An Adaptive and Parallel Scheme for HD Video De-interlacing

Jie Dong and King Ngi Ngan
{jdong, knngan}@ee.cuhk.edu.hk

Department of Electronic Engineering
The Chinese University of Hong Kong

ICME2008, June 23-26 2008, Hannover, Germany

Outline

- Introduction
- Proposed solution
- Experimental results
- Conclusion

Interlaced Video

- Benefit of interlaced video
  - trade off between frame rate and vertical resolution
- Format of HD interlaced videos
  - 1920x1080, 50/60 field per second
- Application of HD interlaced videos
  - HD broadcasting (Australia, USA, Canada, Singapore, Hong Kong and some European countries)
- Problem
  - compatible with CRT-based HDTV sets (never entered the market in large volume)
  - not compatible with the PC community sets

De-interlacing

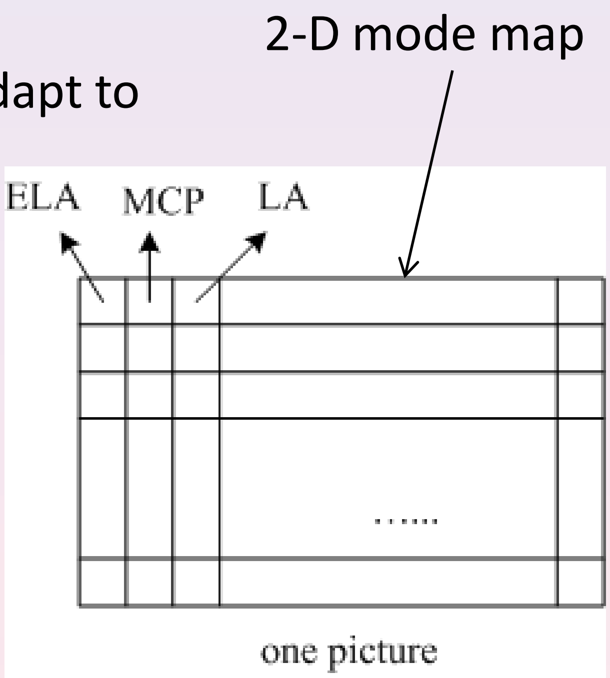
- Problem statement
  - Interpolating the missing lines in a frame
- Challenge of real-time de-interlacing for HD videos
  - Simple
  - Smart (inevitably employ implicit or explicit motion/texture detectors)
- Related Work
  - Aiming at one requirement
    - Simple enough to fulfill the real-time requirement at the expense of visual quality (various linear and median filters)
    - Slow but employ advanced techniques (motion/texture detection, MCP)
  - Taking both aspects into consideration

Overview of the Proposed Solution

- General Idea
  - Make full use of the values of the SE in bitstreams
    - MV: the direction and speed of motion
    - MB partition: the smoothness of motion or texture
    - Direction of intra prediction: texture
    - Transform coefficients: edge
  - Decision of interpolation method
    - SE values
    - Accuracy of SE values
- Advantages
  - Adapt to local activities
  - Save time for texture or motion detection
  - Reliable

