

Assignment #1 CS676: Computer Vision and Image Processing

HW : Implementation of Mean Shift algorithm for image segmentation

Group # 15

Abhishek Meena - 12028

Jitendra Kumar Nagar - 12325

Algorithm

Step 1: Mean Shift Clustering:

Let \mathbf{x}_i and $\mathbf{z}_i, i = 1, \dots, n$, be the d -dimensional input and filtered image pixels in the joint spatial-range domain. For each pixel,

1. Initialize $j = 1$ and $\mathbf{y}_{i,1} = \mathbf{x}_i$.
2. Compute $\mathbf{y}_{i,j+1}$ according to (20) until convergence,
 $\mathbf{y} = \mathbf{y}_{i,c}$.
3. Assign $\mathbf{z}_i = (\mathbf{x}_i^s, \mathbf{y}_{i,c}^r)$.

The superscripts s and r denote the spatial and range components of a vector, respectively. The assignment specifies that the filtered data at the spatial location \mathbf{x}_i^s will have the range component of the point of convergence $\mathbf{y}_{i,c}^r$.

The kernel (window) in the mean shift procedure moves in the direction of the maximum increase in the *joint* density gradient, while the bilateral filtering uses a fixed, static window. In the image smoothed by mean shift filtering, information *beyond* the individual windows is also taken into account.

Step2: Mean Shift Image Segmentation

Let \mathbf{x}_i and $\mathbf{z}_i, i = 1, \dots, n$, be the d -dimensional input and filtered image pixels in the joint spatial-range domain and L_i the label of the i th pixel in the segmented image.

1. Run the mean shift filtering procedure for the image and store *all* the information about the d -dimensional convergence point in \mathbf{z}_i , i.e., $\mathbf{z}_i = \mathbf{y}_{i,c}$.
2. Delineate in the joint domain the clusters $\{\mathbf{C}_p\}_{p=1\dots m}$ by grouping together *all* \mathbf{z}_i which are closer than h_s in the spatial domain and h_r in the range domain, i.e., concatenate the basins of attraction of the corresponding convergence points.
3. For each $i = 1, \dots, n$, assign $L_i = \{p \mid \mathbf{z}_i \in \mathbf{C}_p\}$.
4. Optional: Eliminate spatial regions containing less than M pixels.

Results:

Different image segmentation obtained by varying the 3 parameters:

Kernel: Gaussian

- hs = spatial bandwidth of the kernel
- hr = range bandwidth of the kernel
- th = threshold set for the convergence

Image # 42049

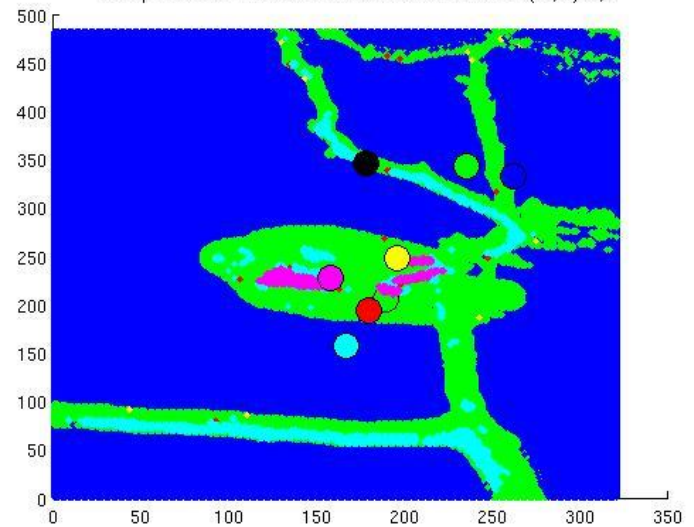


Original image (42049)

