Table Of Contents

1. Problem #3: Image Segmentation 3

1.1 MS + Super-pixel Segmentation 3
1.1.1 Motivation 3
1.1.2 Approach and Implementation 3
1.1.3 Results 6
1.1.4 Discussion 9

1.2 Color Palettes Generation 11
1.2.1 Motivation 11
1.2.2 Approach and Implementation 11
1.2.3 Results 17
1.2.4 Discussion 18

1.3 Segmentation Result Evaluation 19
1.3.1 Motivation 19
1.3.2 Approach and Implementation 19
1.3.3 Results 20
1.3.4 Discussion 20

2. References 21

Table of Figures

1.1 (a) Original Man image 6

1.1 (b) Man image with 150 super-pixels 6

1.2 (a) Original Rhinos image 7

1.2 (b) Rhinos image with 150 super-pixels 7

1.3 (a) Man image with Hs = 15, Hr = 10, Iterations = 4 8

1.3 (b) MS Rhinos image with Hs = 15, Hr = 10, Iterations = 5 8

1.4 (a) Man image with Hs = 8, Hr = 7, Iterations = 2 9

1.4 (b) Man image with Hs = 40, Hr = 32, Iterations = 16 9

1.5 (a) Rhinos image with Hs = 8, Hr = 7, Iterations = 3 10

1.5 (b) Rhinos image with Hs = 40, Hr = 32, Iterations = 14 10

1.6 (a) Original Man Image 11

1.6 (b) Denoised Man Image 11

1.6 (c) Original Rhinos Image 12

1.6 (d) Denoised Rhinos Image 12

1.7 (a) SE Man Image 13

1.7 (b) SE Long Contours Man image 13

1.7 (c) SE Rhinos Image 14

1.7 (d) SE Long Contours Rhinos image 14

1.8 (a) Color Palette for Man image 15

1.8 (b) Color Palette for Rhinos image 16

1.9 (a) CCP Segmented Man image 17

1.9 (b) CCP Segmented Rhinos image 18

1.10 Table of Evaluation Metrics 20

Amruta Kulkarni
USC ID: 6914970827
Email: arkulkar@usc.edu
Date: 15 Oct 2015

EE 569 HOMEWORK #2

---

1. Problem #3: Image Segmentation

1.1 MS + Super-pixel Segmentation

1.1.1 Motivation

Image segmentation is an important technique mainly used in the field of object recognition and computer vision. It mainly partitions an image into multiple segments so as to represent the image in a simple manner. Image in this form is easy to analyze and we can locate different objects by studying their boundaries. In this problem, we are first going to represent an image in the form of super-pixels and then we will apply the Mean Shift algorithm on this super-pixel representation of our original image.

1.1.2 Approach and implementation

- The Mean Shift segmentation is an algorithm in which every pixel is replaced with the mean of the pixels in a specified range only when the value is within a distance (suppose 'd').
- To calculate the distance between pixels, it uses the Euclidean distance.
- We first need to specify a radius and then the pixels within this radius will be considered for operations on them.
- As mentioned above, from all the pixels inside the radius, only those whose value is within the distance 'd' are considered for calculating the mean.
- This is the basic idea of the algorithm but the details are somewhat complex as they deal with shifted windows, clustering of data and also convergence.
- Let us look at the flow chart of this algorithm for a better understanding:

Part 1: To delineate clusters of pixels

↓

Map the image into its feature space by extracting the features that will lead to significant features forming clusters

↓

Perform the Mean shift procedure which includes computation of the mean shift vector and translation of the kernel / window by this vector

↓

Run this multiple times to cover the entire feature space and till the windows evolve towards the modes. This is convergence.

↓

Run this procedure to find stationary points at which the gradient of the density estimate is zero and then remove these points by retaining only the local maxima

↓

After convergence, the data points visited by all the mean shift procedures converging to a particular mode (basin of attraction) will delineate a cluster.

Part 2: Image representation using Mean shift segmentation

↓

Convert RGB image into LAB space where we can calculate the Euclidean distance between pixels

↓

Specify the kernel bandwidths Hs and Hr in the spatial and range domain respectively.