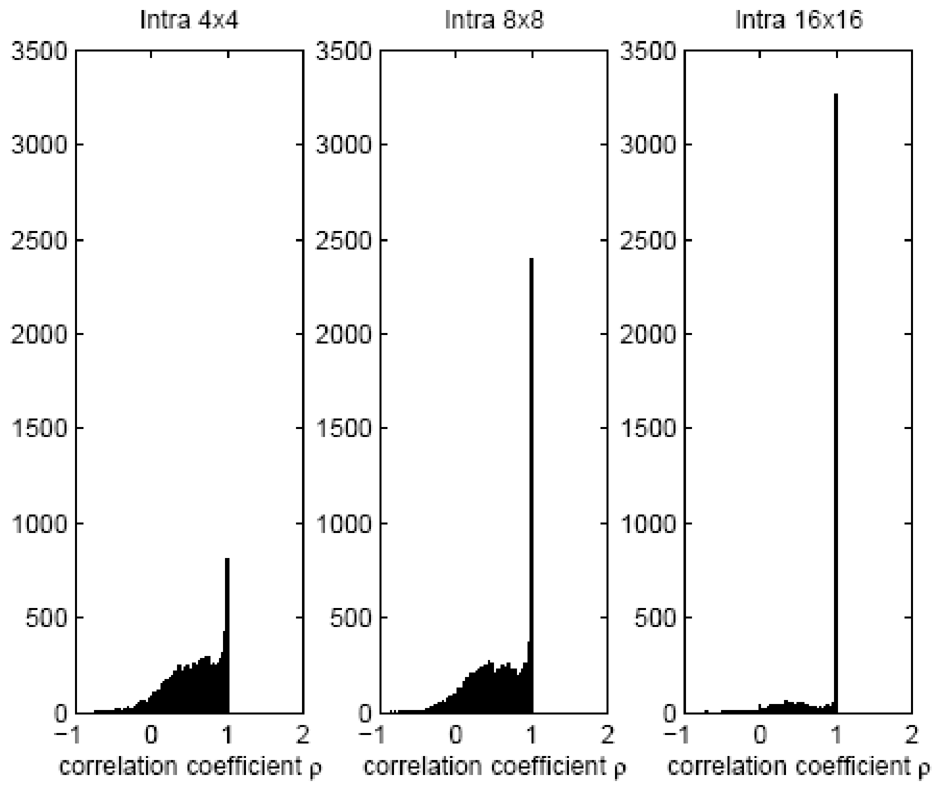
Realization of the Proposed Solution

- Processing unit
  - the smallest area interpolation methods adapt to
- 2-D Mode map
  - 2-D array storing the interpolation method of each processing unit in one picture
- Decision stage
  - Produce 2-D mode map
  - In decoding order
- Interpolation stage
  - Apply the interpolation indicated in the 2-D mode map
  - Behind decision stage, in the display order
  - Parallel performed according to the number of CPU cores



Decision Stage for Intra-coded MB

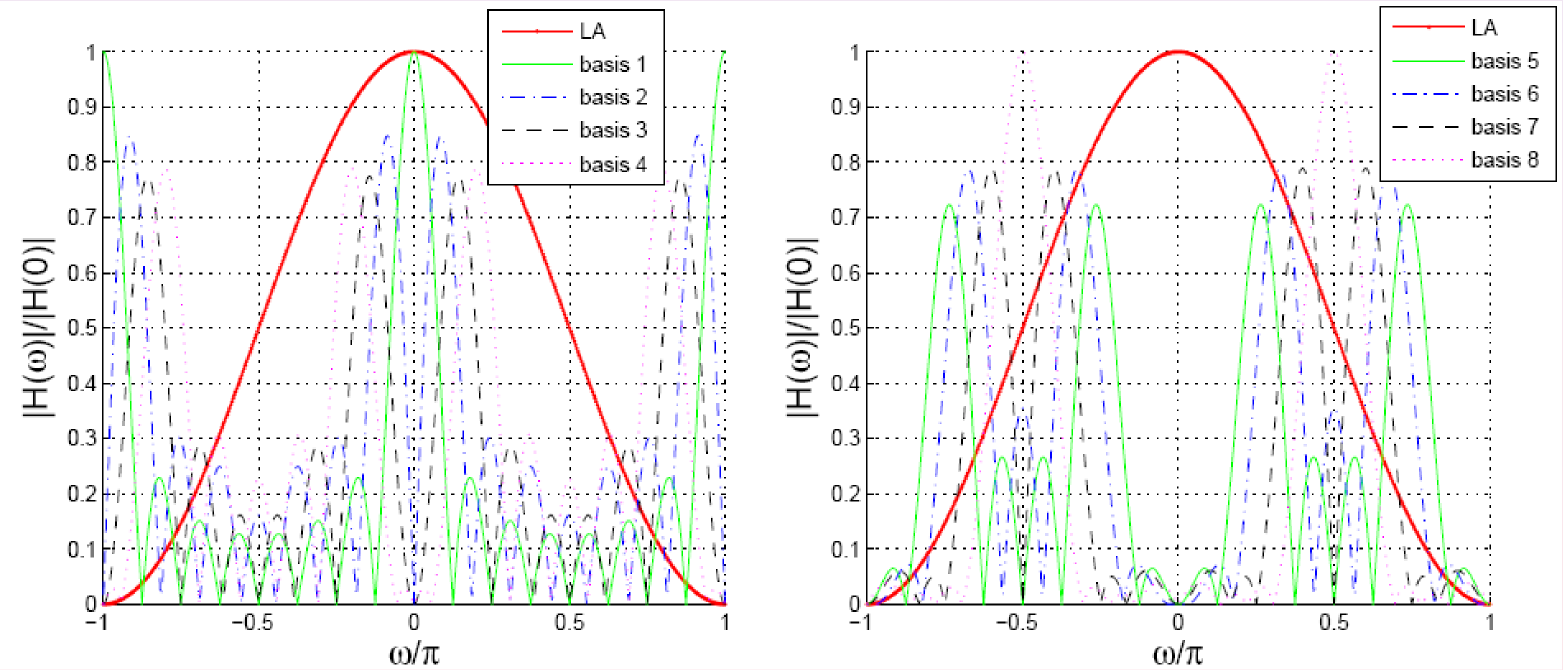
- The relationship between intra prediction size and the texture
  - 16x16: smooth (LA)
  - 4x4: fine (ELA)
  - 8x8: smooth or fine
- For intra8x8-coded MB
  - Vertically smooth (LA)
  - Otherwise (ELA)



