Depth-Map Generation by Image Classification

Team: Phoenix

Y V S HARISH - 20171402 K S S Varma - 20171406

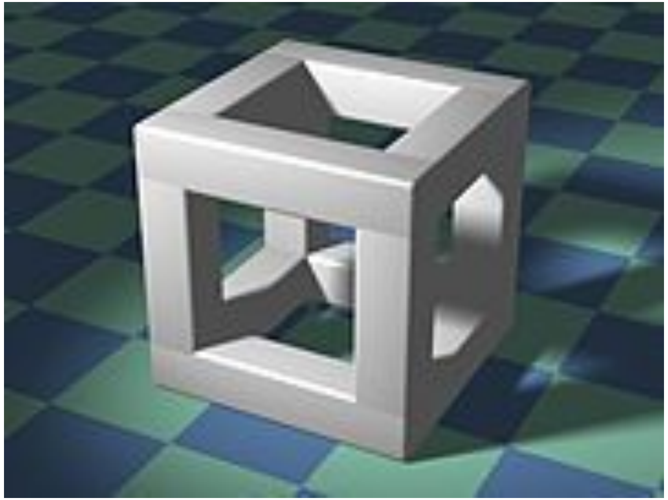
Introduction

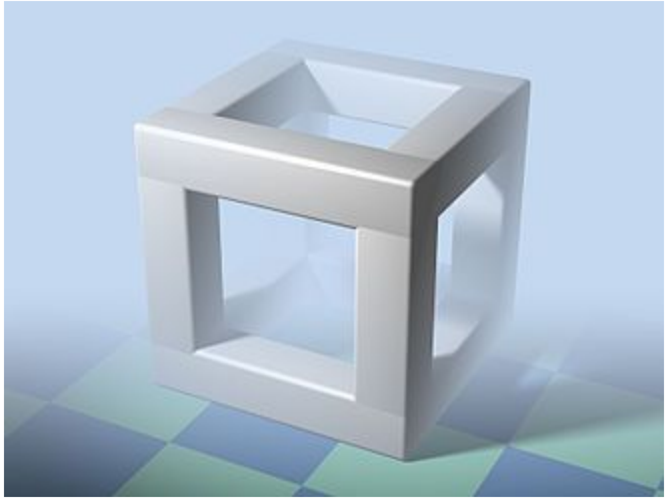
Depth Map

Estimation of depth map from a stereo camera has always been easy as we get the depth value (z) from the trigonometric equations , what we deal here in this project is estimating the depth from a single view camera (monocular) taking some heuristic conditions .

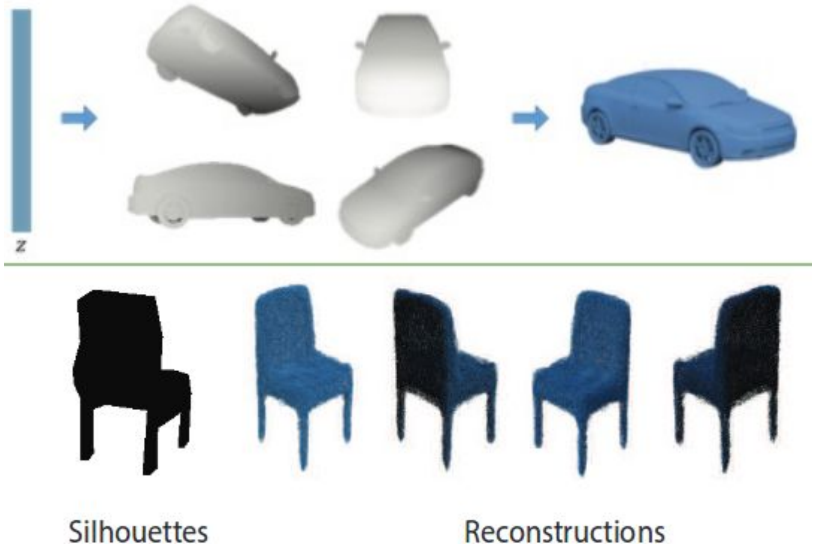
Motivation - Uses

Motivation - Uses





Motivation - Uses



Motivation - Requirement

What are we looking at ??.....

Actual Image

Final Depthmap

Implementation

Overview

Series of steps

- Color-based segmentation
- Rule-based regions detection
- Image classification
- Qualitative depth map

Implementation ::

Segmentation

The color-based segmentation identifies chromatically homogeneous regions.

Region Detection

The semantic region detection can be based on color-based rules aimed to characterize specific regions.

Image Classification

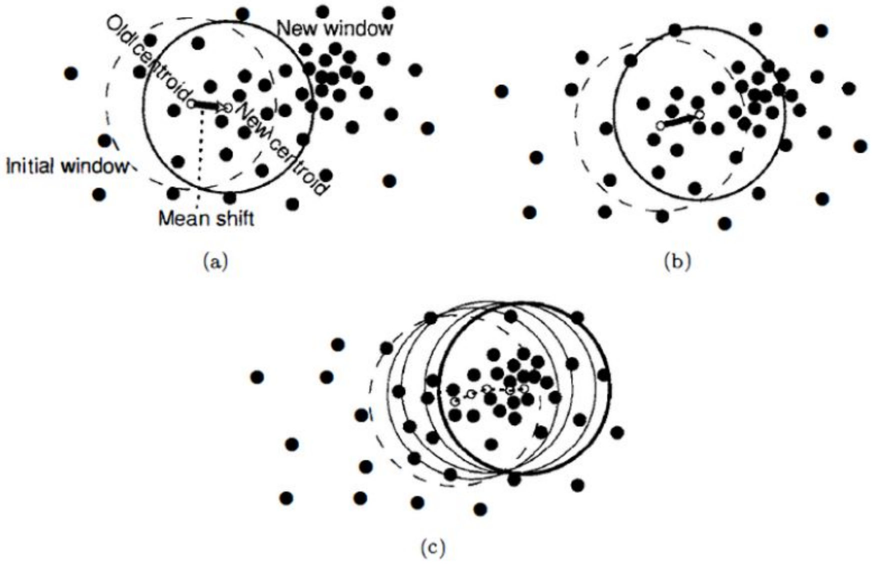
The classifier classifies the image into one of the following categories: Outdoor/Landscape, Outdoor with geometric elements, and Indoor

Segmentation

The mean shift algorithm able to group together pixels depending on their interdependency.

It generates a color segmented image in RGB format where the chromatic values of each identified region, are directly related to the original chromatic values.

