
画像処理シリーズ
Windowsパソコンで始める
コピペで動く
150
プログラム
付き
初めての
OpenCV
画像処理
吉田 大海 著
Image
enhancement
Region extraction
Noise
remove
Quality
evaluation
CQ出版社

第0部

第1章

150プログラムの中身

本書で取り上げる画像処理プログラムを**表1**，**表2**，**表3**，**表4**に示します．

●第1部 静止画像の画像処理

第1部では静止画像を対象とした画像処理を紹介します（**表1**）．基礎的な濃淡値処理を始めとして，しきい値処理や拡大縮小，エッジ検出などの空間フィルタリング，画像の合成から分析まで幅広く網羅しています．

表1　まずはここから！基本の画像処理

番　号	内　容
1-1	B/G/Rの入れ替え
1-2	セピア・カラー処理
1-3	グレー化処理
1-4	輝度反転
1-5	ポスタリゼーション
1-6	ソラリゼーション
1-7	モザイク画像処理

（**a**）第1章：色の変換

番　号	内　容
2-1	バイアス調整
2-2	γ補正
2-3	折れ線トーンカーブ処理
2-4	正弦波揺らぎ
2-5	コントラスト伸長
2-6	ヒストグラム平坦化

（**b**）第2章：明るさの調整

番　号	内　容
3-1	ヒストグラム描画
3-2	YIQ変換のY画像
3-3	YIQ変換のI画像
3-4	YIQ変換のQ画像
3-5	YCbCrのCb画像
3-6	YCbCrのCr画像
3-7	HSV変換のH画像
3-8	HSV変換のS画像
3-9	HSV変換のV画像

（**c**）第3章：色フォーマット変換

番　号	内　容
4-1	回転変換
4-2	スキュー変換（水平）
4-3	スキュー変換（垂直）
4-4	サイン揺らぎ（水平）
4-5	サイン揺らぎ（垂直）
4-6	上下反転画像
4-7	左右反転画像
4-8	平行移動（水平）
4-9	平行移動（垂直）

（**d**）第4章：変形/移動

番　号	内　容
5-1	ガウス補間拡大
5-2	最近傍補間拡大
5-3	線形補間拡大
5-4	ニアレスト・ネイバー補間（拡大/縮小）
5-5	バイキュービック補間（拡大/縮小）
5-6	バイリニア縮小

（**e**）第5章：拡大/縮小

番　号	内　容
6-1	平均値フィルタ
6-2	中央値フィルタ
6-3	最小値フィルタ
6-4	最大値フィルタ
6-5	ガウシアン・フィルタ
6-6	特定方向への平滑化

（**f**）第6章：ぼかし（ローパス）

プログラムは本書サポート・ページから入手できます．
https://interface.cqpub.co.jp/opencv-1/

表1　まずはここから！基本の画像処理（つづき）

番　号	内　容
7-1	横方向の1次微分
7-2	縦方向の1次微分
7-3	横方向の2次微分
7-4	縦方向の2次微分
7-5	ラプラシアン・フィルタ
7-6	ハイパス・フィルタ（平均値フィルタ）
7-7	ハイパス・フィルタ（ガウシアン）
7-8	DOGフィルタ
7-9	プリューウィット・フィルタ（横）
7-10	プリューウィット・フィルタ（縦）
7-11	ソーベル・フィルタ（横）
7-12	ソーベル・フィルタ（縦）
7-13	ロバーツ・フィルタ（横）
7-14	ロバーツ・フィルタ（縦）
7-15	自己商フィルタ

（**g**）第7章：輪郭の検出（ハイパス）

番　号	内　容
8-1	ダイレーション
8-2	エロージョン
8-3	オープニング
8-4	クロージング
8-5	ブラック・トップハット
8-6	ホワイト・トップハット
8-7	コンディショナル・ダイレーション

（**h**）第8章：拡大縮小によるノイズ除去

番　号	内　容
9-1	しきい値による2値化処理
9-2	マスキング処理
9-3	判別分析法
9-4	モード法
9-5	Pタイル法
9-6	グロウカット
9-7	レベル・スライス2値化法
9-8	局所平均2値化法

（**i**）第9章：ターゲット抽出

番　号	内　容
10-1	HDR合成
10-2	αブレンディング
10-3	ホワイト・ノイズの平滑化
10-4	ハイブリット・イメージ
10-5	ポアソン合成
10-6	鮮鋭化フィルタ
10-7	バイアス消去処理
10-8	簡単なインペインティング
10-9	高解像度化

（**j**）第10章：合成

番　号	内　容
11-1	電子透かし
11-2	レベル表示画像
11-3	ガウシアン・ピラミッド
11-4	ラプラシアン・ピラミッド
11-5	局所フラクタル次元
11-6	平均隣接数
11-7	ブランケット法によるフラクタル次元
11-8	テンプレート・マッチング
11-9	ヒルディッチの細線化

（**k**）第11章：分析

番　号	内　容
A-1	ホワイト・ノイズによるハーフトーニング
A-2	ハーフトーン型ハーフトーニング
A-3	ベイヤー型ハーフトーニング
A-4	フロイドスタインバーグ型誤差拡散法
A-5	バークス型誤差拡散法

（**l**）Appendix1：白黒の粗密 濃淡表現

番　号	内　容
B-1	ポスター風画像
B-2	鉛筆画風ハッチング
B-3	鉛筆画風クロス・ハッチング
B-4	鉛筆画風ブレンディング
B-5	ペンシル・ストローク・マップ
B-6	鉛筆画風変換

（**m**）Appendix2：特殊加工

●第２部 動画像の画像処理

第2部は動画像を対象とし，フレーム間差分や時空間フィルタ，ディゾルブなど時間軸情報を利用した画像処理を中心に解説します（**表2**）．さらに，動画の部分切り出しや逆再生など動画の分析/加工に便利な処理まで幅広く取り扱っています．

表2　動画像の画像処理

番　号	内　容
1-1	フレーム切り出し
1-2	動画作成
1-3	モノクロ動画
1-4	部分切り出し
1-5	高速再生
1-6	低速再生
1-7	逆再生
1-8	拡大/縮小
1-9	任意サイズの切り取り
1-10	動画並列出力

（**a**）第1章：基本動画処理

番　号	内　容
2-1	フレーム間差分
2-2	局所時間平均値フィルタ
2-3	局所時間中央値フィルタ
2-4	局所時間最小値フィルタ
2-5	局所時間最大値フィルタ
2-6	局所時間差分鮮鋭化フィルタ

（**b**）第2章：動きや明るさの変化検出

番　号	内　容
3-1	全時間平均画像
3-2	全時間最小画像
3-3	全時間最大画像
3-4	全時間最頻値画像

（**c**）第3章：全体の特徴を知る

番　号	内　容
4-1	時空間中央値フィルタ
4-2	時空間最小値フィルタ
4-3	時空間最大値フィルタ
4-4	時空間ラプラシアンフィルタ
4-5	時空間鮮鋭化フィルタ

（**d**）第4章：時空間フィルタの世界

番　号	内　容
5-1	背景差分による異物検出
5-2	パーティクルフィルタ

（**e**）第5章：物体検出

番　号	内　容
6-1	合成処理
6-2	ディゾルブ
6-3	ライフ・ゲーム
コラム	平滑化を伴うエッジ抽出

（**f**）第6章：特殊な効果

番　号	内　容
	テストプログラム
	入力動画セット

（**g**）共通

●第３部 画像処理の応用例

第3部では応用事例を紹介します（**表3**）．第1部で解説した複数の基本画像処理を組み合わせて作成した画像処理システムや，VR/ARに欠かせないステレオ画像を対象とした画像処理など，少しだけ高度な画像処理をピックアップしています．

表3　画像処理の応用例

章番号	内　容
1	基礎画像処理を組み合わせるだけ! ひび割れ検出
2	HSV表色系を使用した領域抽出
3	マーカにシールを利用したカメラ位置のずれ検知
4	2枚の画像を任意の横サイズでステレオ画像化する
5	ステレオ画像から距離計測…SADによるステレオ・マッチング
6	垂直成分を手がかりとした水平エッジ付きSADによるステレオ・マッチング
7	立体感最大のエッジ検出結果を獲得する「視差情報評価に基づくケニー・エッジ検出」
8	カメラとフレーム間差分を使用した動き検知

●第４部 画像の評価

第4部では画像の評価方法を紹介します（**表4**）．領域抽出結果のノイズ評価方法，画質の評価方法，動画像の評価方法について解説します．

表4　画像の評価方法

番　号	内　容
1-1	適合率
1-2	再現率
1-3	F値
1-4	正解率

（**a**）第1章：領域抽出結果のノイズ評価

番　号	内　容
2-1	RMSE
2-2	PSNR
2-3	輝度のRMSE
2-4	彩度のRMSE
2-5	色相のRMSE

（**b**）第2章：画質の評価方法

第1章

色の変換

1-1 大胆に色を変える「B/G/Rの入れ替え」

収録フォルダ：BGR入れ替え処理

画像の色を，全体のバランスを維持したまま変更するのは本来難しい処理となります．B/G/Rの入れ替え処理は，簡単な操作で実現できる上に，色を大胆に変更できます．しかも，各チャネルに格納されている値には変更を加えないため，全体のバランスを維持できます．

●仕組み

B/G/Rの入れ替えの仕組みを**図1**に示します．各チャネル強度を交換すると，出力するB/G/Rの画素値そのものは変化するため色は変わるものの，チャネルの値自体はそのままのため，色の関係は保存されます．従って，全体のバランスを維持したまま色の変更を行うことができます．