The histogram of vertical correlation in intra-coded MBs (based on *StockholmPan*)

Decision Stage for Intra-coded MB

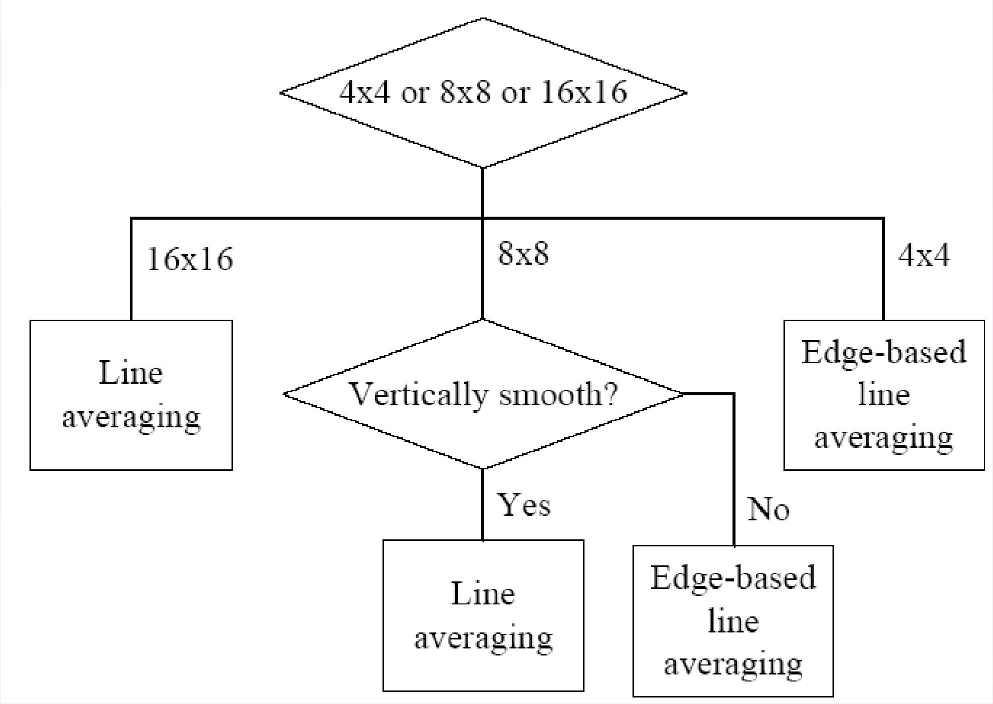
- Why LA is suitable for vertically smooth intra8x8-coded MB?
  - The spectra of the low-frequency bases of the 8x8 ICT in H.264 are in the pass-band of the LA



Frequency response of LA, and the spectra of the first 4 bases (the left figure) and the rest 4 bases (the right figure) of the 8x8 ICT in H.264

Decision Flow for Intra-coded MB

- Problem
  - Switch between LA and ELA
  - At the mercy of aliasing
  - For all-intra-coded sequences, the defect becomes obvious.



