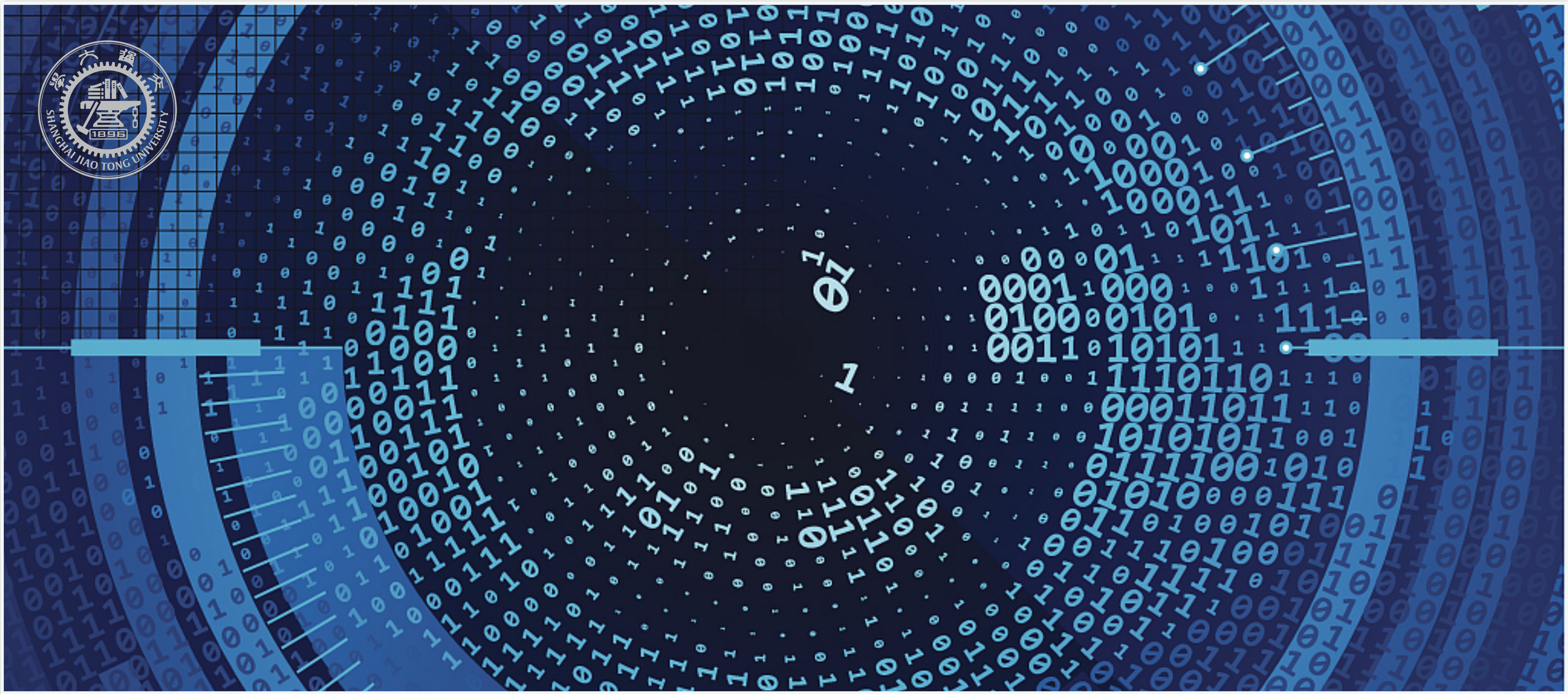


上海交通大学

计算机视觉

教师：赵旭
班级：AI4701

2024 春

3. 2D变换估计

2D变换的对比

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times3}$	8	straight lines	

齐次坐标的意义

- 使非线性映射（如透视投影变换），可以用线性矩阵方程表示
- 很好地表示“无穷远”点和线

主要内容

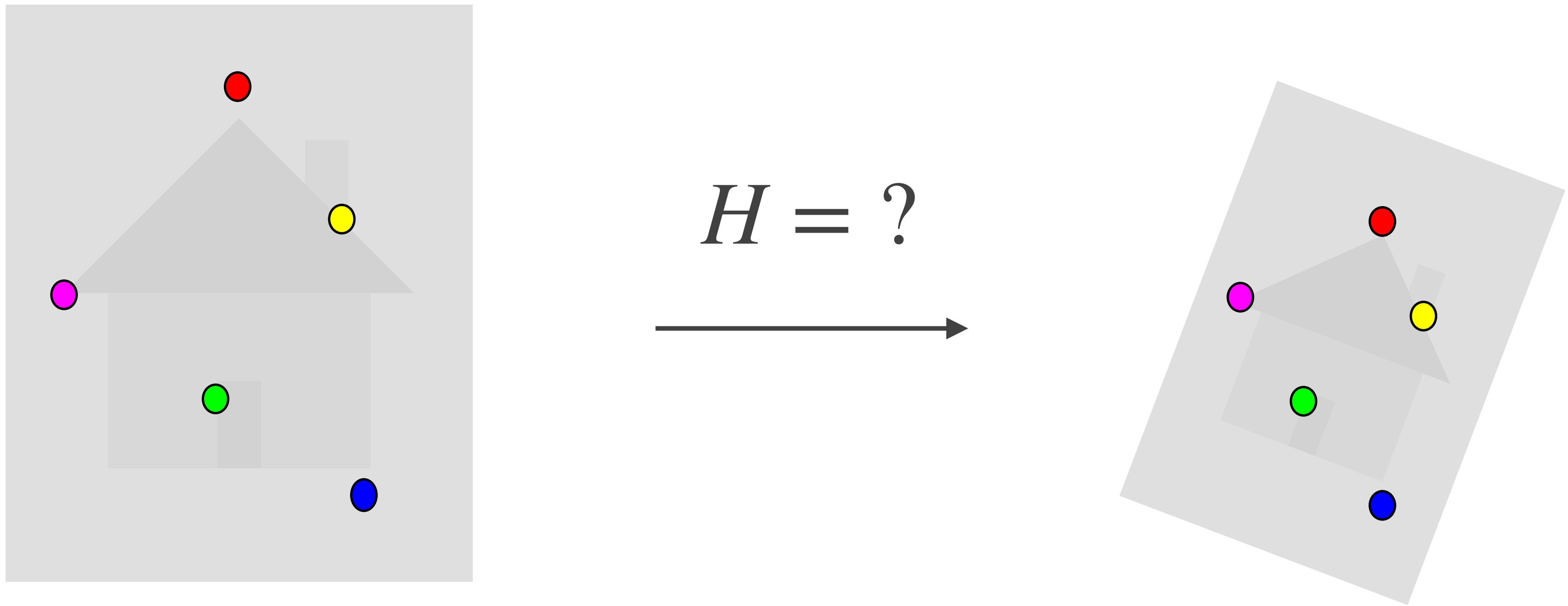
- 2D射影变换估计的直接线性变换算法（DLT）
- 2D射影变换估计的迭代近似算法

2D射影变换

- 射影变换（单应-Homography）：$\mathbb{P}^2 \mapsto \mathbb{P}^2$

$$\mathbf{x}' = \mathtt{H}_{\mathrm{P}}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix} \mathbf{x}$$

$$\mathbf{v} = (v_1, v_2)^{\mathsf{T}}$$



求解H需要多少点对应$\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$?

- $H: 3 \times 3$ 矩阵，9 个未知参数，8个自由度
- 1对点对应产生2个方程（2个约束）
