Midterm Review

15-463: Computational Photography
Review Topics

• Sampling and Reconstruction
• Frequency Domain and Filtering
• Blending
• Warping
• Data-driven Methods
• Camera
• Homographies
• Modeling Light
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Sampling and Reconstruction
Sampling and Reconstruction

- Sampled representations
  - How to store and compute with continuous functions?
  - Common scheme for representation: samples
    - Write down the function's values at many points

Reconstruction
- Making samples back into a continuous function
  - For output (need realizable method)
  - For analysis or processing (need mathematical method)
  - Amount to "guessing what the function did in between"

Mathematically guess what happens in between
Sampling and Reconstruction

- Effects of Undersampling
  - Lost information
  - High frequency signals get indistinguishable from low frequency ones (aliasing)
Sampling and Reconstruction

- How to avoid aliasing?
  - Sample more often
  - Low pass filter the signal (anti-aliasing)
- Filters work by convolution
Sampling and Reconstruction

- Examples of filters
  - Moving average
  - Weighted moving average
    - Equal weights
    - Gaussian weights
  - Sobel
Sampling and Reconstruction

- Gaussian Filters
  - Smooth out images
  - Convolution of two Gaussians each with standard deviation $\sigma$, gives Gaussian with standard deviation $\sigma/2$
Sampling and Reconstruction

• Matching
  • Use normalized-cross correlation or SSH over patches

• Subsampling
  • Filter with Gaussian then subsample
  • Double filter size with every half-sizing
    • Forms image pyramids
Sampling and Reconstruction
Sampling and Reconstruction
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Frequency Domain

This change of basis has a special name…

Decompose signal into different frequencies

Blurring takes away fast vs. slow changes in the image.

smoothed (5x5 Gaussian)
Frequency Domain and Filtering

A sum of sines

Our building block:

Add enough of them to get any signal \( f(x) \) you want!

How many degrees of freedom?

What does each control?

Which one encodes the coarse vs. fine structure of the signal?

\[
\sum_n A_n \sin(\omega_n x)
\]

Fourier Transform

We want to understand the frequency of our signal. So, let's reparametrize the signal by \( \omega \) instead of \( x \):

\[
F(\omega) = \hat{f}(\omega)
\]

Inverse Fourier Transform

For every \( \omega \) from 0 to \( \infty \), \( F(\omega) \) holds the amplitude \( A \) and phase \( \phi \) of the corresponding sine.

How can \( F \) hold both? Complex number trick!

\[
\begin{align*}
F(\omega) &= A(\cos(\phi) + i\sin(\phi)) \\
&= A e^{i\phi}
\end{align*}
\]

We can always go back:

\[
\begin{align*}
\hat{f}(x) &= \sum_n F_n e^{i\omega_n x} \\
&= \sum_n A_n e^{i\phi_n}
\end{align*}
\]

Sum of sine waves of different frequencies
Frequency Domain and Filtering

In 2D

A nice set of basis

This change of basis has a special name…

Teases away fast vs. slow changes in the image.

What does blurring take away?

smoothed (5x5 Gaussian)
Frequency Domain and Filtering
Frequency Domain and Filtering

- Blurring takes away details.
- Original vs. smoothed (5x5 Gaussian).
- High-pass filter applies Laplacian of Gaussian.

Why Laplacian?
- Delta function.
- Laplacian of Gaussian approximates the high-pass filter.

Gaussian

delta function

Laplacian of Gaussian
Frequency Domain and Filtering

Block-based Discrete Cosine Transform (DCT)
Frequency Domain and Filtering

The gradient of an image:

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

The gradient points in the direction of most rapid change in intensity.
Frequency Domain and Filtering

\[ f \]
\[ h \]
\[ h * f \]
\[ \frac{\partial}{\partial x} (h * f) \]

\[ f \rightarrow \frac{\partial}{\partial x} h \rightarrow (\frac{\partial}{\partial x} h) * f \]
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Blending

\[ I_{\text{blend}} = \alpha I_{\text{left}} + (1-\alpha)I_{\text{right}} \]

- Window size = size of largest feature (to avoid strong seams)
- Window size <= 2 * size of smallest feature (to avoid ghosting)
Blending

Pyramid Blending

Lowpass Images

Bandpass Images
Blending

Gradient Domain

- Result image: \( f \)
  Gradients: \( f_x, f_y \)

