

画像処理の基礎

Fundamentals of Image Processing

中央大学 梅田和昇

1. は じ め に

画像処理の歴史はかなり古く、1960年代には研究開発が始まっている。以前の画像処理応用では、特定の目的のための専用機が開発され利用された。現在でも専用機は重要であるが、一方で、デジタルスチルカメラ（いわゆるデジカメ）や携帯電話の爆発的普及、ならびにパーソナルコンピュータ（PC）の高速化により、画像処理が身近にそして安価になり、とても使いやすくなってきている。

本学会には、画像応用技術専門委員会¹⁾を中心に、画像処理の最先端を研究・開発対象としている会員もかなりいらっしゃるが、多くの会員は、画像処理は直接の対象ではなく、うまく利用できれば十分という立場だと思う。本稿では、そういう方を対象として、基礎的かつ重要なポイントを紹介することを試みる。

2. ハードウェア・ソフトウェア

まず、画像処理に必要なハードウェア・ソフトウェアを紹介する。どういう目的でどういう対象に対して画像処理を行うかによって様々であるが、典型的な場合を説明する。処理がオンライン（画像を入力しながら実時間処理）かオフライン（画像入力後に処理）かによって色々違ってくる。

2.1 カメラ

画像処理を行うためには画像が必要であり、画像を得るためにはカメラが必要である。オンラインの場合、画像取得には、一般に CCD カメラが用いられる。最近では、CMOS カメラも用いられるようになってきた。CCD、CMOS イメージセンサとも、固体撮像素子と呼ばれるもので、フォトダイオードを2次元（スキャナなどに用いられるライン CCD など1次元のものもある）に配列させたものである。光電効果で発生した電荷を、バケツリレー方式で転送しながら読み出すのが CCD、アドレス指定で読み出すのが CMOS イメージセンサであり、一般に画質の点で前者、それ以外の点で後者が優れている。産業用の CCD カメラにはモノクロのものが多く、カラー CCD カメラには3CCD、1CCDの2種類ある。前者は、3枚の CCD を利用し、各 CCD に R (Red), G (Green), B (Blue) の色フィルタを装着することで R, G, B の画像を取得する。後者は、1枚の CCD の各フォトダイオードに R, G, B の超小型（数 μm 角）の色フィルタを装着し、1枚の CCD だけで R, G, B の画像を取得する。R, G, B の配列にはベイヤー

配列と呼ばれるものが良く利用され、全画素の半分以上が G、1/4 ずつが R と B である。各色で全画素数分の画像を得るには補間処理が用いられる。画質は当然 3CCD の方が上である。また、1CCD で同じ画素数なら、モノクロの方が画質が良い。なお、そもそも色を R, G, B の3色の合成で表現するのは、人間の眼の色を感じる細胞（錐体）が3種類だからである。

これまでは、テレビ信号を出力するアナログインタフェースのカメラが多く用いられてきた。日本やアメリカで用いられているテレビ信号の規格は、NTSC（モノクロは EIA）と呼ばれ、約 30 Hz で走査線 525 本（うち有効なのは 480 本程度）の画像を出力する。ほかに、ヨーロッパなどでは PAL（モノクロは CCIR）や SECAM という規格が用いられており、25 Hz で走査線 625 本である。これらのテレビ信号では、1枚（フレーム）の画像を得るのに、まず奇数行だけの画像（奇数フィールド）、次に偶数行だけの画像（偶数フィールド）を送り、合成する。これをインタレースと呼ぶ。画像が移動物体を含む場合、奇数フィールドと偶数フィールドとでずれが生じてしまうので注意が必要である。これを防ぐため、テレビ信号の規格からはずれるノンインタレースのカメラも市販されている。テレビ信号は、コンポジット信号と呼ばれる輝度情報と色情報とを混合した信号で通常送られる。テレビやゲーム機などの黄色い端子を使うものである。S 端子は輝度情報と色情報とを分けたものである。R, G, B それぞれを別の端子にしたものもある。この順に画質が良くなる。

最近では、デジタルインタフェースのカメラが普及してきた。そもそも固体撮像素子を用いていれば空間的に標本化された画像が得られる訳で、アナログインタフェースの場合それをわざわざ D/A 変換している。D/A 変換してさらに A/D 変換するより、デジタルのまま直接信号を扱った方が当然画質は良くなる。インタフェースには、CameraLink と呼ばれる規格、IEEE1394, USB, Ethernet を用いる GigE などが用いられる。テレビ信号の規格に比べられないので、メガピクセル（100 万画素以上）のものを含め、色々な画素数のカメラが市販されている。ただし、画素数が多くなれば、それだけ転送には時間かかる。なお、DV ビデオカメラも IEEE1394 を用いるが、画像が圧縮されている。

