Adaptive Resolution Change using Uncoded Areas and Dictionary Learning-based Super-Resolution in Versatile Video Coding
Contents

1. Motivation and ARC Fundamentals

2. Uncoded Areas for ARC

3. Dictionary Learning-based Super-Resolution

4. Simulation Setup and Experimental Results
Motivation

- Dictionary Learning-based super-resolution showed promising results when applied to inter-layer prediction in SHVC [1].
Motivation

- Dictionary Learning-based super-resolution showed promising results when applied to inter-layer prediction in SHVC [1].
- The concept of adaptive resolution change is already known from MPEG 4 [2] and raised attention recently [3].
Motivation

- Dictionary Learning-based super-resolution showed promising results when applied to inter-layer prediction in SHVC [1].
- The concept of adaptive resolution change is already known from MPEG 4 [2] and raised attention recently [3].
- The convex hull of the RD curve can be estimated by downsampling the video before coding [4].

![RD-curves for Campfire sequence (left) and Basketballdrive (right). First 100 frames, RA coding configuration.](image)
### Adaptive Resolution Change

- On which level of the encoding scheme should the resolution change happen?

<table>
<thead>
<tr>
<th>Option</th>
<th>Signaling Cost</th>
<th>Spatial Adaptivity</th>
<th>Temporal Adaptivity</th>
<th>Boundary Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU level</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CTU level</td>
<td>moderate</td>
<td>yes</td>
<td>yes</td>
<td>moderate</td>
</tr>
<tr>
<td>Picture level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>TID level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>Intra period level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>Sequence level</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>almost none</td>
</tr>
</tbody>
</table>

SVCP-19 ARC scheme on the CTU level:
- Tricky regarding the implementation
- Limited to intra frames and in terms of coding gains.

SVCP-20/21 ARC scheme on the picture level:
- Code the picture at full and half resolution.
- Upsample or apply SR to downsampled reconstructed pictures.
- Decide based on RD-cost which one is coded into the bitstream.
Adaptive Resolution Change

On which level of the encoding scheme should the resolution change happen?

<table>
<thead>
<tr>
<th>Option</th>
<th>signaling cost</th>
<th>spatial adaptivity</th>
<th>temporal adaptivity</th>
<th>boundary issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU level</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CTU level</td>
<td>moderate</td>
<td>yes</td>
<td>yes</td>
<td>moderate</td>
</tr>
<tr>
<td>picture level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>TID level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>intra period level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>sequence level</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>almost none</td>
</tr>
</tbody>
</table>

SVCP-19 ARC scheme on the CTU level
– tricky regarding the implementation
– limited to intra frames and in terms of coding gains.
Adaptive Resolution Change

On which level of the encoding scheme should the resolution change happen?

<table>
<thead>
<tr>
<th>Option</th>
<th>signaling cost</th>
<th>spacial adaptivity</th>
<th>temporal adaptivity</th>
<th>boundary issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU level</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CTU level</td>
<td>moderate</td>
<td>yes</td>
<td>yes</td>
<td>moderate</td>
</tr>
<tr>
<td>picture level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>TID level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>intra period level</td>
<td>low</td>
<td>no</td>
<td>yes</td>
<td>almost none</td>
</tr>
<tr>
<td>sequence level</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>almost none</td>
</tr>
</tbody>
</table>

SVCP-19 ARC scheme on the CTU level
- tricky regarding the implementation
- limited to intra frames and in terms of coding gains.

SVCP-20/21 ARC scheme on the picture level.
- Code the picture at full and half resolution.
- Upsample or apply SR to downsampled reconstructed pictures.
- Decide based on RD-cost which one is coded into the bitstream.
1. Motivation and ARC Fundamentals

2. Uncoded Areas for ARC

3. Dictionary Learning-based Super-Resolution

4. Simulation Setup and Experimental Results
Based on the cost function $J = D \cdot \lambda_R$, the decision on which area will be coded is made.

The quantization parameter for the low resolution video is lowered such that $Q_{\text{LR}} = Q_{\text{HR}}$

Upsampling and downsampling – Using the SHVC filters $h^\# = »2, 3, 9, 6, 39, 58, 39, 6, 9, 3, 2 \frac{1}{128}$ $h^" = »1, 0, 4, 0, 11, 0, 40, 64, 40, 0, 11, 0, 4, 0, 1 \frac{1}{64}$ or a machine learning-based method?
Based on the cost function $J = D + \lambda R$ the decision on which area will be coded is made.

- The quantization parameter for the low resolution video is lowered such that $QP_{\text{LR}} = QP_{\text{HR}} - 6$ [5].
Uncoded Areas for ARC

- Based on the cost function $J = D + \lambda R$ the decision on which area will be coded is made.