Segmented Image (hs, hr)=8,7



data points with their cluster centres Gaussian Kernel (hs, hr)=8,7



Segmented image and data points with their clusters for $(hs,hr) = (8,7)$

Segmented Image $(hs,hr)=12,18$



data points with their cluster centres Gaussian Kernel $(hs,hr)=12,18$



Segmented Image $(hs,hr)=20,8$



data points with their cluster centres Gaussian Kernel $(hs,hr)=20,8$

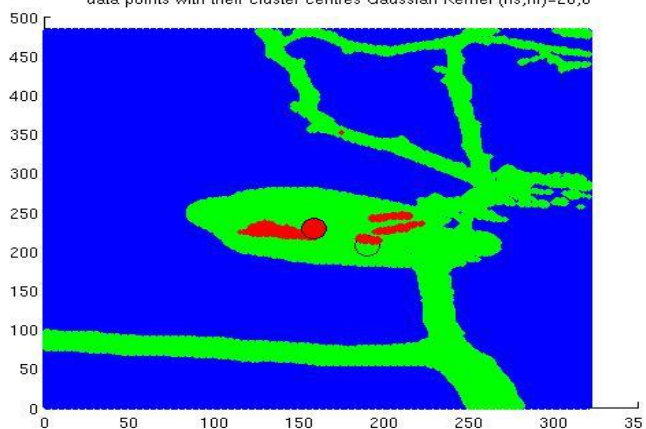
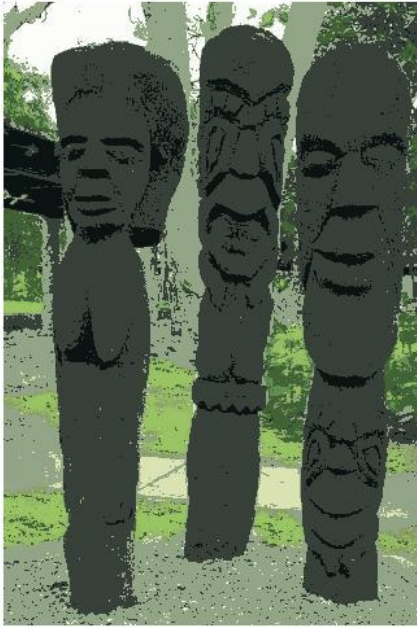


Image # 101085

Original Image



Segmented Image (hs,hr)=8,7



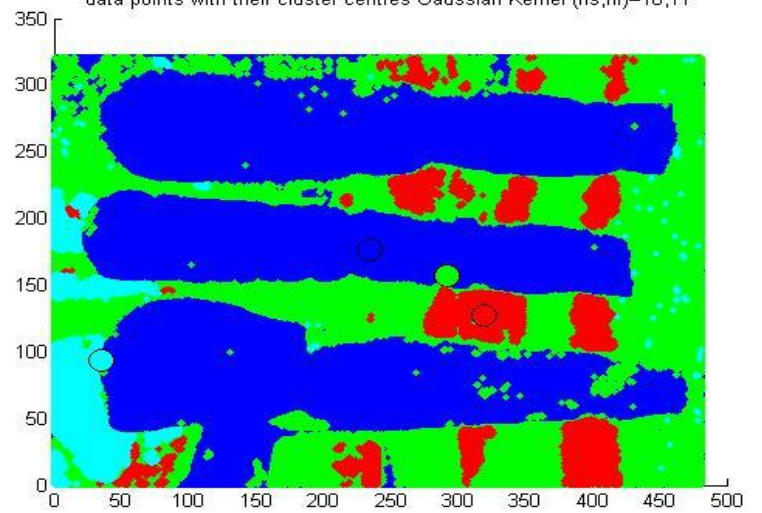
data points with their cluster centres Gaussian Kernel (hs,hr)=8,7