CCD カメラで得られる画像は、CCD での計測値そのままではなく補正されていることが多く、特に、計測値をベ

き乗することが一般的である。この時のべき値をガンマ値、この補正をガンマ補正と呼ぶ。これは、TV モニタが持つべき乗の特性（ガンマ値 2.2）を打ち消すための処理で、ガンマ値 0.45 が標準である。産業用 CCD カメラではガンマ値を 1（補正なし）と 0.45 に切り替えられるものがあるが、1 に切り替えて使用すべきである。

オフラインで画像を取得するには、デジカメや、携帯電話のカメラ（これもデジカメであるが）を利用することも可能である。近年のデジカメの急速な発展のおかげで、一般の CCD カメラをはるかに越える画素数の高精細な画像を安価に取得可能である。ただし、デジカメの画像は写真を綺麗に見せるために色々画像処理が加えられているので、注意が必要である。ガンマ値なども画像ごとに異なる。また、良く知られているように、jpeg は不可逆な圧縮がかけられており、画質がかなり劣化している（人の眼にはわかりにくい）。高性能なデジカメだと、RAW 形式という原則として何も処理されていない（R, G, B 画像の補間すら行われていない）画像が撮像可能であり、これを利用するのが一番良い。

2.2 ハードウェア

最初に述べたように、画像処理専用のハードウェアも未だ健在である。また、それに類する形で、PC の PCI バスにさして用いる画像処理専用ボードもある。しばらく前には、日立製作所の IP-5000 シリーズ（現在の後継機種は IP-7000 シリーズ）がロボティクス分野などで広く使われ、我々もお世話になった。最近では、PC が用いられるケースが増えてきた。画像処理はデータ量が膨大なので処理時間がかかることが 1 つの問題であるが、PC が劇的に速くなったおかげで、専用のハードウェアでなく汎用の PC を用いてもかなりの処理をリアルタイムで実行可能になって来たわけである。なお、画像処理の分野では、テレビ信号の周波数、すなわち NTSC の場合 30 Hz で処理が行われることをリアルタイムと言うことが多い。

画像を PC に取り込むためのハードウェアも考えなければならない。アナログカメラの場合、PCI バスにさして用いる画像入力ボードが比較的安価に市販されている。このボードで A/D 変換を行い、通常 640 × 480 画素、R, G, B 各 8 ビットで計 24 ビットの画像を得る。デジタルの場合、CameraLink なら対応するボードが必要である。IEEE1394 や USB を用いるものなら、PC に標準装備のものを利用可能である。また、デジタルインタフェースのカメラでは 10 ないし 12 ビットの高階調の画像を得られるものもある。

一方、オフラインで画像処理を行う場合のハードウェアは、大抵 PC が用いられる。

2.3 ソフトウェア

専用のハードウェア（ボード含め）を用いる場合は、画像処理を行うためのソフトウェアも付属であることが多い。主要な画像処理がライブラリ化されており、開発しやすい。一方、PC を用いる場合には、いくつか選択肢が考えられる。

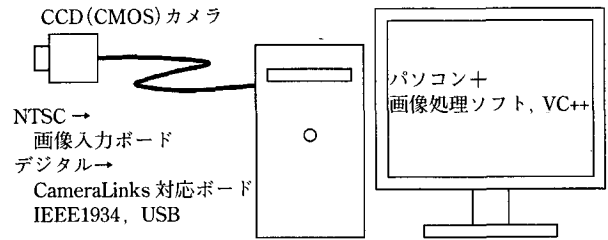


図1 画像処理システムの標準的な構成

1 つは、市販の画像処理専用ソフトウェアの利用である。ライブラリが充実しているものを選べば、高機能の画像処理をかなり容易に実現可能である。我々はドイツのMVTec という会社が市販している Halcon というソフトを使っている。本ソフトウェアのデモ版は、文献²⁾に付属しており、画像処理ソフトウェアでどうことができるかを知る意味でも試してみても良いと考える。ただし、この書籍は通常の書店では販売されていない。なお、市販のソフトウェアを利用する欠点は、高価なこと、それと多くの処理がブラックボックス化されていることである。

