

## Missing Data Filling

1. `int ***readPPMImage(int imgsize[],char *filename,int band)`

Description : Reads a given image file with known number of bands and stores the data into a 3d array.

Type : Outside function

Input Arguments :

Imgsize[] : array of size 2 passed by reference. The function assigns the number of rows and columns to the two values.

Filename : file to be read

Band : number of bands passed in the image

Return :

Returns a 3d array containing the color values of the image –

Bands = 3 : rgb values corresponding to each pixel

Bands = 1 : only one dimension contains the values, others contain 0s.

2. `int ***cloud_detection(int imgsize[],int ***base, int n)`

Description : Detects the cloud pixels in the base image and returns an image with highlighted cloud pixels.

Type: Outside Function

Input Arguments:

Imgsize : 2 size array containing the rows and cols number of the image

Base: a 3d array containing the color values of the image containing the clouds

N : size of the structuring element (user input)

Return:

3d array containing cloud pixels as white and rest as black

3. `void dilate(int ***op, int ***&final_op, int imgsize[], int n)`

Description : Takes an image containing clouds and dilates each cloud region present in the image

Type : Outside function (support function of cloud\_detection function)

Input Arguments:

Op : image containing cloud pixels as white and rest black

Final\_op : image containing dilated cloud regions  
Imgsize : 2 size array containing the size of the image  
N : size of the structuring element (user input)

Return:  
Void

4. `int ***csf(int ***base, int ***aux, int ***cloud, int img1_size[], int img2_size[])`

Description : Takes the dilated cloud image and replaces the cloud region with data using the auxiliary image.

Type : Outside function

Input Arguments :

Base : a 3d array containing the color values of the image containing the clouds

Aux : a 3d array containing the color values of the image without the clouds

Cloud : a 3d array containing the dilated cloud areas as white

Img1\_size : dimension of base image

Img2\_size : dimension of auxiliary image

Return:  
3d array containing the color values after the filling of cloud regions

5. `float distance(int ***img1, int r1, int c1, int r2, int c2)`

Description : calculates euclidean distance between the color values of two given pixels

Type : Outside function

Input Arguments :

Img1 : 3d array containing auxiliary image color values

R1 : x coordinate of pixel in base image

C1 : y coordinate of the pixel in base image

R2 : x coordinate of pixel in aux image

C2 : y coordinate of pixel in aux image

Return:  
Float distance value

6. `int ***maxfilter(int ***op, int ***cloud, int imgsize[])`

Description : Applies Maximum Value occurrence Filter on the given image around the cloud regions only.

Type : Outside function

Input Arguments :

Op : 3d array containing the color values after the filling of cloud regions

Cloud : a 3d array containing the dilated cloud areas as white

Imgsize : dimension of image

Return:

3d array containing the color values after applying max-filter on cloud pixels

7. void writePPMImage(int \*\*\*op,int imgsize[],std::string fname)

Description : writes the given data into a ppm image.

type : Outside function

Input Arguments :

Op : 3d array containing the final color values

Imgsize[] : a 2 size array containing the rows and cols of the image to be written

Fname : name of the file to be written

Return:

Void