CS 532: 3D Computer Vision
2nd Set of Notes

Instructor: Philippos Mordohai
Webpage: www.cs.stevens.edu/~mordohai
E-mail: Philippos.Mordohai@stevens.edu
Office: Lieb 215
Lecture Outline

• 2D projective transformations
  – Homographies
• Robust estimation
  – RANSAC
• Radial distortion
• Two-view geometry

Based on slides by R. Hartley, A. Zisserman, M. Pollefeys and S. Seitz
Projective Transformations in 2D

Definition:

A projectivity is an invertible mapping $h$ from $\mathbb{P}^2$ to itself such that three points $x_1, x_2, x_3$ lie on the same line if and only if $h(x_1), h(x_2), h(x_3)$ do.

Theorem:

A mapping $h: \mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exist a non-singular 3x3 matrix $H$ such that for any point in $\mathbb{P}^2$ represented by a vector $x$ it is true that $h(x) = Hx$.

Definition: Projective transformation

$$
\begin{pmatrix}
    x'_1 \\
    x'_2 \\
    x'_3
\end{pmatrix} =
\begin{bmatrix}
    h_{11} & h_{12} & h_{13} \\
    h_{21} & h_{22} & h_{23} \\
    h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{pmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{pmatrix}
\quad \text{or} \quad
x' = Hx
$$

8DOF

projectivity=collineation=projective transformation=homography
central projection may be expressed by $x' = Hx$
(application of theorem)
Removing Projective Distortion

select four points in a plane with known coordinates

\[ x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \]
\[ y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \]

\[ x'(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13} \]
\[ y'(h_{31}x + h_{32}y + h_{33}) = h_{21}x + h_{22}y + h_{23} \]

(linear in \( h_{ij} \))

(2 constraints/point, 8DOF \( \Rightarrow \) 4 points needed)

Remarks: no calibration at all necessary,
better ways to compute (see later)
A Hierarchy of Transformations

Projective linear group
  Affine group (last row (0,0,1))
    Euclidean group (upper left 2x2 orthogonal)
      Oriented Euclidean group (upper left 2x2 det 1)

Alternatively, characterize transformation in terms of elements or quantities that are preserved or \textit{invariant}

e.g. Euclidean transformations leave distances unchanged
Class I: Isometries

\[
\begin{pmatrix}
    x' \\
    y' \\
    1
\end{pmatrix}
= \begin{bmatrix}
    \varepsilon \cos \theta & -\sin \theta & t_x \\
    \varepsilon \sin \theta & \cos \theta & t_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}
\]

\(\varepsilon = \pm 1\)

orientation preserving: \(\varepsilon = 1\)
orientation reversing: \(\varepsilon = -1\)

\[x' = H_E \cdot x = \begin{bmatrix}
    R & t \\
    0^T & 1
\end{bmatrix} \cdot x\]

\[R^T R = I\]

3DOF (1 rotation, 2 translation)
special cases: pure rotation, pure translation

**Invariants:** length, angle, area
Class II: Similarities

\[
\begin{pmatrix}
  x' \\
  y' \\
  1
\end{pmatrix} = \begin{bmatrix}
  s \cos \theta & -s \sin \theta & t_x \\
  s \sin \theta & s \cos \theta & t_y \\
  0 & 0 & 1
\end{bmatrix} \begin{pmatrix}
  x \\
  y \\
  1
\end{pmatrix}
\]

(isometry + scale)

\[
x' = H_s x = \begin{bmatrix}
  sR \\
  0^T \\
  1
\end{bmatrix} x
\]

\[R^T R = I\]

4DOF (1 scale, 1 rotation, 2 translation)

also known as equi-form (shape preserving)

metric structure = structure up to similarity (in literature)

**Invariants:** ratios of length, angle, ratios of areas, parallel lines
Class III: Affine Transformations

\[
\begin{pmatrix}
x' \\
y'
\end{pmatrix}
= \begin{bmatrix}
a_{11} & a_{12} & t_x \\
a_{21} & a_{22} & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

\[x' = H_A x = \begin{bmatrix} A & t \end{bmatrix} x\]

\[A = R(\theta)R(-\phi)DR(\phi)\]

\[D = \begin{bmatrix} \lambda_1 & 0 \\
0 & \lambda_2 \end{bmatrix}\]

