STEREO VISION

SUBMITTED BY:

Shivangi

Ruturaj Hagawane

Approach – 1

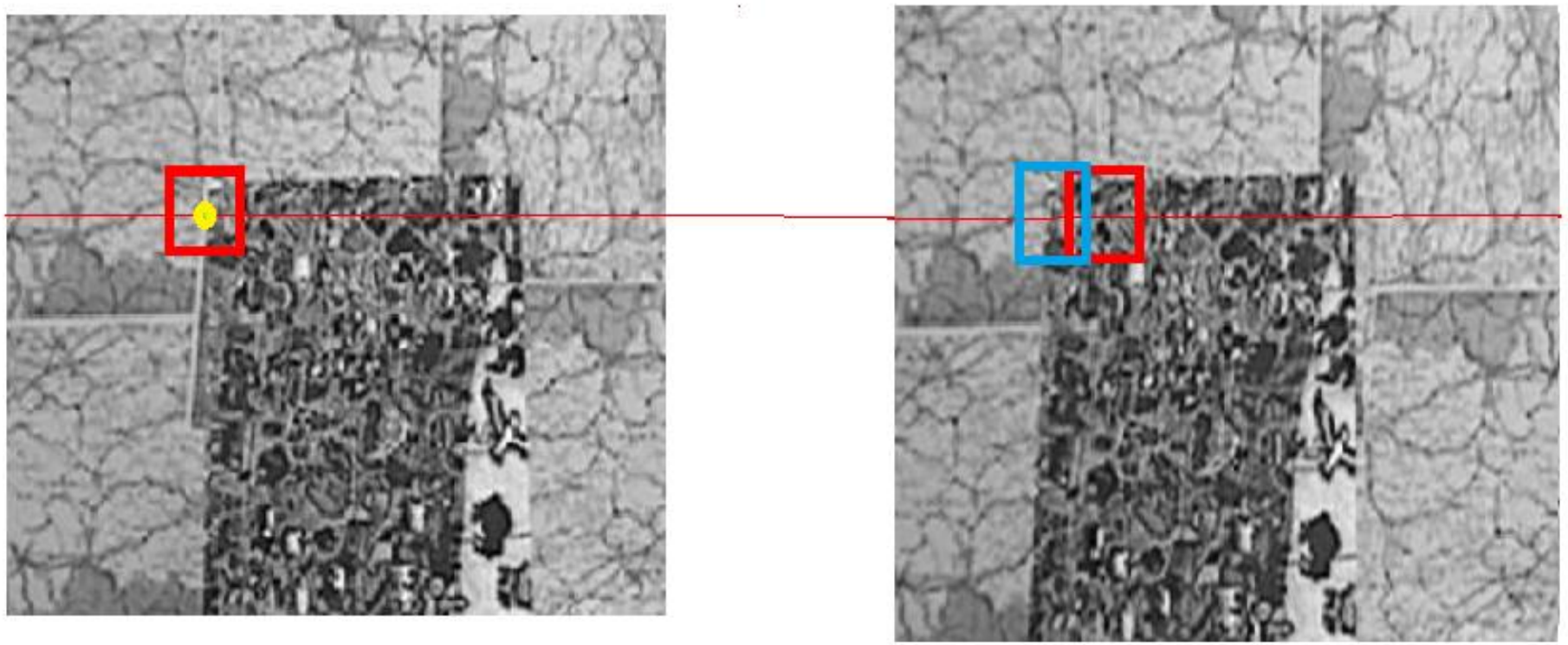
SSD

SUM OF SQUARE DIFFERENCES

Algorithm :

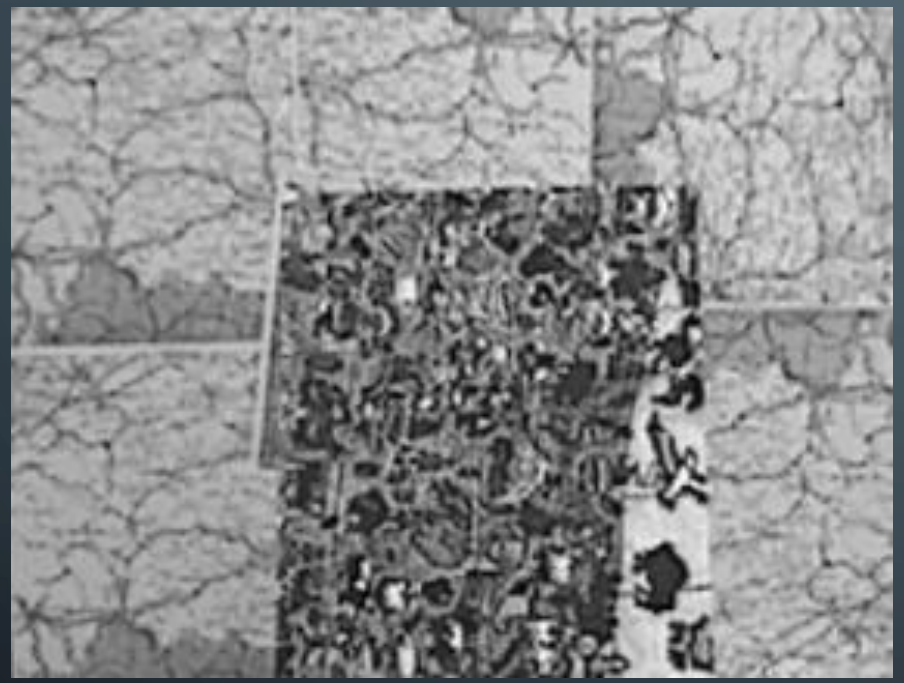
- Input Left image
- Input Right image
- Take a pixel in left image
- Create a window around that pixel
- Find corresponding pixel in right image with window match technique
- Choose the best pixel using SSD

$$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i, y+j) \right)^2$$

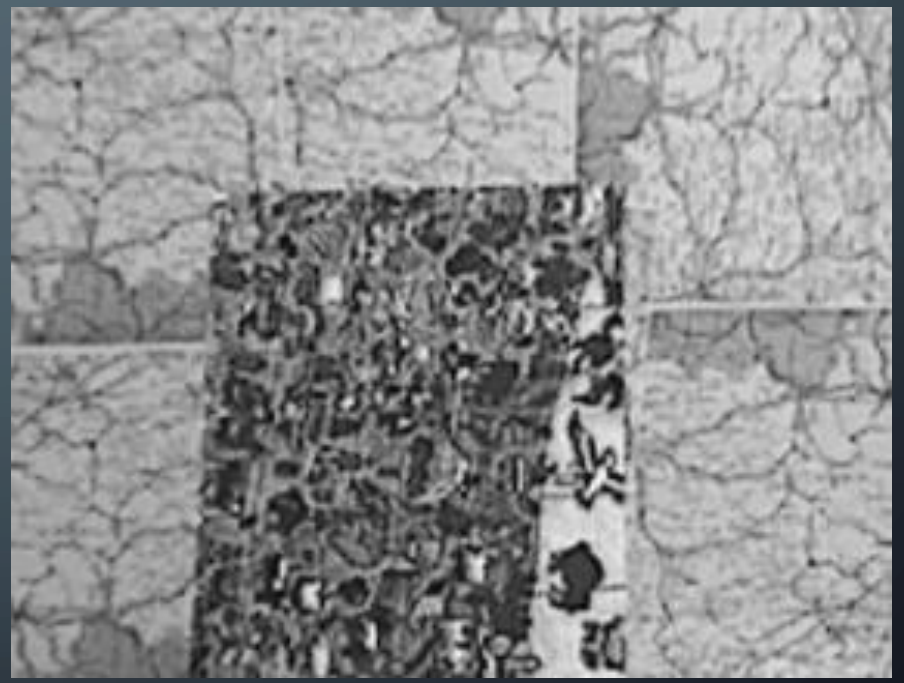


$$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i, y+j)\right)^2$$

MAP

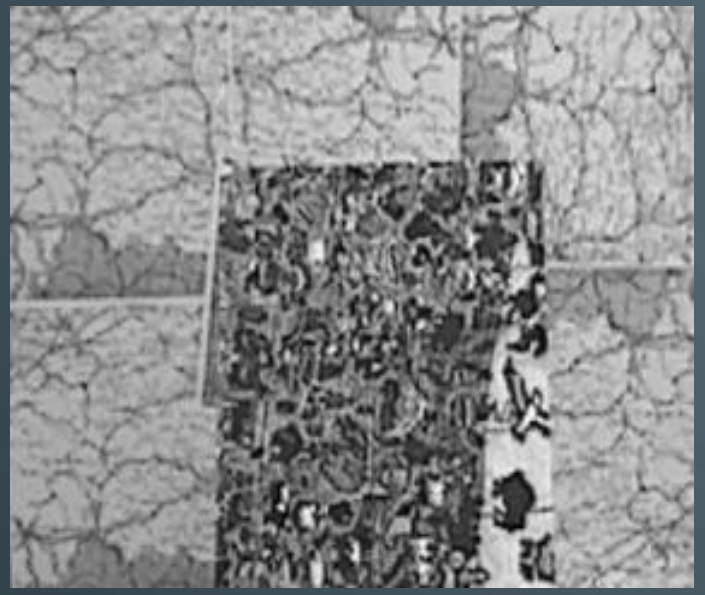


Left Image



Right Image

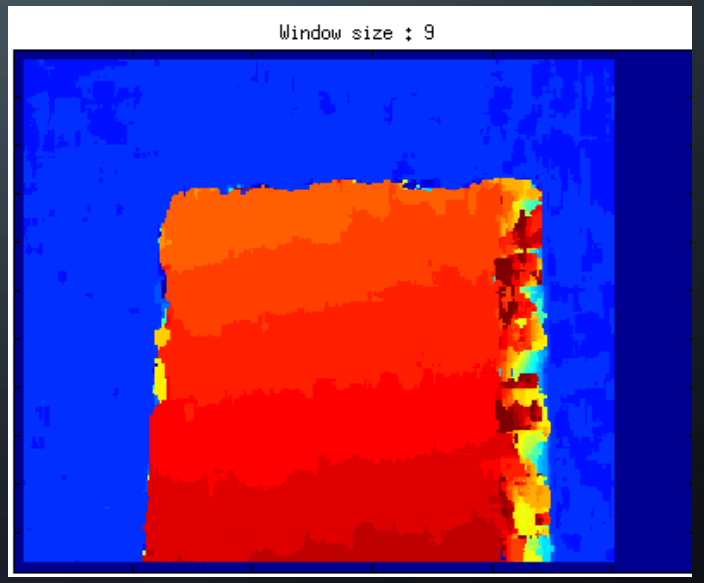
Left image



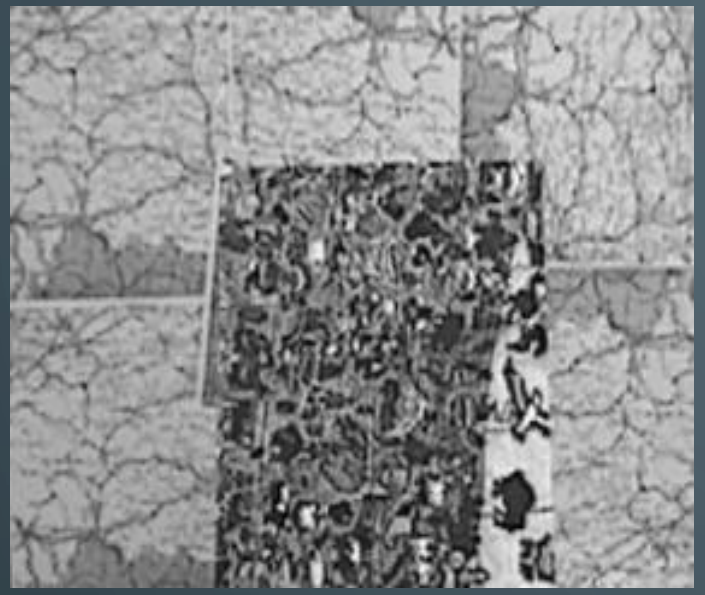
B/W SSD

Ground Truth

Color SSD



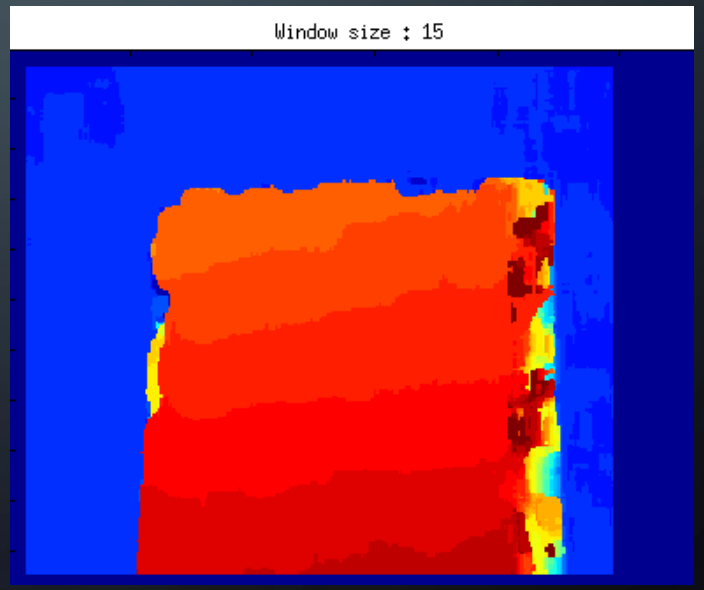
Left image



B/W SSD

Ground Truth

Color SSD



SAWTOOTH

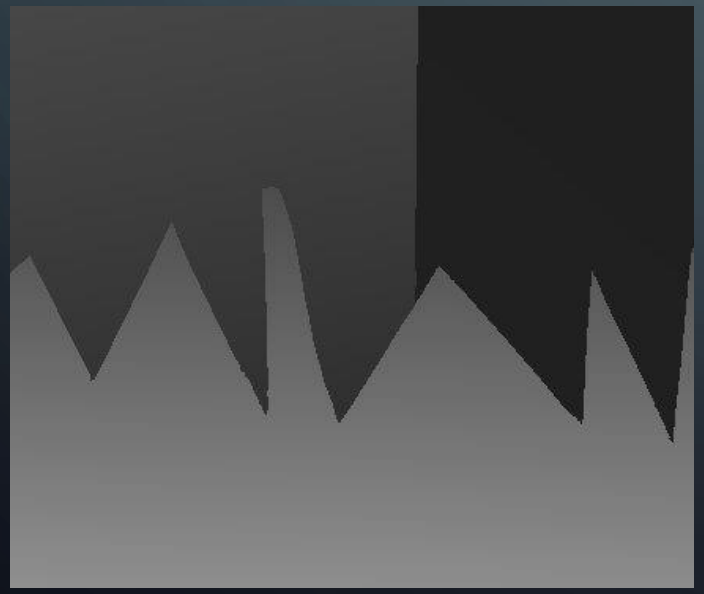
Left Image

Right Image

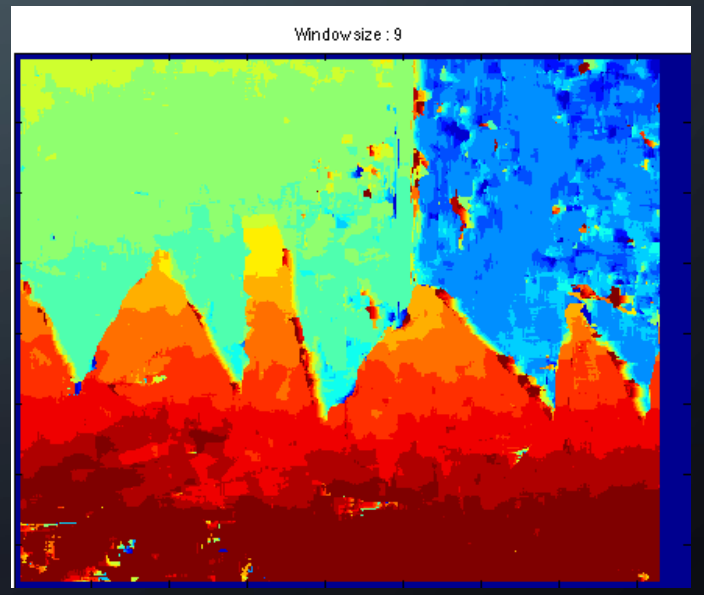
Left image

B/W SSD

Ground Truth

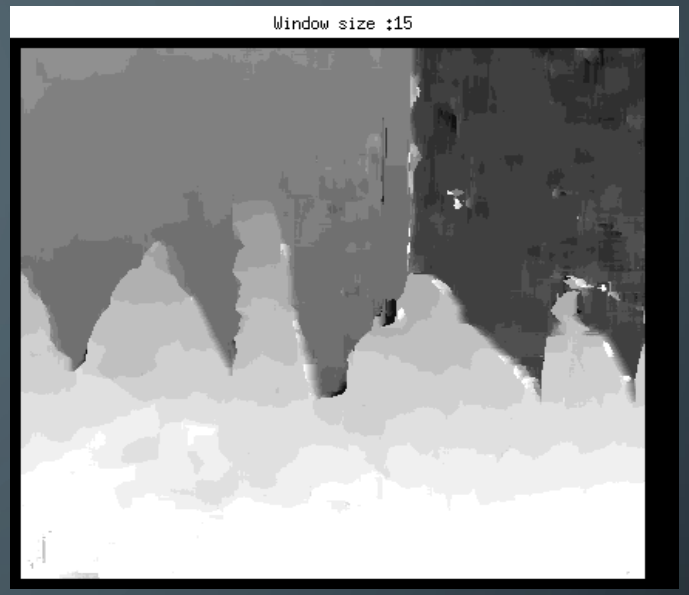


Color SSD

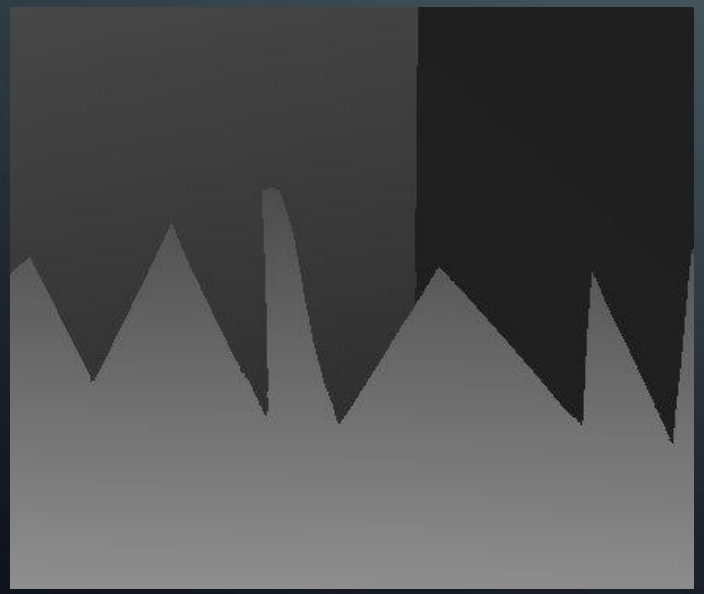


Left image

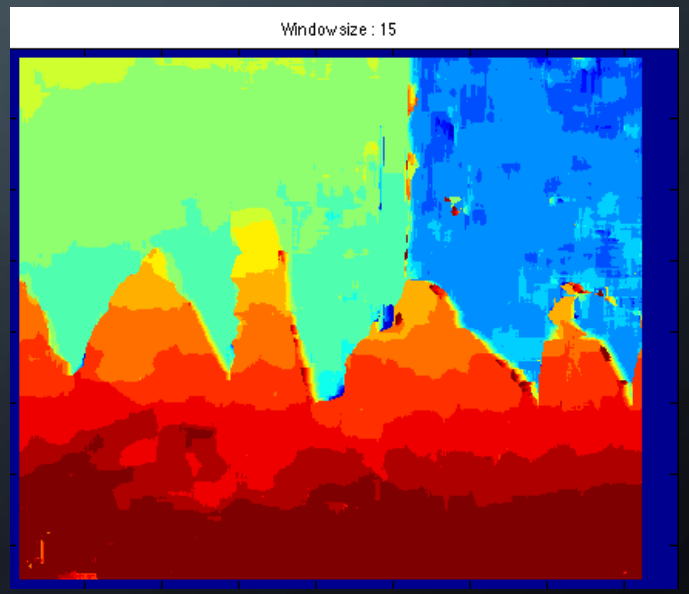
B/W SSD



Ground Truth



Color SSD



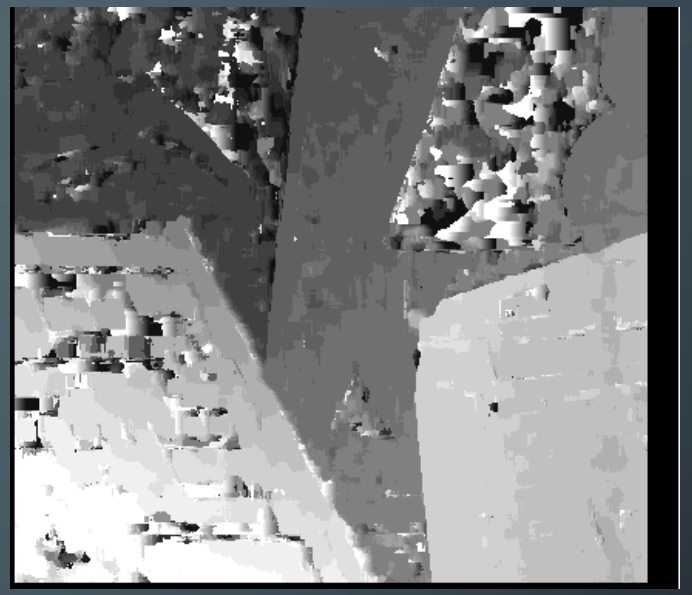
VENUS

Left Image

Right Image

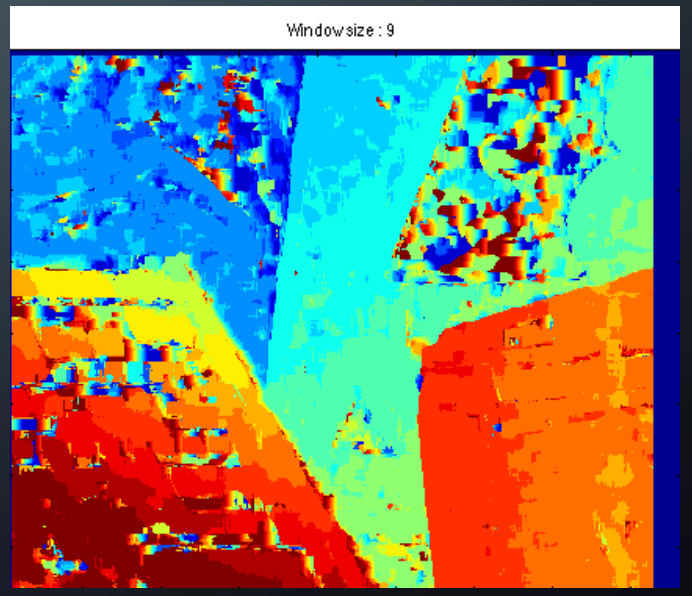
Left image

B/W SSD



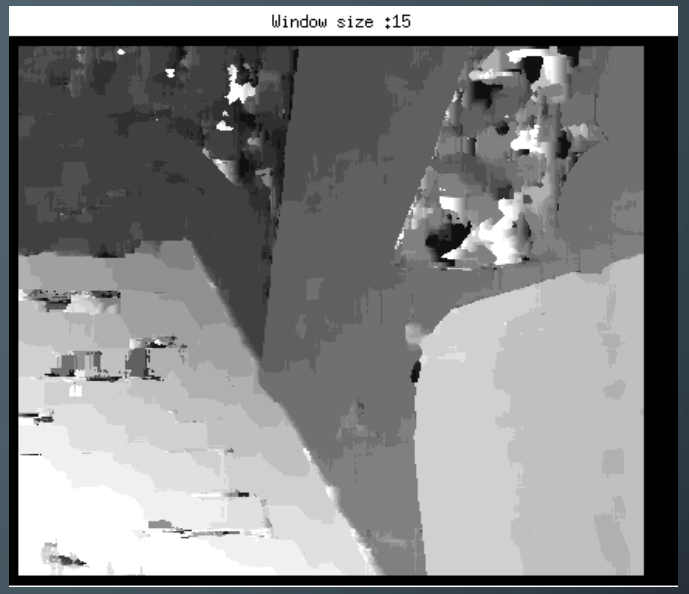
Ground Truth

Color SSD

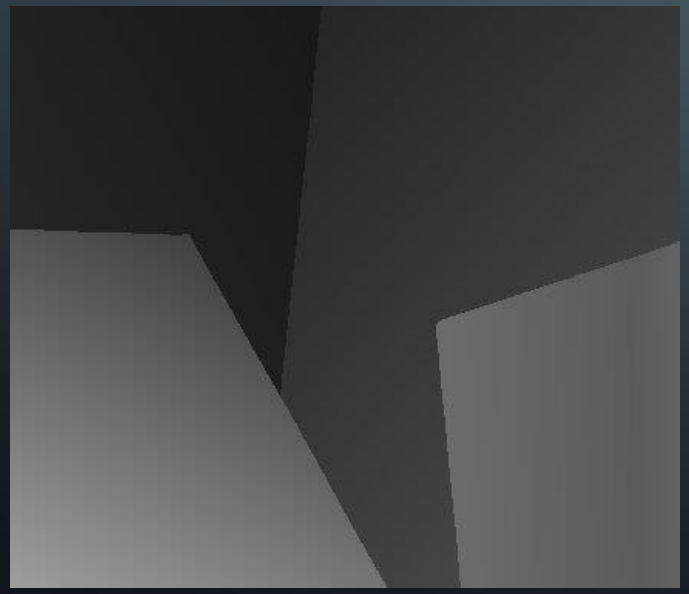


Left image

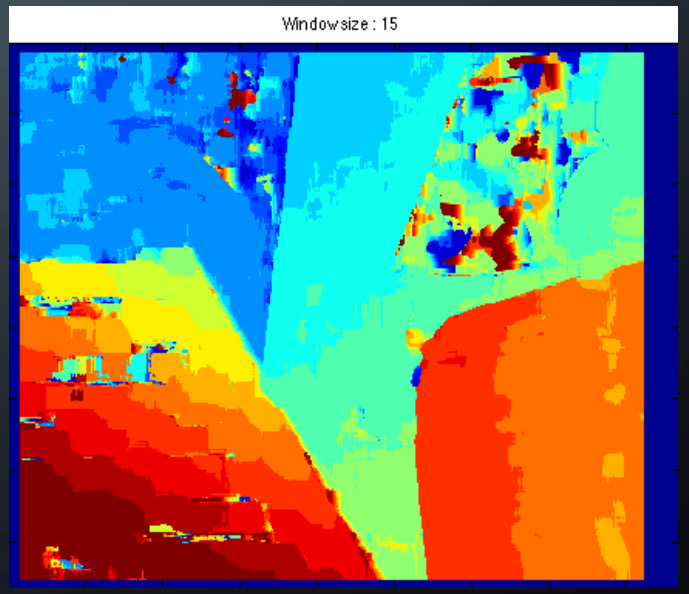
B/W SSD



Ground Truth



Color SSD



BULL

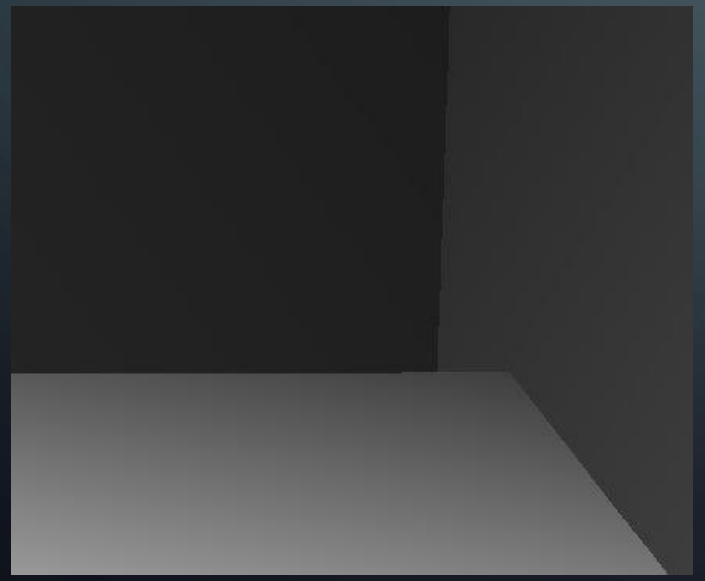
Left Image

Right Image

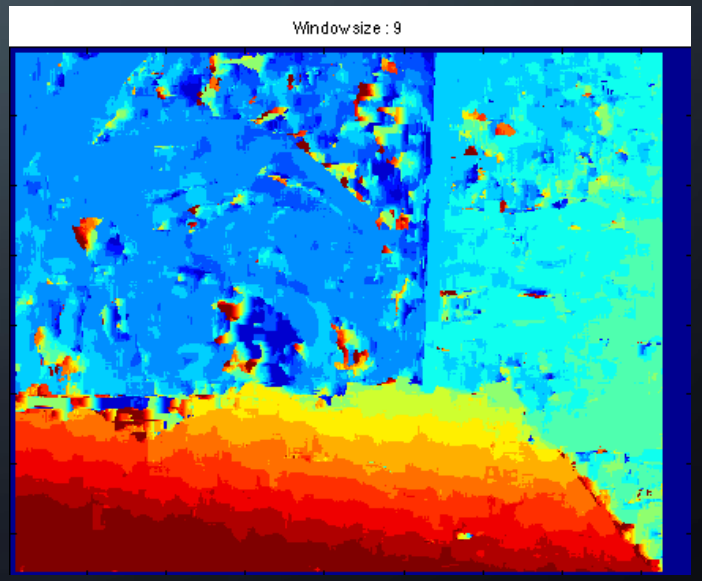
Left image

B/W SSD

Ground Truth

