ชื่อ-สกุล_____ห้อง_____รหัสนักศึกษา_____

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิชา Image Processing Laboratory

การทดลองที่ 3 : การสร้างวิดีโอจากภาพนิ่งการประมวลผลการหมุนภาพในแกนพิกเซลและการประมวลผลในแกนความถี่เพื่อกรองสัญญาณในรูปแบบต่างๆ

วัตถุประสงค์

1. เพื่อศึกษาและทดลองการหมุนภาพด้วยเทคนิคการประมาณค่า Bilinear interpolation
2. เพื่อศึกษาและทดลองฟังก์ชั่นการแปลงฟูริเยร์เพื่อย้ายแกนข้อมูลจากแกนตำแหน่งไปอยู่ในแกนความถี่
3. เพื่อศึกษาองค์ประกอบข้อมูลภาพในแกนความถี่
4. เพื่อศึกษาและทดลองเทคนิคการกรองสัญญาณตามช่วงความถี่ที่กำหนด
5. เพื่อศึกษาและทดลองเทคนิคการกรองสัญญาณตามเงื่อนไขที่กำหนด

อุปกรณ์ และเครื่องมือที่ใช้ในการทดลอง

1. โปรแกรม Python

ข้อกำหนดในการตรวจการทดลอง

1. นศ.จะได้รับการตรวจตามลำดับการ upload ผลการทดลองไปที่ facebook group ในส่วน comment ของการทดลองที่ 3 เมื่ออาจารย์ตรวจเรียบร้อย จะได้รับการเช็คส่งงานในระบบ
2. ในการตรวจให้นศ.แสดงโค๊ดและผลการทดลองที่ทำพร้อมอธิบาย
3. นศ.ทุกคนส่ง source code และ ให้ตอบคำถามท้ายการทดลองใน https://forms.gle/EABnpJ4z5UdW9LCS6
4. ให้นศ. นำภาพ figure ที่ให้แสดงทุกภาพ โพสลง facebook group พร้อมชื่อกลุ่ม รหัสนศ.และชื่อสมาชิกในกลุ่ม ถ้าการทดลองใดมีการปรับค่า ให้แสดงค่าที่เลือกใช้สำหรับผลลัพธ์ภาพนั้นๆ ด้วย ส่งภายในวันที่ 14 ตุลาคม 2562 เวลา 18.00 น

ตอนที่ 1: การทดลองการหมุนภาพด้วย Backward Mapping แบบเทคนิค Bilinear interpolation

1.1 Import Lib (cv2, numpy, skimage, scipy)

1.2 อ่านไฟล์ภาพที่เตรียมมา

1.3 ปรับภาพเป็นภาพ Grayscale โดยใช้ฟังก์ชั่น cvtColor() ใน cv2 lib

1.4 คำนวณ Forward_Mapping_Matrix ตามมุม θ ที่กำหนด

$$F\_Matrix = \begin{bmatrix} \cos\theta & -\sin\theta \\ sin\,\theta & \cos\theta \end{bmatrix}$$

1.5 คำนวณ Backward_Mapping_Matrix ตามมุม θ ที่กำหนด

$$B\_Matrix = \begin{bmatrix} \cos\theta & \sin\theta \\ -sin\,\theta & \cos\theta \end{bmatrix}$$

1.6 คำนวณตำแหน่งมุมทั้ง 4 ของภาพต้นฉบับเมื่อการหมุนภาพด้วย ด้วย F_Matrix สำหรับมุม θ เพื่อกำหนดตำแหน่งของขอบภาพผลลัพธ์

1.7 คำนวณภาพผลลัพธ์จากการหมุนภาพด้วยมุม θ โดยใช้เทคนิค Backward mapping ด้วย Bilinear interpolation

- กำหนดให้ทุกตำแหน่งในภาพผลลัพธ์มีค่าเป็น 0

  f(x',y') = 0

- คำนวณตำแหน่งในภาพผลลัพธ์ (x',y') ย้อนกลับไปที่ภาพต้นฉบับ (x,y) ด้วย B_Matrix

- เลือกเฉพาะตำแหน่ง (x,y) ที่อยู่ในกรอบของภาพต้นฉบับ

  ทำการคำนวณ Bilinear interpolation

  x0 = floor(x), y0 = floor(y)

  x1 = x0+1, y1 = y0+1

  dx = x-x0, dy = y-y0

  f(x',y') = f(x0,y0) + (f(x1,y0) - f(x0,y0))*dx +
  (f(x0,y1) – f(x0,y0))*dy +
  ((f(x0,y0) + f(x1,y1)) - (f(x1,y0)+f(x0,y1)))*dx*dy

1.8 ทำการ save ภาพผลลัพธ์ f(x',y')

