

IMAQ™

IMAQ Vision コンセプトマニュアル

インターネットサポート

サポート電子メール：supportjapan@ni.com

電子メール：infojapan@ni.com

FTPサイト：ftp.ni.com

日本語ホームページ：<http://www.ni.com/jp>

電話サポート（日本）

Tel：03-5472-2981

Fax：03-5472-2977

海外オフィス

イスラエル 03 6393737、イタリア 02 413091、インド 91 80 535 5406、英国 01635 523545、オーストラリア 03 9879 5166、オーストリア 0662 45 79 90 0、オランダ 0348 433466、カナダ（オタワ）613 233 5949、カナダ（カルガリー）403 274 9391、カナダ（ケベック）514 694 8521、カナダ（トロント）905 785 0085、カナダ（モントリオール）514 288 5722、韓国 02 3451 3400、ギリシャ 01 42 96 427、シンガポール 65 6 226 5886、スイス 056 200 51 51、スウェーデン 08 587 895 00、スペイン 91 640 0085、スロベニア 3 425 4200、台湾 02 2528 7227、中国 86 21 6555 7838、チェコ 02 2423 5774、デンマーク 45 76 26 00、ドイツ 089 741 31 30、ニュージーランド 09 914 0488、ノルウェー 32 27 73 00、フィンランド 09 725 725 11、フランス 01 48 14 24 24、ベルギー 02 757 00 20、ブラジル 55 11 3262 3599、ポーランド 22 3390 150、ポルトガル 210 311 210、香港 2645 3186、マレーシア 603 9596711、南アフリカ 11 805 8197、メキシコ 001 800 010 0793、ロシア 095 238 7139

National Instruments Corporation

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

日本ナショナルインスツルメンツ株式会社

〒105-0011 東京都港区芝公園 2-4-1 秀和芝パークビル A 館 4F Tel：03-5472-2970

サポート情報の詳細については、付録 B 「技術サポートおよびプロフェッショナルサービス」を参照してください。本書に対するご意見は、techpubs@ni.com まで電子メールでお送りください。

必ずお読みください

保証

限定的保証：National Instruments Corporation（以下「NI」という）のハードウェア製品は、NIがお客様に製品を出荷した日（以下「配送日」）から次の一定期間、素材及び製作技術上の欠陥に対して保証されています。すなわちIEEE 488に未対応のハードウェア製品については1年間、IEEE 488対応のハードウェア製品については2年間、ケーブルについては90日間の保証が適用されます。ソフトウェア製品の場合は、該当するNIのライセンス条項に基づき、お客様にライセンスが供与されます。配送日から90日間は、NIのソフトウェア製品（但しNIのハードウェア製品に正しくインストールされている場合）について、(a)付属のマニュアル文書に従い実質的に機能すること、および(b)ソフトウェア製品が記録されている媒体は、通常の利用やサービスにおいて素材及び製作技術上の欠陥を有しないこと、が保証されています。ライセンスが供与されたソフトウェア製品の交換については、当初の保証期間の残存期間または30日間のいずれか長い期間について保証されます。お客様が保証期間中の製品をNIに返却するには、事前にNIから返品確認（Return Material Authorization: RMA）番号を取得してください。また、修理・交換品をお客様からNIへ、NIからお客様へ返送する送料は、お客様の負担になります。返却された製品を検査、試験した後、同製品には欠陥がないとNIが判断した場合、その旨をお客様に通知します。同製品の返送にかかる費用はお客様に負担いただき、試験にかかった費用については後日請求致します。製品の不具合が事故、乱用、誤用、お客様による不適切なキャリブレーションによって発生した場合や、お客様が当該NIソフトウェアと共に使用することが予定されていない第三者のソフトウェアと共に利用した場合、不適切なハードウェアまたはソフトウェアのキーを利用した場合、独断で保守または修理を行った場合、本書に定める限定的保証は無効となります。

救済方法：上記の限定的保証において、NIの唯一の義務（およびお客様の唯一の救済方法）は、NIの選択により、支払われた料金の返還、または欠陥製品の修理・交換に限定されます。ただし、NIが、当該製品に適用される保証期間内に、こうした欠陥について書面で通知を受け取った場合に限りです。お客様は、訴訟原因の発生から1年を超えて経過した後は、上記の限定的保証に基づく本救済方法を強制するために訴訟を提起することはできません。

返品および解約に関する方針：お客様は、不要な製品については、配送日から30日以内であれば、当該製品を返却することができます。この場合の送料はお客様にご負担いただきます。上記30日間満了後は不要な製品の返品は受け付けません。特殊機器または特殊なサービスが係わる場合、お客様は、進行中の関連作業全てに対して責任を負うものとします。ただし、お客様から書面による解約の通知を受領した場合、NIはただちに損害を軽減するための責任ある対策を講ずるものとします。製品の返却の際は、NIから返品確認番号を取得してください。お客様がNIに対して行った説明・表示等が虚偽または誤解を生じさせるものであった場合には、NIは注文を取り消すことがあります。

本書の内容については万全を期しており、技術的内容に関するチェックも入念に行っております。技術的な誤りまたは乱丁・落丁につきましては、お客様への事前の通告なく、NIにて次の版から修正する権利があるものとします。本書で誤りと思われる箇所については、NIにご確認ください。NIは、本書およびその内容により、またはそれに関連して発生した損害に対して一切責任を負いません。

本書に規定する保証を唯一の保証とします。NIは、明示・暗示を問わず、ここに記載された以外の保証は行いません。特に、商品適合性の保証や特定用途に対する適合性についての保証は行いません。NIの過失または不注意により発生した損害に関するお客様の賠償請求権は、お客様が製品に支払われた金額を上限とします。NIは、データの消失、利益の逸失、製品の使用から生じた損失や、付随的または結果的に生じた損害に対して、その損害が発生する可能性を通知されていた場合でも、一切の責任を負いません。かかるNIの限定的責任は、訴訟方式、過失責任を含む契約上の責任または不法行為責任を問わず適用されます。NIに対する訴訟は、訴訟原因の発生から1年以内に提起する必要があります。NIは、NIが合理的に支配可能な範囲を超えた原因により発生した履行遅延に関しては一切の責任を負いません。所有者が、NIの指示通りインストール、操作、保守を実施しないことにより発生した損害、欠陥、誤作動、動作不良について、また、所有者による製品の改変、乱用、誤用、または不注意な行動、さらに停電、電源サージ、火災、洪水、事故、第三者の行為、その他の合理的に支配可能な範囲を超えた事象により発生する損害、欠陥、誤作動、動作不良については本書に定める保証の対象となりません。

著作権

著作権法に基づき、National Instruments Corporationの事前の承諾なく、複製、記録、情報検索システムへの保存および翻訳を含め、本書のすべてまたは一部をいかなる手段によっても複製または転載することを禁止します。

商標

CVI™、IMAQ™、LabVIEW™、Measurement Studio™、National Instruments™、NI™、ni.com™、NI-IMAQ™は、National Instruments Corporationの商標です。本書に掲載されている製品および会社名は該当各社の商標または商号です。

特許

National Instruments製品を保護する特許については、ソフトウェアに含まれている特許情報（ヘルプ→特許）、CDに含まれているpatents.txtファイル、またはni.com/patentsのうち、該当するリソースから参照してください。

National Instrumentsの製品を医療用に使用することに関する警告

(1) National Instruments Corporation（以下「NI」という）の製品は、外科移植もしくはそれに関連する用途、または作動不良により人体に深刻な傷害を及ぼすことが合理的に予期される生命維持装置の重要なコンポーネントとしての用途に適した信頼性のレベルでのコンポーネントや試験を採用して設計されておりません。(2) 上記用途を含む、あらゆるアプリケーションにお

いて、不利な要因によってソフトウェア製品の操作の信頼性が損なわれる可能性があります。これには、電力供給の変動、コンピュータハードウェアの誤作動、コンピュータ・オペレーティングシステム・ソフトウェアの適応性、アプリケーション開発に利用したコンパイラや開発ソフトウェアの適応性、インストールの間違い、ソフトウェアとハードウェアの互換性の問題、電子監視機器または制御機器の誤作動または故障、電気システム（ハードウェア及び/又はソフトウェア）の一時的な障害、予期せぬ使用または誤用、ユーザまたはアプリケーション設計者側のミスなどがありますが、これに限定されません（本書においてこのような不利な要因を総称して「システム故障」といいます）。システム故障が財産または人体に危害を及ぼす可能性（身体の損傷および死亡の危険を含む）があるアプリケーションにおいては、システム故障の危険があるため、単独の電気システム方式のみに依存すべきではありません。損害、人体への傷害、または死亡といった事態を避けるため、ユーザまたはアプリケーション設計者は、システム故障から保護するための合理的に慎重な対策を取る必要があります。これには、バックアップメカニズム、または非常停止メカニズムなどがありますが、これに限定されません。各エンドユーザのシステムはカスタマイズされており、NIの試験プラットフォームとは異なること、またユーザやアプリケーション設計者が、NIが評価したことのない方法や、予期しない方法でNI製品を他の製品と組み合わせて使用する可能性があることから、NI製品をシステムまたはアプリケーションに統合する場合は、ユーザまたはアプリケーション設計者が、最終的にNI製品の適合性（かかるシステムまたはアプリケーションの適切な設計、処理、安全レベルが含まれますが、これに限定されません。）の検証および確認における責任を負うものとなります。

目次

本書について

本書で使用する表記規則.....	xv
関連文書.....	xvi

第1部 ビジョンの基本概念

第1章

デジタル画像

デジタル画像の定義.....	1-1
デジタル画像のプロパティ.....	1-2
画像の解像度.....	1-2
画像の鮮明度.....	1-2
平面の数.....	1-3
画像タイプ.....	1-3
グレースケール画像.....	1-4
カラー画像.....	1-5
複素画像.....	1-5
画像ファイル.....	1-5
IMAQ Vision 画像の内部表示形式.....	1-6
画像ボーダー.....	1-8
画像マスク.....	1-10
使用目的.....	1-10
概念.....	1-10
カラースペース.....	1-13
使用目的.....	1-13
概念.....	1-14
RGB カラースペース.....	1-15
HSL カラースペース.....	1-17
CIE-Lab カラースペース.....	1-17
CMY カラースペース.....	1-18
YIQ カラースペース.....	1-18
詳細説明.....	1-18
RGB からグレースケールへ.....	1-18
RGB および HSL.....	1-19
RGB および CIE L*a*b*.....	1-20
RGB および CMY.....	1-21
RGB および YIQ.....	1-21

第 2 章 表示

画像表示	2-1
画像表示の概念	2-1
使用目的	2-1
詳細情報	2-2
表示モード	2-2
16 ビット画像表示のマッピング方法	2-3
パレット	2-4
使用目的	2-4
概念	2-5
詳細情報	2-5
グレーレベルパレット	2-5
温度パレット	2-6
虹パレット	2-7
傾斜パレット	2-7
バイナリパレット	2-8
関心領域	2-9
使用目的	2-9
ROI の概念	2-10
非破壊オーバーレイ	2-11
使用目的	2-11
非破壊オーバーレイの概念	2-11

第 3 章 システムのセットアップとキャリブレーション

イメージングシステムをセットアップする	3-1
高品質の画像を集録する	3-3
解像度	3-3
コントラスト	3-5
被写界深度	3-5
視点	3-5
歪み	3-7
空間キャリブレーション	3-7
使用目的	3-7
概念	3-8
キャリブレーションプロセス	3-8
座標系	3-9
キャリブレーションアルゴリズム	3-11
キャリブレーションの品質情報	3-12
画像補正	3-14
スケールモード	3-14
補正領域	3-15

シンプルキャリブレーション.....	3-16
座標系を再定義する.....	3-17

第 II 部 画像処理および解析

第 4 章 画像解析

ヒストグラム.....	4-1
使用目的.....	4-1
ヒストグラムの概念.....	4-2
線形ヒストグラム.....	4-3
累積ヒストグラム.....	4-3
解釈.....	4-4
ヒストグラムの目盛.....	4-4
カラー画像のヒストグラム.....	4-5
ラインプロファイル.....	4-6
使用目的.....	4-6
輝度測定.....	4-6
使用目的.....	4-6
概念.....	4-7
濃度測定.....	4-7

第 5 章 画像処理

ルックアップテーブル.....	5-1
使用目的.....	5-1
LUT 変換の概念.....	5-1
例.....	5-2
定義済みのルックアップテーブル.....	5-3
対数および逆ガンマ補正.....	5-4
指数およびガンマ補正.....	5-6
平均化.....	5-8
コンボリューションカーネル.....	5-10
概念.....	5-10
空間フィルタ処理.....	5-13
使用目的.....	5-13
空間フィルタの概念.....	5-14
空間フィルタの分類のまとめ.....	5-14
線形フィルタ.....	5-15
非線形フィルタ.....	5-27
詳細の説明.....	5-32
線形フィルタ.....	5-32
非線形プレウィットフィルタ.....	5-33

非線形ソーベルフィルタ	5-33
非線形傾斜フィルタ	5-34
ロバーツフィルタ	5-34
微分フィルタ	5-34
シグマフィルタ	5-35
ローパスフィルタ	5-35
メディアンフィルタ	5-35
N次フィルタ	5-35
グレースケールモフォロジー	5-36
使用目的	5-36
グレースケールモフォロジーの概念	5-36
収縮関数	5-37
膨張関数	5-37
収縮と膨張の例	5-37
オープン関数	5-38
クローズ関数	5-39
オープンとクローズの例	5-39
適正オープン関数	5-40
適正クローズ関数	5-40
自動メディアン関数	5-40
詳細の説明	5-41
収縮の概念および数学的演算	5-41
膨張の概念および数学的演算	5-41
適正オープンの概念と数学的演算	5-42
適正クローズの概念と数学的演算	5-42
自動メディアンの概念と数学的演算	5-43

第 6 章 演算子

概要	6-1
使用目的	6-1
演算子の概念	6-1
算術演算	6-2
論理および比較演算	6-2
例 1	6-5
例 2	6-6

第 7 章**周波数領域解析**

概要.....	7-1
使用目的.....	7-2
高速フーリエ変換の概念.....	7-3
FFT 表現.....	7-3
ローパス FFT フィルタ.....	7-6
ハイパス FFT フィルタ.....	7-9
マスク FFT フィルタ.....	7-11
詳細の説明.....	7-12
フーリエ変換.....	7-12
FFT 表示.....	7-13

第 III 部
粒子解析

概論.....	III-1
使用目的.....	III-2
粒子解析の概念.....	III-2

第 8 章
二値化

概論.....	8-1
使用目的.....	8-1
二値化の概念.....	8-2
輝度二値化.....	8-2
二値化の例.....	8-2
自動二値化.....	8-3
詳細の説明.....	8-6
自動二値化テクニック.....	8-6
クラスタ化.....	8-7
エントロピー.....	8-7
相互分散.....	8-8
メトリック.....	8-8
モーメント.....	8-9
カラー二値化.....	8-9
使用目的.....	8-9

第9章

バイナリモフォロジー

概要	9-1
構造要素	9-1
使用目的	9-1
構造要素の概念	9-2
構造要素のサイズ	9-2
構造要素の値	9-3
ピクセルフレームの形状	9-4
連結性	9-7
使用目的	9-7
連結性の概念	9-7
詳細の説明	9-9
連結性 4	9-9
連結性 8	9-9
一次モフォロジー操作	9-10
使用目的	9-10
一次モフォロジーの概念	9-10
収縮および膨張関数	9-11
オープンおよびクローズ関数	9-13
内側傾斜関数	9-14
外側傾斜関数	9-14
ヒットミス関数	9-15
細線化関数	9-17
太線化関数	9-19
適正オープン関数	9-21
適正クローズ関数	9-21
自動メディアン関数	9-22
上級モフォロジー操作	9-22
使用目的	9-22
上級モフォロジー変換の概念	9-23
ボーダー関数	9-23
穴埋め関数	9-23
ラベリング関数	9-23
ローパスフィルタとハイパスフィルタ	9-24
分離関数	9-25
スケルトン関数	9-26
セグメント化関数	9-28
距離関数	9-29
ダニエルソン関数	9-30
円形関数	9-30
凸関数	9-31

第 10 章 粒子測定

デジタル粒子	10-1
使用目的	10-1
デジタル粒子の概念	10-1
面積	10-1
長さ	10-2
詳細の説明	10-8
基本測定値の定義	10-8
誘導測定値	10-9

第 IV 部 マシンビジョン

第 11 章 エッジ検出

概論	11-1
使用目的	11-1
ゲージング	11-2
検出	11-3
アライメント	11-4
エッジ検出の概念	11-4
エッジの定義	11-4
エッジの特徴	11-5
エッジ検出メソッド	11-7
エッジ検出を二次元検索領域に拡張する	11-11

第 12 章 パターンマッチング

概論	12-1
使用目的	12-1
パターンマッチングの概念	12-3
パターンマッチングツールの機能	12-3
パターンマッチングのテクニック	12-4
既存のパターンマッチング	12-5
新しいパターンマッチングのテクニック	12-6
詳細の説明	12-8
相互相関	12-8
形状マッチング	12-10
使用目的	12-10
形状マッチングの概念	12-10

第 13 章 寸法測定

概論.....	13-1
使用目的.....	13-1
寸法測定 の概念.....	13-2
画像内で部品を検出する.....	13-2
特徴を検出する.....	13-2
測定を行う.....	13-2
測定値を判定する.....	13-3
座標系.....	13-3
使用目的.....	13-3
概念.....	13-4
詳細の説明.....	13-5
特徴または測定ポイントを検出する.....	13-9
エッジベースの特徴.....	13-9
ラインおよび円形の特徴.....	13-10
形状ベースの特徴.....	13-12
画像上で測定を行う.....	13-12
距離測定.....	13-12
解析幾何学.....	13-13

第 14 章 カラー検査

カラースペクトル.....	14-1
スペクトルの生成に使用するカラースペース.....	14-1
カラースペクトルを生成する.....	14-3
カラーマッチング.....	14-6
使用目的.....	14-6
カラー識別.....	14-6
カラー検査.....	14-7
カラーマッチングの概念.....	14-9
カラー分布の学習.....	14-9
カラー分布を比較する.....	14-9
カラーの位置検出.....	14-10
使用目的.....	14-10
検査.....	14-11
識別.....	14-12
カラーの位置検出ツールの機能.....	14-14
パターンの向きおよび複数のインスタンス.....	14-14
周辺光状態.....	14-15
ブレとノイズの状態.....	14-15
カラーの位置検出の概念.....	14-15

カラーパターンマッチング	14-18
使用目的	14-18
カラーパターンマッチングツールの機能	14-21
パターンの向きおよび複数のインスタンス	14-21
周辺光状態	14-22
ブレとノイズの状態	14-22
カラーパターンマッチングの概念	14-22
カラーマッチングとカラーの位置検出	14-23
グレースケールパターンマッチング	14-23
カラーの位置検出とグレースケールパターンマッチング	14-24

第 15 章

計測器リーダー

概要	15-1
使用目的	15-1
メータ関数	15-1
メータアルゴリズムの制約	15-2
LCD 関数	15-2
LCD アルゴリズムの制約	15-3
バーコード	15-3
バーコードアルゴリズムの制約	15-4

付録 A

カーネル

付録 B

技術サポートおよびプロフェッショナルサービス

用語集

索引

本書について

『IMAQ Vision コンセプトマニュアル』は、イメージングの経験が浅いユーザやイメージングを初めて行なうユーザがマシンビジョンや画像処理の基本概念を習得する上で役立ちます。また、このマニュアルには、上級ユーザのためのマシンビジョンや画像処理関数に関する詳細についても説明しています。

本書で使用する表記規則

本書では以下の表記規則を使用します。

<>

DBIO<3..0> などのように、省略記号によって左右に分けられた数を含む山形かっこは、ビットまたは信号名に関連付けられた値の範囲を示します。



このアイコンは、ユーザへのアドバイスを表しています。



このアイコンは、注意すべき重要な情報があることを示しています。

太字

太字のテキストは、メニュー項目やダイアログボックスなど、ソフトウェアでユーザが選択（クリック）する必要のある項目を表します。また、フロントパネル上のパラメータ名、制御器やボタン、ダイアログボックスまたはその一部、メニュー名、パレット名も表します。

下線

下線付きのテキストは、重要な事項を示します。

斜体

このフォントスタイルは変数を示します。または、ユーザが入力する必要がある語または値のプレースホルダを示します。

monospace

このフォントのテキストは、キーボードから入力する必要のあるテキストや文字、コードの一部、プログラムサンプル、構文例を表します。また、ディスクドライブ名、パス名、ディレクトリ名、プログラム名、サブプログラム名、サブルーチン名、デバイス名、関数名、演算名、変数名、ファイル名と括弧、引用するコードにも使います。ただし、日本語の文字の入力や表示は、前後の文と区別するため、「」で囲んでいる場合もあります。

関連文書

以下の文書には、このマニュアルを使用する上で役立つ情報が記載されています。

- 『IMAQ Vision for LabVIEW User Manual』
- 『IMAQ Vision for Measurement Studio User Manual』
- IMAQ Vision for LabVIEW online VI reference
- IMAQ Vision for Measurement Studio LabWindows™/CVI™ online function reference
- IMAQ Vision for Measurement Studio ActiveX Controls online function reference

第 I 部

ビジョンの基本概念

このセクションでは、デジタル画像、表示、およびシステムキャリブレーションの概念について説明します。

第 I 部「ビジョンの基本概念」には、以下の章が含まれます。

第 1 章「[デジタル画像](#)」では、デジタル画像のプロパティ、画像タイプとファイル形式、IMAQ Vision における画像の内部表現、画像ボーダー、画像マスク、およびカラースペースについて説明します。

第 2 章「[表示](#)」では、画像表示、パレット、関心領域、および非破壊オーバーレイについて説明します。

第 3 章「[システムのセットアップとキャリブレーション](#)」では、ピクセル座標を実世界座標に変換できるようにするため、イメージングシステムをセットアップしキャリブレーションする方法を説明します。

デジタル画像

本章では、デジタル画像のプロパティ、画像タイプとファイル形式、IMAQ Vision での画像の内部表現、画像ボーダー、画像マスク、およびカラースペースについて説明します。

デジタル画像の定義

画像とは、輝度を表す値の 2 次元配列です。画像処理の場合、画像という言葉はデジタル画像を指します。画像は次の輝度を関数で表したものです。

$$f(x, y)$$

ここで、 f は点 (x, y) の輝度を表し、 x と y は画素（ピクセル）の空間座標を表します。

通常、画像の左上の角に座標 $(0,0)$ があり、それがピクセルの空間座標の基準点になります。図 1-1 では、左から右へ移動するにつれ、 x の値が増え、上から下へ移動するにつれ、 y の値が増えます。

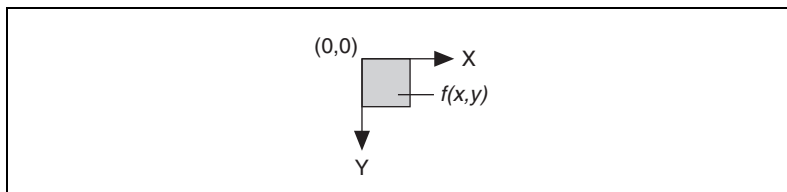


図 1-1 (0,0) ピクセルの空間座標の基準点

デジタル画像処理では、画像センサにより画像を有限数のピクセルに変換します。画像センサは、各ピクセルに数値的位置と、ピクセルの輝度や色を指定するためのグレーレベルまたはカラーの値を割り当てます。

デジタル画像のプロパティ

デジタル画像には、解像度、鮮明度、平面数という 3 つの基本プロパティがあります。

画像の解像度

画像の**空間解像度**は、ピクセルの列数 × 行数で表します。 m 列と n 行で構成された画像の場合、解像度は $m \times n$ です。つまり、その画像は横軸が m ピクセル、縦軸が n ピクセルで構成されているということになります。

画像の鮮明度

画像の鮮明度は、画像で表示できる明度の数を示します。画像の**ビット深度**は、ピクセル値のエンコードに使用するビット数を表します。ビット深度が n の場合、画像の鮮明度は 2^n で表されます。これは、ピクセルが 2^n の値を取ることができる、という意味です。たとえば、 n が 8 ビットの場合、ピクセルは、0 ~ 255 の間の 256 の値を取ることができます。 n が 16 ビットの場合、0 ~ 65,535 (-32,768 ~ 32,767) の間の 65,536 の値を取ることができます。現在、IMAQ Vision では、16 ビットの画像で、-32,768 ~ 32,767 の範囲のみサポートしています。

IMAQ Vision は、8 ビット、10 ビット、12 ビット、14 ビット、16 ビット、浮動小数点、およびカラーエンコードの画像を処理できます。画像をエンコードする方法は、画像の性質、使用する画像処理のタイプ、実行する解析のタイプによって異なります。たとえば、画像でオブジェクトの形状情報を取得する場合、8 ビットのエンコードで十分です。ただし、画像の輝度、または画像の領域を詳細に測定する必要がある場合は、16 ビット、または浮動小数点のエンコードを使用してください。

マシンビジョンまたは画像処理アプリケーションが、検査または解析するオブジェクトのカラーに依存している場合は、カラーエンコードされた画像を使用します。

IMAQ Vision は、他のタイプの画像エンコードを直接サポートしていません。特に、1 ビット、2 ビット、4 ビットの画像としてエンコードされた画像には対応していません。この場合、IMAQ Vision は、画像ファイルを開くと、その画像を 8 ビットの画像 (IMAQ Vision の最小ビット深度) に自動変換します。

平面の数

画像における平面の数は、その画像を構成するピクセルの配列数に対応しています。グレースケール画像や擬似カラー画像は、1つの平面で構成されていますが、TrueColor 画像は、赤、青、緑の各成分に対する3つの平面で構成されています。

TrueColor 画像の場合、ピクセルのカラー成分の輝度は、3つの異なる値にエンコードされています。カラー画像は、RGB 画像の赤、緑、青の各成分に対応する3つのピクセルの配列を組み合わせたものです。HSL 画像は、色相、彩度、輝度の値で定義します。

画像タイプ

IMAQ Vision ライブラリは、グレースケール画像、カラー画像、複素画像という3種類の画像を処理することができます。IMAQ Vision は、これらの3種類の画像をサポートしていますが、ある画像タイプでは、一部の操作を実行できません（たとえば、論理演算子の AND を複素画像に適用すること）。

表 1-1 は、グレースケール画像、カラー画像、複素画像が使用する1ピクセル当たりのバイト数を表しています。理想的な空間分解能としては、カラー画像の場合、8ビットのグレースケール画像のメモリ容量の4倍、複素画像の場合、8ビットのグレースケール画像のメモリ容量の8倍が必要です。

表 1-1 1ピクセル当たりのバイト数







画像タイプ	1ピクセルデータ当たりのバイト数
8ビット (符号なし) 整数のグレースケール (1バイトまたは8ビット)	 グレースケールの輝度：8ビット
16ビット (符号付き) 整数のグレースケール (2バイトまたは16ビット)	 グレースケールの輝度：16ビット

表 1-1 1ピクセル当たりのバイト数 (続き)

<p>32 ビットの 浮動小数点の グレー スケール (4 バイトまた は 32 ビット)</p>	 <p>グレースケールの輝度：32 ビットの浮動小数点</p>
<p>RGB カラー (4 バイトまた は 32 ビット)</p>	 <p>アルファ値 (未使用)：8 ビット 赤の輝度：8 ビット 緑の輝度：8 ビット 青の輝度：8 ビット</p>
<p>HSL カラー (4 バイトまた は 32 ビット)</p>	 <p>8 ビット (未使用) 色相：8 ビット 彩度：8 ビット 明度：8 ビット</p>
<p>複素画像 (8 バイトまた は 64 ビット)</p>	 <p>実数部：32 ビットの浮動小数点 虚数部：32 ビットの浮動小数点</p>

グレースケール画像

グレーレベル画像は、1つのピクセル面で構成されています。各ピクセルは、1つの数を使ってエンコードされます。この数は、以下のいずれかになります。

- 0 ~ 255 の間のグレースケール値を表す 8 ビットの符号なし整数
- -32,768 ~ +32,767 の間のグレースケール値を表す 16 ビットの符号付き整数
- $-\infty \sim \infty$ の間で、グレースケール値を表す一つの単精度浮動小数点数 (4 バイトを使ってエンコード)

カラー画像

カラー画像は、RGB 画像または HSL 画像として、メモリにエンコードされます。カラー画像のピクセルは、4 つの値で構成されています。RGB 画像は、赤、緑、青の平面に対し、それぞれ 8 ビットで、色の情報を保存します。HSL 画像は、色相、彩度、明度に対し、それぞれ 8 ビットで、色の情報を保存します。すべてのカラーモデルで、残りの 8 ビットの値は、使用されません。これは、4×8 ビット（つまり 32 ビット）のエンコード表記法です。



複素画像

複素画像には、グレースケール画像の周波数情報が含まれています。フーリエ変換（FFT）をグレースケール画像に適用して、複素画像を作成します。グレースケール画像を複素画像に変換すると、その画像上で周波数領域の操作を実行できます。

複素画像の各ピクセルは、複素ピクセルの実数部と虚数部を表す 2 つの単精度浮動小数点値としてエンコードされています。複素画像から、実数部、虚数部、振幅、位相の 4 つの成分を抽出することができます。

画像ファイル

画像ファイルは、ヘッダとピクセル値で構成されています。ファイル形式によって、ヘッダには、水平分解能、垂直分解能、ピクセルの鮮明度、オリジナルパレットに関する画像情報が含まれています。また、画像ファイルには、キャリブレーション、パターンマッチングテンプレート、オーバーレイの情報が含まれています。以下は、一般的な画像ファイル形式です。

- **BMP** (Bitmap)
- **TIFF** (Tagged image file format)
- **PNG** (Portable network graphics)：空間補正、パターンマッチングテンプレート、オーバーレイなどの画像情報を保存することができます。

- **JPEG** (Joint Photographic Experts Group)
- ナショナルインストルメンツの画像ファイル形式 (**AIPD**) : 浮動小数点画像、複素画像、HSL 画像の保存に使用)

標準の 8 ビットグレースケール形式は、BMP、TIFF、PNG、JPEG、AIPD です。標準の 16 ビットグレースケール形式は、PNG と AIPD です。RGB 画像向けの標準のカラーファイル形式は、BMP、TIFF、PNG、JPEG、AIPD です。標準の複素画像のファイル形式は、PNG と AIPD です。

IMAQ Vision 画像の内部表示形式

図 1-2 は、IMAQ Vision 画像がシステムメモリでどのように表されるかを示しています。保存されている画像には、画像のピクセルの他に、画像ボーダーや左 / 右側のアラインメントと呼ばれる行や列も含まれています。ピクセルの近傍操作を行う特定の処理関数で、画像ボーダーを使用します。アラインメント領域では、画像の最初のピクセルがメモリで調整された 8 バイトであることを確認します。アラインメントブロックのサイズは、画像の幅とボーダーのサイズによって決まります。画像を調節することによって、処理速度を 30% も上げることができます。

ライン幅は、画像の水平方向の総ピクセル数です。ライン幅は、水平分解能、画像ボーダー、左右のアラインメントを足したものです。水平分解能が 8 バイトの倍数、画像ボーダーのサイズが 0 の場合、水平分解能とライン幅は、同じ長さである場合もあります。

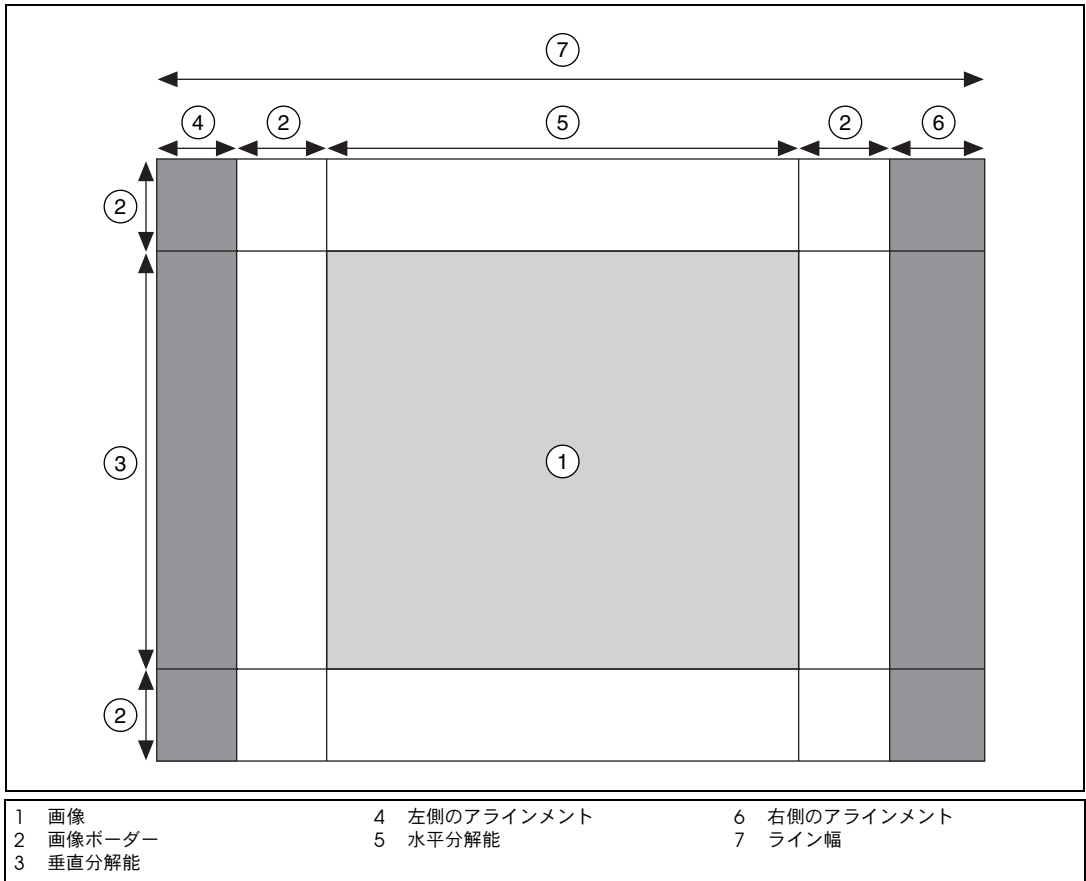


図 1-2 画像の内部表記

画像ボーダー

多くの画像処理関数は、その近接ピクセルの値を使って、ピクセルを処理します。**近接ピクセル**とは、画像処理の際、その値が近隣のピクセルの値に影響するピクセルを指します。画像のエッジに沿ったピクセルは、4つの側面すべてにおいて、近接ピクセルを持つわけではありません。近接ピクセルの値に基づいてピクセルを処理する関数を使う必要がある場合は、境界にあるピクセルを考慮に入れるため、画像を取り囲む**画像ボーダー**を指定します。画像ボーダーを指定するには、そのサイズとピクセル値を指定します。

ボーダーのサイズは、使用している関数で必要となる最大の近接ピクセルに対応している必要があります。近接ピクセルのサイズは、2D 配列のサイズで指定します。たとえば、関数が処理の際にピクセルの 8 つの近接ピクセルを使う場合、近傍のサイズは、 3×3 (3 つの列と 3 つの行からなる配列) で表すことができます。ボーダーサイズは、2D 配列の列数または行数の半分を、最も近い整数値に丸めた値以上に設定します。たとえば、関数が 3×3 の近傍を使用する場合、画像のボーダーサイズは、最低 1 になります。関数が 5×5 の近傍を使用する場合、画像のボーダーサイズは、最低 2 になります。IMAQ Vision では、画像はデフォルトのボーダーサイズの 3 で作成されます。これは、最大 7×7 の近傍を使う関数をサポートします。

IMAQ Vision では、画像ボーダーのピクセル値を指定する方法として、3 種類あります。図 1-3 は、その方法を表しています。図 1-3a は、画像のピクセル値を表しています。図 1-3b で示すように、すべての画像ボーダーのピクセル値をゼロに設定することができます (これはデフォルトの場合)。図 1-3c で示すように、画像のエッジ上のピクセルの値をボーダーピクセルにコピーすることができます。あるいは、図 1-3d のように、画像のエッジ上のピクセル値の鏡像をボーダーピクセルにコピーすることもできます。

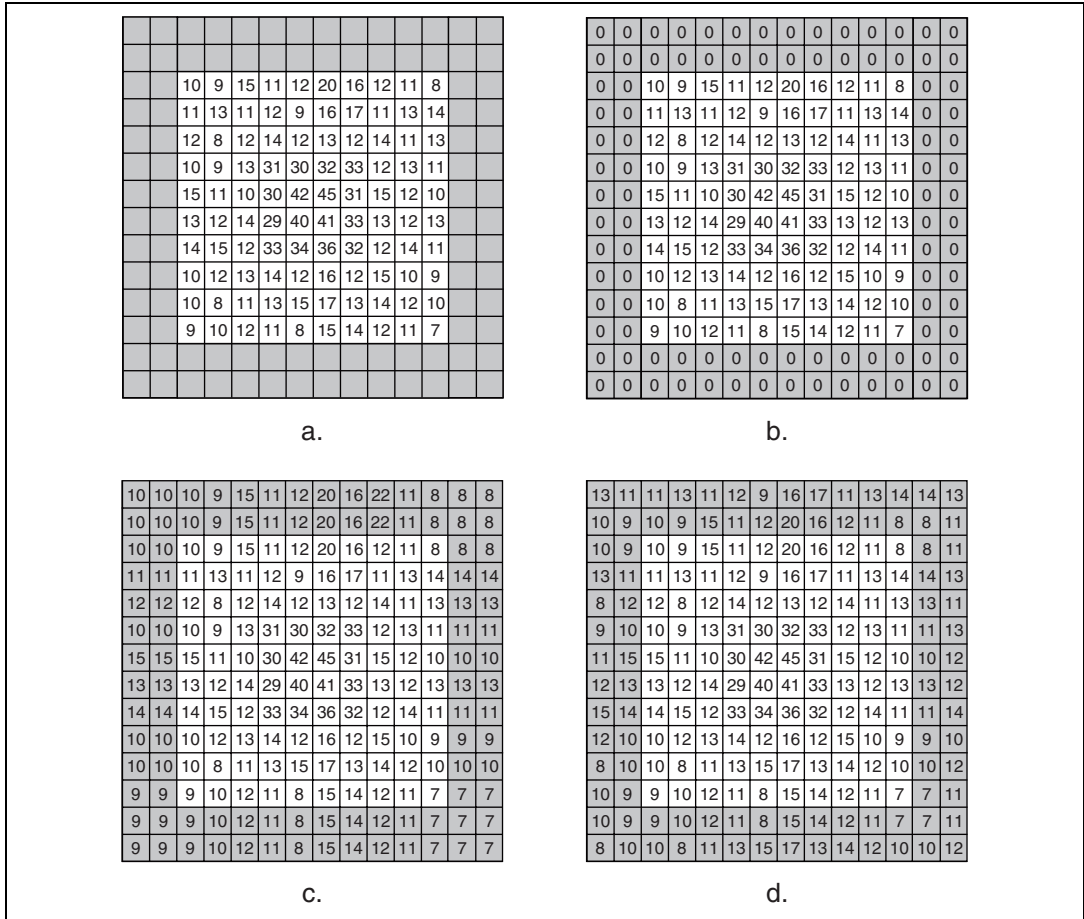


図 1-3 画像ボーダーのピクセル値を設定する

ボーダーピクセルを指定するのにどの方法を使うかは、アプリケーションで必要となる処理関数によって異なります。ボーダーを指定する方法を選択する前に、どのように関数が動作するかを調べてください。どれを選ぶかによって、処理結果に大きく影響します。たとえば、ピクセルと近接ピクセルの違いを元に、画像のエッジを検出する関数を使う場合、ボーダーピクセル値をゼロに設定しないでください。図 1-3b に示すように、ゼロを含む画像ボーダーでは、ボーダーのピクセル値とボーダー上の画像ピクセル値にかなりの違いがあり、これによって、関数は画像のボーダー上で誤ったエッジを検出してしまいます。エッジ検出関数を使う必要がある場合、より高精度の結果を得るには、ボーダー上のピクセル値またはその鏡像をボーダーの領域にコピーしてください。

IMAQ Vision において、近接ピクセルを使う画像処理関数の多くは、近傍を使って、画像ボーダーにピクセル値を自動設定します。グレースケールフィルタ処理のローパス、N 次、およびエッジ検出は、鏡像コピーで画像ボーダーにピクセルを設定します。バイナリモフォロジー関数、グレースケールモフォロジー関数、およびセグメンテーション関数は、ボーダー上のピクセル値をボーダー領域にコピーします。相関関数、円形関数、ボーダー拒否関数、粒子削除関数、スケルトン関数、およびラベル関数では、ボーダーにおけるピクセル値をゼロに設定します。



メモ 画像のボーダーは、処理の際にのみ考慮されます。ボーダーは、表示されることもファイルに保存されることもありません。

画像マスク

画像マスクは、処理用に画像の一部を分離します。関数に画像マスクのパラメータがある場合、関数の処理または解析は、元の画像と画像マスクの両方に依存します。

画像マスクは、検査画像と同じかそれ以下のサイズの 8 ビットのバイナリ画像です。画像マスクのピクセルによって、検査画像の対応するピクセルを処理するかどうかが決まります。画像マスク内のピクセルがゼロ以外の値の場合、検査画像内の対応するピクセルが処理されます。画像マスクのピクセルがゼロの場合、検査画像上の対応するピクセルは処理されません。

使用目的

画像の一部を処理または検査したい場合に、画像マスクを使います。

概念

元の画像のピクセルは、画像マスクで対応するピクセルがゼロ以外の値の場合に処理されます。図 1-4 は、画像におけるピクセル値を反転する関数の出力に、マスクがどう影響するかを表しています。図 1-4a は、検査画像です。図 1-4b は、画像マスクを表しています。図 1-4c は、画像マスク

を使って、検査画像を反転した様子を表しています。図 1-4d は、画像マスクを使わない場合の、検査画像を反転した様子を表しています。

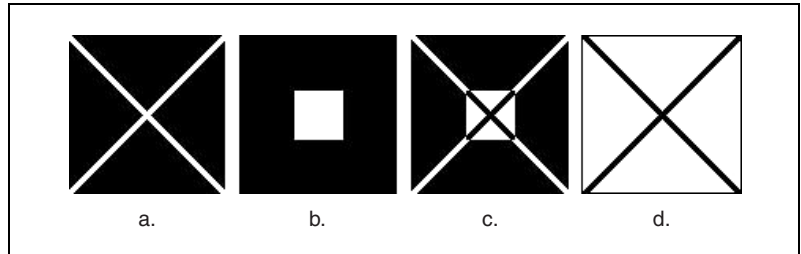


図 1-4 画像マスクの効果

画像マスクを適用する範囲を、処理対象領域の四角形の枠に制限することができます。この方法を使用すると、画像の重要な情報を含む部分にのみ画像マスクを制限することで、メモリを節約することができます。元の画像におけるこの**関心領域 (ROI)**の位置を追跡するため、IMAQ Vision は**オフセット**を設定します。オフセットによって、元の画像の座標の中で画像マスクの起点を配置する位置が設定されます。

図 1-5 は、画像マスクを適用する他の方法を示しています。図 1-5a は、画像マスクを適用する ROI を示しています。図 1-5b は、検査画像と同じサイズの画像マスクを示しています。この場合、オフセットは (0,0) に設定されています。画像マスクは、図 1-5c のように、ROI の四角形の枠と同じサイズにもできます。この場合、オフセットによって参照画像内のマスク画像の位置が指定されています。検査画像の別の領域にマスク画像を適用するには、このオフセットを定義します。

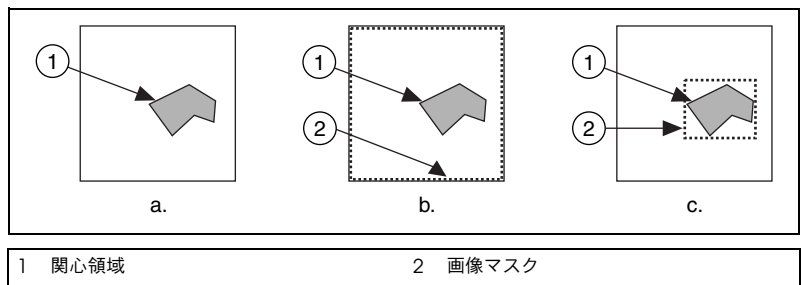


図 1-5 オフセットを使用した画像マスクの制限

図 1-6 は、2つの異なるオフセットを使ったマスクの利用法を示しています。図 1-6a は、検査画像を表し、図 1-6b は、画像マスクを表しています。図 1-6c と図 1-6d は、オフセットがそれぞれ (0, 0) と (3, 1) の場合に、画像マスクを使った関数の結果を示します。

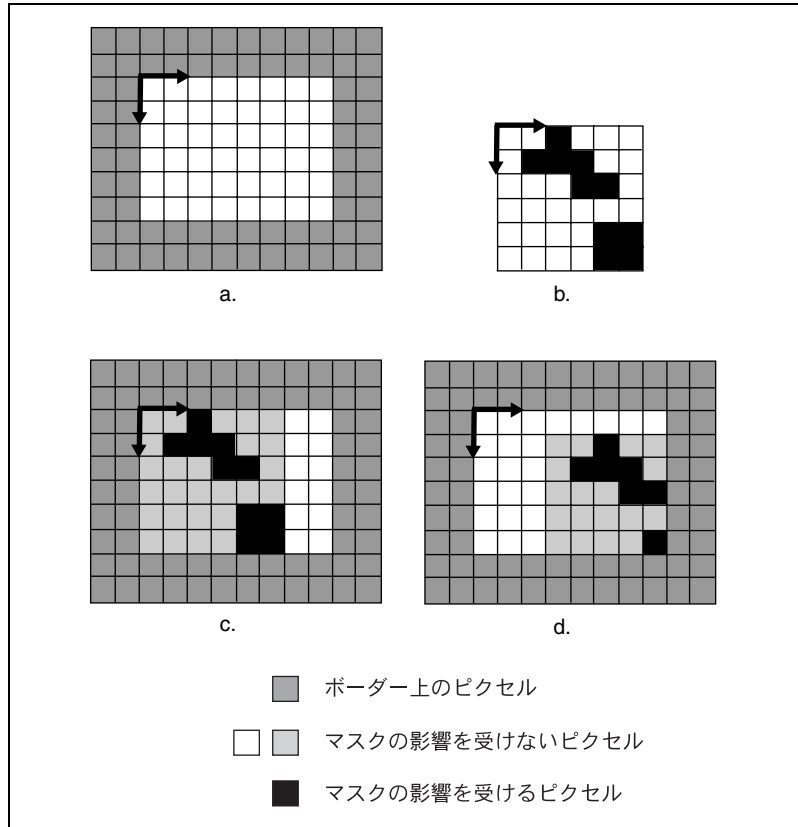


図 1-6 異なるオフセットでマスクを適用した場合の効果

ROIの詳細については、第2章、「表示」の「[関心領域](#)」を参照してください。

カラースペース

色を表現するには、カラースペースを使用することができます。カラースペースとは、各色をポイントで表すことができる 3 次元の座標システムのサブスペースです。カラースペースを使って、人、マシン、ソフトウェアプログラムの間での色の記述を容易にすることができます。

様々な業界やアプリケーションで、異なるカラースペースを使用しています。人間は、輝度、彩度、色相などのパラメータに従って色を認識する一方、コンピュータは、赤、緑、青の組み合わせとして色を認識します。印刷業界では、シアン、マゼンタ、イエローを使って、色を指定しています。以下は、一般的なカラースペースのリストです。

- RGB：赤、緑、青が基準。コンピュータで画像を表示するのに使用されています。
- HSL：色相、彩度、明度が基準。画像処理アプリケーションで使用されています。
- CIE：輝度、色相、色彩が基準。人間の脳が認識する色の感覚として、Commission Internationale de l'Eclairage (国際照明委員会)により定義されています。
- CMY：シアン、マゼンタ、イエローが基準。印刷業界で使用されています。
- YIQ：色の情報 (I と Q) から明度の情報 (Y) を分離したものの。テレビ放送で利用されています。

使用目的

カラー画像を処理する際は、常にカラースペースを指定する必要があります。IMAQ Vision では、画像の作成時に画像に関連するカラースペースを指定します。IMAQ Vision は、RGB と HSL カラースペースをサポートしています。

カラーマシンビジョンアプリケーションの実行中に、照明の状態が大きく変化すると予測される場合は、HSL カラースペースを使用します。HSL カラースペースは、カラーマッチング、カラー検出、カラーパターンマッチングなどのカラー処理関数を実行する際、RGB スペースよりも精度の高いカラー情報を提供します。様々な照明やノイズ条件下で実行する、IMAQ Vision のカラー処理用の上級アルゴリズムは、HSL カラースペースで画像を処理します。

アプリケーションの実行中に照明の状態が大きく変化することがなく、さらに赤、緑、青を使って使用したい色を簡単に指定できる場合は、RGB スペースを使用します。また、アプリケーションでカラー画像を表示するだけで処理しない場合は、RGB スペースを使用してください。RGB スペースは、ユーザが表示したいと思うおりの画像を再生します。IMAQ Vision は、常に RGB スペースでカラー画像を表示します。HSL スペースで画像を作成した場合、IMAQ Vision は、画像を表示する前に RGB スペースに自動変換します。

カラー画像の使い方の詳細については、第 14 章、「[カラー検査](#)」を参照してください。

概念

カラーは、特定の視覚的な刺激に対する脳の反応であるため、人間の脳が認識するさまざまな色の感覚として記述するのが最もよい方法です。目の網膜にある色感知細胞は、赤、緑、青の光に対応する 3 つの帯域を使って色をサンプリングしています。これらの細胞からの信号は、脳に送られ、そこで組み合わせられて異なる色彩感覚が生成されます。国際照明委員会は、以下の感覚を定義しています。

- 輝度：領域が示す光の量の感覚
- 色相：赤、緑、青の組み合わせに似た領域の感覚
- 色彩：領域が示すように見える色相の量の感覚
- 明度：画像内での白基準に対する領域の輝度の感覚
- クロマ：画像内での白基準に対する領域の色彩
- 彩度：輝度に対する領域の色彩

三原色説では、3 つの異なる光（赤、緑、青）の組み合わせによってどのように色を表現するかを説明しています。これは、人間の目が 3 種類の色のセンサを持っている、ということを基本にした理論です。印刷や写真では、この三原色説に基づき、3 つの異なる染料を組み合わせ、色を再現しています。コンピュータのカラースペースでも、3 つのパラメータを使って色を定義しています。

多くのカラースペースは、カラーモニターやプリンタなどのハードウェアを使って画像を表示したり、コンピュータグラフィックスや画像処理などの色情報処理アプリケーションで利用したりするためのものです。カラー CRT モニタ、多くのカラービデオカメラやコンピュータグラフィックスシステムでは、RGB カラースペースを使用しています。HSL スペースは、RGB や YIQ と組み合わせ、画像処理など、色を処理するアプリケーションでよく利用されています。カラー写真の出版業界では、シアン、マゼンタ、イエロー (CMY) カラースペース、つまり CMYK を使用しています。YIQ スペースは、カラー TV 放送の基準です。

RGB カラー空間

RGB カラー空間は、最も一般的に使用されているカラー空間です。人間の目は、錐体（人間の目にある色の受容器）を通じて、赤、緑、青の成分の色情報を受信します。これらの 3 つの色を加法三原色と呼びます。加法混色法によると、人間の脳は、3 つの基本的な光源を処理し、それを組み合わせて、1 色の「画像」を生成します。三原色の各成分を組み合わせて、ほとんどすべての色を生成できます。

図 1-7 に示すように、RGB 空間は、各軸の端に赤、緑、青がある三次元の立方体として視覚化することができます。黒が立方体の基点にあり、白がその立方体の反対側の角にあります。立方体の各側面には、0 と 1 の間の値があります。RGB の立方体の各軸に沿って、その成分がまったく入っていない色から完全な飽和色までの範囲になっています。その立方体内の任意のポイント、つまり色は、3 つの数字（R、G、B の 3 つの組み合わせ）で指定します。黒 (0,0,0) から白 (1,1,1) までの立方体の対角線は、すべてのグレースケール値、つまり赤、緑、青の成分がすべて等しい値を表しています。コンピュータのハードウェアとソフトウェアの組み合わせによって、色の範囲は異なります。一般的な組み合わせは、各成分に対し 0-255 と 0-65,535 です。この範囲内のカラーの値を RGB 立方体の値にマップするには、カラーの値をその範囲の最大値で割ります。

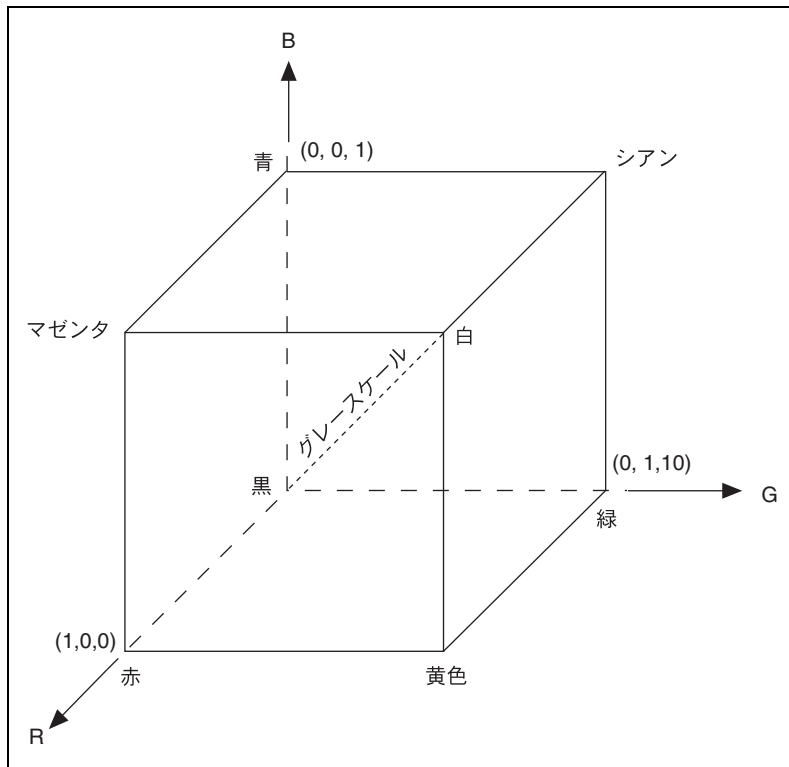


図 1-7 RGB 立方体

RGB カラー空間は、人間が認識できる範囲内にあります。つまり、RGB 立方体は、人間が見ることができる色よりも少ない色しか表すことができません。

RGB 空間を使用することによってコンピュータのモニタの設計が容易になりますが、すべてのアプリケーションに最適という訳ではありません。RGB カラー空間では、色を表すために赤、緑、青の成分がすべて必要です。したがって RGB は、他のカラー空間のように直感的ではありません。HSL カラー空間では、色相の成分だけを使って色を表すため、カラーマッチングなど多くの画像処理アプリケーションでは、HSL が最適ということになります。

HSL カラースペース

HSL カラースペースは、人間が色を簡単に定量化できるようにするために開発されました。色相、彩度、および輝度は、HSL スペースで 1 つの色と他の色を区別するための特性です。色相は、色の主波長に対応します。色相成分は、オレンジ、緑、紫などの色を表します。色相の範囲は虹のように視覚化することができます。彩度とは、色相に追加された白の量のことで、色の相対的な純度を表します。白が混ざっていない色は、純度 100% です。彩度の度合いは、加えられている白の光の量に反比例します。ピンク（赤と白）やラベンダー（紫と白）のような色は、赤や紫よりも純度が低い色です。輝度は、明度の色彩的な概念、つまり光の振幅（または強度）を具体化したものです。クロマは、色相と彩度の組み合わせで、クロマと輝度の関係によって色の特性を表します。色相を処理するシステムでは、HSL カラースペースを使用します。これは、HSI（色相、彩度、輝度）または HSV（色相、彩度、明度）と表すこともできます。

HSL カラースペースの座標系は、円柱形です。第 14 章の「カラー検査」の図 14-1 に示すように、色は六角の円錐形で定義されます。色相の値は、 $0 \sim 360^\circ$ の範囲になります。彩度 S の範囲は $0 \sim 1$ の間で、1 は最も純度の高い色（白が混ざっていない）を表します。また、明度の範囲もやはり $0 \sim 1$ の間で、0 は黒を表し、1 は白を表します。

つまり、色情報からの輝度の分離と、クロマと人間による色の認識の関係という、2 つの主要な要素により、マシンビジョンのアプリケーション開発には HSL スペースが最適です。

CIE-Lab カラースペース

CIE カラースペースシステムでは、人間の視覚システムに従って色を区分しています。このシステムは、CIE 座標で色を指定し、CIE 座標の 1 つの色と他の色を比較するための基準となります。

CIE 1976 $L^*a^*b^*$ は、CIE ベースのカラースペースの 1 つで、色の違いの認識度を線形化する方法です。 L^* 、 a^* 、 b^* の非線形の関係は、目の対数的反応を模倣したものです。

CMY カラースペース

CMY は、シアン、マゼンタ、イエローというもう一つの身近な原色の組み合わせです。CMY は減法カラースペースで、白の光からこれらの原色を引いて希望の色を生成します。CMY カラースペースは、ほとんどのカラー印刷や写真処理で基本となっています。シアン、マゼンタ、イエローは、赤、緑、青の補色であるため、CMY は RGB カラースペースの補色です。

YIQ カラースペース

YIQ スペースは、カラー TV 放送用に、National Television System Committee (NTSC) で採用された基本のカラースペースです。通信の効率を上げ、モノクロの TV 基準との互換性を維持するため、RGB 立方体を線形変換したものです。YIQ システムの Y 成分は、モノクロ TV に必要なすべてのビデオ情報を提供します。画像処理に YIQ スペースを使用する主な利点は、明度情報 (Y) がカラー情報 (I と Q) から分離される点です。明度は、人間の目が識別する光の量に比例しているため、画像のグレースケールに修正を加えても、色の情報には影響しません。

詳細説明

RGB をグレースケールに変換したり、1つのカラースペースを他のカラースペースに変換する標準的な方法があります。RGB からグレースケールへの変換は線形です。ただし、一部のカラースペースは、他のカラースペースで表すことができない色を表現するため、1つのカラースペースから他のカラースペースに変換した場合、非線形になる場合があります。

RGB からグレースケールへ

以下の式を使用して、ピクセルベースで RGB 画像をグレースケール画像に変換します。

$$\text{グレースケール値} = 0.299R + .587G + 0.114B$$

この式は、明度の NTSC 基準の一部です。RGB をグレースケールに変換するには、平均値を計算するという方法もあります。

$$\text{グレースケール値} = (R + G + B) / 3$$

RGB および HSL

RGB カラー空間を HSL カラー空間に変換するための行列演算はありません。以下の式は、RGB カラー空間を HSL カラー空間にマップする非線形の変換を説明しています。

$$L = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

$$V2 = \sqrt{3} \times (G - B)$$

$$V1 = 2 \times R - G - B$$

$$H = 256 \times \tan^{-1}(V2 / V1) / (2 \times \Pi)$$

$$S = 255 \times (1 - 3 \times \min(R, G, B) / (R + G + B))$$

以下の式は、HSL カラー空間を RGB カラー空間にマップするためのものです。

$$h = H \cdot \frac{2\pi}{256}$$

$$s = S/255$$

$$s' = (1 - s)/3$$

$$f(h) = (1 - s \cdot \cos(h) / \cos(\pi/3 - h)) / 3$$

$$\left. \begin{array}{l} b = s' \\ r = f(h) \\ g = 1 - r - b \end{array} \right\} [0 < h \leq 2\pi/3]$$

$$\left. \begin{array}{l} h' = h - 2\pi/3 \\ r = s' \\ g = f(h') \\ b = 1 - r - g \end{array} \right\} [2\pi/3 < h \leq 4\pi/3]$$

$$\left. \begin{array}{l} h' = h - 4\pi/3 \\ g = s' \\ b = f(h') \\ r = 1 - g - b \end{array} \right\} [4\pi/3 < h \leq 2\pi]$$

$$l = 0.299.r + 0.587.g + 0.114.b$$

$$l' = l/l$$

$$R = r.l'$$

$$G = g.l'$$

$$B = b.l'$$

RGB および CIE L*a*b*

RGB を CIE L*a*b* に変換する前に、RGB の値を中間のカラー空間 (CIE XYZ カラー空間) に変換する必要があります。以下の 3×3 の行列で、RGB を CIE XYZ に変換します。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

次に、以下の式を使用して、CIE XYZ の値を CIE L*a*b* の値に変換します。

$$Y/Y_n > 0.008856 \text{ の場合は、 } L^* = 116 \times (Y/Y_n)^{1/3} - 16$$

$$\text{または、 } L^* = 903.3 \times Y/Y_n$$

$$a^* = 500 \times (f(X/X_n) - f(Y/Y_n))$$

$$b^* = 200 \times (f(Y/Y_n) - f(Z/Z_n))$$

$$\text{ここで、 } t > 0.008856 \text{ の場合は、 } f(t) = t^{1/3}$$

$$\text{または、 } f(t) = 7.787 \times t + 16/116$$

ここで、 X_n 、 Y_n 、 Z_n は、白基準の 3 刺激値です。

CIE L*a*b* の値を RGB に変換するには、まず以下の式を使って、CIE L*a*b* の値を CIE XYZ に変換します。

$$X = X_n \times (P + a^* / 500)^3$$

$$Y = Y_n \times P^3$$

$$Z = Z_n \times (P - b^* / 200)^3$$

$$\text{ここで、 } P = (L^* + 16) / 116$$

次に以下の行列演算を使って、CIE XYZ の値を RGB の値に変換します。

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

RGB および CMY

以下の行列演算で、RGB カラー空間を CMY カラー空間に変換します。

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

この変換式を使う前に、すべてのカラー値が 0 と 1 の間になるように正規化します。CMY の値から RGB の値を得るには、1 から個々の CMY 値を引きます。

RGB および YIQ

以下の行列演算で、RGB カラー空間を YIQ カラー空間に変換します。

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

以下の行列演算で、YIQ カラー空間を RGB カラー空間に変換します。

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.956 & 0.621 \\ 1.0 & -0.272 & -0.647 \\ 1.0 & -1.105 & 1.702 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

表示

本章では、画像表示、パレット、関心領域、および非破壊オーバーレイについて説明します。

画像表示

画像の表示は、データの視覚化を可能にするため、ビジョンアプリケーションでは重要なコンポーネントです。[画像処理](#)と[画像の視覚化](#)は、まったく別の要素です。画像処理とは、画像の作成、集録、および解析のことをいいます。画像の視覚化とは、画像データの表示方法および表示された画像との対話方法のことです。一般的な画像アプリケーションでは、表示ウィンドウの数よりはるかに多い数の画像をメモリ内で使用します。

画像表示の概念

表示関数は、外部画像ウィンドウへの画像の表示、ウィンドウの属性の設定、画像ウィンドウへのカラーパレットの割り当て、ウィンドウのクローズ、および外部ウィンドウでの画像ブラウザのセットアップや使用などを行います。[関心領域 \(ROI\)](#) 関数 (表示関数のサブセット) の中には、ROI を外部画像ウィンドウ内で対話式に定義するものがあります。それらの ROI 関数は、異なる描画ツールの構成と表示、ウィンドウでの描画イベントの検知、ウィンドウ上の描画された領域に関する情報の回収、および ROI の移動や回転を行います。非破壊オーバーレイでは、画像ピクセルの値を変更せずに、画像上に重要な情報を表示することができます。

使用目的

表示関数を使用すると、画像データの視覚化、画像ウィンドウからのイベントと関連データの回収、画像からの ROI の対話式選択、および画像への追加情報の添付などを行うことができます。

詳細情報

このセクションでは、IMAQ Vision で利用できる表示モードおよび 16 ビットグレースケール表示のマッピング方法について説明します。

表示モード

画像表示における重要なコンポーネントの 1 つに、ビデオアダプタが動作する表示モードがあります。表示モードは、表示画面上でピクセルの色を指定するビット数を示します。一般に、ビデオアダプタで使用できる表示モードは、ビデオアダプタ上の利用可能なビデオメモリの量と選択した画面解像度により異なり、ピクセルに対して 8 ビットから 32 ビットの範囲になります。

8 ビット表示モードの場合、ピクセルは 256 色のいずれか 1 色になります。16 ビット表示モードの場合、ピクセルは 65,536 色のいずれか 1 色になります。24 ビットまたは 32 ビット表示モードでは、画面上のピクセルの色はそれぞれ 3 バイトまたは 4 バイトでエンコードされます。それらのモードでは、情報はピクセルの赤、緑、および青の各成分に対し 8 ビットを使用して格納されます。また、それらのモードでは約 1670 万色を表示することが可能です。

IMAQ Vision で異なる画像タイプを画面に表示する方法について理解するためには、表示モードについて理解することが重要です。画像処理関数では、しばしばグレースケール画像を使用します。表示画面のピクセルは赤、緑、および青の成分で構成されているため、グレースケール画像のピクセルを直接表示することはできません。