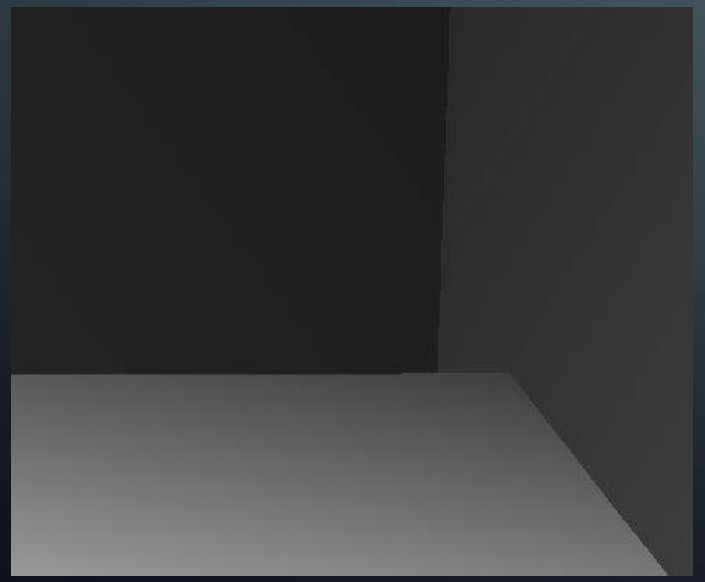


Color SSD

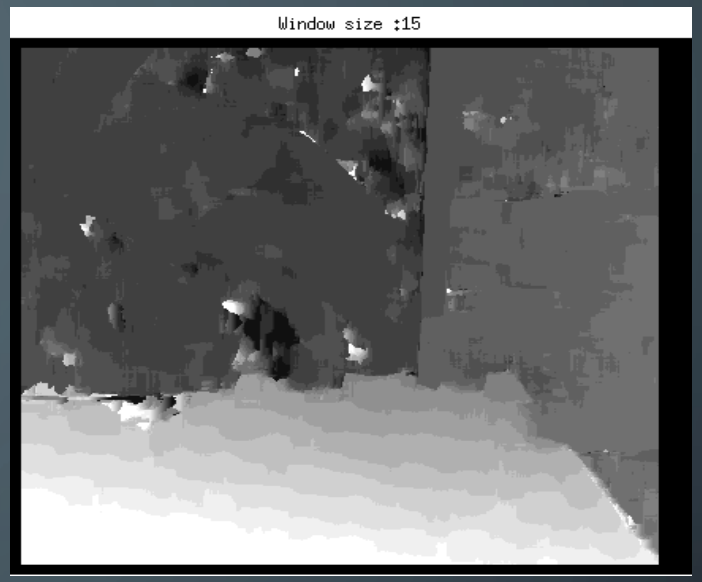


Left image

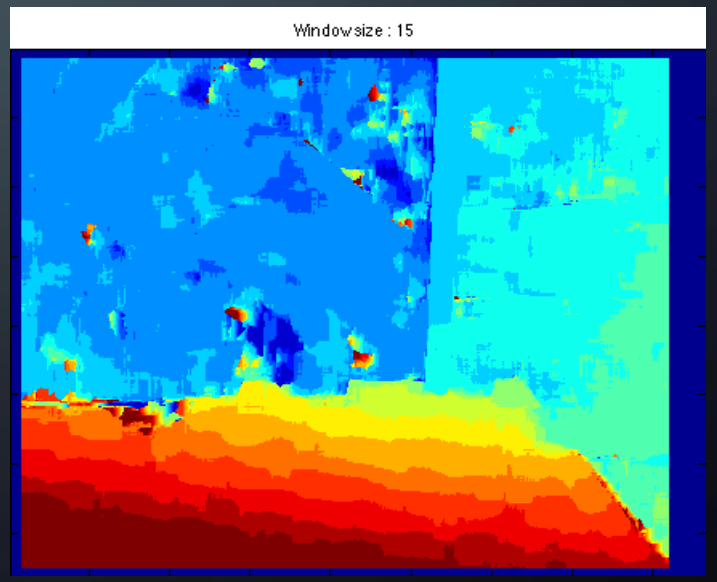
Ground Truth



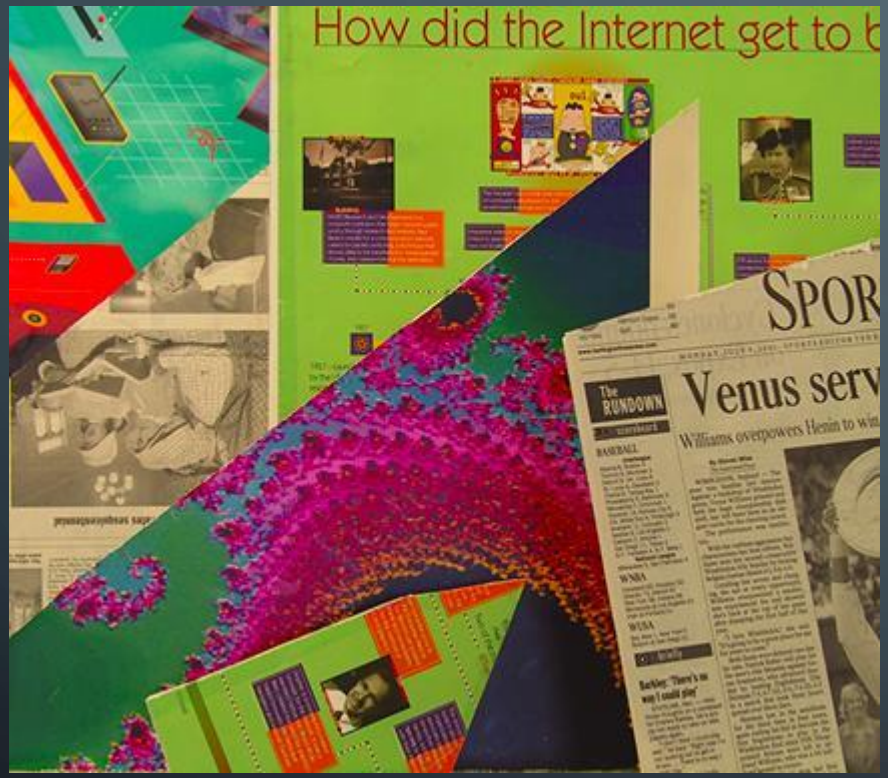
B/W SSD



Color SSD



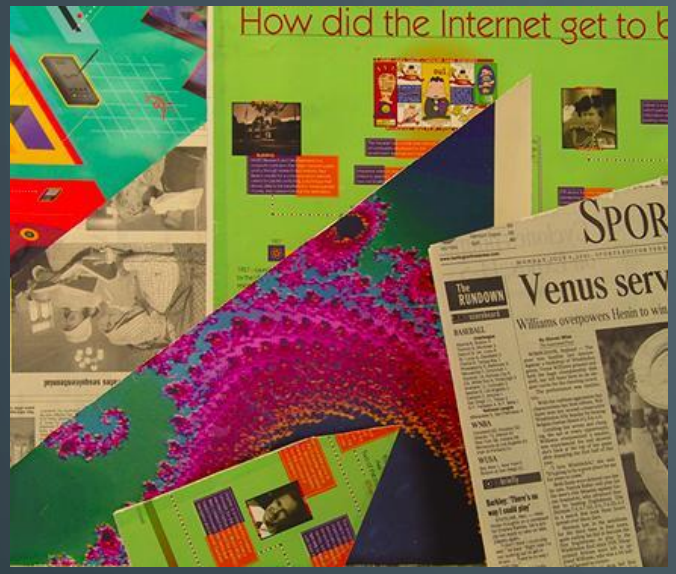
POSTER



Left Image

Right Image

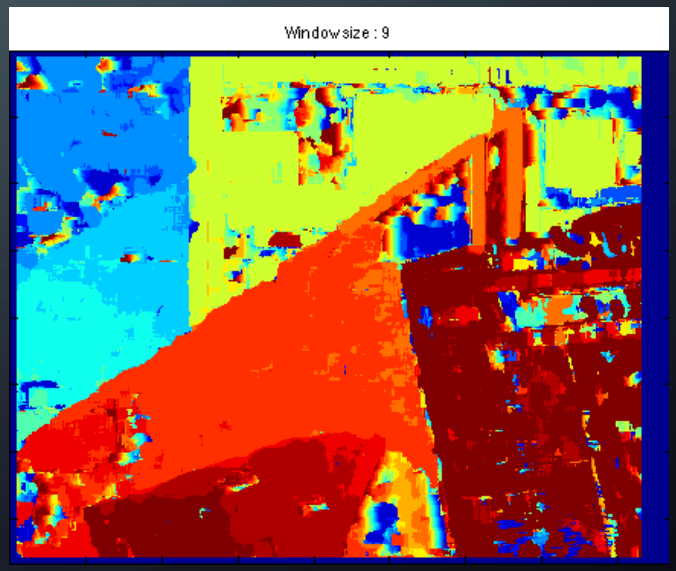
Left image



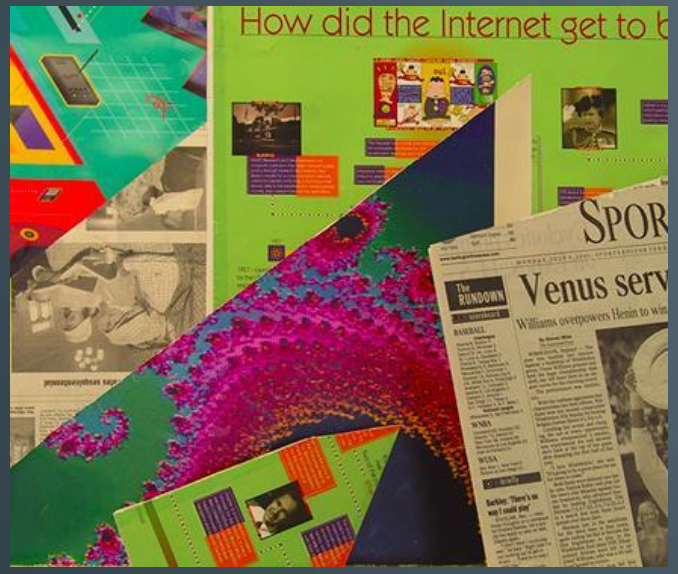
B/W SSD

Ground Truth

Color SSD



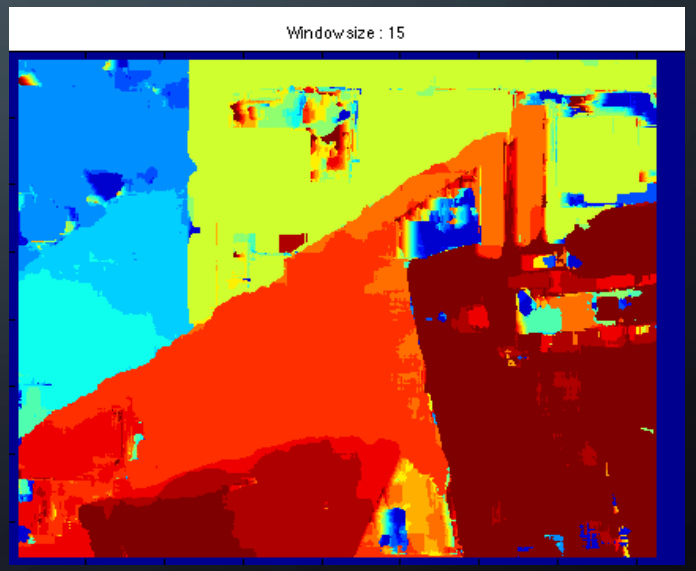
Left image



B/W SSD

Ground Truth

Color SSD



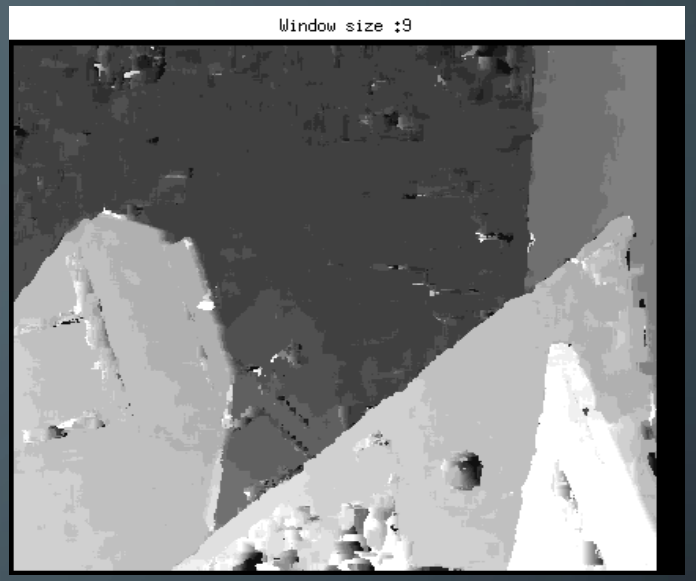
BARN 1

Left Image

Right Image

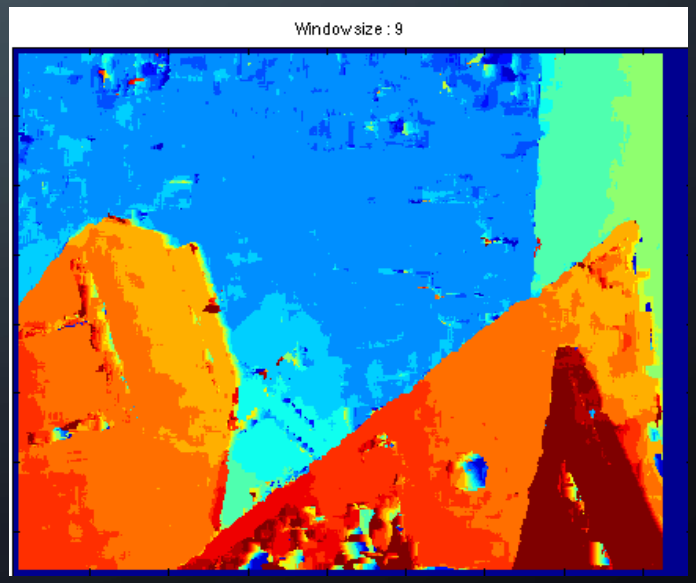
Left image

B/W SSD



Ground Truth

Color SSD

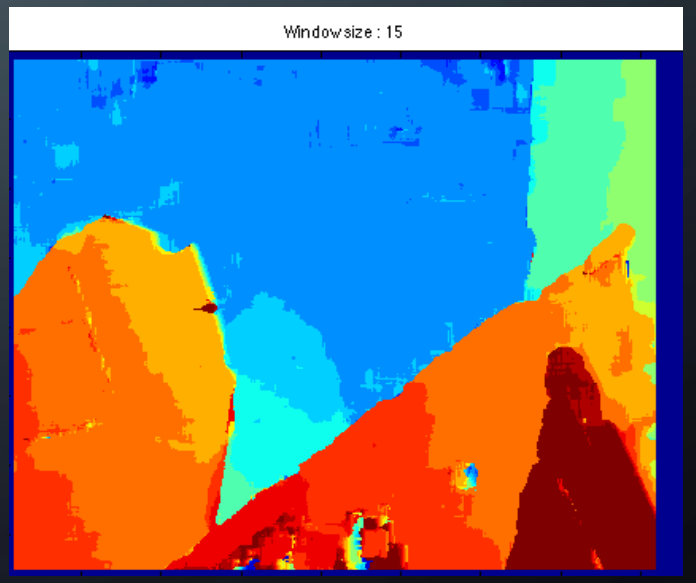


Left image

Ground Truth

B/W SSD

Color SSD



BARN 2

Left Image

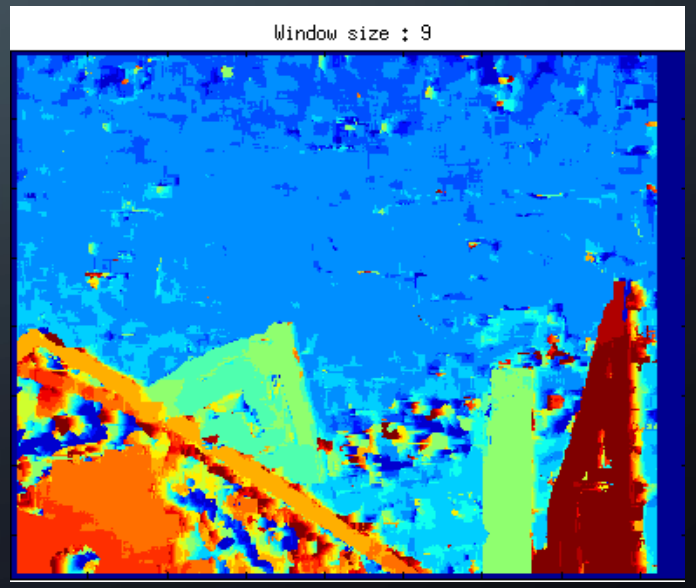
Right Image

Left image

B/W SSD

Ground Truth

Color SSD

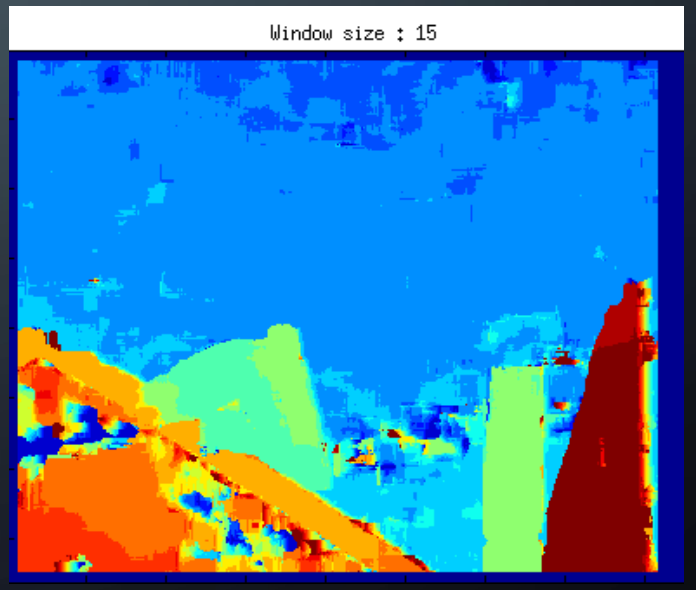


Left image

B/W SSD

Ground Truth

Color SSD

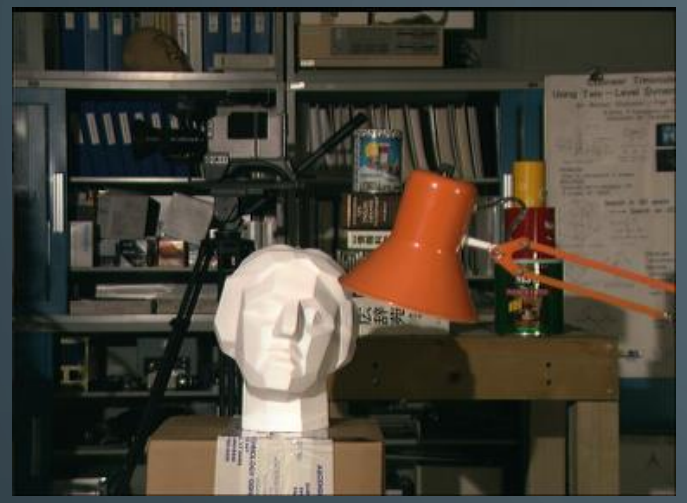


TSUKUBA

Left Image

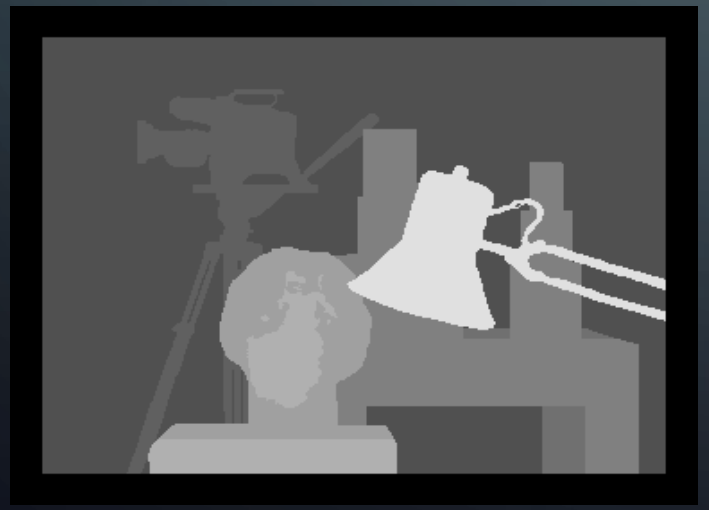
Right Image

Left image

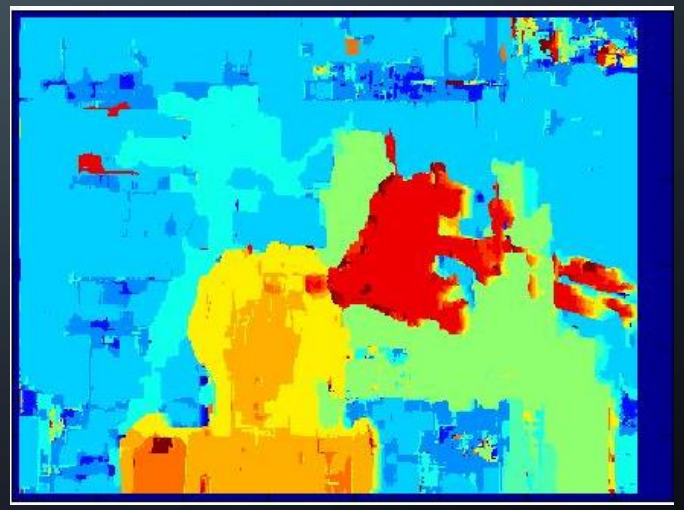


B/W SSD

Ground Truth



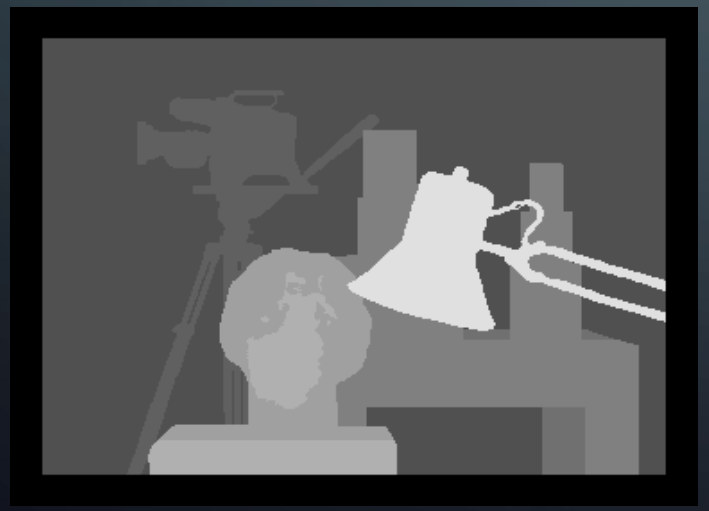
Color SSD



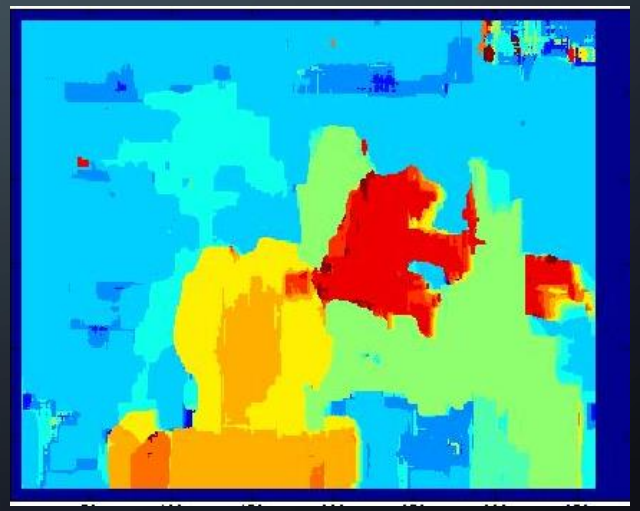
Left image

B/W SSD

Ground Truth



Color SSD



Approach – 2