Mean Shift Algorithm

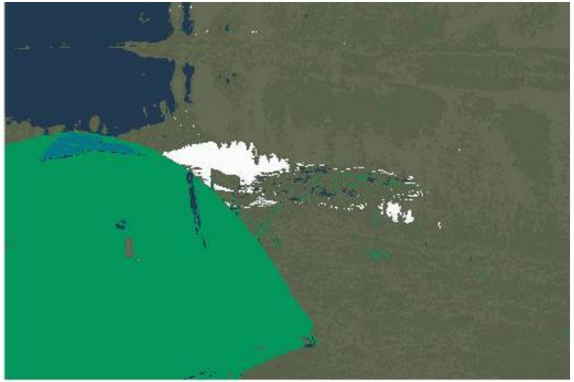


Algorithm ::

- Initialize random seed points.
- Define a window radius.
- For points which are inside this distance threshold (radius) compute new mean.
- Repeat the above step until there is convergence that is the new mean is same as the old mean.

Segmentation - Results On our images

Segmentation - Results



Results : On other Images

Implementation

Segmentation

The color-based segmentation identifies chromatically homogeneous regions.

Region Detection

The semantic region detection can be based on color-based rules aimed to characterize specific regions.

Image Classification

The classifier classifies the image into one of the following categories: Outdoor/Landscape, Outdoor with geometric elements, and Indoor

Regions Detection

The identification of semantic regions in a generic image is a crucial step needed to obtain a robust image classifier.

Regions are detected depending on the color based rules which are designed depending on some experiments.

The following regions have been taken up :

1. **Sky**
2. **Far Mountain**
3. **Near Mountain**
4. **Land**
5. **Other**

Regions Detection

Sky

Rules to be classified as Sky:

- Intensity > 0.65
- Blue > 160
- Green > 70
- Blue > Green -15
- Blue > Red - 15

Regions Detection

Far Mountain

Rules to be classified as Far Mountain:

- Intensity > 0.1
- 20 <= Blue <=160
- Green => 20
- Blue >= Green AND Blue >= Red

OR

- 0.4 < Intensity < 0.8
- 80 <= (Red, Blue, Green) <=160

Regions Detection

Near Mountain

Rules to be classified as Near Mountain:

- Intensity > 0.45
- Blue <=100
- Red >=100
- Red >= Blue AND Red >= Green

OR

- 0.15 < Intensity < 0.65
- Red, Blue, Green <= 120
- Green +10 >= Blue
- Green +10 >=Red

Regions Detection

Land

Rules to be classified as Land:

- Intensity > 0.5
- 100 <= Red <=200
- 100 <= Green <=190
- 140 <= Blue <=180

OR

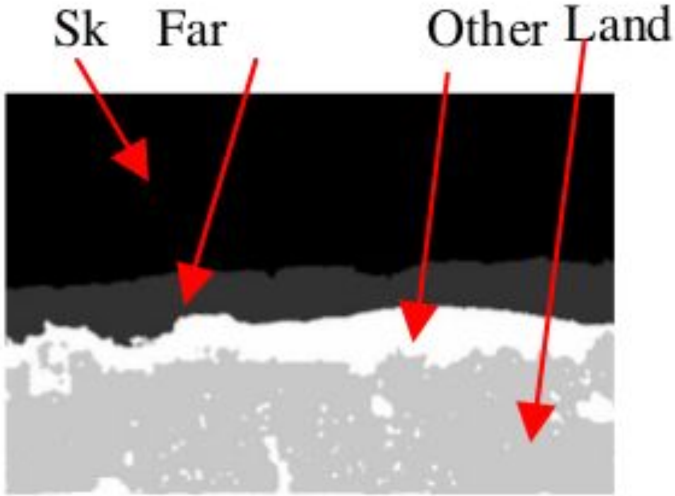
- Intensity > 0.4
- Blue <=100
- Red <= 200
- Green >= Blue
- Green >= Red

Regions Detection

Other

If NONE of the above conditions are satisfied then we classify it as "Other"

Example output for Region Detection ::



Implementation

Segmentation

The color-based segmentation identifies chromatically homogeneous regions.

Region Detection

The semantic region detection can be based on color-based rules aimed to characterize specific regions.

Image Classification

The classifier classifies the image into one of the following categories: Outdoor/Landscape, Outdoor with geometric elements, and Indoor

Image Classification

Region	Labels
Sky	s
Farthest Mountain	m
Far Mountain	m
Near Mountain	m
Land	l
Other	x

Regions are detected based the previous pipeline that is Region detection.

The jump in the discrete assigned gray values is noted and then the quantitative analysis on output sequence is compared some defined threshold values.

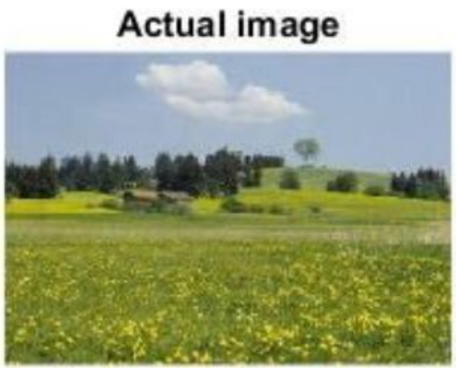
Steps :

1) Sequences and jumps detection for each sample column. A jump is the number of regions encountered in the examined column.

2) Each sequence is compared to the set of typical sequences. If the sequence is recognized and the jumps number is smaller than a threshold J_B, then the value N_1 is increased, where N_1 represents the number of accepted sequences. If the sequence isn't a typical landscape sequence or if the jumps number is bigger than J_B then the sequence is rejected.

3) Final classification. The image is classified as Outdoor if the value of N_1 is bigger than $R_1 N$, where N is the number of analyzed sequences and R_1 is a threshold in [0,1]. Otherwise if the number of sequences with the first region Sky is bigger than $R_2 N$, where R_2 is another threshold in [0,1] the image is classified as Outdoor with geometric appearance else it is classified as Indoor.

