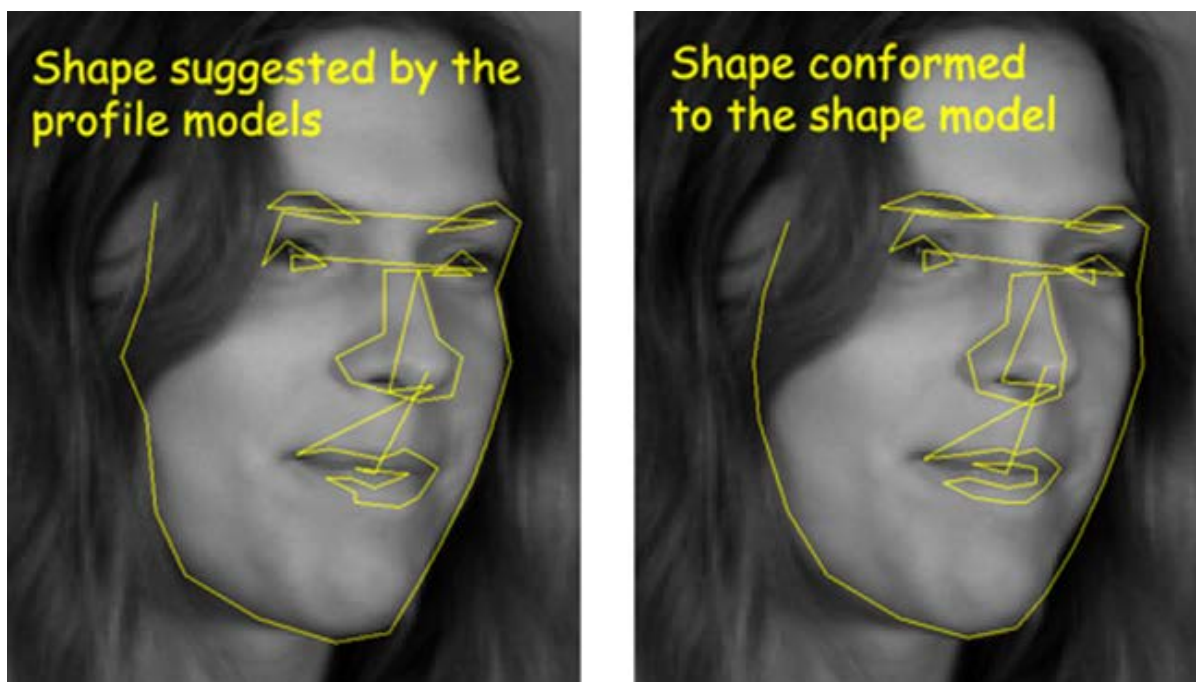


ビジュアル・コンピューティング 2017: OpenCV* について

この記事は、インテル® デベロッパー・ゾーンに公開されている「[Visual Computing 2017: A Look At OpenCV](#)」の日本語参考訳です。



出典: インテル

広範な言語とプラットフォームにおいて画像操作を単純化

OpenCV* (Open Computer Vision の略) バージョン 1.0 のリリースから 10 年が経ちました。リアルタイム・コンピューター・ビジョンを利用してソフトウェアを作成できる C++ ライブラリーとして、元々インテル向けにロシアで開発された OpenCV* は、完全にオープンソースで、マシンラーニングも含まれます。OpenCV* は BSD ライセンスの下に提供されるため、無料で使用できます。

さまざまな DLL で構成される非常に大きなライブラリーで 200MB を超えます。必要に応じて、スタティックにリンクすることもできます。C/C++, Python*, Java*, MATLAB* インターフェイスを備えた数百のアルゴリズムがあり、Windows*, Linux*, Android*, iOS*, および macOS* で動作します。バージョン 1.0 は C ベースでしたが、バージョン 2.0 以降は C++ ベースになりました。最新のバージョンは [3.2](#) (英語) で、2016 年 12 月にリリースされました。

OpenCV* に含まれるものは?