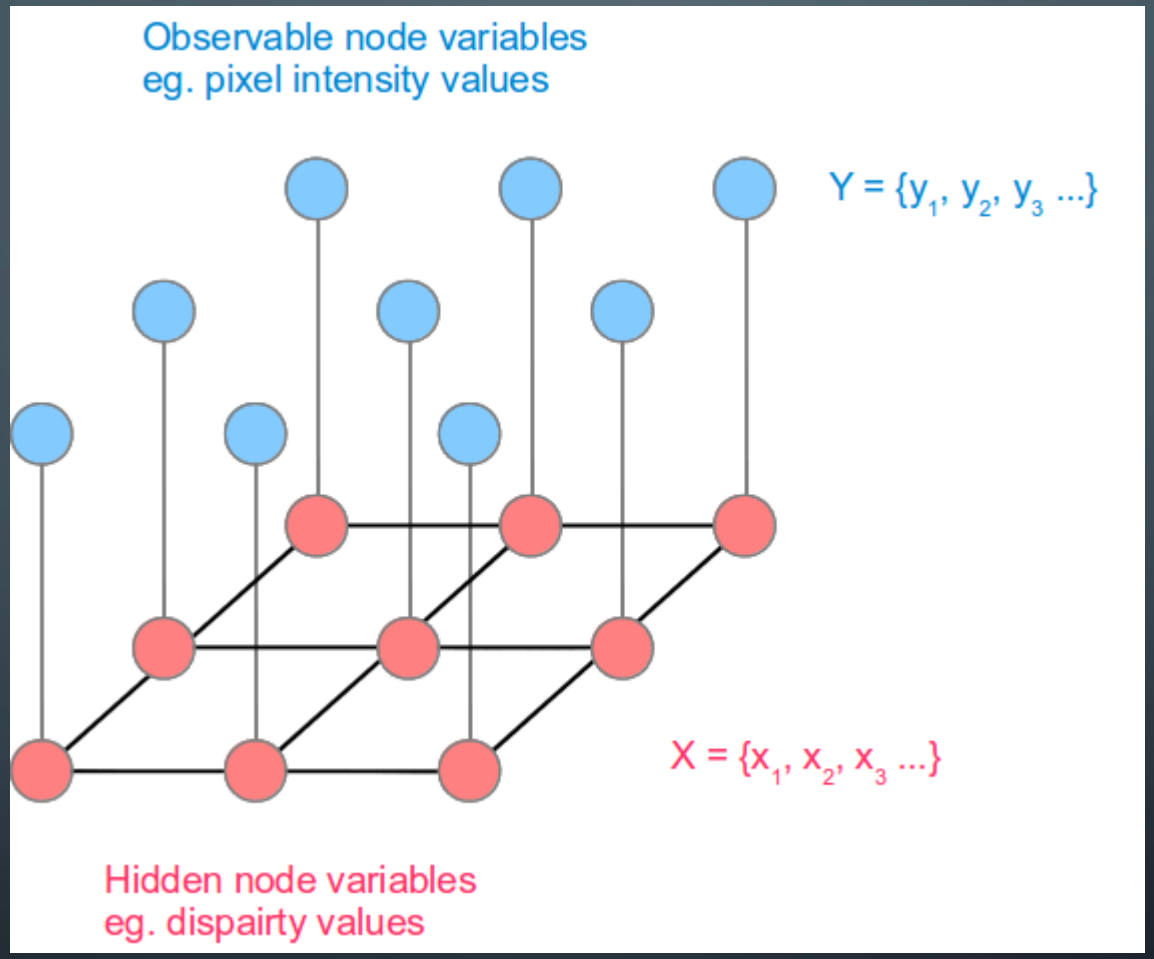
BELIEF PROPAGATION

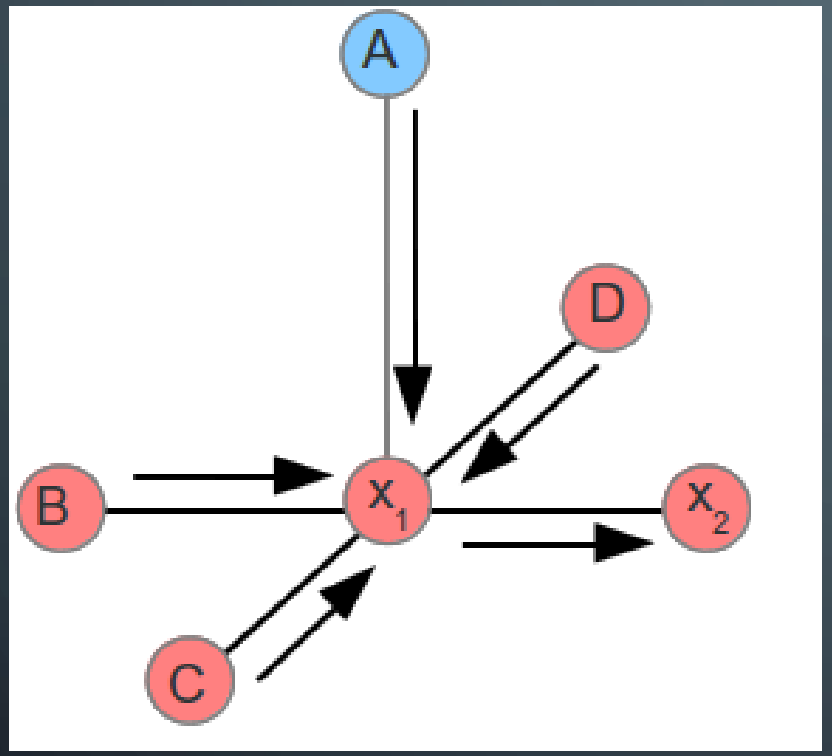
(MARKOV RANDOM FIELD)

TERMS :

- Data Cost : Cost of matching pixel y_i to x_i.
- Smoothness Cost : Cost of matching disparities to neighboring disparities.

$$energy(Y, X) = \sum_i DataCost(y_i, x_i) + \sum_{j=\text{neighbours of i}} SmoothnessCost(x_i, x_j)$$


Observable node variables
eg. pixel intensity values
Y = {y1, y2, y3 ...}
X = {x1, x2, x3 ...}
Hidden node variables
eg. dispairty values



Hidden node (disparity) receives message from neighboring nodes which contains probabilities of different disparities.

Example : For disparities 1- 5 , x1 will receive
probability of disparity = 1 from C
probability of disparity = 2 from C
probability of disparity = 3 from C
probability of disparity = 4 from C
probability of disparity = 5 from C

BELIEF PROPAGATION

- Send message in ‘X’ direction of operation of belief propagation.
- Get belief propagation from all directions except ‘X’.
- Add them up with data and smoothness cost
- Send them in ‘X’ direction

Algorithm :

- For each iteration

    do belief propagation on right.

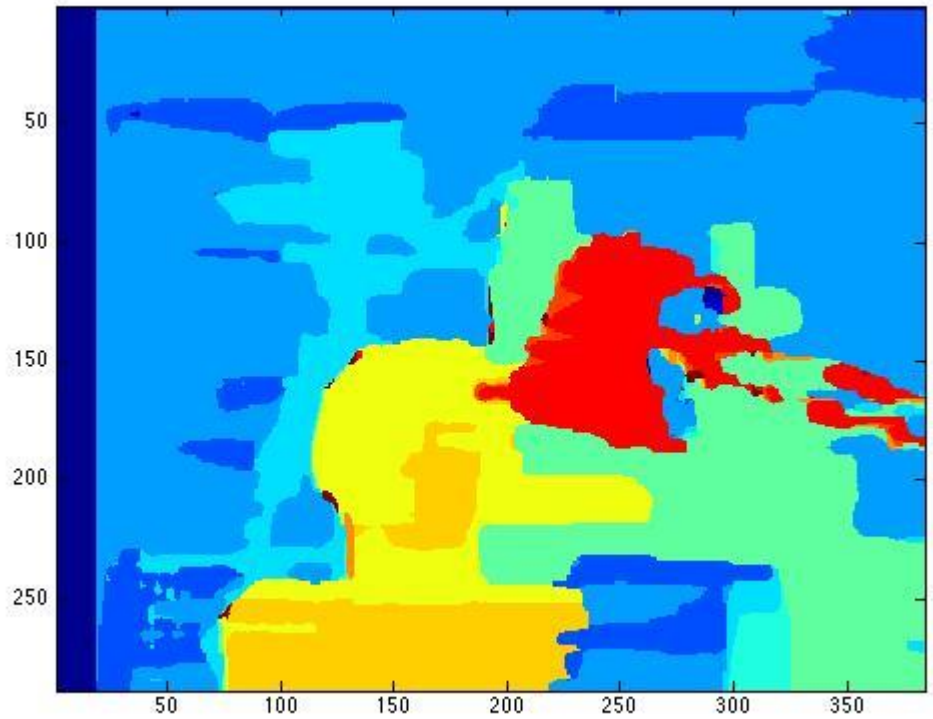
    do belief propagation on left.

    do belief propagation on up.

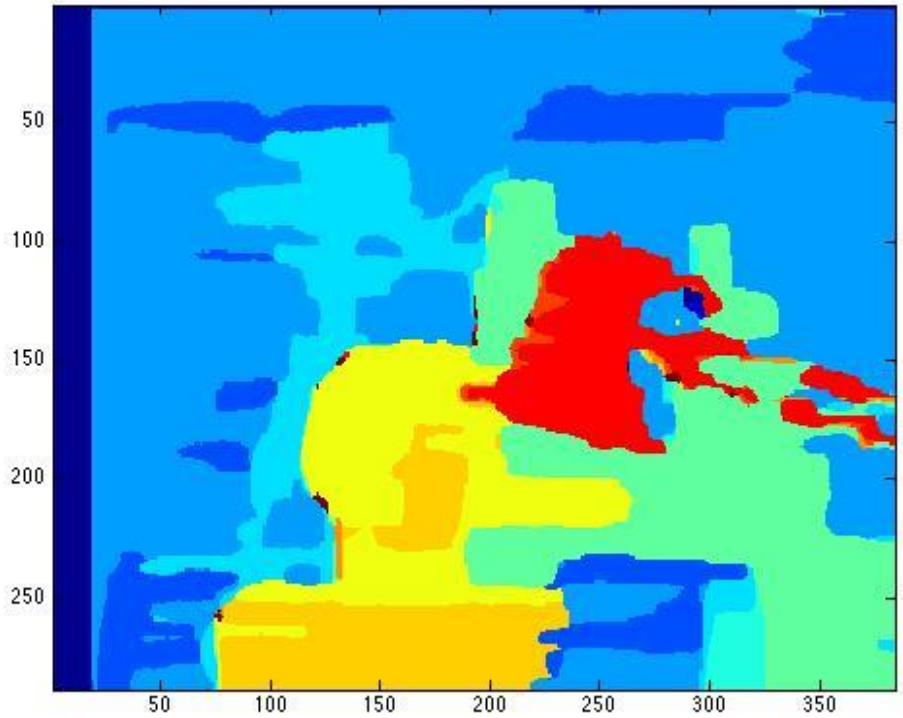
    do belief propagation on down.

