Computer vision: models, learning and inference

Chapter 15

Models for transformations

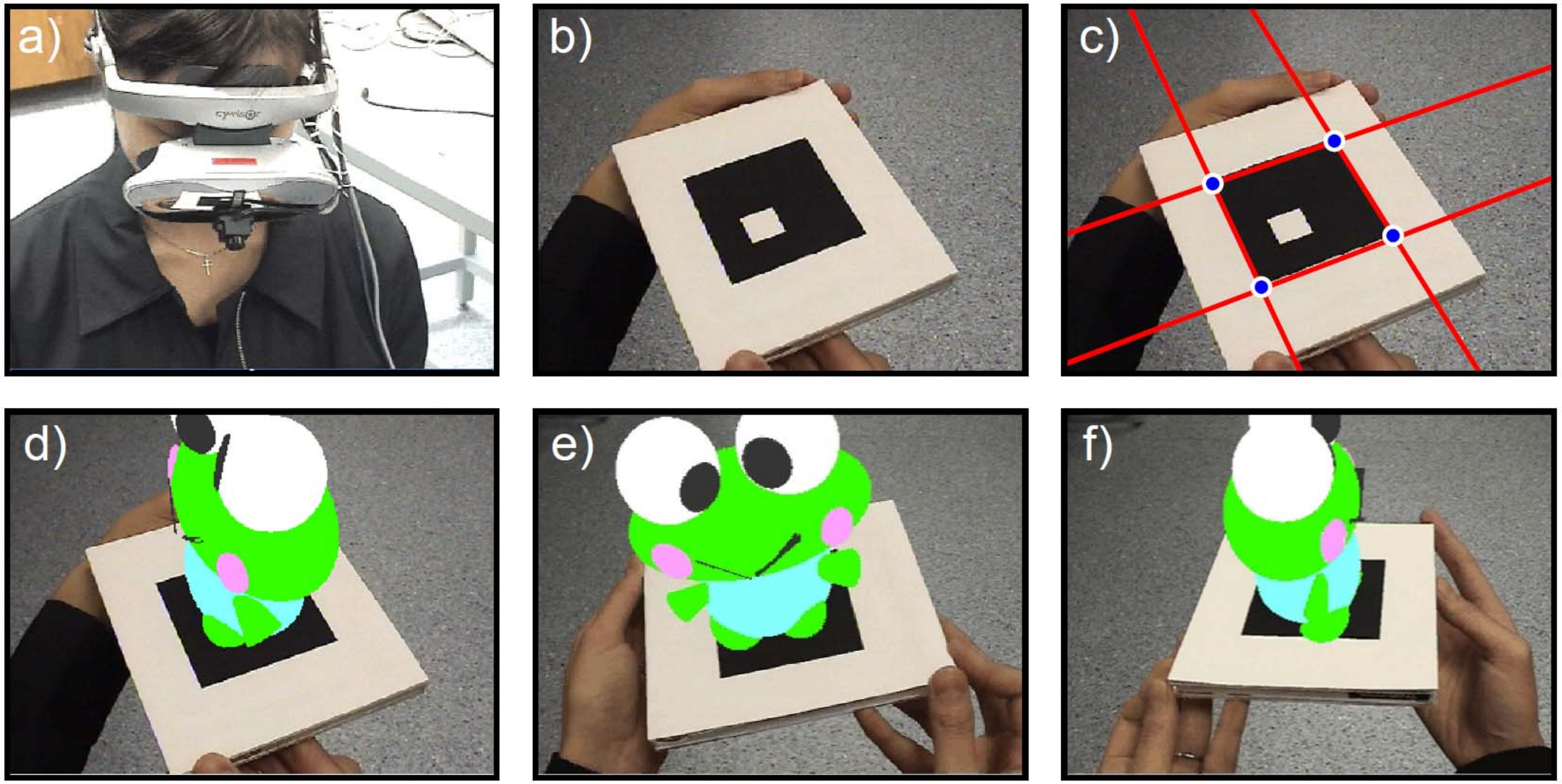
Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Transformation models

- Consider viewing a planar scene
- There is now a 1 to 1 mapping between points on the plane and points in the image
- We will investigate models for this 1 to 1 mapping
  - Euclidean
  - Similarity
  - Affine transform
  - Homography

Motivation: augmented reality tracking



Euclidean Transformation

- Consider viewing a fronto-parallel plane at a known distance D.
- In homogeneous coordinates, the imaging equations are:

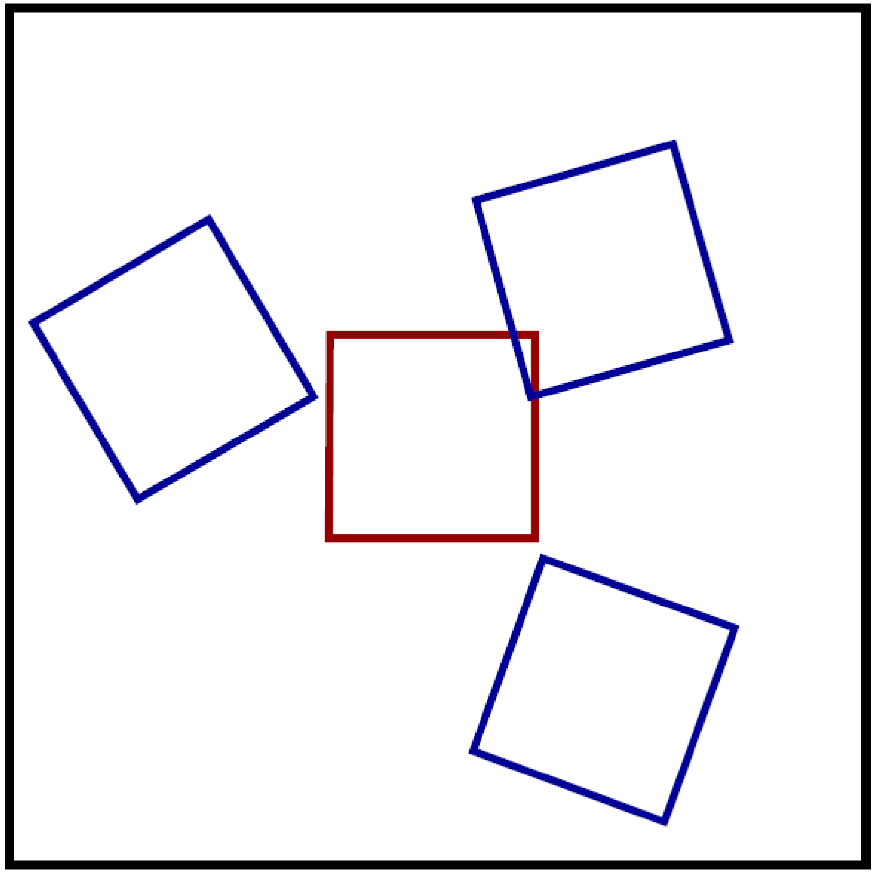
$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & 0 & \tau_x \\ \omega_{21} & \omega_{22} & 0 & \tau_y \\ 0 & 0 & 1 & D \end{bmatrix} \begin{bmatrix} u \\ v \\ 0 \\ 1 \end{bmatrix}$$

3D rotation matrix becomes 2D (in plane)

Plane at known distance D

Point is on plane (w=0)

Euclidean transformation



Euclidean transformation

- Simplifying

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & 0 & \tau_x\\\omega_{21} & \omega_{22} & 0 & \tau_y\\0 & 0 & 1 & D\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

$$= \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix},$$

- Rearranging the last equation

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Euclidean transformation

- Simplifying

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & 0 & \tau_x\\\omega_{21} & \omega_{22} & 0 & \tau_y\\0 & 0 & 1 & D\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

$$= \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix},$$

- Rearranging the last equation

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

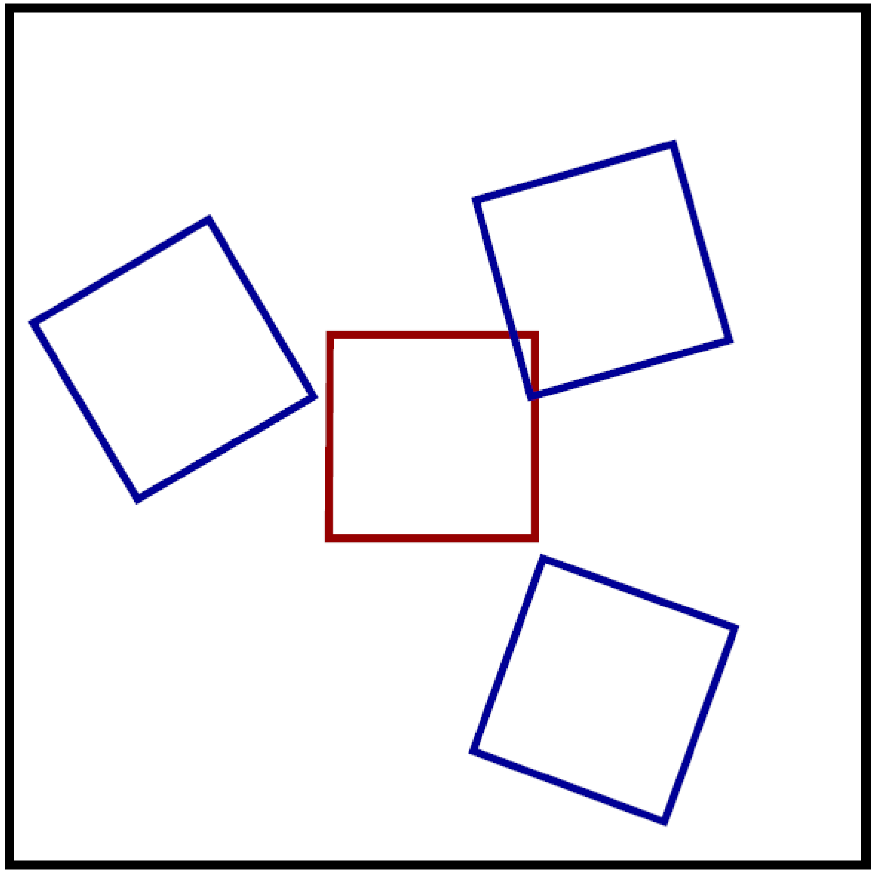
Euclidean transformation

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

- Pre-multiplying by inverse of (modified) intrinsic matrix

$$\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix}=\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Euclidean transformation



Homogeneous:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

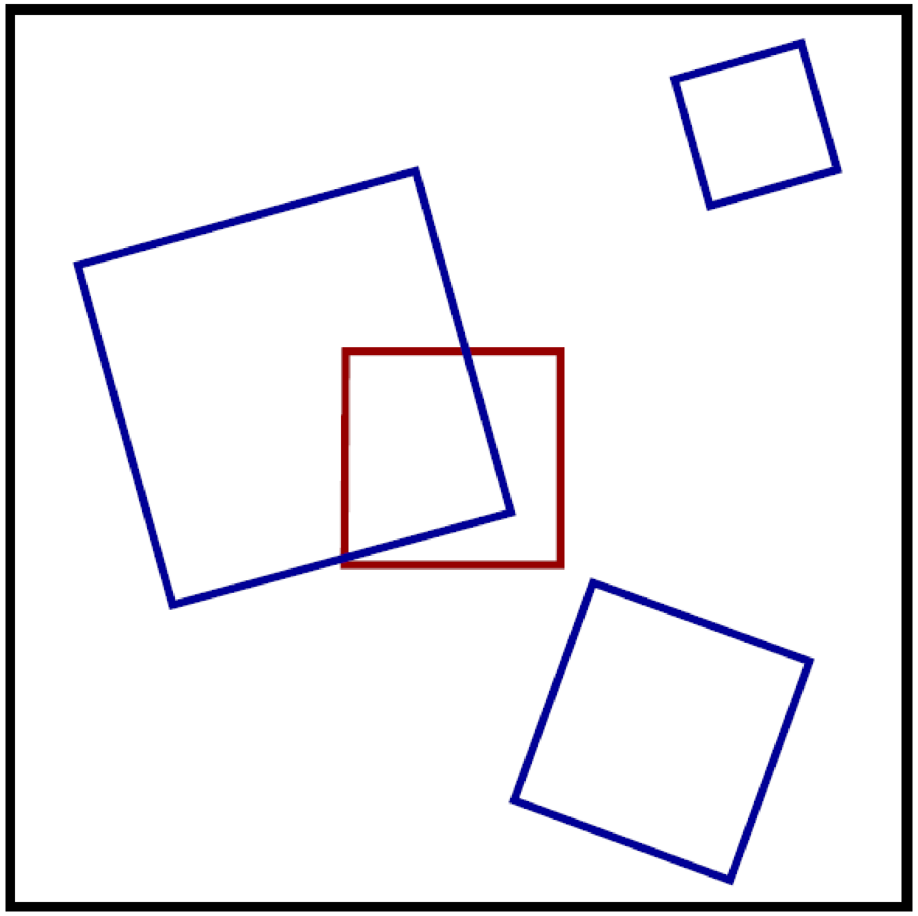
Cartesian:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}$$

For short:

$$\mathbf{x}' = \mathbf{euc}[\mathbf{w}, \boldsymbol{\Omega}, \boldsymbol{\tau}]$$

Similarity Transformation



Similarity Transformation

- Consider viewing fronto-parallel plane at unknown distance D
- By same logic as before we have

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & 0 & \tau_x\\\omega_{21} & \omega_{22} & 0 & \tau_y\\0 & 0 & 1 & D\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

- Premultiplying by inverse of intrinsic matrix

$$\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix} = \begin{bmatrix}\omega_{11} & \omega_{12} & 0 & \tau_x\\\omega_{21} & \omega_{22} & 0 & \tau_y\\0 & 0 & 1 & D\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

