Intra-Prediction in Daala

Nathan Egge

Daala Code Party
1 October 2013
How Daala is Different

- Multiple Block Sizes
  - 4x4, 8x8, 16x16 (32x32?)
- Lapped Transforms
  - No need for adaptive loop-filter
- Frequency Domain Intra-Prediction
  - Exploit compact frequency representation
- Pyramid Vector Quantization (PVQ)
- Chroma From Luma
Time Domain Intra-Prediction

Pros:
- Uses image data from neighboring blocks
  - Only need to remember 1 pixel border
- Parameterizable for any angle $\theta$
- Predicts difficult to code features well
  - edges are extended
- Efficient implementation (no multiplies)

Cons:
- Blocks L, UL, U, UR must be decoded
- Poor prediction in textured areas
Time Domain Intra-Prediction

The intra-prediction modes for 4x4 blocks in WebM (VP8).
Lapped Transforms

Diagram showing the process of applying a prefiltre to input blocks, transforming them with DCT and DCT^{-1} operations, and then outputting the transformed blocks.
Decoding an Intra Frame

Neighboring blocks:

- Decoded Image
- Predicted
- Unpredicted
- Currently Predicting
- Needs Post-filter
- Prediction Support
Freq. Domain Intra-Prediction

\[
X = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots
\end{bmatrix}, \quad Y = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots
\end{bmatrix}
\]

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix}^T \begin{bmatrix}
X \\
Y
\end{bmatrix} = \begin{bmatrix}
X^TX & Y^TX \\
X^TY & Y^TY
\end{bmatrix}
\]

\[
C = X^TX \\
D = X^TY \\
E = Y^TX \\
F = Y^TY
\]

Online update:

\[
C = X^TX \\
D = X^TY \\
E = Y^TX \\
F = Y^TY
\]

Normalize:

\[
C' = S_x C S_x \\
D' = S_x D S_y \\
\beta_1' = C'^{-1} D'
\]

Model Fitting:

\[
\beta_1 = S_x \beta_1' S_x^{-1} \\
\beta_0 = \bar{Y} - X \beta_1 \\
y = \beta_0 + \beta_1 x
\]

Least Squares:

\[
\beta_1' = C'^{-1} D'
\]

Predict:

\[
y = \beta_0 + \beta_1 x
\]
K-Means Training

Classify Image Blocks (initially use VP8 modes)
Fit Model (linear prediction, least squares)
For $k = 1$ to $N$
  Reclassify Image Blocks (use Daala modes)
  Fit Model
Done
Demo

How good is a predictor?

- Coding Gain (Cg) measures how well energy is compacted into a few coefficients:

\[
C_g = 10 \log_{10} \frac{\sigma_x^2}{\left( \prod_{i=0}^{N-1} \sigma_{y_i}^2 \|h_i\|^2 \right)^{1/N}}
\]

- Prediction Gain (Pg) is the improvement in coding gain due to prediction.
## Non-Sparse Results

<table>
<thead>
<tr>
<th>Subset</th>
<th>Size</th>
<th>VP8 Cg Ref</th>
<th>VP8 Cg Res</th>
<th>VP8 Pg</th>
<th>Daala Cg Ref</th>
<th>Daala Cg Res</th>
<th>Daala Pg</th>
</tr>
</thead>
<tbody>
<tr>
<td>subset1</td>
<td>4x8</td>
<td>11.4386</td>
<td>14.2018</td>
<td>2.76327</td>
<td>12.3286</td>
<td>14.6232</td>
<td>2.29461</td>
</tr>
<tr>
<td>subset3</td>
<td>4x8</td>
<td>13.8511</td>
<td>16.7665</td>
<td>2.9154</td>
<td>14.96</td>
<td>17.1703</td>
<td>2.21032</td>
</tr>
<tr>
<td>subset1</td>
<td>8x16</td>
<td>12.4887</td>
<td>13.3316</td>
<td>0.84291</td>
<td>12.8817</td>
<td>13.8135</td>
<td>0.93176</td>
</tr>
<tr>
<td>subset3</td>
<td>8x16</td>
<td>15.1202</td>
<td>15.9894</td>
<td>0.8694</td>
<td>15.6468</td>
<td>16.3716</td>
<td>0.72476</td>
</tr>
<tr>
<td>subset1</td>
<td>16x32</td>
<td>12.9038</td>
<td>13.0725</td>
<td>0.16871</td>
<td>13.1332</td>
<td>13.5621</td>
<td>0.42895</td>
</tr>
<tr>
<td>subset3</td>
<td>16x32</td>
<td>15.587</td>
<td>15.7818</td>
<td>0.19483</td>
<td>15.8721</td>
<td>16.248</td>
<td>0.37593</td>
</tr>
</tbody>
</table>
K-Means with Sparsification

