Computer Assigment

Code ทั้งหมดอยู่บน Github

Url: https://github.com/bongtrop/DIPhomwork

Problem 1

Object 1

Center of Mess: 85.512980479, 116.130408533

Object 2

Center of Mess: 215.044995964, 189.080306699

Object 3

Center of Mess: 95.1498206933, 280.547748705

Object 4

Center of Mess: 100.979768786, 428.010404624

Object 5

Center of Mess: 227.531382442, 391.438748739

Quantity

Obj1 is 0.301531111245

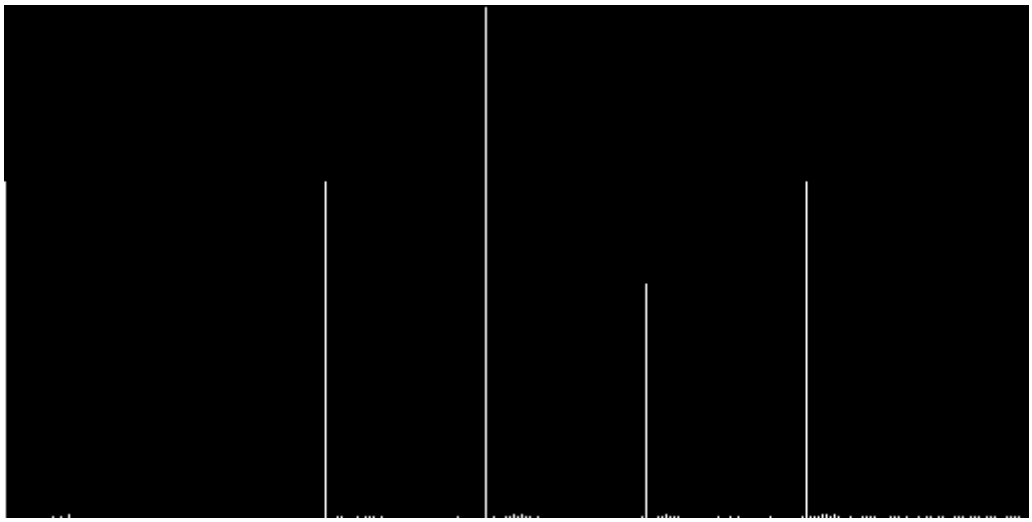
Obj2 is 0.301193318142

Obj3 is 0.288181948056

Obj4 is 0.26479854065

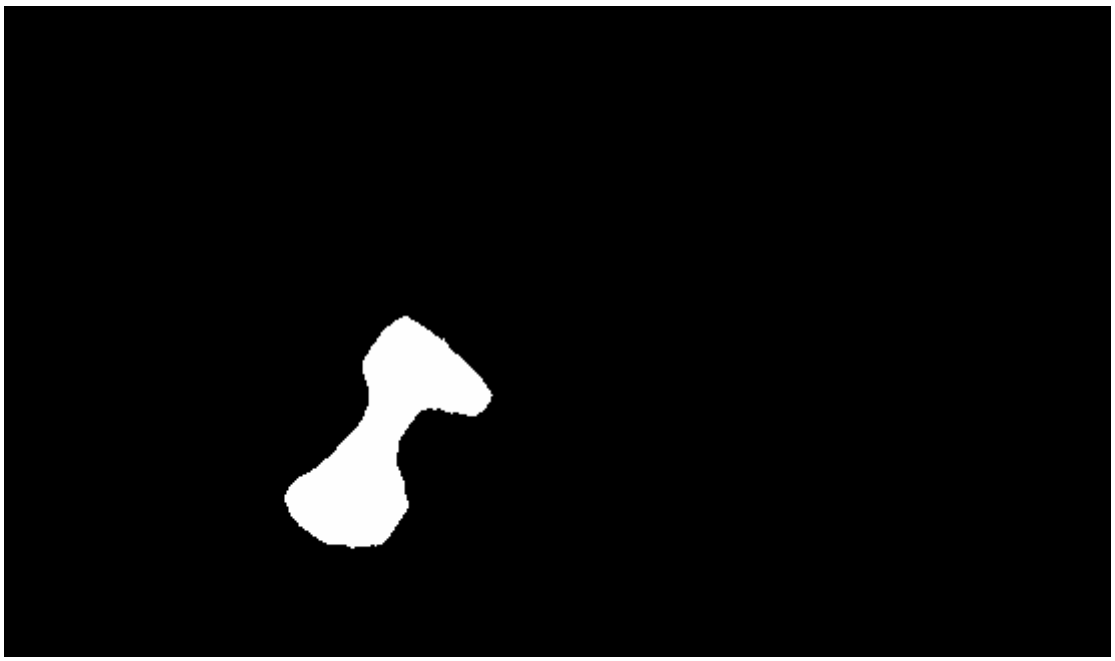
Obj5 is 0.317671394937

การที่ Quantity Obj1 Obj2 Obj5 มีค่าใกล้ๆ กันเพราะว่าขนาดของวัตถุไม่ค่อยต่างกัน หมุนอย่างเดียว



(Histogram จะเห็นว่ามีวัตถุอยู่ 5 อัน)

(Object 1)



(Object 2)

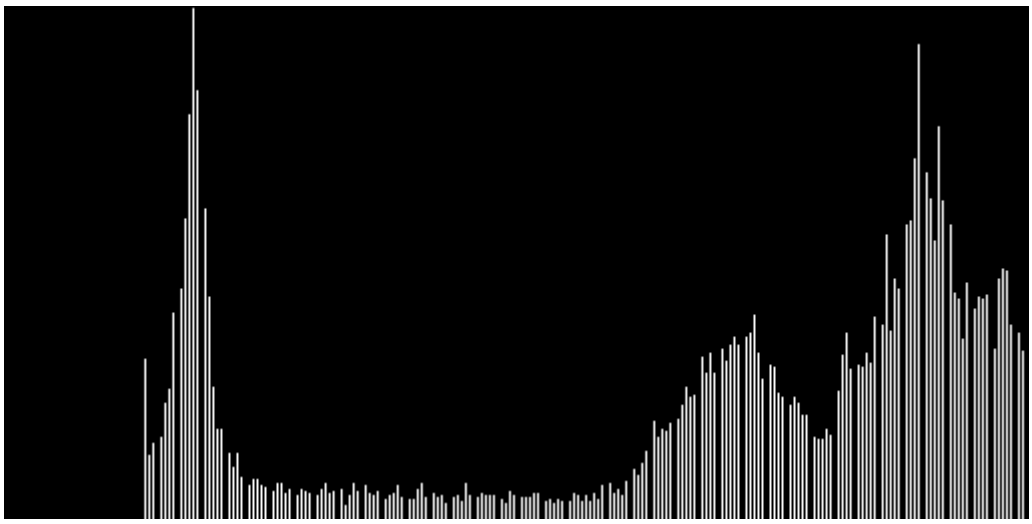
(Object 3)

(Object 4)

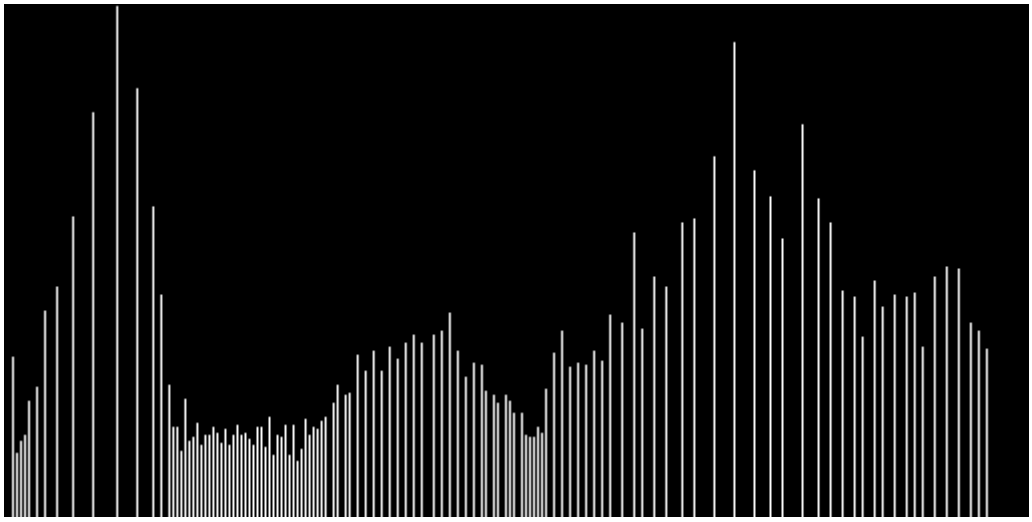
(Object 5)

