

まえがき デジタル画像処理は身近で実用的な先端技術である。人間の視覚機能を理解し再現しようという、未知の分野を拓く科学でもある。それでいて、処理対象や結果を画像として見て確かめることができるため、初心者にとってはとっつきやすい学問分野である。一方、非常に奥が深い研究分野でもある。画像生成の成り立ちを理解するための物理学、3次元空間と2次元の画像空間との関係を記述する幾何学、画像データのモデル化を助ける代数学など、知識や道具として援用する学問分野は多義に渡っている。高校数学や線形微分方程式等の大学の高等数学が、一体何の役に立つのだろうと訝っていた人も、これらを駆使した様々な画像処理アルゴリズムや理論に出会うたびに新たな感動を覚えることだろう。

また、実際にプログラムを書き画像処理システムを構築して、人間の眼から脳に至る視覚機能を人工的に再現してみると、いかに人間が複雑で精巧な処理を行っているか、人間の知能の高度さ複雑さに驚嘆するだろう。画像処理研究は人工知能研究の一分野として始まり、いまはそれ自身が複数の分野を形成するほど大きく成長している学術分野であり、人工知能学の一翼を担っている。人工知能研究の目的は人工的な知能を作ることと、それを通して人間の知能を理解することである、と言われている。画像処理研究も、人間の視覚機能を人工的に実現することと、それを通して人間の視覚を理解することに他ならない。是非、この高度で複雑な視覚機能の解明と実現に挑戦して、未知の世界を探索していただきたい。

本書は、これらの挑戦と探求の入り口を読者に紹介するのを役割とし、情報系の学部学生向けに書かれている。また、学生のみならず、画像処理を道具として使っているユーザと画像処理の研究開発者を読者として想定している。そのため基本的な画像処理手法を中心に説明し、高度な手法は数学的道具だてがどのように使われるかの原理を紹介して、画像処理の面白さを伝えることを目指している。

画像処理は、いつのまにか日常的な技術になっていて、読者は画像処理技術のユーザとしてその恩恵にあずかっている。例えばデジタルカメラ（デジカメ）が一般人でも利用できるほど広く普及するようになり、デジタル画像処理が非常に身近になったのはつい最近のことである。撮影の基本機能において既に画像処理が多用されているデジカメで、顔検出や笑顔検出のパターン認識の機能が付与されて、画像処理とは何かを画像処理の研究開発者でなくとも具体的にかつ身近に体験ができるようになった。また、Google Map¹などの衛星画像と航空写真を統合した地図検索システムなどは大規模画像データベースシステムの具体的な成果である。一方、画像を用いたパターン認識の能力は、飛躍的な発展を遂げているといってよい。郵便番号の数字自動読み取りをはじめとして、指紋認証、LSIのパターン検査やキートップの印刷検査など、人間が行っていた振り分けや、工業製品の品質検査を機械に任せられるようになった。また、テレビ会議やデジタル放送も画像処理の成果の賜物である。情報処理に携わる読者には、それらの背後にある理論を理解して、それがどのように使われているのか、また、新しい問題に対してどう適用すればよいのかといった応用のきく知識を学んでほしい。日々研究が進められている最先端技術については、最新の研究論文やその解説書にゆだねることにするが、簡単な紹介や文献への引用（コラムとして？）を随所に提示している。卒業研究や大学院で画像処理に取り組もうと考えている読者には、これらの最先端の技術に早く触れることを勧める。

画像処理は理論やアルゴリズムを理解するだけではつまらない。やはりプログラムを書いて処理を体験することが手っ取り早く学習する道である。とはいえ、本書は画像処理プログラミングブックではない。アルゴリズムをいかにプログラムで実現するかの習得は、自学自習に任せる。幸いなことに、オープンソースの画像処理ツールキットとしてOpenCV²などが公開されており、画像処理プログラミングへの取り付きは非常に簡単になった。本書

¹ Google 社のサービス，登録商標

² <http://opencv.jp/>

で紹介するアルゴリズムは言うに及ばず、最新のアルゴリズムも公開されているものがあるので、是非利用を勧める。ただし、便利になった分、モジュールがブラックボックス化してしまっている。画像処理を研究対象とする読者は実現されているアルゴリズムが具体的にどのような処理をしているか、大事なツールについてはプログラムコードを見て理解して使うことを心がけるべきである。そのためのてがかりを提供することも、本書の役目である。

また、最近、最先端の手法が学会のチュートリアルで紹介されたり、勉強会と称して有志が集まって情報交換する場がある。これら志を同じくする学会やコミュニティはぜひ使うべきである。インターネットの発展で、ネット上で膨大な情報が手に入るようになったが、細かいノウハウなどは、顔の見える同士、席を並べて尋ねたほうが抜群に効率がよい³。

画像処理は実用的な学問である。すでにさまざまな分野での応用がなされているが、著者は長年、マンマシンインタフェースのための、画像処理技術を組み込んだインタラクティブなシステムの研究を進めてきた。現在に至って、この分野の研究開発従事者が増え、一部は読者らの身近なところで使われ始めている。実世界でのインタラクティブなシステムは、環境の変動要因の中で動作することが必要であり、様々な工夫がなされている。そこで、本書では、画像処理の数ある応用のうち、このインタフェース構築における人間動作のセンシング処理の手法やインタフェース構築のための基礎知識を構築することを目指して、章立てをした。それぞれが用いている画像処理技術を説明しながら、画像処理を組み込むシステムの設計指針について解説する。読者が画像処理を使ったインタフェースを構築する際に適用すべき知識を獲得することを念頭においた。具体的には、本書はまずデジタル画像処理の概念に始まり基本的なアルゴリズムの根本の原理を理解することを目指す。そして画像処理を用いたインタラクティブなシステムを構築しうる基礎知識を授けることを目指す。

出版社、関係者への謝辞。

2011 年 10 月

間瀬健二

³ いずれネット上でも席を並べているかのような画像を含む、雰囲気通信環境ができれば、物理的に集まらなくてもよくなる日がくるかもしれないが、人間のインタラクションの能力にコンピュータが到達するには、さらに時間がかかるだろう。

目 次

1 概 要

1.1	画 像 処 理 と は	1
1.1.1	インタフェースのセンサとしての画像処理	2
1.1.2	情報メディアの 1 要素としての画像処理	3
1.1.3	パターン認識システムにおける画像処理	5
1.2	画像処理のさまざまな型	6
1.3	画像処理の応用分野	7
1.4	デジタル画像の表現	8
1.4.1	様々なデジタル画像	9
1.4.2	画像の 1 次元分布	12
1.4.3	デジタル画像における距離の概念	12
1.4.4	画像の計算機内表現	14

2 画像の形成過程

2.1	画 像 の 形 成	19
2.1.1	投影モデル	19
2.1.2	レンズモデル	21
2.2	色の認知と表現	22
2.2.1	光の物理学と人間の色認識	22
2.2.2	色 空 間	24
2.3	画像センシング	26
2.3.1	画 像 センサ	26
2.3.2	リモートセンシング	27
2.4	画像表示装置	28
2.5	立体知覚と 3D 表現	29

3 デジタル画像の基礎と基本演算

3.1	標本化と量子化	31
3.1.1	標本化定理	31
3.1.2	量 子 化	33

3.1.3	画面走査と画素値の参照	33
3.2	基本的な画像処理	33
3.2.1	点演算	33
3.2.2	画像間演算	36
3.2.3	局所演算	36
3.2.4	その他の局所演算	40
3.2.5	セグメンテーションのための画像処理	41
3.2.6	画像の幾何学	43
3.3	特徴抽出	43
3.3.1	点特徴	43
3.3.2	線特徴	43
3.3.3	領域特徴	44

4 画像の動きと立体構造抽出

4.1	オプティカルフロー	49
4.1.1	時空間画像	49
4.2	立体構造の抽出	49
4.2.1	ステレオ視の幾何学	49
4.2.2	画像パターン投影ステレオ	49
4.2.3	動きからの構造再現	49

5 画像処理を用いたインタフェース

5.1	ジェスチャの認識	51
5.2	個人の識別と表情の認識	51
5.3	拡張現実: AR	51

1

概要

1.1 画像処理とは

人間は3次元の外界を、眼から脳に至る視覚 (vision) を通して観測・理解し、ほかの様々な感覚と組み合わせて外界の情報を手に入れて、それをもとに判断し、行動している。視覚は、この人間の認知・判断・行動からなる過程における重要な外界知覚の一つである。この人間の知覚の機能である視覚情報処理を解明し、それを人工的に再現することは、外界から情報を得て自動的に作業する人工物を構築するために重要である。カメラなどの視覚センサは、人間の眼の機能を模倣してつくられていて、2次元平面パターン情報である画像 (image) として外界を観測することができる。一旦デジタル信号として画像がデータ化されると、コンピュータによる情報処理が可能になる。

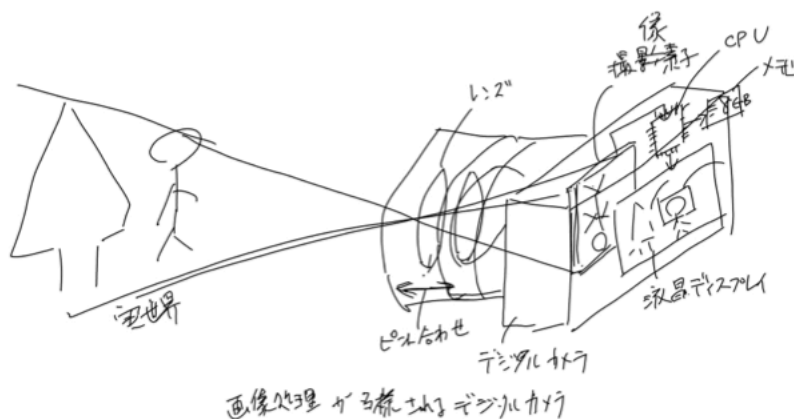
画像処理 (image processing) は、この画像に関して何らかの処理をする視覚情報処理の一分野である。「何らかの処理」とは、対象世界をとらえて画像形式で信号化・パターン化する「イメージング (画像入力)」、画像情報を見やすくするための「画像補正」や「画像強調」、パターン認識のための「特徴抽出」、伝送・蓄積のための「画像符号化」と「画像データベース」、そして、画像を生成表示するための「画像出力」「コンピュータグラフィックス」などの多彩な分野がある。これら全体をとらえて「広義の」画像処理ということもあれば、画像強調、特徴抽出、画像符号化の部分だけに注目した「狭義の」画像処理を対象とすることもある。一方、画像処理と並行して、コンピュータビジョン (Computer Vision, CV) と呼ばれる分野がある。CV とはコンピュータの視覚のことであり、画像の被写体となった対象世界を記号で記述することをいう。

デジタルカメラ内部の構造を見ながら、一連の画像処理の役割を説明すると以下ようになる (図 1.1) 図 1.2)。

1. 実世界をレンズでとらえて、カメラの奥にある撮像素子に、画像パターンとして結像する (イメージング)。
2. 画像の光量の過不足などのせいでコントラストが低い場合には、コントラストを強める (画像強調)。
3. 人の顔の特徴を使って顔領域を見つけて、撮影時に顔の位置にピントを合わせる (特徴抽出)。
4. 得られた画像を符号化圧縮してファイルとしてメモリなど記録媒体に書き込む (画像符号化、画像データベース)。
5. 得られた画像の顔領域に矩形を描画し (コンピュータグラフィックス)、液晶ディスプレイに表示する (画像出力)。

ここで、撮像素子上にパターンとして結像された画像データは、デジタルカメラ内の組込みコンピュータのメモリに転送されて、上記の様々な処理が行われている。また、カメラ全体をイメージング装置と考えて、得られたファイルをコンピュータに移してさまざまな画像処理をすることもある。この際、メモリ上の画像データは、一般的には2次元配列として扱われ、その行列番号で指定された場所には、その点の画像の明るさや色彩を示す値が入っており、画像処理とは、この値を何らかの計算式で操作することに帰着する。その操作は、2次元配列内の

図1.1 画像処理技術の例：デジタルカメラの場合



2

図 1.1 画像処理の例：デジタルカメラの場合 (図中「多様される」は「多用される」の誤り)

個々の値の変換もあれば、2次元配列から領域の重心や面積などの統計量へ変換など、いくつかの型がある。この型については、画像処理の様相として後述する。

このような画像処理の基本的な概念は、19世紀初頭のカメラの発明により開花したといってもよい。3次元の情景をレンズシステムを通して2次元の写真乾板に焼き付けるというイメージングの基本理論がまず構築された。また、フィルムの化学反応を使った現像技術などアナログ画像処理技術が多く開発されている。そして、1843年にファクシミリの原理が発明され、さらに1926年に電子式テレビ受像機が日本で発明されるなど、画像を走査線に分解して入力して伝送再生処理できることがわかると、1940年代のデジタルコンピュータの発明により、デジタル画像処理が行われるようになった。

1.1.1 インタフェースのセンサとしての画像処理

一方、デジタルコンピュータを使う際のユーザインタフェースも徐々に進化をとげた。初期のデジタルコンピュータでは、オペレータがトグルスイッチで機械語命令を直接入力していたが、まもなくテレタイプと呼ばれるタイプライタ式の入出力ユーザインタフェースが現れると、キーボードでプログラムやコマンドを入力すると、返答が用紙に刻印されるというものであった。テキストユーザインタフェース (Text User Interface, TUI) と呼ばれている。1960年代初頭に、マウスが発明され、Sketch Pad と呼ばれるインタラクティブ・グラフィックスのシステムが発表され、コンピュータを、数値計算のための計算機としてではなく、インタラクティブ (対話的) なシステムを実現する道具として使う概念が定着した。それまでコンピュータは、主にシミュレーションなどの科学技術計算に用いられていたが、画像や図形をコンピュータでインタラクティブに処理するという全く新しいパラダイムの始まりであり、現在主流のグラフィカル・ユーザ・インタフェース (Graphical User Interface, GUI) として定着した。

図1.2 デジタルカメラにおける画像処理

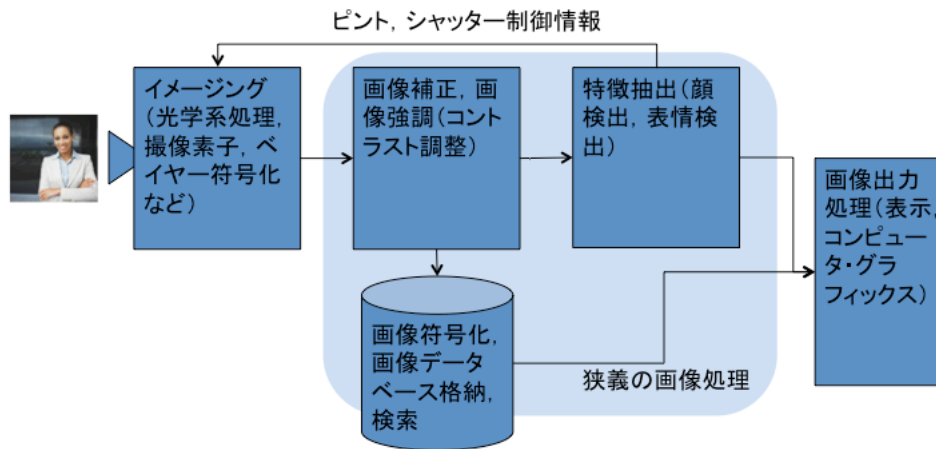


図 1.2 デジタルカメラで用いられる画像処理

さらに、1990年代にコンピュータの性能が格段に向上して、カメラで撮影した画像をリアルタイムで処理して、人間と機械間のインタフェースとして用いることが可能になってきた。このインタフェースは、現在主流となっている音声・ジェスチャインタフェース¹と呼ばれるものの一部を担い、1990年代から多数の研究成果がある。これはコンピュータの“耳”，“眼”，“頭”を良くして、人間の言葉や行動を理解したり、人間の意図や状況を理解し、適切に対応し、人間の負荷を軽減するヒューマンインタフェースである。それは、音声や画像のメディア処理にかぎらず、センサ、AIなどの要素技術から社会心理や認知科学まで広い分野の横断的な研究を統合した技術の結晶でもある³⁾。

インタフェースのセンサとしての画像処理システムの構成は図1.3のようになる。ここで、画像処理インタフェースは、人間の動作から、GUIで用いられるマウスポインタやキースイッチなどの位置・大きさなどの量的な情報や、キーボード入力にかわる記号ラベルを、リアルタイムで取得し主アプリケーションに通知しなければならないという要求条件が課せられる。

1.1.2 情報メディアの1要素としての画像処理

インターネットと画像処理の普及により、デジタル画像が人と人のコミュニケーションの媒体（メディア，media）として使われるようになった。とくに、デジタル画像は、計算可能なメディアすなわち計算メディア（computational media，情報メディアと呼ぶこともある）として、様々な有用性を備えている。計算メディアは、デジタル計算可能であるという性質にもとづき、一般に、

- (i) 再現性が高い (reproducible)
- (ii) 変換可能である (transformable)

¹ パーセプチュアル・ユーザ・インタフェース (Perceptual User Interface, PUI) またはマルチモーダル・インタフェース (Multi-modal Interface) とも呼ばれる

図1.3 ジェスチャインタフェースのための画像処理



図 1.3 インタフェースのセンサとしての画像処理システム

(iii) 移動性が高い (portable)

(iv) 処理可能である (processable)

という特徴がある。このようなメディアでメッセージや情報を表現して、広範囲に発信すると、瞬時に他者に伝えたり、共有するだけでなく、他者が変換・加工して利用できるようになる。Web がその良い例である。

たとえば、図??に示すように、Web による画像情報発信と共有における画像処理は、最初のコンテンツの素材作成（表現）におけるデスクトップパブリッシング（DTP）、発信時の図形・画像記述言語の仕様、記録蓄積時の画像データベースの構成と画像ファイル形式、画像の効率的伝送・蓄積のための画像符号化圧縮、大量の画像から希望の画像を見つける画像検索、そして、得られた画像を表示する表示装置とコンピュータグラフィックスなどがある。

大量の画像データが集積されると、今度は、大量のデータの中から自分が必要なものを選別する手段が必要になる。現在は、画像データにタグ²(tag) データまたは注釈（アノテーション, annotation）データを情報発信者らが付けることによって、検索の際のつながりを提供している。最近の研究では、これらのタグとして、画像が表現している内容を画像処理によって識別した名前を使ったり、すでにタグがついている類似している画像を見つけてそのタグを割り当てるなど、画像処理とパターン認識を駆使して自動的にタグ情報を抽出して、画像検索を効率化する研究が盛んである。

このように、情報の表現と配布流通が自由・広範囲になると、画像処理を使った芸術的創造性の発揮も行われる。市販、無償にかかわらず多くのペインティングソフトウェアが、画像処理の多くの概念やアルゴリズムをコンパクトにまとめている。たいてい、計算メディアの特徴を活かした、レタッチ（筆入れ, processable）やアンドウ

² 当初タグはテキストデータに埋め込む、当該テキストの属性等を表すマークアップ記号のことを表していたが、現在は、テキストに限らず画像や音声データの属性表現などに使われる。

図 1.4 イラスト：目や耳の機能をもつコンピュータとの対話

(やり直し, reproducible) などができるようになっており, 初心者にも易しいお絵かきツールとして提供されている. また, 一旦ファイル化された画像データは, 他のコンピュータに移動 (portable) して再現できる. 一旦決定した色合いを変換 (transformable) して修正することも可能である.