Classify Image Blocks (initially use VP8 modes)

Fit Model (linear prediction, least squares)

For \( k = 1 \) to \( N \)

   Reclassify Image Blocks (use Daala modes)

   If \( k > M \)

      Drop prediction coefficients (use prediction gain, \( P_g \))

   End

   Fit Model

Done
K-Means with Sparsification

Classify Image Blocks (initially use VP8 modes)

Fit Model (linear prediction, least squares)

For $k = 1$ to $N$

Reclassify Image Blocks (use Daala modes)

If $k > M$

Drop prediction coefficients (use prediction gain, $P_g$)

End

Fit Model

Done
K-Means with Sparsification

Classify Image Blocks (initially use VP8 modes)
Fit Model (linear prediction, least squares)

For \( k = 1 \) to \( N \)

Reclassify Image Blocks (use Daala modes)

If \( k > M \)

Drop prediction coefficients (use prediction gain, \( P_g \))

End

Fit Model

Done

Each image processed independently
Online update of covariance matrix

Compute \( \Delta P_g \) per drop coeff.
Incremental update of table
K-Means with Sparsification

Classify Image Blocks (initially use VP8 modes)

Fit Model (linear prediction, least squares)

For $k = 1$ to $N$

Reclassify Image Blocks (use Daala modes)

If $k > M$

Drop prediction coefficients (use prediction gain, $P_g$)

End

Fit Model

Done

Each image processed independently
Online update of covariance matrix

Linear model independent for each coefficient

Compute $\Delta P_g$ per drop coeff.
Incremental update of table
K-Means with Sparsification

Classify Image Blocks (initially use VP8 modes)

Fit Model (linear prediction, least squares)

For $k = 1$ to $N$

Reclassify Image Blocks (use Daala modes)

If $k > M$

Drop prediction coefficients (use prediction gain, $Pg$)

End

Fit Model

Done

Highly Parallelizable

Each image processed independently
Online update of covariance matrix

Linear model independent for each coefficient

Compute $\Delta Pg$ per drop coeff.
Incremental update of table
## Sparse Results

<table>
<thead>
<tr>
<th>Subset</th>
<th>Size</th>
<th>VP8 Cg Ref</th>
<th>VP8 Cg Res</th>
<th>VP8 Pg</th>
<th>Daala Cg Ref</th>
<th>Daala Cg Res</th>
<th>Daala Pg</th>
</tr>
</thead>
<tbody>
<tr>
<td>subset1</td>
<td>4x8</td>
<td>11.4386</td>
<td>14.2018</td>
<td>2.76327</td>
<td>12.3286</td>
<td>14.4887</td>
<td>2.16006</td>
</tr>
<tr>
<td>subset3</td>
<td>4x8</td>
<td>13.8511</td>
<td>16.7665</td>
<td>2.9154</td>
<td>14.96</td>
<td>17.0399</td>
<td>2.07994</td>
</tr>
<tr>
<td>subset1</td>
<td>8x16</td>
<td>12.4887</td>
<td>13.3316</td>
<td>0.84291</td>
<td>12.8817</td>
<td>13.6266</td>
<td>0.74487</td>
</tr>
<tr>
<td>subset3</td>
<td>8x16</td>
<td>15.1202</td>
<td>15.9894</td>
<td>0.8694</td>
<td>15.6468</td>
<td>16.2364</td>
<td>0.58957</td>
</tr>
<tr>
<td>subset1</td>
<td>16x32</td>
<td>12.9038</td>
<td>13.0725</td>
<td>0.16871</td>
<td>13.1332</td>
<td>13.2572</td>
<td>0.124</td>
</tr>
<tr>
<td>subset3</td>
<td>16x32</td>
<td>15.587</td>
<td>15.7818</td>
<td>0.19483</td>
<td>15.8721</td>
<td>15.9675</td>
<td>0.09541</td>
</tr>
</tbody>
</table>
Texture Predictors

- TrueMotion mode now predicts texture!
Ideas for larger block sizes

- Predict using HEVC modes
- Use TF to reduce dimensionality of training support
- Generate textured modes for initial classification
Questions?