Implementation and analysis of Directional DCT in H.264

EE 5359 Multimedia Processing
Guidance: Dr K R Rao

Priyadarshini Anjanappa
UTA ID: 1000730236
priyadarshini.anjanappa@mavs.uta.edu
Introduction

- A popular scenario in image blocks is the occurrence of directional edges.

- By recognizing such characteristics, the video coding standard H.264/advanced video coding (AVC) [2] has developed a number of directional predictions in the coding of all intra blocks - called intra predictions.

- But it is still the conventional discrete cosine transform (DCT) [3] that is used after each intra prediction.

- The transform used by H.264/AVC to process both intra and inter prediction residuals is related to an integer 2-D DCT, implemented using horizontal and vertical transforms.

- It has been found that the coding efficiency can be improved by using directional transforms [1][6], since the residuals often contain textures that exhibit directional features.
Conventional DCT [3]

The 2D DCT of a square or a rectangular block is used for almost all block-based transform schemes for image and video coding.

Forward 2D DCT (NXM)

\[ X^{C2}(k,l) = \frac{4}{MN} C(k) C(l) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n,m) \cos \left(\frac{(2m+1)l\pi}{2M}\right) \cos \left(\frac{(2n+1)k\pi}{2N}\right) \]

\[
\begin{align*}
  k &= 0,1, \ldots, N-1 & C(p) &= \frac{1}{\sqrt{2}}, p = 0 \\
  l &= 0,1, \ldots, M-1 & C(p) &= 1, \ p \neq 0 \\
\end{align*}
\]

\[ x(n,m) = \text{samples in the 2D data domain} \]
\[ X^{C2}(k,l) = \text{coefficients in the 2D DCT domain} \]

Inverse 2D DCT (NXM)

\[ x(n,m) = \sum_{k=0}^{N-1} C(k) \left( \sum_{l=0}^{M-1} C(l) X^{C2}(k,l) \cos \left(\frac{(2m+1)l\pi}{2M}\right) \right) \cos \left(\frac{(2n+1)k\pi}{2N}\right) \]

\[
\begin{align*}
  n &= 0,1, \ldots, N-1 \\
  m &= 0,1, \ldots, M-1 \\
\end{align*}
\]
The conventional 2D DCT is implemented separately through two 1D transforms, one along the vertical direction and the other along the horizontal direction, as shown in Fig. 1.

These two processes can be interchanged, as the 2D DCT is a separable transform.

Fig. 1. 2D DCT implementation: A combination of 1D DCTs along horizontal and vertical directions.

The conventional DCT seems to be the best choice for image blocks in which vertical and/or horizontal edges are dominating.
H.264 Encoder

Fig. 2. Basic coding structure for H.264/AVC for a macroblock [2]
H.264 Profiles

Fig. 3. Illustration of H.264 profiles [14]
Intracoding in H.264 [8]

- Intra coding predicts the image content based on the values of previously decoded pixels. It has
  - 9 prediction modes for 4x4 blocks
  - 9 prediction modes for 8x8 blocks
  - 4 prediction modes for 16x16 blocks.

- For each intra prediction mode, an intra prediction algorithm is used to predict the image content in the current block based on decoded neighbors.

- The intra prediction errors are transformed using an integer DCT.

- An additional 2x2 Hadamard transform is applied to the four DC coefficients of each chroma component.

- If a macroblock is coded in intra- 16x16 mode, a 4x4 Hadamard transform is performed for the 4x4 DC coefficients of the luma signal as shown in Fig. 5a.[2].
**16x16 luma intra prediction modes**

- **Mode 0 (vertical):** extrapolation from upper samples (H).
- **Mode 1 (horizontal):** extrapolation from left samples (V).
- **Mode 2 (DC):** mean of upper and left-hand samples (H+V).
- **Mode 3 (Plane):** a linear “plane” function is fitted to the upper and left-hand samples H and V.

  - This works well in areas of smoothly-varying luminance.

![Fig. 4. 16x16 luma intra prediction modes [5]](image)
Intra prediction modes in H.264

Fig. 5. 4x4 luma intra prediction modes [5]

A-H -> they are the previously coded pixels of the upper macroblock and are available both at encoder/decoder.

I-L -> they are the previously coded pixels of the left macroblock and are available both at encoder/decoder.

