OpenCV祭り発表資料
iOS&便利マクロ&Tips

@dandelion1124

OpenCV

自己紹介（というかこれまでやったこと）

- **OpenCVプログラミングブック**

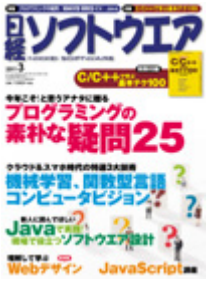
NAIST在学時に執筆．著者（の一人）．

- **「詳解OpenCV」**

和訳本出版時の原著コンテンツチェック等．

- **日経ソフトウェア特集**

2011年3月号記事執筆．



最近は主に動向を追ったり，bug報告したり，コード読んだり．

Agenda

- **iOS SDKでOpenCVを使おう**

対象：**iOS**アプリ開発者

- **知ってると便利なOpenCVマクロ**

対象：**OpenCV**開発者全般

- **Tips**

対象：**OpenCV**開発者全般

OpenCV

iOS SDKでOpenCVを使おう

今回のお話は下記バージョンを対象にします．

- **iOS SDK 4.2**
- **OpenCV 2.2**

また，インストールから説明すると発表時間が
足りなくなるので割愛．詳細は下記サイトを参照．

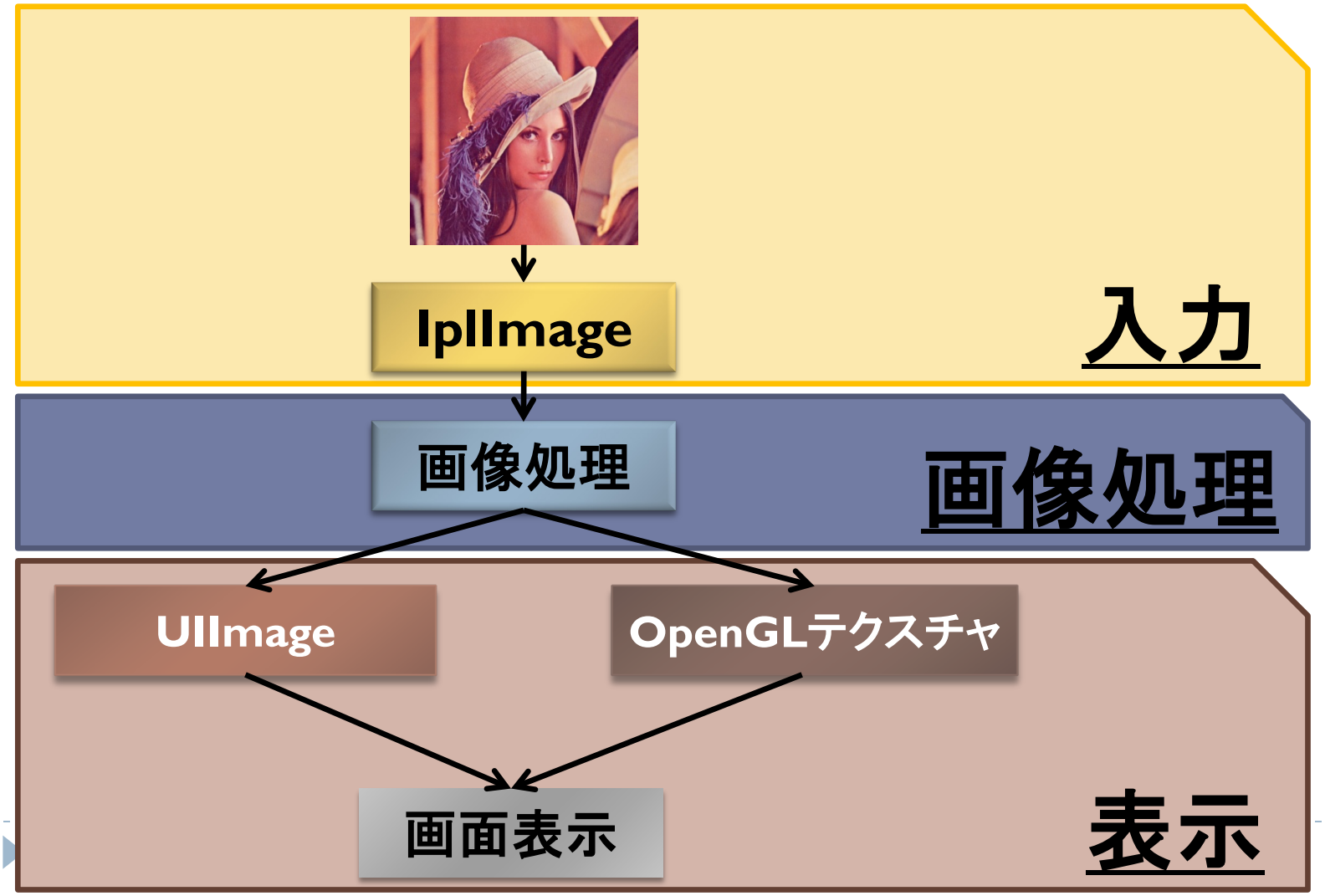
http://www.atinfinity.info/wiki/index.php?OpenCV

