

```
#ifndef CM_H
#define CM_H

//On Rivanna
//module load gsl
//gcc -O3 -o cov_mat cov_matrix.c -lgsl -lgslcblas -lm

typedef struct {
    float x;
    float y;
    float z;
} point_rgb;

typedef struct {
    int width;
    int height;
    point_rgb *const *pc;
} point_cloud;

void initialize_pc(const int m, const int n, point_rgb *const *pc);
void calc_cov_mat(int m, int n, float ***int_imgs, float** dt_map);
float convert2gray(point_rgb point);
void die(const char *error);
float area(float ** int_img, int x, int y);
float **generate_integral_image (point_rgb *const *pci, int m, int n, int code);
float dc_smoothing_area(point_rgb p, point_rgb p1, point_rgb p2, float scale_factor);
float dd_smoothing_area(point_rgb point, float beta);
float **allocate_float_array(const int m, const int n);
point_rgb **allocate_pc(const int width, const int height);
float **dtBin(float **dci, int m, int n);
void dt2D(float **dci, int m, int n);
float *dt1D(float *f, int n);
float **smoothing_windows_map(point_rgb *const *pc, int m, int n, float ** dt_map, int beta);
float avg(float ** int_img, int m, int n, float radius);
int max(int a, int b);
const double alpha = 0.0028;
point_cloud cloud;
float window_average(float ** int_img, int m, int n, float r);
void compute_evec(float cov_mat[3][3]);

#endif
```