Bridging Between Image-Based and Geometry-Based Graphics

Feathering / Compositing
Generalized Resampling
Rasterization

Compositing - basic version
This over that w/ as opacity
New Color = \( \alpha F + (1 - \alpha) B \)
\( \begin{array}{c}
\text{foreground} \\
\text{background}
\end{array} \)

Done per-pixel

Other per-pixel operations: \( c = f(c) \)
brighten, color correct, various enhancements, ...

Feathering
What happens if you change part of an image

Hard edge between new and old \( \Rightarrow \) hard edges attract the eye

Hack - make the transition gradual

\( \alpha = 0 \)
\( \alpha = 1 \)

Close tool, retouch brush, ... \( \Rightarrow \) lots of Photoshop tools
Generalized Resampling
Transform image
\[ f(x, y) \Rightarrow x', y' \]
change positions, not colors (they get carried along)

This is re-sampling

Scaling is a special case

WARPING
Morphing - warp to align, blend to combine

2 parts of warping
1. how to define \( f : IR \rightarrow IR \)
2. how to re-sample
Why is generalized Re-Sampling hard:
- Pixels map to odd shapes
- Pixels map to different sizes
- Sizes vary - can have stretch and squish in some images

**Algorithm 1 - Forward**
- For each pixel $P(x, y)$, $x', y' = f(x, y)$
- $\text{dst}(x', y') = \text{src}(x, y)$
- What if $x', y'$ not at an integer?
- What if scale up?
- What if scale down?

**Better: Splat**
- Splats are the resampling kernel
- Need to accumulate
- Had to get right...

**Algorithm 2 - Reverse**
- For each pixel $p \in \text{dst}(x', y')$
- $x, y = f^{-1}(x', y') \iff \text{need inverse}$
- $\text{dst}(x', y') = \text{sample}(\text{src}(x, y))$

**Good:** no gaps
- Point sampling easy
- Better sampling "averages" over region