OpenCV* には、(ストリームまたはディスクからロードされた) 画像やビデオをキャプチャーし、メモリーに格納して、操作するために必要なすべてのものがまとめられています。

操作には、個々のピクセルの変更、ヒストグラムの計算、顔認識、動作追跡、拡張現実、ジェスチャー認識などが含まれます。[ウィキペディア](#)に OpenCV* ライブラリーで可能な操作の詳しいリストがあります。

[OpenCV* のユーザー](#) (英語) には、主要企業 (IBM*、Microsoft*、インテル、SONY*、Siemens*、および Google*) および研究機関 (スタンフォード大学、マサチューセッツ工科大学、カーネギーメロン大学、ケンブリッジ大学、フランス国立情報学自動制御研究所など) が含まれます。

OpenCV* のインストール

Windows*/macOS* 向けにはバイナリーが提供されています。Linux* では、tar ファイルをダウンロード後展開して、make を実行してから、sudo make でインストールする必要があります。手順については、さまざまな資料で紹介されていますが、[OpenCV* チュートリアル](#) (英語) が特に役立つでしょう。インストールには、コーヒープレイクに十分な時間がかかるでしょう。私の PC (Ubuntu* 16.04 LTS) では、問題なくインストールできました。

また、[無料](#)で利用可能なインテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP) もインストールすべきです。インテル® IPP は OpenCV* と統合され、多くの操作をスピードアップします。次の関数を呼び出します。

```
cvGetModuleInfo( 0, &libraries, &modules );
```

この関数は、インストールされているライブラリーとモジュールの説明を返します。OpenCV* アプリケーションを記述する場合、この情報を使用してターゲット PC にインテル® IPP がインストールされていることを確認し、インストールされていない場合はダウンロードしてインストールするように促します。

OpenCV* での作業

インテル® IPP は非常に大きなライブラリーです。使用する前に、いくつかの基本を理解する必要があります。メインデータ構造の Mat クラス (OpenCV* 1.0 では cvMAT 型) は、メモリーにある画像を管理し、ヘッダーと画像データの両方を含みます。データバッファーを共有していない限り、メモリーの割り当て/解除を自動で行います。これは強力なクラスで、vector<>、Mat<>、またはスカラー値のような異なるデータ型を使用できます。

インテル® IPP には多数のメソッドがあり、[HighGui モジュール](#) (英語) は画像の表示・保存用のウィンドウを作成し使用できるようにします (OS の描画コールバックは不要です)。簡単なマウスイベントを処理するトラックバーを追加して、画像をディスクやメモリーに読み書きし、ビデオをカメラから読み取ってファイルへ書き込みます。

GPU の使用

OpenCL* は、NVidia* の GPU テクノロジーである [CUDA*](#) (英語) とうまく統合できます。別の GPU テクノロジーである OpenCL* 用の OCL モジュールもあるようですが、[StackOverflow に関する答え](#) (英語) が示すように、OpenCV* 3 ではこのモジュールは不要です。

マシンラーニング

OpenCV* は、ベイズ分類器、K 近傍、SVM、決定木、ブースティング、ランダムフォレスト、ニューラル・ネットワーク、ロジスティック回帰を含む、独自のマシンラーニング・ライブラリーを備えています。ディープ・ニューラル・ネットワーク用のモジュール dnn (追加モジュールとしてリストされています) もあります。ただし、フォワードパスの計算 (つまり、ネットワーク・テスト) のみ対応しています。ネットワークの訓練はサポートされません。

まとめ

コンピューター・ビジョン・アプリケーションの開発に興味がある方は、最初に OpenCV* を試してみるべきです。OpenCV* は C++ でプログラムされていますが、大部分はテンプレートを使用せず、代わりに多相性とランタイム・ディスパッチを使用しています。ピクセルアクセスなどの一部の操作は、小さなテンプレート・クラスと関数を使用しています。

OpenCV* アプリケーションの開発は主に C++ で行われていますが、Python* の人気の高まりを受けて、OpenCV* 3.2 のドキュメントには [Python* 向けのチュートリアル](#) (英語) もいくつか追加されています。[インテル® Distribution for Python*](#) の NumPy* (OpenCV* に必要) による高速化を考慮すると、Python* の使用は理想的に見えます。