CPSC 4040/6040
Computer Graphics
Images

Joshua Levine
levinej@clemson.edu
Lecture 20
Removing Warp Artifacts
Nov. 5, 2015

Slide Credits:
Szymon Rusinkiewicz
Agenda
Refresher from Lec19
Projective Warps

• What is the matrix?

• What the knowns? Unknowns? How many?
Algebra

• A general 3x3 matrix can express this entire class of warps (9 unknowns)

  • \( a_{33} \) acts as a global scale parameter, so we can always set it to 1 without losing generality

• The remaining 8 unknowns can be solved by 4 pairs of equations using the 8 known \( x_i, y_i \) values and the 8 known \( u_i, v_i \) values

• Solving these 8 equations gives the 8 remaining \( a_{ij} \) unknowns

\[
x_i = \frac{a_{11}u_i + a_{12}v_i + a_{13}}{a_{31}u_i + a_{32}v_i + 1}
\]

\[
y_i = \frac{a_{21}u_i + a_{22}v_i + a_{23}}{a_{31}u_i + a_{32}v_i + 1}
\]
Bilinear Warping

• Key Idea: Instead of using bilinear interpolation for pixel color values, we can use it to interpolate the positions in the warped image.
Step 2: Forward Warp with \( u,v \) offsets
Inverse Bilinear Warp Can Be Computed from the Forward Warp

• Forward warp for a pixel \((s, t)\) is equivalent to the following equations:

\[
\bar{x} = (x_{12} - x_{03})u + x_{03} \\
x_{03} = (x_3 - x_0)u + x_0 \\
x_{12} = (x_2 - x_1)u + x_1
\]

• where \((s_0, t_0)\) is the lower left of the image, \((s_1, t_1)\) is the upper right of the image, and (effectively normalizing)

\[
u = \frac{s - s_0}{s_1 - s_0} \quad v = \frac{t - t_0}{t_1 - t_0}
\]
Inverse Bilinear Warp Can Be Computed from the Forward Warp

• Taking the inverse of

\[
x = a_0 + a_1 u + a_2 v + a_3 uv
\]

\[
y = b_0 + b_1 u + b_2 v + b_3 uv,
\]

• Leads to

\[
v = \frac{-c_1}{2c_2} \pm \frac{1}{2c_2} \sqrt{c_1^2 - 4c_2c_0}
\]

\[
u = \frac{x - a_0 - a_2 v}{a_1 + a_3 v},
\]

• Where 0<u<1, 0<v<1, and

\[
c_0 = a_1(b_0 - y) + b_1(x - a_0),
\]

\[
c_1 = a_3(b_0 - y) + b_3(x - a_0) + a_1b_2 - a_2b_1,
\]

\[
c_2 = a_3b_2 - a_2b_3.
\]
Recall: Converting Between Image Domains

- When an image is acquired, an image is taken from some continuous domain to a discrete domain.

- **Reconstruction** converts digital back to continuous through interpolation.

- The reconstructed image can then be **resampled** and **quantized** back to the discrete domain.

![Diagram showing the process of converting between image domains](image.png)

*Figure 7.7. Resampling.*
Lesson 1: To reduce magnification artifacts we need to do a better job of reconstruction.

Fixing Jaggies / Magnification Artifacts
Reconstruction Artifacts

- Leads to staircasing or “jaggies”

b) reconstruction artifacts
Do a Better Reconstruction?

• Basic Idea: If we interpolate the data samples better we will have a superior reconstruction

• How? Bilinear Interpolation, Bicubic, etc.
Recall: Nearest Neighbor
Recall Bilinear Example
Recall Bicubic (from Photoshop)

Ignore small color issues
Lesson 2: To reduce minification artifacts we must either 1) sample more finely than once for each output pixel, or 2) smooth the reconstructed input before sampling.
• Aliasing leads to missing and/or unwanted features

• Example: 12x12 images scaled to a 4x4 image.

a) aliasing artifacts
Aliasing

- When we minify, we use only a few samples to represent lots of data
- High frequencies “masquerade” as low ones
- Images look “ropey”. This is not jaggies!

(Barely) adequate sampling

Inadequate sampling
Aliasing Described by Sampling Theory

• What happens if we use too few samples?

• Aliasing: when high frequencies masquerade as low ones
How many samples are enough to avoid aliasing?

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?
How many samples are enough to avoid aliasing?

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?
How many samples are enough to avoid aliasing?

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?
How many samples are enough to avoid aliasing?

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?
How many samples are enough to avoid aliasing?

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?
Antialiasing Filters

- Basic Idea: Smooth the image first to reduce the overall frequency
- How? Use filters!
Resampling with Filters

Output is weighted average of inputs:

```c
float Resample(src, u, v, k, w)
{
    float dst = 0;
    float ksum = 0;
    int ulo = u - w; etc.
    for (int iu = ulo; iu < uhi; iu++) {
        for (int iv = vlo; iv < vhi; iv++) {
            dst += k(u,v,iu,iv,w) * src(u,v)
            ksum += k(u,v,iu,iv,w);
        }
    }
    return dst / ksum;
}
```
Local Convolution?

Compute weighted sum of pixel neighborhood

- Output is weighted average of input, where weights are normalized values of filter kernel \((k)\)

\[
dst(ix,iy) = 0;
for (ix = u-w; ix <= u+w; ix++)
    for (iy = v-w; iy <= v+w; iy++)
        d = dist (ix,iy)\rightarrow(u,v)
        dst(ix,iy) += k(ix,iy)*src(ix,iy);
\]

\(k(ix,iy)\) represented by gray value
Smoothing an Image in This Way Limits the Frequency Bands

Point Sampled: Aliasing!  Correctly Bandlimited
Another Approach

- Instead of smoothing, we can also sample better and then aggregate the samples.

Figure 11.12: Sampling Techniques Under Inverse Mapping
Circled Samples are Different from Average

Figure 11.14: Adaptive Supersampling
Artifact “Free” Warping Pipeline
Figure 11.3: A Conceptual View of Digital Image Warping
Warping Pipeline

- Ideal resampling requires correct filtering to avoid artifacts
- **Reconstruction** filter especially important when magnifying
- **Bandlimiting** filter especially important when minifying
Lec21 Required Reading
• House, Ch. 13
Feature-Based Image Metamorphosis

Thaddens Beier
Silicon Graphics Computer Systems
2011 Shoreline Blvd, Mountain View CA 94043

Shawn Neely
Pacific Data Images
1111 Karlstad Drive, Sunnyvale CA 94089

1 Abstract
A new technique is presented for the metamorphosis of one digital image into another. The approach gives the animator high-level control of the visual effect by providing natural feature-based specification and interaction. When used effectively, this technique can give the illusion that the photographed or computer generated subjects are transforming in a fluid, surrealistic, and often dramatic way. Comparisons with existing methods are drawn, and the advantages and disadvantages of each are examined. The new method is then extended to accommodate keyframed transformations between image sequences for motion image work. Several examples are illustrated with resulting images.

Keywords: Computer Animation, Interpolation, Image Processing, Shape Transformation.

2.2 3D Computer Graphics Techniques
We can use technology in other ways to help build a metamorphosis tool. For example, we can use computer graphics to model and render images which transform over time.

One approach involves the representation of a pair of three-dimensional objects as a collection of polygons. The vertices of the first object are then displaced over time to coincide in position with corresponding vertices of the second object, with color and other attributes similarly interpolated. The chief problem with this technique is the difficulty in establishing a desirable vertex correspondence; this often imposes inconvenient constraints on the geometric representation of the objects, such as requiring the same number of