EE 569 HOMEWORK 2

Issued: 9/18/2015   Due: 10/11/2015

Name: Meiyi Yang

Email: meiyiyan@usc.edu

USC ID: 6761-0405-85

Problem 3: Image Segmentation

3.1 Motivation

Image segmentation is a technique that segments the digital image to several small regions by labeling each pixels for later analyzing and processing [1]. In this problem, I will implement two image segmentation methods: Mean-shift and superpixel segmentation, Color Palettes Generation, and then evaluate the results using: Cov, PRI, VoI, GCE, BDE.

3.2 Approach and procedure

3.2.1 MS + superpixel Segmentation

Mean-shift. Mean-shift method for image segmentation include two step: mean-shift filtering, and mean-shift clustering.

(1) Filtering: Filtering the image by Gaussian kernel function

(2) Cluster: Use gradient descent to find the local maximum;

Combine the pixels within distance to the same label;

Merge small regions; [2]

Superpixel: Superpixel means segmenting image to sets of pixels by computing local image features. This can reduce the complexity of image processing and be used in several areas [3]. Here we use superpixel to pre-processing the image, and get the mean of value for each region which has the same label.

Implementation. Then apply MS to the image. The superpixel algorithm is used by Pacal mettes' script (From: https://github.com/PSMM/SLIC-Superpixels) and the MS algorithm is referred from Kanglai Qian (From: https://github.com/qiankanglai/opencv.meanshift). All this two use Opencv as a library. I adjust above code and set different parameter to implement this problem. The flow chart of this MS + superpixel segmentation method is shown in the Figure 1.

There are four parameters are using in the Mean-Shift and superpixel segmentation:

(1) nr: number of superpixel

(2) nc: weighted factor for superpixel processing

(3) hs: spatial radius in Mean-Shift

(4) hc: color radius in Mean-Shift

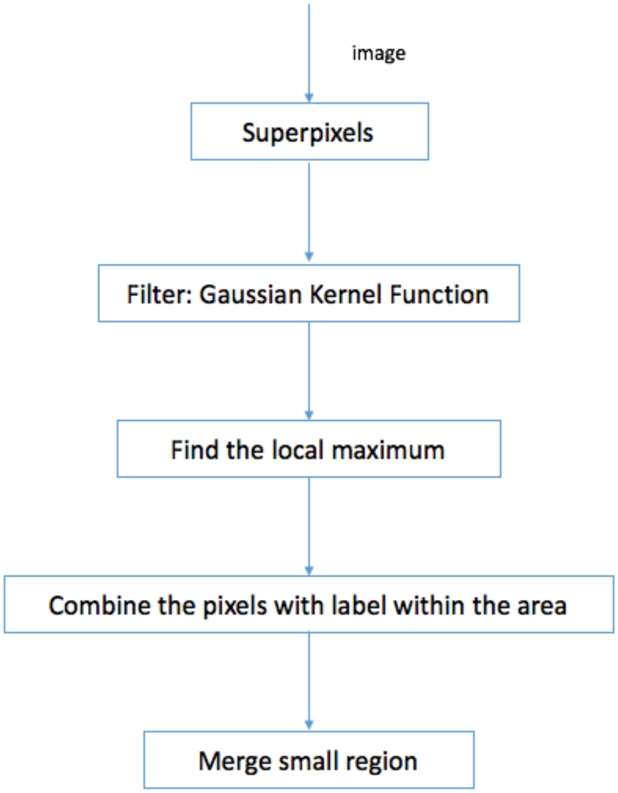


image
Superpixels
Filter: Gaussian Kernel Function
Find the local maximum
Combine the pixels with label within the area
Merge small region

3.2.2 Color Palettes Generation

CPP (Color Palettes Generation). Compared to the MS, CPP also consider the pixels near the detected long contours as an important factor for calculating the image segmentation. This method uses Structured Edge (problem2.3) algorithm to detect the contour of the image. And then combine the contour result with the color palette generation result to get the segmented image. To improve the segmented image, CPP apply three post-processing: leakage avoidance, fake boundary removal and small region mergence [4][5].

Implementation. I use the source code of CPP provided from https://github.com/fuxiang87/MCL_CCP.

3.2.3 Segmentation result evaluation

To evaluate the performance of image segmentation, here we use five new region quality assessment parameter to measure the results I get. These five parameter are: Cov, PRI, VoI, GCE, BDE.

Implementation. This evaluation code is referred by John Wright, and Allen Y. Yang, University of California, Berkeley, 2007.

3.3 Results

3.3.1 MS + superpixel Segmentation

First I apply only superpixel to the image in Figure 2, 3. The red line in the left figure is the boundary.

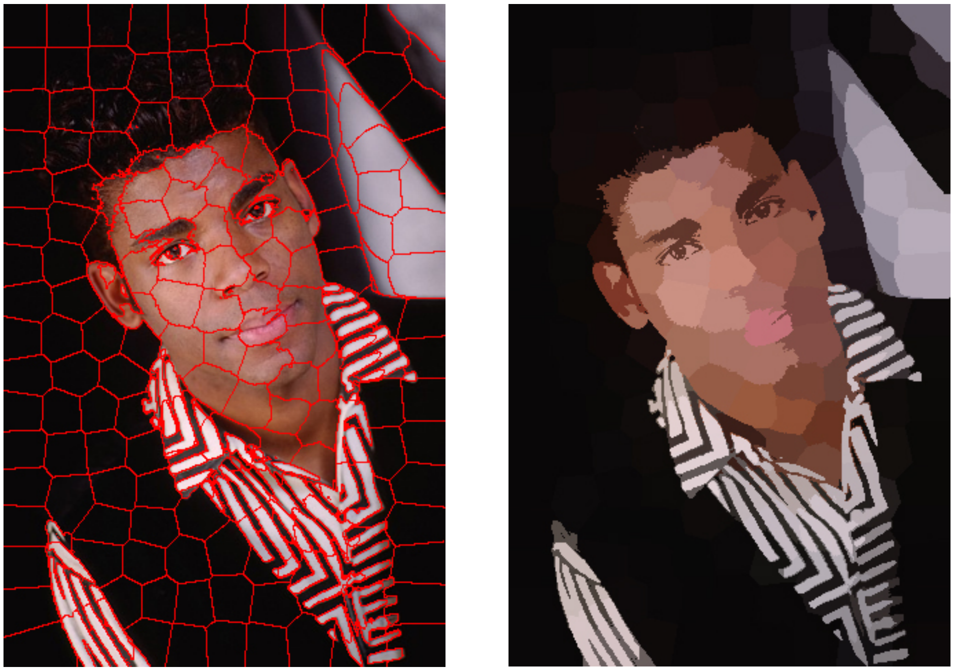


Figure 2. Superpixel for Man, nr = 200, nc = 50. Left: display contour. Right: display color region

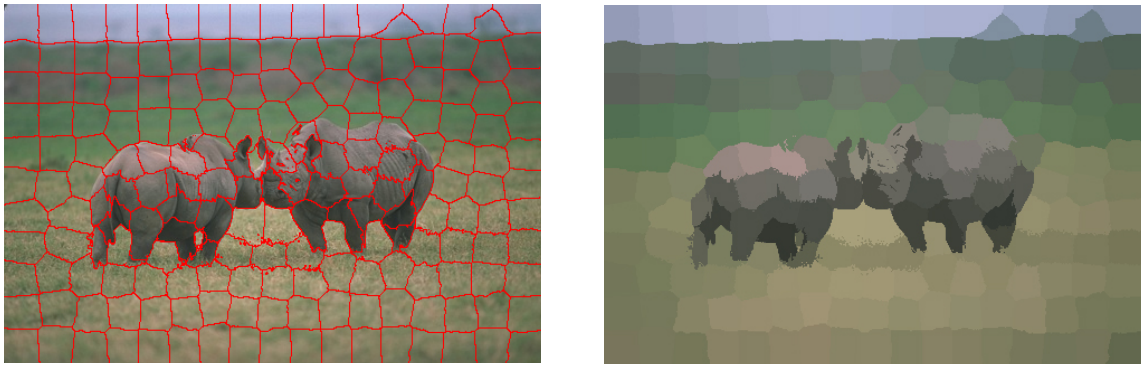


Figure 3. Superpixel for Rhinos, nr = 200, nc = 50. Left: display contour. Right: display color region

Try different nr (number of superpixel) in Figure4, 5.

