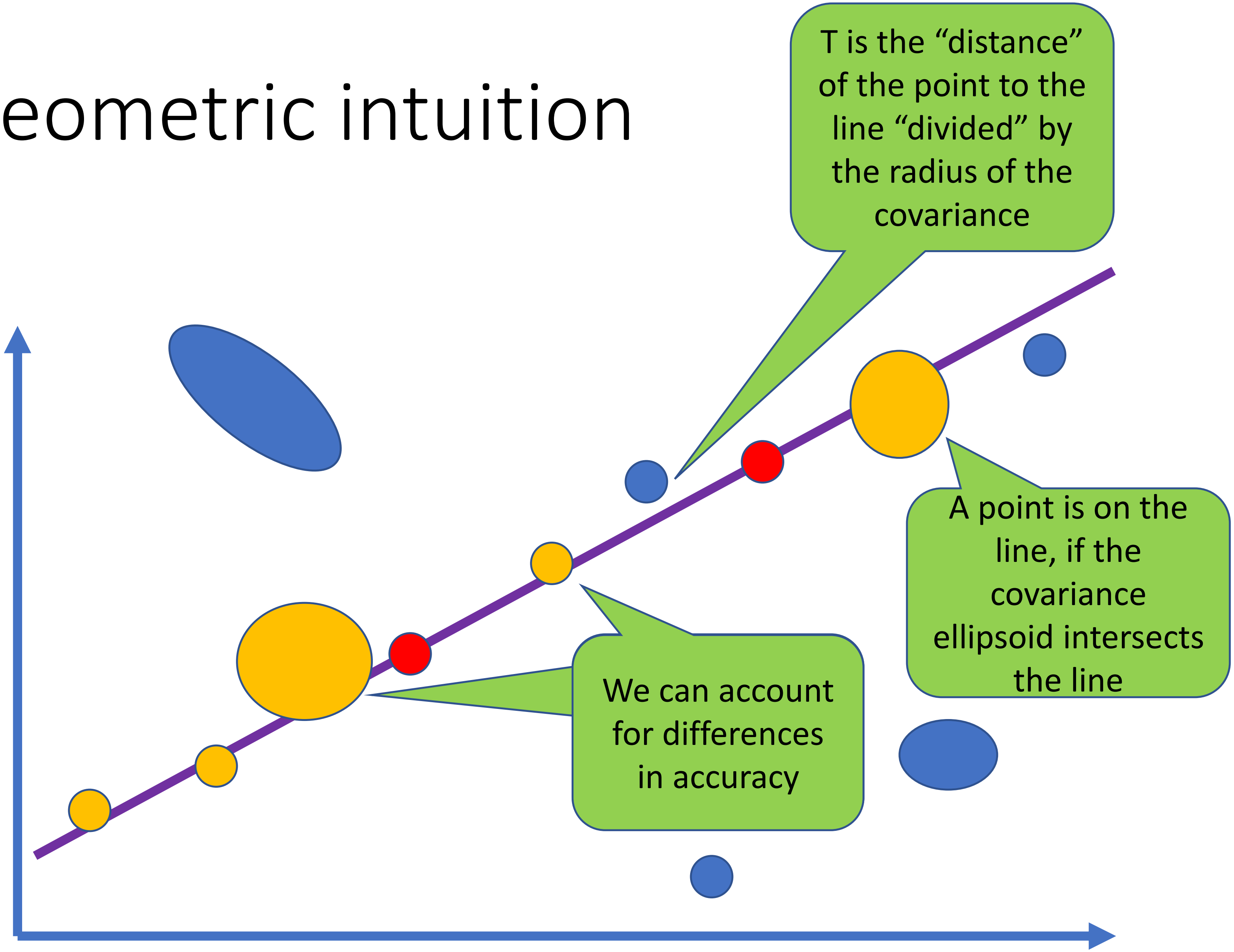
Geometric intuition



T is the “distance” of the point to the line “divided” by the radius of the covariance