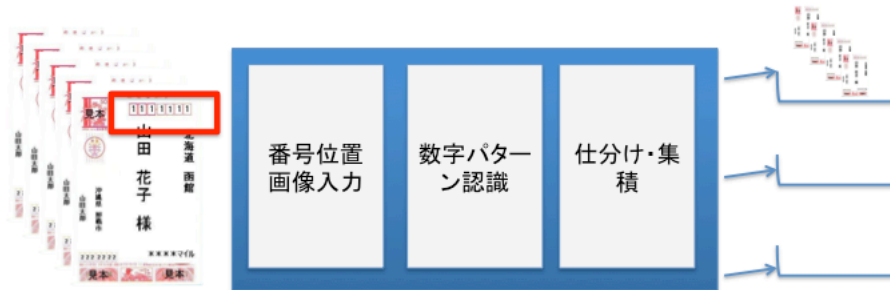
1.1.3 パターン認識システムにおける画像処理

前述の, デジタルカメラにおける画像中の顔領域の抽出や, 画像に写っている内容のタグ情報を自動抽出するために, パターン認識 (pattern recognition) と呼ばれる処理が行われる. パターン認識は, 画像だけでなく, テキスト, 音声, その他のデジタル信号を含む, あらゆるパターンを対象にした概念である. 本書では, 原則的に, 画像の中から目的のパターンを見つける手法としてパターン認識をとらえる. パターン認識の基本的な考えは, 与えられた画像パターンを, パターンのいくつかのクラス (カテゴリともいう) の一つに割り当ててクラス名のラベルをつけることである. たとえば, “0” から “9” までの数字がかかれた画像を入力して, 「ゼロ」「イチ」「ニ」,, 「キュウ」のどれかの数値のクラスに割り振る作業である. これを実現したのが, 郵便番号の自動読み取り装置である (図 1.6 参照). 画像を用いたパターン認識システムには, キーボードのキートップ印刷や LSI の品質 (良品) 検査, 指紋認証や網膜認証などのバイオメトリックス認証による入退室管理, デジカメの人検出・表情検出機構, バーコード認識など, 多くが実用化されている.

図 1.7 は, 一般的なパターン認識 (pattern recognition) システムのプロセスを 5 ステップで図示しており, 画像パターンの認識もこれに沿う. 各ステップは以下ようになる.

- (1) パターン観測: 画像の入力 (input) ・観測によって外界の対象をデジタル画像として取り込む.
- (2) 前処理 (pre-processing): 入力された画像から, 雑音除去 (noise reduction), 処理対象のみの切り出し (cropping), 細線化 (thinning), セグメンテーション (segmentation) などパターン認識対象となる画像を整える.

図1.6 郵便番号自動読み取り装置 (区分機)



7

図 1.6 郵便番号自動読み取り装置

- (3) 特徴抽出 (feature extraction): 前処理済みの画像から、線分ならばストロークの方向コードによる表現や、セグメントの幾何学特徴を計算したり、パターンのフーリエ特徴量や多次元空間尺度などを求めて、特徴ベクトル $\mathbf{x} = (x_1, x_2, \dots, x_n)$ を作る。
- (4) パターン識別 (pattern classification): 特徴ベクトルをパターン識別器に入力して、パターンの識別クラスを出力する。パターン識別器は、あらかじめ辞書パターンを登録しておいて特徴ベクトル間の類似性をもとにテストパターンの識別を行う。あるいは、学習パターンを用いた機械学習 (machine learning) により、パターン識別モデルを学習することが行われる。ベイズ決定則、最近傍則、隠れマルコフモデル (Hidden Markov Model, HMM)、ニューラルネットワークなどの手法がある。
- (5) 後処理 (post-processing): 識別結果を整理したり、外界に出力 (output) する。画像中に見つけた目的パターンの計数や計量、強調表示などを行う。

これらのステップは、はっきりと分割されることもあれば、複数のステップを統合した処理が行われることもある。また、中間処理結果を前のステップにフィードバックかける方法もある。パターン認識と機械学習の詳細については、専門書 (たとえば、石井ら¹⁾⁾, Bishop²⁾⁾) を参考にすること。

1.2 画像処理のさまざまな型

これまで画像処理が使われる場面を、典型的な画像処理システム、ヒューマンインタフェースのための画像処理、画像メディアのための画像処理、およびパターン認識における画像処理について見てきた。それぞれ画像処理がどのように使われているかを見ることができる。これらの例をみるときに画像処理部の機能として、以下に列挙する3つの型があることがわかる。(図1.8参照)。

図1.7 パターン認識のプロセス

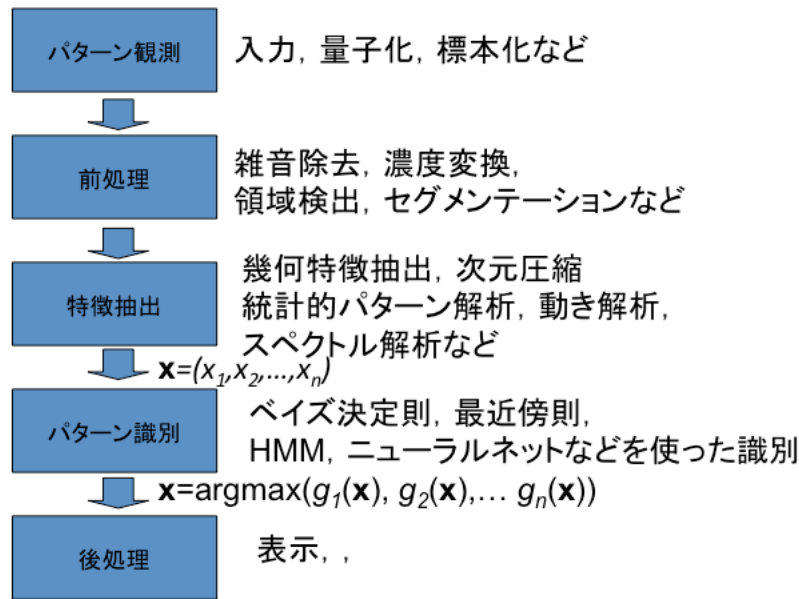


図 1.7 パターン認識と画像処理

- (PP 型: 画像から画像への変換) デジカメの例のように画像の強調, ノイズ除去, 画像の符号化, フィルタリングなどは, 画像を入力とし, 画像を出力する. 入出力画像の大きさは等しいこともあれば, 拡大縮小されることもある.
- (PV 型: 画像から記号または値への変換) パターン認識や, 幾何学特徴の抽出などをおこなう画像処理は, 画像を入力とし, 記号や数値を出力する.
- (VP 型: 記号から画像への変換) コンピュータグラフィクスなどの画像生成に関する画像処理は, 記号や数値を入力とし, 画像を出力する. たとえば $\text{draw_line}(x_1, y_1, x_2, y_2)$ という関数を呼び出すと, 2 点を結ぶ直線を引いた画像を出力する.

1.3 画像処理の応用分野

画像処理の応用分野は, すでにいくつか紹介したとおり多岐に渡る. 産業分野では, まず, 製造工程での, 産業用ロボットの位置合わせのためや部品のピッキングなどのための視覚センサとして, また, 製品の品質検査のための, LSI パターンの良不良検査, キーボードのキートップ文字の印刷品質検査などがある. また, 医療の分野では, CT や MRI などに使われるコンピュータ断層像生成の原理は, 不可視であったものを見えるようにしたことで大きな貢献がある. また, これらの画像や顕微鏡画像から, 血球や腫瘍の大きさを計測したり計数することに画像処理が活躍する. 安心安全に関しては, 事業所や街中での監視カメラ映像を監視センターで集中管理する際の伝送や記録のほか, 立ち入り禁止区域への入場者の検出, 空港などで手配者を発見する顔認証, 車載レーダによるとび出し歩行者の検出などが実用化されている. 生活の場面ではデジタルカメラやテレビ電話など, デジタル映像メディアを使った生活の記録やコミュニケーションの道具として手軽に使われ, WWW を通した映像情報の発信・入手

図1.8 画像処理の3つの型

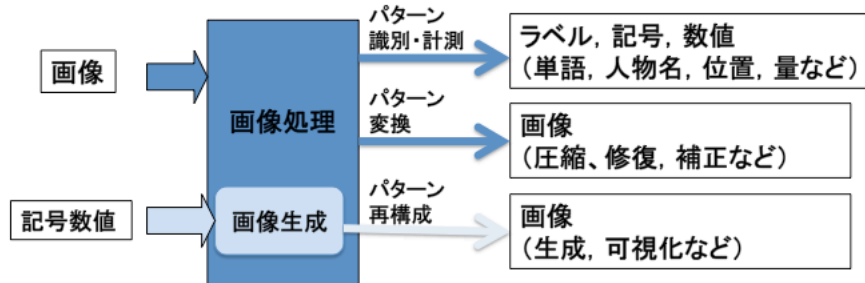


図 1.8 画像処理の 3 つの型

も盛んになった。新聞・雑誌、書籍などの印刷媒体であたりまえとなった電子出版も画像処理の貢献が大である。映画、アニメーション、デジタル絵画などデジタル画像を使った芸術・娯楽作品も多く制作されるようになった。また、ゲームなどのインタフェースにも画像処理が使われはじめている。これらは、上記で述べた、計算メディアの良さを活かして人間の情報処理能力や創造性を拡張・補完するものや、パターン認識の能力を活かして人間の知的作業を支援したり人間が不得意とする作業を行うことで、業務全体の精度や効率を高めるものを目指している。(図 1.9 参照)

1.4 デジタル画像の表現

画像処理は、2次元平面パターンの処理を基本とする。2次元平面パターンは一般に次式の実関数で表される。

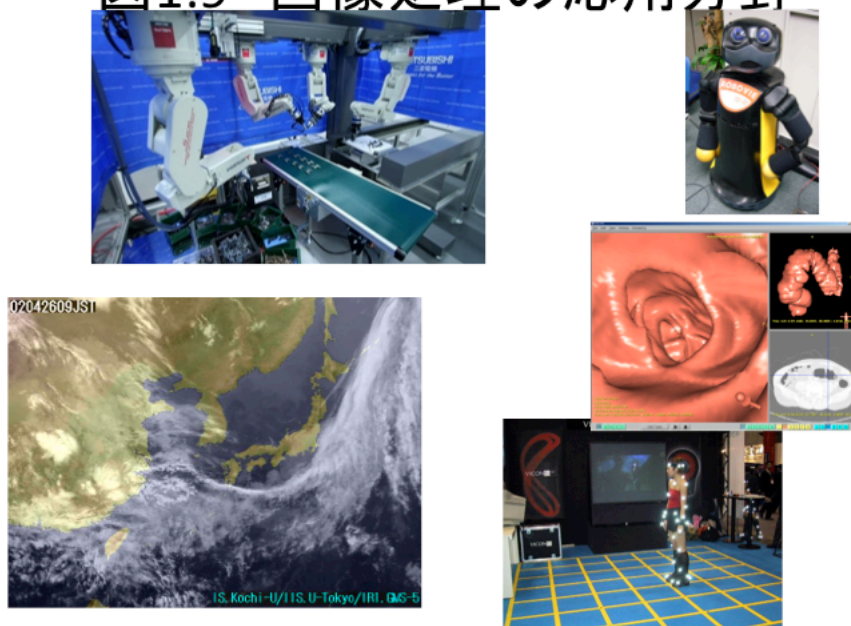
$$Z = g(X, Y); \quad \{X, Y, Z\} \in \mathcal{R}$$

ここで X と Y は 2次元の点座標を表す実数で、 Z はその点の明るさの強度を表す実数であり、それぞれ連続値をとる。デジタル画像は、点座標と明るさを離散値で表現し、一般に次の形式の関数で表す。ここで、 i, j, z は整数値である。

$$z = f(i, j); \quad \{i, j, z\} \in \mathcal{Z}$$

画像における点座標 i, j のことを格子点と呼び (i, j) で表す。格子点を中心とするある範囲(近傍)を画素またはピクセル (pixel, picture cell の略) と呼びある格子点の画素の値 $f(i, j)$ を画素値 (pixel value) と呼ぶ (図 1.10 参

図1.9 画像処理の応用分野



11

図 1.9 画像処理の応用分野—ロボットアームの制御，ロボットの目，仮想化内視鏡，リモートセンシング，モーションキャプチャー—

照). ここでは，格子点配列は正方格子とし，画素は画像全体を等方分割して得られる正方形とする³．画素値は簡単のため $f_{i,j}$ または f_{ij} で表されることがある．画素値は，画像においては一般に画素 (i,j) の明るさを表すが，以下に示すように様々な対象の表現に対応したデジタル画像がある．なお，「明るさ」は輝度，濃度などの呼び名があり，輝度値は光の強弱を表し，濃度値は色の濃淡を表すため，反対の概念であるが，特に断らない限り区別せず画素値を指す．

なお，ディスプレイやカメラにおける画素は，2次元平面上の物理量としての明るさを扱うため，便宜的に，矩形（多くは正方形）として考える．矩形の切り方は，離散値の点座標の取り方に依存する．詳しいことは標本化定理の章で説明する．

1.4.1 様々なデジタル画像

以上のデジタル画像表現を拡張して，次のような様々な画像表現が行わる．

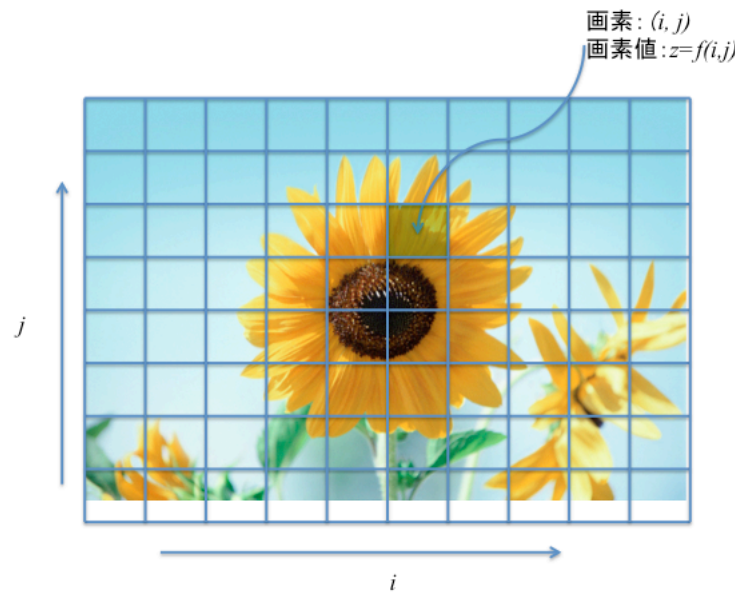
(1) **二値画像と濃淡画像** 画素値が $\{0,1\}$ の二値のどちらかのみとる場合を，二値画像 (binary image) と呼ぶ．例えば文書画像 (document image) のような白地に黒の文字が書かれたものは二値画像で，白地を 0，黒の文字を 1 で表すことが通例である (図 1.11 (a))．⁴ これに対して一般の画像では，画素値が多値をとり濃淡を表すため，濃淡画像 (gray image) と呼ぶ．モノクロ写真と呼ばれる白黒の濃淡のある写真はこの濃淡画像の例である．濃淡画像では，画素値が明るさを表し，大きな値ほど白に，小さな値ほど黒に近く，二値画像と割り当てが反対になるので注意が必要である．

(2) **画像以外のデジタル画像表現** 一般に画素値は画像の各点での明るさを格納するが，数学的な表現と

³ 格子点配列を三角格子として，画素を正六角形とする方法もあるが，計算機の配列表現との親和性から現在では正方格子と正方形が主流である．

⁴ プログラム中でとる値としては，表示の簡便さから， $\{0,255\}$ の 2 値を用いることも多い．

図1.10 デジタル画像の表現



12

図 1.10 デジタル画像の表現

してはいかなる物理量や意味を示す値でもよい。例えば、地形の凸凹を表すデータでは、標高値を画素値とする画像データとして表現される (図 1.11 (b))。また、赤外線センサ (infra-red sensor) で撮影すると 2 次元的な温度分布を測定できる。この場合、温度分布が画素値として表される。さらに、レーザの反射光を使ったレーザレンジセンサ (range sensor) は、距離画像 (range image) と呼ぶセンサから物体の各点までの奥行分布が簡単に得られる。奥行き情報は人間などの処理対象を背景から抜き出すのに非常に有効な特徴であり、ジェスチャインタフェースデバイスで多用される。さらには、画像に写っている塊や領域の識別番号が画素値に割り当てられることもある。これをラベル画像 (label image) と呼ぶ。