Similarity Transformation

Simplifying:

$$\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix}=\begin{bmatrix}\omega_{11}&\omega_{12}&0&\tau_x\\\omega_{21}&\omega_{22}&0&\tau_y\\0&0&1&D\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

$$=\begin{bmatrix}\omega_{11}&\omega_{12}&\tau_x\\\omega_{21}&\omega_{22}&\tau_y\\0&0&D\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Multiply each equation by $\rho=1/D$:

$$\rho\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix}=\begin{bmatrix}\rho\omega_{11}&\rho\omega_{12}&\rho\tau_x\\\rho\omega_{21}&\rho\omega_{22}&\rho\tau_y\\0&0&1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Similarity Transformation

Simplifying:

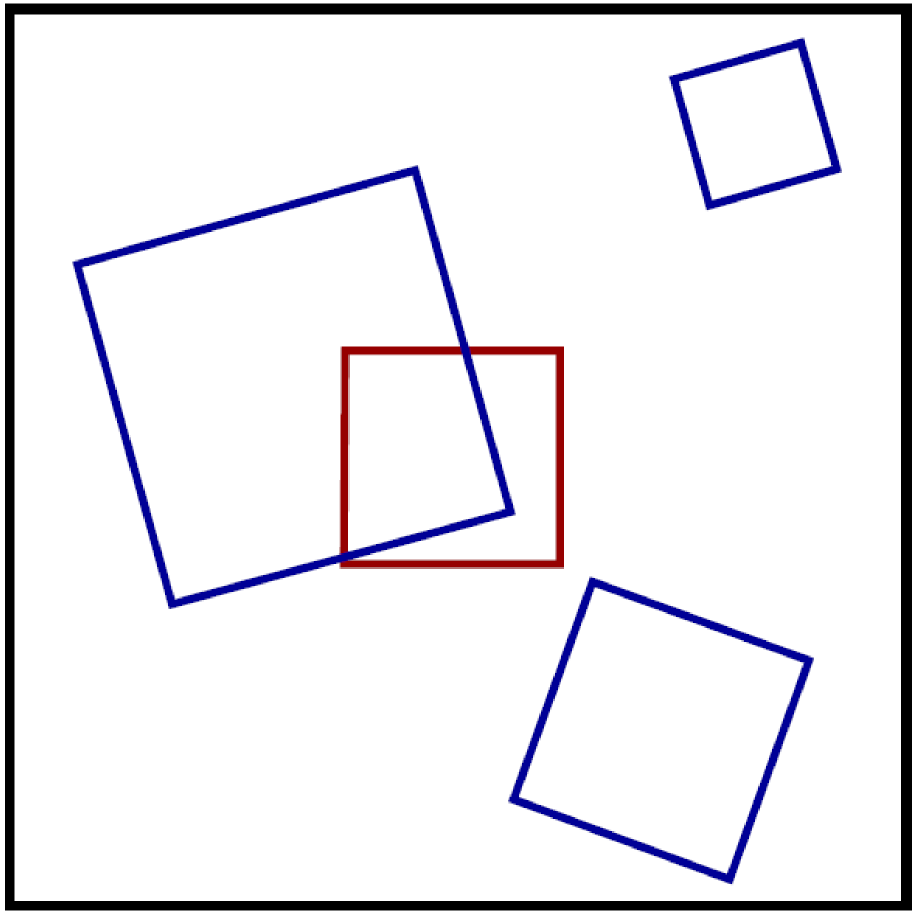
$$\rho\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix} = \begin{bmatrix}\rho\omega_{11} & \rho\omega_{12} & \rho\tau_x\\ \rho\omega_{21} & \rho\omega_{22} & \rho\tau_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Incorporate the constants by defining:

$$\tau_y \leftarrow \rho\tau_y \qquad \tau_x \leftarrow \rho\tau_x \qquad \lambda \leftarrow \rho\lambda$$

$$\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix} = \begin{bmatrix}\rho\omega_{11} & \rho\omega_{12} & \tau_x\\ \rho\omega_{21} & \rho\omega_{22} & \tau_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Similarity Transformation



Homogeneous:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \rho\omega_{11} & \rho\omega_{12} & \tau_x \\ \rho\omega_{21} & \rho\omega_{22} & \tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

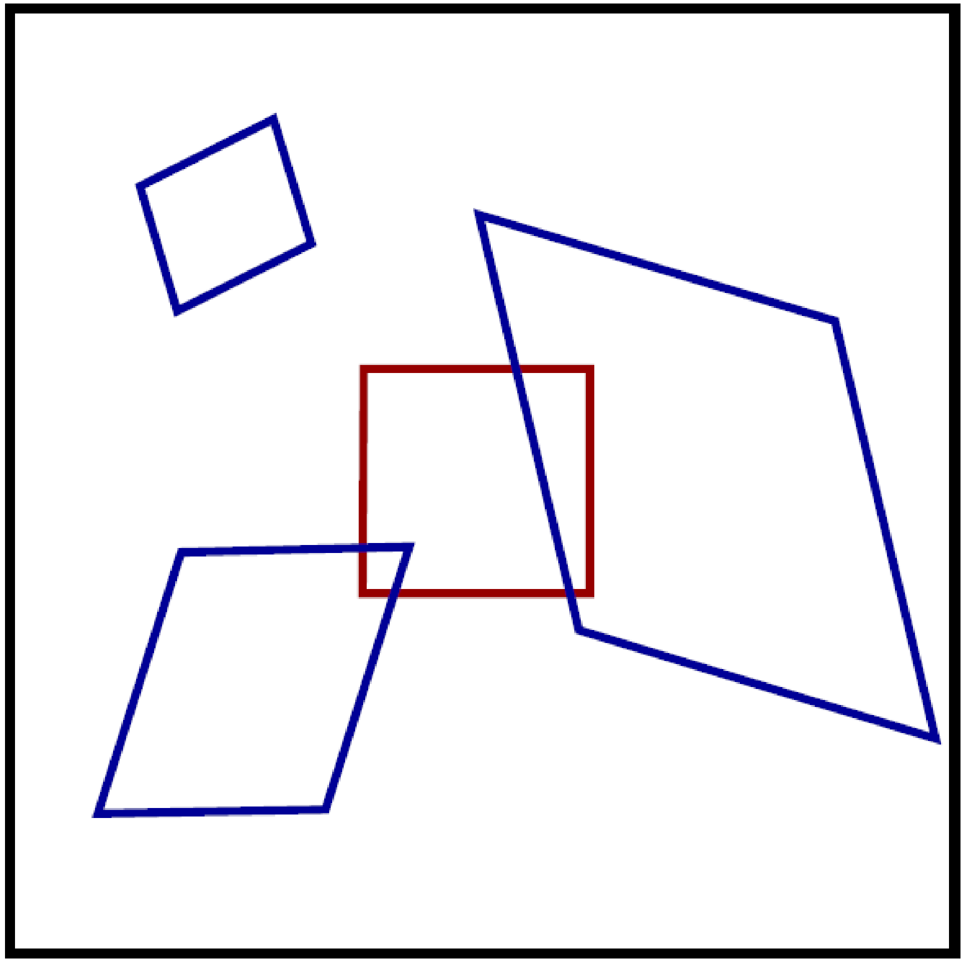
Cartesian:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \rho\omega_{11} & \rho\omega_{12} \\ \rho\omega_{21} & \rho\omega_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}$$

For short:

$$\mathbf{x}' = \mathbf{sim}[\mathbf{w}, \boldsymbol{\Omega}, \boldsymbol{\tau}, \rho]$$

Affine Transformation



Homogeneous:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \tau_x \\ \phi_{21} & \phi_{22} & \tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

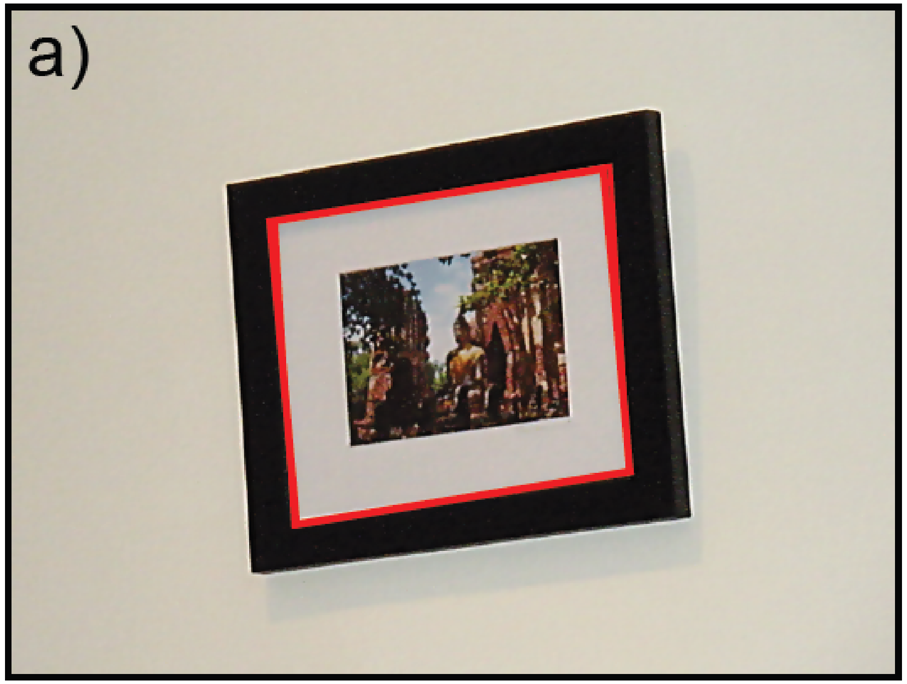
Cartesian:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}$$

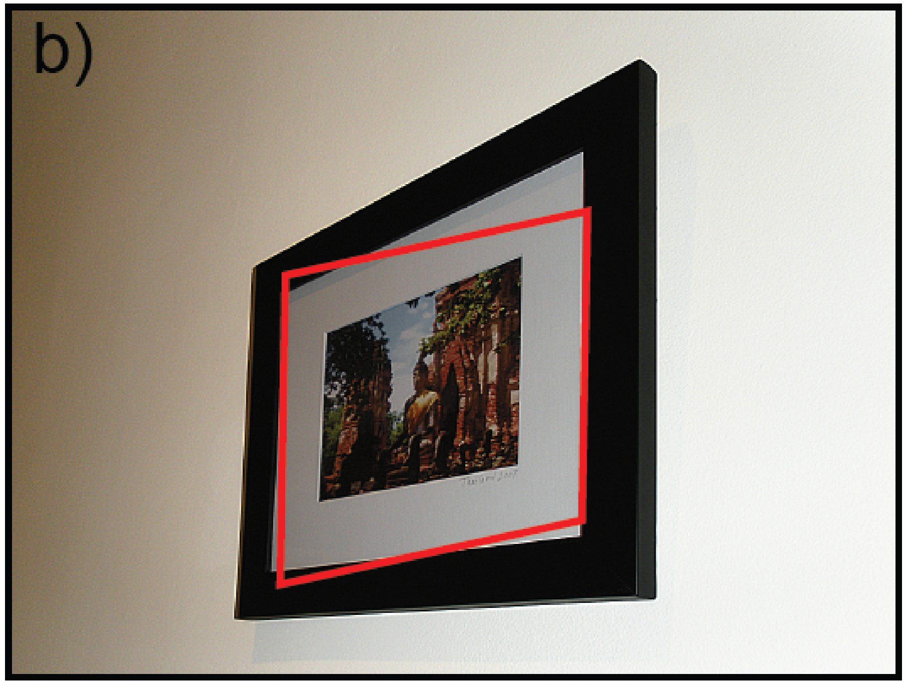
For short:

$$\mathbf{x}' = \mathbf{aff}[\mathbf{w}, \boldsymbol{\Phi}, \boldsymbol{\tau}]$$

Affine Transform

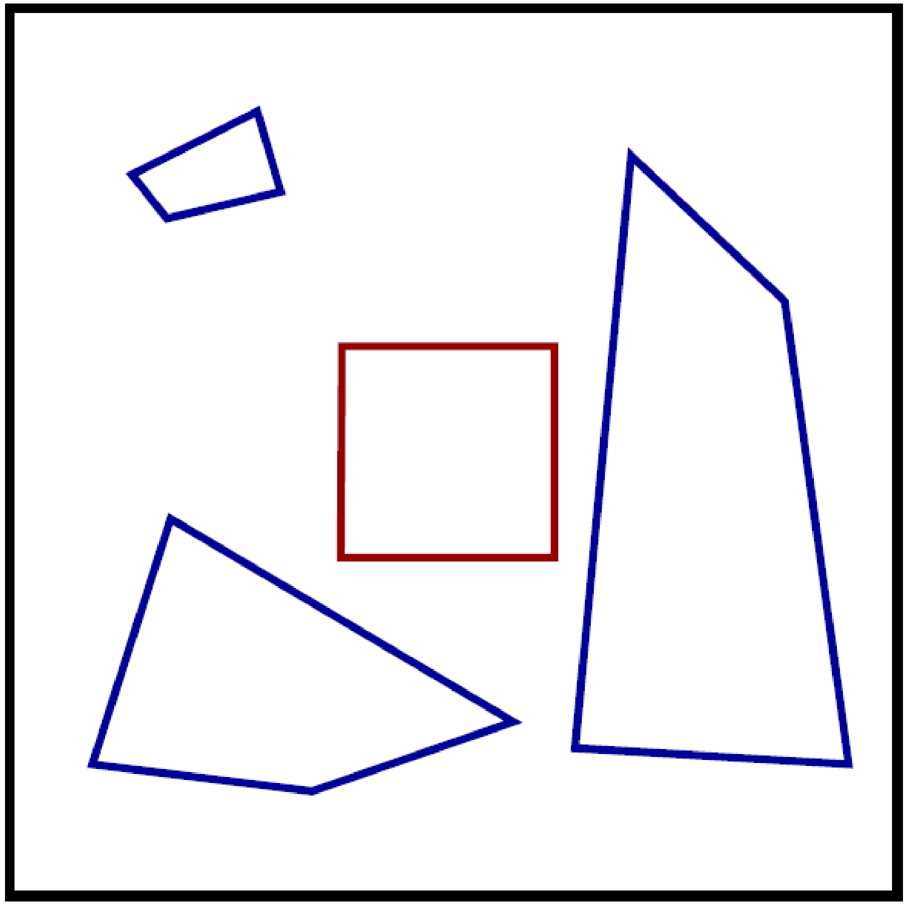


Affine transform describes mapping well when the depth variation within the planar object is small and the camera is far away



