OpenCV + α　自習用資料

2006 Apr.13 梶本裕之
2007 Apr.21 改変
2008/12/30 改変
2010/03/22 改変
2010/07/24 改変

ChangeLog

2008/12/30

・OpenCV が cvcam 関数をサポートしなくなったのに対処

・研究上良く使う「ビデオストリームの再生」を追加．音にも対応．

・GLUI と OpenAL のサンプルを追加

2010/3/22

・OpenCV2.0 へのバージョンアップへの対応

2010/07/24

・OpenCV2.1 へのバージョンアップへの対応．ライブラリの自作が不要となった．

前提

この自習教材の内容は，C 言語，OpenGL および GLUT に関してはテクスチャマップまで自習済みであることを前提としています。対象者は研究室の新入生です．

GL 関係に関しては、下記で自習してください．和歌山大学床井先生の有名なページです．
手抜き GLUT 入門：http://www.wakayama-u.ac.jp/~tokoi/opengl/libglut.html
これはテクスチャを扱っていないので，同じ床井先生が書かれている下記テキスト等で自習してください．
http://marina.sys.wakayama-u.ac.jp/~tokoi/?date=20040913

情報源

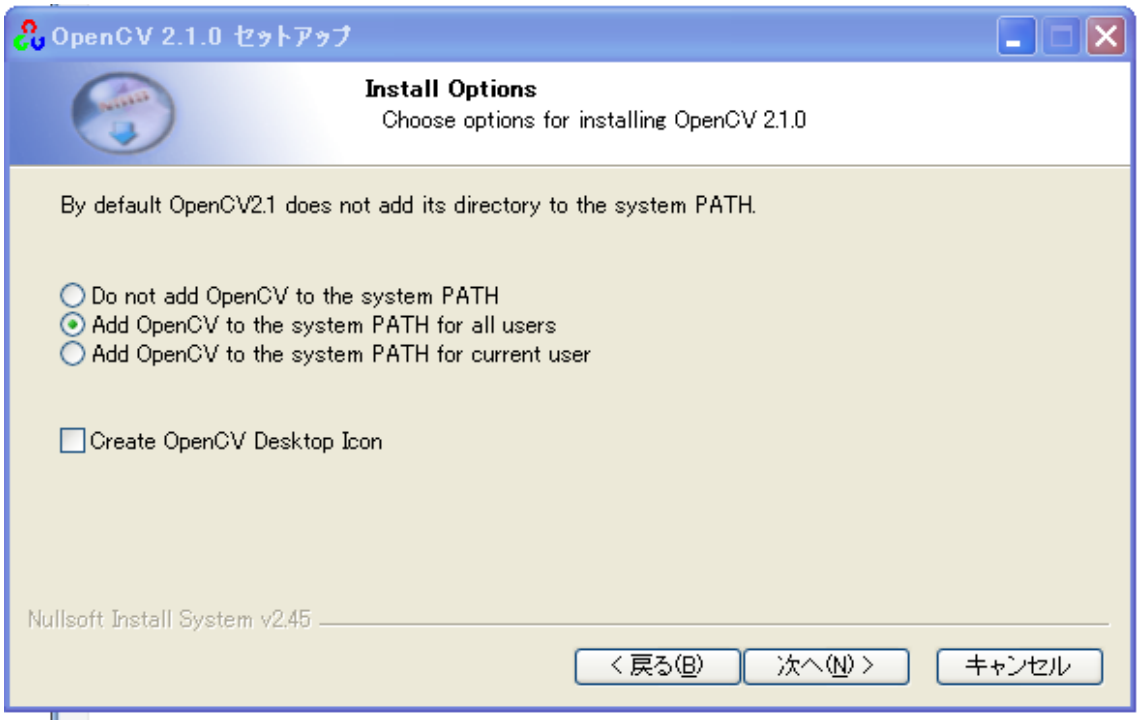
OpenCV に関しては日本語の情報も増えてきました．　http://opencv.jp/
Yahoo のニュースグループ(英語)もあります．

インストール

GLUT と OpenCV をインストールしてください。

- GLUT 「手抜き GLUT 入門」を参照

- OpenCV　http://opencv.jp/　経由で.
  - OpenCV-2.0 ではなく、OpenCV-2.1.0-win32-vs2008.exe をインストールする。これは Visual Studio によるライブラリコンパイル済みバージョン。これでないとライブラリを自分で構築する必要があり、かなり大変。
    http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.1/OpenCV-2.1.0-win32-vs2008.exe/download
  - インストール時は“Add OpenCV to the system PATH for all users”を選択。