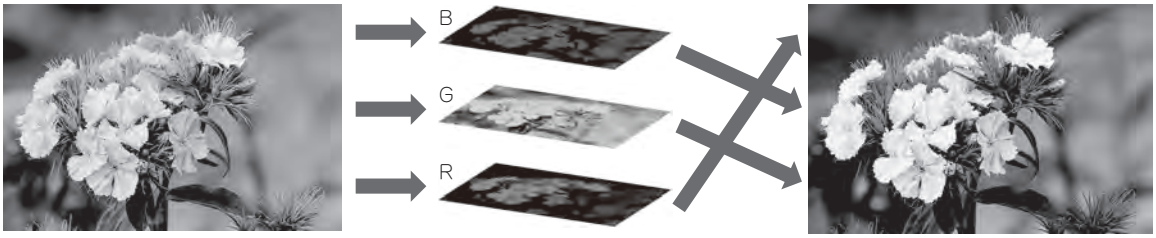


図1　B/G/Rの入れ替え…全体のバランスを維持したまま色の変更ができる

リスト1　B/G/Rの入れ替えのプログラム（抜粋）

```
for (y = 0; y < Y; y++) {
  for (x = 0; x < X; x++) {
    //入力画像から画素値を読み込む
    p[0] = img.at<cv::Vec3b>(y, x)[0]; //B(青色)
    p[1] = img.at<cv::Vec3b>(y, x)[1]; //G(緑色)
    p[2] = img.at<cv::Vec3b>(y, x)[2]; //R(赤色)

    //BGRの画素値を入れ替えて出力画像として書き込む
    img.at<cv::Vec3b>(y, x)[0] = p[2];  //Bへ代入
    img.at<cv::Vec3b>(y, x)[1] = p[0];  //Gへ代入
    img.at<cv::Vec3b>(y, x)[2] = p[1];  //Rへ代入
  }
}
```

第1部第1章のカラー画像はこちらから
https://interface.cqpub.co.jp/opencv-11/

第2章

明るさの調整

2-1　直線的に明るさを増減する「バイアス調整」

収録フォルダ：バイアス調整

画像を明るくしたり，暗くしたりする最もシンプルな方法は，画像の全画素に同じ数を加えることです．画像データでは，画素値が大きいほど画像は明るくなり，画素値が小さいほど画像は暗くなります．従って，画素値に正の定数を加えると画像は明るく，負の定数を加えると画像は暗くなります．

●仕組み

画像のバイアス調整の仕組みを**図1**に示します．画像から画素値を抽出し，変更後，再度格納して画像に保存するという，画像処理の最も基本的な操作ができれば実現できます．

図1では，画素値に70を加えています．全ての画素に70を加えることで，直線全体が，切片を与えられた1次関数のように位置を上げています．

この例では入力画像において186以上の画素値を持つ画素は，出力結果で256以上の値をとってしまいます．従って，「256以上の画素は全て255にする」というクリッピング処理が必要になります．

●実行結果

バイアス調整処理のプログラムを**リスト1**に示します．結果は**図1**に示した通りです．

今回の例ではバイアスを70としているため，画像全体が明るくなっています．一方で，画素値が186以上の画素は白つぶれとなっているのも確認できます．この白つぶれは一般的に弊害となるものですが，被写体のささいな汚れや傷を消去する効果を狙って，意図的に行う場合もあります．

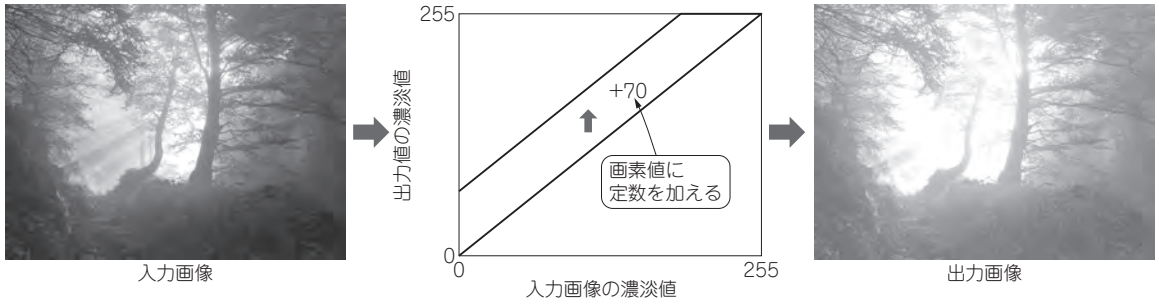


図1　バイアス調整…画素値に定数を加える

第3章 色フォーマット変換

3-1 画像の濃淡がひと目でつかめる「ヒストグラム描画」

収録フォルダ：ヒストグラム描画

画像のヒストグラムは，どのぐらいの明るさの画素が，どのぐらいの割合で含まれているかという分布を示すグラフです．このデータは画像の性質を知るだけでなく，物体認識や画像2値化，コントラスト調整，ノイズ除去など多くの画像処理に活用できます．

●仕組み

濃淡ヒストグラム描画の方法を**図1**に示します．画像に含まれる同じ明るさ（一般的な8ビット画像なら画素値が0 ～ 255）の画素数をカウントします．そして横軸を明るさ，縦軸を画素数としてグラフとして描画すれば，濃淡ヒストグラムとなります．

このとき，ヒストグラムの各ビンの長さを描画画素数と対応する場合は，画像サイズに注意が必要です．画像サイズが大きいとビンが長くなるためです．ビンの長さは正規化した方がよいでしょう．

●実行結果

入力画像のヒストグラムを描画するプログラムを**リスト1**に示します．実行結果は**図1**に示した通りです．

画素値が50近辺の暗めのグレーと，150近辺の明るめのグレーに多くの画素があると確認できます．実際に入力画像を確認すると，明るめのグレーはカップ，暗めのグレーは背景部分であると推測できます．

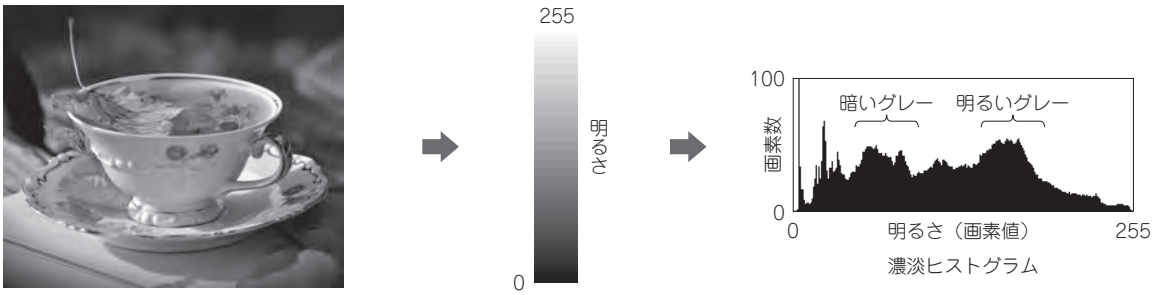


図1　ヒストグラム描画…画素値ごとに画素数をカウントしてグラフ化する

第4章

変形/移動

4-1　水平/斜め/垂直に傾ける「回転変換」

収録フォルダ：回転変換

画像の回転変換処理は，画像を設定した角度に回転させる処理です．写真を撮影をしたときに，うまく水平に撮れなかったときは，この変換処理によって補正できます．

●仕組み

画像の回転処理の仕組みを**図1**に示します．入力画素の座標を(x, y)とすると，変換結果の座標(x_2, y_2)は以下の式で求まります．

$$x_2 = x \times \cos\theta - y \times \sin\theta$$
$$y_2 = x \times \sin\theta - y \times \cos\theta$$

ただし，この式をそのまま適用すると画像の水平位置が大きく移動してしまうため，必要に応じてx_2に$(Y-1) \times \sin\theta$を加えるようにします（Yは縦方向の画素数）．

変換結果の座標(x_2, y_2)は小数点になることがあります．このため切り捨てや切り上げによって空白になってしまう場所がでてきます．これを取り除きたい場合は，中央値フィルタ（第6章参照）などを適用します．

●実行結果

画像を回転するプログラムを**リスト1**に示します．実行結果は**図1**に示した通りです．

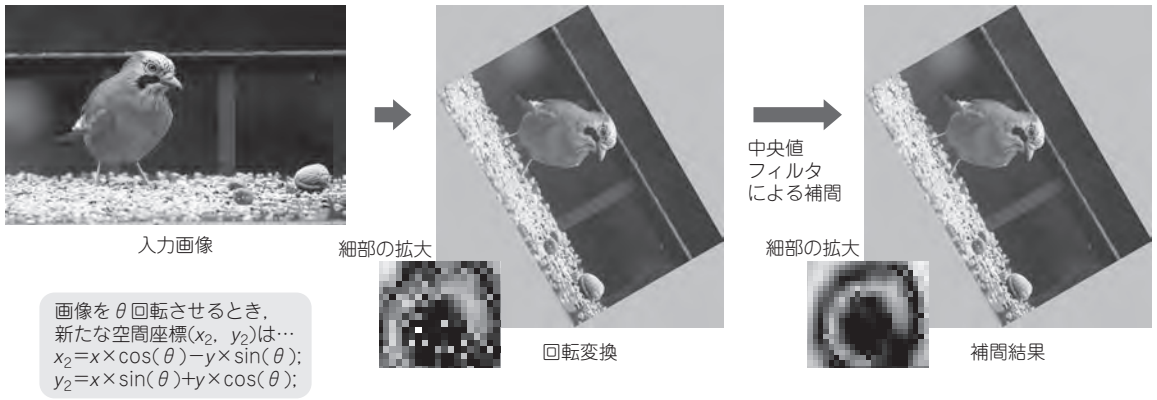


図1　回転変換…座標変換により画像を回転させることができる

第5章

拡大/縮小

5-1　滑らかに拡大する「ガウス補間拡大」

収録フォルダ：ガウス補間拡大

出力結果を滑らかに拡大する方法としては，ガウス補間拡大が優秀です．見た目に自然で滑らかな出力結果が得られます．

●仕組み

ガウス補間拡大の仕組みを**図1**に示します．

拡大したいサイズを元のn倍とします．

出力画像として得たい画素の空間座標を(x, y)とします．入力画像の$(x/n, y/n)$の画素を中心とするブロックにガウス分布の係数を掛けて，総和をとった値が出力画像の画素値になります．