A point is on the line, if the covariance ellipsoid intersects the line

Geometric intuition

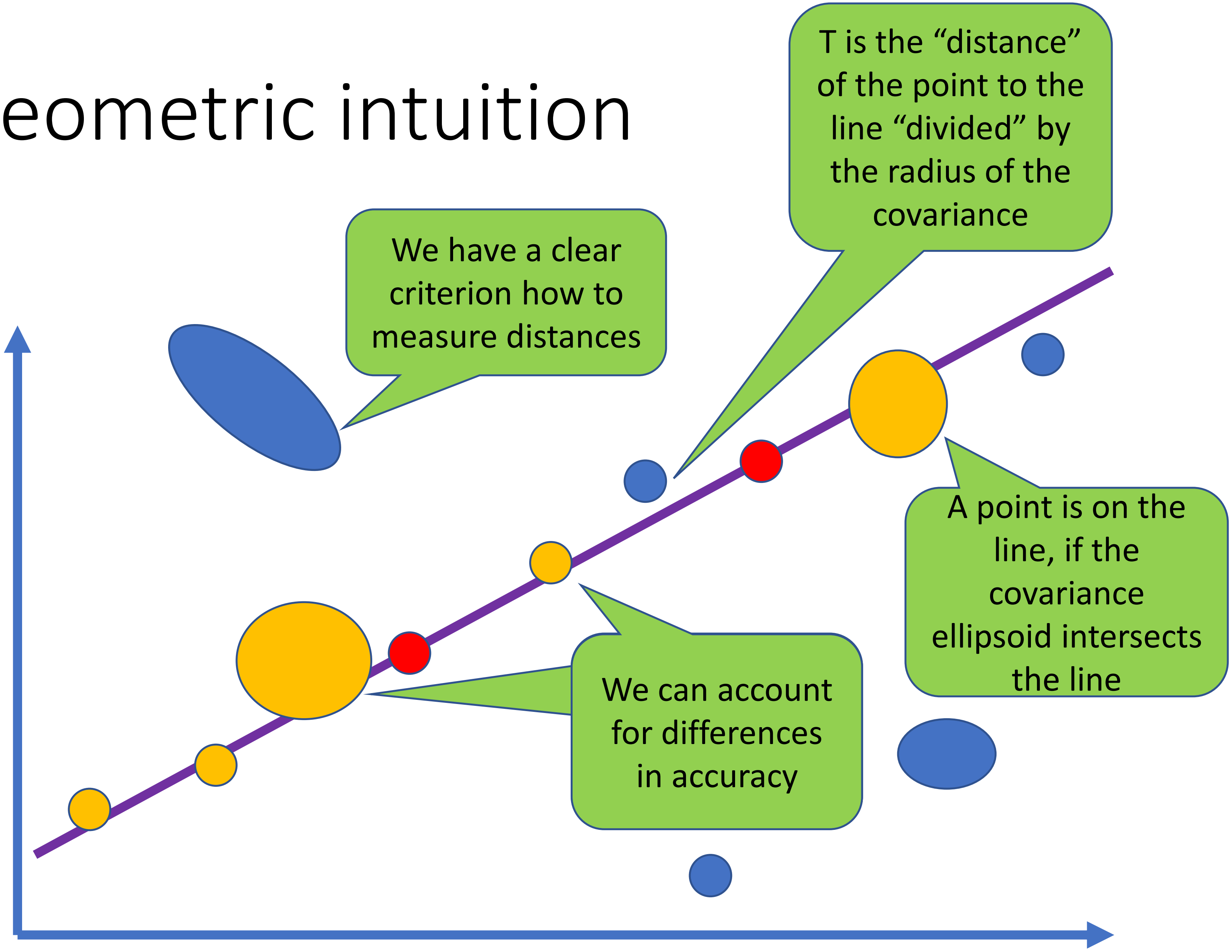


T is the “distance” of the point to the line “divided” by the radius of the covariance

A point is on the line, if the covariance ellipsoid intersects the line

We can account for differences in accuracy

Geometric intuition



T is the "distance" of the point to the line "divided" by the radius of the covariance

We have a clear criterion how to measure distances

A point is on the line, if the covariance ellipsoid intersects the line

We can account for differences in accuracy

Example: Homographies

- We have been calculating homographies from 4 point correspondences using the DLT algorithm
- The model function we used is $\boldsymbol{g}[\boldsymbol{h}, \boldsymbol{x}_i, \boldsymbol{x}'_i] = \boldsymbol{x}'_i \times \boldsymbol{H}\boldsymbol{x}_i = \mathbf{0}$
- It is bi-linear (as are most of the expressions we have been looking at), which means we can re-write it any of the following ways

$$\underbrace{\left(\boldsymbol{x}_i^T \otimes \boldsymbol{S}[\boldsymbol{x}_i{}']\right)}_{\boldsymbol{A}_i} \boldsymbol{h} = \boldsymbol{A}_i\boldsymbol{h} = \mathbf{0}$$

$$\underbrace{\boldsymbol{S}[\boldsymbol{x}'_i]\boldsymbol{H}}_{\boldsymbol{B}_i} \boldsymbol{x}_i = \boldsymbol{B}_i\boldsymbol{x}_i = \mathbf{0}$$

$$\underbrace{-\boldsymbol{S}[\boldsymbol{H}\boldsymbol{x}_i]}_{\boldsymbol{B}'_i}\boldsymbol{x}'_i = \boldsymbol{B}'_i\boldsymbol{x}'_i = \mathbf{0}$$

- Remember, we are only using two rows of each equation

Example: Homographies

- First, we select four point correspondences $\boldsymbol{x}_i' \leftrightarrow \boldsymbol{x}_i$ for which we calculate the Jacobians $\boldsymbol{A}_i$ to obtain the homography $\boldsymbol{H}$ using the 4-point-DLT algorithm

- Now we go through all remaining point correspondence $\boldsymbol{x}_j' \leftrightarrow \boldsymbol{x}_j$ for which we can compute $\boldsymbol{B}_j$ and $\boldsymbol{B}_j'$ (note, that these depend on the $\boldsymbol{H}$ obtained in the first step)
- The model function evaluates now to (we can choose either $\boldsymbol{B}_j$ or $\boldsymbol{B}_j'$)