(3) **カラー画像** カラー画像 (color image) は、一般には赤 (Red), 青 (Blue), 緑 (Green) の 3 原色の加法混色で表される。そこで、同じ画素の 3 原色それぞれの明るさを $R(i, j), G(i, j), B(i, j)$ の三つ組で表して、カラー画像を表現する (図 1.11 (c))。天気予報などで用いられる衛星画像は、光の異なる波長ごとのセンサを用いて大気や植生の状況を観測する。複数のスペクトル波長を取得し、波長ごとのスペクトル強度分布を画像で表現する。この複数の画像をまとめてマルチスペクトル画像と呼ぶ。人間の肌色は色空間で特徴的であり、顔や手の検出においてカラー画像表現は有効である。

(4) **3 次元画像** 3 次元画像 (three dimensional image, 3D image) は、CT などで撮影されたパターンで、高さ方向の座標値を k として $f_k(i, j)$ とした複数の 2 次元画像、あるいは関数を 3 次元に拡張した $f(i, j, k)$ であらわす。点の位置 i, j, k を中心とする直方体を 3 次元画素、立体素またはボクセル (voxel, volume cell の略) と呼ぶ。

なお、最近注目されている 3D 映画や立体テレビのデータは、この形式の 3 次元画像ではなく、両眼立体視のために左目用と右目用の 2 台のカメラ (または仮想カメラ) で撮影した 2 枚の画像 $f_l(i, j)$ と $f_r(i, j)$ で表されてい

図1.11 さまざまなデジタル画像データ

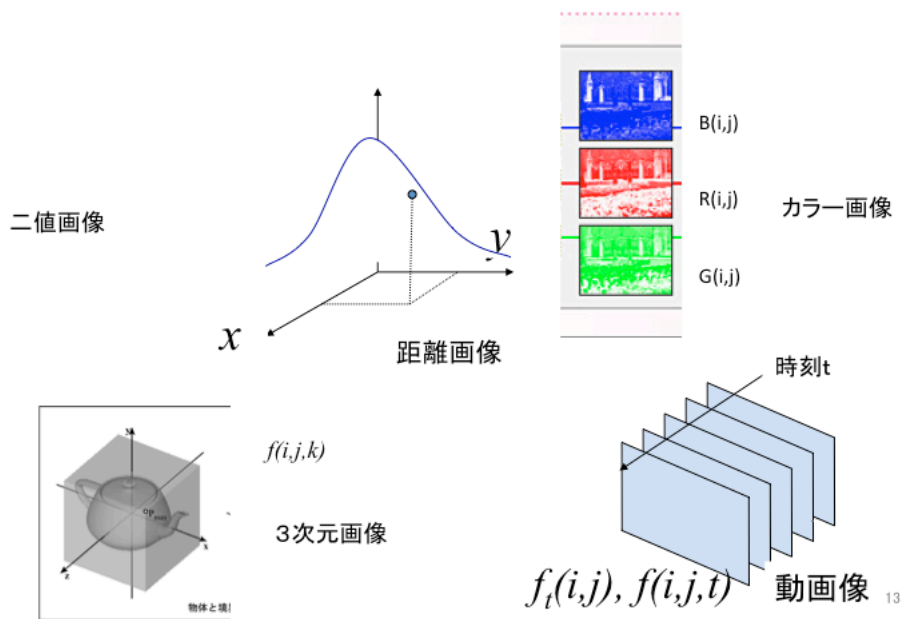


図 1.11 様々なデジタル画像

る。これをステレオ立体視画像、またはステレオ画像 (stereo pair image, stereoscopic image) と呼ぶ⁵。

さらに、コンピュータグラフィクス (computer graphics, CG) では立体感のある画像表現をしている。この場合、部屋や人物などシーンの記述には3次元座標系での点、線、面、体などが用いられるが、画像生成（レンダリングともいう）をした時点で2次元画像となっており、上記の意味での3次元画像とは異なる。いってみれば、透視表現や陰影を上手く使って3次元的な効果や錯覚を生む様々な技法が用いられて、立体感はあるが、基本的に2次元画像で表現されている。CGでは、仮想カメラ位置が自由に設定できるので、ステレオ画像を生成して両眼立体視と組み合わせることでさらに強い立体感を表現できる。一方、3次元座標系をつかったシーンの記述は、シーン内のモノの配置や動きを表現する際に多用される。例えば、人間を頭、胴、腕、手など、その部品の位置と大きさを記述することにより、人間のジェスチャを表現し、それを使って動作パターンの認識が行われる。

(5) 動 画 像 2次元の動画像 (motion image, image sequence) は、3次元画像の3番目の座標軸が時間軸になったものと考えられる。すなわち時刻を t として、 $f_t(i,j)$ または $f(i,j,t)$ で表す(図1.11(d))。動画像では、この1枚1枚の画像をフレームと呼ぶことがある。動画像を撮影するビデオカメラではこのフレームを毎秒約30枚⁶の速度で撮影し、次々と記録していく。動画像に対して、時間に対して変化しない画像あるいは1枚のフレームだけを取り出したものを静止画像 (still image) と呼ぶ。ヒトの行動やジェスチャの認識においては、動画像の処理が欠かせない。この場合、画像処理の過程でフレーム間の相関や画像の時空間特徴を使ったタイプの動画像処理と、各フレームをまず静止画像として処理してその結果を時間的に統合するタイプの動画像処理がある。

⁵ ステレオ画像のペアを一枚の画像フレームに収めて伝送する方式を、サイドバイサイド方式と呼ぶ。

⁶ 30フレーム/秒は、パラパラとフレームをめくった際に、人間の視覚が滑らかに動画として、知覚し鑑賞できる速度と言われている。コンピュータ画像処理にとって適切な速度は、対象と目的に強く依存する。例えば1フレームあたり33ミリ秒 (ms) で画像処理できない場合は、フレームレートを落とす必要がある。

これらの各種デジタル画像の表現は、さらに複数を組み合わせることもできる。

1.4.2 画像の1次元分布

以上の例のように、デジタル画像は一般に2次元空間を表現しているが、画像処理においては、画像の画素値分布を1次元的にとらえることがよく用いられる。それは、画像のある直線上の画素値分布である**プロファイル** (外形, profile), 画像をある軸にそって画素値を合計した**周辺分布** (marginal distribution), および画像全体の画素値分布である**ヒストグラム** (histogram) である。

(1) **プロファイル** プロファイルは、**画像の断面**とも呼ぶ。2次元画像の断面とは不可思議であるが、画素 (i, j) の格子点の画素値 $f(i, j)$ を高さとする3次元曲面として画像をとらえると、その高さ方向の断面をとることで1次元分布を得る。画像 $f(i, j)$ に対して $i = K$ (K : 定数) のプロファイル $f_K(j)$ は、

$$f_K(j) = f(K, j)$$

で表される。例えば、図 1.12 のように画像のある断面における明るさの変化をグラフで見ることができる。

(2) **周辺分布** 周辺分布は、2変数の同時確率分布に対する周辺確率関数にあたるもので、画像 $f(i, j)$ に対し、 i 座標方向の周辺分布 $m(i)$ は、

$$m(i) = \sum_j f(i, j)$$

で表される1次元分布である。周辺分布は、画像中の対象領域の特定に用いられ、たとえば横書き文書画像の処理においては、縦方向の周辺分布から、文字列と行間を区別することができ、文字列領域を見つけて文字認識処理にかけるなど有用である。

(3) **ヒストグラム** ヒストグラムは、対象画像領域全体にわたる画素値の頻度分布を表す。たとえば、 $[0, 255]$ の8bitで画素値が表されているときに、それぞれの画素値をもつ画素の数を数え上げることで、1次元データが得られる。このとき、画素値の1の差が雑音 (noise) の範疇であるなど詳細にみても意味が無い場合があり、数え上げの範囲を一つの値ではなく、区間として分布を調べることが多い。例えば、 $[0, 15]$, $[16, 31]$, $[16^i, 16^{i+1} - 1]$, ..., $[240, 255]$ という具合である。このとき、一つ一つの区間をビン (bin) と呼ぶ。ビンの幅は、対象の画像の性質や、ヒストグラムの使い方によって調整する。ビンの幅を等間隔として、ビンに順番に番号 n ($0 \leq n < N$) (N はビンの数) をふると、ヒストグラム $h(n)$ は、

$$h(n) = \sum_{i,j} B(i, j) \quad B(i, j) = \begin{cases} 1 & \text{if } b^n \leq f(i, j) < b^{n+1} - 1 \\ 0 & \text{else} \end{cases} \quad (1.1)$$

で与えられる。ここで b はビンの幅を表す。また、対象領域の面積 (画素数) を M とすると、

$$p(n) = h(n)/M$$

は、画素値の確率密度関数 ($0 \leq p(n) \leq 1, \sum_n p(n) = 1$) となる。図 1.13 にヒストグラムの例を示す。ヒストグラムは、濃度値ヒストグラムあるいは輝度値ヒストグラムと呼ばれることもある。プロファイルと周辺分布は1次元分布の変数が空間座標であるが、ヒストグラムの変数は、画素値区間に対応するビンの番号であることに注意すること。

1.4.3 デジタル画像における距離の概念

ものの大きさや長さを測ることは、分析をしたり、特徴をとらえて分類するために必要となる。その際に、距離の尺度 (metric) が必要になる。デジタル画像上での2画素間の距離は、4近傍距離、8近傍距離、ユークリッド

図1.12 画像のプロファイル(断面)

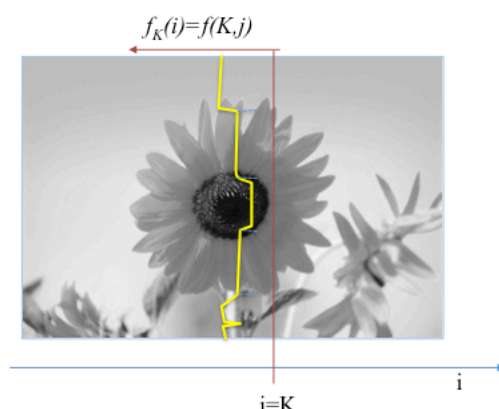


図 1.12 画像のプロファイル

距離など、いくつかの尺度を考えることができる。距離尺度の説明の前に、近傍 (neighbourhood) と連結成分について説明する。

(1) **近傍と連結成分** 近傍とは、ある注目画素 x に隣接する画素の集合のことである。ここで、画素の辺を共有して隣接する画素を「4 近傍」といい $\mathcal{N}^{[4]}(x)$ で表し、画素の頂点を共有して隣接する画素を「8 近傍」といい $\mathcal{N}^{[8]}(x)$ で表す⁷。数字の 4 と 8 は近傍となりうる画素数を表している。図 1.14 に中心画素 $X0$ に対する近傍を図示する。 $\{X1, X3, X5, X7\}$ が 4 近傍, $\{X1, X2, \dots, X8\}$ が 8 近傍である。画素 x, y が次式の関係にあるとき、画素 x と y は互いに n 隣接 ($n = 4$ または 8) しているという。

$$y \in \mathcal{N}^{[n]}(x)$$

つぎに、同じ値をもつ画素 x_1, x_n に対して、これらと同じ値をもつ画素系列 x_2, \dots, x_{n-1} が存在して、すべての画素を 4(8) 隣接の関係でたどれるとき、画素 x_1, x_2 は、4(8) 連結の意味で「連結している (connected)」という。

画像上の同じ値をもつ画素で互いに連結している画素をまとめると一つの塊をつることができる。これを連結成分 (connected component) という。

(2) **距離尺度** デジタル画像においては、連続空間で用いられるユークリッド距離も定義できるが、距離の公理を満たす距離尺度として、隣接している画素への距離を 1 単位とする尺度を定義することができる。それが 4 近傍距離と 8 近傍距離である。4 近傍距離はダイヤモンド距離 (市街地の道路をたどる様子からマンハッタン距離あるいは市街地距離) とよばれ等距離線がダイヤモンド状になる。一方、8 近傍距離は、チェス盤距離と呼ばれ等距離線が正方形となる。それぞれ、4(8) 隣接-4(8) 近傍-4(8) 連結-4(8) 連結成分との親和性が高く距離値も整数演算となり、便利に使うことができる。ただし等距離線が円でないことからわかるように異方性がある

⁷ 3 次元の voxel の場合は、面を共有する 6 近傍、辺を共有する 18 近傍、頂点を共有する 26 近傍が定義される

図1.13 画像のヒストグラム



図 1.13 画像のヒストグラム

などの問題がある．たとえば，ある画像中の水平の2点が4近傍距離で10画素だった場合，その画像が45度回転すると同じ距離が20画素になる．これを異方性という．

n 次元空間でのノルムの計算の観点から距離尺度を定義すると，2点間のユークリッド距離 d^E ，4近傍距離 d^4 ，8近傍距離 d^8 はそれぞれ以下のようにになる．

$$d^{(E)}(\mathbf{x}, \mathbf{y}) = \{\sum_{i=1}^n \{|\mathbf{x}_i - \mathbf{y}_i|^2\}\}^{\frac{1}{2}} \rightarrow \mathbf{L}_2 \text{ノルム}$$

$$d^{(4)}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \{|\mathbf{x}_i - \mathbf{y}_i|\} \rightarrow \mathbf{L}_1 \text{ノルム}$$

$$d^{(8)}(\mathbf{x}, \mathbf{y}) = \max_i \{|\mathbf{x}_i - \mathbf{y}_i|\} \rightarrow \mathbf{L}_\infty \text{ノルム}$$

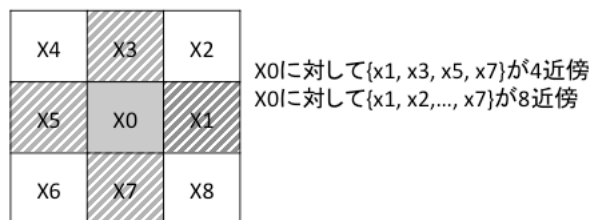
それぞれの距離尺度による中心の画素から各画素への距離は図 1.15 のようになる．

1.4.4 画像の計算機内表現

2次元のデジタル画像は，前述のように2次元配列と親和性が高いため，コンピュータプログラム内では2次元配列のデータとして表現される．多次元配列を主記憶の1次元のメモリ領域にどう配置するかは処理系の実現方法に依存するが，画像データの場合は歴史的経緯により，テレビジョン放送やファクシミリにおける順次走査方法が現在も慣習的に用いられている．

(1) **画像データの2次元配列表現と走査方向** たとえば，c言語ではI行J列の2次元画像 $a(i,j)$ は $a[i][j]$ と表し⁸，図 1.16 に示すように1行ずつ列方向（横方向，J方向）に走査して，1行終わると次の行に進むように各画素が1次元メモリ領域に納められる．各画素の値は $a[i][j]$ で参照または代入ができる．画像がカラー画像の場合

⁸ ポインタ表現では $*((\mathbf{a} + i)) + j$ と表す．



16

図 1.14 4近傍と8近傍

合には、1画素の値として3次元配列 $c[i][j][n]$ ($n=0,1,2$; RGBの各色チャンネルを表す) と表記して、RGBのデータを順に並べて走査することがある。

(2) **画像のファイル形式** 画像データを保存したり他者と交換するためには、コンピュータの外部記憶におくファイル形式を定めておく。ファイルに書かれたデータを読めば、それがどのような形式の大きさがどれだけの画像であるか知ることができるように、共通の規則に沿った必要な情報が書き込まれているのが通常である。JPEG, PBM(PGM, PPM), BMP, PNG など多様な形式が流通しており、公表されている。

1つのファイルは、大きくは [ヘッダ部][データ部][EOF] の順に並び、画像データはデータ部に ASCII 文字形式やバイナリ形式で定義された配列のデータ順に値が格納されている。そのデータの並びはヘッダ部に記述されている。ヘッダ部は、[ファイル形式識別記号][データ形式][画素数: height, width][1画素のバイト数] などが決められた順に格納されている。図 1.17 は PGM 形式のファイルの中身の例である。

かこみ：インタレースとノンインタレース

テレビジョンやデジカメのムービーなど動画データには、インタレース (interlace) とプログレッシブ (progressive) という2種類の画像の走査形式がある。⁹ それぞれ日本語では、飛び越し走査と順次走査と呼ばれる。これを知らないで動画処理をすると、正しく抽出したはずの人物がぎざぎざになってしまうという経験が初心者によくある。本文の図 1.16 の走査は、配列の各行を順次走査しておりプログレッシブ走査になる。インタレース走査は、まず奇数行だけ画像走査したあと次に偶数行を画像走査して1枚の画像全体を走査する。画像伝送において、伝送レートを保ったまま画像の描画回数を増やして高い画質の画像を表示しようとして生み出された便宜的な方法で

⁹ プログレッシブはノンインタレースとも呼ばれる

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

4近傍距離

4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

8近傍距離

$2\sqrt{2}$	$\sqrt{3}$	2	$\sqrt{3}$	$2\sqrt{2}$
$\sqrt{3}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{3}$
2	1	0	1	2
$\sqrt{3}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{3}$
$2\sqrt{2}$	$\sqrt{3}$	2	$\sqrt{3}$	$2\sqrt{2}$

ユークリッド距離

図 1.15 異なる距離尺度による 2 画素間距離

17

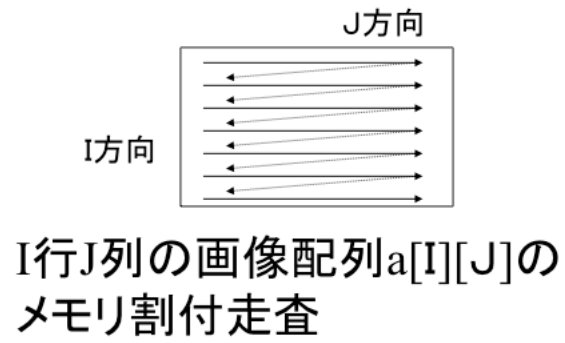
ある．図 1.18 の 1 枚の画像（フレームという）を 2 枚のフィールド画像（奇数番目の行を集めた画像と偶数番目の行を集めた画像）に分解して，順番に送ることで，フレーム周波数の倍のフィールド周波数をかせぐことができる．1 枚のフィールド画像は走査線（走査する行のこと）が間引きされているが人間の目の残像効果により，映像品質を高めることができる．動画は単位時間あたりのデータ量が多いため，伝送だけでなく，撮影，ファイル記録，ファイル読み出し，表示のすべての段階でインタレース走査は有利に働くため，多用されている．