●実行結果

ガウス補間拡大のプログラムを**リスト1**に示します．実行結果は**図1**に示した通りです．

出力結果は2倍のサイズに拡大したものです．画像を拡大してみると，細部が滑らかな要素で構成されており，見た目にも自然な結果になっていることが分かります．

この滑らかさはガウシアン平滑化により出力結果を決定していることにあります．この方法は見た目には滑らかですが，2値(白黒)画像を拡大する場合は意図しない濃淡値(グレー)を生成してしまうため，注意が必要です．

図1　ガウス補間拡大…ガウシアン平滑化により見た目に自然で滑らか

第6章

ぼかし（ローパス）

6-1　ボカしの基本「平均値フィルタ」

収録フォルダ：平均値フィルタ

平均値フィルタは，画像をピンぼけのようにボカすフィルタ処理です．この処理は，画像中の詳細さを消去したり，ノイズを低減したり，ギザギザと角ばった輪郭を滑らかにする効果があります．

●仕組み

平均値フィルタの仕組みを**図1**に示します．注目画素とその周辺画素の平均値を新たな画素値として代入することで実現します．具体的には，係数フィルタを画像に畳み込みます（積和をとる）．

画像に局所的な平均値を代入していくと，画像の局所的な最大値と最小値の差が小さくなり（平均に近づいていくため），画像の大域的な情報が浮かび上がってきます．このような効果を，大域通過（ローパス）特性と呼びます．具体例を挙げると，画像ノイズは外れ値のような特性を持つことが多いため，このフィルタを適用すると平均外として消去されます．しかし同時に，画像の詳細さも大域的な観点から見るとノイズの一種となるため，同様に消去されてしまいます．

平均値を計算する範囲をウィンドウ・サイズもしくは走査窓と呼び，奇数で設定するのが一般的です．ウィンドウ・サイズを大きくすることで，平均値フィルタの効果は大きくなります．

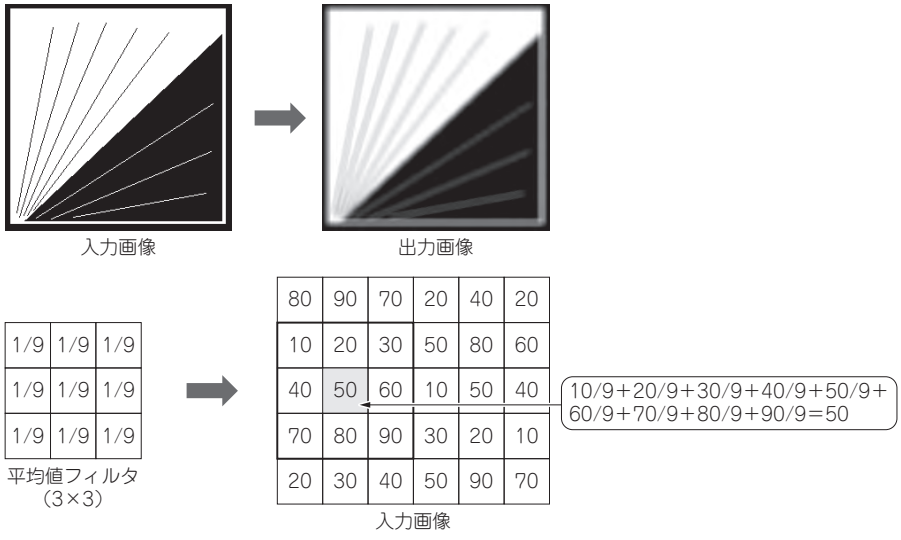


図1
平均値フィルタ…注目画素とその周辺画素の平均値を新たな画素値として代入する

第7章

輪郭の検出（ハイパス）

7-1 基本的な縦方向輪郭検出「横方向の１次微分」

収録フォルダ：一次微分￥横

１次微分で画像の縦方向の輪郭を検出する仕組みを**図1**に示します．

画像の輪郭とは，簡単にいえば色や明るさの変わり目のことを指します．われわれはその変わり目がつながって長い線になっている場合に，輪郭として認識しています．

画像から色や明るさの変わり目を検出する場合は，画像を微分します．最も単純な微分は1次微分であり，特にディジタル画像の場合は引き算に近似されます．

１次微分で画像から縦方向の輪郭を検出するプログラムを**リスト1**に示します．実行結果は**図1**に示した通りです．

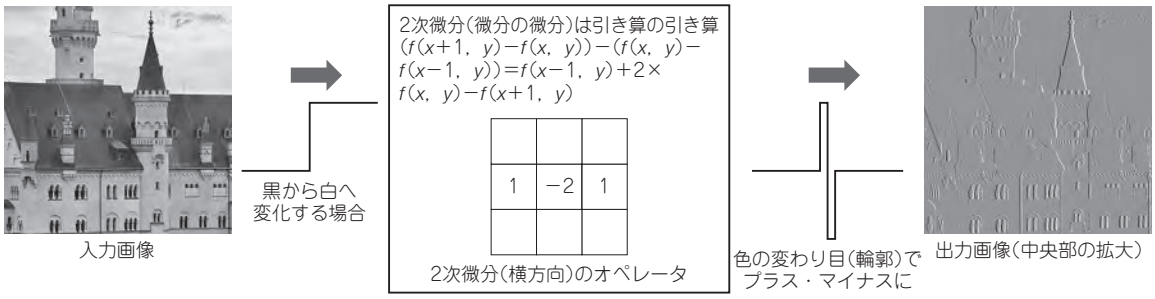


図1　横方向の１次微分…横方向で引き算を行うと縦方向の輪郭を検出できる

リスト1　横方向の１次微分のプログラム（抜粋）

```
//入力画像から隣り合う2画素の画素値を読み込む
p[0] = img2.at<cv::Vec3b>(y, x-1)[0];//Bの画素
p[1] = img2.at<cv::Vec3b>(y, x)[0];//Bの画素

//1次微分してグレー色を求める
P = p[1] - p[0] + 128;
if (P > 255) {
  P = 255;
}
if (P < 0) {
  P = 0;
}
p[3] = P;
```

第8章

膨張収縮によるノイズ除去

8-1 白い領域を膨張させて文字等を消せる基本「ダイレーション」

収録フォルダ：ダイレーション

ダイレーション（Dilation）は，モルフォロジー（Morphology）演算における基本処理の1つです．モルフォロジー演算とは，構造化要素という領域定義と集合演算を組み合わせた画像処理です．ダイレーションは膨張処理とも呼ばれており，白い領域を一回り「大きく」します．

ダイレーションは黒色の領域を消失させることができます．しかし，文字以外の部分でも全体的に輝度が上がってしまうなどの影響を与えるため，その点は注意が必要です．

●仕組み

ダイレーションの仕組みを**図1**に示します．入力画像を十字型の構造化要素に従って上下左右に動かします．作成した画像5枚（原点含めて）を全て含む画像がダイレーション結果になります．

