Video Algorithms and Architectures

K.A. Vissers
Philips Research

and
A.K. Riemens, R.J. Schutten, G. de Haan
Philips Research
Eindhoven, The Netherlands

Contents

• Context of TV systems
• Video Format Conversion
• De-interlacing and Up-Conversion
• Motion Estimation and Motion Compensation
• Several Architectures
• Conclusion
Context of TV systems

- Consumer electronics: price for electronics $50 - 100
- Real-time performance: no loss of data, guaranteed response
- Embedded systems, certainly for the display processing
- Image quality is important

Frequencies for standard resolution TV:
- PAL: 864 pixels, 625/2=312 lines, 50 fields/sec, interlace
- NTSC: 864 pixels, 525/2=262 lines, 60 fields/sec, interlace

In total:
13.5 million samples per second,
luminance (y) and chrominance (u and v, subsampled),
typically 8-10 bits data for luminance and 8 bits for color

System signal processing: 100 - 1000 operations per pixel:
1.35 - 13.5 Billion operations per second,
1.35 - 13.5 Gbyte per second internal bandwidth
Context of TV systems

Algorithms for picture quality improvements:
- Sharpness “improvement”
- Noise Reduction
- De-interlace
- Up-conversion

Increasing demand for conversion
Good reasons for different formats

Standardization is no option
- Channel capacity differs
- Viewing distance differs
- History may be different
- Screen brightness differs
- Resolution requirements differ
- Motion portrayal of different importance

Large Area Flicker on TV

Perception threshold for large area flicker as a function of brightness

- Brightness (cd/m² or nit)
- Picture update frequency (Hz)

- Brightness values: 0.01, 0.1, 1, 10, 100, 1000
- Picture update frequencies: 30, 38, 47, 60, 72
Field Rate Conversion from 50 Hz to 100Hz

- Original field
- Interpolated field

Video format conversion problem

- Video contains time-discrete information:
  - In the temporal domain (discrete number of pict/s)
  - In the vertical domain (discrete number of lines)
  - Often in the horizontal domain (number of pels/line)

- Why not use interpolating and decimating low-pass filters to achieve our goal?
  - TV does not fulfill demands of sampling theorem in V and T domains
  - Tracking viewers transform temporal frequencies
Field rate conversion

First **De-interlace**:  
- vertical temporal filter  
- motion estimation, followed by motion compensated techniques

Next perform **Up-conversion**:  
- motion estimation, followed by motion compensated techniques

In the end:  
- Motion compensated processing in any algorithm that uses previous fields or frames: intra-field noise reduction etc.

What is interlace

<table>
<thead>
<tr>
<th>Spatial domain</th>
<th>Temporal domain</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Spatial domain diagram" /></td>
<td><img src="image2" alt="Temporal domain diagram" /></td>
</tr>
</tbody>
</table>

- **n-1** denotes the previous field or frame.

- **n** denotes the current field or frame.
Why de-interlace

- Some displays require progressive video (matrix type of displays)
- Eliminate line flicker, resolution loss with motion, and alias
- Basic requirement for all scan conversion (even when converting from interlaced to interlaced format)
  - e.g. field rate doubling preventing odd-odd-even-even field sequence

De-interlacing, what is it?

Calculate picture data at TV-lines not transmitted in the current field
Vertical Temporal Filter (VT)

- Original pixel
- Interpolated pixel

VT Median and moving edge

- Ideal interpolation
- VT-median interpolation
- Edge field n
- Edge field n-1
De-interlacing moving images

- The problem with motion is fundamental for all methods without motion compensation
  
  Since

- information of successive fields cannot be combined because of motion, while single fields cannot provide full vertical detail
  
- Motion compensation aims at achieving the same quality for moving image parts as for stationary parts

De-interlacing summary

- For stationary images many methods perform well.

- For moving images, only motion compensated methods perform reasonably good.

- Motion compensated methods have been introduced in consumer products already.

Reference articles:
- G. de Haan and E.B. Bellers, “De-interlacing-An overview, Overview article accepted for publication in the Proceedings of the IEEE.”
**Up-conversion**

- t-T
- t
- t+T
- t+2T

50 Hz input fields
100 Hz output fields

**Picture rate conversion**

moving object at picture rate

- n-1
- n
- n+1
- n+2

original
Picture rate conversion, what if we just repeat the most recent image at the output?

Picture rate conversion, this is what we hoped for.

Original

Repeated

Motion compensated

Original
Tracking by viewer
so moving images need to be sharp

- **Fixed eye**
  - Moving detailed object
  - High temporal frequency
  - Brightness
  - Time

- **Tracking eye**
  - Moving detailed object
  - Moving field of view
  - Zero temporal frequency
  - Brightness
  - Time

Motion Compensated picture rate up-conversion

- n-1
- n-1/2
- picture number
- n
- D/2
- -D/2
Picture rate conversion, can we notice the improvement?

Non - Motion Compensated  Motion Compensated

Motion Estimation

So this is what we need:

- Is there any motion?
- How fast?
- into which direction?
Up-conversion summary

- Motion compensated upconversion is required for good quality
- Robust methods are important

- Reference articles:
  - G. de Haan et al., “IC for Motion Compensated 100Hz TV, with a Smooth Motion Movie-Mode”, IEEE Tr. on Consumer Electronics, May 1996, pp. 165-174.

