

demo3D
threshold: int
show_ground_plane_plot: bool
run_all: bool
void
dir_data: String
dir_files: struct
data: struct
n_superquadrics: int

functionName
arguments: type
return: type
function_variables: type

extract2DFrom3D
data: struct
show_ground_plane_plot: bool
x2Dscan: float[1..*]
y2Dscan: float[1..*]
data_container_above: float[1..*][4]
rotated_points[3][1..*]
x, y, z: float[1..*]
data_container: float[1..*][4]
paramval: float[3]
nx, ny, nz: float
nnorm: float
d: float
threshold: float
pos: float[1..*]
indices_above: float[1..*]
z: float[1..*]
plane_normal: int[3]
lambda: float[3]
norm_lambda: int
beta_ang: float
e: float
e_0: float[3]
R: float[3][3]
rotated_xy: float[2][1..*]
min_x, max_x, min_y, max_y: float
grid_size: int
delta_x, delta_y: float
points_x, points_y: float[1..*]
indices_x, indices_y: int[1..*]
equ_image: bool[1..points_x][1..points_y]
equ_image_trf: bool[1..points_x][1..points_y]
image_connected: int[1..points_x][1..points_y]
nregions: int
linreg, colreg: int[1..*]
w_regionpix: struct
w_mean_regionpix: float[1..*]
x2Dscan_reg, y2Dscan_reg: struct
sorted_mean_region_idx: int[1..*]

objFunGroundPlane
paramval[11]
data_container: float[1..*][4]
show_plot: bool
errfit: float
nx, nt, nz: float
d: float
pos: float[1..*]
dimsys: int
err_side: float
err_fit1: float

shapeSuperquadric
x2Dscan: float[1..*]
y2Dscan: float[1..*]
data_container_above: float[1..*][4]
rotated_points[3][1..*]
threshold: int
n_superquadrics: int
x2Dv, y2Dv: float[1..*]
all_index_min: int[1..*]
n_possible_sqs: int
data_for_3D: struct
fval: float
x2Dvseg, y2Dvseg: float[1..*]
index_min: int
n_segmentations: int
npoints_nextseg: int
paramval2D: float[6]
corrected: int
index_mom: int
corrected_mon: int
superquadric_params: struct
all_data_above: float[1..*][4]
all_data_project: float[1..*][3]
absorve: int[1..*]
n_superquad: int
theta_bound: float[4]
paramval2D_plot float[6]
a: float[1..*]
meanval: float[1..*]
theta: float
fepsilon2: float
Rmat: float[2][2]
x2Dvest, y2Dvest: float[4]
rotatedxy: float[2][4]
x2Dvm, y2Dvm: float[4]
F1_2d, F2_2d, F: float[4]
lcurr, npoints, lnew: int
paramval: float[11]
epsilon: float[1..*]
n: int
etamax, etamin: float;
wmax, wmin: float,
deta: float
dw: float
eta: float[1..*][1..*]
w: float[1..*][1..*]
x_n, y_n, z_n: float[1..*]

minimizeSuperquadric2D
x2Dvseg: float[1..*]
y2Dvseg: float[1..*]
paramval2D: float[6]
fval: float
X_tolerance: float
fun_tolerance: float
theta: float
epsilon2: float
max_x, max_y: float
min_x, min_y: float
a1_lb, a1_ub: float
a2_lb, a2_ub: float
avg_x_lb, avg_x_ub: float
avg_y_lb, avg_y_ub: float
theta_lb, theta_ub: float
epsilon2_lb, epsilon2_ub: float
avg_x, avg_y: float
sgn_avg_x, sgn_avg_y: int
norm_x2Dvseg, norm_y2Dvseg: float
a: float[2]
paramval2D_init: float[6]
lb, ub: float[6]

objFunFit2D
paramval2D: float[6]
x2Dv: float[1..*]
y2Dv: float[1..*]
error_fit_2d: float
a: float[2]
meanval: float[2]
theta: float
epsilon2: float
ctheta: float
stheta: float
Rmat: float[2][2]
x2Dvm: float[1..*]
y2Dvm: float[1..*]
rotate_d_xy[2][1..*]
F1_2d: float[1..*]
F2_2d: float[1..*]
F: float[1..*]
grad_x: float[1..*]
grad_y: float[1..*]
rotate_d_grad[2][1..*]
grad_r: float[1..*]
dimsys: int
err_grad: float
rzero: float[1..*]
err1: float

gradF2D
F1, F2: float[1..*]
xm, ym: float[1..*]
epsilon_est: float
grad_x: float
grad_y: float
threshold: float

superquadricCheckInnerPoint
x2Dvseg: float[1..*]
y2Dvseg: float[1..*]
paramval2D: float[6]
x2Dv: float[1..*]
y2Dv: float[1..*]
index_min: int
corrected: int
index_mom: int
corrected_mom: int
a: float[2]
theta: float
epsilon2: float
ctheta: float
stheta: float
Rmat = float[2][2]
x2Dvm_e: float[1..*]
y2Dvm_e: float[1..*]
rotate_d_xy: [2][1..*]
x2Dvm_i: float
intermediate_point: int
F1_2d, F2_2d: float
F: float
F1_2d_e, F2_2d_e: float
F_e: float
grad_x, grad_y: float
grad_x_p1, grad_y_p1: float
split_mom: float
xmin, ymin: float
x1, x2: float
y1, y2: float
d1, d2: float
test, test_e: float

minimizeSuperquadric3D
paramval2D_plot: float[6]
data_for_3D: struct
paramval: float[11]
fval: float
X_tolerance: float
fun_tolerance: float
fun_evaluations: int
lambda: int[3]
mu3: float
a: float[3]
epsilon: float[2]
meanval: float[2]
beta_angle: float;
x_noisy, y_noisy, z_noisy: float[1..*]
a_lb, a_ub: float[3]
epsilon_lb, epsilon_ub: float[2]
meanval_lb, meanval_ub: float[2]
mu3_lb, mu3_ub: float
lambda_lb, lambda_ub: int[2]
beta_angle_lb, beta_angle_ub: float
parameters: float[11]
lb, ub: float[11]

objFunSQ
paramval: float[11]
x, y, z: float[1..*]
quaderr: float
beta_ang: float
lambda: int[3]
a_est: float[3]
epsilon_est: float[2]
meanval: float[3]
xm, yx, zm: float[1..*]
e_0: float
e: float[1..0]
R: float[3][3]
rotatedxyz: float[3][1..*]
rzero: float[1..*]
F1, F2, F3, F: float[1..*]
gradx, grady, gradz: float[1..*]
rotate_grad: float[3][1..*]
gradr: float[1..*]
dimsys: int
errgrad: float
quaderr1: float

gradF3D
F1, F2, F3: float[1..*]
xm, ym, zm: float[1..*]
epsilon_est: float
grad_x: float
grad_y: float
grad_z: float
threshold: float