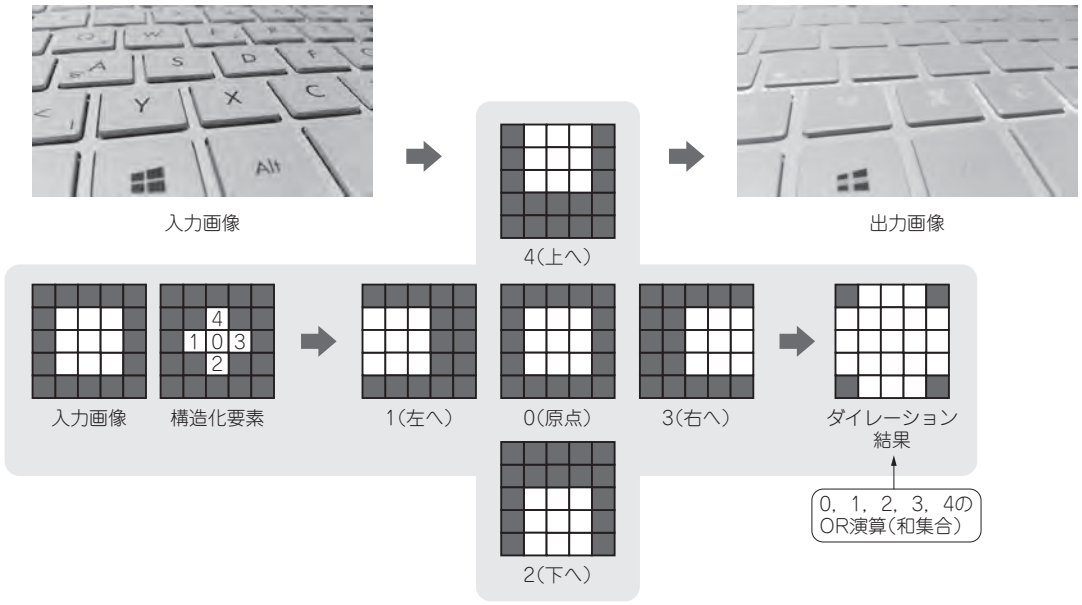


図1　ダイレーション…画像を構造化要素に従って動かして和集合を取ると膨張する

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第9章 ターゲット抽出

9-1 画像分離/抽出の基本「しきい値による2値化処理」

収録フォルダ：任意での閾値処理

2値化処理は，濃淡画像を白黒画像に変換する処理です．画像からの領域抽出や，画像伝送時に情報量を節約することを主な目的として使用されます．

2値化において，しきい値を自動決定する方法はたくさんあります．最もやさしいのは，1つの決まったしきい値だけで画像全体を2値化する（白と黒に分ける）方法です．

●仕組み

任意のしきい値による2値化処理の仕組みを**図1**に示します．

2値化処理は，設定したしきい値に対して注目画素の値が高い場合は白に，低い場合は黒に変換することで実現します．

●実行結果

任意のしきい値による2値化処理のプログラムを**リスト1**に示します．実行結果は**図1**に示した通りです．

しきい値（t）を128に設定しています．濃淡画像が白黒に変換されています．入力画像は自然画像に比べてシンプルな色構成のため，2値化によって大きく情報量が損なわれても，元の画像の様子をしっかりと確認できます．

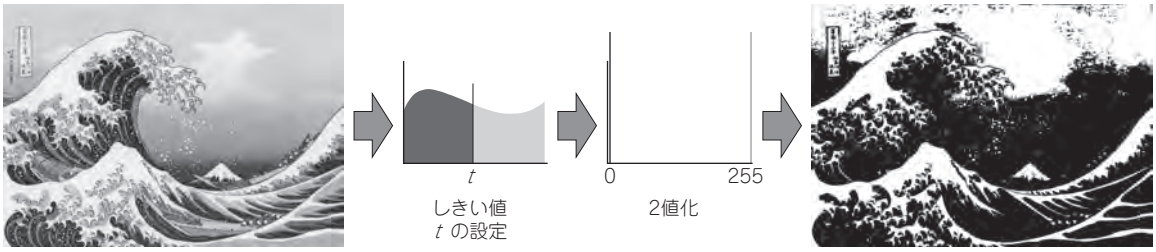


図1　任意のしきい値による2値化処理…1つの決まったしきい値だけで画像全体を白と黒に分ける

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第10章

合成

10-1 明暗や色彩のチャンピオン画像を合成する「HDR合成」

収録フォルダ：HDR化処理

カメラで同じ被写体を撮影しても，そのときの天候や時間帯によってきれいに見える部分が変わってきます．例えば，暗いところに合わせて撮影すると明るいところは白飛びしてしまいます．反対に，明るいところに合わせてると暗いところは黒つぶれしてしまいます．これは「カメラが表現できる情報量（リソース）を明るいところに割くか，もしくは暗いところに割くか」という問題です．HDR（High Dynamic Range）合成は，これら一長一短の写真から長所のみを合成し，明るいところも暗いところもきれいに見えるようにするための処理です．

●仕組み

HDR合成の仕組みを**図1**に示します．明るいところがきれいな画像A，暗いところがきれいな画像B，そしてその中間（通常の環境で撮影）に位置する画像Cを合成することで実現します．

画像Cにγ変換を適用して明るめの画像B，暗めの画像Aを作成します．さらに，3枚の画像の平均濃

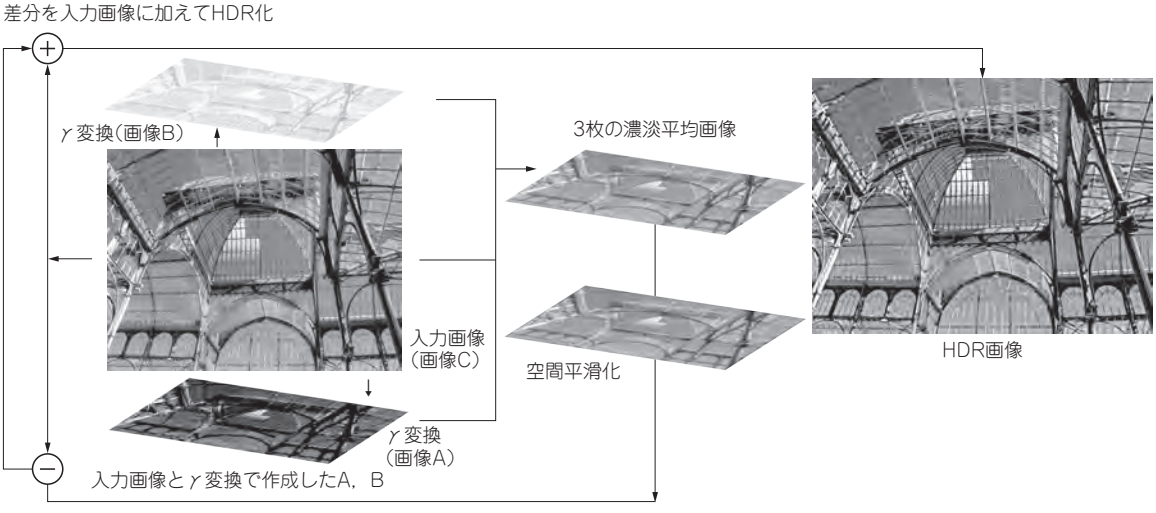


図1　HDR化処理…明るいところがきれいな画像A，暗いところがきれいな画像B，中間の画像Cを合成する

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第11章 分析

画像は，単に「絵」として見えるだけのものではありません．分析を行うことで，目には見えない，さまざまな情報が得られます．

この章では，画像データの分析手法について解説します．基本的な処理ばかりですが，それでも各技術は第10章までの処理と比べると複雑で，プログラムも長くなります．詳しくは，ダウンロード・データに収録のソースコードを参照してください．紙面では，概要のみ紹介します．**（編集部）**

11-1 著作権等の情報を埋め込める「電子透かし」

収録フォルダ：電子透かし

電子透かしは，画像に著作権などの情報を埋め込む技術の1つです．この技術はデータ・ハイディング（Data Hiding）と呼ばれ，一見して加工されたように見えません．

電子透かしの仕組みを**図1**に示します．埋め込む画像は白と黒の2値画像，埋め込み先の画像はグレー画像とします．埋め込む画像が白ならば，埋め込み先のグレー画像の画素値を偶数に変換（もともと偶数

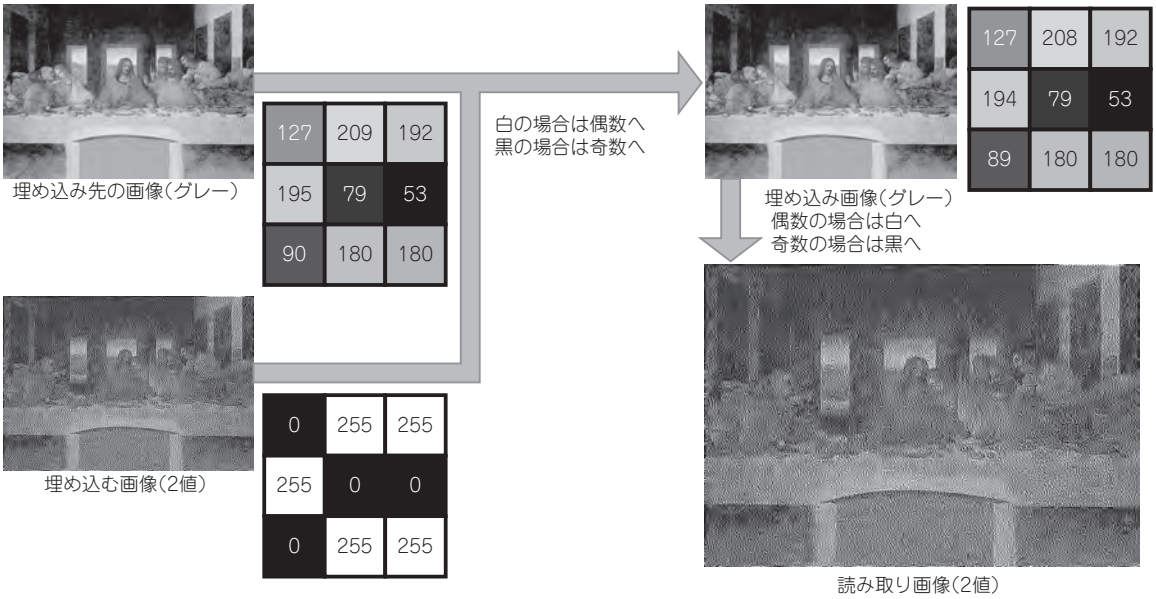


図1　電子透かし…埋め込む画像の画素値が白ならば埋め込み先のグレー画像を偶数に変換する

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第1章 基本動画処理

1-1 動画から決定的瞬間の画像を抽出する「フレーム切り出し」

収録フォルダ：フレーム切り出し

フレーム切り出しとは，動画から任意のフレームを切り出して保存する処理です．

動画はフレームと呼ばれる静止画が何枚も集まってできています．従って，その中から重要なフレームを切り出すというのは，映像から決定的瞬間を保存する技術であるとも言えます．例えば，サッカーの映像からゴールの瞬間を切り出す，ホーム・ビデオから子供がバースデー・ケーキのロウソクを吹く瞬間を切り出す，などが挙げられます．その他，プレゼンテーションの資料で動画の特定シーンを見せたいときに使える，非常に便利な技術です．

