



(a) 原画

(b) エッジ処理後に線画

(c) (a), (b)を合成(一部拡大)

図1 前処理にエッジ処理をすれば画像処理の幅も広がる

● 取り出したり分割したりの前処理として

エッジを抽出すると、画像内の境目が明確になり、さまざまな処理の基とすることができます。例えば、図1(b)は図1(a)の原画にエッジ処理をして線画と呼ばれる絵を生成したものです。以前は図1(a)から人が画像編集ソフトウェア使って、手作業で線を引いていました。Pythonを使えばこの作業がプログラムで一発処理できます。

図1(c)は、2つの画像を合成して生成したパステル画風の画像です。このようにプログラム1つで、どんな画像もパステル画風に仕上げることができるようになります。

その他にも白い紙から文字だけを取り出したり、前景と背景を分けたりとさまざまな用途に使うための前処理に使えます。

● エッジ強調の種類

OpenCVでは、関数によっては関数内部でエッジ処理をするものもあります。直接はエッジ抽出関数を使

わないこともありますが、ここでは、幾つか用意されている関数の使い方の解説をしていきます。

それぞれ実行結果には違った特徴があり、処理速度も差があります。各テスト・プログラムを実行して、目的に応じて使い分けてください。

● 画像深度について

解説中に画像深度としてCV_8U, CV_8UC1などの表記が出てきます。どちらも1チャンネル・データで、CV_ビット数, U=符号なし整数, S=符号付き整数, F=浮動小数点を表しています。

例えばCV_8UはCV_8UC1とも表記可能で、8ビット1チャンネルのデータで通常のグレー・スケール画像を表現し、CV_8UC3と表記してあれば8ビット×3チャンネルでフルカラー画像を表現していることとなります。

この項目ではCV_8U, CV_16U, CV_16S, CV_32F, CV_64Fなどが出てきますが、これらは全てグレー・スケール画像となります。

6-1 きれいなエッジを抽出する キャニー

プログラム名: Canny.py (GPU版はCanny_gpu.py)

CPU版 <https://interface.cqpub.co.jp/6-1canny-py/>, GPU版/6-1canny_gpu-py/



CPU版



GPU版

● シンプルで検出漏れが少ない

複数のステップを経てエッジを検出する手法です。まず、ノイズ除去のための平滑化フィルタを実行し、次に画素値の勾配と勾配の方向を求めます。ここまでの計算結果の画素値の勾配方向の最大値を確認して、そのほかの画素を取り除きます。この時点で線画が出来上がっているの、最後にしきい値に応じてエッジを強調すれば出来上がりです。

● CPU版…プログラム名: Canny.py

▶リスト1: 008行…スレッショルド設定

```
cv2.Canny(image=img, threshold1=100,
threshold2=200)
```

第1引数 image は入力画像です。

第2引数 threshold1 は設定値が大きくなるほどエッジ表示が減ります。

第3引数 threshold2 は設定値が大きくなるほど、エッジのつながりが少なくなります。

第2, 第3引数の設定でどの程度のエッジを強調するかが決まります。値は対象画像によって変わります。取りあえずは、threshold1, threshold2を同じ値から始めて、適度な結果になったらthreshold2で調整することをお勧めします。

● 画像処理の結果をしてみる

図1(a)の原画に対して、threshold1=100は