設定

Visual Studio の設定（一回だけ行えばよい）

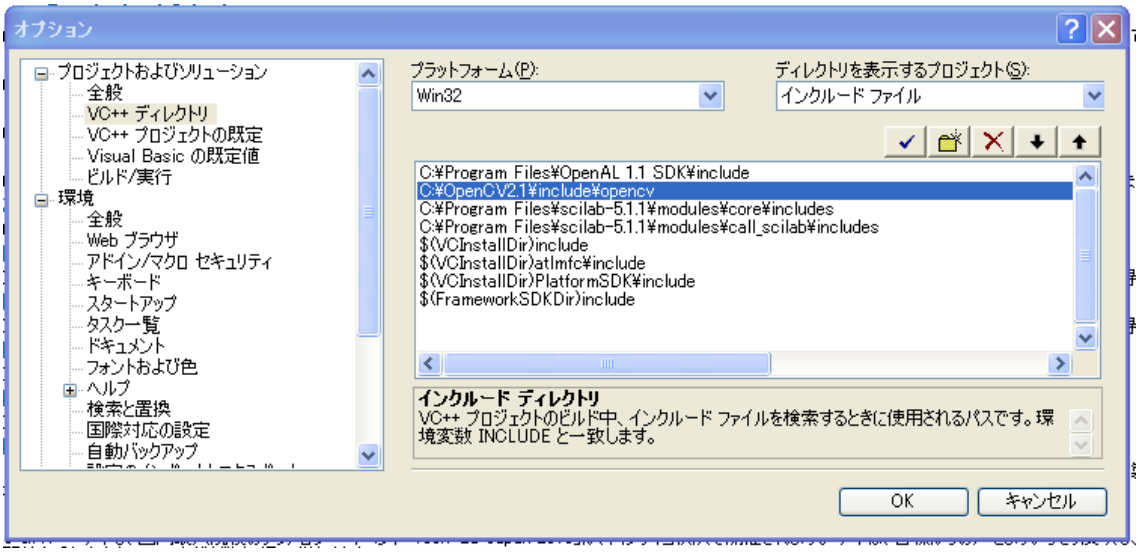
１．include ファイルの存在するフォルダを設定する。

ツール⇒オプション⇒プロジェクト⇒VC++ディレクトリを選択

「インクルードファイル」を選択

C:¥OpenCV2.1¥include¥opencv

を追加。



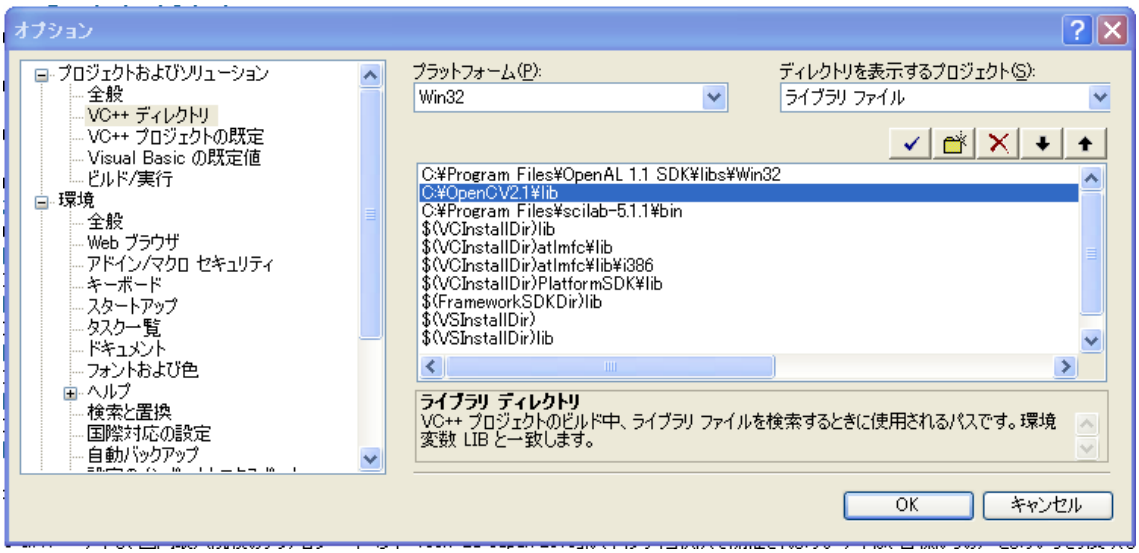
2.lib ファイルの存在するフォルダを設定する

ツール⇒オプション⇒プロジェクトおよびソリューション⇒VC++ディレクトリを選択

「ライブラリファイル」を選択.