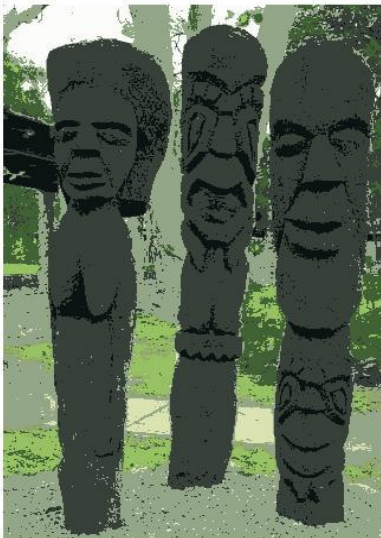
Segmented Image (hs,hr)=18,11



data points with their cluster centres Gaussian Kernel (hs,hr)=18,11



Segmented Image (hs,hr)=8,7



data points with their cluster centres Gaussian Kernel (hs,hr)=8,7



Image # 119082



Segmented Image (hs,hr)=8,20



Segmented Image (hs,hr)=8,7



Segmented Image (hs,hr)=25,30

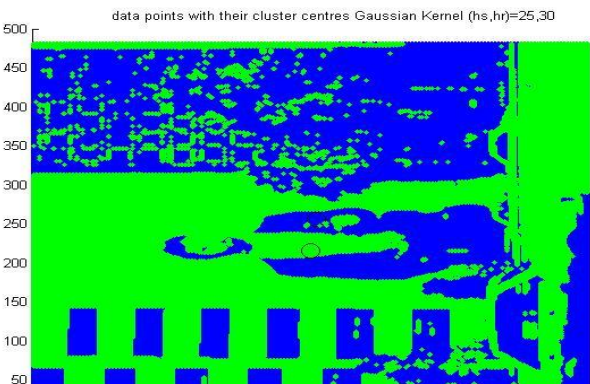
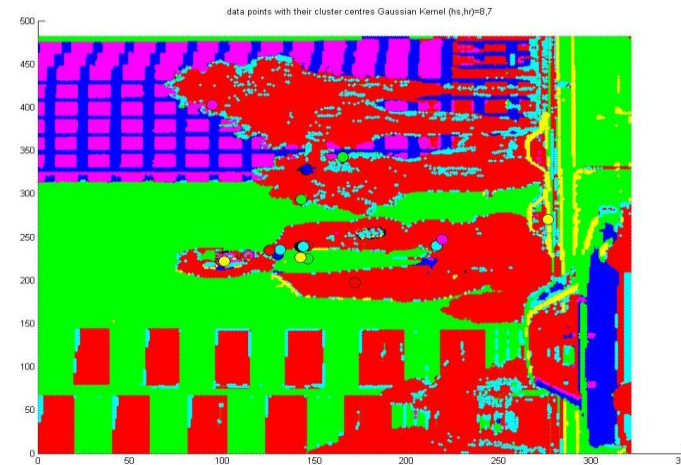
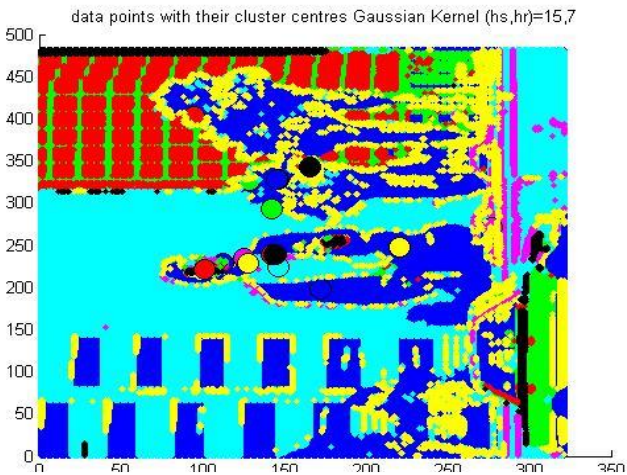
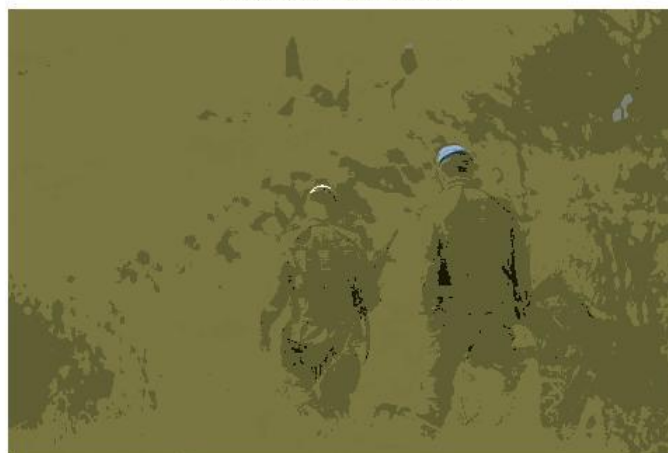


