Image warping/morphing

Digital Visual Effects, Spring 2005

Yung-Yu Chuang

2005/3/9

with slides by Richard Szeliski, Steve Seitz and Alexei Efros
Announcements

• Class time: 1:30-4:20 (with a 20-minute break)
• Last call: send cyy@csie.ntu.edu.tw to subscribe vfx
• Course forum is set up (see course page)
• Scribe volunteers for today and next week
• A schedule for scribes will be posted in forum soon. Please fill in the schedule.
Outline

• Images
• Image warping
• Image morphing
• Project #1
Image fundamentals
Image formation

Illumination (energy) source

Imaging system

Scene element

(Internal) image plan

Output (digitized) image

Quantization
Sampling and quantization
What is an image

- We can think of an image as a function, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$:
  - $f(x, y)$ gives the intensity at position $(x, y)$
  - defined over a rectangle, with a finite range:
    - $f : [a,b] \times [c,d] \rightarrow [0,1]$

• A color image

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$
A digital image

- We usually operate on digital (discrete) images:
  - Sample the 2D space on a regular grid
  - Quantize each sample (round to nearest integer)
- If our samples are D apart, we can write this as:
  \[ f[i, j] = \text{Quantize}\{ f(iD, jD) \} \]
- The image can now be represented as a matrix of integer values

\[
\begin{array}{cccccccc}
  & & & & & & & \\
  & i & & & & & & \\
 62 & 79 & 23 & 119 & 120 & 105 & 4 & 0 \\
10 & 10 & 9 & 62 & 12 & 78 & 34 & 0 \\
10 & 58 & 197 & 46 & 46 & 0 & 0 & 48 \\
176 & 135 & 5 & 188 & 191 & 68 & 0 & 49 \\
 2 & 1 & 1 & 29 & 26 & 37 & 0 & 77 \\
 0 & 89 & 144 & 147 & 187 & 102 & 62 & 208 \\
255 & 252 & 0 & 166 & 123 & 62 & 0 & 31 \\
166 & 63 & 127 & 17 & 1 & 0 & 99 & 30
\end{array}
\]
Aliasing
Image processing

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

\[ h(a) = 1 - a \]

negative
Image enhancement

\[ h(a) = a^r \]

\( r = 3 \)

\( r = 5 \)
Contrast stretching
Histogram

(1) Dark image

(2) Bright image

(3) Low-contrast image

(4) High-contrast image
Accumulated histogram
Histogram equalization
Histogram matching

It is useful for calibrating exposure.
Neighborhood Processing (filtering)

• Q: What happens if I reshuffle all pixels within the image?

• A: It’s histogram won’t change. No point processing will be affected...

• Need spatial information to capture this.
Noise

Original
Salt and pepper noise
Impulse noise
Gaussian noise
Noise reduction

- Mean filter
- Median filter
- Gaussian filter

![Noise reduction diagram](image-url)
Comparison: salt and pepper noise

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Gaussian</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
</tr>
<tr>
<td>5x5</td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
</tr>
<tr>
<td>7x7</td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
</tr>
</tbody>
</table>
Comparison: Gaussian noise

<table>
<thead>
<tr>
<th>3x3</th>
<th>Mean</th>
<th>Gaussian</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="image2.jpg" alt="Image" /></td>
<td><img src="image3.jpg" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5x5</th>
<th>Mean</th>
<th>Gaussian</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.jpg" alt="Image" /></td>
<td><img src="image5.jpg" alt="Image" /></td>
<td><img src="image6.jpg" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7x7</th>
<th>Mean</th>
<th>Gaussian</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7.jpg" alt="Image" /></td>
<td><img src="image8.jpg" alt="Image" /></td>
<td><img src="image9.jpg" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
Image warping
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 filtering: change **range** of image

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

image warping: change **domain** of image

\[ f(x) = g(h(x)) \]
Parametric (global) warping

Examples of parametric warps:

- Translation
- Rotation
- Aspect
- Affine
- Perspective
- Cylindrical
2D coordinate transformations