Problem 2

ใช้ Histogram Equalization ในการปรับภาพ



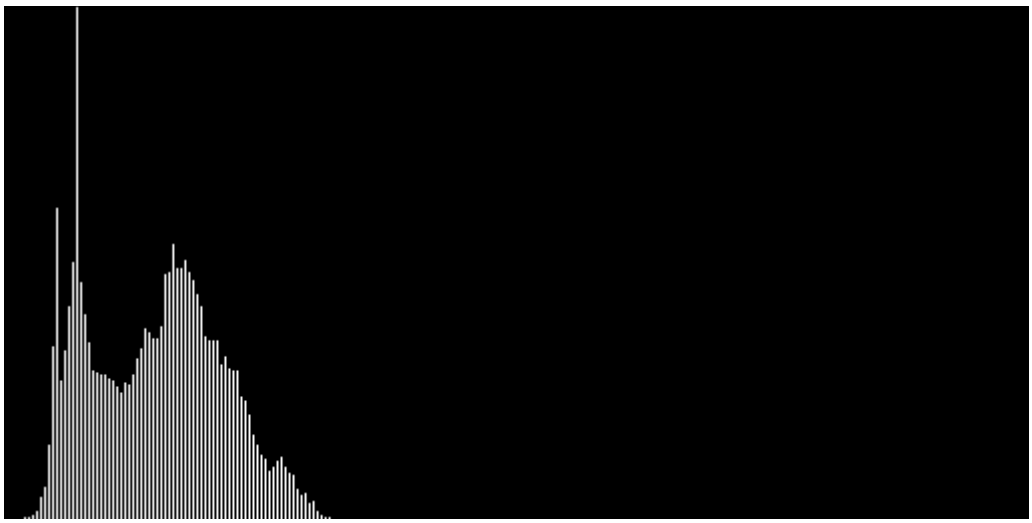
(Histogram cameraman original)



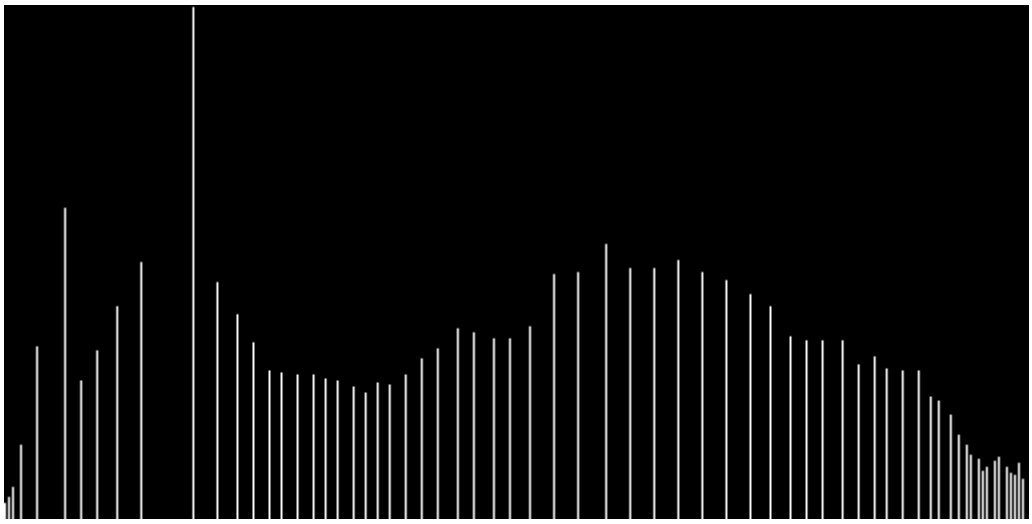
(Histogram cameraman before Histogram Equalize)

(Before)

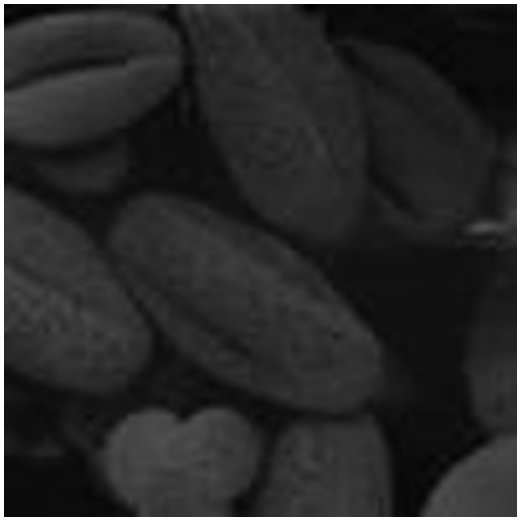
(After)



(Histogram SEM256_256 Original)



(Histogram SEM256_256 before Histogram Equalize)



(Before)

(After)

Problem 3

(Excess Green)

(Red-Blue Differrent)

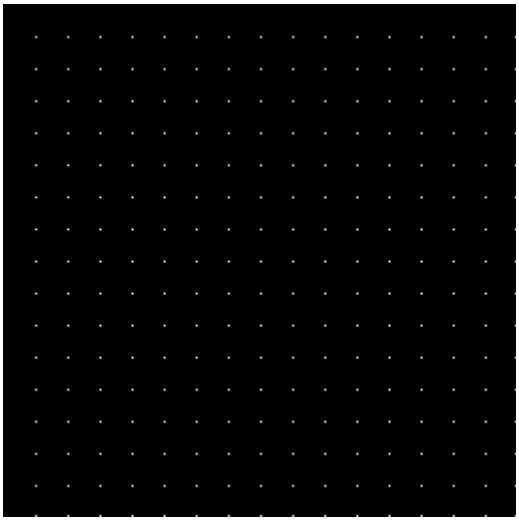
(Gray Scale)

(Excess Blue)

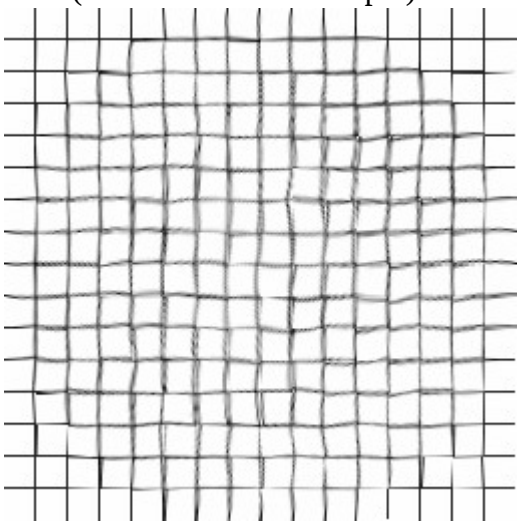
Problem 4

หามุมของ grid ใน grid.pgm ใช้ Convolute ในการหาจุด แต่ distgrid.pgm ไม่สามารถหาได้จึงไปขอ distgrid.json จากเทพต้าที่ใช้มือจิ้มเอง ทำได้ไงไม่รู้ (อดทนมาก)

แก้สมาการหา W โดยใช้ Gaussian Elimination Method จากกนั้นประมาณค่าสี โดยใช้ Bilinear



(Convolution Grid Output)



(distgrid ที่ทำการ Backward Mapping Control Grid แล้ว)

(distlenna ที่ทำการ Backward Mapping Control Grid แล้ว)

Code

ใช้ภาษา Python ใช้ Module numpy สำหรับจัดการ Matrix

Module opener (opener.py)

```
import numpy as np

# PGM File to Matrix
# Input pgm2mat(Filename)
def pgm2mat(filename):
    f = open(filename, "rb")

    # Read Detail
    i = 0
    detail = []
    while i<3:
        line = f.readline()
        line = line.replace("\n","").replace("\r","")
        if line[0]=='#':
            continue

        detail.append(line)
        i+=1

    # Read Image
    [w, h] = detail[1].split(' ')
    h = int(h)
    w = int(w)
    mat = np.zeros((h,w), dtype=np.int32)

    byte = f.read(1)
```

```
    for i in range(0,h):
        for j in range(0,w):
            mat[i][j] = ord(byte)
            byte = f.read(1)

    f.close()
    return mat

# Matrix to PGM File
# Input mat2pgm(Filename, Matrix)
def mat2pgm(filename, mat, pgmtype="P5", level="255"):
    mat[mat>255] = 255
    mat[mat<0] = 0
    f = open(filename, "wb")
    w = mat.shape[1]
    h = mat.shape[0]

    # Write Detial
    f.write(pgmtype+"\n")
    f.write(str(w)+" "+str(h)+"\n")
    f.write(level+"\n")

    # Write Image
    for i in range(0,h):
        for j in range(0,w):
            f.write(chr(mat[i][j]))

    f.close()
```