    Calculate MAP (Maximum a posteriori)

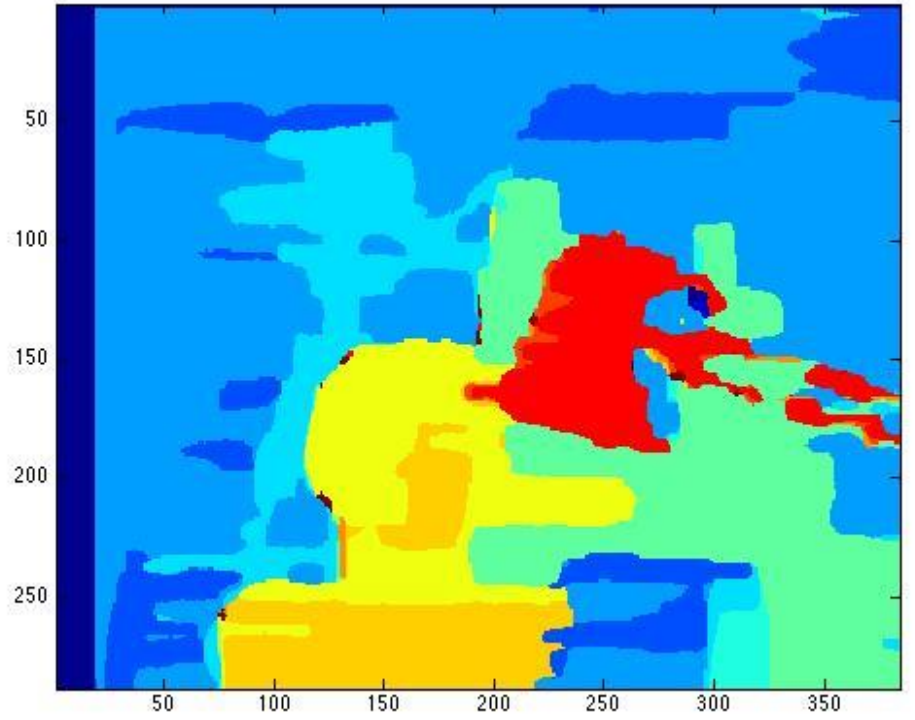
Iteration 1



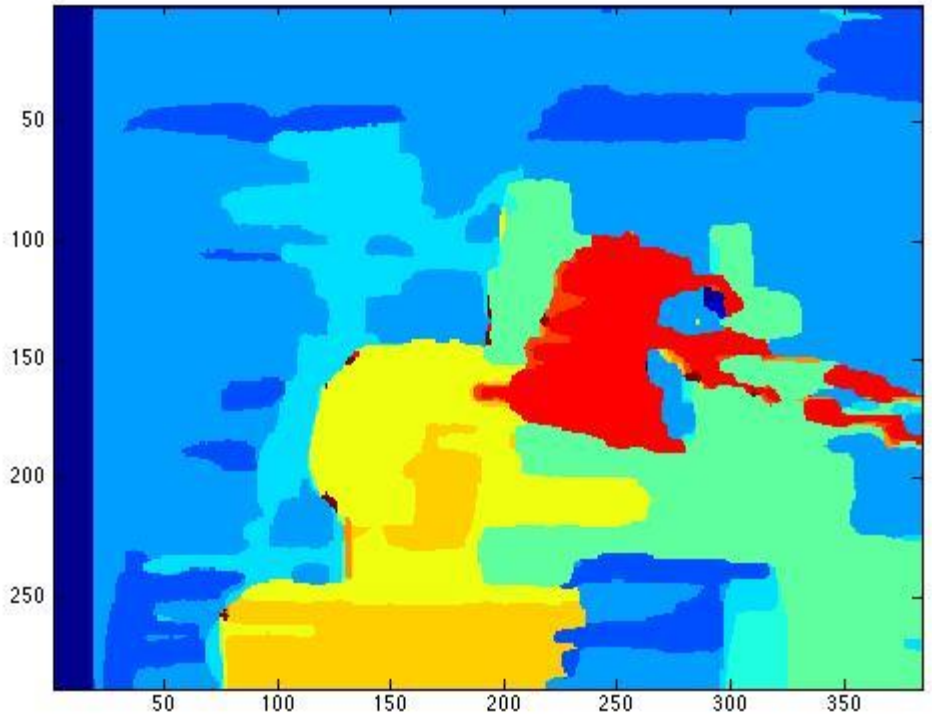
Iteration 2



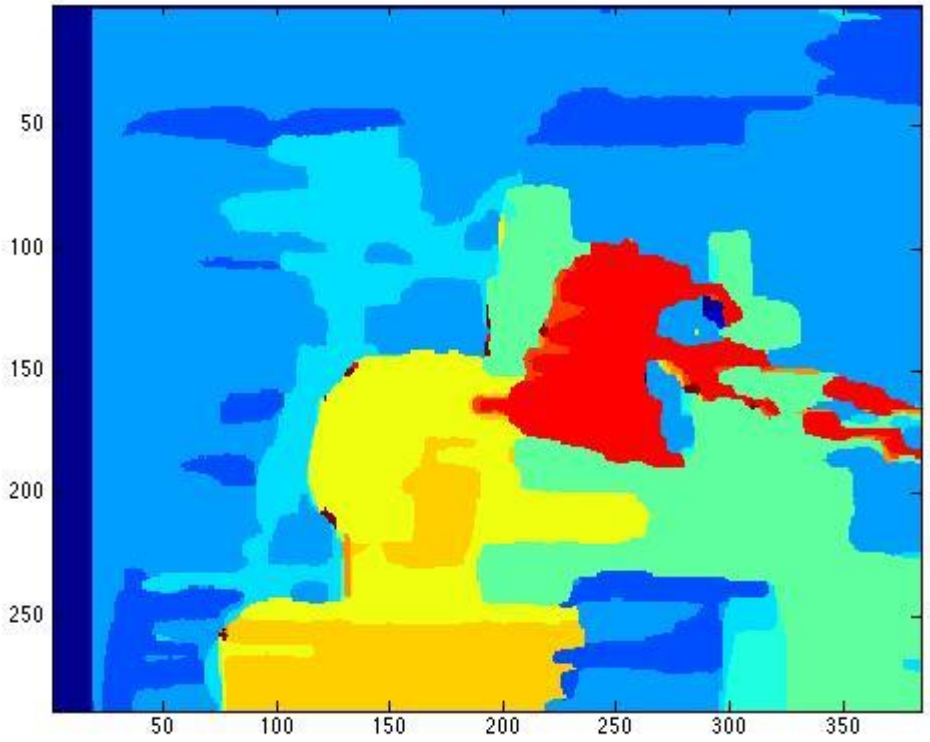
Iteration 3



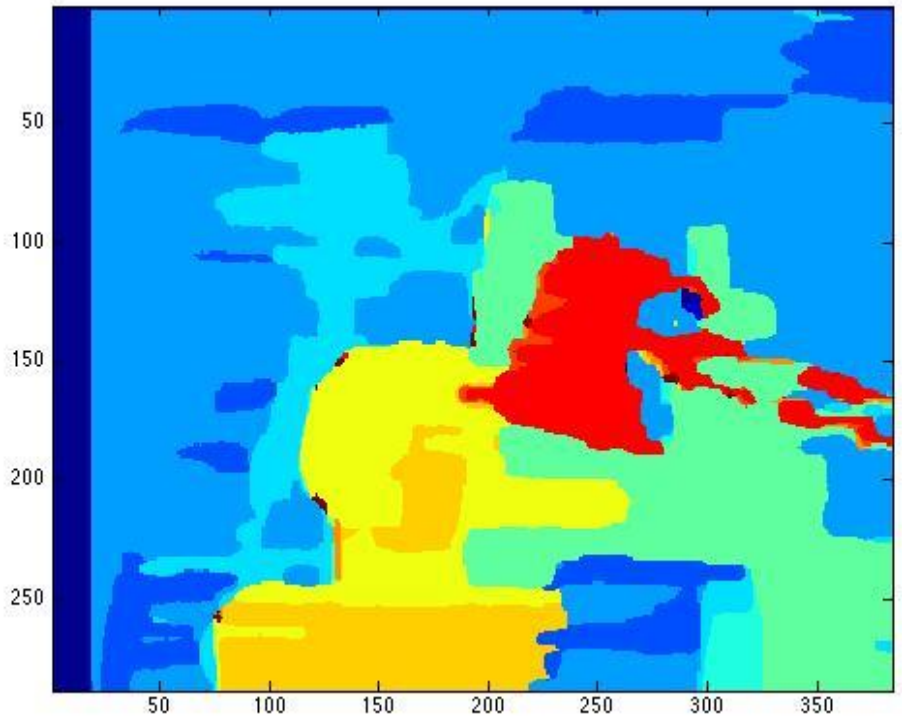
Iteration 4



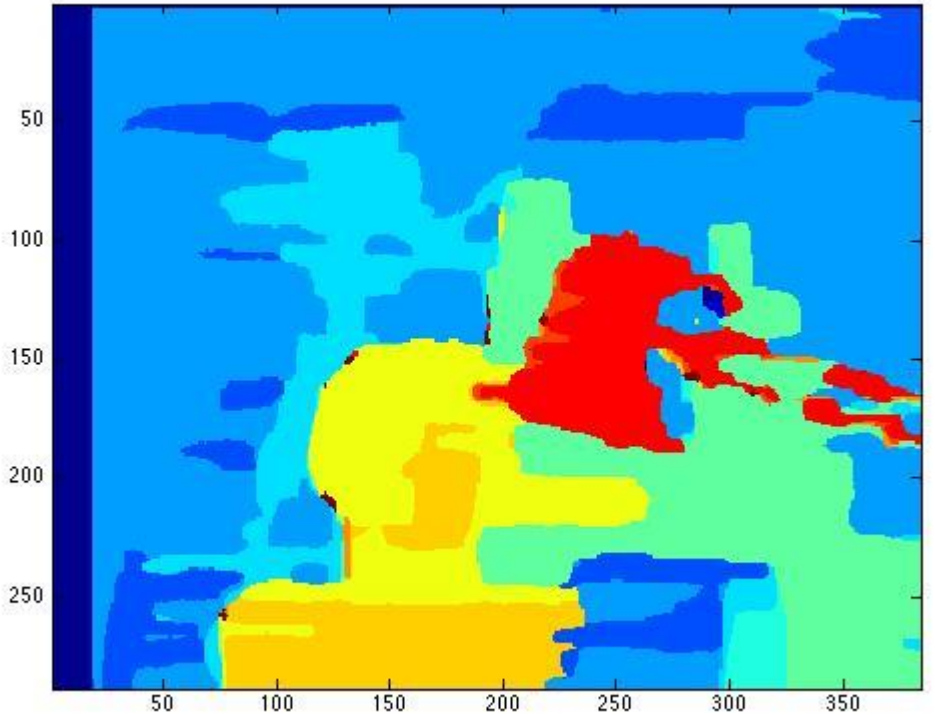
Iteration 5



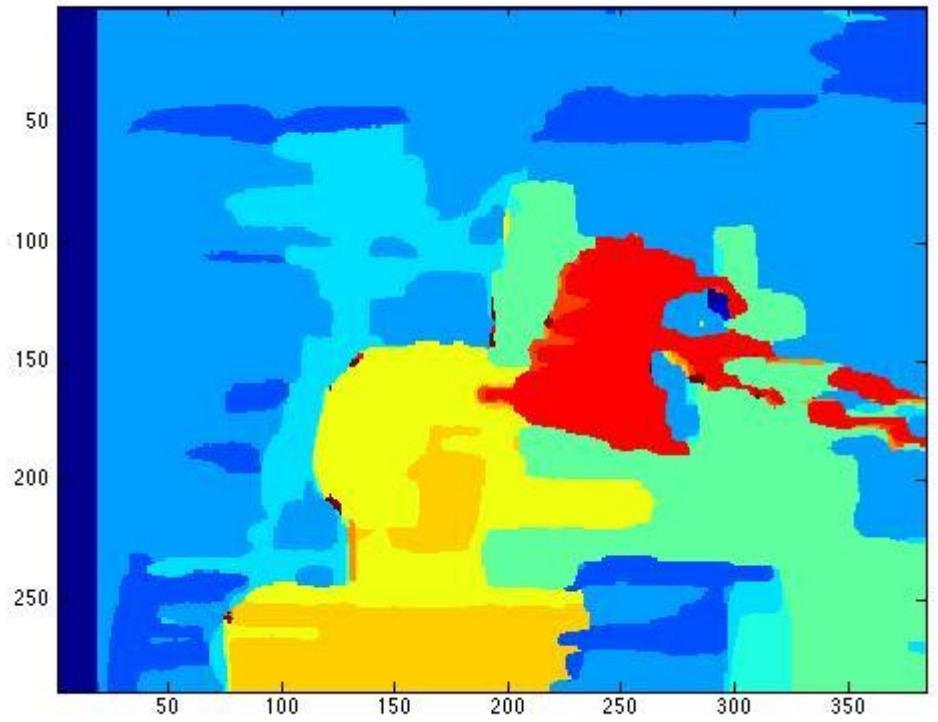
Iteration 6



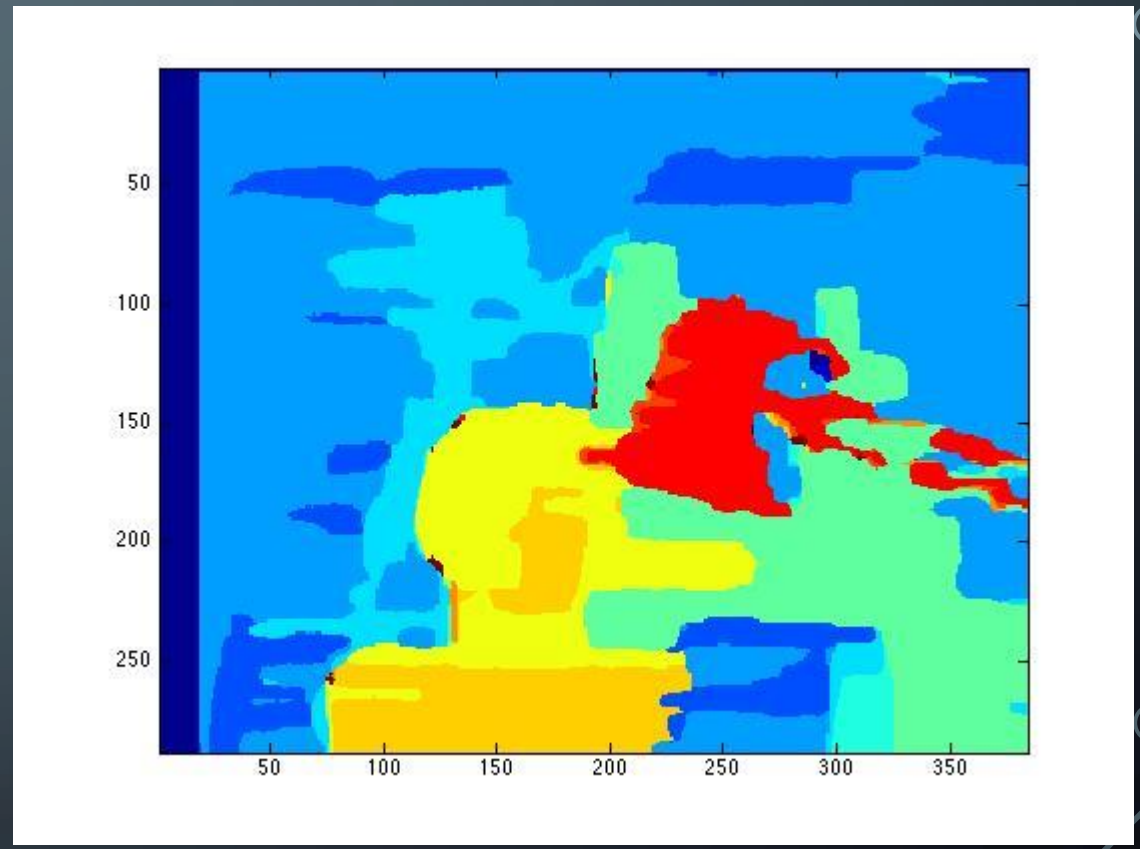