6DOF (2 scale, 2 rotation, 2 translation)
non-isotropic scaling! (2DOF: scale ratio and orientation)

**Invariants:** parallel lines, ratios of parallel lengths, ratios of areas
Class VI: Projective Transformations

\[ x' = H_P x = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix} x \]

\[ v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}^T \]

8DOF (2 scale, 2 rotation, 2 translation, 2 line at infinity)
Action is non-homogeneous over the plane

**Invariants:** cross-ratio of four points on a line
(ratio of ratio)
### Overview of Transformations

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Degrees of Freedom (dof)</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projective</td>
<td>8</td>
<td>Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio</td>
</tr>
<tr>
<td>Affine</td>
<td>6</td>
<td>Paralllism, ratio of areas, ratio of lengths on parallel lines (e.g. midpoints), linear combinations of vectors (centroids). <strong>The line at infinity</strong> $l_\infty$</td>
</tr>
<tr>
<td>Similarity</td>
<td>4</td>
<td>Ratios of lengths, angles. <strong>The circular points</strong> $I, J$</td>
</tr>
<tr>
<td>Euclidean</td>
<td>3</td>
<td>Lengths, areas.</td>
</tr>
</tbody>
</table>

---

Matrix forms for each transformation:

- **Projective**
  
  \[
  \begin{bmatrix}
  h_{11} & h_{12} & h_{13} \\
  h_{21} & h_{22} & h_{23} \\
  h_{31} & h_{32} & h_{33}
  \end{bmatrix}
  \]

- **Affine**
  
  \[
  \begin{bmatrix}
  a_{11} & a_{12} & t_x \\
  a_{21} & a_{22} & t_y \\
  0 & 0 & 1
  \end{bmatrix}
  \]

- **Similarity**
  
  \[
  \begin{bmatrix}
  sr_{11} & sr_{12} & t_x \\
  sr_{21} & sr_{22} & t_y \\
  0 & 0 & 1
  \end{bmatrix}
  \]

- **Euclidean**
  
  \[
  \begin{bmatrix}
  r_{11} & r_{12} & t_x \\
  r_{21} & r_{22} & t_y \\
  0 & 0 & 1
  \end{bmatrix}
  \]
Homework 1

Warp the basketball court from this image to a new image so that it appears as if the new image was taken from directly above.

What are we missing?
Image Warping

Slides by Steve Seitz
Image Transformations

image filtering: change *range* of image
\[ g(x) = T(f(x)) \]

image warping: change *domain* of image
\[ g(x) = f(T(x)) \]
Parametric (Global) Warping

- Transformation $T$ is a coordinate-changing machine:
  \[ p' = T(p) \]
- What does it mean that $T$ is global?
  - It is the same for any point $p$
  - It can be described by just a few numbers (parameters)
- $T$ is represented as a matrix:
  \[ p' = M \cdot p \]
Image Warping

Given a coordinate transform \((x',y') = h(x,y)\) and a source image \(f(x,y)\), how do we compute a transformed image \(g(x',y') = f(T(x,y))\)?
Forward Warping

Send each pixel $f(x,y)$ to its corresponding location $(x',y') = T(x,y)$ in the second image.

Q: what if the pixel lands “between” two pixels?
Forward Warping

Send each pixel $f(x,y)$ to its corresponding location $(x',y') = T(x,y)$ in the second image.

Q: what if the pixel lands “between” two pixels?
A: Distribute color among neighboring pixels (splatting)
Inverse Warping

- Get each pixel \( g(x',y') \) from its corresponding location \( (x,y) = T^{-1}(x',y') \) in the first image.

- Q: what if pixel comes from “between” two pixels?
Inverse Warping

- Get each pixel $g(x', y')$ from its corresponding location $(x, y) = T^{-1}(x', y')$ in the first image.