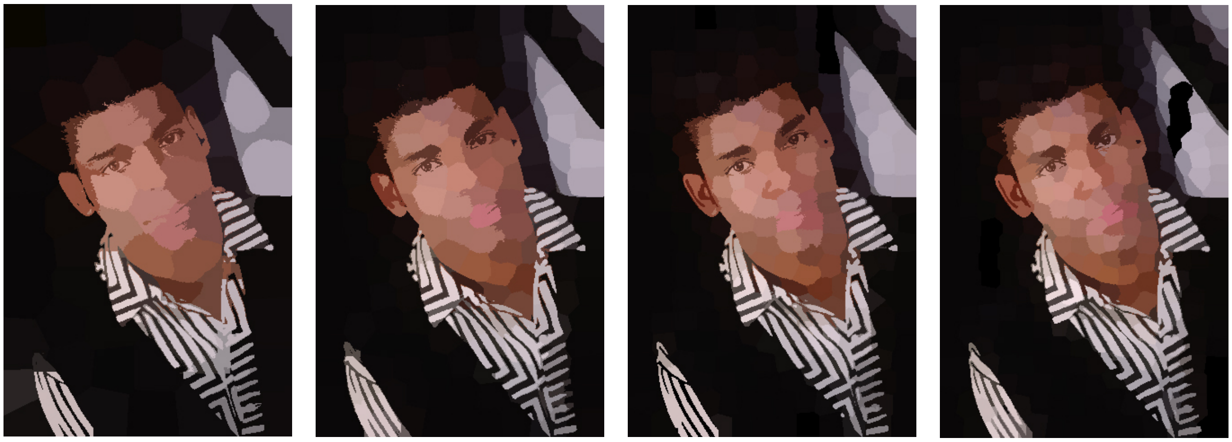


Figure 4. Superpixel for Man, nc = 50. Left to Right: nr = 100, 200, 300, 400

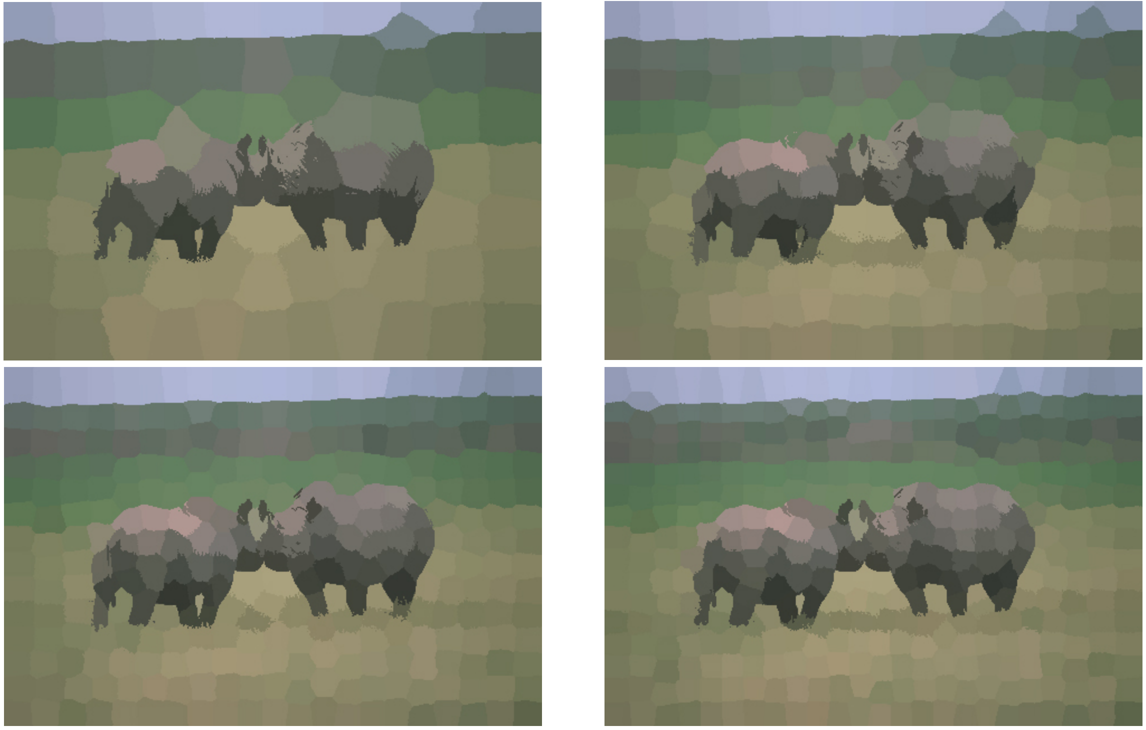


Figure 5. Superpixel for Rhinos, nc = 50. Top-Left to Bottom-Right: nr = 100, 200, 300,

Try different nc (weighted factor) in Figure 6, 7.