ところが，コンピュータで画像処理をしようとするとう都合なことが生じる．撮影時からインタレース走査しているため，2 枚のフィールド画像の取得の間に被写体が動いてしまうと，その 2 枚をつかって 1 枚の画像フレームを作ると，図 1.19 楕形の歯抜け画像ができてしまう．画像処理には適さないということになる．

この解決には，(i) プログレッシブ走査のカメラで撮影しそのままファイル化する，(ii) 奇数または偶数フィールドの画像を使って処理をする（高さ方向の画素数が半分になる），(iii) 動き補償や上下走査線を使い欠損している走査線の画像を補間する，などの方法がとられる．撮影環境を自分で制御できるときは (i) が望ましいが，(ii) の方法も良く用いられる．

演習問題

- (1) 画像処理の科学的な意義と工学的な意義を述べなさい．
- (2) 身近なシステムのユーザインタフェースに画像処理技術を使っているものを 3 つあげなさい．それぞれの画像処理部において，使われている画像の特徴と，画像処理部の出力形式を推測して述べなさい．
- (3) コミュニケーションメディア（電子メール，web など）をひとつ選び，その計算メディアとしての 4 つの特徴を具体的に述べなさい．
- (4) 3 次元の $6(18,26)$ 近傍をそれぞれ図示しなさい．



18

図 1.16 画像の走査による 1 次元メモリへの割付

図 1.17 ファイル形式の例, PGM

図 1.18 インタレース走査

図 1.19 高速度で移動する人間のインタレース走査画像の例

2

画像の形成過程

画像処理の数々のアルゴリズム、とりわけ画像理解や3次元シーン記述などのコンピュータビジョンの理論を理解するためには、画像の形成過程をよく理解しておくことが必要である。というのは、画像の形成は、3次元空間を2次元平面に投影することであるのに対し、3次元空間を再構成するアルゴリズムが行っているのは、画像投影モデルを使って2次元画像から3次元への逆問題を解くという作業だからである。またパターン認識においても、ノイズや外乱などを信号に重畳した画像形成過程をモデル化した、エレガントな解法が数多く提案されている。

2.1 画像の形成

画像の形成過程を理解するには、ピンホールカメラ (pinhole camera, 針穴写真機) のモデルが用いられる。ピンホールカメラは、暗箱の一面に感光紙を張り、その反対側の面に小さな穴を開けただけの簡単な構造で、針穴を通して3次元の世界を感光紙に焼き付けるものである¹。ピンホールカメラの原理は簡単で、物体表面から放射や反射によって四方に放出された光線が、針穴の開口部で絞られて感光面に到着して画像を形成するものである。

2.1.1 投影モデル

(1) **透視投影** まず、ピンホールカメラモデルを用いての透視投影 (perspective projection) の幾何を示す (図 2.1 参照)。3次元の世界座標系を (X, Y, Z) とし、原点 O の位置に針穴が、 XY 平面に平行で $z' = -f$ の位置に投影面 Π' があるとする。投影面の X' 軸、 Y' 軸はそれぞれ X 軸、 Y 軸に平行とする。このとき3次元の対象世界の点 $P(X, Y, Z)^t$ の光線は、原点の針穴を通り投影面の点 $p'(x', y', z')^t$ に上下左右が反転した実像を結像する。ここで、投影面を原点を通過して Z 軸にそって平行に $z = f$ まで移動すると、仮想的な投影面 Π を構成でき、面上に虚像としての点 $p(x, y, z)^t$ が投影される。

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}, \quad z = f \quad (2.1)$$

このとき、 f をピンホールカメラの焦点距離と呼ぶ。透視投影は x, y の式からわかるように奥行 Z の情報を含むため X, Y に対し非線形である。この性質を使ったカメラオブスクラは中世から絵画で使われ、ピンホールカメラによる像を写しとって、遠近感のある写実的な絵画を書くのに用いられた。さらに2枚以上の透視投影画像を使うと投影点の座標から奥行を推定できる。

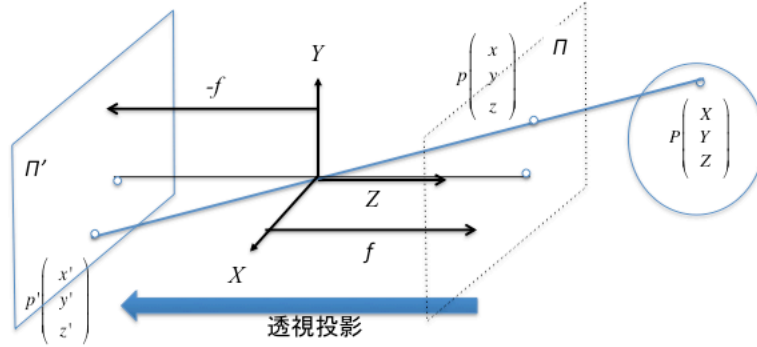
(2) **平行投影** 透視投影に対して、全ての対象世界の点の光線を Z 軸に平行に投影面に投影する場合を平行投影 (parallel projection) または正射影 (orthographic projection) と呼ぶ。平行投影は対象世界での線分の長さの比が投影像でも保持されるため、平行な直線群は投影像においても平行となる。(図 2.2 参照)。

$$x = X, \quad y = Y, \quad z = \text{const.} \quad (2.2)$$

図2.1 透視投影

- 点 $P(X, Y, Z)^t$ の透視投影像 $p(x, y, z)^t$

$$(X, Y, Z)^t \rightarrow (f \frac{X}{Z}, f \frac{Y}{Z}, f)^t$$



14

図 2.1 透視投影

図 2.2 平行投影

(3) 弱透視投影

透視投影は、奥行きを表現できる利点があるが、非線形のモデルのため解析が複雑になる。一方、平行投影は線形であるが、奥行情報が消えてしまい、3次元の再構成には不向きである。そこで、その中間をとった弱透視投影 (weak perspective projection) がよく用いられる。

まず、対象世界を対象物に近い位置 $z = Z^\circ (> 0)$ の平面 Π° (正面平行平面) に平行投影し²、つぎに、その平行投影像を射影投影するものである (図 2.3 参照)。

$$x = \frac{f}{Z^\circ} X = mX, \quad y = \frac{f}{Z^\circ} Y = mY, \quad z = Z^\circ \quad (2.3)$$

ここで、 $m = f/Z^\circ (> 0)$ は倍率と呼ばれる。 Z° が固定であるため、 x, y が X, Y に関して線形り、線形モデルで記述できる。

平行投影と弱透視投影はアフィン投影とも呼ばれる。弱透視投影は対象世界の再構成に多用される投影モデルである。

(4) 同次行列による表現 投影は、3次元空間から2次元空間への写像であるので、ベクトルの変換行列でコンパクトに表現することができる。ここで N 次元空間を表現するのに $N + 1$ 次元ベクトルによる同次座標 (homogeneous coordinate) を使った表現をすると、回転、拡大縮小だけでなく、並進 (移動) も表現できるため

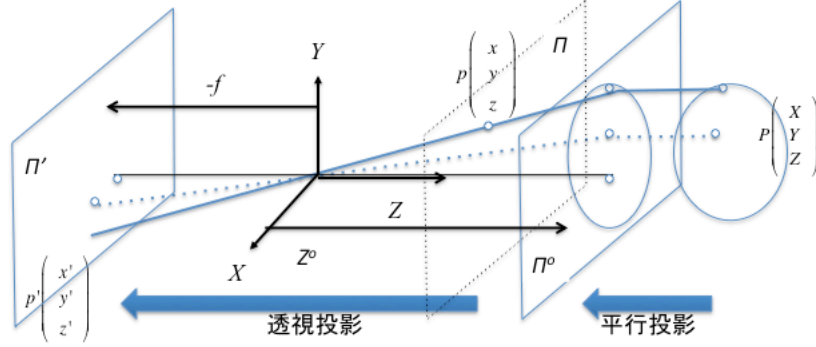
¹ 感光紙の代わりに CCD などの電子撮像素子を用いれば、電子的なカメラになる。

² 他の教科書では、世界座標軸の向きを反対にして $Z^\circ < 0$ として定式化しているものがあるが、ここでは、正方向に正面平行平面を置いている

図2.2 弱透視投影

- 点 $P(X, Y, Z)^t$ の弱透視投影像 $p(x, y, z)^t$

$$(X, Y, Z)^t \rightarrow (f \frac{X}{Z}, f \frac{Y}{Z}, Z^o)^t$$



15

図 2.3 弱透視投影

便利である．透視投影の場合の同次座標のベクトル，行列による表現は次式のようにになる．

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \quad (2.4)$$

同次座標 $(U, V, W)^t$ の 2 次元座標は， $W = 1$ になるように定数倍した $(x, y, 1)^t \rightarrow (U/W, V/W, 1)^t$ で求められる．同様に同次座標 $(X, Y, Z, T)^t$ の 3 次元座標は， $T = 1$ になるように定数倍した $(X/T, Y/T, Z/T, 1)^t$ で求められる．

なお，本書では一般に空間の座標は列ベクトルであらわし $(X, Y, Z, T)^t$ のように記述する． $(\cdot)^t$ はベクトルの転置を表し， (a, b, c) を行ベクトルという．(別のところでまとめる?)

2.1.2 レンズモデル

ピンホールカメラは，理想的にはどのような焦点距離でも，対象世界のすべての点をはっきりと投影できる³．しかしながら，針穴を通るわずかの光では明るい投影像を得ることは困難である．そこで，集光のためにレンズを使ったカメラが発明された．平行光線が凸レンズを通ると焦点に集まるという性質を使うと，位置 z の物体上の点 P から発せられた光は，レンズの焦点距離 f に従って，次式のようにレンズの反対側の位置 z' の点 P' に集光する (図 2.4 参照)．

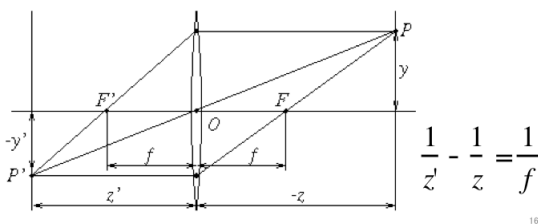
$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f} \quad (2.5)$$

³ 現実には，はっきりと結像できる焦点距離は，針穴の直径と光の波長に依存する．

ただし、式 2.5 は、理想的に薄いレンズ（薄レンズ）の場合である．一般にレンズは物理的な厚みがあり（厚レンズ）、ボケやゆがみなどの収差の問題が生じる．レンズの屈折は、異なる屈折率の媒体間を通過する光の性質によっており、屈折率は光の波長に依存する．従って、波長によって屈折率が異なるため、光収差が生じ色成分が分離した結像となる．また、球面レンズでは、光線が光軸から離れると焦点が近づくため大きなレンズではボケが生ずる．これら为了避免のために、非球面レンズを使ったり、複数の凹レンズや凸レンズを組み合わせた組レンズが用いられる．

レンズを用いることの最大の難点は、式 2.5 の関係がくずれると、レンズの光学系の点拡がり関数 (point-spread function, PSF) により点光源がボケて結像することである．いわゆる、ピントが合わない、という状況である⁴．カメラや望遠鏡は、対象物までの距離 z が変化したときに、レンズ系の焦点距離 f を調整して、結像位置でのピントを合わせる．ここで、レンズの開口面積を小さくすれば、ピンホールカメラに近づくため、ピントの合う奥行区間が広がる．この開口面積の調整は、レンズの絞り値 (F 値=焦点距離/レンズの有効口径) の調整で行われる．この奥行区間を被写界深度 (depth of field) と呼び、F 値に比例する．被写界深度の浅い（小さい）レンズ設定は、フィルムや受光素子の感度に十分な光が得られて鮮明な映像を結像できるが、ピントが合わないところでボケが生じる．このようなボケは時には写真撮影やディスプレイ表示時の表現的効果を与えるが、画像処理をして物体を抽出する目的では、対象物がボケていると処理方法の変更を余儀なくされ、その意味では邪魔な副作用である（図 2.4 参照）．最近では PSF のモデルを積極的に用いてボケのない映像を得る手法を用いたカメラが造られている．

図2.3 レンズモデルと被写界深度の異なる写真



ぼけ表現 (Bokeh) と f 値



図 2.4 レンズモデルと被写界深度の異なる写真

2.2 色の認知と表現

人間は色をどのように認知しているのだろうか．コンピュータの中で色はどのように表現すればよいのだろうか．

2.2.1 光の物理学と人間の色認識

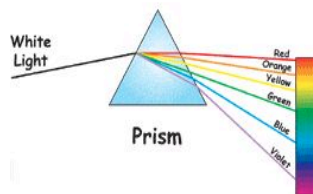
可視光は電磁波の仲間であり、我々は特定の波長帯の電磁波を可視光として目の視細胞でとらえている．熱源からの熱放射があると電磁波が放出される．太陽のような高温の熱源は広い波長の電磁波スペクトルを含んでいる⁵．

⁴ 手ぶれによるボケとは異なることに注意．

⁵ 物理的には、黒体放射とよび、黒体の絶対温度 T に従って黒体放射強度は決まる．プランクの法則に従って、波長スペクトル分布のピークは、絶対温度が高

そのため太陽光をプリズムで分光すると可視光の様々な波長に対する虹色が現れる．分光が起きるのは波長によって屈折率が異なるためである（図 2.5）．人体や地表は，熱放射によって遠赤外領域の波長の不可視光の電磁波を放射している．サーモグラフィは，黒体放射の特性を使って熱源の温度を推定して遠隔から体温や表面温度を測ることができる．また，リモートセンシングにおいて人工衛星のセンサが地表の温度を測るときも同じ原理である．

図2.4 プリズムで分光された光



18

図 2.5 プリズムで分光された光

光が物体にあたると物体表面で反射し，ヒトの目やセンサに届いて色を認識する．反射の強度は入射光の強さ，入射角度，物体表面の反射特性によって決まる．反射光は，鏡面反射と拡散反射の成分からなる（図 2.6）．鏡面反射成分は，入射光に対して正反射方向にのみ反射した光であり，拡散反射成分は，入射した光があらゆる角度に反射しているように半球方向に一様に分布して反射する光である．拡散反射は，物体表面の薄膜構造の中に入った光が，反射，屈折，吸収を繰り返して，表面から出てくるために半球方向に一様に分布する．

拡散反射の反射輝度 I_r は，最も単純なモデルでは，入射輝度 I_i をもちいて次の式で与えられる．

$$I_r = k_a \times I_i \quad (2.6)$$

ここで， k_a は反射係数である．このモデルは反射光の強度が，反射面をどの角度からみても同じ強度であることを示す．この反射をランバート反射 (Lambertian reflectance) とよびそのような表面を理想拡散面あるいはランバート面 (Lambert surface, あるいは単にランベルト (Lambert)) と呼ぶ．ランバート面は石こうの表面仕上げや綿布などがよい例である⁶．ランバート反射の計算モデルは，CG で物体表面の輝度を計算する際に多用される．

くなるにつれ低い波長に推移し，全放射エネルギーは T^4 に比例する．また，黒体の絶対温度を使って，黒体放射の光源の色を色温度 (color temperature) として表現することが可能である．色温度は工学的には，色の再現性を保持するために，撮影環境における光源の色合いの設定に用いられる．光源の色合いが変わると，その反射光も当然色合いが変わる．光源である太陽光は空気の散乱によって，日中光，日没光，晴天および曇天で色温度が異なる．そこで，撮影のときに晴天 6300K (ケルビン)，曇天 4300K，室内 3200K などの色温度値をもったフィルターから適正な値選んで補正する（この補正をホワイトバランスをとる，という）．

⁶ 陰影からの形状復元の研究で，石こう像を使ってランバート反射を仮定したものが多いのはこのためである

図2.5拡散反射と鏡面反射

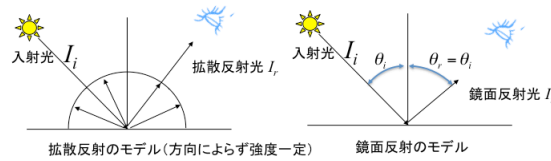


図 2.6 拡散反射と鏡面反射

このときの反射係数 k_a を拡散反射率またはアルベド (albedo, アルビード) と呼ぶ。

物体の色は、入射した光から反射特性にしたがってある波長のスペクトルが吸収され、特定の波長のスペクトルを拡散反射することで決まる。すなわち、反射係数が波長に依存した $k_a(\lambda)$ で表され、その分布が異なることによって、物体の色が決まる。例えば赤色の物体に白色光を投射すると、赤の波長以外のスペクトルが物体表面で吸収され、赤の波長のスペクトルのみが反射されて物体から出てくるために赤色を感じることができる⁷。一方、赤色の物体に緑色の光を投射すると反射されるスペクトル成分がないので、黒く見える。

さて、上記で説明した入射輝度の式 2.6 は、一本の光線の反射の様子を概念的に示しているに過ぎない。光がエネルギーであることを考慮すれば、ある方向の光源から照射されたエネルギーを微小表面で受け止め、その微小表面からどちらにどの量を放射しているかを定式化することができる。これを双方向反射率分布関数 (BRDF, bidirectional reflectance distribution function) と呼び次式の入射光と反射光の比で定義される。

$$\rho(\theta_o, \phi_o, \theta_i, \phi_i) = \frac{L_o(\theta_o, \phi_o)}{E(\theta_i, \phi_i) \cos \theta_i d\omega_i} \quad (2.7)$$

ここで、 $d\omega_i$ は物体上の微小面積の立体角、 $E(\theta_i, \phi_i)$ は角度 (θ_i, ϕ_i) から入射する放射輝度の強さ、 $L_o(\theta_o, \phi_o)$ が角度 (θ_o, ϕ_o) 方向へ、微小表面から反射される放射輝度の強さである。球座標を (θ, ϕ) で、 i と r の添字は入射光と反射光を表す。

