Specification and Computation of Warping and Morphing Transformations

Bruno Costa da Silva
Microsoft Corp.
Morphing Transformations

Representation of Transformations

Specification of Transformations

Specification Techniques
Morphing Transformation

- **Shape & Attribute Transformation**
  - Shape transformation ➔ Alignment
  - Shape Combination ➔ Blending
  - Attribute transformation ➔ Combination

- **Input** = two different graphical objects
- **Output** = a new graphical object
Morphing

\[ g = A(W(f_1), W(f_2)) \]
Example: Shape Transformation

- Alignment through Warping
Example: Attribute Transformation

Cross-dissolve

\[ \text{Image 1} + \text{Image 2} = \text{Result Image} \]
Example: Morphing
Transformation Dimension

- Intrinsic Dimension of Objects:
  - Letters are 2D
  - Buildings are 3D

- Transformations can occur in:
  - Intrinsic dimension
  - Visualization dimension
2D x 3D Transformations

Project

2D Metamorphosis

3D Metamorphosis

Project

Project
2D Warp x 3D Warp
Reduced Dimension Morphing

- 2.5D morphing
  - 3D warp + 2D cross-dissolve

- View Morphing
  - Infer projection parameters
  - New views from other viewpoints
View Morphing
Transformations ➔ Animations

- **Input graphical object:**
  - Static
  - Dynamic

- **Transformation:**
  - Fixed
  - Variable

- **Output graphical object:** Dynamic
Producing Animations

- **Input**: Static → Fixed → Static
- **Transformation**: Static → Variable → Dynamic
- **Output**: Dynamic → Fixed → Dynamic
- **Input**: Dynamic → Variable → Dynamic
Morphing Transformations

Representation of Transformations

Specification of Transformations

Specification Techniques
Abstraction Levels

Physical Universe → Mathematical Universe → Representation Universe → Implementation Universe
The Transformation Process

1. User
2. Specification
3. Representation
4. Computation
5. Results
The Transformation Process

Characteristics

- User Input:
  - Short and simple

- Representation:
  - Can be complex, but
  - Simple to obtain from specification and
  - Simple to use for computation
Selecting a Representation

- Goals
  - Computer storage and manipulation
  - Finiteness and continuity
  - Control and user friendliness
  - Possibility of efficient computation
  - Simplicity
Representation Example

- Projective Warp
Representation Example

Some Possible Decisions

User Inputs:
- Perspective parameters
- Four corners and their deformation

Representations:
- Four corners and their deformation
- Transformation matrix
Morphing Transformations
Representation of Transformations
Specification of Transformations
Specification Techniques
Specifying Transformations

- Globally
  - Few parameters

- Locally
  - Large number of parameters
  - Complex specification/computation
  - Warping and Morphing
Examples

- Global
- Local
Specification Principles

What happens to each point of the domain?

Can be described:

- for every individual point (not feasible)
- for a few regions and interpolated
- indirectly through parameters
Examples

- Interpolated

- Parameters
  (80% of pinch)
Specification

Definitions

Correspondence

- association between two regions
- defines original and final states

Specification

- set of correspondences
Specification Example

Three Correspondences

Original Set

Final Set
Warping x Morphing

- **Warping**
  - Single object
  - Specification of original and deformed states

- **Morphing**
  - Two objects
  - Specification of initial and final states
Warping x Morphing

- Warping

- Morphing
Morphing Transformations

Representation of Transformations

Specification of Transformations

Specification Techniques
Types of Specification

- Parametric
- Partition-based
- Feature-based
- Automatic or semi-automatic
Parametric Specification

- Controlled by few parameters
- Examples: rotation, twist, visualization
Specification by Partition

- Regions cover the entire object
- Regular x Irregular partitions
- Same topology: original and final sets
Partition Examples
Specification by Features

- Regions do *not* cover the object
- Dimension of features $\leq$ object dimension
- Examples: points, vectors, planes, boxes,...
Feature Examples
Automatic Specification

- No region correspondences
- Reduced specification (if any)

Approaches:
- Automatic computation
- Automatic detection of object features
- Digitization (rotoescoping)
Automatic Example
Resulting Transformation

Behavior in:

- Specified Areas
  - “close” to specification

- Unspecified areas
  - should maintain continuity
  - “close” to expected behavior
Specification Goals

- *Minimal* input

- Predictable, common sense behavior

- Leads to Dualities:
  - Specification x Computation
  - Specified x Inferred
  - Amount of input x Predictability
Next

- Comparison of specification techniques
- Videos
- Specification and computation duality
- Computation of transformations
Specification and Computation of Warping and Morphing Transformations (part II)

Bruno Costa da Silva
Microsoft Corp.
Specification + Computation

Computation of Transformations

Conclusion
Specification + Computation

- Specification techniques influence computation:
  - Split responsibilities
  - Specification-wise analysis
  - Computation considered
Case Analysis

Techniques and their specification

- Point-based
- Vector-based
- Spline mesh
- Feature-based spline mesh
- Drawing-based
Point-based

+ Simple interpolations
  Simple user interface
  Higher dimensions
  Different types of graphical objects

- May need many points
  Consistency problems
  Predictability
  Computation
Point-based Example
Vector Based

+ Few vectors needed
  Simple user interface
  Higher dimensions
  Different types of graphical objects

− Computation
  Consistency problems
  Predictability
Vector Based Example
Spline Mesh

+ Efficient multiple pass implementation
Less consistency problems
Predictability
Higher dimensions

- Difficult specification
User interface
Restricted types of graphical objects
Multiple pass anomalies
Spline Mesh Example
Feature-based Spline Mesh

- Feature specification
- Spline Computation
- User works with features
- Features deform spline meshes
- Spline meshes warp the objects
User

Computation

Warp Grid

Warp Object
Feature-based Spline Mesh

+ 
  - Feature specification
  Efficient spline mesh computation
  Higher dimensions

- 
  Loose control
  Interpolation
  Consistency
  Restricted types of graphical objects
  Multi-pass anomalies
Feature-based Spline Mesh Example
Drawing Based

+ Flexible specification
  Rotoscoped features
  Different types of graphical object

- Computation
  Consistency problems
  Higher dimensions
Drawing-based Example
Specification + Computation

Computation of Transformations

Conclusion
Computation of Transformations

- Object representation
  - Continuous x discrete
- Forward mapping
- Inverse mapping
- Multi-pass transformations
Forward Mapping

- Traverse input pixels
- Miss/overlap output pixels
Inverse Mapping

- Traverse output pixels
- Does not waste work

\[ W^{-1} \]

Input \[ p_i \] \rightarrow Output \[ p_o \]
Multi-pass Transformations

- **Separable transformation**

\[ T = f_n \circ f_{n-1} \circ \ldots \circ f_2 \circ f_1 \]

- **Computation of** \( f_i \) **simpler than** \( T \)
  - Efficient multi-pass computation
Multi-pass Transformations
Specification + Computation

Computation of Transformations

Conclusion
Specification Trends

■ Minimization of user input favors:
  ◆ Feature specification
  ◆ Automatic feature extraction
  ◆ Automatic specification

■ Efficiency favors:
  ◆ Partition based techniques (regularity)
  ◆ Simpler forms of interpolation
Conclusion

- Transformation Representation
- Importance of User and Specification
- Specification and Computation Duality
- Specification Techniques Comparison
- Simple Specification & Predictable Results