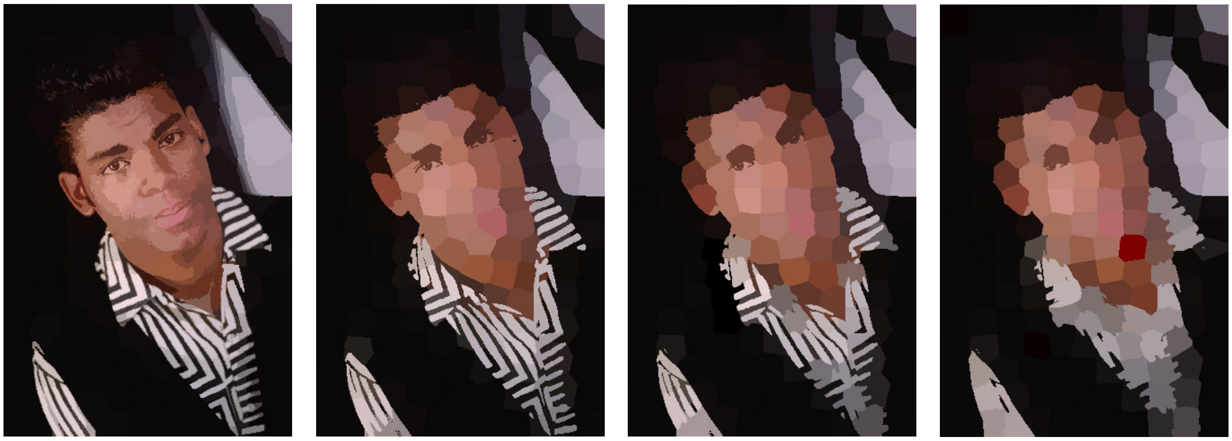


Figure 5. Superpixel for Man, nr = 200. Left to Right: nc = 10, 50, 100, 200

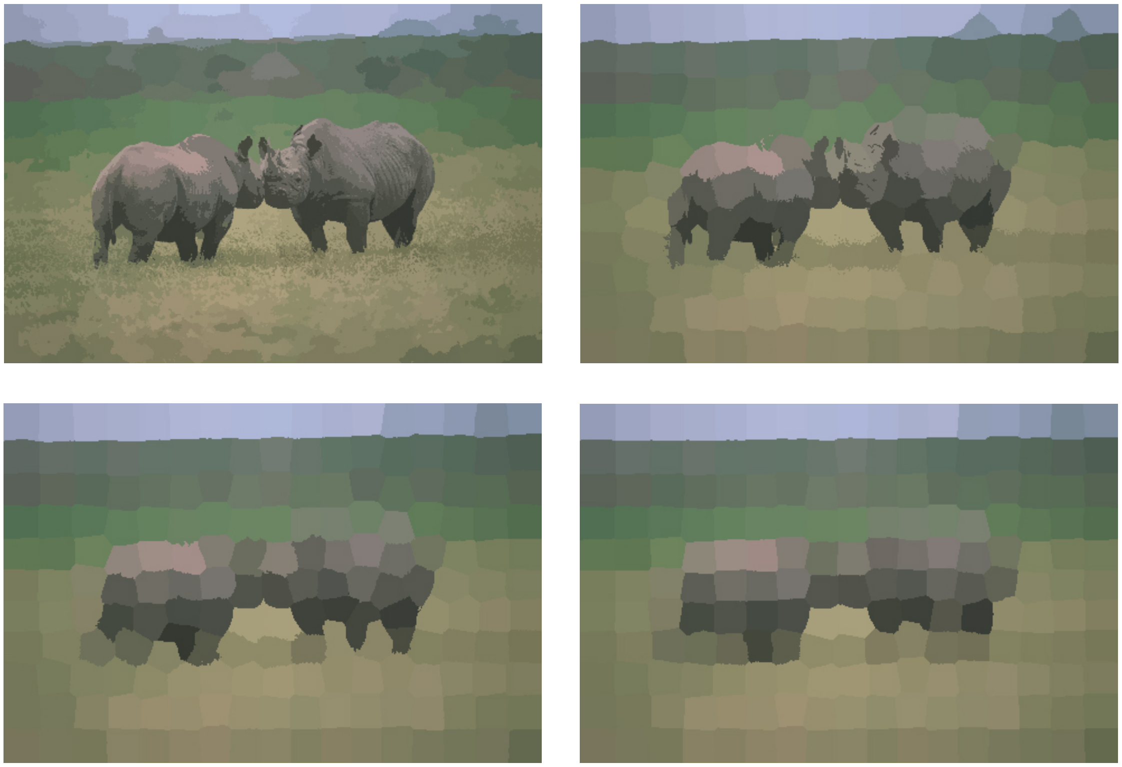


Figure 6. Superpixel for Rhinos, nr = 200. Top-left to bottom-right: nc = 10, 50, 100, 200

Try different hs (spatial radius in Mean-Shift) in Figure7, 8.

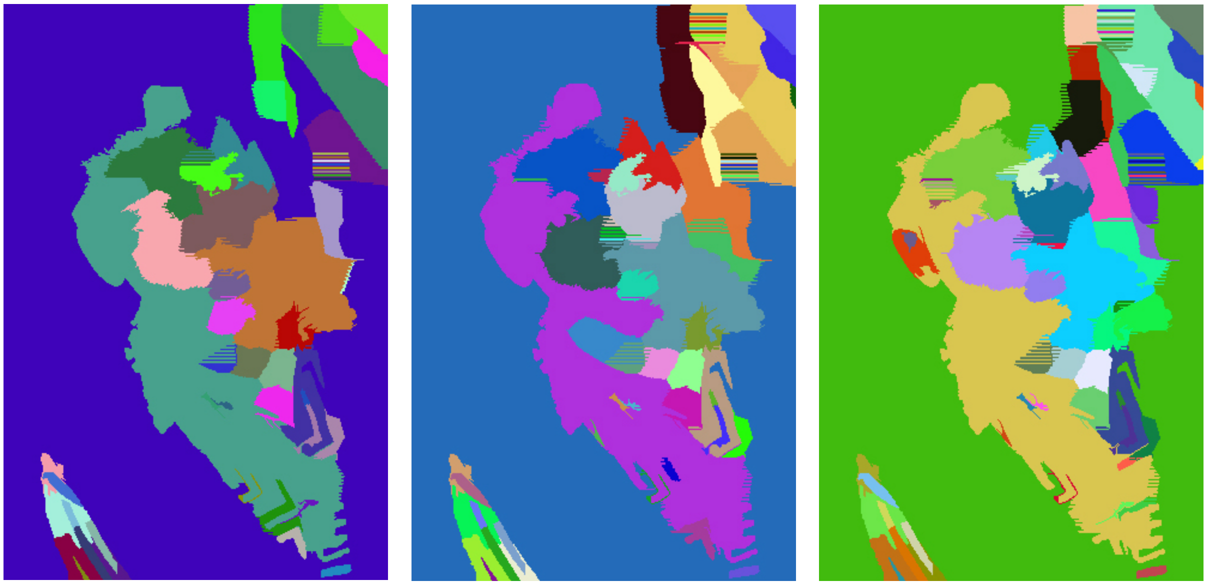


Figure 7. MS+superpixel for Man, nr = 200, nc = 50, hc = 6.5. Left to right: hs = 4, 10, 14

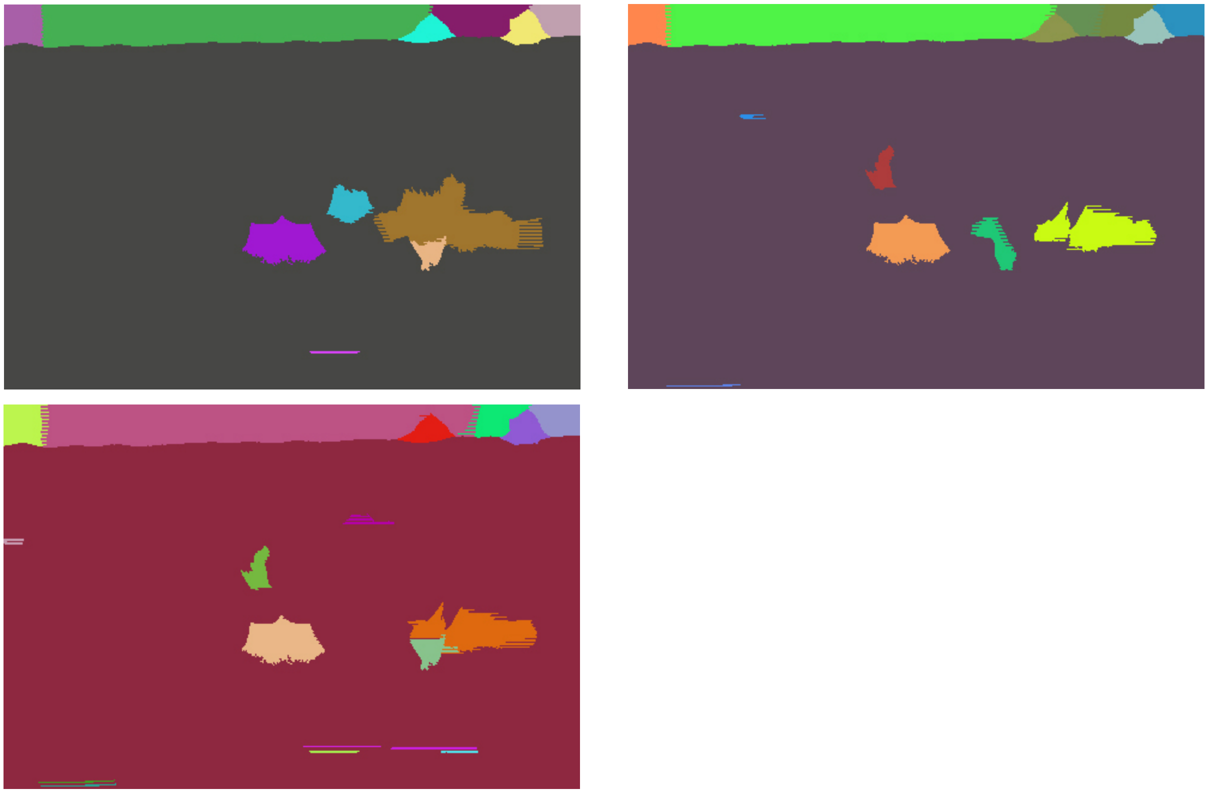


Figure 8. MS+superpixel for Rhinos, nr = 200, nc = 50, hc = 6.5. Top-Left to bottom: hs = 4, 10, 14

Try different hc (color radius in Mean-shift) in Figure 9, 10.