24 ビットまたは 32 ビットの表示モードでは、表示アダプタは 8 ビットを使ってグレースケール値をエンコードし、256 色のグレイを表示可能にします。このカラー解像度で、8 ビットのグレースケール画像を表示することができます。ただし、16 ビットグレースケール画像など、ビット深度の高い画像は、24 ビットまたは 32 ビットの表示モードでは正確に表示されません。16 ビットグレースケール画像を表示するには、最下位ビットを無視するか、マッピング関数を使って 16 ビットを 8 ビットに変換します。

16 ビット画像表示のマッピング方法

次のテクニックは、IMAQ Vision で 16 ビット画像を 8 ビット画像に変換し、マッピング関数を使ってそれを表示する方法を示すものです。マッピング関数は、16 ビット画像のダイナミックレンジを 8 ビット画像に均等に分散します。

- フルダイナミック：16 ビット画像の最小輝度値を 0 に、最大輝度値を 255 にマップします。画像内の他のすべての値は、次の方程式を使って 0 から 255 の間になるようにマップされます。このマッピングメソッドは、画像のすべてのダイナミックレンジの表示が確実にできるため、汎用で用いられています。画像内の最大ピクセル値と最小ピクセル値を使用してその画像の全体のダイナミックレンジを決定するため、最大値や最小値にノイズの多いピクセルや欠損ピクセル（クラス A 以外のセンサの場合）があると、表示される画像の見た目に影響します。IMAQ Vision では、デフォルトでこのテクニックを使用します。

$$z = \frac{x-y}{v-y} \times 255$$

ここで、z は 8 ビットピクセル値

x は 16 ビット値

y は最小輝度値

v は最大輝度値です。

- 90% ダイナミック：累積ヒストグラムの 5% に相当する輝度が 0 に、累積ヒストグラムの 95% に相当する輝度が 255 にマップされます。0 ~ 5% の範囲の値が 0 にマップされ、95 ~ 100% の範囲の値が 255 にマップされます。このマッピングメソッドは、フルダイナミックメソッドよりも堅牢で、画像内の小さな乱れまで捉えることはありません。このメソッドでは、累積ヒストグラムの算出またはヒストグラムの推定値が必要です。ヒストグラムの詳細については、第 4 章、「[画像解析](#)」を参照してください。
- ユーザ定義パーセント範囲：このメソッドは 90% ダイナミックメソッドと似ていますが、ソフトウェアによってマップする累積ヒストグラムの最大および最小パーセント値は、ユーザが定義します。
- ユーザ定義範囲：このテクニックはフルダイナミックメソッドと似ていますが、0 と 255 にマップする最小値と最大値は、ユーザが定義します。このメソッドを使用すると、画像内のある領域の最大値と最小値を割り出してその領域のヒストグラムを計算することにより、その領域のコントラストを強調することができます。その領域のヒストグラムには、ピクセルの最大輝度と最小輝度が表示されます。それらの値は、画像全体のダイナミックレンジを拡張するのに使用します。

- ダウンシフト：このテクニックはピクセル値のシフトがベースになっています。このメソッドは、16 ビットピクセル値に特定の数の右シフトを適用し、最下位バイトを表示します。このテクニックを使用すると、下位ビットのいくつかは切断され、表示されません。このメソッドは非常に高速ですが、これを使用するとセンサの実ダイナミックが 8 ビットセンサの機能に低下します。また、使用している画像センサのビット深度に関する知識が必要です。たとえば、12 ビットカメラで集録した画像を、そのカメラで集録した最上位 8 ビットで表示するには、4 つの右シフトを使って視覚化する必要があります。IMAQ 画像集録デバイスを使用している場合、Measurement & Automation Explorer ではデフォルトでこのテクニックを使用しません。

パレット

画面にグレースケール画像が表示されると、IMAQ Vision は画面に表示された対応ピクセルに対して、画像内の各ピクセルの値を赤、緑、および青の輝度に変換します。このプロセスでは、**パレット**というカラーテーブルを使用します。カラーテーブルは、画像の各グレースケール値と色を関連付けます。IMAQ Vision では、8 ビットグレースケール画像を表示するために使用するパレットをカスタマイズすることができます。

使用目的

パレットを使用すると、ピクセルデータを変えることなく、画像をさまざまな視覚的表現で表示することが可能です。パレットでは、フォトネガティブ表示やカラーコード表示などの各種効果を生成することができます。カラーコード表示では、色の総数に制限のある特定の画像構成要素を詳細に表現するのにパレットが便利です。

画像をさまざまなパレットに表示することにより、特定の輝度の領域を強調したり、グレーレベルのゆるやかな変動や急激な変動を識別したり、またグレースケール画像では認識が困難な細部を表示したりすることができます。たとえば、人間の目は暗いところよりも明るいところの方が輝度のわずかな変化を認識しやすくなっています。カラーパレットを使用すると、そのようなわずかな変化を見分けやすくなります。

概念

パレットとは、定義済みまたはユーザ定義の RGB 値配列です。パレットは、各グレーレベル値に対して対応するカラー値を定義し、ピクセルを表します。ピクセルのグレーレベル値は、表に指標付けされたアドレスとして機能し、赤、緑、および青 (RGB) の各輝度に対応する 3 つの値を返します。その RGB 値によって、赤、緑、青がさまざまな割合で混合され、その値の範囲のカラー表現を生成するパレットが定義されます。

8 ビットグレースケール画像の場合、ピクセルは 0 から 255 までの 2^8 、つまり 256 の値をとることができます。カラーパレットは 256 の RGB 要素から構成されています。3 つのカラー成分 (赤、緑、青) のそれぞれに 0 から 255 の値を適用することにより、特定の色が作成されます。赤、緑、および青の成分がまったく同じ値を持つ場合、グレーレベルのピクセル値になります。

グレーパレットは、階調の異なるグレーを各明度に関連付けて、黒から白までのグレーの連続的な線形グラデーションを生成します。パレットを設定する際、黒を明度 0 に、白を 255 に割り当てるか、その逆にすることができます。他のパレットには、赤から青、薄茶から濃茶などの線形または非線形グラデーションを指定することができます。

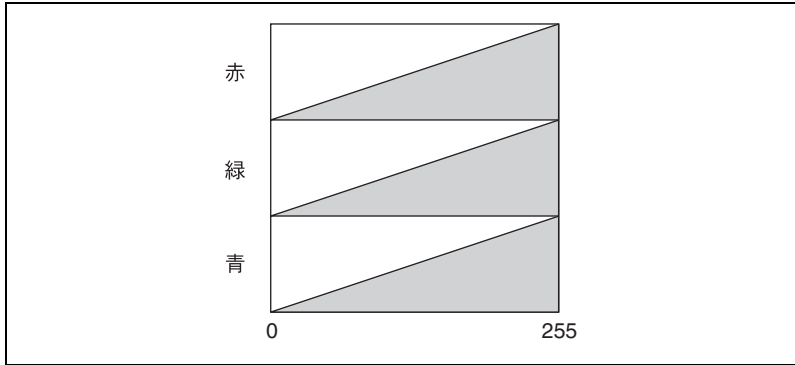
IMAQ Vision には、5 つの定義済みのカラーパレットがあります。各パレットは、異なる階調のグレーを表します。

詳細情報

以下のセクションでは、IMAQ Vision で使用できる定義済みのパレットについて説明します。各セクションのグラフは、それぞれのパレットで使用するカラーテーブルを表します。グラフの横軸は入力グレーレベル範囲 (0, 255) を、縦軸はそれぞれのグレーレベル値に割り当てられた RGB 輝度を表します。

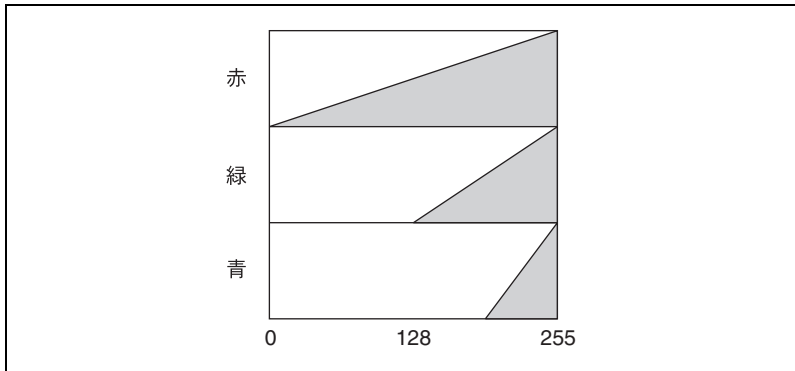
グレーレベルパレット

このパレットは、黒から白へのさまざまなグレーレベルがグラデーションになったものです。各値が同じ量の赤、緑、および青に割り当てられ、グレーレベルを形成しています。



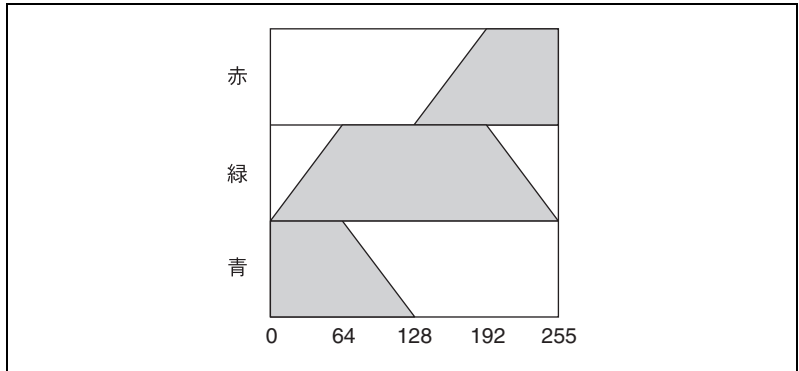
温度パレット

このパレットは、薄茶から濃茶へのグラデーションです。0 が黒で 255 が白になっています。



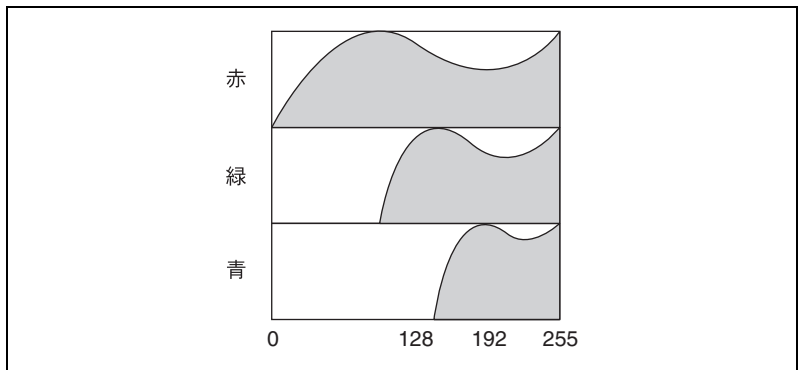
虹パレット

このパレットは、赤から青のグラデーションになっていて、その中間に際立った緑の領域があります。0 が青で 255 が赤になっています。



傾斜パレット

このパレットは、赤から白のグラデーションになっていて、その中間に際立った明るい青の領域があります。0 が赤で 255 が白になっています。

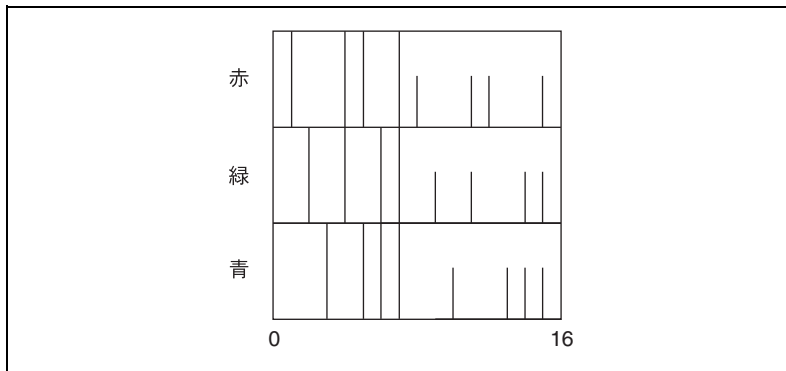


バイナリパレット

このパレットは、異なる 16 色が 16 サイクルになっています。表 2-1 では、 g がグレーレベル値を示しています。

表 2-1 バイナリパレットのグレーレベル値

$g =$	R	G	B	生成される色
0	0	0	0	黒
1	255	0	0	赤
2	0	255	0	緑
3	0	0	255	青
4	255	255	0	黄色
5	255	0	255	紫
6	0	255	255	水色
7	255	127	0	オレンジ色
8	255	0	127	マゼンタ
9	127	255	0	若草色
10	127	0	255	すみれ色
11	0	127	255	空色
12	0	255	127	青緑色
13	255	127	127	ばら色
14	127	255	127	黄緑色
15	127	127	255	青紫色



この周期パレットは、バイナリ画像やラベル付き画像の表示に適しています。

関心領域

関心領域 (ROI) とは、画像の中の画像解析を行う領域のことです。

使用目的

画像の一部に対して処理や解析を行うには、ROI を使用します。楕円や四角形、フリーハンドなど、標準の輪郭を使用して ROI を指定することができます。また、以下のオプションのいずれかを実行することができます。

- 画像ウィンドウ内で ROI を構築
- ROI を画像ウィンドウ内に関連付け
- 画像ウィンドウに関連付けられた ROI を抽出
- 画像ウィンドウから現在の ROI を消去
- ROI を画像マスクに変換
- 画像マスクを ROI に変換

ROI の概念

ROI は、画像内で処理や解析を行う 1 つまたは複数の領域を示します。そのような領域は、特定の輪郭で定義します。IMAQ Vision では、次の輪郭タイプを使用できます。

表 2-2 ROI に使用できる輪郭のタイプ

アイコン	輪郭名
	ポイント
	ライン
	長方形
	回転した長方形
	楕円
	環状領域
	壊れたライン
	多角形
	フリーハンドライン
	フリーハンド

ROI は、対話式、プログラムの、または画像マスクを使って指定することができます。ROI を対話式に定義するには、ツールパレットのツールを使って表示されている画像上に ROI を描きます。ROI をプログラムのに定義したり画像マスクを使って指定する方法についての詳細は、『IMAQ Vision for LabVIEW User Manual』または『IMAQ Vision for Measurement Studio User Manual』を参照してください。

非破壊オーバーレイ

非破壊オーバーレイを使用すると、実際に画像を変更することなく、画像に重要情報の注釈を表示することができます。画像上にテキスト、ライン、ポイント、複雑な幾何学的図形、およびビットマップを重ねることができますが、その下にある画像のピクセル値は変更されず、画像の表示のみが変わります。図 2-1 は、オーバーレイを使用して画像内の各粒子の向きを表す方法を示します。

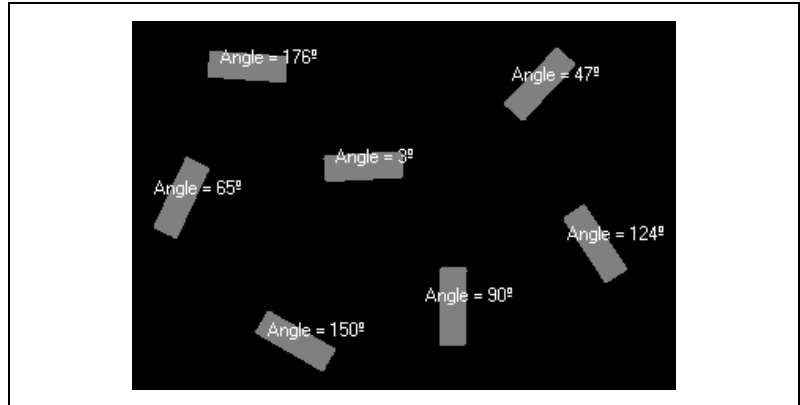


図 2-1 非破壊オーバーレイ

使用目的

非破壊オーバーレイはさまざまな目的で使用できますが、例として次のような場合に使用します。

- 画像内でオブジェクトが検出された部分をハイライトする。
- 表示されている画像に、パターンマッチング関数で得られたスコアなど、量的または質的情報を追加する。
- ルーラーグリッドやアラインメントマークを表示する。

非破壊オーバーレイの概念

オーバーレイによって、解析関数や処理関数の結果が影響を受けることはありません。表示目的のみです。オーバーレイは画像に関連付けられているため、特別なオーバーレイデータタイプというものはありません。単に画像にオーバーレイを追加するだけです。画像のサイズや向きが変わるとオーバーレイの意味がなくなるため、そのような場合 IMAQ Vision はオーバーレイをクリアします。オーバーレイ情報の入った画像は、PNG ファイル形式を使って保存することができます。

システムのセットアップと キャリブレーション

本章では、ピクセル座標を実世界座標に変換できるようにするため、イメージングシステムをセットアップしキャリブレーションする方法を説明します。ピクセル座標を実世界座標に変換することは、実世界単位を使って検査画像から正確な測定値をとることが必要な場合に便利です。

イメージングシステムをセットアップする

画像を集録、解析、および処理する前に、イメージングシステムをセットアップする必要があります。イメージングシステムには、視野、作動距離、解像度、被写界深度、およびセンササイズという5つの要素があります。図 3-1 は、その概念を表しています。

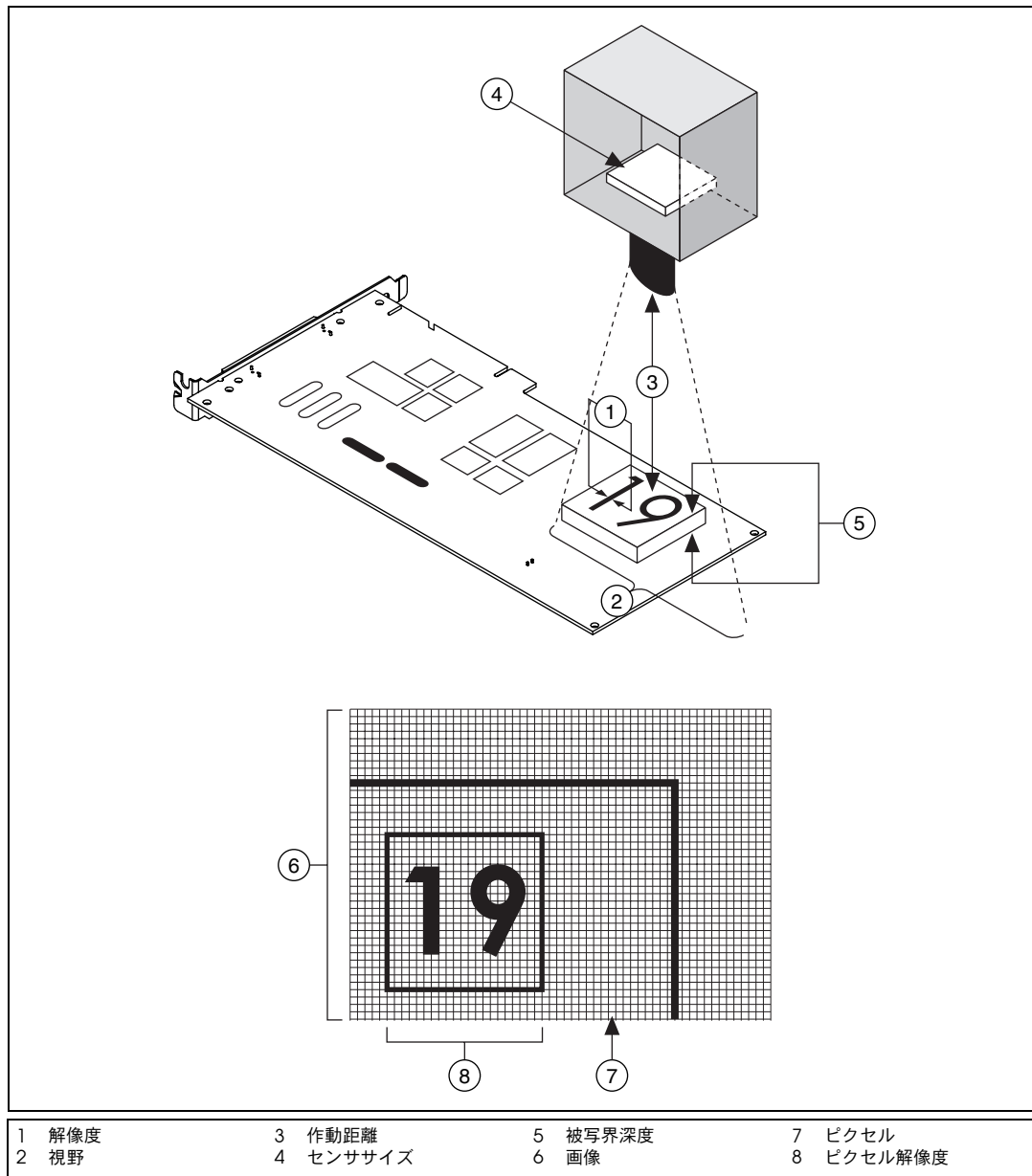


図 3-1 イメージングシステムの基本パラメータ

- 解像度：イメージングシステムによって識別可能なオブジェクトの特徴の最小サイズ。
- ピクセル解像度：検査対象のオブジェクトを表すのに必要な最小ピクセル数。
- 視野：検査対象のオブジェクトの中で、カメラが集録できる領域。
- 作動距離：カメラレンズの前面から検査対象のオブジェクトまでの距離。
- センササイズ：センサの有効領域のサイズで、通常センサの横の寸法によって決まります。
- 被写界深度：オブジェクトの焦点が合う最大深度。

イメージングシステムの基本パラメータに関する詳細は、『Edmund Industrial Optics Electronic Imaging Resource Guide』の「Application Notes」を参照するか、Edmund Industrial Optics のサイト (edmundoptics.com) をご覧ください。

高品質の画像を集録する

システムのセットアップ方法は、行う解析または処理のタイプにより異なります。画像から必要な情報を抽出するために、十分な品質の画像を生成できるイメージングシステムを構築する必要があります。全体の画質には、解像度、コントラスト、被写界深度、視点、および歪みの5つの要素が関係します。

解像度

イメージングシステムをセットアップするには、ピクセル解像度と解像度の2つの解像度について考慮する必要があります。ピクセル解像度とは、検査対象のオブジェクトを表すのに必要な最小ピクセル数のことをいいます。ピクセル解像度は、検査に必要な最小の特徴によって決まります。少なくとも2ピクセルで最小の特徴を表すようにします。次の式によって、構築するイメージングシステムに求められる最小ピクセル解像度が決まります。

$$\left(\frac{\text{オブジェクトの最長軸の長さ}}{\text{オブジェクトの最小の特徴のサイズ}} \right) \times 2$$

オブジェクトが視野全体を占めない場合、画像サイズはピクセル解像度より大きくなります。

解像度は、イメージングシステムがオブジェクトの細部をどれほど再生できるかを示します。低解像度の画像は、細部が欠如ししばしばぼやけて見えます。イメージングシステムの解像度には、視野、カメラのセンササイズ、およびセンサのピクセル数という 3 つの要素が関係します。これらの 3 つの要素がわかれば、カメラレンズの焦点距離を決定することができます。

視野

視野とは、検査対象のオブジェクトの中で、カメラが集録できる領域のことをいいます。図 3-2 は、ピクセル解像度と視野の関係を示します。

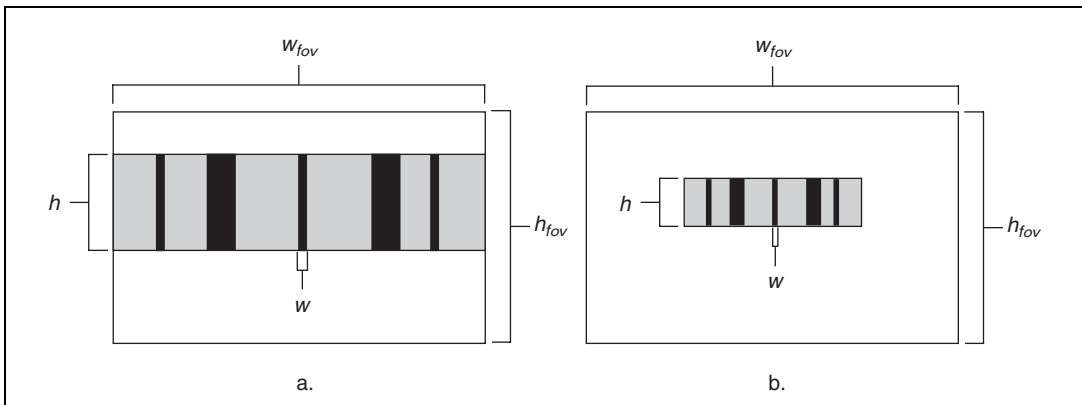


図 3-2 ピクセル解像度と視野の関係

図 3-2a のオブジェクトは、視野全体を占めています。図 3-2b のオブジェクトは、視野全体よりも少ないスペースしか占めていません。 w が最小の特徴の x 方向へのサイズで、 h が最小の特徴の y 方向へのサイズを表す場合、最小の x ピクセル解像度は

$$\frac{W_{fov}}{w} \times 2$$

となり、最小の y ピクセル解像度は

$$\frac{h_{fov}}{h} \times 2.$$

となります。2 つのうちのいずれか大きい方のピクセル解像度を選ぶようにしてください。



メモ 図 3-2b では、画像サイズはピクセル解像度より大きくなります。

センササイズとセンサのピクセル数

カメラのセンササイズは視野を決める上で重要であり、視野は解像度の最小条件を決める際の主要要素になります。センサの対角線長により、センサの有効領域が指定されます。センサのピクセル数は、ピクセル解像度以上である必要があります。カメラを選ぶ際は、解像度の最小条件を満たすものを選ぶようにしてください。

レンズの焦点距離

視野と適当なセンササイズが決定したら、次はイメージングアプリケーションに適したカメラレンズのタイプを決めます。レンズは、基本的に焦点距離で定義されます。レンズ、視野、およびセンササイズ間の関係は次のようになっています。

$$\text{焦点距離} = (\text{センササイズ} \times \text{作動距離}) / \text{視野}$$

作動距離が変更不可能な場合、レンズの焦点距離の選択肢は限定されます。作動距離が固定していて焦点距離が短いと、画像が歪んで見えることがあります。ただし、作動距離を柔軟に変更することができる場合は、適切な焦点距離を選んで歪みを最小限にできるように作動距離を調節してください。

コントラスト

解像度とコントラストは、互いに密接に関連する要素で、画質に影響を及ぼします。コントラストは、検査対象のオブジェクトと背景との輝度値の差を表します。構築するイメージングシステムには、背景とオブジェクトを区別できるだけのコントラストが必要です。正しい照明テクニックによって、システムのコントラストを高めることができます。

被写界深度

レンズの被写界深度とは、高さの異なるオブジェクトに焦点を合わせることでできるレンズの能力のことをいいます。高さの異なるオブジェクトを検査することが必要な場合は、オブジェクトが近づいたり遠ざかったりしても一定の画質を保つことができるレンズを選びます。

視点

視点エラーは、検査対象のオブジェクトに対してカメラ軸が垂直でない場合によく発生します。図 3-3a は、理想的なカメラ位置を示しています。図 3-3b は、カメラが斜めの位置からオブジェクトを撮影している状態を示しています。

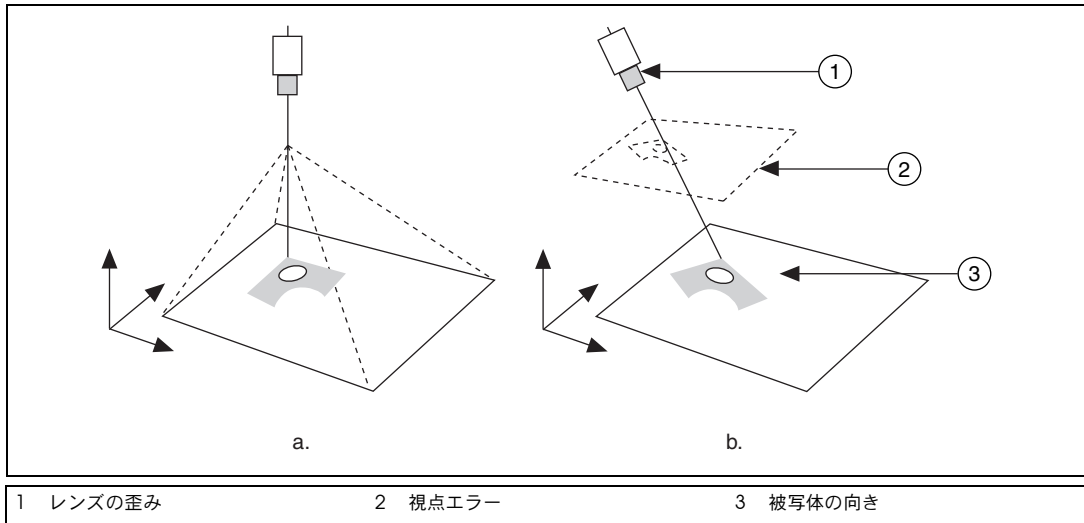


図 3-3 検査対象のオブジェクトに対するカメラアングル

視点エラーとは、オブジェクトのレンズからの距離によって、オブジェクトの倍率が変化したように見えることをいいます。図 3-4a は、点のグリッドを示します。図 3-4b は、カメラがグリッドを斜めの位置からとらえた場合の視点エラーを表しています。

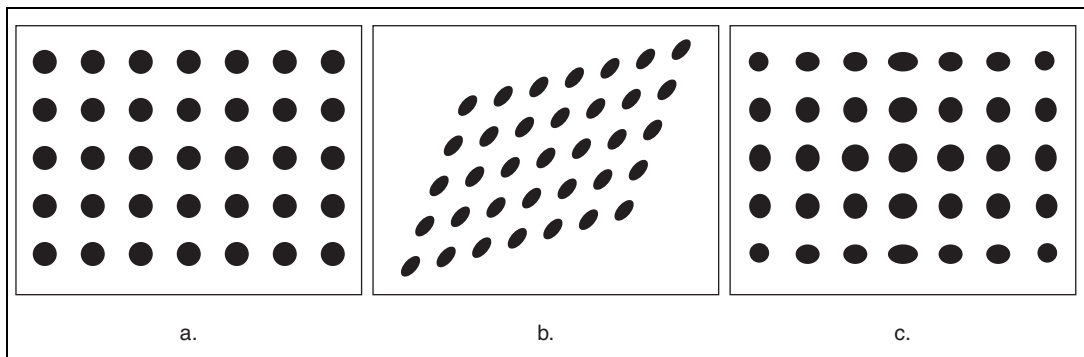


図 3-4 視点エラーと歪みエラー

視点エラーを防ぐためには、検査対象のオブジェクトに対してカメラが垂直になるように設置します。画像から正確な測定を取るためには、画像に対しキャリブレーションテクニックを適用して視点エラーを補正します。

歪み

非線形歪みは、カメラレンズの光学的誤差によって起こる幾何収差です。一般的なカメラレンズでは、放射状歪みが発生します。放射状歪みでは、レンズの光学的中心から遠い点が、中心から実際よりも遠いように写ります。図 3-4c は、点のグリッドにおける歪みの効果を示します。歪みが発生すると、画像内の情報は視野の中心に対して誤った位置に配置されますが、必ずしも失われるわけではありません。したがって、空間キャリブレーションによって画像の歪みを修正することができます。

空間キャリブレーション

空間キャリブレーションとは、イメージングセットアップにより発生した多くのエラーを補償して、ピクセルを実際に使用する単位に変換するプロセスです。実世界単位で正確な測定を得る必要がある場合は、イメージングセットアップのキャリブレーションが重要です。

画像には、情報がピクセルの形で含まれています。空間キャリブレーションを行うと、ピクセル単位の測定値をインチやセンチメートルなどの単位に変換することができます。ピクセルと実世界単位との変換率がわかっていると、この変換は簡単に行えます。たとえば、1 ピクセルが 1 インチだとすると、10 ピクセルは 10 インチになります。

ただし透視投影やレンズ歪みによりピクセル測定が影響を受けるため、この変換はそれほど簡単ではありませんこともあります。キャリブレーションでは、ピクセルと実世界単位との変換に使用できるマッピングを構築することにより、発生した可能性のあるエラーを補償します。また、キャリブレーション情報を使用して、画像表示や形状測定の視点エラーや歪みエラーを修正することもできます。

使用目的

イメージングシステムのキャリブレーションは、正確かつ確実な測定が必要なときに行います。IMAQ Vision のキャリブレーションは、以下のような場合に使用します。

- 標準パターン（キャリブレーションテンプレート）の画像化またはリファレンスポイントによるイメージングセットアップのキャリブレーション
- 測定値（長さ、面積、幅）を実世界単位からピクセル単位へ変換およびその逆

- 学習したキャリブレーションマッピングを適用し、キャリブレーション済みのセットアップによって集録された画像を修正
- 任意の座標系を割り当てて、位置を実世界単位で測定



メモ カメラ軸がオブジェクト面に対して 90° にできる限り近い場合は、視点によるエラーを無視して測定を推測することもできます。

概念

イメージングセットアップをキャリブレーションするために、キャリブレーションソフトウェアは画像内のポイントと実世界でそれに対応する位置の間で既知のマッピングを行います。キャリブレーションソフトウェアは、それらのマッピングを使用して画像全体のピクセルを実世界のマッピングに変換します。そうして得られたキャリブレーション情報は、そのマッピングを作成するために使用したイメージングセットアップのみに対して有効です。イメージングセットアップでマッピング情報に反する変更が加えられると、キャリブレーション情報の精度が低下します。

キャリブレーションプロセス

キャリブレーションソフトウェアが画像全体のキャリブレーション情報を計算するには、ピクセルから実世界単位への既知のマッピングリストが必要です。リストを指定するには、2つの方法があります。

- 図 3-5a に示すような点のグリッドを描きます。実世界単位で dx および dy の距離情報を指定します。キャリブレーションソフトウェアは、図 3-5b に示すグリッドの画像と、グリッドの点の間の距離情報を使用して、キャリブレーションプロセスに必要なピクセルから実世界単位へのマッピングのリストを生成します。
- 実世界の点のリストと対応するピクセル座標を直接キャリブレーションソフトウェアに入力します。

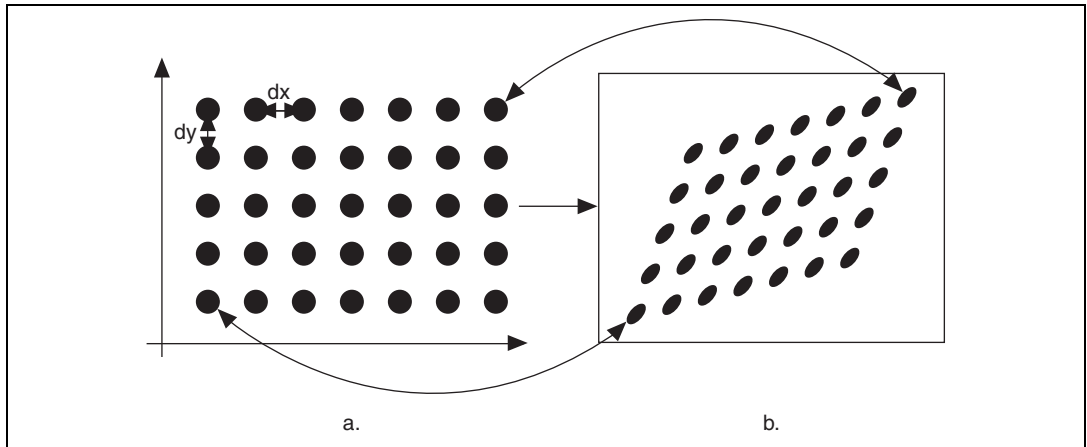


図 3-5 キャリブレーションセットアップ

キャリブレーションプロセスでは、ピクセルから実世界単位へのマッピングのリストとユーザ定義のアルゴリズムを使用して、画像全体に対するマッピングを作成します。またキャリブレーションソフトウェアは、エラーマップも生成します。エラーマップは、ピクセル座標が実世界座標に変換されたときに発生しうる最悪の場合のエラーを予測して返します。

キャリブレーションプロセスから得られたキャリブレーション情報を使用して、任意のピクセル座標を対応する実世界座標に、あるいはその逆に変換します。

座標系

測定値を実世界単位で表すには、座標系を定義する必要があります。原点、角度、および軸方向によって座標系を定義します。図 3-6a は、実世界のキャリブレーショングリッドの座標系を示します。図 3-6b は、対応するキャリブレーショングリッドの画像の座標系を示します。ピクセルで表した原点により、座標系の中心が定義されます。図 3-6 に示す座標系の原点は、円で囲まれた点の中心にあります。角度は、実際の環境での横軸に対する座標系の方向を指定します。図 3-6b で、横軸が自動的にグリッド画像の最上段の点に揃えられていることに注意してください。このキャリブレーション手順により、実世界での横軸の方向が決まります。その方向はグリッド画像の最上段の点に沿っています。

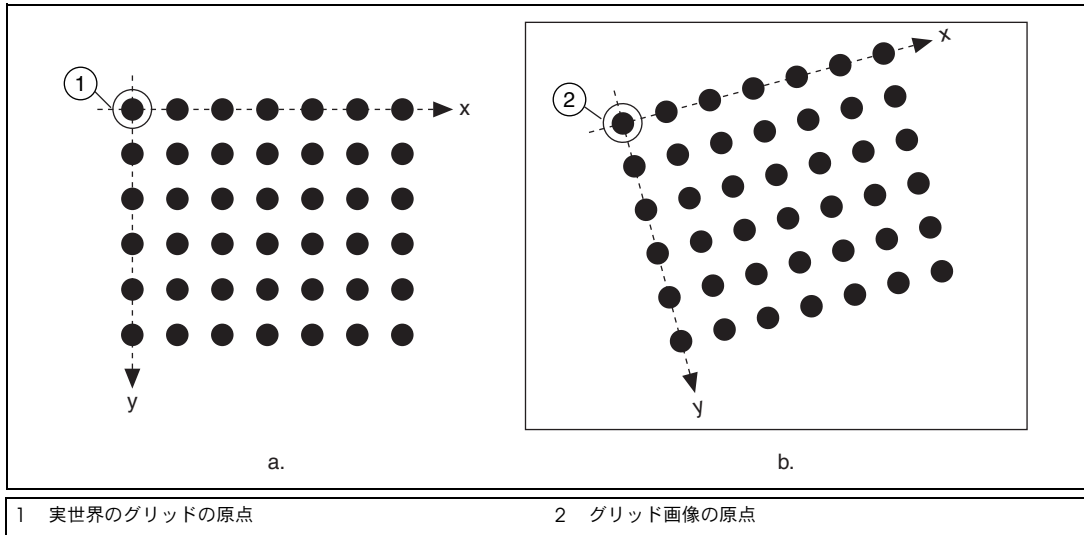


図 3-6 座標系の原点と角度

2つめの軸方向は、図 3-7a のように間接位置であるか、図 3-7b のように直接位置であるかのいずれかです。間接位置座標は、デジタル画像で座標系がどのように存在するかを示します。直接位置座標は、実世界で座標系がどのように存在するかを示します。

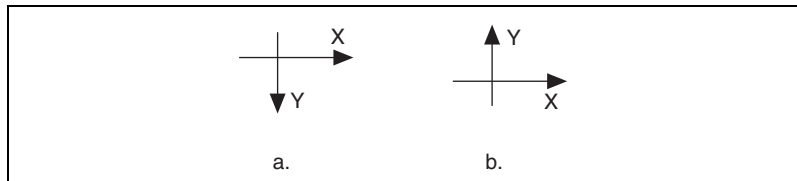


図 3-7 座標系の軸方向

座標系を指定しないと、キャリブレーションプロセスはデフォルトの座標系を次のように定義します。

1. 原点はキャリブレーショングリッドの左上の点の中心に配置されません。
2. 角度は 0 に設定されます。それにより、x 軸はグリッドの点の最上段に合わせられます。
3. 軸方向は間接に設定されます。それにより、y 軸はグリッドの点の最も左の列に合わせられます。

キャリブレーションプロセスでグリッドでなくポイントのリストを指定すると、ソフトウェアはデフォルト座標系を次のように定義します。

1. 原点はリスト内の最も低い x 座標値と最も低い y 座標値のポイントに配置されます。
2. 角度は 0 に設定されます。
3. 軸方向は間接位置に設定されます。

座標系を自分で定義する場合は、次の点に注意してください。

- 原点はピクセルで表示します。その位置を実世界単位に変換できるように、原点の位置はキャリブレーショングリッド内に入るようにします。
- 角度は新規座標系と実世界での水平方向の間の角度に指定します。使用中のイメージングシステムに視点エラーのみあってレンズの歪みはない場合、この角度は図 3-11 に示すように表示できます。ただし、非線形歪みが発生している場合は、画像内に座標系を表示するのは重要なことです。

キャリブレーションアルゴリズム

IMAQ Vision には、視点と非線形の 2 つのキャリブレーションアルゴリズムがあります。視点キャリブレーションは視点エラーを修正し、非線形キャリブレーションは視点エラーとレンズの非線形歪みを修正します。非線形歪みに比べ視点の方が速く学習されます。

視点アルゴリズムは、画像全体のピクセルから実世界単位へのマッピングを計算します。このマッピングを使用して、画像内のいずれのピクセルの座標でも実世界単位に変換することができます。

非線形アルゴリズムは、図 3-8 に示すように、キャリブレーショングリッドの各点を中心とした四角形の領域で、ピクセルから実世界単位へのマッピングを計算します。IMAQ Vision は、近接の点に基づいて、各点のマッピング情報を推測します。グリッド点のある領域内でピクセル単位を実世界単位に変換することができます。IMAQ Vision は各点の周辺のマッピングを計算するため、画像内のグリッド点のある領域のみ正確にキャリブレーションされます。

キャリブレーション関数の キャリブレーション ROI 出力は、画像内で正確なキャリブレーション情報を持つ領域を定義します。視点メソッドにおけるキャリブレーション ROI には画像全体が含まれます。非線形メソッドにおけるキャリブレーション ROI には、グリッド点の周辺の四角形領域すべてを囲む四角形の境界内が含まれます。図 3-8 は、キャリブレーション ROI の概念を表しています。

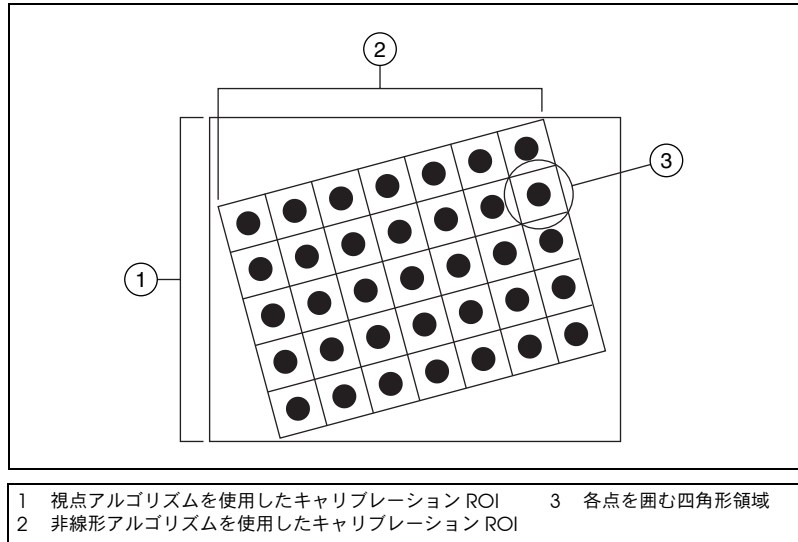


図 3-8 キャリブレーション ROI



メモ キャリブレーション ROI の外にあるピクセルを実世界単位に変換することもできますが、そのような変換は正確な結果が出ないことがあります。

キャリブレーションの品質情報

キャリブレーション関数の *quality score* および *error map* 出力は、システムに対するキャリブレーションの結果を示します。

quality score は 0 から 1000 まであり、キャリブレーション関数が与えられたグリッドまたはポイントセットを学習した度合いを示します。品質スコアはキャリブレーションの正確さを示すものではなく、キャリブレーションマッピングが学習したグリッドまたは特徴となるポイントにどの程度適合しているかを表します。

quality score を使用して、選択したキャリブレーションアルゴリズムが適切かどうかを判断します。視点メソッドを使用して強い非線形歪みを持つ画像をキャリブレーションしたり、目の粗いグリッドを使って強い非線

形歪みを持つ画像をキャリブレーションすると、IMAQ Vision は低い品質スコアを返します。

このスコアを使用すると、セットアップが予測どおりに動作しているかどうかも判定することができます。たとえば、レンズ歪みの非常に少ないレンズを使用している場合、視点キャリブレーションの結果は正確になります。ただし、キャリブレーションテンプレートが物理的に歪んでいるなど、システムセットアップに問題があると、レンズの品質にかかわらず品質スコアは低くなる場合があります。

エラーマップとは、ピクセル座標を実世界座標に変換する際に、発生する可能性のある位置エラーの予測です。エラーマップは、画像内の各ピクセルに対して予測される位置エラーを含む二次元配列です。ピクセル座標 (i, j) のエラー値は、実世界での真の位置と比べて、推定の実際の座標 (x, y) で考えられる最大の位置エラーを示します。エラー値は次の式を使って計算します。

$$e(i, j) = \sqrt{(X - X_{true})^2 + (Y - Y_{true})^2}$$

エラー値は、推定される実世界座標が存在できる真の実世界での位置からの根号距離を示します。エラー値の信頼区間は 95% で、それはつまり推定される実世界座標の位置エラーが全体の 95% 以下であることを意味します。ピクセル座標のエラー値が小さい場合、推定される実世界座標が非常に正確に計算されていることを示します。エラー値が大きい場合は、ピクセルの推定される実世界座標が不正確な可能性があることを意味します。

エラーマップを使用して、イメージングセットアップとキャリブレーション情報が検査アプリケーションに必要な精度条件を満たしているかどうかを判定します。エラー値がアプリケーションの許容可能な位置エラーを超えている場合は、イメージングセットアップに改善を加える必要があります。強いレンズ歪みのあるイメージングシステムでは、通常エラーマップの値は高くなります。かなり歪みが強いレンズを使用している場合は、エラーマップを使ってアプリケーションの精度条件に合うピクセルの位置を確認することができます。レンズの歪みの効果は画像の縁に近づくにつれて大きくなるため、画像の中心に近いピクセルの方が画像の縁にあるピクセルよりもエラー値は低くなります。

画像補正

画像補正とは、キャリブレーション済みのセットアップで集録された歪みのある画像を変換して、視点エラーとレンズ歪みを補正することをいいます。IMAQ Vision は、入力画像の各ピクセルに対してピクセルから実世界座標への変換を適用することにより、画像を補正します。次に、IMAQ Vision は簡単な推移変換とスケーリング変換を適用して、新しい画像に実世界座標を位置付けます。IMAQ Vision はスケーリングプロセスで補間を使用して新しい画像を生成します。

補正について学習する場合は、補正テーブルを構築することができます。補正テーブルとは、メモリに格納されたルックアップテーブルで、画像内のすべてのピクセルの実世界での位置情報を含んでいます。ルックアップテーブルを使用すると画像補正の速度は大幅に向上しますが、多くのメモリが必要で学習時間は長くなります。このオプションは、ビジョンアプリケーションで一度に複数の画像を補正したいときに使用します。

スケールモード

スケールモードは、補正された画像のスケール方法を指定します。スケールモードには、大きさに合わせてスケールと面積を維持してスケールの2つのオプションがあります。図 3-9 は、スケールモードを示しています。図 3-9a は、元の画像を示します。大きさに合わせてスケールオプションでは、図 3-9b に示すように、補正された画像は元の画像と同じ大きさの画像に収まるようにスケールされます。面積を維持してスケールオプションでは、図 3-9c に示すように、補正された画像は画像内の物体の面積が元の画像のときと同じになるようにスケールされます。面積を維持するようにスケールされた画像は、通常元の画像より大きくなります。面積を維持してスケールすると画像のサイズが大きくなるため、関数の処理時間も長くなります。

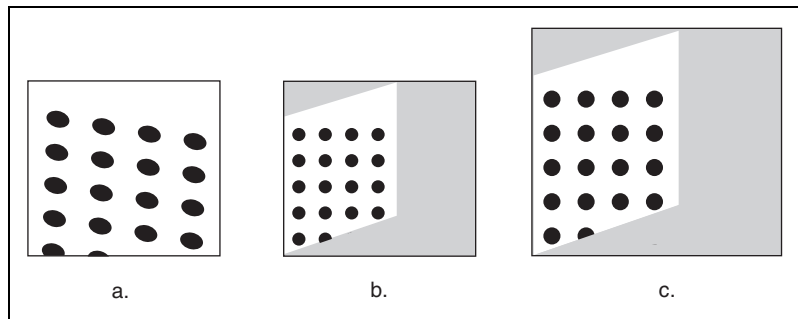


図 3-9 スケールモード

どちらのスケールモードを選ぶかは、アプリケーションにより異なります。ビジョンアプリケーションで画像内のオブジェクトの実際の面積が必要な場合は、面積を維持してスケールを選択します。その他のすべてのアプリケーションでは、大きさに合わせてスケールを使用します。

補正領域

ユーザ定義の ROI またはキャリブレーションソフトウェアが定義するキャリブレーション ROI に基づいて、画像全体または画像内の領域を補正することができます。図 3-10 は、補正用に指定できるさまざまな領域を示します。IMAQ Vision は、指定した領域のみのキャリブレーション情報を学習します。

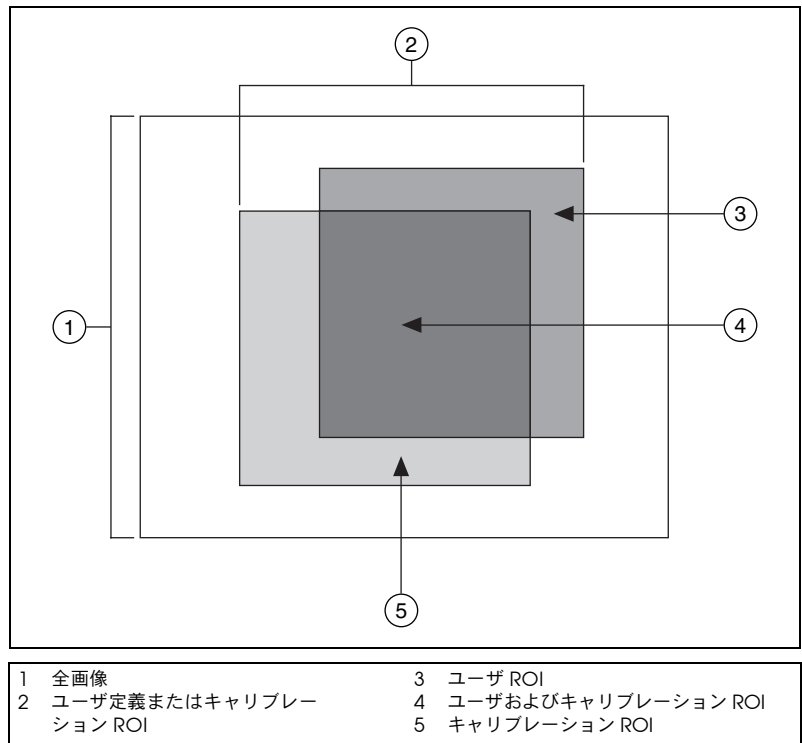


図 3-10 ROI モード

- 全画像：キャリブレーション ROI やユーザ定義の ROI に関係なく、画像全体を補正します。
- ユーザまたはキャリブレーション ROI：ユーザ定義の ROI およびキャリブレーション ROI の両方のピクセルを補正します。

- ユーザ ROI：キャリブレーションの学習段階で指定したユーザ定義の ROI 内のピクセルのみを補正します。
- ユーザおよびキャリブレーション ROI：ユーザ定義の ROI とキャリブレーション ROI の交差する部分にあるピクセルのみを補正します。
- キャリブレーション ROI：選択したキャリブレーションアルゴリズムにより出力されたキャリブレーション ROI 内のピクセルのみ補正します。スケールモードを大きさに合わせてスケールに設定すると、この ROI は画像全体に合わせられます。

有効な座標は、実世界座標にマップしようとしているピクセル座標が補正した画像領域内にあるかどうかを示します。たとえば、キャリブレーション ROI 内のピクセルのみを補正して、キャリブレーション ROI 以外の部分のピクセルを実際の座標にマップしようとしている場合、Corrected Image Learn ROI パラメータはエラーを示します。

シンプルキャリブレーション

カメラ軸が画像面に対して垂直で、レンズ歪みがほとんどない場合は、シンプルキャリブレーションを使ってイメージングセットアップをキャリブレーションすることができます。シンプルキャリブレーションでは、ピクセル座標は x および y （水平および垂直）方向にスケールすることにより実世界座標に変換されます。

シンプルキャリブレーションは、キャリブレーショングリッドなしで直接ピクセル座標を実世界座標にマップします。ソフトウェアは、定義済みの座標基準とスケールファクタにしたがって、ピクセル座標を回転しスケールします。

シンプルキャリブレーションを行うには、座標系とスケールモードを定義します。図 3-11 は、座標系の定義方法を示します。座標基準を設定するには、 x 軸と画像の横軸の角度を度単位で指定します。中心を、座標基準の原点にする位置として、ピクセルで表します。軸方向を直接位置または間接位置に設定します。スケールモードオプションを大きさに合わせてスケールか面積を維持してスケールのいずれかに設定します。シンプルキャリブレーションには、補正テーブルオプションもあります。



メモ 角度を 0 に設定してシンプルキャリブレーションを行う場合は、画像補正の必要がないため補正用に学習する必要もありません。

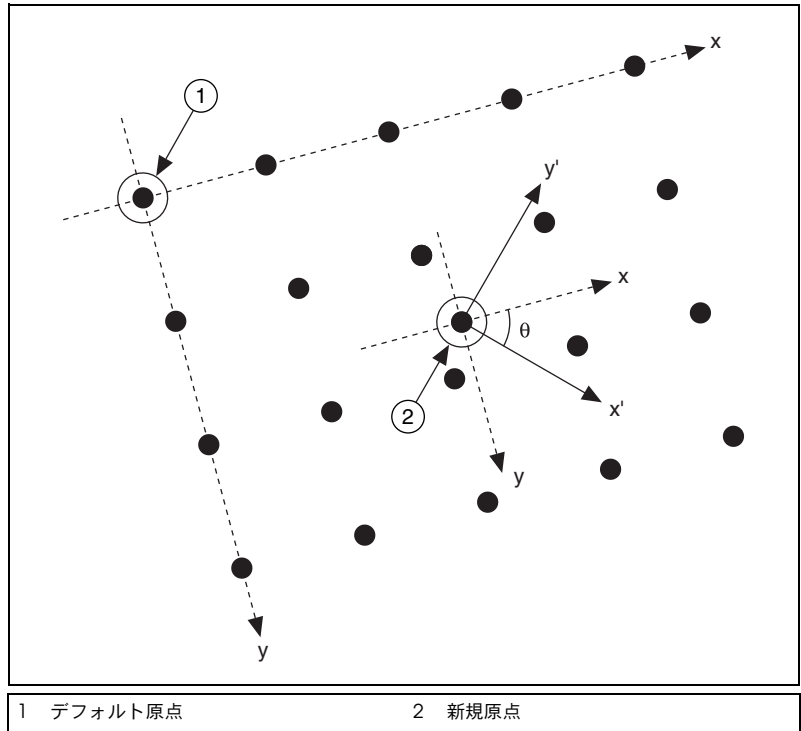


図 3-11 新規座標系の定義

座標系を再定義する

シンプルキャリブレーションを使用して、キャリブレーション済みの画像に割り当てられた座標系を変更することができます。新規座標系を定義する場合は、次の点に注意してください。

- 原点はピクセルで表示します。その位置を実世界単位に変換できるように、原点の位置はキャリブレーショングリッド内に入るようにします。
- 角度は新規座標系と実世界での水平方向の間の角度に指定します。

ビジョンアプリケーションによっては、オブジェクトを完全に検査するために数カ所の領域を画像化する必要があることもあります。それらの異なる領域を画像化するには、カメラの下にその領域が来るまでオブジェクトを移動させるか、カメラを移動させてその領域の上に来るようにします。いずれの場合でも、各画像は実世界では別の領域にマップします。座標系の原点と方向を新規に指定して、原点が検査対象のオブジェクト上のポイントに来るようにすることができます。

図 3-12 は、ボードの角に対するボード上の穴の位置を測定するための検査アプリケーションを示します。ボードは、x 方向と y 方向に移動可能で中心を固定した状態で回転可能な台の上に乗っています。ボードの角は台の中心にあります。図 3-12a に示すセットアップの初期段階では、シンプルキャリブレーションを使って、座標系をボードの角に合わせて定義します。座標系の原点はピクセル単位でボードの角に指定し、軸の角度は 180° に、軸方向は間接に設定します。パターンマッチングを使用して穴の位置をピクセルで求めます (図 3-12a に十字で示した部分)。ピクセル単位の穴の位置を実世界での位置に変換します。この変換により、定義済みの座標系に対する穴の実世界での位置が返されます。検査プロセスで、台はある特定の距離を移動したり回転したりします。このため、カメラの視野の中でボードは移動したり回転したり見えます。図 3-12b に示すように、以降の画像ではボードが移動したり回転したように見えます。ボードが移動したため、元の座標系はボードの角と揃わなくなりました。したがって、この座標系を使って行った測定値は不正確になります。台が移動した距離に関する情報を使って、画像内でのボードの角の新しい位置を決め、シンプルキャリブレーションを使って座標系を更新してその変更を反映させます。更新された座標系の原点がボードの角の新しいピクセル位置になり、座標系の角度は台が回転した角度になります。更新された座標系を図 3-12c に示します。新しい座標系では正確な測定値が得られます。

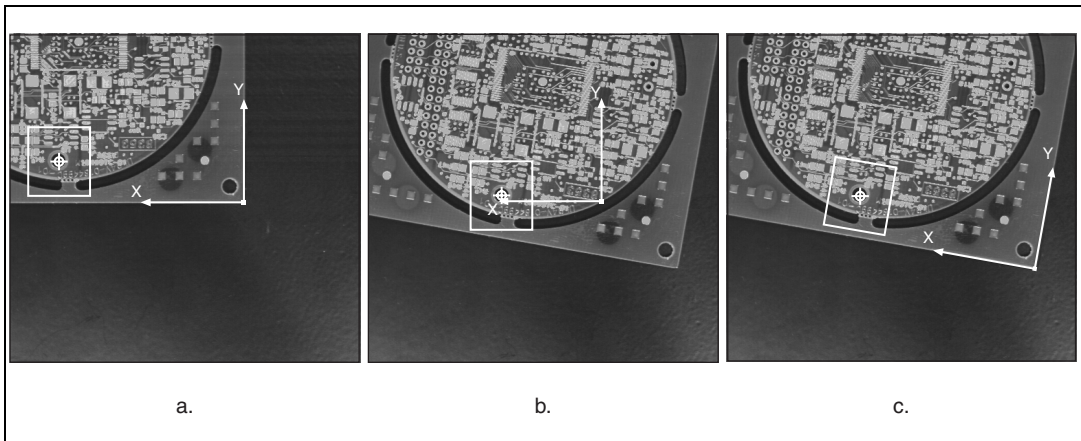


図 3-12 座標系の移動

第 II 部

画像処理および解析

このセクションでは、画像解析／処理、演算子、および周波数領域解析の概念について説明します。

第 II 部「画像処理および解析」では、以下について説明します。

第 4 章「[画像解析](#)」では、ヒストグラム、ラインの輪郭、および輝度測定について説明します。

第 5 章「[画像処理](#)」では、ルックアップテーブル、カーネル、空間フィルタ、およびグレースケールモフォロジーについて説明します。

第 6 章「[演算子](#)」では、画像をマスク、合成、比較するための算術演算子と論理演算子について説明します。

第 7 章「[周波数領域解析](#)」では、周波数領域解析、FFT、および周波数領域における画像解析や処理について説明します。

画像解析

本章では、ヒストグラム、ラインプロファイル、および輝度測定について説明します。

画像解析とは、画像のピクセルのグレースケール輝度に基づいて、統計値と測定値を計算するテクニックです。画像解析関数を使用すると、画像の内容を理解し、アプリケーションのソリューションとしてどのタイプの検査ツールを使用するかを決定することができます。画像解析関数にはまた、存在や欠如の確認などの基本的検査タスクを実行するための測定も備わっています。

ヒストグラム

ヒストグラムは、各グレースケールレベルの総ピクセル数をカウントしてグラフにします。そのグラフから、画像に特定の際立ったグレースケール値の領域が含まれているかがわかります。

ヒストグラムは、画像の外観の一般的な概要を表すもので、背景やオブジェクト、ノイズなどのあらゆる成分を識別するのに役立ちます。

使用目的

ヒストグラムは、画像内でのピクセル輝度の分布を表すための基本的な画像解析ツールです。ヒストグラムを使用して、画像全体の輝度が検査タスクを行うために十分であるかどうかを判断します。また、ヒストグラムを使用すると、画像に特定のグレースケール値の際立った領域が含まれているかどうかを確認することができます。さらに、ヒストグラムを使用すると、画像集録条件を調整することもできます。

ヒストグラムによって、2つの重要な要素を確認することができます。

- **飽和**：イメージング環境で光量が不足するとイメージングセンサで露光不足になり、逆に光量が多すぎるとイメージングセンサで露光過多、つまり飽和が起きます。露光不足または露光過多状態で集録された画像には、その中から検出しようとしている情報がすべて含まれません。そのようなイメージング環境を見つけ出し、イメージングシステムのセットアップの際に修正することが重要です。センサが露光不足か露光過多であることをを見つけるには、ヒストグラムを使用します。露光不足の画像には、グレースケール値の低いピクセルが多く含まれます。そのような場合、ヒストグラムには低い方の端にピークが表

示されます。露光過多、つまり飽和状態の画像には、グレーレベル値の非常に高いピクセルが多く含まれます。そのような場合は、図 4-1 に示すように、ヒストグラムでは高い方の端にピークが表示されま

- コントラスト不足：よく使用されているタイプのイメージングアプリケーションでは、画像の中の対象部分の検査および測定（カウント）が一般的に行われます。オブジェクトを背景と区別する方法では、たとえば明るい部分と暗い背景との輝度差が重要になります。その場合、画像のヒストグラムを解析することにより、図 4-2 に示すように、複数の明確に分離された輝度の集合が現れます。集録した画像のヒストグラムで、アプリケーションに必要なコントラストが得られるまで、イメージングセットアップを調整してください。

ヒストグラムの概念

ヒストグラムは、グレースケール範囲 $[0, \dots, k, \dots, 255]$ で定義された関数 H で、グレーレベル値 k に等しいピクセルの数は次のようになります。

$$H(k) = n_k$$

ここで k はグレーレベル値、
 n_k は画像内で k に等しいグレーレベル値を持つピクセルの数、
 $k n_k = n$ は画像内の総ピクセル数です。

次のヒストグラムプロットは、発生頻度の多いグレーレベルと発生頻度の少ないグレーレベルを表しています。

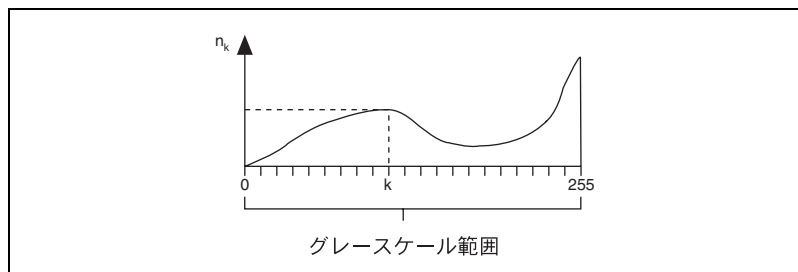


図 4-1 ヒストグラムプロット

線形ヒストグラムと累積ヒストグラムの 2 種類のヒストグラムを計算することができます。

いずれの場合でも、横軸は 0 から 255 の範囲を持つグレーレベル値を表しています。グレーレベル値 k では、線形ヒストグラムの縦軸は値 k のピクセルの数 n_k を表し、累積ヒストグラムの縦軸は k 以下のピクセルの割合を表します。