↓

Run the mean shift procedure on the image and store the convergence points

↓

Delineate the clusters by grouping the convergence points that are closer than Hs and Hr and then assign labels to each pixel according to the cluster it belongs to.

The above flowchart can be explained as follows:

- Convert the image data into feature space by using feature extraction
- Distribute windows over the feature space and specify Hs and Hr for the kernel
- Compute the mean vector of each window (kernel)
- Shift the window to the locations of these mean vectors
- Repeat the above two steps till the windows become stable i.e. till convergence
- Merge the windows if they are at the same location
- The data that was covered by each of the obtained windows will form a cluster around the location of this window.
- These clusters are nothing but the image segments.

In this problem, we have first computed the super-pixel representation [1] of our image and then operated the MS segmentation on this representation to get the final segmented image. The code has been implemented in MATLAB.

1.1.3 Results

The results obtained after using the SLIC super-pixels algorithm [1] on man and rhinos image are as shown below:

Figure No. 1.1(a) Original Man image

Figure No. 1.1(b) Man image with 150 super-pixels

Figure No. 1.2(a) Original Rhinos image

Figure No. 1.2(b) Rhinos image with 150 super-pixels

After applying the mean shift segmentation on images shown in Figure No. 1.1(b) and 1.2(b), we get the following segmented images for Man and Rhinos Image:

Figure No. 1.3(a) Man image with Hs = 15, Hr = 10, Iterations = 4

Figure No. 1.3(b) MS Rhinos image with Hs = 15, Hr = 10, Iterations = 5

1.1.4 Discussion

In the mean shift segmentation, as we have learnt in the above algorithm, we need to specify two key parameters namely Hs and Hr. Hs is for the spatial domain and Hr is for the range domain. The pixels that lie within the range 'Hr' are considered and now the pixels whose value is within the distance 'Hs' are chosen to calculate the mean. Thus, if we specify appropriate values of Hs and Hr then the important information of the image objects can be successfully retained. But if we use too large values for Hs and Hr then it will consider a large area of the image and form large clusters due to which we lose the image content as the segments fail to represent the image accurately. Moreover, the number of iterations required to converge are also more for larger values, thus the time required is more. Thus, we need to be careful when choosing these parameters. The effects of the two key parameters on the segmentation results are as follows:

Figure No. 1.4(a) Man image with Hs=8,Hr=7, Iterations = 2

Figure No. 1.4(b) Man image with Hs=40,Hr=32, Iterations = 16

Figure No. 1.5(a) Rhinos image with Hs = 8, Hr = 7, Iterations = 3

Figure No. 1.5(b) Rhinos image with Hs = 40, Hr = 32, Iterations = 14

1.2 Color Palettes Generation

1.2.1 Motivation

The Color-guided Color Palette (CCP) is another method of Image segmentation. It is the state-of-the-art method that accurately finds the representative colors of an image and integrates them with the contours. The CCP technique operates in the spatial domain to generate segmented images and then performs post-processing techniques such as leakage avoidance, small region mergence and fake boundary removal. This technique is superior to the Mean Shift method as it results in a more accurate representation of the image segments.

1.2.2 Approach and Implementation

We saw how the performance of the Mean Shift Segmentation is dependent on the bandwidth parameters. Choosing these parameters is a difficult task as they are image dependent. This drawback is removed in CCP. In this algorithm, we deal with just one key parameter and depending upon its appropriate range, the performance can be improved.

The steps of implementation of this algorithm are shown below along with the images obtained by performing them.