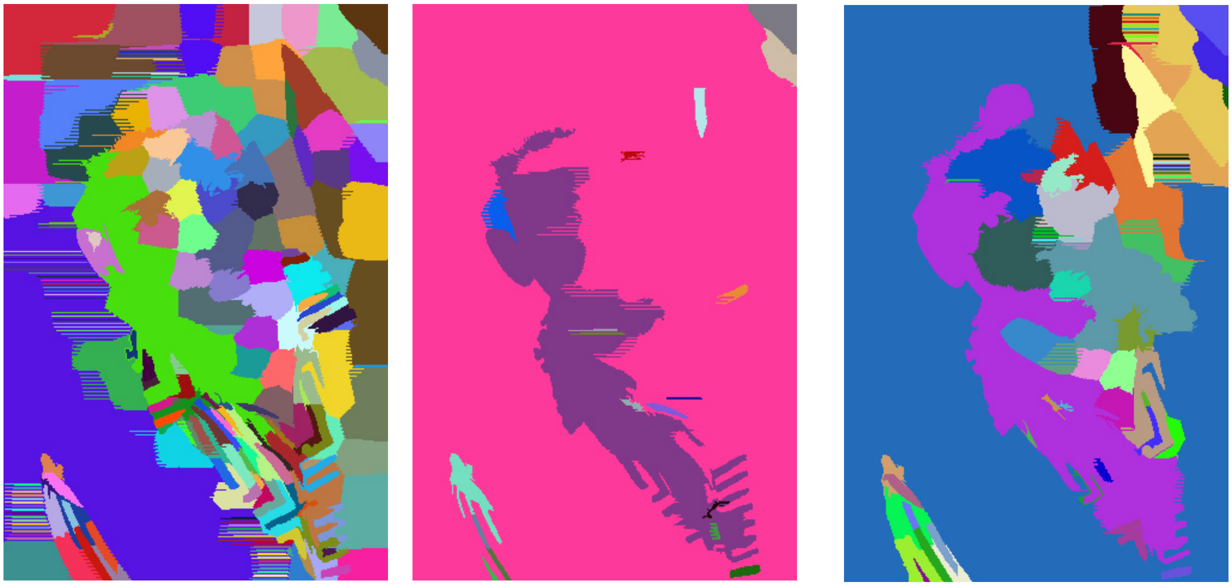


Figure 9. MS + superpixel for Man, nr = 200, nc = 50, hs = 10. Left to right: hs = 2, 6.5, 12

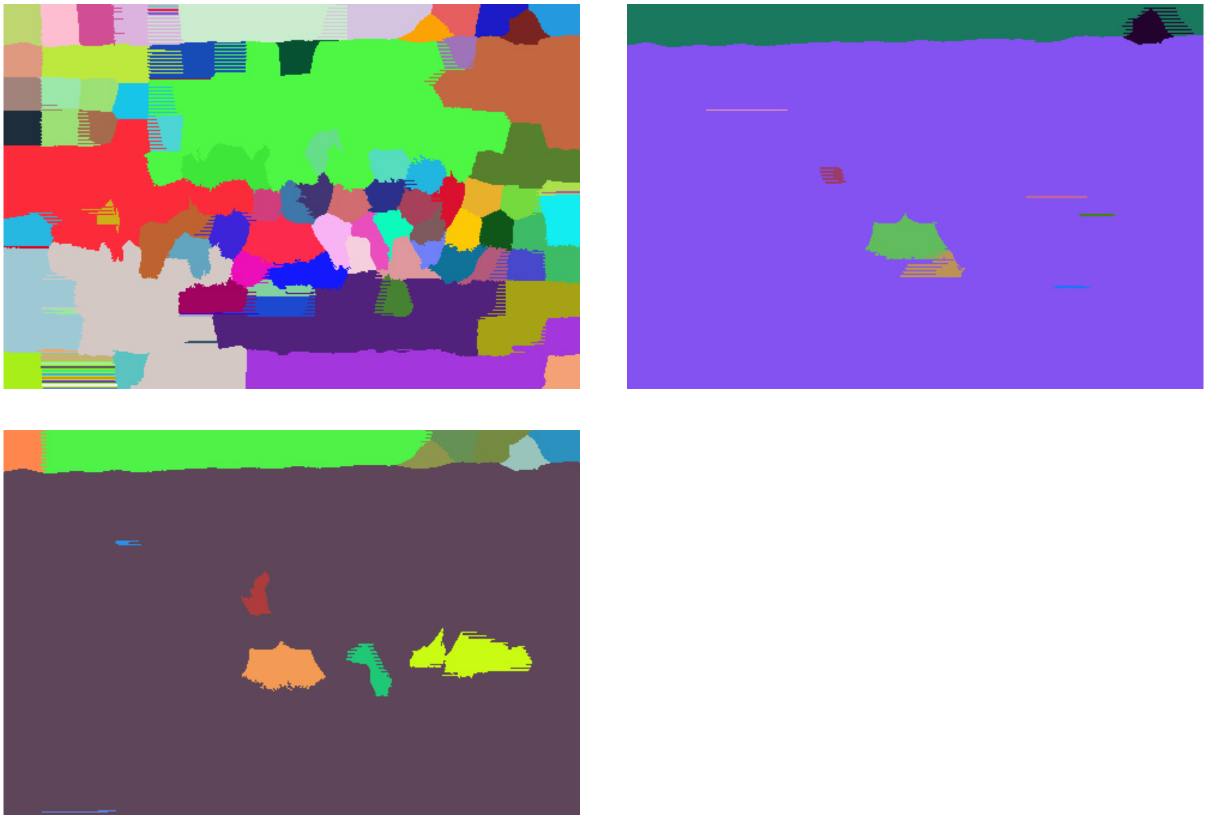


Figure 10. MS + superpixel for Rhinos, nr = 200, nc = 50, hs = 10. Top-Left to bottom: hc = 2, 6.5, 12

3.3.2 Color Palettes Generation

Apply CPP to Man and Rhinos image.