線形ヒストグラム

濃度関数は次のようになります。

$$H_{Linear}(k) = n_k$$

ここで $H_{Linear}(k)$ は、 k に等しいピクセルの数です。

確率関数は次のようになります。

$$P_{Linear}(k) = n_k / n$$

ここで $P_{Linear}(k)$ は、ピクセルが k に等しくなる確率です。

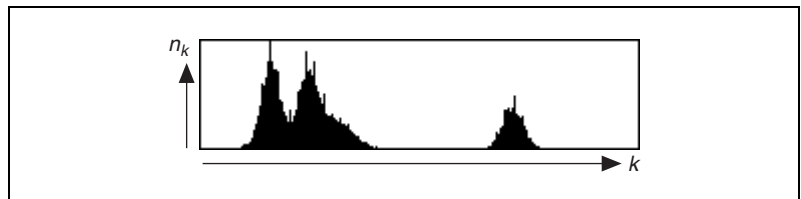


図 4-2 線形ヒストグラムの例

累積ヒストグラム

分布関数は次のようになります。

$$H_{Cumul}(k) = \sum_{i=0}^k n_i$$

ここで $H_{Cumul}(k)$ は、 k 以下のピクセルの数です。

確率関数は次のようになります。

$$P_{Cumul}(k) = \sum_{i=0}^k \frac{n_i}{n}$$

ここで $P_{Cumul}(k)$ は、ピクセルが k 以下になる確率です。

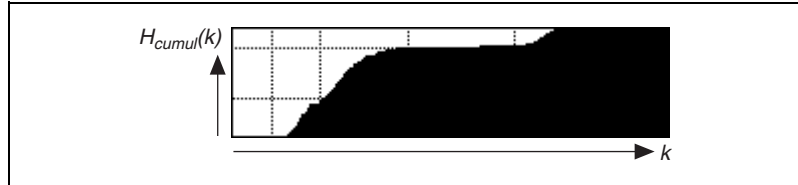


図 4-3 累積ヒストグラムの例

解釈

ピクセルが集中した部分を表すグレーレベルの区間は、画像内に顕著な成分があることと、それぞれの輝度範囲を表します。

図 4-2 の線形ヒストグラムにより、画像が 3 つの主要な要素から成っていることがわかります。図 4-3 の同じ画像の累積ヒストグラムによると、左の 2 つのピークが画像のおよそ 80% を占め、残りの 20% が 3 つめのピークに対応していることがわかります。

ヒストグラムの目盛

ヒストグラムプロットの縦軸は、線形目盛または対数目盛で表すことができます。対数目盛を使用すると、少数のピクセルによって使用されているグレーレベル値をはっきりと表現することができます。ヒストグラムを線形目盛で表示すると、それらの値は使用されていないように見える場合があります。

対数目盛では、ヒストグラムの縦軸がグレーレベル値に対応するピクセル数の対数を表します。低いグレーレベル値の使用が顕著になる代わりに、突出したグレーレベル値が目立たなくなります。対数目盛を使用すると、線形目盛では目に付かなかったような小さなヒストグラム値が強調されます。図 4-4 は、同じ画像を線形目盛と対数目盛で表示した場合の違いを示したものです。この画像においては、3 つのピクセルが 0 になっています。

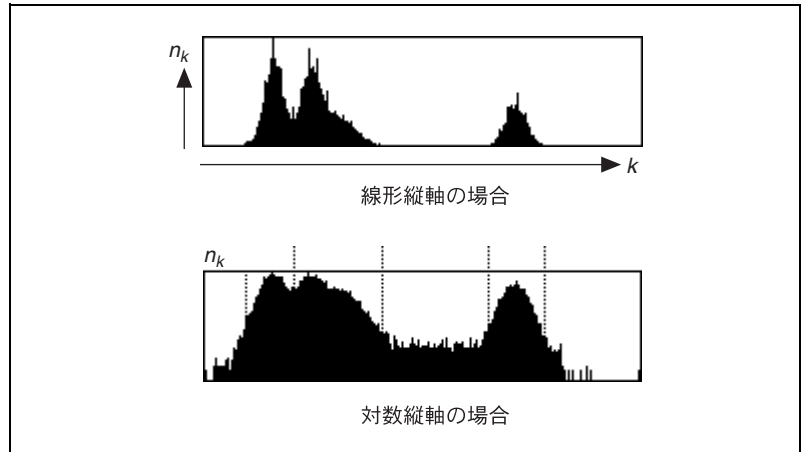


図 4-4 縦軸に線形目盛と対数目盛を使用した場合の同じ画像のヒストグラム

カラー画像のヒストグラム

カラー画像のヒストグラムは3つのテーブルで表現され、それぞれが表 4-1 に示すカラーモデルの3つの主要成分のヒストグラムに対応しています。

表 4-1 カラーモデルと主要成分

カラーモデル	成分
RGB	赤、緑、青
HSL	色相、彩度、明度

ラインプロファイル

ラインプロファイルとは、ラインに沿った輝度の変化をプロットしたものです。ライン上のピクセルのグレースケール値を返し、それをグラフ化します。

使用目的

ラインプロファイルユーティリティは、成分間の境界線の確認、輝度の変化の幅の定量化、および反復パターンの検出などに役立ちます。図 4-5 は、一般的なラインプロファイルを示します。

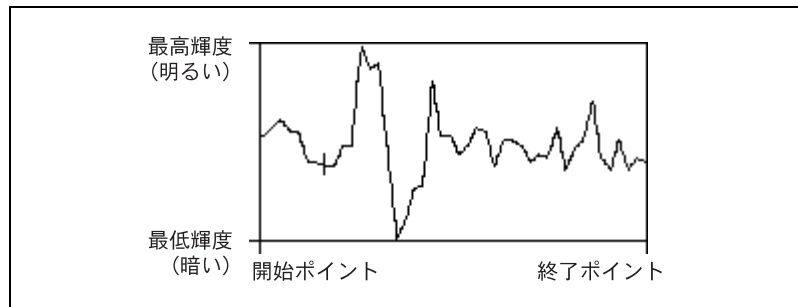


図 4-5 ラインプロファイル

ピークや谷は、画像内で選択したライン上の光の輝度の上昇や低下を表します。その幅や高さは、関連する領域のサイズや輝度に比例します。

たとえば、均一の輝度を持つ明るいオブジェクトは、プロットでは平坦域として表示されます。オブジェクトと周辺の背景とのコントラストが大きいほど、平坦域の傾斜も大きくなります。一方、ノイズの多いピクセルは、一連の小さなピークとして表示されます。

輝度測定

輝度測定とは、画像または画像内の領域のグレースケール画像統計を測定するものです。

使用目的

輝度測定を使用すると、画像内の領域の平均輝度値を測定し、たとえばある部分の有無や欠陥などを検出することができます。

概念

濃度測定

IMAQ Vision には、次の濃度測定パラメータがあります。

- 最小グレー値：グレーレベル単位で表した最小輝度値
- 最大グレー値：グレーレベル単位で表した最大輝度値
- 平均グレー値：グレーレベル単位で表した粒子の平均輝度値
- 標準偏差：輝度値の標準偏差

画像処理

本章では、ルックアップテーブル、コンボリューションカーネル、空間フィルタ、およびグレースケールモフォロジーについて説明します。

ルックアップテーブル

ルックアップテーブル (LUT) 変換とは、他の領域の情報量を落として、重要な情報を含む領域の細部をハイライト表示する基本画像処理関数です。そのような関数には、**ヒストグラム平均化**、**ガンマ補正**、**対数補正**、および**指数補正**があります。

使用目的

LUT 変換を使用して、コントラストの少ない領域のダイナミック輝度を修正し、それによって画像のコントラストと輝度を向上させます。

LUT 変換の概念

LUT 変換により、入力されたグレーレベル値 (元の画像からの値) が他のグレーレベル値 (変換後の画像) に変換されます。

LUT 変換は、指定した入力範囲 (rangeMin, rangeMax) に変換 $T(x)$ を次のように適用します。

$$T(x) = \begin{array}{ll} \text{dynamicMin} & \text{if } x \leq \text{rangeMin} \\ f(x) & \text{if } \text{rangeMin} < x \leq \text{rangeMax} \\ \text{dynamicMax} & \text{if } x > \text{rangeMax} \end{array}$$

x は入力されたグレーレベル値を表し、この場合 $\text{dynamicMin} = 0$ (8 ビット画像) または最小初期ピクセル値 (16 ビットおよび浮動小数点画像) です。

$\text{dynamicMax} = 255$ (8 ビット画像) または最大初期ピクセル値 (16 ビットおよび浮動小数点画像)

$$\text{dynamicRange} = \text{dynamicMax} - \text{dynamicMin}$$

$f(x)$ は新規の値を表します。

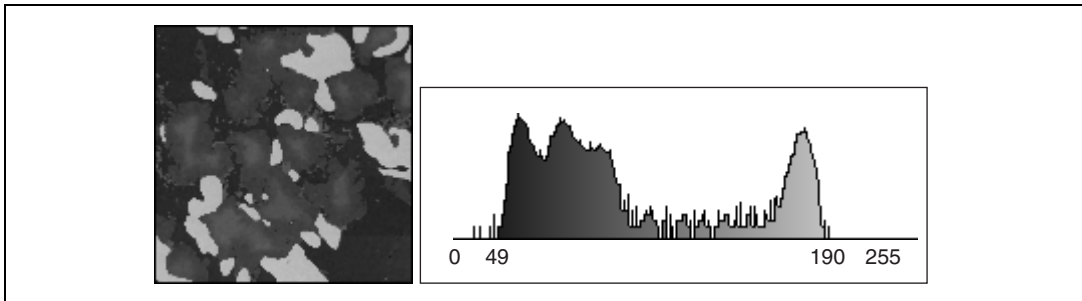
関数は $f(x)$ をスケールして、 $f(\text{rangeMin}) = \text{dynamicMin}$ 、 $f(\text{rangeMax}) = \text{dynamicMax}$ になるようにします。 $f(x)$ は、 $(\text{rangeMin}, \text{rangeMax})$ に対して選択したメソッドに従って動作します。

8 ビット解像度の場合、LUT は 256 個の要素を持つ表です。配列要素の指標は入力されたグレーレベル値を示します。各要素の値は出力値を示します。

LUT に関連する伝達関数は、画像の輝度やコントラストに対して、意図したとおりの効果があります。

例

次の例では以下のソース画像を使用します。ソース画像の線形ヒストグラムで、グレーレベル区間 $(0, 49)$ と $(191, 254)$ には有意情報が含まれていません。



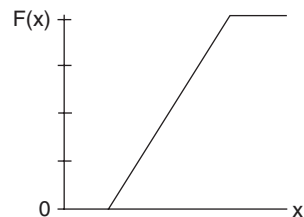
以下の LUT 変換を使用して、49 未満の値のピクセルが 0 に、191 を超える値のピクセルが 255 に設定されます。区間 $(50, 190)$ が $(1, 254)$ に拡大され、グレーレベル範囲 $(50, 190)$ のピクセルが集中することによってその領域のダイナミック輝度が広がります。

$$x \in (0, 49), F(x) = 0$$

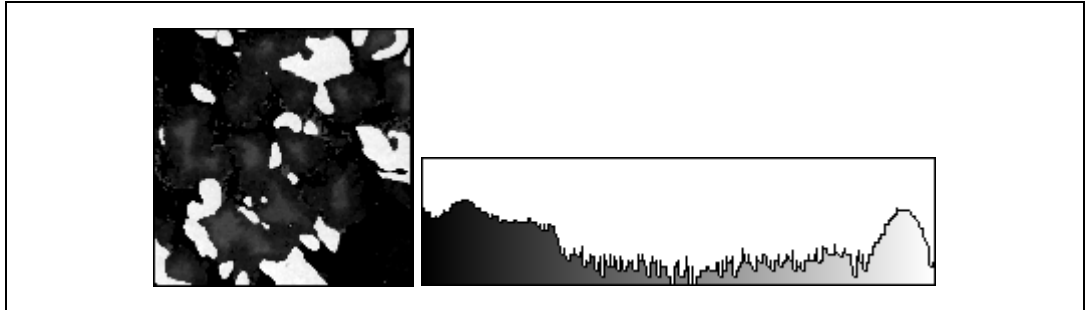
$$x \in (191, 254), F(x) = 255$$

それ以外の場合

$$F(x) = 1.81 \times x - 89.5$$






LUT 変換により、以下のような画像が生成されます。新しい画像の線形ヒストグラムには、区間 (50, 190) の 2 つのピークしか含まれていません。



定義済みのルックアップテーブル

IMAQ Vision では、線形、対数、 $1/Y$ 乗、平方根、指数、 Y 乗、二乗の 7 つの定義済み LUT を使用できます。表 5-1 は、各 LUT の伝達関数を示し、暗い色が低輝度値に、明るい色が高輝度値に関連付けられたパレット（グレーパレットなど）に表示されている画像への効果を説明します。

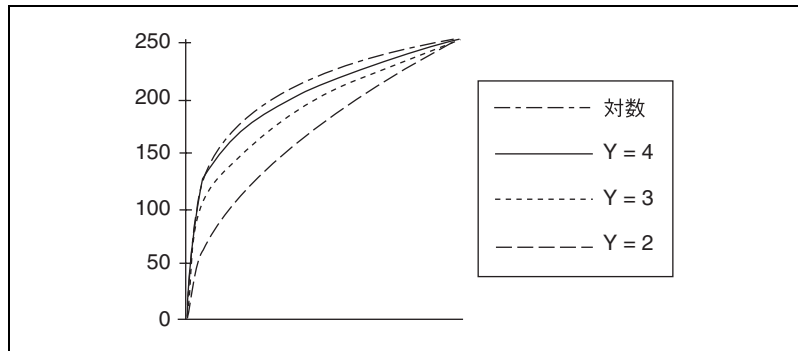
表 5-1 LUT 伝達関数

LUT	伝達関数	シェーディング補正
線形		グレースケール全体 (0, 255) に対して特定のグレーレベル区間 (min, max) を均一に分散することにより、ダイナミック輝度を上げます。デフォルトの最小値と最大値は、8 ビット画像で 0 と 255 です。
対数 $1/Y$ 乗 平方根		暗い部分の輝度とコントラストを上げ、明るい部分のコントラストを下げます。
指数 Y 乗 二乗		暗い部分の輝度とコントラストを下げ、明るい部分のコントラストを上げます。

対数および逆ガンマ補正

対数および逆ガンマ補正は、低グレーレベル範囲を拡大して、高グレーレベル範囲を縮小します。グレーパレットを使用する際は、これらの変換により画像全体の輝度が高くなり、暗い部分のコントラストが強まって逆に明るい部分のコントラストが弱くなります。

次のグラフは、変換の動作方法を示します。横軸が入力グレーレベル範囲、縦軸が出力グレーレベル範囲を表します。各入力グレーレベル値のルックアップ曲線との交差点を横にプロットして、出力値を表します。



対数、**平方根**、および **$1/Y$ 乗**の各関数は、低グレーレベル値を含む区間を拡大し、高グレーレベル値を含む区間を縮小します。

ガンマ係数 Y が高いほど、輝度補正は強くなります。対数補正は $1/Y$ 乗関数よりも強い効果があります。

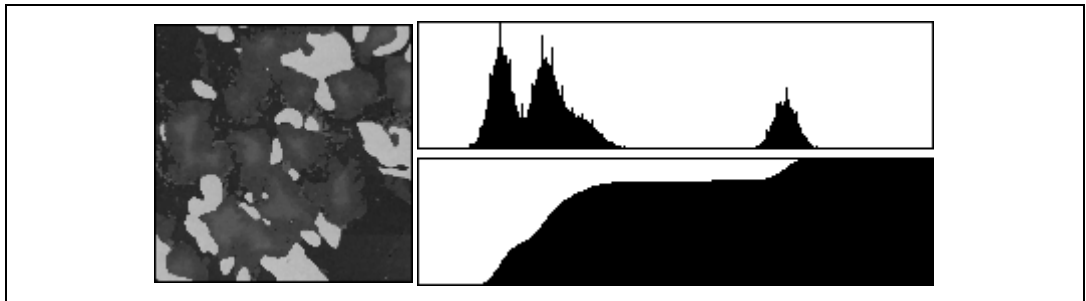
対数および逆ガンマ補正の例

以下の図は、さまざまな LUT 変換を行った後の画像の線形ヒストグラムと累積ヒストグラムを示します。ヒストグラムが右側に圧縮されているほど、画像は明るくなります。

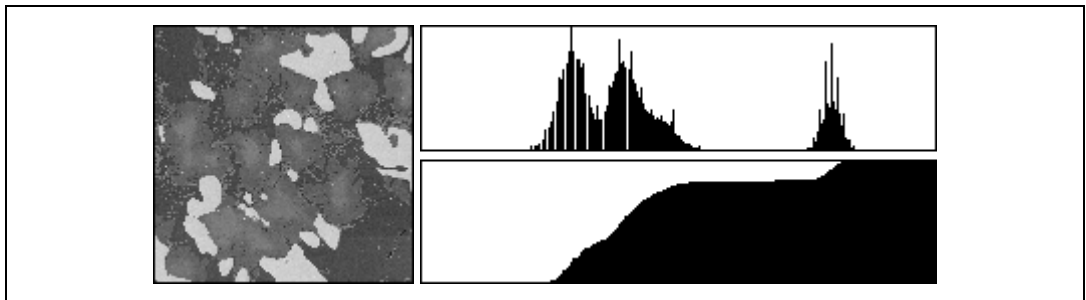


メモ 左の図は元の画像を示し、右上の画像が線形ヒストグラムを、右下の画像が累積ヒストグラムを示します。

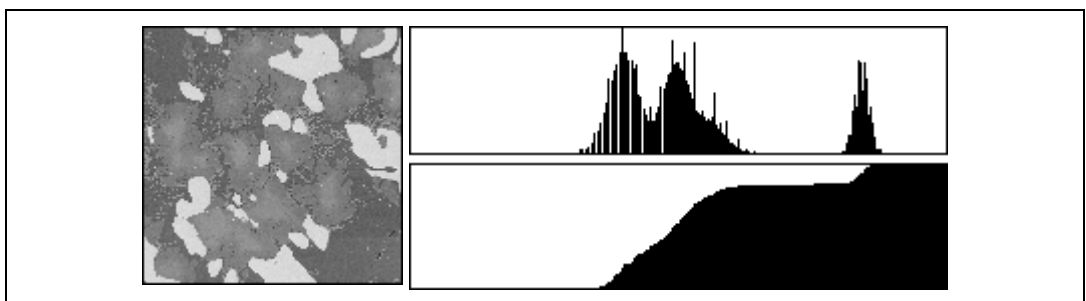
元の画像とヒストグラムを以下に示します。



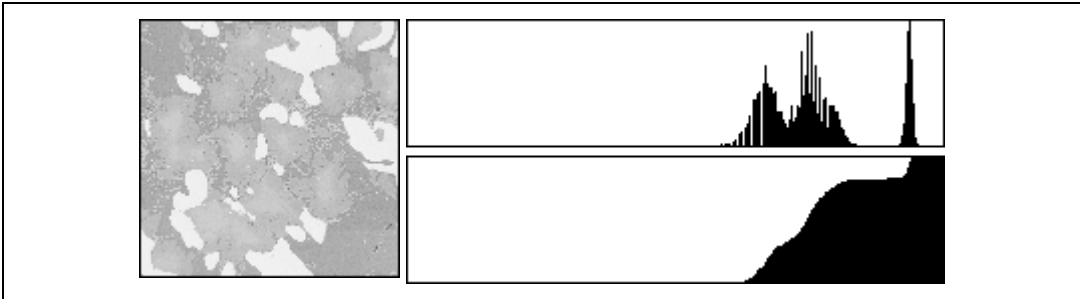
$1/Y$ 乗変換 ($Y = 1.5$ の場合) により次の画像とヒストグラムが生成されます。



平方根または $1/Y$ 乗変換 ($Y = 2$ の場合) により次の画像とヒストグラムが生成されます。



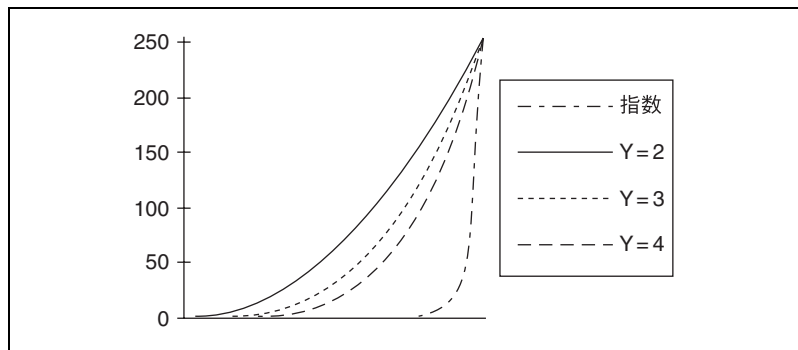
対数変換により次の画像とヒストグラムが生成されます。



指数およびガンマ補正

指数および逆ガンマ補正は、高グレーレベル範囲を拡大して、低グレーレベル範囲を縮小します。グレーパレットを使用する際は、これらの変換により画像全体の輝度が低くなり、明るい部分のコントラストが強まって逆に暗い部分のコントラストが弱くなります。

次のグラフは、変換の動作方法を示します。横軸が入力グレーレベル範囲、縦軸が出力グレーレベル範囲を表します。各入力グレーレベル値のルックアップ曲線との交差点を横にプロットして、出力値を表します。



指数、**二乗**、および**Y乗**の各関数は、高グレーレベル値を含む区間を拡大し、低グレーレベル値を含む区間を縮小します。

ガンマ係数 γ が高いほど、輝度補正は強くなります。指数補正は γ 乗関数よりも強い効果があります。

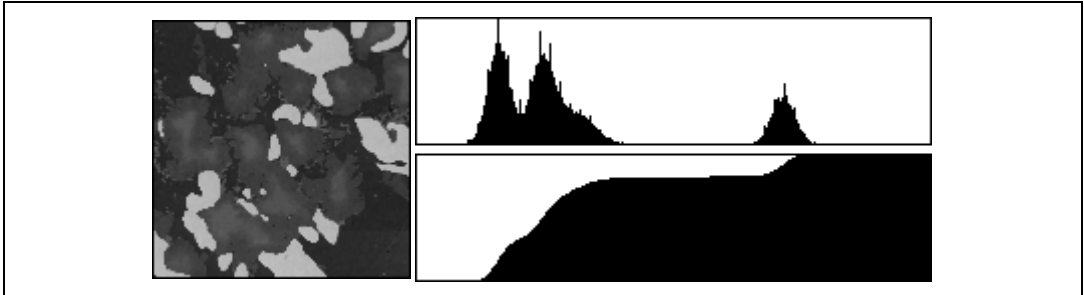
指数およびガンマ補正の例

次の図は、さまざまな LUT 変換を行った後の画像の線形ヒストグラムと累積ヒストグラムを示します。ヒストグラムが圧縮されているほど画像は暗くなります。

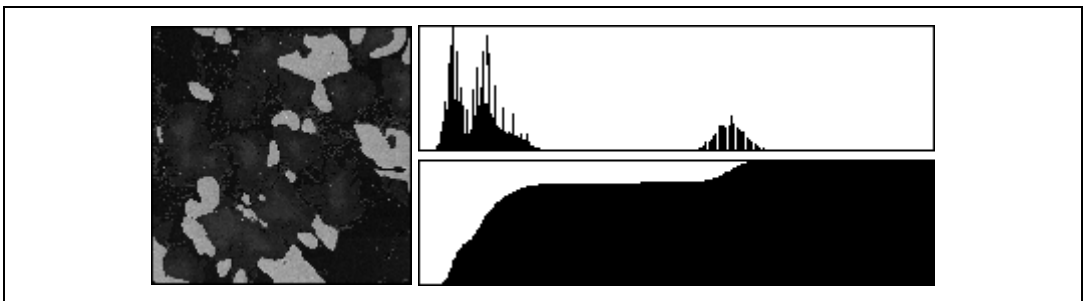


メモ 左の図は元の画像を示し、右上の画像が線形ヒストグラムを、右下の画像が累積ヒストグラムを示します。

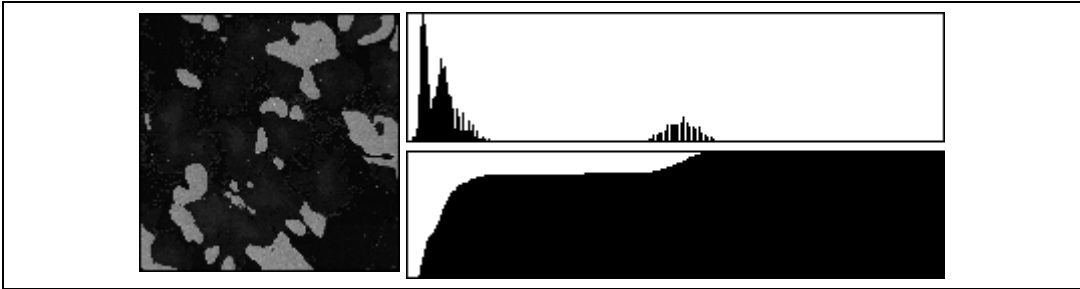
元の画像とヒストグラムを以下に示します。



1/Y 乗変換 ($Y = 1.5$ の場合) により次の画像とヒストグラムが生成されます。



二乗または Y 乗変換 ($Y=2$ の場合) により、次の画像とヒストグラムが生成されます。



指数変換により、次の画像とヒストグラムが生成されます。



平均化

平均化関数とは、定義済みの LUT では実行できないルックアップテーブル操作のことをいいます。LUT は関数が適用される画像の内容に基づいて計算されます。

Equalize 関数は、ピクセルのグレーレベル値を変更して、定義されているグレースケール範囲 (8 ビット画像で 0 から 255) に均等に分散するようにします。この関数は、一定のグレーレベル区間につき同じ量のピクセルを関連付けて、利用できるすべての階調のグレーを活用します。この変換を使用して、すべてのグレーレベルを使用していない画像のコントラストを高めます。

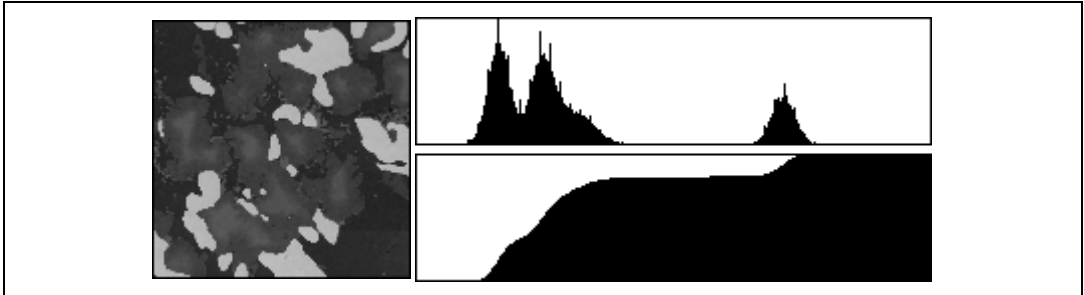
平均化をあるグレーレベル区間に限定することもできます。それは平均化範囲ともいいます。この場合、関数は平均化範囲内のピクセルを全区間 (8 ビット画像で 0 から 255) に分散し、他のピクセルを 0 に設定します。生成される画像では、平均化範囲の輝度を持つ領域の細部が表示され、他の領域は除去されます。

平均化の例 1

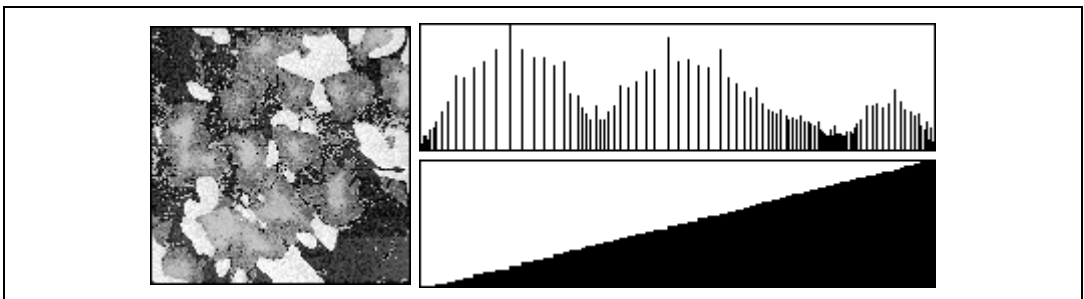
この例は、区間 $(0, 255)$ の平均化によって、元の 3 つのピークに含まれる情報がより大きな区間に拡大される方法を示します。変換された画像には、各コンポーネントの情報が元の画像より詳しく表示されます。元の画像とヒストグラムを以下に示します。



メモ 例 1 と 2 で、左の図は元の画像を示し、右上の画像が線形ヒストグラムを、右下の画像が累積ヒストグラムを示します。



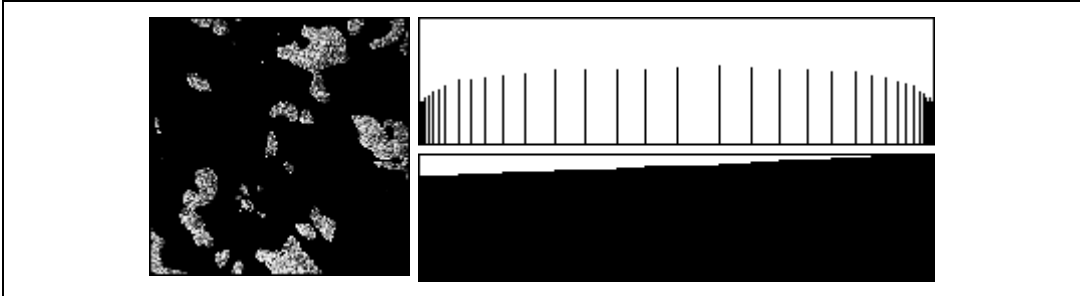
$(0, 255)$ から $(0, 255)$ への平均化により、次の画像とヒストグラムが生成されます。



メモ ヒストグラムの平均化を行った後の画像の累積ヒストグラムは、常に上図に示すような線形になります。

平均化の例 2

この例では、区間 (166, 200) の平均化により、元の 3 つのピーク (166 から 200 の範囲) に含まれる情報が区間 (0, 255) に拡大される方法を示します。変換された画像には、元の輝度範囲が (166, 200) の成分の詳細が表示され、他の成分はすべて黒に設定されます。(166, 200) から (0, 255) への平均化により、次の画像とヒストグラムが生成されます。



コンボリューションカーネル

コンボリューションカーネルとは、グレースケール画像に適用できる二次元フィルタを定義するものです。コンボリューションカーネルは、係数とそのコンボリューションフィルタの特性を定義する二次元構造体です。一般的なフィルタ処理では、コンボリューションカーネルの係数により画像内の各ピクセルのフィルタ処理後の値が決まります。IMAQ Vision では、いくつかの異なるタイプのフィルタ処理に使用できるコンボリューションカーネルが用意されています。また、独自のコンボリューションカーネルを定義することもできますので、カスタムフィルタの作成も可能です。

グレースケール画像をフィルタ処理する場合は、コンボリューションカーネルを使用します。グレースケール画像をフィルタ処理することにより、画質が向上してアプリケーションの条件に合うものになります。画像を滑らかにしたり、ノイズを除去したり、エッジ情報を強調したりする際に、フィルタを使用します。

概念

コンボリューションカーネルは、グレースケール画像のピクセル値がフィルタによってどのように変わるかを定義します。コンボリューションカーネルとは、その係数によって各ピクセルのフィルタ処理後の値の計算方法を定義する二次元構造体です。ピクセルのフィルタ処理後の値は、元の値と近接ピクセルの値の加重結合です。コンボリューションカーネル係数は、各近接ピクセルによる、更新されるピクセルに対する貢献度を定義します。コ

ンボリューションカーネルサイズは、フィルタ処理の際に値が考慮される近接ピクセルの数を決定します。

図 5-1a に示す 3 × 3 カーネルの場合、中央のピクセル（黒で表示）は 8 つの近接ピクセル（グレーで表示）の値から得られたものです。図 5-1b に示す 5 × 5 カーネルでは 24 個の近接ピクセルを指定し、7 × 7 カーネルでは 48 個の近接ピクセルを指定するというようになります。

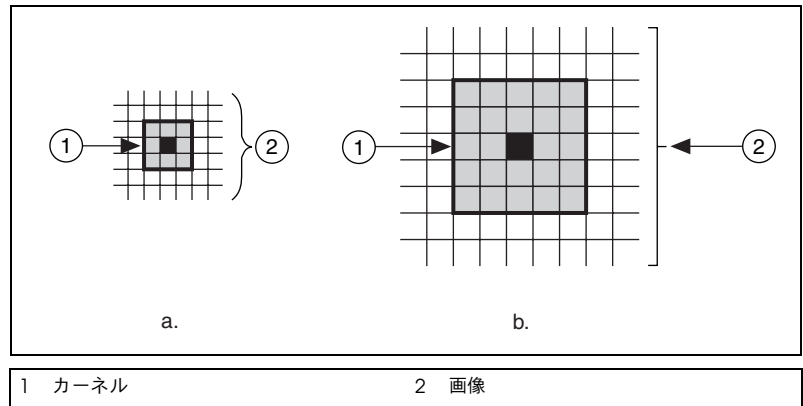


図 5-1 カーネルの例

画像に対してフィルタ処理を行う際は、カーネルを画像の左上隅のピクセルから右下隅のポイントに移動します。図 5-2 に示すように、画像の各ピクセルに対し、カーネルにある値を使って新しい値が計算されます。

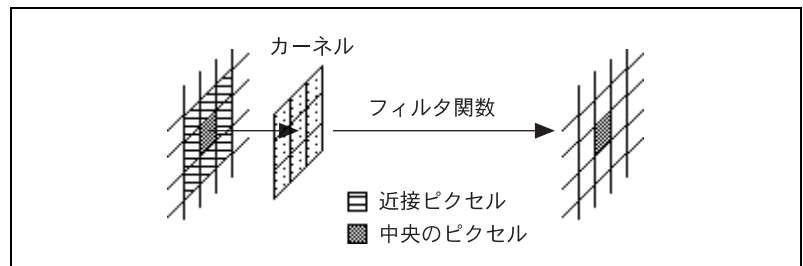


図 5-2 フィルタ処理の仕組み

画像の縁にあるピクセル（最初の行、最後の行、最初の列、最後の列のピクセル）のフィルタ処理値を計算するときは、カーネルの一部が画像の外にはみ出します。たとえば、図 5-3 では、左上隅の値を計算する際に、 3×3 カーネルの 1 行 1 列が画像の外にはみ出しています。

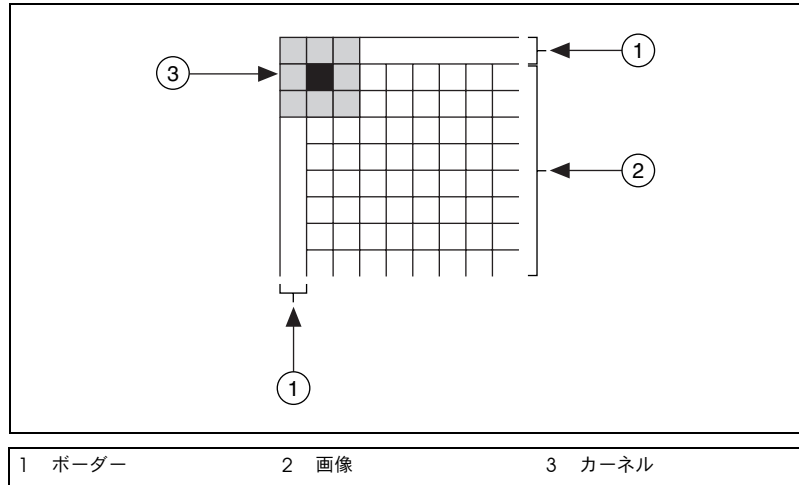


図 5-3 ボーダー部分のピクセルのフィルタ処理

IMAQ Vision は、画像を作成する際に自動的にボーダー領域を割り当てます。デフォルトのボーダー領域は、深度が 3 ピクセルでピクセル値 0 を含んでいます。また、カスタムボーダー領域を定義して、領域内でピクセル値を指定することもできます。ボーダー領域のサイズは、カーネルの行または列の数の半分以上でなければなりません。画像のボーダー上でのフィルタ処理の結果は、これらの値の計算に必要な近接ピクセルが不在のため、不確実なものになります。そのため、フィルタの効率が下がり、画像の残りの部分では指定よりもはるかに少ない数のピクセルに対して動作することになります。ボーダー領域に関する詳細は、本書の第 1 章、「[デジタル画像](#)」を参照してください。

空間フィルタ処理

フィルタは、線形（コンボリューションともいいます）と非線形の 2 つのタイプに分けられます。

コンボリューションとは、ピクセル値とコンボリューションカーネルの係数によって加重された近接ピクセルの値に基づいて、そのピクセルの値を再計算するアルゴリズムです。この計算で得られた合計値をカーネルの要素の合計で割り、新規のピクセル値を求めます。コンボリューションカーネルのサイズには理論上の制限はなく、正方形でも長方形でも（ 3×3 、 5×5 、 5×7 、 9×3 、 127×127 ）かまいません。コンボリューションは、傾斜、ラプラシアン、平滑化、ガウスの 4 つの種類に分けられます。このグループ分けは、コンボリューションカーネルまたは各ピクセルに割り当てられた加重によって決まり、それは中央のカーネルピクセルに対するそのピクセルの物理的位置により異なります。

IMAQ Vision には、各種類用の通常のサイズ（ 3×3 、 5×5 、および 7×7 ）の標準コンボリューションカーネルが含まれています。また、独自のカーネルを作成してそれに何を入れるかを選ぶこともできます。ユーザー定義のカーネルのサイズは、事実上無制限です。この機能により、特定の特徴を持つフィルタを作成することができます。

使用目的

空間フィルタは、特定方向のエッジ検出、パターンの輪郭描画、ノイズの除去、および細部のアウトラインやスムージングなど、さまざまな目的に使用することができます。フィルタを使って画像の平滑化、先鋭化、変形、およびノイズの削除などを行い、必要な情報を抽出できる状態にします。

非線形フィルタは、輪郭を抽出（エッジ検出）したり孤立したピクセルを削除するのに使用します。IMAQ Vision には、輪郭抽出に使用できる 6 つのメソッド（微分、傾斜、プレウィット、ロバーツ、シグマ、またはソーベル）があります。Canny Edge Detection フィルタは、画像の SN 比が低い条件下でもエッジを正確に検出する特殊なエッジ検出メソッドです。

ピクセル値を調整するには、異なるメソッドを使用する、NthOrder と LowPass の 2 つのフィルタのいずれかを選択します。これらの関数では、カーネルサイズと次数または割合のいずれかが入力において指定されている必要があります。

空間フィルタ は、近接ピクセルの輝度の変化に応じて、ピクセル値を変更します。ピクセルの近接ピクセルは、そのピクセル自体を中心とした行列のサイズ、つまりマスクにより定義されます。それらのフィルタは、輝度の変化の有無に影響を受けることがあります。

空間フィルタは、2つのカテゴリに分類されます。

- **ハイパスフィルタ** は、通常オブジェクトの境界部分に見られる顕著な輝度変化を強調します。ハイパス周波数フィルタを使用すると、鋭いエッジや細部、ノイズなどに相当する、急激に変動するパターンを分離することができます。
- **ローパスフィルタ** は、光の輝度の変化を減衰します。ローパス周波数フィルタを使用すると、オブジェクトと背景など、徐々に変化するパターンを強調することができます。このフィルタは、詳細を取り除いてエッジを不鮮明にし、画像全体を滑らかにする傾向があります。

空間フィルタの概念

空間フィルタの分類のまとめ

表 5-2 は、さまざまなタイプの空間フィルタをまとめたものです。

表 5-2 空間フィルタの分類

フィルタのタイプ	フィルタ
線形フィルタ	
ハイパス	傾斜、ラプラシアン
ローパス	平滑化、ガウシアン
非線形フィルタ	
ハイパス	傾斜、ロバーツ、ソーベル、プレウィット、微分、シグマ
ローパス	メディアン、N 次、ローパス

線形フィルタ

線形フィルタは、各ピクセルをその近接ピクセルの加重和に置き換えます。ピクセルの近接ピクセルを定義する行列は、各近接ピクセルに割り当てられたウェイトも指定します。この行列のことを、**コンボリューションカーネル**といいます。

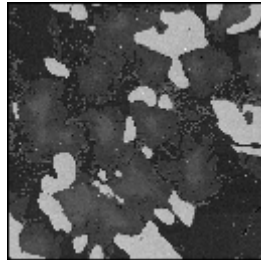
フィルタカーネルに正と負の係数が両方含まれている場合、伝達関数は加重微分と同等になり、先鋭化つまりハイパスフィルタを生成します。一般的なハイパスフィルタには傾斜フィルタとラプラシアンフィルタがあります。

カーネルの係数がすべて正の場合、伝達関数は加重合計と同等になり、平滑化またはローパスフィルタを生成します。一般的なローパスフィルタには、平滑化フィルタとガウシアンフィルタがあります。

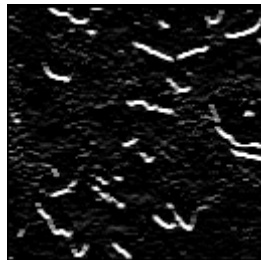
傾斜フィルタ

傾斜フィルタは、特定の方向に沿った輝度の変化をハイライトして、エッジを明確にしテクスチャを浮き立たせます。

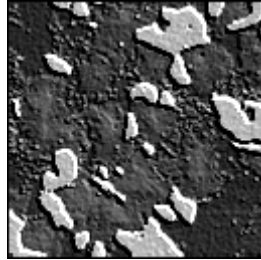
次のようなソース画像があるとします。



傾斜フィルタは、水平エッジを抽出して次のような画像を生成します。



傾斜フィルタは、対角エッジを抽出して次のような画像を生成します。



カーネルの定義

傾斜コンボリューションフィルタは、一次導関数です。そのカーネルでは、次のモデルを使用しています。

$$\begin{array}{ccc} a & -b & c \\ b & x & -d \\ c & d & -a \end{array}$$

ここで a 、 b 、 c および d は整数で $x=0$ または 1 です。

フィルタ軸および方向

このカーネルには、中央の要素を通してカーネルの正の係数と負の係数間を結ぶ対称軸があります。この対称軸により、描くエッジの方向が決まります。たとえば、

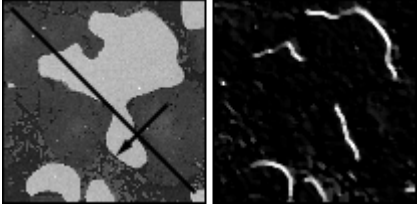
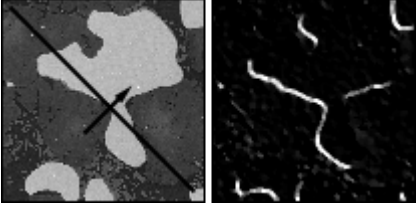
$a=0$ 、 $b=-1$ 、 $c=-1$ 、 $d=-1$ 、 $x=0$ の場合、カーネルは次のようになります。

$$\begin{array}{ccc} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{array}$$

対称軸は 135° に位置します。

ある方向に対し、傾斜フィルタを作成してその方向のエッジをハイライトしたり暗くしたりすることができます。フィルタは、実際にそのカーネルの対称軸に対して、垂直方向の輝度変化に敏感です。カーネルの負の係数から正の係数へ向かう方向 D に対し、フィルタは方向 D に沿って輝度が高くなるピクセルをハイライトし、輝度が低くなるピクセルを暗くします。

次の2つのカーネルは、135°方向のエッジを強調します。

傾斜 #1	傾斜 #2
$\begin{matrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix}$
<p>傾斜 #1 は、北東から南西方向に沿って輝度が高くなるピクセルをハイライトします。その同じ方向に沿って輝度が低くなるピクセルは暗くなります。この処理により、図に示すような明るい領域の北東方面のフロントエッジが明確に描かれます。</p>	<p>傾斜 #2 は、南西から北東方向に沿って輝度が高くなるピクセルをハイライトします。その同じ方向に沿って輝度が低くなるピクセルは暗くなります。この処理により、図に示すような明るい領域の南西方面のフロントエッジが明確に描かれます。</p>
	

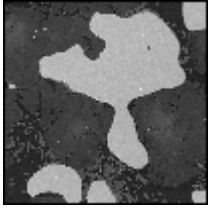



メモ 画像のルックアップテーブルを反転すると、明るい領域が暗く、暗い領域が明るくなるため、画像に傾斜 #1 を適用した場合と、その画像のフォトネガティブに傾斜 #2 を適用した場合は、同じ結果が得られます。

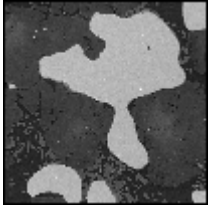
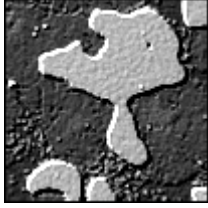
エッジの抽出とエッジのハイライト

傾斜フィルタでは、中央係数 x が 1 か 0 かによって、2つの効果が得られます。

- 中央係数がヌル ($x = 0$) の場合、傾斜フィルタは係数 a 、 b 、 c 、および d の構成で指定した方向に沿って輝度の変化が発生するピクセルをハイライトします。変換した画像には元のエッジの白黒の境界線が表示され、パターン全体の明度は暗くなります。

元の画像	傾斜 #1	フィルタ処理後の画像
	$\begin{matrix} -1 & -1 & 0 \\ -1 & \mathbf{0} & 1 \\ 0 & 1 & 1 \end{matrix}$	

- 中央係数が 1 ($x = 1$) に等しい場合、傾斜フィルタは上述の場合と同じ変化を検出しますが、それを元の画像の上に重ねて表示します。変換した画像は、元の画像のエッジをハイライトしたようになります。このタイプのカーネルは、粒子の抽出やテクスチャの認識に使用します。

元の画像	傾斜 #2	フィルタ処理後の画像
	$\begin{matrix} -1 & -1 & 0 \\ -1 & \mathbf{1} & 1 \\ 0 & 1 & 1 \end{matrix}$	

傾斜 #2 のカーネルは、下記のように分解することができます。

$$\begin{matrix} -1 & -1 & 0 \\ -1 & \mathbf{1} & 1 \\ 0 & 1 & 1 \end{matrix} = \begin{matrix} -1 & -1 & 0 \\ -1 & \mathbf{0} & 1 \\ 0 & 1 & 1 \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 \end{matrix}$$



メモ

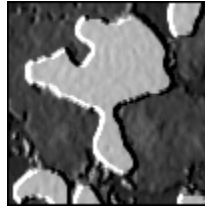
方程式の右辺の 2 つめのカーネルを使用したコンボリューションフィルタにより、元の画像が復元されます。すべての近接ピクセルに 0 が乗算され、中央のピクセルはそのままになります。 $(P_{(i,j)} = 1 \times P_{(i,j)})$

この方程式は、傾斜 #1 で抽出したエッジを、傾斜 #2 が元の画像に追加することを示します。

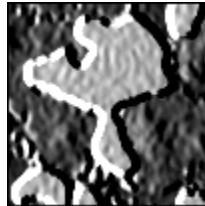
$$\text{傾斜 \#2} = \text{傾斜 \#1} + \text{元の画像}$$

エッジの太さ

カーネルが大きいほど、エッジは太くなります。次の画像は西-東 3×3 の傾斜を示します。



次の画像は西-東 5×5 の傾斜を示します。



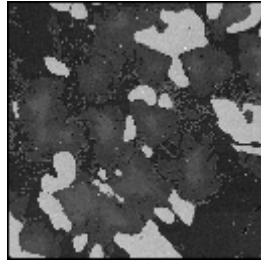
最後の画像は西-東 7×7 の傾斜を示します。



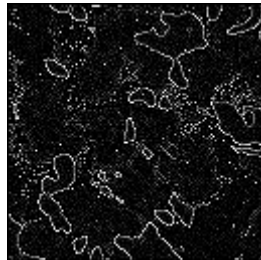
ラプラシアンフィルタ

ラプラシアンフィルタは、ピクセルの周辺の輝度変化をハイライトします。このフィルタは、オブジェクトの輪郭を抽出して詳細を明確に表示します。傾斜フィルタと異なり、このフィルタは全方向性です。

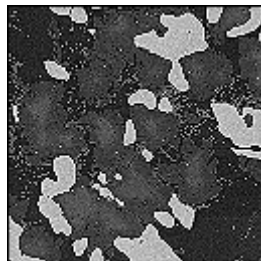
次のようなソース画像があるとします。



ラプラシアンフィルタは、輪郭を抽出して次のような画像を生成します。



ラプラシアンフィルタは、輪郭をハイライトして次のような画像を生成します。



カーネルの定義

ラプラシアンコンボリューションフィルタは二次導関数で、カーネルは以下のモデルを使用します。

$$\begin{array}{ccc} a & d & c \\ b & x & b \\ c & d & a \end{array}$$

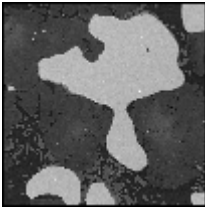

ここで a、b、c、および d は整数です。

ラプラシアンフィルタを使用すると、中央係数 x が外側の係数の絶対値の和以上かどうかにより、2 つの異なる結果が得られます。

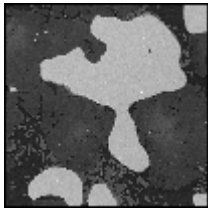

輪郭の抽出とハイライト

中央係数がこの和に等しい場合 ($x = 2(|a| + |b| + |c| + |d|)$)、ラプラシアンフィルタは明確な輝度変化が見られるピクセルを抽出します。鋭いエッジ、オブジェクト間の境界線、背景のテクスチャの変更、ノイズ、その他の効果により、そのような変化がもたらされます。変換された画像には、黒の背景上に白い輪郭が表示されます。

元の画像、ラプラシアンカーネル、およびフィルタ処理後の画像は以下のようになります。

元の画像	ラプラシアン #1	フィルタ処理後の画像
	$\begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$	

中央係数が外側の係数の合計より大きい場合 ($x > 2(a + b + c + d)$)、ラプラシアンフィルタは上述の場合と同じ変化を検出しますが、それを元の画像上に重ねて表示します。変換された画像は、元の画像と似ていますが、すべての明確な輝度変化がハイライトされています。

元の画像	ラプラシアン #2	フィルタ処理後の画像
	$\begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$	

ラプラシアン #2 のカーネルは、次のように分解することができます。

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & 1 & -1 \end{matrix} = \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$



メモ

方程式の右辺の 2 つめのカーネルを使用したコンボリューションフィルタにより、元の画像が復元されます。すべての近接ピクセルに 0 が乗算され、中央のピクセルはそのままです。 ($P_{(i,j)} = 1 \times P_{(i,j)}$)

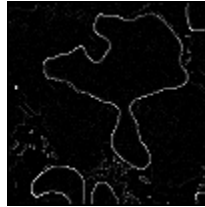
この方程式は、ラプラシアン #1 カーネルで抽出した輪郭をラプラシアン #2 カーネルが元の画像に追加することを示します。

$$\text{ラプラシアン \#2} = \text{ラプラシアン \#1} + \text{元の画像}$$

たとえば、ラプラシアン #2 カーネルの中央係数が 10 の場合、ラプラシアンフィルタは、ラプラシアン #1 カーネルで抽出した輪郭を、元の画像に 2 をかけたものに加算します。値の大きい中央係数は、あまり目立たない輪郭やフィルタによってハイライトされた詳細に相当します。

輪郭の太さ

カーネルが大きい場合輪郭は太くなります。次の画像はラプラシアン 3×3 です。



次の画像はラプラシアン 5×5 です。



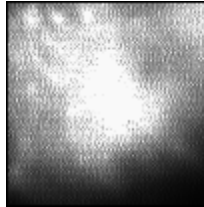
次の画像はラプラシアン 7×7 です。



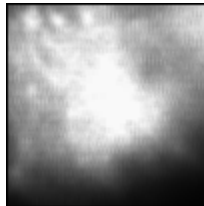
平滑化フィルタ

平滑化フィルタは、近接ピクセルの輝度変化を減衰します。オブジェクト全体の形状を滑らかにし、エッジを不鮮明にして、詳細を取り除きます。

次のようなソース画像があるとします。



平滑化フィルタを使用すると、次のような画像が生成されます。



カーネルの定義

平滑化コンボリューションフィルタは、カーネルで以下のモデルを使用する平均化フィルタです。

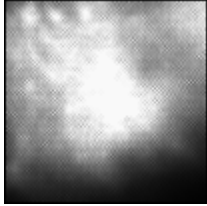
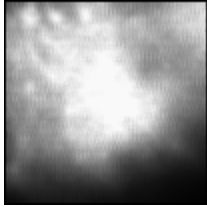
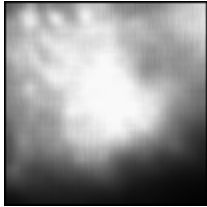
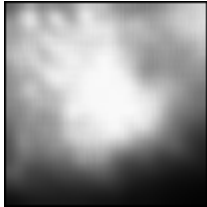
$$\begin{array}{ccc} a & d & c \\ b & x & b \\ c & d & a \end{array}$$

ここで a 、 b 、 c 、および d は正の整数で、 $x=0$ または 1 です。

平滑化カーネルの係数はすべて正であるため、各中央ピクセルはその近接ピクセルの加重平均になります。近接ピクセルのウェイトが強いほど、中央ピクセルの新しい値に対するその影響は大きくなります。

ある特定の係数 (a 、 b 、 c 、 d) に対し、中央係数が 0 ($x=0$) に等しい平滑化カーネルは、中央係数が 1 ($x=1$) の平滑化カーネルよりも、より強いぼかし効果があります。

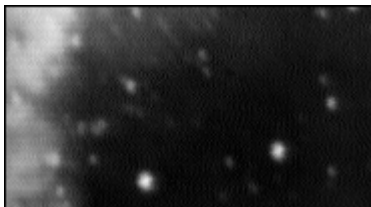
平滑化カーネルとフィルタ処理後の画像は以下のようになります。カーネルサイズの値が大きいと、より強い平滑化効果が得られます。

<p style="text-align: center;">カーネル #1</p> <pre style="text-align: center;"> 0 1 0 1 0 1 0 1 0 </pre>	<p style="text-align: center;">フィルタ処理後の画像</p> 
<p style="text-align: center;">カーネル #2</p> <pre style="text-align: center;"> 2 2 2 2 1 2 2 2 2 </pre>	<p style="text-align: center;">フィルタ処理後の画像</p> 
<p style="text-align: center;">カーネル #3</p> <pre style="text-align: center;"> 1 </pre>	<p style="text-align: center;">フィルタ処理後の画像</p> 
<p style="text-align: center;">カーネル #4</p> <pre style="text-align: center;"> 1 </pre>	<p style="text-align: center;">フィルタ処理後の画像</p> 

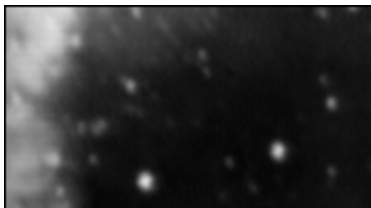
ガウシアンフィルタ

ガウシアンフィルタは、近接ピクセルの輝度変化を減衰します。オブジェクト全体の形状を滑らかにして、詳細を減衰します。平滑化フィルタと似ていますが、ぼかし効果はより控えめになっています。

次のようなソース画像があるとして。



ガウシアンフィルタを使用すると、以下のような画像が生成されます。



カーネルの定義

ガウシアンコンボリューションフィルタは平均化フィルタで、カーネルは以下のモデルを使用します。

$$\begin{matrix} a & d & c \\ b & x & b \\ c & d & a \end{matrix}$$

ここで a 、 b 、 c 、および d は整数で、 $x > 1$ です。

ある特定のサイズのガウシアンコンボリューションカーネルの係数は、ガウス曲線の整数を使用した最適近似です。たとえば、

3 × 3	5 × 5
1 2 1	1 2 4 2 1
2 4 2	2 4 8 4 2
1 2 1	4 8 16 8 4
	2 4 8 4 2
	1 2 4 2 1

ガウシアンカーネルの係数はすべて正であるため、各ピクセルはその近接ピクセルの加重平均になります。近接ピクセルのウェイトが強いほど、中央ピクセルの新しい値に対するその影響は大きくなります。

平滑化カーネルとは異なり、ガウシアンフィルタの中央係数は 1 より大きくなります。したがって、ピクセルの本来の値には、その近接ピクセルのいずれのウェイトよりも大きいウェイトが乗算されます。その結果、中央係数の値が大きいほど平滑化効果は少なくなります。カーネルサイズが大きいと、より強い平滑化効果が得られます。

非線形フィルタ

非線形フィルタは、各ピクセル値をその周辺ピクセルの非線形関数に置き換えます。線形フィルタと同様、非線形フィルタは近接ピクセルに作用します。

非線形プレウィットフィルタ

非線形プレウィットフィルタは、オブジェクトの外側の輪郭を抽出するハイパスフィルタです。このフィルタは、縦軸および横軸上に検出される顕著な輝度変化をハイライトします。

各ピクセルに、次のプレウィットコンボリューションカーネルを使用して得られた垂直および水平傾斜の最大値が割り当てられます。

カーネル #1	カーネル #2
-1 0 1	-1 -1 1
-1 0 1	0 0 0
-1 0 1	1 1 1

非線形ソーベルフィルタ

非線形ソーベルフィルタは、オブジェクトの外側の輪郭を抽出するハイパスフィルタです。このフィルタは、縦軸および横軸上に検出される顕著な輝度変化をハイライトします。

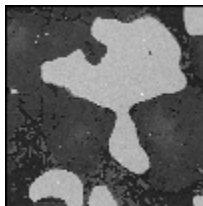
各ピクセルに、次のソーベルコンボリューションカーネルを使用して得られた垂直および水平傾斜の最大値が割り当てられます。

カーネル #1	カーネル #2
-1 0 1	-1 -2 -1
-2 0 2	0 0 0
-1 0 1	1 2 1

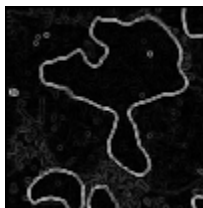
プレウィットフィルタとは反対に、ソーベルフィルタは中央ピクセルの縦および横の近接ピクセルにより高いウェイトを割り当てます。

非線形プレウィットおよび非線形ソーベルの例

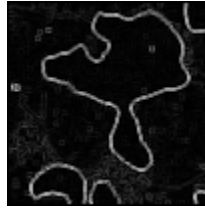
この例では以下のソース画像を使用します。



非線形プレウィットフィルタを使用すると、以下のような画像が生成されます。



非線形ソーベルフィルタを使用すると、以下のような画像が生成されます。



いずれのフィルタも、オブジェクトの輪郭を明確に描きます。異なるコンボリューションカーネルを組み合わせているため、非線形プレウィットフィルタは曲線の輪郭を、非線形ソーベルフィルタは四角形の輪郭を強調する傾向があります。この違いは、孤立したピクセルの輪郭の場合特に顕著です。

非線形傾斜フィルタ

非線形傾斜フィルタは、縦軸に沿って輝度変化が発生する輪郭を明確に表示します。

ロバーツフィルタ

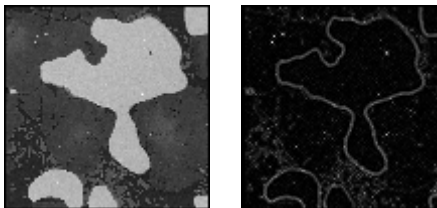
ロバーツフィルタは、対角軸に沿って輝度変化が発生するピクセルをハイライトする輪郭を描きます。

微分フィルタ

微分フィルタは、そのピクセルと左上 3 つの近接ピクセルの間に輝度変化が発生するピクセルをハイライトすることにより、連続的な輪郭を生成します。

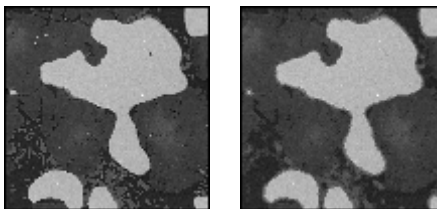
シグマフィルタ

シグマフィルタはハイパスフィルタです。このフィルタは、偏差があまり大きくない場合、ピクセルをその近接ピクセルの平均値に設定して、輪郭と詳細を明確にします。左側の例はフィルタ処理前の画像です。右側の例はフィルタ処理後の画像を示します。



ローパスフィルタ

ローパスフィルタは、偏差が大きい場合に、ピクセルをその近接ピクセルの平均値に設定することにより、詳細を少なくしてエッジを不鮮明にします。左側の例はフィルタ処理前の画像です。右側の例はフィルタ処理後の画像を示します。



メディアンフィルタ

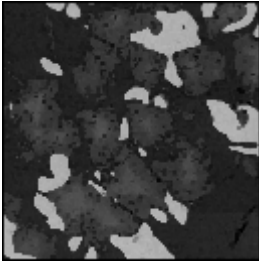
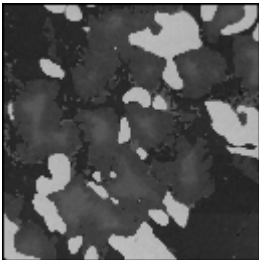
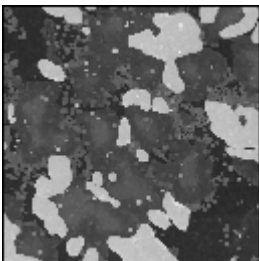
メディアンフィルタはローパスフィルタです。各ピクセルにその近接ピクセルの中央値を割り当て、孤立したピクセルを取り除いて詳細を少なくします。ただし、メディアンフィルタによってオブジェクトの輪郭がぼやけることはありません。

メディアンフィルタを実装するには、N 次フィルタを実行して、フィルタサイズ $f \times f$ に対し次数を $(f^2 - 1) / 2$ に設定します。

N次フィルタ

N次フィルタは、メディアンフィルタを拡張したものです。このフィルタは、各ピクセルにその近接ピクセルの N 次値を割り当てます（昇順に並べた場合）。値 N はフィルタの次数を指定し、これを使用して画像全体の輝度に対するフィルタの効果を加減することができます。次数が低い場合変換された画像は暗くなり、高い次数では明るくなります。

N 次フィルタの効果を確認するため、暗い背景と明るいオブジェクトを含む画像の例を示します。この画像をグレースケールで表示すると、オブジェクトは背景よりも高いグレースケール値になっています。

フィルタサイズ $f \times f$ に対して	フィルタサイズ 3×3 の例	
<ul style="list-style-type: none"> $N < (f^2 - 1)/2$ の場合、N 次フィルタは明るい領域を収縮（または暗い領域を膨張）する傾向があります。 $N = 0$ の場合、各ピクセルはローカルの最小値に置き換えられます。 	次数 0 (画像を平滑化し、明るいオブジェクトを収縮)	
<ul style="list-style-type: none"> $N = (f^2 - 1)/2$ の場合、各ピクセルはローカルの中央値に置き換えられます。オブジェクト内に孤立している暗いピクセルと、背景に孤立している明るいピクセルも取り除かれます。背景とオブジェクトの全体の面積に変化はありません。 	次数 4 (メディアンフィルタと同等)	
<ul style="list-style-type: none"> $N > (f^2 - 1)/2$ の場合、N 次フィルタは明るい領域を膨張（または暗い領域を収縮）する傾向があります。 $N = f^2 - 1$ の場合、各ピクセルはローカルの最大値に置き換えられます。 	次数 8 (画像を平滑化し、明るいオブジェクトを膨張)	

詳細の説明

$P_{(i,j)}$ が座標 (i,j) にあるピクセル P の輝度を示すとする場合、 $P_{(i,j)}$ の周辺のピクセルは、次のように指標付けすることができます (3×3 行列の場合)。

$P_{(i-1,j-1)}$	$P_{(i,j-1)}$	$P_{(i+1,j-1)}$
$P_{(i-1,j)}$	$P_{(i,j)}$	$P_{(i+1,j)}$
$P_{(i-1,j+1)}$	$P_{(i,j+1)}$	$P_{(i+1,j+1)}$

線形フィルタ は、 $P_{(i,j)}$ にその周辺値の線形結合である値を割り当てます。たとえば、

$$P_{(i,j)} = P_{(i,j-1)} + P_{(i-1,j)} + 2P_{(i,j)} + P_{(i+1,j)} + P_{(i,j+1)}$$

非線形フィルタ は、 $P_{(i,j)}$ にその周辺値の線形結合でない値を割り当てます。たとえば、

$$P_{(i,j)} = \text{最大値}(P_{(i-1,j-1)}, P_{(i+1,j-1)}, P_{(i-1,j+1)}, P_{(i+1,j+1)})$$

5×5 の近接ピクセルの場合、 i と j の指標は -2 から 2 までの開きがあります。 $P_{(i,j)}$ とその周辺ピクセルを含む一連のピクセルは、 $P_{(n,m)}$ として表します。

線形フィルタ

画像内の各ピクセル $P_{(i,j)}$ に対し (ここで i と j はピクセルの座標を表します)、コンボリューションカーネルは $P_{(i,j)}$ を中心とします。カーネルによってマスク処理されたピクセルは、その上に配置された係数で乗算します。 $P_{(i,j)}$ は、その積を係数の合計または 1 (いずれか大きい方) で除算したものに なります。

3×3 の近接ピクセルの場合、 $P_{(i,j)}$ の周辺ピクセルとカーネル K は、次のように指標付けすることができます。

$P_{(i-1,j-1)}$	$P_{(i,j-1)}$	$P_{(i+1,j-1)}$
$P_{(i-1,j)}$	$P_{(i,j)}$	$P_{(i+1,j)}$
$P_{(i-1,j+1)}$	$P_{(i,j+1)}$	$P_{(i+1,j+1)}$

