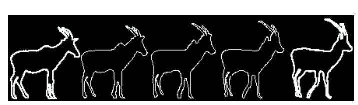
Metamorphosis

Metamorphosis

Metamorphosis is the gradual evolution of a source object, through intermediate objects to a target object

Object = image, polygon, curve
volume, polyhedron, surface

You must do things right!



Cross dissolves

Problem - misaligned regions

Issues in morphing

- Correspondence:
  - Feature specification
  - Warp generation
- Transition control

Some image metamorphosis methods

- Mesh Warping (Wolberg)
- Field Morphing (Beier & Neely)
- Radial Basis Functions (Arad et al, Edge&Maddock)
- Energy minimization (Lee et al)
- Compatible triangulation (Aronov et al, Tal&Elber, Surazhsky et al)

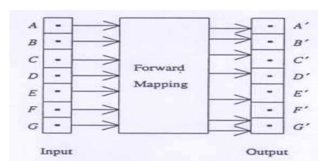
Distortion of a single image

There are two ways to warp an image:

- Forward mapping
- Reverse mapping

Forward mapping

Scan through the source (input) image pixel by pixel and copy them onto the destination (output) image at positions determined by the X and Y mapping functions

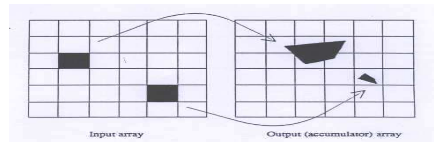


Properties

- Input pixels are mapped from the set of integers to the set of real numbers
- This corresponds to the regularly spaced input samples and the irregular output distribution
- It can cause two types of problems: *Gaps* and *Overlaps*

Avoiding gaps

Use *Four-corner-mapping* = Consider input pixels as square patches that may be transformed into arbitrary quadrilaterals in the output image

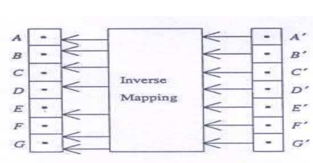


Properties

- Input remains contiguous after the mapping
- Input pixels often straddle several output pixels or lie embedded in one
- An *Accumulator Array* is required to properly integrate the input contributions at each output pixel
- Problem - Costly intersection tests are needed to derive the weights

Reverse mapping

- Go through the destination image, pixel by pixel, and sample the correct pixel from the source image
- Guaranteed that all output pixels are computed



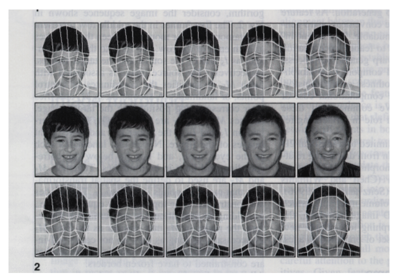
Properties

- More convenient approach than forward mapping
- No *Accumulator Array* is necessary
- Output pixels that lie outside a clipping window need not be evaluated

Mesh-based morphing (Wolberg)

Ms

Is



It

Mt

Correspondence in mesh warping

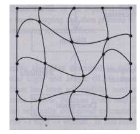
A combination of warping two images so they have the same “shape” and then cross-dissolving the resulting images

- Image warping: Specify a warp that distorts the first image into the second
- Cross-dissolve between image elements:
  The color of each pixel is interpolated over time from the first image to the corresponding second image value

Mesh warping

For each frame f

- Linearly interpolate mesh M between Ms and Mt
- Warp Is to I_1 using meshes Ms and M
- Warp It to I_2 using meshes Mt and M
- Linearly interpolate image I_f between I_1 and I_2



Warping algorithm

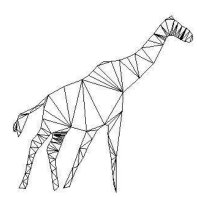
Given a source image and two 2D arrays of coordinates S and D, fit splines to produce a continuous mapping

Apply a two-pass algorithm:

- Map (u,v) to (x,v)
- Map (x,v) to (x,y)

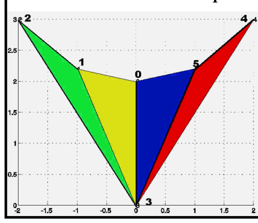
Triangulation-based morphing

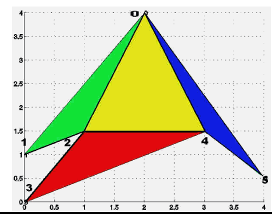
Polygon triangulation = A decomposition of a polygon into triangles by a maximal set of non-intersecting diagonals