When variation in depth is comparable to distance to object then the affine transformation is not a good model. Here we need the homography

Homography



Projective transformation / collinearity / homography

Start with basic projection equation:

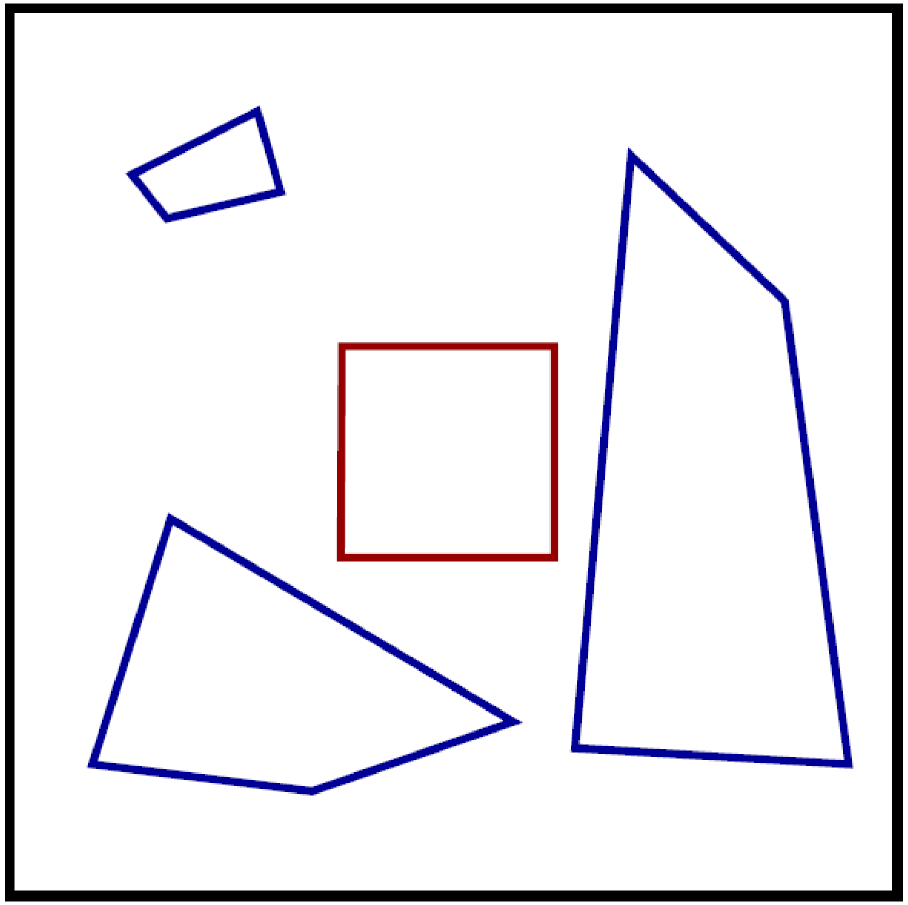
$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \omega_{13} & \tau_x\\\omega_{21} & \omega_{22} & \omega_{23} & \tau_y\\\omega_{31} & \omega_{32} & \omega_{33} & \tau_z\end{bmatrix}\begin{bmatrix}u\\v\\0\\1\end{bmatrix}$$

$$= \begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\\omega_{31} & \omega_{32} & \tau_z\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Combining these two matrices we get:

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\\phi_{21} & \phi_{22} & \phi_{23}\\\phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

Homography



Homogeneous:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Cartesian:

$$x = \frac{\phi_{11}u + \phi_{12}v + \phi_{13}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

$$y = \frac{\phi_{21}u + \phi_{22}v + \phi_{23}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

For short:

$$\mathbf{x} = \mathbf{hom}[\mathbf{w}, \boldsymbol{\Phi}]$$

Modeling for noise

- In the real world, the measured image positions are uncertain.
- We model this with a normal distribution
- e.g.

$$Pr(\mathbf{x}|\mathbf{w}) = \text{Norm}_{\mathbf{x}}\left[\mathbf{hom}[\mathbf{w}, \boldsymbol{\Phi}], \sigma^2\mathbf{I}\right]$$

Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

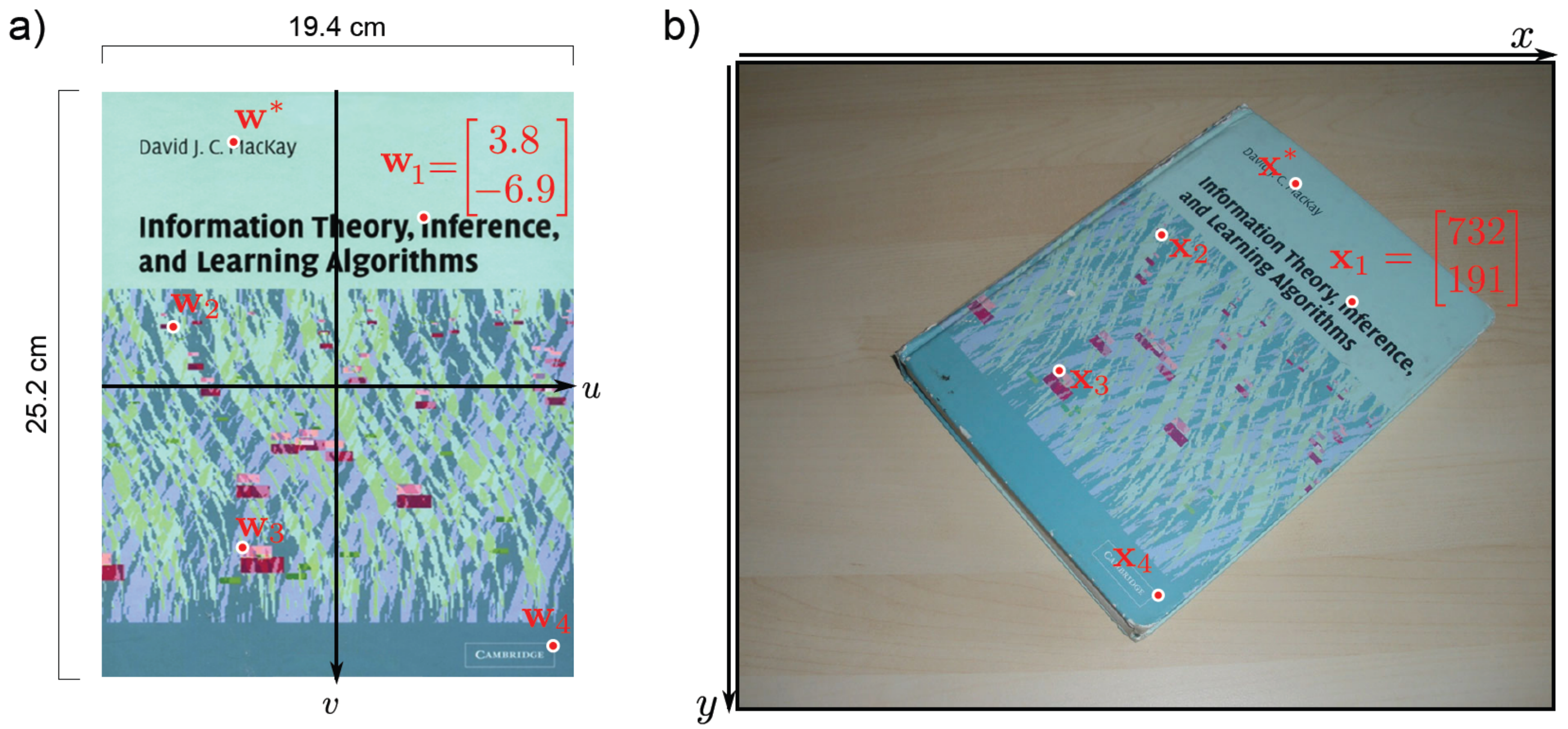
Learning and inference problems

$$Pr(\mathbf{x}|\mathbf{w}) \quad = \quad \text{Norm}_{\mathbf{x}}\left[\mathbf{hom}[\mathbf{w}, \boldsymbol{\Phi}], \sigma^2\mathbf{I}\right]$$

- Learning – take points on plane and their projections into the image and learn transformation parameters

- Inference – take the projection of a point in the image and establish point on plane

Learning and inference problems



Learning transformation models

- Maximum likelihood approach

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left[\prod_{i=1}^{I}\operatorname{Norm}_{\mathbf{x}_i}\left[\mathbf{trans}[\mathbf{w}_i,\boldsymbol{\theta}],\sigma^2\mathbf{I}\right]\right]\\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left[\sum_{i=1}^{I}\log\left[\operatorname{Norm}_{\mathbf{x}_i}\left[\mathbf{trans}[\mathbf{w}_i,\boldsymbol{\theta}],\sigma^2\mathbf{I}\right]\right]\right]\end{aligned}$$

- Becomes a least squares problem

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}\left(\mathbf{x}_i-\mathbf{trans}[\mathbf{w}_i,\boldsymbol{\theta}]\right)^T\left(\mathbf{x}_i-\mathbf{trans}[\mathbf{w}_i,\boldsymbol{\theta}]\right)\right]$$

Learning Euclidean parameters

$$
\begin{aligned}
\hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\tau}} &= \underset{\boldsymbol{\Omega},\boldsymbol{\tau}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} (\mathbf{x}_i - \mathbf{euc}[\mathbf{w}_i, \boldsymbol{\Omega}, \boldsymbol{\tau}])^T (\mathbf{x}_i - \mathbf{euc}[\mathbf{w}_i, \boldsymbol{\Omega}, \boldsymbol{\tau}]) \right] \\
&= \underset{\boldsymbol{\Omega},\boldsymbol{\tau}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} (\mathbf{x}_i - \boldsymbol{\Omega}\mathbf{w}_i - \boldsymbol{\tau})^T (\mathbf{x}_i - \boldsymbol{\Omega}\mathbf{w}_i - \boldsymbol{\tau}) \right],
\end{aligned}
$$

Solve for transformation:

$$
\hat{\boldsymbol{\tau}} = \frac{\sum_{i=1}^{I} \mathbf{x}_i - \boldsymbol{\Omega}\mathbf{w}_i}{I} = \boldsymbol{\mu}_x - \boldsymbol{\Omega}\boldsymbol{\mu}_w
$$

Remaining problem:

$$
\hat{\boldsymbol{\Omega}} = \underset{\boldsymbol{\Omega}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} ((\mathbf{x}_i - \boldsymbol{\mu}_x) - \boldsymbol{\Omega}(\mathbf{w}_i - \boldsymbol{\mu}_w))^T ((\mathbf{x}_i - \boldsymbol{\mu}_x) - \boldsymbol{\Omega}(\mathbf{w}_i - \boldsymbol{\mu}_w)) \right]
$$

Learning Euclidean parameters

$$\hat{\boldsymbol\Omega} = \underset{\boldsymbol\Omega}{\operatorname{argmin}} \left[\sum_{i=1}^{I} ((\mathbf{x}_i - \boldsymbol\mu_x) - \boldsymbol\Omega(\mathbf{w}_i - \boldsymbol\mu_w))^T ((\mathbf{x}_i - \boldsymbol\mu_x) - \boldsymbol\Omega(\mathbf{w}_i - \boldsymbol\mu_w)) \right]$$

Has the general form:

$$\hat{\boldsymbol\Omega} = \underset{\boldsymbol\Omega}{\operatorname{argmin}} \left|\mathbf{B} - \boldsymbol\Omega\mathbf{A}\right|_F \qquad \text{subject to } \boldsymbol\Omega\boldsymbol\Omega^T = \mathbf{I}, |\boldsymbol\Omega| = 1$$

- This is an orthogonal Procrustes problem. To solve:
  - Compute SVD $\mathbf{ULV}^T = \mathbf{BA}^T$
  - And then set $\hat{\boldsymbol\Omega} = \mathbf{VU}^T$

Learning similarity parameters

$$\begin{aligned}\hat{\boldsymbol\Omega}, \hat{\boldsymbol\tau}, \hat{\rho} &= \underset{\boldsymbol\Omega,\boldsymbol\tau,\rho}{\operatorname{argmin}}\left[\sum_{i=1}^{I}(\mathbf{x}_i - \mathbf{sim}[\mathbf{w}_i, \boldsymbol\Omega, \boldsymbol\tau, \rho])^T(\mathbf{x}_i - \mathbf{sim}[\mathbf{w}_i, \boldsymbol\Omega, \boldsymbol\tau, \rho])\right] \\ &= \underset{\boldsymbol\Omega,\boldsymbol\tau,\rho}{\operatorname{argmin}}\left[\sum_{i=1}^{I}(\mathbf{x}_i - \rho\boldsymbol\Omega\mathbf{w}_i - \boldsymbol\tau)^T(\mathbf{x}_i - \rho\boldsymbol\Omega\mathbf{w}_i - \boldsymbol\tau)\right]\end{aligned}$$

