

3D Computer Vision Parameter Estimation

Nassir Navab

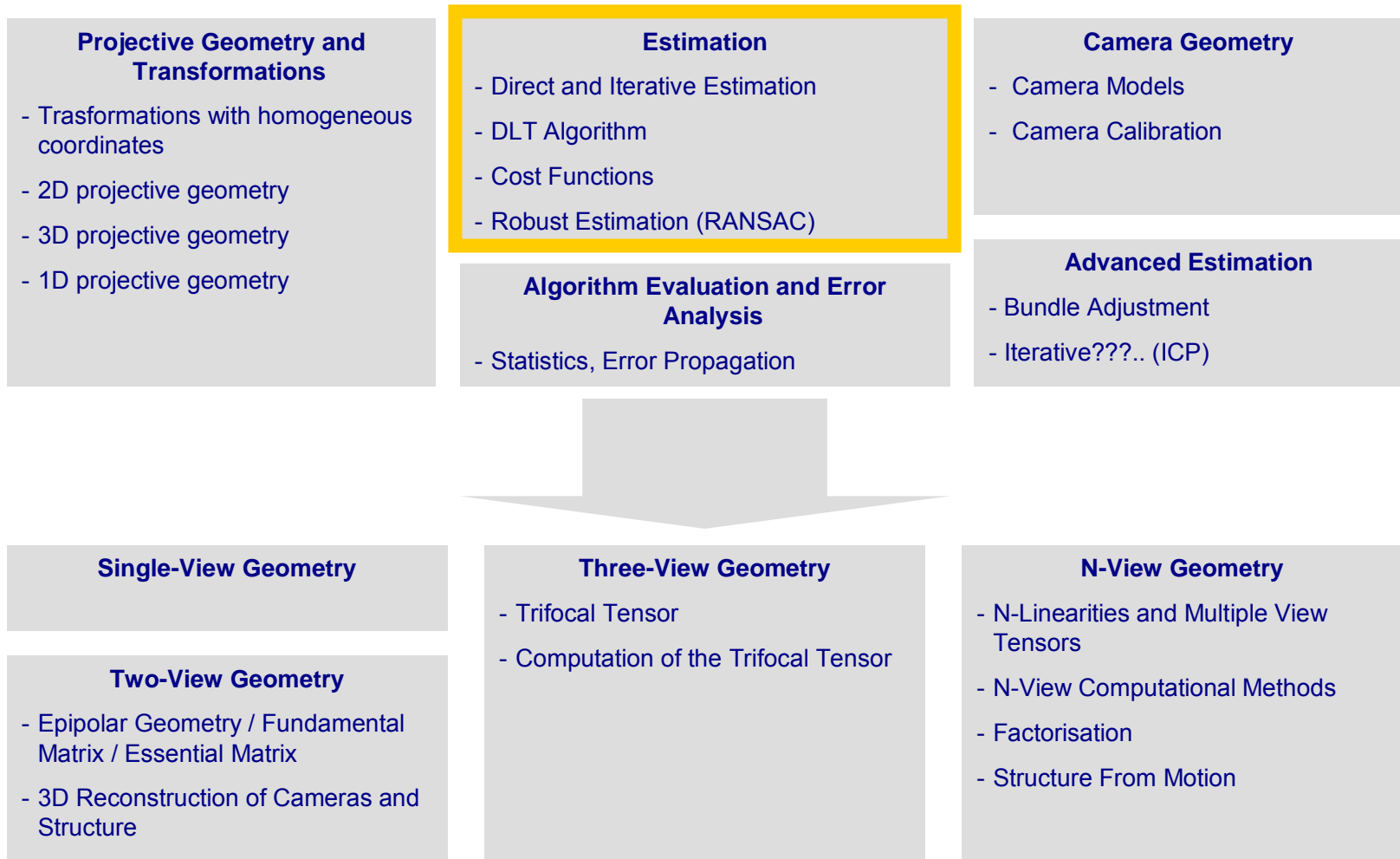
based on a course given at UNC by Marc Pollefeys & the book "Multiple View Geometry" by Hartley & Zisserman

18 November 2006

chair for computer aided medical procedures

department of computer science | technische universität münchen

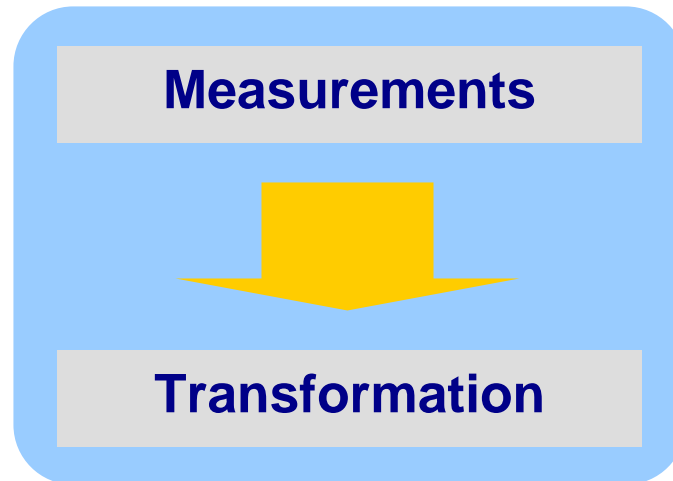
Outline of the Computer Vision Lecture



Parameter Estimation

- “Computation of a **transformation** (or some other mathematical quantity) based on **measurements** of some nature“

(Hartley&Zisserman)



Parameter Estimation Examples

- 2D homography
 - Given a set of (x_i, x_i') , compute H (such that $x_i' = Hx_i$)
- 3D to 2D camera projection
 - Given a set of (X_i, x_i) , compute P (such that $x_i = PX_i$)
- Fundamental matrix
 - Given a set of (x_i, x_i') , compute F (such that $x_i'^T F x_i = 0$)
- Trifocal tensor
 - Given a set of (x_i, x_i', x_i'') , compute T

Parameter Estimation Example used throughout the Lesson

- The estimation example for this lesson: 2D homography

Given a set of **measurements** (x_i, x_i') ,
compute the **transformation** H ,
such that $x_i' = Hx_i$



$$Hx_i = x_i'$$

→

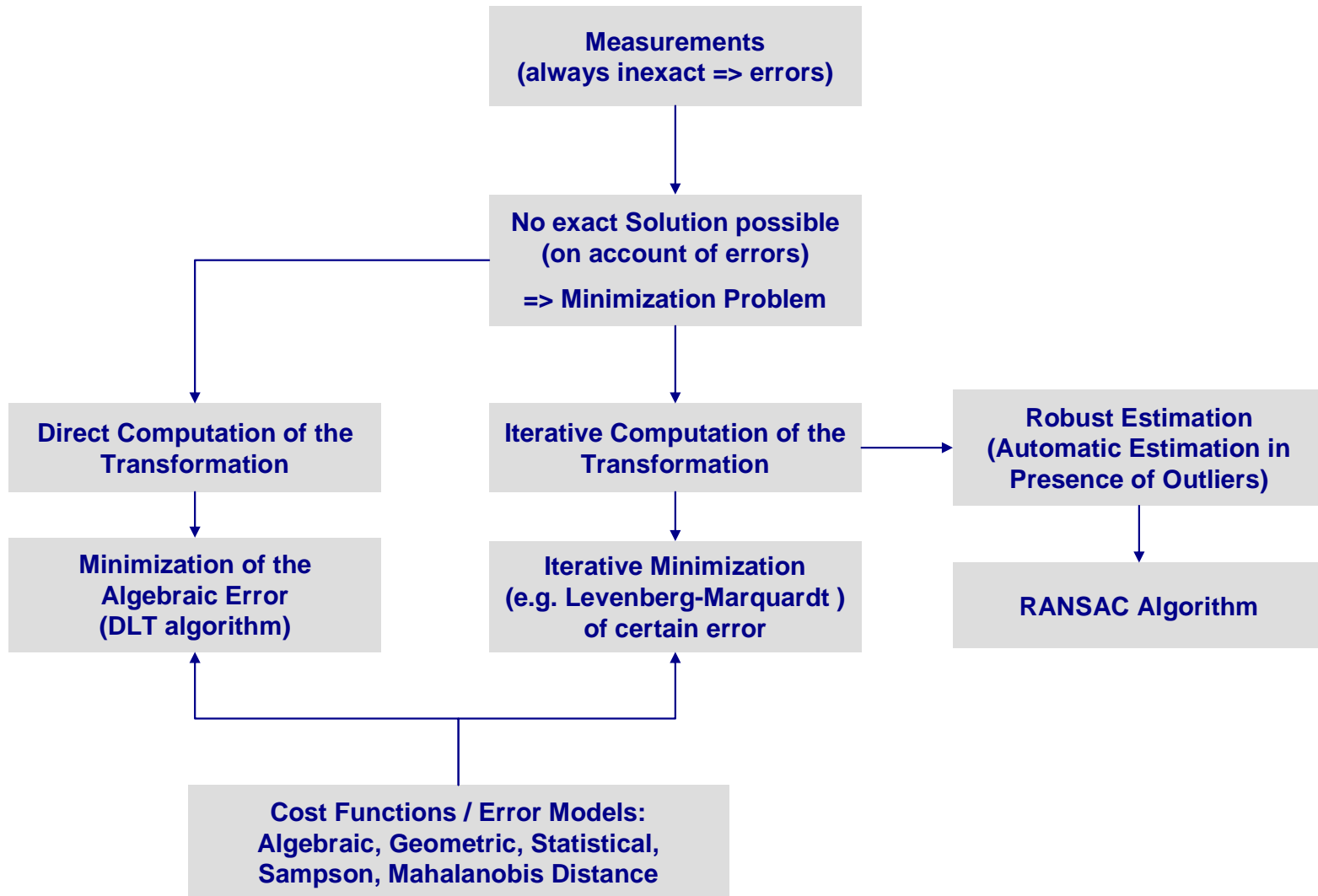
$$H = ?$$



Outline – Parameter Estimation

- Introduction and Overview
- First Estimation Example: DLT-Algorithm (Direct Estimation)
- Iterative Estimation
- Cost Functions:
 - Algebraic Distance
 - Geometric Distance:
 - Error in one Image
 - Error in both Images (Simple Transfer Error)
 - Error in both Images (Symmetric Transfer Error)
 - Reprojection Error
- Example: 1D
- Cost Functions II:
 - Sampson Error
 - Statistical Cost Function (Maximum Likelihood Estimation)
 - Mahalanobis Distance
- Robust Estimation (RANSAC-Algorithm)
- Example: Fully Automatic Estimation of Homography from two Images
- Further Questions

Outline of the Estimation Process



Number of Measurements Required

- At least as many independent equations as degrees of freedom required

- Example:
$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

2 independent equations / point

8 degrees of freedom

$$4 \times 2 \geq 8$$

Number of Measurements Required

- At least as many independent equations as degrees of freedom required

- Example:
$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2 independent equations / point
8 degrees of freedom

$$4 \times 2 \geq 8$$

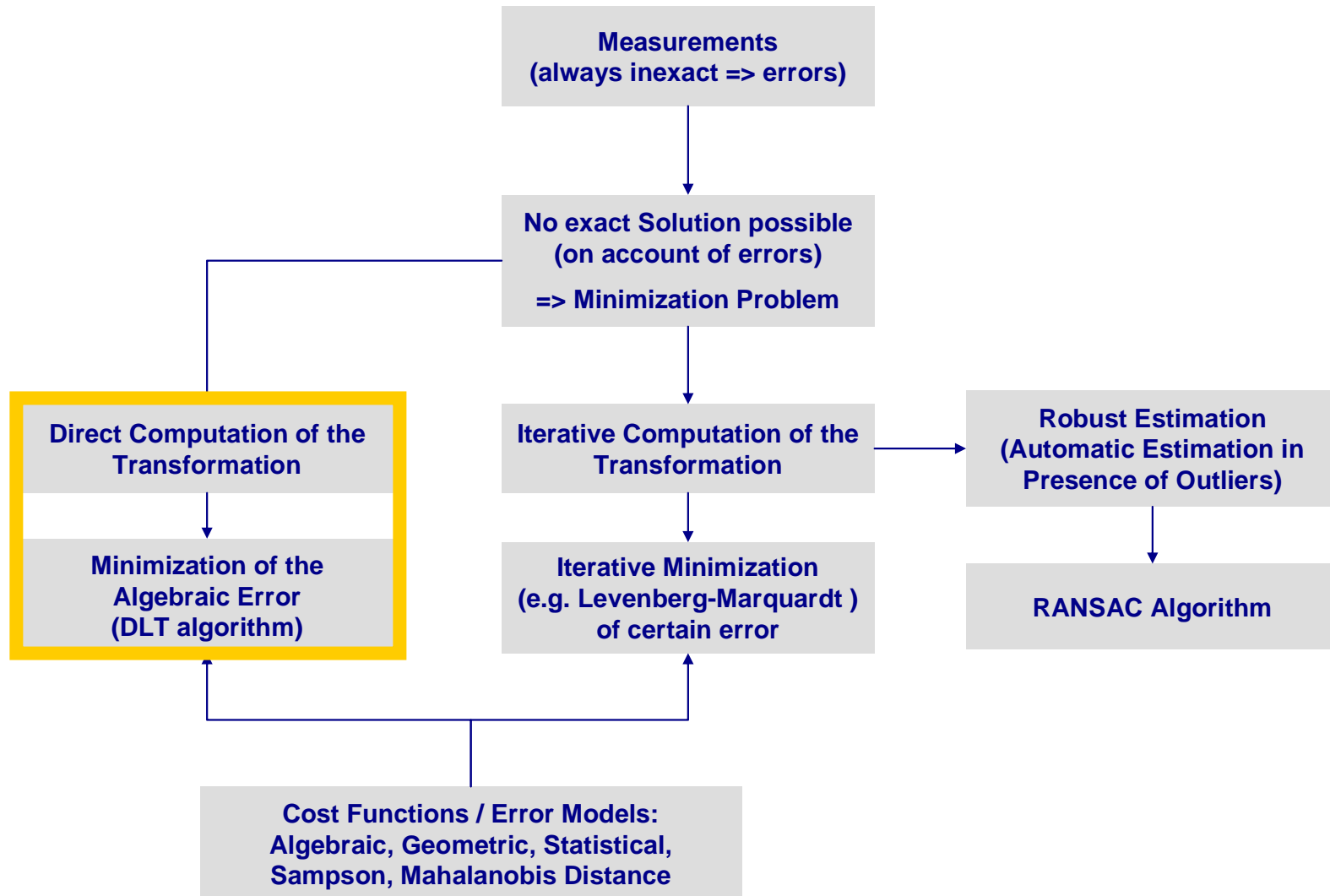
Approximate Solutions

- Minimal solution
 - 4 points yield an exact solution for H
- More points
 - No exact solution, because measurements are inexact (“noise”)
 - Search for “best” according to some cost function
 - Algebraic or geometric/statistical cost