(1) Denosing images

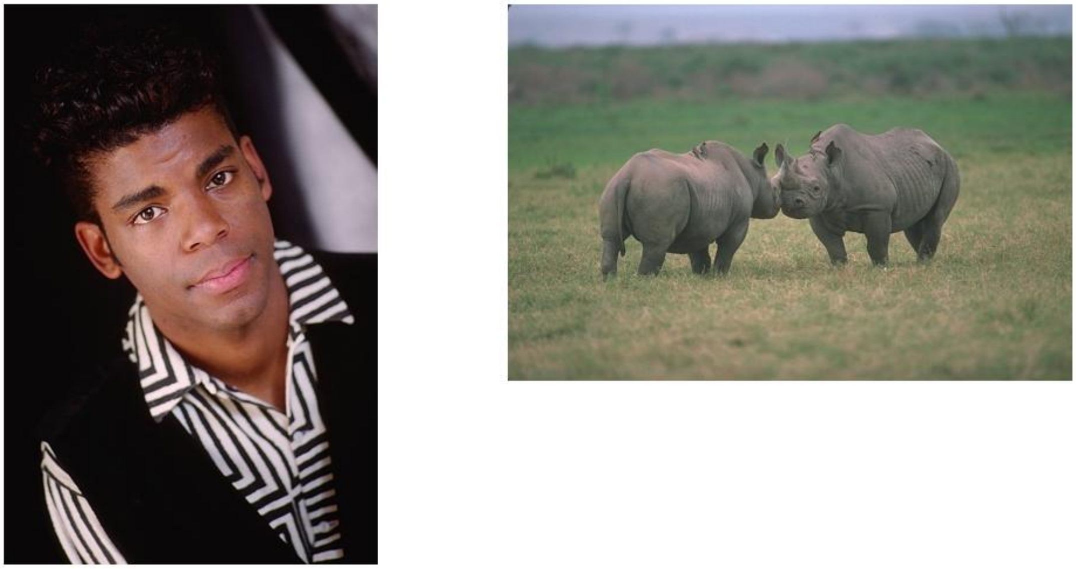


Figure 11. CPP, original images. Left: Man; Right: Rhinos

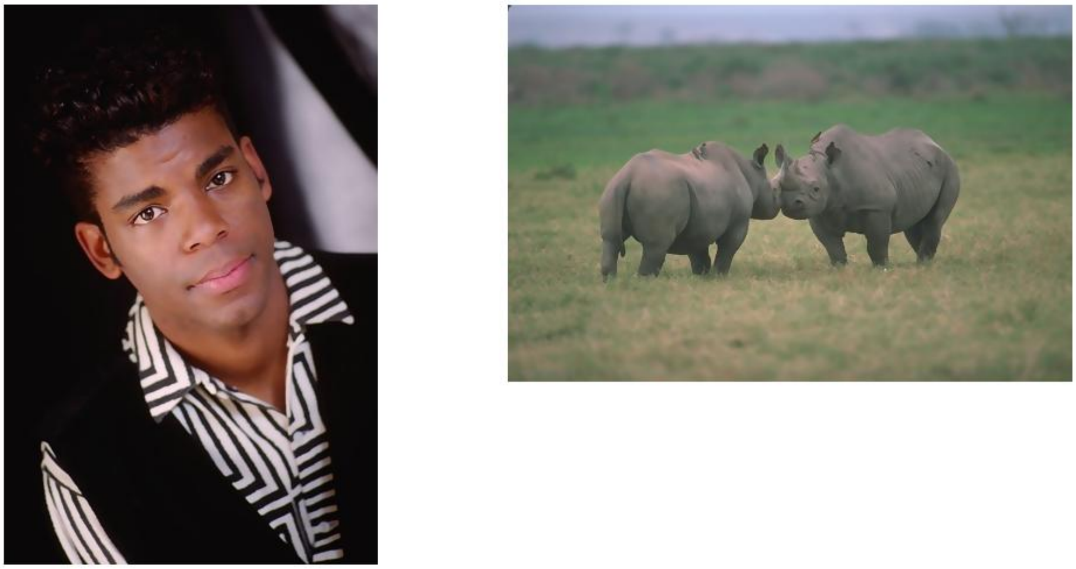


Figure 12. CPP, denoising images. Left: Man; Right: Rhinos

(2) Contour image

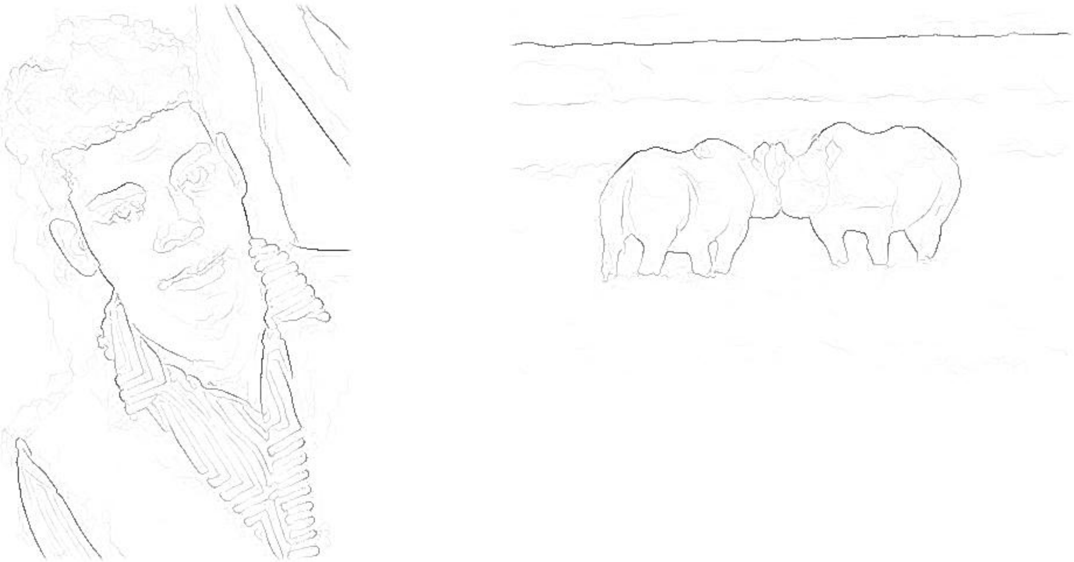


Figure 13. CPP, Probability maps from SE detector. Left: Man; Right: Rhinos

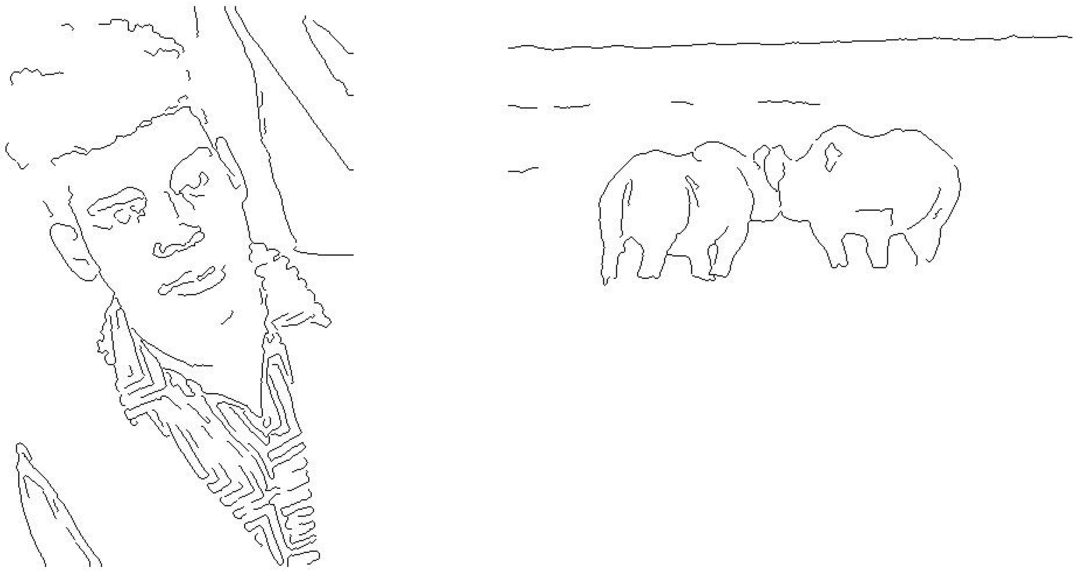


Figure 14. CPP, contour maps from SE detector. Left: Man; Right: Rhinos

(3) Sample color images

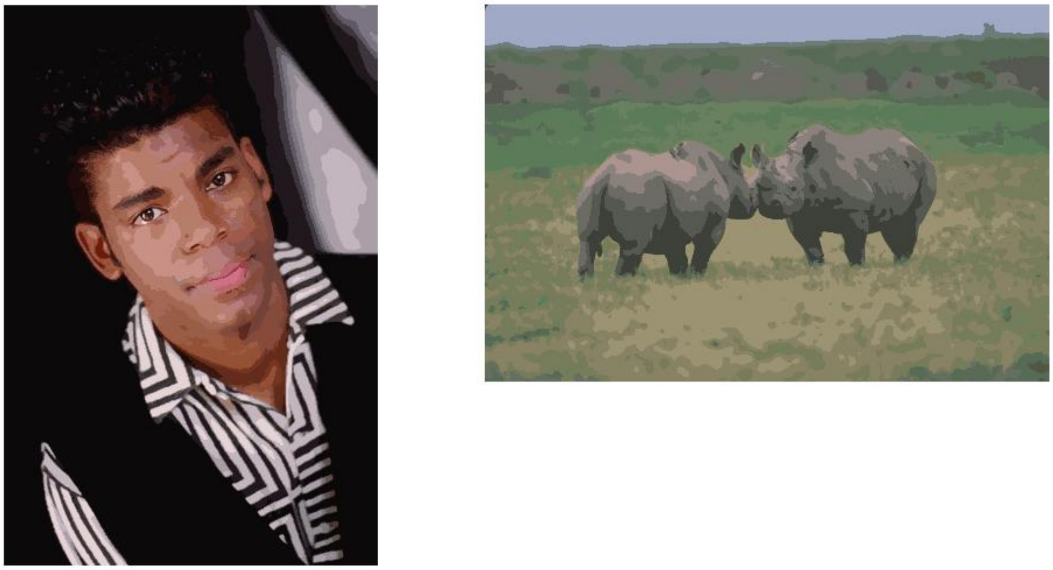


Figure 15. CPP, sample color images. Left: Man; Right: Rhinos

(4) Segmentation images (images with contour and color segmented images)

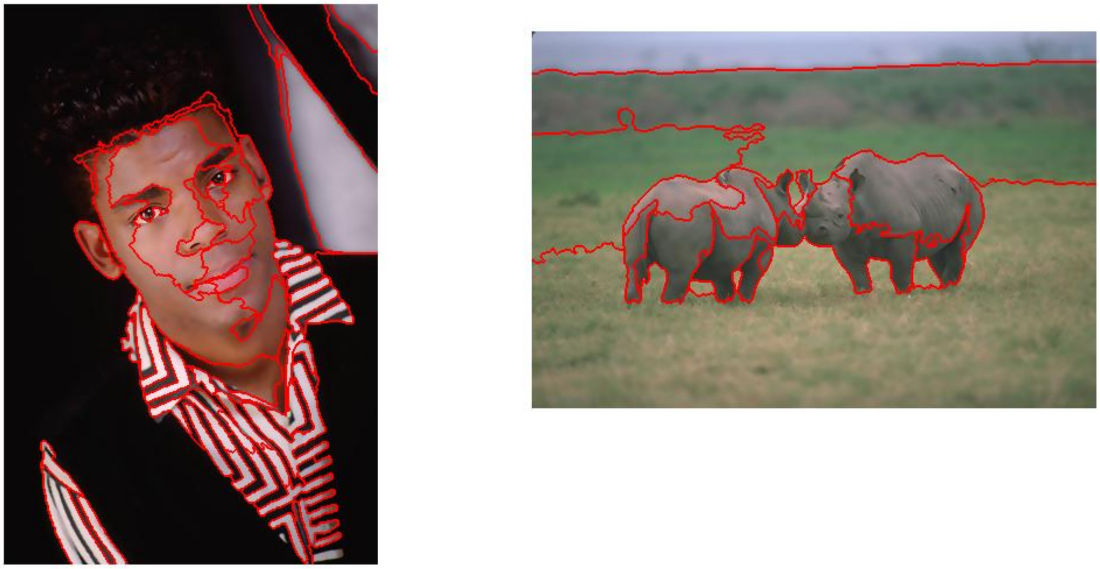


Figure 16. CPP, contour results. Left: Man; Right: Rhinos

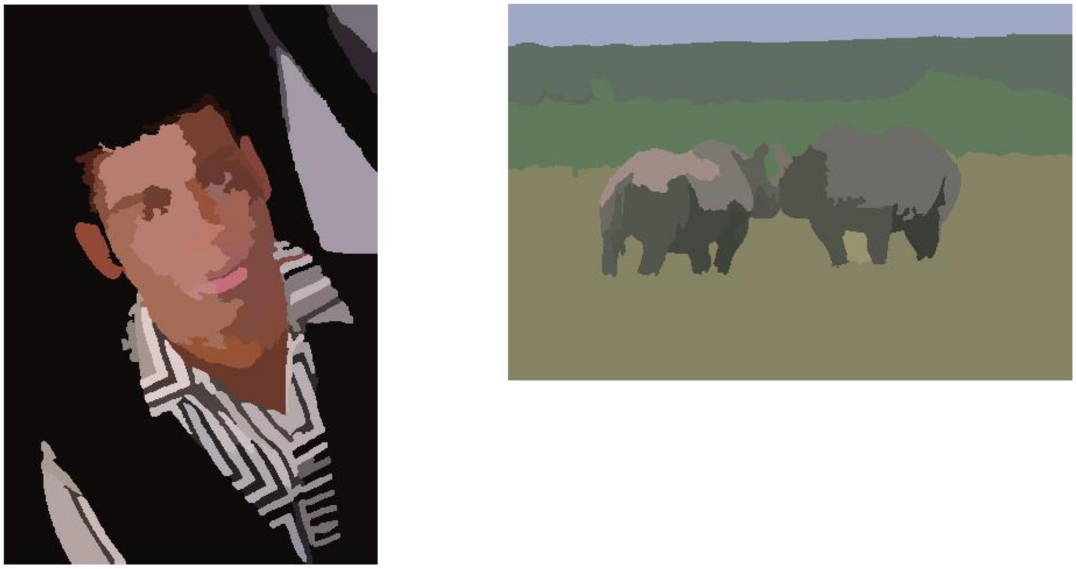
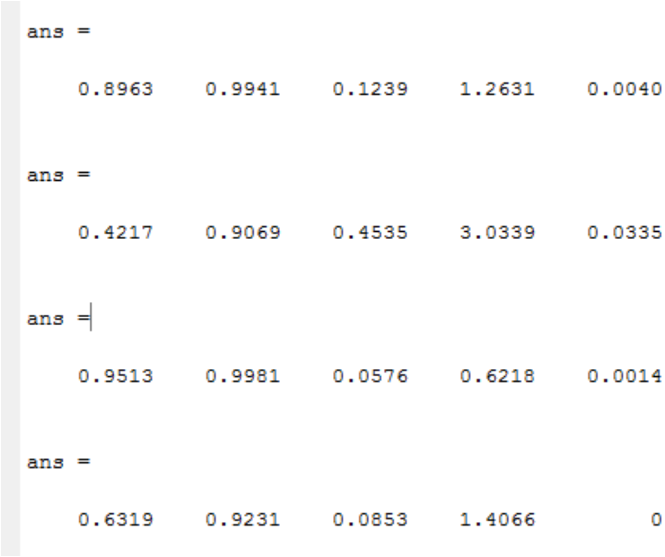


Figure 17. CPP, post-processing color segmented results. Left: Man; Right: Rhinos

3.4.3 Segmentation result evaluation

The results of processing evaluation are shown in Figure 18, and Table1.



```
ans =

    0.8963    0.9941    0.1239    1.2631    0.0040


ans =

    0.4217    0.9069    0.4535    3.0339    0.0335


ans =

    0.9513    0.9981    0.0576    0.6218    0.0014


ans =

    0.6319    0.9231    0.0853    1.4066         0
```

Figure 18. Evaluation result for Man and Rhinos

	Cov	PRI	GCE	VoI	BDE
Man: bound	0.8963	0.9941	0.1239	1.2631	0.0040
Man: color	0.4217	0.9069	0.4535	3.0339	0.0335
Rhinos: bound	0.9513	0.9981	0.0576	0.6218	0.0014
Rhinos: color	0.6319	0.9231	0.0853	1.4066	0

Table 1. Result for evaluation

3.4 Discussion

3.4.1 MS + superpixel Segmentation

Mean-shift. The results I got by using mean-shift and superpixel are worse than CPP. For displaying, I assign random color value to different labels in the final results.

