Lecture 9: 2D Transformation & Alignment

COMP 590/776: Computer Vision
Instructor: Soumyadip (Roni) Sengupta
TA: Mykhailo (Misha) Shvets
Steps of creating a Panorama

This is your next homework assignment!
Why extract features? (last week)

• Motivation: panorama stitching
  • We have two images – how do we combine them?
Why extract features? (last week)

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
How to align and combine two images?

• Motivation: panorama stitching
  • We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
Step 3: align images
Step 4: blending images

Last Week
This Week
What is the geometric relationship between these two images?
What is the geometric relationship between these two images?

Very important for creating mosaics!
First, we need to know what this transformation is.
Second, we need to figure out how to compute it using feature matches.
Image Warping

• image filtering: change range of image
  \[ g(x) = h(f(x)) \]

• image warping: change domain of image
  \[ g(x) = f(h(x)) \]
Image Warping

• image filtering: change range of image
  \[ g(x) = h(f(x)) \]

• image warping: change domain of image
  \[ g(x) = f(h(x)) \]
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Parametric (global) warping

- Examples of parametric warps:

  - translation
  - rotation
  - aspect
Parametric (global) warping

Transformation $T$ is a coordinate-changing machine:

$$p' = T(p)$$

What does it mean that $T$ is global?

- Is the same for any point $p$
- Can be described by just a few numbers (parameters)

Let’s consider linear transforms (can be represented by a 2x2 matrix):

$$p' = Tp$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix}$$
Common linear transformations

• Rotation by angle $\theta$ (about the origin)

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

What is the inverse?

For rotations:

$$\mathbf{R}^{-1} = \mathbf{R}^T$$
2-D Rotation

\[
x = r \cos(\phi)
\]
\[
y = r \sin(\phi)
\]
\[
x' = r \cos(\phi + \theta)
\]
\[
y' = r \sin(\phi + \theta)
\]
2-D Rotation

\[ x = r \cos(\phi) \]
\[ y = r \sin(\phi) \]
\[ x' = r \cos(\phi + \theta) \]
\[ y' = r \sin(\phi + \theta) \]

Trig Identity…
\[ x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta) \]
\[ y' = r \sin(\phi) \cos(\theta) + r \cos(\phi) \sin(\theta) \]
2-D Rotation

\[
x = r \cos (\phi) \\
y = r \sin (\phi) \\
x' = r \cos (\phi + \theta) \\
y' = r \sin (\phi + \theta)
\]

Trig Identity…
\[
x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta) \\
y' = r \sin(\phi) \cos(\theta) + r \cos(\phi) \sin(\theta)
\]

Substitute…
\[
x' = x \cos(\theta) - y \sin(\theta) \\
y' = x \sin(\theta) + y \cos(\theta)
\]
2-D Rotation

• This is easy to capture in matrix form:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} =
\begin{bmatrix}
  \cos(\theta) & -\sin(\theta) \\
  \sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]
2x2 Matrices

• What types of transformations can be represented with a 2x2 matrix?

**2D Identity?**

\[
x' = x \\
y' = y
\]

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
\]

**2D Scale around (0,0)?**

\[
x' = s_x \cdot x \\
y' = s_y \cdot y
\]

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
\]
2x2 Matrices

• What types of transformations can be represented with a 2x2 matrix?

2D Rotate around (0,0)?

\[
x' = \cos \Theta \cdot x - \sin \Theta \cdot y \\
y' = \sin \Theta \cdot x + \cos \Theta \cdot y
\]

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
\]

2D Shear?

\[
x' = x + \text{sh}_x \cdot y \\
y' = \text{sh}_y \cdot x + y
\]

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \text{sh}_x \\ \text{sh}_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
\]
2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

**2D Mirror about Y axis?**

\[
\begin{align*}
    x' &= -x \\
    y' &= y
\end{align*}
\]

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} =
\begin{bmatrix}
    -1 & 0 \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix}
\]

**2D Mirror over (0,0)?**

\[
\begin{align*}
    x' &= -x \\
    y' &= -y
\end{align*}
\]

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} =
\begin{bmatrix}
    -1 & 0 \\
    0 & -1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix}
\]
2x2 Matrices

• What types of transformations can be represented with a 2x2 matrix?

2D Translation?

\[ x' = x + t_x \]  
\[ y' = y + t_y \]  

NO!

