

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

2011/03/07 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.

2011/03/07

- ・ OpenCV2.2 へのバージョンアップへの対応. ビデオ関係にバグ?

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。

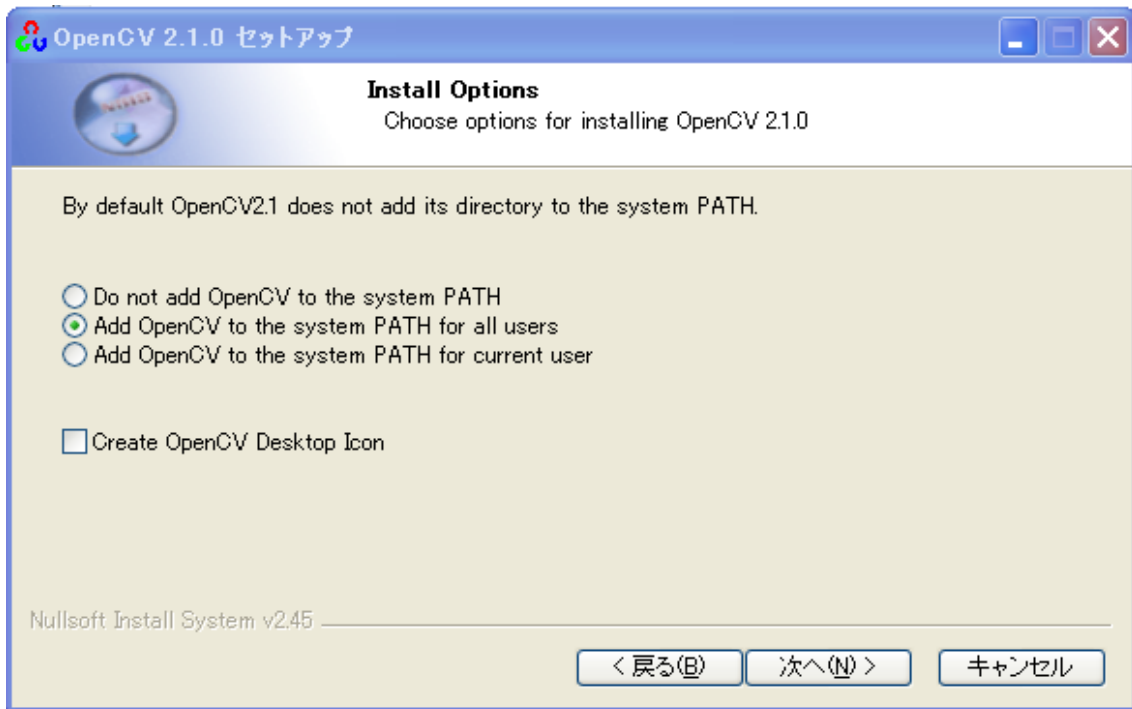
インストール

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-2.2-win ではなく、OpenCV-2.2.0-win32-vs2010.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/2.2/OpenCV-2.2.0-win32-vs2010.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）