- Solve for rotation matrix as before
- Solve for translation and scaling factor

$$\hat{\boldsymbol\tau} = \frac{\sum_{i=1}^{I}(\mathbf{x}_i - \hat{\rho}\hat{\boldsymbol\Omega}\mathbf{w}_i)}{I}$$

$$\hat{\rho} = \frac{\sum_{i=1}^{I}(\mathbf{x}_i - \boldsymbol\mu_x)^T\hat{\boldsymbol\Omega}(\mathbf{w}_i - \boldsymbol\mu_w)}{\sum_{i=1}^{I}(\mathbf{w}_i - \boldsymbol\mu_w)^T(\mathbf{w}_i - \boldsymbol\mu_w)}$$

Learning affine parameters

$$\begin{aligned}\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\tau}} &= \underset{\boldsymbol{\Phi},\boldsymbol{\tau}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} (\mathbf{x}_i - \mathbf{aff}[\mathbf{w}_i, \boldsymbol{\Phi}, \boldsymbol{\tau}])^T (\mathbf{x}_i - \mathbf{aff}[\mathbf{w}_i, \boldsymbol{\Phi}, \boldsymbol{\tau}]) \right] \\ &= \underset{\boldsymbol{\Phi},\boldsymbol{\tau}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} (\mathbf{x}_i - \boldsymbol{\Phi}\mathbf{w}_i - \boldsymbol{\tau})^T (\mathbf{x}_i - \boldsymbol{\Phi}\mathbf{w}_i - \boldsymbol{\tau}) \right].\end{aligned}$$

- Affine transform is linear

$$\boldsymbol{\Phi}\mathbf{w}_i + \boldsymbol{\tau} = \begin{bmatrix} u_i & v_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_i & v_i & 1 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \tau_x \\ \phi_{21} \\ \phi_{22} \\ \tau_y \end{bmatrix} = \mathbf{A}_i \mathbf{b}$$

- Solve using least-squares solution

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} (\mathbf{x}_i - \mathbf{A}_i\mathbf{b})^T (\mathbf{x}_i - \mathbf{A}_i\mathbf{b}) \right]$$

Learning homography parameters

$$\begin{aligned}\hat{\boldsymbol{\Phi}} &= \underset{\boldsymbol{\Phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}(\mathbf{x}_i - \mathbf{hom}[\mathbf{w}_i,\boldsymbol{\Phi}])^T(\mathbf{x}_i - \mathbf{hom}[\mathbf{w}_i,\boldsymbol{\Phi}])\right]\\ &= \underset{\boldsymbol{\Phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}\left(x_i - \frac{\phi_{11}u_i+\phi_{12}v_i+\phi_{13}}{\phi_{31}u_i+\phi_{32}v_i+\phi_{33}}\right)^2 + \left(y_i - \frac{\phi_{21}u_i+\phi_{22}v_i+\phi_{23}}{\phi_{31}u_i+\phi_{32}v_i+\phi_{33}}\right)^2\right]\end{aligned}$$

- Homography is not linear – cannot be solved in closed form. Convert to other homogeneous coordinates

$$\lambda\begin{bmatrix}x_i\\y_i\\1\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\\ \phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}u_i\\v_i\\1\end{bmatrix}$$

- Both sides are 3x1 vectors; should be parallel, so cross product will be zero

$$\tilde{\mathbf{x}} \times \boldsymbol{\Phi}\tilde{\mathbf{w}} = \mathbf{0}$$

Learning homography parameters

$$\tilde{\mathbf{x}} \times \boldsymbol{\Phi}\tilde{\mathbf{w}} = \mathbf{0}$$

- Write out these equations in full

$$\begin{bmatrix} y(\phi_{31}u+\phi_{32}v+\phi_{33}) - (\phi_{21}u+\phi_{22}v+\phi_{23}) \\ (\phi_{11}u+\phi_{12}v+\phi_{13}) - x(\phi_{31}u+\phi_{32}v+\phi_{33}) \\ x(\phi_{21}u+\phi_{22}v+\phi_{23}) - y(\phi_{11}u+\phi_{12}v+\phi_{13}) \end{bmatrix} = \mathbf{0}$$

- There are only 2 independent equations here – use a minimum of four points to build up a set of equations

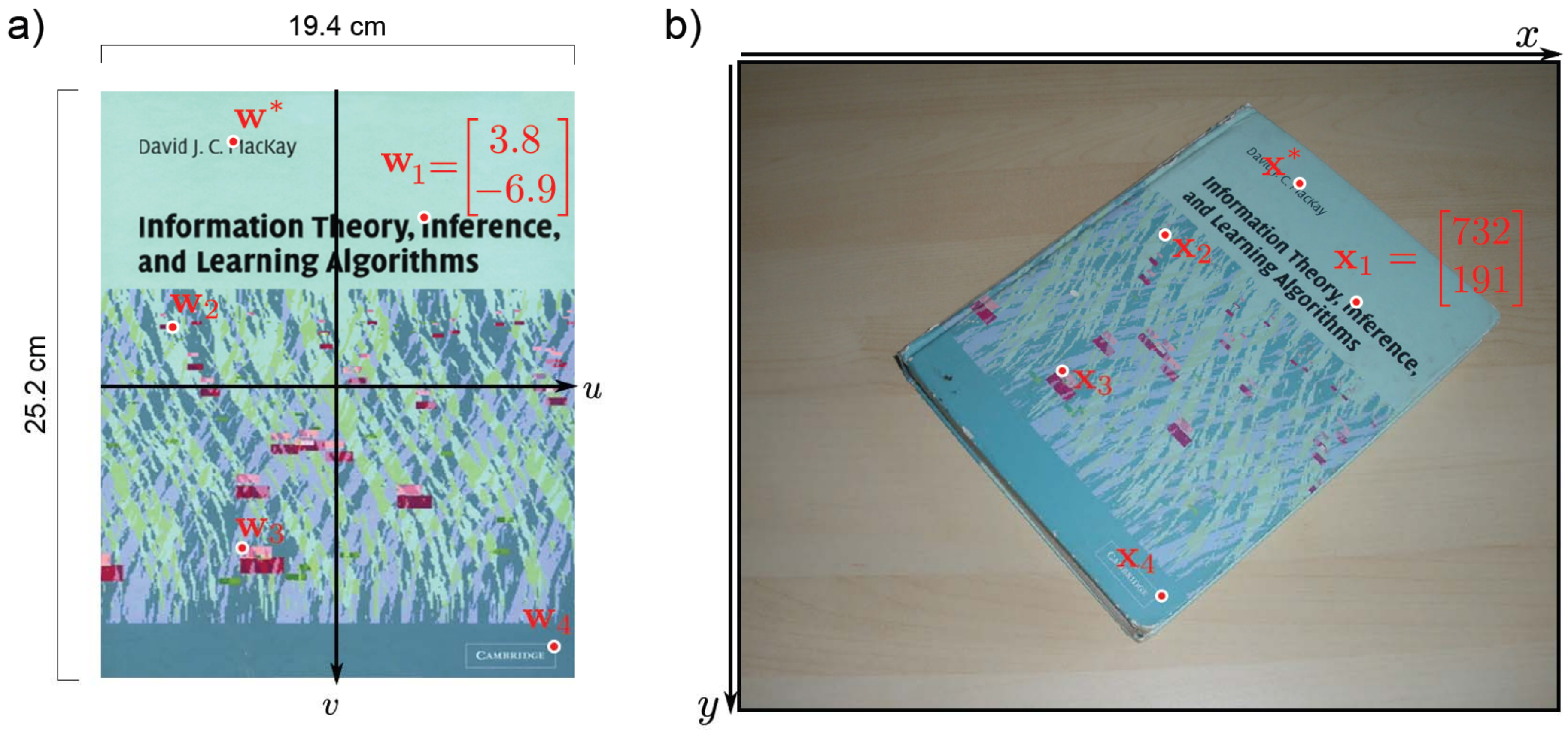
$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & y_1u_1 & y_1v_1 & y_1 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 & -x_1 \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & y_2u_2 & y_2v_2 & y_2 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -x_2v_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_I & -v_I & -1 & y_Iu_I & y_Iv_I & y_I \\ u_I & v_I & 1 & 0 & 0 & 0 & -x_Iu_I & -x_Iv_I & -x_I \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \phi_{13} \\ \phi_{21} \\ \phi_{22} \\ \phi_{23} \\ \phi_{31} \\ \phi_{32} \\ \phi_{33} \end{bmatrix} = \mathbf{0}$$

Learning homography parameters

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & y_1u_1 & y_1v_1 & y_1 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 & -x_1 \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & y_2u_2 & y_2v_2 & y_2 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -x_2v_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_I & -v_I & -1 & y_Iu_I & y_Iv_I & y_I \\ u_I & v_I & 1 & 0 & 0 & 0 & -x_Iu_I & -x_Iv_I & -x_I \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \phi_{13} \\ \phi_{21} \\ \phi_{22} \\ \phi_{23} \\ \phi_{31} \\ \phi_{32} \\ \phi_{33} \end{bmatrix} = \mathbf{0}$$

- These equations have the form $\mathbf{A}\boldsymbol{\phi} = \mathbf{0}$, which we need to solve with the constraint $\boldsymbol{\phi}^T\boldsymbol{\phi} = 1$
- This is a "minimum direction" problem
  - Compute SVD $\mathbf{A} = \mathbf{U}\mathbf{L}\mathbf{V}^T$
  - Take last column of $\mathbf{V}$
- Then use non-linear optimization

Inference problems



- Given point **x*** in image, find position **w*** on object

Inference

$$\begin{aligned}\hat{\mathbf{w}} &= \underset{\mathbf{w}}{\operatorname{argmax}}\left[\log\left[\operatorname{Norm}_{\mathbf{x}}\left[\mathbf{trans}[\mathbf{w},\boldsymbol{\theta}],\sigma^2\mathbf{I}\right]\right]\right]\\ &= \underset{\mathbf{w}}{\operatorname{argmin}}\left[(\mathbf{x}-\mathbf{trans}[\mathbf{w},\boldsymbol{\theta}])^T(\mathbf{x}-\mathbf{trans}[\mathbf{w},\boldsymbol{\theta}])\right]\end{aligned}$$

In the absence of noise, we have the relation $\mathbf{x}=\mathbf{trans}[\mathbf{w},\boldsymbol{\theta}]$, or in homogeneous coordinates

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\\ \phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

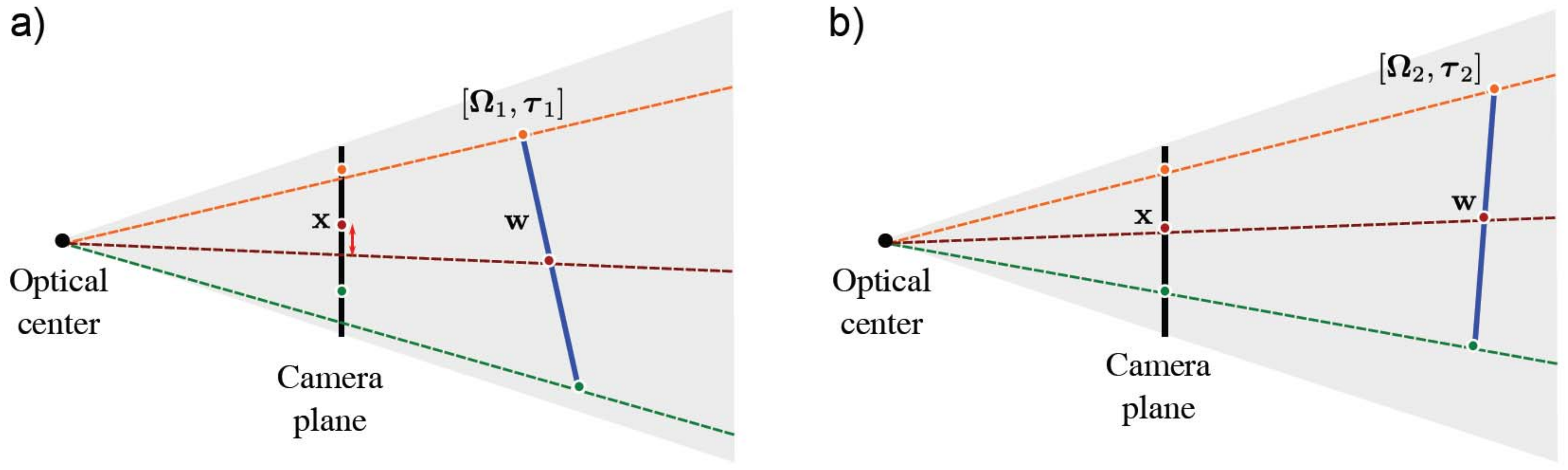
To solve for the points, we simply invert this relation

$$\lambda'\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\\ \phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}^{-1}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Problem 1: exterior orientation



$$
\begin{aligned}
\hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\tau}} &= \underset{\boldsymbol{\Omega},\boldsymbol{\tau}}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log \left[Pr(\mathbf{x}_i | \mathbf{w}_i, \boldsymbol{\Lambda}, \boldsymbol{\Omega}, \boldsymbol{\tau}) \right] \right] \\
&= \underset{\boldsymbol{\Omega},\boldsymbol{\tau}}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log \left[\operatorname{Norm}_{\mathbf{x}_i} \left[\mathbf{pinhole}[\mathbf{w}_i, \boldsymbol{\Lambda}, \boldsymbol{\Omega}, \boldsymbol{\tau}], \sigma^2 \mathbf{I} \right] \right] \right]
\end{aligned}
$$

Problem 1: exterior orientation

- Writing out the camera equations in full

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix} = \lambda'\begin{bmatrix}\phi_x & \gamma & \delta_x\\0 & \phi_y & \delta_y\\0 & 0 & D\end{bmatrix}\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\\omega_{31} & \omega_{32} & \tau_z\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

$$= \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\\phi_{21} & \phi_{22} & \phi_{23}\\\phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}$$