●仕組み

フレーム切り出しの仕組みを**図1**に示します．再生中の動画から任意のタイミングでフレームを静止画像として保存することで実現します．タイミングの設定方法は，大きく分けて3種類の方法が考えられます．

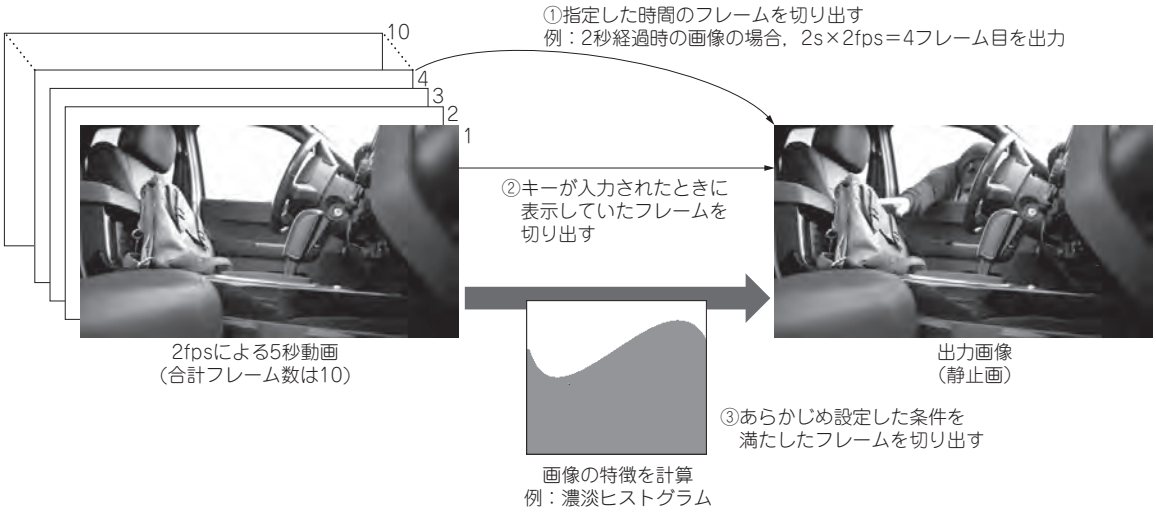


図1　フレーム切り出し…動画から任意のフレームを切り出して静止画像を保存する

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第2章

動きや明るさの変化検出

2-1 動き検出で最も重要な方法「フレーム間差分」

収録フォルダ：フレーム間差分

フレーム間差分処理は，動画の動きを検出することができる処理です．数ある方法の中で最も基本的かつ重要なものの1つです．

フレーム間差分を利用した動物体の検知，追跡方法は，時空間ラプラシアン・フィルタ，オプティカル・フローなど，これまでに数多くの応用が報告されています．また，対象が定点カメラである場合は，この処理だけでも十分に動物体を検知できます．

●仕組み

フレーム間差分の仕組みを**図1**に，例を**図2**に示します．動画内で連続する2枚のフレームを取り出し，その差分を検出することで実現できます．

時刻tにおいて表示されているフレームの空間座標$(x,\ y)$の画素データを$\mathrm{f}(x,\ y,\ t)$と表現するとします．その場合，フレーム間差分処理は，

$$\mathrm{f}(x,\ y,\ t) - \mathrm{f}(x,\ y,\ t-1)$$

となります．

直感的には，現在のフレームと過去のフレームとを比較すると，動きがある場合には違いが生じ，動かない場合には違いが生じないという考えです．そのフレーム間差分の値を動画として書き出していくことで，フレーム間差分動画となります．

図1ではヒマワリの動画を例にしています．入力動画においては花や葉が風で揺れています．出力動画では，その動きの変化が大きいところが白や黒で表示されます．

このとき，動きの変化の仕方によって値がマイナスになることがあります．具体的には，動画が明るいものから暗いものに変化するときはマイナスの値になり，暗いものから明るいものに変化するときはプラスの値になります．そこで出力値には127を足して正の値にするのが一般的です．

定点カメラにおける利用例を**図2**に示します．入力動画はクルマの中に設置された定点カメラの映像です．座席のバッグが奪われる様子が確認できます．盗難前，犯人の出現，バッグの把持，バッグの持ち去りのそれぞれの場面が含まれています．

この様子を出力動画のフレーム間差分で確認すると，動きのない状態では何も検出しません．しかしその他のフレームでは犯人の顔やバッグなど，動いたり動かされたりしたものを検出しています．

第3章

全体の特徴を知る

3-1 物体の動き情報を画像で表現する「全時間平均画像」

収録フォルダ：全時間平均画像

全時間平均画像処理は，動画を1枚の画像情報で表現する方法の1つです．基本的には定点カメラで撮影した動画が対象となり，そこに含まれる物体がどんな動きをしたのか，それを1枚の画像で表現することができます．

具体的には，対象が動かなければ静止画そのままで出力され，動くとその軌跡が出力されます．速度や動き方によって軌跡が変わるため，特性を理解すれば画像から，

- 動いたもの
- 動いた向き
- 動いた位置

を大まかですが判断することができます．

また，全時間平均画像はその性能だけでなく，出力結果の画像自体が非常に面白く，動画によっては芸術性の高い画像が作成できます．

●仕組み

全時間平均画像処理の仕組みを**図1**に示します．動画を構成する全てのフレームから平均フレームを出力することで実現します．

時間軸，すなわちフレーム・ナンバ方向に各フレームの画素値を足していき，全フレーム数で割ることによって平均フレームを求めることができます．

このとき，撮影範囲内で動く物体は軌跡となって画像に現れます．その現れ方は**図2**に示す通りです．

例えば黒い背景の中を青いボールが一定の速度で横切ったとします．すると，時間平均画像では均質な横方向の軌跡が生じます．これは，動画におけるボールの位置を平均的に表した場合，等速度であれば軌跡内で均質となるためです．

一方，ボールの移動速度が変化すると結果は異なります．ボールの移動速度が中，高，低と変化したとします．高速で通過した位置と低速で通過した位置では，後者の方が長くボールが滞在します．従って，軌跡にはムラが生じ，より具体的には中，薄，濃という形で出力されます．

このように，全時間平均画像では動き方によって出力結果に違いが生じてきます．

第4章

時空間フィルタの世界

4-1 突発的に現れるノイズを効果的に除去する「時空間中央値フィルタ」

収録フォルダ：時空間中央値フィルタ

時空間中央値フィルタ処理は，画像に生じた飛び値のようなノイズを除去する場合に有効です．

静止画像処理における中央値フィルタの考え方を，時間軸に対しても適用します．中央値フィルタは代表的なインパルス・ノイズ除去フィルタです．時空間中央値フィルタでは，空間座標と時空間の両方の情報を手掛かりとするため，同じインパルス・ノイズ除去を行うにしても，より高い効果が得られます．

時空間中央値フィルタは繰り返し適用することで，ノイズ除去性能を上げることができます．

●仕組み

時空間中央値フィルタの仕組みを**図1**に示します．動画内で連続する奇数枚のフレームを対象に，注目画素を中心とした時空間の近傍の画素を抽出し，画素値を大きさの順に並べ替えます．その中央値を画素として，出力動画のフレームを作成していきます．

図1では，取り出すフレーム数を3とし，注目フレームの時刻をtとしています．つまり，取り出すフレームは$t-1$，t，$t+1$の3枚です．

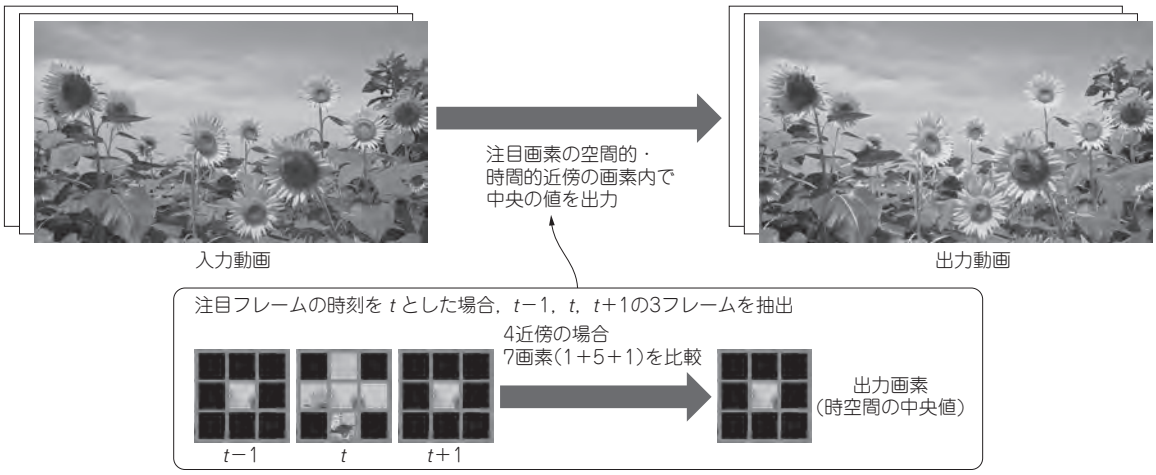


図1　時空間中央値フィルタ…空間座標と時空間の両方の情報を手掛かりにして効果的にインパルス・ノイズを除去する

第5章

物体検出

5-1 監視カメラの侵入者検出にも使える「背景差分による異物検出」

収録フォルダ：背景差分による異物検出

背景差分による異物検出処理は，定点カメラで撮影された動画から，移動物体や本来存在すべきでない異物を検出するための基本処理です．

処理のアイデアはフレーム間差分と似ており，非常にシンプルです．しかし適切な状況で使用すれば，実用レベルの精度と効果を期待できます．例えば，立ち入り禁止区域に設置した防犯カメラから侵入者を検知するなどが主な利用方法です．

