MAC0317 - Introduction to Digital Signal Processing

19-11-29 PE3 - Overview of lossless DCT comrpession

Professor Marcel Parolin Jackowski mjack@ime.usp.br

Undergraduate Vitor Santa Rosa Gomes, 10258862, vitorssrg@usp.br

The purpose of this last Programming Exercise is to examine a technique capable of losslessly compressing images. You already know that the cosine transform (DCT) is applied to several data compression algorithms (e.g. JPEG, MP3, etc), however this DCT encoding usually yields to data loss. Here, you will study about a technique that compresses without loss based on DCT. You should write an overview about the selected article, concisely describing i) its concept, ii) how the technique works, iii) its advantages and disadvantages, and iv) how this technique compares to more recent ones. You then shall submit a PDF file with your overview (this is an individual work) with at most two pages, which might be tailored through your favourite text editor or language (e.g. Latex, MS Word, etc).

OVERVIEW: LOSSLESS IMAGE COMPRESSION USING THE DISCRETE COSINE TRANSFORM

Synopsis

This synopsis has **emphasized text**, as well as contains **analysis and critique** reviews.

Abstract

Key ideia.

- **What. Lossless gray-scale image compression.**
- **How.**
  1. Quantize **discrete cosine transform** (DCT) coefficients.
  2. Save inverse DCT and errors.
  3. Feed an entropy encoder.
- **Why.** Comparable to current (1997) JPEG lossless formats.

Introduction

- Applications of image compression.
- DCT has energy-compaction properties.
- Some applications are **sensible to lossy compression**.
- **Comment.** How much sensitive?
- DCT is used with image compression.
- JPEG used DCT.
- There exists a **correlation between high-energy** DCT coefficients of neighbouring blocks.
- **Comment.** The introduction text feels unnatural, the relation between the sentences is not clear.

The algorithm

- Square block F of $m \times m$ from an image.
- DCT transform matrix.

$$
\begin{aligned}
C_{ij} &= \frac{1}{\sqrt{m}} i = 0 \;\; j = 0 \ldots m-1 \\
&= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)j\pi}{2m} i = 1 \ldots m-1 \;\; j = 0 \ldots m-1.
\end{aligned}
$$

- **Comment.** This math format is highly misleading.
- **Comment.** Throughout all text there is no reference to a DCT definition.
- **Define DCT transform** as $f = CFC^T$.
- **Define inverse DCT transform** as $F = C^T f C$.
- **Comment.** The notation for F and f is nonstandard.
- Finite precision causes quantization of C.
- **Comment.** There are no details or references of how the decimal (not binary!) places are quantized.
- If C has B **decimal places of precision**, f would have $2B$.
- C is not unitary, so redefine $F = C^{-1} f (C^T)^{-1}$.
- **Comment.** There are no mentions to the orthogonality of C, necessary to $C^{-1} = C^T$.
- $B > 1$ otherwise C^{-1} doesn't exist.

- **Comment.** Because $det(C) = 0$ em $B = 1$.
- F requires about $2m^3$ multiplications and $m - 1$ additions.
- **Comment.** I couldn't verify the number of additions.
- Consider the following **ranking of the coefficients**. Choose to retain only the first w, zero out the others.
- **Comment.** Seams like the RLE method that we saw in class, used in JPEG.

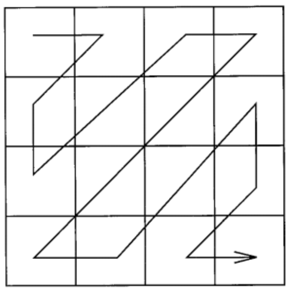
FIG. 1. Highest to lowest energy coefficients.

- Fn is the F cropped to the first w.
- **Residual error** $E = F - Fn$.
- E and Fn can reconstruct F.
- With the first w coefficients, save their **difference from the respective coefficient form the block on its left**. This decorrelates (removes) the first-order entropy.
- **Comment.** Why? This is very baffling. The reference [7] should have been mentioned here.
- Use an entropy encoder on E like Huffman or arithmetic.
- **Comment.** References? I don't know the second one.
- B has low impact on E's entropy.
- Given $m = 8$, $B = 2$ seams to achieve maximal compression.
- There seams to be $w \propto -E$.
- However larger w increases E entropy and the overall memory consumption.
- **Comment.** Why? Could have showed some graphs varying m, B and w.
- **Potential compression statistic** p:

$$p(w) = \sum_i \{\text{No. of bits needed to store } w \text{ coefficients at block } i + \quad (5)$$

$$\text{First Order entropy of } \mathbf{E} \text{ at block } i\}. \quad (6)$$

- $p(w)$ has an global minimal as w increases to m^2. The value depends on the image and on the scan [6].
- The scan form fig.1 is **optimal if the image data follows a first-order Markov process** [7].

Experiments

- Images **Cameraman**, **Baboon** and **Moon**, 256×256 gray scale 256; $m = 8$, $w = 3$ from $p(w)$, $B = 2$; compared to the seven present (1997) JPEG lossless standards.
- **Comment.** Classical choices, however the pdf versions have unusual artifacts. Though commonly used as test images, they are just too few examples for the sake of comparison. Large datasets with imagens sensitive to loss should have been benchmarked.
- Overhead of 3072, against almost nothing from JPEG.
- E and Fn have less entropy than the JPEG preprocessors.
- **Comment.** The compression ratio for the test images using standard entropy encoder is inconclusive to whether it is better than any of the JPEG standards.
- The **Rice coder** [11, 12] is adaptive and has efficient hardware implementation [12], used to compress E.
- An adaptive **Huffman coder** is used for the overhead.
- **Comment.** The compression ratio for the test images using the Rice and the Huffman coders is still inconclusive. All the methods vary strongly and non-proportionally.

Conclusion

- The method performed as average against the JPEG standards, even using a standard entropy coder.

Final comment. Not only the seven JPEG standards are still the principal ones, but also there not newer great advances on general lossless compression after 1997, regarding only the references sited on en.wikipedia.org/wiki/Lossless_JPEG. There are no comments about the resources reuired by algorithm, as well as the benchmarking is pretty limited. As the JPEG already offers on average the same compression ratio with a more studied and optimized implementation based roughly on the same core ideas, I couldn't find the advantages of the proposed method.