Iteration 7



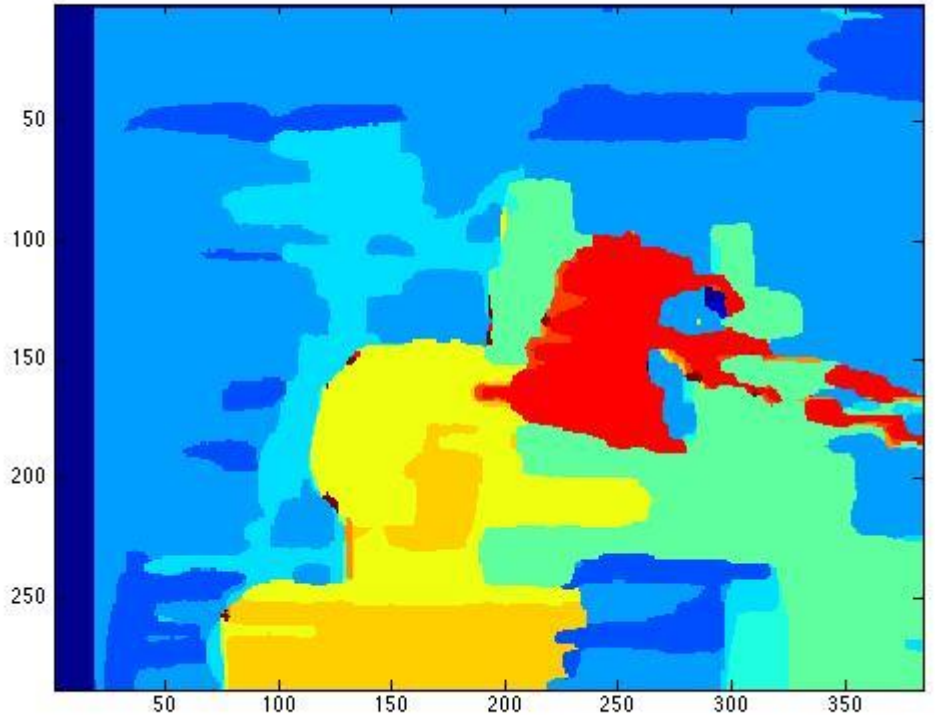
Iteration 8



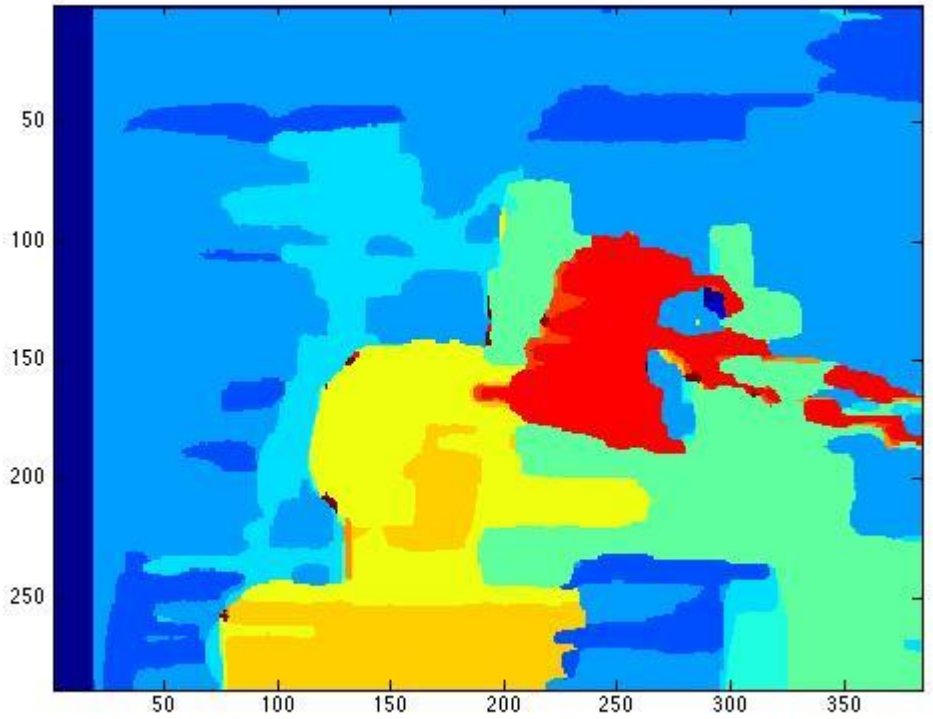
Iteration 9



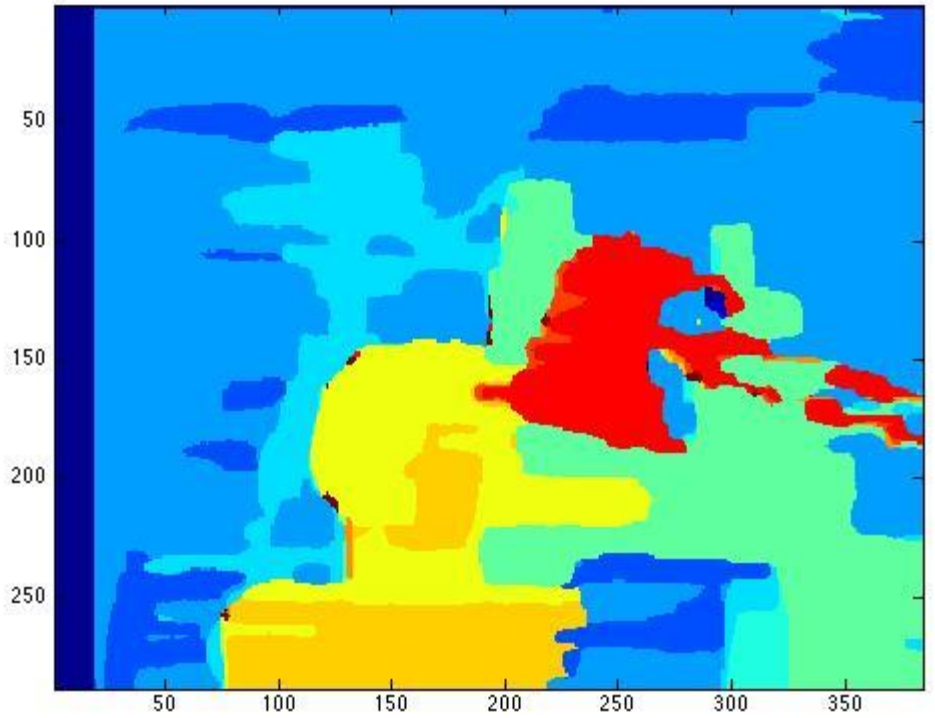
Iteration 10



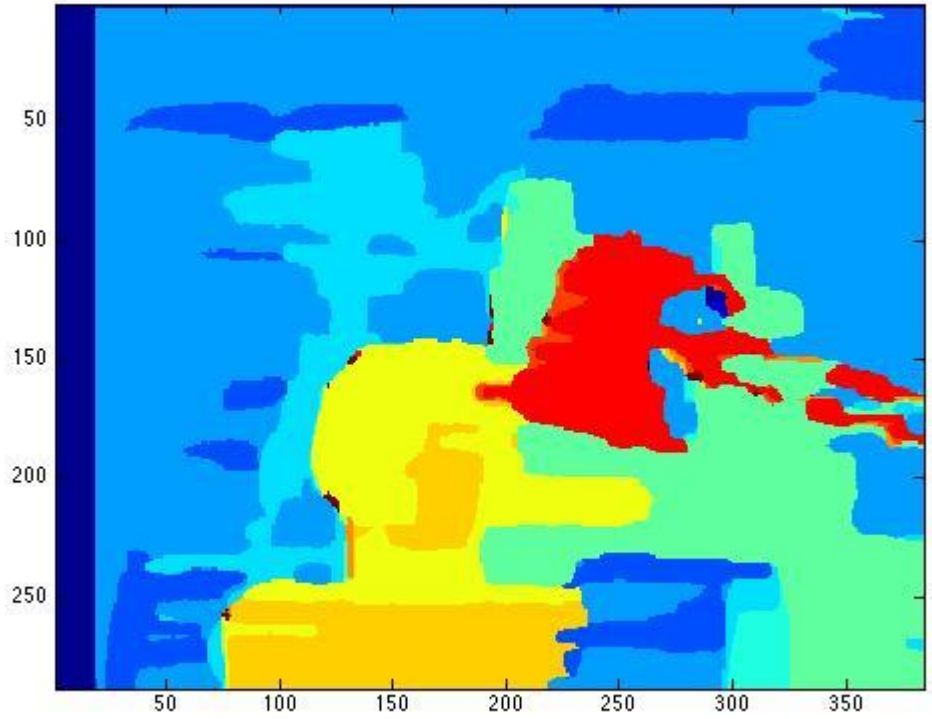
Iteration 11



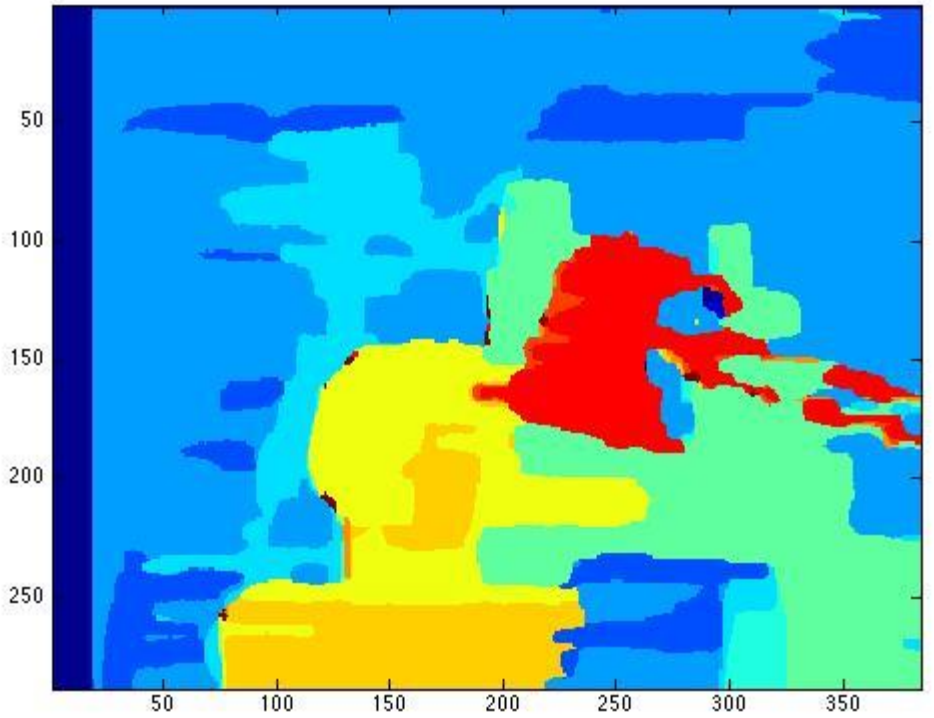
Iteration12



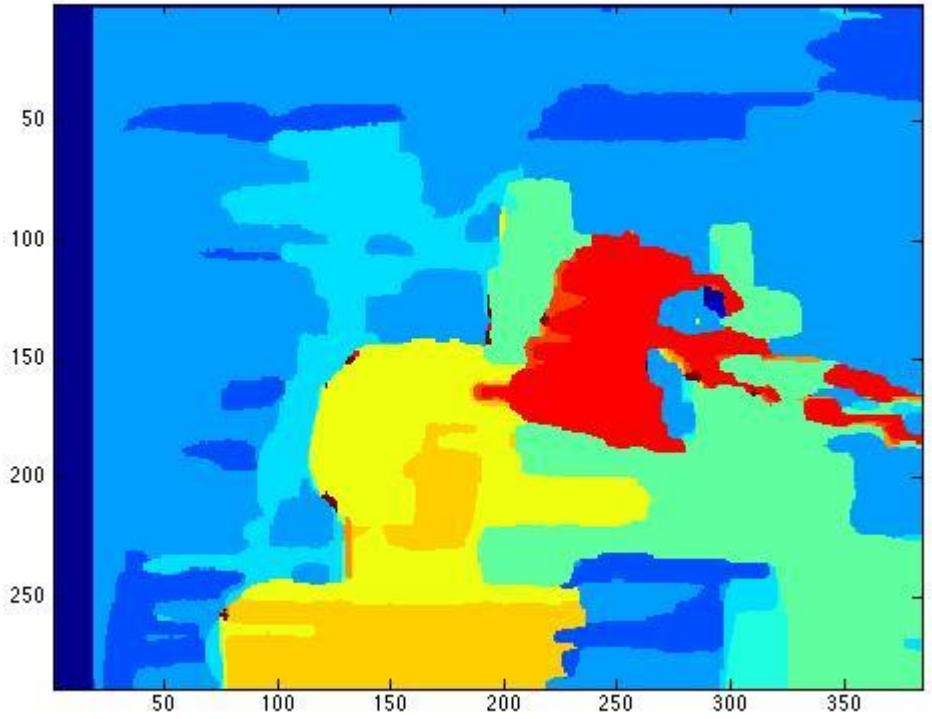
Iteration 13



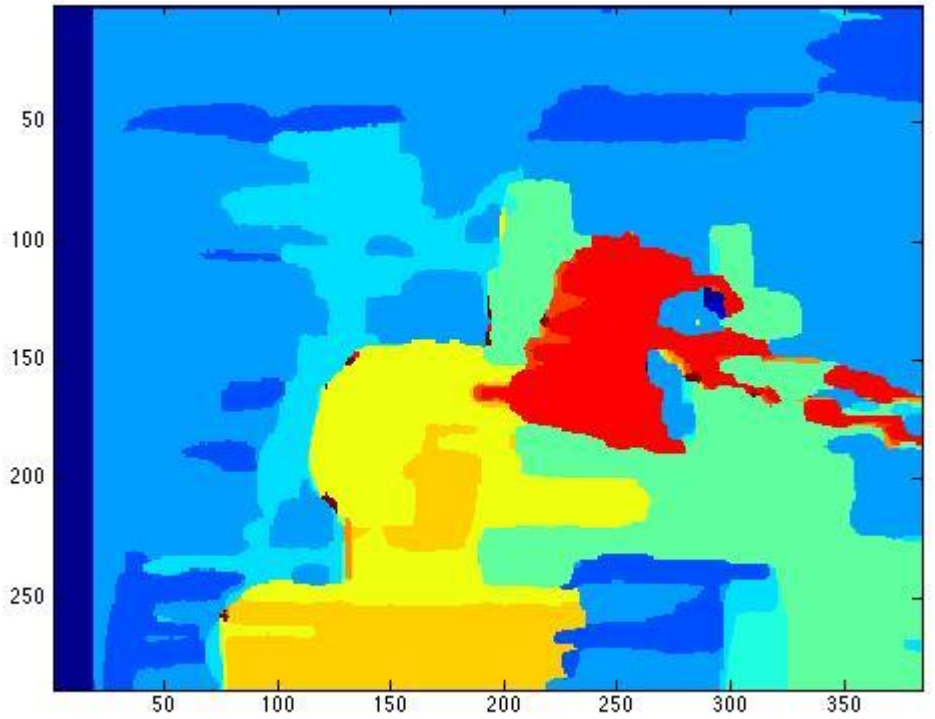