Different parameters. There are four parameters are using in the Mean-Shift and superpixel segmentation: nr (number of superpixel), nc (weighted factor for superpixel processing), hs (spatial radius in Mean-Shift), hc (color radius in Mean-Shift).

nr: different nr means different number of superpixels in the image. More nr, more superpixels. The default number of nr I use here is 200.

nc: this is a weighted factor for the color value of each superpixel. The default number of nr I use here is 50.

hs: this parameter represents the spatial radius using in mean-shift when getting the area to label the same area. When hs increase, the result image become harder recognize to human being.

hc: When label the area, we need hc as a color radius to calculate the gradient of color and get the area. When hc increase, the different number of color assigned in the current area become lower.

3.4.2 Color Palettes Generation

Purpose of Denoising. This denoising filter is very important to CPP. We use a shift able bilateral filtering in this method. If we don't apply denoising filter, the result will be show in Figure 19. Without denoising filter, we get a terrible sample color image and lost a lot of details in the final segmented image.

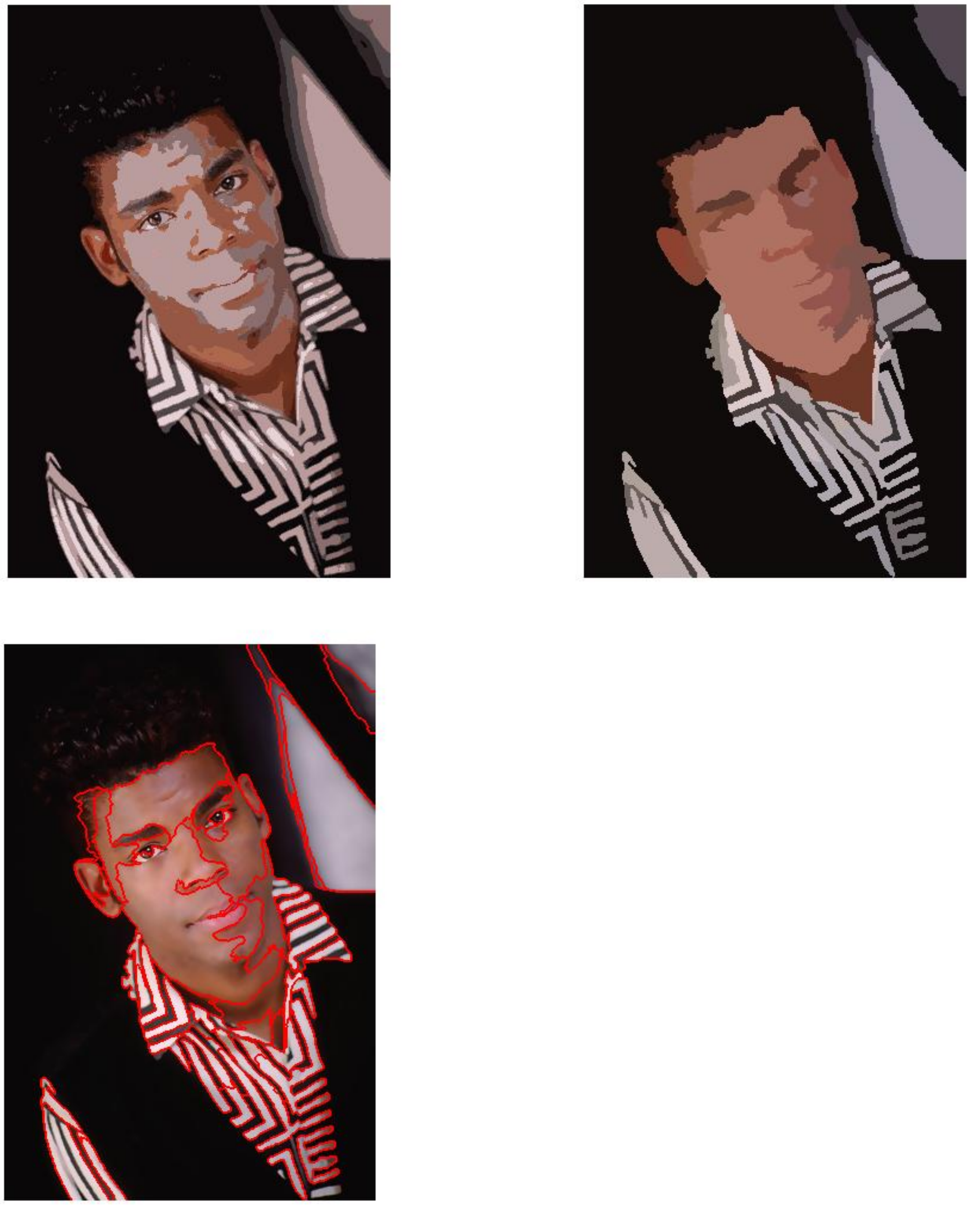


Figure19. CPP. Without denoising. Top Left: sample color image. Top-Right & Bottom: final CPP results.

Edge extension. The edge extension method using SE here is for calculating the long contour in the image and use this contour we can deter the sample color beside the contour.

Effect of spectral radius in Mean-shift. Please see in 3.4.1

Post-processing: leakage avoidance by contours. This post-processing partition the region that are separated by long contour. If we don't use this post-processing. The results will become Figure 20. We can see that with leakage avoidance the result lost some color between long contour.

