7-9 Monday – 309-GD2

Xử lý ảnh
INT3404 1

Giảng viên: TS. Nguyễn Thị Ngọc Diệp
Email: ngocdiep@vnu.edu.vn
Slide & code: https://github.com/chupibk/INT3404_1

Ôn lại tuần 6: Các phép toán hình thái Morphological operations

translation $(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$

erosion $A \ominus B = \{z \mid (B)_z \subseteq A\}$

$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$ reflection

$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$ **dilation**

Opening $A \circ B = (A \ominus B) \oplus B$

$= \bigcup \{(B)_z \mid (B)_z \subseteq A\}$

Closing $A \bullet B = (A \oplus B) \ominus B$

$= \left[\bigcup \{(B)_z \mid (B)_z \cap A = \varnothing\}\right]^c$

Some applications of morphology

Background extraction

Hole filling

Object detection

Connected components

Homework 2

Homework 2: Detect barcode

- Description: Detect barcode region in an image

  There're 5 images that have barcode in each image.

  The barcodes vary in size, color, orientation, and slightly, shape.

  Build a script using OpenCV and Python to detect the barcode region.
- Requirements:
  - Submit the source code that includes detecting + displaying the region in the original data

Data

Input:

Expected output:

Note: Overlapping with the groundtruth about 80% is acceptable!

Submission

- Form: https://forms.gle/ftoUvATYeEGj55KY7
- Deadline: Oct 13, 2019, 23:59 (Hanoi time)

Lịch trình

Tuần	Nội dung	Yêu cầu đối với sinh viên
1	Giới thiệu môn học Làm quen với OpenCV + Python	Cài đặt môi trường: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Phép toán điểm (Point operations) – Điều chỉnh độ tương phản – Ghép ảnh	Làm bài tập 1: điều chỉnh gamma tìm contrast hợp lý
3	Histogram - Histogram equalization - Phân loại ảnh dùng so sánh histogram	Thực hành ở nhà
4	Phép lọc trong không gian điểm ảnh (linear processing filtering) - làm mịn, làm sắc ảnh	Thực hành ở nhà Tìm hiểu thêm các phép lọc
5	Tìm cạnh (edge detection)	Thực hành ở nhà
6	Các phép toán hình thái (Erosion, Dilation, Opening, Closing) - tìm biển số	Làm bài tập 2: tìm barcode
7	Chuyển đổi không gian - miền tần số (Fourier) - Hough transform	Thực hành ở nhà
8	Phân vùng (segmentation) - depth estimation - threshold-based - watershed/grabcut	Đăng ký thực hiện bài tập lớn
9	Mô hình màu Chuyển đổi giữa các mô hình màu	Làm bài tập 3: Chuyển đổi mô hình màu và thực hiện phân vùng
10	Mô hình nhiễu -Giảm nhiễu -Khôi phục ảnh -Giảm nhiễu chu kỳ - Ước lượng hàm Degration -Hàm lọc ngược, hàm lọc Wiener	Thực hành ở nhà
11	Template matching – Image Matching	Làm bài tập 4: puzzle
12	Nén ảnh	Thực hành ở nhà
13	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
14	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
15	Tổng kết cuối kỳ	Ôn tập

Frequency domain

The lecture slides are partially given by Dr. Le Thanh Ha (HMI Lab, UET-VNU)

Frequency Spectra

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$

Frequency = rate of change

Slides: Efros

Frequency Spectra

Frequency Spectra

Frequency Spectra

Frequency Spectra

Frequency Spectra

Frequency Spectra

Frequency Spectra

$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$

1D Discrete Fourier transform

- Complex series: X_k, x_n

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0,\ldots,N-1$$

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} \qquad n = 0,\ldots,N-1$$

2D Discrete Fourier transform

- Discrete domain:

$$F(w_x, w_y) = \sum_{x=1}^{M} \sum_{y=1}^{N} f(x,y) e^{-i(w_x x + w_y y)}$$

$$f(x,y) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} F(w_x, w_y) e^{i(w_x x + w_y y)}$$

2D Fourier transform

- f – image
- F – frequency image.

For every ω from 0 to ∞ (actually $-\infty$ to ∞), $F(\omega)$ holds the amplitude A and phase f of the corresponding sinusoid

$$F(\omega) = R(\omega) + iI(\omega) \qquad A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$$

$$\varphi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$$

Which one is more informative, magnitude or phase?

Fourier analysis in images

Intensity Image

Fourier Image

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

2D Fourier transform

Left: shows the vertical and horizontal frequencies as the edges of bricks.

Right: shows that the edges generate high frequencies with high energy.

2D Fourier transform

FT of each character symbol is different from others, especially at the low frequencies.

2D Fourier transform

- 2 real images
- FTs show that:
  - The high frequencies of the left image is less than those of the right image.

2D Fourier transform

- Ideal low-pass filter:
  - Completely remove the high frequencies of the left image.
  - The right image is obtained by applying the inverse FT (IFT)

2D Fourier transform

- High pass filter
  - Remove allmost all low frequencies
  - The resulted image is too dard because high energy is condensed in low frequencies which are removed.
  - There edges left (high frequencies)

2D Fourier transform - IP

- The left image is added with horizontal wave
- There are very bright dots in frequency domain.
- The image looks good after remove the noise frequency.