Image Classification - Outdoor example



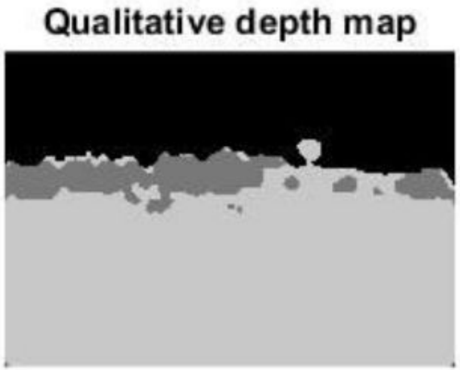


Image Classification - Outdoor with geometric elements

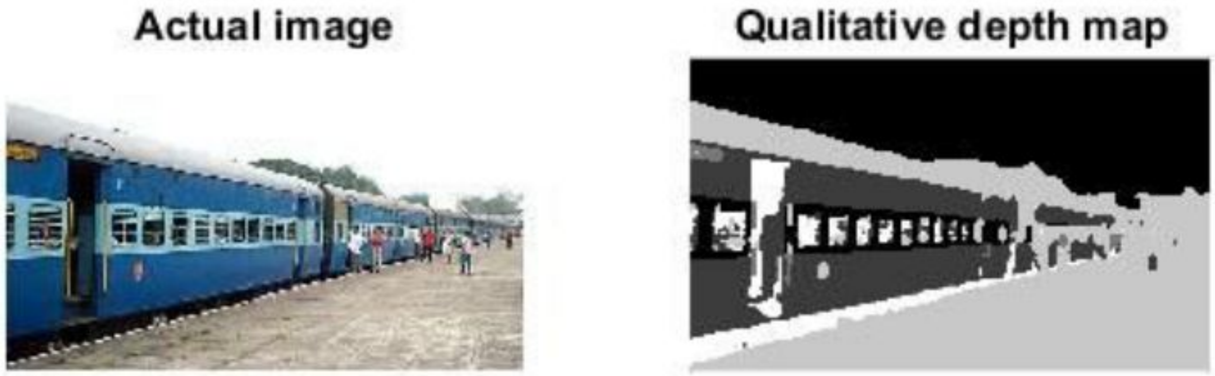
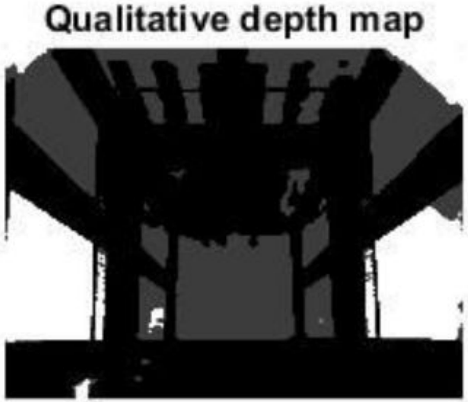


Image Classification - Indoor Example



Vanishing Lines detection

Regions are detected based the previous pipeline that is Region detection.

The jump in the discrete assigned gray values is noted and then the quantitative analysis on output sequence is compared with some defined threshold values.

Steps :

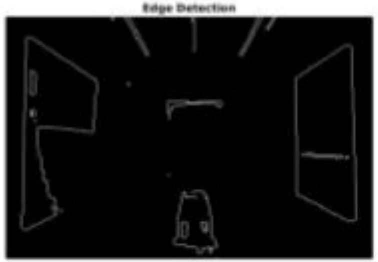
If the image is outdoor without geometric elements vanishing we consider vanishing points as the lowest point of intersection between land U other and other regions (x_b,y_b) the coordinates of the Vanishing point s fixed to $(W/2,y_b)$.

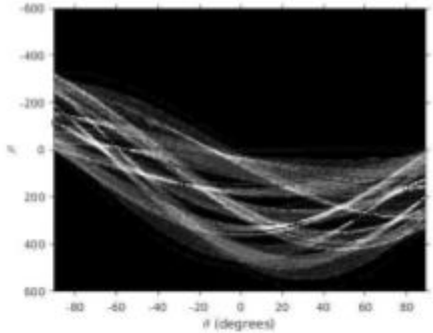
Otherwise :

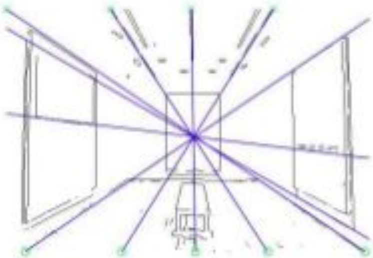
1) Edge detection using a 3X3 Sobel masks . The resulting images I_{sx} , I_{sy} are the normalized and converted into a binary image I E , eliminating reductant information .
2) Noise reduction of I_{sx} and I_{sy} using a standard low-pass filter 5X5 .
3) Detection of line through lines by keeping voting threshold.
4) Compute of intersection between each pair of straight lines .
5) The Vanishing Point is chosen as the intersection point with greatest number of intersections around it, while the vanishing lines detected are the main straight lines passing close to Vanishing Point .

Outputs ::


Grayscale Image


Edge Detection


-600
-400
-200
0
200
400
600
-80 -60 -40 -20 0 20 40 60 80
θ (degrees)



Implementation: Depth Map Generation

Vanishing lines detection

This is an important step in the computation of the geometric map where we roughly estimate the vanishing lines and hence the vanishing point.

Gradient Plane Generation

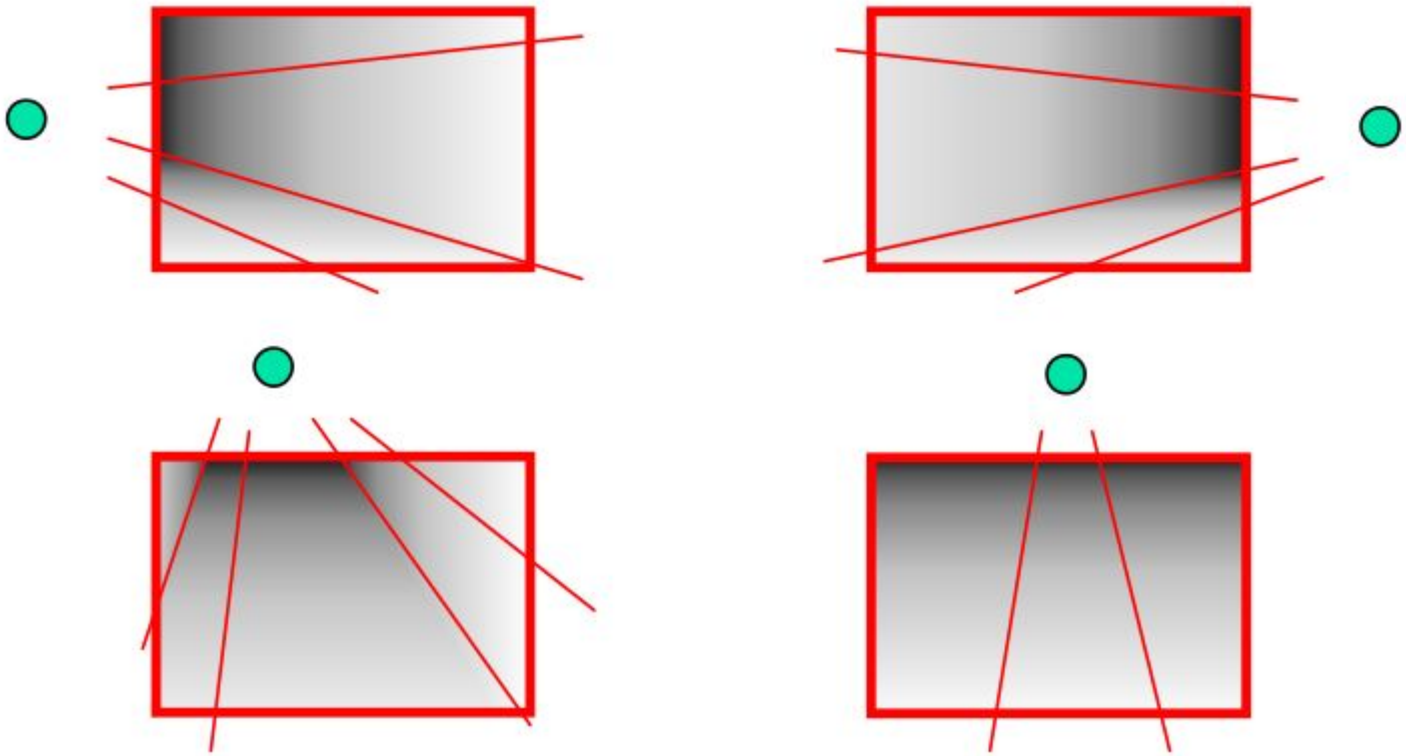
Depending upon the previous output of the vanishing point coordinates we impose a set of conditions in order to get the location of the vanishing point.

