第4章

基本中の基本フィルタで試す!
撮ったらすぐ処理でレベルアップ!

ダウンロード・データあります

実験ビフォー・アフタ! リアルタイム画像処理の基本テクニック

本格的な説明に入る前に

森岡 澄夫

図1　平均値フィルタ適用の効果

Raspberry Pi開発環境のセットアップを行って顔検出などのサンプルを動かしたとき，何もくふうしなければ，その速度の遅さに驚くことでしょう．

Raspberry Piは組み込み系のマイコンとして見れば極めて高速なのですが，一方で画像処理はもろもろのコンピュータ処理の中でも最も重い部類の一つであり，かなり荷が重いと言えます．第3部では高速化に取り組むうえで基本となる考え方を説明します．

直面する課題…単純な処理でも意外に時間がかかる

OpenCV関数を使って処理を組む過程で，「実は思いがけずむだなこと，不必要に複雑なことをしてしまっているのではないか，コンピュータ内のデータの流れに逆らったことをしているのではないか」という疑いを持ってかかると，高速化の糸口が見えてきます．

手始めに，よく使われる平均値フィルタをリアルタイムで動かすことを考えてみましょう．これは画像をぼかす処理です．画像をぼかす目的としては，細かいノイズを除去したり，画像のおおまかな特徴だけを残したりすることが挙げられます．画像処理としては単純なもので，これ単体が処理の目的というよりは，むしろプリミティブ演算（画像処理を組み立てるために使う部品）として使うものです．

OpenCVでは，`cv::blur()`という関数が用意されていますから，ブラック・ボックス扱いで呼び出すだけでも済みます．単純にもかかわらず，意外に時間のかかる処理であり，高速化は必要になってきます．

平均値フィルタを例に…実行時間を測ってみる

● 計算の内容

フィルタ適用の効果の例を**図1**に，具体的な演算内容を**図2**に示します．このフィルタの現理は，ある画素に着目してその周辺の領域の輝度値の平均をとっていくものです．画面内の輝度の変化を減らす処理となり，その結果，画像がぼけて見えるようになります．画像をぼかす度合いを設定するフィルタ・パラメータとしてカーネル・サイズ（ここでは平均をとる範囲）があり，これを大きくすればするほど**図1**のように効果が強くなります．

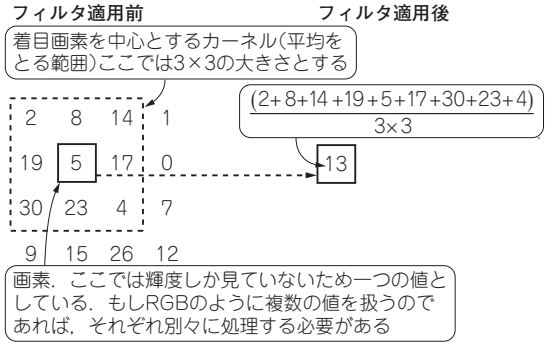

図2　平均値フィルタの演算内容