1) Denoising:

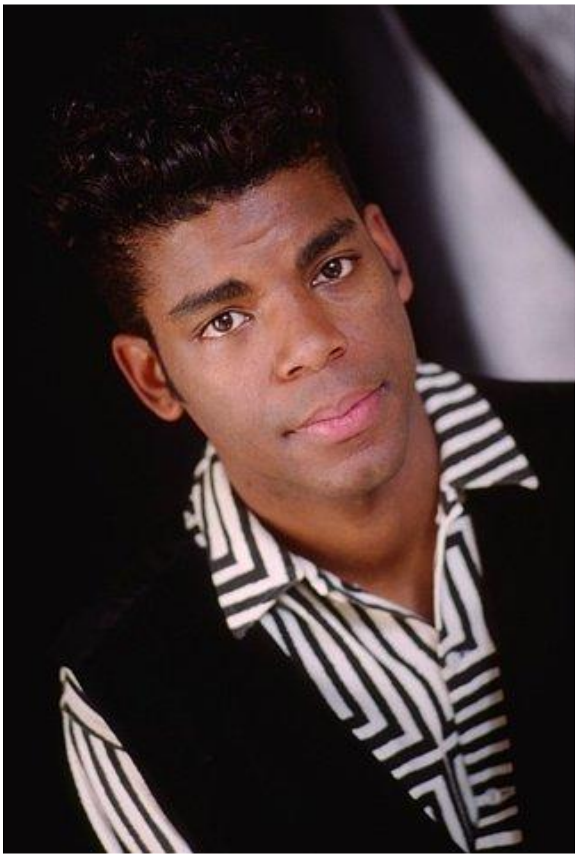

Figure No. 1.6(a) Original Man image

Figure No. 1.6(b) Denoised Man image

Figure No. 1.6(c) Original Rhinos image

Figure No. 1.6(d) Denoised Rhinos image

CCP uses bilateral denoising on the original images. If we closely observe the original and the denoised images for Man and Rhinos respectively, we can conclude that the denoised image is very smooth and it has no noise. Thus the purpose of this step is to remove noise while retaining the underlying image content and features. It kind of applies a blurring effect to the image. We can see how the face of the man is now smooth and similarly the skin of the Rhinos appears to be smooth. Also, the grass looks a bit blurry.