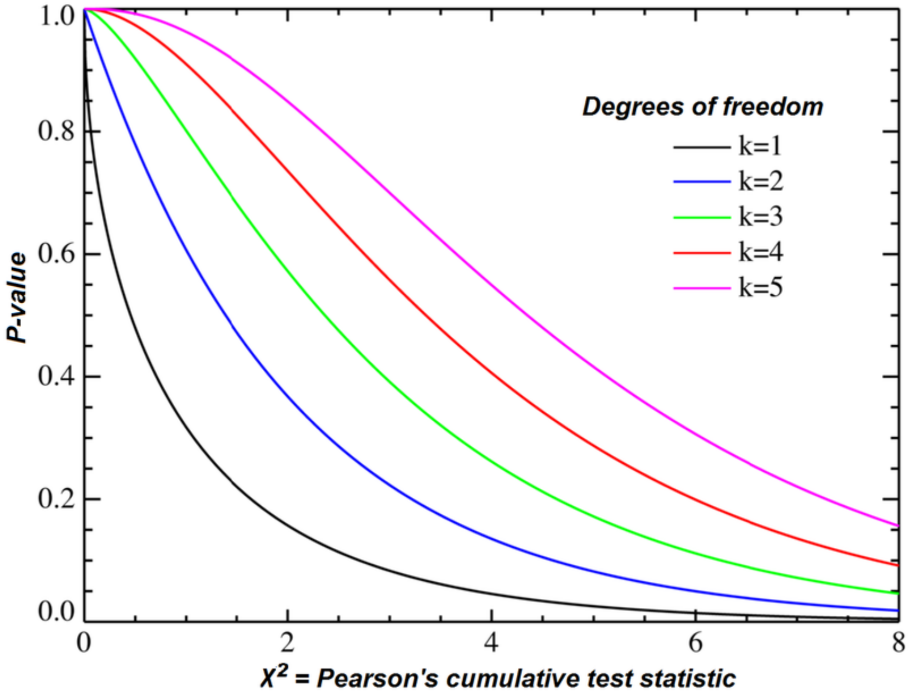
$$\boldsymbol{y}_j = \boldsymbol{B}_j \boldsymbol{x}_j = \boldsymbol{B}_j' \boldsymbol{x}_j'$$

- The covariance matrix propagates as

$$\boldsymbol{C}_{\boldsymbol{y}_j \boldsymbol{y}_j} = \boldsymbol{B}_j \boldsymbol{C}_{\boldsymbol{x}_j \boldsymbol{x}_j} \boldsymbol{B}_j^T + \boldsymbol{B}_j' \boldsymbol{C}_{\boldsymbol{x}_j' \boldsymbol{x}_j'} \boldsymbol{B'}_j^T$$

Example: Homographies

- Finally we calculate for each of the correspondences which we need to test the normalised test statistic
$$T_j = \boldsymbol{y}_j^T \boldsymbol{C}_{\boldsymbol{y}_j \boldsymbol{y}_j}^{-1} \boldsymbol{y}_j$$
- The threshold can be derived from the χ^2 distribution given the degrees of freedom (2 in case of 2d point equivalence) and a P-value (e.g. 99%)
- In this case the thresholds are
  - Inliers: $T \leq 9.21$
  - Outliers: $T > 9.21$
- We have not accounted for the uncertainty of $\boldsymbol{H}$ in this approach, so we might choose the thresholds a bit larger



RANSAC

- Every iteration of the RANSAC algorithm calculates
  - The split of observations into inliers and outliers
  - An approximate parameter vector $\tilde{\boldsymbol{p}}$

- The iteration with the smallest number of outliers “wins”
- We retain the parameter vector $\tilde{\boldsymbol{p}}$ of that iteration and remove all outliers for further processing

- From that point onward, we assume the model function holds for all observations

RANSAC

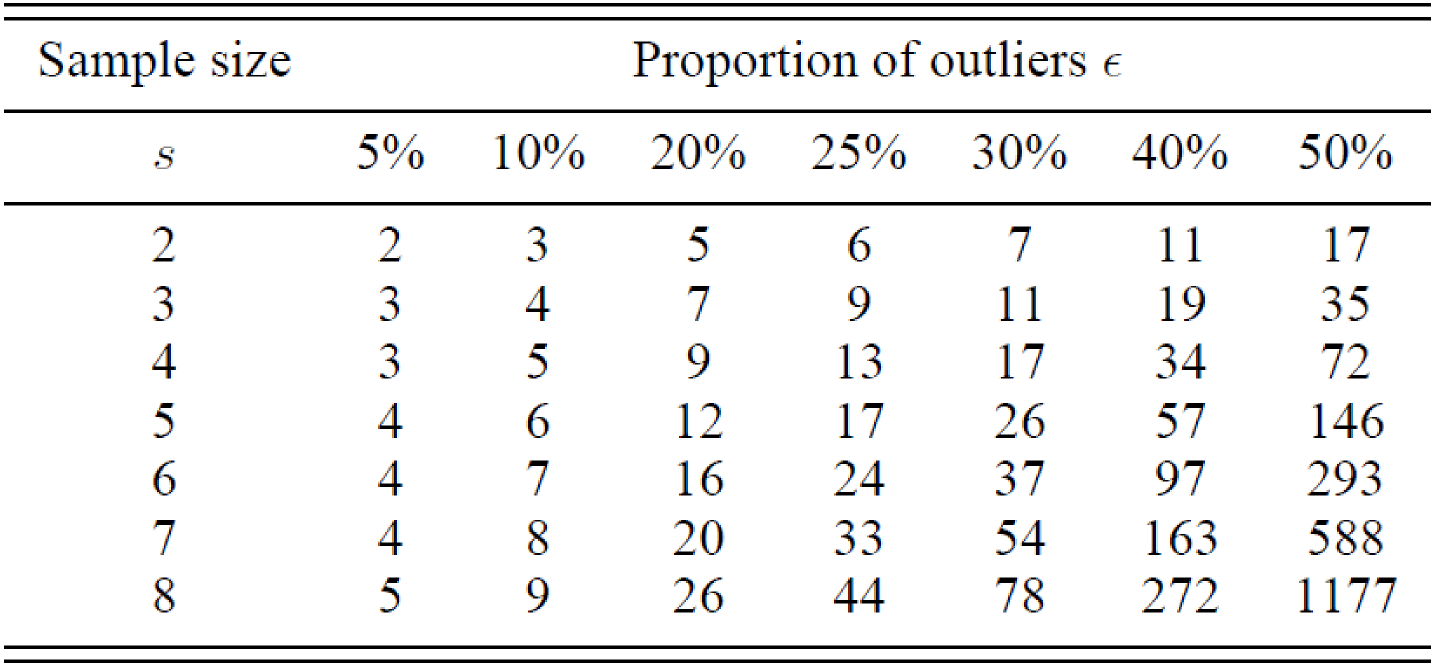
- How many iterations do we need?
- It depends on the minimal number of observations s the direct algorithm $\mathcal{A}$ requires and the percentage ϵ of outliers
- To ensure with probability p that at least one iteration contains only inliers the number of iterations needed is