M -> it is the previously coded pixel of the upper left macroblock and is available both at encoder/decoder.
Fig. 5a. 4x4 DC coefficients for intra 16x16 mode
Directional DCT (DDCT)

- A new DDCT framework [1] has been developed which provides a remarkable coding gain as compared to the conventional DCT.

- In H.264, the intra prediction errors are transformed using an integer DCT.

- In the new framework, DDCT is used to replace the AVC transforms by taking into account the intra prediction mode of the current block.
Fig. 6. Six directional modes in DDCT defined in a similar way as in H.264 for the block size 8x8. [1] (The vertical and horizontal modes are not included here)
DDCT implementation

- DDCT provides 9 transforms for 4x4, 9 transforms for 8x8, and 4 transforms for 16x16 [4][5][8].
- For each intra prediction mode, DDCT consists of two stages:

**Stage 1 – along the prediction direction:**

- Pixels that align along the prediction direction are grouped together and 1-D DCT is applied.
- In cases of prediction modes that are neither horizontal nor vertical, the DCT transforms used are of different sizes.

![Diagram](image)

Fig. 7. NXN image block in which 1D DCT is applied along diagonal down left direction (mode 3) [1]
Stage 2 – across the prediction direction:

Another stage of 1-D DCT is applied to the transform coefficients resulted in the first stage. Again, the DCTs may be of different sizes.

Fig. 8. Arrangement of coefficients after the first 1-D DCT, followed by rearrangement of coefficients after the second DCT as well as the modified zig zag scanning [1]
Fig. 9. Implementation of mode 3 DDCT

**STEP 1**

(a,b,..., p) = 4x4 block of pixels in the 2D spatial domain

**STEP 2**

(A, B, ..., P) = 1D DCT coefficients

DCTs are of lengths L= 1, 2, 3, 4, 2 and 1 along the dominant direction in mode 3

**STEP 3**

Rearrangement of coefficients after the first 1D DCT

**STEP 4**

(A, B, ..., P) = 1D DCT coefficients

Horizontal DCTs of lengths L= 7, 5, 3, 2 and 1

**STEP 5**

Rearrangement of coefficients after the second 1D DCT for zig zag scanning
After step 3 in Fig. 9, for each basis image, repeat step 4 in Fig. 9 by replacing the corresponding coefficient with 1 and the remaining coefficients with 0.

For a 4x4 block, 16 basis images are obtained.

The same procedure is applied for all the other DDCT modes.

Fig. 10. Procedure to obtain basis images
Fig. 11. Basis Images for mode 3 DDCT, 4x4 block

Fig. 12. Basis images for mode 0/1 DDCT, mode 3 DDCT and mode 5 DDCT [1]
Fig. 13. Getting mode 4 from mode 3 [8]
Fig. 14. Getting mode 6 from mode 5 [8]
Fig. 15. Getting mode 7 from mode 5 [8]
Fig. 16. Getting mode 8 from mode 5 [8]
Compression efficiency of DDCT [8]

To make the transform sizes more balanced, the DDCTs group pixels in the corners together in order to use DCT of longer size, hence more efficient in terms of compression.

Properties of DDCT [8]

- **Adaptivity**

  DDCT assigns a different transform and scanning pattern to each intra prediction mode unlike the integer DCT.

- **Directionality**

  By first applying the transform along the prediction direction, DDCT has the potential to minimize the artifacts around the object boundaries.
Symmetry

Although there are 22 DDCTs for 22 intra prediction modes (9 modes for 4x4, 9 modes for 8x8, and 4 modes for 16x16), these transforms can be derived, using simple operators such as rotation and/or reflection, from only 7 different core transforms:

- 16x16: one transform for all 16x16 modes

- 8x8 and 4x4:
  - Modes 0, 1: The same transform similar to AVC, DCT is used, first horizontally, then vertically

  - Modes 3 and 4: The DDCT for mode 4 can be obtained from the transform for mode 3 using a reflection on the vertical line at the center of the block

  - Modes 5 to 8: The DDCT for modes 6-8 can be derived from mode 5 using reflection and rotation.
OBJECTIVE

- The objective of the project is to implement DDCT in place of Integer DCT in the transform block of the encoder in the H.264/AVC Reference Software JM17.2 [7].

- A single intra prediction frame will be considered for the DDCT implementation.

- The coding will be done using Microsoft Visual Studio 2008.