C:¥OpenCV2.1¥lib

を追加



１， ２共に、OpenCV のための設定であるが、glut のための設定を行ってもよい。サンプルでは glut はライブラリも include ファイルもプロジェクトと同じ場所においているため設定していない。

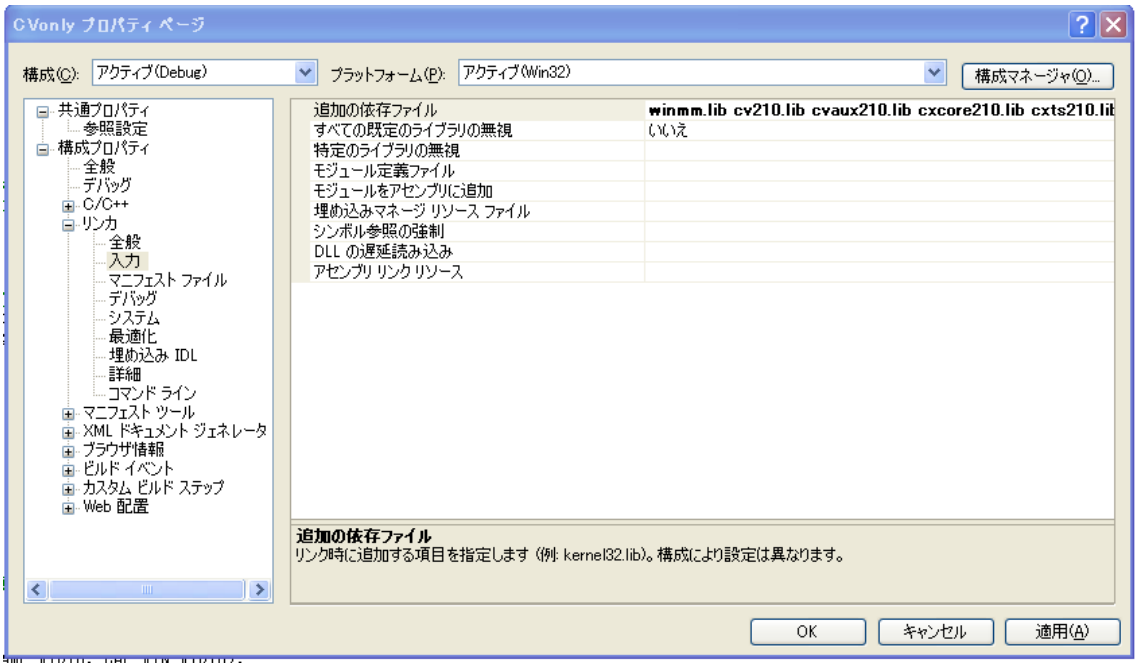
プロジェクトの設定（各プロジェクトで行う。サンプルプロジェクトではすでに設定済み）

プロジェクト⇒プロパティ⇒リンカ⇒入力を選択

追加の依存ファイルとして下記のものを記入

winmm.lib cv210.lib cvaux210.lib cxcore210.lib cxts210.lib highgui210.lib ml210.lib opencv_ffmpeg210.lib glut32.lib

Visual Studio の設定で OpenCV のライブラリの存在場所を教えているので、各ファイルを絶対パスで指定する必要がなくなっている。これらのライブラリをプロジェクトのソースと同じ場所においておいてもよい．プロジェクトの設定として行わなくても，ソースの中に#pragma で指定することもできる．



DLL ファイルのコピー

DLL ファイルをコピーする．

講習では後述する OpenAL、GLUI も使うので、これらの DLL も含めて DLL フォルダに入れた。これをまるごと C:¥ WINDOWS¥system32 にコピーすればよい．

課題：

exe(実行ファイル)と lib（静的リンクライブラリ）と DLL（動的リンクライブラリ）のそれぞれの役割について調べる．

サンプルの中身と課題

CVonly

OpenCV で画像を取得する部分のみの簡単なプログラムです。
以前は cvcam という関数で画面のキャプチャが windows のイベントとして扱われていましたが，廃止されました．highGUI 関数を使ったサンプルになっています．
また画像データに関してピクセルごとにアクセスする方法のサンプルにもなっています。

課題：

（１）CVonly 中の pixel_access_example を改変し，横に走っている格子縞を縦にする．
（２）CVonly 中の pixel_access_example を改変し，各ピクセルの青成分と赤色成分を入れ替える．
（３）CVonly 中の pixel_access_example を改変し，全体的に暗くしてみる．
（４）（２）は実は自分で関数を書かなくても関数が用意されている．次の関数を試す．

```
cvCvtColor(frame, frame, CV_BGR2RGB);
```

（５）（４）を参考に，画像を平滑化せよ．cvSmooth を使う．

GL_Jpeg_Texture

OpenGL 上でテクスチャ画像を貼り付けるサンプルです。通常、GL は画像を読み込むルーチンがないため、jpeg などの画像を扱うには非常に苦労します。libjpeg などを使うのが一般的ですが敷居は高いです。
しかし実は OpenCV のほうで、画像を読み込む関数が用意されています。これを使えば簡単に jpeg ファイルを読み込み、テクスチャとして貼り付けられます。

課題：
GL_Jpeg_Texture を改変し，別の画像を提示する．

CVandGL

OpenCV と OpenGL を組み合わせるプログラムです。
CV で得た画像を GL 上でテクスチャとして貼り付けています。カメラキャプチャは GL の idol 関数中で行っています．他のもっと良い方法があるかもしれません．また正確な時間更新のためには glutTimerFunc を使う必要があるでしょう．

課題：
（１）GL で回転する立方体を書く。これは「手抜き OpenGL 入門」をなぞればできます。
（２）立方体の各面にビデオ映像が表示されるようにする。

CVandGL_応用(1)エッジ抽出

OpenCV の画像処理機能を使った OpenCV らしいプログラムです。ここではエッジ抽出フィルタを作っています。実質２行程度で出来ていることがわかるでしょう。

CvandGL_応用(2)色抽出

OpenCV の画像処理機能を使った OpenCV らしいプログラムその２です。画像のピクセルごとの加減乗除が一つの命令で出来ることがわかると思います。

ここでは色抽出フィルタを作っています。スペースキーを押したときに画像中心にある物体の色をキーとして抜き出します。上下矢印キーで閾値を設定できます。

「色空間」に関して、「RGB」ではなく「YCrCb」を採用し、色の比較を CrCb 平面状で行っていることに注目してください。これにより、照明の明るさに依存しにくい色比較が出来ます。

CvandGL_応用(3)背景差分

その３です。スペースキーを押した時の画像をもとに、それとの差分だけを表示します。初めに青色スクリーンを提示しておくなどすると、背景差分でブルーバック合成ができます。web カメラにありがちな自動追尾機能等があるとうまくいかないので，切ってください.

課題：

この辺は今の段階では動かしてみる程度で結構です.

とにかく非常にたくさんの関数があります。下手な画像処理を自分で書くことを考えず、OpenCV の関数を使うことを心がけましょう。

CvandGL_応用(4)他の PC への画像転送

CvandGL_応用(5)他の PC からの画像転送

共有ファイルを用いた通信です。

送信側プログラムは OpenCV でカメラ画像を取り込み、それを fwrite でファイル先頭から書き込んできます。100ms ごとに更新しています．サンプルでは自分自身のプロジェクトのあるフォルダ中の imgdata というファイルに書き込む設定になっていますが，このファイルをネットワーク上の共有ファイルにすることによってデータを通信出来ます．正統的とは言えない方法ですが, LAN 内であれば 1ms 程度の遅延で通信できてしまうので実験レベルでは十分実用になります. なにより ANSI-C の知識だけで出来てしまうのが利点です.

受信側プログラムでは OpenCV は画像の色変換にだけ使っています． fread でファイルから読み込んだ画像データを OpenGL でテクスチャとして描画しています。GL では glutTimerFunc を使うことによって 50ms ごとに画像を更新しています。サンプルでは自分自身のプロジェクトのあるフォルダ中の imgdata ファイルを読みだす設定になっていますが，これを先ほどの送信プログラムと合わせることで通信出来ます.

課題：
自分の PC の中でこの二つのプログラムを動かし, 画像の転送実験を行う. 例えば書き込み, 読み出しファイルとして c:¥imgdata を指定すればよい. （C 言語のソースの中では c:¥¥imgdata となることに注意）.
もし 2 台の PC がある場合は PC 間の画像の転送実験を行う.

CvandGL_応用(6)ムービー読み出し

OpenCV は動画ファイルを扱うことが出来ます. これによって例えば web カメラのない環境でもテストすることが出来ますし，動画を書きだすこともできます. この例は動画読み出しの例です. フレーム単位で制御することが出来ます.