$$N = \log[1 - p] / \log[1 - (1 - \epsilon)^s]$$

- Note, that this number does not depend on the number of observations

RANSAC

- How many iterations do we need?
- Iterations to achieve at least one no-outlier subset



Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Minimising algebraic distance

- Now that we have removed all outliers, we can use all remaining observations to estimate the parameter vector from all data (remember, $\widetilde{\boldsymbol{p}}$ is only calculated from a small subset)
- We have already seen how the DLT algorithm can be used to find solutions of a homogeneous equation system

$$\boldsymbol{Ap} = \boldsymbol{0}$$

- Obviously $\boldsymbol{p} = \boldsymbol{0}$ is a trivial but invalid solution, so we use the singular vector corresponding to the smallest singular value, which can be shown to be the minimum of

$$\boldsymbol{p^T A^T A p} \rightarrow min$$

- Subject to the constraint

$$\boldsymbol{p^T p} = 1$$

Minimising algebraic distance

- The problem is, that this algebraic distance has no useful interpretation and that therefore minimising it does not yield the “best” result
- For example, for the homography estimation the algebraic distance is the cross product of the two homogeneous vector

$$d_{Alg}[\boldsymbol{x_i}, \boldsymbol{H}\boldsymbol{x_i'}] = \boldsymbol{x_i'} \times \boldsymbol{H}\boldsymbol{x_i}$$

- As a measure of distance this is problematic, not least because we can multiply each homogeneous entity with a different constant factor, which will alter the result
- Also, distances measured closer to the origin of the coordinate system are longer, which is certainly very undesirable
- As a conclusion, the DLT algorithm is only an approximation of the final result

Coordinate normalisation

- Another issue is that homogeneous vectors mix units in the sense that the Euclidean part is a measure of distance, while the 1 we put at the end of the vector is unitless
- Unless we take care of this through means of the Mahalonbis distance with singular covariance matrix as seen before, we need to make sure that the ratio between the Euclidean and the homogeneous part in the vector is accounted for
- This is typically done by moving the coordinate origin of all entities considered into the centre $\boldsymbol{\mu}$ and scaling all units by a factor $1/\sigma$ to be in a similar range as the homogeneous part, i.e. between -1 and 1
- This can be achieved for points by applying a simple isometry prior to all operations

$$\boldsymbol{T} = \begin{pmatrix} \frac{1}{\sigma}\boldsymbol{I} & -\frac{\boldsymbol{\mu}}{\sigma} \\ 0 & 1 \end{pmatrix}$$

- In particular the DLT algorithm is very sensitive to this problem, so coordinate normalisation should always be applied

Maximum likelihood estimate

- Remember, that our inputs are as follows
  - Observations $\boldsymbol{l}$
  - Observation covariances $\boldsymbol{C_{ll}}$
  - Model function $g[\boldsymbol{p}, \boldsymbol{l}] = \boldsymbol{0}$
  - Intrinsic constraints $h[\boldsymbol{p}] = \boldsymbol{0}$

- Because we removed all outliers, we can now assume that the log-likelihood of the observation vector given the unknown model compliant mean vectors $\boldsymbol{\bar{l}}$ is the Mahalanobis distance

$$\Omega = \log\left[\Pr\left\{\boldsymbol{l} \mid \boldsymbol{\bar{l}}\right\}\right] + x = \left(\boldsymbol{l} - \boldsymbol{\bar{l}}\right)^T \boldsymbol{C_{ll}^{-1}} \left(\boldsymbol{l} - \boldsymbol{\bar{l}}\right)$$

Maximum likelihood estimate

- Remember, that our inputs are as follows
  - Observations $\boldsymbol{l}$
  - Observation covariances $\boldsymbol{C_{ll}}$
  - Model function $g[\boldsymbol{p},\boldsymbol{l}] = \boldsymbol{0}$
  - Intrinsic constraints $h[\boldsymbol{p}] = \boldsymbol{0}$