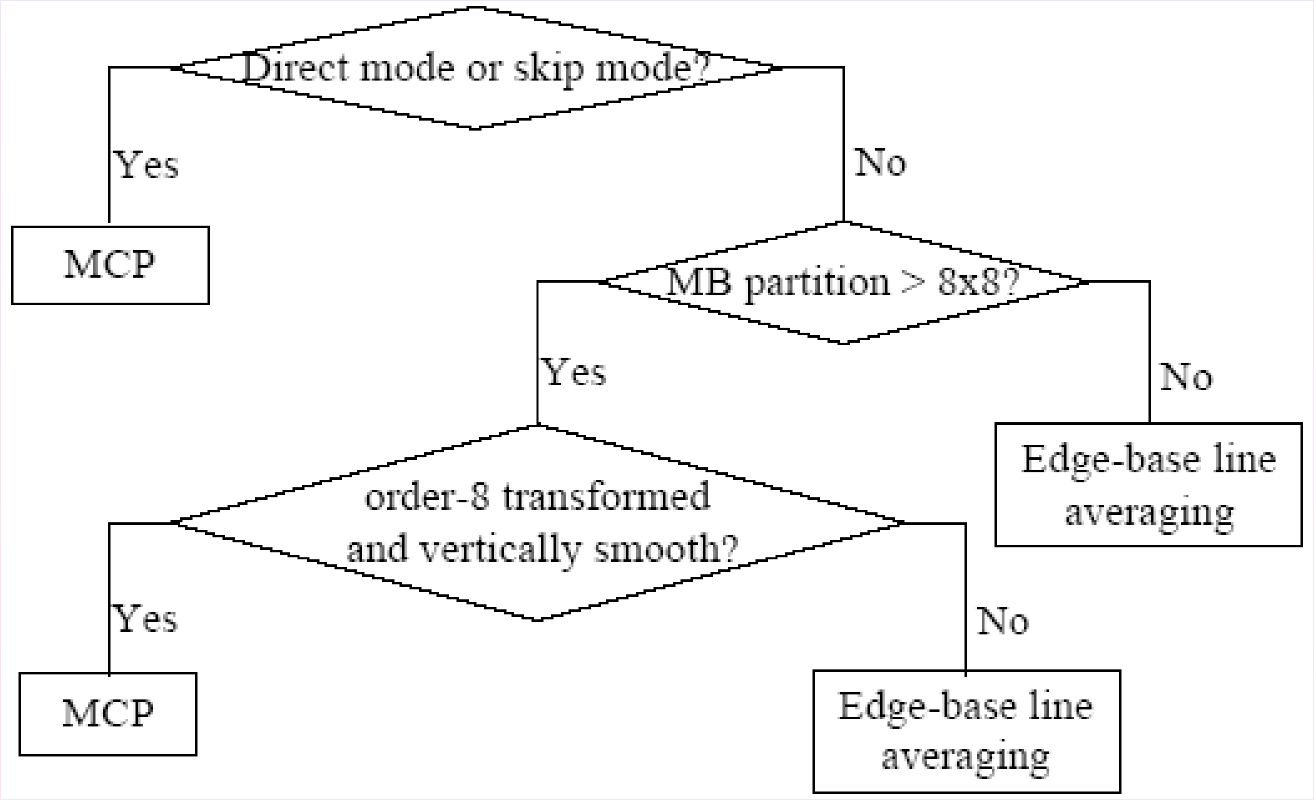
Decision Stage for Inter-coded MB

- The percentage of different MB types in B frames
  - The proportion of skip and direct modes is large for HD videos.
  - With the two modes, MVs are derived from those of spatially or temporally neighbouring blocks and thus indicate uniform translation.