Module cal (cal.py)

```
import numpy as np
import math

# Gaussian Elimination Partial Pivoting
# Input GEPP(Ax = b)
def GEPP(A, b):
    n =  len(A)
    if b.size != n:
        raise ValueError("Invalid argument: incompatible sizes between A & b.", b.size, n)
    for k in xrange(n-1):
        maxindex = abs(A[k:,k]).argmax() + k
        if A[maxindex, k] == 0:
            raise ValueError("Matrix is singular.")
        if maxindex != k:
            A[[k,maxindex]] = A[[maxindex, k]]
            b[[k,maxindex]] = b[[maxindex, k]]
        for row in xrange(k+1, n):
            multiplier = A[row][k]/A[k][k]
            A[row][k] = multiplier
            for col in xrange(k + 1, n):
                A[row][col] = A[row][col] - multiplier*A[k][col]
            b[row] = b[row] - multiplier*b[k]
    #print A
    #print b
    x = np.zeros(n)
    k = n-1
    x[k] = b[k]/A[k,k]
    while k >= 0:
        x[k] = (b[k] - np.dot(A[k,k+1:],x[k+1:]))/A[k,k]
        k = k-1
```

```
    return x


# Bilinear Interpolate
# Input bilinear(Matrix Image, Position y, Position x)
def bilinear(mat, posy, posx):
    if posx>mat.shape[1]-1 or posy>mat.shape[0]-1:
        return mat[math.floor(posy)][math.floor(posx)]

    f00 = mat[math.floor(posy),math.floor(posx)]
    f01 = mat[math.floor(posy),math.ceil(posx)]
    f10 = mat[math.ceil(posy),math.floor(posx)]
    f11 = mat[math.ceil(posy),math.ceil(posx)]

    a = f01 - f00
    b = f10 - f00
    c = f11 + f00 - f01 - f10
    d = f00

    posx = posx-math.floor(posx)
    posy = posy-math.floor(posy)

    return a*posx + b*posy + c*posx*posy + d
```

```
Module view (view.py)
'''
This module use opencv to show image
Install opencv first
'''

import cv2
import numpy as np

def pgm(filename, name):
  im = cv2.imread(filename)
  cv2.imshow('Image '+name, im)

def mat(mat, name):
  mat[mat>255] = 255
  mat[mat<0] = 0
  mat = mat.astype(np.uint8)

  cv2.imshow('Mat '+name, mat)

def hist(hist, name):
  hist[255] = 0
  m = np.max(hist)
  x = 255.0/m
  hist = hist * x

  im = np.zeros((256,512), dtype=np.uint8)
  for i in range(0, 512, 2):
    f = int(round(hist[i/2]))
    for j in range(255, 255-f, -1):
      im[j][i] = 255
```

```
  cv2.imshow('Histogram '+name, im)


def show():
  cv2.waitKey(0)
```

Module dip (dip.py)

```
import numpy as np
import cal

# Create Histogram From Image
# Input mat2hist(Matrix Image)
def mat2hist(im):
  hist = np.zeros(256, dtype=np.uint32)
  for i in range(0,256):
      hist[i] = sum(sum(im==i))

  return hist

# Draw Histogram Image
# Input mat2hist(Matrix Histogram)
def hist2im(hist):
    hist[255] = 0
    m = np.max(hist)
    x = 255.0/m
    hist = hist * x

    im = np.zeros((256,512), dtype=np.uint8)
    for i in range(0, 512, 2):
      f = int(round(hist[i/2]))
      for j in range(255, 255-f, -1):
        im[j][i] = 255

    return im

# Change Gray Matrix to Back-White Matrix
# Input mat2bw(Matrix Image, Selection Gray Level)
```

```
def mat2bw(mat, V):
  bw = np.zeros(mat.shape, dtype=np.uint8)
  for v in V:
    bw[mat==v] = 255

  return bw

# Find Moment
# Input moment(Matrix Image, p, q)
def moment(mat, p, q):
  mat = mat.astype(np.float64)
  if (np.max(mat)>1):
    mat = mat/255

  w = mat.shape[1]
  h = mat.shape[0]
  return np.sum(np.array([[(x**p)*(y**q)*mat[y][x] for x in range(0, w)] for y in range(0,h)]))

# Find Central Moment
# Input central_moment(Matrix Image, p, q)
def central_moment(mat, p, q):
  mat = mat.astype(np.float64)
  if (np.max(mat)>1):
    mat = mat/255

  w = mat.shape[1]
  h = mat.shape[0]

  M00 = moment(mat, 0, 0)
  M10 = moment(mat, 1, 0)
  M01 = moment(mat, 0, 1)
```

```
  xc = M10/M00
  yc = M01/M00
  return np.sum(np.array([[((x-xc)**p)*((y-yc)**q)*mat[y][x] for x in range(0, w)] for y in range(0,h)]))

# Find Normalize Moment
# Input norm_moment(Matrix Image, p, q)
def norm_moment(mat, p, q):
  mat = mat.astype(np.float64)
  if (np.max(mat)>1):
    mat = mat/255

  Cpq = central_moment(mat, p, q)
  C00 = central_moment(mat, 0, 0)

  return Cpq/(C00**((p+q)/2+1))

# Find cdf from Histogram
# Input hist2cdf(Histogram Matrix)
def hist2cdf(hist):
  area = np.sum(hist)
  prop = 0
  i = 0
  cdf = np.zeros(hist.shape)

  for h in hist:
    prop = prop + (h*1.0/area)
    cdf[i] = prop
    i+=1

  return cdf
```

```
# Histogram Equalizetion
# Input histeq(Histogram Matrix)
def histeq(mat):
  out = np.zeros(mat.shape, np.int32)
  hist = mat2hist(mat)
  cdf = hist2cdf(hist)
  f = np.round(cdf*255).astype(np.uint8)

  for i in range(0,256):
    out[mat==i] = f[i]

  return out

def lpo(mat, m, c):
  mat = mat*m + c

  return mat

# Normalization Matrix by maximum value
# Input norm(Matrix)
def norm(mat):
  mat = mat.astype(np.float64)
  mat = (mat/np.max(mat))*255
  return mat.astype(np.int32)

# Convolution Image
# Input convolute(Matrix Image, Matrix Kernal, Origin)
def convolute(F, G, origin):
  G = np.fliplr(G)
  G = np.flipud(G)
```

```
  origin[0] = G.shape[0] - origin[0]
  origin[1] = G.shape[1] - origin[1]

  result = np.zeros(F.shape, dtype=np.int32)
  for i in range(origin[0], F.shape[0]-origin[0]):
    for j in range(origin[1], F.shape[1]-origin[1]):
      try:
        result[i][j] = np.sum(F[i-origin[0]:i-origin[0]+G.shape[0],j-origin[1]:j-origin[1]+G.shape[1]] *
G)
      except:
        pass

  return result

# OTSU Threshold
# Input convolute(Matrix Image)
def otsu(mat, bias=0):
  hist = mat2hist(mat)
  total = np.sum(hist)
  summ = 0
  for i in range (1,256):
    summ += i*hist[i]

  sumB = 0
  wB = 0
  WF = 0
  maxx = 0.0
  between = 0.0
  th1 = 0.0
  th2 = 0.0
```

```
  for i in range(0,256):
    wB+= hist[i]
    if wB==0:
      continue

    wF = total - wB
    if wF==0:
      break

    sumB += i*hist[i]
    mB = sumB/wB
    mF = (summ - sumB)/wF
    between = wB * wF * (mB-mF)**2
    if between>=maxx:
      th1 = i
      if between>maxx:
        th2=i

      maxx = between

  th = int((th1+th2)/2)+bias
  bw = mat2bw(mat, range(th,256))
  return bw

# Backward Mapping Control Grid
# Input controlgrid(Matrix Image, Grid, Distgrid)
def controlgrid(mat, grid, distgrid, gridsize):
    res = np.zeros(mat.shape, dtype=np.int32)
    for i in range(0, len(grid)):
        A = [[grid[i]["x1"]*1.0, grid[i]["y1"]*1.0, grid[i]["x1"]*grid[i]["y1"]*1.0, 1.],
             [grid[i]["x2"]*1.0, grid[i]["y2"]*1.0, grid[i]["x2"]*grid[i]["y2"]*1.0, 1.],
```

```
        [grid[i]["x3"]*1.0, grid[i]["y3"]*1.0, grid[i]["x3"]*grid[i]["y3"]*1.0, 1.],
        [grid[i]["x4"]*1.0, grid[i]["y4"]*1.0, grid[i]["x4"]*grid[i]["y4"]*1.0, 1.]]

    B = [distgrid[i]["x1"], distgrid[i]["x2"], distgrid[i]["x3"], distgrid[i]["x4"]]
    wx = cal.GEPP(np.array(A), np.array(B))

    B = [distgrid[i]["y1"], distgrid[i]["y2"], distgrid[i]["y3"], distgrid[i]["y4"]]
    wy = cal.GEPP(np.array(A), np.array(B))



    for j in range((i/gridsize)*gridsize, ((i/gridsize)+1)*gridsize):
        for k in range((i%gridsize)*gridsize, ((i%gridsize)+1)*gridsize):
            posx = wx[0]*k + wx[1]*j + wx[2] * j * k + wx[3]
            posy = wy[0]*k + wy[1]*j + wy[2] * j * k + wy[3]


            res[j][k] = cal.bilinear(mat, posy, posx)

return res
```

