H.264 MEPSO
Motion Estimation & Partitioning Selection Optimization

Students: Sharon Gingold, Yaniv Mazor
Supervisor: Ran Bar-Sella
Sponsor: Pivot Signal Processing Ltd.
Talk Outline

• Quick overview: Video Coding & H.264
• Motion Estimation (ME) Problem
• Partitioning Selection (PS) Problem
• Project’s Goal
• Literature Review
• System Level Solution
• Performance Simulations and Results
H.264 Encoder Scheme (baseline)

~ 70% Of the time
Motion Estimation is a **BIG** problem!

**Tradeoff:**
Best estimation improves bit-rate…
BUT impossible to find in Real Time.

**Concept:**
Find sub-optimal, much faster estimation.

**Currently no single DSP solution for H.264**

**History:**
A lot of work was already done, however
- No System Level Real Time Solution
- Doesn’t integrate H.264 new features, such as Variable Block Shapes.

Click to show sequence
Variable Block Size

• **Improvement:**
One of H.264’s most important additions is a variable block size option – to better model the motion field.

• **Tradeoff:**
Block size Partitioning needs to be based on the image
Each partitioning needs a separate ME stage

• **Concept:**
Estimate sub-optimal partitioning.

• **History:**
No existing low-complexity alg.
All algorithms are ‘greedy’
with full ME for each candidate partitioning

This makes our problem 259 times more difficult....
Project Goals

Develop Motion Estimation (ME) and Partitioning selection (PS) algorithms optimized for Real-Time single DSP implementation

Challenges

• **System Level Solution**, covering all aspects of estimation.
• **One Chance**: Minimal number of calculations in one pass.
• **Re-use** of calculations in different stages.
• **Little** quality degradation and **significant** bits reduction.

Uniqueness

• Combination of independent solutions + original ideas
• one search per block
• Solution is part of a whole system
• Sharing of calculations
Proposed ME System

**System Level Approach**
Each block optimizes the whole system, rather than itself

**Hybrid Motion Estimation System Scheme:**

- Current Frame
- Reference Frame
- Initial Block Size Map Generator
- Budget Constraints
- One Pass Motion Search (ME)
- Block Size Optimization
- Encode one Frame

Reference Frame
Bit alloc.
Partitioning
Initial map
Final map
stream
Literature Review

optimizations are based on one of the following

How to calculate the error? (what pixels to compare)

Scan subset of pixels for comparison

What Motion Vectors to Check?

Scan subset of solution space
Innovative ME Approach

• Traditional flows reduce complexity by either searching less points OR calculating error with less pixels.

• Our algorithm integrates both approaches as well as our own original ideas to form a novel efficient solution.

• Perfectly integrates in the System Level.
Example: Pixel Decimation

an Example for innovative original ideas:

Difference Estimation

full error calculation – most accurate, but VERY expansive

Choose a subset of the pixels in the block

Use Variability of Pixels in each block

Choose randomly only one pixel

Choose randomly two pixels
Variable Block Size Optimization

H.264 allows **259** different macroblock partitioning options for INTER prediction.

<table>
<thead>
<tr>
<th>16x16</th>
<th>16x8</th>
<th>8x16</th>
<th>8x8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- 8x8
- 8x4
- 4x8
- 4x4

8 pels

- 16x16 allows low-computational cost
- Intensive motion require smaller partitions (PSNR vs. Bit Rate)

**Sub-Optimal Solution required:**

*Predict best out of 259 options (One Time Pass) based on:*

- Activity Detector
- Previous frame’s partitioning (reference)
- Previous frame’s bit consumptions (adaptive)
No One Pass algorithm was found in literature.

All multi-search-per-block techniques involve with initial partitioning and refinements.

Basic Scheme of Quad-Tree based algorithm (Top-Down)

- Too complex
- Insufficient

There is also bottom up…
Partitioning Selection Observations

Explored Full-Search coded sequences in order to

- Characterize nature of partitioning
- Find its properties
- Correlate the final partition map to some possible inputs.

Significant interesting behavior was discovered, based on which the algorithms were developed.
Partitioning Optimization

Let’s say that two adjacent blocks found the same motion vector.

We’d like to unite them to one, and by that to save the motion information.
Partitioning Optimization

- Unifying adjacent blocks is possible if: \( dX + dY \leq 2 \)
Partitioning Optimization

- Unifying adjacent blocks is possible if: 
  $dX + dY \leq 4$
Low complexity System

Intentional Error !!!

Texture detector

- Get Bit Usage of MB in reference frame
- Below TH?
- No: Decrease block size group = 1
- Yes: Increase block size group = 3

Activity detector

- Decimated MB SAD energy
- Below TH?
- No: More than two blocks are high
- Yes: Only one 8x8 block is high

Budget Constraints

One Pass Motion Search (ME)

Block Size Optimization

Encode one Frame

Initial Block Size Map Generator

Current Frame

Reference
Innovative Concepts

The Five-“S” ace !!!

• System Level Optimization (intentional error)
• Single Pass Decision Making
• Sharing of Computations
• Statistical Activity Decimation
• Simple (Efficient) Detectors
Simulation and Performance Results

Simulation and Performance Results

H.264 MEPSO by Yaniv Mazor & Sharon Gingold
Kasher Contest, June, 2005
MEPSO – Analysis of distance from optimal solution

Simulation and Performance Results
Conclusions

• Low complexity algorithm:
  – 2 KSAD/frame (<0.5% of FS complexity)
  – One pass operation
  – DSP friendly algorithm

• Quality penalty – minimal (<0.5dB PSNR)

• Currently being patented

• DSP implementation
Thank you

The End