Intelligent design & image warping

- D'Arcy Thompson
  - [http://www-groups.dcs.st-and.ac.uk/~history/Miscellaneous/darcy.html](http://www-groups.dcs.st-and.ac.uk/~history/Miscellaneous/darcy.html)

- Importance of shape and structure in evolution

Skulls of a human, a chimpanzee and a baboon and transformations between them
Important scientific question

• How to turn Dr. Jekyll into Mr. Hyde?
• How to turn a man into a werewolf?

• Powerpoint cross-fading?
Important scientific question

- How to turn Dr. Jekyll into Mr. Hyde?
- How to turn a man into a werewolf?
- Powerpoint cross-fading?
- or
- Image Warping and Morphing?

From An American Werewolf in London
Digression: old metamorphoses

- Unless I’m mistaken, both employ the trick of making already-applied makeup turn visible via changes in the color of the lighting, something that works only in black-and-white cinematography. It’s an interesting alternative to the more familiar Wolf Man time-lapse dissolves. This technique was used to great effect on Fredric March in Rouben Mamoulian’s 1932 film of *Dr. Jekyll and Mr. Hyde*, although Spencer Tracy eschewed extreme makeup for his 1941 portrayal.
• Jekyll & Hide 1932:
  – 35:13
  – ch18 1:06:45
  – ch19 1:17:50

• Jekyll & Hide 1941:
  – ch20 1:25:13
Averaging images

- Cross-fading
  - Pretty much the compositing equation
  \[ C = t \ F + (1-t) \ B \]
Averaging vectors (location)

- $V = t \mathbf{P} + (1-t) \mathbf{Q}$
Warping & Morphing combine both

- For each pixel
  - Transform its location like a vector
  - Then linearly interpolate like an image
Morphing

• Input: two images $I_0$ and $I_N$

• Expected output: image sequence $I_i$, with $i \in 1..N-1$

• User specifies sparse correspondences on the images
  – Pairs of vectors $\{(P^0_j, P^N_j)\}$
Morphing

- For each intermediate frame $I_t$
  - Interpolate feature locations $P_{ti}^t = (1 - t) P_{0i}^0 + t P_{1i}^1$
  - Perform two warps: one for $I_0$, one for $I_1$
    - Deduce a dense warp field from the pairs of features
    - Warp the pixels
  - Linearly interpolate the two warped images
Warping
Warping

- Imagine your image is made of rubber
- warp the rubber

No prairie dogs were armed when creating this image
Careful: warp vs. inverse warp

How do you perform a given warp:

• **Forward warp**
  – Potential gap problems

• **Inverse lookup the most useful**
  – For each output pixel
    • Lookup color at inverse-warped location in input
Image Warping – parametric

- Move control points to specify a spline warp
- Spline produces a smooth vector field
Warp specification - dense

- How can we specify the warp?
  Specify corresponding \textit{spline control points}
  - \textit{interpolate} to a complete warping function

But we want to specify only a few points, not a grid
Warp specification - sparse

• How can we specify the warp?
  Specify corresponding *points*
    • *interpolate* to a complete warping function
    • How do we do it?

How do we go from feature points to pixels?

Slide Alyosha Efros
Warp as interpolation

- We are looking for a warping field
  - A function that given a 2D point, returns a warped 2D point
- We have a sparse number of correspondences
  - These specify values of the warping field
- This is an interpolation problem
  - Given sparse data, find smooth function
Interpolation in 1D

• You know the function at a sparse set of points
• Simplest interpolation?
• Piecewise linear!
Interpolation in 1D

- What is that function we interpolate?
- Warp (absolute or relative displacement)
- In 1D, pairs
Now what about 2D

• What is the equivalent of piecewise linear in 2D?
• What is the equivalent of a segment?
• Triangles and triangulation!
1. Input correspondences at key feature points
2. Define a triangular mesh over the points
   - *Same mesh in both images!*
   - Now we have triangle-to-triangle correspondences
3. Warp each triangle separately from source to destination
Triangulations

A *triangulation* of set of points in the plane is a *partition* of the convex hull to triangles whose vertices are the points, and do not contain other points.