distgrid.json

[{"n":0,"x1":0,"y1":0,"x2":16,"y2":0,"x3":0,"y3":16,"x4":16,"y4":16},
{"n":1,"x1":16,"y1":0,"x2":32,"y2":0,"x3":16,"y3":16,"x4":32,"y4":16},
{"n":2,"x1":32,"y1":0,"x2":48,"y2":0,"x3":32,"y3":16,"x4":48,"y4":16},
{"n":3,"x1":48,"y1":0,"x2":64,"y2":0,"x3":48,"y3":16,"x4":64,"y4":16},
{"n":4,"x1":64,"y1":0,"x2":80,"y2":0,"x3":64,"y3":16,"x4":79,"y4":16},
{"n":5,"x1":80,"y1":0,"x2":96,"y2":0,"x3":79,"y3":16,"x4":97,"y4":17},
{"n":6,"x1":96,"y1":0,"x2":112,"y2":0,"x3":97,"y3":17,"x4":114,"y4":19},
{"n":7,"x1":112,"y1":0,"x2":128,"y2":0,"x3":114,"y3":19,"x4":130,"y4":18},
{"n":8,"x1":128,"y1":0,"x2":144,"y2":0,"x3":130,"y3":18,"x4":146,"y4":19},
{"n":9,"x1":144,"y1":0,"x2":160,"y2":0,"x3":146,"y3":19,"x4":160,"y4":18},
{"n":10,"x1":160,"y1":0,"x2":176,"y2":0,"x3":160,"y3":18,"x4":176,"y4":17},
{"n":11,"x1":176,"y1":0,"x2":192,"y2":0,"x3":176,"y3":17,"x4":192,"y4":16},
{"n":12,"x1":192,"y1":0,"x2":208,"y2":0,"x3":192,"y3":16,"x4":208,"y4":16},
{"n":13,"x1":208,"y1":0,"x2":224,"y2":0,"x3":208,"y3":16,"x4":224,"y4":16},
{"n":14,"x1":224,"y1":0,"x2":240,"y2":0,"x3":224,"y3":16,"x4":240,"y4":16},
{"n":15,"x1":240,"y1":0,"x2":255,"y2":0,"x3":240,"y3":16,"x4":255,"y4":16},
{"n":16,"x1":0,"y1":16,"x2":16,"y2":16,"x3":0,"y3":32,"x4":16,"y4":32},
{"n":17,"x1":16,"y1":16,"x2":32,"y2":16,"x3":16,"y3":32,"x4":33,"y4":32},
{"n":18,"x1":32,"y1":16,"x2":48,"y2":16,"x3":33,"y3":32,"x4":48,"y4":32},
{"n":19,"x1":48,"y1":16,"x2":64,"y2":16,"x3":48,"y3":32,"x4":67,"y4":31},
{"n":20,"x1":64,"y1":16,"x2":79,"y2":16,"x3":67,"y3":31,"x4":85,"y4":35},
{"n":21,"x1":79,"y1":16,"x2":97,"y2":17,"x3":85,"y3":35,"x4":103,"y4":37},
{"n":22,"x1":97,"y1":17,"x2":114,"y2":19,"x3":103,"y3":37,"x4":121,"y4":40},
{"n":23,"x1":114,"y1":19,"x2":130,"y2":18,"x3":121,"y3":40,"x4":136,"y4":42},
{"n":24,"x1":130,"y1":18,"x2":146,"y2":19,"x3":136,"y3":42,"x4":150,"y4":43},
{"n":25,"x1":146,"y1":19,"x2":160,"y2":18,"x3":150,"y3":43,"x4":162,"y4":41},
{"n":26,"x1":160,"y1":18,"x2":176,"y2":17,"x3":162,"y3":41,"x4":177,"y4":37},
{"n":27,"x1":176,"y1":17,"x2":192,"y2":16,"x3":177,"y3":37,"x4":192,"y4":35},
{"n":28,"x1":192,"y1":16,"x2":208,"y2":16,"x3":192,"y3":35,"x4":208,"y4":32},
{"n":29,"x1":208,"y1":16,"x2":224,"y2":16,"x3":208,"y3":32,"x4":224,"y4":32},
{"n":30,"x1":224,"y1":16,"x2":240,"y2":16,"x3":224,"y3":32,"x4":240,"y4":31},
{"n":31,"x1":240,"y1":16,"x2":255,"y2":16,"x3":240,"y3":31,"x4":255,"y4":32},
{"n":32,"x1":0,"y1":32,"x2":16,"y2":32,"x3":0,"y3":48,"x4":16,"y4":48},
{"n":33,"x1":16,"y1":32,"x2":33,"y2":32,"x3":16,"y3":48,"x4":32,"y4":48},
{"n":34,"x1":33,"y1":32,"x2":48,"y2":32,"x3":32,"y3":48,"x4":51,"y4":49},
{"n":35,"x1":48,"y1":32,"x2":67,"y2":31,"x3":51,"y3":49,"x4":72,"y4":49},
{"n":36,"x1":67,"y1":31,"x2":85,"y2":35,"x3":72,"y3":49,"x4":94,"y4":53},
{"n":37,"x1":85,"y1":35,"x2":103,"y2":37,"x3":94,"y3":53,"x4":112,"y4":56},
{"n":38,"x1":103,"y1":37,"x2":121,"y2":40,"x3":112,"y3":56,"x4":128,"y4":60},
{"n":39,"x1":121,"y1":40,"x2":136,"y2":42,"x3":128,"y3":60,"x4":141,"y4":63},
{"n":40,"x1":136,"y1":42,"x2":150,"y2":43,"x3":141,"y3":63,"x4":154,"y4":65},
{"n":41,"x1":150,"y1":43,"x2":162,"y2":41,"x3":154,"y3":65,"x4":166,"y4":65},