課題：
この辺は今は動かしてみる程度で結構です. 余裕があればカメラ画像のムービーファイルへの書き出しを試みてください.

その他の OpenGL，GLUT ライクなライブラリ

少し凝ったアプリケーションを作ろうとすると GLUT と OpenCV では足りないものが出てきます. 特にサウンドと GUI が無いのが気になります.

サウンドは例えば DxLib でも簡単に扱えますが, 3D 空間を運動するようなものは難しいでしょう. DirectSound は敷居が高く，Vista 以降では推奨されないようです.

GUI は Windows のプログラミングに習熟するか，Visual Basic，C#等の Microsoft 推奨環境に移行すれば容易ですが，オープンプラットフォーム性は失われます.

そこでここまでの経験を生かせる二つのライブラリ，OpenAL と GLUI を紹介します.

インストールと情報

- OpenAL
  http://connect.creativelabs.com/openal/Downloads/Forms/AllItems.aspx
  (coreSDK と ALUT をインストール)
  仕様書の日本語訳
  http://www.memorize-being.net/releases/oal11spec-ja/
  サンプルソース等. 一番の情報源
  http://www.devmaster.net/articles.php?catID=6

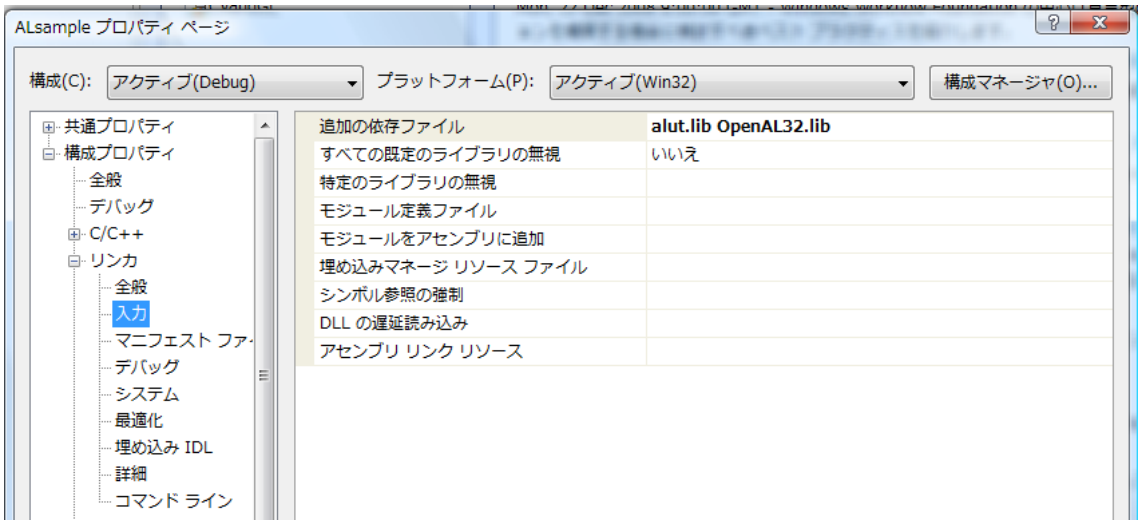
- GLUI
  http://www.cs.unc.edu/~rademach/glui/src/release/glui_v2_1_beta.zip
  阪大の日浦先生がイントロを書かれています.
  http://www-sens.sys.es.osaka-u.ac.jp/wakate/tutorial/group3/glui/
  NAIST の千原研にもイントロがありました.
  http://chihara.naist.jp/people/2004/kenta-t/OpenCV/pukiwiki/index.php?%B9%E2%C5%D9%A4%CAGUI
  オフィシャルの日本語訳
  http://ktm11.eng.shizuoka.ac.jp/glui/glui.html

ALsample

OpenAL およびその上にかぶせる ALUT という 3D サウンドライブラリのサンプルです. ALUT は GLUT と極力似せて作られているのですんなりと理解できます. いまいち普及していませんが, ドップラー効果まで計算される優れモノです.

OpenAL SDK と ALUT をインストールしたのち, 環境設定をする必要があります. CV の場合と同様に, インクルードファイルのパス, ライブラリのパスをそれぞれ設定します.
C:¥Program Files¥OpenAL 1.1 SDK¥include
C:¥Program Files¥OpenAL 1.1 SDK¥libs¥Win32

また使用するライブラリはプロジェクトの設定中で書いておく必要がある点は OpenCV と同じです. 今回は alut32.lib と OpenAL32.lib を設定しています.