というわけで，使い方の話に進みます！！

OpenCV

iOS SDKでOpenCVを使おう

- iOSアプリ実装におけるデータの流れ（例）



OpenCV

iOS SDKでOpenCVを使おう

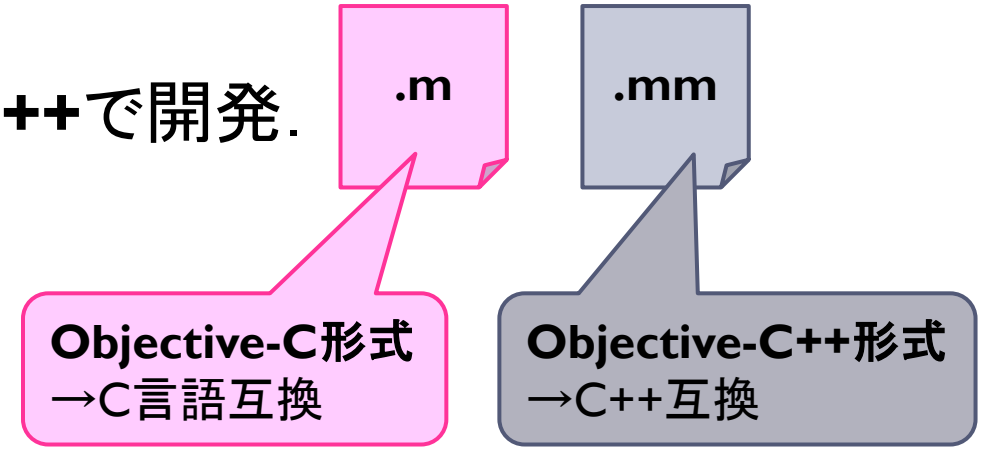
- **OpenCV**

C言語APIとC++ APIが提供.

- **iOS アプリ開発**

Objective-C，**Objective-C++**で開発.

拡張子は.mと.mm



デフォルトは**Objective-C**なので，このコード中では

OpenCVのC++ APIが使えません。。。

OpenCV

iOS SDKでOpenCVを使おう

- **選択肢1：**

OpenCVのC言語APIのみを使う

個人的にはこちらがオススメ
※「選択肢2」はちょっと上級者向け

メリット：（選択肢2に比べて）パフォーマンスが落ちない＋導入がシンプル

デメリット：C++ APIが使えない~~ので**C++er**には苦行~~

- **選択肢2：**

（C++として処理させるために）ソースコードを.mmにして，
xxxx.pchファイル中でOpenCVのヘッダをインポートする

メリット：C++ APIが使える~~ので**C++er**歓喜！！~~

デメリット：Objective-C++として処理されるのでアプリ全体のパフォーマンスが劣化

OpenCV

iOS SDKでOpenCVを使おう

- 気になるパフォーマンスのお話

計測条件

- ✓ **iPod touch 4th gen**
- ✓ **iOS SDK 4.2**
- ✓ **OpenCV 2.2**(Compile for Thumb無効でビルド)
- ✓ 使用画像：lenna.jpg(512x512)
- ✓ 計測は5回実施し，平均値をとる