$K_{(i-1,j-1)}$	$K_{(i,j-1)}$	$K_{(i+1,j-1)}$
$K_{(i-1,j)}$	$K_{(i,j)}$	$K_{(i+1,j)}$
$K_{(i-1,j+1)}$	$K_{(i,j+1)}$	$K_{(i+1,j+1)}$

ピクセル $P_{(i,j)}$ の値は、 $(1/N)\sum K_{(a,b)}P_{(a,b)}$ になります。ここで a は $(i-1)$ から $(i+1)$ 、 b は $(j-1)$ から $(j+1)$ です。 N は正規化因子で、 $\sum K_{(a,b)}$ または 1 のいずれか大きい方と等しくなります。

新しい値 $P_{(i,j)}$ が負の場合は、0 に設定されます。新しい値 $P_{(i,j)}$ が 255 より大きい場合は、255 に設定されます (8 ビット解像度の場合)。

係数 $K_{(\alpha,b)}$ の絶対値が大きいほど、ピクセル $P_{(\alpha,b)}$ の新規値 $P_{(i,j)}$ に対する影響度は高くなります。係数 $K_{(\alpha,b)}$ が 0 の場合、近接ピクセル $P_{(\alpha,b)}$ は $P_{(i,j)}$ の新規値に影響を与えません ($P_{(\alpha,b)}$ が $P_{(i,j)}$ になることがあります)。

コンボリューションカーネルが

$$\begin{array}{ccc} 0 & 0 & 0 \\ -2 & \mathbf{1} & 2 \\ 0 & 0 & 0 \end{array}$$

の場合、次のようになります。

$$P_{(i,j)} = (-2P_{(i-1,j)} + P_{(i,j)} + 2P_{(i+1,j)})$$

コンボリューションカーネルが

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & \mathbf{0} & 1 \\ 0 & 1 & 0 \end{array}$$

の場合、次のようになります。

$$P_{(i,j)} = (P_{(i,j-1)} + P_{(i-1,j)} + P_{(i+1,j)} + P_{(i,j+1)})$$

非線形プレウィットフィルタ

$$P_{(i,j)} = \max\left(\begin{array}{l} |P_{(i+1,j-1)} - P_{(i-1,j-1)} + P_{(i+1,j)} - P_{(i-1,j)} + P_{(i+1,j+1)} - P_{(i-1,j+1)}|, \\ |P_{(i-1,j+1)} - P_{(i-1,j-1)} + P_{(i,j+1)} - P_{(i,j-1)} + P_{(i+1,j+1)} - P_{(i+1,j-1)}| \end{array} \right)$$

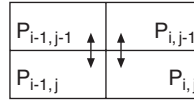
非線形ソーベルフィルタ

$$P_{(i,j)} = \max\left(\begin{array}{l} |P_{(i+1,j-1)} - P_{(i-1,j-1)} + 2P_{(i+1,j)} - 2P_{(i-1,j)} + P_{(i+1,j+1)} - P_{(i-1,j+1)}|, \\ |P_{(i-1,j+1)} - P_{(i-1,j-1)} + 2P_{(i,j+1)} - 2P_{(i,j-1)} + P_{(i+1,j+1)} - P_{(i+1,j-1)}| \end{array} \right)$$

非線形傾斜フィルタ

ピクセルの新しい値は、上隣の近接ピクセルからの偏差と左の 2 つの近接ピクセルの偏差の絶対値の最大値になります。

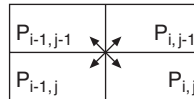
$$P_{(i,j)} = \max(|P_{(i,j-1)} - P_{(i,D)}|, |P_{(i-1,j-1)} - P_{(i-1,D)}|)$$



ロバーツフィルタ

ピクセルの新しい値は、左上の近接ピクセルの偏差とその他の 2 つの近接ピクセルの偏差の絶対値の最大値になります。

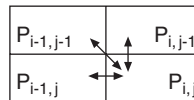
$$P_{(i,j)} = \max(|P_{(i-1,j-1)} - P_{(i,D)}|, |P_{(i,j-1)} - P_{(i-1,D)}|)$$



微分フィルタ

ピクセルの新しい値は、左上の近接ピクセルからの偏差の絶対値の最大値になります。

$$P_{(i,j)} = \max(|P_{(i-1,D)} - P_{(i,D)}|, |P_{(i-1,j-1)} - P_{(i,D)}|, |P_{(i,j-1)} - P_{(i,D)}|)$$



シグマフィルタ

$$\begin{aligned} & \text{If } P_{(i,j)} - M > S, \\ & \text{then } P_{(i,j)} = P_{(i,j)}, \\ & \text{else } P_{(i,j)} = M \end{aligned}$$

$P_{(i,j)}$ とその近接ピクセルの平均値を M 、それらの標準偏差を S とすると、各ピクセル $P_{(i,j)}$ は、値が $(M - S, M + S)$ の範囲内である場合、平均値 M になります。

ローパスフィルタ

$$\begin{aligned} & \text{If } P_{(i,j)} - M < S, \\ & \text{then } P_{(i,j)} = P_{(i,j)}, \\ & \text{else } P_{(i,j)} = M \end{aligned}$$

$P_{(i,j)}$ とその近接ピクセルの平均値を M 、それらの標準偏差を S とすると、各ピクセル $P_{(i,j)}$ は、値が $(M - S, M + S)$ の範囲外である場合、平均値 M になります。

メディアンフィルタ

$P_{(i,j)}$ = 一連のピクセル $(P_{(n,m)})$ の中央値

N 次フィルタ

$P_{(i,j)}$ = 一連のピクセル $(P_{(n,m)})$ の N 次値

ここで $P_{(n,m)}$ は昇順に並べられています。

次の例では、 3×3 の近接ピクセルを使用しています。

13	10	9
12	4	8
5	5	6

次の表は、各 N 次数値の中央ピクセルの新しい出力値を示します。

N 次	0	1	2	3	4	5	6	7	8
新しいピクセル値	4	5	5	6	8	9	10	12	13

特定のフィルタサイズ f に対し、 N 次は 0 から $f^2 - 1$ までの範囲になることができます。たとえば、フィルタサイズ 3 の場合、 N 次は 0 から 8 ($3^2 - 1$) までの範囲になります。

グレースケールモフォロジー

モフォロジー変換は、画像内の粒子のストラクチャを抽出して変更します。それは 2 つのカテゴリに分けられます。

- **バイナリモフォロジー**関数は、バイナリ画像に適用されます。
- **グレースケールモフォロジー**関数は、グレースケール画像に適用されません。

グレースケールモフォロジーでは、ピクセルをその周辺ピクセルと比較して、最小（収縮）または最大（膨張）のピクセル値を保持します。

使用目的

グレースケールモフォロジー関数は、画像のピクセル輝度のフィルタ処理や平滑化に使用します。アプリケーションとしては、ノイズのフィルタ処理、不均等な背景の補正、およびグレースケール特徴抽出などがあります。

グレースケールモフォロジーの概念

グレースケールモフォロジー関数は、グレースケール画像に適用されます。これらの関数を使用して、明るい領域を拡張し暗い領域を縮小したり、またはその逆をすることにより、領域の形状を変更することができます。これらの関数を使用すると、徐々に変化するパターンが滑らかになり、境界部分でのコントラストが強まります。このセクションでは、次のグレースケールモフォロジー関数について説明します。

- 収縮
- 膨張
- オープン
- クローズ

- 適正オープン
- 適性クローズ
- 自動メディアン

これらの関数は、構造要素を使用するグレーレベル収縮および膨張を組み合わせたものです。

収縮関数

グレーレベル収縮は、輝度の低い近接ピクセルに囲まれたピクセルの輝度を下げます。近接ピクセルは、構造要素によって定義されます。

膨張関数

輝度の高い近接ピクセルに囲まれたピクセルの輝度を上げます。近接ピクセルは、構造要素によって定義されます。グレーレベル膨張は、グレーレベル収縮とは逆の効果があり、明るい領域を膨張すると暗い領域が収縮されます。

収縮と膨張の例

この例では以下のソース画像を使用します。

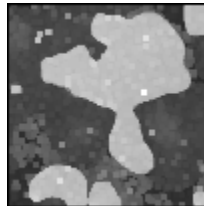
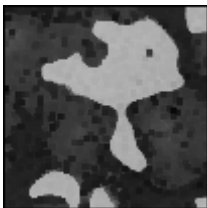
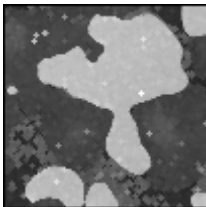
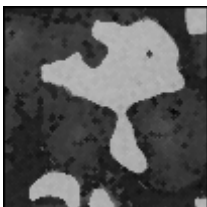
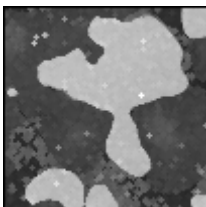


表 5-3 は、構造要素の例と、それに対応する収縮および膨張処理された画像を示します。

表 5-3 収縮と膨張の例

構造要素	収縮	膨張
<pre> 1 1 1 1 1 1 1 1 1 </pre>		
<pre> 0 1 0 1 1 1 0 1 0 </pre>		

オープン関数

グレーレベル**オープン関数**は、グレーレベル収縮の後にグレーレベル膨張を実行するものです。それにより、暗い領域内に孤立している明るい点を取り除かれ、境界線が滑らかになります。この関数の効果は、構成している要素の構成によって加減されます。

$$\text{オープン}(f) = \text{膨張}(\text{収縮}(f))$$

収縮と膨張は反対のモフォロジー処理であるため、この操作によって粒子の形状が大きく変わることはありません。収縮によって縮小された明るい境界部分は、膨張で復元されます。ただし、収縮で消滅した小さな明るい粒子は、膨張によっても復元されません。

クローズ関数

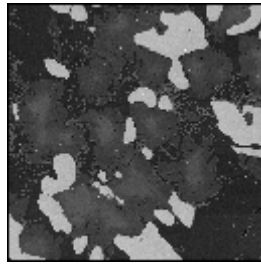
グレーレベルクローズ関数は、グレーレベル膨張の後にグレーレベル収縮を実行するものです。それにより、明るい領域内に孤立している暗い点を取り除かれ、境界線が滑らかになります。この関数の効果は、構造要素の配置によって加減されます。

$$\text{クローズ}(I) = \text{収縮}(\text{膨張}(I))$$

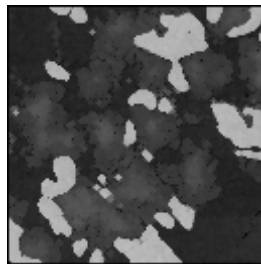
膨張と収縮は反対のモフォロジー処理であるため、この操作によって粒子の形状が大きく変わることはありません。膨張によって拡張された明るい境界部分は収縮で縮小されます。ただし、収縮で消滅した小さな暗い粒子は、膨張によっても復元されません。

オープンとクローズの例

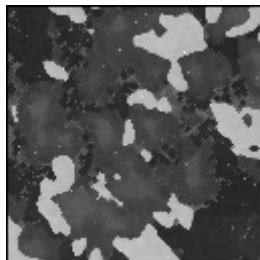
この例では以下のソース画像を使用します。



オープン関数により、以下のような画像が生成されます。



クローズ関数により、以下のような画像が生成されます。



メモ

オープンまたはクローズ関数を連続して適用しても、結果は常に同じになります。

適正オープン関数

グレースケール**適正オープン関数**は、オープンおよびクローズの有限的な組み合わせです。この関数は暗い領域内に孤立する明るいピクセルを取り除き、明るい領域の境界線を滑らかにします。この関数の効果は、構造要素の配置によって加減されます。

適正クローズ関数

適正クローズ関数は、クローズおよびオープンの有限的な組み合わせです。この関数は明るい領域内に孤立する暗いピクセルを取り除き、暗い領域の境界線を滑らかにします。この関数の効果は、構造要素の配置によって加減されます。

自動メディアン関数

自動メディアン関数は、オープンとクローズの組み合わせを使用します。この関数は、細かい部分の少ないよりシンプルな粒子を生成します。

詳細の説明

収縮の概念および数学的演算

画像内の各ピクセルがその近接ピクセルの最小値と等しくなります。

特定のピクセル P_0 に対し、構造要素は P_0 を中心とします。そして、1 に等しい構造要素の係数によってマスク処理されたピクセルは、 P_i となります。

$$P_0 = \min(P_i)$$



メモ

すべての係数を 1 に設定して、構造要素 $f \times f$ を使用したグレーレベル収縮を行うのは、フィルタサイズを $f \times f$ 、値 N を 0 として N 次フィルタ処理を行うのと同じです。詳細については、本章の「[非線形フィルタ](#)」のセクションを参照してください。

膨張の概念および数学的演算

画像内の各ピクセルは、その近接ピクセルの最大値と等しくなります。

特定のピクセル P_0 に対し、構造要素は P_0 を中心とします。そして、1 に等しい構造要素の係数によってマスク処理されたピクセルは、 P_i となります。

$$P_0 = \max(P_i)$$



メモ

すべての係数を 1 に設定して、構造要素 $f \times f$ を使用したグレーレベル膨張を行うのは、フィルタサイズを $f \times f$ 、値 N を $f^2 - 1$ として N 次フィルタ処理を行うのと同じです（[非線形空間フィルタ](#)を参照）。詳細については、本章の「[非線形フィルタ](#)」のセクションを参照してください。

適正オープンの概念と数学的演算

I が元の画像の場合、適正オープン関数は元の画像 I と、オープン、クローズ、さらにもう 1 回オープンを行うことによって得られる変換後の画像との間の各ピクセルの最小値を抽出します。

$$\begin{aligned} \text{適正オープン}(I) &= \min(I, \text{OCO}(I)) \\ \text{または適正オープン}(I) &= \min(I, \text{DEEDDE}(I)) \end{aligned}$$

ここで I は元の画像、
 E は収縮、
 D は膨張、
 O はオープン、
 C はクローズ、
 $F(I)$ は画像 I に関数 F を適用した後に得られる画像、そして
 $GF(I)$ は画像 I に関数 F と、次に関数 G を適用した後に得られる画像です。

適正クローズの概念と数学的演算

I が元の画像の場合、適正クローズ関数は元の画像 I と、クローズ、オープン、さらにもう 1 回クローズを行うことによって得られる変換後の画像との間の各ピクセルの最大値を抽出します。

$$\begin{aligned} \text{適性クローズ}(I) &= \max(I, \text{COC}(I)) \\ \text{または適性クローズ}(I) &= \max(I, \text{EDDEED}(I)) \end{aligned}$$

ここで I は元の画像、
 E は収縮、
 D は膨張、
 O はオープン、
 C はクローズ、
 $F(I)$ は画像 I に関数 F を適用した後に得られる画像、そして
 $GF(I)$ は画像 I に関数 F と、次に関数 G を適用した後に得られる画像です。

自動メディアン の概念と数学的演算

I が元の画像の場合、自動メディアン関数は元の画像 I に適正オープンと適正クローズを適用して得られる 2 つの画像の間のピクセルの最小値を抽出します。

$$\text{自動メディアン}(I) = \min(\text{OCO}(I), \text{COC}(I))$$

$$\text{または自動メディアン}(I) = \min(\text{DEEDDE}(I), \text{EDDEED}(I))$$

ここで I は元の画像、
 E は収縮、
 D は膨張、
 O はオープン、
 C はクローズ、
 $F(I)$ は画像 I に関数 F を適用した後に得られる画像、そして
 $GF(I)$ は画像 I に関数 F と、次に関数 G を適用した後に得られる画像です。

演算子

本章では、画像をマスク処理、合成、比較するための算術および論理演算子について説明します。

概要

演算子とは、画像に対して基礎的な算術および論理演算を実行するものです。演算子を使用して、画像と他の画像または定数との間で加算、減算、乗算、および除算を実行します。また、論理積 / 否定論理積、論理和 / 否定論理和、排他的論理和 / 排他的否定論理和などの論理演算や、画像と他の画像や定数とのピクセルの比較を実行することもできます。

使用目的

そのような演算子の一般的なアプリケーションとしては、時間遅延比較、画像間での結合や共通部分の識別、画像の背景の補正による光のドリフトの除去、および複数の画像とモデルとの間の比較などがあります。さらに、演算子を使用すると、画像を二値化またはマスク処理してコントラストや輝度を変更することもできます。

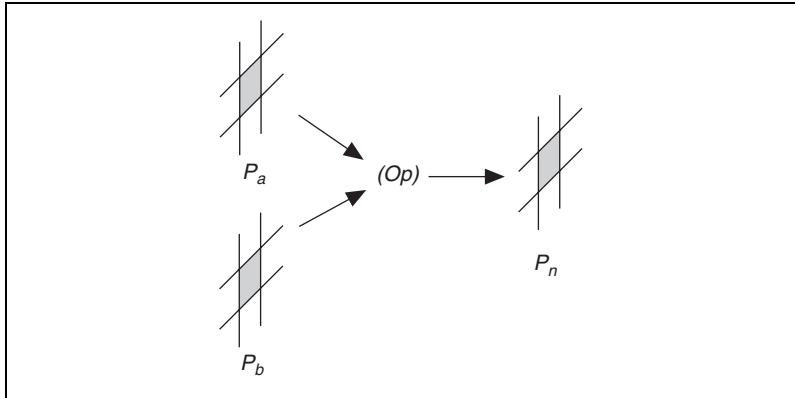
演算子の概念

画像間での算術または論理演算は、ピクセルごとの変換です。それによって生成される画像では、各ピクセルの値は他の画像の同じ座標のピクセル値から引き出されています。

A が解像度 XY の画像、 B が解像度 XY の画像、 Op が演算子の場合、 A と B に演算子 Op を適用した結果得られる画像 N では、各ピクセルに次の値が割り当てられています。

$$p_n = (p_a)(Op)(p_b)$$

ここで p_a は画像 A のピクセルの値で、 p_b は画像 B のピクセルの値です。



算術演算

表 6-1 の方程式は、8 ビット解像度の画像 a と b での算術演算の使用方法を示します。

表 6-1 算術演算

演算子	方程式
乗算	$p_n = \min(p_a \times p_b, 255)$
除算	$p_n = \max(p_a / p_b, 0)$
加算	$p_n = \min(p_a + p_b, 255)$
減算	$p_n = \max(p_a - p_b, 0)$
剰余	$p_n = p_a \text{ Mod } p_b$
絶対差分	$p_n = p_a - p_b $

生成されたピクセル値 p_n が負の場合は、0 に設定されます。255 より大きい場合は、255 に設定されます。

論理および比較演算

論理演算子は、ビットごとの演算です。論理演算子は、ビットレベルで 1 バイトにコード化されたグレーレベル値を操作します。表 6-2 の方程式は、論理演算子の使用方法を記述するものです。論理演算の真理値表は、本章の「真理値表」のセクションにあります。

表 6-2 論理および比較演算

演算子	方程式
論理演算子	
論理積	$p_n = p_a \text{ AND } p_b$
否定論理積	$p_n = p_a \text{ NAND } p_b$
論理和	$p_n = p_a \text{ OR } p_b$
否定論理和	$p_n = p_a \text{ NOR } p_b$
排他的論理和	$p_n = p_a \text{ XOR } p_b$
論理差分	$p_n = p_a \text{ AND } (\text{NOT } p_b)$
比較演算	
マスク	$p_b = 0$ 、の場合 $p_n = 0$ 、 それ以外の場合 $p_n = p_a$
平均	$p_n = \text{mean}(p_a, p_b)$
最大値	$p_n = \text{max}(p_a, p_b)$
最小値	$p_n = \text{min}(p_a, p_b)$

8 ビット解像度の画像の場合、論理演算子は主に、グレースケール画像を 0 または 255 のピクセル（バイナリ形式では 0 は 00000000、255 は 11111111 と表されます）からなるマスク画像と合成するか、画像をバイナリまたはラベル付きの内容（画像を二値化後）と合成または比較することを目的としています。

表 6-3 は、論理演算子を使用して画像内の情報を抽出したり削除したりする方法を示します。

表 6-3 バイナリ画像マスクでの論理演算子の使用

特定の p_a に対し	$p_b = 255$ の場合	$p_b = 0$ の場合
論理積	$p_a \text{ AND } 255 = p_a$	$p_a \text{ AND } 0 = 0$
否定論理積	$p_a \text{ NAND } 255 = \text{NOT } p_a$	$p_a \text{ NAND } 0 = 255$
論理和	$p_a \text{ OR } 255 = 255$	$p_a \text{ OR } 0 = p_a$
否定論理和	$p_a \text{ NOR } 255 = 0$	$p_a \text{ NOR } 0 = \text{NOT } p_a$
排他的論理和	$p_a \text{ XOR } 255 = \text{NOT } p_a$	$p_a \text{ XOR } 0 = p_a$
論理差分	$p_a - \text{NOT } 255 = p_a$	$p_a - \text{NOT } 0 = 0$

真理値表

次の真理値表は、論理演算で使用される規則を記述したものです。一番上の行と左の列は、入力ビットの値を示します。表のセルは、2つの入力ビットの組み合わせに対する出力値を示します。

AND		
	$b=0$	$b=1$
$a=0$	0	0
$a=1$	0	1

NAND		
	$b=0$	$b=1$
$a=0$	1	1
$a=1$	1	0

OR		
	$b=0$	$b=1$
$a=0$	0	1
$a=1$	1	1

NOR		
	$b=0$	$b=1$
$a=0$	1	0
$a=1$	0	0

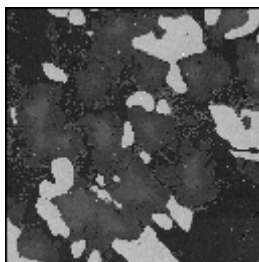
XOR		
	$b=0$	$b=1$
$a=0$	0	1
$a=1$	1	0

XNOR		
	$b=0$	$b=1$
$a=0$	1	0
$a=1$	0	1

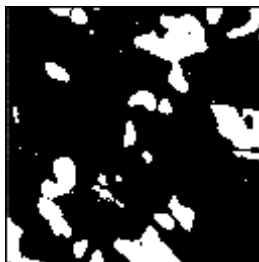
NOT	
	NOT a
$a=0$	1
$a=1$	0

例 1

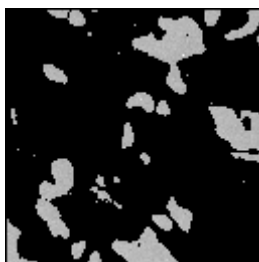
次の図は、この例で使用する元のグレースケール画像を示します。



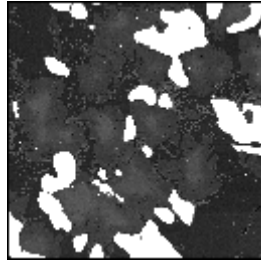
バイナリ形式で関心領域を分離し、モフォロジー操作で画像処理して、最後に 255 を乗算して次のマスク画像を生成します。



演算 (元の画像 AND マスク画像) により、マスク内のオブジェクト領域の本来の輝度が復元されます。



演算 (元の画像 OR マスク画像) により、マスク内の背景領域の本来の輝度が復元されます。

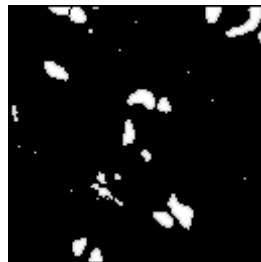


例 2

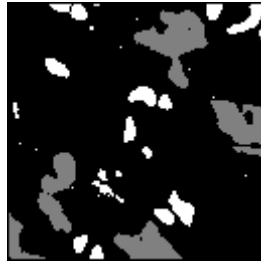
この例では、論理和演算を使用して 2 つのバイナリ画像の結合を含む画像を生成する方法を示します。次の画像は、背景が 0、オブジェクトがグレーレベル値 128 の、最初の画像を表します。



次の画像は、背景が 0、オブジェクトがグレーレベル値 255 の、2 つめの画像を示します。



画像 #1 OR 画像 #2 により、次の画像に示すような結合画像が生成されます。



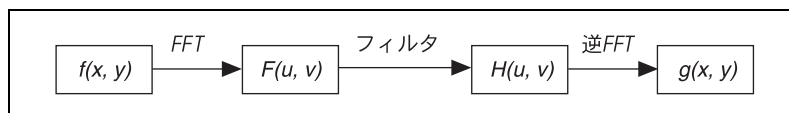
周波数領域解析

本章では、高速フーリエ変換を使用した画像の周波数領域への変換、および周波数領域での画像の解析と処理について説明します。

概要

周波数フィルタは、画像内の輝度変化の周期性と空間分布に基づいてピクセル値を変更します。空間フィルタと異なり、周波数フィルタは、空間的画像ではなく、その周波数表現に対して直接適用されます。画像の周波数表現は、**高速フーリエ変換 (FFT)** という関数を使用して生成します。それにより、元の画像に見られるパターンの周期性と分散に関する情報が得られます。

FFT 画像に見られる空間周波数をフィルタ処理することができます。そして、逆 FFT 関数によってフィルタ処理後の FFT 画像の空間表現を復元します。



周波数処理は、画像から情報を抽出するためのもう 1 つのテクニックです。輝度変化の位置と方向を使用する代わりに、周波数処理を使用して、空間領域でそれらの変化が発生する周波数に対し操作することができます。この新しい成分は**空間周波数**といい、画像内の輝度が空間座標に比例して変化する周波数のことをさします。

画像の空間周波数は、高速フーリエ変換 (FFT) を使用して計算します。FFT の計算には 2 つのステップがあります。まず行の一次元高速フーリエ変換を行い、次にその結果の列の一次元高速フーリエ変換を行います。FFT プレーンを形成する複素数は、複素画像と呼ばれる 64 ビット浮動小数点画像にコード化されます。複素画像は、実数部分を表す 32 ビット浮動小数点数と、虚数部分を表す 32 ビット浮動小数点数で形成されます。

画像において、詳細部分と鋭いエッジは短い距離内で著しいグレーレベル変化があることから、中から高空間周波数に関連付けられます。徐々に変化するパターンは低空間周波数に関連付けられます。空間周波数をフィルタ処理することによって、関連する空間コンポーネントを削除、減衰、またはハイライトすることが可能です。

ローパス周波数フィルタを使用して、画像内に存在する高周波数を減衰または削除（トランケート）します。このフィルタは、空間画像での輝度の急激な変化に関連する情報を抑圧します。ローパス周波数フィルタの後に使用する逆 FFT は、ノイズ、詳細、テクスチャ、および鋭いエッジが平滑化された画像を生成します。

ハイパス周波数フィルタは、複素画像に存在する低周波数を減衰または削除（トランケート）します。このフィルタは、空間画像での輝度の緩い変化に関連する情報を抑圧します。この場合、ハイパス周波数フィルタの後に使用する逆 FFT は、パターン全体が鋭く、詳細が強調された画像を生成します。

マスク周波数フィルタは、ユーザが指定したマスクに含まれる周波数を削除します。マスクを使用して画像のフーリエ変換を変更すると、ローパスフィルタやハイパスフィルタを適用するよりさらに多くのことが可能になります。画像マスクはユーザによって作成され、画像内のきわめて特定の周波数および方向を表すことができます。たとえば、このテクニックを応用して、周波数領域の優位周波数や高調波をフィルタ処理することができます。

使用目的

画像で、詳細部分と鋭いエッジは中から高空間周波数に関連付けられます。それは、詳細部分や鋭いエッジでは短い距離内で著しいグレーレベル変化が見られるためです。徐々に変化するパターンは、低空間周波数に関連付けられます。

デジタル処理プロセスで、周期的な縞模様など無関係のノイズが発生することがあります。周波数領域では、周期的パターンは限られた数の高空間周波数に減少されます。そのような特定の周波数をトランケートしてフィルタ処理後の FFT 画像を空間領域に戻すと、新しい画像が生成され、その画像ではグリッドパターンが消滅し、全体的な特徴は維持されます。

高速フーリエ変換の概念

画像のFFTは、複素数の2次元配列で、複素画像としても表されます。これは、空間領域での輝度変化の周波数を表現するものです。低周波数は、元の画像の全体的なパターンに見られる滑らかで緩やかな輝度変化に対応します。高周波数は、オブジェクトのエッジ、ノイズの多いピクセル周辺、および詳細部分周辺に見られる急激で短い輝度変化に対応します。

FFT 表現

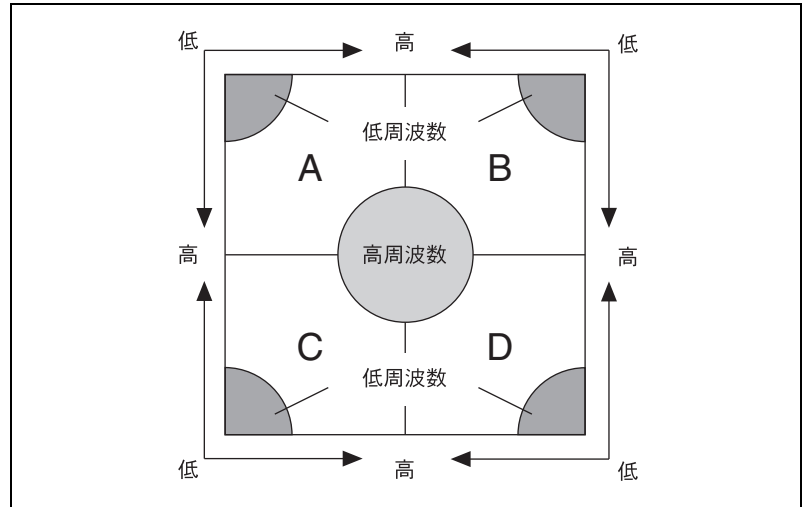
画像の高速フーリエ変換には、**標準表現**と**光学表現**の2つの表現方法があります。

標準表現

標準表現では、高周波数を画像の中央にまとめ、低周波数を端に配置します。定数項、つまりヌル周波数は、画像の左上コーナー部分になります。周波数範囲は以下の通りです。

$$[0, N] \times [0, M]$$

ここで M は画像の水平方向の解像度で、
 N は垂直方向の解像度です。



メモ IMAQ Vision では、この表現を使用してメモリ内の複素画像を表します。画像マスクを構築する際はこの表現を使用してください。

図 7-1a は、画像を示します。図 7-1b は、同じ画像の FFT を標準表現を使用して表したものです。

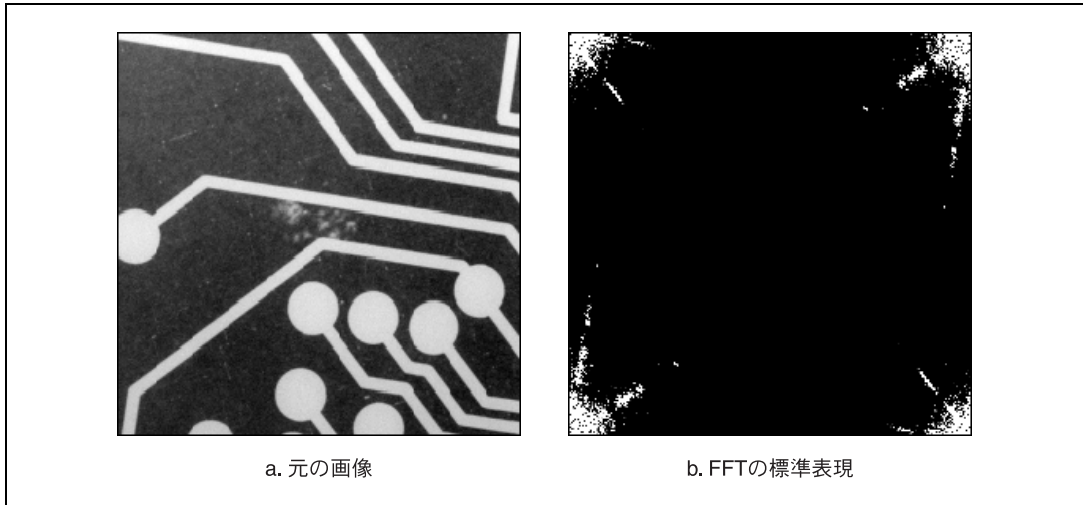


図 7-1 標準表現で表した画像の FFT

光学表現

光学表現では、低周波数を画像の中央にまとめ、高周波数を端に配置します。定数項、つまりヌル周波数は、画像の中央になります。周波数範囲は以下の通りです。

$$\left[-\frac{N}{2}, \frac{N}{2} \right] \times \left[-\frac{M}{2}, \frac{M}{2} \right]$$

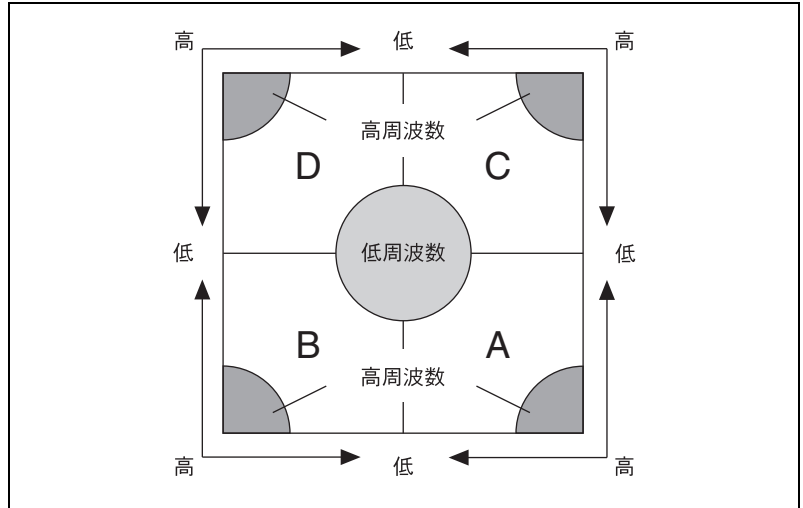


図 7-2a は、図 7-1a と同じ元の画像を示します。図 7-2b は、同じ画像の FFT を光学表現を使用して表したものです。

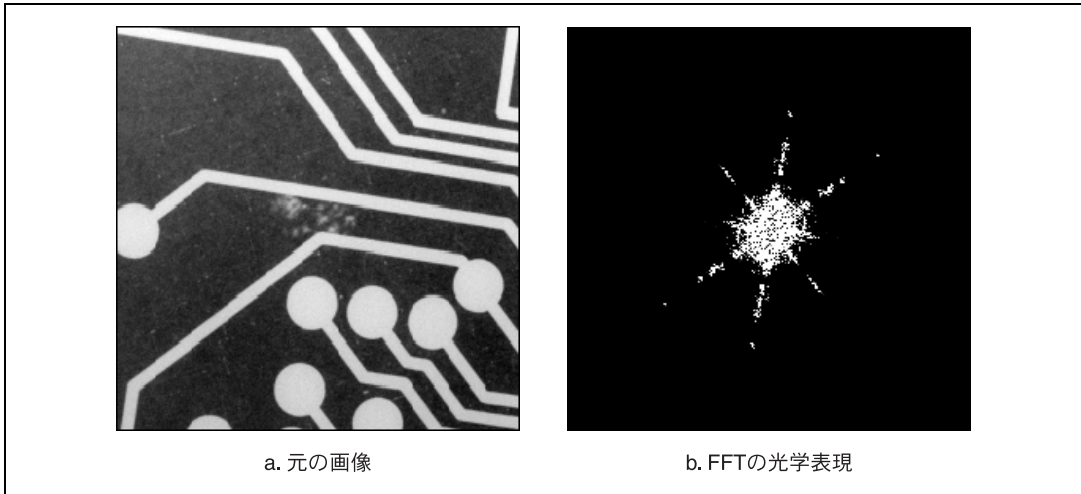


図 7-2 光学表現で表した画像の FFT



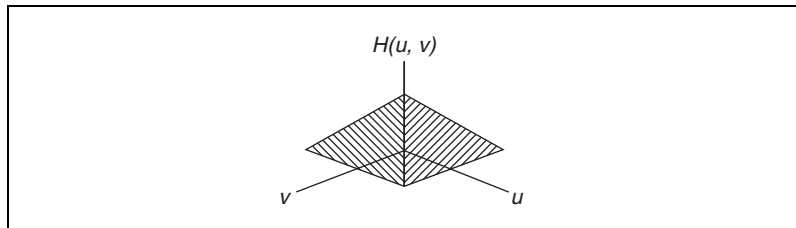
メモ この表現方法は、IMAQ Vision で複素画像を表示する際に使用するものです。

標準表現から光学表現に切り替えるには、A、B、C、および D の各領域の順序を変更します。

FFT 画像の輝度は、表示されている成分の振幅に比例します。

ローパス FFT フィルタ

ローパス周波数フィルタは、FFT プレーンに存在する高周波数を減衰または削除します。このフィルタは、空間画像での輝度の急激な変化に関連する情報を抑圧します。この場合、逆 FFT は、ノイズ、詳細部分、テクスチャ、および鋭いエッジが平滑化された画像を生成します。



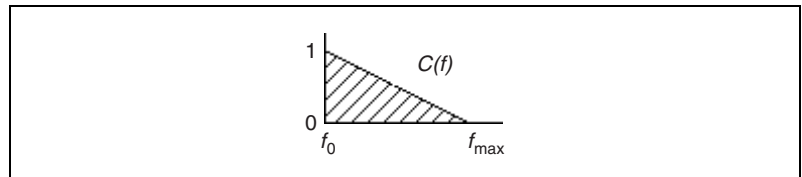
ローパス周波数フィルタは、基本（またはヌル）周波数を中心とした周波数範囲外に位置する空間周波数を削除または減衰します。

ローパス減衰

ローパス減衰は、全周波数範囲に線形減衰を適用し、ヌル周波数 f_0 から最大周波数 f_{max} に上げます。これを行うには、各周波数に、基本周波数および最大周波数からの偏差の関数である係数 C を乗算します。

$$C(f) = \frac{f_{max} - f}{f_{max} - f_0}$$

ここで、 $C(f_0) = 1$ 、 $C(f_{max}) = 0$ です。



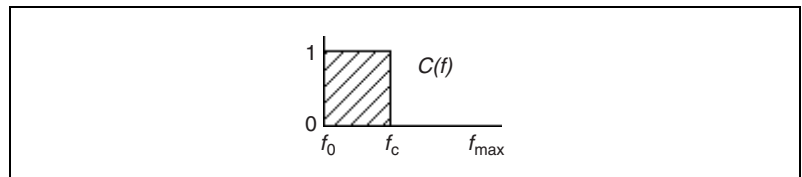
ローパストランケーション

ローパストランケーションは、遮断周波数またはトランケーション周波数 f_c より高い場合に、周波数 f を削除します。そのためには、周波数 f に係数 C を乗算します。係数 C は、周波数 f がトランケーション周波数 f_c より大きいかどうかによって 0 または 1 になります。

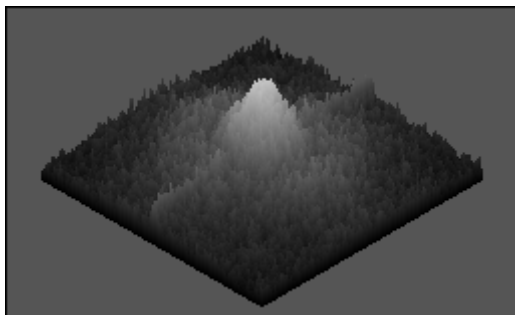
if $f > f_c$

then $C(f) = 0$

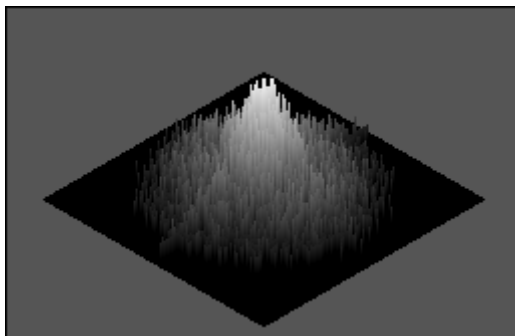
else $C(f) = 1$



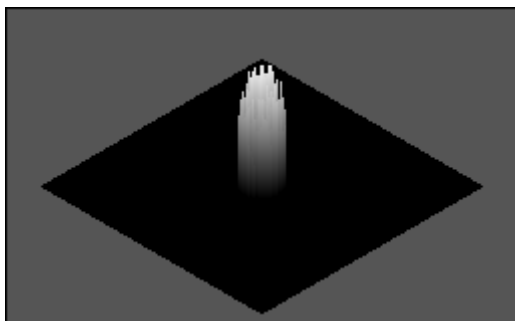
以下の一連の図は、両方のタイプのローパスフィルタの動作を示します。FFT の振幅を 3 次元表示のプロファイルで示したものです。この例では以下の元の FFT を使用します。



ローパス減衰の後、中央のピークの振幅は同じで、端部分の変動がなくなっています。

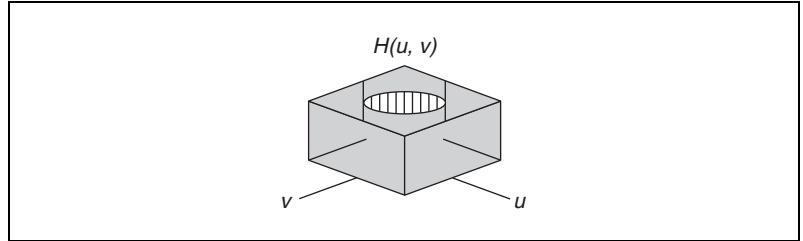


$f_c = f_0 + 20\%(f_{\max} - f_0)$ でローパストランケーションを行った後、 (f_0, f_c) のトランケーション範囲外の空間周波数は削除されます。中央のピークの残っている部分は、元の FFT プレーンのもと同様です。



ハイパス FFT フィルタ

ハイパス FFT フィルタは、FFT プレーンに存在する低周波数を減衰または削除します。このフィルタは、空間画像での輝度の緩い変化に関連する情報を抑圧する効果があります。この場合、逆 FFT コマンドを使用すると、全体的なパターンが減衰され詳細が強調された画像が生成されます。

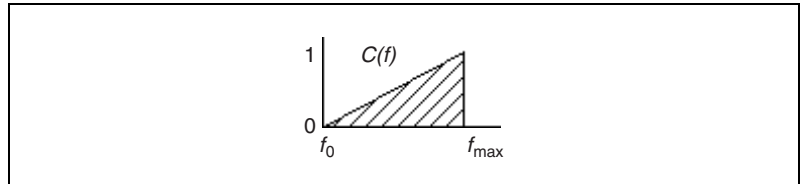


ハイパス減衰

ハイパス減衰は、全周波数範囲に線形減衰を適用し、最大周波数 f_{\max} からヌル周波数 f_0 に上げます。これを行うには、各周波数に、基本周波数および最大周波数からの偏差の関数である係数 C を乗算します。

$$C(f) = \frac{f - f_0}{f_{\max} - f_0}$$

ここで、 $C(f_0) = 1$ 、 $C(f_{\max}) = 0$ です。



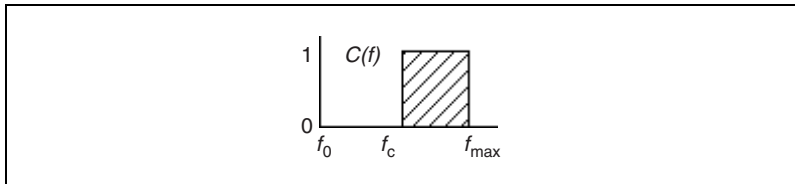
ハイパストランケーション

ハイパストランケーションは、遮断周波数またはトランケーション周波数 f_c より低い場合に、周波数 f を削除します。そのためには、周波数 f に係数 C を乗算します。係数 C は、周波数 f がトランケーション周波数 f_c より大きいかどうかによって 1 または 0 になります。

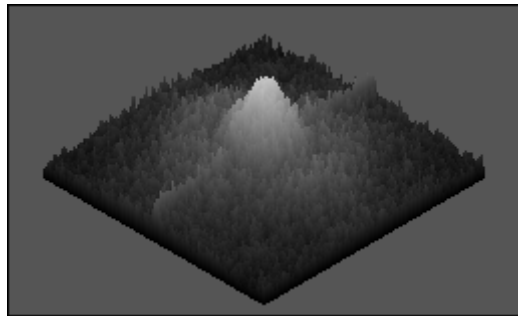
if $f < f_c$

then $C(f) = 0$

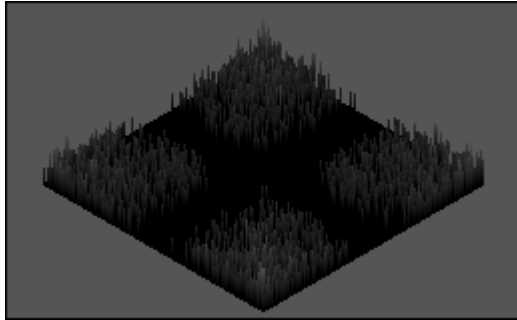
else $C(f) = 1$



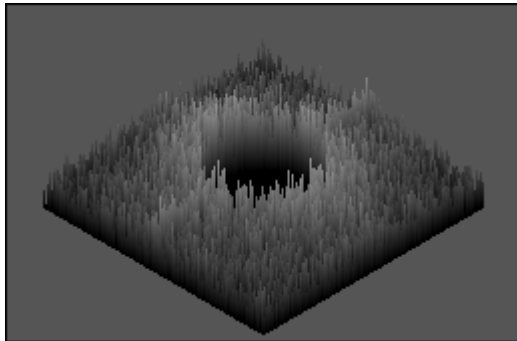
以下の一連の図は、両方のタイプのハイパスフィルタの動作を示します。FFT の振幅を三次元表示のプロファイルで示したものです。この例では以下の元の FFT を使用します。



ハイパス減衰を行った後、中央のピークは削除され、端の変動部分は残ったままです。

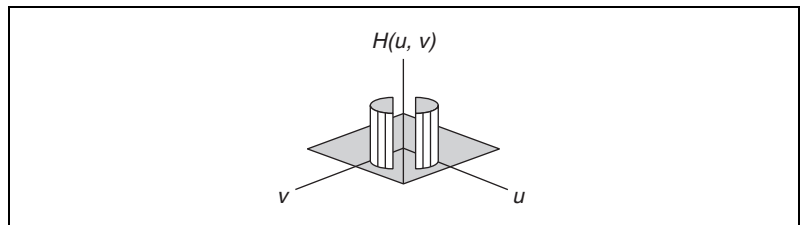


$f_c = f_0 + 20\%(f_{\max} - f_0)$ でハイパストランケーションを行った後、 (f_0, f_c) のトランケーション範囲内の空間周波数は 0 に設定されます。その他の周波数は、元の FFT プレーンの周波数とまったく同じです。



マスク FFT フィルタ

マスク FFT フィルタは、ユーザが指定したマスクに含まれる周波数を削除します。マスクの定義により、このフィルタはローパス、バンドパス、ハイパス、または任意のタイプの選択式フィルタとして動作することができます。



詳細の説明

フーリエ変換

画像の空間周波数は、**フーリエ変換**という関数で計算します。これは連続領域で以下のように定義します。

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(xu + yv)} dx dy$$

ここで、 $f(x, y)$ は点 (x, y) の輝度、 (u, v) は水平および垂直の空間周波数です。フーリエ変換で、各セット (u, v) に複素数を割り当てます。

逆に、高速フーリエ変換 $F(u, v)$ は、次の式を使って、解像度 NM の空間画像 $f(x, y)$ に変換することができます。

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)}$$

不連続領域では、フーリエ変換は高速フーリエ変換 (FFT) という効率的なアルゴリズムを使って計算します。

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)}$$

ここで、 $N \times M$ は空間画像 $f(x, y)$ の解像度です。

$e^{-j2\pi ux} = \cos 2\pi ux - j \sin 2\pi ux$ であるため、 $F(u, v)$ は、正弦項と余弦項の無限和からなります。各組み合わせ (u, v) により、対応する正弦項と余弦項の対の周波数が決まります。ある特定の対 (u, v) に対し、すべての値 $f(x, y)$ が $F(u, v)$ に作用している点に注意してください。そのように複雑なものであるため、FFT の計算は時間がかかります。

解像度 $N \times M$ の画像と、空間ステップ増分 Δx および Δy の場合、元の画像の FFT の解像度は同じ NM になり、周波数ステップ増分 Δu および Δv は次の方程式で求められます。

$$\Delta u = \frac{1}{N \times \Delta x} \quad \Delta v = \frac{1}{M \times \Delta y}$$

FFT 表示

FFT 画像は、実数部、虚数部、振幅、または位相の4つの複素成分のいずれかを使って視覚化することができます。それらの成分間の関係は、次の2つの式で表すことができます。

$$F(u, v) = R(u, v) + jI(u, v)$$

ここで、 $R(u, v)$ は実数部、 $I(u, v)$ は虚数部です。

$$F(u, v) = |F(u, v)| \times e^{j\varphi(u, v)}$$

ここで、 $|F(u, v)|$ は振幅、 $\varphi(u, v)$ は位相です。

$F(u, v)$ の振幅のことをフーリエスペクトルともいい、次の式に等しくなります。

$$|F(u, v)| = \sqrt{R(u, v)^2 + I(u, v)^2}$$

2の累乗に対するフーリエスペクトルは、パワースペクトルまたはスペクトル密度と呼ばれます。

位相 $\varphi(u, v)$ は位相角ともいい、次の式に等しくなります。

$$\varphi(u, v) = \text{atan} \left[\frac{I(u, v)}{R(u, v)} \right]$$

複素画像を表示する際、複素画像の振幅プレーンは、デフォルトで光学表現を使って表示されます。振幅値を正しく視覚化するために、振幅値は表示前に因数 m によってスケールされます。因数 m は、次の方程式で求められます。

$$\frac{128}{w \times h}$$

ここで w は画像の幅、
 h は画像の高さを表します。

第 III 部

粒子解析

このセクションでは、二値化、モフォロジー、粒子測定などを含む、粒子解析の概念について説明します。

第 III 部、「粒子解析」には、以下の章が含まれます。

第 8 章、「[二値化](#)」では、二値化およびカラー二値化について説明します。

第 9 章、「[バイナリモフォロジー](#)」では、構造要素、連結性、基礎および上級モフォロジー変換について説明します。

第 10 章、「[粒子測定](#)」では、粒子の面積、長さ、座標、弦と軸、形状の等価および特徴、およびさまざまな測定について説明します。

概論

[粒子](#)とは、画像内にあるすべてのピクセルが同じ論理状態にある、互いに接触したピクセルの領域です。画像内で粒子に属するピクセルは、すべて前景状態にあるといます。その他のすべてのピクセルは背景状態にあります。バイナリ画像では、背景のピクセルは 0 に等しい値を持ち、0 以外の値を持つピクセルはすべてバイナリのオブジェクトに属します。

粒子解析を使用して、画像内でピクセルの連結した領域やまとまった部分を検出し、それらの領域に対して選択した測定を行います。そのような領域のことを一般的に[粒子](#)といます。

粒子解析は、画像内の粒子に関する情報を生成する一連の処理操作と解析関数からなります。粒子解析を使用すると、画像内にある任意の 2 次元形状を検出し解析することができます。

使用目的

ある特定の基準を満たす空間的特徴を持つ粒子を検出する場合に、粒子解析を使用します。計算に時間のかかる多くのアプリケーションで、粒子解析を使用して空間的特徴が基準に満たない粒子を取り除き、重要性の高い粒子のみを以降の解析に使用することができます。

粒子解析を使用すると、粒子のサイズや数、位置、粒子領域の有無などの空間情報を検出することもできます。そのような情報を使用して、さまざまなマシンビジョン検査タスクを実行することができます。そのタスクには、シリコンウェハの欠陥検出、電子基板上のはんだ不良の検出、木版の構造的欠陥やプラスチック板のひびを検出するウェブ検査アプリケーションなどがあります。部品の形状や向きに有意な分散がある場合、モーショントラッキングアプリケーションでオブジェクトを検出することができます。

オブジェクトの形状や向きに有意な分散が見られるアプリケーションでは、粒子解析はオブジェクトを検索するための柔軟かつ強力なツールです。粒子解析によって得られた測定結果を組み合わせ使用し、オブジェクトの形状を固有に定義する特徴を見つけることができます。

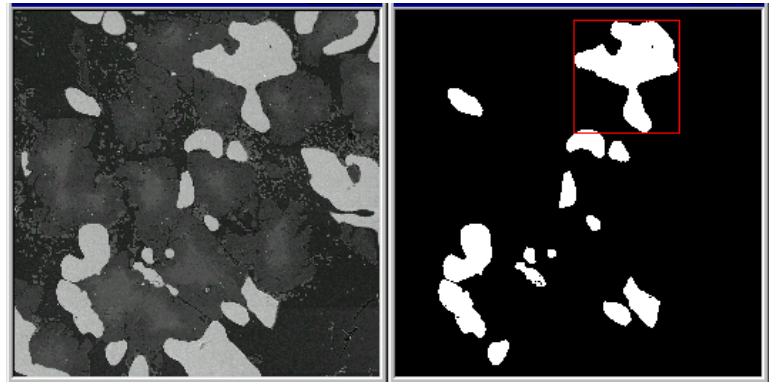
粒子解析の概念

一般的な粒子解析プロセスでは、画像全体をスキャンし、画像内にあるすべての粒子を検出して、各粒子に関する詳細なレポートを作成します。通常このレポートには、画像内での粒子の位置、サイズ、形状、他の粒子に対する向き、最長セグメント、慣性モーメントなど、その粒子に関する 50 種類ほどの情報が含まれています。

周囲、角度、面積、重心など複数のパラメータを使用して、それらの粒子を識別し分類することができます。多くのアプリケーションで、複数のパラメータを使用した方がパターンマッチングより速くより効果的に結果を出すことができます。

また、異なる組み合わせのパラメータを使用することにより、画像内の特徴を固有に識別することが可能です。例えば、テンプレートの粒子のサイズを基準として使用し、許容範囲に入らないすべての粒子を取り除くことができます。そして、さらに別のパラメータの許容範囲を使用することによって、残りの粒子をさらに絞り込むことができます。それには、各粒子の最長セグメントや分散度の要因（粒子を囲む最小四角形の面積に対する、粒子の面積の割合）などのパラメータを使用できます。

次の図は、粒子解析アプリケーションで取得できるパラメータリストのサンプルを示します。この例のバイナリ画像は、元の画像を二値化して、画像の縁にある粒子を取り除くことにより得られたものです。これらのパラメータを使用して、粒子を識別し分類することができます。以下の表は、図に表示されている四角形に囲まれた粒子に関する値を示します。



境界四角形：	
x1 左辺	125
y2 上辺	7
x2 右辺	198
y2 底辺	85
粒子の面積（ピクセル単位）	2456
穴の数	1
面積（キャリブレーション済み）	2456.00
穴の面積（ピクセル）	2
X 軸の和	406482.00
X 軸の二乗の和	67885136.00
Y 軸の和	89909.00
Y 軸の二乗の和	4158045.00
X 軸と Y 軸の和	14856285.00
投影 x	99
投影 y	94
周囲	289.02

穴の周囲	5.01
最長セグメントの座標 (x および y)	$x = 125$ $y = 36$
最長セグメントの長さ	72

粒子解析を使用するには、まず二値化によってバイナリ画像を作成してください。そして、モフォロジー変換を使用してバイナリ画像の質を向上させ、画像内の粒子に対し測定を行います。

二値化

本章では、二値化およびカラー二値化について説明します。

概論

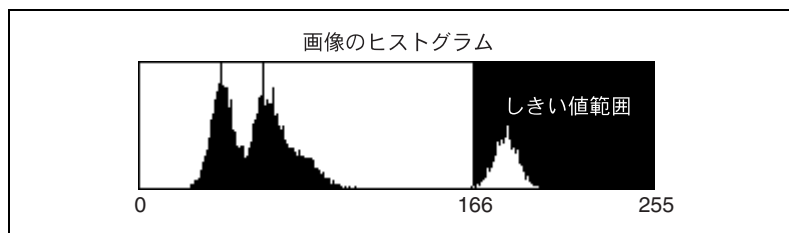
二値化とは、画像を粒子領域と背景領域の2つの領域にセグメント化することです。このプロセスでは、しきい値範囲というグレースケール範囲に属するピクセルをすべて1に設定し、その他のピクセルをすべて0に設定します。

二値化を使用して、画像内の関心対象オブジェクトを分離することができます。二値化を行うと、0から255までのピクセル値を持つグレースケール画像が、ピクセル値0または1のバイナリ画像に変換されます。

二値化により、グレースケールおよびカラー画像のピクセル値範囲を選択して、処理対象のオブジェクトを背景から分離することができます。

使用目的

画像内の有意構造体に対応する領域を抽出し、その領域に対して集中的に解析を行う際に、二値化を使用します。



しきい値範囲外のピクセルは0に設定され、背景領域の一部とみなされます。しきい値範囲内のピクセルは1に設定され、粒子領域の一部とみなされます。

二値化の概念

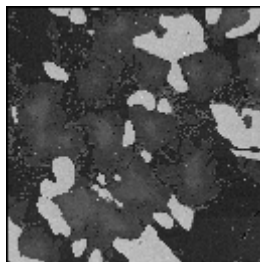
輝度二値化

粒子は、**輝度範囲**によって定義されます。ある特定のしきい値（全体的な光度またはグレイの階調）範囲のグレイレベル値を持つピクセルからなるのが粒子です。その他のピクセルはすべて背景とみなされます。

しきい値範囲は、2つのパラメータ、(**最小値**, **最大値**) により定義されます。最小値以上または最大値以下のグレイレベル値を持つピクセルはすべて、画像の粒子に属するピクセルとして選択されます。

二値化の例

この例では以下のソース画像を使用します。



しきい値範囲 (166, 255) (最も明るい領域) に属するピクセルをハイライトすると、次のような画像が生成されます。



境界がはっきり区別されていない場合、画像を粒子領域と背景領域にセグメント化する際に重大な問題が発生することがあります。そのような場合は、ユーザ自身が正しいしきい値を選択する必要があります。したがって、二値化前に画像の内容を強調して、正しい境界線がはっきり表示されるようにすることが必要な場合もあります。画像の内容を強調するには、ルックアップテーブル、フィルタ、FFT、または平均化関数を使用します。

境界部分と交差するラインの輝度プロファイルを確認することも、正しいしきい値を選ぶ上で役立ちます。さらに、モフォロジー変換を使用してバイナリ粒子の形状を修正すると、二値化の際の良くない選択を修正することができます。

自動二値化

自動二値化にはさまざまなテクニックがあります。

- クラスタ化
- エントロピー
- メトリック
- モーメント
- 相互分散

手動二値化とは反対に、これらの方法では最小と最大輝度値を設定する必要はありません。これらのテクニックは、輝度が変動する状況での使用に適しています。

元の画像によっては、モーメントやエントロピーなどの自動二値化関数を適用する前に、元のグレースケール画像を反転した方がいい場合があります。特に、背景が前景より明るい場合などです。

クラスタ化は、利用できる唯一のマルチクラス二値化メソッドです。クラスタ化は複数のクラスに対して操作するため、三次以上の画像を作成することができます。その他の 4 つのメソッド（エントロピー、メトリック、モーメント、相互分散）は、完全なバイナリ二値化テクニック用のものです。どのアルゴリズムを使用するかは、二値化する画像のタイプにより異なります。

クラスタ化

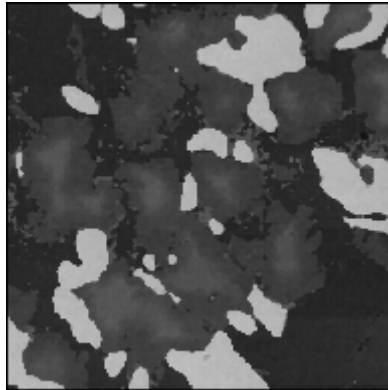
このテクニックでは、画像内に認識される位相の数に相当する別個の数のクラスの中にある画像のヒストグラムをソートします。グレー値が決定され、各クラスに対して**質量中心**が決まります。このプロセスは、各位相またはクラスの質量中心を表す値が得られるまで繰り返されます。

最も頻繁に使用される自動二値化メソッドはクラスタ化で、これはマルチクラス二値化とも呼ばれます。

クラスタ化の例

この例では、クラスタ化テクニックを画像の 2 位相および 3 位相で使用します。この関数の結果は、一般に画像のヒストグラム値や照明状態とは無関係です。

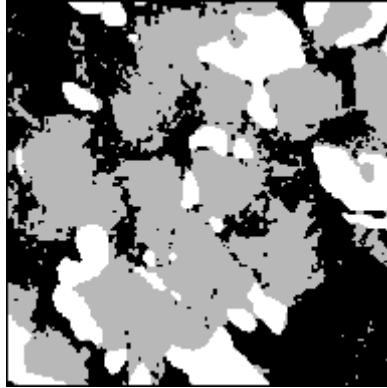
この例では以下の元の画像を使用します。



2 位相にクラスタ化すると、以下のような画像が生成されます。



3 位相にクラスタ化すると、以下のような画像が生成されます。



エントロピー

エントロピーは、従来の画像解析テクニックに基づくもので、画像上に極めて小さい割合で存在する粒子を検出するのに適しています。たとえば、欠陥の検出に最適です。

メトリック

このテクニックは、相互分散と同様の状況で使用します。各しきい値に対し、初期のグレースケールを表す面によって決まる値が計算されます。最適しきい値は、最小値に対応します。

モーメント

このテクニックは、コントラストの低い画像に適しています。モーメントメソッドは、表示されている画像が理論的にバイナリの画像の不鮮明なバージョンであるという仮定に基づいています。電子的ノイズや焦点不良などにより、集録プロセスでぼやけが生じた場合、ぼやけた画像と元の画像のいずれに対しても統計モーメント（平均および分散）は同じであると仮定されます。この関数は、理論上のバイナリ画像について再計算します。

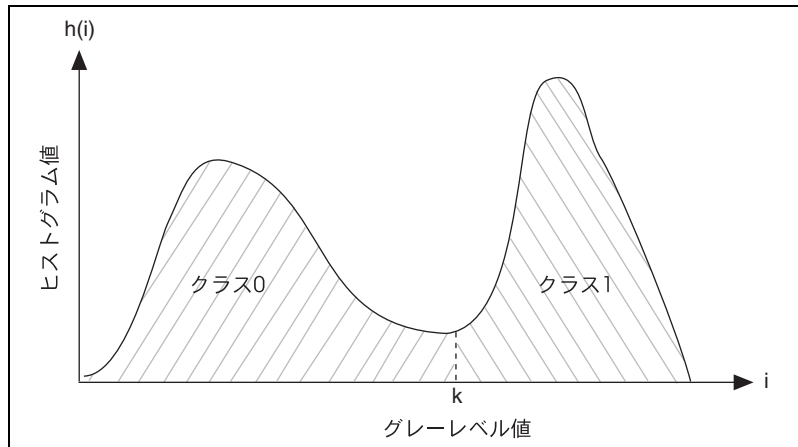
相互分散

相互分散とは、階乗解析を識別するのに使用する従来の統計テクニックです。このメソッドは、クラスがあまり不均衡でない画像に適しています。満足な結果を得るには、最小クラスが最大クラスの少なくとも 5% でなければなりません。2 つのクラスにかなりの差異がある場合、このメソッドでは、標準偏差の小さいクラスを過小評価する傾向があります。

詳細の説明

自動二値化テクニック

すべての自動二値化メソッドで、画像のヒストグラムを使用してしきい値を確定します。ヒストグラムのパラメータには、下記の記号を使用します。



i : グレーレベル値を表します。

k : しきい値として選択したグレーレベル値を表します。

$h(i)$: 画像内にある各グレーレベル値のピクセル数を表します。

N : 画像内のグレーレベルの総数を表します (8 ビット画像では 256)。

n : 画像内の総ピクセル数。

IMAQ Vision には、現在 5 つの自動二値化テクニックがあります。

- クラスタ化
- エントロピー
- 相互分散
- メトリック
- モーメント

これらの 5 つの方法はすべて、画像を 2 つのクラスに二値化するのに使用することができます。上図のように、 k 以下のグレーレベル値はすべてクラス 0 に、他のグレーレベル値はすべてクラス 1 に属するように、自動二値化テクニックを使用してしきい値のピクセル値 (k) を決定します。

クラスタ化メソッドは、画像を 3 つ以上のクラスに二値化する際に使用します。

クラスタ化

しきい値とは、以下の条件を満たすピクセル値 k です。

$$\frac{\mu_1 + \mu_2}{2} = k,$$

ここで、 μ_1 は画像内の 0 から k までの全ピクセル値の平均で、 μ_2 は $k + 1$ から 255 までの全ピクセル値の平均です。

エントロピー

このメソッドでは、ヒストグラムデータに情報理論を適用してしきい値を指定します。情報理論では、ヒストグラムのエントロピーはヒストグラムに関連する情報量を表します。次の式

