Panorama Stitching
Panorama stitching

- Task: join two images (acquired from the same point) into a panorama
- Why do we do it in this course
  - includes many processing steps we have discussed
  - not as messy as relative orientation
  - in fact several “panoramic cameras” are built that way
Panorama stitching

• Task: join two images (acquired from the same point) into a panorama

• Why do we do it in this course
  • includes many processing steps we have discussed
  • not as messy as relative orientation
  • in fact several “panoramic cameras” are built that way
Recap: homography

- Projection with a straight ray from a 2D plane to another 2D plane (the retina of the camera)
- No dimensionality reduction
- Ray intersects each plane exactly once
- Inverse projection along the same ray bundle
Recap: homography

- In homogeneous coordinates given by (3x3) matrix $H$
  - $H$ is called a **homography** (or **projectivity**)
  - the most general projective 2D transformation
  - maps a square to a general quadrangle
  - has 8 degrees of freedom
  - $H$ is a one-to-one mapping, hence invertible

\[ x = H x_\pi \quad x_\pi = H^{-1} x \]
Rotating camera

- Relation between two images, if the camera only rotates (baseline = 0)
  - Projection rays are identical
  - Ray maps directly from one image plane to the other
  - The two images are related by a homography

\[
x = K[I|0]X = K\tilde{X}
\]
\[
x' = K'[R|0]X = K'R\tilde{X}
\]
\[
\tilde{X} = K^{-1}x
\]
\[
x' = K'RK^{-1}x
\]
\[
x' = Hx
\]
Recap: estimating the homography

- Eight unknowns
- Two linearly independent equations per point
- Need 4 point correspondences

\[ \lambda x' = Hx \quad , \quad x' \propto Hx \]
Recap: estimating the homography

- Eight unknowns
- Two linearly independent equations per point
- Need 4 point correspondences

\[ x' \propto Hx \quad \rightarrow \quad Hx \times x' = 0 \]

\[
\begin{bmatrix}
0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \\
-x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x'
\end{bmatrix}
\begin{bmatrix}
h_{11} \\
h_{12} \\
h_{13} \\
h_{21} \\
h_{22} \\
h_{23} \\
h_{31} \\
h_{32} \\
h_{33}
\end{bmatrix} = 0
\]
Recap: estimating the homography

- Stack equations from at least 4 correspondences
- Remember
  - a homogeneous linear equation system
  - solve s.t. $\|h\|=1$, using SVD

\[
Ah = 0
\]

\[
[U, D, V] = \text{svd}(A)
\]

\[
h = \begin{bmatrix}
  v_{19} \\
v_{29} \\
  \vdots \\
v_{99}
\end{bmatrix}
\]
Estimating the homography

- A note on numerical stability
  - Coefficients of an equation system should be in the same order of magnitude so as not to lose significant digits
  - in pixels: $xx' \sim 1e6$
  - Conditioning: scale and shift points to be in $[-1..1]$
  - apply inverse transform to estimated homography
  - a general rule, not only for homography

\[
\begin{bmatrix}
0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \\
-x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x'
\end{bmatrix}
\begin{bmatrix}
h_{11} \\
h_{12} \\
h_{13} \\
h_{21} \\
h_{22} \\
h_{23} \\
h_{31} \\
h_{32} \\
h_{33}
\end{bmatrix} = 0
\]
Estimating the homography

- A note on numerical stability
  - Conditioning: scale and shift points to be in [-1..1]

\[
\begin{align*}
  s &= \frac{1}{2} \max_i (\|x_i\|) \\
  t &= \text{mean}(x_i) \\
  s' &= \frac{1}{2} \max_i (\|x'_i\|) \\
  t' &= \text{mean}(x'_i)
\end{align*}
\]

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

\[
T' = \begin{bmatrix}
  \frac{1}{s'} & 0 & -\frac{t'_x}{s'} \\
  0 & \frac{1}{s'} & -\frac{t'_y}{s'} \\
  0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{aligned}
  u &= Tx \\
  u' &= T'x' \\
  u' &= \bar{H}u \\
  T'x' &= \bar{H}Tx \\
  H &= (T')^{-1} \bar{H}T
\end{aligned}
\]
Estimating the homography

• The cookbook recipe

• condition all image coordinates
• RANSAC loop: iterate
  • pick 4 correspondences
  • build equations, estimate homography
  • transform all points \( x \)
  • measure Euclidean distance to all points \( x' \)
  • count inliers (scale down threshold too!)
• re-estimate final homography
• undo conditioning
Recap: bilinear resampling

- **Backward resampling**
  - create empty target area
  - loop through target area
    - transform target pixels to source image with inverse homography
    - interpolate RGB values from neighbouring pixels
    - write grey-value to target pixel
Recap: bilinear resampling

- Interpolating intensities for non-integer coordinates
  - bilinear interpolation in (2 x 2) neighbourhood
  - linear in each coordinate, quadratic surface

\[ I(x, y) = (1 - \Delta x) \cdot (1 - \Delta y) \cdot I(x_-, y_-) + \Delta x \cdot (1 - \Delta y) \cdot I(x_+, y_-) + (1 - \Delta x) \cdot \Delta y \cdot I(x_-, y_+) + \Delta x \cdot \Delta y \cdot I(x_+, y_+) \]
Recap: blending

• Intensity differences
  • images may have different brightness, contrast, ...
  • intensities vary between images (lighting changes, exposure, ...)

![Images showing intensity differences](image1.png) ![Images showing intensity differences](image2.png)
Recap: blending

• Linear blending
  • Weighted averaging of intensity values
  • weights decrease linearly towards the boundary
Recap: blending

- Linear blending
  - Weighted averaging of intensity values
  - weights decrease linearly towards the boundary
Application: AutoStitch

• Given a collection of unordered photographs...
Application: AutoStitch

• ... automatically create a panorama
  • find images related by a homography
  • extract relative rotation
  • remap to a sphere
  • blend colours along seams
Application: AutoStitch

- ... automatically create a panorama
  - find images related by a homography
  - extract relative rotation
  - remap to a sphere
  - blend colours along seams
Application: AutoStitch

• ... automatically create a panorama
  • find images related by a homography
  • extract relative rotation
  • remap to a sphere
  • blend colours along seams