{"n":42,"x1":162,"y1":41,"x2":177,"y2":37,"x3":166,"y3":65,"x4":178,"y4":62},
{"n":43,"x1":177,"y1":37,"x2":192,"y2":35,"x3":178,"y3":62,"x4":192,"y4":57},
{"n":44,"x1":192,"y1":35,"x2":208,"y2":32,"x3":192,"y3":57,"x4":206,"y4":52},
{"n":45,"x1":208,"y1":32,"x2":224,"y2":32,"x3":206,"y3":52,"x4":224,"y4":48},
{"n":46,"x1":224,"y1":32,"x2":240,"y2":31,"x3":224,"y3":48,"x4":240,"y4":48},
{"n":47,"x1":240,"y1":31,"x2":255,"y2":32,"x3":240,"y3":48,"x4":255,"y4":48},
{"n":48,"x1":0,"y1":48,"x2":16,"y2":48,"x3":0,"y3":64,"x4":16,"y4":64},
{"n":49,"x1":16,"y1":48,"x2":32,"y2":48,"x3":16,"y3":64,"x4":34,"y4":64},
{"n":50,"x1":32,"y1":48,"x2":51,"y2":49,"x3":34,"y3":64,"x4":56,"y4":63},
{"n":51,"x1":51,"y1":49,"x2":72,"y2":49,"x3":56,"y3":63,"x4":80,"y4":66},
{"n":52,"x1":72,"y1":49,"x2":94,"y2":53,"x3":80,"y3":66,"x4":99,"y4":68},
{"n":53,"x1":94,"y1":53,"x2":112,"y2":56,"x3":99,"y3":68,"x4":116,"y4":72},
{"n":54,"x1":112,"y1":56,"x2":128,"y2":60,"x3":116,"y3":72,"x4":132,"y4":76},
{"n":55,"x1":128,"y1":60,"x2":141,"y2":63,"x3":132,"y3":76,"x4":144,"y4":80},
{"n":56,"x1":141,"y1":63,"x2":154,"y2":65,"x3":144,"y3":80,"x4":156,"y4":84},
{"n":57,"x1":154,"y1":65,"x2":166,"y2":65,"x3":156,"y3":84,"x4":167,"y4":85},
{"n":58,"x1":166,"y1":65,"x2":178,"y2":62,"x3":167,"y3":85,"x4":177,"y4":83},
{"n":59,"x1":178,"y1":62,"x2":192,"y2":57,"x3":177,"y3":83,"x4":190,"y4":80},
{"n":60,"x1":192,"y1":57,"x2":206,"y2":52,"x3":190,"y3":80,"x4":204,"y4":74},
{"n":61,"x1":206,"y1":52,"x2":224,"y2":48,"x3":204,"y3":74,"x4":222,"y4":66},
{"n":62,"x1":224,"y1":48,"x2":240,"y2":48,"x3":222,"y3":66,"x4":240,"y4":64},
{"n":63,"x1":240,"y1":48,"x2":255,"y2":48,"x3":240,"y3":64,"x4":255,"y4":64},
{"n":64,"x1":0,"y1":64,"x2":16,"y2":64,"x3":0,"y3":80,"x4":16,"y4":80},
{"n":65,"x1":16,"y1":64,"x2":34,"y2":64,"x3":16,"y3":80,"x4":37,"y4":78},
{"n":66,"x1":34,"y1":64,"x2":56,"y2":63,"x3":37,"y3":78,"x4":63,"y4":78},
{"n":67,"x1":56,"y1":63,"x2":80,"y2":66,"x3":63,"y3":78,"x4":84,"y4":78},
{"n":68,"x1":80,"y1":66,"x2":99,"y2":68,"x3":84,"y3":78,"x4":103,"y4":81},
{"n":69,"x1":99,"y1":68,"x2":116,"y2":72,"x3":103,"y3":81,"x4":119,"y4":85},
{"n":70,"x1":116,"y1":72,"x2":132,"y2":76,"x3":119,"y3":85,"x4":132,"y4":89},
{"n":71,"x1":132,"y1":76,"x2":144,"y2":80,"x3":132,"y3":89,"x4":144,"y4":94},
{"n":72,"x1":144,"y1":80,"x2":156,"y2":84,"x3":144,"y3":94,"x4":154,"y4":100},
{"n":73,"x1":156,"y1":84,"x2":167,"y2":85,"x3":154,"y3":100,"x4":165,"y4":103},
{"n":74,"x1":167,"y1":85,"x2":177,"y2":83,"x3":165,"y3":103,"x4":176,"y4":102},
{"n":75,"x1":177,"y1":83,"x2":190,"y2":80,"x3":176,"y3":102,"x4":188,"y4":100},
{"n":76,"x1":190,"y1":80,"x2":204,"y2":74,"x3":188,"y3":100,"x4":203,"y4":94},
{"n":77,"x1":204,"y1":74,"x2":222,"y2":66,"x3":203,"y3":94,"x4":221,"y4":85},
{"n":78,"x1":222,"y1":66,"x2":240,"y2":64,"x3":221,"y3":85,"x4":240,"y4":80},
{"n":79,"x1":240,"y1":64,"x2":255,"y2":64,"x3":240,"y3":80,"x4":255,"y4":80},
{"n":80,"x1":0,"y1":80,"x2":16,"y2":80,"x3":0,"y3":96,"x4":16,"y4":96},
{"n":81,"x1":16,"y1":80,"x2":37,"y2":78,"x3":16,"y3":96,"x4":41,"y4":93},
{"n":82,"x1":37,"y1":78,"x2":63,"y2":78,"x3":41,"y3":93,"x4":65,"y4":91},
{"n":83,"x1":63,"y1":78,"x2":84,"y2":78,"x3":65,"y3":91,"x4":86,"y4":90},
{"n":84,"x1":84,"y1":78,"x2":103,"y2":81,"x3":86,"y3":90,"x4":102,"y4":90},
{"n":85,"x1":103,"y1":81,"x2":119,"y2":85,"x3":102,"y3":90,"x4":118,"y4":96},

{"n":86,"x1":119,"y1":85,"x2":132,"y2":89,"x3":118,"y3":96,"x4":130,"y4":102},
{"n":87,"x1":132,"y1":89,"x2":144,"y2":94,"x3":130,"y3":102,"x4":141,"y4":108},
{"n":88,"x1":144,"y1":94,"x2":154,"y2":100,"x3":141,"y3":108,"x4":152,"y4":116},
{"n":89,"x1":154,"y1":100,"x2":165,"y2":103,"x3":152,"y3":116,"x4":161,"y4":117},
{"n":90,"x1":165,"y1":103,"x2":176,"y2":102,"x3":161,"y3":117,"x4":172,"y4":119},
{"n":91,"x1":176,"y1":102,"x2":188,"y2":100,"x3":172,"y3":119,"x4":184,"y4":116},
{"n":92,"x1":188,"y1":100,"x2":203,"y2":94,"x3":184,"y3":116,"x4":200,"y4":112},
{"n":93,"x1":203,"y1":94,"x2":221,"y2":85,"x3":200,"y3":112,"x4":217,"y4":105},
{"n":94,"x1":221,"y1":85,"x2":240,"y2":80,"x3":217,"y3":105,"x4":237,"y4":97},
{"n":95,"x1":240,"y1":80,"x2":255,"y2":80,"x3":237,"y3":97,"x4":255,"y4":96},
{"n":96,"x1":0,"y1":96,"x2":16,"y2":96,"x3":0,"y3":112,"x4":18,"y4":110},
{"n":97,"x1":16,"y1":96,"x2":41,"y2":93,"x3":18,"y3":110,"x4":42,"y4":106},
{"n":98,"x1":41,"y1":93,"x2":65,"y2":91,"x3":42,"y3":106,"x4":65,"y4":103},
{"n":99,"x1":65,"y1":91,"x2":86,"y2":90,"x3":65,"y3":103,"x4":84,"y4":101},
{"n":100,"x1":86,"y1":90,"x2":102,"y2":90,"x3":84,"y3":101,"x4":100,"y4":102},
{"n":101,"x1":102,"y1":90,"x2":118,"y2":96,"x3":100,"y3":102,"x4":114,"y4":105},
{"n":102,"x1":118,"y1":96,"x2":130,"y2":102,"x3":114,"y3":105,"x4":127,"y4":112},
{"n":103,"x1":130,"y1":102,"x2":141,"y2":108,"x3":127,"y3":112,"x4":136,"y4":119},
{"n":104,"x1":141,"y1":108,"x2":152,"y2":116,"x3":136,"y3":119,"x4":145,"y4":126},
{"n":105,"x1":152,"y1":116,"x2":161,"y2":117,"x3":145,"y3":126,"x4":154,"y4":130},
{"n":106,"x1":161,"y1":117,"x2":172,"y2":119,"x3":154,"y3":130,"x4":167,"y4":132},
{"n":107,"x1":172,"y1":119,"x2":184,"y2":116,"x3":167,"y3":132,"x4":180,"y4":132},
{"n":108,"x1":184,"y1":116,"x2":200,"y2":112,"x3":180,"y3":132,"x4":196,"y4":128},
{"n":109,"x1":200,"y1":112,"x2":217,"y2":105,"x3":196,"y3":128,"x4":215,"y4":122},
{"n":110,"x1":217,"y1":105,"x2":237,"y2":97,"x3":215,"y3":122,"x4":237,"y4":115},
{"n":111,"x1":237,"y1":97,"x2":255,"y2":96,"x3":237,"y3":115,"x4":255,"y4":112},
{"n":112,"x1":0,"y1":112,"x2":18,"y2":110,"x3":0,"y3":128,"x4":19,"y4":126},
{"n":113,"x1":18,"y1":110,"x2":42,"y2":106,"x3":19,"y3":126,"x4":41,"y4":120},
{"n":114,"x1":42,"y1":106,"x2":65,"y2":103,"x3":41,"y3":120,"x4":64,"y4":113},
{"n":115,"x1":65,"y1":103,"x2":84,"y2":101,"x3":64,"y3":113,"x4":81,"y4":112},
{"n":116,"x1":84,"y1":101,"x2":100,"y2":102,"x3":81,"y3":112,"x4":96,"y4":112},
{"n":117,"x1":100,"y1":102,"x2":114,"y2":105,"x3":96,"y3":112,"x4":109,"y4":115},
{"n":118,"x1":114,"y1":105,"x2":127,"y2":112,"x3":109,"y3":115,"x4":121,"y4":120},
{"n":119,"x1":127,"y1":112,"x2":136,"y2":119,"x3":121,"y3":120,"x4":129,"y4":128},
{"n":120,"x1":136,"y1":119,"x2":145,"y2":126,"x3":129,"y3":128,"x4":137,"y4":135},
{"n":121,"x1":145,"y1":126,"x2":154,"y2":130,"x3":137,"y3":135,"x4":148,"y4":141},
{"n":122,"x1":154,"y1":130,"x2":167,"y2":132,"x3":148,"y3":141,"x4":161,"y4":143},
{"n":123,"x1":167,"y1":132,"x2":180,"y2":132,"x3":161,"y3":143,"x4":174,"y4":144},
{"n":124,"x1":180,"y1":132,"x2":196,"y2":128,"x3":174,"y3":144,"x4":193,"y4":142},
{"n":125,"x1":196,"y1":128,"x2":215,"y2":122,"x3":193,"y3":142,"x4":213,"y4":137},
{"n":126,"x1":215,"y1":122,"x2":237,"y2":115,"x3":213,"y3":137,"x4":236,"y4":131},
{"n":127,"x1":237,"y1":115,"x2":255,"y2":112,"x3":236,"y3":131,"x4":255,"y4":128},
{"n":128,"x1":0,"y1":128,"x2":19,"y2":126,"x3":0,"y3":144,"x4":18,"y4":141},
{"n":129,"x1":19,"y1":126,"x2":41,"y2":120,"x3":18,"y3":141,"x4":40,"y4":135},

