

DSCM026 Обработка на изображения и разпознаване на образи

Първа домашна работа: Филтър за високи честоти

Изготвил:

Невена Ангелова

Факултетен номер:

f119176

Специалност:

Софтуерни технологии в интернет

1. Приложение на Дискретна трансформация на Фурие при филтър за високи честоти

В проекта е реализиран филтър за високи честоти реализиран след извършена Дискретна трансформация на Фурие (DFT) върху черно-бяло изображение с помощта на библиотеката FFTW.

DFT се представя чрез формулата:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) * e^{(-j * 2 * \pi * (u * x / M + v * y / N))}, u=0..M-1, v=0..N-1$$

където $f(x, y)$ представлява дигитална снимка с M редове и N колони, а u и v са индекси на честота.

За центриране на честотите в изходния спектър $f(x, y)$ се заменя с $f_{centered}(x, y)$

$$f_{centered}(x, y) = f(x, y) \cdot (-1)^{(x+y)} = f(x, y) \cdot e^{(j \pi (x+y))}$$

Така нулевата честота се премества в центъра.

След преобразуване на изображението в честотната област се прилага филтър. Вземат се централните координати и се увеличават с 10 %. Така се определят границите след които честотите се нулират.

Накрая се прилага Обратна трансформация на Фурие:

$$f(x, y) = 1/(MN) \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{(j2\pi i (ux/M + vy/N))}, x=0..M-1, y=0..N-1$$

Генерира се изображение, като стойностите се нормализират и се отменя центрирането.

2. Алгоритъм

2.1 main функция

Стъпки:

- зареждане на изображение image
- прилагане на DFT: `fft2d(image, trs);`
- прилагане на low pass filter върху trs, със запазване на 10%
- прилагане на inverse DFT
- показване на изображението

2.2 fft2d void функция

Тук е реализиран DFT алгоритъм

вход: двумерно grayscale изображение `image[m][n]`

изход: двуизмерен комплексен масив `trs[m][n]`

Стъпки:

- създаване на масив `in` от комплексни числа с размер $m * n$
- създаване на масив `out` със същия размер

- за всеки ред r от 0 до $m-1$: и за всяка колона c от 0 до $n-1$:

$value = image[r][c] * (-1)^{(r+c)}$ // центриране на спектъра

$in[r*n + c] = \text{комплексно число}(value, 0)$

- създаване на FFTW план за 2D FORWARD трансформация

- изпълняване на DFT

- трансформиране на всеки елемент (r,c) в комплексно число, с което C++ може да работи:

$trs[r][c] = \text{комплексно число}(out.real, out.imag)$

- освобождаваме паметта

2.3. lowPassFilter void функцията

Реализира филтриране на високите честоти.

вход: $trs[m][n]$ – спектър на изображението

$keep_ratio$ – процент от ниските честоти

Стъпки:

- определяне на централните координати:

$half_m = m / 2$

$half_n = n / 2$

- определяне на зоната за запазване

$keep_m = half_m * keep_ratio$

$keep_n = half_n * keep_ratio$

- за всеки r , c в изображението:

ако r е извън $(half_m - keep_m)$ или $(half_m + keep_m)$

и c е извън $(half_n - keep_n)$ или $(half_n + keep_n)$

тогава:

$trs[r][c] = 0$ // зануляване на високите честоти

2.4. `inverseFFT void` функцията

вход: $trs[m][n]$ – честотен спектър

изход: възстановено изображение $result[m][n]$

Стъпки:

- създаване на масив `in` от комплексни числа m, n

- създаване на масив `out` от комплексни числа m, n

- за всеки r, c данните се преобразуват във формат подходящ за FFTW:

$in[r * n + c] = trs[r][c]$

- създаване на FFTW план за BACKWARD

- изпълняване на DFT

- за всеки pixel (r, c) :

$value = out.real / (m * n)$ // нормализация

$value *= (-1)^{(r + c)}$ // премахване на центрирането

$pixel = value$

ограничаване $pixel$ в диапазон $[0..255]$

- освобождаване на паметта

3. Извод

Високите честоти в изображенията характеризират рязкия контраст и контурите.

При филтрирането им остава фонът и по-плавните нюанси.