Implementation and Analysis of Directional Discrete Cosine Transform in Baseline Profile in H.264

INSTRUCTOR: DR. K. R. RAO
List of acronyms

AIC advanced image coding
AVC advanced video coding
BMP bit map format
CABAC context adaptive binary arithmetic coding
DCT discrete cosine transform
EBCOT embedded block coding with optimized truncation
FRExt fidelity range extensions
HD-photo high-definition photo
I-frame intra frame
IP intra prediction
JM joint model
MSE mean square error
PSNR peak signal to noise ratio
SSIM structural similarity index metric
VLC variable length coding
LIST OF FIGURES

Figure

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

2: Different prediction modes used for prediction in AIC and H.264

3: The specific coding parts of the profiles in H.264

4: Basic coding structure for H.264/AVC encoder for a macroblock

5: Six directional modes in DDCT defined in a similar way as in H.264 for the block size 8x8.

6: D-DCT image coding
Implementation and Analysis of Directional Discrete Cosine Transform in Baseline Profile in H.264

Abstract:

Nearly all block-based transform schemes for image and video coding developed so far choose the 2-D discrete cosine transform (DCT)[2] of a square block shape. With almost no exception, this conventional DCT is implemented separately through two 1-D transforms[1], one along the vertical direction and another along the horizontal direction. Developing a new block based DCT framework in which the first transform may choose to follow a direction other than the vertical or horizontal one. The coefficients produced by all directional transforms in the first step are arranged appropriately so that the second transform can be applied to the coefficients that are best aligned with each other.

Compared with the conventional DCT, the resulting directional DCT framework is able to provide a better coding performance for image blocks that contain directional edges—a popular scenario in many image signals. By choosing the best from all directional DCTs (including the conventional DCT as a special case) for each image block, the rate-distortion coding performance can be improved remarkably. Finally, a brief theoretical analysis is presented to justify why certain coding gain (over the conventional DCT) results from this directional framework.

Introduction:

H.264/AVC is a video coding standard that has a wide range of applications ranging from high-end professional camera and editing systems to low-end mobile applications. They strive to achieve maximum compression efficiency without compromising the quality of video. To this end many coding tools are defined in them. Transform coding is one among them. Transform coding represents the signal/image (that is currently in time/spatial domain in another domain (transform domain), where most of the energy in signal/image is concentrated in a fewer number of coefficients. Thus the insignificant coefficients can be discarded after transform coding to achieve compression. In images/videos, the DCT-II [3] (which represents a signal/image as the weighted sum of cosine functions with different frequencies) is primarily used for transform coding.

The 2D DCT of a square or a rectangular block is used for almost all block-based transform schemes for image and video coding. 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. The conventional DCT seems to be the best choice for image blocks in which vertical and/or horizontal edges are dominating.
Fig. 1: 2D DCT implementation: A combination of 1D DCTs along horizontal and vertical directions.

**FORWARD 2D DCT (NXM):**

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

\[
k = 0, 1, \ldots, N-1,
\]

\[
\ell = 0, 1, \ldots, M-1
\]

\[
C(p) = \begin{cases} 
\frac{1}{\sqrt{2}} & p=k, l \\
1 & p=k, l
\end{cases}
\]

\[x(n,m) = \text{Samples in the 2D data domain.}
\]

\[X(k,l) = \text{Coefficients in the 2D- DCT domain.}
\]

**INVERSE 2D DCT (NXM):**

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

\[n = 0, 1, \ldots, N-1,
\]

\[m = 0, 1, \ldots, M-1
\]

\[x(n,m) = \text{Samples in the 2D data domain.}
\]

\[X(k,l) = \text{Coefficients in the 2D- DCT domain.}
\]
H.264 [4]:

H.264 technology aims to provide good video quality at considerably low bit rates, at reasonable level of complexity while providing flexibility to wide range of applications. H.264 or MPEG-4 part 10 aims at coding video sequences at approximately half the bit rate compared to MPEG-2 at the same quality. It also aims at having significant improvements in coding efficiency using CABAC entropy coder, error robustness and network friendliness. Parameter set concept, arbitrary slice ordering, flexible macroblock structure, redundant pictures, switched predictive and switched intra pictures have contributed to error resilience / robustness of this standard. Adaptive (directional) intra prediction (Fig.2) is one of the factors which contributed to the high coding efficiency of this standard.

Different modes used for block prediction are shown in Fig.2 [4].
Fig. 2: Different prediction modes used for prediction in AIC and H.264 [4]

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 a 4x4 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 similar 4x4 transform is performed for the 4x4 DC coefficients of the luma signal. In this framework, DDCT is to replace the AVC transforms by a set of transforms taking into account the prediction mode of the current block. Hence, DDCT provides 9 transforms for 4x4, 9 transforms for 8x8, and 4 transforms for 16x16, although many of them are the same or can be simply inferred from a core transform. For each transform, the DDCT also provides a fixed scanning pattern based on the quantization parameter (QP) and the intra prediction mode to replace the zigzag scanning pattern of DCT coefficients in AVC.
Each profile specifies a subset of entire bitstream of syntax and limits that shall be supported by all decoders conforming to that profile. There are three profiles in the first version: baseline, main, and extended.

- **Baseline profile** is primarily for low-cost applications that require additional data loss robustness, this profile is used in some videoconferencing and mobile applications. This profile includes all features that are supported in the Constrained Baseline Profile, plus three additional features that can be used for loss robustness (or for other purposes such as low-delay multi-point video stream compositing).

- **Main profile** is designed for digital storage media and television broadcasting. H.264 main profile which is the subset of high profile was designed with compression coding efficiency as its main target.

- **High profile** is the primary profile for broadcast and disc storage applications, particularly for high-definition television applications (for example, this is the profile adopted by the Blu-ray Disc storage format and the DVB HDTV broadcast service). Fidelity range extensions provide a major breakthrough with regard to compression efficiency. The profiles are shown in Fig.2.

- **Extended profile** intended as the streaming video profile, this profile has relatively high compression capability and some extra tricks for robustness to data losses and server stream switching.
Fig. 4: Basic coding structure for H.264/AVC encoder for a macroblock [2].

**Transform Block [9]:**

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. Note that, in cases of prediction modes that are neither horizontal nor vertical, the DCTs used are of different sizes.

- **Stage 2** – across the prediction direction: another stage of DCT is applied to the transform coefficients resulted in the first stage. Again, the DCTs may be of different sizes.

Six directional modes of DDCT are shown in Fig. 5. 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.
MODES OF D-DCT

Fig. 5: Six directional modes in DDCT defined in a similar way as in H.264 for the block size 8x8.

DDCT IMAGE CODING:

Step wise DDCT image coding is shown in the block diagram below.

Fig 6: D-DCT image coding
Repeat the same procedure for the other modes.

STEP 1: \(x_{00}, x_{01}, ..., x_{32}, x_{33}\) - Pixels in the 2-D spatial domain.

STEP 2: 1D-DCT is performed for the \(4\times4\) block in diagonal down-left position with lengths \(L=1, 2, 3, 4, 3, 2, 1\).

STEP 3: The coefficients of step 2 after 1D DCT are arranged vertically as shown in the figure.

STEP 4: Apply Horizontal 1D-DCT for lengths \(L=7, 5, 3\) and 1 and arranged in the same pattern.

STEP 5: After Step 4, move all 2D \((4\times4)\) Directional DCT coefficients to the left.

Implement quantization followed by 2D VLC for compression/coding along zig-zag scan.

This scanning helps to increase the runlength of zero (transform) coefficients leading to reduced bit rate in the 2D-VLC coding (similar to JPEG [12]).
REFERENCES: