

```

#ifdef _WIN32
#include <windows.h>
#endif
#include <stdio.h>
#include <stdlib.h>
#ifdef __APPLE__
#include <GL/gl.h>
#include <GL/glut.h>
#else
#include <OpenGL/gl.h>
#include <GLUT/glut.h>
#endif
#include <AR/gsub.h>
#include <AR/video.h>
#include <AR/param.h>
#include <AR/ar.h>

```

Mac OS 9 OpenGL + GLUT Frameworks

OS X OpenGL + GLUT Frameworks

AR Frameworks/Libraries Shared between gsub + Lite

```

/*****
// modified by Thomas Pintaric, Vienna University of Technology

```

```

#ifdef _WIN32
char *vconf = "flipV,showDlg"; // see video.h for a list of supported parameters
#else
char *vconf = "";
#endif
/*****

```

```

int xsize, ysize; // Video Size
int thresh = 100; // Pattern Recognition Threshold
int count = 0;

```

```

char *cparam_name = "Data/camera_para.dat"; // where camera info is.
ARParam cparam; // ? Config?

```

```

char *patt_name = "Data/patt.hiro"; // where pattern info is (comp)
int patt_id; // ? THIS IS codes each marker
double patt_width = 80.0; // 80 mm?
double patt_center[2] = {0.0, 0.0}; // ? ID Array, 2 elements
double patt_trans[3][4]; // ?

```

```

static void init(void);
static void cleanup(void);
static void keyEvent( unsigned char key, int x, int y);
static void mainLoop(void);
static void draw( void );

```

```

int main(int argc, char **argv)
{

```



setup  
The internal procedures/symbols

- procedure  
what is diff between proced. & symbol?

init();

arVideoCapStart(); *- start video capture?*  
argMainLoop( NULL, keyEvent, mainLoop );  
return (0);

static void keyEvent( unsigned char key, int x, int y)

{  
/\* quit if the ESC key is pressed \*/  
if( key == 0x1b ) {  
printf("\*\*\* %f (frame/sec)\n", (double)count/arUtilTimer());  
cleanup(); *- close video ports*  
exit(0);

*if esc pressed print fps stats + quit*  
*main() { X = RND; GetCos(X); }*

/\* main loop \*/  
static void mainLoop(void)

{  
ARUint8 \*dataPtr; *- data pointer*  
ARMarkerInfo \*marker\_info;  
int marker\_num;  
int j, k; *- loop index variables*

/\* grab a vide frame \*/  
if( (dataPtr = (ARUint8 \*)arVideoGetImage()) == NULL ) {  
arUtilSleep(2); *- Slowdown, wait for an image (2ms?)*  
return;

*pass to window?*  
*Get (var)*  
*Var 2*

if( count == 0 ) arUtilTimerReset();  
count++;

*Must be re-rendered by an ~~arUtil~~ = Cos(VAR) program.*

argDrawMode2D(); *- Draws The video*  
argDispImage( dataPtr, 0,0 ); *- Display video image*

/\* detect the markers in the video frame \*/  
if( arDetectMarker(dataPtr, thresh, &marker\_info, &marker\_num) < 0 ) {  
cleanup();  
exit(0);

*Threshold for detecting Markers*  
*Array returned for marker info*  
*# of Markers detected*  
*find markers, exit if error*  
*Video image in*

arVideoCapNext(); *- gets next frame?*  
/\* check for object visibility \*/  
k = -1;

for( j = 0; j < marker\_num; j++ ) {  
if( patt\_id == marker\_info[j].id ) {  
if( k == -1 ) k = j;

*Loop through as many Markers as if loaded were detected.*  
*Set k to current marker if first time through*

```

else if( marker_info[k].cf < marker_info[j].cf ) k = j;
}
}
if( k == -1 ) {
  argSwapBuffers();
  return;
}
/* get the transformation between the marker and the real camera */
arGetTransMat(&marker_info[k], patt_center, patt_width, patt_trans);
draw();
argSwapBuffers();
}

