MULTIPLE FEATURE-BASED CLASSIFICATIONS
ADAPTIVE LOOP FILTER (MCALF)

Johannes Erfurt, W. Lim, H. Schwarz, D. Marpe, T. Wiegand
CONTENT

- Introduction to In-Loop Filtering
- Adaptive Loop Filtering
- ALF and GALF
- Multiple Feature-based Classifications
  - Feature Descriptors
  - Concept of Confidence Level
- Simulation Results
- Conclusion
Introduction to In-Loop Filtering

- In-loop filtering is applied after reconstruction of coding blocks.
- Filtered picture is stored in decoded picture buffer and may be used for prediction.

**DBF** = Deblockling Filter

**SAO** = Sample Adaptive Offset
Adaptive Loop Filtering

\(X = \) original samples, \(Y = \) reconstructed samples

- Each pixel location is classified into one of \(L\) classes \(C_1, ..., C_L\) based on local features
- Estimate multiple Wiener filters \(F_l\) for each \(C_l, \ l = 1, ..., L\)
- \(F_l\) minimizes mean square error (MSE) between \(X\) and \(\tilde{X}\)

\[
\tilde{X} = \sum_{\ell=1}^{L} \chi_{C_\ell} \cdot (Y * F_l) \quad \text{with} \quad \chi_{C_\ell}(i, j) = \begin{cases} 
1, & \text{if } (i, j) \in C_\ell \\
0, & \text{otherwise}
\end{cases}
\]
Adaptive Loop Filtering

\( X = \) original samples, \( Y = \) reconstructed samples

- Each pixel location is classified into one of \( L \) classes \( \mathcal{C}_1, \ldots, \mathcal{C}_L \) based on local features
- Estimate multiple Wiener filters \( F_l \) for each \( \mathcal{C}_l, \ l = 1 \ldots L \)
- \( F_l \) minimizes mean square error (MSE) between \( X \) and \( \tilde{X} \)

\[
\tilde{X} = \sum_{\ell=1}^{L} \chi_{\mathcal{C}_\ell} \cdot (Y * F_l) \quad \text{with} \quad \chi_{\mathcal{C}_\ell}(i, j) = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{C}_\ell \\ 0, & \text{otherwise} \end{cases}
\]

How do we perform classification into \( \mathcal{C}_1, \ldots, \mathcal{C}_L \)?
ALF and GALF

- ALF was proposed for HEVC standard and further developed resulting in GALF
- Certain coding tools make ALF/GALF very efficient:
  - Classification including directional gradients
  - Adaptively chosen filter support for each frame (5x5, 7x7, 9x9)
  - Block-wise on/off-flag
  - Temporal prediction: Use previously coded filter coefficients
  - Class merging
Multiple Feature-based Classifications

- MCALF: Multiple Feature-based Classifications ALF
- Test at encoder side $M$ classifiers $C_{l_1}, \ldots, C_{l_M}$
- Each classifier has a certain feature descriptor $D$ to group each pixel location into classes

$$C_{\ell} = \{(i, j) \in I : D(i, j) = \ell\} \quad \text{for } \ell = 1, \ldots, K$$

- Classifier with best RD performance chosen
- Filter index and possible filter information are signaled
Feature Descriptors

- Each classifier is described through different feature descriptors $D$
- Laplacian feature descriptor $D_L$: includes computation of directional gradients for squared blocks, e.g. in vertical direction:

$$g(i, j) = |2 \cdot Y(i, j) - Y(i - 1, j) - Y(i + 1, j)|$$
Feature Descriptors

- Pixel-based feature descriptor $D_P$:

$$D_P(i, j) = \left\lfloor \frac{(K - 1)}{2^B} Y(i, j) \right\rfloor$$

for bit depth $B$ and $K$ classes

$Y(i, j) \in [0, 2^B - 1]$
Feature Descriptors

- **Pixel-based feature descriptor** $D_P$:
  
  $$D_P(i, j) = \left\lfloor \frac{(K - 1)}{2^B} Y(i, j) \right\rfloor \quad \text{for bit depth } B \text{ and } K \text{ classes}$$

- **Ranking-based feature descriptor** $D_R$:
  
  $$D_R(i, j) = \# \{(k_1, k_2) : Y(i, j) \geq Y(k_1, k_2) \text{ for } |k_1 - i| \leq l, |k_2 - j| \leq h \} + 1$$

![Diagram showing the calculation of $D_R$ with example values]
Feature Descriptors