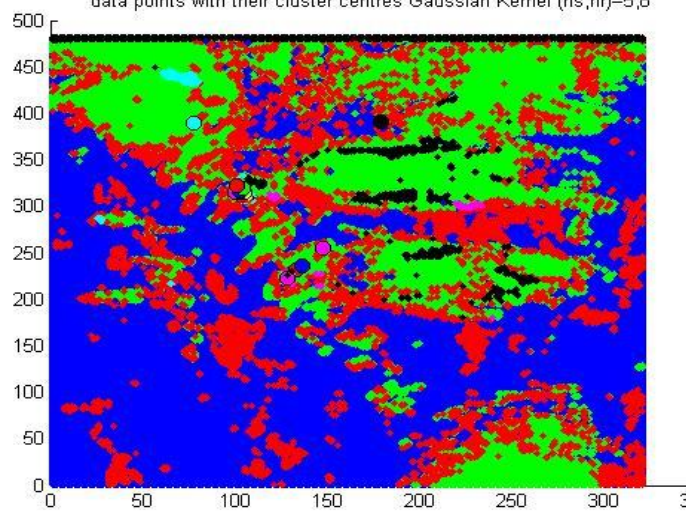
Image # 170057



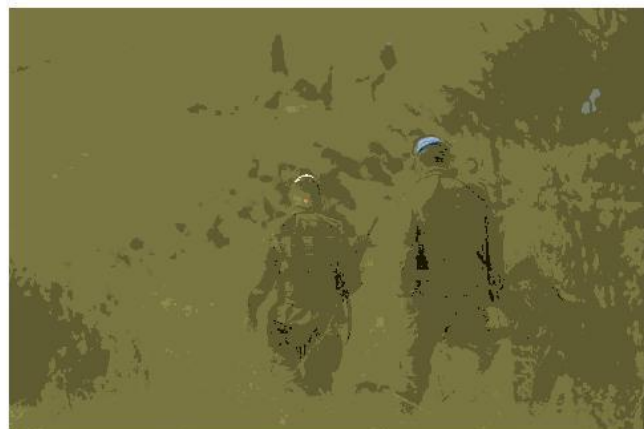
Segmented Image (hs,hr)=5,8



data points with their cluster centres Gaussian Kernel (hs,hr)=5,8



Segmented Image (hs,hr)=10,8



data points with their cluster centres Gaussian Kernel (hs,hr)=10,8

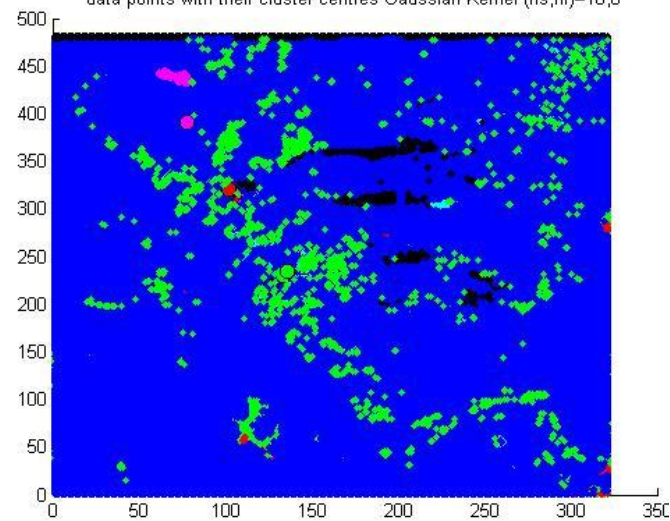
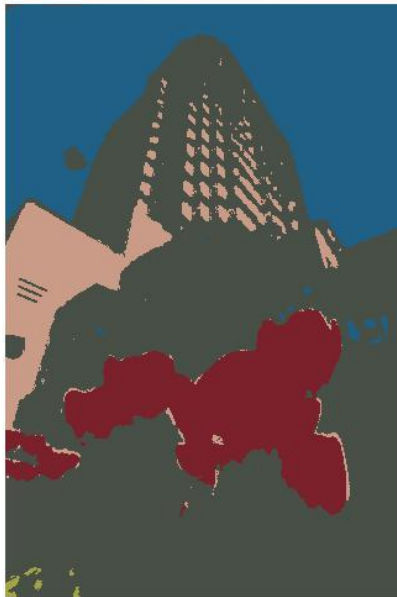


