Image Segmentation (EN 2550)

Niruhan Viswarupan

1. Thresholding

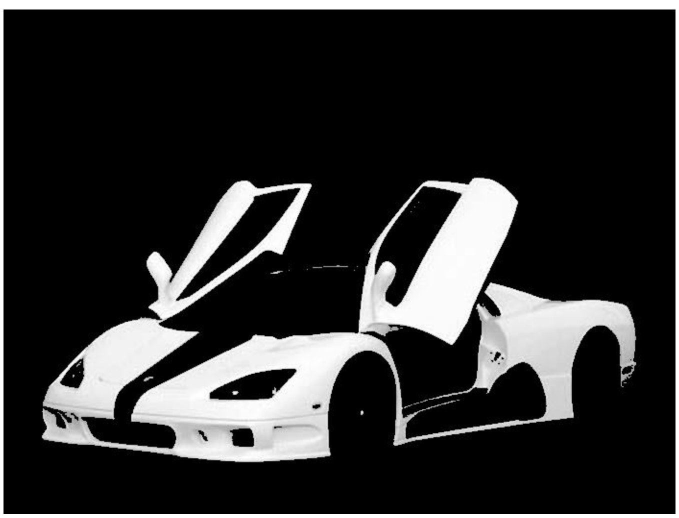
Method: Input image is converted into grayscale image. Every pixel of the gray image is visited and is the value is above a threshold it is passed into the output.

Code snippet:

```
for i=1:h
   for j=1:w
      if (imGray(i, j) > threshold)
         thresholdedImGray(i, j) = imGray(i, j);
      end
   end
end
```

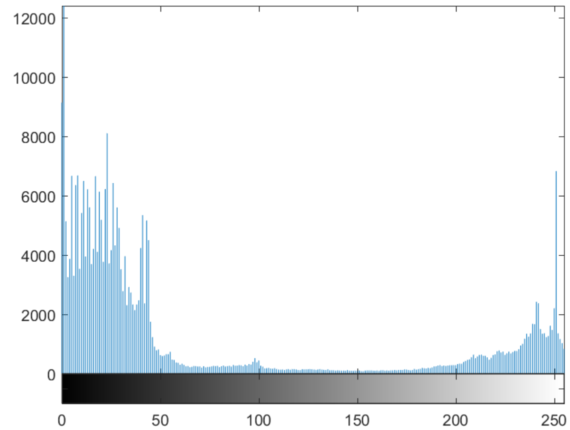
Result:

Original Image



After Thresholding with 150

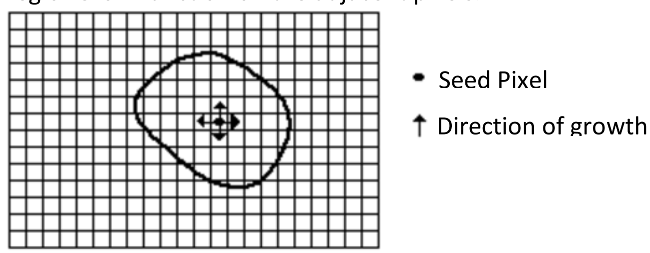
Analysis: This simple technique is powerful but only works when there is a good intensity variation between object of interest and background. The histogram of the image used is as follows. Note that it has two peaks separated by a valley. Around intensity of 150 the valley is minimum. So choosing 150 as threshold will result in a clearly segmented image.



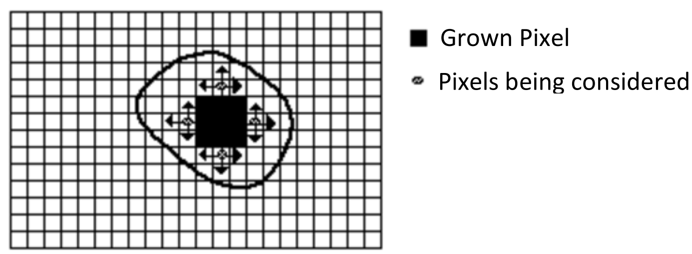
Histogram of the Image

2. Region Growing

Method: Select a seed pixel and pass it to the output. Then check its adjacent 4 pixels and if their intensities differ from the seed's by less than a stipulated threshold then recursively call regionGrow function on the adjacent pixels.