Gold Standard Algorithm

- Cost function that is optimal for some assumptions
- Computational algorithm that minimizes it is called “Gold Standard” algorithm
- Other algorithms can then be compared to it

Direct Linear Transformation



Direct Linear Transformation (DLT)

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$$

$$\Rightarrow \mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$$

$$\Leftrightarrow \mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3\top} \mathbf{x}_i - w'_i \mathbf{h}^{2\top} \mathbf{x}_i \\ w'_i \mathbf{h}^{1\top} \mathbf{x}_i - x'_i \mathbf{h}^{3\top} \mathbf{x}_i \\ x'_i \mathbf{h}^{2\top} \mathbf{x}_i - y'_i \mathbf{h}^{1\top} \mathbf{x}_i \end{pmatrix} = \mathbf{0}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \\ -y'_i \mathbf{x}_i^\top & x'_i \mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$$\Leftrightarrow \mathbf{A}_i \mathbf{h} = \mathbf{0}$$

□ given:

$$\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top$$

$$\mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top} \mathbf{x}_i \\ \mathbf{h}^{2\top} \mathbf{x}_i \\ \mathbf{h}^{3\top} \mathbf{x}_i \end{pmatrix}$$

Direct Linear Transformation (DLT)

- Equations are linear in \mathbf{h} : $\mathbf{A}_i \mathbf{h} = 0$
- Only 2 out of 3 are linearly independent (indeed, 2 eq/pt)

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \\ -y'_i \mathbf{x}_i^\top & x'_i \mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = \mathbf{0}$$

$$x'_i \mathbf{A}_i^1 + y'_i \mathbf{A}_i^2 + w'_i \mathbf{A}_i^3 = \mathbf{0}$$

Direct Linear Transformation (DLT)

- Equations are linear in \mathbf{h} : $\mathbf{A}_i \mathbf{h} = 0$
- Only 2 out of 3 are linearly independent (indeed, 2 eq/pt)

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{X}_i^\top & y'_i \mathbf{X}_i^\top \\ w'_i \mathbf{X}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{X}_i^\top \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = \mathbf{0}$$

(only drop third row if $w'_i \neq 0$)

- Holds for any homogeneous representation, e.g. $(x'_i, y'_i, 1)$

Direct Linear Transformation (DLT)

- Solving for H

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} \mathbf{h} = \mathbf{0}$$

Direct Linear Transformation (DLT)

- Solving for H

$$Ah = 0$$

size A is 8×9 or 12×9 , but rank 8

Trivial solution is $h=0_9^T$ is not interesting

1-D null-space yields solution of interest

pick for example the one with $\|h\| = 1$

Direct Linear Transformation (DLT)

- Over-determined solution

$$Ah = 0$$

- No exact solution because of inexact measurement i.e. “noise”
- Find approximate solution
 - Additional constraint needed to avoid 0, e.g. $\|h\| = 1$
 - $Ah = 0$ not possible, so minimize $\|Ah\|$

Direct Linear Transformation (DLT)

- Find approximate solution
 - Additional constraint needed to avoid 0, e.g. $\|\mathbf{h}\| = 1$
 - $A\mathbf{h} = \mathbf{0}$ not possible, so minimize $\|A\mathbf{h}\|$

- This is a constrained optimization problem
- Solution with Lagrange Multiplier Method and SVD

Direct Linear Transformation (DLT)

$$\min(h^t A^t A h + \lambda(1 - h^t h))$$

$$\text{Derivative} \rightarrow 2A^t A h - 2\lambda h = 0$$

$$\rightarrow A^t A h = \lambda h \rightarrow \text{eigenvector}$$

$$\rightarrow \min(\lambda h^t h + \lambda - \lambda h^t h) = \min(\lambda)$$

$$\rightarrow \text{eigenvec} - \text{associated} - \text{to} - \text{smallest} - \text{eigenvalue}(A^t A)$$

DLT algorithm

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i' = Hx_i$

Algorithm

- (i) For each correspondence $x_i \leftrightarrow x_i'$ compute A_i . Usually only two first rows needed.
- (ii) Assemble n 2×9 matrices A_i into a single $2n \times 9$ matrix A
- (iii) Solution for h is the eigenvector associated with the smallest eigenvalue of $A^t A$
- (iv) Determine H from h

SVD Reminder (Singular Value Decomposition)

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T \quad m \geq n$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

$$U^T U = I$$

$$V^T V = I$$

$$A = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + \cdots + U_n \sigma_n V_n^T$$

$$U \Sigma V^T X$$

SVD Reminder II

$$U \Sigma V^T X$$

$$A = U \Sigma V^T \rightarrow A^t A = V \Sigma U^T U \Sigma V^t = V \Sigma^2 V^{-1}$$

Homogeneous least-squares

$$\min \|AX\| \text{ subject to } \|X\| = 1 \quad \text{solution } X = V_n$$

DLT algorithm

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i' = Hx_i$

Algorithm

- (i) For each correspondence $x_i \leftrightarrow x_i'$ compute A_i . Usually only two first rows needed.
- (ii) Assemble n 2×9 matrices A_i into a single $2n \times 9$ matrix A
- (iii) Obtain SVD of A (i.e. $U \Sigma V^T$). Solution for h is last column of V
- (iv) Determine H from h

Inhomogeneous Solution

- Since h can only be computed up to scale, we introduce \tilde{h}
- we pick $h_j=1$, e.g. $h_9=1$, and solve for 8-vector \tilde{h}

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\ x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & x_i x_i' & y_i x_i' \end{bmatrix} \tilde{h} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix}$$

- Solve using Gaussian elimination (4 points) or using linear least-squares (more than 4 points)

Least Square Solution

$$Ax = b$$

$$\text{Min}((Ax - b)^t (Ax - b))$$

$$\text{Derivative} \rightarrow 2A^t Ax - 2A^t b = 0$$

$$x = (A^t A)^{-1} A^t b$$

$$\text{residual} - \text{Error} = A(A^t A)^{-1} A^t b - b$$

Inhomogeneous Solution

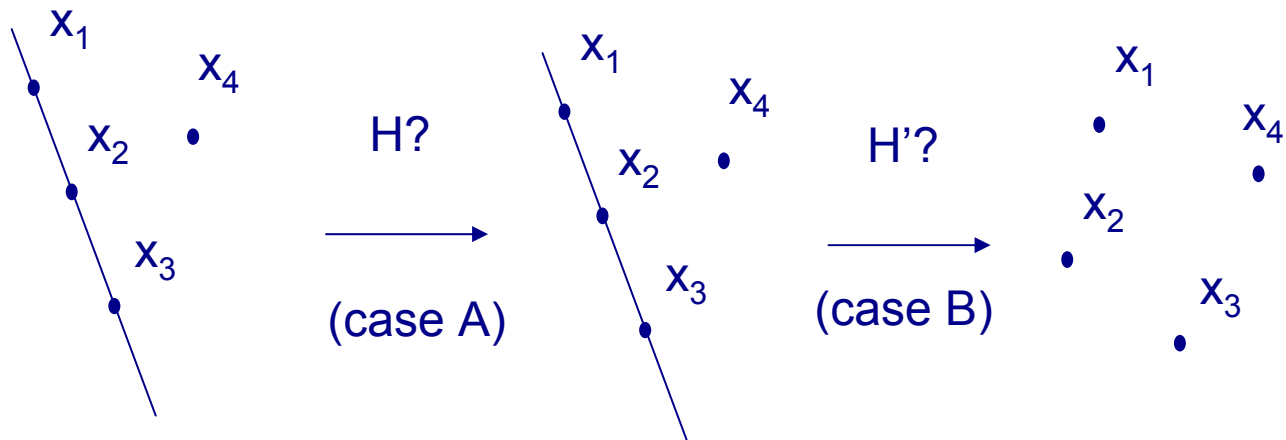
Since h can only be computed up to scale, pick $h_j=1$, e.g. $h_9=1$, and solve for 8-vector $\tilde{\mathbf{h}}$

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\ x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & x_i x_i' & y_i x_i' \end{bmatrix} \tilde{\mathbf{h}} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix}$$

However, if $h_9=0$ this approach fails
 also poor results if h_9 close to zero
 Therefore, not recommended

Note $h_9=H_{33}=0$ if origin is mapped to infinity $\mathbf{1}_\infty^T \mathbf{H} \mathbf{x}_0 = [0 \quad 0 \quad 1] \mathbf{H} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$

Degenerate Configurations



Solutions from Lines, etc.

2D homographies from 2D lines

$$l'_i = H^{-T} l_i \quad Ah = 0$$

Minimum of 4 lines

2D affinities (6 dof)

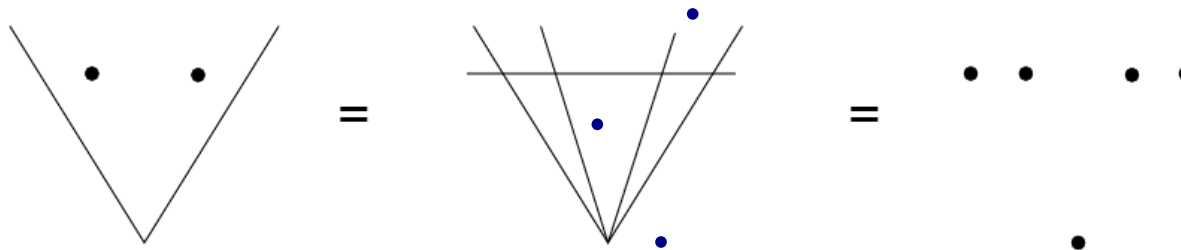
Minimum of 3 points or lines

3D Homographies (15 dof)

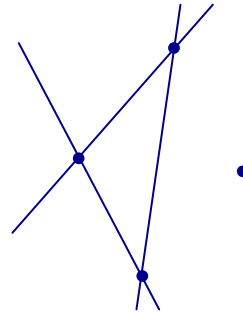
Minimum of 5 points or 5 planes

Conic provides 5 constraints

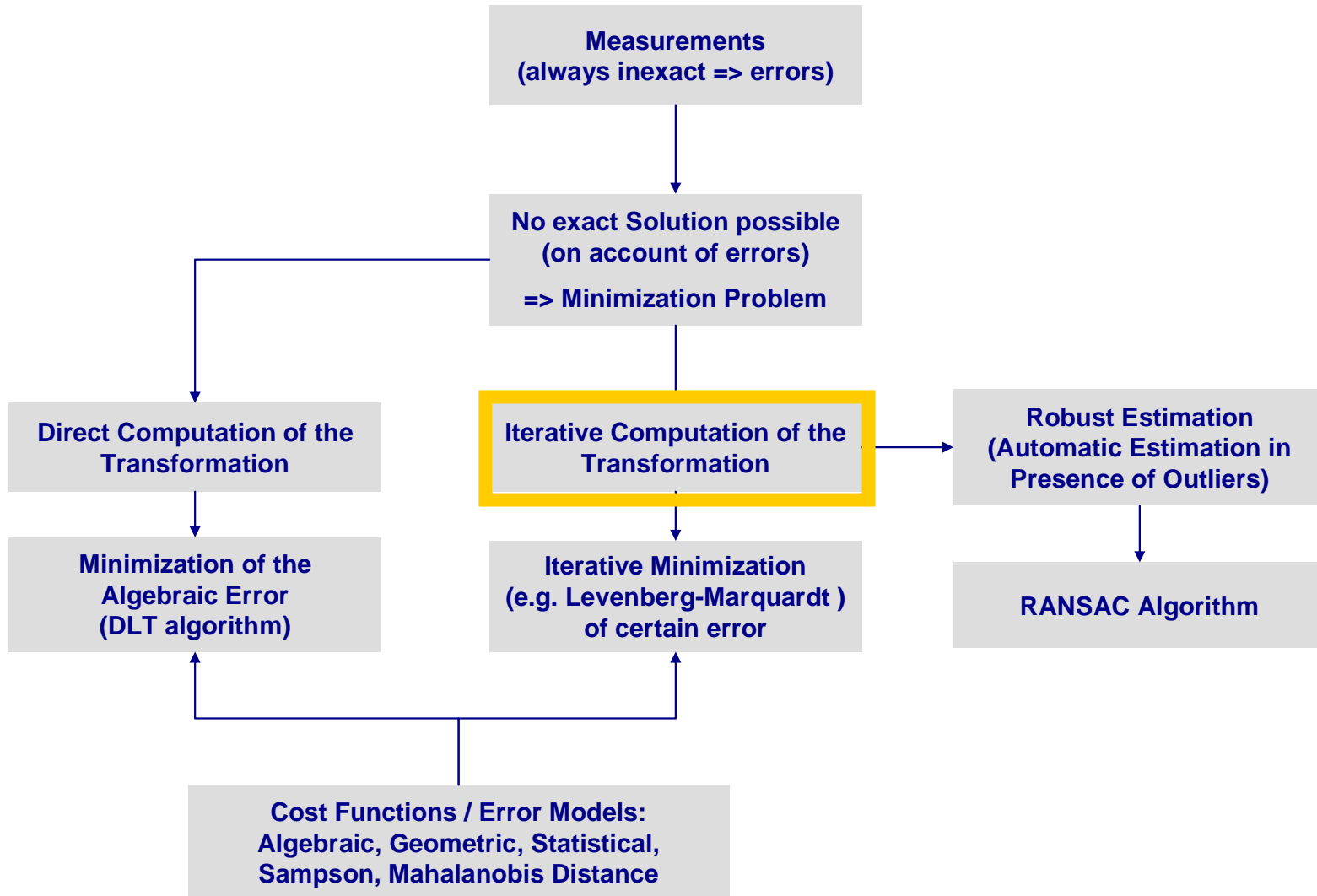
Mixed Configurations



Mixed Configurations



Iterative Estimation



Iterative Estimation

- Geometric Cost Functions cannot be minimized directly
- Iterative / Non-Linear Algorithms needed