よく使いそうな関数について計測結果を紹介．

OpenCV

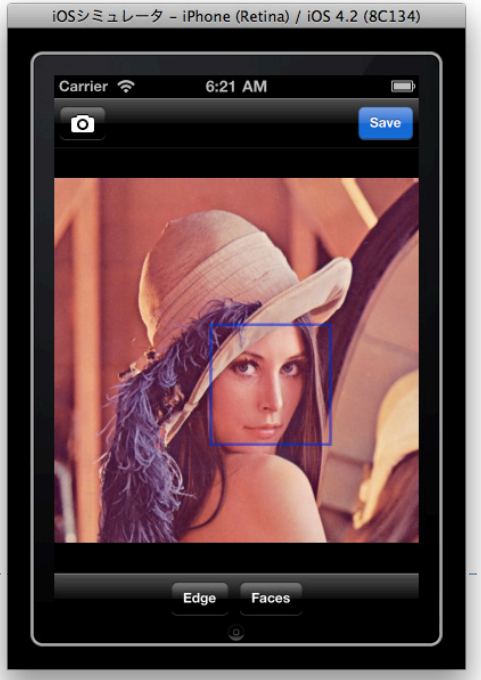
iOS SDKでOpenCVを使おう

- **顔認識**
  - 関数1：**cvLoad** **0.58[sec]**

学習データXMLの読み込みに案外時間が掛かるので，
極力，初期化で1回だけ読み込むようにすることを推奨．

  - 関数2：**cvHaarDetectObjects 0.86[sec]**

リアルタイムで処理するにはチューニングや
工夫が必要そう．



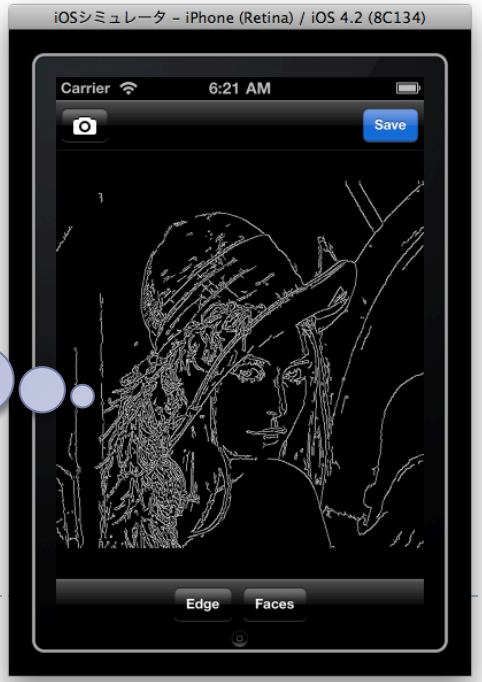
OpenCV

iOS SDKでOpenCVを使おう

- エッジ検出
  - 関数3：cvCvtColor　　0.01[sec]
  - 関数4：cvCanny　　0.05[sec]

この2つの関数についてはそこまで気にしなくてよさそう

実機によるデモは
懇親会の時にでも



OpenCV

iOS SDKでOpenCVを使おう

- パフォーマンス向上に向けて

NEON(ARM7のSIMD命令)を使って，OpenCVを**魔改造**すればパフォーマンス向上できるかも.

参考：NEONによるBGRA->GRAY変換処理概要（※詳細は[1]を参照）

```
for(8ピクセル毎に処理)
{
  //Y = 0.2126 R + 0.7152 G + 0.0722B
  tmp = valB * fracB;
  tmp + = valG * fracG;
  tmp + = valR * fracR;
  result = tmpを8bitシフト（RGBAのイメージなので）
  参照イメージのポインタをずらす
}
```