2.2.2 色空間

色の認識は、照明、記憶、物体同定、感情などのコンテキストが関係する複雑な機能である。ヒトの眼球奥の網膜には、可視光の波長の光を吸収し反応する 3 種類の錐体 (すいたい) とよぶ視細胞がある。それぞれ吸収波長分布が異なっており、感度がピークとなる波長の長いほうから L 錐体、M 錐体、S 錐体と呼ばれる⁸。ヒトはこの 3 種類の視細胞の興奮の割合を使って色を認識する。このため、ヒトは 3 原色をつかった表色系で色を認識している。そこで、3 次元空間を使ったさまざまな色空間が定義される。

(1) 線形色空間

□ RGB 色空間

まず、線形色空間として赤 (red, R)、緑 (green, G)、青 (blue, B) の 3 原色を用いた RGB 色空間がよく用

⁷ 同じ物体でも、環境の光源が変化すると色味も変化する。夕焼け時にとった写真が赤っぽくなるのはこのためである。

⁸ それぞれ赤錐体、緑錐体、青錐体とも呼ばれるが、赤錐体は黄色に近い波長の吸収感度が高いので注意が必要である。また、感度はなめらかな山形であり特定の波長のみに急峻な反応をするわけではない。

いられる．各原色のスペクトル強度分布をそれぞれ $(r(\lambda), g(\lambda), b(\lambda))$ で表すと，目的の色は各分布の重み付き線形和として

$$U(\lambda) = R \times r(\lambda) + G \times g(\lambda) + B \times b(\lambda)$$

で表すことができる．ここで， (R, G, B) が重みである．このように，原色を加法的に混ぜて色を表すことを加法混色と呼ぶ．RGB 色空間は加法混色系である (図 2.7 参照)．

□ CMY 色空間

一方，シアン (cyan, C)，マゼンタ (magenta, M)，イエロー (yellow, Y) の 3 原色を用いた CMY 色空間は減法混色系である．一般に光の 3 原色は加法混色で，色材の 3 原色は減法混色と説明されるが，色材を使う印刷の場合は，スペクトルを吸収する顔料 (または染料) 成分を重畳することになるので，減法混色で表すことができる．例えば，シアンの顔料は赤を吸収し ($C=W-R$) とマゼンタの顔料は緑を吸収する ($M=W-G$)．シアンとマゼンタの顔料を足すと，それぞれが R と G を吸収するので白色光 W に対し， $C+M=W-R+W-G=W-R-G=R+G+B-R-G=B$ となり青が反射されるという仕組みである (ここで $W+W=W$ である)．

□ CIE XYZ 色空間

RGB 色空間は混色により別の波長の色に近い色を生成することを説明できるが，あらゆる色を生成できるわけではない．例えば，G と B の混色で青緑系の色を生成できるが，太陽光を分光した青緑色の単一スペクトル光に比べるとどんなに割合を調整しても同じ色を生成することができない．色認知の考えを適用すると，青緑色 $+ rR = gG + bB$ で等号が成立する (すなわち左辺と右辺の加法照合ができる) ので，式を変形すると 青緑色 $= -rR + gG + bB$ が得られる．そこで R (赤) の負値の混色によって明るい青緑色は生成できることになるが，負値 ($-r$) をとるということは減法照合が必要になってしまい不便である．

そこで，すべての可視光の波長を加法照合的に表現できる理想的⁹な線形色空間として，CIE (Commission international d'éclairage, 国際照明委員会) は CIE XYZ 色空間を定めた．XYZ 色空間は，混色の重みが常に正になるように決めた仮想的な色空間である．逆に，X,Y,Z の原色をある実際の可視光色として見ることはできない．XYZ 空間を $X+Y+Z=1$ 平面で切り取った断面の図を (x,y) 平面で描き，xy 色空間と呼ぶ．

(2) 非線形色空間

□ HSV 色空間

以上の線形色空間は演算が容易で工学的な価値が高く多用されているが，人間が色を操作するときの直感とは合わないことが多い．そのために古くからマンセル表色系など，色を色相，彩度，明度で表す方法が提案され美術・デザインの分野で多用されてきた．しかしマンセル表色系は，RGB 色空間との計算上の整合性が悪いので，マンセル表色系に類似した HSV 色空間が提案された．H,S,V はそれぞれ Hue (色相)，Saturation (彩度)，Value of Brightness (明度) の頭文字である．HSV は非線形色空間であるが，RGB から HSV への変換は次式で表すことができる (その逆変換も可能である)．

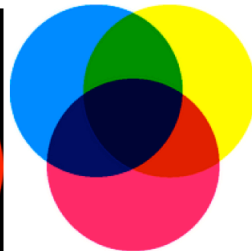
```
max = MAX(r,g,b);
min = MIN(r,g,b);
rt  = ( max - r ) / (max - min);
gt  = ( max - g ) / (max - min);
bt  = ( max - b ) / (max - min);
```

⁹ 物理的な量としては定義できないが計算モデル上は表現できる，という意味．

画像センシング(番外2) 色の合成



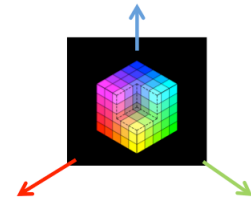
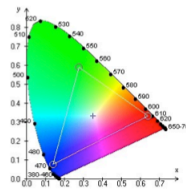
光の3原色
加法混色



色の3原色
減法混色

21

色の知覚と表現(6) 線形色空間



22

HSV(HSI)色空間

- 色相H(0度~360度)
- 彩度S: $S = WP/WE$ (0~1)
- 明度V: 高さ

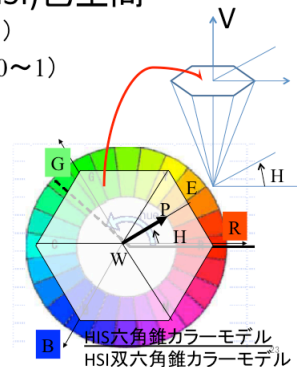


図 2.7 線形色空間 ((a) RGB 色空間, (b)CMY 色空間, (c)CIE-xy 色空間) と非線形色空間 ((d) HSV 色空間)

```

If(r == max) then H = 1/3 π (bt - gt) ;
If(g == max) then H = 1/3 π (2 + rt - bt) ;
If(b == max) then H = 1/3 π (4 + gt - rt) ;
(0 <= H <= 2 π)
V = max;
S =(max-min)/max;

```

2.3 画像センシング

対象世界をとらえてデジタル画像として信号化・パターン化する処理を、イメージング（画像入力）と呼ぶ。イメージングは、カメラのように実世界をレンズをとおして2次元画像として結像する方法や、イメージスキャナのように2次元の写真や書類を2次元画像として入力する方法が用いられる。

2.3.1 画像センサ

カメラのイメージングは、格子状に配列された光電素子（フォトダイオード）を持つ半導体撮像素子を撮像位置に設置して、光電素子に照射された光エネルギー量を計測して実現される。この撮像素子として、現在は、CCD(Charge Coupled Device, 電荷結合素子) センサと CMOS(Complementary Metal Oxide Semiconductor, 相補性金属酸

化膜半導体) センサが多用される。CCD と CMOS は信号の読み出し方法が異なる。一般に CCD より CMOS が安価であるが、CCD のほうが感度がよいとされる。

光電素子が記録する電子の量 I は、次式で与えられる。

$$I(i, j) = T \int_{\lambda} \int_{p \in S(i, j)} E(p, \lambda) R(p) q(\lambda) dp d\lambda \quad (2.8)$$

ここで、 (i, j) は画素の位置、 T は電子集積時間 (すなわち露光時間)、 $E(p, \lambda)$ は単位面積単位波長あたりの入射光強度、 $R(p)$ は素子の空間応答で素子上の位置 p の関数、 $q(\lambda)$ は入射光のエネルギー単位あたりの生成電子数で波長に依存する。

式 2.8 が表すように、得られる電子の量は、対象とする波長と画素の面積にわたって積分する。このような光電素子を使ってカラー画像を撮影するためには、光電素子ごとに異なる波長の光が入射するように、色フィルターを通せばよい。普及型のデジタルカメラでは、各光電素子の前に赤、青、緑の波長帯を通過させ、補色光をブロックするフィルターをコーティングして画素ごとに異なる色の感度を上げている。よく用いられるのは、 2×2 の格子で対角に赤と青、逆の対角に 2 つの緑に対応して光をとらえる画素を置くようにしている。この並びをベイヤー (Bayer) 符号化と呼ぶ。カメラのイメージングの回路では、ベイヤー符号化されたパターンから、それぞれの色の画素パターンを読みだして、色が抜けている画素を補間することによって、R,G,B の 3 チャンネルからなる完全なカラー画像を再構成する (図 2.8)。これを一般的にはカラーデモザイキング (color de-mosaicing) と呼ぶ。

ベイヤー符号化は、フィルターを使うことで、単板の撮像素子でカラー画像のイメージングを行うため安価であるが、画素値を補間するためセンサの解像度を十分に活用できないという弱点がある。そこで、高品質を目指すカメラでは、RGB の各チャンネルごとに撮像素子 (3CCD, 3CMOS) を用いて、合計 3 枚の撮像素子を使って解像度の高い画像のセンシングを行っている。このとき、レンズの後段に波長ごとの屈折率の違いを利用した分光プリズムを置いて、各チャンネルの画像を目的の撮像素子まで誘導する光回路が使われている。

撮像素子で得られた色信号はコンポジット映像信号 (NTSC, PAL, SECAM) やコンポーネント映像信号に直接変換されることもある。コンポジット映像信号はアナログ放送時代の標準方式であり、現デジタル放送時代では、RGB の各色成分を独立に記録する R:G:B=4:4:4 や、差分輝度信号を使って信号圧縮する Y:Cr:Cb=4:2:2 のコンポーネント映像信号が用いられる。

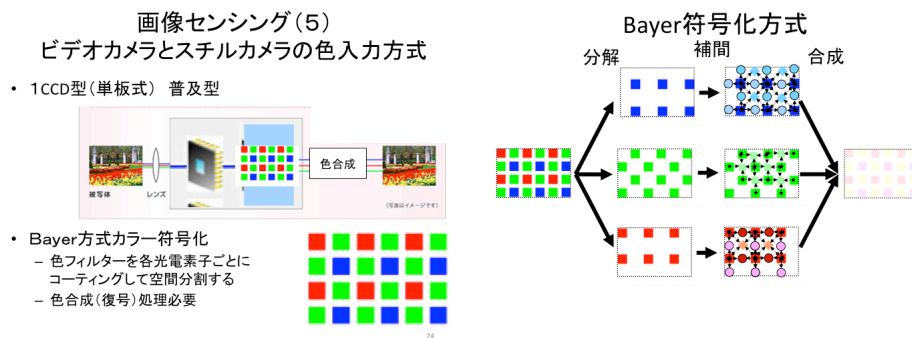


図 2.8 ベイヤー符号化

2.3.2 リモートセンシング

人工衛星や航空機を使った気象・地理・資源探査に用いられるリモートセンシング (remote sensing) では、イメージングの技術が直接応用されている。衛星軌道からの観測の際には、太陽光の反射光成分である可視光だけで

なく、熱放射による可視光や波長の長いマイクロ波などの電磁波を観測することで、大気に影響されない地表の観測が可能である。リモートセンシング人工衛星には、8 個程度のマルチバンド（複数波長帯）を使ったセンサが積まれ、数 100nm から数 $10\mu\text{m}$ までの波長の電磁波をカバーしている。

我々が一般に気象衛星の画像としてみている画像は、必ずしも人が宇宙から地球を見たときの色彩ではなく、得られた多チャンネルの画像から気象が分かり易い色割り付けをして可視化していることがある。

2.4 画像表示装置

カラー画像のイメージングすなわち入力、色成分に基づいて行われたように、スクリーンプロジェクタやフラットパネルディスプレイなどの電子的な画像表示装置をはじめ、カラー印刷などの画像の表示もまた原色の合成によって行われる。

近年のコンピュータディスプレイは、LED ディスプレイとプラズマディスプレイが主流である。発色は、1 つの画素を 3 分割した発光素子で形成し、それぞれの素子の前に RGB の 3 原色それぞれを通過させるフィルターをコーティングしてある。そうしておいて、各素子の発光強度を調整して、空間的に任意の色合成をする。人間の視覚の空間分解能があまり高いことを利用して、網膜で加法混色をしていると考えてよい。現在は、カラーフィルターの印刷技術が著しく高度化したためこの原理による単板型の表示装置が主流となっている（図 2.9）。

それに対して、プロジェクタの初期モデルには、3 つの異なる光源からの像をスクリーン上で加法混色する装置も存在したが、精密な位置合わせが必要であり、現在はあまり用いられない。

LED パネルが発明される以前には、CRT(Cathode Ray Tube) 型のディスプレイ（ブラウン管とも呼ばれる）がテレビ受像機の発明以来 80 年以上使われてきた。これは、電子銃と呼ばれる陰極線を発出する装置をつかって、ディスプレイの表面の内側に塗布してある蛍光体に電子を走査して当てて発光させる仕組みである。蛍光体の手前にシャドウマスクと呼ぶ小穴の空いた金属の薄膜を置き、穴を通過する電子線が RGB の各色蛍光体に当たるようにして色の表示を行っている。

液晶ディスプレイ(Liquid Crystal Display)の構造

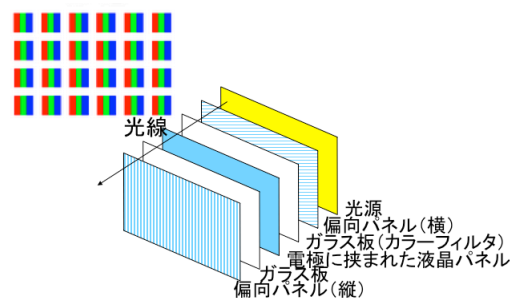


図 2.9 LCD ディスプレイの構造

2.5 立体知覚と 3D 表現

人間の視覚が立体を知覚（立体視）する手掛かりには、様々なものがある。典型的なものは両眼視差と呼ばれるもので、右眼と左眼では対象点の奥行きの違いが網膜上での結像する（相対的な）位置の違いとなり、それを脳が受け取って奥行きの違いと認識する。図柄に意味のないランダムドットパターンで平面上に対応する点の視差を合成しても立体を知覚するのは、このためである。そのほか、運動視差、焦点調節、輻輳角、陰影、オクルージョン（物体輪郭による隠蔽）をはじめ、ものの色や大きさやなどでも立体を知覚する。

画像処理における立体視（すなわち奥行きの計測や形状認識）と立体表示の多くは、この人間の特性を利用している。

画像処理における立体視では、両眼視差や陰影、物体輪郭などの特徴が用いられ、それぞれ、ステレオ画像処理、陰影からの形状抽出 (shape from shading, photometric stereo, 動きからの立体再構成、線画の理解などの手法がある。本書では、ステレオ画像処理について後述する。ステレオ画像処理のための画像入力は、2 台以上のカメラによる撮影を行う。右眼用と左眼用の写真を、両眼の間隔に合わせてレンズの光軸を平行に置いた 2 台のカメラで同時に撮影する。

一方、立体的な表示をするための方法として、コンピュータグラフィクス技術を使った画像の陰影表現や遠近感による立体感の付与、インタラクティブな操作による運動視差をつかった立体感の付与、ステレオペア画像を提示して両眼視差を与えることによる立体表示、さらに、光の波の位相情報も含めて 3 次元像を直接記録して再生するホログラフィーがある。

立体写真は両眼視差の効果を使って、ステレオペア画像を両眼でみるものである。2 枚の写真までの距離と違う奥行き位置に結像する必要があるが、裸眼の立体視には多少の訓練を要するが、レンズを使うと簡単に立体視できる。3D（立体）映画と呼ばれるものも両眼視差の効果を使っており、視聴者はメガネをかけて、左右の眼に別の映像が映るように様々な工夫をして立体視を可能にしている。古くは、アナグリフという赤と青のフィルターを両眼にそれぞれ付けて、波長分割する手法がとられた。そのほか、偏光フィルターを映像プロジェクタとメガネに使って偏光分割する方法、両眼用の映像を交互に表示し、そのタイミングと同期して左右のメガネの液晶シャッターの開閉を行うという時間分割する方式などがある。3D 映画館は、メガネの価格が安くでき、カラー画像が上映できる偏光フィルター方式が多用されているが、3D（立体）テレビは、明度を重視した液晶シャッター式（フレームシーケンシャル方式）が多い。また、ディスプレイ面に微少レンズ（例：レンチキュラーレンズ）を置くことで左右の眼に違う映像が見えるようにしたパララックスバリア方式では、メガネが必要ない。その際多くの視点画像を用意しておくと、立体視できる視点位置範囲が広がりかつ運動視差も提示できるため効果が高い。

演習問題

- (1) スネルの法則を使って、薄レンズ方程式を導出しなさい。ただし、レンズの曲率半径 R 、屈折率 n とする。
ヒント：薄レンズの中心を通る光は直進する。
- (2) ビデオカメラの色温度フィルターは何のためにあるのか説明しなさい。
- (3) コンピュータディスプレイの 3 原色からでは全ての可視光が再現できないことを説明しなさい。
- (4) 立体テレビを見て人間が立体を認識する原理と、そのためのディスプレイの構造を説明しなさい。

3

デジタル画像の基礎と基本演算

第2章では、レンズをとおして実世界をセンサーに投影して画像化する仕組みを学んだ。しかしこれは連続空間での変換であり、コンピュータ処理ができるようにするためには、画像のデジタル化が必要である。そこでデジタル空間表現での処理をこれから学ぶ。この章では、まず画像のデジタル化を数学的表現を使って説明する。今日ではデジカメを使えば目の前の実世界を映したデジタル画像を簡単に入手できるが、数学的表現を理解しておくことでデジタル画像処理の深い理解が可能である。さらに、画像処理の基本演算である、点演算、局所演算などを原理とアルゴリズムの組で示す。最後に、様々なパターン認識で使われる特徴量の計算方法について基本的なものを紹介する。