Architectural Considerations

Combine:
- Motion estimation: 3D recursive search
- Deinterlace
- Upconversion
- Preferably noise reduction and aspect ratio scaling

Into:
ONE consumer priced IC, or a part of a platform in ONE consumer based platform.
Motion Estimation

The problem: find the best block position at a number of candidate positions, comparing data of the current field/frame with date of the previous field/frame.

Motion Estimation Questions:

Find a perceptively **good** ME that requires limited:
- external memory
- internal memory
- computational load.
- What is a good ME? Iterative development loop:
  - Propose an algorithm for ME
  - Implement de-interlace and or up-conversion with it
  - Evaluate the video quality and the cost of implementation!

Trade-offs with very non-linear and implicit cost functions
**Choices for Motion Estimation**

Algorithmic choices have a video quality and cost impact:

- Number of previous fields/frames used, e.g. one frame (~ 1 Mbyte, external off chip memory)
- Search range, typically +/- 12 - 16 in vertical direction, +/- 30-40 in horizontal direction (~ 10-100kByte internal memory)
- Block size for comparison, typically 8x8 to 16x16
- Accuracy of vectors, typically 0.25 pixel!, so 2D interpolation is required inside the motion estimation
- Number of pixels used in the calculation of the Sum of differences: typically a sub-sample of a factor 2-4.

**Combination of Motion Estimation, De-interlace and Up-conversion**

- Combine the field/frame memories,
- Combine the De-interlacing with Motion Estimation: recursive de-interlacing
- Use motion vectors for De-interlace
- Calculate new motion vectors for Up-conversion at proper temporal location (vector split)
System Trade-offs, solution 1

Existing IC (SAA 4991), used in high end Philips TVs

- All pixel processing in dedicated synthesized hardware.
- Frame/field memories of chip
- Line memories for the search range on-chip.
- Control settings and field level adjustment in a micro-controller (8051)
- Good video quality, implementation tuned for the TV market
- Includes noise reduction and vertical scaling of the image
- Runs synchronous with the video scanning frequency

IC characteristics:

<table>
<thead>
<tr>
<th>Process</th>
<th>CMOS 0.8 µm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die Size</td>
<td>97 mm²</td>
</tr>
<tr>
<td>Transistor Count</td>
<td>980,000</td>
</tr>
<tr>
<td>Data Clock</td>
<td>27 or 32 MHz</td>
</tr>
<tr>
<td>Package</td>
<td>PLCC84</td>
</tr>
<tr>
<td>Dissipation</td>
<td>1.8 W</td>
</tr>
<tr>
<td>Interface</td>
<td>UART-bus</td>
</tr>
<tr>
<td>ME/MC Range</td>
<td>±16 (H), ±9 (V) pixels</td>
</tr>
</tbody>
</table>
System Trade-offs solution 1)

- Dedicated, synthesized, tuned solution

System trade-offs, solution 2)

Attempt to implement the de-interlacing and up-conversion completely in software exploiting the opportunities of Philips Trimedia VLIW core:

- Data parallelism 4 bytes in a word in the 32 bit architecture,
- Special Media instructions making SAD calculations and Median calculations very efficient
- Instruction Level Parallelism exploiting: 4.5 out of 5 issueslot effectively used over the complete program
- Some video quality limitations to achieve the software solution
System Trade-offs, solution 2)

**PCI-card**
- CVBS, or Y/C input
- Digital multi-standard decoder (SAA7111)

**SDRAM (8MB)**

**Philips TriMedia (TM1000)**

**PCI-bus**
- MPEG in
- MC-video out

**TM1000 function:**
- Software:
  - Object based ME
  - Robust MVS interpol.
  - Film detection (2-2 & 2-3)
  - De-interlacing
  - VT-median / weave
- Hardware:
  - Video I/O
  - PCI-bridge

**DAC - System Level Design with Embedded Platforms**

---

**TM1000 overview**

- **SDRAM**
  - video-in
  - Serial I/O
  - PCI bridge
  - I²C I/O
  - audio-out
  - audio-in
  - VLIW cpu
  - I$ D$
TM1000 VLIW core

32 KB instr cache
16 KB data cache,
quasi dual ported,
8-way set associative

128 words x 32 bits register file
5 ALU, 5 const, 2 shift,
3 branch
2 l/FPmul, 2 FPalu, 1 FPdivsqrt, 1 FPcomp
2 loadstore, 2 DSPalu, 2 DSPmul
Pipelined, latency 1 to 3 cycles (except FPdivsqrt)

System Trade-offs, proposal 3)

- New proposal: joint effort of Philips Research, Philips
  Trimedia and Philips Semiconductors Business Line
  Video.
- Best video quality
- Partitioning of total functionality in Software on the
  TM-core and a dedicated new coprocessor, with on
  chip internal memory for the search range
- Combined with many other functions and features in
  the context of TV processing
- Runs completely de-coupled from the video input
  frequency or the video output frequency, and
  independent of the video scanning direction.
Conclusion

- The feasibility of Motion Estimation and Motion Compensated De-interlacing and Up-Conversion has been shown.
- Several implementations with a range in video quality have been illustrated.
- Quantifying the system trade-offs for next generation systems is essential.
- The combination of powerful Media processor cores with flexible coprocessors is unique in this field.
- De-coupling of video scanning opens new algorithmic opportunities.