There are an exponential number of triangulations of a point set.
“Quality” Triangulations

Let $\alpha(T) = (\alpha_1, \alpha_2, ..., \alpha_{3t})$ be the vector of angles in the triangulation $T$ in increasing order. A triangulation $T_1$ will be “better” than $T_2$ if $\alpha(T_1) > \alpha(T_2)$ lexicographically.

The Delaunay triangulation is the “best”
- Maximizes smallest angles

Slide Alyosha Efros
Improving a Triangulation

In any convex quadrangle, an edge flip is possible. If this flip improves the triangulation locally, it also improves the global triangulation.

If an edge flip improves the triangulation, the first edge is called illegal.
Naïve Delaunay Algorithm

Start with an arbitrary triangulation. Flip any illegal edge until no more exist.
Questions?
Recap

We have correspondences at a sparse set of feature points
We have compatible triangulations
Now we need to interpolate linearly inside a triangle
Barycentric Definition of a Plane

- \( P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c \)
  with \( \alpha + \beta + \gamma = 1 \)

P is the \textit{barycenter}: the single point upon which the plane would balance if weights of size \( \alpha, \beta, \) & \( \gamma \) are placed on points \( a, b, \) & \( c. \)
Barycentric Definition of a Triangle

- \( P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c \)
  with \( \alpha + \beta + \gamma = 1 \)

- \text{AND} \ 0 < \alpha < 1 \ & \ 0 < \beta < 1 \ & \ 0 < \gamma < 1
How Do We Compute $\alpha$, $\beta$, $\gamma$?

- Ratio of opposite sub-triangle area to total area
  - $\alpha = A_a / A$  
  - $\beta = A_b / A$  
  - $\gamma = A_c / A$

- Use signed areas for points outside the triangle
Intuition Behind Area Formula

- P is barycenter of a and Q
- $A_a$ is the interpolation coefficient on $aQ$
- All points on lines parallel to bc have the same \( \alpha \) (All such triangles have same height/area)
Simplify

- Since $\alpha + \beta + \gamma = 1$, we can write $\alpha = 1 - \beta - \gamma$

\[
P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c
\]

\[
P(\beta, \gamma) = (1-\beta-\gamma)a + \beta b + \gamma c
\]

\[
= a + \beta(b-a) + \gamma(c-a)
\]
Questions?
recall: Applying a warp: USE INVERSE

- **Forward warp:**
  - For each pixel in **input** image
    - Paste color to **warped** location in output
  - Problem: gaps

- **Inverse warp**
  - For each pixel in **output** image
    - Lookup color **from inverse-warped** location
Problems with triangulation morphing

- Not very continuous
  - only $C^0$

- Folding problems

Fig. L. Darsa

Slide Alyosha Efros
Hardcore Photoshop for portrait

figure 9.35

BEFORE

figure 9.36

AFTER
figure 9.37
Selecting the entire left side of the image avoids potential artifacts.

figure 9.38
Dragging a Free Transform handle to narrow the selected area.

figure 9.39
The Liquify filter’s Warp tool pushes pixels forward as you drag.
Step Three:
Get the Push Left tool from the Toolbar (as shown here). It was called the Shift Pixels tool in Photoshop 6 and 7, but Adobe realized that you were getting used to the name, so they changed it, just to keep you off balance.