```

If k has already reassigned then  
 if no markers if last marker of lesser confidence than  
 then or wrong current marker - if so then  
 Show non-augmented image reassigned k to the current  
 found marker at higher confidence

```

static void init( void )
{
  ARParam wparam;
  /* open the video path */
  if( arVideoOpen( vconf ) < 0 ) exit(0);
  /* find the size of the window */
  if( arVideoInqSize(&xsize, &ysize) < 0 ) exit(0);
  printf("Image size (x,y) = (%d,%d)\n", xsize, ysize);
  /* set the initial camera parameters */
  if( arParamLoad(cparam_name, 1, &wparam) < 0 ) {
    printf("Camera parameter load error !!\n");
    exit(0);
  }
  arParamChangeSize( &wparam, xsize, ysize, &cparam );
  arInitCparam( &cparam );
  printf("*** Camera Parameter ***\n");
  arParamDisp( &cparam );
  if( (patt_id=arLoadPatt(patt_name)) < 0 ) {
    printf("pattern load error !!\n");
    exit(0);
  }
}

```

Draw augmented content over video image in rear buffer  
 2x4 array Var (4x3 array)  
 just 2 element ID array?  
 Show Augmented Image  
 ? window ARVideoWin, 32.lib  
 - If -1 then couldn't open + exit  
 otherwise print dimensions + exit.  
 if paramLoad returns -1 then something wrong with loading camera parameters.  
 Reconcile window size to SEC  
 AR Library. Do stuff manually  
 Both assigns patt\_id - see through  
 determine 50 bmp loaded ok into  
 pattern matching array - if didn't  
 then print error msg + exit  
 L. 129

```

/* open the graphics window */
argInit( &cparam, 1.0, 0, 0, 0, 0 );
}
/* cleanup function called when program exits */
static void cleanup(void)
{
  arVideoCapStop(); - stop video capture?
}

```

```

arVideoClose();
argCleanup();
}

static void draw( void )
{
  double   gl_para[16];
  GLfloat  mat_ambient[] = {0.0, 0.0, 1.0, 1.0};
  GLfloat  mat_flash[]   = {0.0, 0.0, 1.0, 1.0};
  GLfloat  mat_flash_shiny[] = {50.0};
  GLfloat  light_position[] = {100.0, -200.0, 200.0, 0.0};
  GLfloat  ambi[]        = {0.1, 0.1, 0.1, 0.1};
  GLfloat  lightZeroColor[] = {0.9, 0.9, 0.9, 0.1};

  argDrawMode3D();
  argDraw3dCamera( 0, 0 );
  glClearDepth( 1.0 );
  glClear(GL_DEPTH_BUFFER_BIT);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LEQUAL);

  /* load the camera transformation matrix */
  argConvGlpara(patt_trans, gl_para);
  glMatrixMode(GL_MODELVIEW);
  glLoadMatrixd( gl_para );

  glEnable(GL_LIGHTING);
  glEnable(GL_LIGHT0);
  glLightfv(GL_LIGHT0, GL_POSITION, light_position);
  glLightfv(GL_LIGHT0, GL_AMBIENT, ambi);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightZeroColor);
  glMaterialfv(GL_FRONT, GL_SPECULAR, mat_flash);
  glMaterialfv(GL_FRONT, GL_SHININESS, mat_flash_shiny);
  glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
  glMatrixMode(GL_MODELVIEW);
  glTranslatef( 0.0, 0.0, 25.0 );
  glutSolidCube(50.0);
  glDisable( GL_LIGHTING );

  glDisable( GL_DEPTH_TEST );
}

```

*close video paths*  
*external cleanup procedure possibly for other mysterious things*  
*variables for 3D env. specs*  
*Material specs.*  
*lighting specs*  
*setting up 3D Augmented content drawing.*  
*orientation of marker/Augmented object.*

*Not Over*

*Not Over*

*Open GL*  
*3D drawing environment*  
*Mumbo - Jumbo*  
*(executing the settings from above)*  
*using the variables*  
*Graphics Library Utility Toolkit draws cube function size = 50 (30 less than Marker Size)*