2) Structured Edge Detection

Figure No. 1.7(a) SE Man image

Figure No. 1.7(b) SE long contours Man image

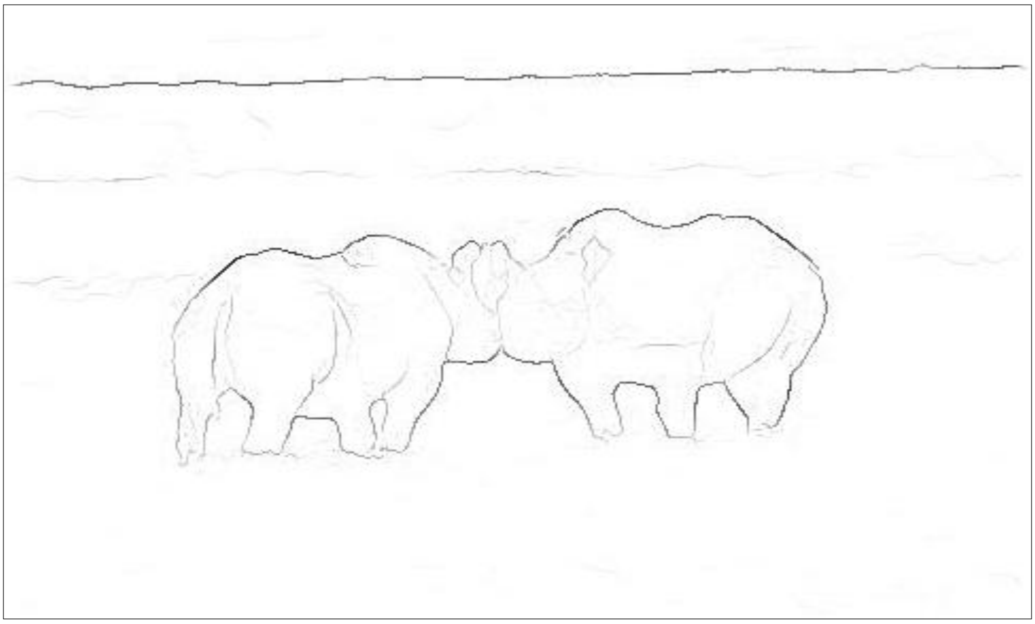

Figure No. 1.7(c) SE Rhinos image

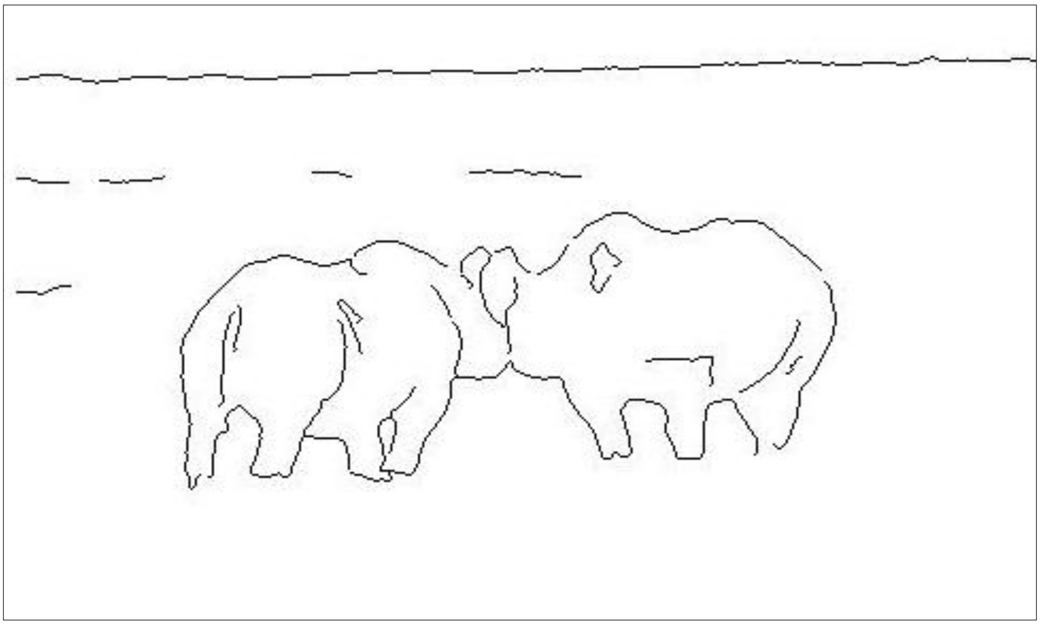

Figure No. 1.7(d) SE Long Contours Rhinos image

To detect the boundaries of the objects present in the image, we apply the Structured edge detection. SE image is the probability edge map that might contain both the edge as well as non-edge pixels. We want only the true edge pixels to be present in the image. Thus we use some threshold value on the probabilities to retain only the true edge pixels. The short contours in the image are not reliable because they might not be able to separate two regions with low contrast. Thus, we want to connect these contours to form long contours that can reliably separate two regions. For this purpose, we need edge extension.

3) Color Palette Generation

This is the initial segmentation of the image. It considers the denoised image as well as the long contours for generating the color palette. We select color pixels along two sides of long contours. Long contours provide useful spatial information about the region separation as we saw in the above images. Unfortunately, due to leakage problems, they might not be able to form closed boundaries. Thus, we collect color samples along both sides, and then after quantizing the clusters of these samples we get closed regions with neat boundaries. Now again, we cannot just uniformly collect color samples as sometimes we might end up choosing a sample that might not belong to that specific region. Thus, first we need to ensure that the samples we are choosing correspond to that specific region whose boundary is shown by the long contour. Due to this, we need to use contour-guided color samples.

The color palettes obtained by clustering of the Man and Rhinos image are shown below:

Figure No. 1.8(a) Color Palette for Man image

Figure No. 1.8(b) Color Palette for Rhinos image

The color palettes shown above are obtained by applying the MS clustering with bandwidth parameter Hr in the spectral domain. As we saw in the previous problem, we need to choose this parameter carefully to get reliable segments of the image. The radius cannot be too large because it will increase the computational complexity. The spectral radius is image dependent and needs to be found out by applying a range of different values. If we try to use the K-means color clustering algorithm, then we need to consider one more parameter i.e. the number of output clusters. However, finding an optimal solution to the K means clustering is difficult. It is computationally complex and the execution time is more, as it usually requires a lot of steps to converge.