- 最小配置解（minimal solution）：理论上讲，2张图像上的4组对应点，可以确定H.

直接线性变换算法（DLT）

- $\mathbf{x}_i{}' = H\mathbf{x}_i$, $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$
- $\mathbf{x}_i{}'$ 和 $H\mathbf{x}_i$ 之间有尺度问题，方向相同，但可能大小不一样
- 因此，更合适的方程可表达为：
  - $\mathbf{x}_i' \times H\mathbf{x}_i = 0$
- $A_i\mathbf{h} = 0$
  - A_i 为2×9 或者3×9矩阵

直接线性变换算法（DLT）

- $A\mathbf{h}=0$
- 平凡解$\mathbf{h}=0$不是我们想要的解
- 由于A的秩为8（4个点，2*4=8，3*4=12，A为12*9，或8*9矩阵）
    - 解存在尺度因子问题，可设定约束：$\|\mathbf{h}\|=1$

直接线性变换算法（DLT）

- 超定解：当点对应大于4对时
  - 优化问题：$\min \|A\mathbf{h}\|$ 服从约束：$\|\mathbf{h}\| = 1$
- 通过SVD分解求解

直接线性变换算法（DLT）

- 目标：给定$n \geq 4$ 2D-2D点对应$\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$，确定2D单应矩阵$H$，使得：$\mathbf{x}_i' = H\mathbf{x}_i$
- 算法：