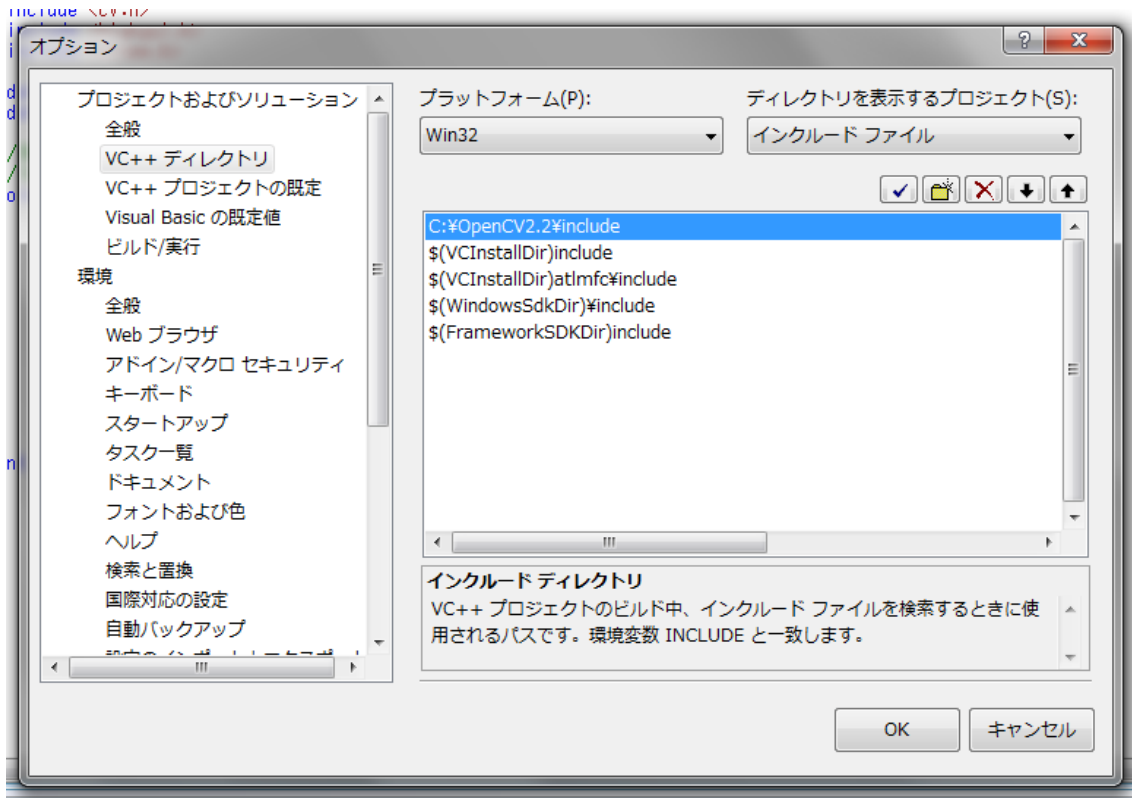
1. include ファイルの存在するフォルダを設定する。

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:¥OpenCV2.2¥include

を追加。

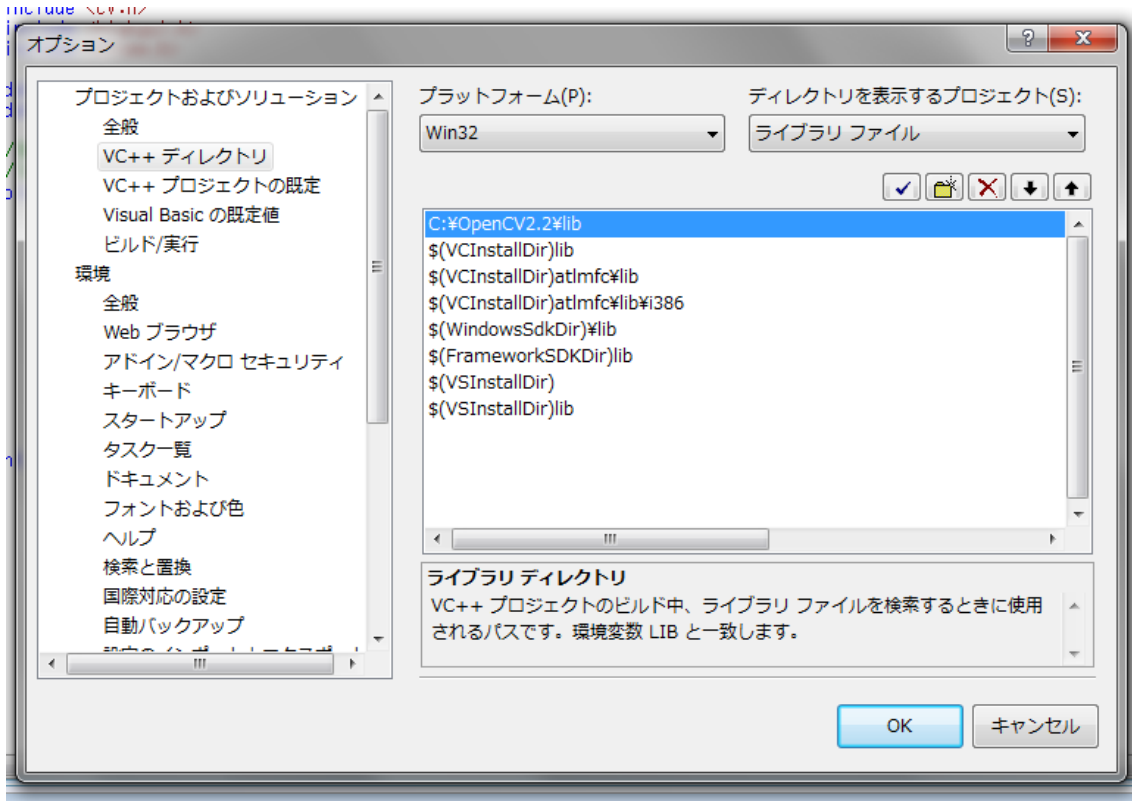


2.lib ファイルの存在するフォルダを設定する

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:¥OpenCV2.2¥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(Visual Studio のバージョンによって異なるので適当に探してください)

追加の依存ファイルとして下記のを記入

winmm.lib opencv_calib3d220.lib opencv_contrib220.lib opencv_core220.lib
 opencv_features2d220.lib opencv_features2d220.lib opencv_ffmpeg220.lib
 opencv_flann220.lib opencv_gpu220.lib opencv_highgui220.lib opencv_imgproc220.lib
 opencv_legacy220.lib opencv_ml220.lib opencv_objdetect220.lib opencv_ts220.lib
 opencv_video220.lib glut32.lib

(glut のライブラリとして freeglut を使う場合、最後が freeglut.dll になります。全てのプロジェクトで修正してください)

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

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。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

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出来ます。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、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めます。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 ファイルを読み出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, 動画を書き出すこともできます. 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.

※現在 Windows7-64bit 環境でこのプログラムが動かないというバグがあります.

課題:

この辺は今で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 推奨環境に移行することも考えられますが, オープン性は失われます.