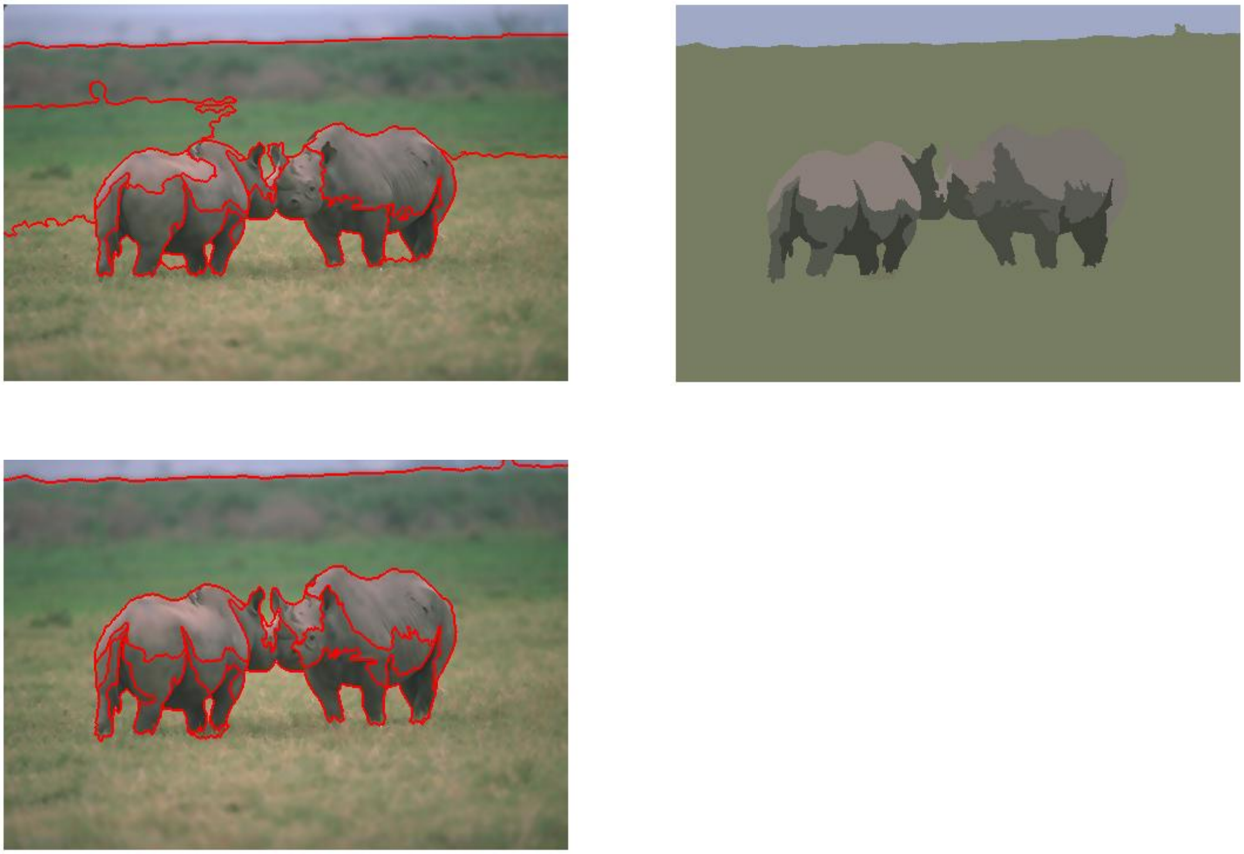


Figure 20. Top-Left: correct result. Top-Right & bottom-left: results without leakage avoidance by contours

Post-processing: fake boundary removal. This post-processing checks each common boundary between two adjacent regions. This give us a lot of detail of contours image. See details in Figure 21.

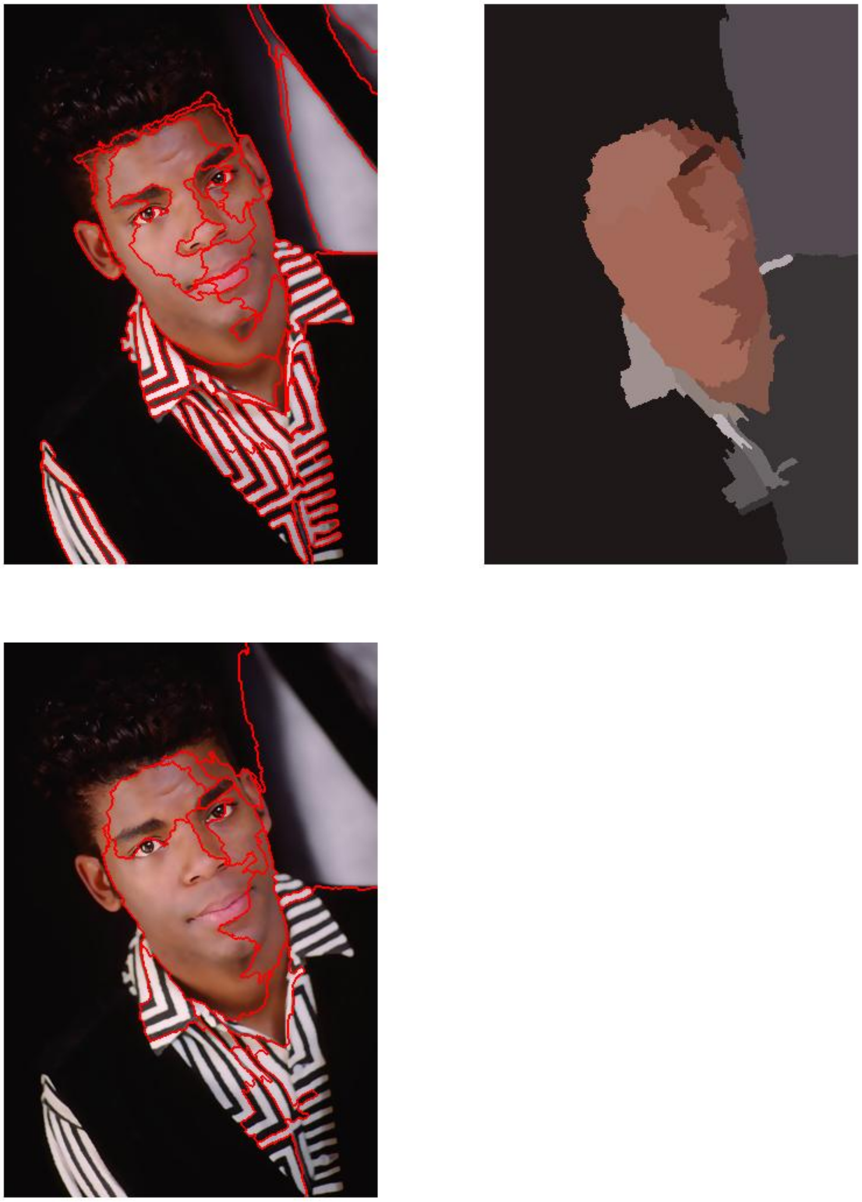


Figure 21. CPP. Top-Left: correct result. Top-Right & bottom-left: results without fake boundary removal

Post-processing: small region mergence. This post-processing checks each small region with area less than areaTH.small. To be specific, I change the parameter of merging in the small region from 0.1 to 5. Figure 22 shows this. We can see that if I increase the merging parameter, too many pixels are merged.

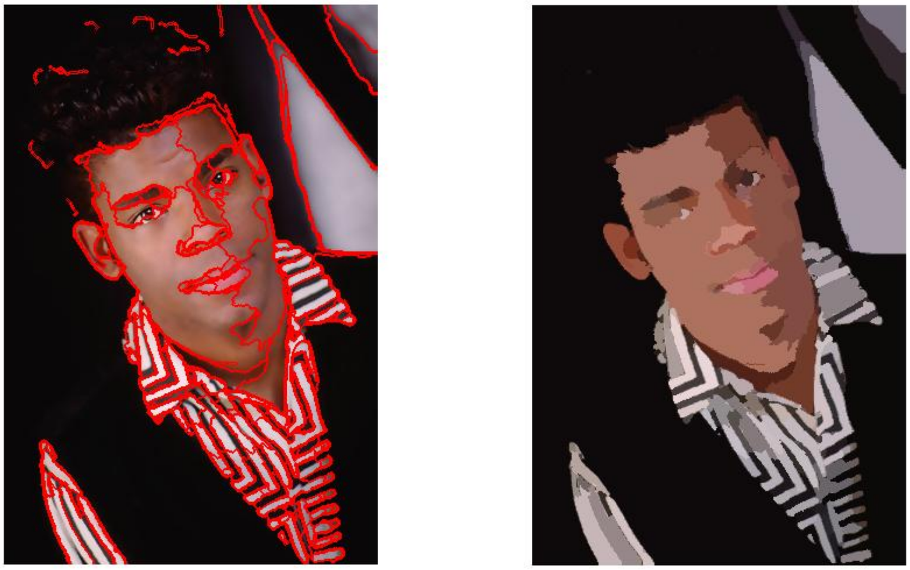


Figure 22. CPP. Results for changing merging parameter from 0.1 to 5.

3.4.3 Segmentation result evaluation

(1) Cov: this is segment covering parameter.

(2) PRI: Probabilistic Rand index

(3) VoI: Variation of Information

(4) GCE: this is a parameter for Global Consistency Error

(5) BDE: this is a parameter for calculate the displacement of ground-true boundary image and result boundary image.

Reference

[1] https://en.wikipedia.org/wiki/Image_segmentation
[2] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 5, pp. 603–619, 2002.
[3] Van den Bergh, Michael, et al. "Seeds: Superpixels extracted via energy-driven sampling." Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012. 13-26.
[4] EE569 Discussion #6, Chun-Ting Huang, Xiang Fu

[5] Xiang Fu, Chien-Yi Wang, Chen Chen, Changhu Wang, C.-C. Jay Kuo. Robust Image Segmentation Using Contour-guided Color Palettes. Submitted to ICCV 2015.