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)}$$

がグレーレベル i の発生確率を表すとします。 $\{0, N-1\}$ の範囲のグレーレベル値を持つ画像のヒストグラムのエントロピーは、次の式で求められます。

$$H = - \sum_{i=0}^{i=N-1} p(i) \log_e p(i)$$

k がしきい値の場合、2つのエントロピー

$$H_b = - \sum_{i=0}^{i=k} p(i) \log_e p(i),$$

$$H_w = - \sum_{i=k+1}^{i=N-1} p(i) \log_e p(i)$$

は、二値化後の画像の黒および白のピクセルに関連するエントロピーを表します。最適なしきい値は、次の式で得られる二値化画像のエントロピーを最大にするグレーレベル値です。

$$H = H_b + H_w$$

わかりやすく言うと、しきい値とは、以下の式が最大になるピクセル値 k です。

$$- \sum_{i=0}^{i=k} \log(h(i)+1)(h(i)) - \sum_{i=k+1}^{i=N-1} \log(h(i)+1)(h(i)) + \log \left(\sum_{i=0}^{i=k} h(i) \sum_{i=k+1}^{i=N-1} h(i) \right)$$

相互分散

クラスタ化テクニックと同様に動作します。

メトリック

しきい値とは、以下の式が最小になるピクセル値 k です。

$$\sum_{i=0}^{i=k} h(i) |i - \mu_1| = \sum_{i=k+1}^{i=N-1} h(i) |i - \mu_2| ,$$

ここで、 μ_1 は画像内の 0 から k までの全ピクセル値の平均で、 μ_2 は $k+1$ から 255 までの全ピクセル値の平均です。

モーメント

このメソッドでは、二値化する画像のモーメントが出力（バイナリ）画像でも保持されるように、しきい値を計算します。

画像の k 番目のモーメント m は、次の式で計算します。

$$m_k = \frac{1}{n} \sum_{i=0}^{i=N-1} i^k h(i),$$

ここで n は画像内の総ピクセル数です。

カラー二値化

カラー二値化は、カラー画像をバイナリ画像に変換します。

カラー画像を二値化するには、各カラー成分に対して 1 つずつ、3 つのしきい値を指定する必要があります。出力画像のピクセルは、カラー成分が指定範囲内にある場合のみ設定されます。それ以外の場合、ピクセルは 0 に設定されます。

使用目的

解析や処理に必要な特徴を分離し、不要なものを取り除く場合に、カラー画像を二値化します。



メモ カラー二値化を実行する前に、ルックアップテーブルや平均化関数を使用して画像の特徴を強調することが必要な場合があります。

カラー画像を二値化するには、3 つのカラー成分 1 つ 1 つに対ししきい値を指定します。元の画像のピクセル値は、カラー成分が指定範囲内にある場合のみ 1 に設定されます。それ以外の場合、ピクセル値は 0 に設定されます。

図 8-1 は、RGB 形式で保存されたカラー画像の各プレーンのヒストグラムを示します。グレーで表した領域は、各カラープレーンのしきい値範囲を示します。カラー画像のピクセルをバイナリ画像の 1 に設定するには、赤の値を 130 から 200、緑の値を 100 から 150、青の値を 55 から 115 の間に設定する必要があります。

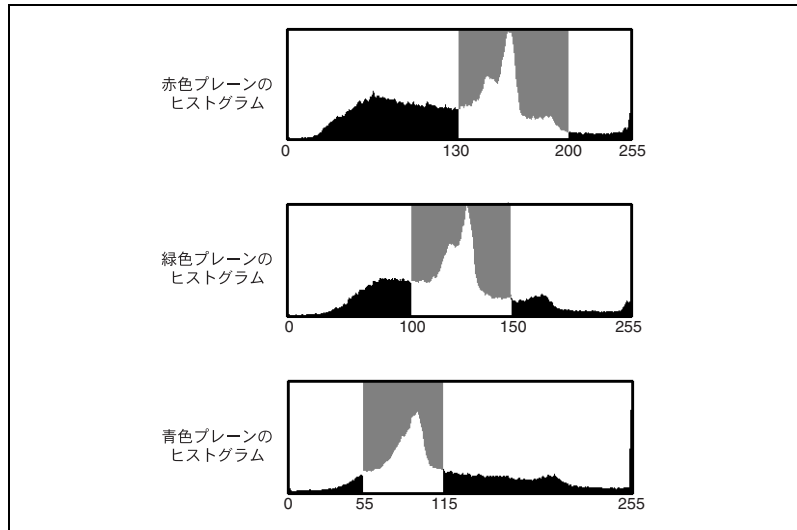


図 8-1 RGB 画像のしきい値範囲

RGB 画像を二値化するには、まず二値化後に解析するオブジェクトを形成するピクセルの赤、緑、および青の値を指定する必要があります。次に、各カラープレーンに対し、処理対象のカラー値を包含するしきい値範囲を指定します。処理対象のカラーを分離するためには、3つのカラープレーンすべてに対して正しい範囲を指定する必要があります。

図 8-2 は、HSL 形式で保存されたカラー画像の各プレーンのヒストグラムを示します。グレーで表した領域は、各カラープレーンのしきい値範囲を示します。カラー画像のピクセルをバイナリ画像の 1 に設定するには、色相値を 165 から 215、彩度値を 0 から 30、明度値を 25 から 210 の間に設定する必要があります。

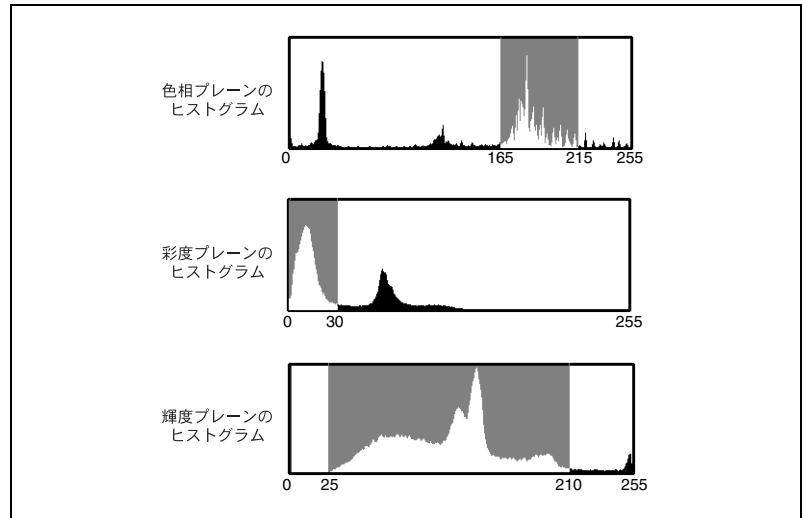


図 8-2 HSL 画像のしきい値範囲

色相プレーンには、画像の主要なカラー情報が含まれています。HSL 画像を二値化するには、まず二値化後に解析するピクセルの色相値を指定する必要があります。アプリケーションによっては、同じ色相値で彩度値のみが異なるカラーを選択することが必要な場合もあります。明度プレーンは画像の輝度レベルに関する情報のみを含むため、明度しきい値をすべての明度値が含まれるように設定すると、輝度情報が二値化プロセスに無関係になるようにすることができます。

バイナリモフォロジー

本章では、構造要素、連結性、基礎および上級バイナリモフォロジー操作について説明します。

概要

バイナリモフォロジー操作は、バイナリ画像内の粒子のストラクチャを抽出して変更します。面積、周囲、方向などの粒子測定を行う前に、検査アプリケーションでこれらの操作を実行して、バイナリ画像の情報内容を向上させることができます。

バイナリ画像とは、ピクセル値 1 の粒子領域とピクセル値 0 の背景領域からなる画像です。バイナリ画像は、**二値化**プロセスの結果として生成されます。二値化は主観的なプロセスであるため、出力されるバイナリ画像には、ノイズ粒子、画像の縁に接触する粒子、互いに接触する粒子、境界線が不均等な粒子など、不要な情報が含まれることがあります。モフォロジー関数は、粒子の形状を操作することによって、そのような不要情報を取り除き、バイナリ画像に含まれる情報の質を向上させることができます。

構造要素

粒子の形状を変更するモフォロジー演算は、近接ピクセルの数と値に基づいてピクセルを処理します。**近接ピクセル**とは、特定の画像処理関数が適用される際に、その値が近隣のピクセルの値に影響するピクセルを指します。モフォロジー変換では、**構造要素**という二次元バイナリマスクを使用して、近接ピクセルのサイズと各ピクセルに対する効果を定義し、粒子の形状と境界線に対するバイナリモフォロジー関数の影響度を制御することができます。

使用目的

すべての一次バイナリモフォロジー操作、または上級バイナリモフォロジー操作の 1 つ、分離操作を行う際に、構造要素を使用します。構造要素のサイズと値を変更して、粒子の形状をある特定の方法で変えることができます。ただし、構造要素を定義する前に、基礎モフォロジー操作について学ぶ必要があります。

構造要素の概念

処理対象のピクセルの新しい値を決める際、構造要素のサイズと内容によって、モフォロジー操作で考慮されるピクセルが指定されます。構造要素は、処理対象のピクセルを中心として、軸が奇数サイズである必要があります。構造要素の内容は常にバイナリで、1と0の値からなります。最も一般的な構造要素は、値1を含む3×3の行列です。この行列（下図）は、ほとんどのバイナリおよびグレースケールモフォロジー変換でのデフォルトの構造要素になっているものです。

```

1 1 1
1 1 1
1 1 1
    
```

モフォロジー変換で構造要素が処理するピクセルを定義する方法を決める要素には、構造要素のサイズ、構造要素のセクターの値、およびピクセルフレームの形状の3つがあります。

構造要素のサイズ

構造要素のサイズは、処理対象のピクセルを取り囲む近接ピクセルのサイズを決定します。処理対象のピクセルの座標は、構造要素の相関として決まります。図9-1で、処理対象のピクセルの座標はそれぞれ(1,1)、(2,2)、および(3,3)です。基点(0,0)は常に左上角のピクセルです。

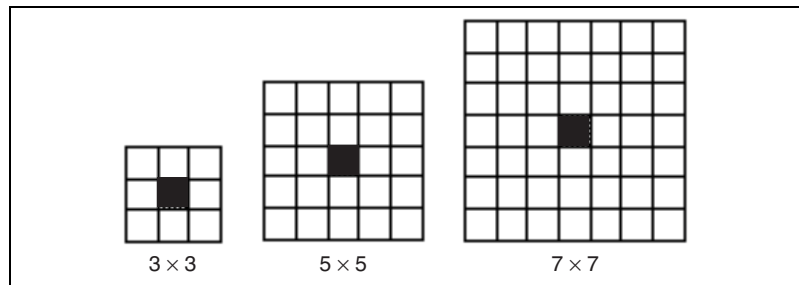


図 9-1 構造要素のサイズ

構造要素を使用するには、画像ボーダーが必要です。3×3の構造要素では、最小で1のボーダーサイズが必要です。同様に、5×5と7×7の構造要素では、最小でそれぞれ2と3のボーダーサイズが必要です。構造要素が大きくなると、画像ボーダーのサイズもそれに伴って大きくなります。ROIの詳細については、第1章、「デジタル画像」の「画像ボーダー」のセクションを参照してください。



メモ IMAQ Vision の画像のデフォルトボーダーサイズは3です。このボーダーサイズにより、一切の変更なく、最大7×7の構造要素を使用することができます。7×7よりも大きい構造要素を使用する場合は、画像を作成する際にその構造要素に対応するサイズのボーダーを指定する必要があります。

構造要素のサイズによって、モフォロジー変換の速度が決まります。構造要素が小さいほど、変換速度は速くなります。

構造要素の値

構造要素のバイナリ値によって、変換の際にどの近接ピクセルを考慮に入れるかが次のように決まります。

- 構造要素のセクターの値が1の場合、変換の際に、対応する元の画像のピクセルの値が中央のピクセルの値に影響を与えます。
- 構造要素の値が0の場合、モフォロジー関数はそれに対応する元の画像のピクセル値を無視します。

図9-2は、モフォロジー関数を実行する際の構造要素の値による影響を示します。構造要素を使用してモフォロジー変換を実行すると、ピクセル P_0 が近接ピクセル値の相関になるように変更されます。

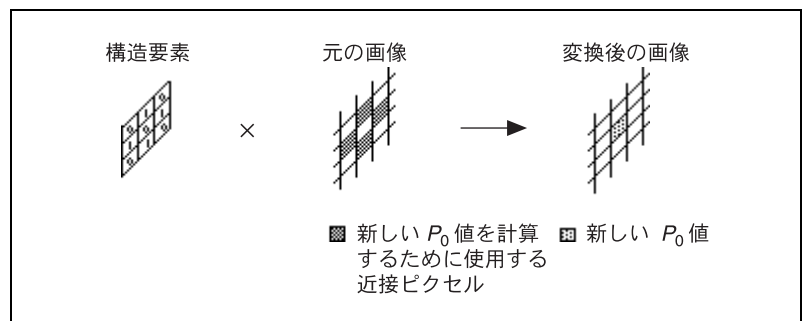


図 9-2 構造要素の値のモフォロジー関数に対する作用

ピクセルフレームの形状

デジタル画像とは、四角形のグリッドに配置されたピクセルの二次元配列です。粒子の構造を抽出し変更するモフォロジー変換により、ピクセルを正方形または六角形のいずれかの構成で処理することができます。そのようなピクセル構成が、ピクセルフレームという概念の元になります。ピクセルフレームは整列（正方形）とシフト（六角形）のいずれかです。ピクセルフレームパラメータは、近接ピクセルの輝度値によってピクセルの値を変更する関数では重要なパラメータです。このフレーム概念を利用した関数で画像を処理する際の、IMAQ Vision の画像解析方法は、正方形と六角形のどちらのフレームを使用するかによって異なります。IMAQ Vision では、デフォルトで正方形フレームを使用します。



メモ

画像内のピクセルは、水平ピクセルフレームでは物理的にシフトしません。ピクセルフレームの形状を設定できる関数では、六角形のフレームを指定した場合、単にピクセル値を異なる方法で処理するのみです。

図 9-3 は、 3×3 と 5×5 の構造要素を適用した場合の、正方形と六角形のピクセルフレームの違いを表したものです。

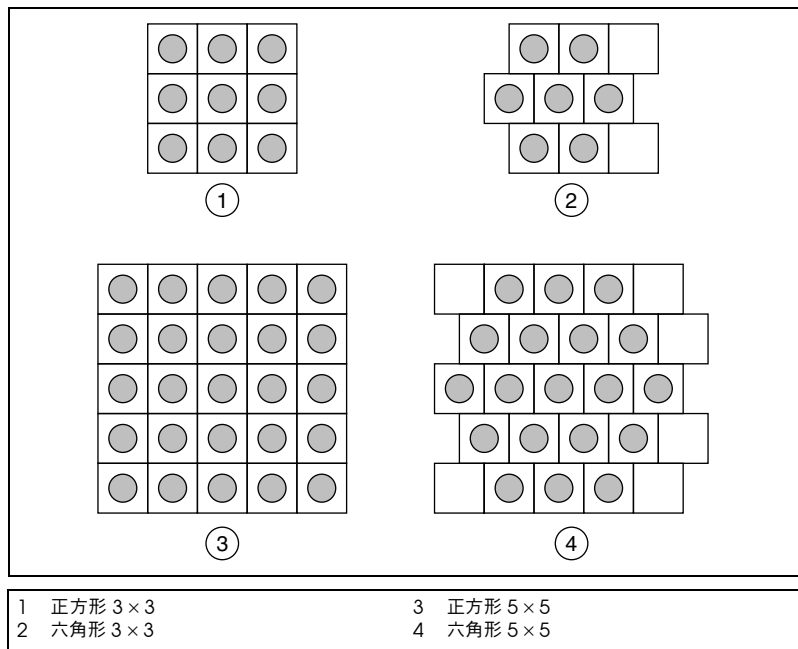


図 9-3 正方形と六角形のピクセルフレーム

モフォロジー関数で 3×3 の構造要素と六角形のフレームモードを使用する場合、処理対象のピクセルに対する近接ピクセルの影響を計算する際に、要素 $(2, 0)$ と $(2, 2)$ は変換に考慮されません。モフォロジー関数で 5×5 の構造要素と六角形のフレームモードを使用する場合、要素 $(0, 0)$ 、 $(4, 0)$ 、 $(4, 1)$ 、 $(4, 3)$ 、 $(0, 4)$ および $(4, 4)$ は変換に考慮されません。

図 9-4 は、 3×3 の構造要素と四角形のフレームモードを使用したモフォロジー変換を示します。

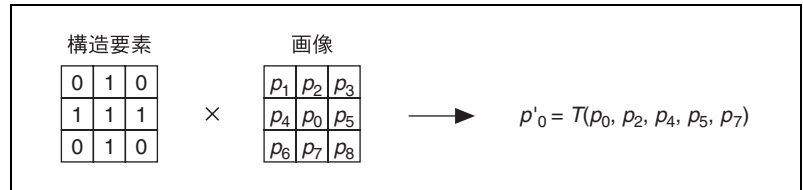


図 9-4 3×3 の構造要素と四角形フレームを使用した変換

図 9-5 は、 3×3 の構造要素と六角形のフレームモードを使用したモフォロジー変換を示します。

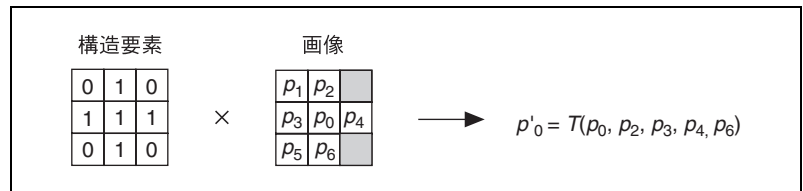


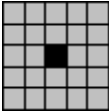
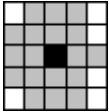
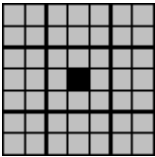
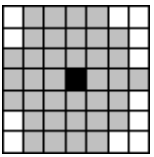


図 9-5 3×3 の構造要素と六角形フレームを使用した変換

表 9-1 は、ピクセルフレームの形状による近接ピクセルへの影響を、構造要素のサイズごとに示したものです。グレーのボックスは、黒で示した中央のピクセルの近接ピクセルを表します。

表 9-1 ピクセルフレームの形状による近接ピクセルの変化

構造要素のサイズ	正方形ピクセルフレーム	六角形ピクセルフレーム
3 × 3		
5 × 5		
7 × 7		

正方形フレーム

正方形フレームの場合、ピクセルは通常どおりに並べられています。図 9-6 は、8つの近接ピクセルに周りを囲まれた、正方形フレームのピクセルを示します。d は、水平および垂直方向の近接ピクセルから中央ピクセルへの距離で、斜め方向の近接ピクセルは中央ピクセルから $\sqrt{2}d$ の距離にあります。

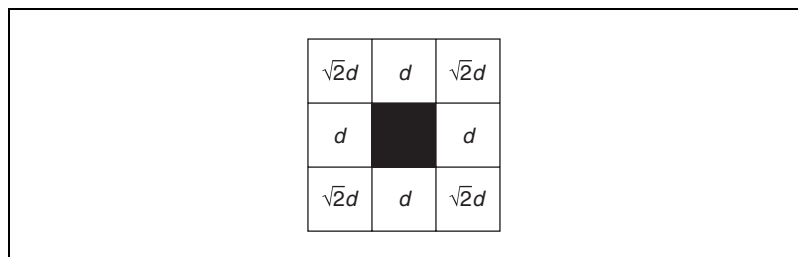


図 9-6 正方形フレーム

六角形フレーム

六角形フレームでは、画像の偶数のラインが右に半ピクセル分シフトします。したがって、六角形フレームの場合ピクセルは真円に似た配置になります。図 9-7 は、6つの近接ピクセルに周りを囲まれた、六角形フレームのピクセルを示します。各近接ピクセルは中央のピクセルから等距離 d にあるため、モフォロジー測定において非常に正確な結果が得られます。

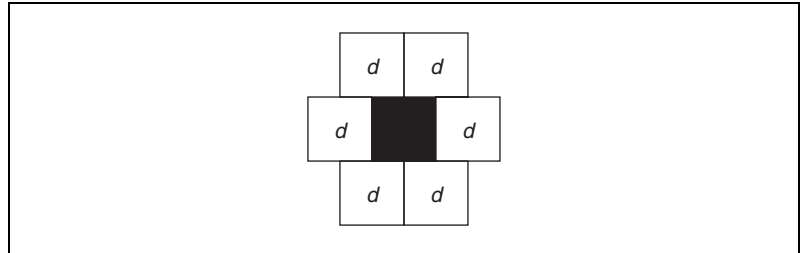


図 9-7 六角形フレーム

連結性

指定した輝度しきい値に該当するピクセルを識別したら、IMAQ Vision はそれらのピクセルを粒子として一まとめにします。このプロセスにより、隣接のピクセル、つまり**連結性**が生じます。関数によっては、ピクセルの連結性を設定して、それにより2つの隣接するピクセルが同じ粒子に含まれるかどうかを決める方法を指定することもできます。

使用目的

バイナリ画像内に1つのポイントで接触する粒子が含まれる場合は、IMAQ Vision が各粒子を認識できるように、連結性4を使用します。狭い領域を含む粒子がある場合は、連結性8を使用します。連結性設定を選択したら、アプリケーション全体を通して同じ連結性設定を使用する必要があります。

連結性の概念

連結性4では、2つのピクセルが水平または垂直方向に隣接している場合、それらは同じ粒子の一部とみなされます。連結性8では、2つのピクセルが水平、垂直、または斜め方向に隣接している場合、それらは同じ粒子の一部とみなされます。図 9-8 は、2つのタイプの連結性を示します。

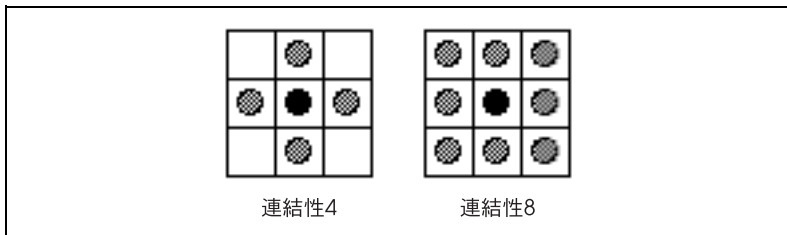


図 9-8 連結性のタイプ

図 9-9 は、連結性 4 と連結性 8 の場合で画像内の粒子数がどのように異なるかを示します。図 9-9a は、連結性 4 の場合で画像には 2 つの粒子があります。図 9-9b は、連結性 8 の場合で画像には 1 つしか粒子がありません。

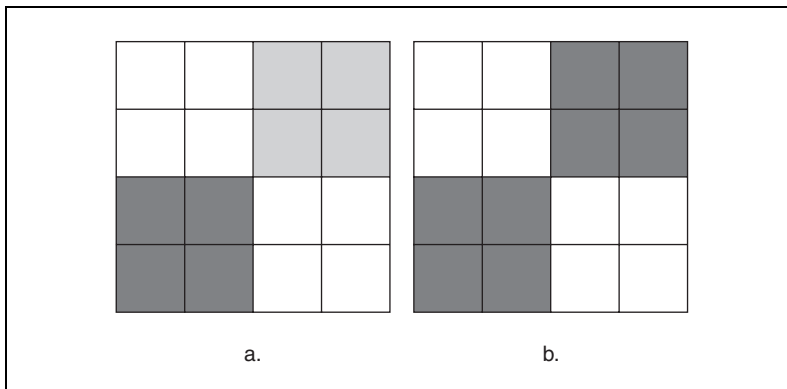


図 9-9 連結性処理の例

IMAQ RemoveParticle、IMAQ RejectBorder、IMAQ FillHole、および IMAQ ParticleFilter の各上級モフォロジー VI では、入力**連結性 4/8**（デフォルトは 8）を使用して、近接ピクセルが同じ粒子に属するかどうかを指定します。

詳細の説明

四角形のピクセルフレームでは、次の図に示すように、各ピクセル P_0 には 8 つの近接ピクセルがあります。数学的に考えた場合、ピクセル P_1 、 P_3 、 P_5 、 P_7 は、ピクセル P_2 、 P_4 、 P_6 、および P_8 よりもピクセル P_0 に近い距離にあります。

$$\begin{bmatrix} P_8 & P_1 & P_2 \\ P_7 & P_0 & P_3 \\ P_6 & P_5 & P_4 \end{bmatrix}$$

D が P_0 から P_1 への距離とした場合、 P_0 とその 8 つの近接ピクセルとの距離は、 D から $\sqrt{2}D$ となり、次の図のようになります。

$$\begin{bmatrix} \sqrt{2}D & D & \sqrt{2}D \\ D & 0 & D \\ \sqrt{2}D & D & \sqrt{2}D \end{bmatrix}$$

連結性 4

ピクセルが粒子内の他のピクセルから D の距離に位置している場合、そのピクセルは粒子の一部ということになります。つまり、2 つのピクセルが水平または垂直方向に隣接している場合、それらは同じ粒子の一部とみなされます。それらのピクセルが斜めに隣接している場合は、2 つの別々の粒子であると考えます。図 9-10 では、粒子の数は 4 になります。

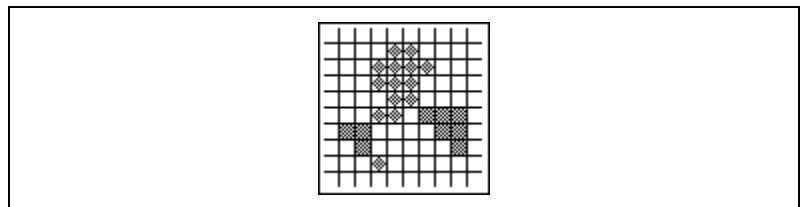


図 9-10 連結性 4

連結性 8

ピクセルが粒子内の他のピクセルから D または $\sqrt{2}D$ の距離に位置している場合、そのピクセルは粒子の一部となります。つまり、2 つのピクセルが水平、垂直、または斜め方向に隣接している場合、それらは同じ粒子の一部とみなされます。図 9-11 では、粒子の数は 1 になります。

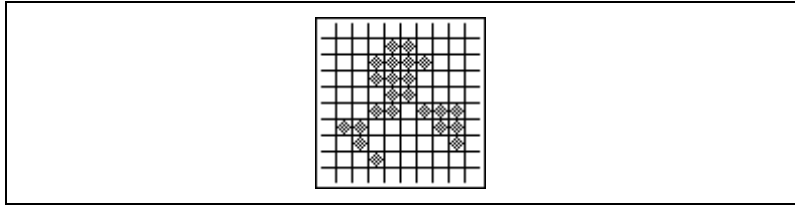


図 9-11 連結性 8

一次モフォロジー操作

一次モフォロジー操作は、バイナリ画像に対して動作し、近接ピクセルに基づいて各ピクセルを処理します。各ピクセルは、近接ピクセルの情報と使用する操作によって1または0に設定されます。これらの操作を行うと、常に画像内の粒子の全体的なサイズと形状が変更されます。

使用目的

一次モフォロジー操作は、粒子の拡張や縮小、オブジェクトのボーダーの平滑化、粒子の外側および内側境界線の検出、および特定の配置のピクセルの検索などに使用します。

また、これらの変換操作により、粒子の量的解析用に画像を調整したり、領域のジオメトリを確認したり、最も単純な形状を抽出してモデル処理や識別に利用したりすることもできます。

一次モフォロジーの概念

一次モフォロジー関数は、粒子が1に、背景が0に設定されたバイナリ画像に適用されます。一次モフォロジー関数には、収縮、膨張、およびヒットミス の3つの基礎バイナリ処理関数があります。その他の変換操作は、これらの3つの関数を組み合わせたものです。

このセクションでは、次の一次モフォロジー変換について説明します。

- 収縮
- 膨張
- オープン
- クローズ
- 内側傾斜
- 外側傾斜
- ヒットミス
- 細線化

- 太線化
- 適正オープン
- 適性クローズ
- 自動メディアン



メモ 以下の説明で、ピクセルという語は値 1 を持つピクセルを示し、粒子という語は値 1 を持つピクセルの集まりを示します。

収縮および膨張関数

収縮は、背景に孤立しているピクセルを取り除き、構造要素で定義されたテンプレートにしたがって粒子の輪郭を収縮します。

特定のピクセル P_0 に対し、構造要素は P_0 を中心とします。そして、1 に等しい構造要素の係数によってマスク処理されたピクセルは、 P_i となります。

- ピクセル P_i の値が 0 の場合、 P_0 は 0 に設定され、それ以外の場合 P_0 は 1 に設定されます。
- $\text{AND}(P_i) = 1$ の場合、 $P_0 = 1$ になり、それ以外の場合は $P_0 = 0$ になります。

膨張は、粒子内に孤立している小さな穴を取り除き、構造要素で定義されたテンプレートにしたがって粒子の輪郭を拡張します。膨張とは背景を収縮することと同じであるため、この関数は収縮と逆の効果があります。

特定のピクセル P_0 に対し、構造要素は P_0 を中心とします。そして、1 に等しい構造要素の係数によってマスク処理されたピクセルは、 P_i となります。

- ピクセル P_i の値が 1 の場合、 P_0 は 1 に設定され、それ以外の場合 P_0 は 0 に設定されます。
- $\text{OR}(P_i) = 1$ の場合、 $P_0 = 1$ になり、それ以外の場合は $P_0 = 0$ になります。

図 9-12 は、収縮と膨張による効果を示します。図 9-12a は、元のバイナリ画像です。図 9-12b は、収縮処理後の画像で、図 9-12c は膨張処理後の画像です。

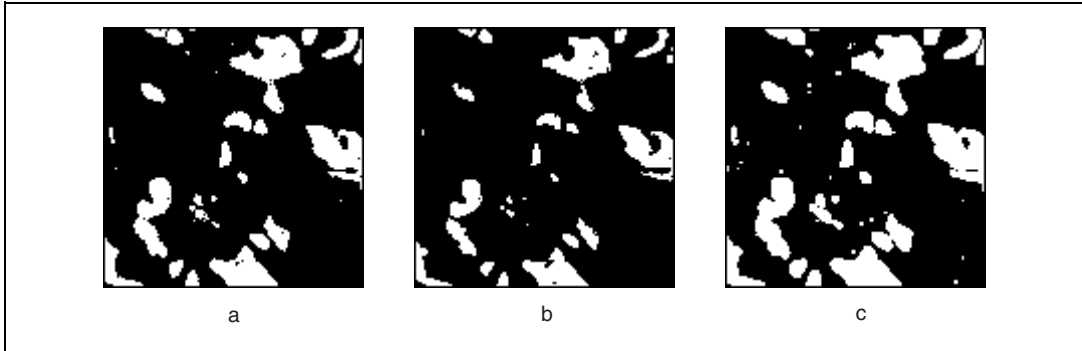


図 9-12 収縮および膨張関数

図 9-13 は、表 9-2 と表 9-3 に示す例のソース画像で、グレーのセルが 1 に等しいピクセルを表します。

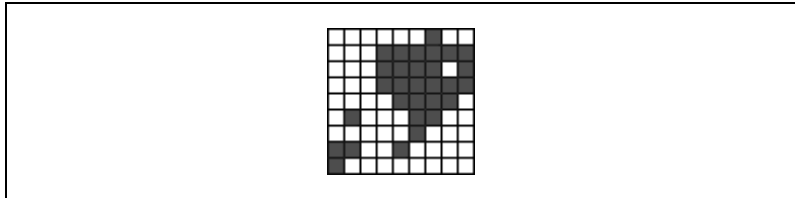





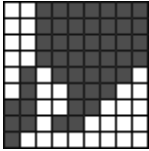
図 9-13 収縮および膨張前のソース画像

表 9-2 および表 9-3 は、構造要素によって収縮や膨張にどのような影響があるかをそれぞれ示したものです。構造要素が大きいほど、テンプレートの編集が可能になり、効果はより選択的になります。

表 9-2 構造要素による収縮への影響

構造要素	収縮後	説明
		ピクセルが 1 に等しく左上 3 つの近接ピクセルが 1 でない場合、そのピクセルは消去されます。収縮により、粒子の左上ボーダーは切り取られます。
		ピクセルが 1 に等しくその下および右の近接ピクセルが 1 でない場合、そのピクセルは消去されます。収縮により、粒子の下および右のボーダーは切り取られますが、角の部分は維持されます。

表 9-3 構造要素による膨張への影響

構造要素	膨張後	説明
		ピクセルが 1 に等しいか、その左上の 3 つの近接ピクセルが 1 の場合、そのピクセルは 1 に設定されます。膨張により、粒子の右下のボーダーが拡張されます。
		ピクセルが 1 に等しいか、その下または右の近接ピクセルが 1 の場合、そのピクセルは 1 に設定されます。膨張により、粒子の上および左のボーダーが拡張されます。

オープンおよびクローズ関数

オープン関数とは、収縮の後に膨張を行うものです。この関数により、小さな粒子が取り除かれ境界線が滑らかになります。収縮と膨張は**二重変換処理**であり、ボーダーは収縮で取り除かれた後膨張で復元されるため、この操作を行っても粒子の面積や形状が大きく変わることはありません。ただし、収縮によって取り除かれた小さな粒子は、膨張によっても再現されません。*I*を画像とすると、以下のようになります。

$$\text{オープン}(I) = \text{膨張}(\text{収縮}(I))$$

クローズ関数とは、膨張の後に収縮を行うものです。この関数は、微小な穴を埋めて境界線を滑らかにします。膨張と収縮は**補完的モフォロジー操作**であり、膨張関数によって拡張されたボーダーは収縮関数で縮小されるため、この操作を行っても粒子の面積と形状が大きく変わることはありません。ただし、膨張によって埋められた微小の穴は収縮によっても再現されません。*I*を画像とすると、以下のようになります。

$$\text{クローズ}(I) = \text{収縮}(\text{膨張}(I))$$

次の図は、オープン関数とクローズ関数の例を示します。

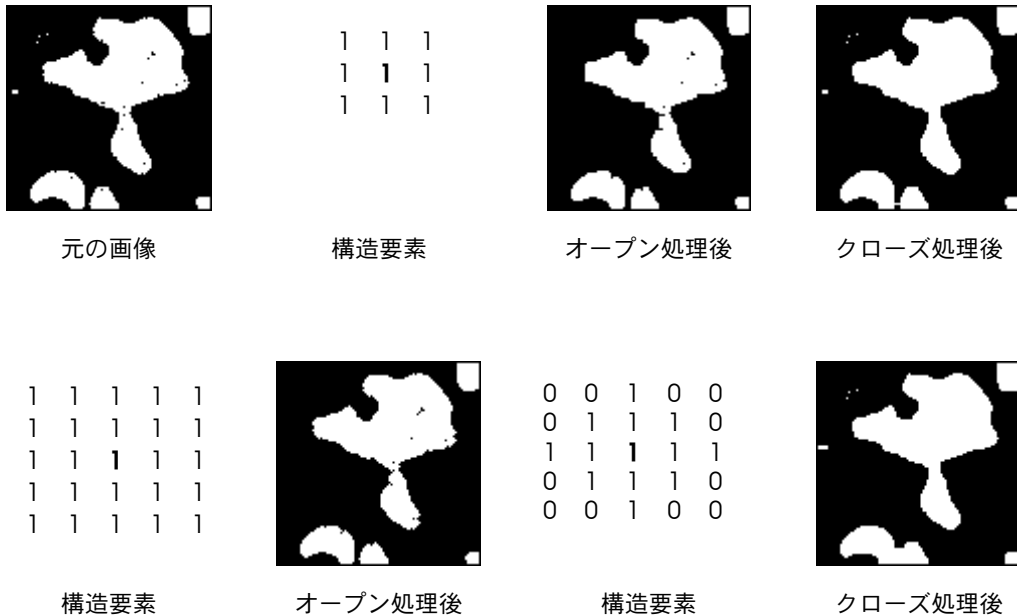


図 9-14 オープンおよびクローズ関数

内側傾斜関数

内側傾斜関数は、元の画像から収縮後の画像を減算します。残ったピクセルは、収縮プロセスで削除されたピクセルに対応します。 I を画像とすると、以下のようになります。

$$\text{内側傾斜}(I) = I - \text{収縮}(I) = \text{XOR}(I, \text{収縮}(I))$$

外側傾斜関数

外側傾斜関数は、膨張後の画像から元の画像を減算します。残ったピクセルは、膨張プロセスで追加されたピクセルに対応します。 I を画像とすると、以下のようになります。

$$\text{外側傾斜}(I) = \text{膨張}(I) - I = \text{XOR}(I, \text{膨張}(I))$$

図 9-15a は、元のバイナリ画像です。図 9-15b は、5×5の構造要素を使った抽出により生成された画像を示します。重ねて表示した内部エッジは白で示し、外図エッジはグレーで示しています。拡張された輪郭の太さは、構造要素のサイズにより異なります。

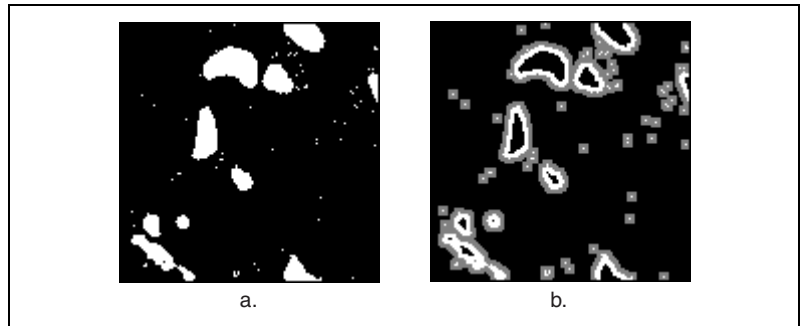


図 9-15 外部エッジ

ヒットミス関数

ヒットミス関数を使用して、特定のピクセルの配置を検出します。この関数は、構造要素で定義されたテンプレートに完全一致する近接のピクセルを抽出します。構造要素の構成によって、ヒットミス関数は、単独の孤立したピクセル、十字または縦長のパターン、粒子のエッジ上の直角、その他のユーザ指定の形状などを検出することができます。構造要素のサイズが大きいほど、学習したテンプレートはより明確になります。ヒットミス関数の使用方法については、表 9-4 を参照してください。

中央係数が 0 の構造要素の場合、ヒットミス関数を使用すると、元の画像で 1 に設定されているすべてのピクセルが値 0 に変更されます。

特定のピクセル P_0 に対し、構造要素は P_0 を中心とします。そして、構造要素によってマスク処理されたピクセルは、 P_1 となります。

- ピクセル P_1 の値がその上に配置された構造要素の係数に等しい場合、ピクセル P_0 は 1 に設定され、それ以外の場合ピクセル P_0 は 0 に設定されます。
- つまり、ピクセル P_1 が構造要素とまったく同一のテンプレートを定義する場合、 $P_0 = 1$ となり、それ以外の場合は $P_0 = 0$ になります。

図 9-16b、図 9-16c、および図 9-16d は、図 9-16a に示す同じソース画像に 3 つのヒットミス関数を適用したものです。各ヒットミス関数で異なる構造要素を使用しており、それは変換後の画像の上に示しています。グレーのセルは 1 に等しいピクセルを示します。

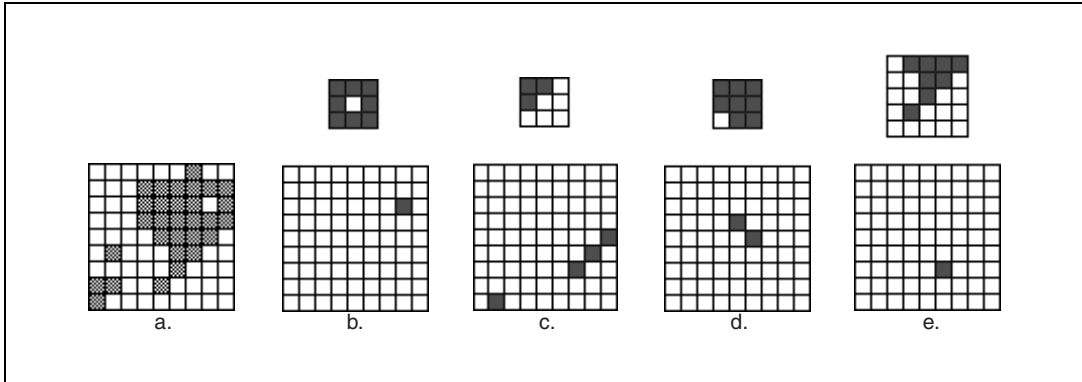


図 9-16 ヒットミス関数

ヒットミス関数の2つめの例は、図 9-17 に示すようなバイナリ画像において、構造要素で指定したさまざまなパターンを検出する状況を示します。その結果は、表 9-4 に示しています。

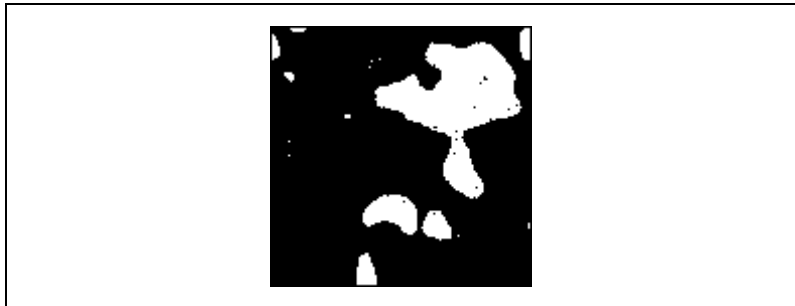
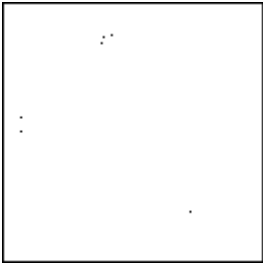
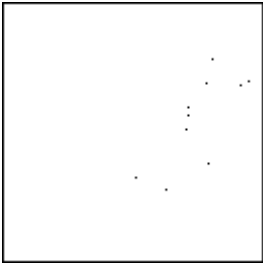
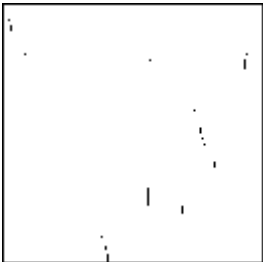


図 9-17 ヒットミス関数適用前のバイナリ画像

表 9-4 ヒットミス関数を使用する

方法	構造要素	生成される画像
<p>ヒットミス関数を使用して、背景に孤立しているピクセルを検出します。</p> <p>右に示す構造要素は、0 に等しいピクセルの少なくとも2つの層によって囲まれた1の値を持つピクセルをすべて抽出します。</p>	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 </pre>	
<p>ヒットミス関数を使用して、粒子内に単独で存在するピクセルの穴を検出します。</p> <p>右に示す構造要素は、1 に等しいピクセルの少なくとも1つの層によって囲まれた0の値を持つピクセルをすべて抽出します。</p>	<pre> 1 1 1 1 0 1 1 1 1 </pre>	
<p>ヒットミス関数を使用して、左の垂直エッジ上のピクセルを検出します。</p> <p>右に示す構造要素は、左方向に1に等しいピクセルと右方向に0に等しいピクセルの少なくとも1つの層で囲まれたピクセルを抽出します。</p>	<pre> 1 1 0 1 1 0 1 1 0 </pre>	

細線化関数

細線化関数は、構造要素で指定したテンプレートに一致する近接ピクセルを削除します。構造要素の構成によっては、細線化を使用して背景に孤立した単一ピクセルと粒子のエッジ上に存在する直角を削除することもできます。構造要素のサイズが大きいほど、テンプレートはより明確になります。

細線化関数は、元の画像とヒットミス関数適用後の変換画像の間での共通部分を抽出します。バイナリ用語で言い換えると、この操作により元の画像からヒットミス変換が減算されるということになります。

構造要素の中央係数が 0 の場合は、この関数を使用しないでください。その場合は、ヒットミス関数を使用しても背景のある特定のピクセルの値を 0 から 1 に変更するのみです。ただし、細線化関数の減算によって結果的にそれらのピクセルは 0 にリセットされます。

I を画像とすると、以下のようになります。

$$\text{細線化}(I) = I - \text{ヒットミス}(I) = \text{XOR}(I, \text{ヒットミス}(I))$$

図 9-18a は、次の細線化の例で使用する元のバイナリ画像を示します。図 9-18b は、元の画像から背景の単一ピクセルが削除された出力画像を示しています。この例では以下の構造要素を使用します。

```

0 0 0
0 1 0
0 0 0
    
```

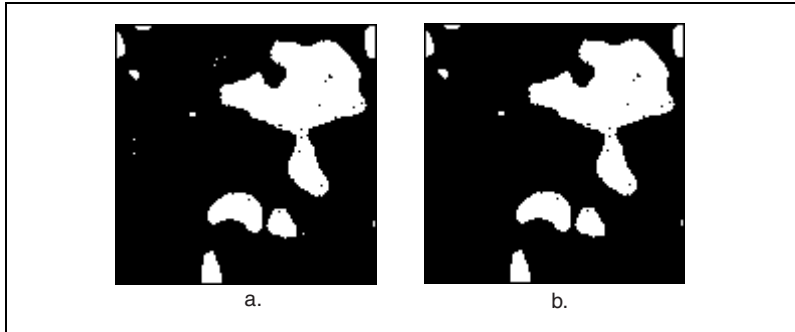


図 9-18 細線化関数

細線化のもう1つの例では、図9-19aに示すソース画像を使用します。図9-19bから図9-19dは、ソース画像に細線化を3回適用した結果を示します。各細線化で異なる構造要素を使用し、それは変換後の画像の上に示しています。グレーのセルは1に等しいピクセルを示します。

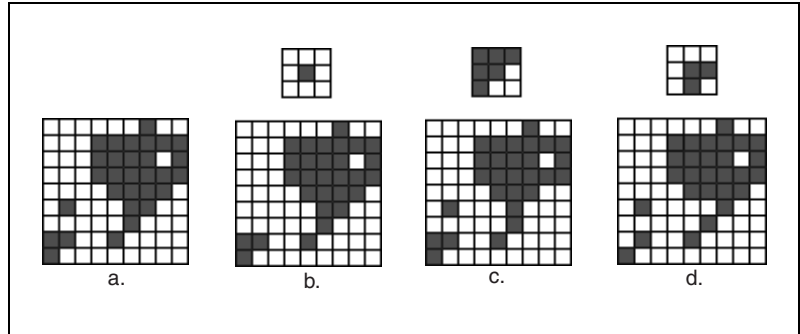


図 9-19 各構造要素による細線化関数

太線化関数

太線化関数は、構造要素で指定したテンプレートに一致する近接ピクセルを画像に追加します。構造要素の配置によっては、太線化を使用して、穴を埋め粒子のエッジ上にある直角を滑らかにすることもできます。構造要素が大きいほど、より明確なテンプレートが可能になります。

太線化関数により、元の画像と変換後の画像の結合が抽出されます。これは、太線化用に指定した構造要素を使ったヒットミス関数によって作成されたものです。バイナリ用語で言い換えると、この操作により元の画像にヒットミス変換が加算されるということになります。

構造要素の中央係数が1の場合はこの関数を使用しないでください。その場合は、ヒットミス関数を使用しても粒子のある特定のピクセルを1から0に変更するのみです。ただし、太線化関数の加算によって結果的にそれらのピクセルは1にリセットされます。

1を画像とすると、以下のようになります。

$$\text{太線化}(I) = I + \text{ヒットミス}(I) = \text{OR}(I, \text{ヒットミス}(I))$$

図 9-20a は、次の太線化の例で使用する元のバイナリ画像を示します。
 図 9-20b は、元の画像に太線化関数を適用した結果を示します。この関数では、次の構造要素を使用して単一ピクセルの穴が埋められています。

```

    1 1 1
    1 0 1
    1 1 1
    
```

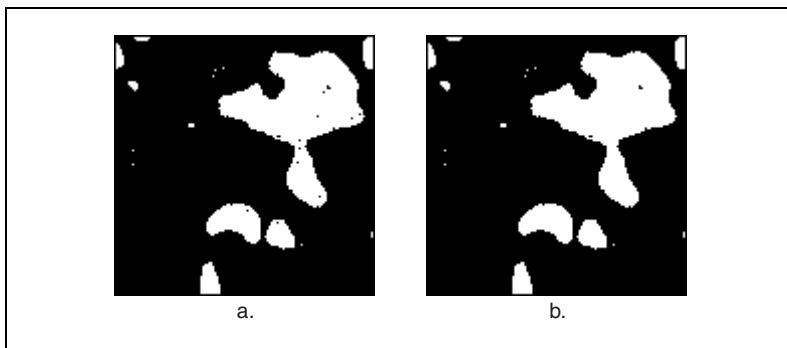


図 9-20 太線化関数

図 9-21a は、もう 1 つの太線化の例に使用するソース画像です。
 図 9-21b から図 9-21d は、ソース画像に太線化を 3 回適用した結果を示します。各太線化で異なる構造要素を使用しており、それは変換後の画像の上に示しています。グレーのセルは 1 に等しいピクセルを示します。

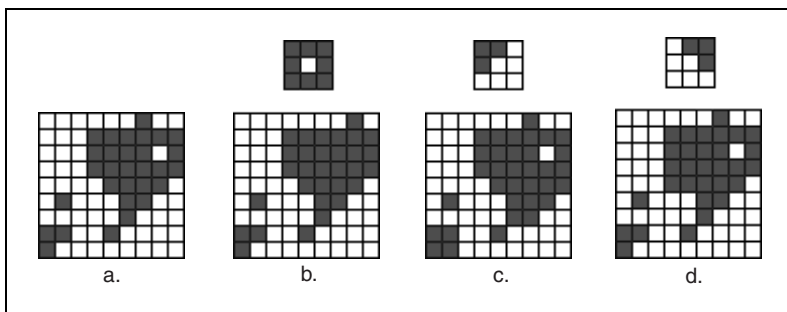


図 9-21 異なる構造要素を使用した太線化関数

適正オープン関数

適正オープン関数は、オープンおよびクローズの有限的な組み合わせです。この関数は、小さな粒子を取り除き、構造要素で定義されたテンプレートにしたがって、粒子の輪郭を滑らかにします。

I が元の画像の場合、適正オープン関数は元の画像 I と、オープン、クローズ、さらにもう1回オープンを行うことによって得られた変換後の画像との間の共通部分を抽出します。

$$\begin{aligned} \text{適正オープン}(I) &= \text{AND}(I, \text{OCO}(I)) \text{ または} \\ \text{適正オープン}(I) &= \text{AND}(I, \text{DEEDDE}(I)) \end{aligned}$$

ここで I は元の画像、
 E は収縮、
 D は膨張、
 O はオープン、
 C はクローズ、
 $F(I)$ は画像 I に関数 F を適用して得られた画像、そして
 $GF(I)$ は画像 I にまず関数 F を適用し、次に画像 I に関数 G を適用して得られた画像です。

適正クローズ関数

適正クローズ関数は、クローズおよびオープンの有限的な組み合わせです。この関数は、小さな穴を取り除き、構造要素で定義されたテンプレートにしたがって、粒子の輪郭を滑らかにします。

I が元の画像の場合、適正クローズ関数は元の画像 I と、クローズ、オープン、さらにもう1回クローズを行うことによって得られた変換後の画像との間の共通部分を抽出します。

$$\begin{aligned} \text{適正クローズ}(I) &= \text{OR}(I, \text{COC}(I)) \text{ または} \\ \text{適正クローズ}(I) &= \text{OR}(I, \text{EDDEED}(I)) \end{aligned}$$

ここで I は元の画像、
 E は収縮、
 D は膨張、
 O はオープン、
 C はクローズ、
 $F(I)$ は画像 I に関数 F を適用して得られた画像、そして
 $GF(I)$ は画像 I にまず関数 F を適用し、次に画像 I に関数 G を適用して得られた画像です。

自動メディアン関数

自動メディアン関数は、オープンとクローズの組み合わせを使用します。この関数は、詳細の少ないよりシンプルな粒子を生成します。

I が元の画像の場合、自動メディアン関数は元の画像 I に適正オープンおよび適正クローズを適用したものの共通部分を抽出します。

$$\begin{aligned} \text{自動メディアン}(I) &= \text{AND}(\text{OCO}(I), \text{COC}(I)) \text{ または} \\ \text{自動メディアン}(I) &= \text{AND}(\text{DEEDDE}(I), \text{EDDEED}(I)) \end{aligned}$$

ここで I は元の画像、

E は収縮、

D は膨張、

O はオープン、

C はクローズ、

$F(I)$ は画像 I に関数 F を適用して得られた画像、そして

$GF(I)$ は画像 I にまず関数 F を適用し、次に画像 I に関数 G を適用して得られた画像です。

上級モフォロジー操作

上級モフォロジー操作は、一次モフォロジー演算に基づいており、画像のピクセルではなく粒子に対して処理を行います。各操作は、バイナリ画像の粒子に対して特定の演算を行うよう開発されています。

使用目的

上級モフォロジー操作は、粒子内の穴埋め、画像の縁にある粒子や大小の不要な粒子の削除、互いに接触している粒子の分離、粒子の凸閉包の検出などに使用します。

また、これらの変換操作により、粒子の量的解析用に画像を調整したり、領域のジオメトリを確認したり、最も単純な形状を抽出してモデル処理や識別に利用したりすることもできます。

上級モフォロジー変換の概念

上級モフォロジー関数は、バイナリ収縮および膨張など、基礎変換を条件的に組み合わせたものです。この関数は、粒子にしきい値 1 が適用され背景が 0 のバイナリ画像に適用されます。このセクションでは、次の上級バイナリモフォロジー関数について説明します。

- ボーダー
- 穴埋め
- ラベリング
- ローパスフィルタ
- ハイパスフィルタ
- 分離
- スケルトン
- セグメント化
- 距離
- ダニエルソン
- 円形
- 凸



メモ 本書のこのセクションでは、ピクセルという語は 1 に等しいピクセルを示し、粒子という語は 1 に等しいピクセルの集まりを示します。

ボーダー関数

ボーダー関数は、画像の縁にある粒子を取り除きます。これらの粒子は画像をデジタル化した際に切り取られた可能性があり、それらを取り除くことによって、誤った粒子測定や統計が得られるのを回避することができます。

穴埋め関数

穴埋め関数は、粒子内の穴を埋めます。

ラベリング関数

ラベリング関数は、各粒子に別々のグレーレベル値を割り当てます。生成される画像は、バイナリ画像ではなく、画像内の粒子の数に背景領域のグレーレベル 0 を足した数に等しいグレーレベル値を使用した、ラベル付き画像です。

ラベリング関数は、連結性 4 または連結性 8 の基準を使って粒子を識別します。連結性についての詳細は、本章の「**連結性**」のセクションを参照してください。

ローパスフィルタとハイパスフィルタ

ローパスフィルタは、**フィルタサイズ**というパラメータで指定した幅にしたがって、小さな粒子を取り除きます。フィルタサイズ N に対し、ローパスフィルタは幅が $(N - 1)$ ピクセル以下の粒子を取り除きます。これらの粒子は収縮を $(N - 1)/2$ 回繰り返すと消滅します。

ハイパスフィルタは、フィルタサイズというパラメータで指定した幅にしたがって、大きな粒子を取り除きます。フィルタサイズ N に対し、ハイパスフィルタは幅が N 以上の粒子を取り除きます。これらの粒子は収縮を $(N/2 + 1)$ 回繰り返しても消滅しません。

ハイパスおよびローパスモフォロジーフィルタは、いずれも収縮処理を使用して粒子を削除します。収縮処理やフィルタ処理では、 $2k$ ピクセルの幅のピクセルと $2k - 1$ ピクセルの幅の粒子を区別できないため、1回の収縮処理で2ピクセル幅の粒子と1ピクセル幅の粒子の両方が取り除かれます。

表 9-5 は、ローパスおよびハイパスフィルタフィルタ処理の効果を示します。

表 9-5 ローパスおよびハイパスフィルタ処理の効果

フィルタサイズ (N)	ハイパスフィルタ	ローパスフィルタ
N は、偶数 ($N = 2k$) です。	<ul style="list-style-type: none"> • $2k$ 以上の幅の粒子を取り除きます。 • $k - 1$ 回の収縮処理を行います。 	<ul style="list-style-type: none"> • $2k - 2$ 以下の幅の粒子を取り除きます。 • $k - 1$ 回の収縮処理を行います。
N は、奇数 ($N = 2k + 1$) です。	<ul style="list-style-type: none"> • $2k + 1$ 以上の幅の粒子を取り除きます。 • k 回の収縮処理を行います。 	<ul style="list-style-type: none"> • $2k$ 以下の幅の粒子を取り除きます。 • k 回の収縮処理を行います。

図 9-22 は、この例で使用した元のバイナリ画像を示します。図 9-22b は、ある特定のフィルタサイズに対し、ハイパスフィルタによって次の画像が生成される方法を示します。グレーの粒子と白の粒子は、それぞれローパスフィルタとハイパスフィルタによってフィルタ処理されます。

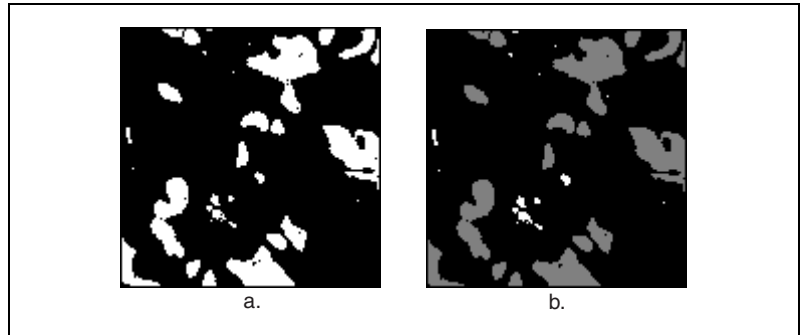


図 9-22 ハイパスフィルタ関数とローパスフィルタ関数

分離関数

分離関数は、ユーザ指定のフィルタサイズに基づいて、細い部分を切断し、互いに接触している粒子を切り離します。この操作では、収縮、ラベリング、および条件付き膨張を使用します。

たとえば、画像を二値化した後、互いに重なり合っている 2 つのグレーレベル粒子は 1 つのバイナリ粒子のように見えることがあります。元の粒子が互いに接触する部分が峽部として表示されます。その峽部の幅が M ピクセルの場合、 $(M + 1)$ のフィルタサイズを使用した分離により、その峽部が切断され元の 2 つの粒子が復元されます。これは N ピクセルより狭い部分を持つすべての粒子に適用されます。

フィルタサイズ N に対し、分離関数は幅が $(N - 1)$ ピクセル以下の峽部を分離します。これらの粒子は収縮を $(N - 1)/2$ 回繰り返すと 2 つの部分に分離されます。

N が奇数の場合は上記の定義どおりになりますが、 N が偶数の場合は、峽部を切り離すかつながったままにするかを定める上での収縮処理の使用により、多少の修正が必要になります。この関数は、 $2k$ ピクセルの幅の峽部と $(2k - 1)$ ピクセルの幅の峽部を区別することができないため、1 回の収縮により、2 ピクセル幅の峽部と 1 ピクセル幅の峽部の両方が切り離されます。

分離の精度は、以下のような場合に、偶数ピクセルより幅の狭い峽部を取り除くときのみ限定されます。

- N が偶数 ($2k$) の場合、 $(2k - 2)$ ピクセル以下の幅の峽部が分離されます。ここでは $(k - 1)$ 回の収縮が行われます。
- N が奇数 ($2k + 1$) の場合、 $2k$ ピクセル以下の幅の峽部が分離されません。ここでは k 回の収縮が行われます。

スケルトン関数

スケルトン関数は、それぞれの粒子の幅が 1 ピクセルになるまで細線化を繰り返し適用します。スケルトン関数は、時間がかかりメモリも多く使用します。これは、さまざまな構成の構造要素での細線化とオープン処理を条件的に適用するものです。

スケルトン L では、次のような構造要素を使用します。

```

0  ?  1
0  1  1
0  ?  1
    
```

スケルトン M では、次のような構造要素を使用します。

```

?  ?  1
0  1  1
?  ?  1
    
```

SKIZ は、逆スケルトンです（反転画像に対するスケルトン L）。

L スケルトン関数

L スケルトン関数は、L 形の構造要素のスケルトン関数のことです。図 9-23a の元の画像を確認し、この画像から L スケルトン関数によって図 9-23b の四角形のピクセルフレーム画像が生成されることを確認してください。

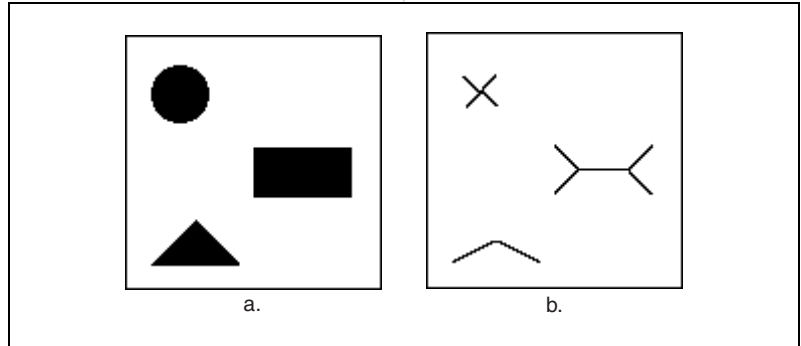


図 9-23 L スケルトン関数

M スケルトン関数

M スケルトン (M 形の構造要素) 関数を使用すると、より多くの樹状突起や枝部分を持つスケルトンが抽出されます。図 9-23a の同じソース画像に M スケルトン関数を適用すると、図 9-24 のような画像が生成されます。

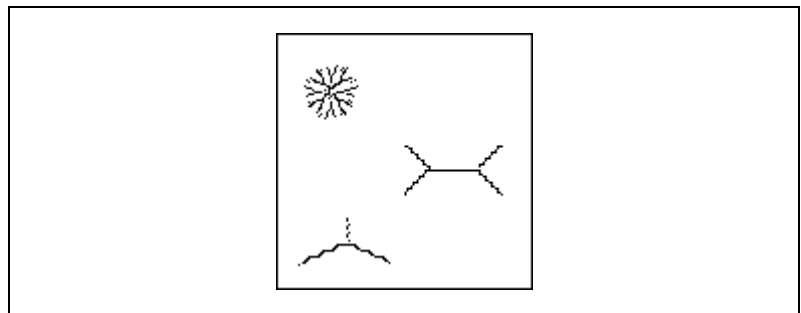


図 9-24 M スケルトン関数

SKIZ 関数

SKIZ (skeleton of influence zones) 関数は、L スケルトンを、粒子領域でなく背景領域に適用したような動作をします。これにより粒子から等距離の中央線が生成されます。

図 9-23a のソース画像にスキッツ関数を適用して、それをソース画像に重ねると、次のような画像になります。

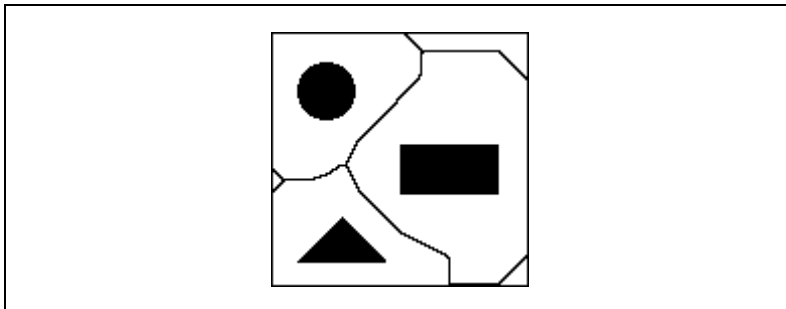


図 9-25 スキッツ関数

セグメント化関数

セグメント化関数は、ラベル付きの画像にのみ適用されます。この関数は、粒子を中心としたセグメントに画像を分割し、セグメント同士が重ならず、また空きスペースが残らないようにします。粒子が互いに接触するまで膨張処理を行うと、空きスペースが発生します。



メモ セグメント化関数は処理に時間がかかります。この関数を選択する前に、画像を最小有意サイズに縮小してください。

図 9-26 では、バイナリ粒子（黒で表示）はセグメント（グレーの階調で表示）の上に重ねて表示されています。

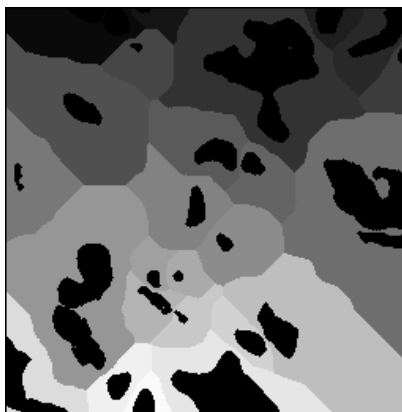


図 9-26 セグメント化関数

バイナリ粒子を含む画像に適用すると、変換後の画像はすべてのピクセルが1に設定されているため、画像全体が赤になります。

セグメント化関数と SKIZ 関数の比較

セグメント化関数は、それぞれ1つの粒子を含むセグメントを抽出し、他の粒子を妨害することなく、(すべての粒子が同じ速度で移動すると仮定して) その粒子を移動できる範囲を表します。

