第4章 USBカメラから得たリアルタイム画像を処理＆表示する方法

吉田 大海

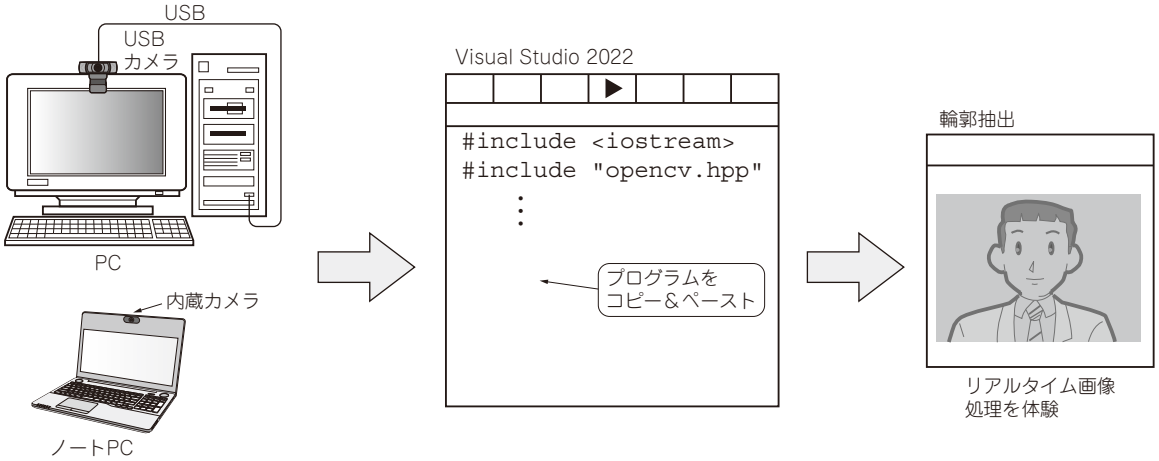

図1　PCとUSBカメラを接続後にプログラムを実行するだけで試せる
ノートPCにある内蔵カメラでも代用できる

前章では，あるフォルダに格納されている画像に対して画像処理を施すことが前提でした．本章で対象とするのは，USBカメラから取り込むライブ画像（リアルタイム）です．プログラムの説明は，前章で解説した内容を基にしていますので，前章を未読の場合はそちらからご覧ください．

● 環境構築

既に画像処理環境を整えたPCにUSBカメラを接続し，USBカメラ用のプログラムを実行すればすぐに試すことができます（**図1**）．なお，ノートPCなどの内蔵カメラでも代用できる可能性があります．

プログラムと実行結果

● 使用するプログラム

今回使用するプログラムUSBカメラ22.cppを**リスト1**に示します．このプログラムはUSBカメラを起動し，リアルタイムで画像を取得しつつ表示して，その画像をグレー化した画像も表示します．ただし，[q]キーを押すと終了します．

もし，自分自身で作成したUSBカメラ用の画像処理を試したいときは，本プログラムに上書きしていく形から始めると手軽です．

● 実行結果

リスト1のプログラムを実行すると**図2**のような結果が得られます．この例ではカメラ画像をグレー化するだけですが，エッジ検出などを実行すると，未来感のある映像になって面白いと思います．ぜひいろいろな画像処理結果をリアルタイム映像として見てみてください．

プログラムの処理内容

● USBカメラの認識確認

まずは接続したUSBカメラがきちんと画像処理環境で認識できるのかを確認する必要があります．そのための記述がメイン関数内の最初にあります．

▶9行目…USBカメラの起動

ここの意味は，USB接続されたカメラを起動し，USBcameraとして管理するとなります．ここで，USBcamera(0)の最後に付いている()内の0は，接続されているカメラに割り当てられる引数（カメラ・ナンバのようなもの）です．

ノートPCを使っている場合，内蔵カメラを認識するかもしれません．その場合，USB接続しているカメラには1番が割り振られることがあります．そのような場合は，()の数字を1に変更するとUSBカメラの映像に変更されます．この数字は使用環境にもよりますので，実際にカメラ起動を試しつつ引数を変更します．

▶11～15行目…エラー確認用の処理

もし，どのカメラもうまく起動できていない場合に