

Digital Video Technology

Homework #2 – 2D-DCT

2011/09/26

Write a 2D-DCT program and test it with the given 10 gray level images (lena64, pepper64, baboon64, gra1, gra2, wildcard, triangle, circle1, circle2, circle3)

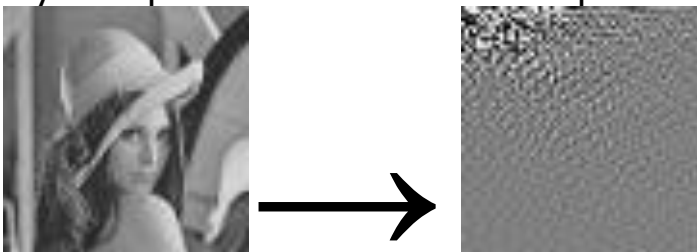
Whole frame DCT

Use **direct 2D** and **fast algorithm** (two 1D DCT) to implement 2D DCT/IDCT, and verify your results with data generated by these two algorithms. Compare and explain their evaluation time.

1. List the DCT coefficients of all images. The results need to be presented with **images** with pixel values as DCT coefficients **added by 128**. After that, remember to **saturate** the results between 0 and 255.

Note: Although the range of DCT output (block size: 64x64) is between -2^{15} and 2^{15} , just saturate it between 0 and 255, **do NOT do scaling**.

Try to explain the results. An example is shown below:



2. Apply IDCT to **reconstruct** the original images from the DCT coefficients, which are presented with **floating-point** numbers. Are they the same as the original images?

3. Use only **16 bits** to present the DC value and use **8 bits** to present the AC values. **Calculate the bits required** to present the image, and **reconstruct** the original images with the quantized coefficients.

Note:

- Just preserve to integer part. For example, if DC value is 184534.3421, you only need to preserve 184534.
- Next, 16-bits representation means you need to preserve this value with top 15 bits (because 1 bit for sign). Decide how many bits need to be truncated.
 $184534_{10} = \underline{101101000011010}110_2$.
- The number that underlined digits represent is the truncation result, which is 23066_{10} .
- When you need to reconstruct the original images with the quantized coefficients, be sure to shift 101101000011010 back to $101101000011010\underline{000}_2$.
- The 8-bits representation of AC is similar to the case of DC. But all AC should be truncated with the same bits. That means you have to find the biggest one of AC coefficients, decide how many bits need to be truncated, and apply to all other AC coefficients.
- The number of truncated bits of **DC** and **AC** needn't to be the same.
- Remember the number of truncated bits of DC and AC by your program. **No need to write out another file to record the information.**

4. When the AC values of (3) is in the range $-7 \sim 7$, set them as 0 and use 0 bit to present those coefficients. Calculate the bits required to present the image and **reconstruct** the original images with the quantized coefficients.

Block-based DCT

5. (Bonus) Use fast algorithm to implement **block-based (8x8) DCT/IDCT**. Use the requirement stated in (1) & (2) to show the results.

Requirements:

1. Deadline: 2011/10/13 (Before class)
-10 points / day
2. All the files need to be compressed as a single ZIP or RAR file.
Send this file to TA via FTP:
Address: 140.112.48.126 Port: 4299
Account (password):
The same as the one used in the course website.
Examples of filename:
DVT_HW2_R00901001.zip
DVT_HW2_R00901001_Ver2.zip
3. Required files
 - a. Report, in PDF format
(With pictures, you can use downscaled images in report)
 - b. Source code (C/C++)
Do NOT send entire project or output files to TA!! (-10%)
All TA needs are related .cpp/.c
 - c. Executable file (*.exe)
 - d. Text document (*.txt), to describe how to execute your program. Also indicate your output file names.
 - e. DCT coefficient images of 1. (named as dct1_lena64.raw, dct1_pepper64.raw, dct1_baboon64.raw, ...).
 - f. Reconstructed images of 2. (named as r2_lena64.raw, r2_pepper64.raw, r2_baboon64.raw, ...).
 - g. Reconstructed images of 3. (named as r3_lena64.raw, r3_pepper64.raw, r3_baboon64.raw, ...).
 - h. Reconstructed images of 4. (named as r4_lena64.raw, r4_pepper64.raw, r4_baboon64.raw, ...).
 - i. DCT coefficient & reconstructed images of 5. (named as dct5_lena64.raw, r5_lena64.raw, dct5_pepper64.raw, r5_pepper64.raw, ...).
4. Any further question, please email to TA.
(吳柏辰, cool77329@gmail.com)

Two-dimensional DCT & IDCT:

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

$$\text{Where } C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$0 \leq x, y, u, v \leq N - 1, \quad N^2: \text{frame size}$$

One-dimensional DCT & IDCT:

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N}$$

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u)F(u) \cos \frac{(2x+1)u\pi}{2N}$$

$$\text{Where } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$0 \leq x, u \leq N - 1, \quad N^2: \text{frame size}$$