- **Product of two feature descriptors** \( D_1 : I \rightarrow \{1, \ldots, K_1\} \) and \( D_2 : I \rightarrow \{1, \ldots, K_2\} \)

\[
D(i, j) = (D_1(i, j), D_2(i, j)) \in \{1, \ldots, K_1\} \times \{1, \ldots, K_2\}
\]
Concept of Confidence Level

- Optimal classes for $L = 2$:

  $\tilde{C}_1 = \{(i, j) \in I \mid Y(i, j) \leq X(i, j)\}$,
  
  $\tilde{C}_2 = \{(i, j) \in I \mid Y(i, j) > X(i, j)\}$
Concept of Confidence Level

- Optimal classes for $L = 2$:
  
  $$\tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\},$$
  
  $$\tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\}$$

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $C_1^{pre}, \ldots, C_K^{pre}$
**Concept of Confidence Level**

- Optimal classes for $L = 2$:
  
  $\tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\}$
  
  $\tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\}$

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $\mathcal{C}_1^{pre}, \ldots, \mathcal{C}_K^{pre}$

Inner ellipse should be mostly contained in left or right half
Concept of Confidence Level

- Optimal classes for $L = 2$:
  \[ \tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\} \]
  \[ \tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\} \]

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $C_1^{pre}, \ldots, C_K^{pre}$
Concept of Confidence Level

- Optimal classes for $L = 2$:
  $$\tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\},$$
  $$\tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\}$$

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $C_{1}^{pre}, \ldots, C_{K}^{pre}$
Concept of Confidence Level

- Optimal classes for $L = 2$:
  \[ \tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\}, \]
  \[ \tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\} \]

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $C_1^{pre}, \ldots, C_K^{pre}$
Concept of Confidence Level

- Optimal classes for $L = 2$:
  \[ \tilde{C}_1 = \{(i, j) \in I : Y(i, j) \leq X(i, j)\} \]
  \[ \tilde{C}_2 = \{(i, j) \in I : Y(i, j) > X(i, j)\} \]

- $X$ not available at decoder: Approximate $\tilde{C}_1$ and $\tilde{C}_2$

- Use certain feature descriptor $D$ to receive $K$ pre-classes $\mathcal{C}_1^{pre}, \ldots, \mathcal{C}_K^{pre}$
Concept of Confidence Level

- Calculate for each class $c_k^{pre}$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

$$p_{k,1} = \frac{\#(c_k^{pre} \cap \tilde{C}_1)}{\#(c_k^{pre})}, \quad p_{k,2} = \frac{\#(c_k^{pre} \cap \tilde{C}_2)}{\#(c_k^{pre})}$$
Concept of Confidence Level

- Calculate for each class $C_k^{pre}$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

$$p_{k,1} = \frac{\#(C_k^{pre} \cap \tilde{C}_1)}{\#(C_k^{pre})}, \quad p_{k,2} = \frac{\#(C_k^{pre} \cap \tilde{C}_2)}{\#(C_k^{pre})}$$

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \rightarrow \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 1 & p_{k,1} > p \\ 2 & p_{k,2} > p \\ 0 & \text{otherwise} \end{cases}$
Concept of Confidence Level

- Calculate for each class $C^\text{pre}_k$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

$$p_{k,1} = \frac{\#(C^\text{pre}_k \cap \tilde{C}_1)}{\#(C^\text{pre}_k)}, \quad p_{k,2} = \frac{\#(C^\text{pre}_k \cap \tilde{C}_2)}{\#(C^\text{pre}_k)}$$

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \to \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 
1 & p_{k,1} > p \\
2 & p_{k,2} > p \\
0 & \text{otherwise.}
\end{cases}$
Concept of Confidence Level

- Calculate for each class $C_k^{pre}$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

$$p_{k,1} = \frac{\#(C_k^{pre} \cap \tilde{C}_1)}{\#(C_k^{pre})}, \quad p_{k,2} = \frac{\#(C_k^{pre} \cap \tilde{C}_2)}{\#(C_k^{pre})}$$

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \rightarrow \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 1 & p_{k,1} > p \\ 2 & p_{k,2} > p \\ 0 & \text{otherwise} \end{cases}$

- $C_\ell^c = \{(i, j) \in I : P_D(D(i, j)) = \ell\}$ for $\ell = 1, 2$ high confidence classes
Concept of Confidence Level

- Calculate for each class $C^\text{pre}_k$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$:

$$p_{k,1} = \frac{\#(C^\text{pre}_k \cap \tilde{C}_1)}{\#(C^\text{pre}_k)}, \quad p_{k,2} = \frac{\#(C^\text{pre}_k \cap \tilde{C}_2)}{\#(C^\text{pre}_k)}$$