Practice question

Match the spatial domain image to the Fourier magnitude image

1

2

3

4

5

A

B

C

D

E

Sampling

Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/

Subsampling by a factor of 2

Throw away every other row and column to create a 1/2 size image

Aliasing problem

- 1D example (sinewave):

Source: S. Marschner

Aliasing problem

- 1D example (sinewave):

Source: S. Marschner

Aliasing problem

- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies”
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”

Source: D. Forsyth

Sampling and aliasing

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version

Anti-aliasing

Solutions:

- Sample more often
- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it's better than aliasing
  - Apply a smoothing filter

Algorithm for downsampling by factor of 2

1. Start with image(h, w)
2. Apply low-pass filter

   im_blur = imfilter(image, fspecial('gaussian', 7, 1))
3. Sample every other pixel

   im_small = im_blur(1:2:end, 1:2:end);

With and without anti-aliasing

Sampling

Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/

Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-high frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it

Early Visual Processing: Multi-scale edge and blob filters

Properties of Fourier Transform

	Spatial Domain *(x)*	Frequency Domain *(u)*
Linearity	$c_1 f(x) + c_2 g(x)$	$c_1 F(u) + c_2 G(u)$
Convolution	$f(x) * g(x)$	$F(u)G(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Differentiation	$\frac{d^n f(x)}{dx^n}$	$(i2\pi u)^n F(u)$

Filtering in Frequency domain

Ideal Lowpass filter

a b c

FIGURE 4.39 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross section.

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$

- Cut off high frequencies specified by a distance d_0.
  - Cannot be realized by electronic component → not practical

Ideal Lowpass filter

Causes ringing effect → How to remove

a b

FIGURE 4.40 (a) Test pattern of size 688 × 688 pixels, and (b) its spectrum. The spectrum is double the image size as a result of padding, but is shown half size to fit. The circles have radii of 10, 30, 60, 160, and 460 pixels with respect to the full-size spectrum. The radii enclose 86.9, 92.8, 95.1, 97.6, and 99.4% of the padded image power, respectively.

a b c
d e f

FIGURE 4.41 (a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.40(b). The power removed by these filters was 13.1, 7.2, 4.9, 2.4, and 0.6% of the total, respectively. We used mirror padding to avoid the black borders characteristic of zero padding, as illustrated in Fig. 4.31(c).

Ideal Lowpass filter

a b c

FIGURE 4.42 (a) Frequency domain ILPF transfer function. (b) Corresponding spatial domain kernel function. (c) Intensity profile of a horizontal line through the center of (b).

Butterworth Filter

a b c

FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

$$H(u,v) = \frac{1}{1 + \left[D(u,v)/D_0\right]^{2n}}$$

Butterworth Filter

a b c d
e f g h

FIGURE 4.47 (a)–(d) Spatial representations (i.e., spatial kernels) corresponding to BLPF transfer functions of size 1000×1000 pixels, cut-off frequency of 5, and order 1, 2, 5, and 20, respectively. (e)–(h) Corresponding intensity profiles through the center of the filter functions.

Ideal lowpass filter vs Butterworth Filter

a b c
d e f

FIGURE 4.41 (a) Original image of size 688×688 pixels. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.40(b). The power removed by these filters was 13.1, 7.2, 4.9, 2.4, and 0.6% of the total, respectively. We used mirror padding to avoid the black borders characteristic of zero padding, as illustrated in Fig. 4.31(c).

a b c
d e f

FIGURE 4.46 (a) Original image of size 688×688 pixels. (b)–(f) Results of filtering using BLPFs with cutoff frequencies at the radii shown in Fig. 4.40 and $n = 2.25$. Compare with Figs. 4.41 and 4.44. We used mirror padding to avoid the black borders characteristic of zero padding.

Gaussian Filter

a b c

FIGURE 4.43 (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0.

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Ideal lowpass filter vs Guassian Lowpass Filter

a b c
d e f

FIGURE 4.41 (a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.40(b). The power removed by these filters was 13.1, 7.2, 4.9, 2.4, and 0.6% of the total, respectively. We used mirror padding to avoid the black borders characteristic of zero padding, as illustrated in Fig. 4.31(c).

a b c
d e f

FIGURE 4.44 (a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.40. Compare with Fig. 4.41. We used mirror padding to avoid the black borders characteristic of zero padding.

Gaussian Lowpass Filter

a b

FIGURE 4.49 (a) Sample text of low resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

Gaussian Lowpass filter example

a b c

FIGURE 4.49 (a) Original 785×732 image. (b) Result of filtering using a GLPF with $D_0 = 150$. (c) Result of filtering using a GLPF with $D_0 = 130$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

Highpass filter

FIGURE 4.51 Top row: Perspective plot, image, and, radial cross section of an IHPF transfer function. Middle and bottom rows: The same sequence for GHPF and BHPF transfer functions. (The thin image borders were added for clarity. They are not part of the data.)

Ideal

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Gaussian

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Butterworth

$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

Highpass filter

a b c
d e f

FIGURE 4.53 Top row: The image from Fig. 4.40(a) filtered with IHPF, GHPF, and BHPF transfer functions using $D_0 = 60$ in all cases ($n = 2$ for the BHPF). Second row: Same sequence, but using $D_0 = 160$.