●仕組み

背景差分による異物検出の仕組みを**図1**に示します．

入力動画を構成するフレームと，背景として用意された特別なフレームとの差分を計算し，それを2値化することで実現します．

背景画像は，最頻値画像として得られるような通常時の画像です．例えば，立ち入り禁止区域に設置

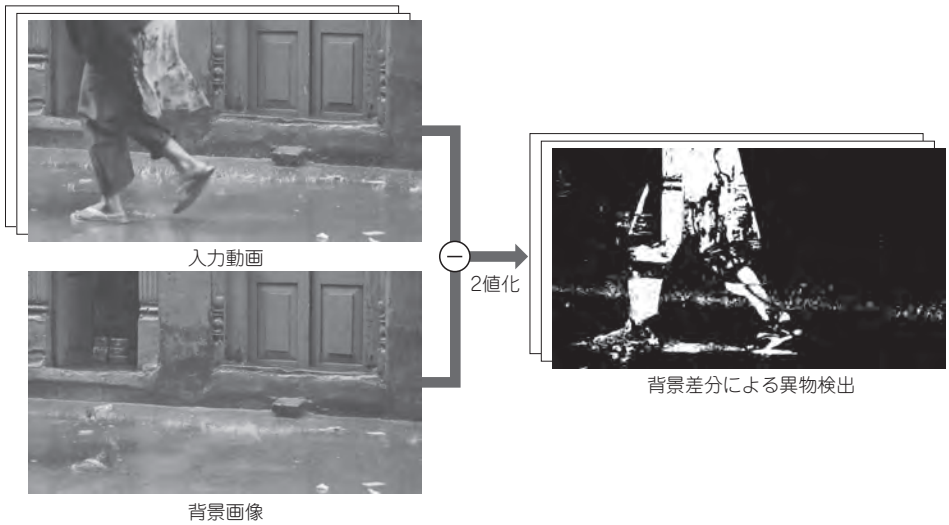


図1　背景差分による異物検出…定点カメラの映像のように背景に動きがない場合に移動物体を検出する

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第6章

特殊な効果

6-1 文字などの情報を重ね合わせる「合成処理」

収録フォルダ：動画への合成処理

ここで紹介するのは静止画を動画に合成するため処理です．合成の際に，その位置をフレームに応じて設定していくことで，静止画を動画内で動かすことができます．

例えば，移動物体検出を目的とした実験用サンプルを作成したいとき，スタッフ・ロールのように文字を移動させたいとき，単純なコラージュ作品を作りたいときなどに，この処理は活躍します．

●仕組み

合成処理の仕組みを**図1**に示します．入力動画を構成するフレームに対して，その都度静止画を合成していくことで実現します．

図1において，静止画のサンプルは「移動物体」という文字画像です．黒の領域は背景，すなわち合成先の動画を表示するように処理します．このとき特定の座標に合成し続けるだけでは，文字画像が動くことはありません．フレームに応じて合成位置を変化させることで移動するように見えます．

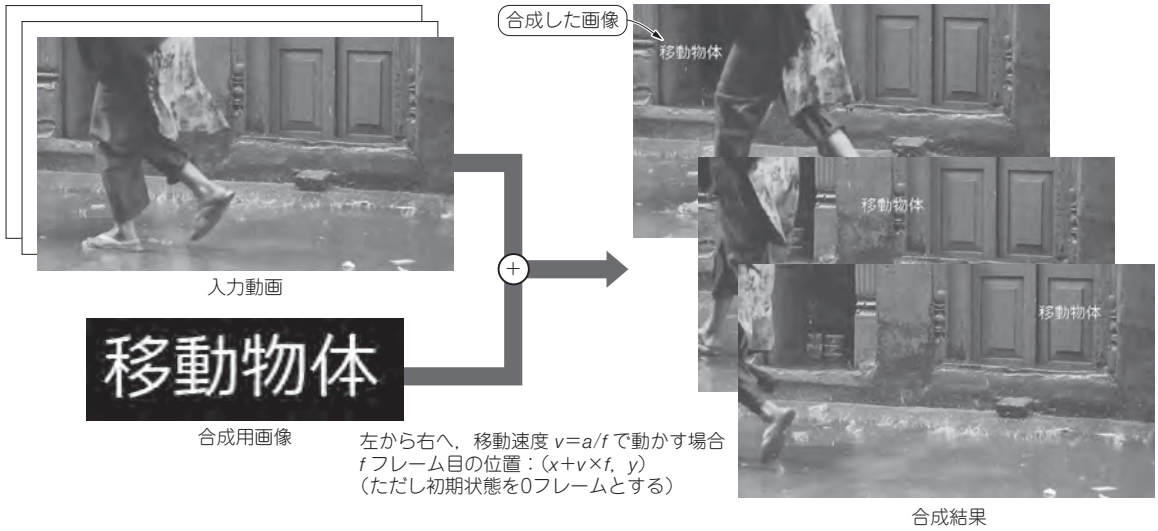


図1　合成処理…文字やアイコンのような静止画を動画に重ね合わせる

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第1章
第1部の基本処理を組み合わせて「ひび割れ検出」

●個別処理の掛け算が大きな処理を成し遂げる

紹介するひび割れ検出システムは，これまでのような単独の画像処理ではなく，その組み合わせに重点を置いた内容です．これは単独の画像処理のように，試したら終わるのではなく，システムだからこそ処理の具体的な目的が存在し，そのためにチューニングすべきパラメータを持ち，そして定量的に評価できる性能を有しています．また，このシステムの改良過程はパズルのような歯応えと楽しみがあり，その過程で得られる知見や感じた困難さは生きた知識になります．

●身近な事象，ひび割れを例に

ひび割れはわれわれにとって身近な存在です．道路面，ビルや住宅の壁面，絵画やガラス窓，あるいは工場の加工品など，普段目にしないものまで含めると，バリエーションには限りがありません．そして，それらは多くの分野で検出対象として研究されています．特に建築物や配管などに生じたものは人命にもかかわるため，今も多くの専門家や技術者が高精度な検出技術の開発に励んでいます．中でも画像処理を使用した検出技術はその主流の1つであり，近年はAI技術の発展に伴って，ますます高精度になっています．

●AIに頼らない画像処理で解く

ひと口にひび割れと言ってもバリエーションは豊かです．太さ，長さ，暗さ，形状，もちろん何に生じたのかという背景も含めると膨大なパターンがあります．これらを効率的に画像処理で検出するなら，対象となるひび割れを学習したAIによる検出システムは，間違いなく最有力候補でしょう．

ですが，その検出システムがひび割れの，

- どんな特徴を学習しているのか
- どんな課題を克服/達成しているのか

について，ニューラル・ネットワークに頼るAIには説明が難しいとされます．画像処理であれば，上記を説明可能です．また，この情報はシステム導入を検討しているユーザにとっても貴重な判断材料になります．

プログラムの入手先は
https://interface.cqpub.co.jp/opencv-1/

第2章 HSV表色系を使用した領域抽出

色情報は画像処理全般において最も重要な指針のひとつです．例えば「新鮮な小松菜は鮮やかで明るい緑色している」「錆(さび)は暗いくすんだ赤色をしている」など，人も色情報からモノの状態をしばしば判別しています．この判断をディジタル画像処理で自動化するにはどうすればよいでしょうか．ディジタル画像の色は，**図1**のようにR（赤色），G（緑色），B（青色）の光が，強度を変えながら発光することで実現しています．このような表色系をRGB表色系と呼びます．しかし，この方法は感覚的な色指定には不向きです．なぜなら，どのようにRGBの値を指定すれば，くすんだ暗い緑色になるか，簡単には分からないからです．そのような場合はHSV表色系が便利です．人の感覚に近い基準で色情報を指定できます．ここではHSV表色系を使用した領域抽出を紹介します．

カラー画像は，
https://interface.cqpub.co.jp/opencv-32/
で確認できます

図1　RGB888であればR：0 ～ 255，G：0 ～ 255，B：0 ～ 255の範囲で色を表現するものの，「くすんだ暗い緑色」などといった感覚的な指定は難しい

第3章
マーカにシールを利用したカメラ位置のずれ検知

「カメラ画像を使って物体位置を対象物に近付ける」と，文章だけ見ても大変困難な印象を受けます．しかし，もしも誘導に必要となる目印（マーカ）をシールに置き換えることができれば，驚くほど簡単かつ飛躍的に実現へ近づくことができます．ここではUSBカメラを物体の目とし，そしてシールを目印（マーカ）として対象物を認識しつつ，物体が対象物に対してどのようにずれているのかを表示できるシステム例を紹介します．