Translation is not a linear operation on 2D coordinates

Only linear 2D transformations can be represented with a 2x2 matrix
All 2D Linear Transformations

• Linear transformations are combinations of ...
  • Scale,
  • Rotation,
  • Shear, and
  • Mirror

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
a & b \\
c & d
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix}
\]

• Properties of linear transformations:
  • Origin maps to origin
  • Lines map to lines
  • Parallel lines remain parallel
  • Ratios are preserved
  • Closed under composition

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
a & b & e & f \\
c & d & g & h \\
i & j & k & l
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix}
\]
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Homogeneous Coordinates

• Q: How can we represent translation as a 3x3 matrix?

\[ x' = x + t_x \]
\[ y' = y + t_y \]

• Homogeneous coordinates
  • represent coordinates in 2 dimensions with a 3-vector
Homogeneous coordinates

Trick: add one more coordinate:

\[(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\]

homogeneous image coordinates

Converting \textit{from} homogeneous coordinates

\[
\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)
\]
Homogeneous Coordinates

- Add a 3rd coordinate to every 2D point
  - \((x, y, w)\) represents a point at location \((x/w, y/w)\)
  - \((x, y, 0)\) represents a point at infinity
  - \((0, 0, 0)\) is not allowed

Convenient coordinate system to represent many useful transformations

\[(2,1,1)\] or \[(4,2,2)\] or \[(6,3,3)\]
Homogeneous Coordinates

• Q: How can we represent translation as a 3x3 matrix?

\[ x' = x + t_x \]
\[ y' = y + t_y \]

• A: Using the rightmost column:

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x + t_x \\
y + t_y \\
1
\end{bmatrix}
\]

\[ \text{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \]
Affine transformations

\[ T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \]

any transformation represented by a 3x3 matrix with last row \([0 \ 0 \ 1]\) we call an affine transformation.
Basic affine transformations

\[
\begin{bmatrix}
x'
\end{bmatrix} = \begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x
y
1
\end{bmatrix}
\]

Translate

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

2D in-plane rotation

\[
\begin{bmatrix}
x'
\end{bmatrix} = \begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x
y
1
\end{bmatrix}
\]

Scale

\[
\begin{bmatrix}
x'
\end{bmatrix} = \begin{bmatrix}
1 & sh_x & 0 \\
sh_y & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x
y
1
\end{bmatrix}
\]

Shear
Matrix Composition

• Transformations can be combined by matrix multiplication

\[
\begin{pmatrix}
  x' \\
  y' \\
  w'
\end{pmatrix} = \begin{pmatrix}
  1 & 0 & tx \\
  0 & 1 & ty \\
  0 & 0 & 1
\end{pmatrix}\begin{pmatrix}
  \cos \Theta & -\sin \Theta & 0 \\
  \sin \Theta & \cos \Theta & 0 \\
  0 & 0 & 1
\end{pmatrix}\begin{pmatrix}
  sx & 0 & 0 \\
  0 & sy & 0 \\
  0 & 0 & 1
\end{pmatrix}\begin{pmatrix}
  x \\
  y \\
  w
\end{pmatrix}
\]

\[p' = T(t_x,t_y) \quad R(\Theta) \quad S(s_x,s_y) \quad p\]

Does the order of multiplication matter?
Affine transformations

- Affine transformations are combinations of...
  - Linear transformations, and
  - Translations

\[
\begin{bmatrix}
  x' \\
  y' \\
  w
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

- Properties of affine transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved
  - Closed under composition
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Where do we go from here?

affine transformation

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix}
\]

what happens when we mess with this row?
Projective Transformations *aka* Homographies *aka* Planar Perspective Maps

\[
H = \begin{bmatrix}
    a & b & c \\
    d & e & f \\
    g & h & 1
\end{bmatrix}
\]

Called a *homography* (or *planar perspective map*)

Any two images of the same planar surface in 3D space are related by a **homography** (assuming a **pinhole camera model**).
Homographies

\[
\begin{bmatrix}
x' \\
y' \\
w'
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix} = \sim \begin{bmatrix}
\frac{ax+by+c}{gx+hy+1} \\
\frac{dx+ey+f}{gx+hy+1} \\
1
\end{bmatrix}
\]

What happens when the denominator is 0?
Points at infinity

\[(x, y, 0)\]  \[\rightarrow\]  \[(x', y', 1)\]
Image warping with homographies

Image plane in front

black area where no pixel maps to

$H_1$

$H_2$
Homographies (Projective Transformation)

• Homographies ...
  • Affine transformations, and
  • Projective warps

\[
\begin{bmatrix}
  x' \\
  y' \\
  w'
\end{bmatrix}
= \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_i' \\
  y_i' \\
  1
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
  h_{00} & h_{01} & h_{02} \\
  h_{10} & h_{11} & h_{12} \\
  h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix}
\]

where the length of the vector \([h_{00} h_{01} \ldots h_{22}]\) is 1.

