COMPUTER VISION AND IMAGE PROCESSING (CS676A)

IMAGE SEGMENTATION USING MEAN SHIFT

GROUP 6

PRAMOD CHUNDURI - 13221

KRITI JOSHI - 13358

INTRODUCTION

We have obtained the following results after implementing Mean shift algorithm for two different kernels namely “Flat kernel” and “Gaussian kernel”. Parameters tuned for both of them are as follows:

- FLAT KERNEL:
    1. Threshold
    2. Window Radius
- GAUSSIAN KERNEL:
    1. Spatial radius (HS)
    2. Color radius (HC)
    3. Threshold
    4. Window Radius

All of these parameters were so adjusted to get the best segmentation.

RESULTS

The first part contains image results for flat kernel with following representation:

(Threshold,Window Radius square, Number of segments)

The second part contains results for Gaussian kernel with following representation:

(HS,HC,Number of segments) - varying HC while keeping HS constant

(HS,HC,Number of segments) - varying HS while keeping HC constant

Code Description

Modes or convergence points for all pixels are computed using standard mean shift algorithm and then stored in a 2-d matrix named mode. Groups are created and all the pixels are classified by using total distance (euclidian distance using color and spatial distance with proper tradeoff) as the deciding parameter.

SPEED-UPS:

- All the points(pixels) traversed during the mode computation of a pixel are stored in an array and once the convergence point (p_{convg}) is obtained, all the pixels in the stack are popped out and assigned (p_{convg}) as mode. A flag matrix is maintained to keep a track of what all pixels have been assigned a mode value.
- Instead of processing all pixels to find nearest neighbour, certain window has been used on the 2-d image to find the point most similar to the successive means found.

Both of these techniques have helped significantly in speeding up the segmentation. Any test image could be segmented within couple of minutes without compromising with the quality of segments.

Compilation and execution:

g++ -o Gaussian Gaussian.cpp `pkg-config --cflags --libs opencv`

./Gaussian imageName.jpg

g++ -o Flat Flat.cpp `pkg-config --cflags --libs opencv`

./Flat imageName.jpg

ORIGINAL:

FLAT KERNEL:

(500,600,19)

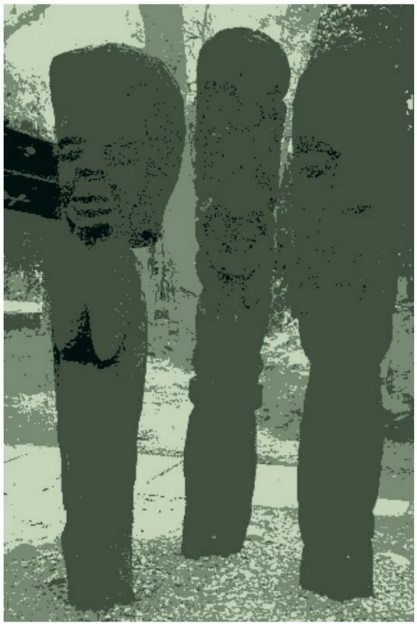
(700,600,13)

(1100,600,11)

GAUSSIAN KERNEL:

(20,35,3)

(20,30,3)



(20,15,4)

(30,25,3)

(25,25,3)

(20,25,4)

ORIGINAL:

Thresholds were decreased as compared to other images to differentiate points with similar color values and obtain finer segments.

FLAT KERNEL:

(400,800,26)

(500,800,17)

(700,800,13)

GAUSSIAN KERNEL:

(10,10,21)

(10,20,15)

(10,30,13)

(10,10,21)

(20,10,19)

(30,10,20)

ORIGINAL:

FLAT KERNEL:

(500,900,74)

(1000,900,35)

(2000,900,20)

GAUSSIAN KERNEL:

(10,35,3)

(10,25,3)

(10,15,5)

(10,20,4)

(20,20,4)

(30,20,4)

ORIGINAL:

FLAT KERNEL:

(500,1500,11)

(1000,1500,7)

(2500,1500,3)

GAUSSIAN KERNEL:

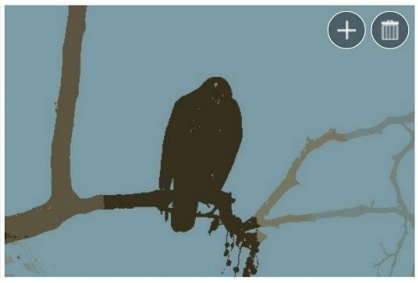
(25,15,12)

(25,25,10)

(25,35,9)

(10,25,12)

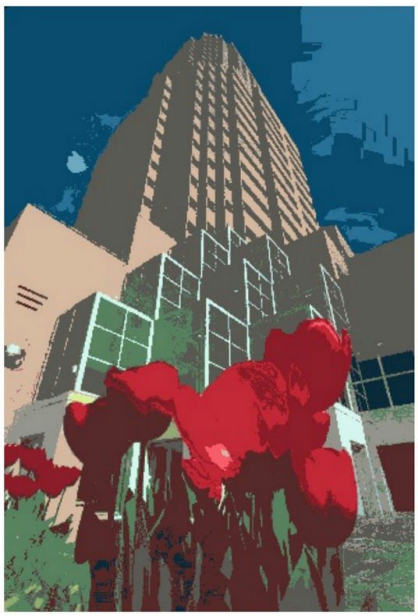
(20,25,9)



(25,25,10)

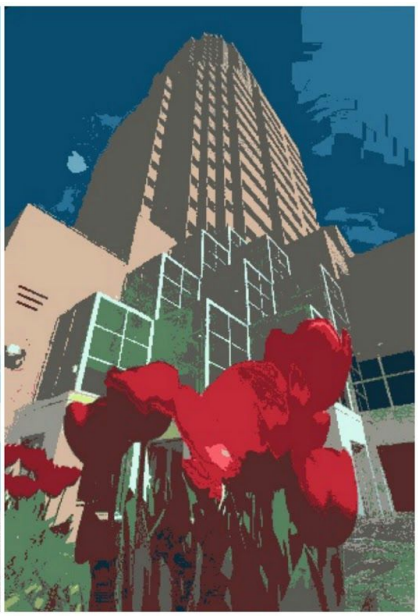
ORIGINAL:

FLAT KERNEL:



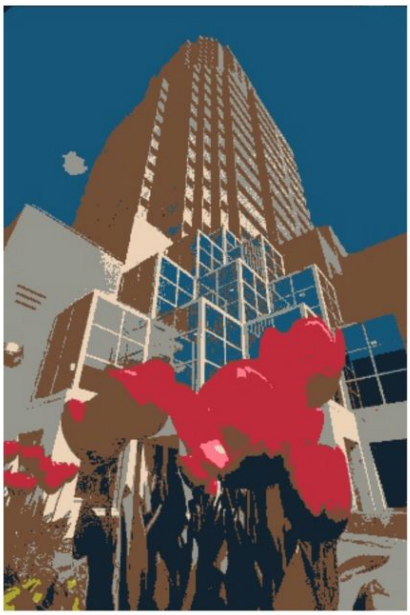
(1500,600,17)

(2000,600,13)



(500,900,100)

GAUSSIAN KERNEL:



(10,10,8)

(10,20,8)

(10,30,7)

(10,15,8)

(20,15,7)

(30,15,7)