- Want \( f \) to ‘look like’ some prespecified \( d \), and \( f_x, f_y \) to ‘look like’ some prespecified \( g^x, g^y \)

\[
\min_{f} w_x (f_x - g^x)^2 + w_y (f_y - g^y)^2 + w_d (f - d)^2
\]

- Weights specify **per-pixel** importance of how much you want \( f \) close to \( d \), \( f_x \) close to \( g^x \), \( f_y \) close to \( g^y \)
Blending

Gradient Domain

\[ f_x = s_x, f_y = s_y \]

\[ f(x, y) - t(x - 1, y) = s_x, f(x, y) - t(x, y - 1) = s_y \]
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Point Processing

Change range of image

\[ g(x) = h(f(x)) \]

Example: \( g(x) = f(x) + 0.3 \)

Histogram Equalization
Warping

Change domain of image

Example: \( g(x) = f(x/2) \)
Warping

- 2D Transformations
  - Translate
  - Rotate
  - Scale
  - Similarity
  - Affine
  - Projective

<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>$[I</td>
<td>t]_{2\times3}$</td>
<td>2</td>
<td>orientation + ⋯</td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$[R</td>
<td>t]_{2\times3}$</td>
<td>3</td>
<td>lengths + ⋯</td>
</tr>
<tr>
<td>similarity</td>
<td>$[sR</td>
<td>t]_{2\times3}$</td>
<td>4</td>
<td>angles + ⋯</td>
</tr>
<tr>
<td>affine</td>
<td>$[A]_{2\times3}$</td>
<td>6</td>
<td>parallelism + ⋯</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>$[\tilde{H}]_{3\times3}$</td>
<td>8</td>
<td>straight lines</td>
<td></td>
</tr>
</tbody>
</table>
Warping

Change of Basis

\[ \mathbf{p}^{uv} = (4,3) \]

\[ \mathbf{p}^{ij} = 4\mathbf{u} + 3\mathbf{v} \]

\[
\begin{bmatrix}
u_x & v_x \\
u_y & v_y
\end{bmatrix}
\begin{bmatrix}4 \\3
\end{bmatrix}
= 
\begin{bmatrix}
u_x & v_x \\
u_y & v_y
\end{bmatrix}
\mathbf{p}^{uv}
\]
Warping

Change of Basis: Inverse Transform

\[ p^{ij} = (5,4) = p_x u + p_y v \]

\[ p^{uv} = (p_x, p_y) = ? \]

\[ p^{uv} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix}^{-1} p^{ij} \]
Warping

- Affine Warp
- Need 3 correspondences

\[ T(x, y) \]

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

- Many ways to find affine matrix
- Warp Source to $[0,0]$, $[1,0]$, $[0,1]$, and then to Destination
- Pose as system of equations in $[a;b;c;d;e;f]$
Warping

- Forward warp

\[ f(x,y) \rightarrow g(x',y') \]

- Inverse warp

\[ g(x',y') \rightarrow f(x,y) \]
Morphing
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Data-Driven Methods

Subspaces Methods (ex: Faces)

Write an image as linear combination of basis images

$$X = \sum_{i=1}^{m} a_i X_i$$
Data-Driven Methods

Subspaces Methods (ex: Faces)

\[
S_{\text{model}} = \sum_{i=1}^{m} a_i S_i \quad T_{\text{model}} = \sum_{i=1}^{m} b_i T_i
\]

\[
s = \alpha_1 \cdot + \alpha_2 \cdot + \alpha_3 \cdot + \alpha_4 \cdot + \ldots = S \cdot a
\]

\[
i = \beta_1 \cdot + \beta_2 \cdot + \beta_3 \cdot + \beta_4 \cdot + \ldots = T \cdot \beta
\]

Shape and Appearance Models
Data-Driven Methods

Subspaces Methods (ex: Faces)

• How to get basis?
• How many basis images to use?
• How to get images that capture important variations?