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` を使ってみます.

インストールと情報

- OpenAL

<http://connect.creativelabs.com/openal/Downloads/Forms/AllItems.aspx>

(coreSDK 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

その他の話題

Wii と KINECT

ゲーム機の発達に伴い、高度な入力用インタフェースが安価に手に入るようになりました。研究室の今後の講習会でも、任天堂 Wii のコントローラによる位置計測、重心計測を取り上げます。

また Microsoft KINECT も扱う予定です。おそらく今後の OpenCV は KINECT のサポートが入りますので、今回の OpenCV 講習の内容はそのまま利用出来るはずです。

物理エンジン等

OpenGL, OpenCV を扱っても、まだ完成度の高い（≒見た目の良い）アプリケーションを作るにはいろいろな壁があります。研究室では物理エンジンとして openFrameworks を講習に取り入れる予定です。

一連の課題で身につけるべきこと

授業での初歩的なプログラミング以外では初めてのプログラミング、という中で、今回の目的は「ある程度の規模のプログラムを書いてみる」ことでした。その中で次の3つの能力を身につけて貰えればと考えています。

1. デバッグ能力：少なくとも初期においては、プログラミング能力≒デバッグ能力です。バグに対して、回避するのではなく、必ず原因を究明する。このバグ解決回数と初期のプログラミング能力は完全に比例しますので、「苦勞ミソ」を舐めるつもりでバグに立ち向かいましょう。

その際には莫大な可能性から素早く問題の場所を特定する「二分木」の考え方を使う必要があるでしょう。これはハードウェア開発でも有効ですので意識して身につけてください。

2. 検索能力：ソースコードがおかしい訳ではないのにエラーが出る... そうした問題に対してスタックしてしまうと何も出来ません。これはデバッグ能力とはまた別の能力、名検索能力が必要。つまり開発環境のヘルプと google 検索に正しく頼るということです。

3. 適応力：1, 2とも関係しますが、今後の研究あるいは就職その後、OpenGL, OpenCV ばかり使うことはありえませんし、C 言語で済む可能性も極めて低いでしょう。マイコン等のハードウェア講習でも同じですが、「それが使える能力」は大したことがなく、「何をを使うと最適かを調査し、その最適解を受け入れて新しい開発環境を採用するという適応力」がはるかに重要です。かと言って、研究室講習会ではこのレベルを直接教えることが出来ないで、自主課題を頑張ることによって身につけてください。

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 glutIdleFunc 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す、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変更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。