- Q: what if pixel comes from “between” two pixels?
  - A: interpolate color value from neighbors
    - Bilinear interpolation typically used
Bilinear Interpolation

\[ f(x, y) = (1 - a)(1 - b) f[i, j] + a(1 - b) f[i + 1, j] + ab f[i + 1, j + 1] + (1 - a)b f[i, j + 1] \]
Forward vs. Inverse Warping

- Which is better?
- ...
Parameter Estimation

Slides by R. Hartley, A. Zisserman and M. Pollefeys
Homography: Number of Measurements Required

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

\[
x' = Hx
\]

\[
\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]

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
Direct Linear Transformation (DLT)

\[ x'_i = Hx_i \quad x'_i \times Hx_i = 0 \]

\[ x'_i = (x'_i, y'_i, w'_i)^T \quad Hx_i = \begin{pmatrix} h^T_{1i} \\ h^T_{2i} \\ h^T_{3i} \end{pmatrix} \]

\[ x'_i \times Hx_i = \begin{pmatrix} y'_i h^3_{1i} x_i - w'_i h^2_{1i} x_i \\ w'_i h^1_{1i} x_i - x'_i h^3_{1i} x_i \\ x'_i h^2_{1i} x_i - y'_i h^1_{1i} x_i \end{pmatrix} \]

\[
\begin{bmatrix}
    0^T & -w'_i x_i^T & y'_i x_i^T \\
    w'_i x_i^T & 0^T & -x'_i x_i^T \\
    -y'_i x_i^T & x'_i x_i^T & 0^T
\end{bmatrix}
\begin{pmatrix}
    h^1 \\
    h^2 \\
    h^3
\end{pmatrix} = 0
\]

\[ A_i h = 0 \]
Direct Linear Transformation (DLT)

Equations are linear in $h$

\[ A_i h = 0 \]

Only 2 of 3 are linearly independent (indeed, 2 eq/pt)

\[
\begin{bmatrix}
0^T & -w_i'x_i^T & y_i'x_i^T \\
w_i'x_i^T & 0^T & -x_i'x_i^T \\
-y_i'x_i^T & x_i'x_i^T & 0^T
\end{bmatrix}
\begin{bmatrix}
h^1 \\
h^2 \\
h^3
\end{bmatrix} = 0
\]

\[ x_i'A_i^1 + y_i'A_i^2 + w_i'A_i^3 = 0 \]
Direct Linear Transformation (DLT)

\[
\begin{bmatrix}
0^T & -w'_ix_i^T & y'_ix_i^T \\
w'_ix_i^T & 0^T & -x'_ix_i^T \\
w'_ix_i^T & 0^T & -x'_ix_i^T \\
\end{bmatrix}
\begin{bmatrix}
h^1 \\
h^2 \\
h^3 \\
\end{bmatrix} = 0
\]

(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 $H$  
  $$Ah = 0$$

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} h = 0$$  
Size of $A$ is 8x9 or 12x9, but rank 8

Trivial solution is $h=0_g^T$ 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

\[
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_n
\end{bmatrix} h = 0
\]

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 \( \| Ah \| \)
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 \( n \) 2x9 matrices \( A_i \) into a single 2nx9 matrix \( A \)
(iii) Obtain SVD of \( A \). Solution for \( h \) is last column of \( V \)
(iv) Determine \( H \) from \( h \)
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 $\sim h$

\[
\begin{bmatrix}
0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\
-x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & x_i x_i' & y_i x_i'
\end{bmatrix}\sim h = \begin{pmatrix}
-w_i y_i' \\
w_i x_i'
\end{pmatrix}
\]

Solve using Gaussian elimination (4 points) or using linear least-squares (more than 4 points)
However, if $h_9 = 0$ this approach fails
Also poor results if $h_9$ close to zero
Therefore, not recommended
Normalizing Transformations

• Since DLT is not invariant to transformations, what is a good choice of coordinates?
  e.g.
  – Translate centroid to origin
  – Scale to a $\sqrt{2}$ average distance to the origin
  – Independently on both images

\[
T_{\text{norm}} = \begin{bmatrix}
  w + h & 0 & w/2 \\
  0 & w + h & h/2 \\
  0 & 0 & 1
\end{bmatrix}^{-1}
\]
Importance of Normalization

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

\[\sim 10^2 \sim 10^2 1 \sim 10^2 \sim 10^2 1 \sim 10^4 \sim 10^4 \sim 10^2\]

orders of magnitude difference!