- The maximum likelihood estimate for $\boldsymbol{p}$ can therefore be obtained by minimising the log-likelihood subject to the two constraints g and h
- Using the Lagrange multipliers $2\boldsymbol{\lambda}$ and $2\boldsymbol{\mu}$ we minimise

$$\mathrm{L} = \left(\boldsymbol{l} - \bar{\boldsymbol{l}}\right)^T \boldsymbol{C_{ll}^{-1}}\left(\boldsymbol{l} - \bar{\boldsymbol{l}}\right) + 2\boldsymbol{\lambda^T} g\left[\boldsymbol{p},\bar{\boldsymbol{l}}\right] + 2\boldsymbol{\mu^T} h[\boldsymbol{p}]$$

Gauss-Helmert-Model

- First we calculate the Taylor approximation of the model function

$$g[\boldsymbol{p}, \boldsymbol{l}] \approx g[\boldsymbol{p_0}, \boldsymbol{l_0}] + \boldsymbol{A}(\mathbf{p} - \boldsymbol{p_0}) + \boldsymbol{B}(\mathbf{l} - \boldsymbol{l_0}) = \boldsymbol{g_0} + \boldsymbol{A\Delta p} + \boldsymbol{B\Delta l}$$

- And of the intrinsic constraints

$$h[\boldsymbol{p}] \approx h[\boldsymbol{p_0}] + \boldsymbol{H}(\boldsymbol{p} - \boldsymbol{p_0}) = \boldsymbol{h_0} + \boldsymbol{H\Delta p}$$

- Note, that we need a linearisation point $(\boldsymbol{p_0}, \boldsymbol{l_0})$ which we can initialise using our observations $\boldsymbol{l_0} = \boldsymbol{l}$ and crucially using the approximate solution $\mathbf{p_0} = \widetilde{\boldsymbol{p}}$ calculated before

Gauss-Helmert-Model

- Substituting $\boldsymbol{\Delta l} = \bar{\boldsymbol{l}} - \boldsymbol{l_0}$ into the objective function we get

$$\Omega = \left(\boldsymbol{l} - \bar{\boldsymbol{l}}\right)^{\boldsymbol{T}} \boldsymbol{C_{ll}^{-1}} \left(\boldsymbol{l} - \bar{\boldsymbol{l}}\right) = (\boldsymbol{l_0} - \boldsymbol{l} + \boldsymbol{\Delta l})^{\boldsymbol{T}} \boldsymbol{C_{ll}^{-1}} (\boldsymbol{l_0} - \boldsymbol{l} + \boldsymbol{\Delta l})$$

- And substituting the Taylor expansion into the Lagrange function yields

$$L[\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \lambda, \boldsymbol{\mu}] = \Omega[\boldsymbol{\Delta l}] + 2\boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{g_0} + \boldsymbol{A\Delta p} + \boldsymbol{B\Delta l}) + 2\boldsymbol{\mu}^{\boldsymbol{T}}(\boldsymbol{h_0} + \boldsymbol{H\Delta p})$$

- Note, that the only unknowns are $\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \boldsymbol{\lambda}, \boldsymbol{\mu}$

Gauss-Helmert-Model

- To find the solution we calculate the partial derivatives of the Lagrange function

$$L = (\boldsymbol{l_0} - \boldsymbol{l} + \boldsymbol{\Delta l})^T \boldsymbol{C}_{ll}^{-\mathbf{1}} (\boldsymbol{l_0} - \boldsymbol{l} + \boldsymbol{\Delta l}) + 2\lambda^T (\boldsymbol{g_0} + \boldsymbol{A\Delta p} + \boldsymbol{B\Delta l}) + 2\boldsymbol{\mu}^T (\boldsymbol{h_0} + \boldsymbol{H\Delta p})$$

- Which are

$$\frac{\partial}{\partial \boldsymbol{\Delta p}} L[\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \boldsymbol{\lambda}, \boldsymbol{\mu}] = 2\boldsymbol{A^T}\boldsymbol{\lambda} + 2\boldsymbol{H^T}\boldsymbol{\mu}$$

$$\frac{\partial}{\partial \boldsymbol{\Delta l}} L[\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \boldsymbol{\lambda}, \boldsymbol{\mu}] = 2\boldsymbol{C}_{ll}^{-\mathbf{1}}(\boldsymbol{l_0} - \boldsymbol{l}) + 2\boldsymbol{C}_{ll}^{-1}\boldsymbol{\Delta l} + 2\boldsymbol{B^T}\boldsymbol{\lambda}$$

$$\frac{\partial}{\partial \boldsymbol{\lambda}} L[\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \boldsymbol{\lambda}, \boldsymbol{\mu}] = 2\boldsymbol{g}_0 + 2\boldsymbol{A\Delta p} + 2\boldsymbol{B\Delta l}$$

$$\frac{\partial}{\partial \boldsymbol{\mu}} L[\boldsymbol{\Delta p}, \boldsymbol{\Delta l}, \boldsymbol{\lambda}, \boldsymbol{\mu}] = 2\boldsymbol{h_0} + 2\boldsymbol{H\Delta p}$$

