QUAD-TREE-BASED IMAGE SHAPE CODING WITH BLOCK COMPENSATION

Lih-Yang Wang
Department of Electronic Engineering
Southern Taiwan University of Technology

Outline

- Introduction
- Binary shape coding
- Quad-tree-based shape coding
- Experimental results
- Conclusion
Introduction

- MPEG4 adopts objects-based coding scheme
- An object corresponds to a “Video Object Plane” (VOP)

A video frame composed of (a)VOP1 (b)VOP2

Binary Alpha Plane

- Use Binary Alpha Plane (BAP) to describe an object shape
- BAP is further divided into 16×16 Binary Alpha Blocks (BAB) for shape coding

Fig: (a) Source image (b) Binary alpha plane (c) 16×16 BAB
Binary Shape Coding

- **Bitmap based Shape coding**
  - Advantage: easy to reconstruct BAP
  - Disadvantage: lower compression rate

- **Contour based Shape coding**
  - Advantage: flexibility, higher compression rate
  - Disadvantage: difficult to reconstruct BAP

![Bitmap based](image1.png) ![Contour based](image2.png)

Quad-Tree Representation

- Represents a binary image with a hierarchy data structure
- A BAB can be represented by a quad tree with no more than 5 levels

![Quad-Tree](image3.png)
Three Types of Blocks

Three types of blocks:

- **Homogenous block**
  - Transparent Block (code 0)
  - Opaque Block (code 1)

- **Non-homogenous block**
  - Boundary Block (code 2)

QT Scan Order

The order for Quad-tree scan:
1. from top to down (16→8→4→2→1)
2. from left to right (z direction)
The Problems of Using QT

- Quad tree structure can be quite complicated to describe an image object with
  - Unsmooth boundary
  - Smooth but “non-aligned to quad-division edges”
Quad-tree based Algorithm for Block Compensation (QTBC)

- A Near-Homogeneous block
  - A macro white (black) block with only a single 1x1 black(white) sub-block at a corner

- Block Compensation:
  - Near-Homogenous block → Homogenous

QTBC Flowchart
Run-Length Coding for QTBC (1)

Bottom Level

Horizontal

<table>
<thead>
<tr>
<th>0011</th>
<th>1100</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>(B)</td>
</tr>
</tbody>
</table>

Vertical

<table>
<thead>
<tr>
<th>0101</th>
<th>1010</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C)</td>
<td>(D)</td>
</tr>
</tbody>
</table>

Unlikely to appear

| 1001 | 0110 |

RLC for QTBC (2)

Huffman code

without RLC: 111 111 111

With RLC: 010

reduce 6 bits
Huffman Code

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Huffman code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0011</td>
<td>111</td>
</tr>
<tr>
<td>0101</td>
<td>110</td>
</tr>
<tr>
<td>0111</td>
<td>00000</td>
</tr>
<tr>
<td>1001</td>
<td>00001</td>
</tr>
<tr>
<td>1010</td>
<td>101</td>
</tr>
<tr>
<td>1100</td>
<td>100</td>
</tr>
<tr>
<td>RLC (A)</td>
<td>010</td>
</tr>
<tr>
<td>RLC (B)</td>
<td>001</td>
</tr>
<tr>
<td>RLC (C)</td>
<td>011</td>
</tr>
<tr>
<td>RLC (D)</td>
<td>0001</td>
</tr>
</tbody>
</table>

Comparison between QTSC and QTBC (1)

Akiyo

QTSC: 1358 bits
QTBC: 818 bits
QTBC(+RLC): 785 bits
Compensation: 46 blocks
Bit-rates decrease: 41.4%
Comparison between QTSC and QTBC (2)

**Coast**

- QTSC: **1220** bits
- QTBC: **875** bits
- QTBC(+RLC): **738** bits
- Compensation: **33** blocks
- Bit-rate decreased: **39.5%**

Comparison between QTSC and QTBC (3)

**Foreman**

- QTSC: **1688** bits
- QTBC: **942** bits
- QTBC(+RLC): **878** bits
- Compensation: **77** blocks
- Bit rate decreased: **48%**
Comparison between QTSC and QTBC (4)

Grandma

QTSC: 1827 bits
QTBC: 1074 bits
QTBC(+RLC): 1032 bits
Compensation: 77 blocks
Decrease 43.5% bit rates

Comparison between QTSC and QTBC (5)

News

QTSC: 1781 bits
QTBC: 1105 bits
QTBC(+RLC): 1071 bits
Compensation: 57 blocks
Bit rate decreased: 40%
**Summary of Comparison Results**

<table>
<thead>
<tr>
<th></th>
<th>Akiyo</th>
<th>Coast</th>
<th>Foreman</th>
<th>Grandma</th>
<th>News</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAE</td>
<td>920</td>
<td>761</td>
<td>1017</td>
<td>1108</td>
<td>1073</td>
</tr>
<tr>
<td>QTSC</td>
<td>1358</td>
<td>1220</td>
<td>1688</td>
<td>1827</td>
<td>1781</td>
</tr>
<tr>
<td>QTBC (without RLC)</td>
<td>818</td>
<td>875</td>
<td>942</td>
<td>1074</td>
<td>1105</td>
</tr>
<tr>
<td>QTBC (with RLC)</td>
<td>779</td>
<td>723</td>
<td>881</td>
<td>1029</td>
<td>1068</td>
</tr>
</tbody>
</table>

**Conclusion**

- A lossy quad-tree-based shape coding algorithm has been proposed.
- A run-length scheme is added to further improve the compression rate.
- Experimental results have demonstrated the efficiency of the proposed algorithm.