• Properties of projective transformations:
  • Origin does not necessarily map to origin
  • Lines map to lines
  • Parallel lines do not necessarily remain parallel
  • Ratios are not preserved
  • Closed under composition
2D image transformations

<table>
<thead>
<tr>
<th>Name</th>
<th>Matrix</th>
<th># D.O.F.</th>
<th>Preserves:</th>
<th>Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>$\begin{bmatrix} I &amp; t \end{bmatrix}_{2 \times 3}$</td>
<td>2</td>
<td>orientation + ⋅ ⋅ ⋅</td>
<td></td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$\begin{bmatrix} R &amp; t \end{bmatrix}_{2 \times 3}$</td>
<td>3</td>
<td>lengths + ⋅ ⋅ ⋅</td>
<td></td>
</tr>
<tr>
<td>similarity</td>
<td>$\begin{bmatrix} sR &amp; t \end{bmatrix}_{2 \times 3}$</td>
<td>4</td>
<td>angles + ⋅ ⋅ ⋅</td>
<td></td>
</tr>
<tr>
<td>affine</td>
<td>$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$</td>
<td>6</td>
<td>parallelism + ⋅ ⋅ ⋅</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$</td>
<td>8</td>
<td>straight lines</td>
<td></td>
</tr>
</tbody>
</table>
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Computing transformations

• Given a set of matches between images A and B
  • How can we compute the transform T from A to B?

• Find transform T that best “agrees” with the matches
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Simple case: translations

How do we solve for $(x_t, y_t)$?
Simple case: translations

Displacement of match $i = (x_i' - x_i, y_i' - y_i)$

$$(x_t, y_t) = \left( \frac{1}{n} \sum_{i=1}^{n} x_i' - x_i, \frac{1}{n} \sum_{i=1}^{n} y_i' - y_i \right)$$
Least squares: linear regression

\[ y = mx + b \]

\( (y_i, x_i) \)

\( y = mx + b \)
Linear regression

$$Cost(m, b) = \sum_{i=1}^{n} |y_i - (mx_i + b)|^2$$
Linear regression

\[
\begin{bmatrix}
  x_1 & 1 \\
  x_2 & 1 \\
  \vdots \\
  x_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
  m \\
  b \\
\end{bmatrix}
= 
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n \\
\end{bmatrix}
\]
Another view

- System of linear equations
  - What are the knowns? Unknowns?
  - How many unknowns? How many matches do we need?

\[
\begin{align*}
  x_i + x_t &= x_i' \\
  y_i + y_t &= y_i'
\end{align*}
\]
• Problem: more equations than unknowns
  • “Overdetermined” system of equations
  • We will find the least squares solution
Least squares formulation

• For each point
  \[(x_i, y_i)\]
  \[x_i + x_t = x'_i\]
  \[y_i + y_t = y'_i\]

• we define the *residuals* as

  \[r_{x_i}(x_t) = (x_i + x_t) - x'_i\]
  \[r_{y_i}(y_t) = (y_i + y_t) - y'_i\]
Least squares formulation

• Goal: minimize sum of squared residuals

\[ C(x_t, y_t) = \sum_{i=1}^{n} (r_{x_i}(x_t)^2 + r_{y_i}(y_t)^2) \]

• “Least squares” solution

• For translations, is equal to mean (average) displacement
Least squares formulation

• Can also write as a matrix equation

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 \\
0 & 1 \\
\vdots \\
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x_t \\
y_t \\
\end{bmatrix}
=
\begin{bmatrix}
x'_1 - x_1 \\
y'_1 - y_1 \\
x'_2 - x_2 \\
y'_2 - y_2 \\
\vdots \\
x'_n - x_n \\
y'_n - y_n \\
\end{bmatrix}
\]
Least squares

\[ At = b \]

• Find \( t \) that minimizes

\[ \| At - b \|^2 \]

• To solve, form the normal equations

\[ A^T At = A^T b \]

\[ t = (A^T A)^{-1} A^T b \]
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Affine transformations

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

- How many unknowns?
- How many equations per match?
- How many matches do we need?
Affine transformations

• Residuals:
  \[ r_{x_i}(a, b, c, d, e, f) = (ax_i + by_i + c) - x_i' \]
  \[ r_{y_i}(a, b, c, d, e, f) = (dx_i + ey_i + f) - y_i' \]

• Cost function:
  \[
  C(a, b, c, d, e, f) = \sum_{i=1}^{n} \left( r_{x_i}(a, b, c, d, e, f)^2 + r_{y_i}(a, b, c, d, e, f)^2 \right)
  \]
Affine transformations