また、我々はこれまで使っていないが（これから使っていく予定）、Intel がフリーで公開している OpenCV というソフトウェアライブラリ³⁾も良く利用されている。こういうライブラリを用いない場合には、Visual C++ などの汎用の開発環境を用いることになる（OpenCV の利用時にも必要である）。この時、C のソースが掲載されている教科書^{4) 5)}は参考になるだろうし、Web で公開されているソースを見つけることもできる。

図1に画像処理システムの標準的なハードウェア・ソフトウェア構成を示す。なお、以上は Windows を前提として書いている。Linux（あるいは Mac）も利用可能であるがここでは触れない。

2.4 画像の取得

順番がやや前後するが、画像の取得方法について簡単に述べておく。

上記のハードウェア・ソフトウェアの組合せにおいて、専用のボードを用いる場合には、一般に SDK（Software Development Kit）やサンプルソフトが提供され、それらを用いることができる。また、カメラに SDK が付属している場合もある。安価な USB カメラなどで SDK が付属していない場合でも、DirectShow（Windows で動画を扱う API）に対応するソフトウェア（Halcon や OpenCV はそうである）を用いれば、画像のオンラインでの取得を容易に行える。そうでない場合には、VC++ で DirectShow を直接扱うプログラミングが必要である。

一方、オフラインで画像をディスクから読み込む場合には、画像フォーマットは、BMP 形式が扱いやすい。BMP 形式は、ヘッダ情報と R, G, B のデータで構成されている。jpeg などの一般的なフォーマットの画像も扱うことは可能である。VC++ では CImage クラスを利用することで読み込みが可能となる。我々は Silicon Graphics が提供する Image Format Library を利用している。前記の RAW 形式

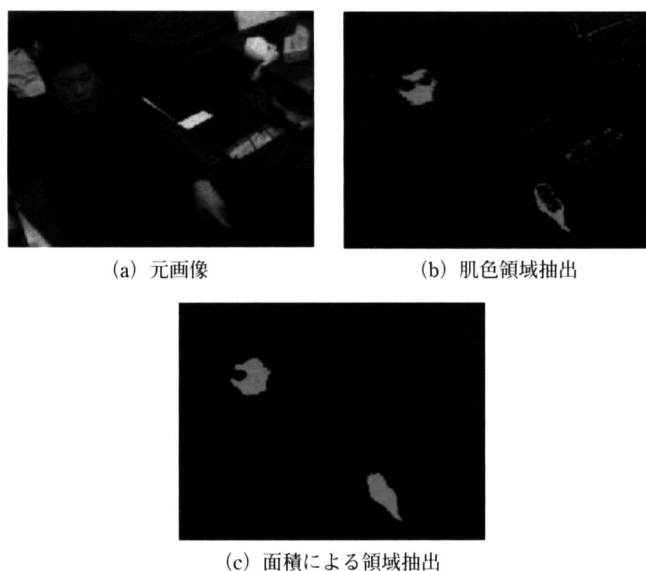


図2 画像中の肌色領域の検出

は、メーカーによってフォーマットが異なり、読み込みに苦労する。

3. 基礎的で重要な画像処理手法

基礎的で重要ないくつかの画像処理手法を紹介する。

3.1 2値化による対象の抽出

画像処理で最も良く使われる処理と言えよう。一般的な意味での2値化とは、適当なしきい値を設定し、画像の輝度値がそれより大きければ1、小さければ0とすることである。しきい値の決め方にも色々手法があるが、適当に定められるならそれでも良い。注意しなければならないのは、一般にカメラレンズには周辺減光という周辺程画像が暗くなる現象が生じ（原理的には、視線方向からの角度の4乗に比例し、コサイン4乗則と言われている）、そのため画像の中央部と周辺部とで適当なしきい値が変わってしまう恐れがあることである。あらかじめ様な明るさのものを観測した時の輝度値の分布から周辺減光の度合いを求めておき、入力された画像の周辺減光を補正した後に2値化を行う、あるいはしきい値を画像の部位によって動的に変化させるなどの手法があり得る。

多少広義に解釈すれば、輝度値がある範囲の場合に1、それ以外を0とすることも2値化であると言えよう。これをR、G、Bそれぞれで行えば、欲しい色の領域のみを選択することもできる。