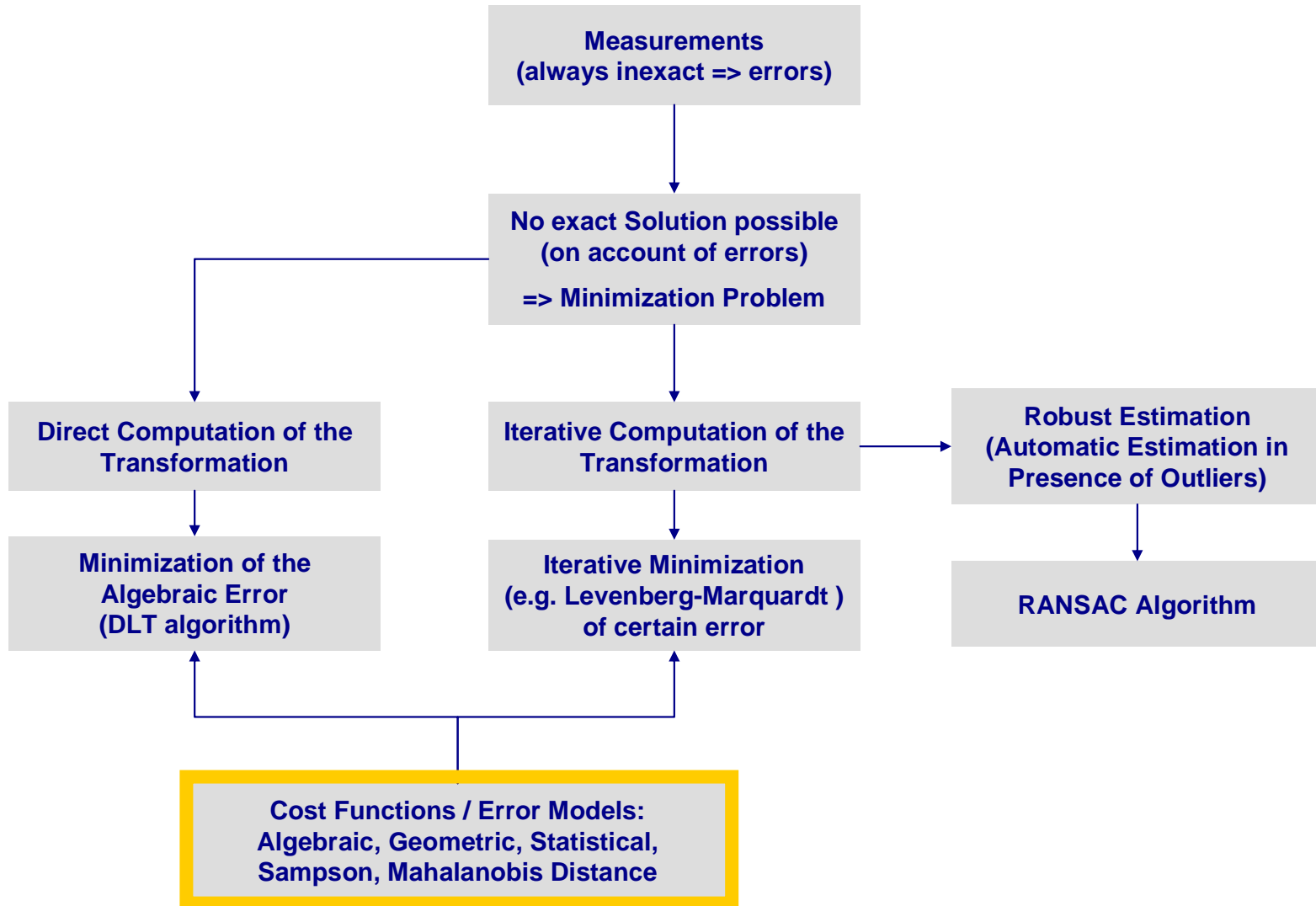
- Drawbacks:
 - slower
 - initial estimate needed
 - risk of converging to local instead of global minimum or not at all
 - selection of stopping criterion difficult

- Careful implementation needed

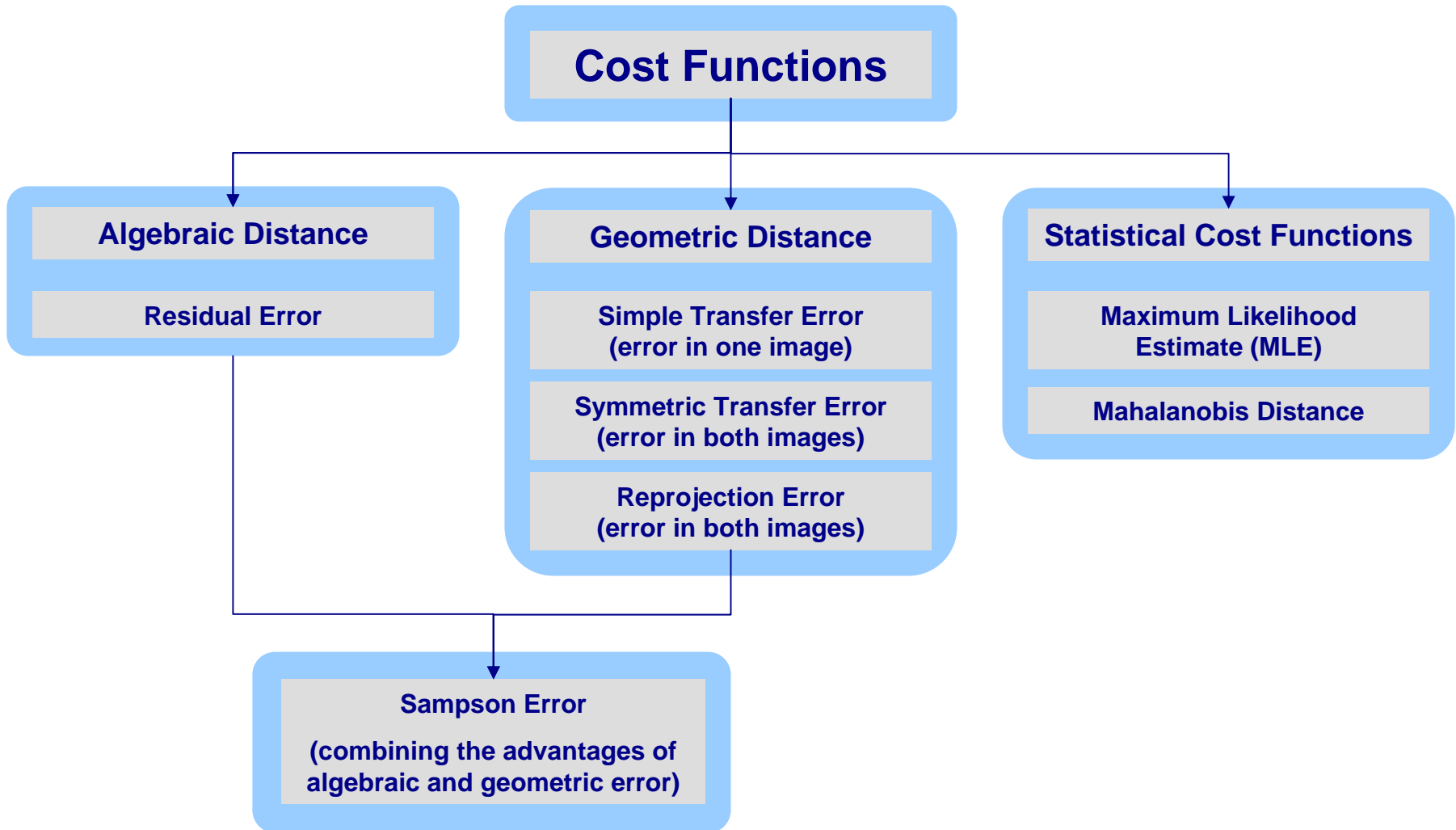
Iterative Estimation

- Needed steps:
 - Choice of suitable **Cost Function**
 - Choice of suitable **Parametrization**
 - Specification of **Function** that expresses Cost in Terms of Parameters
 - Choice of **Initial Estimate** (e.g. use of DLT)
 - **Iteration**

Cost Functions



Overview of Cost Functions / Error Models



Cost Functions

- Algebraic distance
- Geometric distance

- Comparison
- Geometric interpretation
- Sampson Error

Algebraic Distance

DLT minimizes $\|Ah\|$

$e = Ah$ residual vector

e_i partial vector for each $(x_i \leftrightarrow x_i')$
algebraic error vector

$$d_{\text{alg}}(x'_i, Hx_i)^2 = \|e_i\|^2 = \left\| \begin{bmatrix} 0^\top & -w'_i x_i^\top & -y'_i x_i^\top \\ -w'_i x_i^\top & 0^\top & -x'_i x_i^\top \end{bmatrix} h \right\|^2$$

Algebraic Distance

$$d_{\text{alg}}(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 = \|e_i\|^2 = \left\| \begin{bmatrix} 0^\top & -w'_i \mathbf{x}_i^\top & -y'_i \mathbf{x}_i^\top \\ -w'_i \mathbf{x}_i^\top & 0^\top & -x'_i \mathbf{x}_i^\top \end{bmatrix} \mathbf{h} \right\|^2$$

$$d_{\text{alg}}(\mathbf{x}_1, \mathbf{x}_2)^2 = a_1^2 + a_2^2 \quad \text{where } \mathbf{a} = (a_1, a_2, a_3)^\top = \mathbf{x}_1 \times \mathbf{x}_2$$

$$\sum_i d_{\text{alg}}(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 = \sum_i \|e_i\|^2 = \|\mathbf{A}\mathbf{h}\|^2 = \|\mathbf{e}\|^2$$

Not geometrically/statistically meaningful, but given good normalization it works fine and is very fast (use for initialization)

Geometric Distance

- Error in one Image:
 - Simple Transfer Error

- Error in both Images:
 - Symmetric Transfer Error
 - Reprojection Error

Geometric Distance

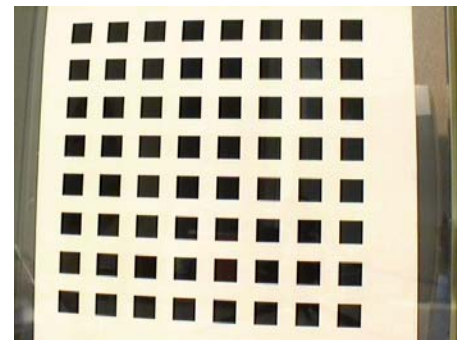
\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates
 $d(.,.)$ Euclidean distance

Error in one image

(assume the measurements in the first image have no error)

$$\hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}} \sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2$$

e.g.: calibration patterns allow for rather exact measurements

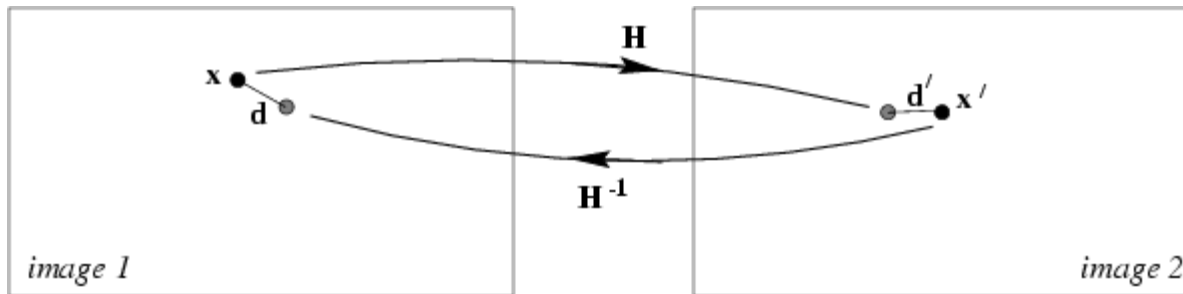


Geometric Distance

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates
 $d(.,.)$ Euclidean distance

Symmetric Transfer Error
 (assume errors in both images)

$$\hat{\mathbf{H}} = \operatorname{argmin}_{\mathbf{H}} \sum_i d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2$$



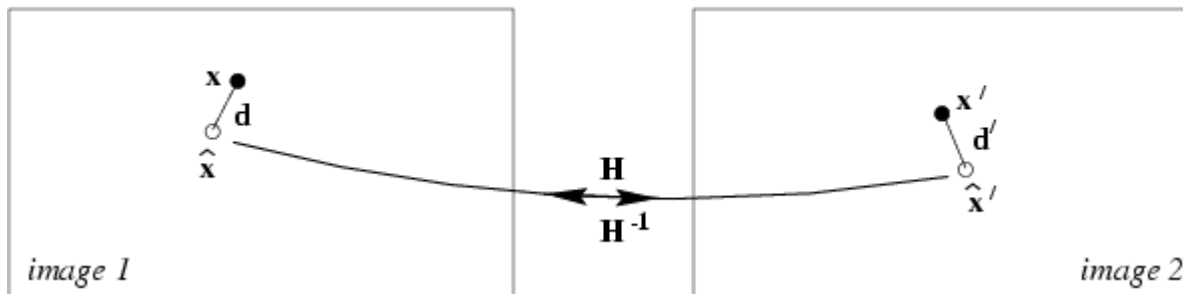
Geometric Distance

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates
 $d(.,.)$ Euclidean distance

- Reprojection Error
 - assume error in both images
 - minimize over the transformation and the correspondences

$$\left(\hat{\mathbf{H}}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i \right) = \underset{\mathbf{H}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i}{\operatorname{argmin}} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

subject to $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$



Geometric Distance - Summary

Error in one image

$$\hat{H} = \operatorname{argmin}_H \sum_i d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2$$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates
 $d(.,.)$ Euclidean distance

Symmetric transfer error

$$\hat{H} = \operatorname{argmin}_H \sum_i d(\mathbf{x}_i, H^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, H\mathbf{x}_i)^2$$

Reprojection error

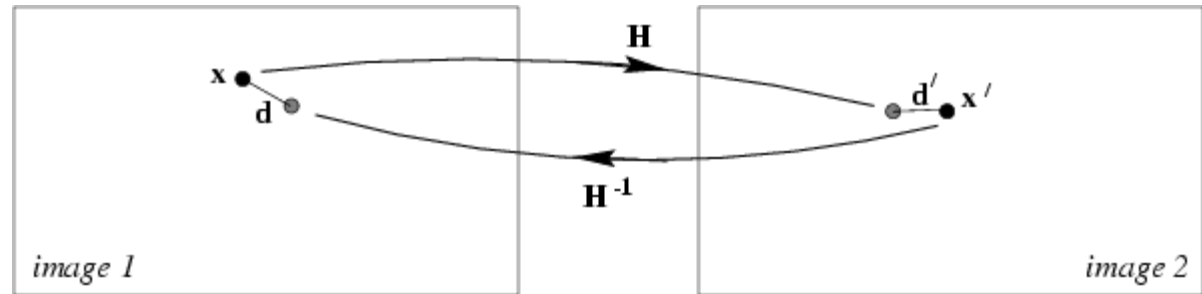
$$(\hat{H}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i) = \operatorname{argmin}_{H, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

subject to $\hat{\mathbf{x}}'_i = \hat{H}\hat{\mathbf{x}}_i$

Comparison of Symmetric Transfer Error and the Reprojection Error

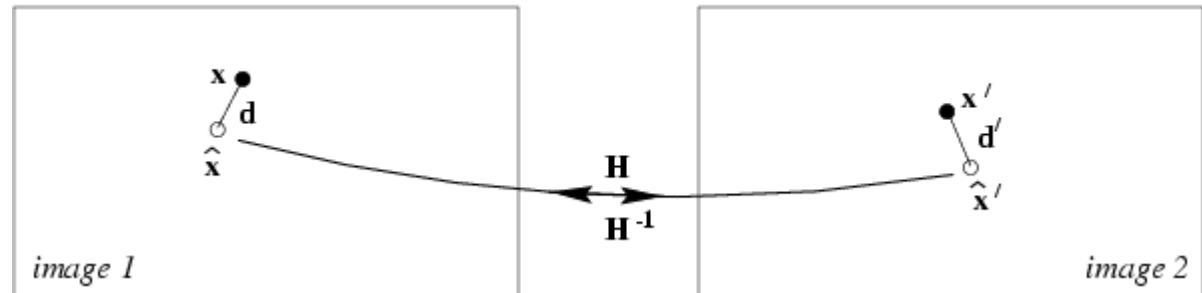
Symmetric Transfer Error:

$$d(x, H^{-1}x')^2 + d(x', Hx)^2$$



Reprojection Error:

$$d(x, \hat{x})^2 + d(x', \hat{x}')^2$$



Comparison of Geometric and Algebraic Distances

Error in one image

$$\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top \quad \hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{y}'_i, \hat{w}'_i)^\top = \mathbf{H}\bar{\mathbf{x}}$$

$$\begin{bmatrix} 0^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & 0^\top & -x'_i \mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix}$$

$$\mathbf{A}_i \mathbf{h} = \mathbf{e}_i = \begin{pmatrix} y'_i \hat{w}'_i - w'_i \hat{y}'_i \\ w'_i \hat{x}'_i - x'_i \hat{w}'_i \end{pmatrix}$$

Comparison of Geometric and Algebraic Distances

Error in one image

$$\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top \quad \hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{y}'_i, \hat{w}'_i)^\top = \mathbf{H}\bar{\mathbf{x}}$$

$$\mathbf{A}_i \mathbf{h} = \mathbf{e}_i = \begin{pmatrix} y'_i \hat{w}'_i - w'_i \hat{y}'_i \\ w'_i \hat{x}'_i - x'_i \hat{w}'_i \end{pmatrix}$$

$$d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = (y'_i \hat{w}'_i - w'_i \hat{y}'_i)^2 + (w'_i \hat{x}'_i - x'_i \hat{w}'_i)^2$$

$$\begin{aligned} d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 &= (y'_i / w'_i - \hat{y}'_i / \hat{w}'_i)^2 + (\hat{x}'_i / \hat{w}'_i - x'_i / w'_i)^2 \\ &= d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 / w'_i \hat{w}'_i \end{aligned}$$

Comparison of Geometric and Algebraic Distances

Error in one image

$$\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top \quad \hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{y}'_i, \hat{w}'_i)^\top = H\bar{\mathbf{x}}$$

$$d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 / w'_i \hat{w}'_i$$

$w'_i = 1$ typical, but not $\hat{w}'_i = h_3 \mathbf{x}_i$, except for affinities

For affinities DLT can minimize geometric distance

$$H_A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Geometric Interpretation of the Reprojection Error

Estimating homography~fit surface \mathcal{V}_H to points $X=(x,y,x',y')^T$ in \mathbb{R}^4
 $\mathbf{x}'_i \times H\mathbf{x}_i = 0$ represents 2 quadrics in \mathbb{R}^4 (quadratic in X)

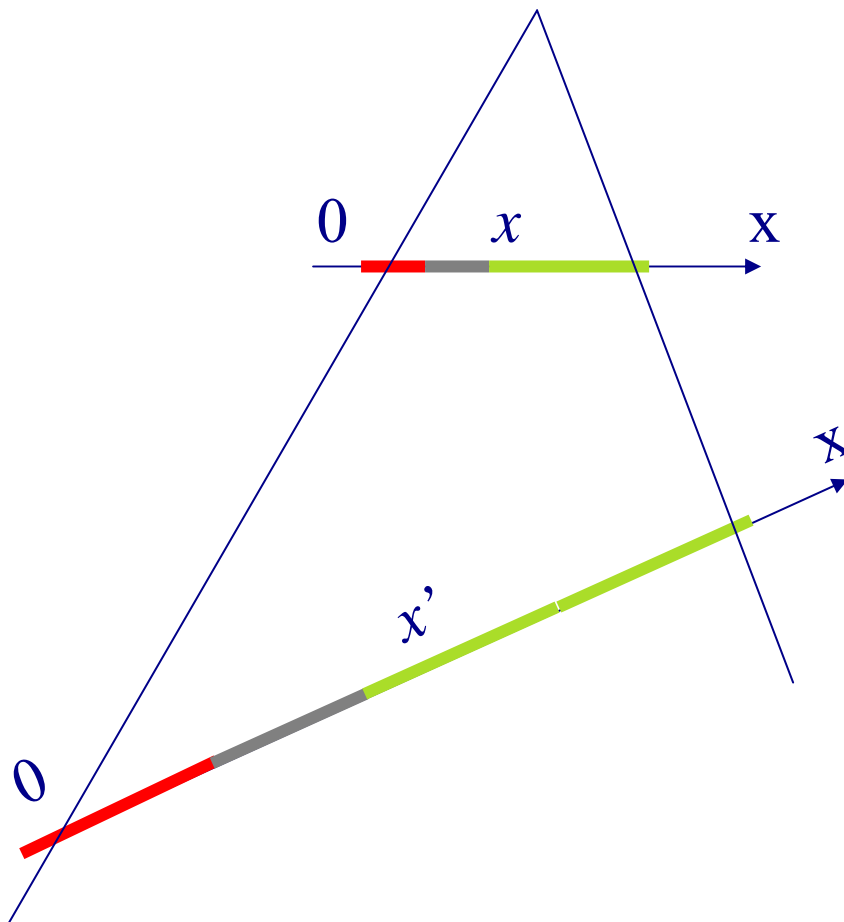
$$\begin{aligned}\|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2 &= (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (x'_i - \hat{x}'_i)^2 + (y'_i - \hat{y}'_i)^2 \\ &= d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2\end{aligned}$$

$$d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = d_{\perp}(\mathbf{X}_i, \mathcal{V}_H)^2$$

1D Example

- In the following we take a look at a 1D example for the different cost functions presented

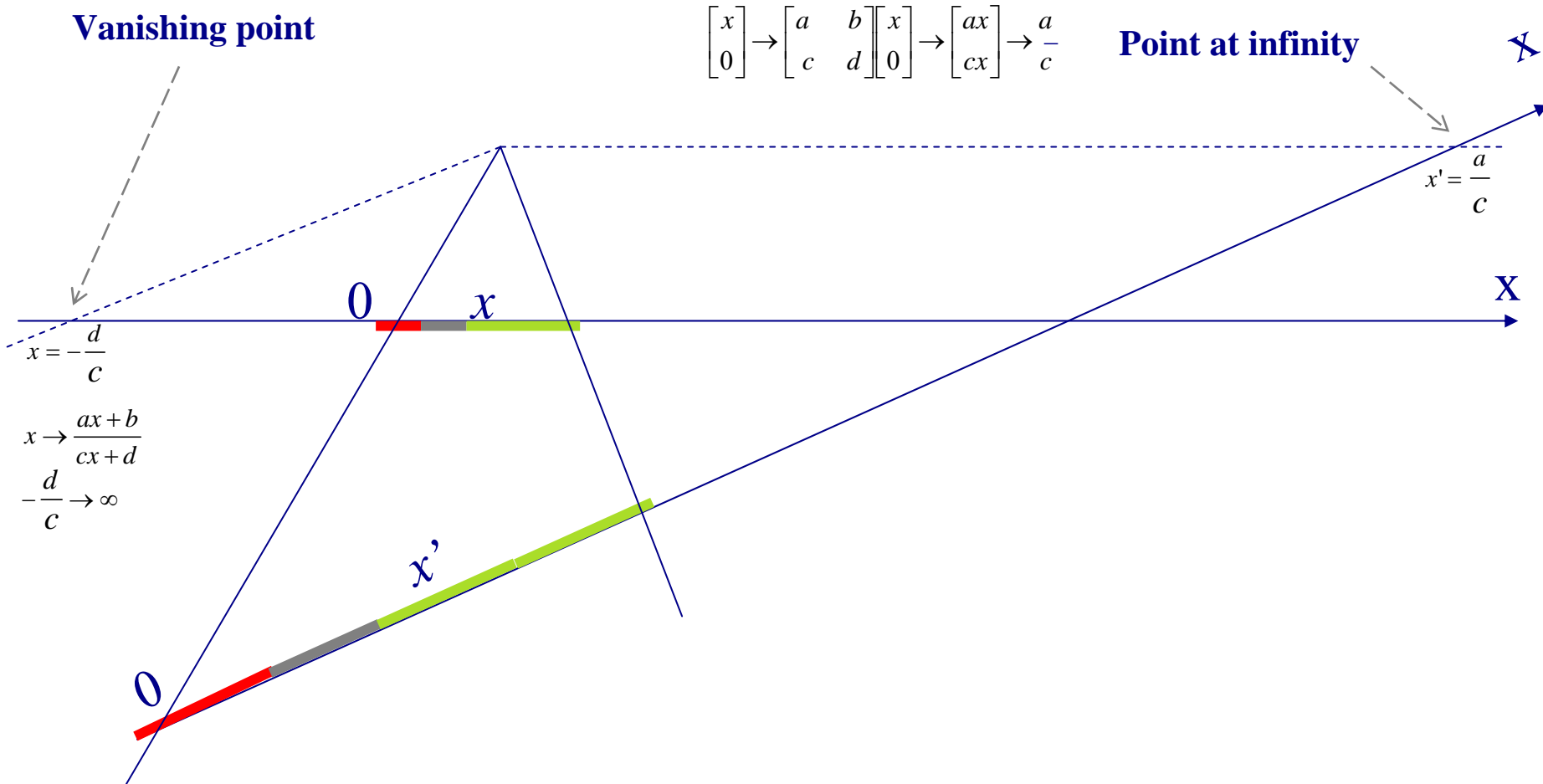
1D Projective Transformation (P_1)