- translation: \( x' = x + t \) \( x = (x, y) \)
- rotation: \( x' = R x + t \)
- similarity: \( x' = s R x + t \)
- affine: \( x' = A x + t \)
- perspective: \( \underline{x}' \cong H \underline{x} \) \( \underline{x} = (x, y, 1) \) (\( \underline{x} \) is a homogeneous coordinate)

- These all form a nested group (closed under composition w/ inv.)
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>$\begin{bmatrix} I &amp; t \end{bmatrix}_{2\times3}$</td>
<td>2</td>
<td>orientation + ⋯</td>
<td></td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$\begin{bmatrix} R &amp; t \end{bmatrix}_{2\times3}$</td>
<td>3</td>
<td>lengths + ⋯</td>
<td>⬤</td>
</tr>
<tr>
<td>similarity</td>
<td>$\begin{bmatrix} sR &amp; t \end{bmatrix}_{2\times3}$</td>
<td>4</td>
<td>angles + ⋯</td>
<td>⬤</td>
</tr>
<tr>
<td>affine</td>
<td>$\begin{bmatrix} A \end{bmatrix}_{2\times3}$</td>
<td>6</td>
<td>parallelism + ⋯</td>
<td>⬤</td>
</tr>
<tr>
<td>projective</td>
<td>$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times3}$</td>
<td>8</td>
<td>straight lines</td>
<td>⬤</td>
</tr>
</tbody>
</table>
Image warping

• Given a coordinate transform $x' = h(x)$ and a source image $f(x)$, how do we compute a transformed image $g(x') = f(h(x))$?
Forward warping

- Send each pixel $f(x)$ to its corresponding location $x' = h(x)$ in $g(x')$

- What if pixel lands “between” two pixels?
Forward warping

• Send each pixel $f(x)$ to its corresponding location $x' = h(x)$ in $g(x')$

• What if pixel lands “between” two pixels?
• Answer: add “contribution” to several pixels, normalize later (splatting)
Inverse warping

- Get each pixel $g(x')$ from its corresponding location $x = h^{-1}(x')$ in $f(x)$

- What if pixel comes from “between” two pixels?
Inverse warping

• Get each pixel $g(x')$ from its corresponding location $x = h^{-1}(x')$ in $f(x)$

• What if pixel comes from “between” two pixels?
• Answer: resample color value from interpolated (prefiltered) source image
Interpolation

• Possible interpolation filters:
  - nearest neighbor
  - bilinear
  - bicubic
  - sinc / FIR
Bilinear interpolation

• A simple method for resampling images

\[ f(x, y) = (1 - a)(1 - b) f[i, j] + a(1 - b) f[i + 1, j] + ab f[i + 1, j + 1] + (1 - a)b f[i, j + 1] \]
Bicubic interpolation

Non-parametric image warping

- Specify a more detailed warp function
- Splines, meshes, optical flow (per-pixel motion)
Demo

- Warping is a useful operation for mosaics, video matching, view interpolation and so on.
Image morphing
Image morphing

- The goal is to synthesize a fluid transformation from one image to another.
- Cross dissolving is a common transition between cuts, but it is not good for morphing because of the ghosting effects.
Image morphing

- Why ghosting?
- Morphing = warping + cross-dissolving

\[\text{shape} \quad \text{(geometric)} \quad \text{color} \quad \text{(photometric)}\]
Image morphing

image #1  cross-dissolving  image #2

warp  morphing  warp
Morphing sequence
Artifacts of cross-dissolving

http://www.salavon.com/
Face averaging by morphing

average faces
Image morphing

create a morphing sequence: for each time t

1. Create an intermediate warping field (by interpolation)
2. Warp both images towards it
3. Cross-dissolve the colors in the newly warped images
An ideal example

t=0  morphing  t=1
An ideal example

t=0  middle face (t=0.5)  t=1
Warp specification (mesh warping)

- How can we specify the warp?
  1. Specify corresponding *spline control points* *interpolate* to a complete warping function

   easy to implement, but less expressive
Warp specification (field warping)

• How can we specify the warp?
  2. Specify corresponding vectors
    • interpolate to a complete warping function
    • The Beier & Neely Algorithm
Beier & Neely (SIGGRAPH 1992)