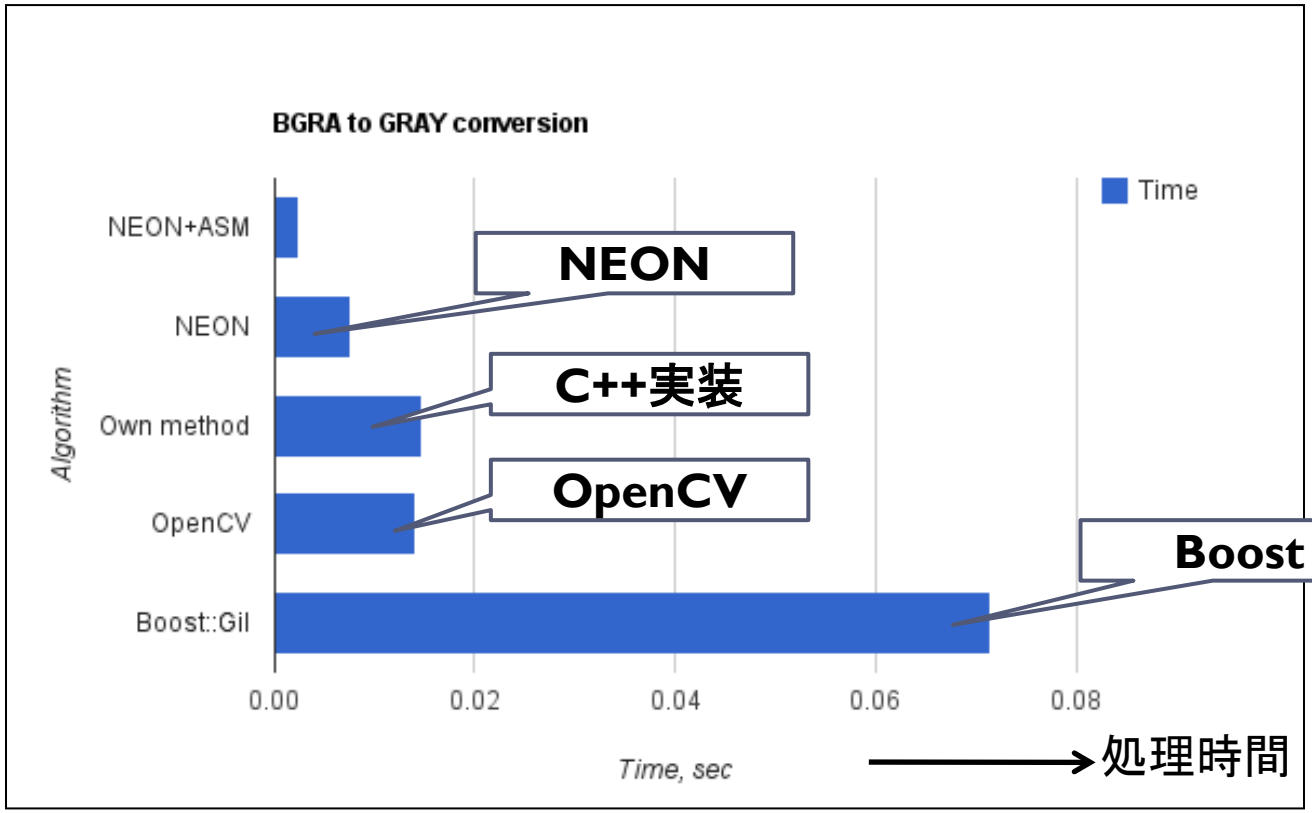
[1] http://computer-vision-talks.com/2011/02/a-very-fast-bgra-to-grayscale-conversion-on-iphone/

OpenCV

iOS SDKでOpenCVを使おう

- パフォーマンス向上に向けて

参考：NEONを使ってBGRAからGRAY変換した処理時間（※[1]より引用）



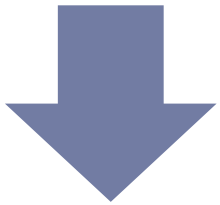
[1] http://computer-vision-talks.com/2011/02/a-very-fast-bgra-to-grayscale-conversion-on-iphone/