Step Four:
Choose a relatively small brush size (like the one shown here) using the Brush Size field near the top-right of the Liquify dialog. With it, paint a downward stroke starting just above and outside the love handle and continuing downward. The pixels shifts back in toward the body, removing the love handle as you paint. (Note: If you need to remove love handles on the left side of the body, paint upward rather than downward. Why? That's just the way it works.) When you click OK, the love handle repair is complete.
Morphing
Input images
The feature locations will be our $x_i$

Yes, in this example, the number of features is excessive
Interpolate feature location

- Provides the $y_i$
Warp each image to intermediate location

Two different warps: Same target location, different source location i.e. the $y_i$ are the same (intermediate locations), the $x_i$ are different (source feature locations)

Note: the $x_i$ do not change along the animation, but the $y_i$ are different for each intermediate image

Here we show $t=0.5$ (the $y_i$ are in the middle)
Warp each image to intermediate location
Interpolate colors linearly

Interpolation weight are a function of time:

$$C = (1-t)f^0_t(I_0) + t f^1_t(I_1)$$
Recap

• For each intermediate frame $I_t$
  – Interpolate feature locations $y^t_i = (1 - t) x^0_i + t x^1_i$
  – Perform two warps: one for $I_0$, one for $I_1$
    • Deduce a dense warp field from the pairs of features
    • Warp the pixels
  – Linearly interpolate the two warped images
Movie time

- MJ BW: 23:01
- Willow:
Resampling
The sampling problem

- Parts are magnified
- Parts are minified
- Sometimes anisotropic

- Same problem for 3D texture mapping
Intuition

Plain lookup is bad
• In magnified regions, not smooth enough
• In minified regions, it creates aliasing

What we want
  In magnified regions, smooth interpolation
  In minified regions, take the average

We need good signal processing framework to do this
Fundamentals of Texture Mapping and Image Warping

Master’s Thesis
under the direction of Carlo Séquin

Paul S. Heckbert

Dept. of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720

©1989 Paul S. Heckbert

June 17, 1989
Resampling

source space

discrete input

reconstruct

reconstructed input

resampling with prefiltering [Heckbert 89]

destination space

discrete output

sample

warped input

prefilter

continuous output
Women in Art video

http://youtube.com/watch?v=nUDloN-_Hxs
Bells and whistles
Morphing & matting

• Extract foreground first to avoid artifacts in the background
Uniform morphing

Figure 4. Uniform metamorphosis
Non-uniform morphing

Figure 5. Nonuniform metamorphosis

http://www-cs.ccny.cuny.edu/~wolberg/pub/cgi96.pdf
Video

- Lots of manual work
View morphing
Problem with morphing

• So far, we have performed linear interpolation of feature point positions

• But what happens if we try to morph between two views of the same object?

Figure 2: A Shape-Distorting Morph. Linearly interpolating two perspective views of a clock (far left and far right) causes a geometric bending effect in the in-between images. The dashed line shows the linear path of one feature during the course of the transformation. This example is indicative of the types of distortions that can arise with image morphing techniques.
View morphing

- Seitz & Dyer
- Interpolation consistent with 3D view interpolation

Figure 1: View morphing between two images of an object taken from two different viewpoints produces the illusion of physically moving a virtual camera.
Main trick

- Prewarp with a homography to "pre-align" images
- So that the two views are parallel
  - Because linear interpolation works when views are parallel