■仕組み

●1，シールの認識

目印となるシールはHSV表色系を使って検出を試みるため，シンプルで発色の鮮やかなものが良いです．今回は**図1**に示すような赤色／緑色／橙色の丸いシールを使用します．このとき，もしも暗い場所での誘導が目的であれば，シールではなくLEDランプなどを使用するのが良いかもしれません．そしてシールの認識に使用したHSVのパラメータは**表1**の通りですが，この数値は参考程度としてください．より良い精度目指すために，ぜひここは実環境でシール認識のみの予備実験を行い，検出精度の高いパラメータを使用するようにしましょう．

検出パラメータが決まったら，その検出結果の座標の平均値を検出位置とします．これで目印となるシールの中心位置になればパラメータ調整は完了です．のちのずれの計算精度を大きく左右するため，入念に調整しましょう．

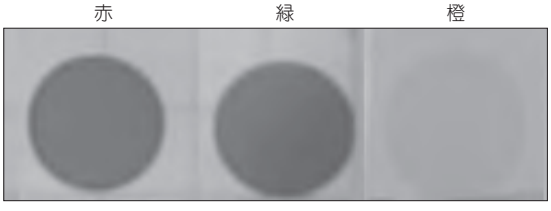


図1　今回は丸いシールを使用する
左から赤，緑，橙

表1　シールの認識に使用したHSVのパラメータ

色	HSV	最大	最小
赤	H（色相）	75	45
	S（彩度）	255	200
	V（輝度）	255	100
緑	H（色相）	75	45
	S（彩度）	255	120
	V（輝度）	200	50
橙	H（色相）	30	15
	S（彩度）	255	200
	V（輝度）	255	100

第4章

2枚の画像を任意の横サイズでステレオ画像化する

ステレオ画像は，視差を持つ2枚の画像を1セットとして横へつなげた画像であり，VRゴーグルや裸眼による平行法で鑑賞することで立体的に見ることができます．この処理は，視差を持つ2枚の画像をそれぞれ左画像，右画像として指定し，そして最終的な横サイズを入力することで，任意サイズのステレオ画像にできます．

●仕組み

まずは視差を持つ2枚の画像の取得方法について説明します．被写体を決めたら，スマートフォンのカメラ機能など，お好きなカメラで水平に注意しながら撮影してください．取得した写真を「左画像」とします．次に，カメラの水平を保ちながら，撮影位置を右に移動し，再び撮影してください．このとき，カメラの角度や高さは変えないよう気をつけてください．取得した写真を「右画像」とします．参考までに**写真1**の治具を作って撮影しています．

これで視差を持つ2枚の画像は取得できます．このとき，右に移動した距離が視差となります．この2枚の画像のうち「左画像」は左に配置し，「右画像」は右側に配置した画像がステレオ画像です．

具体的にどのような処理がなされるか**図1**で説明します．本処理では新たに生成されるステレオ画像の横サイズをまず指定します．仮にそのサイズがXだとすると，左画像，右画像のサイズはそれぞれ$X/2$です．そしてステレオ画像における空間座標で言えば，左画像の横サイズは0から$X/2-1$まで，右画像の横幅は$X/2$からXまでとなります．なお，指定したサイズへの拡縮には，画像の補間処理としては著名なバイキュービック補間法を採用しています（**リスト1**）．

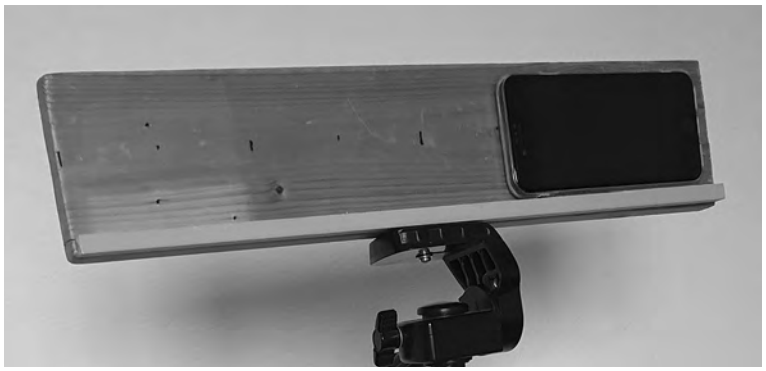


写真1　スマホでステレオ画像を作るための治具

第5章

ステレオ画像から距離計測…SADによるステレオ・マッチング

ステレオ画像からの距離計測を学ぶ場合，その入門として真っ先に上がる画像処理技術といえば「SADを使用したステレオ・マッチング」でしょう．SADはSum of Absolute Difference，絶対値の差の和のことです．後述します．

ステレオ・マッチングとは，ステレオ画像の左半分と右半分の水平位置がどのぐらいずれているか（視差と言う）を求めて，そのずれ量をカメラに対する距離と解釈して画像化する処理です．このズレ量を求めるためには「何が」「どの位置にあるか」という技術，つまり，類似度計算が必要となります．本処理はこの類似度計算にSADを用いたもっとも基本的な処理であり，ステレオ・マッチングのイロハが詰まっています．

●仕組み

ステレオ・マッチングの基本的な概念について学びましょう．ステレオ・マッチングは，ステレオ画像の「左半分と右半分（以後，左画像と右画像）が水平方向でどのぐらいズレているか」というズレ量を求める技術です．**図1**を見てみてみましょう．図には，ステレオ画像を左画像と右画像の2枚に分割し，縦に並べたものがあります．この2枚の画像を見比べると，画像中央に立っている丸太の位置が，水平方向でズレていることが分かります．そしてそのズレ方は「左画像における物体は，右画像では左に移動している」というものです．これは，「左画像は左目に対応する画像」「右画像は右目に対応する画像」という点を考えれば簡単です．

簡単な実験で確認しましょう．鼻先の少し前に人差し指を立ててください．そして，右目を閉じて見たときと，左目を閉じて見たときで，人差し指がどう動いたかを確認してください．左目では右寄りに，右目では左寄りに見えたはずです．この人差し指が移動した距離こそ視差であり，これをステレオ画像から求めることがステレオ・マッチングです．

●類似度計算

では，「何が，どこに動いたか」を特定する技術といえば，パターン・マッチングであり，類似度計算の出番です．この処理では最も基本的な類似度計算法であるSADを用います．SADは**図2**の通り，比較したい2つの画像（ウィンドウ）に対して，対応する画素値の差の絶対値を合計し，これを類似度として，最も似ている位置を特定します．ここで，求める値は「差の絶対値」であるため，この値が小さいほど似ている点に注意しましょう．

では，上記の「左画像における物体は，右画像では左に移動している」という点を踏まえて類似度計算を行うと，「右画像の物体は，左画像の同じ位置から，右方向に類似度計算しながら探していく」ことに

第6章
垂直成分を手掛かりとした水平エッジ付きSADによるステレオ・マッチング

ステレオ画像から距離画像を作成する場合，パターン・マッチングの精度が重要であることを前章で述べました．本章の処理では，マッチング箇所を画像特徴（垂直方向のエッジ）が強い領域に限定することで，領域は限定的ながら信頼性の高い距離画像を得ることができます．

●仕組み

この処理は，前章「SADを使用したステレオマッチング」のしきい値として，水平ソーベル・フィルタを使用する処理です．**図1**の「水平方向のソーベル・エッジ強度画像」を見てみましょう．縦方向のエッジが出ています．これは，水平ソーベル・フィルタを適用後，エッジを絶対値にしてから任意のしきい値で2値化することで得られます．

ソーベル・エッジは平滑化を伴ったエッジ検出処理のため，ノイズの影響を抑制しつつエッジを検出できます．このエッジ検出結果内でのみステレオ・マッチングを行うことで，処理を高速化しつつ信頼性の高いマッチング結果を得ることができます．

●実行結果

図2に入力となるステレオ画像と距離画像，および出力画像（距離画像とステレオ画像の半分を繋げた画像）を示します．距離画像では，ステレオ・マッチングの結果がソーベルエッジの検出結果の範囲内に限定されていることが分かります．どの程度までマッチングさせるかは，水平ソーベル・フィルタに用いるしきい値によって限定できます．しきい値を緩くすると，多くのマッチング結果が得られる一方で，

図1
水平方向のソーベル・エッジの強度（絶対値）**画像**

第7章

立体感最大のエッジ検出結果を獲得する「視差情報評価に基づくケニー・エッジ検出」

ステレオ画像からエッジ検出をする際に，やはり立体視したときの品質が重要です．そして，エッジが高周波成分から検出されることを踏まえると，元のステレオ画像が持つ視差精度をどれだけ維持できるかが重要です．この処理は視差情報評価画像を用いることで，元の高周波成分が持つ視差精度をできるだけ維持したケニー・エッジ検出結果を獲得できます(1)．

●仕組み

この処理は，ケニー・エッジ検出で使用するしきい値2つを変えていきながら画像にケニー・エッジ検出法を適用し，そのエッジ検出結果の視差精度を視差情報評価画像で評価し，その評価結果が最大となるケニー・エッジ検出結果を出力します．

視差情報評価画像とは，元のステレオ画像が持つ視差と，画像処理後の持つ視差を比較することで，画像処理結果を3項目で評価する処理です．ただし，ここで用いる視差精度とは，ステレオ画像の高周波成分が持つ視差情報が基準となります．