（1） 对于每一对对应点$\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$，计算矩阵$A_i$ (前2行)

（2）将n个2×9的矩阵A_i组合成一个$2n \times 9$的矩阵A

（3）将A进行SVD分解，对应于最小特征值的单位特征向量便是解$\mathbf{h}$

直接线性变换算法（DLT）

- 退化情况：计算最小配置解的4组点对应中有3点共线
  - 不存在唯一解

归一化直接线性变换算法（DLT）

- 为什么需要归一化DLT?
  - DLT对坐标系的选择具有一定敏感度：有些坐标系比另外的其他坐标系更适合计算单应
  - DLT对相似变换具有不变性
  - 归一化后对尺度和坐标原点的选择具有不变性，预先消除了坐标变换的影响
  - 数据归一化在DLT中是实质性的，非可有可无

归一化直接线性变换算法（DLT）

- 目标：给定$n \geq 4$ 2D-2D点对应$\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$，确定2D单应矩阵$H$，使得：$\mathbf{x}_i' = H\mathbf{x}_i$
- 算法：

（1）归一化$\mathbf{x}$:计算一个只包含位移和缩放的相似变换T，将点$\mathbf{x}$变到新的点集，使该集合的中心位于原点，并且它们到原点的平均距离是$\sqrt{2}$

（2）归一化$\mathbf{x}'$: 如上，计算相似变换T'

（3）常规DLT得到单应$\bar{H}$

（4）解除归一化：令$H = T'^{-1}\bar{H}T$

迭代优化算法

- 为什么需要引入迭代优化近似算法?
  - 不存在完美的测量，总存在测量误差
  - 通过最小化误差函数（代价函数），从统计意义上最大程度减小误差

代价函数

- 代数距离

$$d_{\text{alg}}(\mathbf{x}_i', \mathtt{H}\mathbf{x}_i)^2 = \|\boldsymbol{\epsilon}_i\|^2 = \left\| \begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \end{bmatrix} \mathbf{h} \right\|^2 .$$

$$\sum_i d_{\text{alg}}(\mathbf{x}_i', \mathtt{H}\mathbf{x}_i)^2 = \sum_i \|\boldsymbol{\epsilon}_i\|^2 = \|\mathtt{A}\mathbf{h}\|^2 = \|\boldsymbol{\epsilon}\|^2.$$

  - 优点：线性解，计算开销小
  - 缺点：精度低，直观几何意义不显著。可以作为其他非线性算法的初始解

代价函数

- 几何距离

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

$$\sum_i d(\mathbf{x}_i, \mathtt{H}^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, \mathtt{H}\mathbf{x}_i)^2.$$