{"n":130,"x1":41,"y1":120,"x2":64,"y2":113,"
x3":40,"y3":135,"x4":60,"y4":129},
{"n":131,"x1":64,"y1":113,"x2":81,"y2":112,"x3":60,"y3":129,"x4":76,"y4":125},
{"n":132,"x1":81,"y1":112,"x2":96,"y2":112,"x3":76,"y3":125,"x4":90,"y4":124},
{"n":133,"x1":96,"y1":112,"x2":109,"y2":115,"x3":90,"y3":124,"x4":101,"y4":125},
{"n":134,"x1":109,"y1":115,"x2":121,"y2":120,"x3":101,"y3":125,"x4":113,"y4":129},
{"n":135,"x1":121,"y1":120,"x2":129,"y2":128,"x3":113,"y3":129,"x4":121,"y4":136},
{"n":136,"x1":129,"y1":128,"x2":137,"y2":135,"x3":121,"y3":136,"x4":131,"y4":144},
{"n":137,"x1":137,"y1":135,"x2":148,"y2":141,"x3":131,"y3":144,"x4":142,"y4":150},
{"n":138,"x1":148,"y1":141,"x2":161,"y2":143,"x3":142,"y3":150,"x4":156,"y4":154},
{"n":139,"x1":161,"y1":143,"x2":174,"y2":144,"x3":156,"y3":154,"x4":172,"y4":154},
{"n":140,"x1":174,"y1":144,"x2":193,"y2":142,"x3":172,"y3":154,"x4":190,"y4":154},
{"n":141,"x1":193,"y1":142,"x2":213,"y2":137,"x3":190,"y3":154,"x4":212,"y4":149},
{"n":142,"x1":213,"y1":137,"x2":236,"y2":131,"x3":212,"y3":149,"x4":236,"y4":145},
{"n":143,"x1":236,"y1":131,"x2":255,"y2":128,"x3":236,"y3":145,"x4":255,"y4":144},
{"n":144,"x1":0,"y1":144,"x2":18,"y2":141,"x3":0,"y3":160,"x4":17,"y4":160},
{"n":145,"x1":18,"y1":141,"x2":40,"y2":135,"x3":17,"y3":160,"x4":38,"y4":151},
{"n":146,"x1":40,"y1":135,"x2":60,"y2":129,"x3":38,"y3":151,"x4":57,"y4":144},
{"n":147,"x1":60,"y1":129,"x2":76,"y2":125,"x3":57,"y3":144,"x4":72,"y4":140},
{"n":148,"x1":76,"y1":125,"x2":90,"y2":124,"x3":72,"y3":140,"x4":85,"y4":138},
{"n":149,"x1":90,"y1":124,"x2":101,"y2":125,"x3":85,"y3":138,"x4":96,"y4":138},
{"n":150,"x1":101,"y1":125,"x2":113,"y2":129,"x3":96,"y3":138,"x4":106,"y4":141},
{"n":151,"x1":113,"y1":129,"x2":121,"y2":136,"x3":106,"y3":141,"x4":115,"y4":148},
{"n":152,"x1":121,"y1":136,"x2":131,"y2":144,"x3":115,"y3":148,"x4":126,"y4":153},
{"n":153,"x1":131,"y1":144,"x2":142,"y2":150,"x3":126,"y3":153,"x4":138,"y4":161},
{"n":154,"x1":142,"y1":150,"x2":156,"y2":154,"x3":138,"y3":161,"x4":153,"y4":165},
{"n":155,"x1":156,"y1":154,"x2":172,"y2":154,"x3":153,"y3":165,"x4":169,"y4":167},
{"n":156,"x1":172,"y1":154,"x2":190,"y2":154,"x3":169,"y3":167,"x4":190,"y4":167},
{"n":157,"x1":190,"y1":154,"x2":212,"y2":149,"x3":190,"y3":167,"x4":214,"y4":163},
{"n":158,"x1":212,"y1":149,"x2":236,"y2":145,"x3":214,"y3":163,"x4":238,"y4":161},
{"n":159,"x1":236,"y1":145,"x2":255,"y2":144,"x3":238,"y3":161,"x4":255,"y4":160},
{"n":160,"x1":0,"y1":160,"x2":17,"y2":160,"x3":0,"y3":176,"x4":16,"y4":177},
{"n":161,"x1":17,"y1":160,"x2":38,"y2":151,"x3":16,"y3":177,"x4":34,"y4":170},
{"n":162,"x1":38,"y1":151,"x2":57,"y2":144,"x3":34,"y3":170,"x4":53,"y4":162},
{"n":163,"x1":57,"y1":144,"x2":72,"y2":140,"x3":53,"y3":162,"x4":66,"y4":156},
{"n":164,"x1":72,"y1":140,"x2":85,"y2":138,"x3":66,"y3":156,"x4":81,"y4":153},
{"n":165,"x1":85,"y1":138,"x2":96,"y2":138,"x3":81,"y3":153,"x4":92,"y4":153},
{"n":166,"x1":96,"y1":138,"x2":106,"y2":141,"x3":92,"y3":153,"x4":102,"y4":156},
{"n":167,"x1":106,"y1":141,"x2":115,"y2":148,"x3":102,"y3":156,"x4":112,"y4":158},
{"n":168,"x1":115,"y1":148,"x2":126,"y2":153,"x3":112,"y3":158,"x4":124,"y4":165},
{"n":169,"x1":126,"y1":153,"x2":138,"y2":161,"x3":124,"y3":165,"x4":137,"y4":171},
{"n":170,"x1":138,"y1":161,"x2":153,"y2":165,"x3":137,"y3":171,"x4":153,"y4":174},
{"n":171,"x1":153,"y1":165,"x2":169,"y2":167,"x3":153,"y3":174,"x4":171,"y4":178},