The percentage of different MB types in B frames

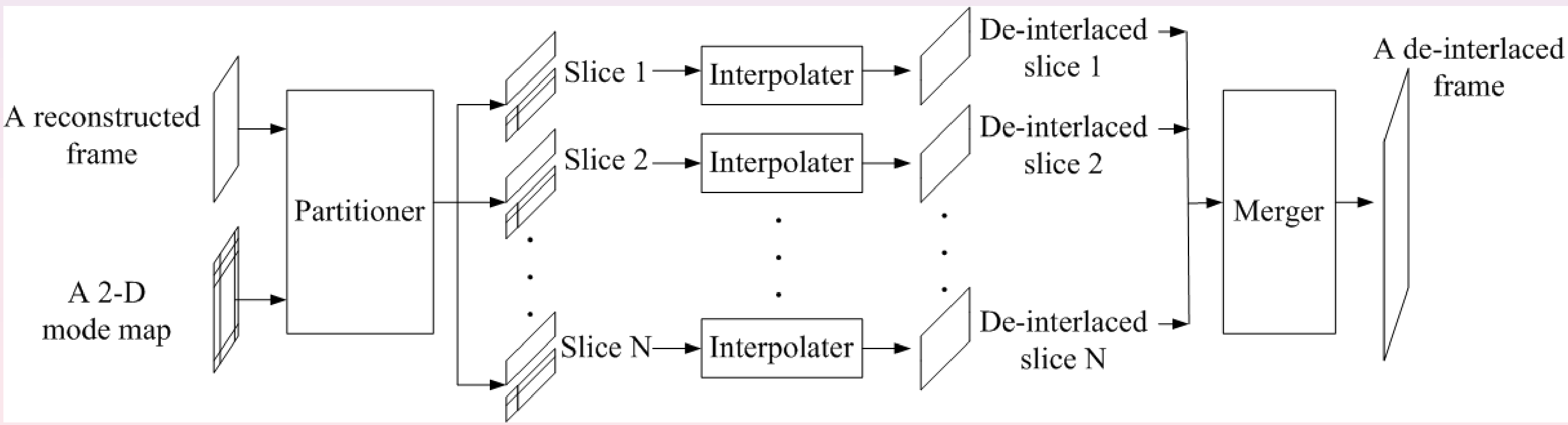
Test sequence	Skip/Direct	16×16	16×8	8×16	8×8	Intra
Parkrun	69.09	19.11	3.07	1.95	6.74	0.04
Shields	68.81	16.49	3.35	2.62	8.57	0.21
StockholmPan	71.22	13.70	3.76	2.85	8.43	0.09

Decision Flow for Inter-coded MB



Interpolation Stage

- Parallel Implementation
  - Perform with the decision stage in parallel
  - Itself can be divided into parts to be interpolated independently according to the number of CPU cores.



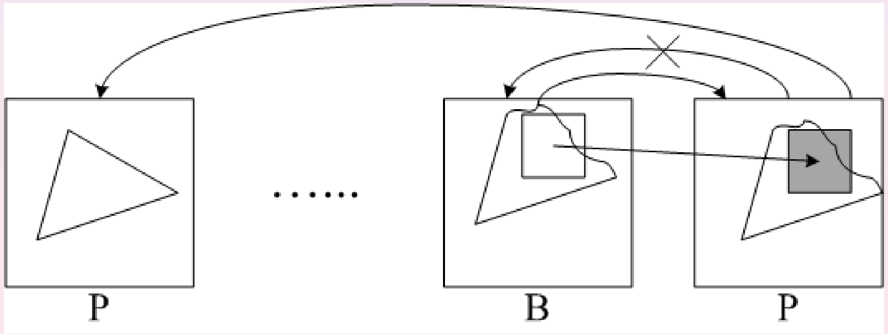
The Motion Compensation Method

- Same partition and directions of prediction as those of the associated MB
- Only the immediately previous and next fields are used as reference for MCP method, regardless of the number of reference frames in H.264 bitstreams.
- MV always approximately points to the existing pixels.
- The pixels used for MCP are passed a 3-tap MF rather than being used directly

$$p_{i,n} = median(p_{i-1,n}, p_{i+1,n}, r)$$

Motivation of the Complementary MCP (CMCP)

- Updated the improper method in the interpolation stage
  - The interpolation method decided in the decision stage may not be proper, as the syntax values do not necessarily reflect the real motion due to the encoding strategies and the restriction of the standards.
  - Common example: the intra-coded MBs in P frames from IBBPBBP-structured sequence