GLUIsample

OpenGL 用のインタフェースとして標準的な GLUI のサンプルです．こちらもいまいち普及していないように見えますが研究室レベルでは使っているところは多いです．
心理実験を行う場合，GLUI はスライダ（トラックバー）が無いので被験者の回答に使いにくいかもしれません．ただしスライダは OpenCV の方でサポートしているので相補的と言えなくもない？
GLUI をインストールしようとするとライブラリをコンパイルしなければならず面倒です．このサンプル中にライブラリ(glui32.lib)を置いておきましたのでこれを使うとよいでしょう．

また使用するライブラリはプロジェクトの設定中で書いておく必要がある点は OpenCV や OpenAL と同じです．今回は glut32.lib と glui32.lib を設定しています．

CVandALandUI

OpenCV と OpenAL と GLUI を合わせたサンプルを一応作りました．
このくらい沢山のライブラリを使うものになると，プロジェクトへのライブラリファイルの登録，Visual Studio の環境へのインクルードファイルのパス指定，DLL 置き忘れ等のミスが多発しますので注意．
サンプルでは音源が左右に 2 つあり，移動すると変化が分かります．非常によくある罠として，**OpenAL ではモノラルの音源を 3 次元的に移動することが出来ますが，ステレオの音源は BGM として扱われます．** Audacity 等を使ってモノラル化してから使いましょう．
http://audacity.sourceforge.net/?lang=ja
http://www.sps.sie.dendai.ac.jp/old/audacity/menu3.html

OpenAL，GLUI 以外のライブラリ

沢山あるようです．特にクロスプラットフォームな UI は X window の流れのものが一通りありますし，UI 部分だけ別の言語で(VB とか)作るとか，いっそのこと UI 部だけ独立したプログラムにしてしまって本体プログラムと通信しあうとか．いくらでも方法はあります．研究室として統一する必要のある部分ではないので，なるべく楽な方法を身に付けましょう．

最終課題：
OpenGL+OpenCV＋OpenAL＋GLUI で，何か作る．動画と音と UI があればこれらにこだわらなくても良い．勉強なので実装する機能が多いほどよい．せっかくのマルチメディアなので美しいほうが良い．作りこみましょう．

その他の話題

触覚ディスプレイとの融合（ループ周期の問題）

GLUT を使う場合，普通は glutMainLoop で glutTimerFunc や glutIdolFunc を使ってループを回しますが，これは触覚ディスプレイにとってはあまりにも遅いループになります．

触覚ディスプレイの **1kHz～10kHz のループ**と，これまでに紹介した**高々数十 Hz のループ**を共存させるには，通常「スレッドプログラミング」を行います．
ただしこのスレッドプログラミングにも穴はあります．研究上では最も固い方法として PC を分けてしまい，共有メモリによる通信を行う方法がとられることもあるくらいです．
http://www.interface.co.jp/catalog/selection/memolink/memolink.asp

簡単な方法として，課題で行ったように**一つの PC の中で二つのプログラムを動かし，プログラム間の通信を前述のファイルで行う**，というのがあります．これだと研究用の触覚のみのプログラムをほとんど改変せずにデモ用のアプリを組めるので，研究室的には多用しています．

複雑なモデルをどのように生成するか

OpenGL で複雑なポリゴン形状をどのように作るか、というのは長年決着のついていない問題です．（DirectX だと、"X-file"というフォーマットがあり、多くの 3D モデリングソフトがその形式で出力してくれるので問題ありません）LightWave 形式や 3DMF、DXF, STL 形式等を解釈するライブラリが色々落ちていますのでそれらを使うことになります。標準化をしないのが OpenGL の文化である，と言う人もいますが困りますね．モデルとシーンを扱える COLLADA がなんとなく良さそうな気がしていますが試していません．
http://codezine.jp/article/detail/843?p=1

物理シミュレーションをどうするか

同様の話題として衝突判定等の物理シミュレーションがあります．最近では PhysX や Bullet を使ったものが目につくようになりました．
http://ja.wikipedia.org/wiki/PhysX

ゲーム環境をどうつくるか

ゲームエンジンを使う手もあります．例えば irrlicht はチュートリアルが充実していて，インストールして 5 分で DOOM ライクなゲームがつくれるようになるそうです．
http://irrlicht.sourceforge.net/

我々はこういういかにも VR は敬遠しがちですが，デモで必要な時にはどんどん使うべき．