- The quantization parameter for the low resolution video is lowered such that $QP_{LR} = QP_{HR} - 6$ [5].

- Upsampling and downsampling
  - Using the SHVC filters
    \[
    h_\downarrow = \left[2, -3, -9, 6, 39, 58, 39, 6, -9, -3, 2\right]/128 \\
    h_\uparrow = \left[-1, 0, 4, 0, -11, 0, 40, 64, 40, 0, -11, 0, 4, 0, -1\right]/64
    \]
  - or a machine learning-based method?
Contents

1. Motivation and ARC Fundamentals

2. Uncoded Areas for ARC

3. Dictionary Learning-based Super-Resolution

4. Simulation Setup and Experimental Results
A dictionary is typically trained using vectorized training patches $x_i$ of a size $s_p \times s_p = 8 \times 8$:

$$D \leftarrow \arg \min_D \sum_{i=1}^{n} \frac{1}{2} \|x_i - D\alpha_i\|^2 + \lambda \|\alpha_i\|_1$$
Dictionary Learning Fundamentals

• A dictionary is typically trained using vectorized training patches $x_i$ of a size $s_p \times s_p = 8 \times 8$:

$$D \leftarrow \arg\min_D \sum_{i=1}^{n} \frac{1}{2} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

• A sparse representation of an image patch is found by sparse encoding the patch $x$ in the dictionary $D$:

$$\alpha \leftarrow \arg\min_\alpha \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_1$$

$$x = D\alpha + \varepsilon$$
Dictionary Learning Fundamentals

• A dictionary is typically trained using vectorized training patches \( x_i \) of a size \( s_p \times s_p = 8 \times 8 \):

\[
D \leftarrow \arg \min_D \sum_{i=1}^{n} \frac{1}{2} \| x_i - D\alpha_i \|_2^2 + \lambda \| \alpha_i \|_1
\]

• A sparse representation of an image patch is found by sparse encoding the patch \( x \) in the dictionary \( D \):

\[
\alpha \leftarrow \arg \min_{\alpha} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1
\]

\[
x = D\alpha + \varepsilon
\]

• The concept of dictionary learning can be used for super-resolution by training coupled dictionaries [6].

Figure: Example Dictionary
DLSR: Coupled dictionaries approach
DLSR: Coupled dictionaries approach

\[ I_{LR}^{\uparrow} \]

\[ D_{LR} \]

\[ D_{HR} \]

\[ x_{LR} \approx D_{LR} \alpha \]

\[ x_{HR} \approx D_{HR} \alpha \]
DLSR: Coupled dictionaries approach

\[ I_{\text{LR}}^\uparrow \]

\[ I_{\text{LR}} \approx \alpha_{164} + \alpha_{171} + \alpha_{179} + \alpha_{206} \]

\[ D_{\text{LR}}, D_{\text{HR}} \]

\[ x_{\text{LR}} \approx D_{\text{LR}} \alpha \]

\[ x_{\text{HR}} \approx D_{\text{HR}} \alpha \]
DLSR: Mathematical Summary

dictionary learning for LR:  \( D_{LR} \leftarrow \arg \min_{D_{LR}} \sum_{i=1}^{n} \frac{1}{2} \| x_{LR,i} - D_{LR} \alpha_i \|_2^2 + \lambda \| \alpha_i \|_1 \)

dictionary learning for HR:  \( D_{HR} \leftarrow \arg \min_{D_{HR}} \| X_{HR} - D_{HR} A \|_2^2 \)
DLSR: Mathematical Summary

dictionary learning for LR: 
\[ D_{LR} \leftarrow \arg \min_{D_{LR}} \sum_{i=1}^{n} \frac{1}{2} \|x_{LR,i} - D_{LR} \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \]

dictionary learning for HR: 
\[ D_{HR} \leftarrow \arg \min_{D_{HR}} \|X_{HR} - D_{HR} \alpha\|_2^2 \]

sparse coding: 
\[ \alpha \leftarrow \arg \min_{\alpha} \frac{1}{2} \|x - D_{LR} \alpha\|_2^2 + \lambda \|\alpha\|_1 \]

general function approximation: 
\[ x_{HR} \approx D_{HR} \left( \arg \min_{\alpha} \frac{1}{2} \|x_{LR} - D_{LR} \alpha\|_2^2 + \lambda \|\alpha\|_1 \right) \]
1. Motivation and ARC Fundamentals