これらのセグメントのエッジが粒子の外部スケルトンを表します。SKIZ 関数と異なり、セグメント化では中央距離は関係ありません。

粒子同士が接触して画像全体を占めるまで連続的に膨張を適用することによって、セグメントを得ることができます。最終的な画像には、元の画像の粒子と同じ数のセグメントが含まれます。ただし、SKIZ ラインに囲まれた部分をセグメントとして考えた場合、元の画像にあった粒子の数より多くのセグメントが生成されることもあります。図 9-27 の右上の領域を見てください。この画像には、以下のものが含まれています。

- 黒で表した元の粒子
- 点のパターンで表したセグメント
- スキッズライン

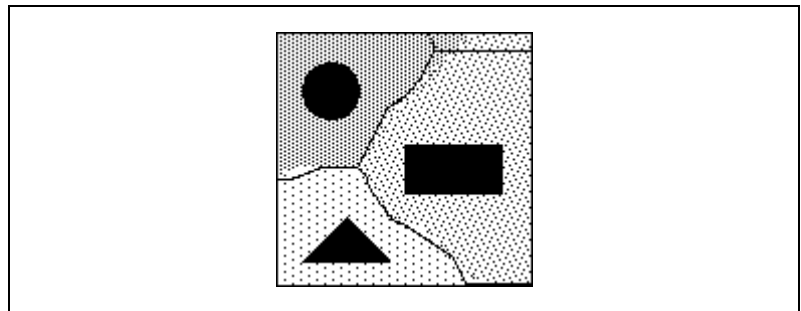


図 9-27 スキッズラインによるセグメント化

距離関数

距離関数は、粒子のボーダーへの最短距離に等しい各ピクセルに、グレースケール値を割り当てます。その距離は、粒子の外側のボーダーへの距離、または粒子内の穴への距離です。

ダニエルソン関数

ダニエルソン関数も距離マップを作成しますが、従来の距離関数よりも精度の高いアルゴリズムです。できる限り、距離関数でなくダニエルソン関数を使用するようにしてください。

図 9-28a は、次の例で使用する元の二値化画像を示します。この画像は、ローパスフィルタ、穴埋め、およびダニエルソン関数で順次処理されています。ダニエルソン関数により、図 9-28b に示すような距離マップ画像が生成されます。

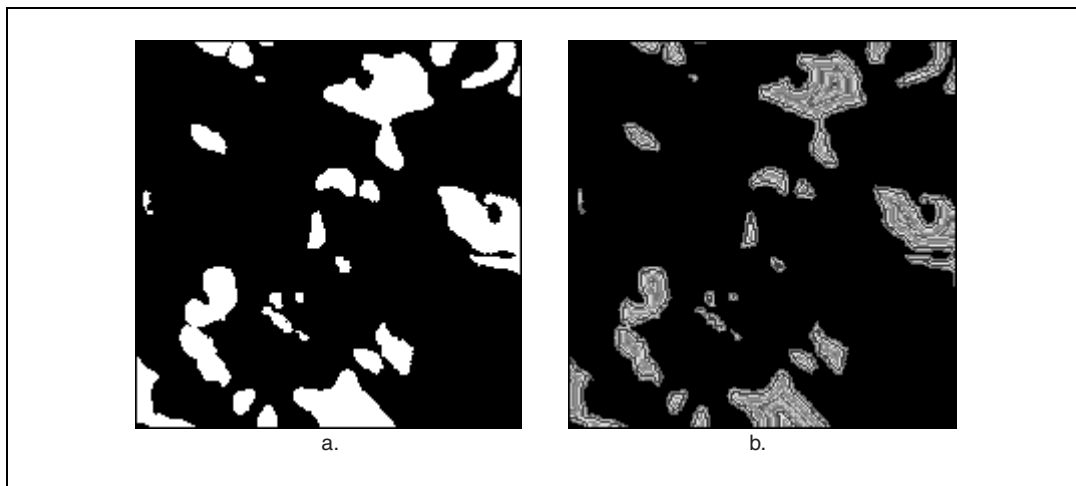


図 9-28 ダニエルソン関数

この最終画像をバイナリパレットで表示します。その場合、各レベルに応じて色分けされます。ピクセルのセットの粒子のボーダーとの関係を容易に識別することができます。最初の層は、ボーダーを形成する層で、色は赤です。ボーダーに最も近い2つめの層は緑で、3つめの層は青、という具合になります。

円形関数

円形関数を使用すると、互いに重なり合う円形粒子を分離することができます。円形関数では、粒子が本質的に円形であるという条件で、ダニエルソン係数を使用して粒子の形状を再構築します。粒子は、当初円形が重なり合っていて、それを別々の円形に切り離すものであると考えます。これにより、各粒子に対応する円を描くことができます。

図 9-29a は、次の円形関数の例で使用するソース画像です。図 9-29b は処理後の画像を示します。

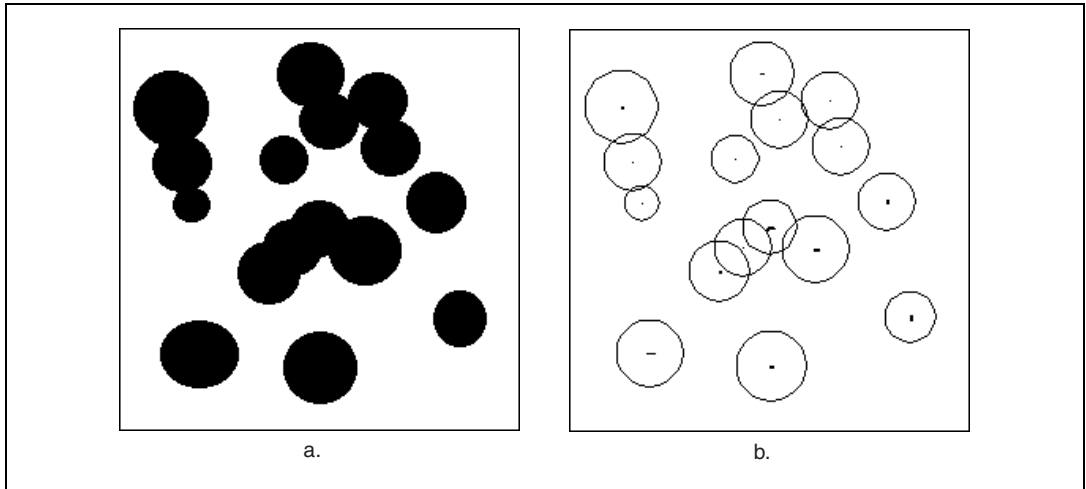


図 9-29 円形関数

凸関数

凸関数は、粒子の輪郭が不連続の場合でも粒子の測定を行えるようにするため、粒子をクローズ処理するのに便利です。このコマンドは通常、集録プロセスでサンプル粒子が切れてしまった場合に必要になります。

凸関数は、各粒子の周囲にある凸閉包を計算して、実質的に粒子をクローズ処理します。処理対象の画像は、バイナリでラベル付きでなければなりません。

図 9-30a は、この例で使用するラベル付きのバイナリソース画像です。
図 9-30b は、その画像に凸関数を適用した後の画像です。

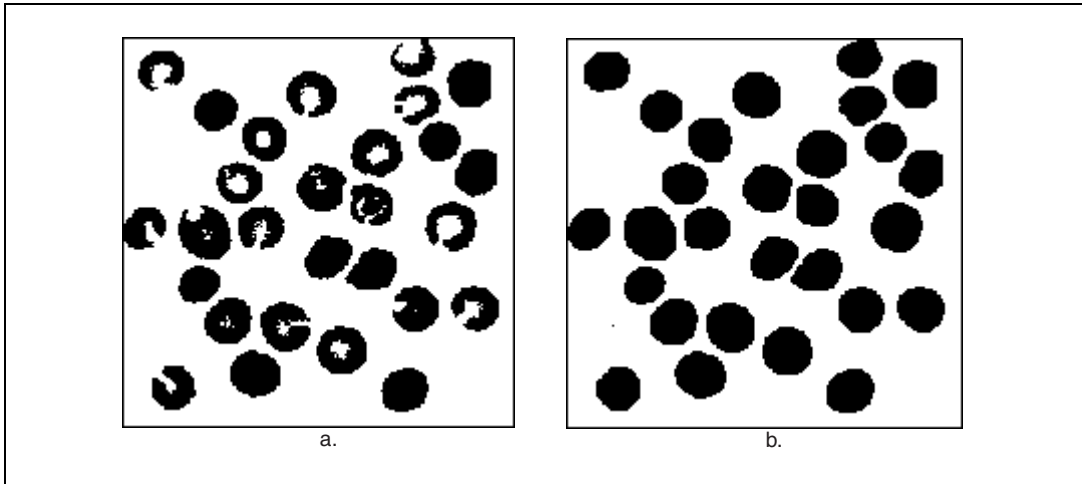


図 9-30 凸関数

粒子測定

本章では、粒子の面積、長さ、座標、弦と軸、形状の等価および特徴、およびさまざまな測定について説明します。

デジタル粒子

「面積」、「長さ」、「座標」、「弦と軸」、「形状の等価」、「形状の特徴」、および「多様な測定値」の各セクションで説明したモフォロジーおよび輝度パラメータによって、デジタル粒子の特性を表すことができます。

使用目的

バイナリ画像で粒子の形状測定を行う場合に、粒子測定を使用します。

デジタル粒子の概念

面積

このセクションでは、次の面積パラメータについて説明します。

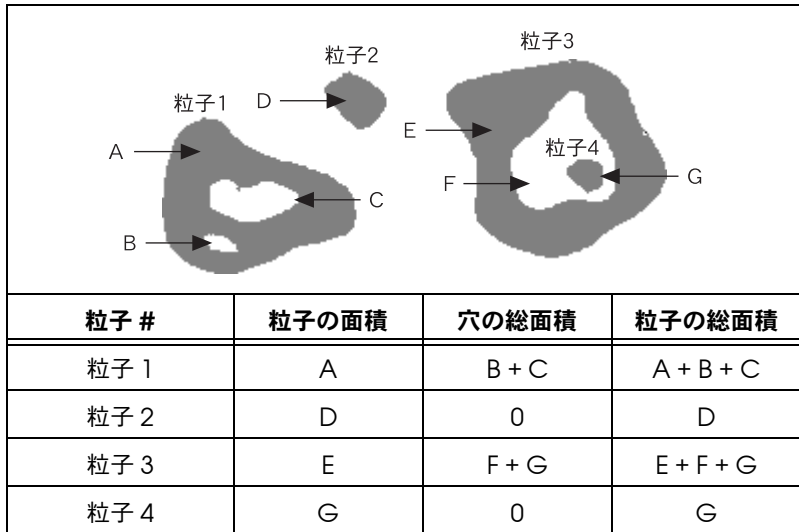
- **ピクセル数**：ピクセル単位で表した、穴を除いた粒子の面積
- **粒子面積**：実単位（画像空間キャリブレーションに基づく）で表した粒子の面積。1ピクセルが1平方単位を表すようにキャリブレーションされている場合、この値は**ピクセル数**と同じになります。
- **画像の面積**：実単位で表した、画像全体の面積。この値は、（解像度 X × X ステップ）（解像度 Y × Y ステップ）の積になります。
- **比率**：画像全体の面積に対する粒子面積の比率。すべての粒子が画像に占めるパーセンテージです。

$$\text{比率} = \frac{\text{粒子の面積}}{\text{画像の面積}}$$

- **穴の数**：粒子内の穴の数。ソフトウェアは、最小1ピクセルの粒子内にある穴を検出します。
- **穴の総面積**：粒子内の穴の総面積。
- **粒子の総面積**：穴の面積を含めた粒子の面積。この値は、**粒子面積 + 穴の総面積**と等しくなります。



メモ 大きな粒子の穴の中にある粒子は、別個の粒子として識別されます。中に粒子を含む穴の面積には、その粒子の面積も含まれます。



長さ

このセクションでは、次の長さパラメータについて説明します。

- **粒子の周囲**：粒子の外側の輪郭の長さ。
- **穴の周囲**：粒子内の穴の周囲の合計。要素 B が要素 A に塞がれているような場合にその 2 つの要素について検討するとき、穴の測定値は非常に便利なデータになります。二値化後に B 領域が A 領域内の穴のようになるよう画像を処理できる場合は、**(穴の総面積 / 粒子の総面積)** の比率により B の A における割合が求められます。**穴の周囲**は、A と B の境界線の長さを表します。
- **幅**：粒子の左端のピクセルと右端のピクセル間の距離で、 $\max(X_i) - \min(X_i)$ で求められます。また幅は、粒子を含む最小の境界四角形の横の辺、つまり $\max X - \min X$ の差に等しくなります。
- **高さ**：粒子の最も上と最も下のピクセル間の距離で、 $\max(Y_i) - \min(Y_i)$ で求められます。また高さは、粒子を含む最小の境界四角形の縦の辺、つまり $\max Y - \min Y$ の差に等しくなります。

座標

座標は、画像の左上角の原点 (0, 0) を中心として表現します。このセクションでは、次の座標パラメータについて説明します。

- **重心 (X, Y)** : 粒子の重心の座標。N ピクセルで構成された粒子、 P_i の重心は、ポイント G として次の式で定義されます。

$$\overline{OG} = \frac{1}{N} \sum_{i=1}^{i=N} \overline{OP}_i$$

$$\text{重心 } X_G = \frac{1}{N} \sum_{i=1}^{i=N} X_i$$

X_G は、粒子の水平セグメントの中心点の平均位置です。

$$\text{重心 } Y_G = \frac{1}{N} \sum_{i=1}^{i=N} Y_i$$

Y_G は、粒子の垂直セグメントの中心点の平均位置です。



メモ 粒子が凸面のない形状の場合、G は粒子の外に位置していることもあります。

- **左辺 X, 上辺 Y** : 粒子を含む最小境界四角形の左上と右下の角の座標。原点 (0, 0) には、(minX, minY) と (maxX, maxY) の座標を持つ 2 つのピクセルがあり、次のようになります。

$$\text{minX} = \min(X_i)$$

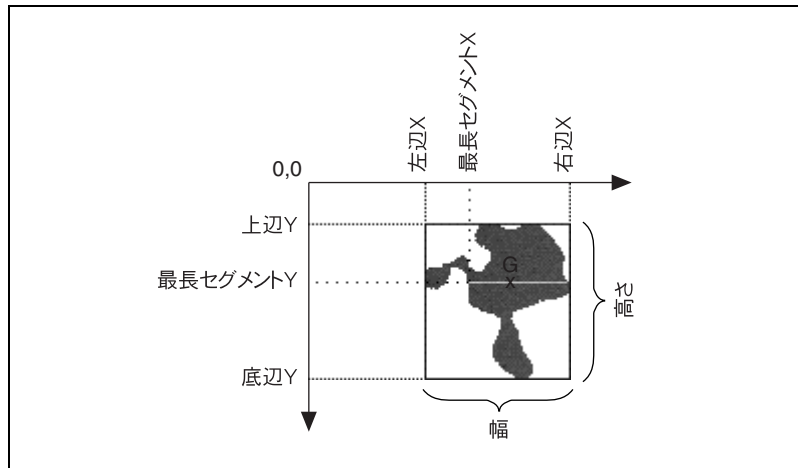
$$\text{minY} = \min(Y_i)$$

$$\text{maxX} = \max(X_i)$$

$$\text{maxY} = \max(Y_i)$$

ここで X_i と Y_i は、粒子内のピクセル P_i の座標です。

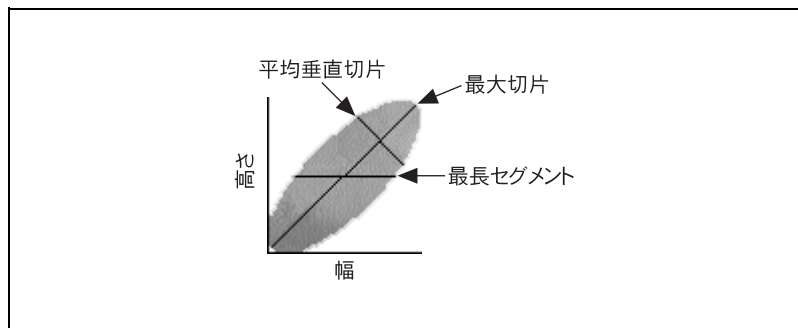
- **右辺 X, 底辺 Y** : 粒子を含む最小境界四角形の右下角。
- **最長セグメント X, Y** : 粒子の最長セグメント上で最も左にあるピクセルの座標。



弦と軸

このセクションでは、次の弦と軸パラメータについて説明します。

- **最長セグメント**：粒子の最大セグメントの長さ。
- **弦平均 X**：粒子の水平セグメントの長さの平均。
- **弦平均 Y**：粒子の垂直セグメントの長さの平均。

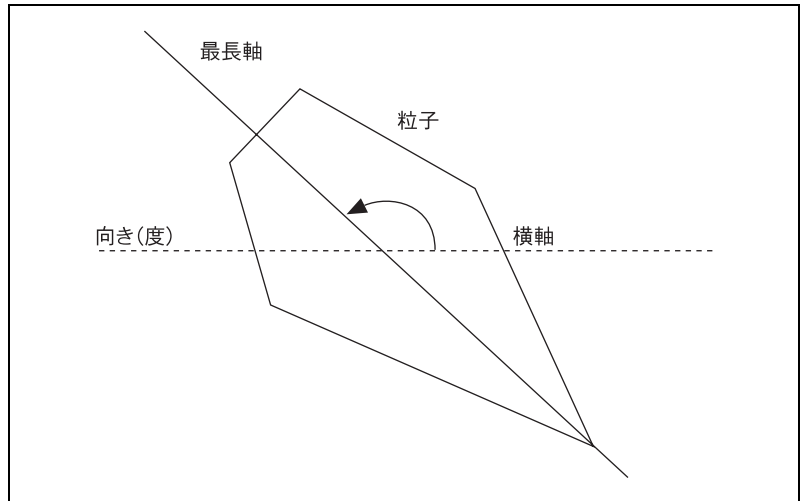


- **最大切片**：粒子の凸包にある（可能なすべての投影方向での）最長セグメントの長さ。
- **平均垂直切片**：最大切片に対して垂直な粒子のセグメントの平均長。

$$\text{平均垂直切片} = \frac{\text{粒子の凸包の面積}}{\text{最大切片}}$$

- **粒子の方向**：最長軸の横軸に対する角度。この値は 0° から 180° の範囲になります。この値から粒子の対称性に関する情報は得られません。

るので注意してください。したがって、 190° という角度は 10° と同じとみなされます。



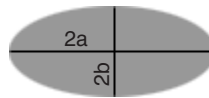
形状の等価

このセクションでは、次の形状の等価パラメータについて説明します。

- **等価楕円短軸**：粒子と同じ面積と粒子の最大切片の半分に等しい長軸を持つ楕円の短軸。

この定義により、次の式が得られます。

$$\text{粒子面積} = \pi ab、\text{最大切片} = 2a$$



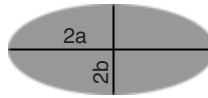
等価楕円短軸は次の式で定義されます。

$$2b = \frac{4 \times \text{粒子の面積}}{\pi \times \text{最大切片}}$$

- **楕円長軸**：粒子と同じ面積および周囲長を持つ楕円の長軸の全長。この長さは、 $2a$ に等しくなります。

この定義により、次の式が得られます。

$$\begin{aligned} \text{面積} &= \pi ab \\ \text{周囲} &= \pi \sqrt{2(a^2 + b^2)} \end{aligned}$$



和 $a + b$ と積 ab がパラメータ **粒子の面積** および **粒子の周囲** の関数になるように、この方程式を表すことができます。そして a と b は、多項式 $X^2 - (a + b)X + ab = 0$ の 2 つの解となります。

特定の面積と周囲長に対し、解は 1 つ (a, b) であることに注意してください。

- **楕円短軸**：粒子と同じ面積および周囲長を持つ楕円の短軸の全長。この長さは、 $2b$ に等しくなります。
- **楕円率**：等価楕円の長軸の短軸に対する比率
これは次のように定義されます。

$$\frac{\text{楕円長軸}}{\text{楕円短軸}} = \frac{a}{b}$$

等価楕円が長いほど、楕円比率も高くなります。等価楕円が真円に近いほど、楕円比率は 1 に近くなります。

- **長方形の長辺**：粒子と同じ面積と周囲長を持つ四角形の長辺 (a) の長さ。

この定義により、次の式が得られます。

$$\begin{aligned} \text{面積} &= ab \\ \text{周囲} &= 2(a + b) \end{aligned}$$



和 $a + b$ と積 ab がパラメータ **粒子の面積** および **粒子の周囲** の関数になるように、この方程式を表すことができます。そして a と b は、次の多項式の 2 つの解になります。

$$X^2 - (a + b)X + ab = 0$$

特定の面積と周囲長に対し、解は 1 つ (a, b) であることに注意してください。

- **長方形の短辺**：粒子と同じ面積と周囲長を持つ四角形の短辺の長さ。この長さは、 b に等しくなります。

- **四角形辺比率**：等価四角形の長辺の短辺に対する比率
これは次のように定義されます。

$$\frac{\text{長方形の長辺}}{\text{長方形の短辺}} = \frac{a}{b}$$

等価四角形が長いほど、四角形辺比率も高くなります。

等価四角形が正方形に近いほど、四角形辺比率は 1 に近くなります。

形状の特徴

このセクションでは、次の形状の特徴パラメータについて説明します。

- **慣性 I_{xx} , I_{yy} , I_{xy} のモーメント**：粒子の重心に対する粒子内のピクセルの分布を表します
- **伸長因子**：粒子内の最長セグメントの垂直セグメントの平均長に対する比率。これは次のように定義されます。

$$\frac{\text{最大切片}}{\text{最大垂直切片}}$$

粒子の形状が長いほど、その伸長因子は高くなります。

- **分散度**：粒子を含む最小四角形の面積における、その粒子の面積の比率。これは次のように定義されます。

$$\frac{\text{粒子の面積}}{\text{最小四角形の面積}}$$

分散度は、インターバル (0, 1) に属します。粒子の形状が四角形に近いほど、分散度は 1 に近くなります。

- **ヘイウッド円形度**：同じ面積の円の周囲長に対する粒子の周囲長の比率。これは次のように定義されます。

$$\frac{\text{粒子の周囲}}{\text{粒子と同じ面積の円周}} = \frac{\text{粒子の周囲}}{2\sqrt{\pi} \times \sqrt{\text{粒子の面積}}}$$

粒子の形状が円に近いほど、ヘイウッド円形度は 1 に近くなります。

- **水力半径**：粒子の面積の周囲長に対する比率。これは次のように定義されます。

$$\frac{\text{粒子の面積}}{\text{粒子の周囲}}$$

粒子が半径 R の円の場合、その水力半径は次のように定義されます。

$$\frac{\pi R^2}{2\pi R} = \frac{R}{2}$$

水力半径は、円の半径 R の半分に等しく、次のようになります。

$$\frac{\text{円の面積}}{\text{円周}} = \frac{\text{粒子の面積}}{\text{粒子の周囲}}$$

- **Waddel ディスクの直径**：粒子と同じ面積を持つ円の直径。これは次のように定義されます。

$$\frac{2\sqrt{\text{粒子の面積}}}{\sqrt{\pi}}$$

以下に示す表は、測定値を求めるための式と、その式から得られる測定値の定義をリストにしたものです。

多様な測定値

これらの一次係数は、慣性モーメントや重心などの測定値の計算に使用されます。IMAQ Vision には、次の多様な測定値パラメータがあります。

- **X 軸の和**：粒子内の各ピクセルの x 座標の和
- **Y 軸の和**：粒子内の各ピクセルの y 座標の和
- **X 軸の二乗の和、Y 軸の二乗の和、X 軸と Y 軸の和**：粒子内の各ピクセルの x 座標の二乗の合計、 y 座標の二乗の合計、および xy 座標の合計
- **補正投影 X**：他の水平セグメントの上に重ならない水平セグメントの合計
- **補正投影 Y**：他の水平セグメントの上に重ならない垂直セグメントの合計

詳細の説明

基本測定値の定義

A	面積
ρ	周囲
Left	最も左のポイント
Top	最も上のポイント
Right	最も右のポイント

Bottom 最も下のポイント

 P_x x 軸への投影 P_y y 軸への投影**誘導測定値**

表 10-1 は、IMAQ Vision における誘導測定値を記述したものです。

表 10-1 誘導測定値

記号	誘導測定値	基本測定値
l	幅	右 - 左
h	高さ	下 - 上
d	対角線	$\sqrt{l^2 + h^2}$
M_x	重心 X	$(\Sigma x)/A$
M_y	重心 Y	$(\Sigma y)/A$
I_{xx}	慣性 XX	$(\Sigma x^2) - A \times M_x^2$
I_{yy}	慣性 YY	$(\Sigma y^2) - A \times M_y^2$
I_{xy}	慣性 XY	$(\Sigma xy) - A \times M_x \times M_y$
C_x	弦平均 X	A/P_y
C_y	弦平均 Y	A/P_x
C	平均垂直切片	凸閉包の面積 / 最大切片
A_{2b}	同等な楕円短軸	$4 \times A / (\pi S_{max})$
d°	方向	$I_{xx} = I_{yy}$ の場合、 $d^\circ = 45$ それ以外は、 $d^\circ = \frac{90}{\text{atan}(2 \times I_{xy} / (I_{xx} - I_{yy}))}$ $I_{xx} \geq I_{yy}$ で $I_{xy} \geq 0$ の場合、 $d^\circ = 180 - d^\circ$ $I_{xx} \geq I_{yy}$ で $I_{xy} < 0$ の場合、 $d^\circ = -d^\circ$ $I_{xx} < I_{yy}$ の場合、 $d^\circ = 90 - d^\circ$ $d^\circ < 0$ の場合、 $d^\circ = 0^\circ$
$E_{2\alpha}$	楕円の長軸 (2α)	$E_{2\alpha} = \sqrt{\frac{p^2}{2\pi^2} + \frac{2\pi}{A}} + \sqrt{\frac{p^2}{2\pi^2} - \frac{2\pi}{A}}$

表 10-1 誘導測定値 (続き)

記号	誘導測定値	基本測定値
E_{2b}	楕円の短軸 ($2b$)	$E_{2b} = \sqrt{\frac{\rho^2}{2\pi^2} + \frac{2\pi}{A}} - \sqrt{\frac{\rho^2}{2\pi^2} - \frac{2\pi}{A}}$
E_{ab}	楕円率	E_{2a} / E_{2b}
R_c	長方形の長辺	$1/4 (\rho + t')$ ここで $t' = \sqrt{\rho^2 - 16A}$
r_c	長方形の短辺	$1/4 (\rho - t')$ ここで $t' = \sqrt{\rho^2 - 16A}$
R_{Rr}	四角形辺比率	R_c / r_c
F_e	伸長因子	最大切片 / C
F_c	分散度	$A / (h \times l)$
F_H	ハイウッド円形因子	$\frac{\rho}{2\sqrt{\pi A}}$
F_f	タイプファクタ	$\frac{A^2}{4\pi\sqrt{I_{xx} \times I_{yy}}}$
R_h	水力半径	A / ρ
R_d	Waddel ディスクの直径	$2\sqrt{\frac{A}{\pi}}$

第 IV 部

マシンビジョン

このセクションでは、エッジ検出、パターンマッチング、寸法測定、カラー検査などのマシンビジョンアプリケーションで一般的に使用される高レベル操作の概念について説明します。

第 IV 部「マシンビジョン」には、以下の章が含まれます。

第 11 章、「[エッジ検出](#)」では、レーキ、同心レーキ、スポーク、およびキャリパなど、エッジの検出に使用するエッジ検出テクニックおよびツールについて説明します。

第 12 章、「[パターンマッチング](#)」では、パターンマッチングおよび形状マッチングについて説明します。

第 13 章、「[寸法測定](#)」では、解析ツール、クランプ、ラインフィット、および座標系について説明します。

第 14 章、「[カラー検査](#)」では、カラースペクトル、カラーマッチング、カラーの位置検出、およびカラーパターンマッチングについて説明します。

第 15 章、「[計測器リーダ](#)」では、メータ、LCD、およびバーコードについて説明します。

エッジ検出

本章では、レーキ、同心レーキ、スポーク、およびキャリパなど、エッジの検出に使用するエッジ検出テクニックおよびツールについて説明します。

概論

エッジ検出は、画像内のピクセルのラインに沿ってエッジを検出するものです。

エッジ検出ツールを使用して、画像のピクセル輝度の不連続部分を識別して検出します。不連続部分は通常、検出対象領域にあるオブジェクトの境界線を表す、ピクセル輝度値の急激な変化と関連しています。IMAQ Vision でエッジ検出ツールを使用するには、まず画像内に検索領域を指定します。

検索パスは、対話式またはプログラムの指定することができます。対話式に指定した場合は、ライン ROI ツールを使用して、解析する検索パスを選択することができます。また、定数値や前回の処理ステップの結果に基づいて、検索領域をプログラムの調整することもできます。たとえば、粒子解析やパターンマッチングアルゴリズムを使用して、前回検出した領域の特定の部分上でエッジを検出したい場合があります。エッジ検出ソフトウェアは、その領域に沿ってピクセルを解析してエッジを検出します。エッジ検出ツールは、領域内のすべてのエッジ、最初のエッジ、または最初とエッジのいずれかを検出するように構成できます。

使用目的

エッジ検出は、多くのマシンビジョンアプリケーションにおいて効果的なツールです。エッジ検出により、オブジェクトの境界線の位置や不連続部分の有無などに関する情報を得ることができます。

エッジ検出は、ゲー징、検出、およびアライメントの3つのアプリケーション領域で使用します。

ゲーシング

長さ、距離、直径、角度、カウントなどの重要な寸法測定を行って、検査対象の製品が正しく製造されていることを確認するためには、**ゲーシング**を使用します。コンポーネントまたは部品は、ゲーシングを行ったパラメータがユーザ定義の許容限度内に収まったかどうかにより、合格または不合格になります。

ゲーシングは、製造時にインラインまたはオフラインで使用されます。インライン処理の場合、各コンポーネントは製造されると同時に検査されます。インラインの目視ゲーシング検査は、機械部品検証、電子パッケージ検査、容器検査、ガラスピン検査、電子コネクタ検査などのアプリケーションで、幅広く使用されている検査テクニックです。

ゲーシングはまた、製品の品質のオフライン測定にも使用されます。製造ラインから製品のサンプルが抽出されます。次に、オブジェクトのある部分間の測定距離を確認して、そのサンプルが許容範囲内に入るかどうかを決定します。画像内にある別のエッジ間や、粒子解析やパターンマッチングテクニックを使用して測定した位置間の距離を測定することができます。また、エッジを連結して最適フィットライン、投影、交差点、および角度を求めることもできます。さらに、エッジの位置を使用して、円、楕円、多角形などの形状の推定測定値を計算することもできます。

図 11-1 は、ゲーシングアプリケーションでエッジ検出を使用して点火プラグの隙間の距離を測定する方法を示します。

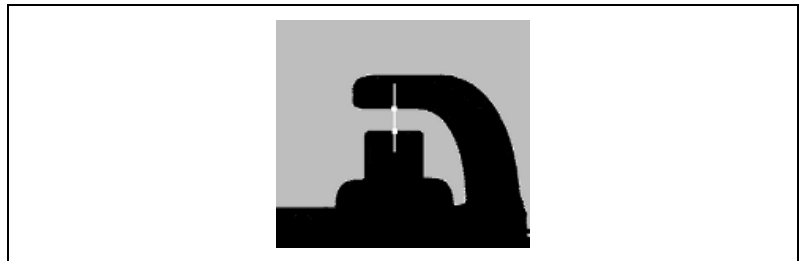


図 11-1 エッジ検出を使用したゲーシングアプリケーション

検出

部品の有無を検査するアプリケーションは、電子コネクタアセンブリや機械部品アプリケーションなどで一般的に使用されます。そのアプリケーションの目的は、ラインプロファイルとエッジ検出を使用して、部品があるかないかを調べることです。ラインプロファイル上のエッジは、背景と前景間のコントラストのレベルとその遷移の傾斜によって定義されます。このテクニックを使用すると、ラインプロファイル上のエッジの数をカウントしたり、推定したエッジ数と実際のエッジ数を比較したりすることができます。この方法は、画像相関やパターンマッチングなどの画像処理メソッドに対し、数値的計算の少ない代替方法として使用できます。

図 11-2 は、検索ラインプロファイル上で検出されたエッジ数によってコネクタが正しく組み立てられているかどうかを判断する、簡単な検出アプリケーションを示します。8つのエッジが検出されたということは、4つのワイヤがあることを示します。それ以外の数のエッジが検出された場合、その部品は正しく組み立てられていないということになります。

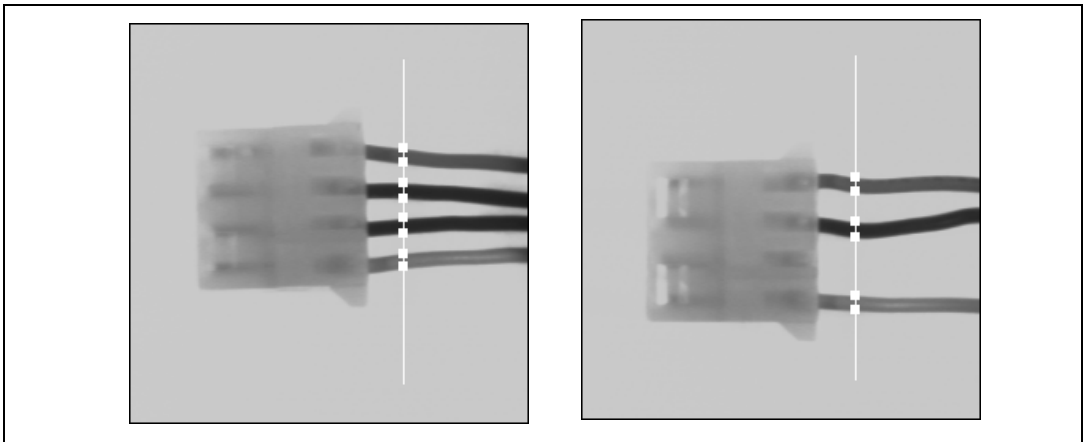


図 11-2 エッジ検出を使用したコネクタ検査

また、エッジ検出を使用して、部品上のひびやかき傷のような外観的欠陥など、構造的欠陥を検出することもできます。部品の輝度が均一の場合、そのような欠陥が輝度プロファイルで急激な変化として表示されます。エッジ検出ではそのような変化を識別します。

アライメント

アライメントは、部品の位置と方向を確定します。多くのマシンビジョンアプリケーションで、検査対象のオブジェクトが画像内で異なる位置にある場合があります。エッジ検出では、関心領域のみを検査できるようにするため、検査を実行する前に画像内のオブジェクトの位置を検出します。部品の位置と方向は、ステージなどの位置決めデバイスに情報をフィードバックする際にも使用することができます。

図 11-3 は、画像内のディスクの左側境界線を検出する例を示します。エッジの位置情報を使用して、ディスクの向きを判断することができます。

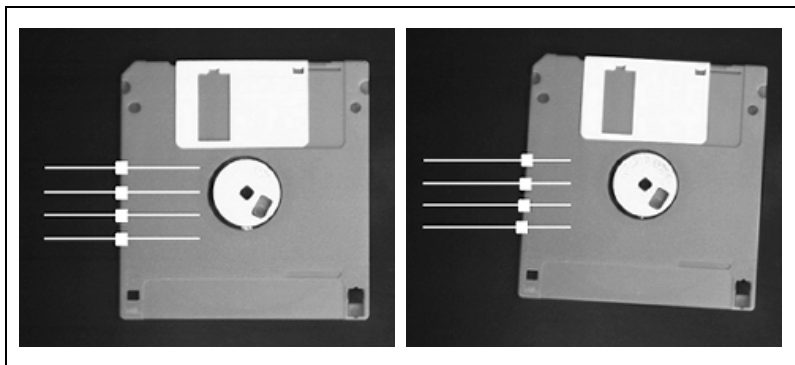


図 11-3 エッジ検出を使用したアライメント

エッジ検出の概念

エッジの定義

エッジとは、画像内の隣接するピクセル間におけるグレースケール値の急激な変化です。IMAQ Vision の場合、エッジ検出は、図 11-4 に示すように、検索領域上のピクセル値の一次元プロファイルに対して動作します。一次元の検索領域は、ライン、円または楕円の周囲、四角形や多角形の境界線、またはフリーハンド領域のいずれかになります。ソフトウェアは、プロファイル上のピクセル値を解析し、輝度の有意変化を検出します。輝度変化の特徴を指定することにより、どのような変化がエッジを表すかわかります。

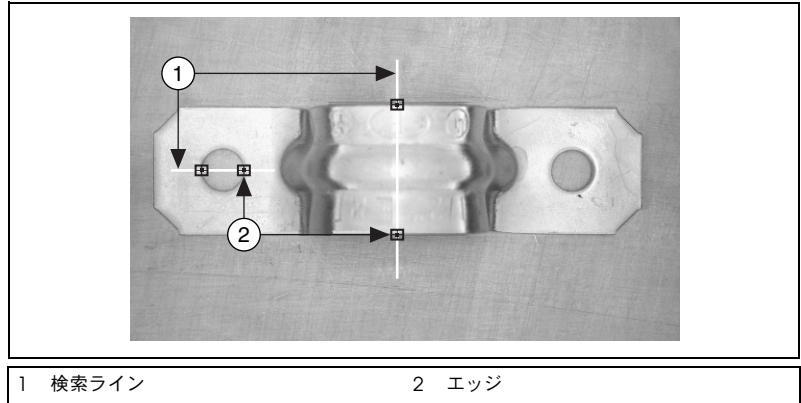


図 11-4 ブラケットのエッジの例

エッジの特徴

図 11-5 は、エッジの特徴を示すための一般的なモデルを示します。

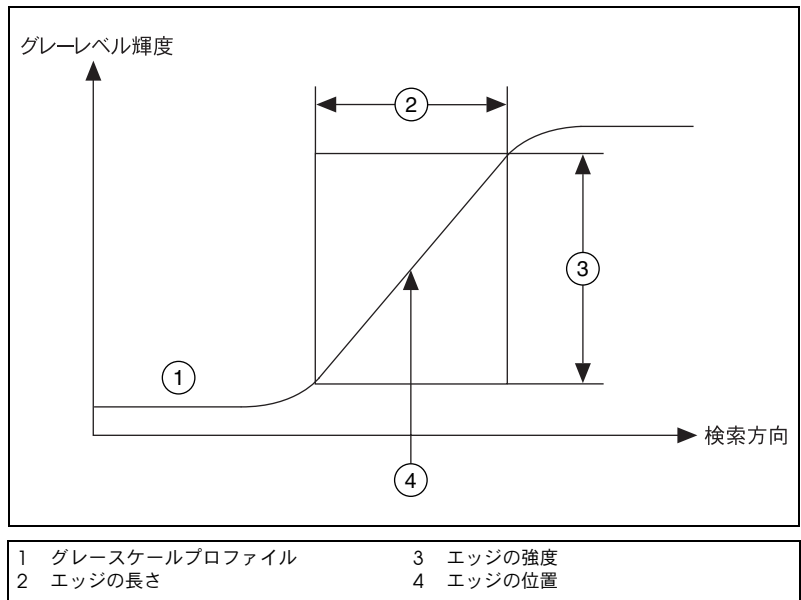


図 11-5 エッジモデル

このモデルの主要なパラメータには、次のようなものがあります。

- **エッジの強度**：背景とエッジの間のグレースケール値の最小差を定義します。エッジの強度は、エッジコントラストともいいます。図 11-6 は、エッジ強度の異なる画像を示します。エッジの強度は、次のようなことを含めさまざまな原因により変化します。
 - 照明状態：その場の全体の光量が低い場合、画像内のエッジの強度も低くなります。図 11-6 は、異なる照明条件による、オブジェクトの境界線上のエッジ強度の変化を示します。
 - グレースケール値の異なるオブジェクト：非常に明るいオブジェクトが存在すると、全体に輝度の低い他のオブジェクトのエッジ強度が低くなります。

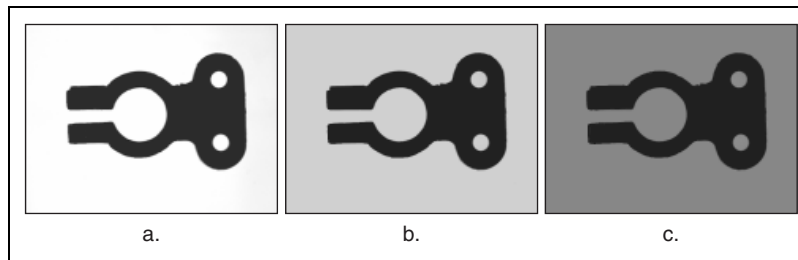


図 11-6 強度の異なるエッジの例

- **エッジの長さ**：エッジと背景の間で、指定のグレースケール値の差が生じる最大距離を定義します。長さは、エッジの傾斜を表すものです。背景とエッジの間に緩やかな傾斜のあるエッジを検出するには、長さを長く設定します。
- **エッジの極性**：エッジが立ち上がりか立ち下がりかを指定します。立ち上がりエッジとは、エッジを交差する際にグレースケール値が上がる部分です。立ち下がりエッジとは、エッジを交差する際にグレースケール値が下がる部分です。エッジの極性は、検索方向と関連しています。図 11-7 は、エッジの極性の例を示します。
- **エッジの位置**：画像内のエッジの x, y 位置を示します。図 11-5 は、エッジモデルのエッジの位置を示すものです。

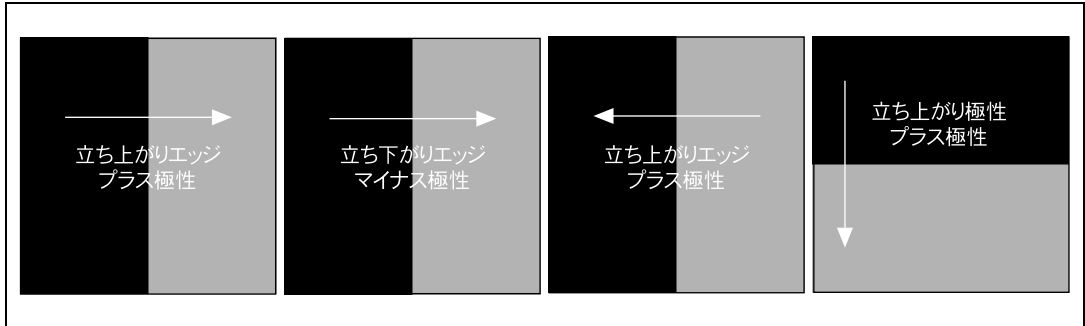


図 11-7 エッジの極性

エッジ検出メソッド

IMAQ Vision でのエッジ検出には 2 つの方法があります。どちらの方法も、一次元プロファイル上の各ピクセルのエッジ強度を計算するものです。エッジの強度が最小強度より大きい場合に、エッジが生じることになります。さらなる検査でエッジの正確な位置を検出します。最小強度を指定するには、ソフトウェアのコントラストパラメータを使用します。

シンプルエッジ検出

ソフトウェアは、ピクセルプロファイル上の任意のポイントのピクセル値を使用して、そのポイントのエッジ強度を定義します。エッジポイントを検出するため、ソフトウェアはピクセルプロファイルを最初から最後までピクセルごとにスキャンします。立ち上がりエッジは、しきい値にヒステリシス値を足した値より、ピクセル値が高くなった最初のポイントで検出されます。このしきい値を設定して、検出しようとしているエッジの最小エッジ強度を定義します。ヒステリシス値を使用して、立ち上がりエッジと立ち下がりエッジに対し異なるエッジ強度を指定します。立ち上がりエッジが検出されると、ソフトウェアは立ち下がりエッジを検索します。立ち下がりエッジは、ピクセル値が指定したしきい値より下がったときに検出されます。このプロセスは、ピクセルプロファイルの最後に達するまで繰り返されます。プロファイル上の最初のエッジは、立ち上がりと立ち下がりどちらの場合もあり得ます。図 11-8 は、シンプルエッジモデルを示しています。

シンプルエッジ検出メソッドは、画像のノイズが少なくオブジェクトと背景の間に明確な境界がある場合に特に有効です。

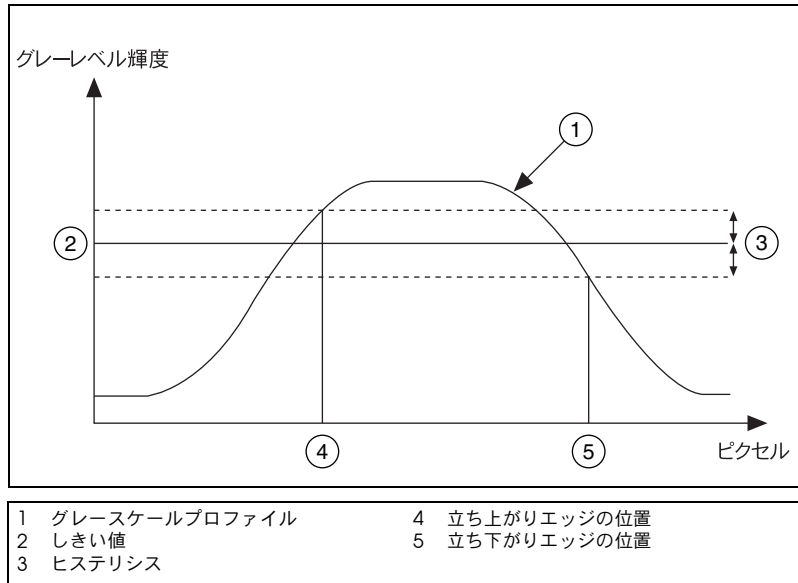


図 11-8 シンプルエッジ検出

上級エッジ検出

ピクセルプロファイル上のあるポイントにおけるエッジ強度を計算するため、ソフトウェアはその解析ポイントの前後のピクセルを平均します。ポイントの後で平均されたピクセルは、そのポイントから一定のピクセル距離にある場合があり、それは傾斜パラメータを設定することにより定義します。この数字は、エッジプロファイルでの予想遷移領域に対応します。ポイントの前と後それぞれで平均するピクセル数を定義するには、幅パラメータを設定します。平均を計算すると、ソフトウェアはそれらの平均値の差を計算してコントラストを求めます。プロファイル上のノイズによる影響を軽減するには、フィルタを使用します。画像に多くのノイズが含まれると考えられる場合は、大きいフィルタ幅を使用します。図 11-9 は、パラメータとエッジプロファイルの関係を示します。

エッジを検出するため、ソフトウェアは一次元のグレースケールプロファイル上を 1 ピクセルずつスキャンします。各ポイントでエッジ強度（コントラスト）を計算します。現在のポイントでのコントラストが、エッジの最小コントラストとしてユーザ設定した値より大きい場合、さらに解析を行うためポイントは保存されます。このポイントから始まっていくつかの連続するポイントを解析し、コントラストが最大値に達した後その値から下がるポイントを検出します。コントラストが最大値に達するポイント

を、開始エッジ位置といいます。傾斜パラメータの値を開始エッジ位置に追加して、終了エッジ位置を求めます。開始エッジ位置と終了エッジ位置の間で、開始エッジ値とポイントの輝度値の差が開始エッジ値と終了エッジ値の差の半分以上になる最初のポイントが、エッジ位置として返されます。

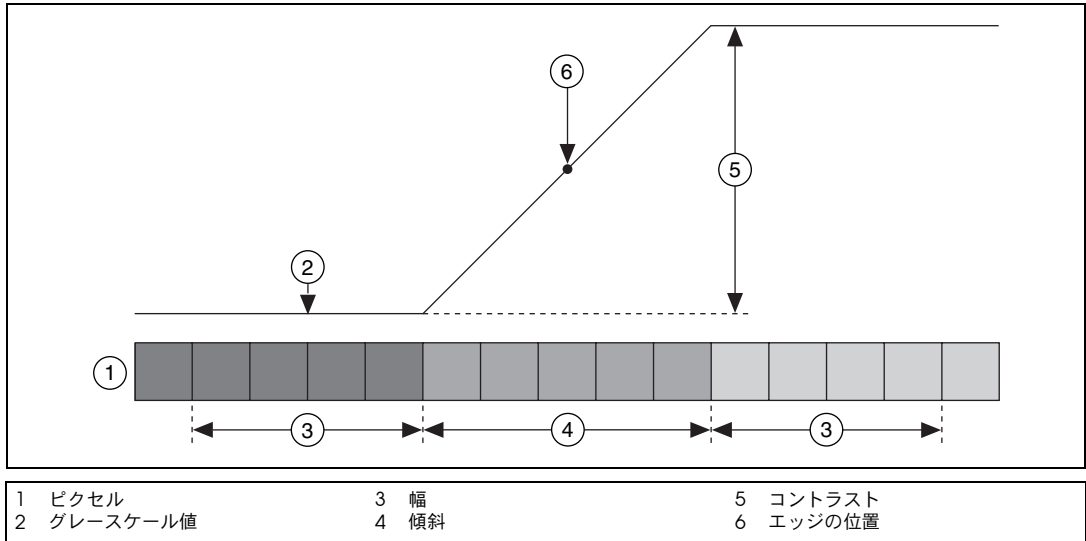


図 11-9 上級エッジ検出

サブピクセル精度

画像の解像度が十分に高い場合は、ほとんどの計測アプリケーションでピクセル精度のみにより正確な測定を行うことができます。ただし、利用可能なセンサのサイズに制限があるため、マシンビジョンアプリケーションで必要な最小画像解像度を実現するのは難しい場合があります。そのような場合は、サブピクセル精度でエッジの位置を検出する必要があります。

サブピクセル解析は、より解像度の高いイメージングシステムで得ることのできるピクセル値を推測するためのソフトウェアメソッドです。サブピクセル精度でエッジの位置を計算するために、エッジ検出ソフトウェアはまず、二次関数や三次関数などの上位補間関数をピクセルの輝度データにフィットします。

補間関数は、エッジ検出アルゴリズムに元のピクセル値の間のピクセル輝度値を提供します。次にソフトウェアは、その輝度情報を使用して、サブピクセル精度によってエッジの位置を検出します。

図 11-10 は、三次スプライン関数が一連のピクセル値にフィットする方法を示します。このフィットを使用して、ピクセル間の位置にある値が推測されます。エッジ検出アルゴリズムでは、これらの値を使用して、サブピクセル精度でエッジの位置を推測します。

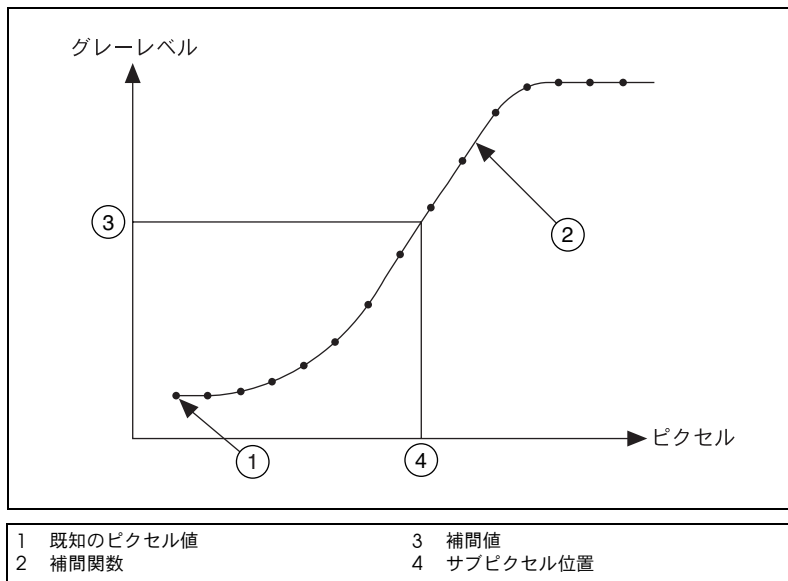


図 11-10 補間を使用したサブピクセル情報の取得

現在利用できるイメージングシステムコンポーネントとソフトウェアツールを使用すれば、1/4 のサブピクセル精度で確実な推測を行うことができます。ただし、推測の結果は照明状態やカメラレンズなどのイメージングセットアップに大きく影響されます。サブピクセル情報を使用する前に、画像の解像度を向上させることを試みてください。画質の向上に関する詳細は、『IMAQ Vision User Manual』を参照してください。

エッジ検出を二次元検索領域に拡張する

IMAQ Vision のエッジ検出は、一次元プロファイルに対して動作します。ただし、レーキ、スポーク、および同心レーキの各ツールを使用して、エッジ検出ツールの使用を二次元検索領域に拡大することができます。そのようなエッジ検出では、二次元検索領域は複数の検索ラインからなり、その上でエッジ検出が行われます。検索領域として使用する検索ラインの数を調整するには、ライン間の幅を設定します。

レーキ

レーキは、四角形の検索領域に対して動作します。検索ラインは四角形の方角と平行に描画されます。領域内のライン数を制御するには、横方向の四角形に対し検索方向を左から右または右から左に指定します。縦方向の四角形では検索方向を上から下または下から上に指定します。図 11-11 は、レーキ関数の基本を示しています。

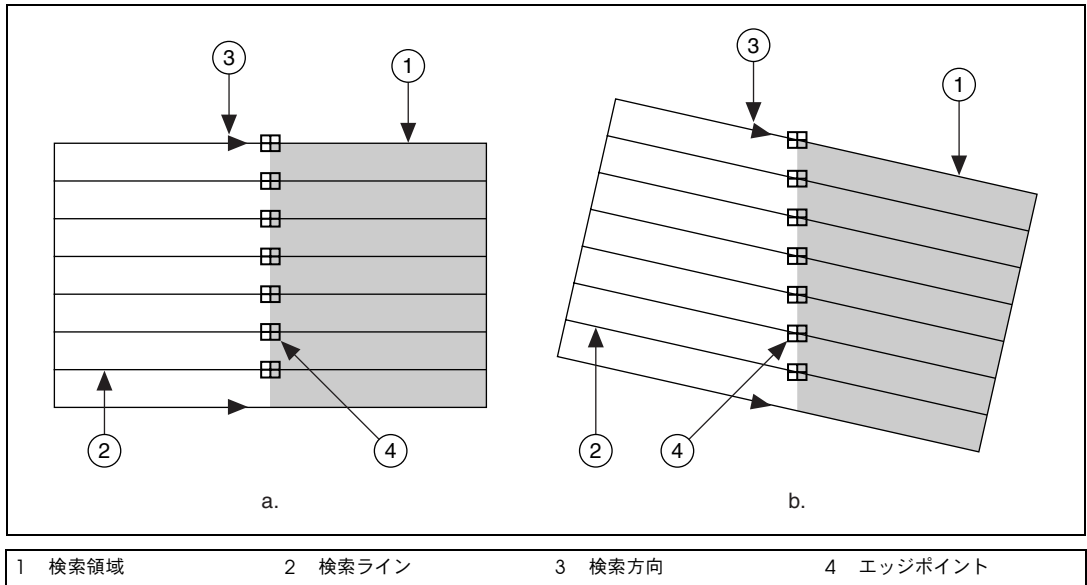


図 11-11 レーキ関数

スポーク

スポークは、環状検索領域に対して動作し、領域の中心から外に向かって検索領域内に描かれた検索ラインに沿って検索します。領域内のライン数を制御するには、各ライン間の角度を指定します。検索方向は、中心から外へ、あるいは外から中心へのいずれかに指定します。図 11-12 は、スポーク関数の基本を示しています。

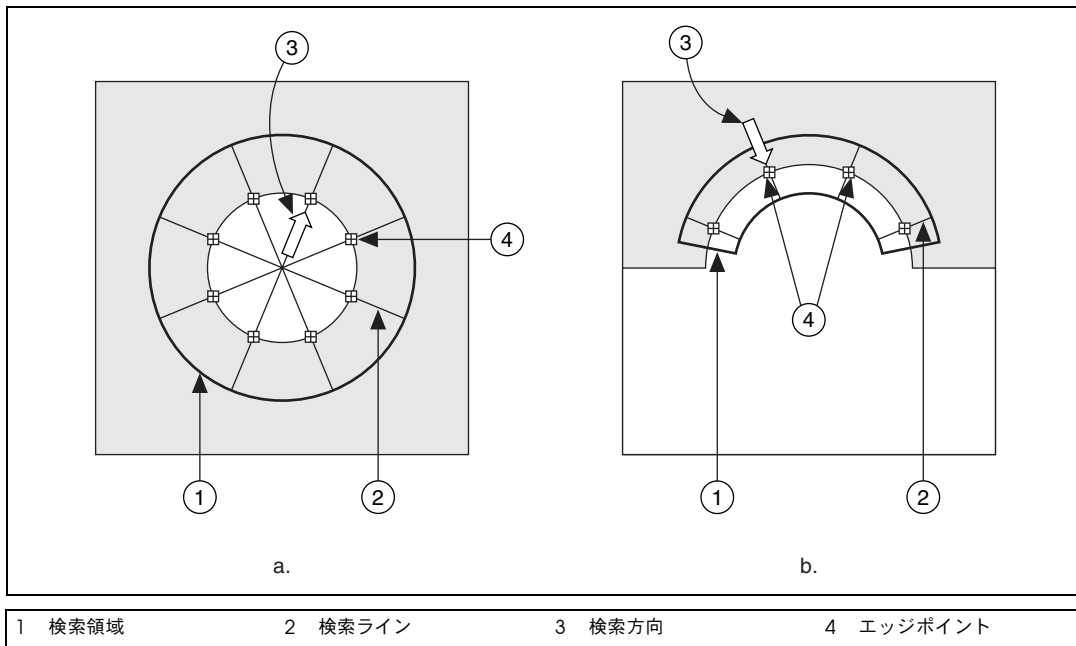


図 11-12 スポーク関数

同心レーキ

同心レーキは、環状検索領域に対して動作します。同心レーキは、レーキを環状領域に適応させたものです。図 11-13 は、同心レーキの基本を示しています。検索領域内にあり、外側の境界線に対して同心の検索ラインに沿って、エッジ検出を行います。エッジ検出に使用する同心検索ラインの数を制御するには、同心ライン間の半径距離を指定します。検索方向は、時計回りまたは時計と反対回りで指定します。

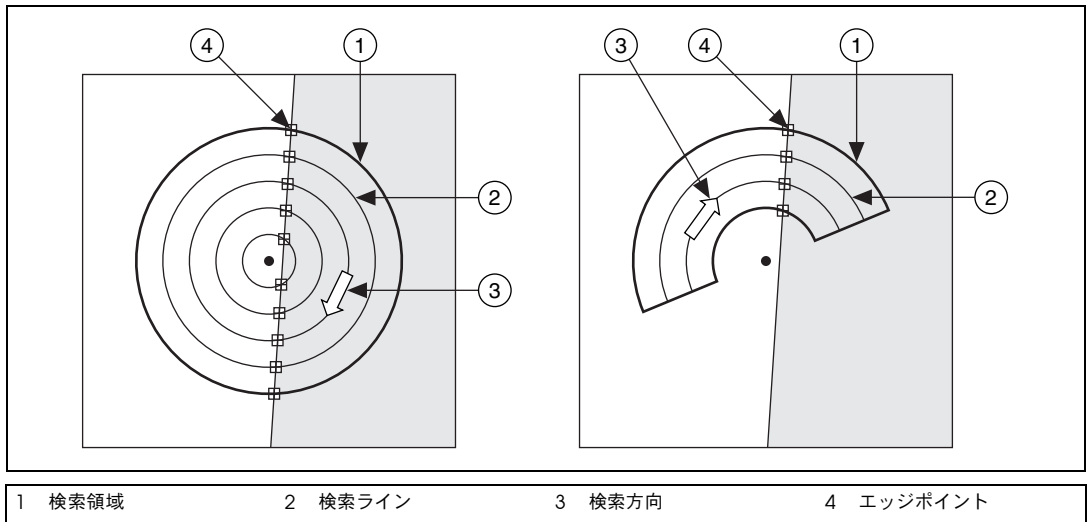


図 11-13 同心レーキ関数

パターンマッチング

本章では、パターンマッチングと形状マッチングについて説明します。

概論

パターンマッチングでは、グレースケール画像の中から、あらかじめ指定したテンプレートと一致する領域を検出します。パターンマッチングは、照明の悪い環境や、ぼやけ、ノイズ、テンプレートのシフトや回転にかかわらず、テンプレートとの一致を検出します。

画像内で既知の基準パターンをすばやく探し出すには、パターンマッチングを使用します。パターンマッチングを実行する際は、検索するオブジェクトを表すモデルまたはテンプレートを作成します。次に、マシンビジョンアプリケーションが各集録画像からモデルを検索し、各一致項目に対しスコアを計算します。このスコアは、検出したパターンがモデルにどの程度近いかを表すものです。

使用目的

パターンマッチングアルゴリズムは、さまざまなアプリケーションに使用され、画像処理関数の中でも最も重要なものです。パターンマッチングは、以下の3つのタイプのアプリケーションで使用することができます。

- **アライメント**：基準パターンを見つけて、既知のオブジェクトの位置と方向を求めます。基準パターンは、オブジェクト上の基準ポイントとして使用します。
- **ゲー징**：長さ、直径、角度、その他の重要な寸法を測定します。測定値が設定した許容レベル内に入らない場合、コンポーネントは不合格になります。パターンマッチングを使用して、測定するオブジェクトを検出します。
- **検査**：欠損している部品や不明確な印刷など、単純な欠陥を検出します。

IMAQ Vision のパターンマッチングツールは、モデルまたはテンプレートと呼ばれる、特徴を理想的に表したものと、画像内に含まれる特徴との間の類似点を測定します。**特徴**とは、画像内にあるピクセルの特定パターンとして定義します。

パターンマッチングは、多くのアプリケーションで重要な役割を果たします。パターンマッチングを使用すると、画像内に存在するテンプレートの発生数と位置に関する情報をアプリケーションに提供することができます。たとえば、プリント基板を含む画像から、1つまたは複数のラインメントマーク（基準パターン）を検索することができます。マシンビジョンアプリケーションは、マークを使用して、チップ取り付け装置からチップを配置するためにボードの位置を合わせます。図 12-1a は、回路基板の一部を示します。図 12-1b は、プリント基板 (PCB) 検査またはチップのピックアンドプレイスアプリケーションで使用する一般的な基準パターンを示します。

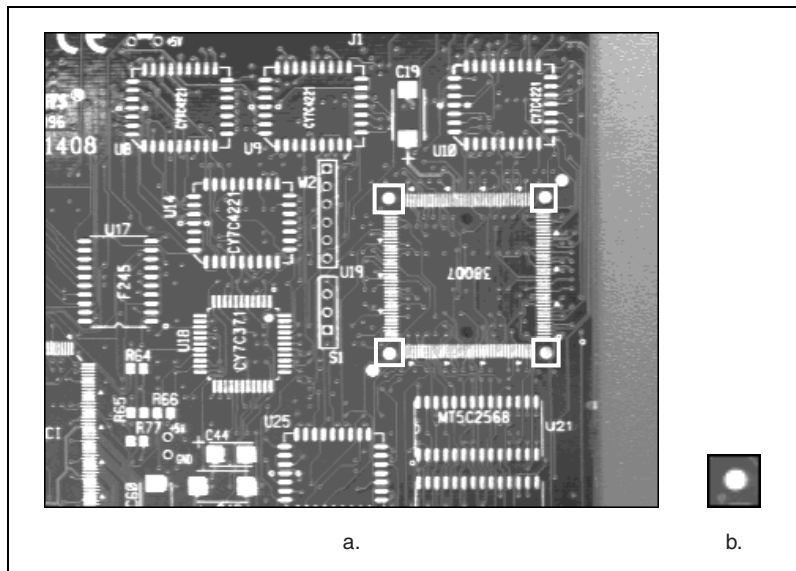


図 12-1 一般的な基準パターンの例

ゲージングアプリケーションは、それらのオブジェクトの位置を検出してからそれらの間の距離を測定、つまりゲージングします。測定値が許容範囲内に収まる場合は、部品は良品とみなされます。許容範囲外の場合は、コンポーネントは不合格となります。特徴を検索して見つけることが、クワッドパックのリード線やアンチロックブレーキセンサの検査など、多くのゲージングアプリケーションの成功を決定付ける、きわめて重要な処理タスクとなります。リアルタイムアプリケーションでは、検索速度が非常に重要なポイントとなります。

パターンマッチングの概念

パターンマッチングツールの機能

パターンマッチングは多くのマシンビジョンアプリケーションにおける最初のステップとなるため、さまざまな状況下で確実に動作する必要があります。自動化されたマシンビジョンアプリケーションでは、検査対象の物質または部品の視覚的外観は、部品の向き、倍率の変化、および照明の変化などにより変わることがあります。パターンマッチングツールを使用すると、そのような変化にかかわらず基準パターンを検出することができます。パターンマッチングツールにより正確な結果が得られる一般的な状況を以下に示します。

パターンの向きおよび複数のインスタンス

パターンマッチングツールを使用すると、画像内のパターンが回転していたり倍率が変化していても、基準パターンを見つけることができます。パターンが画像内で回転したり倍率が変化している場合、パターンマッチングツールは以下を検出します。

