Design of Low-Complexity, Non-Separable 2-D Transforms Based on Butterfly Structures

Haoming CHEN and Bing ZENG
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong SAR, China

An overview

- **Background**
  - Karhunen-Loève transform (KLT): R-D coding performance upper-bound; data-dependent, very costly to implement.
  - Discrete cosine transform (DCT): a robust approximation to KLT, very efficient implementation; a big performance drop in some cases.

- **What’s new**
  - A 2-D model for directional (image) sources
  - New transforms tightly bounded by KLT (in terms of the R-D coding performance) and DCT (in terms of the implementation cost)
  - An analytical (closed-form) solution

2-D correlation model (for images)

- **Non-directional source**: correlation between two pixels A at \((p_a, q_a)\) and B at \((p_b, q_b)\)
  \[ r_{AB}(\theta) = \rho \sqrt{\frac{p_a^2 + p_b^2}{(p_a - p_b)^2} + \frac{q_a^2 + q_b^2}{(q_a - q_b)^2}} \]

- **Directional source**: with a dominating orientation, a rotated elliptical function applies
  \[ r_{AB}(\theta) = \rho \sqrt{\frac{d_1^2 + d_2^2}{(d_1 - d_2)^2}} \]
  \[ \{d_1(\alpha) = (p_a - p_b)\cos\alpha - (q_a - q_b)\sin\alpha\} \]
  \[ \{d_2(\alpha) = (q_a - q_b)\cos\alpha + (p_a - p_b)\sin\alpha\} \]

Basics of KLT and DCT

- **KLT**: derived from the eigen-decomposition of a covariance matrix; achieves the maximum coding gain:
  \[ G_{C} = \frac{1}{N} \sum_{n=0}^{N-1} \log_{2} r_{n}^{2} \]

- **DCT**: implemented through a butterfly structure

Problem formulation

Any unitary matrix can be factorized into a product of Givens rotations \(\Omega(j, k, \theta)\). This problem can be stated as the maximization of the coding gain over all \(\Omega(j, k, \theta)\), \(l = 1, \ldots, L\). \(L\) is selected to control the implementation cost.

- **Unconstrained optimization**
  \[ \max_{j, k, \theta} G_{C} \]
  with \( C = \prod_{l=1}^{L} \Omega(j, k, \theta) \)

- **Standing-alone Givens rotation** (\(N = 2\))
  - Best rotation angle: denote the covariance matrices of \([X_0, X_1]^T\) and \([Y_0, Y_1]^T\) as \(R\) and \(\Omega\). Then \( R = \Omega \cdot \Omega^T \). Best angle can be derived by maximizing the coding gain of \( R \):
    \[ \theta = \begin{cases} \phi/2, & \text{if } (r(0,0) - r(1,1)) \cdot (r(0,1) + r(1,0)) \geq 0 \\ (\pi - \phi)/2, & \text{otherwise} \end{cases} \]
    \[ \phi = \cos^{-1} \sqrt{\frac{|r(0,0) - r(1,1)|^2}{r(0,0) + r(1,1) + r(1,0) + r(0,0)}} \]
  - Using \( \Omega(\theta) \), the coding gain is:
    \[ G_{\Omega} = \frac{1}{2} \log_{2} \left( \frac{R(0,0) \cdot R(1,1)}{r(0,0) \cdot r(1,1) - r(0,1) \cdot r(1,0)} \right) \]
    \[ = \frac{1}{2} \log_{2} \left( r(0,0) \cdot r(1,1) - r(0,1) \cdot r(1,0) \right) \]
    \[ \geq \frac{1}{2} \log_{2} (r(0,0) \cdot r(1,1)) = G_{\text{orig}} \]

- **General case** (\(N > 2\))
  - Iterative approach: in each stage \(l\), we choose a pair of two nodes in such a way that its best angle \(\theta_l\) leads to the maximum coding gain after performing the Givens rotation.
  - As the \(l\)-th Givens rotation takes place on \((j, k)\), the coding gain becomes:
    \[ G = -\frac{1}{N} \log_{2} \left( R(j, k) \cdot \Omega(j, k, \theta_l) \times \prod_{l=1}^{N-1} R(i, i) \right) \]
    \[ = -\frac{1}{N} \log_{2} \left( 1 - r(j, k) \cdot r(k, j) \right) \]

Pairing strategy algorithm (Proposed)

Given \( r = [r(i, j)]_{N \times N} \) of \( f = [x_0, x_1, \ldots, x_N] \); set \( l = 0 \); and initialize the transform matrix with \( C = I \).

1) For \( l = 1, L \), search over \( \Omega = \{0, 1, \ldots, N - 1\} \) to identify \((j, k)\) that lead to the largest ratio of \( r(j, k) \cdot r(k, j) \) w.r.t. \( r(j, j) \cdot r(k, k) \).

2) Calculate the best angle \(\theta_l\) for the selecting pair. Stop if \(\theta_l = 0\); otherwise, update the transform matrix:
    \[ C := C \cdot \Omega(j, k, \theta_l) \]

3) Calculate the covariance matrix of the output node-variables and use it to replace \( r \):
    \[ r \mapsto \Omega(j, k, \theta_l) \cdot r \cdot \Omega^T(j, k, \theta_l) \]

Go to the next iteration.

Experimental results

- **Only diagonal down (DDL) model**: without and with H.264 intra-prediction; \( L = 32 \) (as the traditional DCT for a \(4 \times 4\) block needs 32 butterfly structures):

| 4 × 4 DDL block \((a = 45^\circ, q = 5, \rho = 0.95)\) | KLT | DCT | Proposed |
|---|---|---|
| Original | Coding Gain | 2.4112 | 2.3852 | 2.8748 |
| Residual with intra-prediction | EPE | 0.8929 | 0.8570 | 0.8191 |

- **Learning curve of pairing strategy algorithm**:

[Graphs and tables are not included here.]