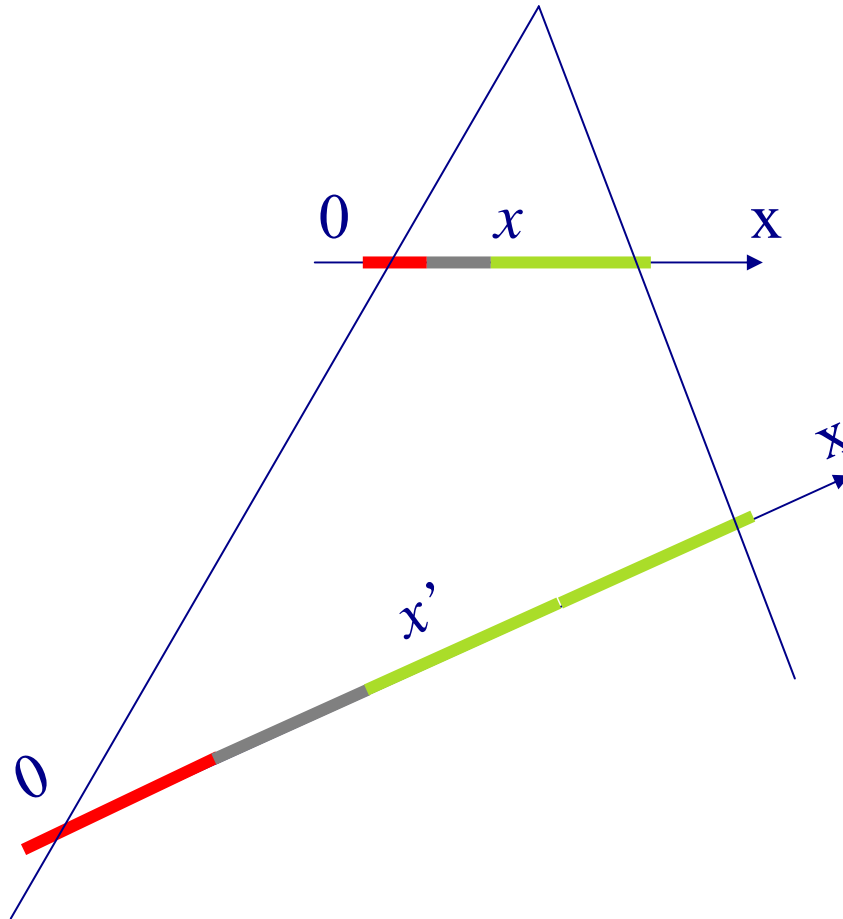
$$x \rightarrow x' = \frac{ax + b}{cx + d}$$

$$\begin{bmatrix} x \\ 1 \end{bmatrix} \rightarrow \lambda \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

1D Projective Transformation (P_1)



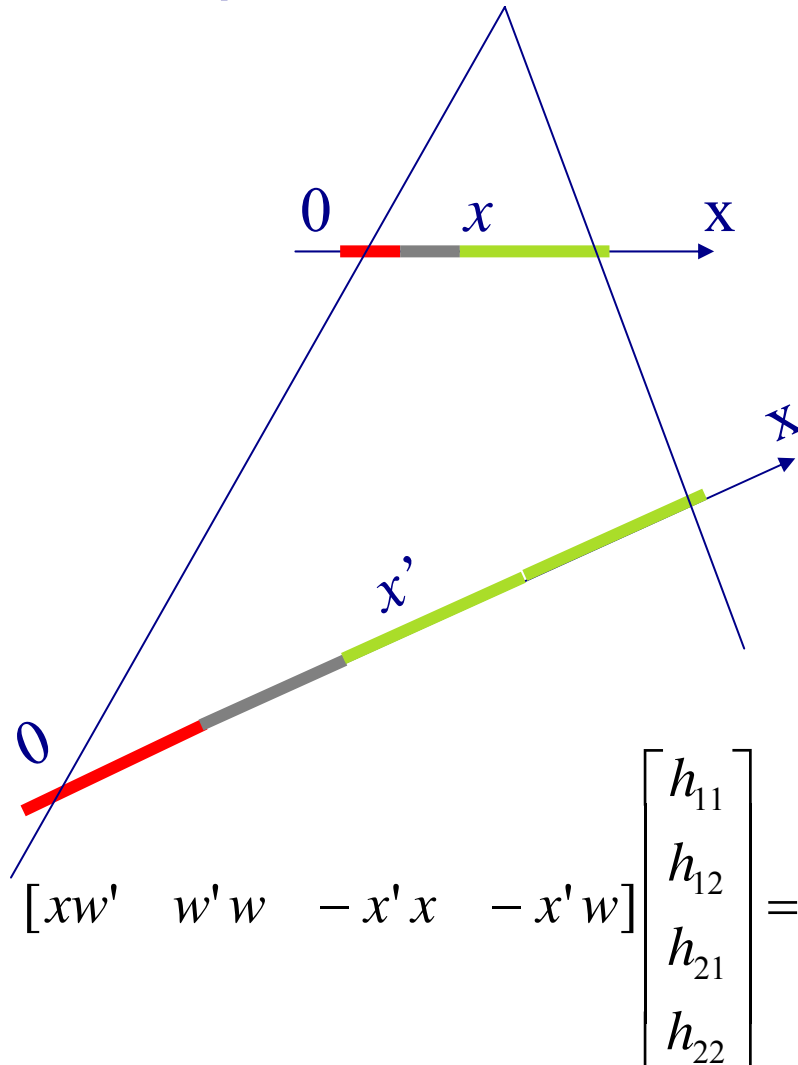
1D Projective Transformation (P_1)



$$\frac{x}{w} \rightarrow \frac{x'}{w'} = \frac{h_{11}x + h_{12}}{h_{21}x + h_{22}}$$

$$\begin{bmatrix} x \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}$$

1D Example – DLT



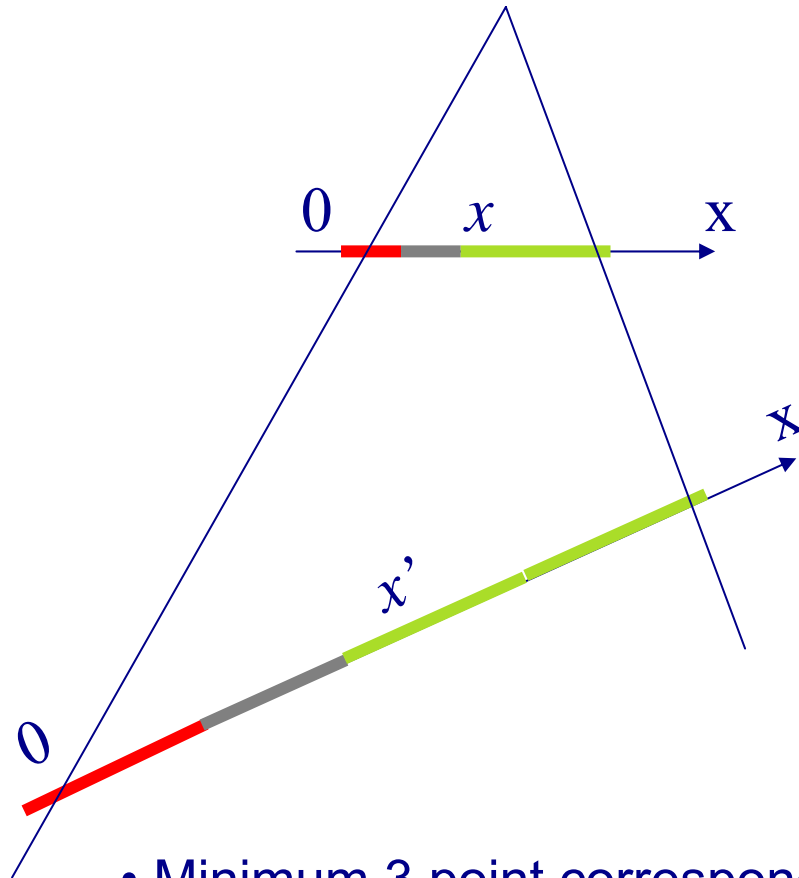
$$\begin{bmatrix} x \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}$$

$$\frac{x}{w} \rightarrow \frac{x'}{w'} = \frac{h_{11}x + h_{12}w}{h_{21}x + h_{22}w}$$

$$w'(h_{11}x + h_{12}w) - x'(h_{21}x + h_{22}w) = 0$$

$$A_i h = 0, (i = 1..n) \rightarrow Ah = 0$$

1D Example – DLT



$$[xw' \quad w'w \quad -x'x \quad -x'w] \begin{bmatrix} h_{11} \\ h_{12} \\ h_{21} \\ h_{22} \end{bmatrix} = 0$$

$$A_i h = 0, (i = 1..n) \rightarrow Ah = 0$$

- Minimum 3 point correspondences (unique solution)
- More points: $A^t A = UDV^t \rightarrow V_n$ is the solution

1D Example – Algebraic Distance

DLT minimizes $\|Ah\|$

$e = Ah$ residual vector

e_i partial vector for each $(x_i \leftrightarrow x_i')$
algebraic error vector

$$d_{\text{alg}}(x_i', Hx_i)^2 = \|e_i\|^2 = \left\| \begin{bmatrix} x_i' w_i' & w_i' w & -x_i' x_i & x_i' w \end{bmatrix} h \right\|^2$$

1D Example – Algebraic Distance

$$d_{\text{alg}}(\mathbf{x}'_i, H\mathbf{x}_i)^2 = \|e_i\|^2 = \left\| \begin{bmatrix} \mathbf{x}'_i \mathbf{w}_i' & \mathbf{w}_i' \mathbf{w} & -\mathbf{x}'_i \mathbf{x}_i & \mathbf{x}'_i \mathbf{w} \end{bmatrix} \mathbf{h} \right\|^2$$

$$\sum_i d_{\text{alg}}(\mathbf{x}'_i, H\mathbf{x}_i)^2 = \sum_i \|e_i\|^2 = \|\mathbf{A}\mathbf{h}\|^2 = \|\mathbf{e}\|^2$$

Not geometrically/statistically meaningful, but given good normalization it works fine and is very fast (use for initialization)

1D Example – Geometric Distance

Error in one image

$$\hat{H} = \operatorname{argmin}_H \sum_i d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2$$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates

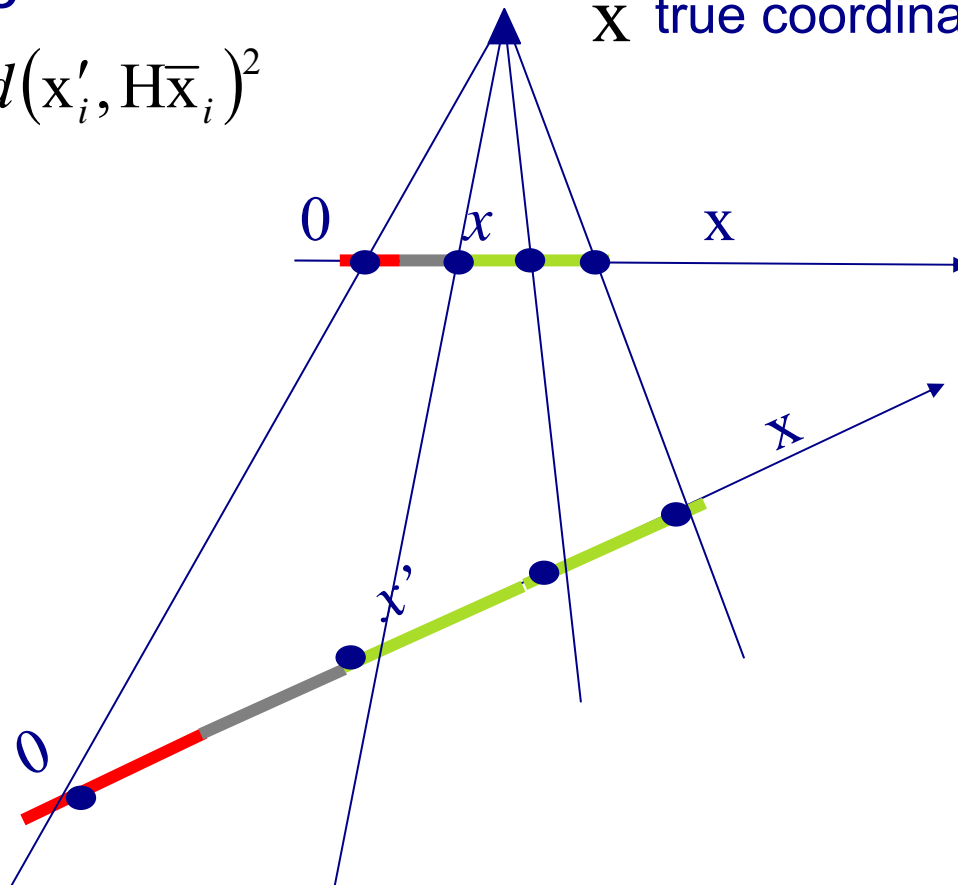


1D Example – Geometric Distance

Error in one image

$$\hat{H} = \operatorname{argmin}_H \sum_i d(x'_i, H\bar{x}_i)^2$$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates

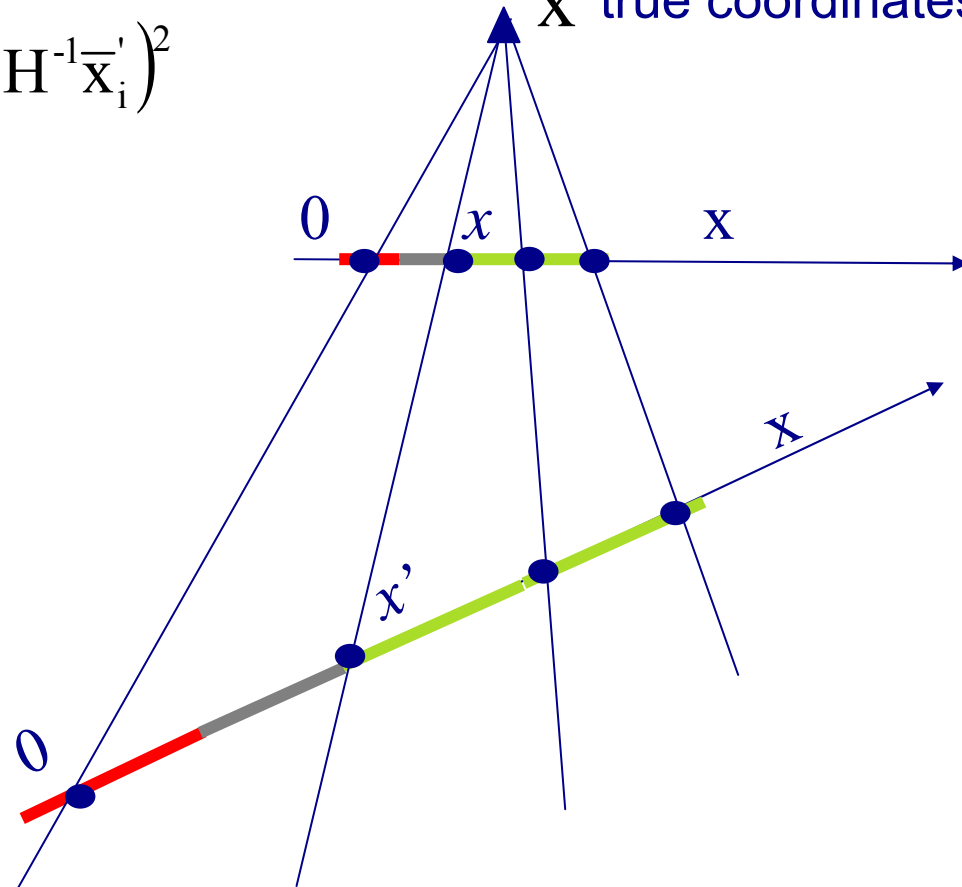


1D Example – Geometric Distance

Error in one image

$$\hat{H} = \operatorname{argmin}_H \sum_i d(x_i, H^{-1}\bar{x}_i)^2$$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates



Comparison of Geometric and Algebraic Distances

Error in one image

$$\mathbf{x}'_i = (x'_i, w'_i)^\top \quad \hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{w}'_i)^\top = H\bar{\mathbf{x}}$$

$$A_i \mathbf{h} = e_i = (w'_i \hat{x}'_i - x'_i \hat{w}'_i)$$

$$d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = (w'_i \hat{x}'_i - x'_i \hat{w}'_i)^2$$

$$\begin{aligned} d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 &= (\hat{x}'_i / \hat{w}'_i - x'_i / w'_i)^2 \\ &= d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 / w'_i \hat{w}'_i \end{aligned}$$

Comparison of Geometric and Algebraic Distances

Error in one image

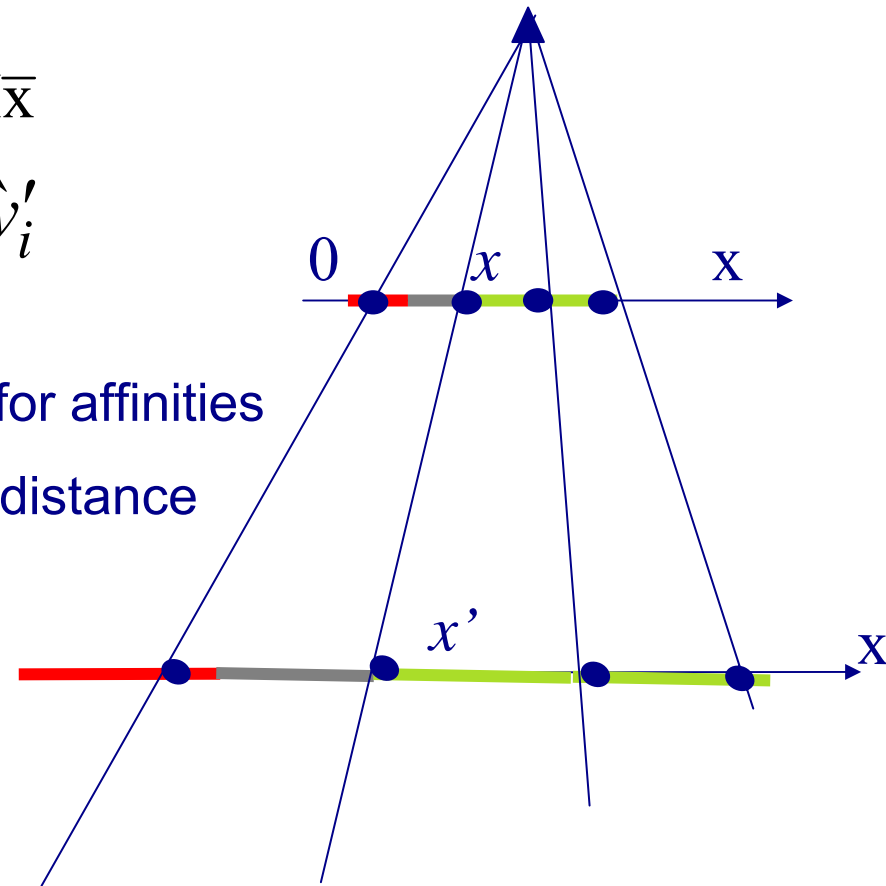
$$\mathbf{x}'_i = (x'_i, w'_i)^\top \quad \hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{w}'_i)^\top = H\bar{\mathbf{x}}$$

$$d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 / w'_i \hat{w}'_i$$

$w'_i = 1$ typical, but not $\hat{w}'_i = h_2 x_i$, except for affinities

For affinities DLT can minimize geometric distance

$$H_A \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ 1 \end{bmatrix}$$

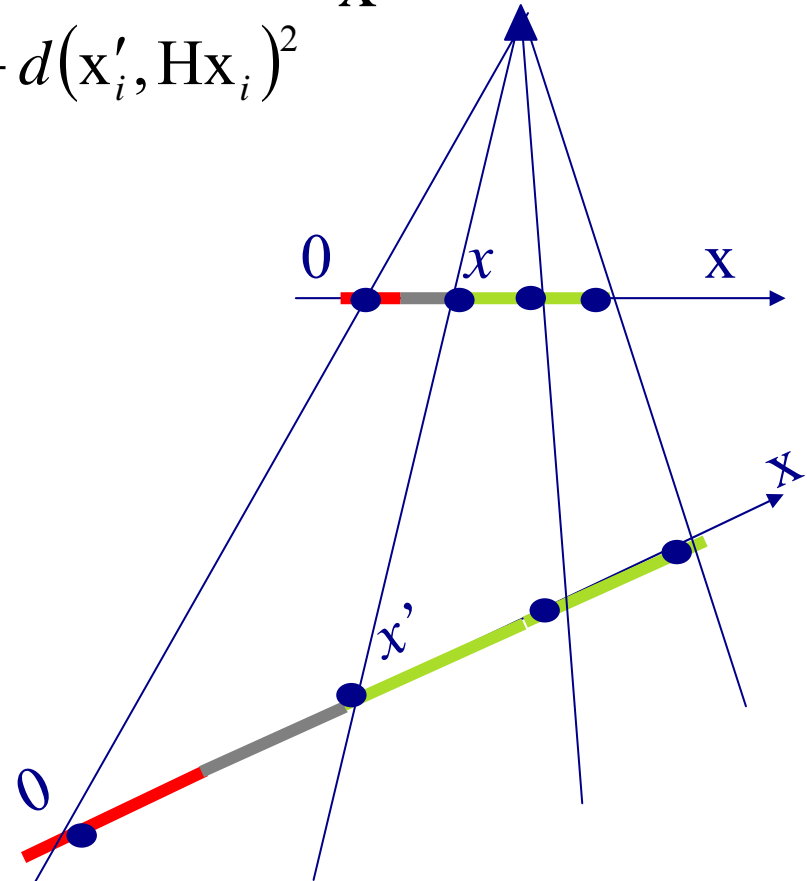


1D Example – Geometric Distance

Symmetric transfer error

$$\hat{H} = \operatorname{argmin}_H \sum_i d(x_i, H^{-1}x'_i)^2 + d(x'_i, Hx_i)^2$$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates



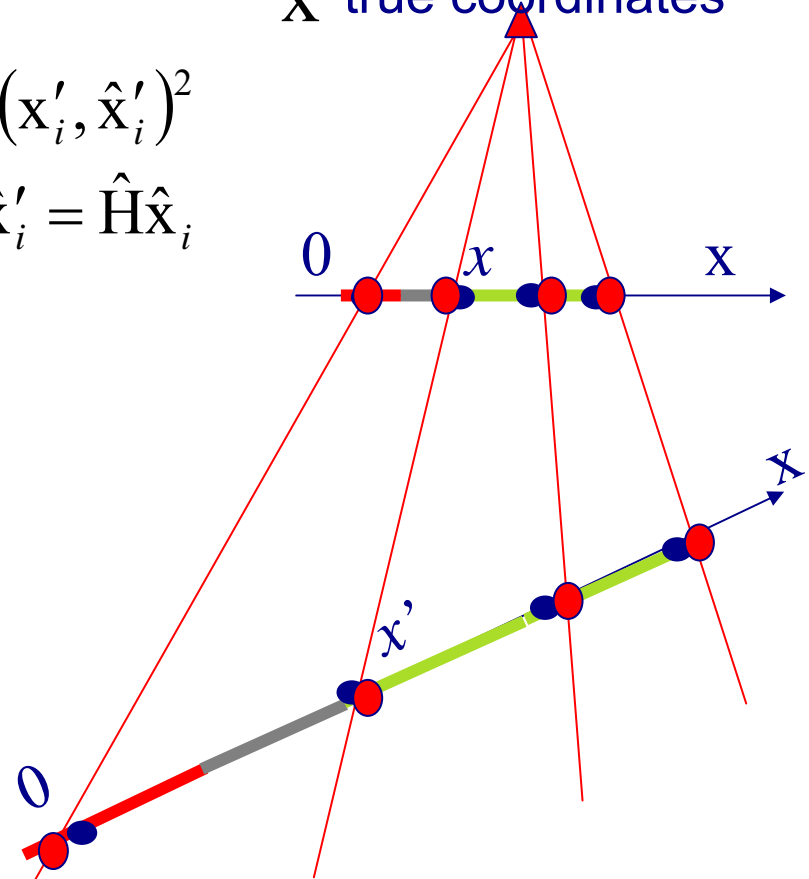
1D Example – Geometric Distance

Reprojection Error

$$(\hat{H}, \hat{x}_i, \hat{x}'_i) = \underset{H, \hat{x}_i, \hat{x}'_i}{\operatorname{argmin}} \sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}'_i)^2$$

subject to $\hat{x}'_i = \hat{H}\hat{x}_i$

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates



1D Example – Geometric Distance

Reprojection error

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates

$$(\hat{H}, \hat{x}_i, \hat{x}'_i) = \underset{H, \hat{x}_i, \hat{x}'_i}{\operatorname{argmin}} \sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}'_i)^2 \quad \text{subject to} \quad \hat{x}'_i = \hat{H} \hat{x}_i$$

$$[xw' \quad w'w \quad -x'x \quad -x'w] \begin{bmatrix} h_{11} \\ h_{12} \\ h_{21} \\ h_{22} \end{bmatrix} = 0$$

$$\hat{h}_{11} \hat{x} + \hat{h}_{12} - \hat{h}_{21} \hat{x}' - \hat{h}_{22} \hat{x}' = 0$$

1D Example – Geometric Distance

\mathbf{X} measured coordinates
 $\hat{\mathbf{X}}$ estimated coordinates
 $\bar{\mathbf{X}}$ true coordinates

Error in one image

$$\hat{H} = \operatorname{argmin}_H \sum_i d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2$$

Symmetric transfer error

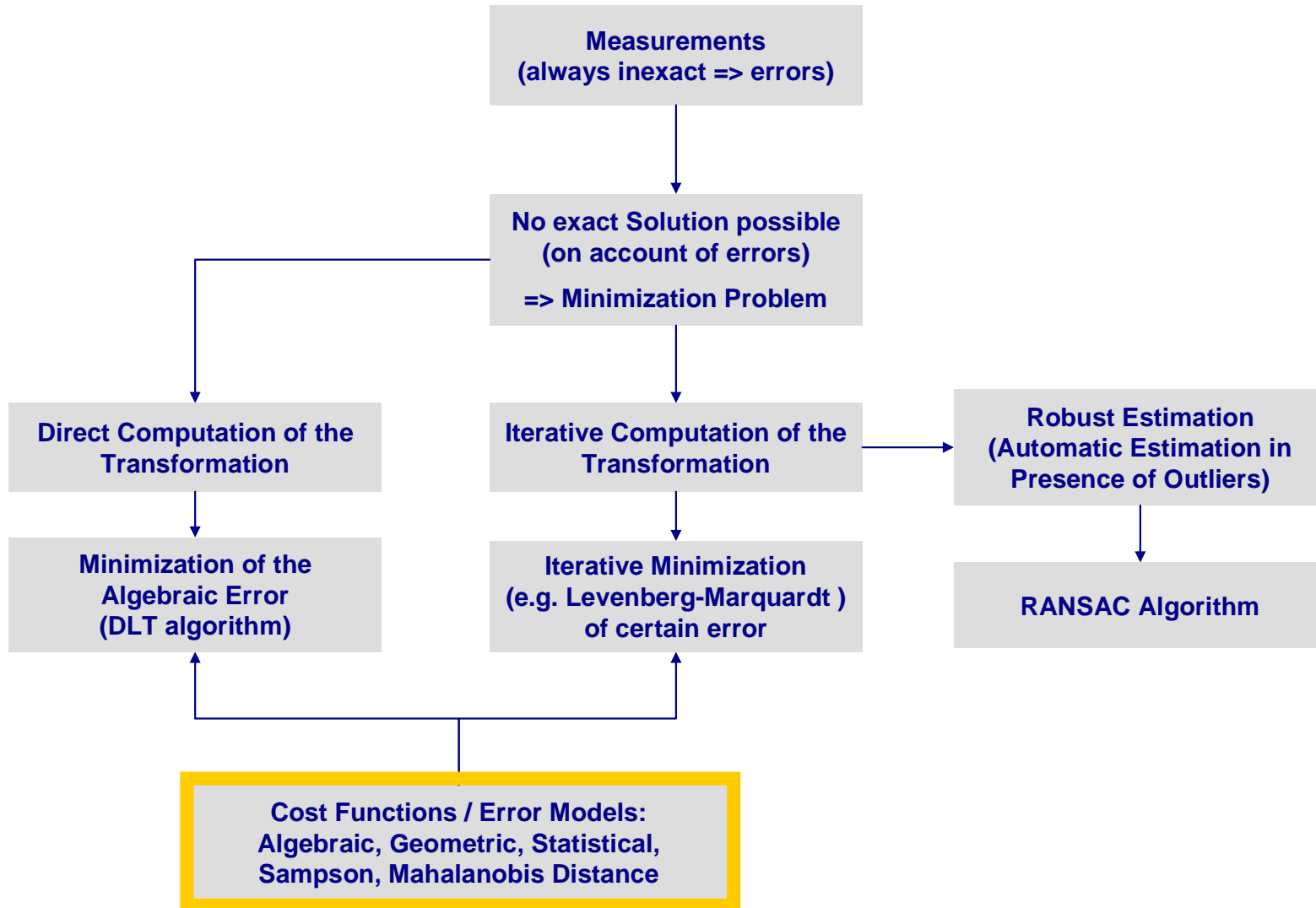
$$\hat{H} = \operatorname{argmin}_H \sum_i d(\mathbf{x}_i, H^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, H\mathbf{x}_i)^2$$