1.9 ทำการสร้างภาพผลลัพธ์จากการหมุนภาพ อย่างน้อย 3 มุม θ

1.10 สร้างวิดีโอไฟล์**ภาพผลลัพธ์ทั้งหมด**ที่ได้จากข้อ 1.9 โดยใช้ฟังก์ชัน cv2.VideoWriter() และ cv2.VideoWriter_fourcc() ให้กำหนดพารามิเตอร์ความเร็วเฟรม (Frame rate: Frame per sec: fps) ให้สามารถเห็นการเปลี่ยนแปลงของภาพผลลัพธ์อย่างช้าๆ ชัดเจน และกำหนดให้แสดงการเปลี่ยนแปลงแบบ reverse playback

1.11 ให้ upload ไฟล์วิดีโอไปที่ youtube โดยตั้งชื่อไฟล์ เป็น รหัสนศ.#1-รหัสนศ.#2-Image Processing–CE KMITL.avi และ แจ้งส่งงานใน comment การทดลองที่ 3.1 พร้อมโพสลิงค์วิดีโอจาก youtube ไปที่ facebook group เพื่อส่งงาน

_____

(อาจารย์ตรวจผลการทดลอง)

ตอนที่ 2: การทดลองกรองสัญญาณความถี่สูงในภาพ

2.1 Import Lib (cv2, numpy, skimage, scipy)

2.2 อ่านไฟล์ภาพที่เตรียมมา

2.3 ปรับภาพเป็นภาพ Grayscale โดยใช้ฟังก์ชั่น cvtColor() ใน cv2 lib

2.4 คำนวณ Fourier Transform ของภาพเฉดเทาในข้อ 1.4 ด้วยฟังก์ชั่น

fft2(); ใน numpy หรือ

dft2(); ใน cv2

2.5 ทำการเลื่อนผลความถี่สูงไปกึ่งกลางภาพ ด้วยฟังก์ชั่น

fftshift();

2.6 สร้างเมทริกซ์ตัวกรองความถี่สูงแบบ Gaussian High Pass Filter ขนาดเท่ากับ power of 2 ที่ใกล้กับขนาดภาพที่สุด

$$r(u,v) = \sqrt{(u - \frac{M}{2})^2 + (v - \frac{N}{2})^2}$$

$$H_{GaussianHighPass}(u,v) = 1 - e^{-r^2(u,v)/2r_0^2}$$

M = ความสูงของภาพ และ N = ความกว้างของภาพ

r_0 = รัศมีความถี่ที่ต้องการกรอง โดยกำหนดให้เลือกค่ารัศมีจำนวนไม่น้อยกว่า 10 ค่า เรียงลำดับจากค่ามากไปค่าน้อย

2.7 ทำการกรองสัญญาณในแกนความถี่โดยทำสัมประสิทธิ์ความถี่ที่ถูกเลื่อนไปอยู่กึ่งกลางภาพในข้อ 2.5 คูณด้วยเมทริกซ์ตัวกรองที่สร้างในข้อ 2.6

$$F(u,v)H_{GaussianHighPass}(u,v)$$

2.8 ทำการแปลงกลับผลลัพธ์การกรองสัมประสิทธิ์ความถี่สูงที่ได้จากข้อ 2.7 ด้วยฟังก์ชั่น

ifft2(); หรือ idft2();

2.9 ทำการกำจัดผลลัพธ์การแปลงกลับที่ได้จากข้อ 2.8 เนื่องจาก round off error ด้วยฟังก์ชั่น

round(abs());

2.10 สร้างวิดีโอไฟล์ภาพผลลัพธ์ทั้งหมดที่ได้จากข้อ 2.9 โดยใช้ฟังก์ชัน cv2.VideoWriter() และ cv2.VideoWriter_fourcc() ให้กำหนดพารามิเตอร์ความเร็วเฟรม (Frame rate: Frame per sec: fps) ให้สามารถเห็นการเปลี่ยนแปลงของภาพผลลัพธ์อย่างช้าๆ ชัดเจน และกำหนดให้แสดงการเปลี่ยนแปลงแบบ reverse playback และเลือก cmap ที่ทำให้ได้ภาพที่สวยงาม

2.11 ให้ upload ไฟล์วิดีโอไปที่ youtube โดยตั้งชื่อไฟล์ เป็น รหัสนศ.#1-รหัสนศ.#2-Image Processing–CE KMITL.avi และ แจ้งส่งงานใน comment การทดลองที่ 3.2 พร้อมโพสลิงค์วิดีโอจาก youtube ไปที่ facebook group เพื่อส่งงาน

_____

(อาจารย์ตรวจผลการทดลอง)

Tutorial

[1] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_transforms/py_fourier_transform/py_fourier_transform.html

[2] https://scipython.com/book/chapter-6-numpy/examples/blurring-an-image-with-a-two-dimensional-fft/

[3] http://www.scipy-lectures.org/advanced/image_processing/

[4] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html

[5] https://www.scipy-lectures.org/packages/scikit-image/index.html

[6] https://mmeysenburg.github.io/image-processing/09-contours/

[7] https://www.programiz.com/python-programming/methods/built-in/complex