Depth map assignment

Here we decide on the geometric depth map based on the first 2 steps and hence assign interpolated gray values to the image depending on their depths.

Gradient Plane detection

Here depending on some conditions we classify the direction of the Vanishing point.



Steps :

$Xvp \leq 0$ AND $(H-1/W-1) * Xvp < Yvp < -(H-1/W-1) * Xvp + H-1$	**Left Case**
$Xvp \geq W-1$ AND $-(H-1/W-1) * Xvp + H-1 < Yvp < (H-1/W-1) * Xvp$	**Right Case**
$Yvp \leq 0$ AND $(W-1/H-1) * Yvp \leq Xvp \leq (W-1/H-1) * (H-1-Yvp)$	**Up Case**
$Yvp \geq H-1$ AND $(W-1/H-1) * (H-1-Yvp) \leq Xvp \leq (W-1/H-1) * Yvp$	**Down Case**
$0 < Xvp < W-1$ AND $0 < Yvp < H-1$	**Inside Case**

(X_{vp}, Y_{vp}) = Vanishing Point $\quad$ (H, W) = Height, Width

Implementation: Depth Map Generation

Vanishing lines detection → **Gradient Plane Generation** → **Depth map assignment**

Vanishing lines detection

This is an important step in the computation of the geometric map where we roughly estimate the vanishing lines and hence the vanishing point.

Gradient Plane Generation

Depending upon the previous output of the vanishing point coordinates we impose a set of conditions in order to get the location of the vanishing point.

Depth map assignment

Here we decide on the geometric depth map based on the first 2 steps and hence assign interpolated gray values to the image depending on their depths.

Depth Map Assignment

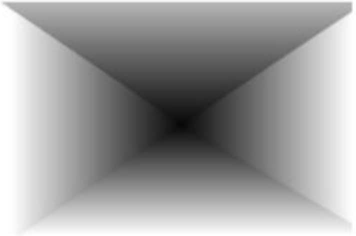
Here depending on some conditions we classify the direction of the Vanishing point.

Steps :

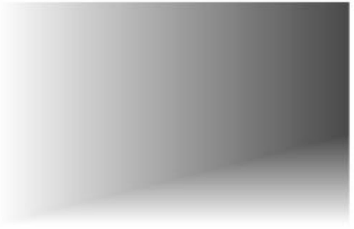
Two main assumptions are used :

- Higher depth level corresponds to lower grey values .
- The vanishing point is the most distant point from the observer(this assumption is almost always true) .

The depth level value is approximated by a linear variations.





Outputs




Outputs

Fusion of Geometric and Qualitative Depth Map :

$M_1(x,y)$ => Qualitative Depth Map $M_2(x,y)$ => Geometric Depth Map

$M(x,y)$ => Final Depth Map

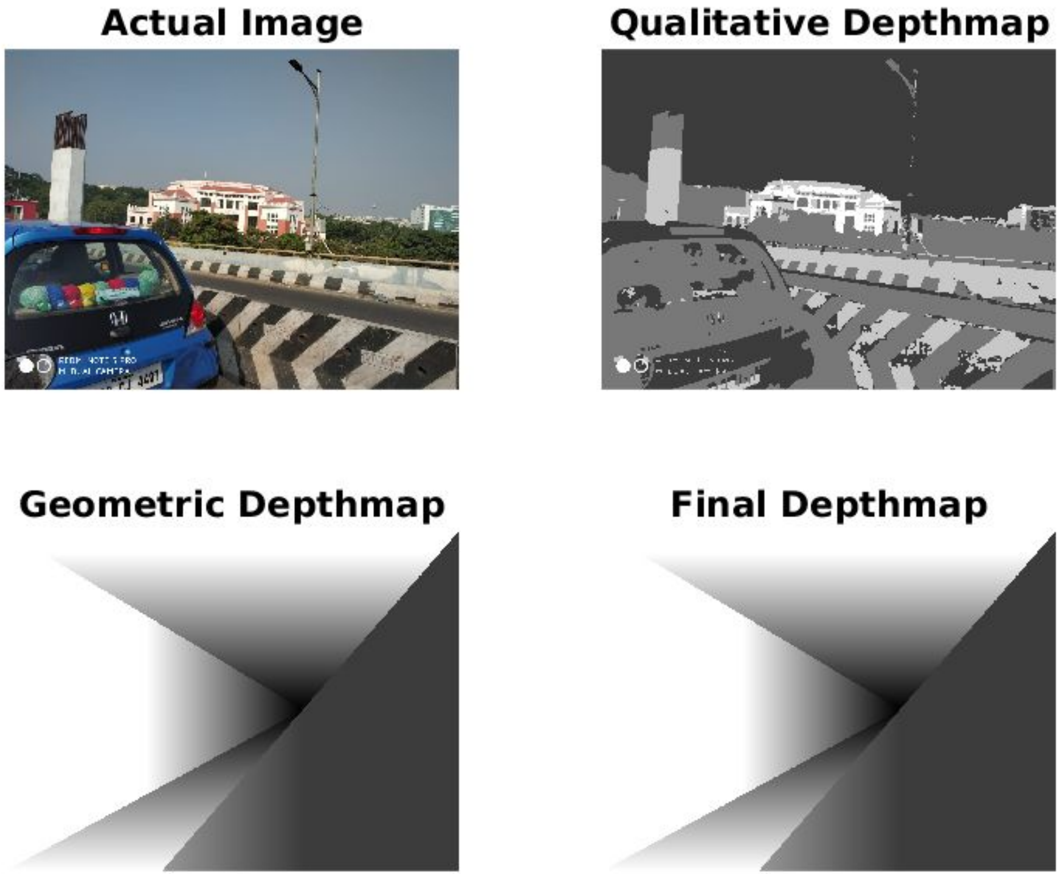
Type	Fusion Operation
Indoor	$M(x,y) = M_2(x,y)$
Outdoor without Geometric Components	$M(x,y) = M_2(x,y)$; $(x,y) \rightarrow$ Land or Other $M(x,y) = M_1(x,y)$; Otherwise
Outdoor with Geometric Components	$M(x,y) = M_1(x,y)$; $(x,y) \rightarrow$ Sky $M(x,y) = M_2(x,y)$; Otherwise