抽出された1（ないし0）の部分に注目し、領域（連結している画素群）ごとにラベルづけを行う。これをラベリングと呼ぶ。そして、面積（＝領域を構成する画素の数）などの特徴量を用いて領域の中から欲しい領域を抽出する。領域の特徴量には、ほかに周囲長、円形度、外接長方形の幅・高さ（フェレ径とも言う）、モーメント、主軸ベクトルなどが用いられる。これらを求めることは手段にも目的にもなりうる。図2に画像中から肌色領域を抽出し、面積が大きな領域2つを手、顔領域として抽出した例を示す。なお、この例では、次節で述べるモルフォロジー処理

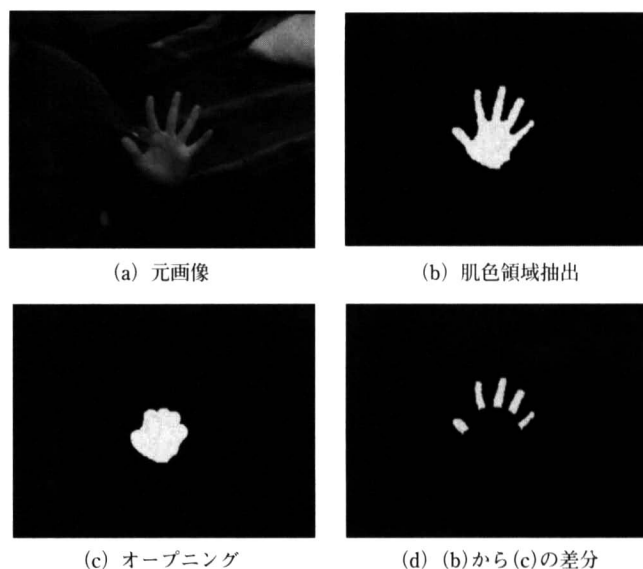


図3 モルフォロジー処理による指領域の検出

を併用している。

3.2 モルフォロジー処理

モルフォロジー（morphology）は形態という意味であるが、画像処理では、領域を膨張させたり収縮させたりすることで、領域の小さな突起や欠けを検出する処理を言う。

膨張：自身を含む近傍画素のいずれかが1であれば1

収縮：自身を含む近傍画素のいずれかが0であれば0

を組合わせる。適当な回数だけ収縮を行った後、同じ回数だけ膨張を行う処理をオープニング、逆をクロージングと言ひ、それぞれ突起、欠けを除去した画像が得られる。また、元画像からこれらの画像を引くことで、突起や欠けが検出される。これらの処理は欠陥検査などで利用される。なお近傍は、注目画素の上下左右を対象とする場合を4近傍、斜め上・下も含める場合を8近傍と呼ぶ。双方利用されるが、4近傍の方が多用されるように思う。図3に、オープニング処理を利用して指（＝突起）を検出した例を示す。

3.3 エッジ抽出

画像中で輝度値に大きな変化を生じている箇所をエッジと呼ぶ。3.1節で述べた領域（エッジと逆に輝度値があまり変わらない箇所）と同様に重要である。エッジの抽出は、変化を検出する訳であるから、微分値が大きい箇所を求めれば良い。第(i, j)番目の画素の輝度値を $f(i, j)$ と表した時、横方向の微分は、差分 $f(i+1, j) - f(i, j)$ で近似することが出来る。これだと微分位置が半画素ずれるので、注目画素の左右の輝度値を用い、さらに平滑化によるノイズ除去を行うために上下の行をあわせて3行分を用い、 $f(i+1, j-1) - f(i-1, j-1) + f(i+1, j) - f(i-1, j) + f(i+1, j+1) - f(i-1, j+1)$ で微分を求めることが良く行われる。縦方向も同様である。これをPrewittオペレータと呼び、図4(a)のように表記する。また、中央の行の重みを2倍にした図4(b)に示すフィルタをSobelオペレータと呼ぶ。これら2つが微分処理によく用いられる。なお、微分値のスケールを合わせるためにはそれぞれ6, 8で割る必要があるが、無視されること

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

(a) Prewitt オペレータ (左：横方向、右：縦方向)

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

(b) Sobel オペレータ (左：横方向、右：縦方向)

図4 画像の空間微分に用いられる代表的なオペレータ

が多い。求まった横、縦方向の微分値の絶対値和あるいは2乗和の平方根で微分値の大きさ、arctangentで方向を求めることができる。

なお、図4に示すような3×3のオペレータは画像処理ではしばしば用いられる。Photoshopなどのソフトウェアがあれば、これらを手軽に試してみることができる。例えばPhotoshop Element3.0なら、フィルタ→その他→カスタムを選べば良い。