Gauss-Helmert-Model

- Setting all these partial derivatives to zero and dividing out the common factor 2 we obtain 4 linear equations

$$\boldsymbol{A^T\lambda + H^T\mu = 0}$$
$$\boldsymbol{C_{ll}^{-1}(l_0 - l) + C_{ll}^{-1}\Delta l + B^T\lambda = 0}$$
$$\boldsymbol{g_0 + A\Delta p + B\Delta l = 0}$$
$$\boldsymbol{h_0 + H\Delta p = 0}$$

- Multiplying the second equation with $\boldsymbol{C_{ll}}$ we get

$$\boldsymbol{\Delta l = l - l_0 - C_{ll}B^T\lambda}$$

- Substituting this into the third equation yields

$$\boldsymbol{BC_{ll}\,B^T\lambda = g_0 + A\Delta p + B\,(l - l_0)}$$

- Which we can solve for $\boldsymbol{\lambda}$ to get

$$\boldsymbol{\lambda = \left(BC_{ll}B^T\right)^{-1}\left(g_0 + A\Delta p + B\,(l - l_0)\right)}$$

Gauss-Helmert-Model

- Finally we substitute $\boldsymbol{\lambda}$ into the first equation and get

$$\boldsymbol{A^T\left(BCB^T\right)^{-1}\left(g_0 + A\Delta p + B\,(l - l_0)\right) + H^T\mu = 0}$$

- Which together with the fourth equation

$$\boldsymbol{h_0 + H\Delta p = 0}$$

- Can be re-arranged into the Normal equation system

$$\boldsymbol{\begin{pmatrix} A^T\left(BCB^T\right)^{-1}A & H^T \\ H & 0 \end{pmatrix}\begin{pmatrix} \Delta p \\ \mu \end{pmatrix} = \begin{pmatrix} A^T\left(BCB^T\right)^{-1}(B(l_0 - l) - g_0) \\ -h_0 \end{pmatrix}}$$

Gauss-Helmert-Model

- The final iterative algorithm is as follows
  1. Choose good initial parameters $\boldsymbol{p_0} = \widetilde{\boldsymbol{p}}$ and initialise $\boldsymbol{l_0} = \boldsymbol{l}$
  2. Calculate the Taylor expansion of $\boldsymbol{g}$ and $\boldsymbol{h}$ at $\boldsymbol{p_0}$ and $\boldsymbol{l_0}$ to obtain the Jacobian matrices $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{H}$
  3. Solve the Normal equation system to obtain $\boldsymbol{\Delta p}$:
  $$\begin{pmatrix} \boldsymbol{\Delta p} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A^T}\left(\boldsymbol{BCB^T}\right)^{-1}\boldsymbol{A} & \boldsymbol{H^T} \\ \boldsymbol{H} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{A^T}\left(\boldsymbol{BCB^T}\right)^{-1}\left(\boldsymbol{B}(\boldsymbol{l_0} - \boldsymbol{l}) - \boldsymbol{g_0}\right) \\ -\boldsymbol{h_0} \end{pmatrix}$$
  4. Calculate
  $$\boldsymbol{\lambda} = \left(\boldsymbol{BCB^T}\right)^{-1}\left(\boldsymbol{g_0} + \boldsymbol{A\Delta p} + \boldsymbol{B}\,(\boldsymbol{l} - \boldsymbol{l_0})\right)$$
  5. Calculate
  $$\boldsymbol{\Delta l} = \boldsymbol{l} - \boldsymbol{l_0} - \boldsymbol{CB^T\lambda}$$
  6. Calculate the parameter estimate and adjusted observation
  $$\widehat{\boldsymbol{p}} = \boldsymbol{p_0} + \boldsymbol{\Delta p}$$
  $$\hat{\boldsymbol{l}} = \boldsymbol{l_0} + \boldsymbol{\Delta l}$$
  7. Update the linearisation point $\boldsymbol{p_0} = \widehat{\boldsymbol{p}}$ and $\boldsymbol{l_0} = \hat{\boldsymbol{l}}$ and iterate

Gauss-Helmert-Model

- To see how the covariance matrix propagates we look at the normal equation system in the last iteration of this procedure

$$\begin{pmatrix} \boldsymbol{\Delta p} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A} & \boldsymbol{H}^{\boldsymbol{T}} \\ \boldsymbol{H} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}(\boldsymbol{B}(\boldsymbol{l_0}-\boldsymbol{l})-\boldsymbol{g_0}) \\ -\boldsymbol{h_0} \end{pmatrix}$$

- We can apply linear error propagation now
- Not considering $\boldsymbol{H}$ we would get

$$\begin{aligned} \boldsymbol{C_{pp}} &= \left(\boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A}\right)^{-\boldsymbol{1}} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{BCB}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{T}}\boldsymbol{A}\left(\boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A}\right)^{-\boldsymbol{T}} \\ &= \left(\boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A}\right)^{-\boldsymbol{1}} \end{aligned}$$

- Considering $\boldsymbol{H}$ is a bit more complicated

Gauss-Helmert-Model

- To see how the covariance matrix propagates we look at the normal equation system in the last iteration of this procedure

$$\begin{pmatrix} \boldsymbol{\Delta p} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A} & \boldsymbol{H}^{\boldsymbol{T}} \\ \boldsymbol{H} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}(\boldsymbol{B}(\boldsymbol{l}_{\boldsymbol{0}} - \boldsymbol{l}) - \boldsymbol{g}_{\boldsymbol{0}}) \\ -\boldsymbol{h}_{\boldsymbol{0}} \end{pmatrix}$$