3.1 標本化と量子化

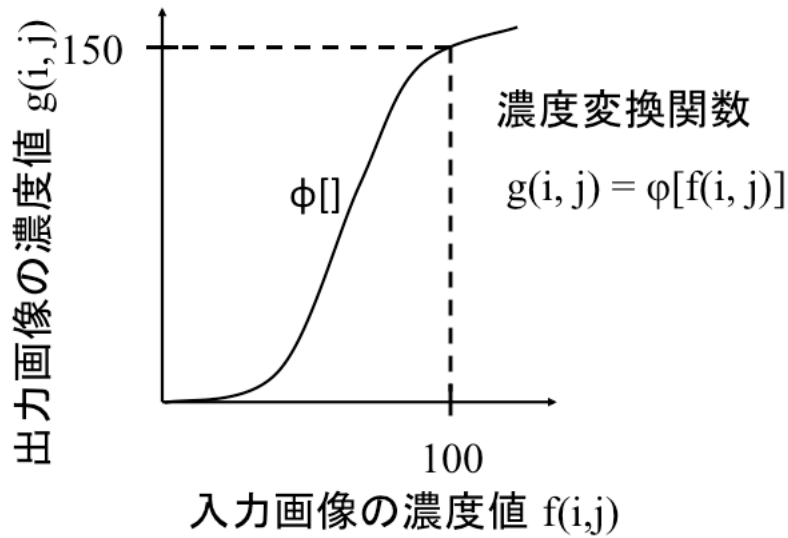
デジタル画像は1章で学んだように、2次元の離散的な格子点上の輝度値の分布 $z = f(i, j)$ として表現できる。2次元連続区間の点座標を格子点に離散化する処理を**標本化 (sampling)** といい、連続した画像の濃淡値 z を離散化して画素の値とする処理を**量子化 (quantization)** と呼び、これらをまとめて**画像のデジタル化 (digitization)** と呼ぶ。

3.1.1 標本化定理

標本化は、連続画像を適当な間隔で矩形（正方形）に区切ったような形になり、この矩形を画素と呼ぶことはすでに学んだ。画像は、画素を単位として縦横の画素数により大きさを定義する。この大きさを解像度 (resolution) と呼ぶ。例えば、横 M 画素、縦 N 画素の画像の解像度は $M \times N$ 画素となる。ディスプレイにおけるいろいろな大きさの画像の解像度と呼び名をを図 3.1 に示す。デジカメなどの画像入力装置の解像度は、CMOS や CCD などの製造技術により決まるセンサの物理的大きさや回路の密度に依存しており、メーカーは解像度の高さを競っているが、どこまで大きければ十分なのだろう。解像度が高いデジタル画像は現実をより精密に写すことがわかっているが、データ量が膨大になっていく。標本化をする際の間隔を決める基準はないのだろうか。ここに、この間隔を定量的に示唆する標本化定理 (サンプリング定理, sampling theorem) と呼ぶ優れた命題がある。

標本化定理 準備: 空間変数 x, y , 空間周波数 u, v として、2次元信号を関数 $f(x, y)$ で表す。 $f(x, y)$ のフーリエ変換 $F(u, v)$ は

$$F(u, v) = \mathcal{F}[f(x, y)]$$



34

図 3.1 画像の解像度

と表現できる. $f(x, y)$ の信号が x, y それぞれの次元についてある周波数 U, V 以上の帯域がカットされているとき, すなわち

$$F(u, v) = 0 \quad (|u| > U \quad \text{or} \quad |v| > V)$$

であれば, 原信号 $f(x, y)$ は, 離散的な点での f の値,

$$f\left(\frac{l}{2U}, \frac{m}{2V}\right), \quad (\{l, m\} \in \mathcal{Z})$$

をもとに, 次式で表すことが出来る.

$$f(x, y) = \sum_{l=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f\left(\frac{l}{2U}, \frac{m}{2V}\right) \frac{\sin \pi(2Ux - l)}{\pi(2Ux - l)} \frac{\sin \pi(2Vy - m)}{\pi(2Vy - m)} \quad (3.1)$$

この定理の意味するところは, サンプリング周波数 $(2U, 2V)$ で標本化された信号 $f(\frac{l}{2U}, \frac{m}{2V})$ を sinc 関数で畳み込みをすることで, 原信号が復元できるというものである¹. ここで, sinc 関数は, 次の形の関数である.

$$\text{sinc}(x) = \frac{\sin x}{x}$$

標本化定理は, 原信号の周波数がある帯域以上でカット (帯域制限) されているという条件により成立する. これは, 時空間での標本化手順がフーリエ変換した周波数空間においては信号の移動と複製で表されていることから, 制限された帯域に限って周波数空間信号をフーリエ逆変換して原信号が求められるという性質に依る. ここで, 周波数空間の移動量がサンプリング周波数により $(2U, 2V)$ であるため, 原信号が $(|U|, |V|)$ 以上の周波数成

¹ すべての信号が必要ではないことを覚えておくように

分を含むと信号の折り返しが生じて正しく復元できないことになる。この現象を折り返し雑音またはエリアシング (aliasing) と呼ぶ。エリアシングは高周波数成分を含むエッジ周辺で発生しやすく、一般的にはモアレパターンとなって観察される。

3.1.2 量子化

標本化が空間的な離散化であるのに対し、量子化は輝度値の離散化であり、濃淡の階調数を決める手続きとなる。階調数が少ないと、たとえ解像度が高くても表示したときに荒い画像として目に映る。従ってなるべく多い階調数で量子化したい反面、多すぎる階調数は冗長でありデータ量が増えるだけである。

残念ながら、標本化定理に相当するような、階調数を決めるための明快な定理はない。1画素の濃淡に1バイト (=256 階調) を割り当てるといった割り切りで扱われているのが実情である。カラー画像では各色1バイトが割り当てられて、違和感なく表現できているが、一部の高性能な画像入出力装置では、1画素の各色を12bitで量子化している場合が見受けられる。

量子化においては、階調数のほかに、階調の区分の決め方も考慮する必要がある。画像センサの特性と人間の視覚特性を比べると、必ずしも画像センサの感度（光電素子の電圧）を線形に均等に割り振ればよいということはない。例えば明るさに対する人間の感度（感覚量）は物理信号に対して対数的であるため、それを考慮した非線形の量子化が有効である。また、近年 HDR(High Dynamic Range) 画像と呼ばれる画像表現が注目されている。

3.1.3 画面走査と画素値の参照

3.2 基本的な画像処理

本章では基本的な画像処理の数々を習う。基本処理には点演算、局所演算、および画像間演算がある。これらの基本処理は、輝度変換、ノイズ除去、エッジ強調、マスク処理などの機能があり、それは、人間にとって画像を見やすくしたり、画像中の物体認識のための特徴抽出の前処理として用いられる。

3.2.1 点演算

点演算は、画像 $f(i, j)$ の1画素の値を入力とし、同じ場所の画素の値 $g(i, j)$ を出力とする単項演算 ψ を画像全体に渡って行う処理である。

$$f(i, j) \xrightarrow{\psi} g(i, j); \text{for all } (i, j)$$

または、

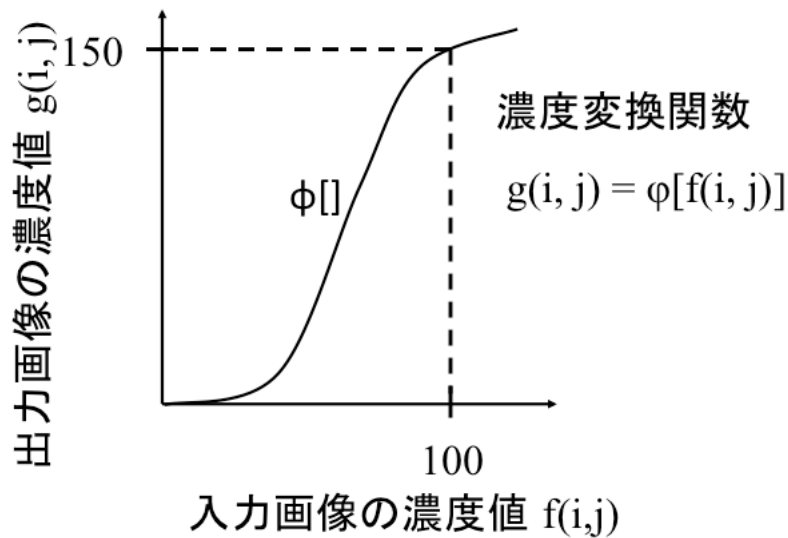
$$g(i, j) = \psi[f(i, j)]$$

と表す。以降、for all (i, j) は、特に断らない限り省略するが、処理は画面全体にわたって行うことを表す。

(1) **輝度変換** 点演算では階調補正を目的とした輝度変換がよく用いられる。輝度変換は、人間にとって画像を見やすくすることが第1の用途である。たとえば、入力画像のある輝度値区間（いま、 $[a, b]$ としよう）に画像中の有益な情報が集まっているとする。この区間 $[a, b]$ の輝度を伸長し、それ以外を圧縮することによって、対象区間の階調数の幅（ダイナミックレンジという）を広げることができる。このため、輝度変換は階調補正ともよばれる。すなわち、図 3.2 のような輝度変換関数 ψ_1 を用意すると区間 $[a, b]$ が区間 $[a', b']$ ($(b' - a') > (b - a)$) に伸長され、 $b - a$ 個の階調を $b' - a'$ 個に増やすことができる。このようにダイナミックレンジを調整することは、後段の画像処理にも有効となる²。例えば、伸長した範囲でエッジ検出すると細かい強度の違いが強調され、弱い

² 輝度変換関数はトーンカーブともよばれる。

エッジ特徴を抽出できる．(要：図 3.2 の変更)



34

図 3.2 輝度変換

輝度変換は，伸長する輝度値区間の選び方と，変換関数の線形性によっていくつかの輝度変換関数が定義される．図 3.3 に，異なる輝度変換関数 ψ に対するいくつかの階調補正の例を示す．写真を撮影する際に全体の光量が不十分なために，ダイナミックレンジが狭くコントラスト（強弱）が弱い画像が得られたときにも輝度変換は用いられる．単純な方法は前述の窓処理型の輝度変換関数が用いられる．しかし，輝度ヒストグラムにおいて明部と暗部にそれぞれピークをもつような写真では，この手法ではバランスが悪い．そこで，限られたビット数で，ダイナミックレンジをバランスよく最大限に広げる輝度変換関数を自動的に計算する手法として，ヒストグラム平坦化処理がある．これは，画像の輝度ヒストグラムをとったときに，分布が（理想として）一様分布になるように輝度ごとの画素数を考慮しながら輝度変換する方法である．また，一枚の画像中でスポットライトが当たっているような図柄の場合には，その領域は輝度の高い範囲で多くのダイナミックレンジを必要とし，また，一方の暗い場所では輝度の低い範囲でなるべく多くのダイナミックレンジを使えるようにする必要がある．そこで，HDR においては画面一様の輝度変換をするのではなく，場所ごとに最大のダイナミックレンジを使えるような変換関数を使う．

(2) 画像の 2 値化 輝度変換のうち，しきい値処理 (thresholding) は，入力画像の輝度値としきい値との大小関係を判定して画像を 2 値化する処理である．画像処理において特徴抽出をする際に多様される．しきい値処理は次式となる．

$$g(i, j) = \begin{cases} 1 & \text{if } f(i, j) \geq T \\ 0 & \text{else} \end{cases} \quad (3.2)$$

ここで， T はしきい値 (threshold) である．

しきい値処理において，しきい値の決定は重要な問題である．2 値化する特徴を抽出する基準が明確な場合には，その値を使えばよい．例えば，後段で説明するエッジ検出では，望ましいエッジ強度を指定すればよい．しかし，

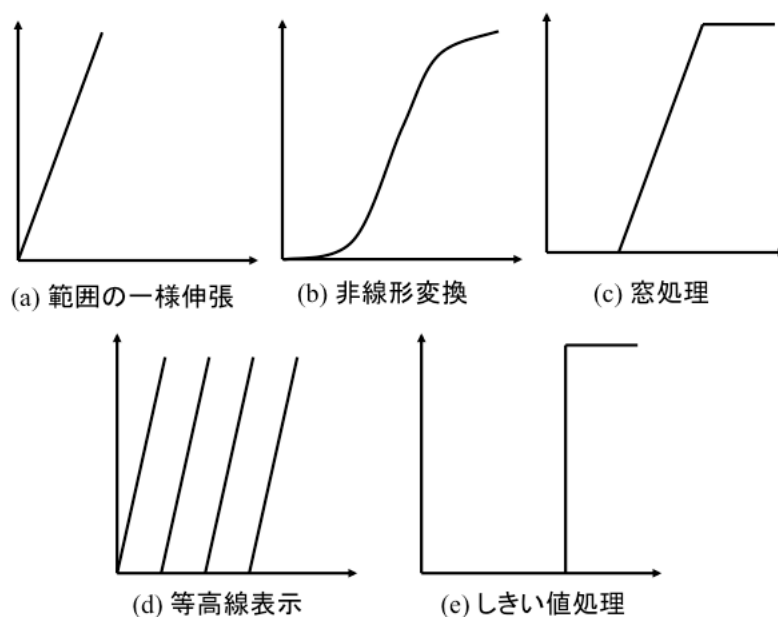


図 3.3 輝度変換関数の例：(1) 範囲の一樣伸張，(2) 非線形変換，(3) 窓処理，(4) しきい値処理

文書画像のような濃淡画像から文字領域を取り出す場合には、画像全体の輝度値分布から求めることが行われる。以下、しきい値決定法として、パーセンタイル法、モード法、および判別分析法の3つを示す(図 3.4)。

パーセンタイル法 パーセンタイル法(P-タイル法)は、輝度ヒストグラムから得られる確率密度分布を用いて、一定の累積確率値(パーセント, %)を与えて、その値を与えるビンの輝度値をしきい値 T_p とする方法である。パーセンタイル法はキーボードのキートップの文字印刷の品質検査やプリント基板のエッチングの品質検査など、良品であれば、対象となる文字や配線パターンの正しい面積割合がわかっているような問題に適している。

モード法 モード法は、輝度ヒストグラムが双峰性分布をしているときに、2つの山の谷間となるビンの輝度値をしきい値 T_m とする方法である。画像内の前景となる「図」と背景となる「地」の輝度値分布が双峰のそれぞれの山を形成するような場合に、よい結果が得られる。ただし、ヒストグラムのビンの幅を変更するとヒストグラムの形状が変化するため、適切なビン幅の設定が必要である。また、実際のヒストグラムは小さな山や谷があるため、安定性が低いという弱点がある。

判別分析法 判別分析法は、モード法と同じように、図と地の輝度値分布が異なることを利用している。図と地のそれぞれが、異なる平均値と分散の輝度ヒストグラムをもつ正規分布からなる混合正規分布を仮定して、2つのクラスに分割した際の分離度を計算し、それが最大化される輝度値をしきい値 T_d とする方法である。安定した値を得られる方法であるが、上記の仮定に合わない画像に対してはうまくいく保証がない。しきい値 T_d は、クラス間分散-クラス内分散比

$$F(t) = \frac{\sigma_B^2(t)}{\sigma_I^2(t)}$$

を最大にする $t_{max} = T_d$ を選択することで得られる．ここで、仮のしきい値 t で分割したときのクラス 1, クラス 2 のそれぞれの分布を $N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)$ とすると、クラス間分散 $\sigma_B^2(t)$ 、クラス内分散 $\sigma_I^2(t)$ は、それぞれ、

$$\sigma_B^2(t) = n_1 n_2 (\mu_1 - \mu_2)^2, \sigma_I^2(t) = n(n_1 \sigma_1^2 + n_2 \sigma_2^2)$$

で表される．ここで、 n は画像全体の画素数、 n_1 はクラス 1 に分類される画素数、 n_2 はクラス 2 に分類される画素数である．この比 $F(t)$ を分離度と呼び、これを最大化する t が求める解である．

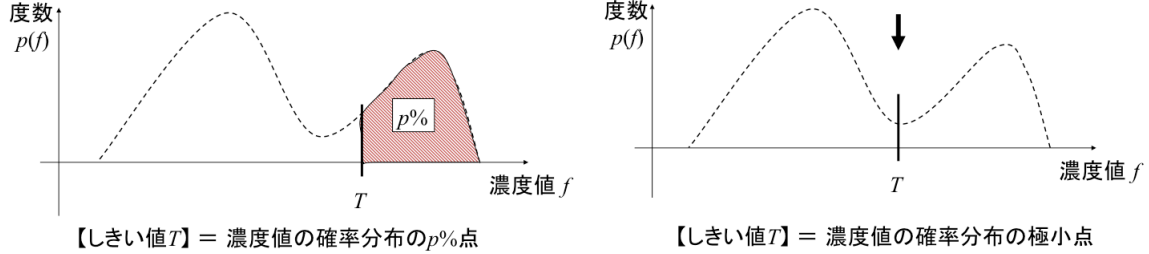


図 3.4 2 値化手法の例：(1) パーセンタイル法，(2) モード法，(3) 判別分析法

3.2.2 画像間演算

2 画像の同じ位置の画素間点演算による画像間演算も定義できる．

(1) 四則演算 2つの画像 $f_1(i, j), f_2(i, j)$ の画像間演算における四則演算は次式のように記述し、すべての (i, j) についてそれぞれ演算する．

$$g(i, j) = f_1(i, j) + f_2(i, j) \quad (3.3)$$

$$g(i, j) = f_1(i, j) - f_2(i, j) \quad (3.4)$$

$$g(i, j) = f_1(i, j) \times f_2(i, j) \quad (3.5)$$

$$g(i, j) = f_1(i, j) / f_2(i, j) \text{ (ただし } f_2(i, j) \neq 0 \text{)} \quad (3.6)$$

3.2.3 局所演算

点演算が単項演算であるのに対し、局所演算は複数の画素の値を入力として、1つの画素の値を出力とする演算

である。すなわち、画像 $f(i, j)$ の注目画素 (i, j) に対し、その近傍の画素集合 $\{(p, q); (p, q) \in \eta(i, j)\}$ における各値 $f(p, q)$ の組を入力とし、同じ場所の画素の値 $g(i, j)$ を出力とする多項演算 ϕ を画像全体に渡って行う処理である。

$$f(p, q); [\{(p, q), (p, q) \in \eta(i, j)\}] \xrightarrow{\phi} g(i, j); \text{for all } (i, j)$$