これらのオペレータで求められたエッジ画像を2値化し、細線化(線の太さを1にすること)すれば、線画を得ることができる。

3.4 テンプレートマッチング

テンプレート画像と呼ばれる参照画像を用意し、入力画像中からテンプレート画像に最も似ている箇所を探す処理をテンプレートマッチングと呼ぶ。類似度を評価するには、差の絶対値の和(SAD, Sum of Absolute Distance)、差の2乗和(SSD, Sum of Squared Distance)、相関係数などが用いられる。相関係数を用いるものは、正規化相関と呼ばれ、比較する画像の間で明るさやコントラストの変動があっても影響がないため、安定したマッチングが可能である。値が1に近い程、類似しているという判定となる。図5において、

$$\text{SAD : } \sum_{i=1}^n |x_i - y_i|$$

$$\text{SSD : } \sum_{i=1}^n (x_i - y_i)^2$$

$$\text{正規化相関 : } \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$$

で与えられる。ただし、 σ_x , σ_y は x , y の標準偏差、 σ_{xy} は共分散である。

テンプレートマッチングは、欲しい対象を画像中で探すのに利用可能であるし、2台のカメラを用いて三角測量で

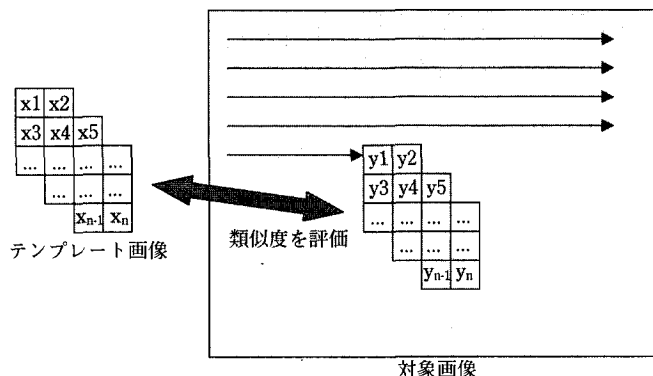


図5 テンプレートマッチング

距離を算出するステレオ計測において左右の画像で対応する箇所を探すのにも用いられる。また、時系列の画像でテンプレートマッチングを行い対象を追跡すること(トラッキングと呼ぶ)もよく行われる。

3.5 画像の差分

2枚の画像の差分を求めることも色々な用途に利用可能である。例えば、同じになるべき画像の差分から欠陥部分を求めることもできる。また、背景画像を登録しておいて入力画像との差分を求めることで(背景差分)、侵入者の検出などにも用いることができる。連続する画像間での差分を求めることで(フレーム間差分)、移動している箇所の検出も可能である。

4. おわりに

以上、駆け足であったが、画像処理に必要なハードウェア・ソフトウェアと基礎的かつ重要な画像処理手法に関して概観した。本稿で画像処理に興味を持たれたら、是非詳しい書籍を当たって頂きたい。数多くの参考書が出版されているが、例えば、基本を押さえている⁶⁾、画像応用技術専門委員会編で応用事例も豊富な⁷⁾、新しい内容も含め読みやすく書かれている⁸⁾、網羅的な⁹⁾あたりがお勧めである。雑誌では、画像ラボ、映像情報インダストリアル²の2誌が発行されている。また、ハードウェアやソフトウェアの情報収集には、パシフィコ横浜で6月に開催される画像センシング展、12月に開催される国際画像機器展への参加がお勧めである。

参考文献

- 1) <http://www.tc-iaip.org/>
- 2) HALCON 活用法, リンクス出版事業部, (2004). 購入は <http://linx.jp/>より.
- 3) <http://www.intel.com/technology/computing/opencv/>
- 4) 井上誠喜他: C言語で学ぶ実践画像処理, オーム社, (1999).
- 5) 長谷川純一他: 画像処理の基本技法〈技法入門編〉, 技術評論社, (1986).
- 6) 田村秀行: コンピュータ画像処理, オーム社, (2002).
- 7) 精密工学会画像応用技術専門委員会: 画像処理応用システム—基礎から応用まで, 東京電機大学出版局, (2000).
- 8) デジタル画像処理, CG-ARTS 協会, (2004).
- 9) 高木幹雄, 下田陽久: 新編 画像解析ハンドブック, 東京大学出版会, (2004).