Start Growing a Region



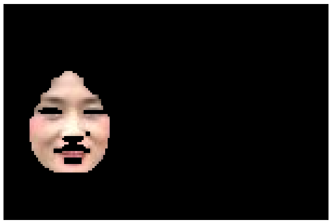
Growing Process after a few Iterations

Code Snippet:

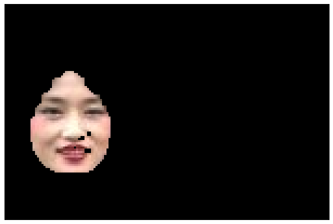
```
if (x-1>0)
   if (abs(image(x, y)-image(x-1,y))<threshold)
      %output(x-1,y)=-2;
      newSeed = [x-1, y];
      regionGrow(newSeed, image, threshold);
   end
end
```

Result:

Original Image

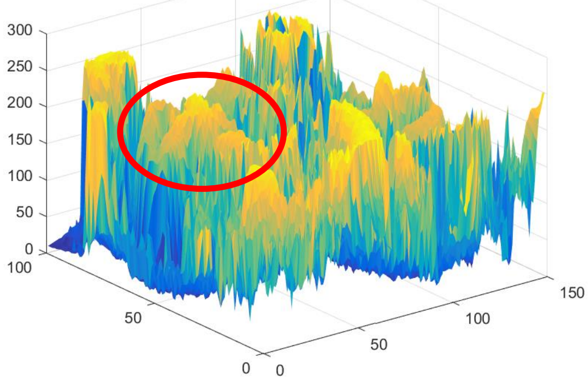


Region Grown with One Seed (Threshold = 20)



With Multiple Seeds at Dark Regions in The face

Analysis: Note that when using a single seed the dark regions inside the object of interest are not passed onto the output. We could eliminate it by selecting multiple seeds. Note the depressions inside the red circle in the following image.



3D Plot of Intensities of the Input

3. Mean Shift

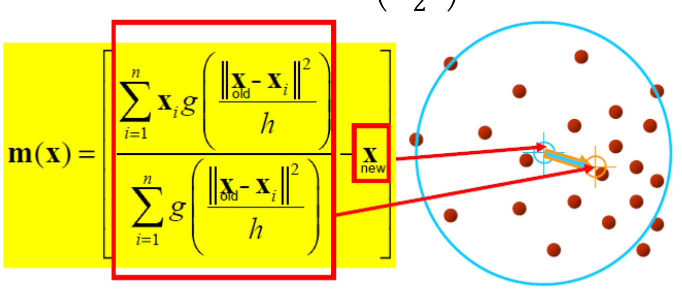
Method: The input RGB image is converted to CIELUV color space because clustering in RGB will result in radical color shifts. Mean shift clustering is done on a 5D space containing 3 dimensions from LUV and 2 spatial dimensions of the image using the following kernel.

$$K_{h_s h_r}(x) = \frac{C}{h_s^2 h_r^p} k\left(\left\|\frac{x^s}{h_s}\right\|^2\right) k\left(\left\|\frac{x^r}{h_r}\right\|^2\right)$$

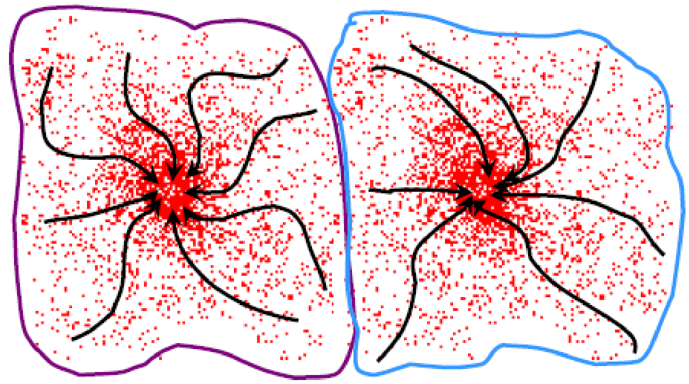
Where, p = dimensionality of range (for LUV p = 3) h_s = spatial parameter, h_r = range parameter, C = normalization constant, x^s = spatial part of x and x^r = range part of x. For k(x) Epanachikov profile (k_E) or truncated normal (k_N) profile gives satisfactory performance.

$$k_E(x) = \begin{cases} 1 - x, & 0 \le x \le 1 \\ 0, & x > 0 \end{cases}$$

$$k_N(x) = exp\left(-\frac{1}{2}x\right)$$



Calculating Mean Shift in 2D



Finding Attracting Basins by Mean Shift Clustering in 2D

The mean shift filtering algorithm: Let x_i and z_i, I = 1,.....,n, be the d-dimensional input and filtered image pixel.