Reprojection error

$$(\hat{H}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i) = \operatorname{argmin}_{H, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

subject to $\hat{\mathbf{x}}'_i = \hat{H}\hat{\mathbf{x}}_i$

Cost Functions – Part II



Cost Functions – Part II

- Sampson Error
 - combines the advantages of the algebraic error and the geometric error
 - linear approximation to the geometrical error

- Statistical Cost Functions
 - Maximum Likelihood Estimate (MLE)
 - Mahalanobis Distance

Sampson Error

(between algebraic and geometric error)

Vector \hat{X} that minimizes the geometric error $\|X - \hat{X}\|^2$ is the closest point on the variety V_H to the measurement X

Sampson error: 1st order approximation of \hat{X}

$$Ah = C_H(X) = 0$$

$$C_H(X + \delta_X) = C_H(X) + \frac{\partial C_H}{\partial X} \delta_X \quad \delta_X = \hat{X} - X \quad C_H(\hat{X}) = 0$$

$$C_H(X) + \frac{\partial C_H}{\partial X} \delta_X = 0 \quad J\delta_X = -e \quad \text{with } J = \frac{\partial C_H}{\partial X}$$

Find the vector δ_X that minimizes $\|\delta_X\|$ subject to $J\delta_X = -e$

Sampson Error

Find the vector δ_X that minimizes $\|\delta_X\|$ subject to $J\delta_X = -e$

Use Lagrange multipliers:

$$\text{minimize} \quad \delta_X^T \delta_X - 2\lambda(J\delta_X + e) = 0$$

$$\begin{aligned} \text{derivatives} \quad 2\delta_X - 2\lambda^T J &= 0^T & \Rightarrow \delta_X &= J^T \lambda \\ 2(J\delta_X + e) &= 0 & \Rightarrow JJ^T \lambda + e &= 0 \\ & & \Rightarrow \lambda &= -(JJ^T)^{-1} e \\ & & \Rightarrow \delta_X &= -J^T (JJ^T)^{-1} e \end{aligned}$$

$$\hat{X} = X + \delta_X \quad \|\delta_X\|^2 = \delta_X^T \delta_X = e^T (JJ^T)^{-1} e$$

Sampson Error

Vector \hat{X} that minimizes the geometric error $\|X - \hat{X}\|^2$ is the closest point on the variety V_H to the measurement X

Sampson error: 1st order approximation of \hat{X}

$$Ah = C_H(X) = 0$$

$$C_H(X + \delta_X) = C_H(X) + \frac{\partial C_H}{\partial X} \delta_X \quad \delta_X = \hat{X} - X \quad C_H(\hat{X}) = 0$$

$$C_H(X) + \frac{\partial C_H}{\partial X} \delta_X = 0 \quad J\delta_X = -e$$

Find the vector δ_X that minimizes $\|\delta_X\|$ subject to $J\delta_X = -e$

$$\|\delta_X\|^2 = \delta_X^T \delta_X = e^T (JJ^T)^{-1} e \quad (\text{Sampson error})$$

Sampson Approximation

$$\|\delta_X\|^2 = e^T (JJ^T)^{-1} e$$

A few points

- (i) For a 2D homography $X=(x,y,x',y')$
- (ii) $e = C_H(X)$ is the algebraic error vector
- (iii) $J = \frac{\partial C_H}{\partial X}$ is a 2x4 matrix, e.g. $J_{11} = \partial(-w'_i x_i^T h^2 + y'_i x_i^T h^3) / \partial x = -w'_i h_{21} + y'_i h_{31}$
- (iv) Similar to algebraic error $\|e\|^2 = e^T e$ in fact, same as Mahalanobis distance
- (v) Sampson error independent of linear reparametrization (cancels out in between e and J)
- (vi) Must be summed for all points $\|e\|_{JJ^T}^2$
- (vii) Close to geometric error, but much fewer unknowns $\sum e^T (JJ^T)^{-1} e$

Statistical Cost Function and Maximum Likelihood Estimation

- Optimal cost function related to noise model
- Assume zero-mean isotropic Gaussian noise (assume outliers removed)

$$\Pr(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-d(\mathbf{x}, \bar{\mathbf{x}})^2 / (2\sigma^2)}$$

Error in one image

$$\Pr(\{\mathbf{x}'_i\} | H) = \prod_i \frac{1}{2\pi\sigma^2} e^{-d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2 / (2\sigma^2)}$$

$$\log \Pr(\{\mathbf{x}'_i\} | H) = -\frac{1}{2\sigma^2} \sum d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2 + \text{constant}$$

Maximum Likelihood Estimate

$$\sum d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2$$

Statistical Cost Function and Maximum Likelihood Estimation

- Optimal cost function related to noise model
- Assume zero-mean isotropic Gaussian noise (assume outliers removed)

$$\Pr(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-d(\mathbf{x}, \bar{\mathbf{x}})^2 / (2\sigma^2)}$$

Error in both images

$$\Pr(\{\mathbf{x}'_i\} | H) = \prod_i \frac{1}{2\pi\sigma^2} e^{-\left(d(\mathbf{x}_i, \bar{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, H\bar{\mathbf{x}}_i)^2\right) / (2\sigma^2)}$$

Maximum Likelihood Estimate

$$\sum d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

Mahalanobis Distance

- General Gaussian case

Measurement \mathbf{X} with covariance matrix Σ

$$\|\mathbf{X} - \bar{\mathbf{X}}\|_{\Sigma}^2 = (\mathbf{X} - \bar{\mathbf{X}})^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$

Error in two images (independent)

$$\|\mathbf{X} - \bar{\mathbf{X}}\|_{\Sigma}^2 + \|\mathbf{X}' - \bar{\mathbf{X}}'\|_{\Sigma'}^2$$

Varying covariances

$$\sum_i \|\mathbf{X}_i - \bar{\mathbf{X}}_i\|_{\Sigma_i}^2 + \|\mathbf{X}'_i - \bar{\mathbf{X}}'_i\|_{\Sigma'_i}^2$$

Cost Functions Summary

Algebraic Error	Geometric Error	Statistical Error
Direct Linear Transformation (DLT)	Simple and Symmetric Transfer Error, Reprojection Error	Maximum Likelihood Estimation (MLE), Mahalanobis Distance
<ul style="list-style-type: none"> + fast (direct computation) + simple implementation 	<ul style="list-style-type: none"> + geometric interpretation 	<ul style="list-style-type: none"> + statistical interpretation given
<ul style="list-style-type: none"> - no geometric interpretation 	<ul style="list-style-type: none"> - slow computation 	<ul style="list-style-type: none"> - slow computation
Sampson Error		leads to same results as the geometric error
<ul style="list-style-type: none"> • combines the advantages of algebraic and geometric error • linear approximation to geometric error 		
<ul style="list-style-type: none"> + fast computation + geometric interpretation 		

Further Questions

- How do the Cost Functions Minimization Algorithms behave in respect to change of coordinate system and scaling
- Geometric Minimization invariant
- DLT not invariant to change of coordinate system and scaling
⇒ normalization needed !!!

Changing Coordinate Systems and Scaling

- Change of coordinate systems is an Euclidean transformation T (rotation and translation) plus scaling: similarity transformation

$$y = \begin{pmatrix} sR & t \\ 0 & 1 \end{pmatrix} x = Tx$$

- What is the behavior of algebraic and geometric error when changing coordinate systems and scaling?

$$d_{alg}(x', Hx) = \|Ah\|$$



$$d_{alg}(y', \tilde{H}y) = ?$$

$$d_{euclid}(x', Hx)$$



$$d_{euclid}(y', \tilde{H}y) = ?$$

Invariance of Geometric Error

- Model change of coordinate system in images with two transformations, T , and $T' \rightarrow y = Tx$ and $y' = T'x'$
- Define $H_T = T'HT^{-1}$
- Geometric error is invariant:

$$d_{euclid}(y', H_T y) = d_{euclid}(T' x', T' HT^{-1}Tx) = d_{euclid}(T' x', T' Hx)$$

Euclidean distance only changes in scale factor under a similarity transformation:

$$d_{euclid}(T' x', T' Hx) = s \cdot d_{euclid}(x', Hx)$$

- Scaling changes the geometric error, but isotropic \rightarrow **Minimizing** geometric error is invariant to similarity transformations

Non-Invariance of Algebraic Error

- Constraint $\|H\|=1$ of DLT algorithm makes Algebraic Error non-invariant

Minimize $\sum_i d_{\text{alg}}(x_i', Hx_i)^2$ subject to $\|H\| = 1$

⇔ Minimize $\sum_i d_{\text{alg}}(y_i', T'HT^{-1}y_i)^2$ subject to $\|H\| = 1$

≠ Minimize $\sum_i d_{\text{alg}}(y_i', T'HT^{-1}y_i)^2$ subject to $\|T'HT^{-1}\| = 1$

- Make DLT invariant to such changes:

Normalization

Normalizing Transformations

- Carry out minimization of DLT in a *fixed canonical coordinate frame* → DLT algorithm invariant to similarity transformations

Transform all points to canonical frame:

- The points are translated so that their *centroid* is in the origin
- The points are then scaled so that the average distance from the origin is equal to $\sqrt{2}$.
- This transformation is applied to each of the two images independently.

Why is normalization important?

$$\begin{bmatrix}
 0 & 0 & 0 & -x'_i & -y'_i & -1 & y'_i x_i & y'_i y_i & y'_i \\
 x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i
 \end{bmatrix}
 \begin{pmatrix}
 h^1 \\
 h^2 \\
 h^3
 \end{pmatrix}
 = 0$$

$\sim 10^2 \quad \sim 10^2 \quad 1 \quad \sim 10^2 \quad \sim 10^2 \quad 1 \quad \sim 10^4 \quad \sim 10^4 \quad \sim 10^2$

orders of magnitude differ!

Monte Carlo simulation:



unnormalized



normalized

Normalized DLT algorithm

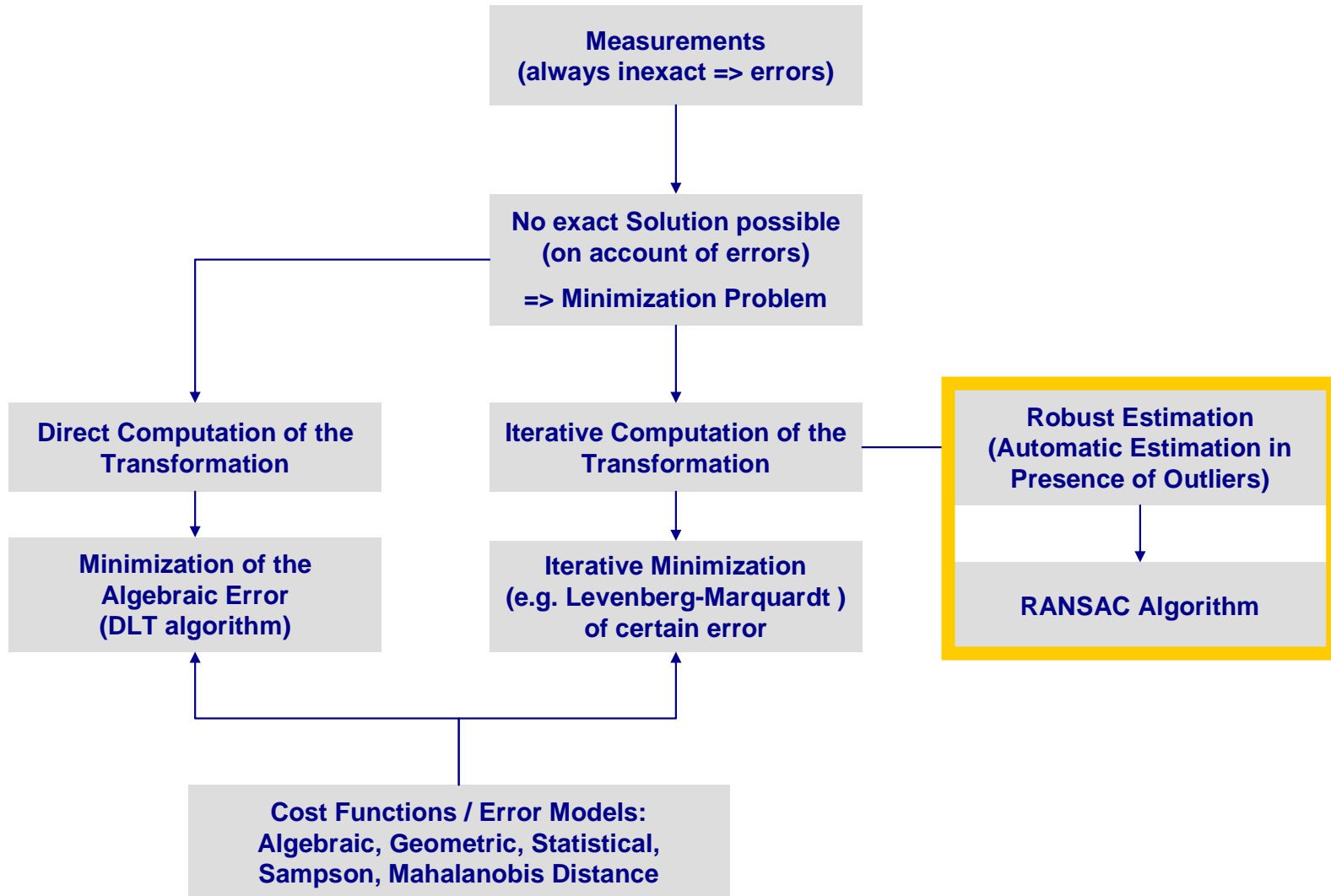
Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x'_i\}$, determine the 2D homography matrix H such that $x'_i = Hx_i$