- Estimate the homography from matched points
- Factor out the intrinsic parameters $\mathbf{\Phi}' = \mathbf{\Lambda}^{-1}\mathbf{\Phi}$

$$\begin{bmatrix}\phi'_{11} & \phi'_{12} & \phi'_{13}\\\phi'_{21} & \phi'_{22} & \phi'_{23}\\\phi'_{31} & \phi'_{32} & \phi'_{33}\end{bmatrix} = \lambda'\begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\\omega_{21} & \omega_{22} & \tau_y\\\omega_{31} & \omega_{32} & \tau_z\end{bmatrix}$$

Problem 1: exterior orientation

$$\begin{bmatrix}\phi'_{11} & \phi'_{12} & \phi'_{13}\\ \phi'_{21} & \phi'_{22} & \phi'_{23}\\ \phi'_{31} & \phi'_{32} & \phi'_{33}\end{bmatrix} = \lambda' \begin{bmatrix}\omega_{11} & \omega_{12} & \tau_x\\ \omega_{21} & \omega_{22} & \tau_y\\ \omega_{31} & \omega_{32} & \tau_z\end{bmatrix}$$

- To estimate the first two columns of rotation matrix, we compute this singular value decomposition

$$\begin{bmatrix}\phi'_{11} & \phi'_{12}\\ \phi'_{21} & \phi'_{22}\\ \phi'_{31} & \phi'_{32}\end{bmatrix} = \mathbf{ULV}^T$$

- Then we set

$$\begin{bmatrix}\omega_{11} & \omega_{12}\\ \omega_{21} & \omega_{22}\\ \omega_{31} & \omega_{32}\end{bmatrix} = \mathbf{U}\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}\mathbf{V}^T$$

Problem 1: exterior orientation

- Find the last column using the cross product of first two columns
- Make sure the determinant is 1. If it is -1, then multiply last column by -1.
- Find translation scaling factor between old and new values

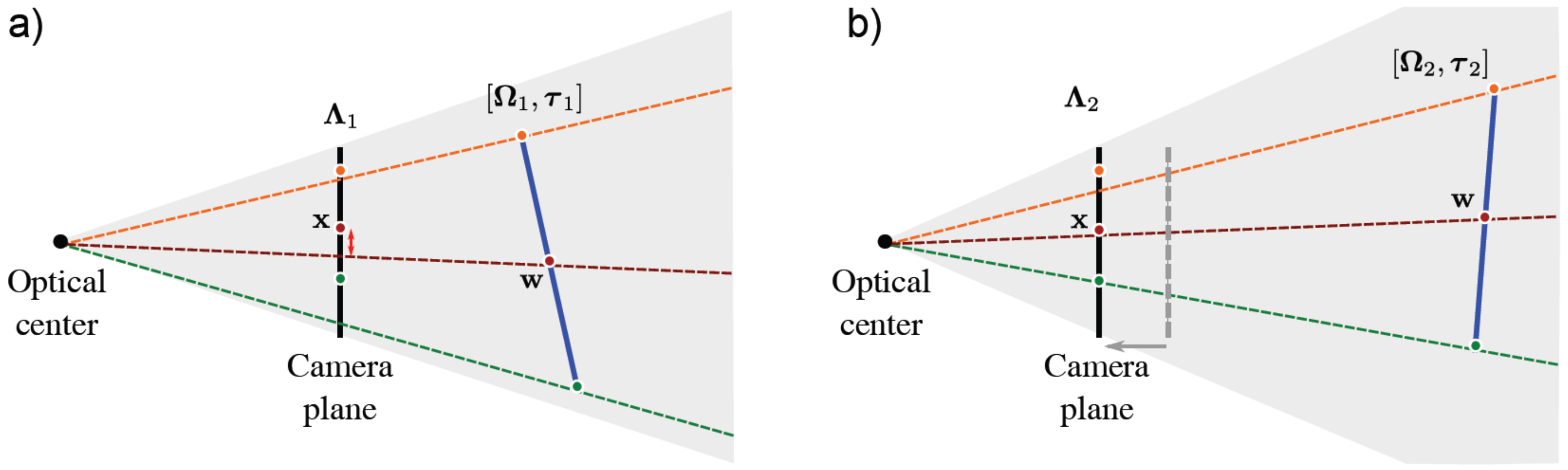
$$\lambda' = \frac{\sum_{m=1}^{3}\sum_{n=1}^{2}\phi'_{mn}/\omega_{mn}}{6}$$

- Finally, set $\boldsymbol{\tau} = [\phi'_{13}, \phi'_{23}, \phi'_{33}]^T/\lambda'$

Structure

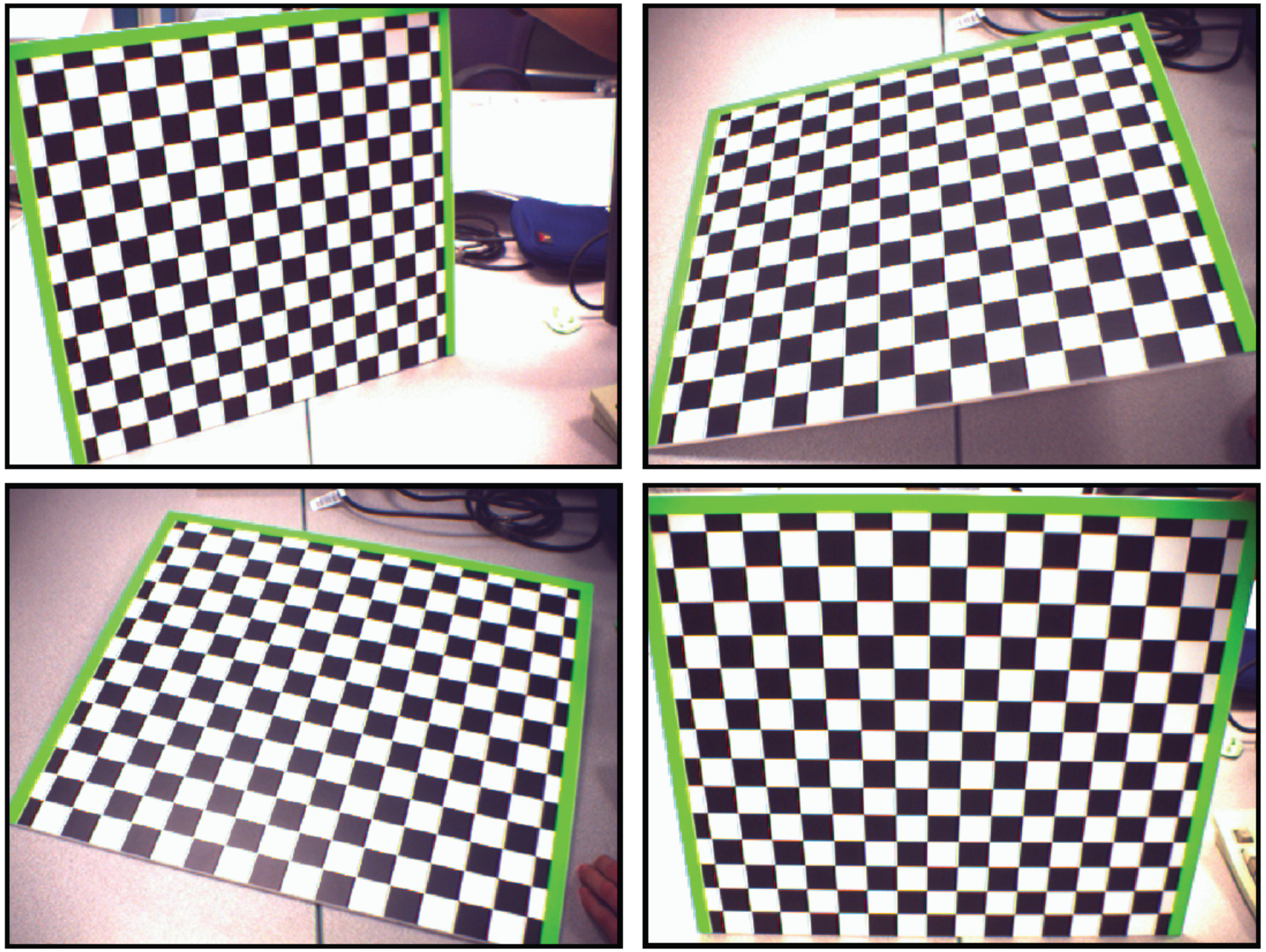
- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Problem 2: calibration



$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} \left[\max_{\Omega_{1\ldots J}, \tau_{1\ldots J}} \left[\sum_{i=1}^{I} \sum_{j=1}^{J} \log \left[Pr(\mathbf{x}_{ij} | \mathbf{w}_i, \Lambda, \Omega_j, \tau_j) \right] \right] \right]$$

Structure



Calibration

$$\hat{\boldsymbol\Lambda} = \underset{\boldsymbol\Lambda}{\operatorname{argmax}}\left[\max_{\boldsymbol\Omega_{1\ldots J},\boldsymbol\tau_{1\ldots J}}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\log\left[Pr(\mathbf{x}_{ij}|\mathbf{w}_i,\boldsymbol\Lambda,\boldsymbol\Omega_j,\boldsymbol\tau_j)\right]\right]\right]$$

One approach (not very efficient) is to alternately

- Optimize extrinsic parameters for fixed intrinsic

$$\hat{\boldsymbol\Omega}_j,\hat{\boldsymbol\tau}_j = \underset{\boldsymbol\Omega_j,\boldsymbol\tau_j}{\operatorname{argmax}}\left[\sum_{i=1}^{I}\log\left[Pr(\mathbf{x}_{ij}|\mathbf{w}_i,\boldsymbol\Lambda,\boldsymbol\Omega_j,\boldsymbol\tau_j)\right]\right]$$

- Optimize intrinsic parameters for fixed extrinsic

$$\hat{\boldsymbol\Lambda} = \underset{\boldsymbol\Lambda}{\operatorname{argmax}}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\log\left[Pr(\mathbf{x}_{ij}|\mathbf{w}_i,\boldsymbol\Lambda,\boldsymbol\Omega_j,\boldsymbol\tau_j)\right]\right]$$

Then use non-linear optimization.

Check code

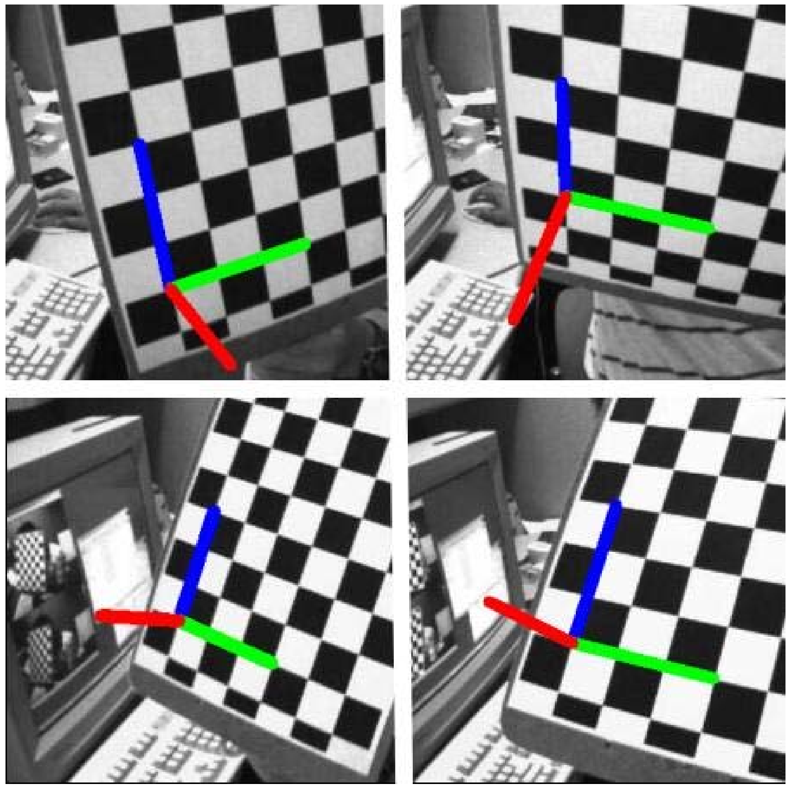
OpenCV-Python Tutorial:

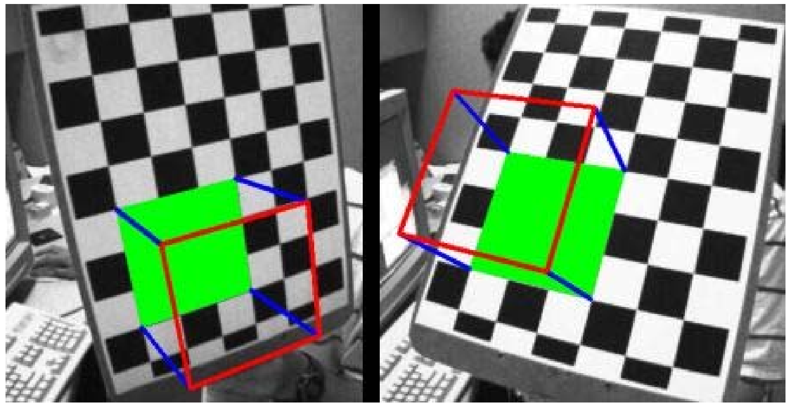
https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html (official)

