

Fusion

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

Type : Outside function

Input :

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

Output :

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 **prepareNewLabels(char *seg_result_label_file, int rows, int cols)`

Type : Outside function

Input :

seg_result_label_file : the segmentation labels result file provided by the meanshift algorithm

rows : rows of the image

cols : cols of the image

Output :

Returns new labels of the pixels in the image in a 2d array of size rows x cols

3. `void fillRegionNewLabels(int **olinfo, int **ninfo, int ol, int nl, int sx, int sy, int **mymap, int rows, int cols)`

Type: Outside Function (support function to prepareNewLabels function)

Input :

Olinfo : the original labels given by the segmentation algorithm

Nlinfo : passed by reference to store the new labels

Ol : old label of pixel (sx,sy)

Nl : new label of pixel (sx,sy)

Sx : x coordinate of the pixel

Sy : y coordinate of the pixel

Mymap : 2d flag array used in determining the start of a new label

Rows : rows of the image

Cols : cols of the image

Output:

Void

4. `vector< vector< int > > prepareObjectInfo(int rows, int cols, int **label_1, int **label_2, int **&dmerge, int &NO)`

Type : Outside function

Input:

Rows : rows of the image

Cols : cols of the image

Label_1 & label_2 : new labels of pan image

Dmerge : 2d flag array passed by reference utilized in identifying a new object

NO : number of objects in the image. To be calculated by this function. Passed by reference.

Output:

2d vector of size number_of_objects x 6 : containing the following information about each object :

Object number, area, xmin, xmax, ymin, ymax

5. `void fillRegionObjectInfo(int **dt1,int **dt2,int **&dmerge,int val1,int val2,int label,int bi,int bj,vector< int > &meta_obj,int rows, int cols)`

Type: Outside Function (support function to prepareObjectInfo function)

Input :

Dt1 & dt 2: the original labels given by the segmentation algorithm

dmerge : 2d flag array used in determining the start of a new object

val1 & val2 : object id of the current object

label : flag variable to check the existence of current object at a particular pixel

bi : x coordinate of the current pixel

bj : y coordinate of the current pixel

meta_obj : passed by reference. To be filled in this function by the information of each object

Rows : rows of the image

Cols : cols of the image

Output:

Void

6. void Fusion_GLSI(int **segmap, int ***&fimage, int ***msimage, int ***panimage, int onum, int xmin, int xmax, int ymin, int ymax,int bandnum)

Type : Outside function

Input :

Segmap : 2d array containing the labels of the image pixels

Fimage : passed by reference. To be populated by fused color values for each pixel

Msimage : 3d array containing MSS image color values

Panimage : 3d array containing PAN image color values

Onum : current object id

Xmin : minimum x coordinate of the bounding box of the current object

Xmax : maximum x coordinate of the bounding box of the current object

Ymin : minimum y coordinate of the bounding box of the current object

Ymax : maximum y coordinate of the bounding box of the current object

Bandnum : current band number

Output :

Void

7. void PrepareStatistics(int **segmap, int ***msimage, int ***panimage,int onum,int xmin,int xmax,int ymin,int ymax,int bandnum,float &msmean,float &msstd,float &panmean,float &panstd,int &panmin,int &panmax, int &msmin, int &msmax)

Type : Outside function (support function of Fusion_GLSI function)

Input:

Segmap : 2d array containing the labels of the image pixels

Fimage : passed by reference. To be populated by fused color values for each pixel

Msimage : 3d array containing MSS image color values

Panimage : 3d array containing PAN image color values

Onum : current object id

Xmin : minimum x coordinate of the bounding box of the current object

Xmax : maximum x coordinate of the bounding box of the current object

Ymin : minimum y coordinate of the bounding box of the current object

Ymax : maximum y coordinate of the bounding box of the current object

Bandnum : current band number

Msmean : passed by reference. Stores the MSS mean value of current object in the current band

Msstd : passed by reference. Stores the MSS standard deviation value of current object in the current band

Panmean : passed by reference. Stores the PAN mean value of current object

Panstd : passed by reference. Stores the PAN standard deviation value of current object
Panmin : passed by reference. Stores the minimum PAN value of the current object
Panmax : passed by reference. Stores the maximum PAN value of the current object
msmin : passed by reference. Stores the minimum MSS value of the current object in the current band
msmax : passed by reference. Stores the maximum MSS value of the current object in the current band

Output:

None

8. void GetStats(int **segmap, std::vector<int> data,int onum,float &mean,float &std, int &min, int &max,int xmin, int xmax, int ymin, int ymax, int nop)

Type : Outside function (support function of PrepareStatistics function)

Input :

Segmap : 2d array containing the labels of the image pixels

Data : 1d vector containing the color values of the image

Onum : current object id

mean : passed by reference. Stores the mean value of current object

std : passed by reference. Stores the standard deviation value of current object

Xmin : minimum x coordinate of the bounding box of the current object

Xmax : maximum x coordinate of the bounding box of the current object

Ymin : minimum y coordinate of the bounding box of the current object

Ymax : maximum y coordinate of the bounding box of the current object

Nop : number of pixels in the current object

Output :

Void

9. void GetHistogram (std::vector<int> data, std::vector<int> &val, std::vector<int> &cnt)

Type : Outside function (support function of GetStats function)

Input :

data: 1d vector containing the color values of the image

val : passed by reference. contains distinct color values In the image

cnt : number of distinct color values in the image

Output:

Void