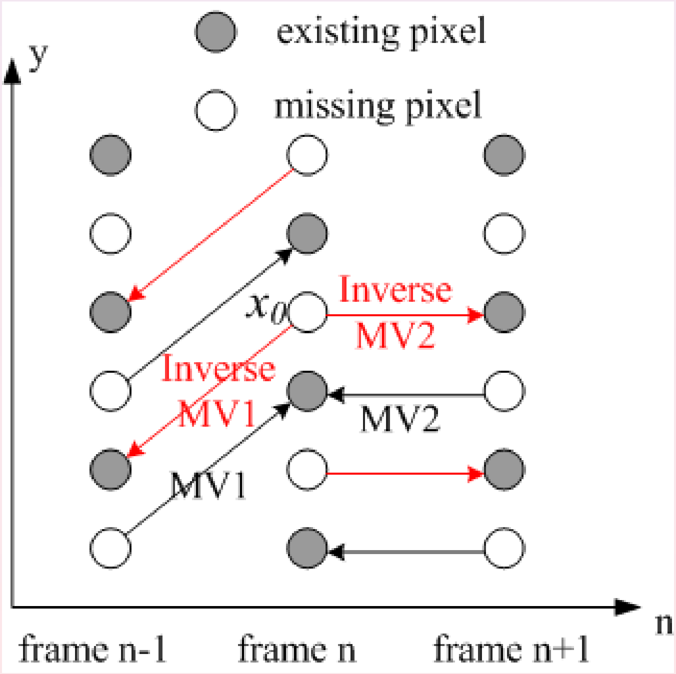


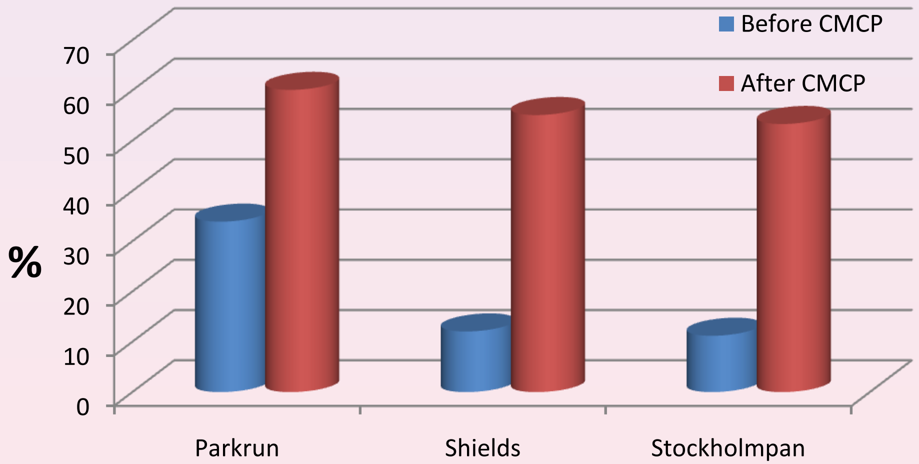
The percentage of different MB types in P frames

Test sequence	Skip	16×16	16×8	8×16	8×8	Intra
Parkrun	11.36	30.82	15.80	11.04	17.76	13.3
Shields	4.15	31.09	16.21	13.21	23.47	12.1
StockholmPan	7.07	22.53	14.46	11.41	24.73	19.92

The Complementary MCP (CMCP)

- frame *n-1*, *n*, *n+1* are B, P, B frames in the bitstream
- frame *n-1*, *n+1* can be used as reference frame of frame *n* by the proposed de-interlacer, which is not allowed by the bitstream.
- The MVs are the inverse of those pointing from B frame to P frame





The percentage of MCP method before and after CMCP

Test Condition

- Compare with common real-time de-interlacers
  - Remove serrations, which often remain after temporal linear filtering.
  - Eliminate flickers, while intra-field methods cannot.
- Compared with
  - M. J. Chen, C. H. Huang and C. T. Hsu, “Efficient de-interlacing technique by inter-field information,” *IEEE Trans. Consum. Electron.*, vol.50, pp.1202-1208, Nov. 2004.

Subjective Tests (1)

Reference

Proposed

Subjective Tests (2)

Reference

Proposed

Subjective Tests (3)

Reference

Proposed

Computational Time

- Execution Time
  - Testing system: Two 3.0 GHz Dual Core Intel Xeon 5160, 1 GB RAM.
  - Sequences: 1080i, 116 fields
  - Unit: second

Test sequence	IPP dec.	IPP dec.+De-interlacing	Computational time
Flamingo	4.354	4.478	0.124
Kayak	4.208	4.423	0.215
Mountain	4.426	4.639	0.213
Parkrun	4.335	4.375	0.040
Shields	4.109	4.697	0.588
Stockholmpan	4.687	4.929	0.242
Average	4.353	4.590	0.237

Conclusion

- An efficient real-time de-interlacer specially for H.264 coded video
  - Makes full use of SE values as the hints of motions and textures
  - Not entirely rely on SE values
  - Suitable for parallel implementation
- Results
  - Provide good visual quality
  - Fulfill the real-time requirement

Thank You !

Q & A