{"n":172,"x1":169,"y1":167,"x2":190,"y2":167,"x3":171,"y3":178,"x4":192,"y4":178},
{"n":173,"x1":190,"y1":167,"x2":214,"y2":163,"x3":192,"y3":178,"x4":217,"y4":177},
{"n":174,"x1":214,"y1":163,"x2":238,"y2":161,"x3":217,"y3":177,"x4":240,"y4":176},
{"n":175,"x1":238,"y1":161,"x2":255,"y2":160,"x3":240,"y3":176,"x4":255,"y4":176},
{"n":176,"x1":0,"y1":176,"x2":16,"y2":177,"x3":0,"y3":192,"x4":17,"y4":192},
{"n":177,"x1":16,"y1":177,"x2":34,"y2":170,"x3":17,"y3":192,"x4":33,"y4":191},
{"n":178,"x1":34,"y1":170,"x2":53,"y2":162,"x3":33,"y3":191,"x4":51,"y4":182},
{"n":179,"x1":53,"y1":162,"x2":66,"y2":156,"x3":51,"y3":182,"x4":66,"y4":175},
{"n":180,"x1":66,"y1":156,"x2":81,"y2":153,"x3":66,"y3":175,"x4":78,"y4":170},
{"n":181,"x1":81,"y1":153,"x2":92,"y2":153,"x3":78,"y3":170,"x4":90,"y4":169},
{"n":182,"x1":92,"y1":153,"x2":102,"y2":156,"x3":90,"y3":169,"x4":101,"y4":172},
{"n":183,"x1":102,"y1":156,"x2":112,"y2":158,"x3":101,"y3":172,"x4":113,"y4":176},
{"n":184,"x1":112,"y1":158,"x2":124,"y2":165,"x3":113,"y3":176,"x4":124,"y4":181},
{"n":185,"x1":124,"y1":165,"x2":137,"y2":171,"x3":124,"y3":181,"x4":139,"y4":184},
{"n":186,"x1":137,"y1":171,"x2":153,"y2":174,"x3":139,"y3":184,"x4":155,"y4":188},
{"n":187,"x1":153,"y1":174,"x2":171,"y2":178,"x3":155,"y3":188,"x4":174,"y4":189},
{"n":188,"x1":171,"y1":178,"x2":192,"y2":178,"x3":174,"y3":189,"x4":198,"y4":193},
{"n":189,"x1":192,"y1":178,"x2":217,"y2":177,"x3":198,"y3":193,"x4":221,"y4":192},
{"n":190,"x1":217,"y1":177,"x2":240,"y2":176,"x3":221,"y3":192,"x4":240,"y4":192},
{"n":191,"x1":240,"y1":176,"x2":255,"y2":176,"x3":240,"y3":192,"x4":255,"y4":192},
{"n":192,"x1":0,"y1":192,"x2":17,"y2":192,"x3":0,"y3":208,"x4":16,"y4":208},
{"n":193,"x1":17,"y1":192,"x2":33,"y2":191,"x3":16,"y3":208,"x4":31,"y4":208},
{"n":194,"x1":33,"y1":191,"x2":51,"y2":182,"x3":31,"y3":208,"x4":49,"y4":204},
{"n":195,"x1":51,"y1":182,"x2":66,"y2":175,"x3":49,"y3":204,"x4":64,"y4":197},
{"n":196,"x1":66,"y1":175,"x2":78,"y2":170,"x3":64,"y3":197,"x4":80,"y4":193},
{"n":197,"x1":78,"y1":170,"x2":90,"y2":169,"x3":80,"y3":193,"x4":89,"y4":190},
{"n":198,"x1":90,"y1":169,"x2":101,"y2":172,"x3":89,"y3":190,"x4":101,"y4":190},
{"n":199,"x1":101,"y1":172,"x2":113,"y2":176,"x3":101,"y3":190,"x4":113,"y4":191},
{"n":200,"x1":113,"y1":176,"x2":124,"y2":181,"x3":113,"y3":191,"x4":128,"y4":195},
{"n":201,"x1":124,"y1":181,"x2":139,"y2":184,"x3":128,"y3":195,"x4":144,"y4":198},
{"n":202,"x1":139,"y1":184,"x2":155,"y2":188,"x3":144,"y3":198,"x4":161,"y4":203},
{"n":203,"x1":155,"y1":188,"x2":174,"y2":189,"x3":161,"y3":203,"x4":182,"y4":205},
{"n":204,"x1":174,"y1":189,"x2":198,"y2":193,"x3":182,"y3":205,"x4":204,"y4":206},
{"n":205,"x1":198,"y1":193,"x2":221,"y2":192,"x3":204,"y3":206,"x4":224,"y4":208},
{"n":206,"x1":221,"y1":192,"x2":240,"y2":192,"x3":224,"y3":208,"x4":240,"y4":208},
{"n":207,"x1":240,"y1":192,"x2":255,"y2":192,"x3":240,"y3":208,"x4":255,"y4":208},
{"n":208,"x1":0,"y1":208,"x2":16,"y2":208,"x3":0,"y3":224,"x4":16,"y4":224},
{"n":209,"x1":16,"y1":208,"x2":31,"y2":208,"x3":16,"y3":224,"x4":32,"y4":224},
{"n":210,"x1":31,"y1":208,"x2":49,"y2":204,"x3":32,"y3":224,"x4":48,"y4":223},
{"n":211,"x1":49,"y1":204,"x2":64,"y2":197,"x3":48,"y3":223,"x4":63,"y4":221},
{"n":212,"x1":64,"y1":197,"x2":80,"y2":193,"x3":63,"y3":221,"x4":80,"y4":217},
{"n":213,"x1":80,"y1":193,"x2":89,"y2":190,"x3":80,"y3":217,"x4":92,"y4":213},
{"n":214,"x1":89,"y1":190,"x2":101,"y2":190,"x3":92,"y3":213,"x4":106,"y4":212},
{"n":215,"x1":101,"y1":190,"x2":113,"y2":191,"x3":106,"y3":212,"x4":119,"y4":212},

{"n":216,"x1":113,"y1":191,"x2":128,"y2":195,"x3":119,"y3":212,"x4":133,"y4":215},
{"n":217,"x1":128,"y1":195,"x2":144,"y2":198,"x3":133,"y3":215,"x4":150,"y4":217},
{"n":218,"x1":144,"y1":198,"x2":161,"y2":203,"x3":150,"y3":217,"x4":168,"y4":220},
{"n":219,"x1":161,"y1":203,"x2":182,"y2":205,"x3":168,"y3":220,"x4":189,"y4":222},
{"n":220,"x1":182,"y1":205,"x2":204,"y2":206,"x3":189,"y3":222,"x4":208,"y4":224},
{"n":221,"x1":204,"y1":206,"x2":224,"y2":208,"x3":208,"y3":224,"x4":223,"y4":224},
{"n":222,"x1":224,"y1":208,"x2":240,"y2":208,"x3":223,"y3":224,"x4":241,"y4":224},
{"n":223,"x1":240,"y1":208,"x2":255,"y2":208,"x3":241,"y3":224,"x4":255,"y4":224},
{"n":224,"x1":0,"y1":224,"x2":16,"y2":224,"x3":0,"y3":240,"x4":16,"y4":240},
{"n":225,"x1":16,"y1":224,"x2":32,"y2":224,"x3":16,"y3":240,"x4":32,"y4":240},
{"n":226,"x1":32,"y1":224,"x2":48,"y2":223,"x3":32,"y3":240,"x4":48,"y4":240},
{"n":227,"x1":48,"y1":223,"x2":63,"y2":221,"x3":48,"y3":240,"x4":64,"y4":240},
{"n":228,"x1":63,"y1":221,"x2":80,"y2":217,"x3":64,"y3":240,"x4":80,"y4":239},
{"n":229,"x1":80,"y1":217,"x2":92,"y2":213,"x3":80,"y3":239,"x4":95,"y4":238},
{"n":230,"x1":92,"y1":213,"x2":106,"y2":212,"x3":95,"y3":238,"x4":110,"y4":237},
{"n":231,"x1":106,"y1":212,"x2":119,"y2":212,"x3":110,"y3":237,"x4":125,"y4":236},
{"n":232,"x1":119,"y1":212,"x2":133,"y2":215,"x3":125,"y3":236,"x4":142,"y4":237},
{"n":233,"x1":133,"y1":215,"x2":150,"y2":217,"x3":142,"y3":237,"x4":158,"y4":238},
{"n":234,"x1":150,"y1":217,"x2":168,"y2":220,"x3":158,"y3":238,"x4":175,"y4":239},
{"n":235,"x1":168,"y1":220,"x2":189,"y2":222,"x3":175,"y3":239,"x4":192,"y4":240},
{"n":236,"x1":189,"y1":222,"x2":208,"y2":224,"x3":192,"y3":240,"x4":208,"y4":240},
{"n":237,"x1":208,"y1":224,"x2":223,"y2":224,"x3":208,"y3":240,"x4":224,"y4":240},
{"n":238,"x1":223,"y1":224,"x2":241,"y2":224,"x3":224,"y3":240,"x4":240,"y4":240},
{"n":239,"x1":241,"y1":224,"x2":255,"y2":224,"x3":240,"y3":240,"x4":255,"y4":240},
{"n":240,"x1":0,"y1":240,"x2":16,"y2":240,"x3":0,"y3":255,"x4":16,"y4":255},
{"n":241,"x1":16,"y1":240,"x2":32,"y2":240,"x3":16,"y3":255,"x4":32,"y4":255},
{"n":242,"x1":32,"y1":240,"x2":48,"y2":240,"x3":32,"y3":255,"x4":48,"y4":255},
{"n":243,"x1":48,"y1":240,"x2":64,"y2":240,"x3":48,"y3":255,"x4":64,"y4":255},
{"n":244,"x1":64,"y1":240,"x2":80,"y2":239,"x3":64,"y3":255,"x4":80,"y4":255},
{"n":245,"x1":80,"y1":239,"x2":95,"y2":238,"x3":80,"y3":255,"x4":96,"y4":255},
{"n":246,"x1":95,"y1":238,"x2":110,"y2":237,"x3":96,"y3":255,"x4":112,"y4":255},
{"n":247,"x1":110,"y1":237,"x2":125,"y2":236,"x3":112,"y3":255,"x4":128,"y4":255},
{"n":248,"x1":125,"y1":236,"x2":142,"y2":237,"x3":128,"y3":255,"x4":144,"y4":255},
{"n":249,"x1":142,"y1":237,"x2":158,"y2":238,"x3":144,"y3":255,"x4":160,"y4":255},
{"n":250,"x1":158,"y1":238,"x2":175,"y2":239,"x3":160,"y3":255,"x4":176,"y4":255},
{"n":251,"x1":175,"y1":239,"x2":192,"y2":240,"x3":176,"y3":255,"x4":192,"y4":255},
{"n":252,"x1":192,"y1":240,"x2":208,"y2":240,"x3":192,"y3":255,"x4":208,"y4":255},
{"n":253,"x1":208,"y1":240,"x2":224,"y2":240,"x3":208,"y3":255,"x4":224,"y4":255},
{"n":254,"x1":224,"y1":240,"x2":240,"y2":240,"x3":224,"y3":255,"x4":240,"y4":255},{
"n":255,"x1":240,"y1":240,"x2":255,"y2":240,"x3":240,"y3":255,"x4":255,"y4":255}]

