Image Alignment and Mosaicing
Image Alignment Applications

- **Local alignment:**
  - Tracking
  - Stereo

- **Global alignment:**
  - Camera jitter elimination
  - Image enhancement
  - Panoramic mosaicing
Image Enhancement

Original

Enhanced
Panoramic Mosaicing
Gigapixel panoramas & images

Mapping / Tourism / WWT

Medical Imaging
Panoramic Mosaicing

1. Align images
2. Merge overlapping regions
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Optical Flow for Image Registration

• Compute local matches
• Least-squares fit to motion model
• Problem: outliers
Outlier Rejection

- Robust estimation: tolerant of outliers
- In general, methods based on absolute value rather than square:

\[ \text{minimize } \sum |x_i - f|, \text{ not } \sum (x_i - f)^2 \]
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Correlation / Search Methods

• Assume translation only
• Given images $I_1, I_2$
• For each translation $(t_x, t_y)$ compute

$$c(I_1, I_2, t) = \sum_i \sum_j \psi(I_1(i, j), I_2(i-t_x, j-t_y))$$

• Select translation that maximizes $c$
• Depending on window size, local or global
Cross-Correlation

- Statistical definition of correlation:

$$\psi(u, v) = uv$$

- Disadvantage: sensitive to local variations in image brightness
Normalized Cross-Correlation

- Normalize to eliminate brightness sensitivity:

\[ \psi(u, v) = \frac{(u - \bar{u})(v - \bar{v})}{\sigma_u \sigma_v} \]

where

\[ \bar{u} = \text{average}(u) \]

\[ \sigma_u = \text{standard deviation}(u) \]
Sum of Squared Differences

- More intuitive measure:
  \[ \psi(u, v) = -(u - v)^2 \]

- Negative: higher values \(\rightarrow\) greater similarity

- Expand:
  \[ -(u - v)^2 = -u^2 - v^2 + 2uv \]
Local vs. Global

• Correlation with local windows not too expensive
• High cost if window size = whole image
• But computation looks like convolution
  – FFT to the rescue!
Fourier Transform with Translation

\[ F(f(x + \Delta x, y + \Delta y)) = F(f(x, y)) e^{i(\omega_x \Delta x + \omega_y \Delta y)} \]
Fourier Transform with Translation

• Therefore, if $I_1$ and $I_2$ differ by translation,

$$F(I_1(x, y)) = F(I_2(x, y)) e^{i(\omega_x \Delta x + \omega_y \Delta y)}$$

$$\frac{F_1}{F_2} = e^{i(\omega_x \Delta x + \omega_y \Delta y)}$$

• So, $F^{-1}(F_1/F_2)$ will have a peak at $(\Delta x, \Delta y)$
Phase Correlation

- In practice, use cross power spectrum

\[
\frac{F_1 F_2^*}{|F_1 F_2^*|}
\]

- Compute inverse FFT, look for peaks

- [Kuglin & Hines 1975]
Phase Correlation

• Advantages
  – Fast computation
  – Low sensitivity to global brightness changes
    (since equally sensitive to all frequencies)
Phase Correlation

- **Disadvantages**
  - Sensitive to white noise
  - No robust version
  - Translation only

- Extensions to rotation, scale
- But *not* local motion
- Not too bad in practice with small local motions
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Correlation plus Optical Flow

• Use e.g. phase correlation to find average translation (may be large)
• Use optical flow to find local motions
Correspondence Approaches

• Optical flow
• Correlation
• Correlation + optical flow
• Any of the above, iterated (e.g. Lucas-Kanade)
• Any of the above, coarse-to-fine
• Feature matching + RANSAC
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Image Pyramids

• Pre-filter images to collect information at different scales
• More efficient computation, allows larger motions
Image Pyramids

Idea: Represent $N \times N$ image as a “pyramid” of $1 \times 1$, $2 \times 2$, $4 \times 4$, $\ldots$, $2^k \times 2^k$ images (assuming $N=2^k$)

- level $k$ (= 1 pixel)
- level $k-1$
- level $k-2$

$\ldots$

level 0 (= original image)
Correspondence Approaches

- Optical flow
- Correlation
- Correlation + optical flow
- Any of the above, iterated (e.g. Lucas-Kanade)
- Any of the above, coarse-to-fine
- Feature matching + RANSAC
Matching features

What do we do about the “bad” matches?
RAndom SAmple Consensus

Select one match, count inliers [Szeliski]
Random Sample Consensus

Select one match, count inliers

[Szeliski]
Least squares fit

Find “average” translation vector
Panoramic Mosaicing

1. Align images
2. Merge overlapping regions
Blending

• Blend over too small a region: seams
• Blend over too large a region: ghosting
Multiresolution Blending

• Different blending regions for different levels in a pyramid [Burt & Adelson]
  – Blend low frequencies over large regions (minimize seams due to brightness variations)
  – Blend high frequencies over small regions (minimize ghosting)
Pyramid Creation

- “Gaussian” Pyramid
- “Laplacian” Pyramid
  - Created from Gaussian pyramid by subtraction
    \[ L_i = G_i - \text{expand}(G_{i+1}) \]
Octaves in the Spatial Domain

Lowpass Images

Bandpass Images
Pyramid Blending

(d) (h) (l)
Minimum-Cost Cuts

- Instead of blending high frequencies along a straight line, blend along line of minimum differences in image intensities
Minimum-Cost Cuts

Moving object, simple blending $\rightarrow$ blur

[Davis 98]
Minimum-Cost Cuts

Minimum-cost cut $\rightarrow$ no blur

[Davis 98]
• Follow gradients of source subject to boundary conditions imposed by dest

\[ \begin{cases} \nabla^2 f = \nabla \cdot v \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases} \]
Poisson Image Blending

- Sources
- Destinations
- Cloning
- Seamless cloning
Poisson Image Blending