Mex conversion to regular c++:

Regular use of mex:

void mexFunction( int nlhs, mxArray \*plhs[], int nrhs, const mxArray \*prhs[] )

nlhs – number of requested output result variables

plhs – returns the results as , for example :

plhs[ 0 ] = sourcesMxArray;

plhs[ 1 ] = targetsMxArray;

nrhs – number of input parameters

prhs – usage like :

mxIsCell( prhs[ 0 ] )

frames = mxGetNumberOfElements( prhs[ 0 ] );

if( mxGetClassID( mxGetCell( prhs[ 0 ], frame ) ) != mxUINT32\_CLASS )

if( mxGetClassID( prhs[ 1 ] ) != mxDOUBLE\_CLASS )

height = mxGetM( mxGetCell( prhs[ 0 ], 0 ) );

width = mxGetN( mxGetCell( prhs[ 0 ], 0 ) );

but most important is :

superpixels = ( unsigned long long )( ( double \* )mxGetData( prhs[ 1 ] ) )[ 0 ];

superpixelMap = ( unsigned int \* )( mxGetData( mxGetCell( prhs[ 0 ], frame ) ) );

so we need to identify which variables are populated by mxGetData , and set them as initial input to the function. And plhs[] as the outputs as refs.