

OpenCVで映えるリアルタイム動画処理に挑戦

安川 章

今回挑戦すること… OpenCVリアルタイム動画処理

YouTube時代です。ここではOpenCVを用いて、プリクラにあるような美肌処理や目を大きくする処理をリアルタイムに動画像に施します。図1に効果を示します。

動画像における画像処理は、基本的には静止画の画像処理と変わりません。動画像の中から1枚1枚のフレーム画像を取得して処理します。

● 開発環境

開発環境は以下です。

Windows 10 64ビット版

OpenCV 4.2.0

Visual Studio 2017 C++

● 美顔処理の流れ

美顔処理のフローを以下に示します。

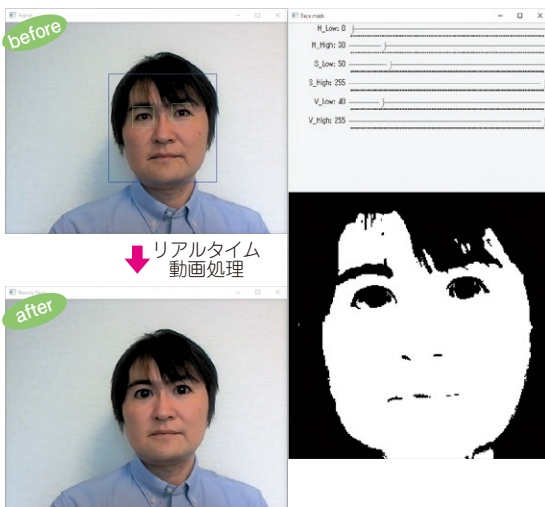


図1 YouTube時代に生かせそうなリアルタイム My 動画画像処理に挑戦

OpenCVによる美顔処理の効果

ステップ1：動画からフレーム画像を取得

ステップ2：顔検出&目検出

ステップ3：目を大きくする

ステップ4：美肌処理

ステップ5：動画ファイルへ書き出す

● プログラムの入手先

プログラムは本誌ウェブ・ページから入手できます。

<https://www.cqpub.co.jp/interface/download/contents.htm>

ステップ1： 動画からフレーム画像を取得

リスト1に動画像からフレーム画像を取得するプログラムを示します。OpenCVでは動画を扱うためのVideoCaptureクラスが用意されています。

● カメラ画像の取り込み

USBカメラを使用するにはopen()関数の引数に整数を渡すとUSBカメラが使用できます。

```
cap.open(0);
```

通常は0を渡すとデフォルトのカメラを使用しますが、PCに複数台のカメラが接続されている場合は、1や2などの数値を指定してください。

動画ファイルを使用するには、open()関数の引数に動画ファイルのパスを指定します。

```
cap.open("sample.avi");
```

USBカメラが接続されていない場合や、動画ファイルが存在しない場合を確認するため、isOpen()関数を実行し、戻り値がtrueの場合は、カメラの接続、動画ファイルのオープンに成功しています。

● フレーム画像の取得

動画像の中からフレーム画像を取得するには、read()関数にて、フレーム画像を格納するMatクラス・オブジェクト(ここではframe)を渡し、フレーム画像を取得します。

read()関数を連続的に繰り返し実行することで、