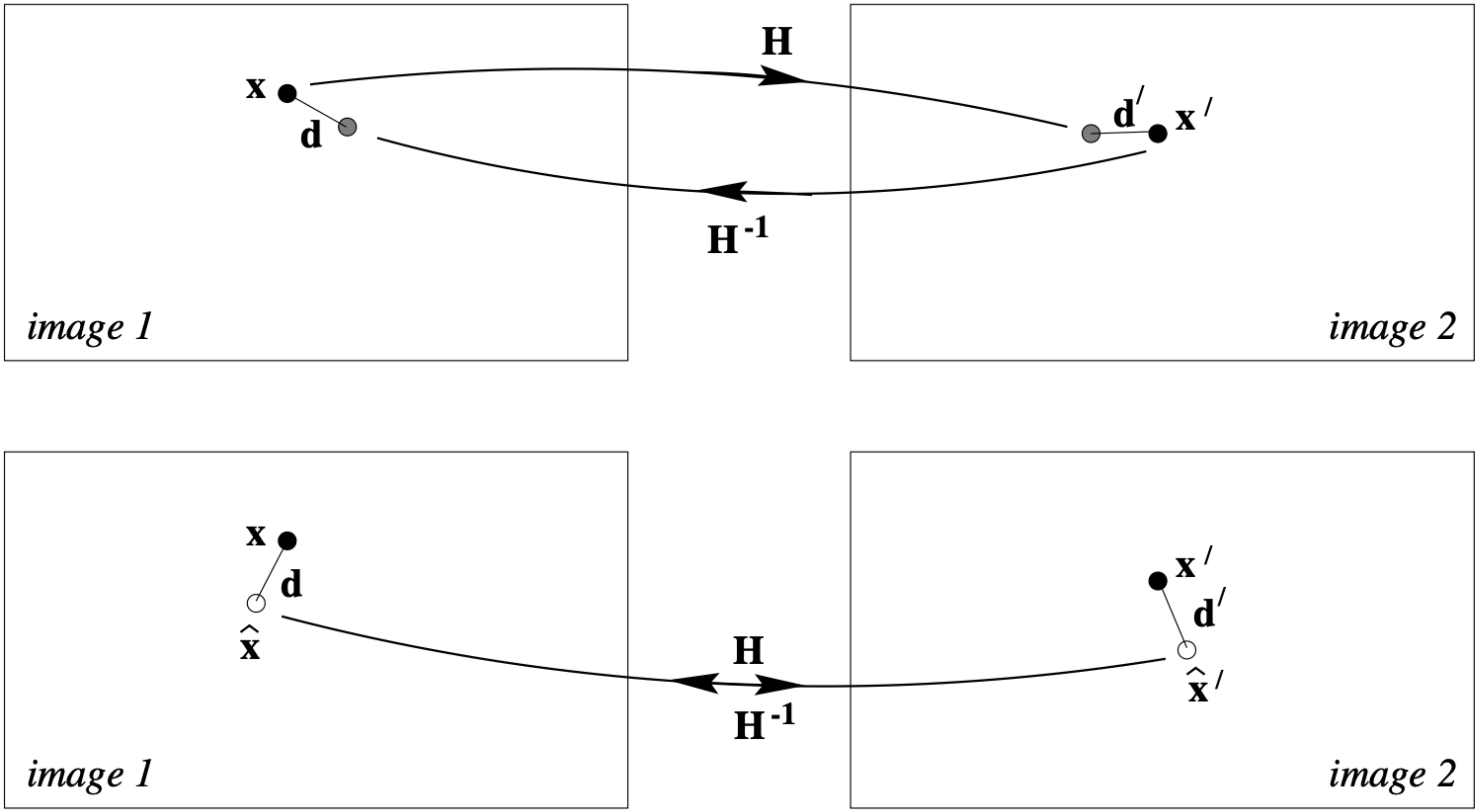
  - 最小化欧氏距离
  - 关注转移误差，具有直观几何意义

代价函数

- 重投影误差

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad \text{subject to } \hat{\mathbf{x}}'_i = \hat{\mathrm{H}}\hat{\mathbf{x}}_i \ \ \forall i.$$

- 同时优化$\mathbf{x}$和H



H
x
d
d′
x′
H⁻¹
image 1
image 2
x
d
x̂
H
H⁻¹
x′
d′
x̂′
image 1
image 2

迭代优化算法：黄金标准算法

Objective

Given $n > 4$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the Maximum Likelihood estimate $\hat{\mathrm{H}}$ of the homography mapping between the images.

The MLE involves also solving for a set of subsidiary points $\{\hat{\mathbf{x}}_i\}$, which minimize

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

迭代优化算法：黄金标准算法

Algorithm

(i) **Initialization**: Compute an initial estimate of $\hat{\mathtt{H}}$ to provide a starting point for the geometric minimization. For example, use the linear normalized DLT algorithm 4.2, or use RANSAC (section 4.7.1) to compute $\hat{\mathtt{H}}$ from four point correspondences.

(ii) **Geometric minimization of – <u>either</u> Sampson error:**

- Minimize the Sampson approximation to the geometric error (4.12–*p*99).
- The cost is minimized using the Newton algorithm of section A6.1(*p*597) or Levenberg–Marquardt algorithm of section A6.2(*p*600) over a suitable parametrization of $\hat{\mathtt{H}}$. For example the matrix may be parametrized by its 9 entries.

<u>or</u> Gold Standard error:

- Compute an initial estimate of the subsidiary variables $\{\hat{\mathbf{x}}_i\}$ using the measured points $\{\mathbf{x}_i\}$ or (better) the Sampson correction to these points given by (4.11–*p*99).
- Minimize the cost

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

  over $\hat{\mathtt{H}}$ and $\hat{\mathbf{x}}_i$, $i = 1, \ldots, n$. The cost is minimized using the Levenberg–Marquardt algorithm over $2n+9$ variables: $2n$ for the n 2D points $\hat{\mathbf{x}}_i$, and 9 for the homography matrix $\hat{\mathtt{H}}$.
- If the number of points is large then the sparse method of minimizing this cost function given in section A6.4(*p*607) is the recommended approach.