Monte Carlo simulation
for identity computation based on 5 points
(not normalized ↔ 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 \( \tilde{x}_i = T_{\text{norm}}x_i, \tilde{x}'_i = T'_{\text{norm}}x'_i \)
(ii) Apply DLT algorithm to \( \tilde{x}_i \leftrightarrow \tilde{x}'_i \)
(iii) Denormalize solution \( H = T'^{-1}_{\text{norm}} \tilde{H}T_{\text{norm}} \)
RANSAC

Slides by R. Hartley, A. Zisserman and M. Pollefeys
Robust Estimation

• What if set of matches contains gross outliers?
RANSAC

Objective
Robust fit of model to data set S which contains outliers

Algorithm
(i) Randomly select a sample of $s$ data points from S and instantiate the model from this subset.
(ii) 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 samples and defines the inliers of S.
(iii) If the subset of $S_i$ is greater than some threshold $T$, re-estimate the model using all the points in $S_i$ and terminate.
(iv) If the size of $S_i$ is less than $T$, select a new subset and repeat the above.
(v) After $N$ trials the largest consensus set $S_i$ is selected, and the model is re-estimated using all the points in the subset $S_i$. 
How Many Samples?

Choose \( N \) so that, with probability \( p \), at least one random sample is free from outliers. e.g. \( p = 0.99 \)

\[
(1 - (1 - e)^s)^N = 1 - p
\]

\[
N = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)}
\]

| \(| e \)| | 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 |
Acceptable Consensus Set

• Typically, terminate when inlier ratio reaches expected ratio of inliers

\[ T = (1 - e)n \]
Adaptively Determining the Number of Samples

$e$ is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield $e=0.2$

- $N=\infty$, $\text{sample\_count}=0$
- While $N>\text{sample\_count}$ repeat
  - Choose a sample and count the number of inliers
  - Set $e=1-(\text{number of inliers})/(\text{total number of points})$
  - Recompute $N$ from $e$
  - Increment the $\text{sample\_count}$ by 1
- Terminate

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)}$$
Other robust algorithms

• RANSAC maximizes number of inliers
• LMedS minimizes median error

• Not recommended: case deletion, iterative least-squares, etc.
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
Determine Putative Correspondences

• Compare interest points
  Similarity measure:
  – SAD, SSD, ZNCC in small neighborhood

• If motion is limited, only consider interest points with similar coordinates
Example: robust computation

<table>
<thead>
<tr>
<th>#in</th>
<th>1-e adapt.</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2%</td>
<td>20M</td>
</tr>
<tr>
<td>10</td>
<td>3%</td>
<td>2.5M</td>
</tr>
<tr>
<td>44</td>
<td>16%</td>
<td>6,922</td>
</tr>
<tr>
<td>58</td>
<td>21%</td>
<td>2,291</td>
</tr>
<tr>
<td>73</td>
<td>26%</td>
<td>911</td>
</tr>
<tr>
<td>151</td>
<td>56%</td>
<td>43</td>
</tr>
</tbody>
</table>

Interest points (500/image)
(640x480)

Putative correspondences (268)
(Best match, SSD<20, ±320)
Outliers (117)
($t=1.25$ pixel; 43 iterations)

Inliers (151)

Final inliers (262)
Radial Distortion and Undistortion

Slides by R. Hartley, A. Zisserman and M. Pollefeys
Radial Distortion

short and long focal length

radial distortion  correction  linear image
Typical Undistortion Model

Correction of distortion

\[ \hat{x} = x_c + L(r)(x - x_c) \quad \hat{y} = y_c + L(r)(y - y_c) \]

Choice of the distortion function and center

\[ x = x_o + (x_o - c_x)(K_1r^2 + K_2r^4 + \ldots) \]
\[ y = y_o + (y_o - c_y)(K_1r^2 + K_2r^4 + \ldots) \]

\[ r = (x_o - c_x)^2 + (y_o - c_y)^2 \]

Computing the parameters of the distortion function

(i) Minimize with additional unknowns
(ii) Straighten lines
(iii) …
Why Undistort?

\[
(x_d, y_d) = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}
\]

\[
(\tilde{x}, \tilde{y}, 1)^\top = [I \mid 0] \mathbf{X}_{\text{cam}}
\]
Two-View Geometry

Slides by R. Hartley, A. Zisserman and M. Pollefeys
Three questions:

(i) **Correspondence geometry:** Given an image point $x$ in the first image, how does this constrain the position of the corresponding point $x'$ in the second image?