- 画像内のパターン
- 画像内におけるパターンの位置
- パターンの向き
- 画像内に発生するパターンの複数のインスタンス（適用される場合）

図 12-2a は、テンプレート画像、つまりパターンを示します。図 12-2b は、テンプレートが画像内で移動したものです。図 12-2c は、テンプレートが画像内で回転したものです。図 12-2d は、テンプレートが画像内で拡大されたものです。図 12-2b から図 12-2d は、テンプレートの複数のインスタンスを示します。

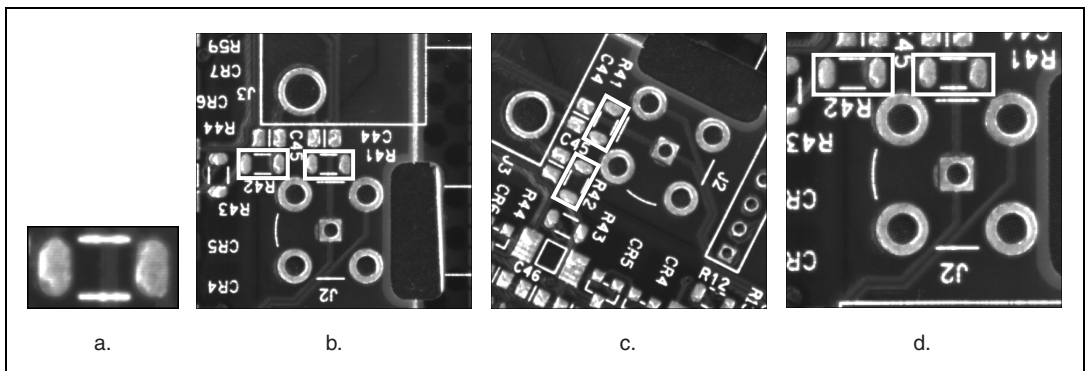


図 12-2 パターンの向きおよび複数のインスタンス

周辺光状態

パターンマッチングツールは、照明が画像全体で均等に变更されている条件下で、画像から基準パターンを検出することができます。図 12-3 は、パターンマッチングが正しく動作する典型的な状況を示します。図 12-3a は、元のテンプレート画像を示します。図 12-3b は、同じパターンを明るい照明下で表したものです。図 12-3c は、そのパターンを暗い照明下で表したものです。

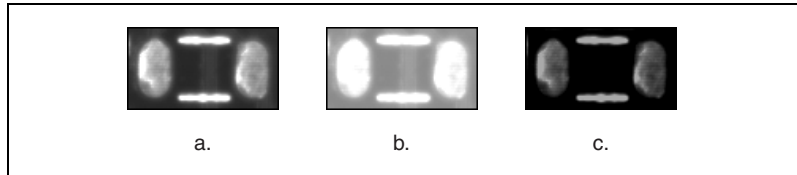


図 12-3 照明状態の例

ぼやけとノイズの状態

パターンマッチングでは、ぼやけやノイズによって変形されたパターンを検出することができます。ぼやけは、通常誤った焦点や被写界深度の変化によって発生します。図 12-4 は、パターンマッチングが正しく動作する一般的なぼやけやノイズの状態を示します。図 12-4a は、元のテンプレート画像を示します。図 12-4b は、ぼやけによる画像の変化を示します。図 12-4c は、ノイズによる画像の変化を示します。

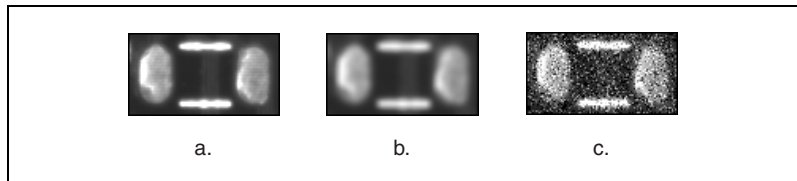


図 12-4 ぼやけとノイズの例

パターンマッチングのテクニック

パターンマッチングには、既存のテクニックや新しいテクニック、さらに粒子解析などその他のテクニックがあります。

既存のパターンマッチング

既存のパターンマッチングテクニックには、正規化相互相関、ピラミッドマッチング、およびスケールに影響されないマッチングがあります。

相互相関

正規化相互相関は、画像からテンプレートを検出するための最も一般的な方法です。相関の基礎となるメカニズムは一連の乗算処理に基づいているため、相関プロセスは時間のかかる処理です。MMX などの新技術を使用すると、並行乗算処理が可能になり、計算にかかる全体の時間が短縮されます。また、画像のサイズを縮小し、画像のマッチングを行う領域を限定すると、マッチングプロセスの速度を上げることができます。ただし、基本的な正規化相互相関操作は、多くのアプリケーションの速度条件を満たすことができません。

ピラミッドマッチング

画像とテンプレートのサイズを縮小することによって、計算にかかる時間を短縮することができます。そのようなテクニックの 1 つに、ピラミッドマッチングがあります。この方法では、画像とテンプレートのどちらも低い空間解像度の方にサブサンプルされます。画像およびテンプレートは、元のサイズの 4 分の 1 に縮小することができます。マッチングは、まず縮小した画像で実行します。画像のサイズが小さいため、マッチングは短時間で終わります。マッチングが完了すると、次は一致度の高い部分のみを元の画像でのマッチング領域として使用することができます。

スケールに影響されないマッチング

正規化相互相関は、画像内のパターンが回転や倍率変更されていない場合に、パターンを検出するのに適した方法です。一般に相互相関では、最大で 5° から 10° 回転した同じサイズのパターンを検出することができます。相関を拡張して、倍率や回転に影響されないようにパターンを検出するのは困難です。

スケールに影響されないマッチングの場合、テンプレートの倍率変更、つまりサイズ変更の操作を繰り返し行ってから、相関処理を実行する必要があります。これにより、マッチングプロセスにかなりの計算量が上乘せされます。回転に影響されないよう正規化するのは、さらに困難です。回転に関する手がかりを画像から得られる場合は、単にテンプレートを回転して相関を行うことができます。ただし、回転に関して何もわからない場合は、最適な一致を見つけるためにテンプレートをくまなく回転させる必要があります。

また、FFT を使用して周波数領域で相関を実行することもできます。画像とテンプレートが同じサイズの場合は、この方法の方が空間領域で相関を行うより効率の高い方法です。周波数領域では、画像の FFT にテンプレートの FFT の複素共役を乗算して相関を求めます。正規化相互相関は、周波数領域で実装する方がはるかに難しい処理になります。

新しいパターンマッチングのテクニック

従来のパターンマッチングメソッドには制約があります。IMAQ Vision で使用しているような新しいメソッドでは、**画像理解**テクニックを組み入れてテンプレート情報の解釈を試み、この情報を使用して画像からテンプレートを検索します。画像理解とは、テンプレート画像の特徴に関する情報を生成する画像処理テクニックです。そのメソッドには次のようなものがあります。

- 画像の幾何学的モデル
- 効率的な画像の不均等サンプリング
- 回転とスケールに影響されないテンプレート情報の抽出

これらのテクニックを使用することにより、画像またはパターンを完全に特徴づけるのに必要な情報量が削減され、検索処理にかかる時間が短縮されます。さらに、テンプレートから有用な情報を抽出し、不要でノイズの多い情報を除去することにより、さらに精度の高い検索結果が得られます。

新しいパターンマッチングテクニックの 1 つでは、不均等サンプリングを使用します。ほとんどの画像には不要な情報が含まれているため、画像内のすべての情報を使ってパターンを検索することは時間がかかる上不正確な作業です。画像をサンプリングして、画像全体の内容を表すいくつかのポイントを抽出することにより、パターンマッチングツールの速度と精度を向上させることができます。図 12-5 は、パターンマッチングで使用するサンプリングテクニックのよい例を示します。図 12-5a は、元のテンプレート、つまり基準画像を示します。図 12-5b の黒い点は、テンプレートを表すのに使用される画像上のポイントを示します。

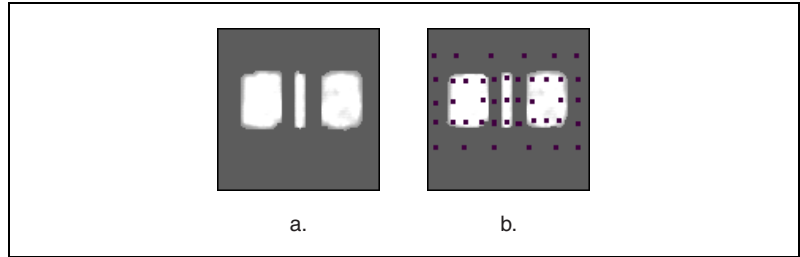


図 12-5 パターンマッチングサンプリングテクニックのよい例

パターンマッチングの新しいテクニックの多くで、画像のエッジ情報を使用して画像の構造に関する情報を提供します。画像内の情報量は、エッジに関する有為データだけに縮小されます。

エッジ画像をさらに処理して、画像内に存在する直線や円の数など、その画像に関する高レベルな幾何学的情報を抽出することができます。その後、テンプレートと画像の間でのエッジや幾何学情報のマッチングに絞って、パターンマッチングを行います。図 12-6 は、エッジと幾何学的モデルの重要性を示すものです。図 12-6a は基準パターンを示し、図 12-6b はパターン内のエッジ情報を示し、図 12-6c はエッジの高レベルな幾何学的解釈を円やラインなどの幾何学的オブジェクトの形で表したものです。

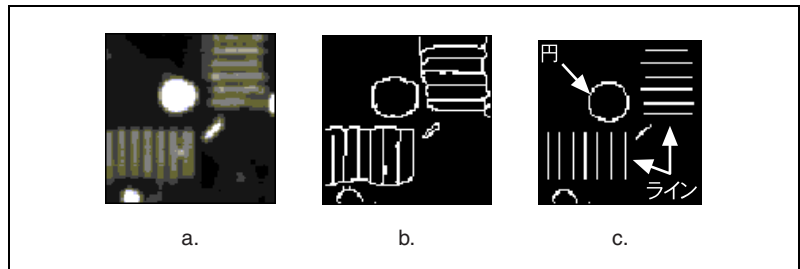


図 12-6 エッジ検出およびパターンマッチングテクニック

IMAQ Vision では、画像のエッジ情報とインテリジェント画像サンプリングテクニックを組み合わせ、パターンのマッチングを行います。画像内でパターンが回転する可能性がある場合は、同様のテクニックを使用しますが、テンプレートにはパターンの回転を反映した値を持つピクセルを特別に選択します。その結果高速で正確なパターンマッチングが実行されます。IMAQ Vision のパターンマッチングを使用すると、サイズ ($\pm 5\%$) や向き (0° から 360°) が変化したり画質が低下していても、正確にオブジェクトを検出することができます。

詳細の説明

このセクションでは、効果的なパターンマッチングツールを構築するために役立つ追加情報を提供します。

相互相関

相関の基本概念は以下の通りです。サイズ $M \times N$ の画像 $f(x,y)$ の中にあるサイズ $K \times L$ のサブ画像 $w(x,y)$ について考えてみます。ここで $K \leq M$ および $L \leq N$ です。ポイント (i,j) における $w(x,y)$ と $f(x,y)$ の相関は、次の式で求められます。

$$C(i,j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x,y)f(x+i,y+j)$$

ここで $i=0,1,\dots,M-1, j=0,1,\dots,N-1$ となり、この合計は画像の w と f が重なる部分を表しています。

図 12-7 は、相関の手順を示しています。画像の基点 f が左上コーナーにあるとします。相関とは、テンプレートまたはサブ画像 w を画像領域内で移動させて、その領域での値 C を計算するプロセスです。それを行うには、テンプレートの各ピクセルにそのピクセルが重なる画像ピクセルを乗算し、次にその結果をテンプレートのすべてのピクセル上に加算します。 C の最大値は、 w が f に最も適合する位置を示します。相関値は、画像の縁部分では正確ではありません。

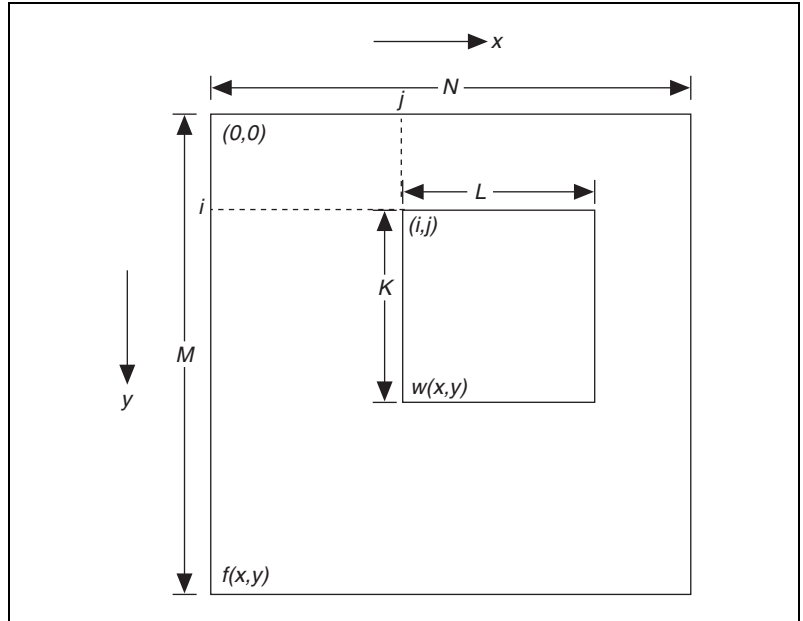


図 12-7 相関の手順

基礎相関は、たとえば輝度など、画像やテンプレートでの振幅の変化に非常に敏感です。たとえば、画像 f の輝度が倍になると、 c の値も同じく倍になります。ただしそれは、次の式を使って正規化相関係数を計算することにより解決できます。

$$R(i,j) = \frac{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \bar{w})(f(x+i,y+j) - \bar{f}(i,j))}{\left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \bar{w})^2 \right]^{1/2} \left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (f(x+i,y+j) - \bar{f}(i,j))^2 \right]^{1/2}}$$

ここで \bar{w} (一度のみ計算) は、テンプレート w のピクセルの平均輝度値です。変数 \bar{f} は、 w の現在位置と一致する領域の f の平均値です。 R の値は -1 から 1 の範囲にあり、 f および w の輝度値のスケールの変化には影響されません。

形状マッチング

形状マッチングでは、バイナリ画像を使用してある形状の有無を検索し、各一致形状の位置を特定します。IMAQ Vision は、回転や倍率の変化にかかわらず、形状を検出します。

バイナリ形状マッチングを実行するには、回転やスケールに影響されない、オブジェクトの形状を表すテンプレートオブジェクトからパラメータを抽出します。そして、それらのパラメータを、他のオブジェクトから抽出した同様のパラメータのセットと比較します。バイナリ形状マッチングは、サイズや向きにかかわらず特徴を検出するという利点があります。

使用目的

形状がわかっているがサイズと向きがわからない特徴を検索していて、画像の二値化が可能な場合は、バイナリ形状マッチングの使用を検討してみてください。

バイナリ形状マッチングは、ロボットアームによる部品の類似形状ごとのソーティングなど、ロボットガイダンスアプリケーションに適しています。

形状マッチングの概念

形状マッチング関数の入力は以下の通りです。

- 検索する形状の画像（テンプレート）
- 分類する部分を含むバイナリ画像
- テンプレートと分類する部分の間に許される不一致の程度を示す許容レベル

検索後は、一致した部分のみを含む画像と、各部分の位置、重心、およびテンプレートと各部分との一致度を示すスコアを記述したレポートが出力されます。

図 12-8 は、バイナリ形状マッチングを使用して、ワイパーの部品をさまざまな形状に分類する方法を示しています。図 12-8a は、形状テンプレートを示します。図 12-8b は、さまざまなワイパーの部品の元のグレースケール画像を示します。図 12-8c は、元の画像をバイナリ（または二値化）にしたものです。図 12-8d は、形状マッチング関数の出力結果を示します。形状マッチング関数は、サイズや向きの変化にかかわらず、画像から検索対象の形状を検出します。

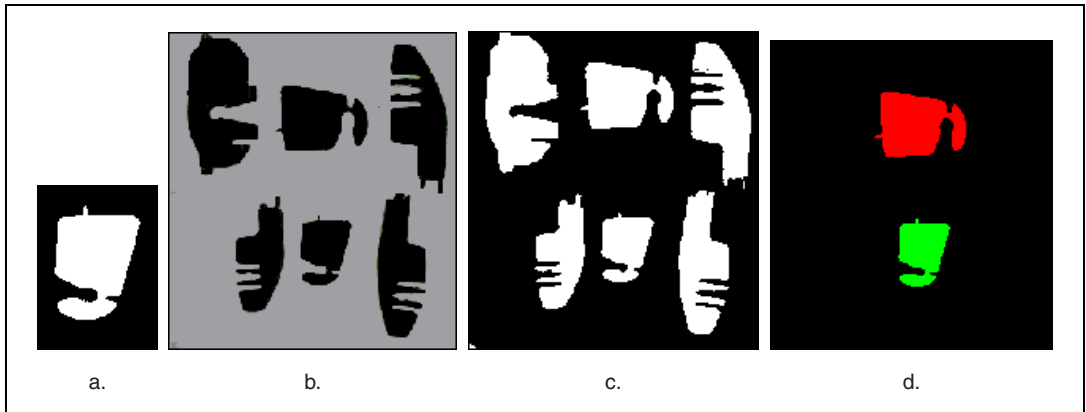


図 12-8 形状マッチングを使用したワイパー部品の検索

寸法測定

本章では、座標系、解析ツール、およびクランプについて説明します。

概論

IMAQ Vision で寸法測定、つまり**ゲージング**ツールを使用して、距離、角度、面積、ラインフィット、円形フィット、カウント数などの重要な定量化可能な距離測定を行い、製品が正しく製造されているかどうかを確認することができます。

コネクタ、スイッチ、リレーなどのコンポーネントは、小さな部品で大量に製造されるものです。そのようなコンポーネントを人間によって目視検査するのは、手間と時間がかかり、また結果が一定ではありません。IMAQ Vision では、それらのコンポーネントの特定の特徴に対してすばやく一貫性のある測定を行い、その結果をレポートに生成することができます。測定した距離やカウントがユーザ指定の許容範囲に入らない場合、そのコンポーネントまたは部品は生産仕様を満たしていないことになり、検査不合格となります。

使用目的

ゲージングは、部品の画像から得られる重要な寸法情報に対して、検査による決定を行うアプリケーションで使用します。ゲージングは、製造時にインラインまたはオフラインで使用されます。インライン処理の場合、各コンポーネントは製造過程で検査されます。インラインゲージング検査は、機械アセンブリ確認、エレクトロニクスパッケージ検査、容器検査、ガラスピン検査、電子コネクタ検査などに使用されます。

また、ゲージングを使用して製品の品質をオフラインで測定することもできます。まず製造ラインから製品のサンプルが抽出されます。次に、オブジェクトの特徴の間の測定距離を使用して、そのサンプルが許容範囲内に入るかどうかを決定します。また、ゲージングテクニックを使用すると、バイナリ画像で粒子やエッジ間の距離を測定でき、画像測定の定量化が容易になります。

寸法測定の概念

ゲーシングプロセスには 4 つのステップがあります。

1. 画像内でコンポーネントまたは部品を検出します。
2. 部品の別々の部分から特徴を検出します。
3. それらの特徴を使って測定を行います。
4. 測定値を仕様と比較して、その部品が検査に合格しているかどうかを判断します。

画像内で部品を検出する

一般的なゲーシングアプリケーションでは、画像全体ではなく ROI から測定値を抽出します。このテクニックを使用するためには、オブジェクトの必要な部分が、定義した ROI 内に常に表示されている必要があります。

通常、検査対象のオブジェクトは、処理画像内で移動したり回転して表示されてしまいます。そのような場合、ROI もオブジェクトと同様に移動または回転する必要があります。オブジェクトに対応して ROI を移動するためには、すべての画像でオブジェクトを検出する必要があります。画像からオブジェクトを検出することは、基準座標系関数を使用して、画像内でのオブジェクトの x, y 位置と向きを求めるといことです。座標基準を構築するには、エッジ検出またはパターンマッチングを使用します。

特徴を検出する

オブジェクトを測定するには、測定のベースとする目印またはオブジェクトの特徴を検出する必要があります。たいていのアプリケーションでは、画像内で検出されたポイントまたはその検出ポイントの幾何学フィットに基づいて測定を行うことができます。測定に利用できるオブジェクトの特徴は、2 つのカテゴリに分けられます。

- エッジ検出メソッドによって検出されたオブジェクトの境界線上にあるエッジポイント
- パターンマッチングによって検出されたオブジェクト内の形状またはパターン

測定を行う

画像内で検出した特徴から、さまざまなタイプの測定を行うことができます。一般的な測定としては、ポイント間の距離、3 つまたは 4 つのポイントで形成された 2 つのライン間の角度、ライン、円形、または楕円最適フィット、検出したポイントに適合する円、楕円、多角形などの幾何学形状の面積などがあります。実行できる測定のタイプに関する詳細は、『IMAQ Vision for LabVIEW User Manual』または『IMAQ Vision for Measurement Studio User Manual』を参照してください。

測定値を判定する

ゲー징アプリケーションの最後のステップでは、画像から得られた測定値に基づいて、部品の品質を判断します。部品の品質は、相対的比較または絶対的比較によって判断します。

多くのアプリケーションでは、検査画像から得られた測定値を、標準仕様または基準画像から得られた同じ測定値と比較します。すべての測定は部品の画像で行うため、直接比較することが可能です。

それ以外のアプリケーションでは、画像から得られた寸法測定値を実単位で指定した値と比較します。その場合は、第 3 章、「[システムのセットアップとキャリブレーション](#)」で説明したキャリブレーションツールを使用して、画像からの測定値を実世界単位に変換します。

座標系

一般的なマシンビジョンアプリケーションでは、測定値は画像全体ではなく ROI から抽出されます。ROI から測定値を抽出するためには、検査対象のオブジェクトが定義済みの ROI に常に表示されている必要があります。

検査対象のオブジェクトの位置と向きが検査画像中で常に同じ場合は、各検査画像からオブジェクトを検出することなく、直接測定を開始することができます。

ただしほとんどの場合、固定検索領域を使用できるほど、検査対象のオブジェクトがカメラの視野の一定の位置にあることはありません。オブジェクトが画像内で移動したり回転したりしている場合は、検索領域もオブジェクトとともに移動および回転させる必要があります。検索領域は座標系に対して定義されます。座標系は、基準点（原点）と基準角度またはその 2 つの軸を形成するラインによって定義されます。

使用目的

ゲー징アプリケーションで座標系を使用するのは、すべての検査画像でオブジェクトが同じ位置に表示されない場合です。また、座標系を使用して、オブジェクトの検索領域をオブジェクトの画像内での位置に対して定義することもできます。

概念

すべての測定値は、座標系に対して定義されます。座標系は、検査対象のオブジェクトの特徴部分に基づくもので、測定の基準として使用されます。オブジェクトを検査する際は、まず検査画像の基準となる特徴を検出します。オブジェクト上で、ソフトウェアがすべての画像で確実に検出可能な特徴を選択します。欠陥部品の画像でその特徴が検出不可能になるような、製造欠陥の影響を受ける特徴を選ばないようにしてください。

特徴を囲むように ROI を指定することにより、ソフトウェアがその特徴を検索する領域を画像内で限定することができます。特徴が検出されると予測される ROI を定義すると、その特徴が画像内の複数の領域で見られる場合に、不一致を防止することができます。ROI を小さくすると、さらに検出速度が向上します。

座標系を定義して、新しい座標系に基づいて測定を行うには、次の一般的な手順に従ってください。

1. 基準座標系を定義します。
 - a. 基準となる特徴、または指定した座標系が基本とする特徴を囲むように、検索領域を定義します。すべての検査画像で、検索領域が特徴を囲むようになっていることを確認してください。
 - b. 検査対象のオブジェクトの基準となる特徴として、見つけやすいものを選びます。その特徴が、基準画像の基準座標系の基本となります。特徴を検出するには、エッジ検出とパターンマッチングの 2 つの主要なテクニックを使用することができます。
 - c. ソフトウェアは、座標系を構築して、画像内でのオブジェクトの位置と向きを追跡します。
2. 基準画像内で測定を行う測定領域を設定します。
3. 検査または測定するオブジェクトの画像を集録します。
4. 座標系を更新します。このステップで、IMAQ Vision は検索領域から特徴を検出し、特徴の新しい位置に基づいて新しい座標系を構築します。
5. 測定を行います。
 - a. IMAQ Vision は、基準座標系と新しい座標系の違いを計算します。この違いに基づき、ソフトウェアは新しい測定領域を新しい座標系に対応して移動します。
 - b. 更新された測定領域内で測定を行います。

図 13-1a は、定義された基準座標系による基準画像を示します。図 13-1b は、更新された座標系による検査画像を示します。

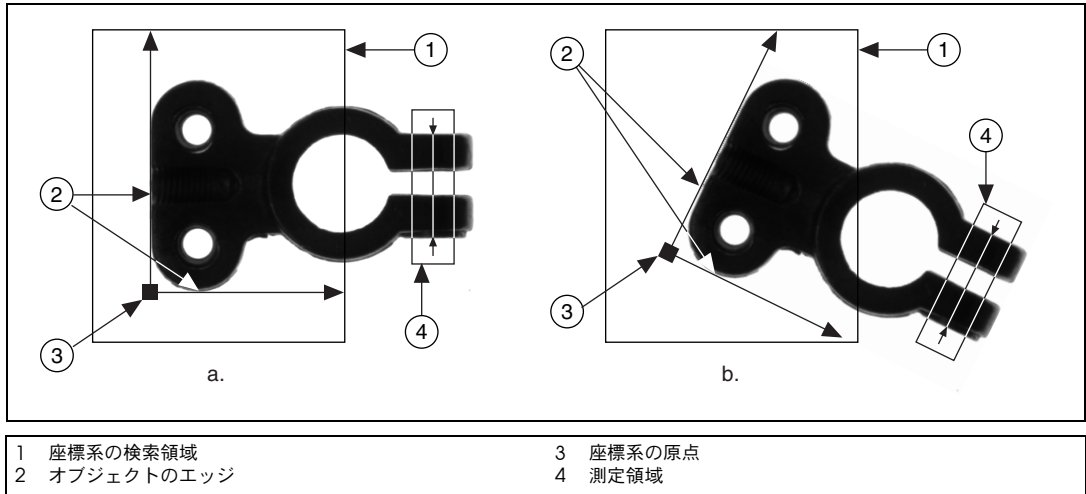


図 13-1 基準画像と検査画像の座標系

詳細の説明

座標系を構築するには、4つの異なる方法を使用することができます。そのうちの2つは、検査対象のオブジェクトの基準エッジの検出が基本となっています。その他の2つでは、パターンマッチングアルゴリズムを使用して特定のパターンを検出します。

エッジベースの座標系関数

これらの関数は、検査対象の部品のエッジを検出することによって、座標系の軸を決定します。検出するオブジェクト上に2つの平行しない明確な直線のエッジが識別される場合は、エッジベースの方法を使用します。ソフトウェアはそれらのエッジを座標系作成のための基準として使用するため、はっきりしていて、検査対象のオブジェクトに常に表示されているエッジを選択してください。

単一検索領域

この方法では、エッジ検出アルゴリズムに基づいて、単一の検索領域で座標系の2つの軸—主軸と副軸—を検出します。まず、図 13-2a に示すように、関数は座標系の主軸を決定します。レーキ関数によって複数の検索ラインとオブジェクトのエッジ間の交差ピクセルを検出します。これらの検索ラインに沿って、検索方向を指定することができます。交差点は、コントラスト、幅、および傾斜によって決定します。エッジ検出に関する詳細は、第 11 章、「[エッジ検出](#)」を参照してください。交差点

を通過してフィットされたラインが主軸を定義します。次に、図 13-2b に示すように、同じ検索領域内で副軸を検索します。ソフトウェアは、主軸に平行する複数の平行ラインを使用してエッジをスキャンしてから、検索領域に最も近く、主軸に対して垂直のオブジェクトのエッジを通過してラインをフィットします。このラインが、座標系の副軸になります。副軸は、主軸に対して平行であってはなりません。主軸と副軸の交差点が、基準座標系の原点となります。

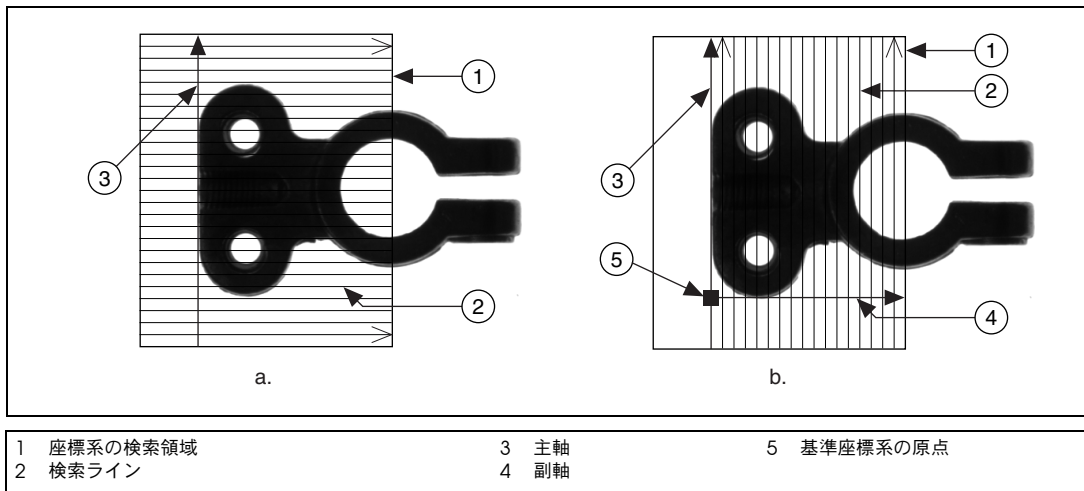


図 13-2 単一の検索領域の座標系を定義する

2つの検索領域

この方法では、単一検索領域の場合と同じ操作モードを使用します。ただし、座標系の軸を定義する2つのエッジは、2つの別々の検索領域にあります。

関数は、まず座標系の主軸の位置を決定します。第一検索領域にある一組の平行検索ラインとオブジェクトの明確な直線エッジの間の交差点を検出します。交差点は、コントラスト、幅、および傾斜によって決定します。エッジ検出に関する詳細は、第 11 章、「エッジ検出」を参照してください。交差点を通過してフィットされたラインが主軸を定義します。この処理を第二検索領域で垂直方向に繰り返して、副軸を検出します。主軸と副軸の交差点が、座標系の原点となります。

図 13-3a は、定義された基準座標系による基準画像を示します。

図 13-3b は、更新された座標系による検査画像を示します。

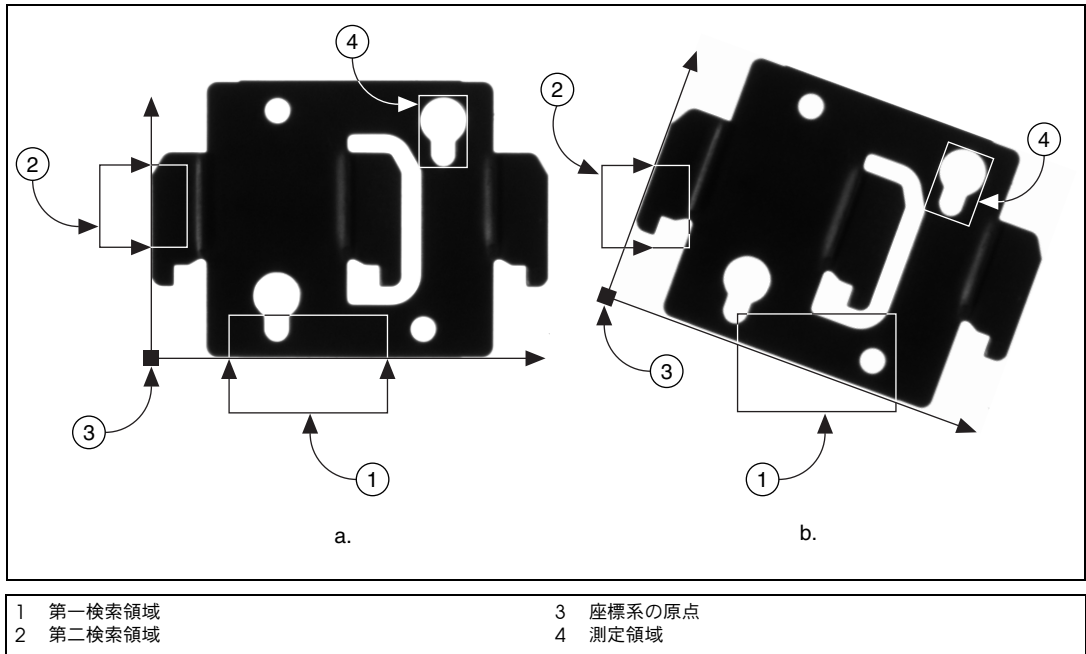


図 13-3 2つの検索領域の座標系を検出する

パターンマッチングベースの座標系関数

パターンマッチングテクニックを使用して基準となる特徴を検出する方法は、画像内に明確な直線エッジが見当たらない場合に、エッジ検出に代わるよい方法です。基準となる特徴、つまりテンプレートは、座標系の基本となるものです。

ソフトウェアは、基準画像内の四角形の検索領域でテンプレート画像を検索します。検出したテンプレートの位置と向きを使用して、座標系の基準位置を作成したり、既存の座標系の現在の位置と向きを更新したりします。

エッジ検出テクニックで必要とされる特徴の安定性や信頼性は、パターンマッチングでも同様に求められます。パターンマッチングでは、シフトに影響されないパターンマッチングと回転に影響されないパターンマッチングの2つの方法のいずれかを使用します。シフトに影響されないパターンマッチングでは、ROIまたは画像全体から、最大 $\pm 5^\circ$ の許容回転角度でテンプレートを検出します。回転に影響されないパターンマッチングでは、テンプレートの向きが変わっても (0° から 360°)、画像内のテン

プレートを検出します。テンプレートとして使用するのに推奨されるパターンのタイプについては、第 12 章、「パターンマッチング」を参照してください。

図 13-4 は、シフトに影響されないパターンマッチングを使用した座標系の検出方法を示します。図 13-4a は、定義された基準座標系による基準画像を示します。図 13-4b は、更新された座標系による検査画像を示します。

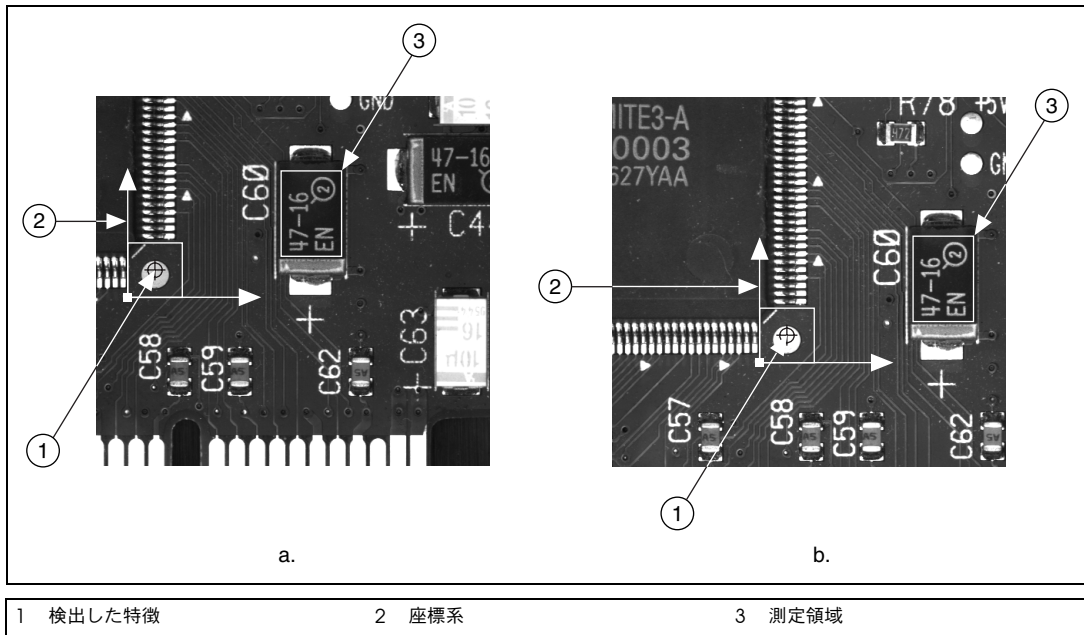


図 13-4 シフトに影響されないパターンマッチングによって座標系を検出する

図 13-5 は、回転に影響されないパターンマッチングを使用した座標系の検出方法を示します。図 13-5a は、定義された基準座標系による基準画像を示します。図 13-5b は、更新された座標系による検査画像を示します。

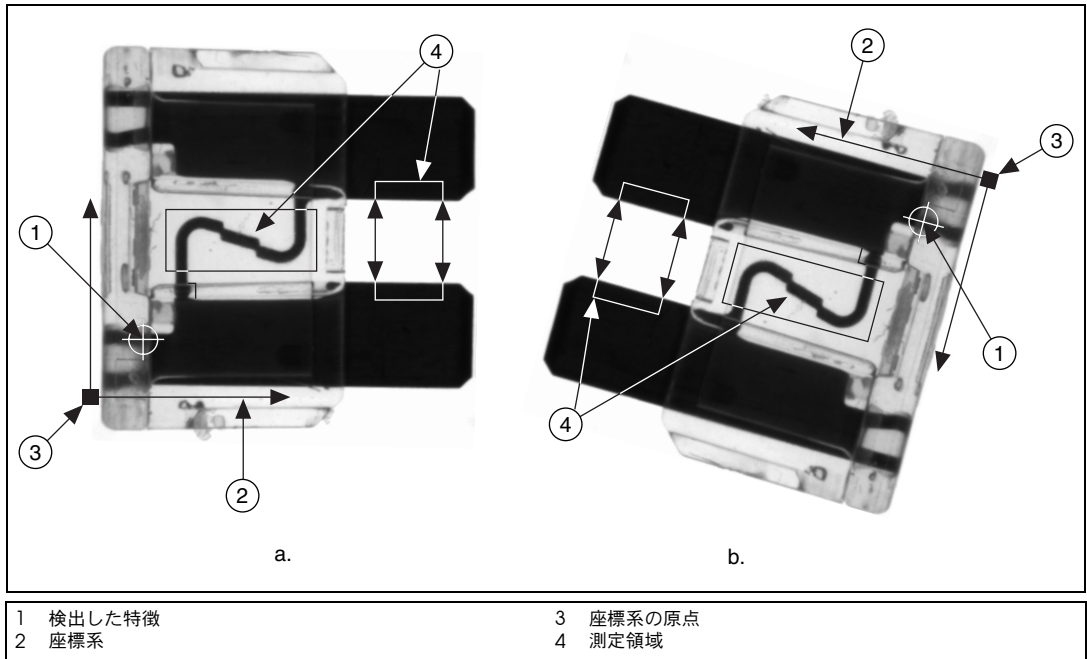


図 13-5 回転に影響されないパターンマッチングによって座標系を検出する

特徴または測定ポイントを検出する

測定を行う前に、測定に使用できる特徴を検出する必要があります。画像上でそのような特徴を検出するには、多くの方法があります。測定に使用する特徴としては、測定する部品の境界線上のポイントが最も一般的です。

エッジベースの特徴

第 11 章、「[エッジ検出](#)」で説明したエッジ検出テクニックを使用して、二次元検索領域内に定義した単一または複数の検索形状上でエッジポイントを検出します。

ラインおよび円形の特徴

IMAQ Vision のライン検出関数を使用して、垂直または水平方向のラインを検出します。それらの関数では、レーキおよび同心レーキ関数を使用して、オブジェクトのエッジ上のポイントを検出し、そのエッジ上でラインをフィットします。

レーキおよび同心レーキ関数の詳細については、第 11 章、「エッジ検出」を参照してください。ラインフィットの方法は、本章で後ほど説明します。図 13-6 は、レーキによって直線エッジを検出する方法を示します。

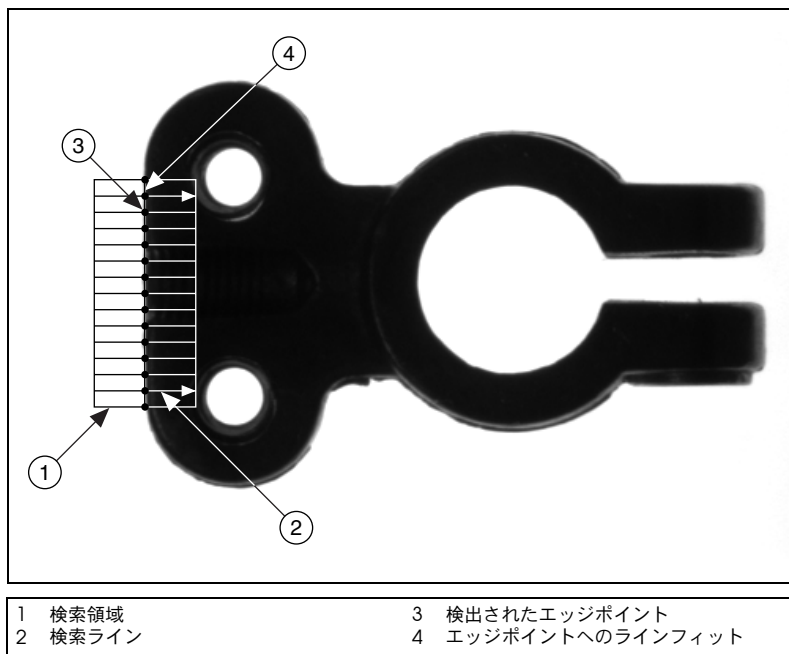


図 13-6 直線の特徴を検出する

円形エッジを検出するには、円形検出関数を使用します。この関数は、スポークを使用して円形エッジ上のポイントを検出し、検出したポイント上で円形をフィットします。図 13-7 は、スポークによって円形エッジを検出する方法を示します。

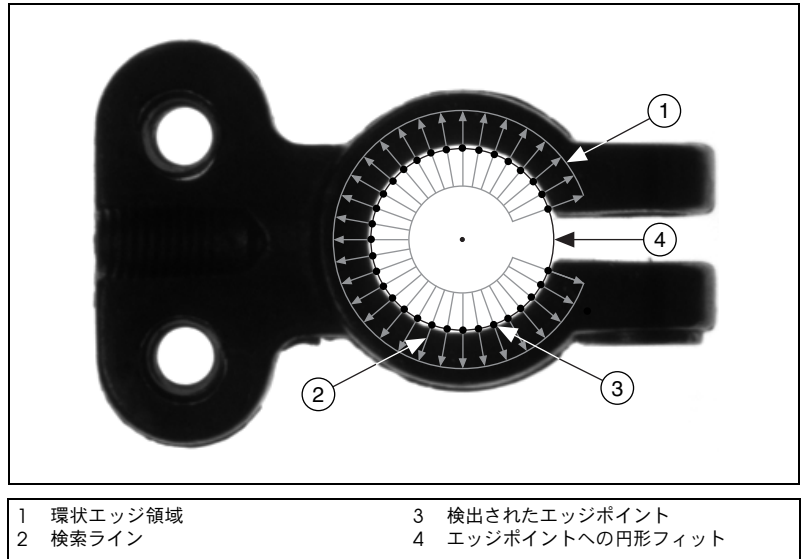


図 13-7 円形検出

形状ベースの特徴

検出する特徴が、部品の境界線より、形状やグレースケールまたはカラーの内容によってよりよく表現できる場合は、パターンマッチングまたはカラーパターンマッチングを使用します。

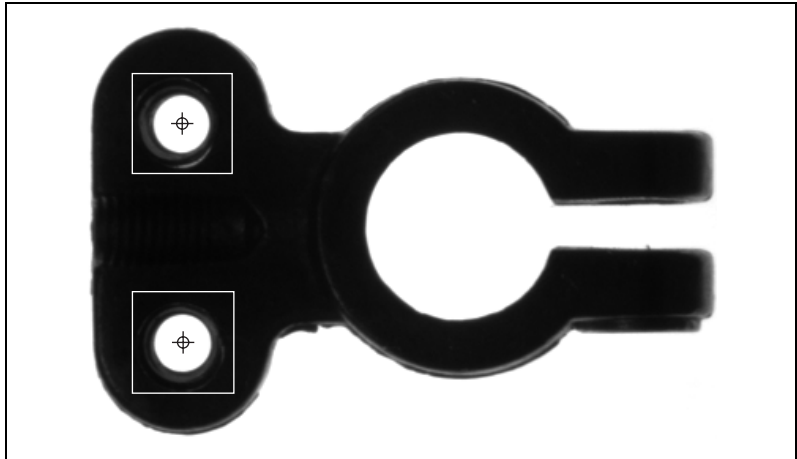


図 13-8 形状の特徴を検出する

画像上で測定を行う

画像内でポイントを検出したら、それらのポイントを基にして距離測定または幾何学測定を行うことができます。

距離測定

距離測定を行うには、以下のいずれかの方法を使用します。

- いずれかの特徴検出メソッドを使用して検出したポイント間の距離を測定。
- IMAQ Vision で利用できるクランプ関数を使用して、オブジェクトの 2 つのエッジ間の距離を測定。クランプ関数は、レーキ関数を使用して、四角形領域内の 2 つのエッジ間の距離を測定します。まず、クランプ関数で、レーキ関数を使用して 2 つのエッジ上のポイントを検出します。次に、検出したポイント間の距離を計算し、最長距離と最短距離を返します。クランプ関数を使用して以下を行います。
 - 2 つの垂直方向のエッジ間の最長または最短水平距離を検出します。
 - 2 つの水平方向のエッジ間の最長または最短垂直距離を検出します。

図 13-9 は、クランプ関数でオブジェクトのエッジ間の最短距離を検出する方法を示します。

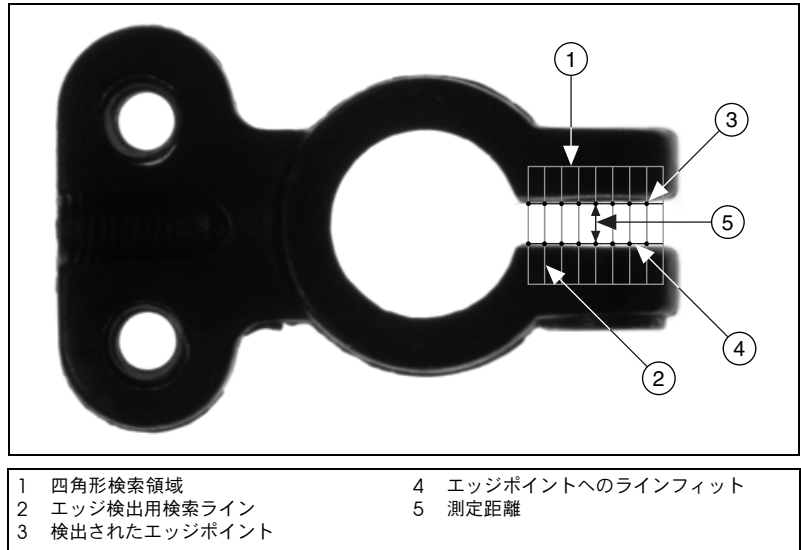


図 13-9 クランプ関数

解析幾何学

画像内で検出した特徴ポイントから、以下の幾何学測定を行うことができます。

- 頂点によって指定された多角形の面積
- ポイントセットにフィットされたラインとそのラインの方程式
- ポイントセットにフィットされた円と、その面積、周囲、および半径
- ポイントセットにフィットされた楕円と、その面積、周囲、および長軸と短軸の長さ
- 開始ポイントと終了ポイントによって指定された 2 つのラインの交差ポイント
- 2 つのラインによって形成された角度を 2 等分するライン
- ポイントと、そのラインに平行するライン間の中間のライン
- ポイントとライン間の垂直距離を算出する垂直ライン

ラインフィット

IMAQ Vision のラインフィット関数では、確実なアルゴリズムを使用してポイントセットに最適フィットするラインを検出します。ラインフィット関数は、特にゲーjingアプリケーションで得られた特徴ポイントに対し動作するようになっています。

一般的なゲーjingアプリケーションでは、レーキまたは同心レーキ関数はオブジェクトの直線エッジ上にあるポイントセットを検出します。理想的なケースでは、すべての検出ポイントにより直線が形成されます。ただし、以下のいずれかの理由で、通常検出されたポイントは直線として表示されません。

- オブジェクトのエッジは、レーキで使用した検索領域全体を占めていない。
- オブジェクトのエッジは連続的な直線ではない。
- 画像内のノイズにより、エッジ上のポイントが実際の位置から動いてしまう。

図 13-10 は、レーキ関数によって検出されたポイントセットの例を示します。図に示すように、すべてのポイントを使用してラインフィットする一般的なラインフィットアルゴリズムは、不正確な結果を返します。IMAQ Vision のラインフィットアルゴリズムは、データセット内の範囲外にあるポイントを補正して、より正確な結果を返します。

IMAQ Vision では、以下のプロセスでラインをフィットします。IMAQ Vision では、ポイントがフィットしたラインからユーザ指定距離内（ピクセル半径）にある場合は、ポイントをラインの一部とみなします。そして、ラインフィットアルゴリズムは、ほぼ直線のライン上にあるポイントのサブセットにラインをフィットします。IMAQ Vision は、平均二乗距離（MSD）を測定することによって、ラインフィットの質を確認します。平均二乗距離とは、各ポイントと予測ラインとの間の二乗距離を平均したものです。図 13-11 は、MSD の計算方法を示します。次に、ラインフィット関数は元のセットからポイントのサブセットを削除します。IMAQ Vision は、すべてのポイントがフィットされるまで、このステップを繰り返します。次に、ラインフィットアルゴリズムは、MSD の最も低いラインを検出します。これは最も質の高いラインフィットを示します。そして、ユーザ定義の最小スコアが得られるか、あるいはユーザ定義の最大反復数を超えるまで、ラインから最も遠くにあるポイントの削除を連続的に行って、ラインの質を向上させます。

ラインフィット関数を実行した結果として、図 13-12 に示すように、範囲外にあるポイントが無視し、最も近いポイントのサブセットにフィットされたラインが得られます。

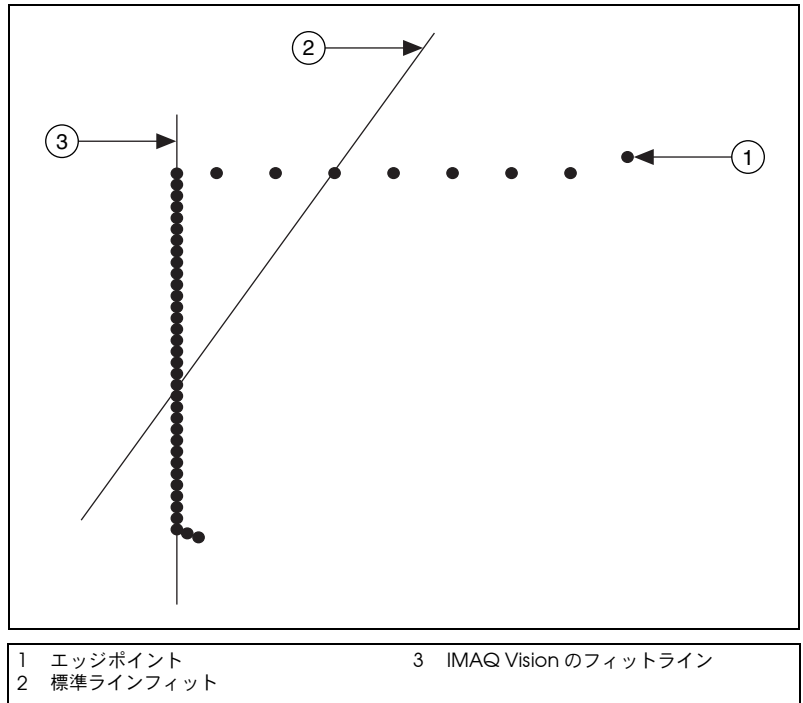


図 13-10 2つの方法を使用したデータセットとフィットライン

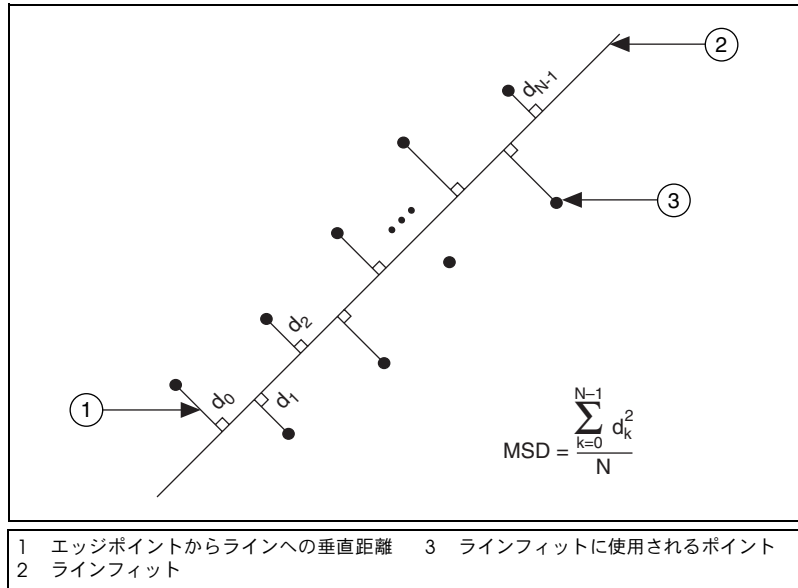


図 13-11 平均二乗距離 (MSD) の計算

ピクセル半径、最小スコア、および最大反復数の各パラメータにより、ラインフィット関数の動作を制御します。

ピクセル半径は、有効ポイントと予測ライン間に許容される最長距離をピクセル単位で定義したものです。アルゴリズムは、少なくともポイントセットの半分がピクセル半径内にあるラインを予測します。ポイントセットにそのようなラインがない場合、関数は有効ポイントの最も多いラインを返すよう試みます。

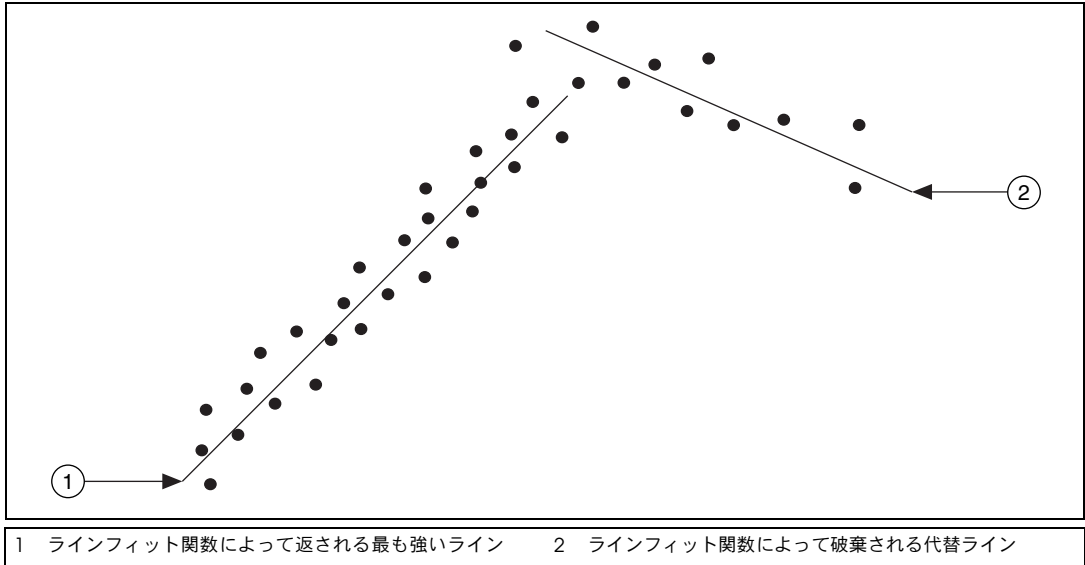


図 13-12 最も強いラインフィット

ピクセル半径を大きくすると、ポイントと予測ラインの間の許容距離も長くなります。一般に、イメージングシステムの解像度とシステム内のノイズ量で、このパラメータを測定することができます。イメージングシステムの解像度が非常に高い場合は、小さいピクセル半径を使用して、範囲外のポイントがラインフィットに考慮されるのを最小限に抑えます。画像にノイズが多い場合は、大きいピクセル半径を使用します。

最小スコアにより、予測ラインの質を向上させることができます。ラインフィット関数は、フィットラインから最も遠いポイントを削除し、それから残りのポイントでラインを再フィットして、そのラインの MSD を計算します。次に、関数は次の式を使用して、新しいフィットのラインフィットスコア (LFS) を計算します。

$$LFS = \left(\frac{1 - MSD}{PR^2} \right) \times 1000$$

ここで PR はピクセル半径です。

IMAQ Vision は、スコアが最小スコア以上になるか、反復数がユーザ指定の最大反復数を超えるまで、全体のプロセスを繰り返します。

最も正確なラインフィットを得るには、最小スコアを高く設定します。たとえば、大きいピクセル半径と高い最小スコアを組み合わせると、ノイズの多いデータセットで正確なフィットが生成されます。小さいピクセル半径と低い最小スコアでは、標準的なデータセットで確実なフィットが得られます。

最大反復数は、最小スコアに達するラインを検索する際のリミットを定義します。アルゴリズムが指定の最小スコアを満たすラインを検索する前に、最大反復数に達した場合、アルゴリズムは停止して現在のラインを返します。希望の結果を得るためにラインの質を向上させる必要がない場合は、ラインフィット関数の最大反復数の値を 0 に設定します。

カラー検査

本章では、カラースペクトル、カラーマッチング、カラーの位置検出、およびカラーパターンマッチングについて説明します。

カラースペクトル

カラースペクトルは、画像または画像の領域に関連する三次元カラー情報を、IMAQ Vision の多くのカラー処理ソフトウェアで使用できる簡潔な一次元形式で表したものです。カラースペクトルは、IMAQ Vision のカラーマッチング、カラーの位置検出、およびカラーパターンマッチングアプリケーションで使用します。

カラースペクトルは、画像の三次元カラー情報を一次元で表します。スペクトルは、画像または画像の領域に関連するすべてのカラー情報を HSL スペースで表現します。情報は IMAQ Vision のカラー処理関数で使用可能な形式に変換されます。

スペクトルの生成に使用するカラースペース

カラースペクトルは、図 14-1 に示すように、画像のカラー分布を HSL スペースで表します。入力画像が RGB 形式の場合、画像はまず HSL 形式に変換され、HSL スペースからカラースペクトルが計算されます。たとえば、IMAQ PCI/PXI-1411 画像集録デバイスを使用し、カラーマッチング用にオンボードで RGB から HSL 変換を行って集録した画像などのような HSL 画像を直接使用すると、処理速度が向上します。

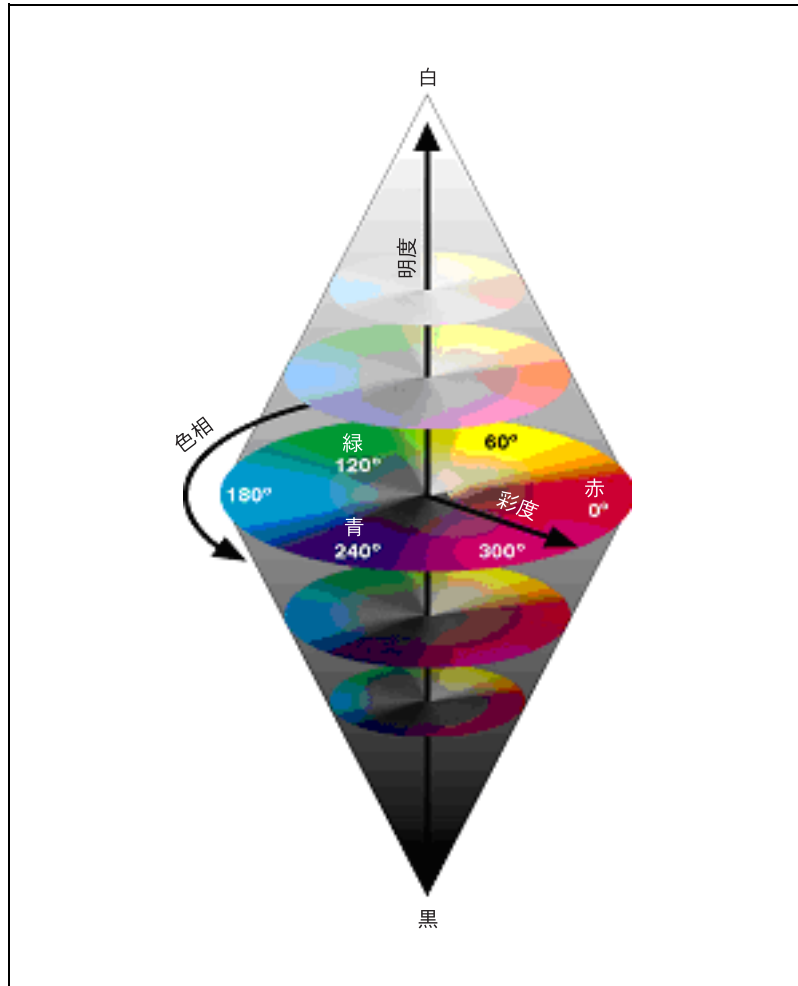


図 14-1 HSL カラースペース

色を HSL モデルスペースで表すと、色の定量化がしやすくなります。HSL スペースの明度（輝度）コンポーネントは、カラー情報から分離されます。この機能により、輝度変化に影響されずにより確実なカラー表示を得ることができます。ただし、色度（色相と彩度）プレーンでは、多くのマシビジョンアプリケーションで背景色として使われる傾向のある白と黒を表すことができません。カラースペースの詳細については、第 1 章、「[デジタル画像](#)」を参照してください。

カラースペクトルを生成する

カラースペクトル配列の各要素は、HSL スペースのカラービンに対応します。配列の最後の 2 つの要素が黒と白を表します。図 14-2 は、HSL カラー空間をビンに分割する方法を示します。色相空間は、いくつかの同じ大きさのセクターに分けられ、さらにそのセクターが高彩度値および低彩度値を表す 2 つの部分に分けられます。それぞれの部分がカラービン (カラースペクトル配列の要素) に対応します。

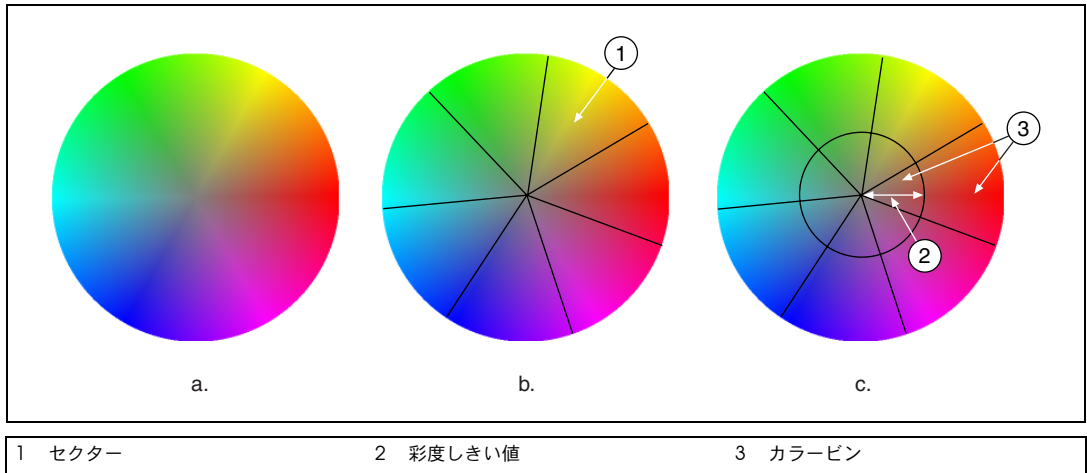


図 14-2 ビンおよびセクターに分割した HSL スペース

カラー感度パラメータは、色相空間を分割するセクター数を指定します。図 14-2a は、輝度が 128 のときの色相カラースペースを示します。図 14-2b は、指定したカラー感度にしたがって、いくつかのセクターに分割した色相空間を示します。図 14-2c は、各セクターをさらに高彩度ビンと低彩度ビンに分割したものです。彩度しきい値は、各セクターをビンに分離する内円の半径を指定します。

図 14-3 は、カラースペクトル要素とカラースペースのピンの間の対応を示すものです。カラースペクトル配列の最初の要素は最初のセクターの高彩度部分、2 つめの要素は低彩度部分、そして 3 つめの要素は 2 つめのセクターの高彩度部分を表す、というようになります。カラースペースに n 個のピンがある場合、カラースペクトル配列には $n + 2$ 個の要素が含まれます。カラースペクトルの最後の 2 つの成分は、それぞれ黒と白を表します。