• Matrix form

\[
\begin{bmatrix}
    x_1 & y_1 & 1 & 0 & 0 & 0 \\
    0 & 0 & 0 & x_1 & y_1 & 1 \\
    x_2 & y_2 & 1 & 0 & 0 & 0 \\
    0 & 0 & 0 & x_2 & y_2 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_n & y_n & 1 & 0 & 0 & 0 \\
    0 & 0 & 0 & x_n & y_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
    a \\
    b \\
    c \\
    d \\
    e \\
    f \\
\end{bmatrix}
= 
\begin{bmatrix}
    x'_1 \\
    y'_1 \\
    x'_2 \\
    y'_2 \\
    \vdots \\
    x'_n \\
    y'_n \\
\end{bmatrix}
\]
Today’s class

• Types of 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)

• Computing 2D Transformations
  • Linear
  • Affine
  • Perspective (Homography)
Homographies

To unwarp (rectify) an image

• solve for homography $H$ given $p$ and $p'$
• solve equations of the form: $wp' = Hp$
  – linear in unknowns: $w$ and coefficients of $H$
  – $H$ is defined up to an arbitrary scale factor
  – how many matches are necessary to solve for $H$?
Solving for homographies

\[
\begin{bmatrix}
    x'_i \\
    y'_i \\
    1
\end{bmatrix} = \begin{bmatrix}
    h_{00} & h_{01} & h_{02} \\
    h_{10} & h_{11} & h_{12} \\
    h_{20} & h_{21} & h_{22}
\end{bmatrix} \begin{bmatrix}
    x_i \\
    y_i \\
    1
\end{bmatrix}
\]

\[
x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}
\]

\[
y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}
\]

\[
x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}
\]

\[
y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}
\]

Not linear!
Solving for homographies

\[
x_i'(h_{20} x_i + h_{21} y_i + h_{22}) = h_{00} x_i + h_{01} y_i + h_{02}
\]
\[
y_i'(h_{20} x_i + h_{21} y_i + h_{22}) = h_{10} x_i + h_{11} y_i + h_{12}
\]

\[
\begin{bmatrix}
  x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\
  0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i
\end{bmatrix}
\begin{bmatrix}
  h_{00} \\
  h_{01} \\
  h_{02} \\
  h_{10} \\
  h_{11} \\
  h_{12} \\
  h_{20} \\
  h_{21} \\
  h_{22}
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0
\end{bmatrix}
\]
Solving for homographies

Defines a least squares problem: minimize \( \|Ah - 0\|^2 \)

- Since \( h \) is only defined up to scale, solve for unit vector \( \hat{h} \)
- \( \text{Rank}(A) = ? \)
Solving for homographies

\[ \mathbf{A} \times \mathbf{h} = \mathbf{0} \]

\[ \text{Rank}(\mathbf{A}) = 8 \]

- Calculate Singular Value decomposition of \( \mathbf{A} \) -> \( \mathbf{A} = \mathbf{UDV}^T \)
- \( \mathbf{U} \) is \( 2n \times r \); \( \mathbf{D} \) is \( r \times r \) (diagonal matrix with singular values); \( \mathbf{V} \) is \( 9 \times r \), where \( r = \text{rank}(\mathbf{A}) \).
- In ideal case \( r = \text{rank}(\mathbf{A}) = 8 \), \( \mathbf{h} \) is in null-space of \( \mathbf{A} \).
- In practice \( \text{rank}(\mathbf{A}) = 9 \), thus the goal is to find the smallest singular value of \( \mathbf{A} \).
- Smallest singular value of \( \mathbf{A} \) also indicates how well the homography can be estimated.

- Calculate \( \mathbf{A}^T \mathbf{A} = \mathbf{VDU}^T \mathbf{UDV}^T = \mathbf{VD}^2 \mathbf{V}^{-1} \) (Since, \( \mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I} \))
- This is eigen-decomposition of \( \mathbf{A}^T \mathbf{A} \)
- Smallest singular value of \( \mathbf{A} \) -> Smallest eigen value of \( \mathbf{A}^T \mathbf{A} \).
- Solution: optimal \( \mathbf{h} = \) eigenvector of \( \mathbf{A}^T \mathbf{A} \) with smallest eigenvalue.
Computing transformations
Image alignment algorithm

Given images A and B

1. Compute image features for A and B
2. Match features between A and B
3. Compute homography between A and B using least squares on set of matches

What could go wrong?
Outliers

Lead to next class on RANSAC
Slide Credits

• **CS5670, Introduction to Computer Vision**, Cornell Tech, by Noah Snavely.


• **Fall 2022 CS 543/ECE 549: Computer Vision**, UIUC, by Svetlana Lazebnik.