The problem (Aronov et al)

- P1 and P2 are two simple polygons
- Generally cannot be triangulated compatibly without extra points

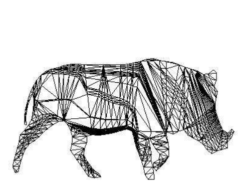


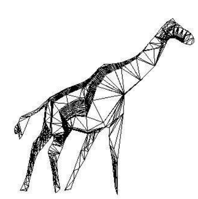


But

If we are allowed to add (Steiner) points, it can be done

Compatible triangulation





Compatible triangulation

Given two polygons P1 and P2 each with n vertices, their compatible triangulation is a joint labeling of their vertices and some of their internal points, such that a triangulation of one polygon admits a triangulation in the other polygon and it is labeled compatibly

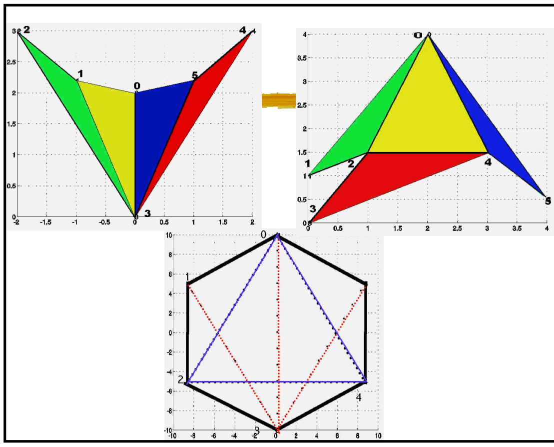
Use in morphing (Tal&Elber)

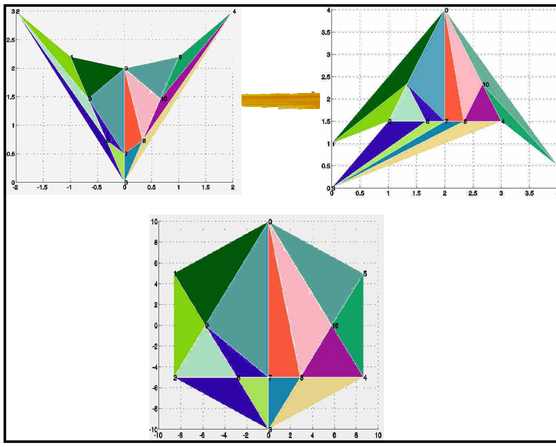
1. Outline extraction
2. Establishment of correspondence between the boundaries
3. Compatible triangulation
4. Texture mapping

Compatible triangulation algorithm

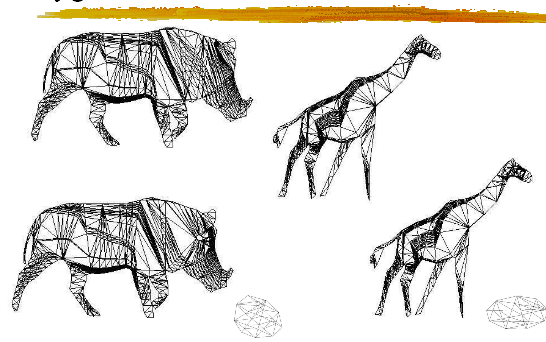
<u>Intuition:</u> "convexize" the polygons

1. Find a triangulation T_1 (T_2) for P_1 (P_2)
2. Map T_1 into T_1' (T_2') of a convex polygon P
3. Overlaying T_1' and T_2' on P yields a convex subdivision, that can be triangulated into T
4. Map T back into T_1 and T_2 to obtain a compatible triangulation of P_1 and P_2





Polygons with holes



Results

Results

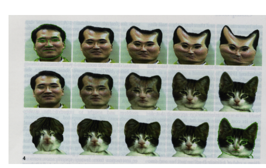
Results

Composing objects like clip-arts

Transition control

- Determine the rate of warping and color blending across the morph sequence
- If transition rates differ from part to part, more interesting animations are possible

Example



Uniform transition

Non-uniform transition