- Single line-pair PQ to P’Q’:

\[ u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \]  \hspace{1cm} (1)

\[ v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \]  \hspace{1cm} (2)

\[ X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \]  \hspace{1cm} (3)
Algorithm (single line-pair)

- For each X in the destination image:
  1. Find the corresponding u, v
  2. Find X' in the source image for that u, v
  3. destinationImage(X) = sourceImage(X')

- Examples:

  Affine transformation
Multiple Lines

\[ D_i = X_i' - X_i \]

weight = \( \left( \frac{\text{length}}{a + \text{dist}} \right)^b \)

\( length \) = length of the line segment,
\( dist \) = distance to line segment

The influence of \( a, p, b \). The same as the average of \( X_i' \)
For each pixel \( X \) in the destination
\[ DSUM = (0,0) \]
\[ weightsum = 0 \]
For each line \( P_i, Q_i \)
  calculate \( u, v \) based on \( P_i, Q_i \)
  calculate \( X'_i \) based on \( u, v \) and \( P_i, Q_i' \)
  calculate displacement \( D_i = X'_i - X_i \) for this line
  \( dist = \) shortest distance from \( X \) to \( P_i, Q_i \)
  \( weight = (length^p / (a + \text{dist}))^b \)
  \( DSUM += D_i \times weight \)
  \( weightsum += weight \)
\[ X' = X + DSUM / weightsum \]
\( \text{destinationImage}(X) = \text{sourceImage}(X') \)
Resulting warp
Animated sequences

- Specify keyframes and interpolate the lines for the inbetween frames
- Require a lot of tweaking
Comparison to mesh morphing

- Pros: more expressive
- Cons: speed and control
Warp interpolation

• How do we create an intermediate warp at time t?
• For optical flow:
  – Easy. Interpolate each flow vector
• For feature point methods:
  – linear interpolation of each feature pair
• For Beier-Neely:
  – Can do the same for line end-points
  – But, a line rotating 180 degrees will become 0 length in the middle
  – One solution is to interpolate line mid-point and orientation angle
  – Not very intuitive
Other Issues

- Beware of folding
  - Can happen in any of the methods
  - You are probably trying to do something 3D-ish

- Extrapolation can sometimes produce interesting effects
  - Caricatures
Results

Michael Jackson’s MTV “Black or White”
Warp specification

- How can we specify the warp
  3. Specify corresponding *points*
    - *interpolate* to a complete warping function
Solution#1: convert to mesh warping

1. Define a triangular mesh over the points
   - Same mesh in both images!
   - Now we have triangle-to-triangle correspondences
2. Warp each triangle separately from source to destination
   - How do we warp a triangle?
   - 3 points = affine warp!
   - Just like texture mapping
Solution#2: scattered point interpolation

- RBF
- Work minimization
Transition control
Transition control
Transition control
Multi-source morphing
Multi-source morphing

\[ \overline{W}_i(p) = \sum_{j=1}^{n} b_j \overline{W}_{ij}(p) \]

\[ \overline{I}_i(r) = \overline{W}_i(p) \cdot b_i I_i(p) \]

\[ I(r) = \sum_{i=1}^{n} \overline{I}_i(r) \]
Multi-source morphing
Multi-source morphing
Project #1: image morphing
Project #1 image morphing

- Assigned: 3/9
- Due: 11:59pm 3/29
- Work in pairs
- Handout will be online by tomorrow noon. I will send a mail to vfx when it is available.
- We will provide a generic image library, gil.
Reference software

- **Morphing software review**
- I used [FantaMorph](#) 30-day evaluation version. You can use any one you like.
Morphing is not only for faces
Morphing is not only for faces
Bells and whistles

- Multi-source morphing
- Automatic morphing
- Morphing for animated sequences
Submission

• You have to turn in your complete source, the executable, a html report and an artifact.

• Report page contains:

  description of the project, what do you learn, algorithm, implementation details, results, bells and whistles...

• Artifacts must be made using your own program. artifacts voting on forum

• Submission mechanism will be announced later.