Iteration 14



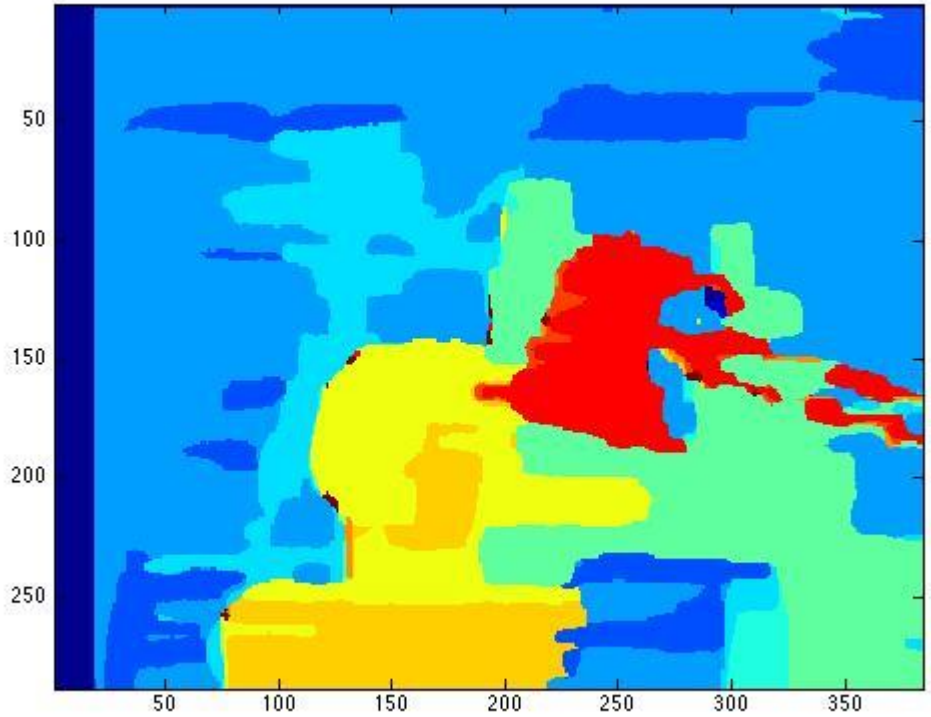
Iteration 15



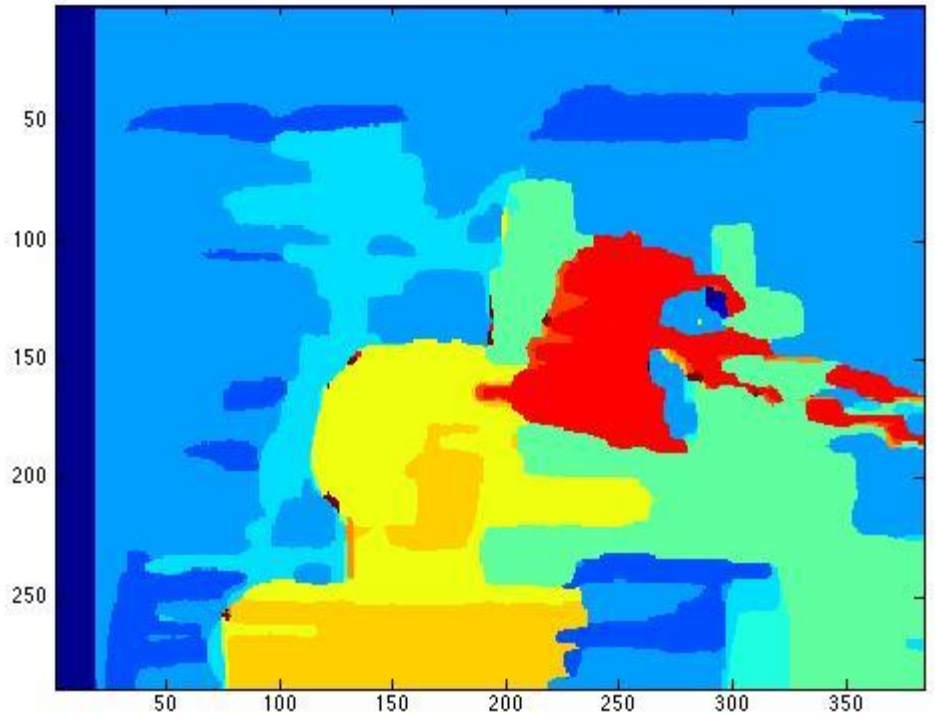
Iteration 16



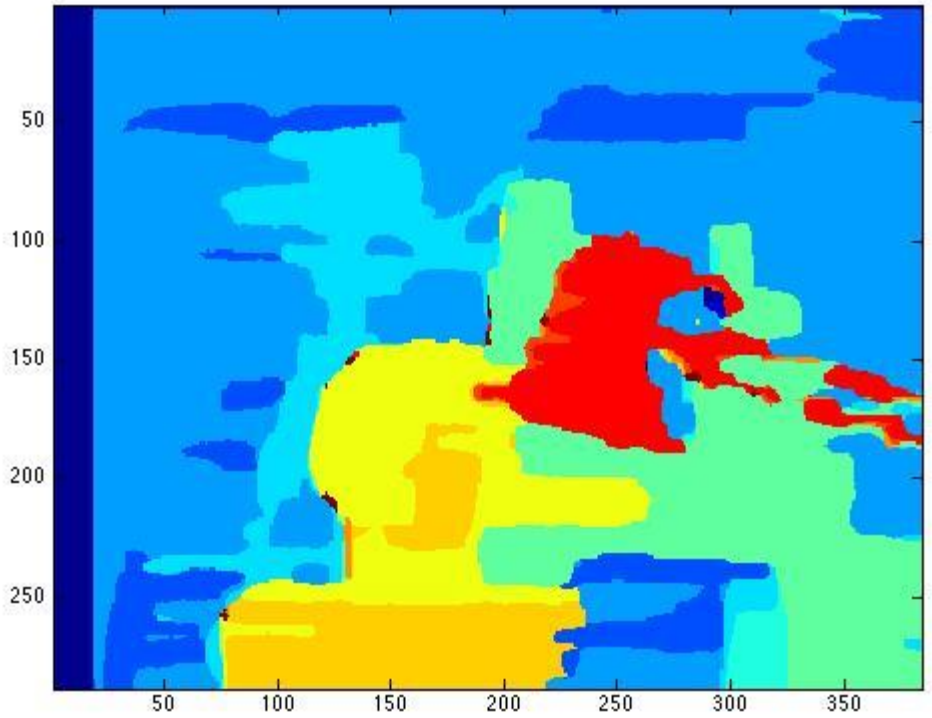
Iteration 17



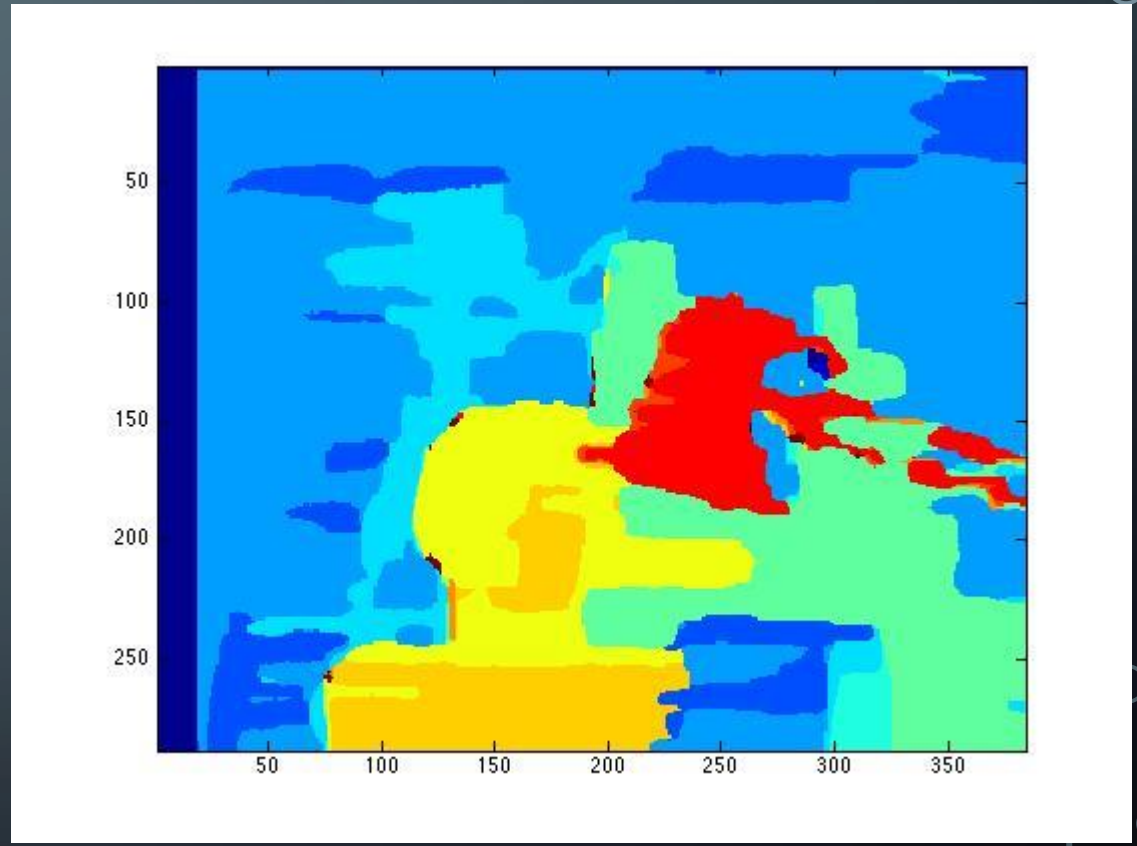
Iteration 18



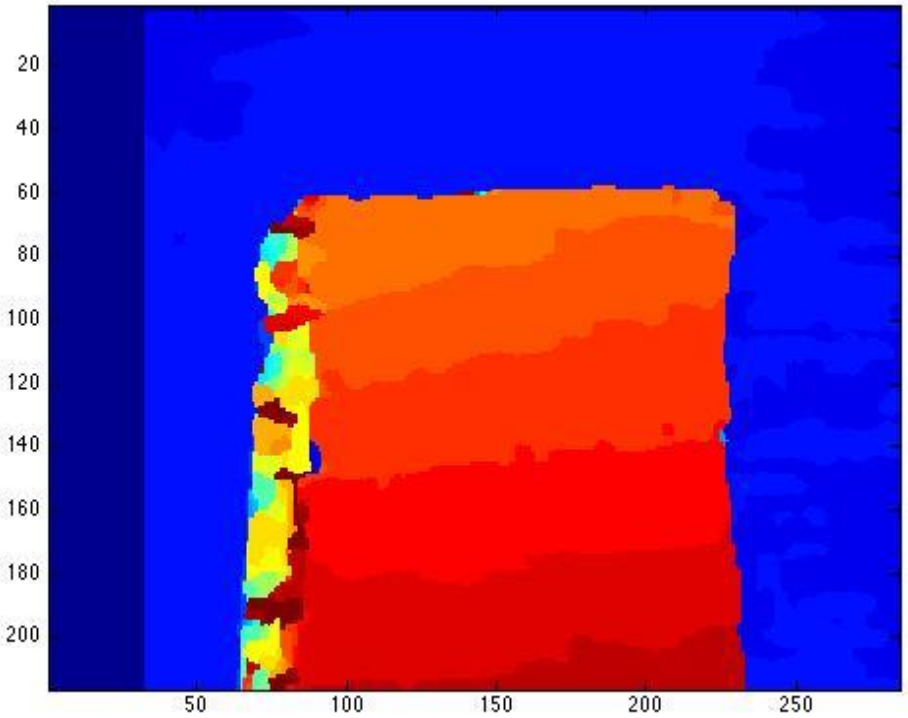
Iteration 19



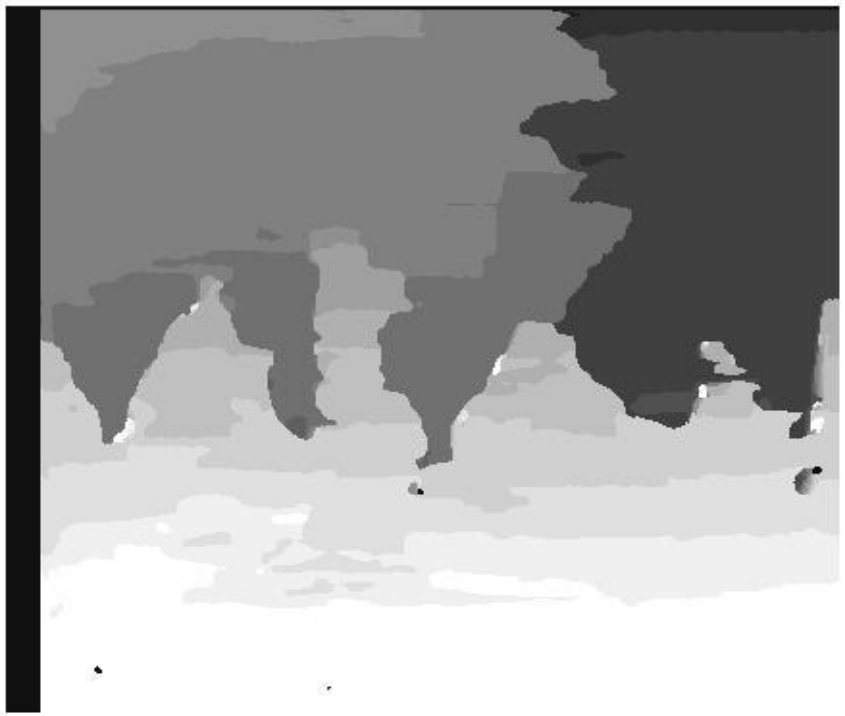
Iteration 20

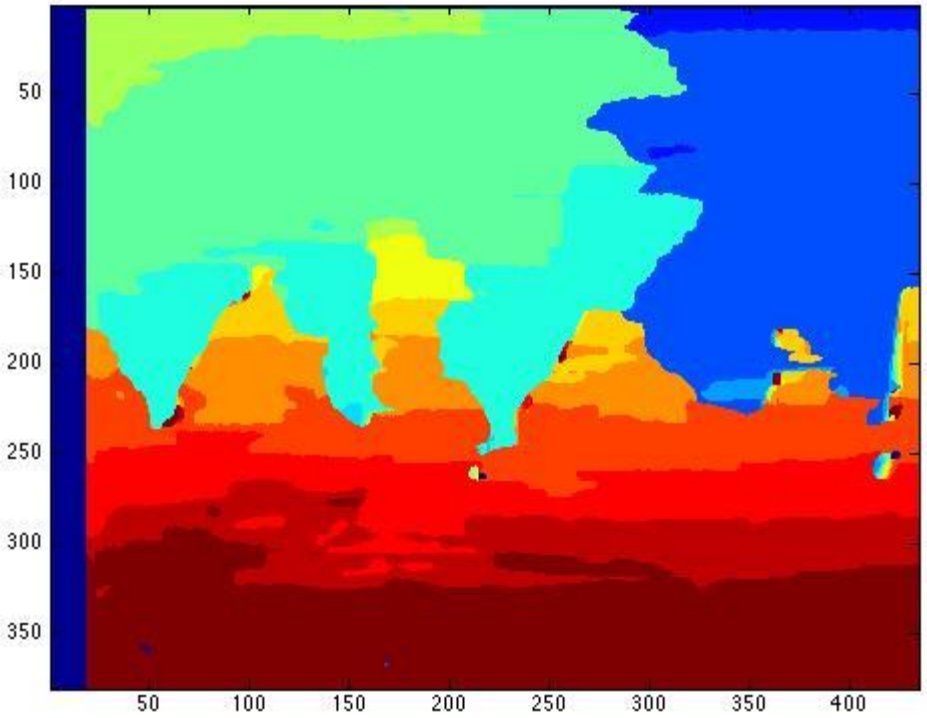


Outputs :

