Shape coding

Touradj Ebrahimi
Touradj.Ebrahimi@epfl.ch
Any data compression (coding) scheme consists of two components:
- Message extraction
- Message coding
Visual data representation and coding

- Visual information can be coded by two distinct approaches depending on the definition and nature of extracted message:
  - First generation coding techniques
  - Second generation coding techniques

For pedagogical reasons, we see examples of a still picture to explain these approaches. However, the approach is valid for any type of visual data.
First generation coding

In 1st generation coding, a picture is represented as a set of small points (called pixels) each bearing a particular shade (gray level) or color. Each pixel represents the message to code.
First generation coding

- Message extraction: Assignment of numbers to every gray level
- Message coding: Coding of one or several numbers or a transformed version of them (PCM, DPCM, SBC, …)

messages to be coded

134  135  132  12  15...
133  134  133  133  11...
130  133  132  16  12...
137  135  13  14  13...
140  135  134  14  12...
First generation coding

• **Advantages**
  – Capture
  – Display/Print
  – Simple (matrix representation)

• **Drawbacks**
  – Differs from human visual system mechanism
  – Not appropriate for interactivity with picture elements
  – Limited potential
• In 2nd generation coding, a picture is represented by the set of its constituting objects (or features). Each object represents the (complex) message to be coded
Second generation coding

• Message extraction: Identification of objects
• Message coding: Coding of objects
Second generation coding

- **Advantages**
  - Similar to human visual system mechanism
  - Easier to interact with objects in the picture
  - Big potential for use in various applications

- **Drawbacks**
  - Problem of definition of objects (segmentation, tracking, …)
  - Needs interface to conventional display (composition)
  - Complexity
• In 1st generation coding:
  – The smallest entity in a picture is a pixel with its associated texture (color), and motion
  – Message to be coded for every pixel:
    texture (color) + motion
Features in 2nd generation coding

- In 2nd generation coding:
  - The smallest entity in a picture is an object with its associated shape, texture (color), and motion.
  - Message to be coded for every pixel: shape + texture (color) + motion.
What is shape and what does it bring?

- Shape is like a force field that keeps atoms (pixels) of an object together
  - This would allow to interact with the entire object as one single entity

- Shape coding approaches:
  - Bitmap
  - Intrinsic
  - Contour
Shape representation

• **Bitmap**

• **Advantages**
  – Simple
  – Can be easily incorporated in 1st generation coding approaches
  – High compression efficiency

• **Drawback**
  – No direct semantic information
• A bounding box is generated around the object. All pixels in the bounding box belonging to the object (inside the object) are coded as 1 (or 0) and all pixels of the bounding box outside of the object are coded as 0 (or 1). This is equivalent to a bi-level image (such as fax).
• Any bi-level image coding scheme can be used to code a shape information represented by this approach
  – CCITT G3, G4 (run-length coding)
  – MMR (run-length)
  – ISO JBIG (arithmetic coding)
  – Quad-tree coding
  – Chroma-keying
  – MPEG-4
Quad-tree coding of bitmaps
MMR based coding of bitmaps

• Modified Modified Reed

![Diagram showing current shape block, video object plane, and changing pixel.]

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Basic Principle: Bitmap arithmetic coding

- Black/white pixels
- Local context prediction
  - 3-line 10-pel template (JBIG):

- Non-adaptive arithmetic coding
Shape representation

• Intrinsic shape coding
  – mathematical morphology: skeleton

• Advantages
  – Semantic representation
  – Scalable
  – Simplifies Object manipulation

• Drawbacks
  – Additional complexity
  – Less efficient compression
Shape representation

• **Contour**
  – chain coding
  – Fourier descriptors
  – geometrical approximation

• **Advantages**
  – Semantic representation
  – Scalable
  – Simplifies image manipulation

• **Drawbacks**
  – Less efficient compression
  – Requires filling algorithms
Lossy vs Lossless Coding

• Bitmap
  – entropy coding (lossless)
  – filtering, downsampling (lossy)

• Contour
  – chain coding (lossless)
  – geometrical approximation (lossy)
Basic Principle: Contour Coding

Contour extraction

lossless

Chain coding

Differential
Huffman
Arithmetic...

lossy

Geometrical approximation

(x1,y1)(x2,y2)...

Entropy coding

0100111...