または全ての画素 (i, j) に関して次の処理を行う。

$$g(i, j) = \phi[f(p, q); \text{for all } (p, q) \in \eta(i, j)]$$

局所演算は、おもにエッジ検出と平滑化に用いられる。また、それぞれのアルゴリズムは線形演算または非線形演算によって実現されている。局所演算は、画像と局所フィルタのたたみ込み演算である。このような処理は、画像中で局所フィルタのパターンが存在するところで強く反応するので、局所フィルタ形状のパターン抽出演算と見られることもできる。局所フィルタがエッジ形状であればエッジ抽出演算を、ガウス分布形状であれば、ガウス平滑化フィルタ演算とあなる。

たとえば、線形フィルタは、重み $\omega_{l,m}$ を使って、

$$g(i, j) = \sum_{(l,m) \in \eta(0,0)} \omega_{l,m} f(i+l, j+m) \quad (3.7)$$

と表すことができる。ここで、 ω の引数 (l, m) は中心画素 $(0, 0)$ とそれに対する近傍 $\{(p, q); (p, q) \in \eta(0, 0)\}$ の各画素である。例えば、8近傍の場合には、自身も含めた9画素 $\{(l, m)\} = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$ に対し、それぞれの重みが $\omega_{-1,-1}, \omega_{-1,0}, \omega_{-1,1}, \omega_{0,-1}, \omega_{0,0}, \omega_{0,1}, \omega_{1,-1}, \omega_{1,0}, \omega_{1,1}$ となる。これは、線形局所演算オペレータの重みとして図式化でき図 3.5 のように表す。この表記は、局所演算の重みを示すのによく用いられる。中心画素の値を p_0 、そして上から時計回りの位置の画素の値を p_1, p_2, \dots, p_8 で表し、各位置の重みを $\omega_1, \omega_2, \dots, \omega_8$ とすると $g(i, j) = \omega_0 p_0 + \omega_1 p_1 + \dots + \omega_8 p_8$ となる。

畳み込み演算としての局所演算 局所演算は、入力画像を $f(i, j)$ 、局所演算フィルタを関数 $h(p, q)$ で表すとき、次式の畳み込み演算で表すことができる。

$$g(i, j) = f(i, j) * h(p, q)$$

(1) **エッジ検出処理** エッジとは、画像中の物体の輪郭だったり、濃淡模様の輪郭を表す図形特徴である。図 3.6 に示すように、ステップエッジ、ランプエッジ、ルーフエッジ、リッジなど様々なエッジがある。エッジ検出の原理は微分演算による。例えば前述のステップエッジを 1 次元信号として見れば、その 1 階微分と 2 階微分は、図 3.7 のようになり、エッジ部のみ信号が強調される。画像は 2 次元信号であるので、各成分の偏微分を計算する。

差分フィルタ 連続画像 $f(x, y)$ の x に関する 1 階偏微分は

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

そこで、 Δx を 1 画素分と考えると、デジタル画像 $f(i, j)$ に対する i 方向の 1 階差分の演算が定義できる。

$$\frac{\partial f}{\partial x} \cong g_{i,j} = f_{i+1,j} - f_{i,j}$$

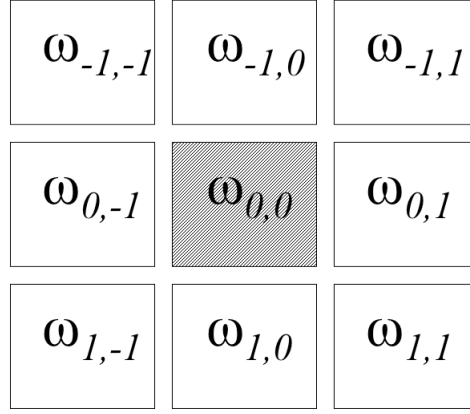


図 3.5 線形局所演算オペレータの重み

図 3.6 様々なエッジ

同様に, j 方向の 1 階差分は次式で与えられる. j 方向の 1 階差分は i 方向に延びるエッジを強調する.

$$\frac{\partial f}{\partial y} \cong g_{i,j} = f_{i,j+1} - f_{i,j}$$

また, 2 階偏微分から 2 階差分の演算を導出できる. 2 階偏微分が 1 階偏微分をさらに微分したものであるから, 2 階差分は 1 階差分の差分である. すなわち, i 方向の 2 階差分は,

$$\frac{\partial^2 f}{\partial x^2} \cong g_{i,j} = (f_{i+1,j} - f_{i,j}) - (f_{i,j} - f_{i-1,j}) = (f_{i+1,j} - 2f_{i,j} + f_{i-1,j}).$$

同様に, j 方向の 2 階差分は次式となる.

$$\frac{\partial^2 f}{\partial y^2} \cong g_{i,j} = (f_{i,j+1} - f_{i,j}) - (f_{i,j} - f_{i,j-1}) = (f_{i,j+1} - 2f_{i,j} + f_{i,j-1})$$

グラディエント 差分がスカラー量であるのに対し, グラディエント (gradient) はベクトル量で, エッジの傾き (方向) と大きさを表す.

$$\nabla f_{x,y} = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \cong \nabla f_{i,j} = (f_{i+1,j} - f_{i,j}, f_{i,j+1} - f_{i,j})$$

ラプラシアン ラプラシアンは次式のように i 方向および j 方向の 2 階差分の和で定義されスカラー量を与える. そのため, 全方向のエッジ強度を計算することができるため多用される.

$$\begin{aligned} \nabla^2 f_{x,y} &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &\cong \nabla^2 f_{i,j} = f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j} \\ &= (f_{i+1,j} - 2f_{i,j} + f_{i-1,j}) + (f_{i,j+1} - 2f_{i,j} + f_{i,j-1}) \end{aligned}$$

その他の差分フィルタ 上記の原理をもとに, 実用的な画像処理で多く用いられる差分フィルタとして, 重み付き平滑化と 1 階差分を組み合わせた Sobel フィルタ, 一様平滑化と 1 階差分を組み合わせた Prewitt フィルタ, 対角方向の 1 階差分の平方 2 乗和の Roberts フィルタなどがある. ユニークなものとして, 後述する Gauss 平滑化

図 3.7 ステップエッジの1階微分と2階微分

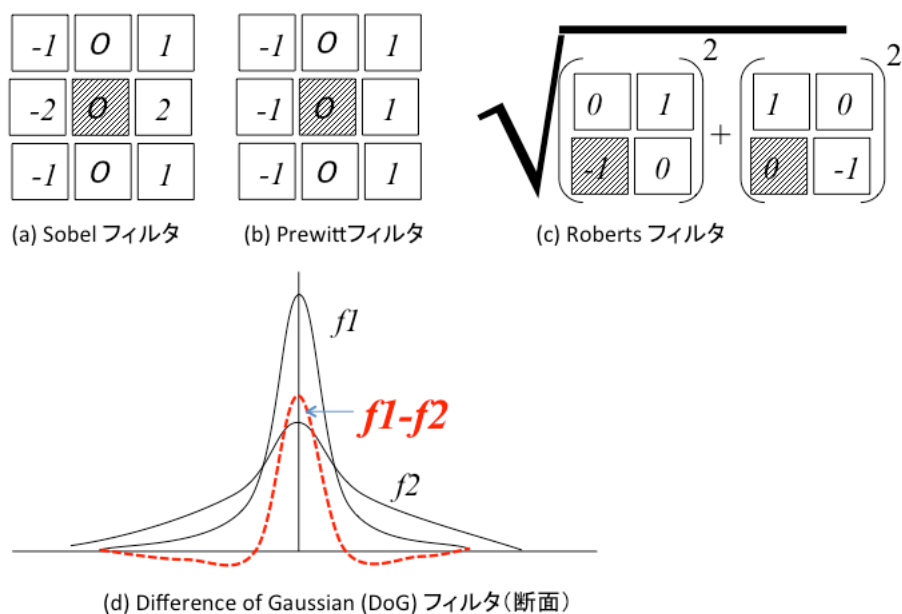


図 3.8 差分フィルタの演算オペレータ: (1) Sobel フィルタ, (2) Prewitt フィルタ, (3) Roberts フィルタ, (4) DoG フィルタ

フィルタにおいて異なるスケール（異なる分散）のフィルタ間の差分をとった Difference of Gaussian (DoG) フィルタがある。DoG は、人間の初期視覚におけるエッジ抽出のモデルといわれる Laplacian of Gaussian (LoG) の良い近似として用いられている。LoG フィルタと DoG フィルタの演算子は以下のようにになっている。

$$LoG = \frac{\delta}{\delta\sigma} G(x, y, \sigma) = \frac{\delta}{\delta\sigma} \left[\frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \right] = -\frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^5} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$\frac{\delta}{\delta\sigma} G(x, y, \sigma) \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{(k-1)\sigma} = DOG$$

ここで、 $G(x, y, \sigma)$ は分散 σ^2 の 2 次元ガウスフィルタを表す。

ゼロ交差 ラプラシアンや LoG(DoG) からエッジを抽出する際に、単純にエッジ強調画像の画素値（またはその絶対値）の大小でエッジを検出するのではなく、ゼロ交差（零交差, zero-crossing）を求める方法がある。ゼロ交差とは、ステップエッジの 2 階微分（図 3.7）をとった際に、微分値が負から正（または正から負）に転換する際に通過する、微分値 0（零）の点のことである。この点を取りだせば、ステップエッジのエッジ部のみをシャープに抽出することができる。Canny エッジ抽出法などで用いられる。

非線形差分フィルタ また、非線形の差分演算を用いた差分フィルタとして、強度の強い方向のみのエッジ強度を計算する方向性 1(2) 階差分フィルタと、近傍内のダイナミックレンジを計算するレンジフィルタなどがある。

- (a) 方向性 1 階差分: $g_{i,j} = \max(f_{i+1,j} - f_{i,j}, f_{i,j+1} - f_{i,j})$
 (b) 方向性 2 階差分: $g_{i,j} = \max(f_{i+1,j} - 2f_{i,j} + f_{i-1,j}, f_{i,j+1} - 2f_{i,j} + f_{i,j-1})$
 (c) レンジフィルタ: $g_{i,j} = \max_{(l,m) \in \eta_{(0,0)}} (f_{i+l,j+m}) - \min_{(l,m) \in \eta_{(0,0)}} (f_{i+l,j+m})$

(2) **平滑化処理** 局所演算のもう一つの重要な用途は平滑化である。信号に含まれているランダム雑音³を除去したり、高周波成分を除去するために用いられる。すなわち、平滑化処理は信号処理の考え方では、低周波通過（ローパス, low-pass）フィルタまたは高周波遮断（ハイカット, high-cut）フィルタにあたる。エッジ検出処理と同様にさまざまな線形フィルタと非線形フィルタがある。線形フィルタは、式 3.7 の近傍 $\eta(l, m)$ の大きさとその重み $\omega_{l,m}$ で定義される。

一様平滑化フィルタ 一様平滑化フィルタの最も簡単なものは、 3×3 の大きさの正方形フィルタで次式のようになる。

$$g(i, j) = 1/C \sum_{l=-1,0,1} \sum_{m=-1,0,1} f(i+l, j+m) \quad (3.8)$$

ここで、定数 $1/C$ は一様重み係数であり、近傍の画素数（この場合、 $C = 3 \times 3 = 9$ ）で決まる。 3×3 フィルタの大きさを 5×5 , 7×7 とすると、さらに平滑化の効果が強まる。

ガウス平滑化フィルタ フィルタの形を 2 次元正規関数にするとガウス平滑化フィルタを構築できる。正規関数 $G(x, y, \sigma)$ の値を毎回計算するのは無駄なので、あらかじめ計算してテーブルに格納しておく方法が用いられる。分散 σ^2 の値により、平滑化のなめらかさを選択する。近傍の範囲は、半径 2σ から 3σ の円内を考えれば十分である。下記は、半径 2σ の例である。

$$g(i, j) = \sum_{-4 \leq l^2 + m^2 \leq 4} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\sigma^2 l^2 + \sigma^2 m^2}{2\sigma^2}\right) f(i+l, j+m) \quad (3.9)$$

$$= \sum_{-4 \leq l^2 + m^2 \leq 4} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{l^2 + m^2}{2}\right) f(i+l, j+m) \quad (3.10)$$

3.2.4 その他の局所演算

そのほかの局所演算として非線形演算を考えると多様な処理を考えられる。

最大値, 最小値, メディアンフィルタ 近傍画素中の最大値, 最小値, あるいは中央値（メディアン, median）を求める演算である。

$$g(i, j) = \max[f(p, q)]; \text{ for all } (p, q) \in \eta(i, j)$$

$$g(i, j) = \min[f(p, q)]; \text{ for all } (p, q) \in \eta(i, j)$$

$$g(i, j) = \text{median}[f(p, q)]; \text{ for all } (p, q) \in \eta(i, j)$$

ここで通常、近傍の大きさは 3×3 , 5×5 などをとる。たとえば、 3×3 近傍では下記となる。

$$g(i, j) = \max[f(i-1, j-1), f(i-1, j), f(i-1, j+1), f(i, j-1), f(i, j), f(i, j+1), f(i+1, j-1), f(i+1, j), f(i+1, j+1)]$$

入力画像が 2 値画像の場合は、最大値フィルタは図形の膨張（拡張, dilation）、最小値フィルタは図形を縮小（浸食, erosion）させる。この膨張、縮小処理は、画像の形状操作を総称したモルフォロジー（morphology）の基本演算である。モルフォロジーではフィルタの形状を構造化要素と呼ぶ。膨張処理は、構造化要素中心を図形の輪郭に

³ 画像の場合も音声信号と同様に、目的の信号でないものは雑音と呼ぶ。

そして動かしたときの構造化要素が覆う範囲まで図形を膨張させる。一方、縮小処理では、構造化要素を図形の輪郭にそって内側を動かしたときの構造化要素の中心が新しい輪郭となる。膨張処理は図形の穴や窪みを埋め、縮小処理は孤立点を消去したり図形の突起を削る処理を行う。本書では、画像 f に対する膨張演算を $f * D$ 、縮小演算を $f * E$ で表す。膨張と縮小を交互に行う $(f * D) * E$ 、 $(f * E) * D$ の処理はそれぞれ、Close 演算、Open 演算と呼ばれ、前者は対象図形の整形に、後者は複数の図形塊の切断に用いられる。(図例?)

(1) **距離変換** 2値画像に対し縮小演算を順次繰り返すと図形塊は徐々に小さくなって最後は塊は消滅する。消滅した回数を N とする。0 回目から $N - 1$ 回目までの n 回目までの縮小演算の結果を $g_n(i, j)$ とすると、

$$d(i, j) = \sum_{0 \leq n < N} g_n(i, j)$$

で表される。ここで得られた $d(i, j)$ を距離変換画像 (distance transformation image) と呼ぶ⁴。この画像の各画素値は、図形塊の輪郭から当該画素までの距離を示す。最小値フィルタの形状を 4 近傍または 8 近傍のどちらを選ぶかによって、距離変換画像のパターンは異なる。

3.2.5 セグメンテーションのための画像処理

ここまでは、点演算や局所演算など処理方法の観点から基本的な画像処理の手法をみてきた。この節ではこれらの基本演算を使って画像中の図形や物体 (以下、総称して図形) のセグメンテーション (segmentation) のための手法を紹介する。セグメンテーションとは、背景から目的の図形を切り取る処理である。なお、画像処理において切り取るとは、その図形領域の画素値として他と区別できる値を代入する処理になる。たとえば、前述の 2 値化処理は、2 値化によって得られた図形に 1 を、残った背景に 0 を代入することで、目的の図形のセグメンテーションを行ったことになる。セグメンテーションの対象となる図形は、領域と線分である。

(1) **領域セグメントの抽出** 画像中に複数の人物が映っていて、それら人物の顔領域を抽出したいとする。そのためには、まず、各画素の顔らしさを計算し、顔らしさの値が画素に格納された”顔らしさ画像”を求め、次にその画像の 2 値化処理をすることによって目的とする領域が抽出できる。式で表すと、入力画像を $f(i, j)$ 、顔らしさを表す中間画像を $g(i, j)$ 、顔領域のセグメント画像を $h(i, j)$ とすると、顔らしさを求める関数 $\phi()$ をもちいて、 $f(i, j)$ から $g(i, j)$ への変換が得られる。

$$g(i, j) = \phi(f(i, j))$$

次に、 $g(i, j)$ をしきい値 T で 2 値化処理を行って人物領域のセグメント画像 $h(i, j)$ が得られる。なお、 $\phi()$ として、ごく単純な例では、肌領域の有無を顔らしさと考えて、カラー入力画像から色相を求める計算をする関数が考えられる。そして、肌色の区間を抽出する 2 値化を行うと、顔領域画像が得られる。

得られた画像には、複数の顔候補領域と、顔に近いと誤判定された領域が含まれる。そこで、領域ごとにさらに別の顔らしさを判定したり、得られた顔候補領域をつかって次の処理に進む。ここで、「領域ごと」の処理をするために、同じ領域に含まれる画素かそうでないかを区別するために、領域のラベリング (labeling) という手法が用いられる。

領域ラベリング 画像を図と地にわけて、図を構成するひとかたまりの画素の集合を連結領域 (connected region) と呼ぶ。ここでは、値 1 の画素を図、値 0 の画素を地とする。1 つの領域において任意の 2 画素が連結しているときにその画素集合を連結領域と呼ぶ。ここで、画素 A と画素 B が連結する (connected) とは、画素 A から、順

⁴ この距離変換画像は、前述の「各画素が画像平面からの奥行き距離を表す【距離画像】」とは異なる。

に8近傍にある（すなわち、画素の頂点または辺を共有する）画素をたどって画素 B にたどり着ける状況を指す。これを8連結と呼ぶ（4近傍の画素をたどってたどり着く場合は4連結と呼ぶ）び、この連結領域は8連結であるという。図 ?? に、8連結の連結領域の例を示す。