- We can apply linear error propagation now
- Considering $\boldsymbol{H}$ one can also show along similar arguments that in this case the covariance propagates as the top-left part of the inverse of the full normal equation matrix

$$\boldsymbol{C}'_{\boldsymbol{pp}} = \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{T}}\left(\boldsymbol{BCB}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}\boldsymbol{A} & \boldsymbol{H}^{\boldsymbol{T}} \\ \boldsymbol{H} & \boldsymbol{0} \end{pmatrix}^{-1}$$

Example: Homographies

- The Taylor expansion of the model function can be easily derived as

$$g_i = \boldsymbol{x}'_{\boldsymbol{0i}} \times \boldsymbol{H}_{\boldsymbol{0}}\ \boldsymbol{x}_{\boldsymbol{0i}} + \boldsymbol{A}_{\boldsymbol{i}}\boldsymbol{\Delta}\mathbf{h} + \boldsymbol{B}_{\boldsymbol{i}}\boldsymbol{\Delta}\boldsymbol{x}_{\boldsymbol{i}} + \boldsymbol{B}'_{\boldsymbol{i}}\boldsymbol{\Delta}\boldsymbol{x}'_{\boldsymbol{i}}$$

- from the bi-linear equations

$$\underbrace{\left(\boldsymbol{x}_{\boldsymbol{0i}}^{T} \otimes \boldsymbol{S}[\boldsymbol{x}_{\boldsymbol{0i}}{}']\right)}_{\boldsymbol{A}_{\boldsymbol{i}}} \boldsymbol{h} = \boldsymbol{A}_{\boldsymbol{i}}\boldsymbol{h} = \boldsymbol{0}$$

$$\underbrace{\boldsymbol{S}[\boldsymbol{x}'_{\boldsymbol{0i}}]\boldsymbol{H}_{\boldsymbol{0}}}_{\boldsymbol{B}_{\boldsymbol{i}}} \boldsymbol{x}_{\boldsymbol{i}} = \boldsymbol{B}_{\boldsymbol{i}}\boldsymbol{x}_{\boldsymbol{i}} = \boldsymbol{0}$$

$$\underbrace{-\boldsymbol{S}[\boldsymbol{H}_{\boldsymbol{0}}\ \boldsymbol{x}_{\boldsymbol{0i}}]}_{\boldsymbol{B}'_{\boldsymbol{i}}}\boldsymbol{x}'_{\boldsymbol{i}} = \boldsymbol{B}'_{\boldsymbol{i}}\boldsymbol{x}'_{\boldsymbol{i}} = \boldsymbol{0}$$

Example: Homographies

- The Taylor expansion of the model function can be easily derived as

$$g_i = \boldsymbol{x}'_{\boldsymbol{0i}} \times \boldsymbol{H_0}\ \boldsymbol{x_{0i}} + \boldsymbol{A_i}\boldsymbol{\Delta}\mathbf{h} + \boldsymbol{B_i}\boldsymbol{\Delta x_i} + \boldsymbol{B}'_{\boldsymbol{i}}\boldsymbol{\Delta x}'_{\boldsymbol{i}}$$

- The vector $\boldsymbol{h}$ is homogeneous, which requires to have the intrinsic constraint

$$\boldsymbol{h}^{\boldsymbol{T}}\boldsymbol{h} = 1$$

- which has the following Taylor expansion

$$h = \left(\boldsymbol{h}_{\boldsymbol{0}}^{\boldsymbol{T}}\boldsymbol{h}_{\boldsymbol{0}} - 1\right) + 2\boldsymbol{h}_{\boldsymbol{0}}^{\boldsymbol{T}}\boldsymbol{\Delta}\mathbf{h}$$

Example: Homographies

- We now create the stacked matrices

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_1} \\ \vdots \\ \boldsymbol{A_n} \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} \boldsymbol{B_1} & \boldsymbol{B'_1} & & & \\ & & \ddots & & \\ & & & \boldsymbol{B_n} & \boldsymbol{B'_n} \end{pmatrix} \qquad \boldsymbol{g_0} = \begin{pmatrix} \boldsymbol{g_1} \\ \vdots \\ \boldsymbol{g_n} \end{pmatrix}$$

- And stack the homogeneous observation covariance matrices

$$\boldsymbol{C_{x_i x_i}} = \boldsymbol{C_{x'_i x'_i}} = \begin{pmatrix} \mathbf{1} & & \\ & \mathbf{1} & \\ & & \mathbf{0} \end{pmatrix} \qquad \boldsymbol{C} = \begin{pmatrix} \boldsymbol{C_{x_1 x_1}} & & & & \\ & \boldsymbol{C_{x'_1 x'_1}} & & & \\ & & \ddots & & \\ & & & \boldsymbol{C_{x_n x_n}} & \\ & & & & \boldsymbol{C_{x'_n x'_n}} \end{pmatrix}$$