> Camera Calibration and 3D Reconstruction
> Pose Estimation

OpenCV-Python Tutorial (Japanese):

http://lang.sist.chukyo-u.ac.jp/classes/OpenCV/

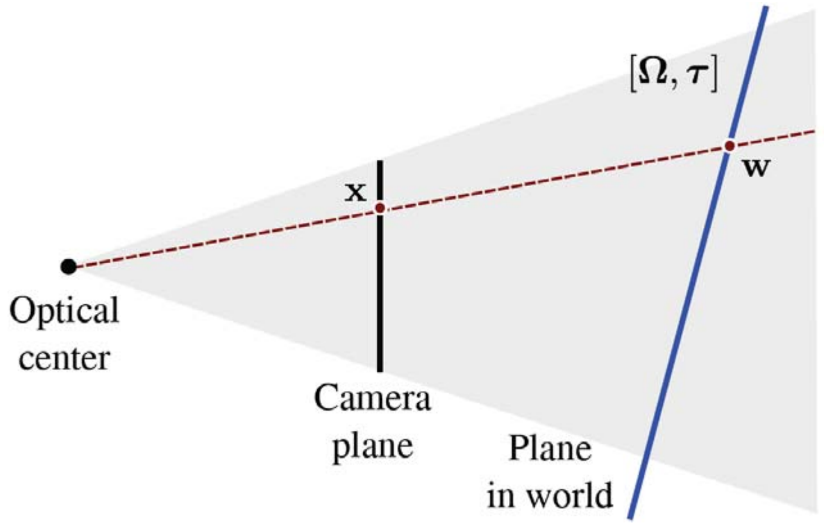




Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Problem 3 - reconstruction



Transformation between plane and image:

$$\mathbf{T} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & D \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ \omega_{31} & \omega_{32} & \tau_z \end{bmatrix}$$

Point in frame of reference of plane: $\tilde{\mathbf{w}} = \mathbf{T}^{-1}\tilde{\mathbf{x}}$

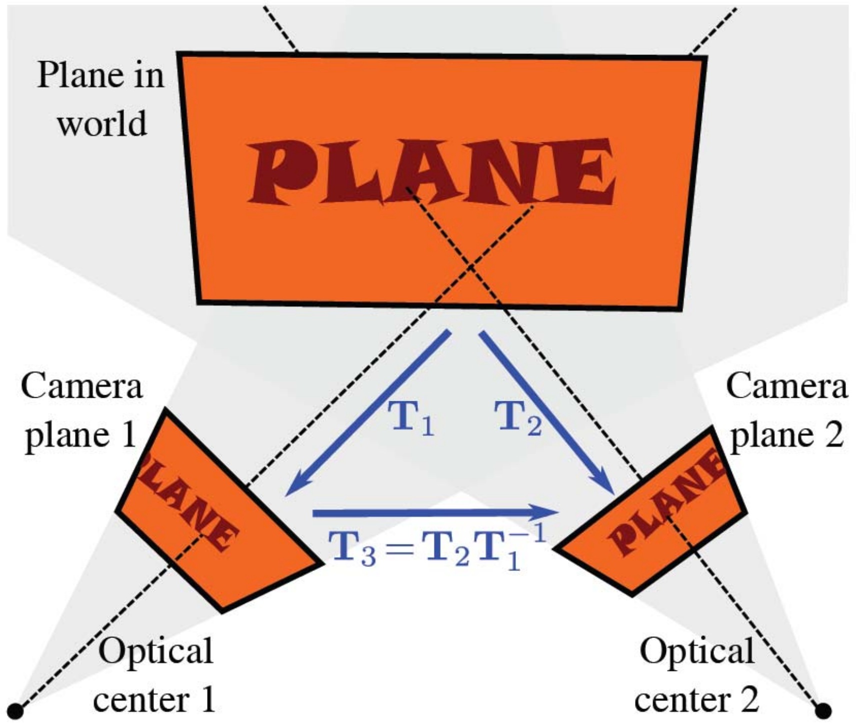
Point in frame of reference of camera $\mathbf{w}' = \mathbf{\Omega}\mathbf{w} + \boldsymbol{\tau}$

Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Transformations between images

- So far we have considered transformations between the image and a plane in the world
- Now consider two cameras viewing the same plane
- There is a homography between camera 1 and the plane and a second homography between camera 2 and the plane



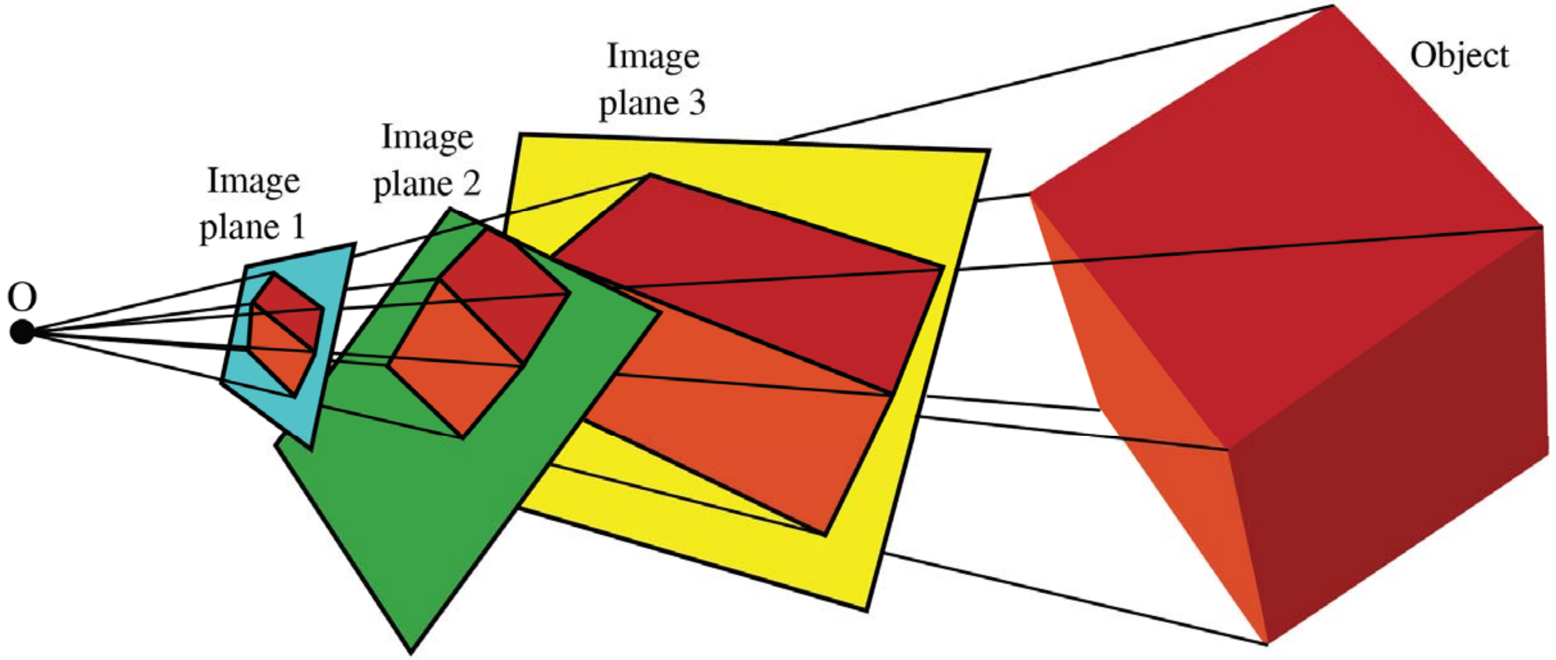
- It follows that the relation between the two images is also a homography

Properties of the homography

Homography is a linear transformation of a ray

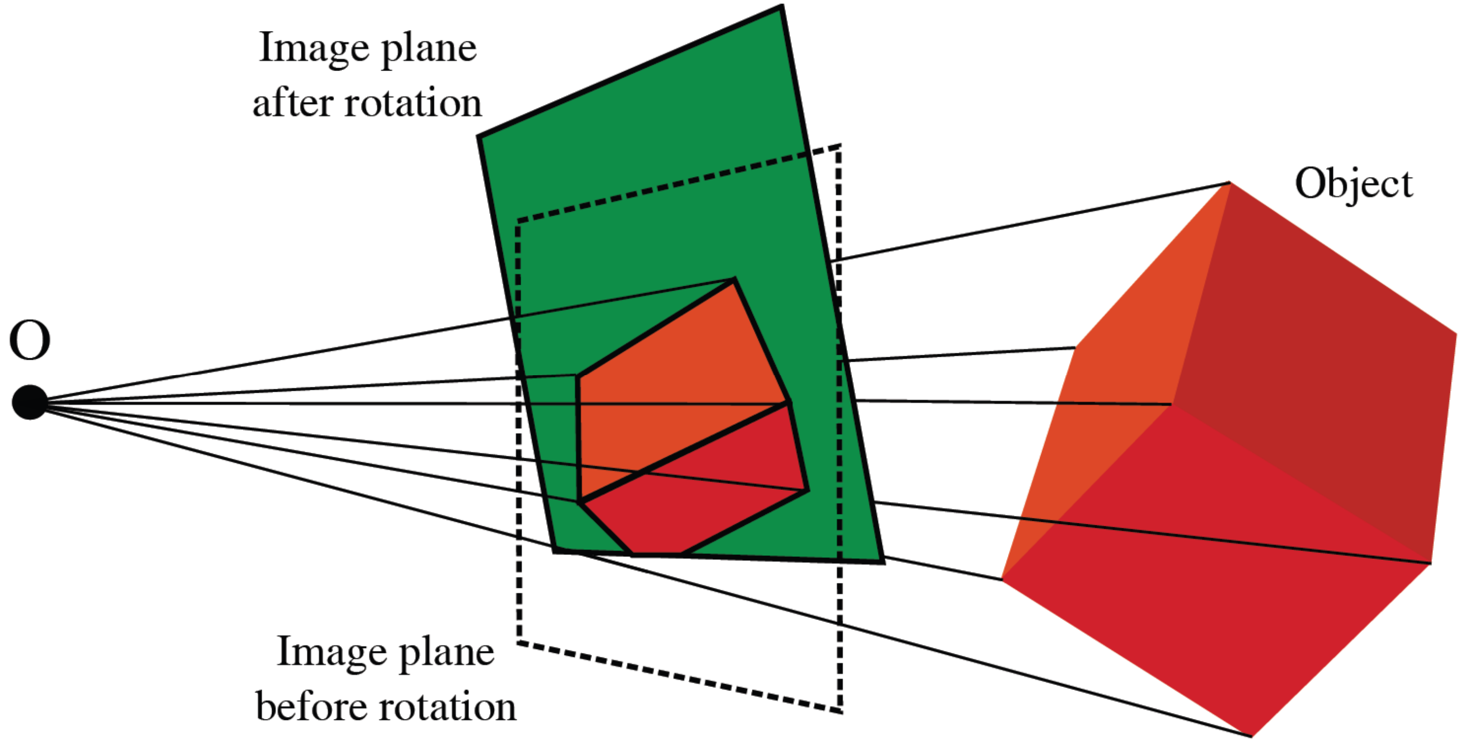
$$\lambda\begin{bmatrix}x_1\\y_1\\1\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\\ \phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}x_2\\y_2\\1\end{bmatrix}$$

Equivalently, leave rays and linearly transform image plane – all images formed by all planes that cut the same ray bundle are related by homographies.



Camera under pure rotation

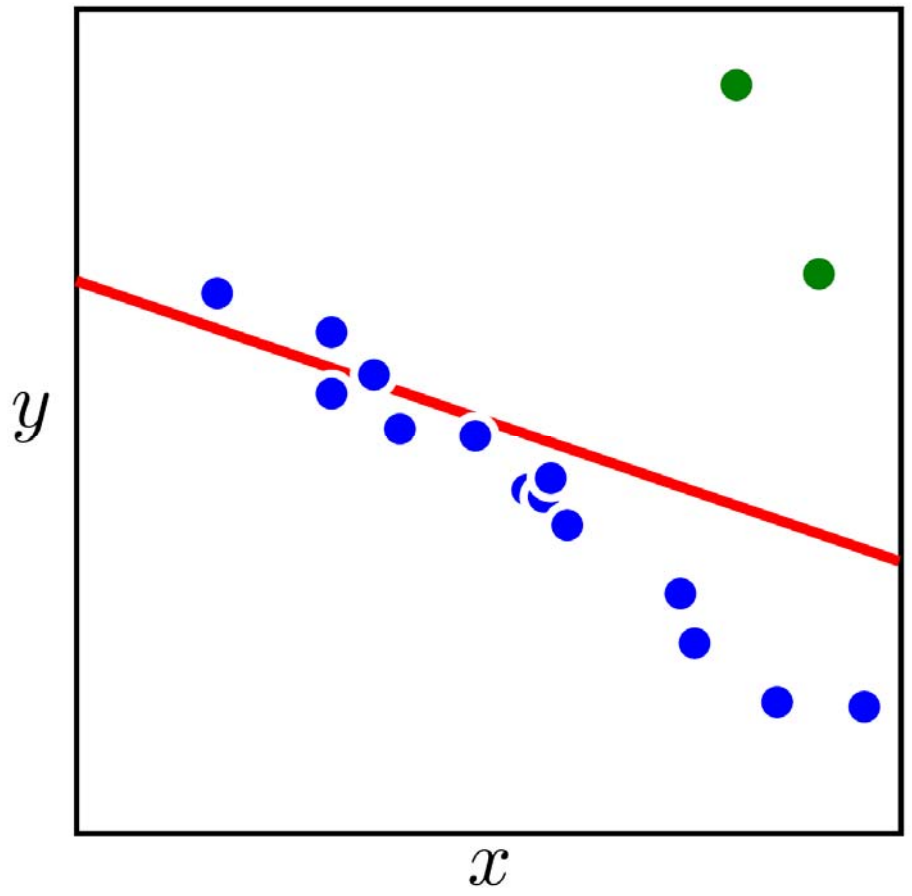
Special case is camera under pure rotation.
Homography can be showed to be $\boldsymbol{\Phi} = \boldsymbol{\Lambda}\boldsymbol{\Omega}_2\boldsymbol{\Lambda}^{-1}$



Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Robust estimation



Least squares criterion is not robust to outliers

For example, the two outliers here cause the fitted line to be quite wrong.