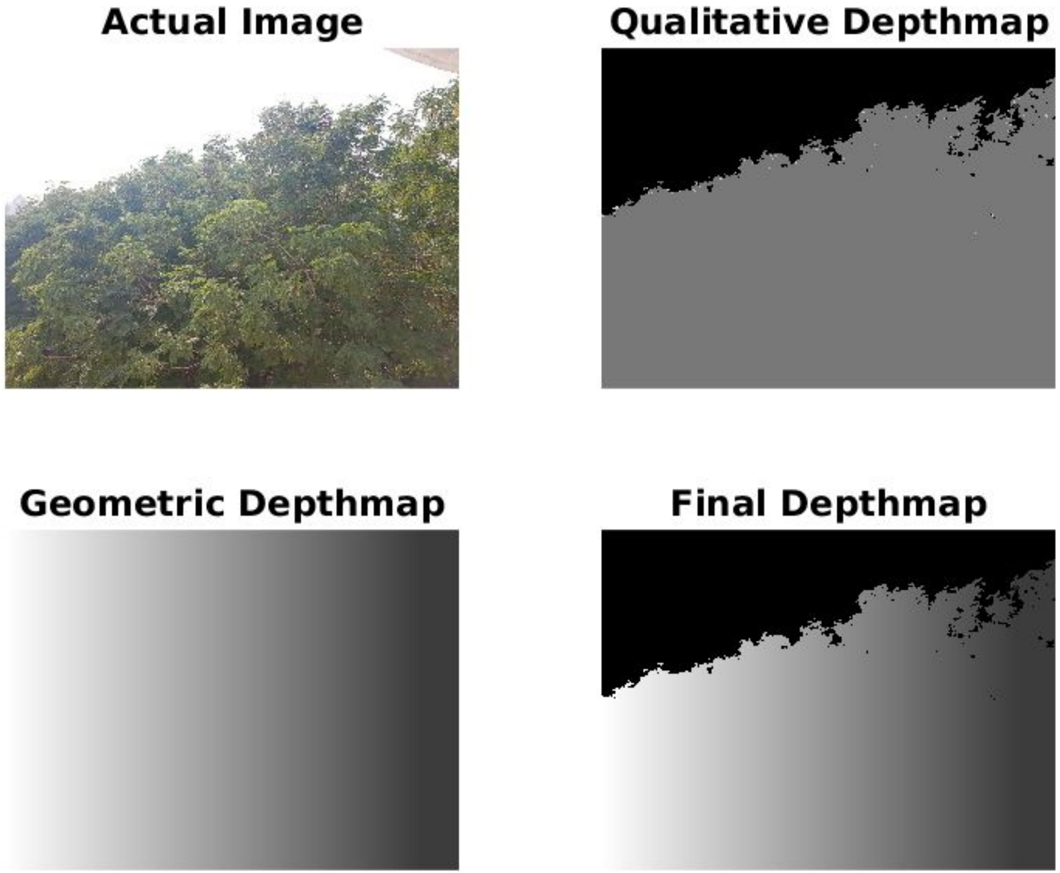
Actual Image

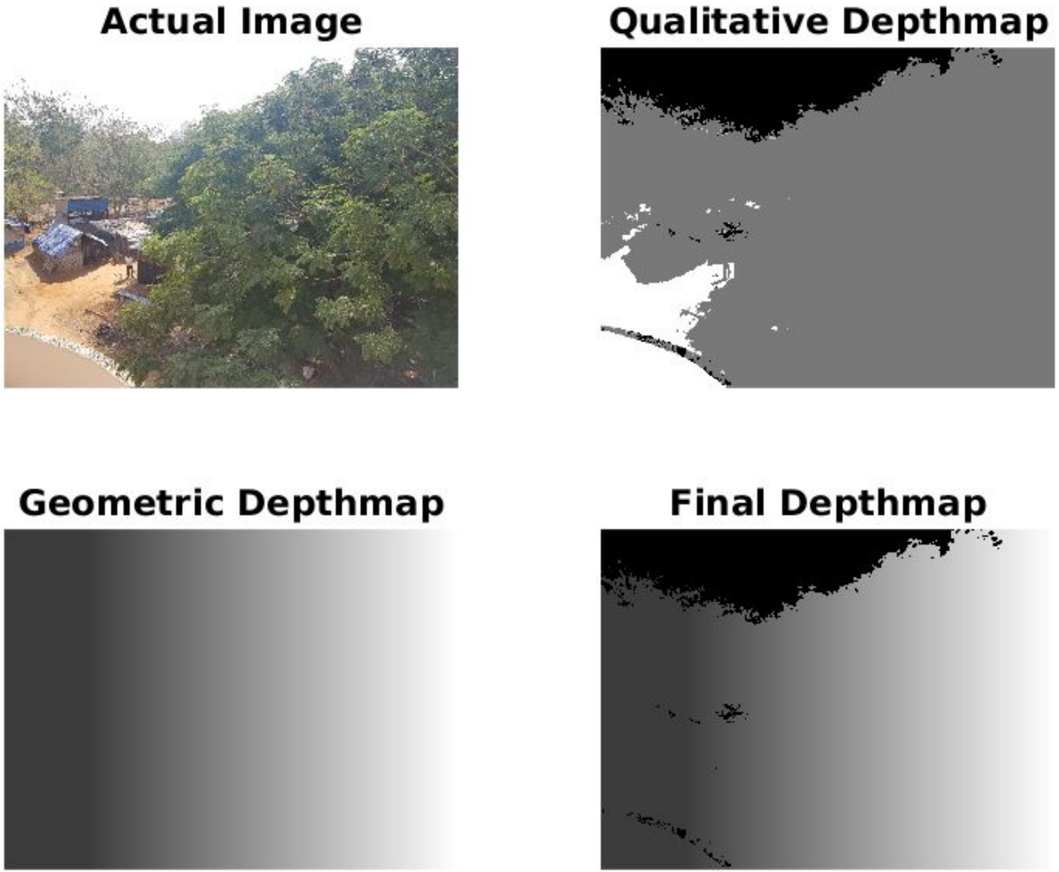
Qualitative Depthmap

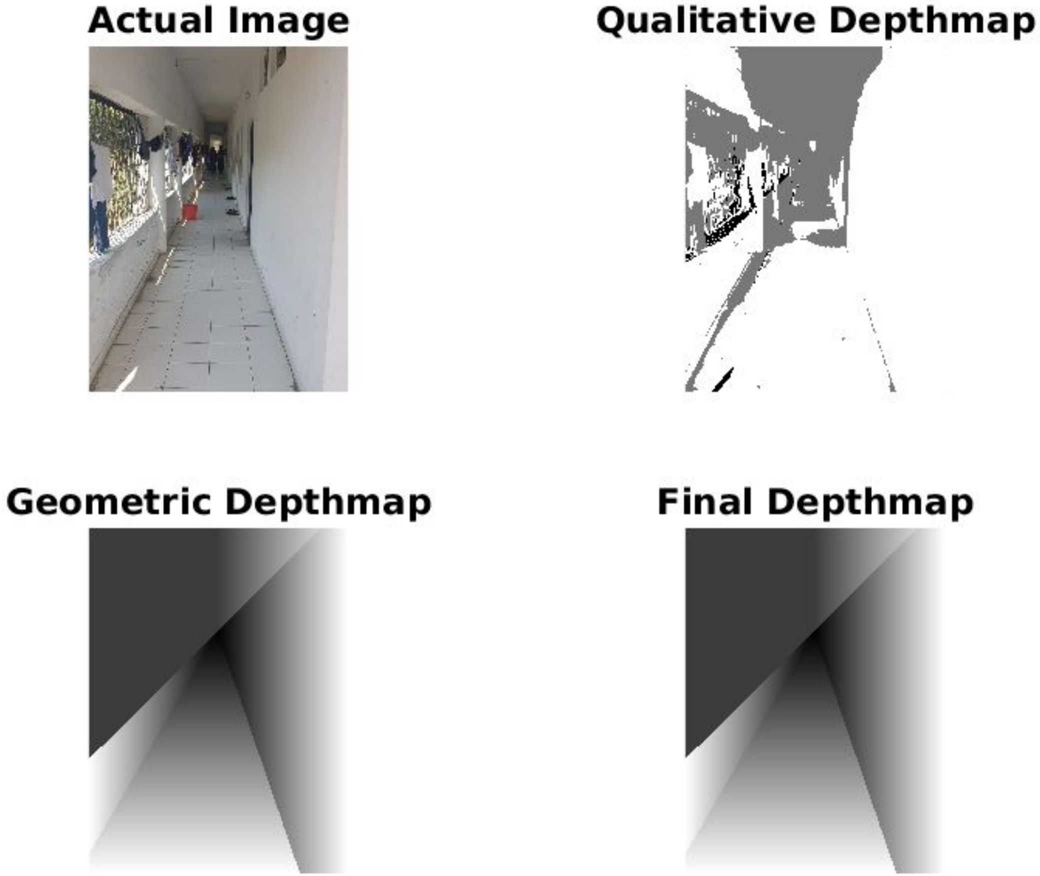
Geometric Depthmap

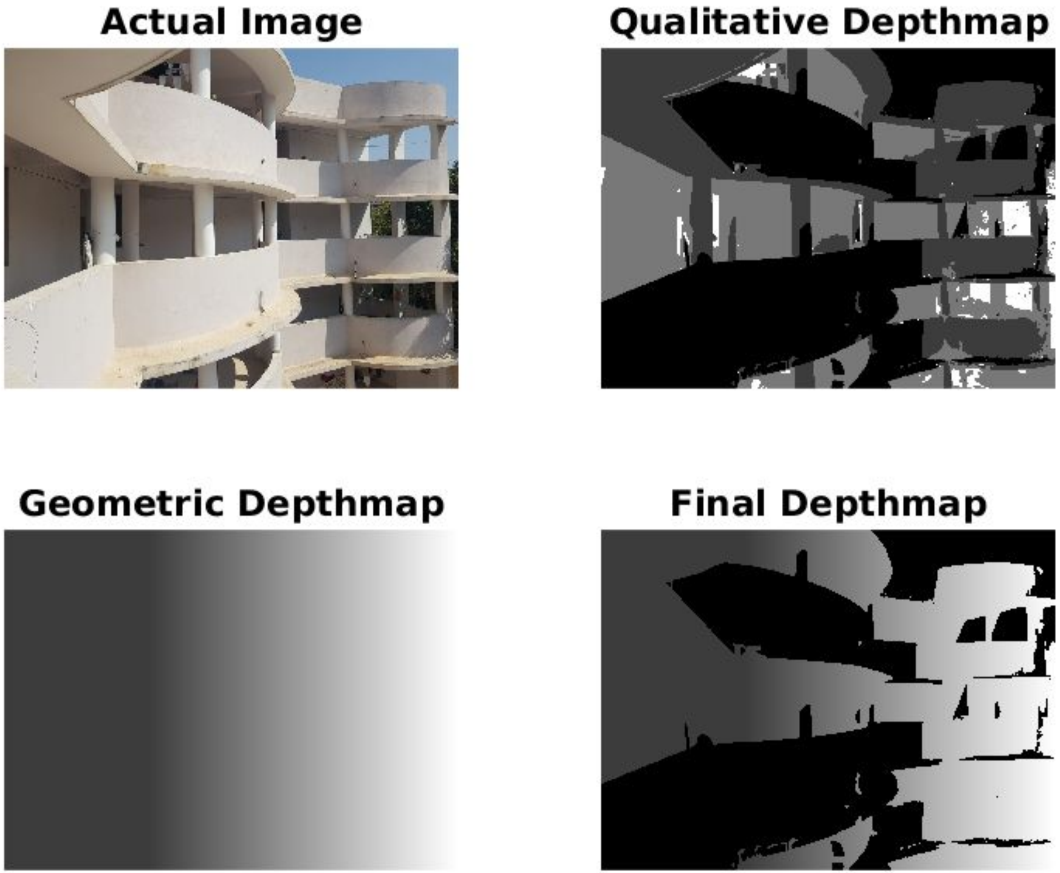
Final Depthmap

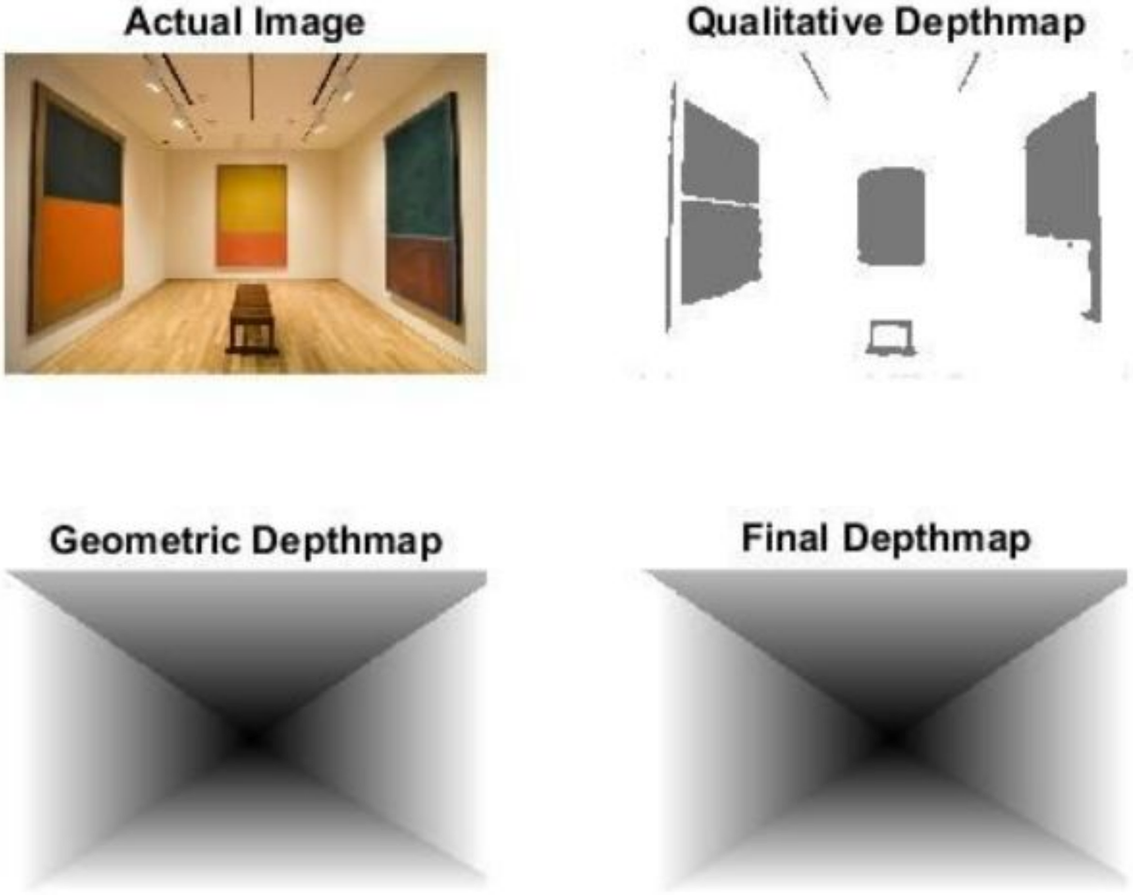

Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap

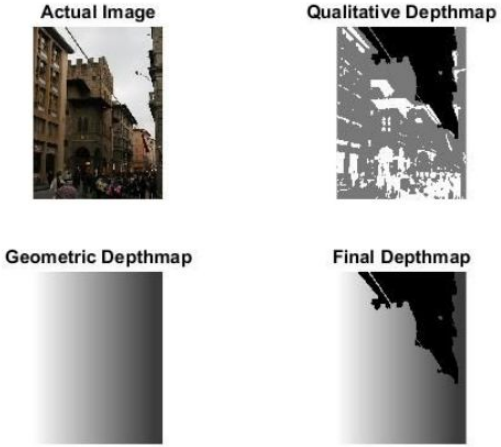

Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap

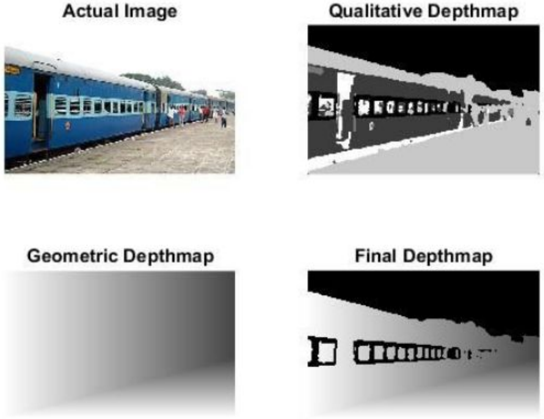

Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap


Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap

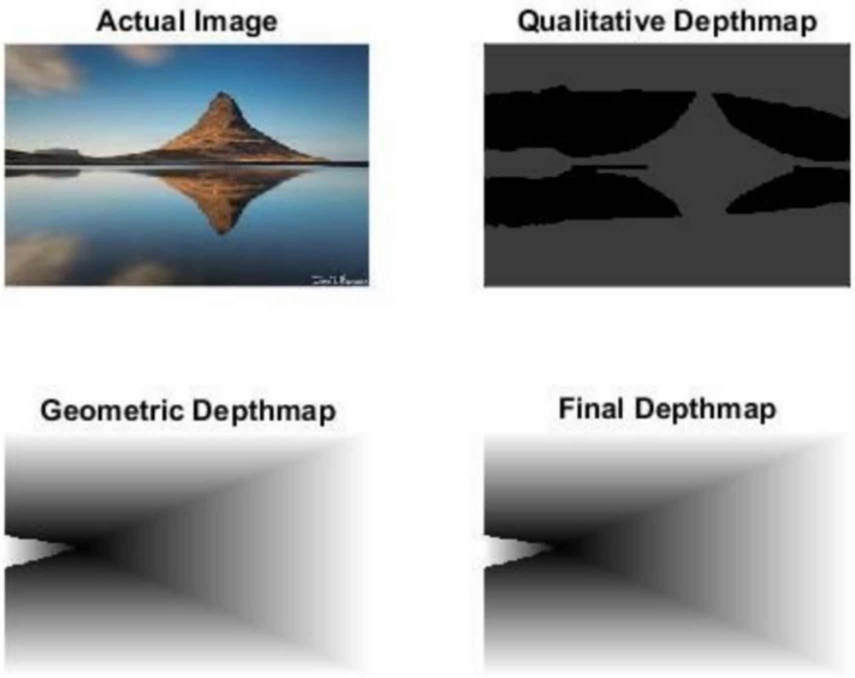

Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap


Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap

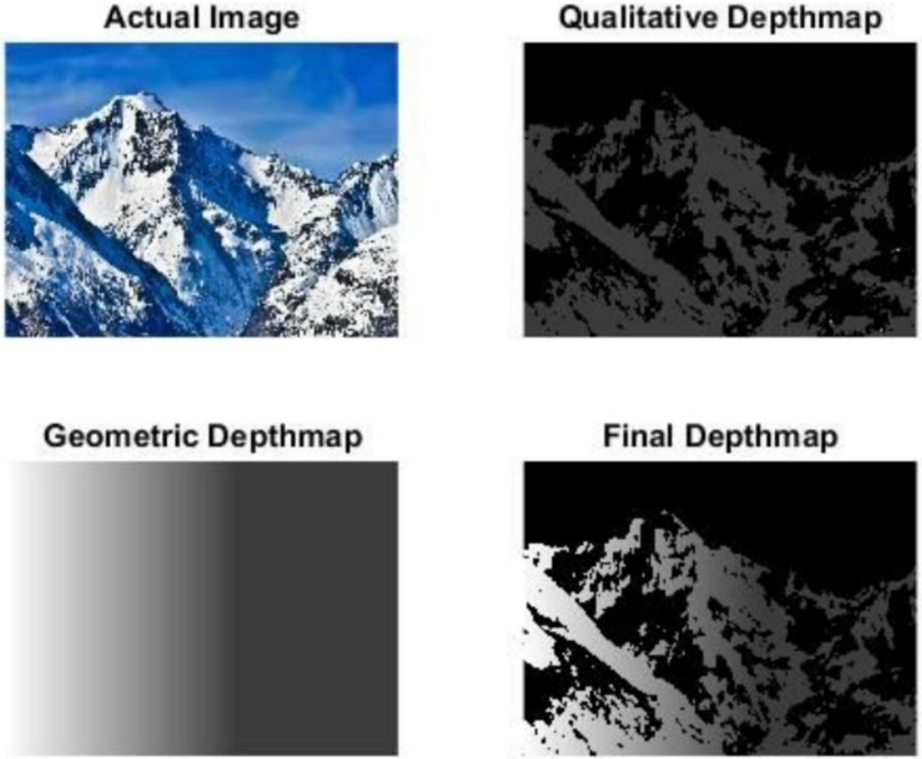

Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap


Actual Image
Qualitative Depthmap
Geometric Depthmap
Final Depthmap

Limitations ::