Figure 4: View Morphing in Three Steps. (1) Original images $I_0$ and $I_1$ are prewarped to form parallel views $\hat{I}_0$ and $\hat{I}_1$. (2) $\hat{I}_s$ is produced by morphing (interpolating) the prewarped images. (3) $\hat{I}_s$ is postwarped to form $I_s$. 
Figure 6: View Morphing Procedure: A set of features (yellow lines) is selected in original images $I_0$ and $I_1$. Using these features, the images are automatically prewarped to produce $\hat{I}_0$ and $\hat{I}_1$. The prewarped images are morphed to create a sequence of in-between images, the middle of which, $\hat{I}_{0.5}$, is shown at top-center. $\hat{I}_{0.5}$ is interactively postwarped by selecting a quadrilateral region (marked red) and specifying its desired configuration, $Q_{0.5}$, in $\hat{I}_{0.5}$. The postwarps for other in-between images are determined by interpolating the quadrilaterals (bottom).
Figure 10: Image Morphing Versus View Morphing. Top: image morph between two views of a helicopter toy causes the in-between images to contract and bend. Bottom: view morph between the same two views results in a physically consistent morph. In this example the image morph also results in an extraneous hole between the blade and the stick. Holes can appear in view morphs as well.
Figure 9: Mona Lisa View Morph. Morphed view (center) is halfway between original image (left) and its reflection (right).
Figure 7: Facial View Morphs. Top: morph between two views of the same person. Bottom: morph between views of two different people. In each case, view morphing captures the change in facial pose between original images $I_0$ and $I_1$, conveying a natural 3D rotation.
Extensions
The actual structure of a face is captured in the shape vector $S = (x_1, y_1, x_2, \ldots, y_n)^T$, containing the $(x, y)$ coordinates of the $n$ vertices of a face, and the appearance (texture) vector $T = (R_1, G_1, B_1, R_2, \ldots, G_n, B_n)^T$, containing the color values of the mean-warped face image.
The Morphable face model

- Again, assuming that we have \( m \) such vector pairs in full correspondence, we can form new shapes \( S_{\text{model}} \) and new appearances \( T_{\text{model}} \) as:

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

\[
s = \alpha_1 \cdot \text{face} + \alpha_2 \cdot \text{face} + \alpha_3 \cdot \text{face} + \alpha_4 \cdot \text{face} + \ldots = S \cdot \alpha
\]

\[
t = \beta_1 \cdot \text{face} + \beta_2 \cdot \text{face} + \beta_3 \cdot \text{face} + \beta_4 \cdot \text{face} + \ldots = T \cdot \beta
\]

- If number of basis faces \( m \) is large enough to span the face subspace then:
Deviations from the mean

\[ \Delta X = X - X \]

\[ X = X + 1.7 \]

\[ \Delta X = X - X \]
Subpopulation means

- Examples:
  - Happy faces
  - Young faces
  - Asian faces
  - Etc.
  - Sunny days
  - Rainy days
  - Etc.
  - Etc.

Average female

Average male
The average face

- [http://www.uni-regensburg.de/Fakultaeten/phil_Fak_II/Psychologie/Psy_II/beautycheck/english/index.htm](http://www.uni-regensburg.de/Fakultaeten/phil_Fak_II/Psychologie/Psy_II/beautycheck/english/index.htm)

On the left: the “real” Miss Germany 2002 (= Miss Berlin) and on the right: the “virtual” Miss Germany, which was computed by blending together all contestants of the final round and was rated as being much more attractive.
Using 3D Geometry: Blanz & Vetter, 1999

show SIGGRAPH video
Automatic morphing

Recap & Significance
Recap

• Idea that linear interpolation introduces blur
• Separation of shape and color
• Idea of non-rigid alignment of different images
  – Applications to medical data
• Applications, related to
  – Special effects
  – Face recognition
  – Video frame interpolation
  – MPEG
• Scattered data interpolation
References
Refs

- http://www.visgraf.impa.br/cgi-bin/morphQuery.cgi?output=html
- http://www.cg.tuwien.ac.at/research/ca/mrm/
- http://w3.impa.br/~morph/sites.html
- http://w3.impa.br/~morph/sig-course/slides.html
- http://www.owlnet.rice.edu/~elec539/Projects97/morphjrks/morph.html
- http://www.fmrib.ox.ac.uk/~yongyue/thinplate.html
- http://www.uoguelph.ca/~mwirth/PHD_Chapter4.pdf
Software

- http://www.morpheussoftware.net/
- http://www.debugmode.com/wimorph/
- http://www.stoik.com/products/morphman/mm1_main.htm
- http://www.creativecow.net/articles/zwar_chris/morph/index.html
- http://meesoft.logicnet.dk/SmartMorph/
- http://www.asahi-net.or.jp/~FX6M-FJMY/mop00e.html
- http://morphing-software-review.toptenreviews.com/
Next time: Panoramas