Example: Homographies

- We also need to create the residual vector

$$\boldsymbol{r} = \begin{pmatrix} \boldsymbol{x_{01}} - \boldsymbol{x_1} \\ \boldsymbol{x_{01'}} - \boldsymbol{x'_1} \\ \vdots \\ \boldsymbol{x_{0n}} - \boldsymbol{x_n} \\ \boldsymbol{x_{0n'}} - \boldsymbol{x'_n} \end{pmatrix}$$

- This are all the necessary components to build and solve the normal equation system

$$\begin{pmatrix} \boldsymbol{\Delta h} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A^T}\left(\boldsymbol{BCB^T}\right)^{\boldsymbol{-1}}\boldsymbol{A} & 2\boldsymbol{h_0} \\ 2\boldsymbol{h_0^T} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{A^T}\left(\boldsymbol{BCB^T}\right)^{\boldsymbol{-1}}(\boldsymbol{Br} - \boldsymbol{g_0}) \\ -\left(\boldsymbol{h_0^T}\boldsymbol{h_0} - 1\right) \end{pmatrix}$$

Example: Homographies

- The homography is the updated according to

$$\boldsymbol{h_0} \mathrel{+}= \boldsymbol{\Delta h}$$

- We also calculate

$$\boldsymbol{\lambda} = \left(\boldsymbol{BCB^T}\right)^{-\mathbf{1}}(\boldsymbol{g_0} + \boldsymbol{A\Delta p} - \boldsymbol{B\,r})$$

$$\begin{pmatrix} \boldsymbol{\Delta x_1} \\ \mathbf{\Delta x_1'} \\ \vdots \\ \boldsymbol{\Delta x_n} \\ \mathbf{\Delta x_n'} \end{pmatrix} = -\boldsymbol{r} - \boldsymbol{CB^T\lambda}$$

- To update the observation linearization point

$$\boldsymbol{x_{0i}} \mathrel{+}= \mathbf{\Delta x_i} \qquad \mathbf{x_{0i}'} \mathrel{+}= \mathbf{\Delta x_i'}$$

Example: Homographies

- Because of the linearization we should make sure that the constraints are not violated after each iteration, by enforcing spherical normalisation of the parameter vector

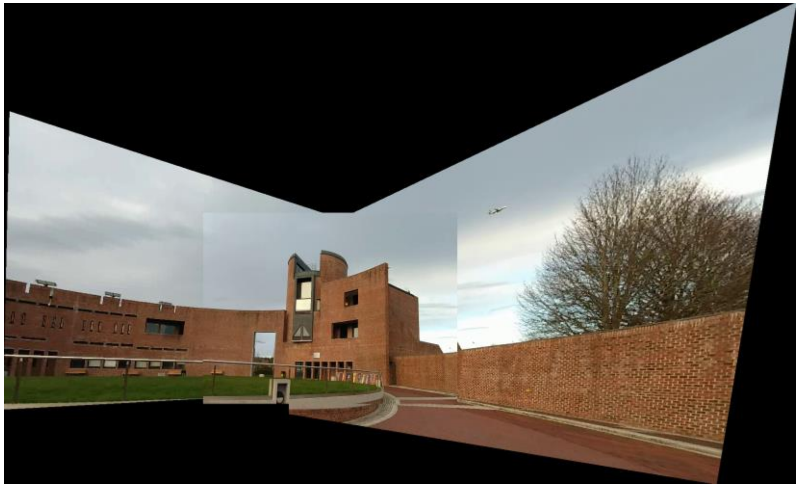
$$\boldsymbol{h_0} /= \sqrt{\boldsymbol{h_0^T h_0}}$$

- As well as Euclidean normalisation of the adjusted observation vector

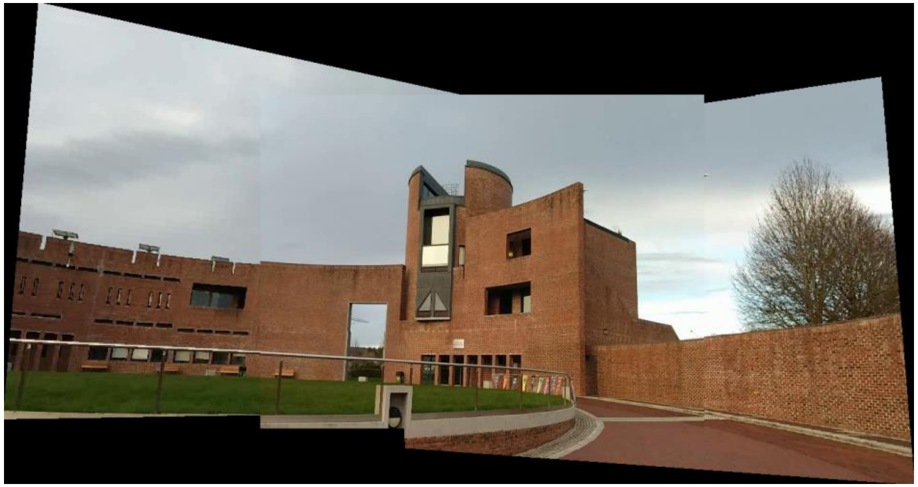
$$\boldsymbol{x_{0i}} /= x_{0i2} \qquad \boldsymbol{x'_{0i}} /= x'_{0i2}$$

Example: Homographies

- The stitching result of the combined RANSAC and Gauss-Helmert-Model parameter estimate are much better than the pure DLT solution



DLT



RANSAC + GHM

Thank you for your attention!