4) Segment Post-Processing

To improve the final segmentation results, we perform 3 post-processing steps as follows:

a) Leakage Avoidance

   This problem is present when two regions with the same color straddle a long contour. For example in the Man image, the area near the neck shows this type of effect and in the Rhinos image, it can be seen near the junction of sky and grass. This can be avoided by retaining the long contour between the two regions even if they have the same color after quantization.

b) Fake Boundary Removal

   This occurs when there is a smooth transition of colors over a large region. For example, on the Man's face we can see many different colors with boundaries and in the Rhinos image there are many fake boundaries on the grass. This effect is removed by checking if the common boundary between two neighboring regions. If the boundary is very significant and not lying on the long contour then the regions are merged.

c) Small Region Mergence

There might be some insignificant small regions in the image. We can get rid of these by merging these small windows with their closes effective neighbors. For example, in the grayish background of the Man, there are some small windows with a different color and these can be merged with their neighboring effective color. Similarly, in the rhino's image, we can see some dark green spots in the grass and again these small windows can be merged to get a more uniform color in that region.

1.2.3 Results

This algorithm [2] was implemented in MATLAB. The final CCP segmentation images of Man and Rhinos are as shown below:

Figure No. 1.9(a) CCP Segmented Man image

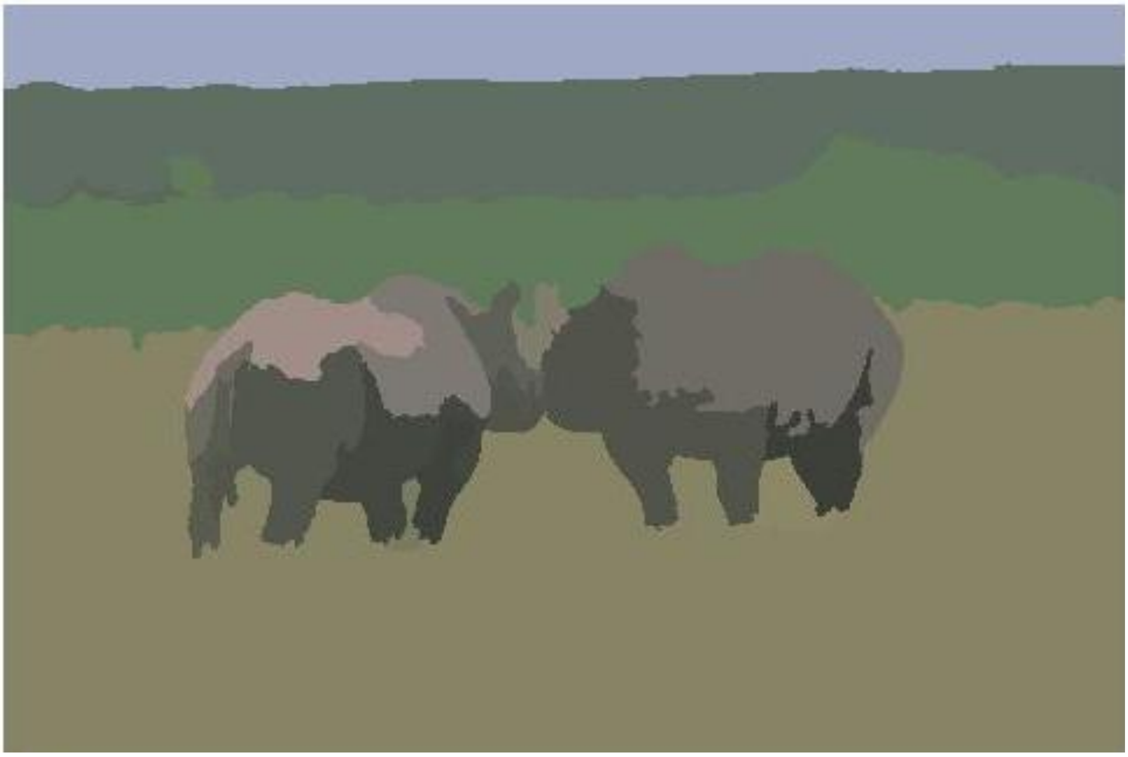

Figure No. 1.9(b) CCP Segmented Rhinos image

1.2.4 Discussion

We can conclude that CCP provides simplified segmentation of an image by choosing an appropriate bandwidth parameter. The number of representative colors in the CCP segmentation is less than the MS segmentation. Moreover, the parameter selection is easier for CCP than for MS. CCP achieves very good boundary F-measures. It does not result into over-segmentation of images. We are going to evaluate a few different metrics for CCP and MS images in the next problem to further support our conclusion that CCP is better than MS.

1.3 Segmentation Result Evaluation

1.3.1 Motivation

To evaluate the performance of a certain algorithm, we need to compare the output results with some ground truth, i.e. the ideal output. In our case, we will use different measures that will represent this comparison. The 5 different metrics that will be considered to evaluate the performance of the above two methods are Segmentation Covering (Cov), Probabilistic Random Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE) and Boundary Displacement Error (BDE). Every metric tells us how good or bad the algorithm performs. We will apply all the 5 metrics on the segmented images obtained from Mean Shift and CCP.