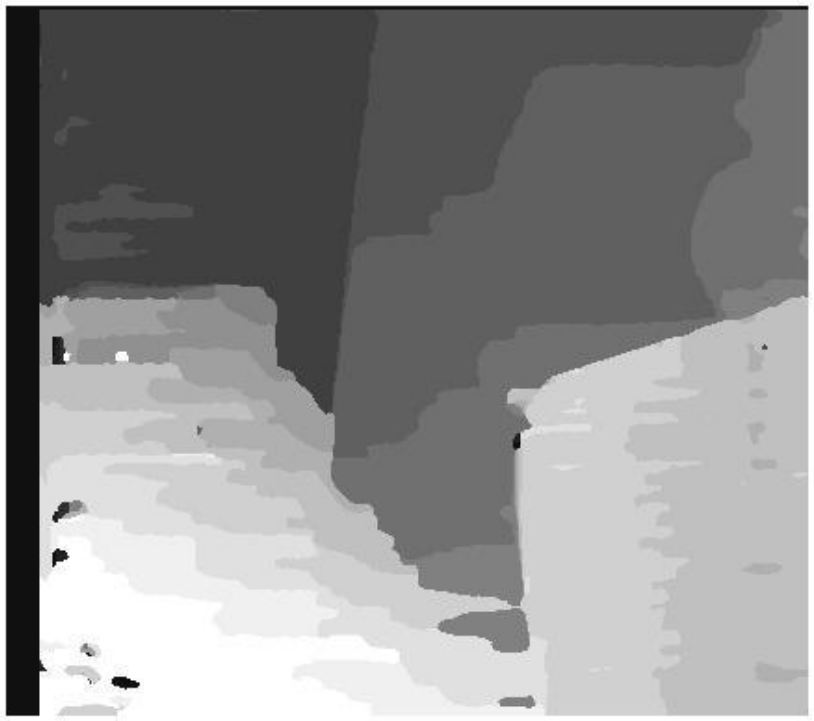


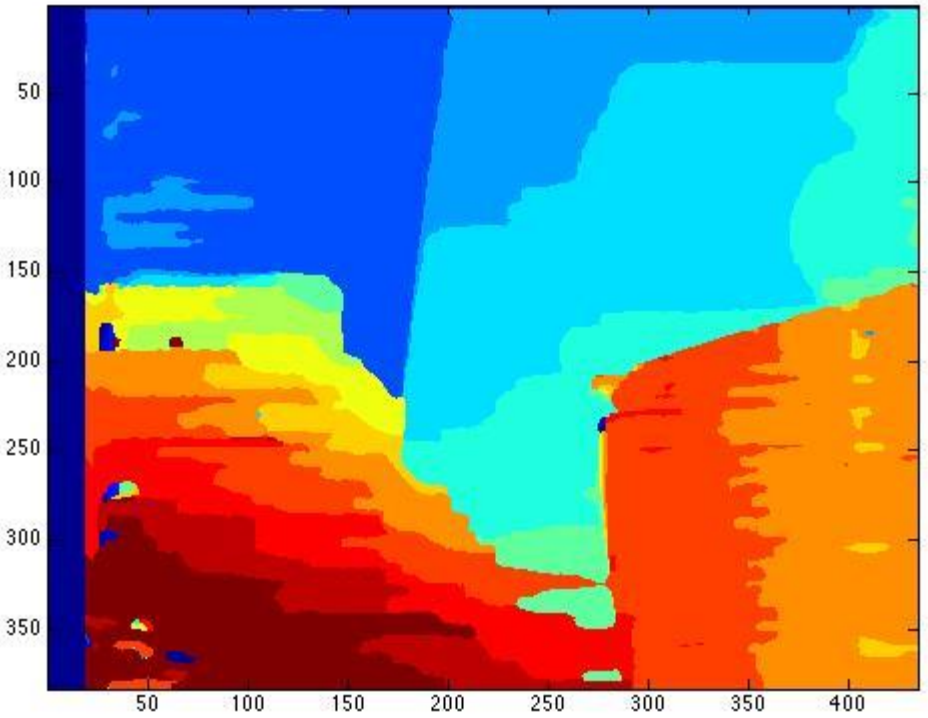
Outputs :



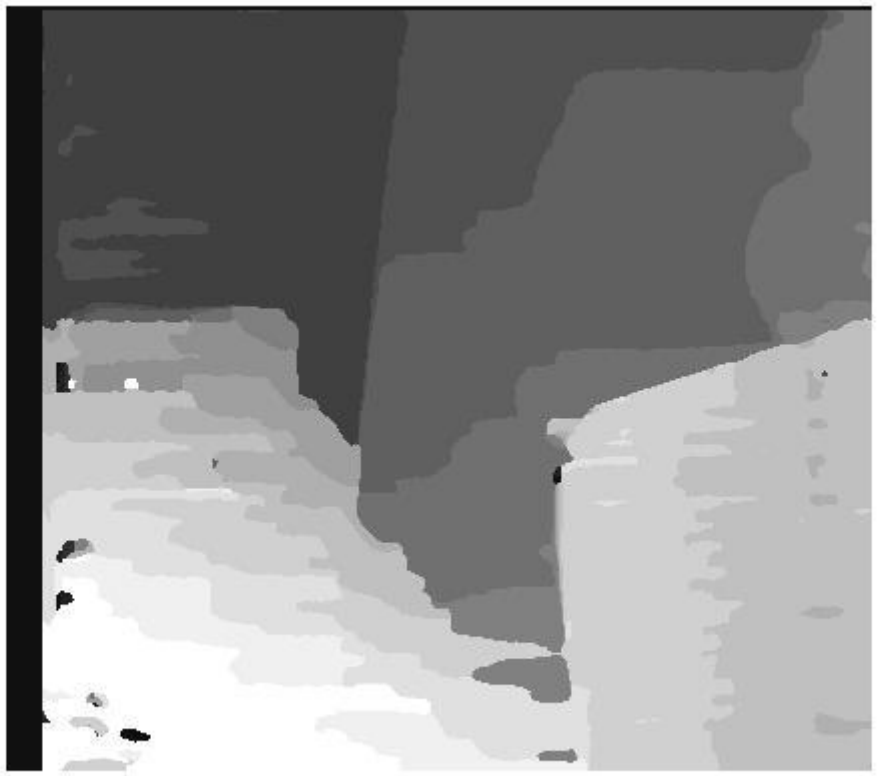


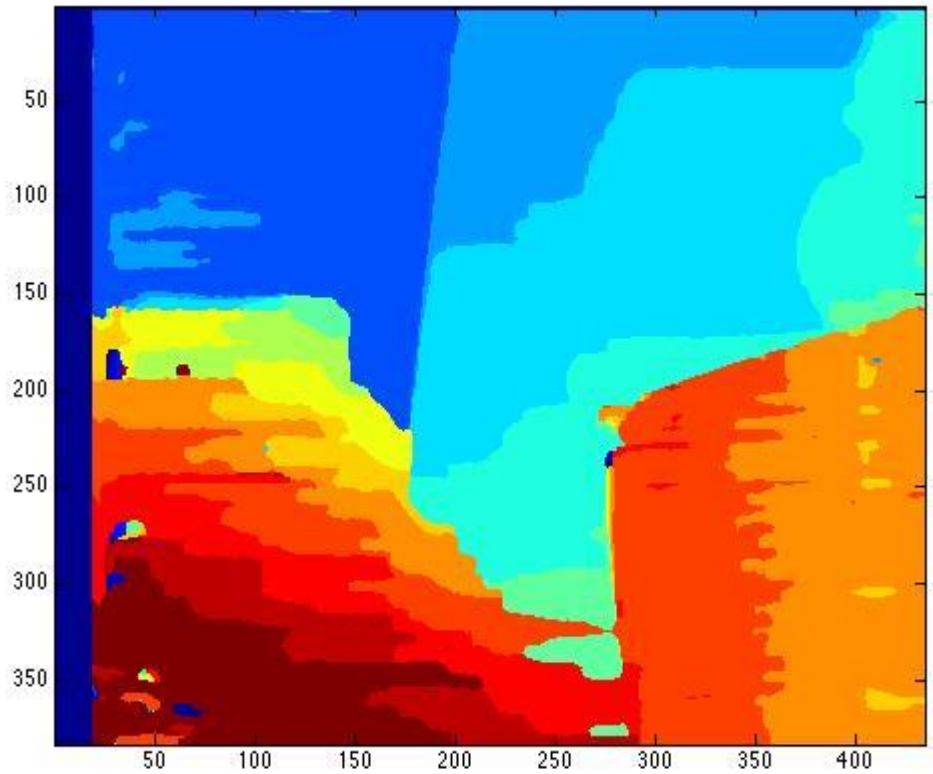
Outputs :



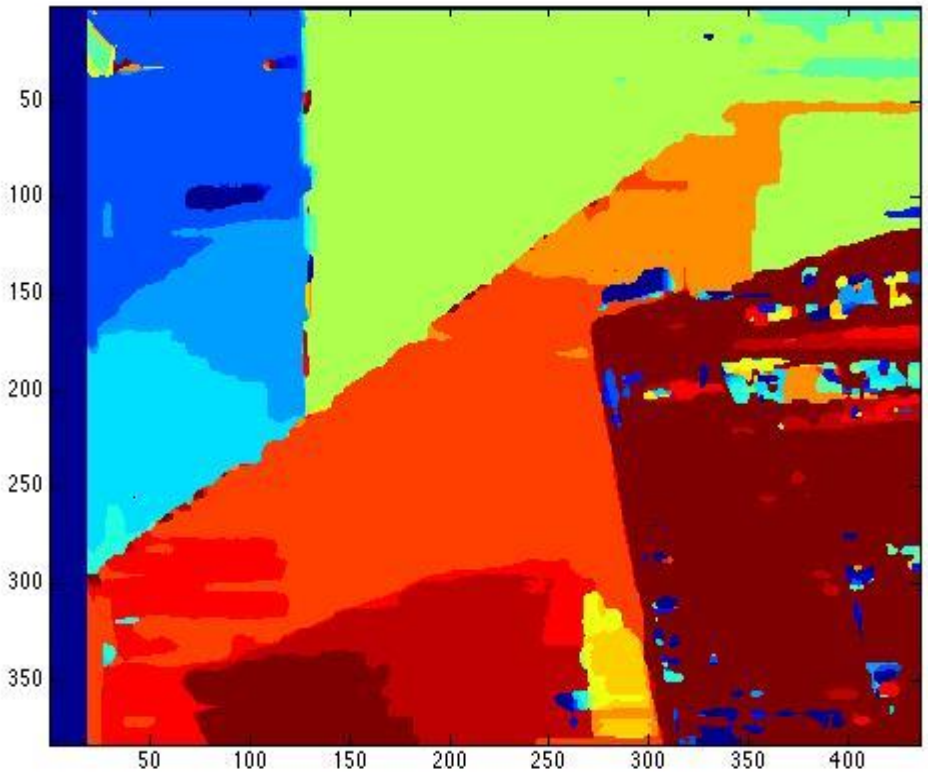


Outputs :

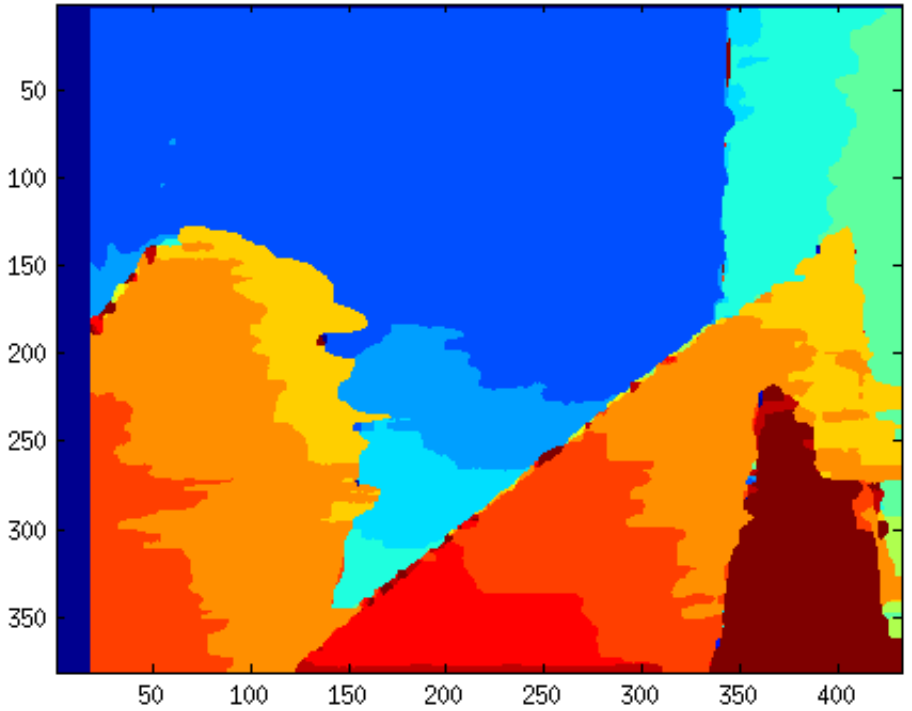




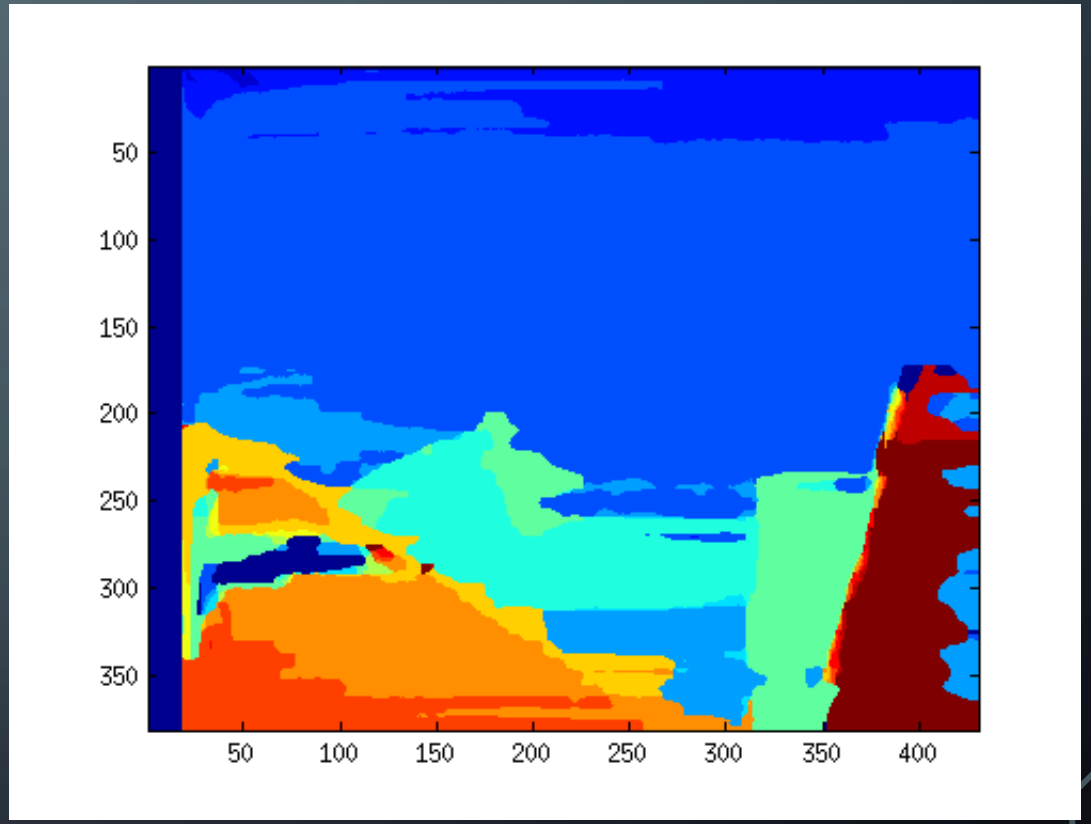
Outputs :



Outputs :



Outputs :



REFERENCES:-

- A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.
- Markov Random Fields with efficient approximations.
- http://vision.middlebury.edu/stereo/data/
- http://nghiaho.com/?page_id=1366