For this image, we can’t segment the water clearly and also some part of sky region is not detected as sky region. Also, the Geometric depth map is not correct because we don’t have huge train data to estimate the proper heuristics.



In this image, regions aren't properly segmented.

Further Improvements :

Improvement 1

Regions detection could detect a greater number of regions (for example : people and objects in foreground in which a gradient of depth should not be assigned).

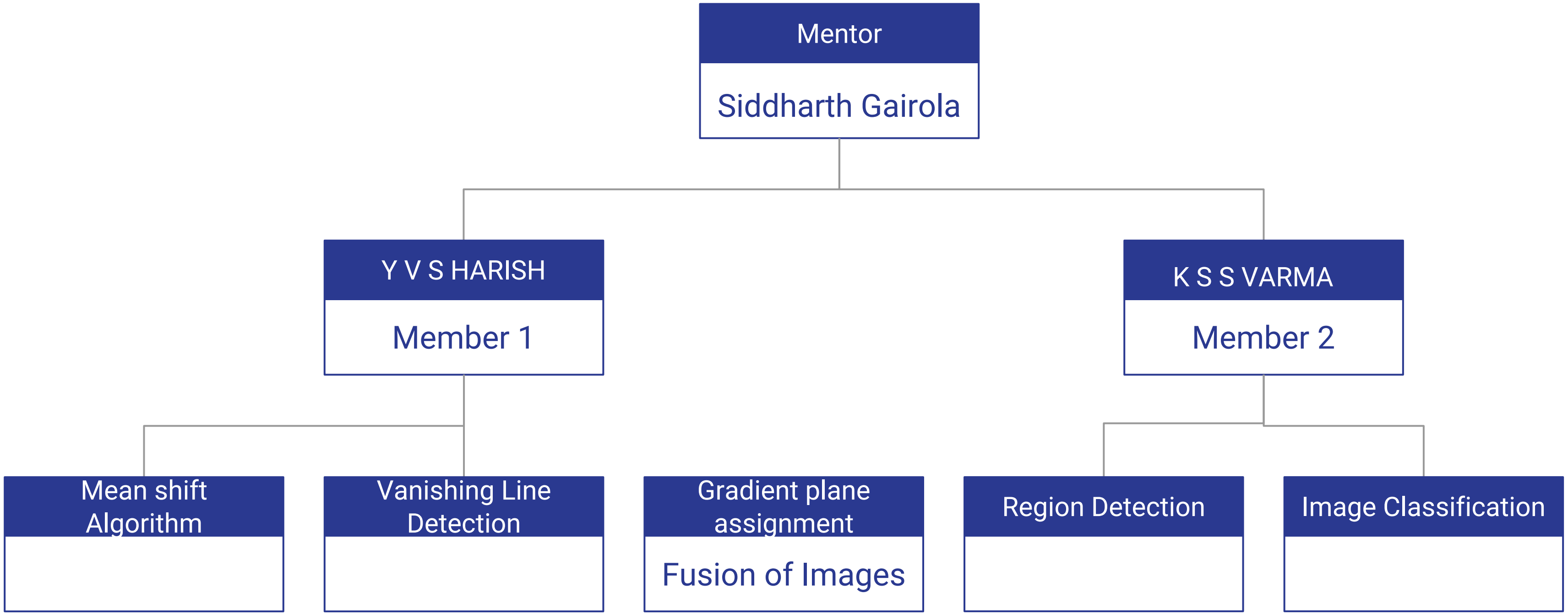
Improvement 2

Vanishing lines detection could detect a possible second vanishing point and its relative vanishing lines.

Improvement 3

We don't have the training data set so we cannot get the appropriate color values which can classify all the regions correctly for all types of images and Geometric Depth Map heuristics.

The team



Mentor
Siddharth Gairola
Y V S HARISH
Member 1
K S S VARMA
Member 2
Mean shift Algorithm
Vanishing Line Detection
Gradient plane assignment
Fusion of Images
Region Detection
Image Classification