1.3.2 Approach and Implementation

The description of each of the metrics stated above is as follows:

1) Segmentation Covering

   This metric calculates the region-wise covering of the ground truth by segmentation. The overlap between two regions is used for calculating this metric.

$$\boxed{\mathcal{C}(S' \to S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} \mathcal{O}(R, R')}$$ [3]

   C is the covering of segmentation S by a segmentation S' and O(R,R') is the overlap between regions R and R'.

2) Probabilistic Random Index

   This metric is calculated by comparing the compatibility between cluster elements. The PRI between the segmented image (A) and the ground truth (B) is given by the sum of pairs of pixels with the same label in A and B and the ones with different labels in a and B. The total number of pairs of pixels then divides this sum to give PRI.

3) Variation of Information

   This metric measures the distance between two segmentations by considering their conditional entropies. Thus, it measures the amount of randomness in a segmentation that cannot be contained by the other.

4) Global Consistency Error

This metric assumes that one of the segmentation is a refinement of the other and then forces all these refinements to be in the same direction.

5) Boundary Displacement Error

It measures the average displacement error of pixels that lie along the boundary by comparing two segmented images i.e. ground truth and test image.

We have obtained the MS as well as CCP segmentation results from the above two problems. These segmented images can be compared with their ground truths to evaluate how good the performance of our algorithm is. The evaluation of the metrics discussed above has been done in MATLAB.

1.3.3 Results

Metric	CCP of Man	MS of Man	CCP of Rhinos	MS of Rhinos
Cov	0.3563	0.2501	0.4445	0.0567
PRI	0.3990	0.2325	0.5317	0.5064
VoI	3.6034	4.5146	2.4458	6.0366
GCE	0.1235	0.4810	0.2483	0.2630
BDE	12.4375	14.6225	5.0639	16.9761

Figure No. 1.10 Table of Evaluation metrics

1.3.4 Discussion

The segmentation result is better when the Cov and PRI are larger and VoI, GCE, BDE are smaller. If we compare the values in orange with the values in black for the CCP man image and MS man image, We can conclude that CCP performs better than MS. Similarly, for Rhinos CCP gives larger Cov and PRI values whereas smaller VoI, GCE, BDE values than MS. This is how we can say that Contour-guided Color Palette is an efficient segmentation technique and its performance is better than the Mean Shift Segmentation.

2. References

[1] SLIC Superpixels by Radhakrishna Achanta et. al.
[2] Robust image segmentation using CCP by Xiang Fu et. al.
[3] Contour Detection and Hierarchial Image segmentation by Pablo A. et. Al.