(ii) **Camera geometry (motion):** Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1,\ldots,n$, what are the cameras $P$ and $P'$ for the two views?

(iii) **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras $P$, $P'$, what is the position of (their pre-image) $X$ in space?
The Epipolar Geometry

C, C’, x, x’ and X are coplanar
What if only C, C’, x are known?
The Epipolar Geometry

All points on \( \pi \) project on \( l \) and \( l' \)
The Epipolar Geometry

Family of planes \( \pi \) and lines \( l \) and \( l' \)
Intersection in \( e \) and \( e' \)
The Epipolar Geometry

epipoles e, e’
= intersection of baseline with image plane
= projection of projection center in other image
= vanishing point of camera motion direction

an epipolar plane = plane containing baseline (1-D family)

an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)
Example: Converging Cameras
Example: Motion Parallel to Image Plane

(e at infinity)  

(e' at infinity)  

(simple for stereo → rectification)
Example: Forward Motion
The Fundamental Matrix F

algebraic representation of epipolar geometry

\[ x \mapsto l' \]

we will see that mapping is a (singular) correlation (i.e. projective mapping from points to lines) represented by the fundamental matrix F
The Fundamental Matrix F

correspondence condition

The fundamental matrix satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ in the two images

$$x'^T F x = 0 \quad (x'^T 1' = 0)$$
The Fundamental Matrix $F$

\[ X(\lambda) = P^+x + \lambda C \]  
\[ l = P'C \times P'P^+x \]  
\[ F = [e']_x P'P^+ \]

(note: doesn’t work for $C=C' \Rightarrow F=0$)
The Fundamental Matrix $F$

$F$ is the unique $3 \times 3$ rank 2 matrix that satisfies $x'^{T}Fx=0$ for all $x \leftrightarrow x'$

(i) **Transpose:** if $F$ is fundamental matrix for $(P,P')$, then $F^T$ is fundamental matrix for $(P',P)$

(ii) **Epipolar lines:** $l'=Fx$ & $l=F^Tx'$

(iii) **Epipoles:** on all epipolar lines, thus $e'^{T}Fx=0$, $\forall x \Rightarrow e'^{T}F=0$, similarly $Fe=0$

(iv) $F$ has 7 d.o.f., i.e. $3 \times 3-1($homogeneous$)-1($rank2$)$

(v) $F$ is a correlation, projective mapping from a point $x$ to a line $l'=Fx$ (not a proper correlation, i.e. not invertible)
Two View Geometry Computation: Linear Algorithm

For every match \((m,m')\):

\[ x^T F x = 0 \]

\[ x' x f_{11} + x' y f_{12} + x' f_{13} + y' x f_{21} + y' y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0 \]

separate known from unknown

\[
\begin{bmatrix}
\[ x' x, x' y, x', y' x, y' y, y', x, y, 1 \]
\end{bmatrix} \begin{bmatrix}
\[ f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33} \]
\end{bmatrix}^T = 0
\]

(data) (unknowns) (linear)

\[
\begin{bmatrix}
\[ x'_1 x_1, x'_1 y_1, x'_1, y'_1 x_1, y'_1 y_1, y'_1, x_1, y_1, 1 \]
\vdots \vdots \vdots \vdots \vdots \vdots \vdots \vdots \vdots \vdots \\
\[ x'_n x_n, x'_n y_n, x'_n, y'_n x_n, y'_n y_n, y'_n x_n, y_n, 1 \]
\end{bmatrix} f = 0
\]

\[ A f = 0 \]
Benefits from having F

• Given a pixel in one image, the corresponding pixel has to lie on epipolar line

• Search space reduced from 2-D to 1-D
Image Pair Rectification

simplify stereo matching
by warping the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines

map epipole $e$ to $(1,0,0)$

try to minimize image distortion

problem when epipole in (or close to) the image

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = H_e$$
Planar Rectification

(standard approach)

Bring two views to standard stereo setup
(moves epipole to $\infty$)
(not possible when in/close to image)