One approach to fitting under these circumstances is to use RANSAC – “Random sampling by consensus”

RANSAC

1. Randomly choose a minimal subset of data.
2. Use this subset to estimate the parameters.
3. Compute the number of inliers for this model.
4. Repeat steps 1-3 a fixed number of times.
5. Re-estimate model using inliers from the best fit.

RANSAC

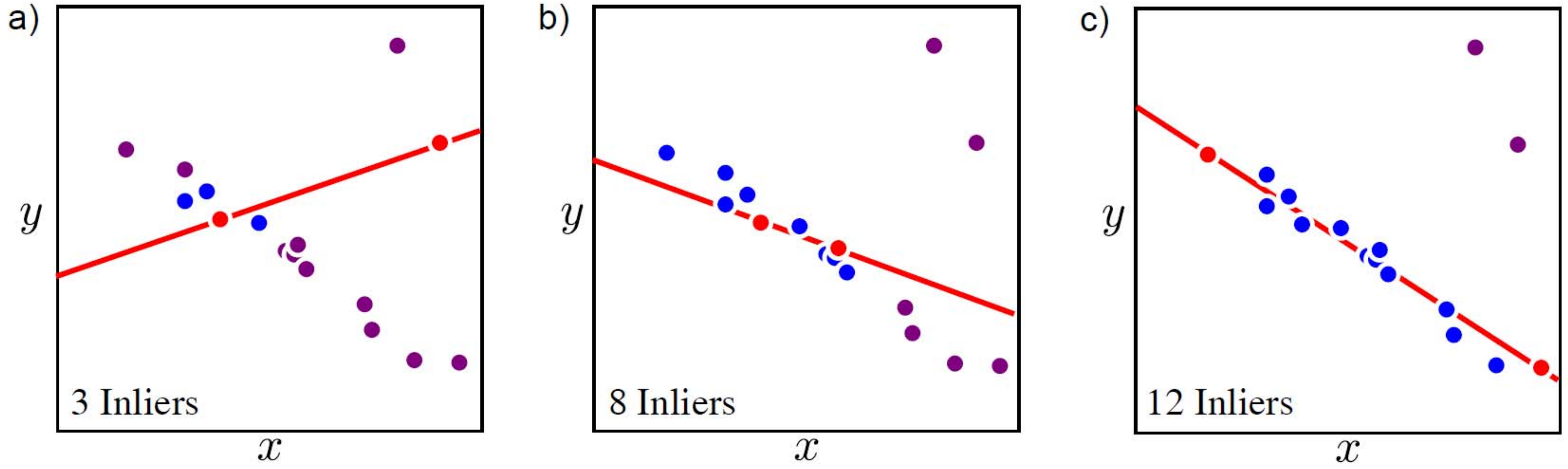
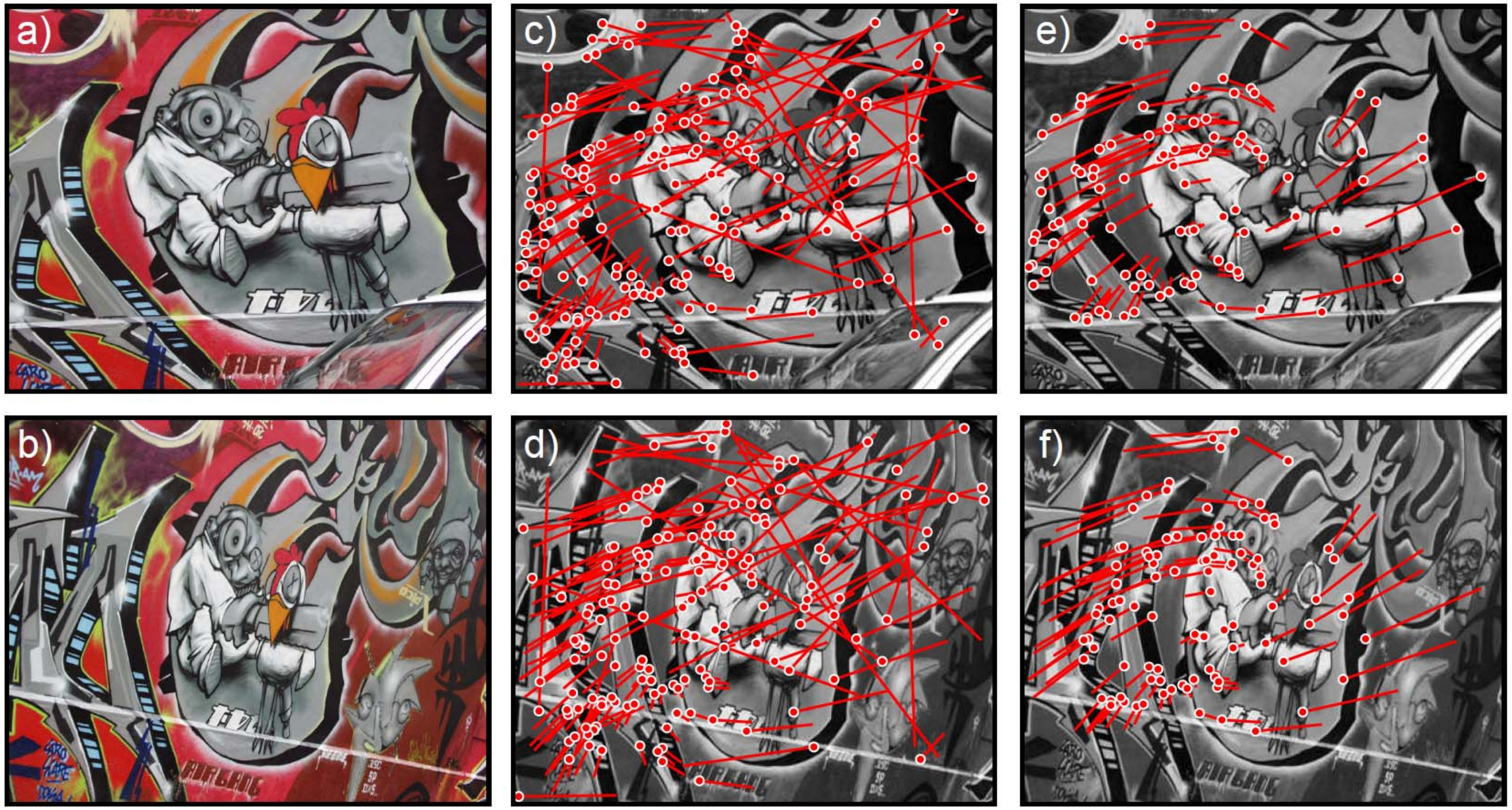


Figure 14.16 RANSAC procedure. a) We select a random minimal subset of points to fit the line (red points). We fit the line to these points and count how many of the other points agree with this solution (blue points). These are termed inliers. Here there are only 3 inliers. b,c) This procedure is repeated with different minimal subsets of points. After a number of iterations we choose the fit that had the most inliers. We refit the line using only the inliers from this fit.

Fitting a homography with RANSAC

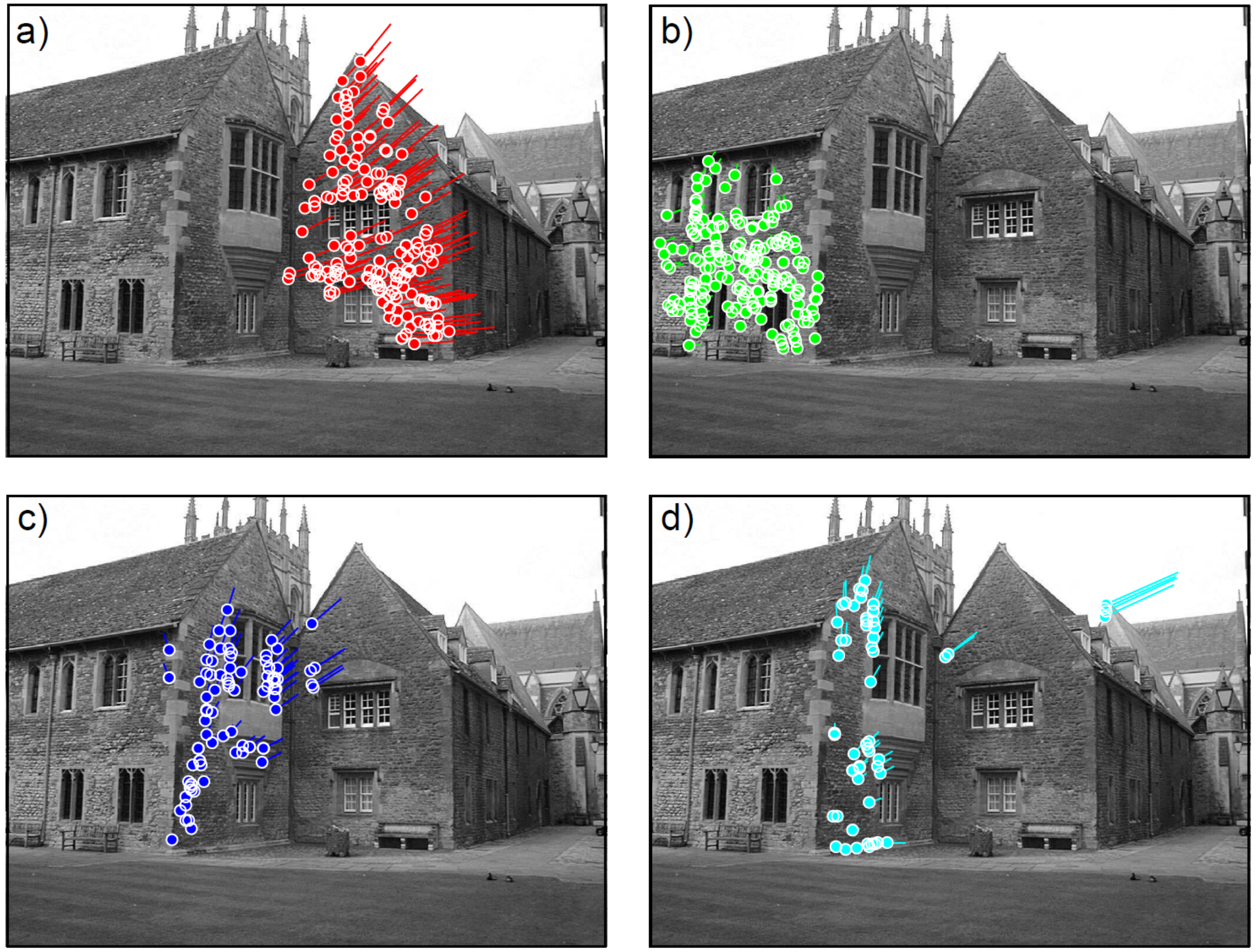


Original images　　Initial matches　　Inliers from RANSAC

Piecewise planarity

Many scenes are not planar, but are nonetheless piecewise planar
Can we match all of the planes to one another?

Approach 1 – Sequential RANSAC



Problems: greedy algorithm and no need to be spatially coherent

Approach 2 – PEaRL (propose, estimate and re-learn)

- Associate label l which indicates which plane we are in
- Relation between points x_i in image1 and y_i in image 2

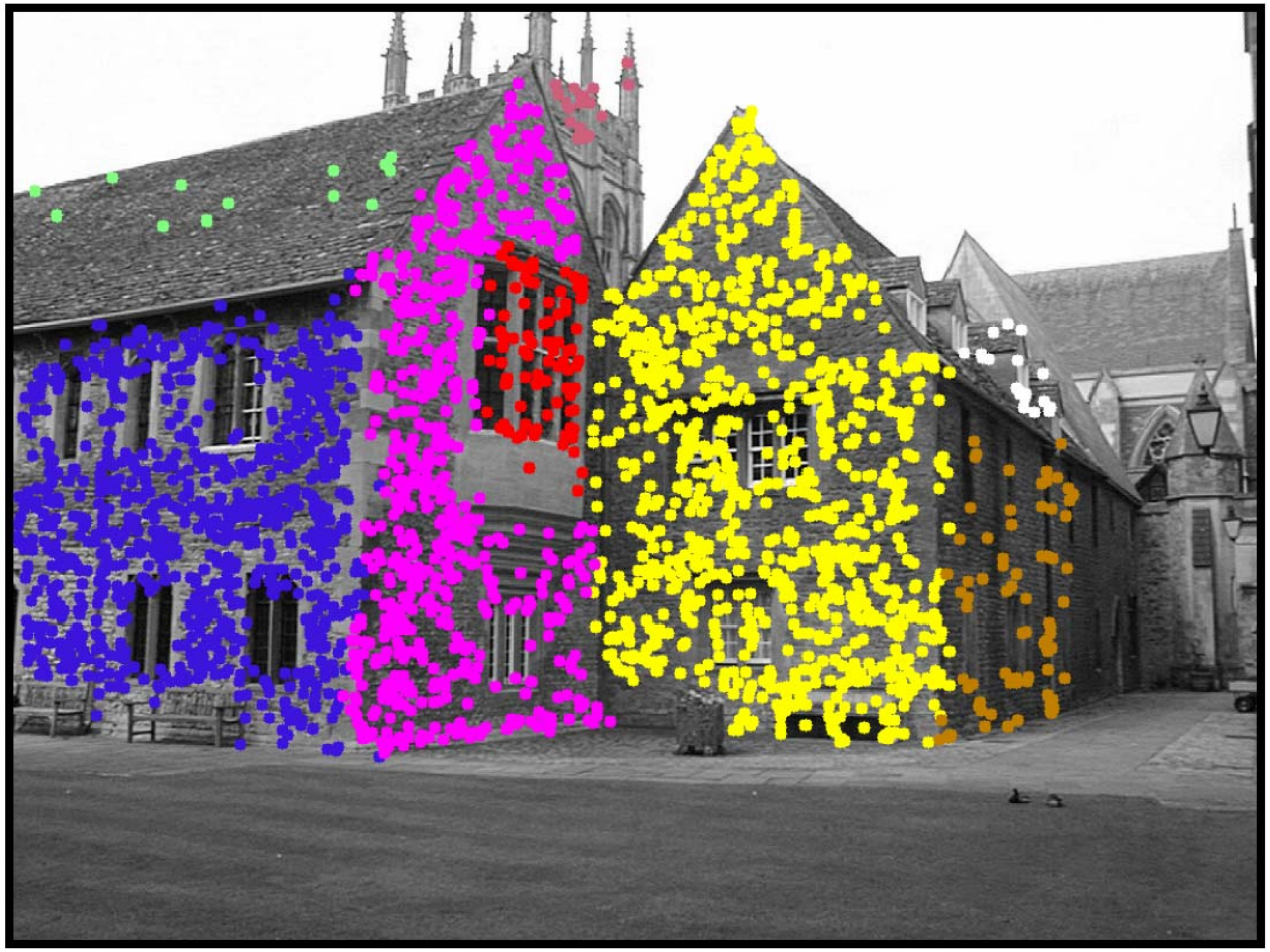
$$Pr(\mathbf{x}_i|\mathbf{y}_i, l_i = k) = \mathrm{Norm}_{\mathbf{x}_i}\left[\mathbf{hom}[\mathbf{y}_i, \mathbf{\Phi}_k], \sigma^2\mathbf{I}\right]$$

