Control Strategies for H.264 video decoding under resources constraints

Work-In-Progress

Anne Marie Alt - Daniel Simon

INRIA Rhône-Alpes
NeCS project-team

Febid Workshop
March 13th, 2010, Paris
Outline

1. ARAVIS Project
2. H.264/SVC Overview
   - Principle
   - Video structure
   - Decoding Process
3. Parallelization strategy : Computing Accelerator
4. Control Strategies
   - Control Architecture
   - Frame controller
   - Quality Controller
   - Model Calibration - Decoder Behavior
   - Frame Controller : Deadline control
   - Preliminary Results
5. Conclusion
Context: **ARAVIS** is a R&D project of the Minalogic pole of excellence.

The partners are:
- STMicroelectronics
- CEA
- TIMA Laboratory
- INRIA Sardes and NeCS teams

Purpose: Integration of **32 nm** scale future **multi-core** chips.

3 control loops needed for energy management

Problems:

- Variability problems in nanometric scale chips
- Energy management: Mobile devices, Green labels
3 control loops needed for energy management

Problems:
- Variability problems in nanometric scale chips
- Energy management: Mobile devices, Green labels

Feedback Control: adaptivity and robustness
- Frequency/Voltage control
- Computing speed controller
- Quality of Service Control:

Joint Control of Video Quality and Energy Consumption
3 types of scalability

- **Temporal scalability**: Frame rate - ex: TV : 25 fps.
- **Spatial scalability**: Resolutions - ex: HD 1920x1080 pixels.
- **Quality scalability**: Quantization step - Contrast.
- Combination of these scalabilities is also possible.
3 types of scalability

- **Temporal scalability**: Frame rate - ex: TV : 25 fps.
- **Spatial scalability**: Resolutions - ex: HD 1920x1080 pixels.
- **Quality scalability**: Quantization step - Contrast.
- Combination of these scalabilities is also possible.
3 types of scalability

- **Temporal scalability**: Frame rate - ex: TV: 25 fps.
- **Spatial scalability**: Resolutions - ex: HD 1920x1080 pixels.
- **Quality scalability**: Quantization step - Contrast.

Combination of these scalabilities is also possible.
**H.264/SVC Overview Principle**

3 types of scalability

- **Temporal scalability**: Frame rate - ex: TV : 25 fps.
- **Spatial scalability**: Resolutions - ex: HD 1920x1080 pixels.
- **Quality scalability**: Quantization step - Contrast.
- **Combination of these scalabilities is also possible.**
Decoding executed from the worst quality to the best quality.

Higher Quality $\Rightarrow$ Higher Computation load and Energy consumption.
Video structure

- Video structure and display order

- Decoding order

- Display and Decoding are not synchronous
- 9 pictures Buffer is mandatory for IBBBPBBBI decoding
- Extra Buffer of 3 pictures depth for control purpose
Decoding Process

$Q_p = \text{Quantization step}$
Decoding Process

Qp = Quantization step

Frame
Resolution QCIF - 176x144
- Qp = 28
- Qp = 26
- Qp = 24

Resolution CIF - 352x288
- Qp = 28
- Qp = 26
- Qp = 24

Slice 1
Slice 2
Slice 3
Slice 4
Slice 5
Slice 6
Slice 7
Slice 8

Initialisation
Slice parsing
Decoding
Loop filter

Timer

actuators sorted by increasing penalty order

Display
Decoding

Fixed Frame Rate

Action means:
- Adjust the deadline
- Skip the loop Filter
- Skip superior quality layers
- Skip the picture
Parallelization strategy: Computing Accelerator

A picture is divided into quality levels and slices. We chose the **slice-level parallelization**. We used the **event-based programming model**.

The video sequence is decoded sequentially. Quality levels are decoded sequentially.

**Reference software**: JSVM.

- **Resolution 176x144**: Qp = 28, Qp = 26, Qp = 24
- **Resolution 352x288**: Qp = 28, Qp = 26, Qp = 24

Frame:

```
Slice 1 | Slice 2 | Slice 3 | Slice 4 | Slice 5 | Slice 6 | Slice 7 | Slice 8
```

```
Cpu 0   | Cpu 1   | Cpu 2   | Cpu 3   | Cpu 4   | Cpu 5   | Cpu 6   | Cpu 7
```
Control Architecture

<table>
<thead>
<tr>
<th>Controller</th>
<th>Goal</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing speed Controller</td>
<td>Voltage and Frequency domain control</td>
<td>Frequency and Voltage set points</td>
</tr>
<tr>
<td>Frame Controller</td>
<td>Decoding control</td>
<td>Deadline</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Estimated cycles number</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>QoS Control : Video Quality and Energy</td>
<td>Requested Quality layer of the video</td>
</tr>
</tbody>
</table>

Control Strategies for H.264 video decoding under resources constraints

Feblid 2010
Frame controller: Compute the deadline and computation load
Objective: Obtain deadlines close to the frame rate.

Inputs:
- Objective: Average deadline = Frame Rate
- Main inputs:
  - actual deadline
  - cycles number for last frames
- Ratio between Vhigh and Vlow
- Final Quality level reached

Error:
- Deadline overshoot

Outputs:
- Estimated computation load
- Deadline Handler
Quality controller: Compute the desired quality level video

Inputs:
- Objective: Available energy and End-users requirements
- State: Ressources state
- Aggregated costs:
  - Current Quality
  - Energy Consumption

Outputs:
- Requested Service:
  - Decoding Quality
  - Energy consumption
Model Calibration - Decoder Behavior

Quantization and Resolution have a significant impact on the computation load.
⇒ these actuators are effective

Control Strategies for H.264 video decoding under resources constraints Febid 2010 12 / 17
Model Calibration - Decoder Behavior

Model Calibration: Decoder Behavior

Decoding times of frames are content-dependent:

- "Flat" sections
- Isolated peaks

Cycles number estimated by last pictures.
Rejection of peaks: Frame controller and extra Buffer.
Graphs: decoding times (ms) functions of Frames.

Peaks are often due to changes of ground.
Purpose: Damping the overshoot - Keep a Buffer of 3 pictures

\[ d_r = t_k - t_{k-1} \]
\[ d_k = t_k + \delta_k \]
\[ \delta_{k+1} = \beta \times \delta_k \]
\[ 0 < \beta < 1 \]
\[ d_{r_{k+1}} = t_{k+1} + \beta \times \delta_k \]

\( t_k \): Ideal deadline for decoding frame k
\( d_r \): Constant inter-frame interval
\( d_k \): Actual deadline for frame k
\( d_{r_{k+1}} \): Requested deadline for frame k+1
\( \delta_k \): Overshoot for frame k
\( \beta \): Gain
Preliminary Results

Beta = 0.8 - Video : 900 pictures - Deadline = 55ms

- Ideal null error
- Overshoot without control
- Overshoot with control

Penalties: (cf Philips)
Missed deadline: -10000
q0 : 5 - q1 : 10
Increase quality: -10
Decrease Quality: -50
Conclusion

Summary:
- Joint Energy Consumption and Video Quality control
- Integrated Feedback in Hardware and Software

Work In Progress:
- Sensors and actuators identification
- Experimental setup
- First sketch of control loops
- Even basic control improves decoder performance
- Control computation cost is negligible.
Thank you for your attention

any questions?