0100111...
Contour location and connectivity

inter-pixel

interior
8-connected

interior

exterior
4-connected

interior

north-west
6-connected
Chain coding

- Start from an arbitrary contour point
- Code its position
- Code the relative position of the next neighbour (left, right, top, down, ...).
- Continue until all contour points are coded

\[(X_s, Y_s) + \text{right} + \text{right} + \text{up} + \text{right} + \ldots\]
Geometrical approximation

- controlled lossy coding:
  - distortion constraint (max. distance)
  - rate constraint (number of vertices)
- approximation by:
  - polygons (1st order- 2 parameters)
  - cubic splines or blending curves (3rd order- 4 parameters)
• **Spatial prediction:**
  – differential coding
  – efficient entropy coding

• **Temporal prediction:**
  – direct (no motion)
  – motion vectors from previous texture frame
  – motion estimation on shape images (vectors transmitted)
Polygonal Approximation

- Contour
- Main axis
- Error control: max. distance
- Recursive edge split
Classical Polygonal Encoding

lossless

error 1 pixel

error 2 pixels

error 3 pixels

error 4 pixels

error 5 pixels

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Progressive Polygonal Encoding

- Error 1 pixel: 481 bits
- Error 2 pixels: 388 bits
- Error 3 pixels: 310 bits
- Error 4 pixels: 234 bits
- Error 5 pixels

Lossless: 669 bits

Total bits: 1501 bits
Progressive Polygon Encoding

- Progressive approximation
- Refinements close to coarser polygon

Stripe Rule
Static Geometrical Rules (4)

- Stripe Rule (1)
- Bounding Box Rule (2)
- Single Child Rule (3)
- Vertex Unicity Rule for Parents (4)
Dynamic Geometrical Rules (8)

- Vertex Unicity Rule for Children (5)
- Conquest Rule (6)
- Utility Rule (7)
- Visibility Rule (8)

Legend:
- invalid position
- locally invalid position
Dynamic Geometrical Rules (12)

Dead End Rule (9)

Connection Rule (10)

Utility Rule for last vertex (11)

Connection Rule for last vertex (12)
Scalable bitmap coding

Spatial

*4

*2

Quality
Scalable polygon approximation

Vector representation:

Vertex (x, y) -> Vertex(αx, αy)

α ∈ R⁺
Scalable Intra Coding: Results

Rate/Distortion

- PPE, Kids
- CAE, Kids
- PPE, Weather
- CAE, Weather

Mismatch pixels (%)

Average bits/frame

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

0 500 1000 1500 2000 2500 3000 3500
Visual shape distortion

original

polygonal approx.
max. dist. = 2.0

Hermite polynomials
+ control blocks

morphological
filter

Quad-tree
What Is Shape good For?

- Very low bit rate, low delay
  - transmit only region of interest
- Editing
  - manipulation
- indexing, retrieval
  - semantic, geometrical
• Objects in a scene may exhibit a certain transparency

• Alpha plane is a generalization of shape information where the value of the bitmap representing the shape is proportional to its transparency

• Typically from 0 (Opaque) to 255 (Fully transparent)
Composition Example

Welcome to MPEG World

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
• **Image** = \( \lambda \cdot \text{Object} + (1-\lambda) \cdot \text{Background} \)
  - \( \lambda = 0 \) the object is transparent
  - \( \lambda = 1 \) the background is invisible

• Successively applied when several objects are to be composited
Composing

Welcome to MPEG World

Composed with binary binary

Composed with alpha plane
Composition

Binary arbitrary

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Alpha planes properties

- Shape and transparency information combined
- 8-bit for the canonical form
- High frequency located on the border
- Presence of large uniform area
- Not directly visible
Utility of an alpha plane

- More flexible than a binary plane.
- Allows a more natural composition of images.
- Anti-aliasing for the composition of natural and synthetic images
- Used to generate synthetic images
Feathering
Alpha plane for synthetic scene composition

Overlay
Compression of an alpha plane

• Problem:
  – More complex than a binary shape.
  – Not a natural image

  Need to code the transparency value of the shape

  Less textures
Destruction (explosion)

Modules of the DFT

Alpha plane

Texture
Alpha plane coding techniques

• An alpha plane can be seen as a gray level picture: all texture coding techniques can be applied to code it

• Binary shape and transparency separation
  – Binary shape coded using a bi-level coding scheme
  – Transparency values coded using a texture coding scheme
Quad-tree VQ coding of alpha planes

- Quad-Tree on 16X16 blocks with VQ

![Quad-tree diagram]

<table>
<thead>
<tr>
<th></th>
<th>y(0)</th>
<th>y(1)</th>
<th>y(2)</th>
<th>y(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>y(4)</td>
<td>y(5)</td>
<td>y(6)</td>
<td>y(7)</td>
<td></td>
</tr>
<tr>
<td>y(8)</td>
<td>y(9)</td>
<td>y(10)</td>
<td>y(11)</td>
<td></td>
</tr>
<tr>
<td>y(12)</td>
<td>y(13)</td>
<td>y(14)</td>
<td>y(15)</td>
<td></td>
</tr>
</tbody>
</table>
• Binary shape and transparency separation
  – DCT on 8X8 blocks or Wavelet
Visual Results

Wavelet

Original

DCT