- Prior on labels is a Markov random field that encourages nearby labels to be similar

$$Pr(\mathbf{l}) = \frac{1}{Z}\exp\left[-\sum_{i,j\in\mathcal{N}_p} w_{ij}\delta[l_i - l_j]\right]$$

- Model solved with variation of alpha expansion algorithm

Approach 2 – PEaRL

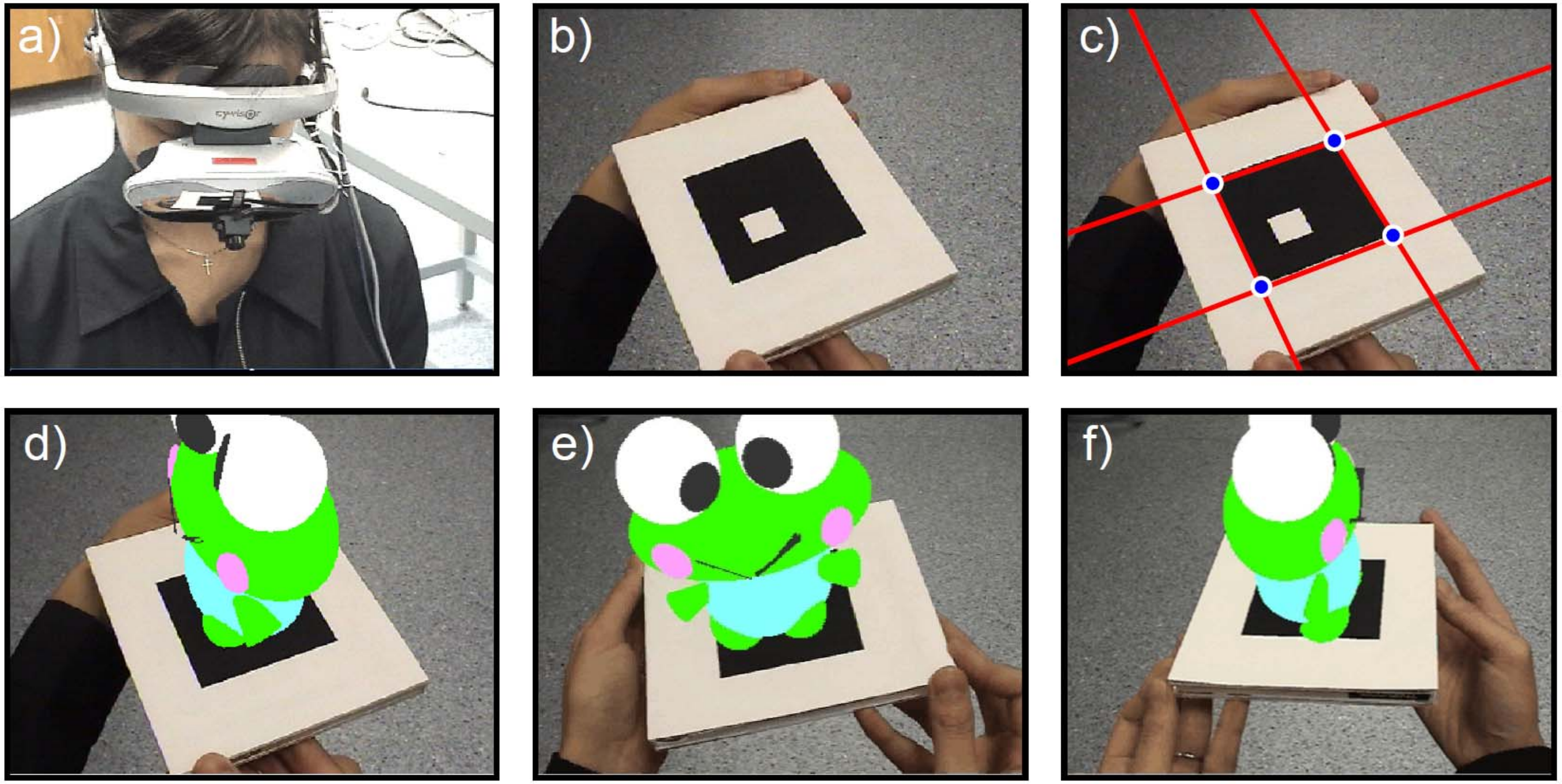


Adapted from Isack & Boykov (forth-
coming). ©2011 Springer.

Structure

- Transformation models
- Learning and inference in transformation models
- Three problems
  - Exterior orientation
  - Calibration
  - Reconstruction
- Properties of the homography
- Robust estimation of transformations
- Applications

Augmented reality tracking



Fast matching of keypoints

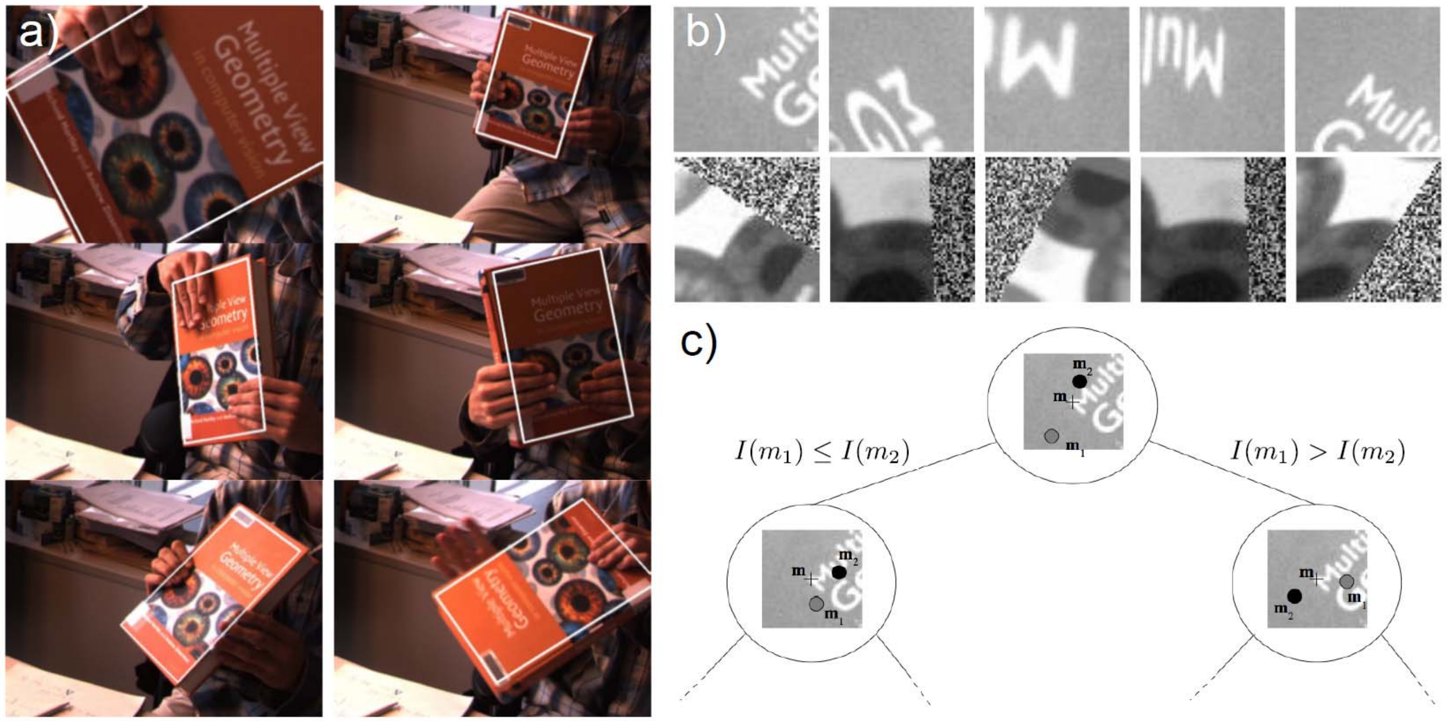
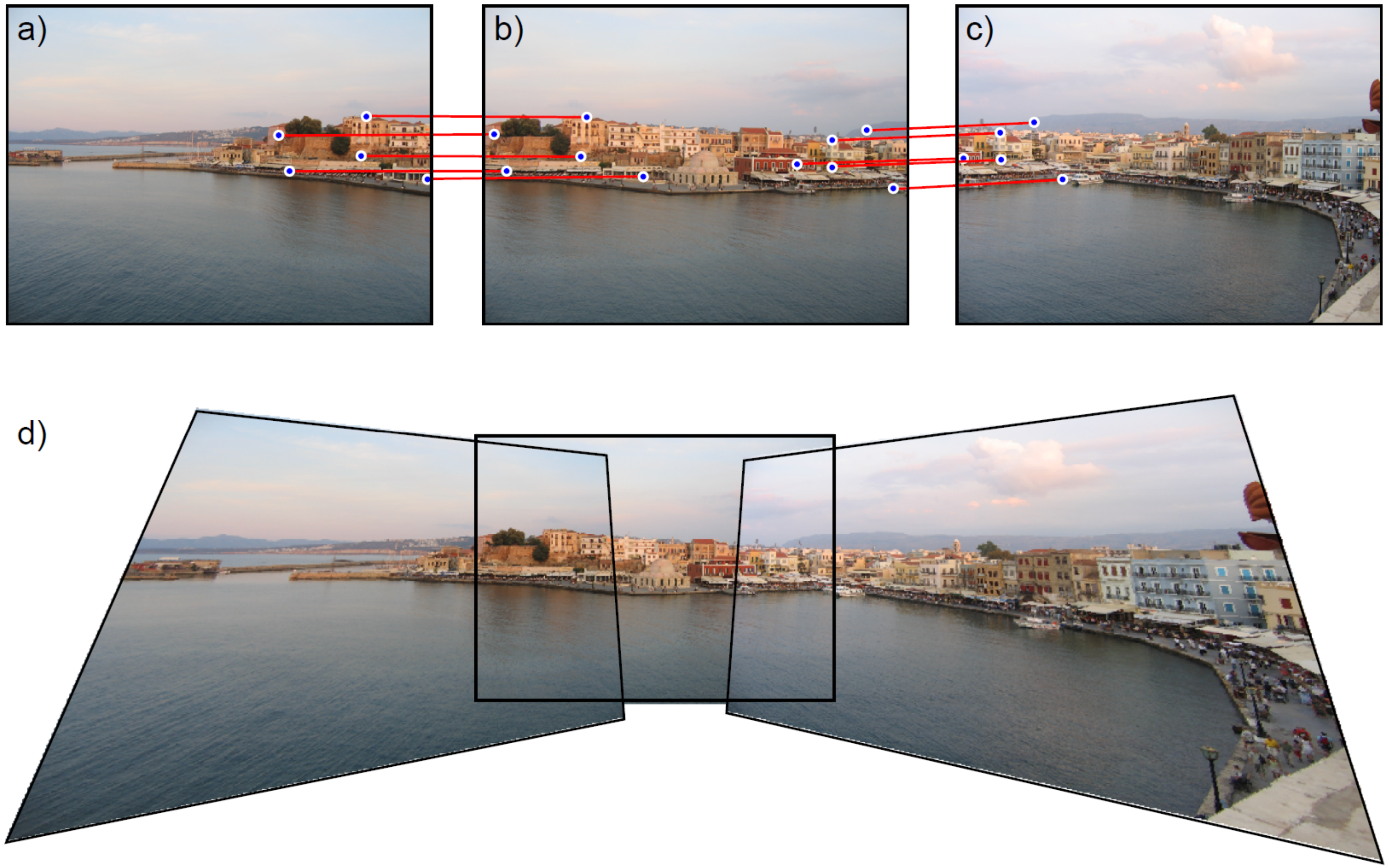


Figure 14.21 Robust tracking using keypoints. a) Lepetit *et al.* (2005) presented a system that automatically tracked objects such as this book. b) In the learning stage, the regions around the keypoints were subjected to a number of random affine transformations. c) Keypoints in the image were classified as belonging to a known keypoint on the object, using a tree-based classifier that compared the intensity at nearby points. Adapted from Lepetit *et al.* (2005). ©2005 IEEE.

Visual panoramas



Conclusions

- Mapping between plane in world and camera is one-to-one
- Takes various forms, but most general is the homography
- Revisited exterior orientation, calibration, reconstruction problems from planes
- Use robust methods to estimate in the presence of outliers