- Coding simulations will be performed on various sets of test images and also on video formats like QCIF (Quarter Common Intermediate Format).

- The coding performance, with different quality assessment metrics like MSE, PSNR, SSIM and MS-SSIM will be observed.

- This project considers the main profile in H.264/AVC.
ENCODER CONFIGURATION:

FramesToBeEncoded = 1  # Number of frames to be coded
ProfileIDC = 77  # Profile IDC
(66=baseline, 77=main, 88=extended; FREXT Profiles: 100=High, 110=High 10, 122=High 4:2:2, 244=High 4:4:4, 44=CAVLC 4:4:4 Intra, 118=Multiview High Profile, 128=Stereo High Profile)

IntraProfile = 0  # Activate Intra Profile for FRExt (0: false, 1: true)
# (e.g. ProfileIDC=110, IntraProfile=1  =>  High 10 Intra Profile)

Transform8x8Mode = 0
# (0: only 4x4 transform, 1: allow using 8x8 transform additionally, 2: only 8x8 transform)

Input YUV file: foreman_part_qcif.yuv
Output H.264 bitstream: test.264
Output YUV file: test_rec.yuv
YUV format: YUV 4:2:0
Frames to be encoded: 1
Frequency used for encoded bitstream: 30.00 fps

DistortionSSIM = 1  # Compute SSIM distortion.
(0: disabled/default, 1: enabled)
DistortionMS_SSIM = 1  # Compute Multiscale SSIM distortion.
(0: disabled/default, 1: enabled)

Fig. 17. QCIF sequence used
Fig. 18. MSE versus bitrate for integer DCT in H.264
Fig. 19. Y-PSNR versus bitrate for integer DCT in H.264
Fig. 20. SSIM vs bitrate for integer DCT in H.264
Fig. 21. MS-SSIM vs bitrate for integer DCT in H.264
<table>
<thead>
<tr>
<th>QP (I frame)</th>
<th>Y PSNR (dB)</th>
<th>Y MSE</th>
<th>Y SSIM</th>
<th>Y MS-SSIM</th>
<th>Total encoding time (s)</th>
<th>Bit rate (kbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>79.159</td>
<td>0.00079</td>
<td>1</td>
<td>1</td>
<td>10.876</td>
<td>5590.56</td>
</tr>
<tr>
<td>4</td>
<td>68.825</td>
<td>0.00852</td>
<td>1</td>
<td>1</td>
<td>10.032</td>
<td>5232.48</td>
</tr>
<tr>
<td>8</td>
<td>57.204</td>
<td>0.12378</td>
<td>0.9995</td>
<td>0.9999</td>
<td>9.183</td>
<td>3891.84</td>
</tr>
<tr>
<td>12</td>
<td>52.28</td>
<td>0.38463</td>
<td>0.9983</td>
<td>0.9997</td>
<td>8.23</td>
<td>2838.72</td>
</tr>
<tr>
<td>16</td>
<td>48.86</td>
<td>0.84553</td>
<td>0.9965</td>
<td>0.9996</td>
<td>7.292</td>
<td>2168.16</td>
</tr>
<tr>
<td>20</td>
<td>45.131</td>
<td>1.99499</td>
<td>0.9921</td>
<td>0.9992</td>
<td>6.433</td>
<td>1560.72</td>
</tr>
<tr>
<td>24</td>
<td>41.803</td>
<td>4.29364</td>
<td>0.9846</td>
<td>0.9985</td>
<td>5.666</td>
<td>1088.88</td>
</tr>
<tr>
<td>28</td>
<td>38.892</td>
<td>8.39157</td>
<td>0.9735</td>
<td>0.9973</td>
<td>4.968</td>
<td>745.68</td>
</tr>
<tr>
<td>32</td>
<td>35.862</td>
<td>16.8695</td>
<td>0.9566</td>
<td>0.9953</td>
<td>4.391</td>
<td>496.8</td>
</tr>
<tr>
<td>36</td>
<td>33.173</td>
<td>31.32035</td>
<td>0.9342</td>
<td>0.9919</td>
<td>4.067</td>
<td>331.92</td>
</tr>
<tr>
<td>40</td>
<td>30.279</td>
<td>60.97929</td>
<td>0.8955</td>
<td>0.984</td>
<td>3.731</td>
<td>221.76</td>
</tr>
<tr>
<td>44</td>
<td>27.981</td>
<td>103.5018</td>
<td>0.8388</td>
<td>0.9705</td>
<td>3.484</td>
<td>152.64</td>
</tr>
<tr>
<td>48</td>
<td>25.293</td>
<td>192.227</td>
<td>0.7532</td>
<td>0.935</td>
<td>3.321</td>
<td>104.88</td>
</tr>
<tr>
<td>51</td>
<td>23.335</td>
<td>301.693</td>
<td>0.6703</td>
<td>0.8967</td>
<td>3.073</td>
<td>72</td>
</tr>
</tbody>
</table>