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \rightarrow \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 1 & p_{k,1} > p \\ 2 & p_{k,2} > p \\ 0 & \text{otherwise.} \end{cases}$

- $C^e_\ell = \{(i, j) \in I : P_D(D(i, j)) = \ell\}$ for $\ell = 1, 2$ high confidence classes
Concept of Confidence Level

- Calculate for each class $C_k^{pre}$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

\[
p_{k,1} = \frac{\#(C_k^{pre} \cap \tilde{C}_1)}{\#(C_k^{pre})}, \quad p_{k,2} = \frac{\#(C_k^{pre} \cap \tilde{C}_2)}{\#(C_k^{pre})}
\]

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \rightarrow \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 1 & p_{k,1} > p \\ 2 & p_{k,2} > p \\ 0 & \text{otherwise.} \end{cases}$

- $C_\ell^e = \{(i, j) \in I : P_D(D(i, j)) = \ell\}$ for $\ell = 1, 2$ high confidence classes

- $C_\ell = \{(i, j) \notin C_1^e \cup C_2^e : \tilde{D}(i, j) = \ell\}$ for $\ell = 1, \ldots, \tilde{K}$ classes for remaining pixel locations
Concept of Confidence Level

- Calculate for each class $C_k^{pre}$ confidence level $p_{k,1}$ and $p_{k,2}$ of $D$

$$p_{k,1} = \frac{\#(C_k^{pre} \cap \tilde{C}_1)}{\#(C_k^{pre})}, \quad p_{k,2} = \frac{\#(C_k^{pre} \cap \tilde{C}_2)}{\#(C_k^{pre})}$$

- For $p \in (1/2, 1)$ define map $P_D : \{1, \ldots, K\} \rightarrow \{0, 1, 2\}$ with $P_D(k) = \begin{cases} 1 & p_{k,1} > p \\ 2 & p_{k,2} > p \\ 0 & \text{otherwise.} \end{cases}$

- $C_\ell^e = \{(i, j) \in I : P_D(D(i, j)) = \ell\}$ for $\ell = 1, 2$ high confidence classes

- $C_\ell = \{(i, j) \notin C_1^e \cup C_2^e : \tilde{D}(i, j) = \ell\}$ for $\ell = 1, \ldots, \tilde{K}$ classes for remaining pixel locations

- This gives $\tilde{K} + 2$ classes $C_1^e, C_2^e, C_1, \ldots, C_{\tilde{K}}$
Simulation Results

- Test conditions:
  - JEM-7.0 with QP points: 27, 32, 37, 42
  - Random Access Main 10 (RA)

- 5 Classifiers for MCALF:
  - $D_P(K = 27)$
  - Product of $D_R$ and $D_P(K = 27)$
  - $D_L(K = 25)$
  - $D_P$ for $C^e_1, C^e_2(p = 0.63, K = 20)$ and $D_L(\tilde{K} = 25)$
  - $D_R$ for $C^e_1, C^e_2(p = 0.63, K = 9)$ and $D_L(\tilde{K} = 25)$

<table>
<thead>
<tr>
<th>Test Sequence</th>
<th>BD Rate (Y) RA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BQTerrace 1920 x 1080</td>
<td>-1.34%</td>
</tr>
<tr>
<td>MarketPlace 1920 x 1080</td>
<td>-0.90%</td>
</tr>
<tr>
<td>Rollercoaster 3840 x 2160</td>
<td>-2.33%</td>
</tr>
</tbody>
</table>

Encoder run-time: 107%
Decoder run-time: 100%

Coding gains of MCALF (5 classifiers) with reference GALF (1 classifier $D_L$)
Conclusion

- Performance of adaptive loop filter highly depends on classification
- Multiple classifications can better adapt to local features in video sequence
- Classification is performed through feature descriptors such as $D_L$, $D_P$, or $D_R$, and classification with confidence level
- We can get more than 2% coding gain on top of GALF with no increase of decoder runtime and only small increase of encoder runtime
Thank you!
## More Results

<table>
<thead>
<tr>
<th>Test Sequence</th>
<th>BD Rate (Y) RA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BQTerrace 1920 x 1080</td>
<td>-5.85%</td>
</tr>
<tr>
<td>MarketPlace 1920 x 1080</td>
<td>-3.35%</td>
</tr>
<tr>
<td>Rollercoaster 3840 x 2160</td>
<td>-6.31%</td>
</tr>
</tbody>
</table>

Coding gains of MCALF (5 classifiers) with reference JEM-7.0 - GALF