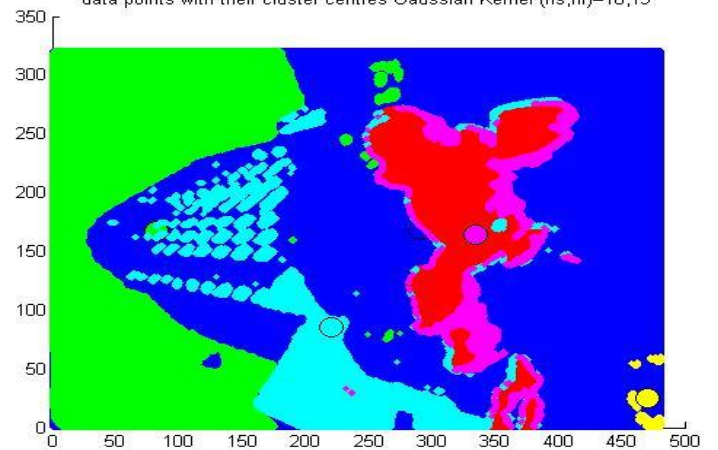
Image # 86000



Segmented Image (hs,hr)=10,19



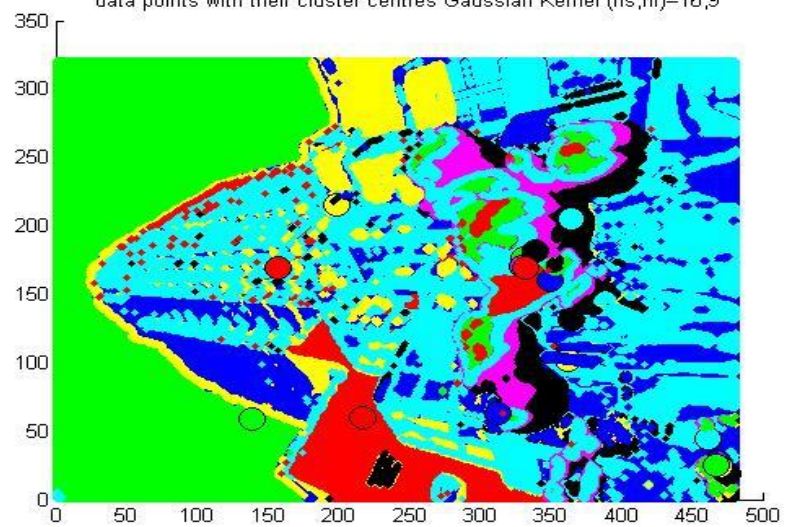
data points with their cluster centres Gaussian Kernel (hs,hr)=10,19



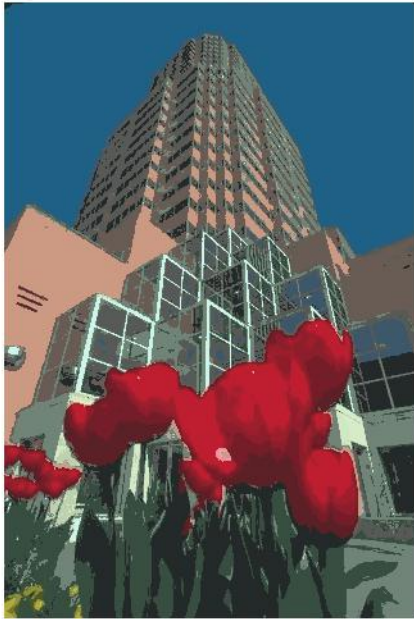
Segmented Image (hs,hr)=16,9



data points with their cluster centres Gaussian Kernel (hs,hr)=16,9



Segmented Image (hs,hr)=10,9



data points with their cluster centres Gaussian Kernel (hs,hr)=10,9