3項目とはそれぞれ，

- G（緑色）：良好な立体感を維持した領域
- B（青色）：立体感が損失した領域
- R（赤色）：不正な立体感が出現した領域

であり，この3項目の割合を画像や数値で出力できます．本処理では，G－Rを評価値として用いて，その値が最大（最も立体感の品質が高い）となる2値化結果を出力します．

●実行結果

図1に入力となるステレオ画像と，出力画像となる立体感最大のケニー・エッジ検出結果を示します．普通に目視しても，エッジ検出結果の左右を見比べると，おおむね同等のエッジ検出結果になっていることが分かります．もし，左右で異なるエッジを検出してしまうと，それは不正な視差となって正しい立体視ができません．そのため，こうして視差情報評価画像を使って左右の視差情報を元の高周波成分に近付けることが重要となります．

また，今回はケニー・エッジ検出のしきい値を画像全体から求めていますが，画像をブロックに分割して適用することで，局所的なしきい値を求めたより高品質な結果が得られます(1)．

第8章
カメラとフレーム間差分を使用した動き検知

カメラを使ったリアルタイム処理の代表と言えば，やはり動き検知ではないでしょうか．閉店後の貴金属店に強盗が押し入り，翌日になって事態を把握した店員が警察に通報する，そのような事態が起きたときの監視カメラ映像を報道番組で見ることがあります．

カメラ映像から侵入者を検知するには，高度なAI技術が必要になると諦める方も居るかも知れません．ですが，環境を限定すれば「フレーム間差分」で簡単かつ安価に実現できます．

●仕組み：フレーム間差分

カメラの映像は動画であり，動画はフレームという静止画の集まりでできています．例えば，アニメ映像は1秒間に何枚もの絵（フレーム）を高速で切り替えることで実現していますが，それと同様です．ここで動きとは，フレーム間に生じた変化と捉えることができるため，フレーム間差分によって検出できます．

●検知結果

図1に結果を示します．ここではフレーム間差分が変化したとき，その画素を白画素で表示するようにしています．その結果，カメラ映像に変化がないときは何も現れませんが，動きがあると白い領域が出現しています．

しかし，システムの環境によってはカメラ映像に動画ノイズが生じ，動きのない映像でもフレーム間差分に変化を生じることがあります．**図2**を見てみましょう．微細な変化でも検知する設定，つまり，**図2**（**a**）のしきい値0では，静止状態でも動画ノイズを誤検知してしまいます．そのため，「画素値がどのぐらい変化したらノイズではなく動きであるか」を判定するしきい値が重要になります．この変化というのは画素値の変化です．

図2（**b**）では，しきい値を50，つまり画素値に50以上の変化が生じた場合，その画素を動きとして白色で表示するように設定しています．しきい値50では静止時には検知されず，動いた時に検知されていることが分かります．反対に，しきい値0では，静止時でも動いた時でも検知され，区別がつきにくくなっています．

●検知の数値化

フレーム間差分としきい値によって動きを検知できるようになりました．ここで動きを数値化してみます．ここでは，動きとして検知した画素の数を表示します．これによって，次の段階として，無視しても良い動きと無視できない動きの区別もできるようになります．**図2**では，カメラに生じた微弱な変化

第1章

領域抽出結果のノイズ評価

領域抽出は代表的な画像処理です．現代社会では人物抽出／文字抽出／車両抽出など，挙げればキリがないほど領域抽出技術が使われています．数多く存在するそれらがどのぐらいの精度を持つのか，優劣をどうつければ良いのか，そしてなにより，手法を改良するためにも評価する方法が必要です．

1-1　適合率

性能を評価する方法の1つに適合率（Precision）があります．適合率は，領域抽出結果にどのぐらいノイズ（不要な領域）が少ないかを示す指標です．抽出結果にノイズが含まれているほど評価値は低くなり，ノイズが少ないほど評価値が高くなります．適合率は領域抽出の代表的な性能評価法ですが，統計学や機械学習，AI分野にも登場する重要な概念です．

●仕組み

適合率を計算するために必要となる画像は2つです．1つは評価対象となる領域抽出システムが出力した出力画像と，もう1つは理想的な領域抽出結果を示す正解画像です．この出力画像が正解画像に近いほど高い評価になります．ここで，「近い」の考え方は複数あり，適合率はいかに正解と無関係な領域が少ないか，つまり，いかにノイズが少ないかが指標になります．

図1に正解画像と出力画像との例を示します．正解画像は「あ」と白色の領域で表現されています．一方で出力画像の方を見てみると，2つの問題点が見当たります．1つは「あ」の領域に抽出漏れがあること

図1
適合率の計算に必要な画像

（a）正解画像　　（b）出力画像

第2章

画質の評価方法

本章では，画質劣化を評価する*RMSE*，ノイズの割り合いを評価する*PSNR*，画像の輝度/彩度/色相の*RMSE*といった画質の評価方法について説明します．

2-1 *RMSE*

●画質劣化の評価

画像は圧縮や伝送を行うと，色の劣化やノイズの発生により画質が変化することがあります．そのため，画像の圧縮法や伝送法を開発/改善する過程では画像の定量的な画質評価は必須です．ここで説明する*RMSE*（Root Mean Squared Error）は代表的な画質評価法です．オリジナルの画像を正解としたとき，評価対象画像の画素値が平均的にどのぐらい変化したかをユークリッド距離で計算します．これにより，元からどのぐらい画質が変化したかを評価できます．

●仕組み

*RMSE*を計算するために必要となる画像は2つです．1つは理想的な画質となるオリジナルの正解画像で，もう1つは評価対象となる画質が変化した出力画像です．*RMSE*はこの出力画像が正解画像に近いほど高い評価になります．ここで定義する「近い」とは，BGRの画素値の近さであり，*RMSE*出力画像の画素と正解画像の画素のユークリッド距離が指標になります．

図1に正解画像と2種類の出力画像の例を示します．正解画像は無劣化の赤ちゃんの画像で，出力画像は2種類用意しています．1つは強いJPEG圧縮をかけた画像であり，部分拡大を見ると凸凹としたブロック状のノイズが生じていることが分かります．もう1つはガウス雑音を付加した画像であり，部分拡大を見ると砂のようなノイズが発生していることが分かります．

ここで，*RMSE*の計算式は次に示す通りです．

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(G_i - O_i)^2}$$

iは画素位置であり，本来は(x, y)座標として2次元で管理する変数を1次元で管理している変数だと考えてください．画像の左上から画素を順番に1，2，3…とカウントしていくイメージです．そしてG_iは正解画像のi番目の画素値であり，O_iは出力画像のi番目の画素値です．Nは正解画像または出力画像の画素数です．この2つの画素値の平均的なユークリッド距離（**図2**）を誤差として求めるのが*RMSE*です．従って，画質が全く変化しない場合の*RMSE*は0となり，最大（最悪）の値は255を取ります．

第3章

動画への応用

画質評価は本書で紹介した通り数多くの方法がありますが，それらを応用して動画の画質評価もできます．本章では，画質の評価を動画にも適用する方法について説明します．

■静止画ごとに評価すればいい

図1に示す通り，そもそも動画はフレームという静止画の集まりであり，それらを高速（1秒間あたり15～60程度，単位はfps）に切り替えながら表示しています．従って，例えば30fpsの動画を1秒間分だけ画質評価したいとすると，それは30枚の静止画を画質評価することとほぼ同じです．

■評価方法…グラフ化して統計量を求める

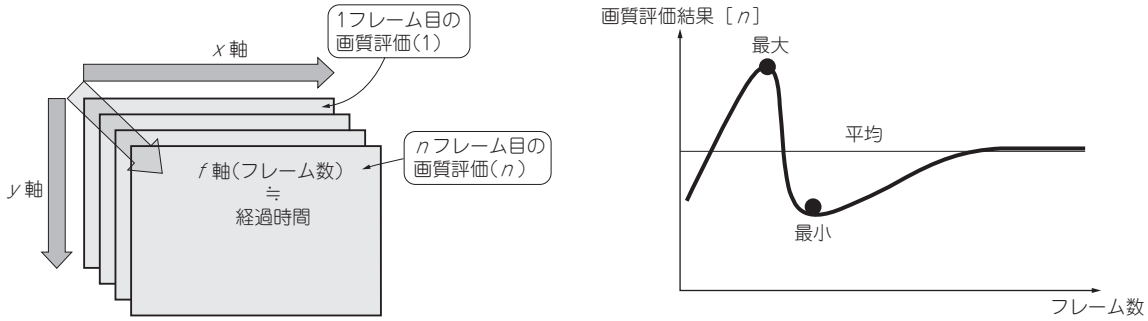
上記の考え方で動画の画質評価を行うと，評価結果はフレームの枚数分出ることになります．これらをそのまま動画の評価結果とするのは少々乱暴です．なにせ30fpsで30分の動画なら，54000枚分の評価結果となります．従って，これらは横軸にフレーム数，縦軸を画質評価とする評価グラフにまとめ，そこから統計量を求めるのが良いでしょう．

●代表的な統計量

以降では，幾つか代表的な統計量を示しておきます．

▶平均値

動画における平均的な画質を知ることができます．代表的な統計量であり，動画評価結果として何か1つだけ採用するならこの値が候補になります．



（a）動画はフレーム（静止画）の集まり　　（b）評価グラフ化し統計量を求める

図1　動画はフレーム単位で画質評価を行い統計量を求めることで画質評価ができる

ISBN978-4-7898-3148-2
C3055 ￥3000E
CQ出版社

定価 3,300円（本体3,000円）⑩

9784789831482

1923055030008

画像処理シリーズ