領域ラベリング（以下、単にラベリングと呼ぶ）は、8連結（または4連結）の連結領域をみつけて、連結領域毎に各画素に同じラベルをつけるアルゴリズムである。たとえば、画像中に5つの領域があれば、1つ目の領域の各画素にラベル「1」を、2つめの領域の各画素にラベル「2」を割り当てる（代入する）ことである。

ラベリングは、シード点となる点から連結している画素に順にラベルをつけるため、逐次的な処理で実行可能であり、シード点の関係を記憶するテーブルをつかって、順次操作走査で画像全体を2回走査するだけで実行できる。

領域ラベリングのアルゴリズム 省略

(2) 背景からの図の抽出

グラフカット

(3) 線分セグメントの抽出 デジタル画像における線分とは、連結性を考慮して順序づけた点（点列）のことである。線分は線幅1画素の画素列として定義され、領域の境界、領域の心線（芯線）、画像中のエッジなどが線分として抽出される。線分セグメントも、領域セグメントと同じように8連結または4連結の連結領域である。エッジ強調の結果の画像を2値化処理すると、線分セグメントの候補が得られる。これらをそのまま境界線などとしてもよいが、幅1画素にするためには細線化処理が必要である。

細線化処理 領域セグメントを、その心線と思われる線分セグメントに変換する処理を細線化と呼ぶ。細線化は、領域セグメントの画素を逐次的に背景に置き換える処理をして、その心線を削り出す処理を行う。ここで、得られた線分セグメントの位相的な特徴が変化しないよう、且つ、最終的に線分が、幅1画素で、元の領域の心線らしさを保つことを判断しながら、順番に処理する。例えば、ある画素を削る（値を0にする）と領域が2つに分割される場合は、その画素は削ってはならない。削る順番の手がかりとなるのは、前述の距離変換を行い、領域の各画素が背景との境界からどれくらい奥にあるかを調べ、距離の小さい（境界に近い）画素から削ればよい。

Hough(ハフ) 変換 上記のように画像から点列を求めるボトムアップの処理に対し、直線（線分）の幾何モデルを仮定し、トップダウンに線分を求める手法として Hough 変換がある。Hough(ハフ) 変換は、画像中の線分を構成する候補となる画素群から線分のモデルを仮定して、同一線分に含まれる画素集合を求める手法、すなわち画像からの直線抽出法の一つである。

まず、 x - y 空間と ρ - θ 空間のパラメータを使って、直線 l を $f(x, y; \rho, \theta) = 0$ で表すと、

$$\rho = x \cos \theta + y \sin \theta \quad (3.11)$$

となる。ここで、 ρ は原点から直線 l への距離であり、 θ は直線 l への原点からの垂線の x 軸からの角度である。図 ?? に、パラメータの関係を図示する。

式 3.11 の示すところは次のようである。図 ?? に示すように、 x - y 空間上の1点 $p_a = (x_a, y_a)$ を与えると、その点を通る任意の直線 (l_1, l_2, l_3, \dots) の方程式は

$$\rho = x_a \cos \theta + y_a \sin \theta$$

で表される。このとき、 l_1, l_2, l_3, \dots は、 ρ - θ 空間において、点 (x_a, y_a) を通る1つの曲線を作る。

逆に、 ρ - θ 空間で 1 点 (ρ_b, θ_b) を決めると、その点を通る曲線 (p_1, p_2, p_3, \dots) の方程式を

$$\rho_b = x \cos \theta_b + y \sin \theta_b$$

で表すことができ、 x - y 空間において、それらに対応する点群 (p_1, p_2, p_3, \dots) として描かれる。ここでこの点群の座標 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ は、方程式が示すとおり直線となる。このように直線の式のパラメータに関して、 x - y 空間と ρ - θ 空間は双対の関係になる。

上記の双対の関係を使って、 x - y 空間の点列から直線をみつけるアルゴリズムを構成できる。すなわち、画像上の直線候補の点ごとに、対応する曲線 $\rho = x \cos \theta + y \sin \theta$ を ρ - θ 空間で描く。もしそれらの点が直線を構成していれば、対応する ρ - θ 曲線は 1 点で交わる。ここで、 ρ - θ 空間に対応する画像配列を用意し、画像上に曲線を描画する。曲線を上書きする際に画素の値を 1 だけ加える操作をするようにすると、曲線が集中した点の画素の値は交差した曲線の数に対応して高くなる。すべての x - y 画像中の候補点について ρ - θ 画像で曲線を描画したあとの値の高い画素 (ρ, θ) で直線が抽出できたことになる。

Hough 変換は、直線のみでなく円や楕円の式をパラメータ化することで、円や楕円の抽出にも応用できる。さらに、パラメータ空間を一般化して投票によってパラメータを推定する一般化 Hough 変換もよく用いられている。

3.2.6 画像の幾何学

(1) 画像のセグメンテーション

3.3 特 徴 抽 出

画像を人間が見やすいように強調したり、対象物を抽出したり、あるいはパターン認識のための特徴をとらえる際に、特徴抽出が行われる。特徴の次元によって、本節では点特徴、線特徴、領域特徴に分けて説明する。

3.3.1 点 特 徴

点特徴は、画像中の各点（画素）を特徴となる属性で表現する。たとえば、1 章で述べた、輝度、色情報などのその点のみで決まる属性もあれば、局所処理によって得られる、一次差分、グラディエント、エッジ強度なども点特徴である。最近では、対象となる点の比較的広い近傍の状況から計算される点特徴として、SIFT(scale invariant feature transform) 特徴、KL(Kanade-Lucas) 特徴などがある。

とりわけ SIFT 特徴は、照明影響に不変、回転に不変、拡大縮小に不変という頑健な特徴であり、例えば 2 枚の同じ場所を違う方向からとった写真（ステレオ写真やパノラマ生成のための移動写真）において、SIFT 特徴のマッチングにより特徴点の対応付けをすることができる。SIFT 特徴は計算量が大きいためそれを高速化した SURF(speed-up robust feature) 特徴も提案されている。

3.3.2 線 特 徴

線特徴は、線分の長さや、線分の各点の曲率などの幾何的な属性で、線分を特徴づける。得られた線分セグメントは座標列としてリスト化され、場合によっては、次のチェーンコードなどを使って符号化表現される。また、閉曲線の場合、ある点から曲線にそって進んだ距離 l と原点からのその点の距離 $u(l) = x(l) + iy(l)$ (複素表現, i は虚数単位) を使うと、曲線の形状をフーリエ記述子で表現できる。

(1) チェインコード チェインコード (chain cord) は、線幅 1 のデジタル線分の画素列に対して、画素の並びの方向を符号化して表現する方法である。符号化の量子化数として一般に用いられるのは、 $360/8 = 45$ 度刻

みの 8 方向である。時計の 3 時の方向を”0”とし、反時計回りに,”1”, ”2”, „, ”7”の方向コードが割り当てられる。

3.3.3 領域特徴

これまでならった基本的な演算を用いると、画像の中に領域要素を定義することができる。例えば点特徴量で際だった抽出対象を画像全体をしきい値処理して、ラベリングにより連結領域として取り出すと、個々の領域についてテクスチャの統計的特徴を計算でき、それをもとに領域抽出をすることができる。また、そのような連結領域についての手がかりがないときには、基本的演算の局所演算と類似の方法で対象画像を矩形の領域で走査し、矩形内の統計的特徴やテンプレートマッチングにより画像全体にわたって領域特徴量を計算して、改めて抽出対象を際立たせ領域抽出を行うことができる。

まず、定められた領域内の統計的特徴としての 1 次統計量と 2 次統計量を説明する。次に、抽出したい対象をテンプレートとして与え、そのテンプレートの輝度分布との類似度や差異を計算するテンプレートマッチングとして、SSDA 法 (Sequential Similarity Detection Algorithm, 逐次残差検定法) と相関係数を用いた方法を説明する。さらに、最近では広域差分フィルタに類似した Haar-like 特徴量や局所エッジ勾配を用いた HOG 特徴量などが提案され、顔や人物の抽出に有効であるとされている。

(1) 2 値図形のモーメント特徴量

2 値画像 $f(x, y)$ における図形の $k(= i + j)$ 次モーメント特徴 m_{ij} は次式で表される。(図形は値 1 とする)

$$m_{ij} = \sum_{x, y \in R} x^i y^j f(x, y) \quad (3.12)$$

これを用いて、0 次、1 次、2 次の画像モーメントが計算できる。0 次モーメントは図形の面積 S であり、次式で計算できる。

$$m_{00} = S = \sum_{x, y \in R} f(x, y)$$

1 次モーメントは画像 $f(x, y)$ の重心ベクトル $(C_x, C_y)^t$ であり、次式で計算できる。

$$(C_x, C_y)^t = (m_{10}/m_{00}, m_{01}/m_{00})^t = \frac{1}{m_{00}} \left(\sum_{x, y \in R} x f(x, y), \sum_{x, y \in R} y f(x, y) \right)^t$$

同様に、2 次モーメント量 m_{11}, m_{20}, m_{02} を式 3.12 から求めることができる。これら 2 次モーメント量を用いると、図形の主軸の重心まわりの傾き ϕ を次の方程式の解として計算することができる。

面積、重心、主軸の傾きは、図形を記述する基本的な特徴となる。

$$\tan^2 \phi + \frac{m_{20} - m_{02}}{m_{11}} \tan \phi - 1 = 0$$

(2) 1 次統計量 統計的特徴のうち 1 次統計量である平均 (mean) μ , 分散 (variance) σ^2 , 歪度 (skewness) s , 尖度 (kurtosis) k , エネルギー e , エントロピー h は、任意の対象領域の画像 $f(i, j)$ の輝度ヒストグラムをもとに、ヒストグラムの面積を正規化した確率密度分布 $p(n); (n = 0, 1, \dots, N - 1), \sum_n p(n) = 1.0$ から計算することがで

きる．ただし， N は量子化レベルであり，画像の面積を S とする．

$$\mu = \sum_n p(n) = \frac{1}{S} \sum_{i,j} f(i,j) \quad (3.13)$$

$$\sigma^2 = \sum_n [(n - \mu)^2 p(n)] = \frac{1}{S} \sum_{i,j} [f(i,j) - \mu]^2 \quad (3.14)$$

$$s = \frac{1}{\sigma^3} \sum_n [(n - \mu)^3 p(n)] = \frac{1}{S} \sum_{i,j} [f(i,j) - \mu]^3 \quad (3.15)$$

$$k = \frac{1}{\sigma^4} \sum_n [(n - \mu)^4 p(n)] = \frac{1}{S} \sum_{i,j} [f(i,j) - \mu]^4 \quad (3.16)$$

$$e = \sum_n [p(n)]^2 \quad (3.17)$$

$$h = - \sum_n p(n) \log_2 p(n) \quad (3.18)$$

歪度は輝度分布の非対称性，尖度はとがり具合を表す．ピークエネルギーは，ある濃度値に分布が集中しているときに高くなり，一様分布のときに最小となる．エントロピーはその逆の傾向を示す．

(3) **2 次統計量** 複雑なテクスチャは輝度値のみの統計量である 1 次統計量では区別することが困難である場合がある．その際，2 つの画素の値の関係を記述できる 2 次統計量が用いられる．距離 (u, v) だけ離れた画素 (x, y) と画素 $(x + u, y + v)$ それぞれ輝度値 a, b であるとき，画素対の同時生起確率を

$$p_{u,v}(a, b) = \frac{N(a, b)}{S}$$

で表す．ここで， $N(a, b)$ は相対位置 (u, v) で，値が a, b である画素対の数， S は対象領域の面積である．

この統計量を用いて，1 次統計量と同様に，共分散 (co-variance) $\sigma(u, v)$ ，慣性 (moment) $M(u, v)$ ，絶対値 (absolute) $Abs(u, v)$ ，エネルギー (energy) $E(u, v)$ ，エントロピー (entropy) $H(u, v)$ ，自己相関関数 (autocorrelation) $R(u, v)$ などが以下のように得られる．

$$\sigma(u, v) = \sum_a \sum_b (a - \bar{a})(b - \bar{b}) p_{u,v}(a, b) \quad (3.19)$$

$$M(u, v) = \sum_a \sum_b (a - b)^2 p_{u,v}(a, b) \quad (3.20)$$

$$Abs(u, v) = \sum_a \sum_b |a - b| p_{u,v}(a, b) \quad (3.21)$$

$$E(u, v) = \sum_a \sum_b p_{u,v}(a, b)^2 \quad (3.22)$$

$$H(u, v) = - \sum_a \sum_b p_{u,v}(a, b) \log_2 p_{u,v}(a, b) \quad (3.23)$$

$$R(u, v) = \frac{1}{S} \sum_{x,y} (f(x, y) f(x + u, y + v)) \quad (3.24)$$

(4) テンプレートマッチング

テンプレートマッチング (template matching) は，あらかじめ目的とする（抽出したい）対象パターンをテンプレートパターンとして用意し画像に順次当てはめ，同じパターンかどうかを調べる処理である．いってみれば，テンプレートパターンをある種のフィルタとして，探索対象の画像全体を順次走査しながら局所的に画像間演算をする処理である．一般的には，大きさ $I \times J$ の画像 $f(i, j)$ に対して，大きさ $L \times M$ ($L < I, M < J$) のテンプレ

レートパターン $t(l, m)$ を畳み込み演算する処理として次式で記述できる.

$$f(i, j) \otimes t(l, m)$$

ここで, \otimes は積和, 差分のほか相関や距離関数などの任意の畳み込み演算子である. 以下, 具体的な関数の例で説明する.

ベクトル間距離関数を使った SSDA 法 テンプレートパターンで対象画像を順次走査していくと, テンプレートの原点 (便宜上左上とする) が置かれた位置 (i, j) に対して画像とテンプレートが重なった各画素の二乗誤差和をマッチング距離 $d_{l_2}(i, j)$ とする次式で計算できる.

$$d_{l_2}(i, j) = \sqrt{\sum_l \sum_m |(f(i+l, j+m) - t(l, m))|^2} \quad (3.25)$$

この計算を全ての i, j について行えば, 得られたマッチング距離画像 $d_{l_2}(i, j)$ のうち値の小さい画素に対象とするテンプレートのマッチング距離の計算は, テンプレートを置いた位置で画像から大きさ $L \times M$ の矩形領域を切り出しそれを列ベクトル \hat{f} と見なして, ベクトル間の L_2 距離 $\|\hat{f} - t\|$ を各位置で計算していると同値と見なすこともできる.

そこで, この距離計算を L_2 距離ではなく L_1 距離を使うと次式のマッチング距離画像 $d_{l_1}(i, j)$ を求めるアルゴリズムを得る.

$$d_{l_1}(i, j) = \sum_l \sum_m |(f(i+l, j+m) - t(l, m))| \quad (3.26)$$

これは, 平方や平方根の計算がないため処理を高速化できるという利点がある.

相関係数法 テンプレートマッチングにおいて, テンプレートパターンと対象画像との類似性の尺度として, 相関係数 (共分散, 相互相関ともいう) を用いる手法である. 次式の相関係数 $R(i, j)$ ($0 \leq R(i, j) \leq 1.0$) が 1.0 に近いほど類似していると判定される.

$$R(i, j) = \frac{\sum_l \sum_m \{f(i-l, j-m) - \bar{f}\} \{t(i, m) - \bar{t}\}}{\sqrt{\sum_l \sum_m \{f(i-l, j-m) - \bar{f}\}^2} \sqrt{\sum_l \sum_m \{t(i, m) - \bar{t}\}^2}} \quad (3.27)$$

SSDA 法がベクトル間距離であるならば, 相関係数は 2 ベクトル間の余弦距離に当たると見なすことができる.

ヒストグラム交差法 上記の方法は, テンプレートと対象画像のパターンの画素値配置の類似性を評価しているが, パターンの画素値分布であるヒストグラムを計算して, ヒストグラム間の距離を評価する方法がある. ヒストグラムの交差面積割合を計測するため, ヒストグラム交差法と呼ばれる. 対象画像 $f(i, j)$ とテンプレートパターン $t(l, m)$ の 2 つの正規化ヒストグラムをそれぞれ $h_f(n), h_t(n)$ で表すとき, ヒストグラム交差は次式で定義される.

$$I(i, j) = \sum_{n=1}^N \min(h_f(n), h_t(n)) \quad (3.28)$$

ただし, $h_f(n)$ は, テンプレート $t(i, j)$ の順次走査で, テンプレートが対象画像を覆っている範囲の画素のヒストグラムである. また N は, ヒストグラムのビン (階数) を表す.

SSDA 法, 相関係数法, ヒストグラム交差法の 3 つは, 類似度の計算原理の違いにより, 概ねそれぞれ次のような特性がある.

表 3.1 テンプレートマッチングにおける各手法の特徴

手法	計算量	対明るさ変化	対大きさ変化	パターン表現性
SSDA 法	中	×	○	◎
相関係数法	大	○	○	◎
ヒストグラム交差法	小	△	◎	△

(厳密な比較ではない, 対大きさ変化はピラミッド表現利用などを前提としている)

(5) Haar-like 特徴

(6) HOG 特徴

演 習 問 題

- (1) 標本化定理を証明しなさい.
- (2) ヒストグラム平坦化処理のアルゴリズムを示しなさい.

4

画像の動きと立体構造抽出

4.1 オプティカルフロー

4.1.1 時空間画像

4.2 立体構造の抽出

4.2.1 ステレオ視の幾何学

4.2.2 画像パターン投影ステレオ

4.2.3 動きからの構造再現

factorization

5

画像処理を用いたインタフェース

5.1 ジェスチャの認識

5.2 個人の識別と表情の認識

5.3 拡張現実：AR

ARToolkit

演習問題

- (1)
- (2)
- (3)

文献

- 1) 石井健一郎, 前田, 村瀬洋: “だれでもわかるパターン認識”, オーム社
- 2) Christopher Bishop: “Pattern Recognition and Machine Learning”, Springer.
- 3) 竹林洋一: “ヒューマンインタフェース”, 人工知能学辞典, オーム社.