2. Uncoded Areas for ARC

3. Dictionary Learning-based Super-Resolution

4. Simulation Setup and Experimental Results
Simulation Setup and Results

- DLSR setup
  - patchsize $s_p \times s_p = 8 \times 8$
  - number of atoms $K = 512$
  - sparse coding penalties
    - $\lambda_{\text{train}} = 0.23$
    - $\lambda_{\text{test}} = 0.19$

- video coding setup
  - 4K sequences from JVET testset
  - anchor VTM-5.0
  - $\text{QP} = 37, 42, 47, 52$
Simulation Setup and Results

• DLSR setup
  – patchsize $s_p \times s_p = 8 \times 8$
  – number of atoms $K = 512$
  – sparse coding penalties
    • $\lambda_{\text{train}} = 0.23$
    • $\lambda_{\text{test}} = 0.19$

• video coding setup
  – 4K sequences from JVET testset
  – anchor VTM-5.0
  – $QP \in \{37, 42, 47, 52\}$
Simulation Setup and Results

• DLSR setup
  – patchsize $s_p \times s_p = 8 \times 8$
  – number of atoms $K = 512$
  – sparse coding penalties
    • $\lambda_{\text{train}} = 0.23$
    • $\lambda_{\text{test}} = 0.19$

• video coding setup
  – 4K sequences from JVET testset
  – anchor VTM-5.0
  – $\text{QP} \in \{37, 42, 47, 52\}$
Simulation Setup and Results

• DLSR setup
  – patchsize $s_p \times s_p = 8 \times 8$
  – number of atoms $K = 512$
  – sparse coding penalties
    • $\lambda_{\text{train}} = 0.23$
    • $\lambda_{\text{test}} = 0.19$

• video coding setup
  – 4K sequences from JVET testset
  – anchor VTM-5.0
  – $\text{QP} \in \{37, 42, 47, 52\}$

<table>
<thead>
<tr>
<th>All Intra</th>
<th>Random Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{\uparrow}$</td>
<td>DLSR</td>
</tr>
<tr>
<td>Campfire</td>
<td>$-19.5%$</td>
</tr>
<tr>
<td>CatRobot1</td>
<td>$-9.6%$</td>
</tr>
<tr>
<td>DaylightRoad2</td>
<td>$-7.2%$</td>
</tr>
<tr>
<td>FoodMarket4</td>
<td>$-8.0%$</td>
</tr>
<tr>
<td>ParkRunning3</td>
<td>$-16.4%$</td>
</tr>
<tr>
<td>Tango2</td>
<td>$-10.0%$</td>
</tr>
<tr>
<td>AVG</td>
<td>$-11.8%$</td>
</tr>
</tbody>
</table>

Table: Bjøntegaard Delta rate savings
Conclusion

- Coding gains with respect to VTM 5.0 can be achieved by performing an adaptive resolution change on the picture level

- Dictionary learning based super-resolution leads to an additional coding gain
  - For All Intra additional 0.7% of rate savings are achieved.
  - For Random Access the additional gain is even higher and the rate can be reduced by further 0.9%.
Conclusion

- Coding gains with respect to VTM 5.0 can be achieved by performing an adaptive resolution change on the picture level.
- Dictionary learning based super-resolution leads to an additional coding gain.
  - For All Intra additional 0.7% of rate savings are achieved.
  - For Random Access the additional gain is even higher and the rate can be reduced by further 0.9%.
Thank you for your attention!

Any questions?

Jens Schneider schneider@ient.rwth-aachen.de

Institut für Nachrichtentechnik, RWTH Aachen University
www.ient.rwth-aachen.de
References


• Downsampling realized by taking e.g. every second sample
• This introduces alias in general
  – The signal is filtered with an anti-aliasing filter
• Upsampling is realized by inserting zeros
  – The signal is filtered with an interpolation filter
• MATLAB’s `imresize` function does not strictly follow this methodology, when using the bicubic kernel
  – samples are shifted when downsampling and shifted back when upsampling
**Fundamentals: Downsampling Filters**

- Bicubic downsampling filter has 8 taps
  - This introduces a phase shift of the downsampled signal

- The downsampling filter used in SHVC has 11 taps
  - no distortion of the phase during downsampling

Figure: different downsampling filters
Fundamentals: Downsampling Filters

Figure: Frequency response of different downsampling filters
Fundamentals: Upsampling Filters

- Bicubic upsampling filter has to be applied several times since we need to “backshift” the phase

- The upsampling filter is derived from the half-pel interpolation filters used in HEVC
  - We need to insert a 1 at position zero and 0s at the odd sample positions

![Diagram](a) bicubic upsampling filter

![Diagram](b) upsampling filter derived from HEVC interpolation filters

Figure: different upsampling filters