OpenCV

知ってると便利なOpenCVマクロ

- **OpenCVマクロを調べる経緯**

  コードリーディングを効率良く行うために調査開始



よくよく考えたら，コーディングにも使えるんじゃね？

というわけで，軽い気持ちで調べてみることに

※後日恐ろしく後悔するわけですが。。。

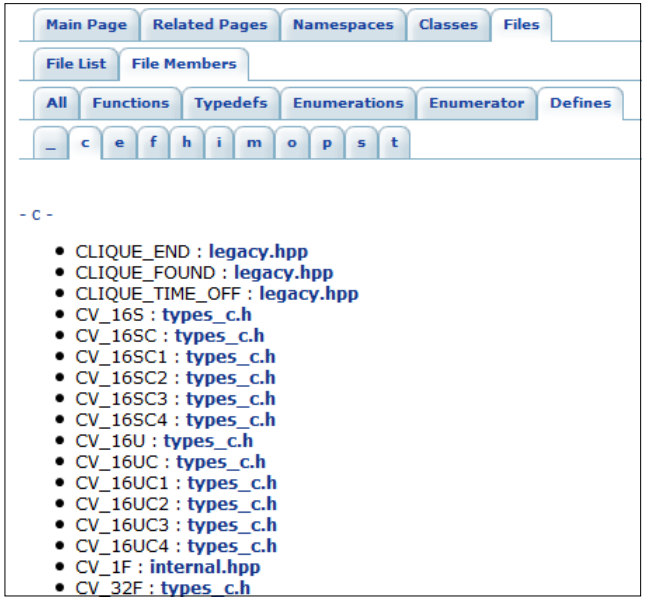
OpenCV

知ってると便利なOpenCVマクロ

- 地味に涙ぐましい道のり。。。

CMakeで「**BUILD_DOXYGEN_DOCS**」を**ON**にして**HTML**生成

HTMLをひたすら読む！（こんなの→）



マクロが多すぎて（ざっと250個強），途中で何度も心が折れそうに。。。orz

OpenCV

そんな前置きはどうでも
よいのでマクロの解説

OpenCV

知ってると便利なOpenCVマクロ

- **CV_IMAGE_ELEM(image, elemtype, row, col)**

機能：指定座標の画素値を参照

使い方：

✓ **座標(100, 100)の画素値を取得**

```
unsigned char p = 0;
p = CV_IMAGE_ELEM(img, unsigned char, 100, 100);
```

✓ **座標(100, 100)の画素値に100をセット**

```
CV_IMAGE_ELEM(img, unsigned char, 100, 100) = 100;
```