1. Set j = 1 and $y_{i,1} = x_i$
2. Compute $y_{i,j+1}$ until convergence $y = y_{i,c}$ using,

$$y_{j+1} = \frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}$$

   Where, $g(x) = -k'(x)$
3. Assign $z_i = (x_i^s, y_{i,c}^r)$

The mean shift segmentation algorithm:

1. Run mean shift filtering on all data points and find convergence point z_i
2. Cluster $\{C_p\}_{p=1\ldots m}$ the data points y grouping z_i which are closer than h_s in the spatial domain and h_r in the range domain.
3. For each I = 1,…,n, assign $L_i = \{p \mid z_i \in C_p\}$
4. Optional: Eliminate clusters containing less than M pixels.

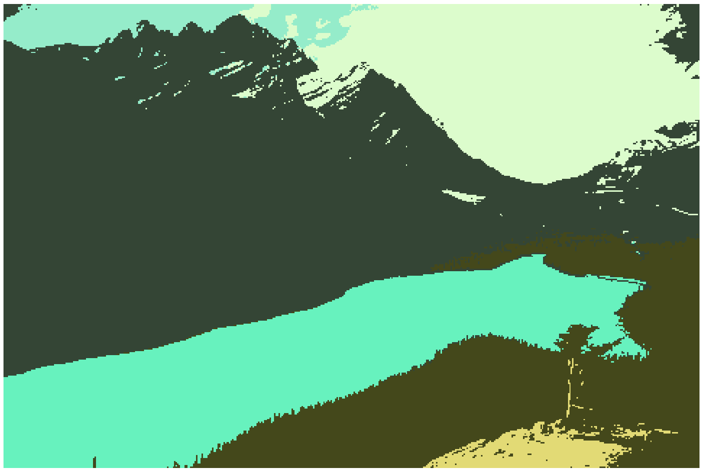
Code Snippet:

```
for i = 1:length(clusterMemberCell)
X(clusterMemberCell {i},:) =
repmat(clusterCenter(:,i)',size(clusterMemberCell {i},2),1);
end
if norm(mean-prevMean) < stopAt
  addTo = 0;
  for cN = 1:amountofClusters
    distance = norm(mean-clusterCent(:,cN));
    if distance < bw/2
      addTo = cN;
      break;
    end
  end
  if addTo > 0
    clusterCent(:,addTo)       =
0.5*(mean+clusterCent(:,addTo));
    preference(addTo,:)    = preference(addTo,:) +
thispreference;
  else    %its a new cluster
    amountofClusters                 = amountofClusters+1;
    clusterCent(:,amountofClusters)       = mean;
    preference(amountofClusters,:)   = thispreference;
  end
```

```
    break;
end
```
Result:

Original Image

Mean Shift Segmented using bw = 0.1 (NumClust = 119)



Mean Shift Segmented using bw = 0.3 (NumClust = 6)

Analysis: The advantages of mean shift segmentation are just two parameters used and the number of modes are found automatically.

Note: For certain parts of the assignment images not provided were used because certain carefully selected images could give very meaningful outputs for the operations.

References:

[1] “Image segmentation” at https://en.wikipedia.org/wiki/Image_segmentation

[2] “Image segmentation” at http://homes.di.unimi.it/ferrari/ImgProc2011_12/EI2011_12_16_segmentation_double.pdf

[3] “Region Growing” at http://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html

[4] “Segmentation - Mean-Shift Algorithm” at http://robot-develop.org/wp-content/uploads/2012/03/seg3.pdf

[5] “Image Segmentation using Mean Shift explained” at http://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained

[6] “Mean Shift Clustering” at https://www.youtube.com/watch?v=Evc53OaDTFc

[7] “Mean Shift: A Robust Approach Towards Feature Space Analysis” by D.Comaniciu and P.Meer on IEEE Transactions on Pattern Analysis and Machine Intelligence

[8] “k-means, mean-shift and normalized-cut segmentation” at https://in.mathworks.com/matlabcentral/fileexchange/52698-k-means--mean-shift-and-normalized-cut-segmentation