Use PCA (principal component analysis)

• Keep those principal components whose eigenvalues are above a threshold
Data-Driven Methods

Video Textures

• Compute SSD between frames
• At frame $i$, transit either to
  • frame $i+1$
  • frame $j$ (if $\text{SSD}(j, i+1)$ is small)
• Decide to go from $i$ to $j$ or $i+1$ by tossing a weighted coin.

$$P_{i \rightarrow j} \sim \exp \left( - \frac{C_{i \rightarrow j}}{\sigma^2} \right)$$
Data-Driven Methods

Texture Synthesis

- Search input image for similar neighborhoods
- Use Gaussian weighted SSD for search to emphasize central pixel
- Sample one neighborhood at random
- Grow texture
Data-Driven Methods

Blocked Texture Synthesis

• Search input image for similar neighborhoods around block
• Grow texture by synthesizing blocks
• Find boundary with minimum error (seam carving)
Data-Driven Methods

Lots of Data

• Ex: Scene completion

• Search millions of images on the Internet to find a patch that will complete your image
Data-Driven Methods

Scene Completion (GIST descriptor)

Histogram matching distance

\[ \chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{\kappa} \frac{(h_i(m) - h_j(m))^2}{h_i(m) + h_j(m)} \]
Data-Driven Methods

Filter bank

Input image
Data-Driven Methods

Lots of Data

- Issues with Data
  - Sampling Bias
  - Photographer Bias
- Reduce Bias
  - Use autonomous techniques to capture data
  - StreetView, satellite, webcam etc.
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Camera

Pinhole Model

Center of Projection
Focal Length
Image Plane

Perspective Projection

\[(x, y, z) \rightarrow (-d \frac{x}{z}, -d \frac{y}{z}, -d)\]
Camera

Pinhole Model

Perspective Projection (Matrix Representation)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1/d & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix} =
\begin{bmatrix}
x \\
y \\
-z/d \\
1
\end{bmatrix} \Rightarrow (-d\frac{x}{z}, -d\frac{y}{z})
\]
Camera

Pinhole Model

Orthographic Projection (Matrix Representation)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= \begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \Rightarrow (x, y)
\]
Camera

Pinhole Model

- Pinhole camera aperture
  - Large aperture --- blurry image
  - Small aperture --- not enough light, diffraction effects
- Lenses create sharp images with large aperture
  - Trade-off: only at a certain focus
Camera

Pinhole Model

- Depth of field
- Distance over which objects are in focus
- Field of view
- Angle of visible region

http://www.cambridgeincolour.com/tutorials/depth-of-field.htm
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Homographies

- Panorama
- Reproject images onto a common plane
- Images should have same center of projection

Mosaic: Synthetic wide-angle camera

Projective Warp (Homography)
Homographies

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

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

- Expand equations and rewrite as
  \[Ph = q\]
  \[h = [a \ b \ c \ d \ e \ f \ g \ h]^T\]
- Solve using least-squares (\(h = P\backslash q\))
Other Projection Models

Cylindrical Projection

- Map 3D point \((X, Y, Z)\) onto cylinder
  \[
  (\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Z^2}} (X, Y, Z)
  \]
- Convert to cylindrical coordinates
  \[
  (\sin \theta, h, \cos \theta) = (\hat{x}, \hat{y}, \hat{z})
  \]
- Convert to cylindrical image coordinates
  \[
  (\tilde{x}, \tilde{y}) = (f \theta, fh) + (\tilde{x}_c, \tilde{y}_c)
  \]
Other Projection Models

Spherical Projection

- Map 3D point \((X,Y,Z)\) onto sphere
  \[
  (\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z)
  \]

- Convert to spherical coordinates
  \[
  (\sin \theta \cos \phi, \sin \phi, \cos \theta \cos \phi) = (\hat{x}, \hat{y}, \hat{z})
  \]

- Convert to spherical image coordinates
  \[
  (\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)
  \]
Review Topics

- Sampling and Reconstruction
- Frequency Domain and Filtering
- Blending
- Warping
- Data-driven Methods
- Camera
- Homographies
- Modeling Light
Modeling Light

(The Omnipotent) Plenoptic Function

- Intensity of Light:
  - From all directions: $\theta, \varphi$
  - At all wavelengths: $\lambda$
  - At all times: $t$
  - Seen from any viewpoint: $V_x, V_y, V_z$
- $P(\theta, \varphi, \lambda, t, V_x, V_y, V_z)$
Modeling Light

Lumigraph (Lightfield)

- Intensity along all lines
- For all views (i.e. s,t), gives intensity at all points (i.e. u,v)
- Captures to some extent $P(\theta, \phi, V_x, V_y, V_z)$

For all $(s,t, u,v)$
Modeling Light

Lumigraph (Lightfield)
Modeling Light

Acquiring Lightfield

- Move camera in known steps over (s,t) using gantry
- Move camera anywhere over (s,t) and recover optimal field
- Use microlens array after main lens
Good Luck!!