※このマクロはIplImageしか受け付けない点，
エラー処理が無い点に注意が必要．

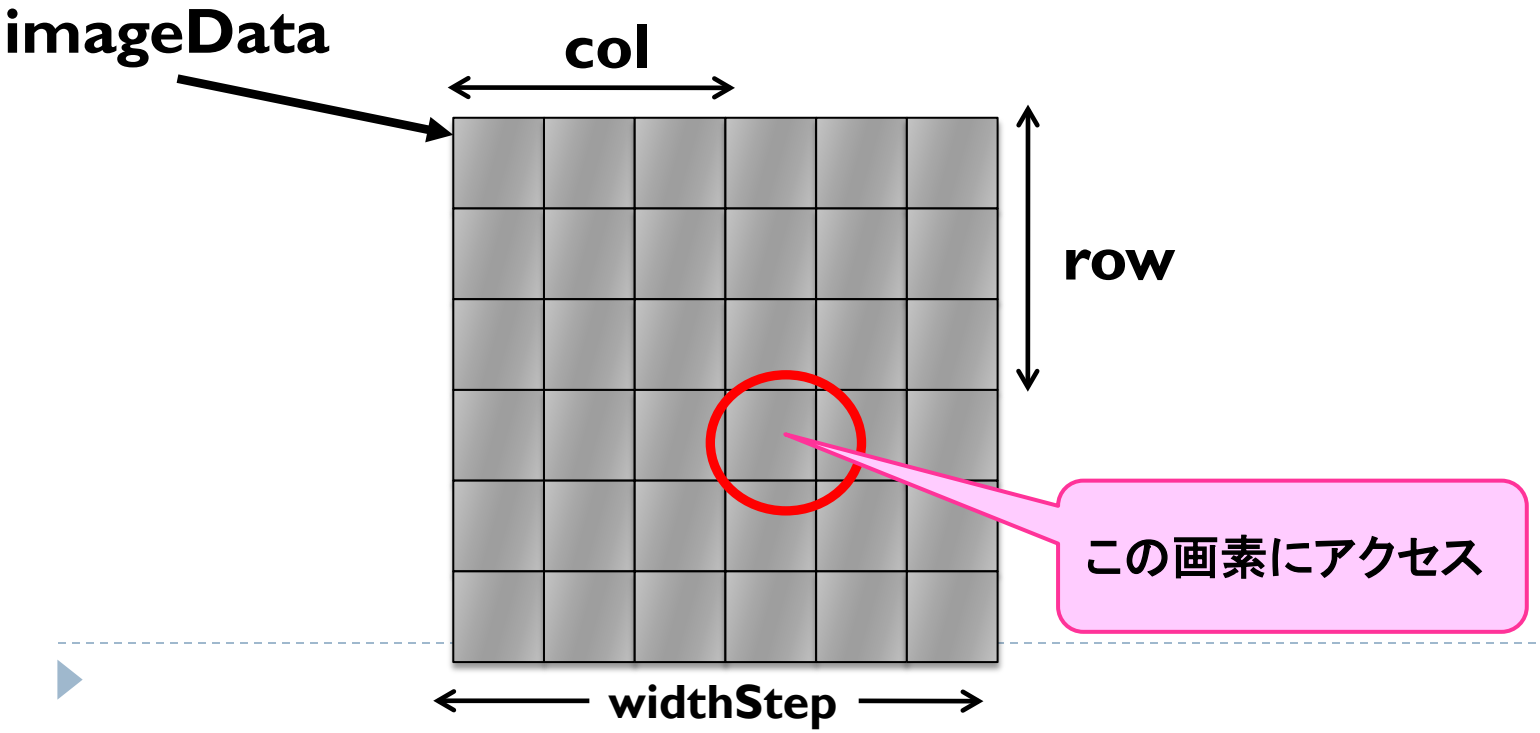
OpenCV

知ってると便利なOpenCVマクロ

- **CV_IMAGE_ELEM(image, elemtype, row, col)**

マクロの中身はこんな感じ（※**IplImage**のポインタ参照）.

(((elemtype*)((image)->imageData + (image)->widthStep*(row)))[(col)])



OpenCV

知ってると便利なOpenCVマクロ

他の型の場合は？

- CvMatの場合

```
CV_MAT_ELEM(image,  elemtype, row, col)
```

- Matの場合

```
Scalar intensity = img.at<uchar>(x, y);
```

OpenCV

知ってると便利なOpenCVマクロ

- **CV_VERSION**

機能：OpenCVのバージョン番号を取得

使い方：

```
printf("OpenCV version = %s¥n", CV_VERSION);
```

出力結果：

OpenCV version = 2.2.0

これだけだとそんなに
うれしくないですが。。。

知ってると便利なOpenCVマクロ

- **CV_VERSIONの発展系**

以下のようなオリジナルマクロを作っておくと実用的かも.

```
#define CV_VERSION_NUMBER
  CVAUX_STR(CV_MAJOR_VERSION)
  CVAUX_STR(CV_MINOR_VERSION)
  CVAUX_STR(CV_SUBMINOR_VERSION)
```

バージョン毎のライブラリを読み込むコードに活用可能.

```
#pragma comment(lib, "opencv_core" CV_VERSION_NUMBER ".lib")
#pragma comment(lib, "opencv_highgui" CV_VERSION_NUMBER ".lib")
#pragma comment(lib, "opencv_imgproc" CV_VERSION_NUMBER ".lib")
```

OpenCV

知ってると便利なOpenCVマクロ

- **CV_IS_MAT（img）**

機能：CvMat形式であるかを判別

- **CV_IS_IMAGE（img）**

機能：IplImage形式であるかを判別

使い方：

```
if(CV_IS_MAT(obj)){proc_for_mat();}
else if(CV_IS_IMAGE(obj)){proc_for_iplimage();}
```

CvMat，IplImageを判別して処理を分けたいときに

OpenCV

Tips

- リンクすべきライブラリを探すTips

OpenCV 2.2になって機能毎にモジュール分割が行われました.

- calib3d
- contrib
- core
- features2d
- ffmpeg　　etc…

あの関数を使うときにはどのライブラリを
リンクすればいいんだ！ということはありませんか？

OpenCV

Tips

- リンクすべきライブラリを探すTips

OpenCV-2.2.0¥doc¥OpenCV.pdfが参考になります！

②章タイトルを確認！
この場合はhighgui.

①cv::imreadを使う場合は
どのモジュールかな？

OpenCV

御静聴ありがとうございました