Table 1. Variation of image metrics for Y component (Foreman QCIF sequence) in Integer DCT implementation of H.264
Prediction residuals and directional transforms in H.264

- A transform is used to transform not only the image intensities but also prediction residuals of image intensities.

- The prediction residuals have different spatial characteristics from image intensities [6].

- Unlike an image, the energy of the prediction residual is not uniformly distributed in the spatial domain.

- Even within a block, many pixels may have zero intensity and the energy may often be concentrated in a region of the block [15].
Fig. 22. Visual comparison between original video and prediction residual [10]
The residuals have more locally 1-D anisotropic features in horizontal/vertical and other directions as compared to the original video.

In this case, the conventional 2-D DCT may not be the best choice to de-corrleate the residual blocks.

There exists a high correlation along the anisotropic edges in the prediction residual. This spatial redundancy can be better exploited by use of directional DCT.

A 2-D DDCT [1] can improve the coding gain for original images; however this may not be the optimal solution for 1-D structures in the residuals.

For prediction residuals, 1-D DDCT along the direction with high correlated pixels can perform better energy compaction than 2D-DDCT.

If 1D-DDCT used in neighboring blocks are with the same size and direction mode, the spatial redundancy can be further exploited by using 1D- conventional DCT for the corresponding coefficients among these blocks.

The difference between 2D-DDCT and 1D-DDCT is that after 1D DCT is applied along the dominant direction in the block, in 1D-DDCT, the second 1D-DCT is not applied along the rows in the rearranged pattern, since the correlation of residuals in such a direction is relatively low.
Also, after 1D-DDCT instead of zigzag scanning, a different scanning pattern can be used as shown in Fig. 23.

Fig. 23. Comparison of scanning orders of DDCT coefficients: (a) mode 4 DDCT (b) scanning order after 2-D DDCT (c) scanning order after 1-D DDCT [10].
Mode Selection

- Selection of the best intra prediction mode can be done by using sum of absolute differences (SAD) or mean absolute differences (MAD).

- Another method used [10] is Gabor filtering.

- Before performing the DDCTs, spatial Gabor filters with different frequencies and orientation of 1-D structures in each inter-block.

- Fig. 24 denotes the filter kernels with eight directions:
  
  $0, \pi/8, \pi/4, 3\pi/8, \pi/2, 5\pi/8, 3\pi/4, 7\pi/8$.

- By using such filters, each residual pixel has an output response with respect to the filter kernel, the largest response value will be chosen as the representative orientation, the number of pixels in each orientation is calculated.

- The orientation with the largest number of pixels is considered as the statistically major direction and the mode corresponding to the dominant direction is thus selected for directional transform [10].
Fig. 24. Spatial Gabor filter kernels with eight directions [10]
FUTURE WORK

- To implement a suitable method for mode decision of the intra prediction residuals based on the various approaches investigated.

- To integrate DDCT into H.264/AVC Reference Software.

- To compare image quality metrics MSE, PSNR, SSIM and MS-SSIM for DDCT with those of integer DCT in H.264.

- To compare performances of 1-D DDCT and 2-D DCCT.
REFERENCES


Website: [http://iphome.hhi.de/suehring/tml/download](http://iphome.hhi.de/suehring/tml/download)

[8] Intra coding with directional DCT and directional DWT, Document: JCTVC-B107_r1  


Website: [http://web.eng.fiu.edu/fanj/pdf/J5_i-manager09h264_camera.pdf](http://web.eng.fiu.edu/fanj/pdf/J5_i-manager09h264_camera.pdf)
Website: http://web.eng.fiu.edu/fanj/pdf/J8_i-mgr10architecture_camera.pdf

Website: http://iphome.hhi.de/wiegand/assets/pdfs/h264-AVC-Standard.pdf


Thank you !