Algorithm

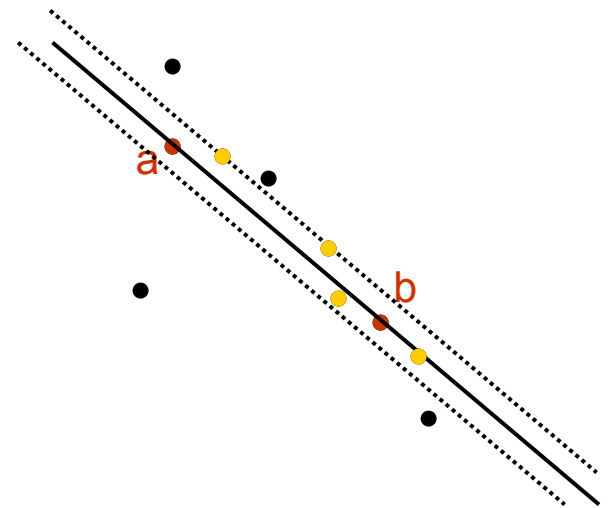
- (i) Normalize points $y_i = T_{\text{norm}} x_i, y'_i = T'_{\text{norm}} x'_i$
- (ii) Apply DLT algorithm to $y_i \leftrightarrow y'_i$,
- (iii) Denormalize solution $H = T_{\text{norm}}^{-1} \tilde{H} T_{\text{norm}}$

Robust Estimation



RANSAC Robust Estimation

- Given the putative correspondences RANSAC determines the set of the right ones
- RANSAC: RANdom SAMple Consensus (Fischer and Bolles)
- Idea for line fitting:
 1. choose randomly two points
 2. these define a line
 3. determine the support for this line (#points within certain distance)
 4. repeat (1-3) for a certain number of times
 5. choose the line with most support



Automatic Computation of H

Objective

Compute homography between two images

Algorithm

- (i) **Interest points:** Compute interest points in each image
- (ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure
- (iii) **RANSAC robust estimation:** Repeat for N samples
 - (a) Select 4 correspondences and compute H
 - (b) Calculate the distance d_{\perp} for each putative match
 - (c) Compute the number of inliers consistent with H ($d_{\perp} < t$)Choose H with most inliers
- (iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt
- (v) **Guided matching:** Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

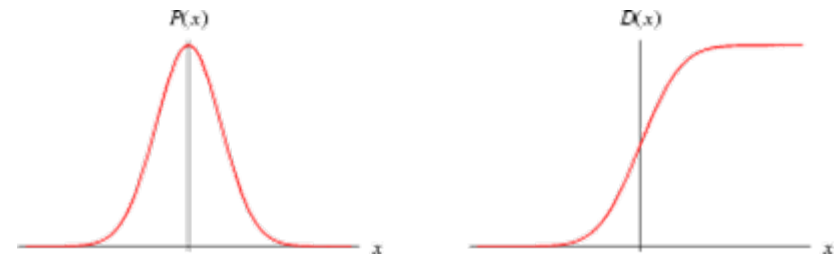
RANSAC (RANDOM Sample Consensus)

- Objective
 - Robust fit of a model to a data set S which contains outliers
- Algorithm
 1. Randomly select a sample of s data points from S and instantiate the model from this subset
 2. Determine the set of data points S_i which are *within a distance threshold t* of the model. The set S_i is the *consensus set* of the sample and defines the inliers of S .
 3. If the size of S_i (the number of inliers) is greater than some threshold T , re-estimate the model using all points in S_i and terminate.
 4. If the size of S_i is less than T , select a new subset and repeat the above.
 5. After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S .

RANSAC – What is the distance threshold?

- Choose distance threshold t such that a point is an inlier with a probability α

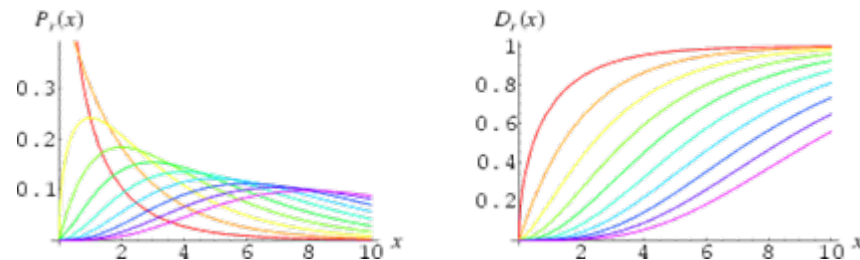
- If measurement error is Gaussian then the square of the point distance d^2 follows a χ^2 distribution



- The probability that d^2 is less than k^2 is given by the cumulative χ^2 distribution

$$F_m(k^2) = \int_0^{k^2} \chi_m^2(\xi) d\xi :$$

- inlier $d^2 < t^2$
- outlier $d^2 \geq t^2$ with $t^2 = F_m^{-1}(\alpha)\sigma^2$



RANSAC - How many samples?

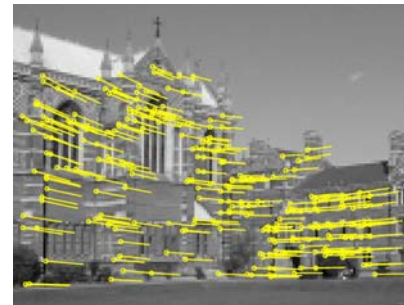
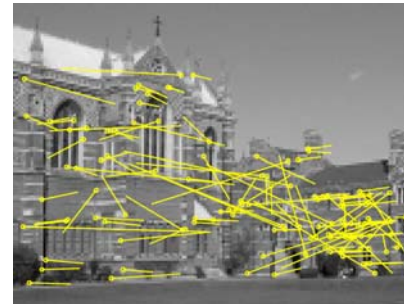
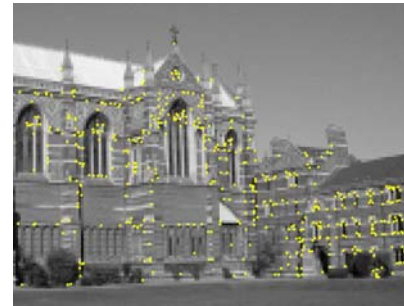
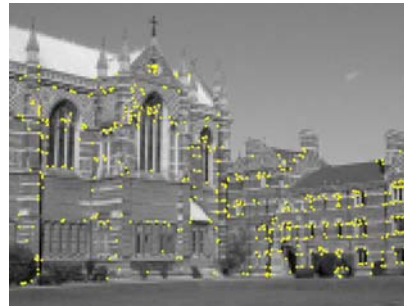
- Choose N (number of samples) to ensure with a probability p that at least one of the random samples of s points is free from outliers
- w is the probability that any selected point is an inlier $\rightarrow \varepsilon = 1 - w$ is the probability for an outlier
- At least N selections are required, where $(1 - w^s)^N = 1 - p$, so that

$$N = \log(1 - p) / \log(1 - (1 - \varepsilon)^s)$$

RANSAC - How many samples?

- Compute number of samples, N , adaptively:
 1. $N = \infty$, $\text{sample_count} = 0$
 2. While $N > \text{sample_count}$ Repeat
 - Choose a sample and count the number of inliers
 - Set $\varepsilon = 1 - (\text{number inliers})/(\text{total number of points})$
 - Set N from ε and equation of previous slide with $p = 0.99$
 - Increment the sample_count by 1
 3. Terminate

Automatic Estimation in Presence of Outliers



Objective

- given two images of the same scene
- compute automatically the homography between them

Outline: Automatic Computation of H

Objective

Compute homography between two images

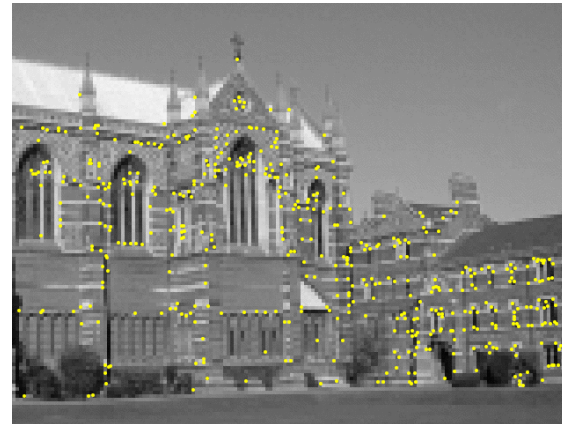
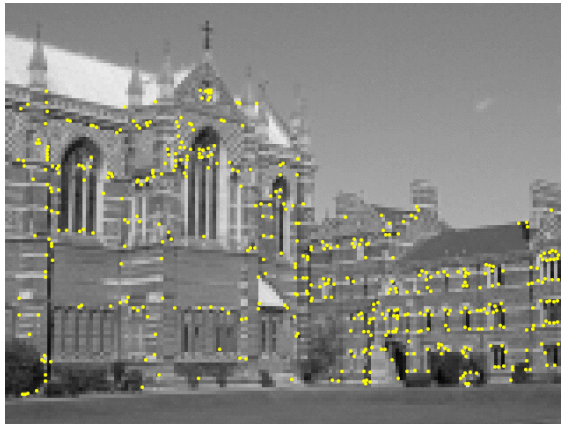
Algorithm

- (i) **Interest points:** Compute interest points in each image
- (ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure
- (iii) **RANSAC robust estimation:** Repeat for N samples
 - (a) Select 4 correspondences and compute H
 - (b) Calculate the distance d_{\perp} for each putative match
 - (c) Compute the number of inliers consistent with H ($d_{\perp} < t$)Choose H with most inliers
- (iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt
- (v) **Guided matching:** Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

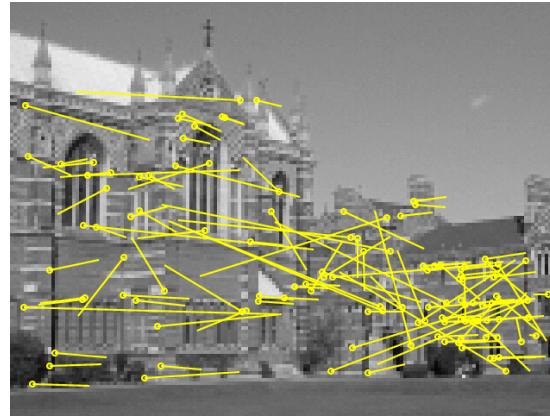
Interest Points

- Automatically compute interest points in each image
- Corner Detection
- Not all points of interest have correspondences in the other image



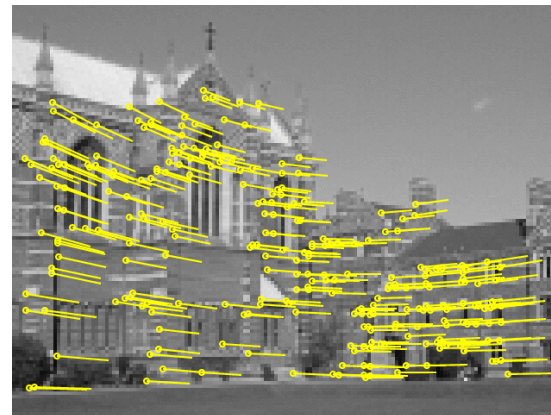
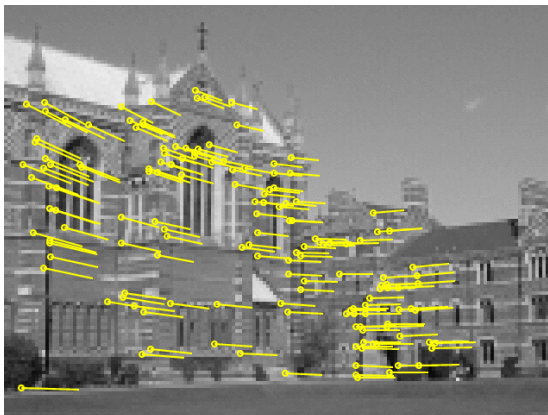
Putative Correspondences

- Compute a set of interest point matches based on some similarity measure
- Many correspondences are not the right ones yet



Iteration

- Now we have the “true” correspondences (inliers) by having applied RANSAC
- What’s next?
- Optimal estimation
re-estimate H from all inliers by minimizing ML cost function
(e.g. with Levenberg-Marquardt)
- Guided matching
Use the re-estimated H to determine more matches



Summary: Automatic Computation of H

Objective

Compute homography between two images

Algorithm

- (i) **Interest points:** Compute interest points in each image
- (ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure
- (iii) **RANSAC robust estimation:** Repeat for N samples
 - (a) Select 4 correspondences and compute H
 - (b) Calculate the distance d_{\perp} for each putative match
 - (c) Compute the number of inliers consistent with H ($d_{\perp} < t$)Choose H with most inliers
- (iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt
- (v) **Guided matching:** Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

RANSAC for homography H – Implementation and run details

Number of inliers	$1 - \epsilon$ (prob for inlier)	Adaptive N
6	2%	20,028,244
10	3%	2,595,658
44	16%	6,922
58	21%	2,291
73	26%	911
151	56%	43