problem1.py

```
import dip
# import view
import cal
import opener

mat = opener.pgm2mat('dataset/scaled_shapes.pgm')
hist = dip.mat2hist(mat)
im = dip.hist2im(hist)
#view.mat(im, "histogram")
opener.mat2pgm('report/histogram_p1.pgm', im)

obj1 = dip.mat2bw(mat, [0])
opener.mat2pgm('report/obj1_p1.pgm', obj1)
obj2 = dip.mat2bw(mat, [80])
opener.mat2pgm('report/obj1_p2.pgm', obj2)
obj3 = dip.mat2bw(mat, [120])
opener.mat2pgm('report/obj1_p3.pgm', obj3)
obj4 = dip.mat2bw(mat, [160])
opener.mat2pgm('report/obj1_p4.pgm', obj4)
obj5 = dip.mat2bw(mat, [200])
opener.mat2pgm('report/obj1_p5.pgm', obj5)

'''
view.mat(obj1, "OBJ 1")
view.mat(obj2, "OBJ 2")
view.mat(obj3, "OBJ 3")
view.mat(obj4, "OBJ 4")
view.mat(obj5, "OBJ 5")
view.show()
'''
```

```
print "\nObject 1"
M00 = dip.moment(obj1, 0, 0)
M01 = dip.moment(obj1, 0, 1)
M10 = dip.moment(obj1, 1, 0)
ctm_x = M01/M00
ctm_y = M10/M00
print "Center of Mess: " + str(ctm_x) + ", " + str(ctm_y)

print "\nObject 2"
M00 = dip.moment(obj2, 0, 0)
M01 = dip.moment(obj2, 0, 1)
M10 = dip.moment(obj2, 1, 0)
ctm_x = M01/M00
ctm_y = M10/M00
print "Center of Mess: " + str(ctm_x) + ", " + str(ctm_y)

print "\nObject 3"
M00 = dip.moment(obj3, 0, 0)
M01 = dip.moment(obj3, 0, 1)
M10 = dip.moment(obj3, 1, 0)
ctm_x = M01/M00
ctm_y = M10/M00
print "Center of Mess: " + str(ctm_x) + ", " + str(ctm_y)

print "\nObject 4"
M00 = dip.moment(obj4, 0, 0)
M01 = dip.moment(obj4, 0, 1)
M10 = dip.moment(obj4, 1, 0)
ctm_x = M01/M00
ctm_y = M10/M00
```

```
print "Center of Mess: " + str(ctm_x) + ", " + str(ctm_y)


print "\nObject 5"
M00 = dip.moment(obj5, 0, 0)
M01 = dip.moment(obj5, 0, 1)
M10 = dip.moment(obj5, 1, 0)
ctm_x = M01/M00
ctm_y = M10/M00
print "Center of Mess: " + str(ctm_x) + ", " + str(ctm_y)


print "\nQuantity"
print "Obj1 is " + str(dip.norm_moment(obj1, 2, 0) + dip.norm_moment(obj1, 0, 2))
print "Obj2 is " + str(dip.norm_moment(obj2, 2, 0) + dip.norm_moment(obj2, 0, 2))
print "Obj3 is " + str(dip.norm_moment(obj3, 2, 0) + dip.norm_moment(obj3, 0, 2))
print "Obj4 is " + str(dip.norm_moment(obj4, 2, 0) + dip.norm_moment(obj4, 0, 2))
print "Obj5 is " + str(dip.norm_moment(obj5, 2, 0) + dip.norm_moment(obj5, 0, 2))
```

problem2.py

```
import dip
# import view
import cal
import opener

mat = opener.pgm2mat('dataset/Cameraman.pgm')
hist = dip.mat2hist(mat)
im = dip.hist2im(hist)
# view.mat(mat, "Cameraman Original")
opener.mat2pgm('report/cameraman_ori_p2.pgm', mat)
opener.mat2pgm('report/cameraman_ori_hist_p2.pgm', im)

nmat = dip.histeq(mat)
hist = dip.mat2hist(nmat)
im = dip.hist2im(hist)
# view.mat(nmat, "Cameraman Edited")
opener.mat2pgm('report/cameraman_edi_p2.pgm', nmat)
opener.mat2pgm('report/cameraman_edi_hist_p2.pgm', im)

mat = opener.pgm2mat('dataset/SEM256_256.pgm')
hist = dip.mat2hist(mat)
im = dip.hist2im(hist)
# view.mat(mat, "SEM256_256 Original")
opener.mat2pgm('report/SEM256_256_ori_p2.pgm', mat)
opener.mat2pgm('report/SEM256_256_ori_hist_p2.pgm', im)

nmat = dip.histeq(mat)
hist = dip.mat2hist(nmat)
im = dip.hist2im(hist)
# view.mat(nmat, "SEM256_256 Edited")
```

```
opener.mat2pgm('report/SEM256_256_edi_p2.pgm', nmat)
opener.mat2pgm('report/SEM256_256_edi_hist_p2.pgm', im)


# view.show()
```

problem3.py

```
import dip
# import view
import cal
import opener

r = opener.pgm2mat('dataset/SanFranPeak_red.pgm')
g = opener.pgm2mat('dataset/SanFranPeak_green.pgm')
b = opener.pgm2mat('dataset/SanFranPeak_blue.pgm')

excess_green = 2*g - r - b
excess_blue = 2*b - g - r
redblue_diff = r - b
gray = (r + g + b)/3

'''
view.mat(excess_green, " Excess Green")
view.mat(redblue_diff, " Red-Blue Different")
view.mat(gray, " Gray Scale")
view.mat(excess_blue, "Excess Blue")
view.show()
'''

opener.mat2pgm('report/excess_green_p3.pgm', excess_green)
opener.mat2pgm('report/redblue_diff_p3.pgm', redblue_diff)
opener.mat2pgm('report/excess_blue_p3.pgm', excess_blue)
opener.mat2pgm('report/gray_p3.pgm', gray)
```

problem4.py

```
import opener
import dip
# import view
import numpy as np
import json

mat = opener.pgm2mat('dataset/grid.pgm')

# Convolute to find cross line
F = 255 - mat
G = np.array([[0, 0, 1, 0, 0],
              [0, 0, 1, 0, 0],
              [1, 1, 1, 1, 1],
              [0, 0, 1, 0, 0],
              [0, 0, 1, 0, 0]
             ])

R = dip.convolute(F, G, [2,2])
R = dip.norm(R)

R[R<200] = 0
R[R>=200] = 255
for i in range(16,256,16):
        R[255, i] = 255
        R[i, 255] = 255

R[255,255] = 255

# view.mat(R, "Convolute Result")
opener.mat2pgm('report/convolute_p4.pgm', R)
```

```
grid_c = np.where(R==255)
grid_x = grid_c[1]
grid_y = grid_c[0]

grid = []

for i in range(0,256):
    g = {}
    g["n"] = i
    g["x1"] = grid_x[i] - 16
    g["y1"] = grid_y[i] - 16
    g["x2"] = grid_x[i]
    g["y2"] = grid_y[i] - 16
    g["x3"] = grid_x[i] - 16
    g["y3"] = grid_y[i]
    g["x4"] = grid_x[i]
    g["y4"] = grid_y[i]

    grid.append(g)

# Distgrid not automatic
json_data = open('distgrid.json')
distgrid = json.load(json_data)
json_data.close()

p = opener.pgm2mat('dataset/distgrid.pgm')
res = dip.controlgrid(p, grid, distgrid, 16)
# view.mat(res, "Result from distgrid")
opener.mat2pgm('report/distgrid_fix_p4.pgm', res)
```

```
p = opener.pgm2mat('dataset/distlenna.pgm')
res = dip.controlgrid(p, grid, distgrid, 16)
# view.mat(res, "Result from distlenna")
opener.mat2pgm('report/distlenna_fix_p4.pgm', res)
# view.show()
```