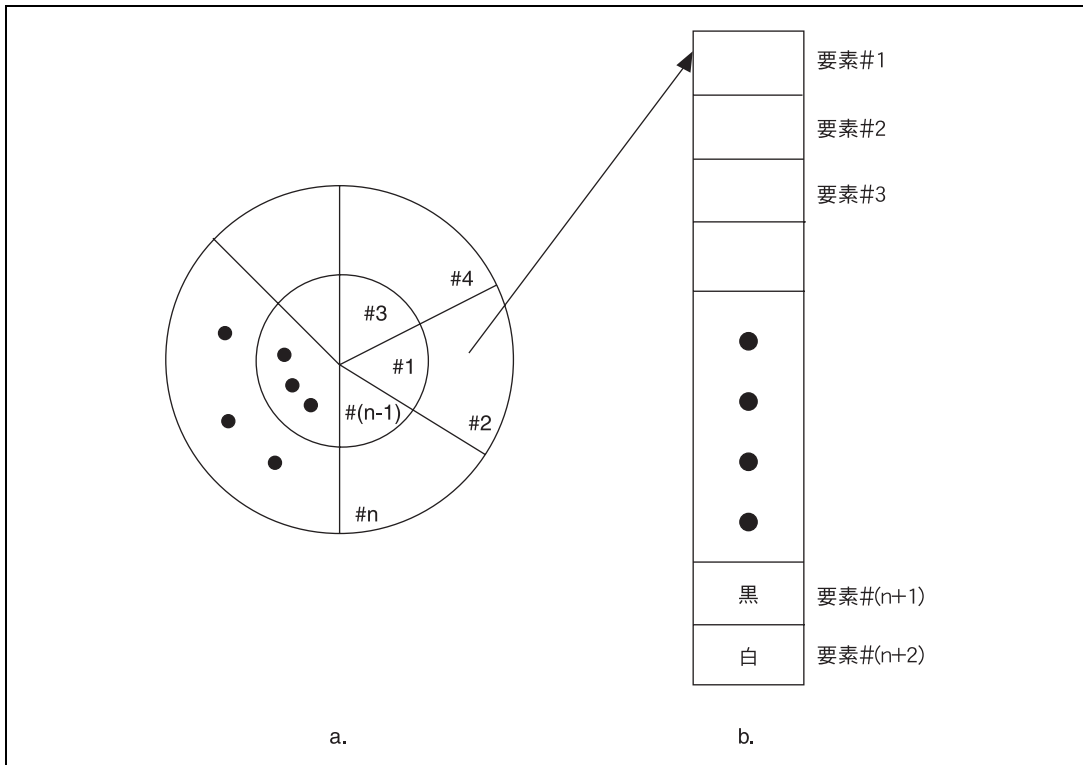


図 14-3 色相カラースペースとカラースペクトル配列の関係

ピン（要素）数の多いカラースペクトルは、ピン数の少ないスペクトルに比べ、高カラー解像度など、詳細を多く含む画像のカラー情報を表します。IMAQ Vision では、低、中、高の 3 つのカラー感度設定から選択できます。低は、色相カラースペースを 7 つのセクターに分け、ピンは全部で $2 \times 7 + 2 = 16$ 個になります。中は、色相カラースペースを 14 のセクターに分け、ピンは全部で $2 \times 14 + 2 = 30$ 個になります。高は、色相カラースペースを 28 のセクターに分け、ピンは全部で $2 \times 28 + 2 = 58$ 個になります。

カラースペクトルの各要素の値は、各カラーピンにおける画像ピクセルの割合を示します。ピン数がカラー感度パラメータにしたがって設定されると、マシンビジョンソフトウェアは画像をスキャンし、各ピンに当てはまるピクセル数をカウントして、カウントと画像内の総ピクセル数の比率をカラースペクトル配列の該当する要素に格納します。

ソフトウェアはまた、特殊な適応学習アルゴリズムを適用して、ピクセルをカラーピンに割り当てる前に、そのピクセルが黒か白かを判別します。図 14-4b は、図 14-4a の低感度カラースペクトルを示します。各棒の高さは、対応するピンに当てはまる画像内のピクセルの割合に対応します。

カラースペクトルには、画像のカラー分布に関する有益な情報が含まれています。カラースペクトルを解析して、画像の主調色などの情報を得ることができます。主調色はカラースペクトルでは最高値を持つ要素です。また、カラースペクトルの配列を、カラー分布の直接解析に使用したり、カラーマッチングアプリケーションに使用したりできます。

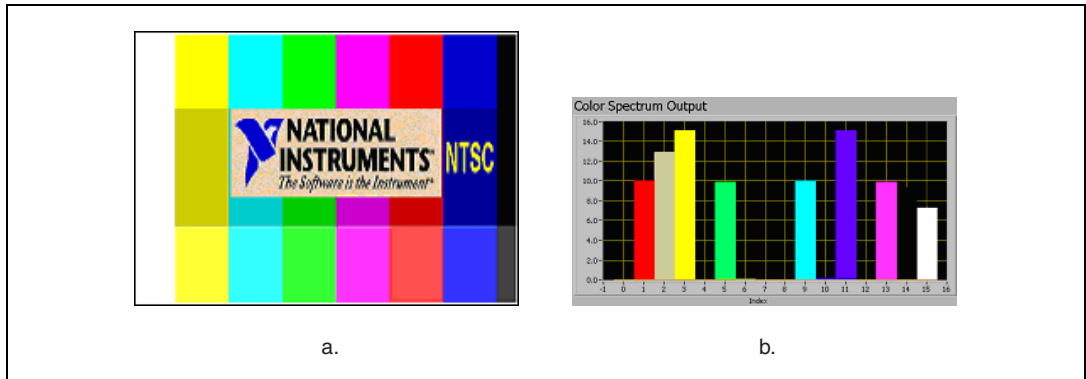


図 14-4 画像に関連するカラースペクトル

カラーマッチング

カラーマッチングは、画像内にどの色がどの程度の量存在するかを数量化し、その情報を使用して他の画像に同じ色が同じ比率で含まれているかどうかを確認します。

カラーマッチングを使用して、画像または画像内の一部の領域のカラー内容を、基準カラー情報と比較することができます。カラーマッチングにより、基準として使用するカラー情報を含む画像を作成したり、画像内の領域を選択したりします。画像内のカラー情報は、1つまたは複数のカラーからなります。マシンビジョンソフトウェアは、画像の三次元カラー情報を学習して、それを一次元のカラースペクトルに表します。マシンビジョンアプリケーションは画像全体または画像の一部の領域のカラー情報を、学習したカラースペクトルと比較して、各領域に対しスコアを計算します。そのスコアは、画像領域のカラー情報が、カラースペクトルに表示された情報にどれほど一致しているかを表すものです。

使用目的

カラーマッチングは、カラー識別、カラー検査、カラーオブジェクトの位置、およびカラー情報の比較によって決定を行う他のアプリケーションに使用することができます。

カラー識別

カラー識別では、オブジェクトの画像のカラー情報を、定義済みのオブジェクトタイプに対応する基準色のデータベースと比較して、オブジェクトを識別します。オブジェクトには、データベースの基準色の中で最も近いオブジェクトのタイプに対応するラベルが割り当てられます。まず、カラーマッチングを使用して、定義済みのすべてのオブジェクトタイプのカラー情報を学習します。定義済みの各オブジェクトタイプに関連するカラースペクトルが基準色になります。マシンビジョンアプリケーションは、カラーマッチングを使用して、オブジェクトの画像のカラー情報を基準色のカラースペクトルと比較します。オブジェクトには、最も高いマッチスコアを持つカラースペクトルのラベルが付けられます。

図 14-5 は、タイルの識別アプリケーションの例を示します。図 14-5a は、識別するタイルの画像を示します。図 14-5b は、基準タイルを使用したカラーマッチングで得られたスコアを示します。

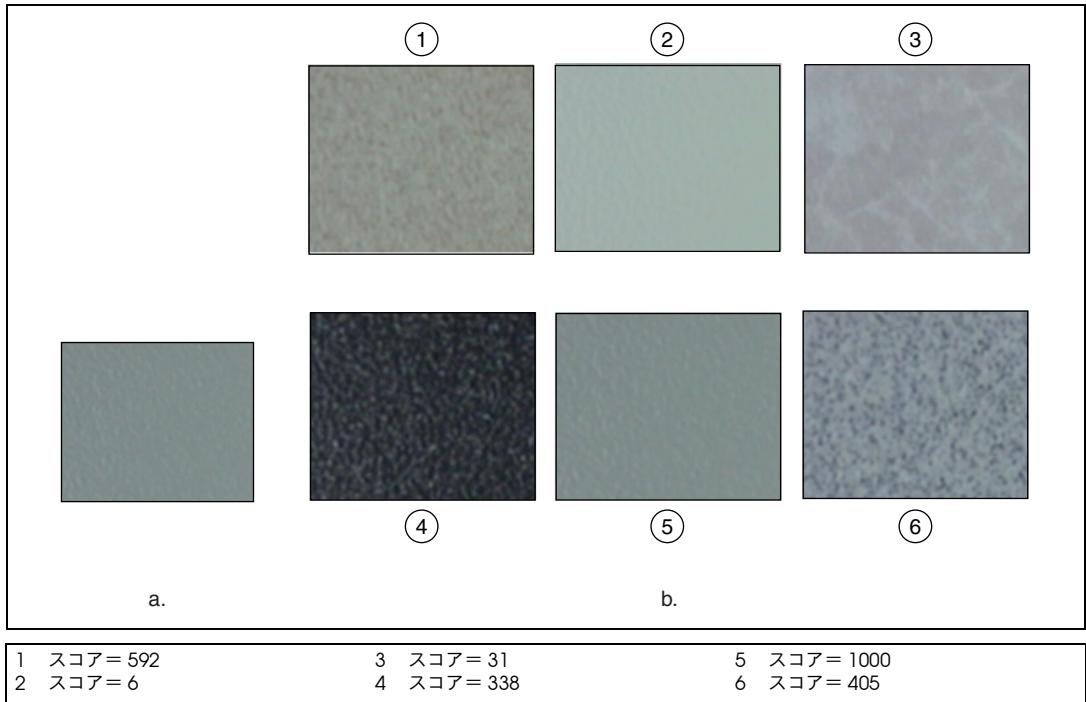


図 14-5 カラーマッチング

カラーマッチングを使用して、自動車組立で正しいコンポーネントが存在していることを確認することもできます。カラー識別タスクの例として、車の内装の布張りの色が仕様どおりであることを確認する作業もあります。

カラー検査

カラー検査は、不在または誤った位置にあるカラーコンポーネント、カラーオブジェクトの表面にある傷やカラーラベルの印刷ミスなど、単純な欠陥を検出します。画像内の検査するオブジェクトまたは部分を、既知の関心領域によってあらかじめ定義する場合に、このようなアプリケーションでカラーマッチングを使用します。そのような領域は、ユーザが定義することもできますが、検査するコンポーネントの検出に使用したパターンマッチングなどのマシンビジョンツールの出力を使用することもできます。

自動車組立のジャンクションボックスでのヒューズの配置は、関心領域によって容易に定義されます。カラーマッチングを使用して、すべてのヒューズがあることと、それらが正しい位置にあることを確認します。図 14-6 は、ヒューズボックスアプリケーションの例を示します。この例では、画像内のヒューズの正確な位置を関心領域によって定義することができます。カラーマッチングにより、各領域のヒューズの色を、その領域にあるべき色と比較します。

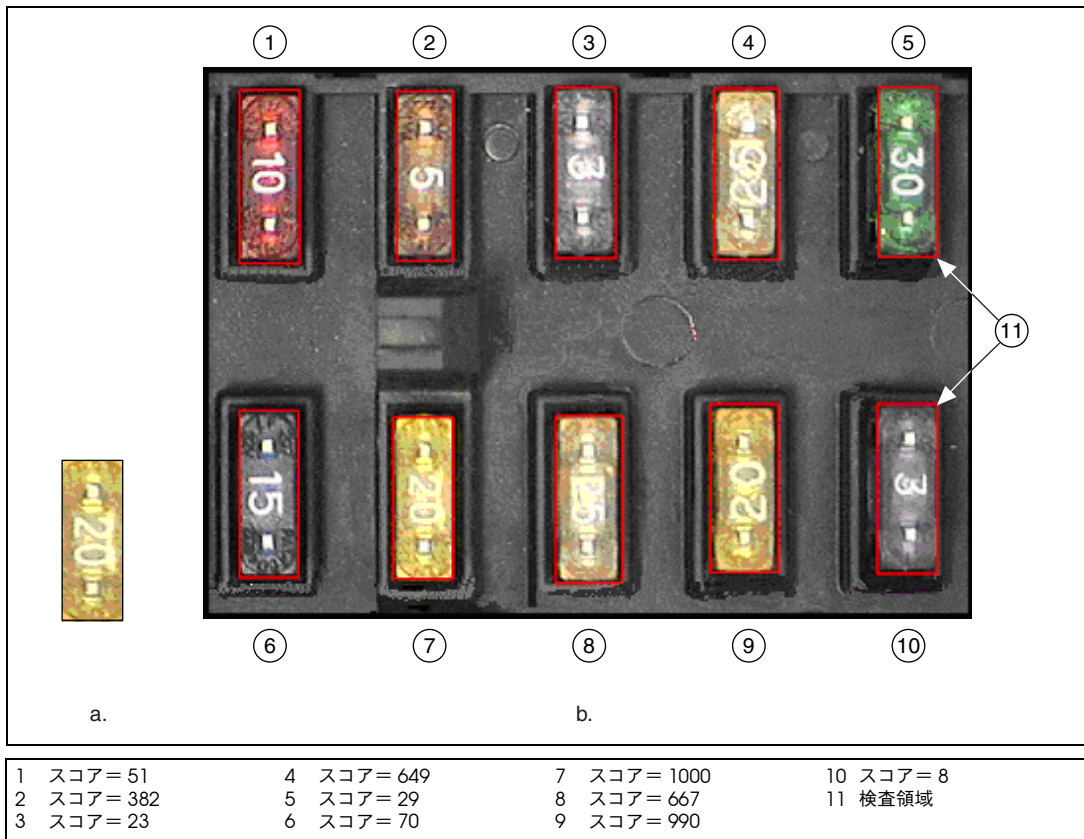


図 14-6 カラーマッチングを使用したヒューズボックス検査

カラーマッチングは、ダイオード、抵抗器、集積回路、コンデンサなどさまざまなコンポーネントを含むプリント基板の検査に使用することができます。製造環境では、欠陥に色の変化が伴う場合、カラーマッチングを使用して製造された製品の欠陥を検出することが可能です。

カラーマッチングの概念

カラーマッチングには 2 つのステップがあります。最初のステップでは、マシンビジョンソフトウェアは基準カラー分布を学習します。次のステップでは、他の画像のカラー情報を基準画像と比較して、類似性を示すスコアを返します。

カラー分布の学習

マシンビジョンソフトウェアは、カラースペクトルを生成して、カラー分布を学習します。ユーザは、アプリケーションで基準として使用するカラー情報を含む画像または画像の領域を提供します。するとマシンビジョンソフトウェアは、ユーザが提供した情報に基づいてカラースペクトルを生成します。カラースペクトルは、マッチングの段階での比較のベースになります。

カラー分布を比較する

マッチング段階では、ターゲット画像またはその領域から得られたカラースペクトルが、学習段階で得られた基準色と比較されます。マッチスコアは、2 つのベクトル間のマンハッタン距離を使用した 2 つのカラースペクトル間の類似点に基づいて計算されます。それらの間の距離を計算する前に、両方のカラースペクトルにファジーメンバーシップ重み関数が適用されます。カラースペースのピン処理の際に発生する可能性のあるエラーは、関数によって補償されます。ファジーカラー比較による方法を使用すると、正確で確実な量的マッチスコアが得られます。マッチスコアは 0 から 1000 までで、カラースペクトルの類似度を定義するものです。スコアが 0 の場合はカラースペクトル間にまったく類似点がなく、1000 の場合は完全な一致を示します。図 14-7 は、比較プロセスを示します。

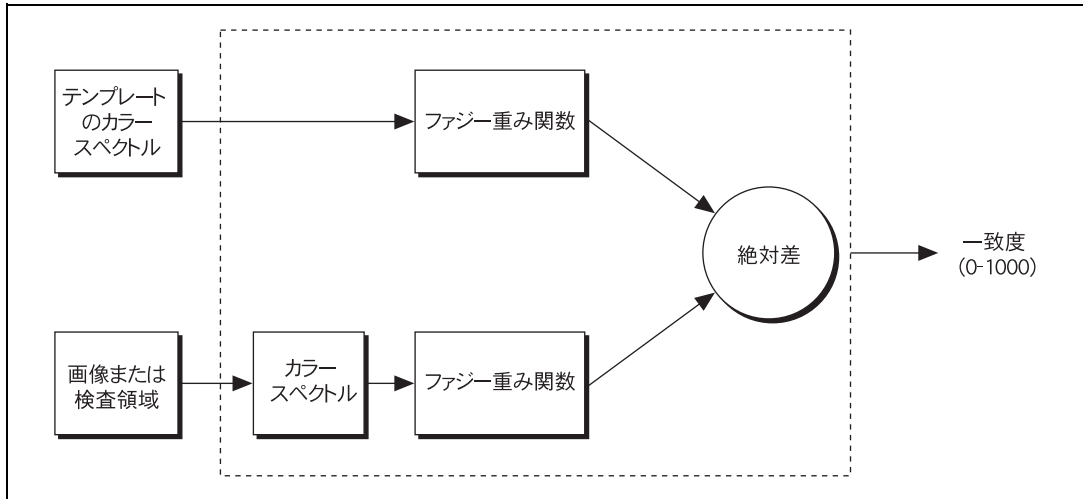


図 14-7 2つのスペクトルを比較して類似度を求める

カラーの位置検出

カラーの位置検出を使用して、画像内の既知のカラー領域をすばやく検出します。カラーの位置検出を実行する際は、検索する色を表すモデルまたはテンプレートを作成します。次に、マシンビジョンアプリケーションが各集録画像からモデルを検索し、各一致項目に対してスコアを計算します。スコアは、モデルのカラー情報が検索された領域のカラー情報にどれほど類似しているかを示します。

使用目的

カラーを使ってコントラストを高めたりオブジェクトを背景から分離することにより、白黒の目視検査での問題点が軽減されます。カラーの位置検出アルゴリズムにより、画像の中で特定の色を持つ領域を簡単に見つけることができます。

カラーの位置検出は、次のようなアプリケーションで使用します。

- 画像内で特定のカラー情報を持つ領域の位置と数が必要なもの。
- 領域内での色の配置ではなく、領域内での累積カラー情報が重要とされるもの。
- 領域の向きを必要としないもの。
- サブピクセル精度での位置情報を必要としないもの。

IMAQ Vision のカラーの位置検出ツールは、特徴を理想的に表したモデル、つまりテンプレートと呼ばれるものと、画像内に含まれる特徴との類似点を測定します。カラーの位置検出で使用する特徴は、画像内で特定の色を持つ領域として定義します。

カラーの位置検出は、多くのアプリケーションで役立つツールです。カラーの位置検出を使用すると、画像内に存在するテンプレートの発生数と位置に関する情報をアプリケーションに提供することができます。カラーの位置検出は、検査、識別、分類などの一般アプリケーションに使用します。

検査

検査は、コンポーネントの欠損、誤った印刷や、布製品上の異なる繊維といった欠陥を検出します。一般的な製薬検査には、薬のパッケージを検査して錠剤が正しいことを確認するアプリケーションがあります。錠剤包装検査は、すべての錠剤が正しい種類であることを確認するもので、それはすべての錠剤のカラー情報が同じであることをチェックすることで簡単に行えます。パッケージ内に正しい錠剤が固定数含まれているかどうかを確認するのが目的であるため、カラーの位置検出は非常に効果的なツールです。

図 14-8a は、検出するカラー情報を含む錠剤の一部分のテンプレート画像を示します。図 14-8b は、正しいパッケージで検出された錠剤を示します。図 14-8c は、パッケージに異なるタイプの錠剤が入っていたり、錠剤が入っていない部分がある画像で検出された錠剤を示します。錠剤の正確な位置情報はこの検査では必要ないため、カラーの位置検出によって返された一致数により、パッケージが検査に合格したかどうかわかります。

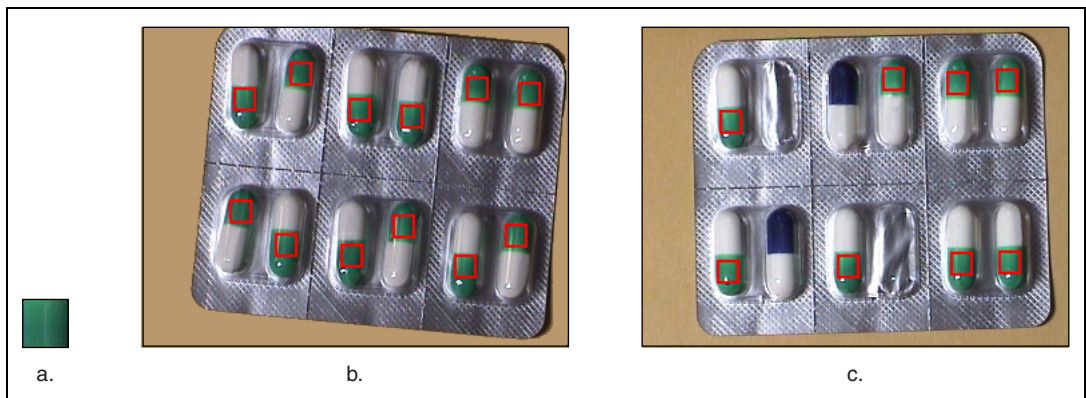


図 14-8 カラーマッチングを使用した錠剤包装検査

識別

識別は、その特徴にしたがってオブジェクトにラベルを付けます。多くのアプリケーションで、カラーコードによる識別マークがオブジェクトに付けられています。そのようなアプリケーションでは、カラーマッチングによりカラーコードを検出してオブジェクトを識別します。スプリング識別アプリケーションでは、コイルに付けられたカラーマークの集まりによって、異なるタイプのスプリングを識別します。スプリングのマーク付けに使用されている異なるタイプのカラーパッチがわかっている場合は、カラーの位置検出により画像に表示されるカラーマークを検出することができます。その後、この情報を使ってスプリングのタイプを識別します。

分類

分類は、オブジェクトを色、サイズ、形状などの属性によって分離します。多くのアプリケーション、特に製薬やプラスチック業界では、錠剤やプラスチックペレットなどのオブジェクトを色によって分類します。図 14-9 は、色の異なるキャンディーを分類する簡単な例を示します。画像内の異なるキャンディーのカラーテンプレートを使用して、カラーの位置検出により異なるキャンディーの位置をすばやく検出します。

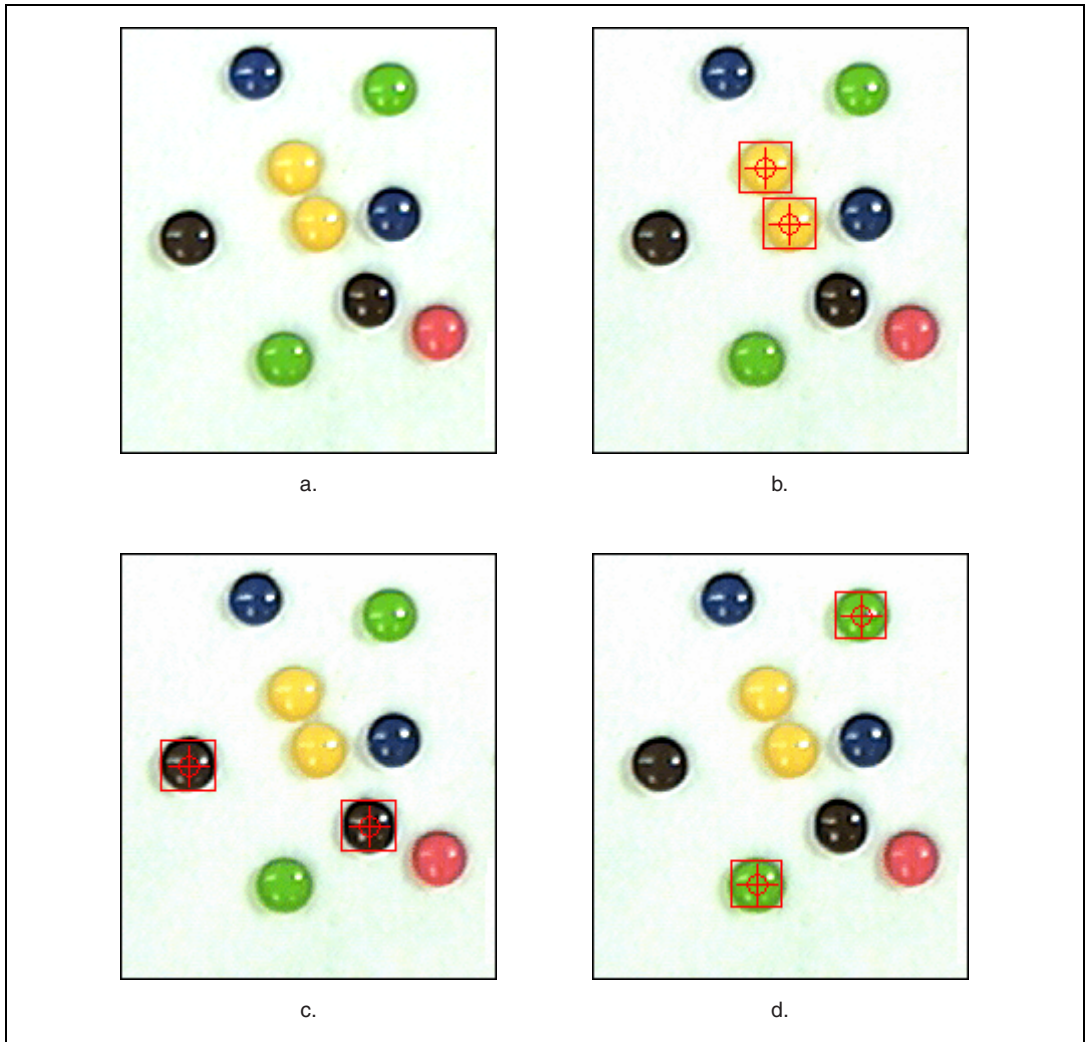


図 14-9 カラー情報によってキャンディーを分類する

カラーの位置検出ツールの機能

自動化されたマシンビジョンアプリケーションでは、検査した物質やコンポーネントは、部品の向きや倍率の変更、照明の変化などにより、外観が変わります。カラーの位置検出ツールを使用すると、そのような変化にかかわらず基準パターンを検出することができます。カラーの位置検出ツールにより、パターンの向きおよび複数のインスタンス、周辺光状態、ブレやノイズの状態などの一般的な状況の中で正確な結果を得ることができます。

パターンの向きおよび複数のインスタンス

カラーの位置検出ツールを使用すると、画像内のパターンが回転していたり倍率が変更されていても、基準パターンを見つけることができます。パターンが画像内で回転したり倍率が変更されている場合、カラーの位置検出ツールは以下を検出します。

- 画像内のパターン
- 画像内におけるパターンの位置
- 画像内に発生するパターンの複数のインスタンス（適用される場合）

カラーの位置検出は、領域内のカラー情報に対してのみ動作し、テンプレートの形状情報については一切考慮しないため、一致オブジェクトの回転角度は検出されません。このツールは、同様のカラー情報を含むテンプレートとサイズが一致する領域の位置のみを検出します。

パターンの向きと複数のインスタンスの例については、図 14-8 を参照してください。図 14-8a は、テンプレート画像を示します。図 14-8b と図 14-8c は、テンプレートがシフトおよび回転した状態を示します。

周辺光状態

カラーの位置検出ツールは、照明が画像全体で均等に变更されている条件下で、画像から基準パターンを検出することができます。また、陰になっている場合など、光が不均等に变化する状況でもパターンを見つけることができます。

図 14-10 は、カラーの位置検出ツールが正しく動作する一般的な状況を示します。図 14-10a は、元のテンプレート画像を示します。図 14-10b は、同じパターンを明るい照明下で表示したものです。図 14-10c は、そのパターンを暗い照明下で表示したものです。

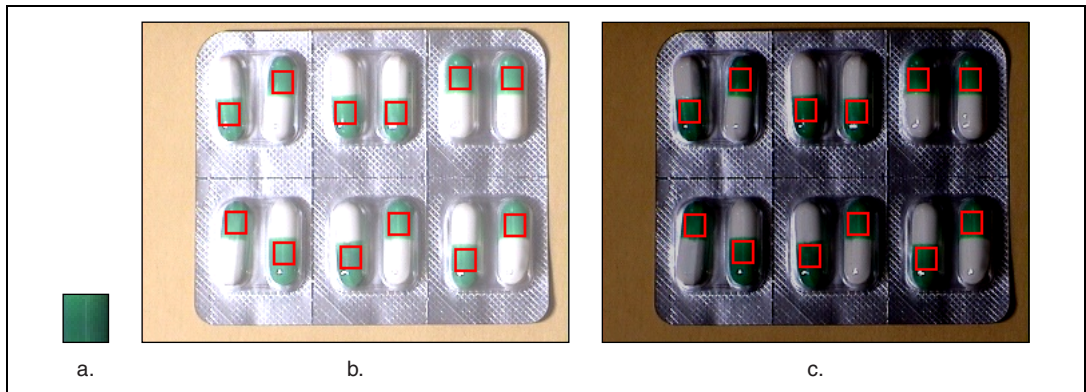


図 14-10 照明状態の例

ブレとノイズの状態

カラーの位置検出では、ブレやノイズによって変形されたパターンを検出することができます。ブレは、通常誤った焦点や被写界深度の変更によって発生します。

カラーの位置検出の概念

カラーの位置検出はカラーマッチングソフトウェアを基に構築され、画像内で特定のカラー情報を持つ領域をすばやく検出します。詳細については、本章の「[カラーマッチング](#)」のセクションを参照してください。

カラーの位置検出ソフトウェアは、カラーマッチングの機能をさらに拡張し、画像内でのオブジェクトの位置が不明なアプリケーションでも使用できるようにしています。カラーの位置検出では、テンプレート画像のカラー情報を使用して、検索画像内に存在するテンプレートを検索します。基本操作では、画像上でテンプレートをピクセル単位で移動させ、カラーマッチングアルゴリズムを使用して、画像の現在の位置でのカラー情報をテンプレートのカラー情報と比較します。

カラーの位置検出のプロセスには、テンプレート情報の学習と画像内でのテンプレートの検索の 2 つの主なステップがあります。図 14-11 は、カラーの位置検出処理の一般的なフローを示します。学習段階では、ソフトウェアはテンプレート画像からカラースペクトルを抽出します。このカラースペクトルを使用して、テンプレートのカラー情報を画像のカラー情報と比較します。

検索ステップで、テンプレートと同じサイズの領域を、画像の上から下に向かってピクセル単位で画像全体に移動させます。各ピクセルで、関数は対象領域のカラースペクトルを計算します。そして、このカラースペクトルをテンプレートのカラースペクトルと比較し、マッチスコアを計算します。

検索ステップは、2 つの段階に分かれています。最初の段階では、粗密検索を行って、マッチスコアが非常に低いものも含めすべての位置を識別します。この段階の目的は、画像内でテンプレート情報と一致する可能性のあるすべての位置をすばやく検出することです。画像内をピクセルごとにステップ処理してマッチスコアを計算するのは時間のかかる作業であるため、検索プロセスの速度を上げるためのテクニックを使用します。次のテクニックを使用すると、すばやい検索ができます。

- サブサンプリング：画像をステップ処理する際、画像内の数カ所のサンプルポイントからカラー情報を取り出し、テンプレートとの比較に使用します。これにより、画像のカラースペクトルの計算に使用するデータ量が削減されるため、検索処理速度が向上します。
- ステップサイズ：画像上でテンプレートをピクセルごとに移動させる代わりに、色の比較のたびに数ピクセルずつスキップするため、検索処理速度が向上します。ステップサイズは、スキップするピクセル数を示します。カラーの位置検出では、ステップサイズの初期値には最大テンプレートの半分のサイズまで使用することができます。

初期検索段階で、画像内で一致する可能性のある位置を生成します。次のステップでは、山登リアルゴリズムを使用して、そのリストから最適一致位置を検索します。

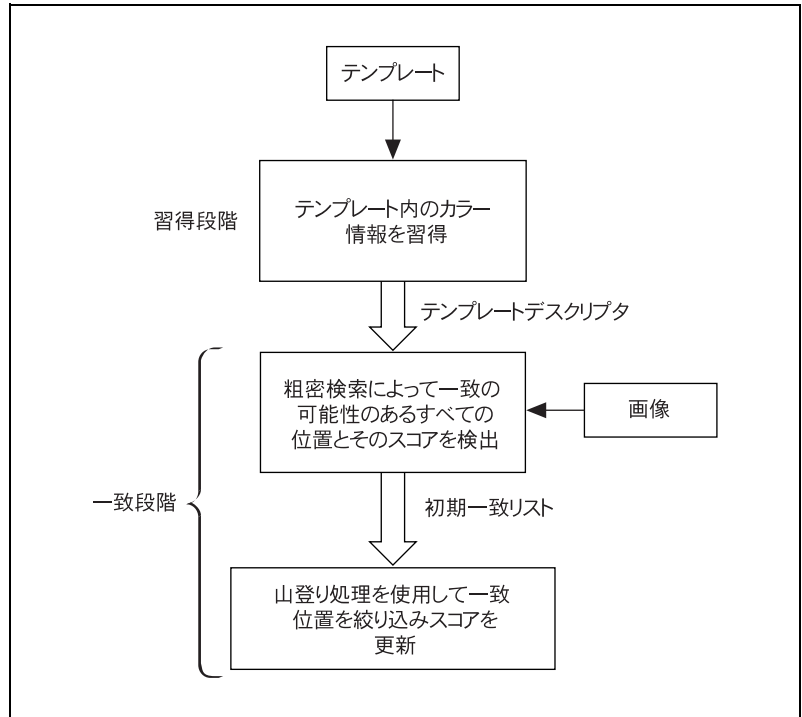


図 14-11 カラーの位置検出処理の概要

カラーパターンマッチング

カラー画像内で既知の基準パターンをすばやく探し出すには、パターンマッチングを使用します。カラーパターンマッチングを実行する際は、検索するオブジェクトを表すモデルまたはテンプレートを作成します。次に、マシンビジョンアプリケーションが各集録画像からモデルを検索し、各一致項目に対してスコアを計算します。このスコアは、検出したパターンがモデルにどの程度近いかを示すものです。パターンの色と空間情報で完全に表される基準パターンを見つけるために、カラーパターンマッチングを使用します。

使用目的

グレースケール、つまり白黒のパターンマッチングは、アライメント、ゲー징、および検査アプリケーションで使用できるツールとして定着しています。（パターンマッチングについての詳細は、第 12 章、「[パターンマッチング](#)」を参照してください。）それらすべてのアプリケーション分野で、色によってオブジェクトと背景とのコントラストを高めて区別しやすくすることにより、白黒での問題点が緩和されます。カラーパターンマッチングアルゴリズムにより、色が存在する場合にオブジェクトをすばやく検出することができます。

次のような場合にカラーパターンマッチングを使用します。

- 検出するオブジェクトには背景と大きく異なるカラー情報が含まれ、画像内でのそのオブジェクトの位置を非常に正確に検出したい場合。そのようなアプリケーションでは、カラーの位置検出よりカラーパターンマッチングの方がより精度の高いソリューションです。カラーの位置検出では検索段階で形状情報を使用しないため、ピクセル精度での一致オブジェクトの検出は非常に困難です。
- 検出するオブジェクトのグレースケールプロパティが、非常に特徴づけにくい、あるいは検索画像内の他のオブジェクトと極めて似通っている場合。そのようなときは、グレースケールパターンマッチングでは正確な結果が得られないことがあります。オブジェクトに画像内の他のオブジェクトと識別できるカラー情報があれば、マシンビジョンソフトウェアにそのオブジェクトを検出するための追加情報を与えることができます。

図 14-12 は、画像から抵抗器を検出するために、カラーの位置検出ではなくカラーパターンマッチングを使用する場合のメリットを示します。カラーの位置検出でも抵抗器は検出されますが、カラー情報のみに頼るため一致結果はあまり正確ではありません。カラーパターンマッチングでは、まずカラーマッチングを使用してオブジェクトを検出し、次にパターンマッチングを使用して位置を確定して、より正確な結果を出力します。

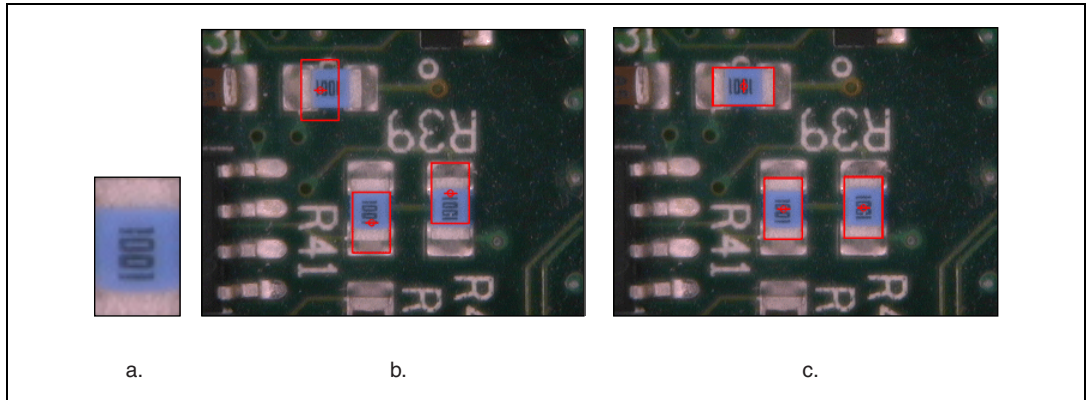


図 14-12 カラーの位置検出とカラーパターンマッチングの比較

図 14-13 は、ヒューズボックスからカラーコードされたヒューズを検出する際に、カラー情報を使用するメリットを示します。図 14-13a は、ヒューズボックスのグレースケール画像を示します。図 14-13a のヒューズボックスの画像の場合、グレースケール輝度が近いこととヒューズの半透明の性質により、グレースケールパターンマッチングではヒューズ 20 とヒューズ 25 を明確に区別することができません。図 14-13b のカラー画像では、色によりヒューズが区別しやすくなっています。色を追加することにより、パターンマッチングツールの精度と確実性をさらに向上させることができます。



図 14-13 ヒューズボックス検査に色を追加するメリット

IMAQ Vision のカラーパターンマッチングツールは、特徴を理想的に表したモデル、つまりテンプレートと呼ばれるものと、画像内に含まれる特徴との間の類似点を測定します。特徴とは、画像内にあるカラーピクセルの特定パターンとして定義します。

カラーパターンマッチングは、多くのアプリケーションで重要な役割を果たします。カラーパターンマッチングを使用すると、画像内に存在するテンプレートの発生数と位置に関する情報をアプリケーションに提供することができます。カラーパターンマッチングは、ゲージング、検査、およびアライメントの 3 つのタイプのアプリケーションで使用することができます。

ゲージング

多くのゲージングアプリケーションでは、画像内のオブジェクト間の距離を検出してから測定、つまりゲージングします。特徴を検索して検出することが、多くのゲージングアプリケーションの成功の決め手となる重要な処理タスクです。測定するコンポーネントが色によって固有に識別できれば、カラーパターンマッチングでコンポーネントをすばやく見つけることができます。

検査

検査は、欠損している部品や不鮮明な印刷など、単純な欠陥を検出します。一般的なアプリケーションとしては、家庭用品の瓶のラベルの欠陥印刷の検査などがあります。たいていのラベルはカラー印刷であるため、ラベルの詳細の検査を行う前に、カラーパターンマッチングを使用して画像内のラベルを検出します。カラーパターンマッチングツールによって返されたスコアは、ラベルが基準に達しているかどうかの決定にも使用できます。

アライメント

アライメントは、基準パターンを見つけて、既知のオブジェクトの位置と方向を求めます。基準パターンは、オブジェクト上の基準ポイントとして使用します。ほとんどのアプリケーションではグレースケールパターンマッチングで十分ですが、アライメントアプリケーションの中には、より確実な結果を出すためにカラーパターンマッチングの使用が必要なものもあります。

カラーパターンマッチングツールの機能

自動化されたマシンビジョンアプリケーションでは、検査対象の物質または部品の視覚的外観は、部品の向き、倍率の変更、および照明の変化などにより変わることがあります。カラーパターンマッチングツールを使用すると、そのような変化にかかわらず基準パターンを検出して正確な結果を出力することができます。

パターンの向きおよび複数のインスタンス

カラーパターンマッチングツールを使用すると、画像内のパターンが回転していたり倍率がわずかに変更されていても、基準パターンを見つけることができます。パターンが画像内で回転したり倍率が変更されている場合、カラーパターンマッチングツールは以下を検出します。

- 画像内のパターン
- パターンの画像内における位置
- パターンの向き
- 画像内に発生するパターンの複数のインスタンス（適用される場合）

図 14-14a は、テンプレート画像、つまりパターンを示します。

図 14-14b と図 14-14c は、テンプレートの複数のインスタンスを示します。図 14-14b は、テンプレートが画像内で移動したものです。

図 14-14c は、テンプレートが画像内で回転したものです。

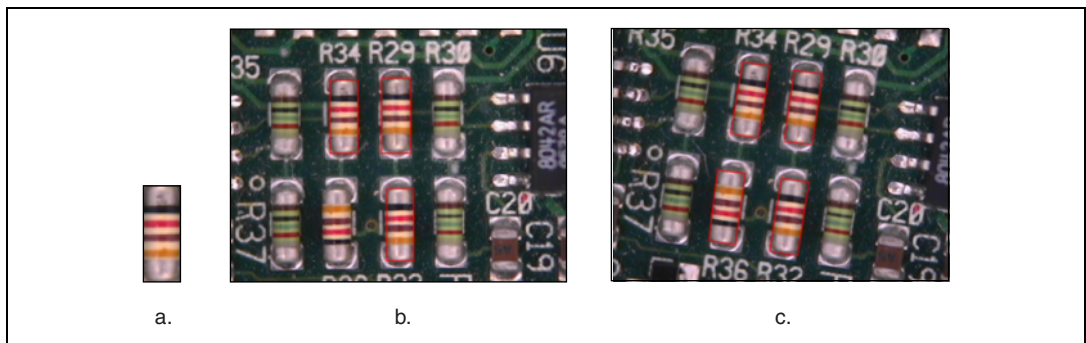


図 14-14 パターンの向き

周辺光状態

カラーパターンマッチングツールは、照明が画像全体で均等に变化している条件下で、画像から基準パターンを検出することができます。照明の変化に対処する場合、カラー解析の方がグレースケール処理より確実な方法であるため、影がある場合など不均等に光が变化する状況では、グレースケールパターンマッチングよりカラーパターンマッチングの方が適しています。

図 14-15a は、元のテンプレート画像を示します。図 14-15b は、同じパターンを明るい照明下で表示したものです。図 14-15c は、そのパターンを暗い照明下で表示したものです。

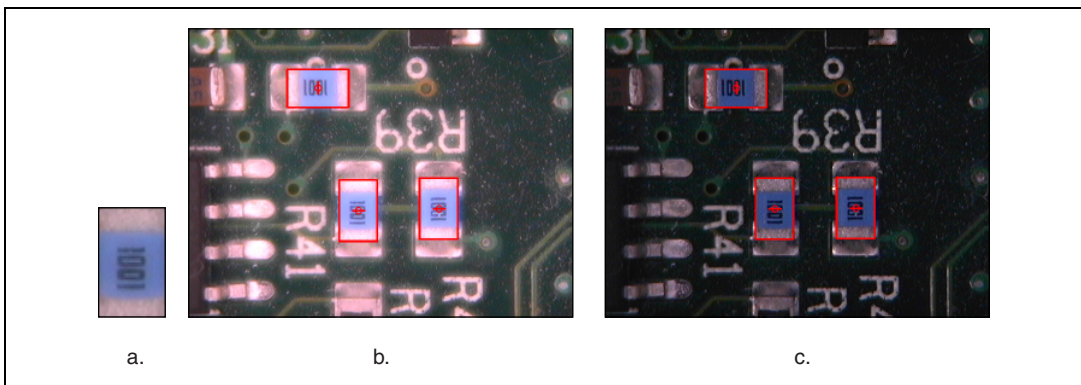


図 14-15 照明状態の例

ブレとノイズの状態

カラーパターンマッチングでは、ブレやノイズによって変形されたパターンを検出することができます。ブレは、通常誤った焦点や被写界深度の変化によって発生します。

カラーパターンマッチングの概念

カラーパターンマッチングは、カラー情報と空間情報を組み合わせて、画像内のカラーパターンをすばやく検出する特有の方法です。このツールは、カラーマッチングとグレースケールパターンマッチングの技術を相乗的に利用して、カラー画像からカラーパターンを検出します。

カラーマッチングとカラーの位置検出

カラーマッチングでは、画像または画像の領域のカラー内容を既存のカラー情報と比較します。画像内のカラー情報は、1 つまたは複数のカラーからなります。カラーマッチングを使用するには、基準として使用するカラー情報を含む画像領域を定義します。マシンビジョンソフトウェアは、画像の三次元カラー情報を学習して、それを一次元のカラースペクトルに表します。マシンビジョンアプリケーションは画像全体または画像の一部の領域のカラー情報を、学習したカラースペクトルと比較して、各領域に対しスコアを計算します。このスコアは、画像領域のカラー情報がカラースペクトルに表示された情報にどれほど一致しているかを表すものです。カラーマッチングを使用するためには、マッチングを行う前に画像内でのオブジェクトの位置がわかっている必要があります。

カラーの位置検出ソフトウェアは、カラーマッチングの機能をさらに拡張し、画像内でのオブジェクトの位置が不明なアプリケーションでも使用できるようにしています。カラーの位置検出では、テンプレート画像のカラー情報を使用して、検索画像内に存在するテンプレートを検索します。基本操作では、画像上でテンプレートをピクセル単位で移動させ、カラーマッチングアルゴリズムを使用して、画像の現在の位置でのカラー情報をテンプレートのカラー情報と比較します。カラーマッチングのために画像全体を検索するのは時間がかかるため、カラーの位置検出ソフトウェアは検出プロセスの速度を向上させるためのテクニックを使用します。粗密検索では、一致オブジェクトの画像内での大まかな位置を検出します。各一致オブジェクトの周辺に対して、山登りアルゴリズムを使用したより詳細な検索を実行し、一致オブジェクトの正確な位置を検出します。カラーの位置検出は、特定のカラー属性を持つ領域を画像から探し出すための効果的な方法です。

詳細については、本章の「[カラーマッチング](#)」および「[カラーの位置検出](#)」の各セクションを参照してください。

グレースケールパターンマッチング

IMAQ Vision で使用するグレースケールパターンマッチングメソッドでは、画像理解テクニックを使用してテンプレート情報を解釈し、その情報を使用して画像からテンプレートを検出します。画像理解とは、テンプレート画像の特徴に関する情報を生成する画像処理テクニックです。そのメソッドには次のようなものがあります。

- 画像の幾何学的モデル
- 効率的な画像の不均等サンプリング
- 回転に影響されないテンプレート情報の抽出

IMAQ Vision では、画像のエッジ情報とインテリジェント画像サンプリングテクニックを組み合わせ、パターンのマッチングを行います。画像のエッジ内容により、画像の構造に関する情報がコンパクトな形で得られます。インテリジェントサンプリングテクニックは、画像全体の内容を表すテンプレートからポイントを抽出します。エッジ情報とインテリジェントサンプリングメソッドにより、画像内の冗長情報を削減し、パターンマッチングツールの速度と精度を向上させます。画像内でパターンが回転する可能性がある場合は同様のテクニックを使用しますが、テンプレートにはパターンの回転を反映した値を持つピクセルを特別に選択します。その結果高速で正確なグレースケールパターンマッチングが実行されます。IMAQ Vision のパターンマッチングを使用すると、サイズ（± 5%）や向き（0° から 360°）が変化したり画質が低下していても、正確にオブジェクトを検出することができます。

グレースケールパターンマッチングの詳細については、第 12 章、「[パターンマッチング](#)」を参照してください。

カラーの位置検出とグレースケールパターンマッチング

カラーパターンマッチングでは、カラーの位置検出とグレースケールパターンマッチングを組み合わせ使用して、テンプレートを検索します。カラーパターンマッチングを使用してテンプレートを検索する際、ソフトウェアはテンプレートのカラー情報を使用して画像内からテンプレートのインスタンスを検索します。次に、検出された部分の周辺領域にグレースケールパターンマッチングを適用し、画像内でのテンプレートの正確な位置を検出します。図 14-16 は、カラーパターンマッチングアルゴリズムの一般的なフローを示します。検索可能領域のサイズは、検索方法やカラー感度などユーザが指定した入力に基づき、ソフトウェアによって決定します。

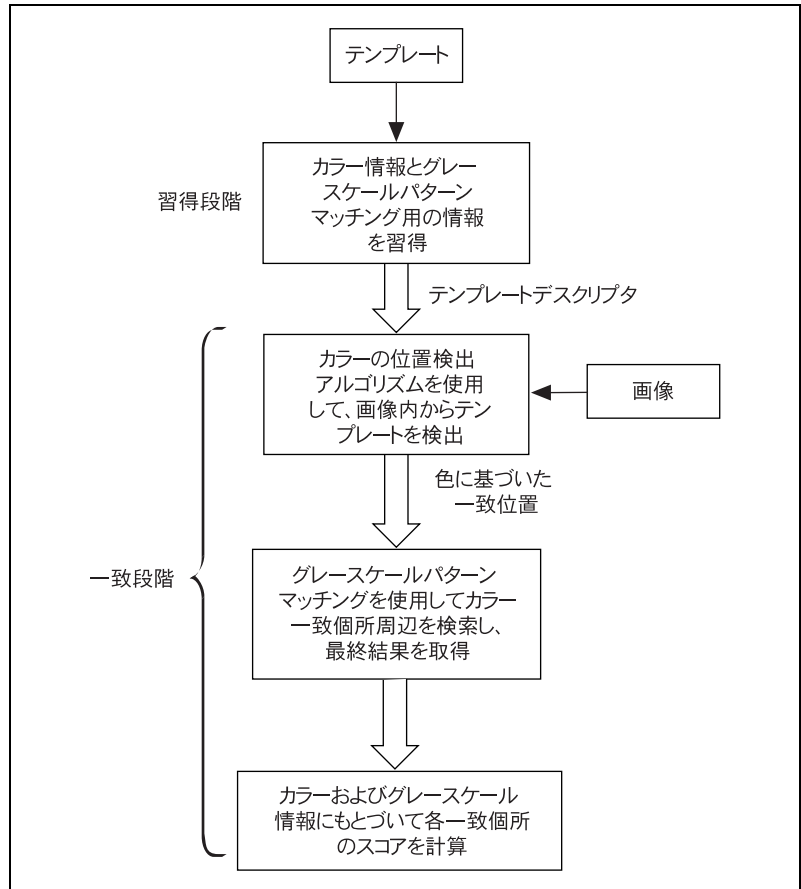


図 14-16 カラーパターンマッチング処理の概要

計測器リーダ

本章では、メータ、液晶ディスプレイ（LCD）、およびバーコードを読み取る計測器リーダについて説明します。

概要

計測器リーダとは、メータ、7セグメント表示装置、バーコードなどの読み取りが必要なアプリケーションをすばやく開発するために使用できる関数です。

使用目的

シンプルなメータ、LCD 表示、バーコードなどの画像から情報を取得する場合に、計測器リーダを使用します。

メータ関数

メータ関数を使用すると、メータやゲージから値を読み取ることが必要なアプリケーションを容易にすばやく開発することができます。これらの関数では、高レベルのビジョンプロセスによりメータやゲージの針の位置を抽出することができます。

この情報によって、ゲージのキャリブレーションなど別のアプリケーションを構築することができます。この関数を使用して、対象の領域から針の基部と先端を計算します。それは針の初期位置とフルスケールの位置を示すものです。次に、それらの VI を使用して、すでに計算したパラメータによって針の位置を読み取ります。

認識プロセスには 2 つの段階があります。

- ユーザが針の先端を指定する学習段階。
- 針の現在位置を確定する解析段階。

メータ関数は、明るい背景上に暗い色の針、または暗い背景上に明るい色の針のあるメータまたはゲージに対して動作するようになっています。

メータアルゴリズムの制約

このセクションでは、メータ関数に使用されるアルゴリズムの制約条件について説明します。このアルゴリズムは、光の変化にあまり影響を受けません。

検出処理では、針の基部の位置が非常に重要です。針の初期位置とフルスケールの位置を示すラインを丁寧に描きます。基部の座標と針の先端によって描かれた弧のポイントの座標は、設定段階で計算されます。これらの座標は、検査時にメータを読み取るために使用します。

LCD 関数

LCD 関数を使用すると、7 セグメント表示から値を読み取ることが必要なアプリケーションを容易にすばやく開発することができます。

その関数を使用して、画像から 7 セグメント数字の情報を抽出します。

読み取りプロセスには 2 つの段階があります。

- 7 セグメント表示を検出する対象の領域をユーザが指定する学習段階。
- ユーザが指定した領域を解析して、7 セグメント数字を読み取る読み取り段階。

IMAQ Vision の LCD 関数は、7 セグメント数字表示器の認識と読み取りに必要な高レベルのビジョンプロセスを提供します。このライブラリにある VI は、LCD または LED のいずれかを使用する 7 セグメント表示（発光ダイオードまたは電子発光表示器による 7 セグメント）用に設計されています。

このライブラリにある関数は、次のタスクを実行することができます。

- 複数桁を含む四角形の領域から各 7 セグメント数字周辺の領域を検出。
- 単一桁の値の読み取り。
- 表示された数の値、符号、および小数点の読み取り。

LCD アルゴリズムの制約

満足な検出結果が得られない場合、4つの要因が考えられます。

- 水平または垂直の高レベルの光のドリフト。
- 背景とセグメントの間のコントラスト不足。
- 多量のノイズ。
- 低い画像解像度。

これらの要因をそれぞれ数値化することにより、アルゴリズムの結果が正確でない理由がわかります。

光のドリフトは、LCD 画面の背景の左上と右下における平均ピクセルの差によって数値化します。8 ビット画像で光のドリフトが 90 より大きい場合、検出結果は正確でないことがあります。

コントラストは、背景の四角形領域とセグメントの四角形領域の平均ピクセル値の差によって求めます。正確な結果を得るには、この差が 8 ビット画像（256 グレーレベル）で 30 より大きい必要があります。

ノイズは、背景の四角形領域内に含まれるピクセル値の標準偏差によって定義します。正確な結果を得るには、この値が 8 ビット画像（256 グレーレベル）で 15 より小さい必要があります。

また、各桁が 18×12 ピクセルより大きくなければ、正確な結果は得られません。

バーコード

バーコードの概念は、一次元のバーコードにコード化された値を読み取るアプリケーションに適用されます。IMAQ Vision では、現在、Code 25、Code 39、Code 93、Code 128、EAN 8、EAN 13、Codabar、MSI、および UPC A のバーコードをサポートしています。

バーコードを認識するためのプロセスには、2つの段階があります。

- ユーザが画像内の対象領域を指定して、バーコードを含む領域を限定する学習段階。
- ユーザが指定した領域を解析してバーコードをデコードする認識段階。

バーコードアルゴリズムの制約

デコード処理中に次のような要因でエラーが発生することがあります。

- 低い画像解像度。
- 水平または垂直の高レベルの光のドリフト。
- 画像のバーのコントラスト。
- 多量のノイズ。

バーの幅とスペースの幅が 2 種類あるバーコード (Code 39、Codabar、Code 25、および MSI code) と、バーの幅とスペースの幅が 4 種類あるバーコード (Code 93、Code 128、EAN 13、EAN 8、および UPC A) では、制約の条件が異なります。

画像の解像度は、最も細いバーとスペースの幅によって決まります。それらの幅は、すべてのバーコードで少なくとも 3 ピクセル必要です。

光のドリフトは、バーコードの背景の左 (上) のラインと右 (下) のラインのグレーレベル平均の差によって数値化します。光のドリフトが、バーとスペースの幅が 2 種類のバーコードでは 120、バーとスペースの幅が 4 種類のバーコードでは 100 を超えると、デコードが正確に行われないことがあります。

露出過度の画像では、バーコードの幅の広いバーと狭いバーでグレーレベルが異なる傾向があります。グレーレベル差が、バーとスペースの幅が 2 種類のバーコードでは 80、バーとスペースの幅が 4 種類のバーコードでは 100 より少ないと、デコードが正確に行われないことがあります。

幅の広いバーと狭いバーの間のグレーレベル差について考えてみます。細いバーはほとんど見えません。8 ビット画像 (256 グレーレベル) でのグレーレベル差が、バーとスペースの幅が 2 種類のバーコードで 115、バーとスペースの幅の差が 4 種類のバーコードで 100 を超えると、正確な結果が得られないことがあります。

ノイズは、背景に描かれた四角形の関心領域の標準偏差として定義します。その値は、バーとスペースの幅が 2 種類のバーコードでは 57、バーとスペースの幅が 4 種類のバーコードでは 27 より少ない必要があります。

バーコードに反射が映り込むと、バーコードから読み取った値がエラーになることがあります。反射によってバーやスペースが隠されていると、エラーが発生します。

カーネル

カーネルとは、ピクセルとその近接ピクセルとの関係を表す構造です。

定義済みの傾斜カーネル

このセクションにある表に、定義済みの傾斜カーネルをリストしています。

プレウィットフィルタ

プレウィットフィルタには以下のカーネルがあります。西 (W)、南 (S)、東 (E)、北 (N) という表記は、明るい領域のどのエッジを描くかを示します。

表 A-1 プレウィットフィルタ

#0 W/ エッジ	#1 W/ 画像	#2 SW/ エッジ	#3 SW/ 画像
-1 0 1	-1 0 1	0 1 1	0 1 1
-1 0 1	-1 1 1	-1 0 1	-1 1 1
-1 0 1	-1 0 1	-1 -1 0	-1 -1 0
#4 S/ エッジ	#5 S/ 画像	#6 SE/ エッジ	#7 SE/ 画像
1 1 1	1 1 1	1 1 0	1 1 0
0 0 0	0 1 0	1 0 -1	1 1 -1
-1 -1 -1	-1 -1 -1	0 -1 -1	0 -1 -1
#8 E/ エッジ	#9 E/ 画像	#10 NE/ エッジ	#11 NE/ 画像
1 0 -1	1 0 -1	0 -1 -1	0 -1 -1
1 0 -1	1 1 -1	1 0 -1	1 1 -1
1 0 -1	1 0 -1	1 1 0	1 1 0
#12 N/ エッジ	#13 N/ 画像	#14 NW/ エッジ	#15 NW/ 画像
-1 -1 -1	-1 -1 -1	-1 -1 0	-1 -1 0
0 0 0	0 1 0	-1 0 1	-1 1 1
1 1 1	1 1 1	0 1 1	0 1 1

ソーベルフィルタ

ソーベルフィルタは、プレウィットフィルタと極めて似ていますが、大きなウェイトを割り当てられている特定の軸に沿って、輝度変化をハイライトする点で異なります。ソーベルフィルタには以下のカーネルがあります。西 (W)、南 (S)、東 (E)、北 (N) という表記は、明るい領域のどのエッジを描くかを示します。

表 A-2 ソーベルフィルタ

#16 W/ エッジ	#17 W/ 画像	#18 SW/ エッジ	#19 SW/ 画像
-1 0 1	-1 0 1	0 1 2	0 1 2
-2 0 2	-2 1 2	-1 0 1	-1 1 1
-1 0 1	-1 0 1	-2 -1 0	-2 -1 0
#20 S/ エッジ	#21 S/ 画像	#22 SE/ エッジ	#23 SE/ 画像
1 2 1	1 2 1	2 1 0	2 1 0
0 0 0	0 1 0	1 0 -1	1 1 -1
-1 -2 -1	-1 -2 -1	0 -1 -2	0 -1 -2
#24 E/ エッジ	#25 E/ 画像	#26 NE/ エッジ	#27 NE/ 画像
1 0 -1	1 0 -1	0 -1 -2	0 -1 -2
2 0 -2	2 1 -2	1 0 -1	1 1 -1
1 0 -1	1 0 -1	2 1 0	2 1 0
#28 N/ エッジ	#29 N/ 画像	#30 NW/ エッジ	#31 NW/ 画像
-1 -2 -1	-1 -2 -1	-2 -1 0	-2 -1 0
0 0 0	0 1 0	-1 0 1	-1 1 1
1 2 1	1 2 1	0 1 2	0 1 2

次の表は、定義済みの傾斜 5×5 カーネルをリストしたものです。

表 A-3 傾斜 5×5

#0 W/ エッジ	#1 W/ 画像	#2 SW/ エッジ	#3 SW/ 画像
0 -1 0 1 0	0 -1 0 1 0	0 0 1 1 1	0 0 1 1 1
-1 -2 0 2 1	-1 -2 0 2 1	0 0 2 2 1	0 0 2 2 1
-1 -2 0 2 1	-1 -2 1 2 1	-1 -2 0 2 1	-1 -2 1 2 1
-1 -2 0 2 1	-1 -2 0 2 1	-1 -2 -2 0 0	-1 -2 -2 0 0
0 -1 0 1 0	0 -1 0 1 0	-1 -1 -1 0 0	-1 -1 -1 0 0
#4 S/ エッジ	#5 S/ 画像	#6 SE/ エッジ	#7 SE/ 画像
0 1 1 1 0	0 1 1 1 0	1 1 1 0 0	1 1 1 0 0
1 2 2 2 1	1 2 2 2 1	1 2 2 0 0	1 2 2 0 0
0 0 0 0 0	0 0 1 0 0	1 2 0 -2 -1	1 2 1 -2 -1
-1 -2 -2 -2 -1	-1 -2 -2 -2 -1	0 0 -2 -2 -1	0 0 -2 -2 -1
0 -1 -1 -1 0	0 -1 -1 -1 0	0 0 -1 -1 -1	0 0 -1 -1 -1
#8 E/ エッジ	#9 E/ 画像	#10 NE/ エッジ	#11 NE/ 画像
0 1 0 -1 0	0 1 0 -1 0	0 0 -1 -1 -1	0 0 -1 -1 -1
1 2 0 -2 -1	1 2 0 -2 -1	0 0 -2 -2 -1	0 0 -2 -2 -1
1 2 0 -2 -1	1 2 1 -2 -1	1 2 0 -2 -1	1 2 1 -2 -1
1 2 0 -2 -1	1 2 0 -2 -1	1 2 2 0 0	1 2 2 0 0
0 1 0 -1 0	0 1 0 -1 0	1 1 1 0 0	1 1 1 0 0
#12 N/ エッジ	#13 N/ 画像	#14 NW/ エッジ	#15 NW/ 画像
0 -1 -1 -1 0	0 -1 -1 -1 0	-1 -1 -1 0 0	-1 -1 -1 0 0
-1 -2 -2 -2 -1	-1 -2 -2 -2 -1	-1 -2 -2 0 0	-1 -2 -2 0 0
0 0 0 0 0	0 0 1 0 0	-1 -2 0 2 1	-1 -2 1 2 1
1 2 2 2 1	1 2 2 2 1	0 0 2 2 1	0 0 2 2 1
0 1 1 1 0	0 1 1 1 0	0 0 1 1 1	0 0 1 1 1

次の表は、定義済みの傾斜 7×7 カーネルをリストしたものです。

表 A-4 傾斜 7×7

#0 W/ エッジ	#1 W/ 画像	#2 S/ エッジ	#3 S/ 画像
0 -1 -1 0 1 1 0	0 -1 -1 0 1 1 0	0 1 1 1 1 1 0	0 1 1 1 1 1 0
-1 -2 -2 0 2 2 1	-1 -2 -2 0 2 2 1	1 2 2 2 2 2 1	1 2 2 2 2 2 1
-1 -2 -3 0 3 2 1	-1 -2 -3 0 3 2 1	1 2 3 3 3 2 1	1 2 3 3 3 2 1
-1 -2 -3 0 3 2 1	-1 -2 -3 1 3 2 1	0 0 0 0 0 0 0	0 0 0 1 0 0 0
-1 -2 -3 0 3 2 1	-1 -2 -3 0 3 2 1	-1 -2 -3 -3 -3 -2 -1	-1 -2 -3 -3 -3 -2 -1
-1 -2 -2 0 2 2 1	-1 -2 -2 0 2 2 1	-1 -2 -2 -2 -2 -2 -1	-1 -2 -2 -2 -2 -2 -1
0 -1 -1 0 1 1 0	0 -1 -1 0 1 1 0	0 -1 -1 -1 -1 -1 0	0 -1 -1 -1 -1 -1 0
#4 E/ エッジ	#5 E/ 画像	#6 N/ エッジ	#7 N/ 画像
0 1 1 0 -1 -1 0	0 1 1 0 -1 -1 0	0 -1 -1 -1 -1 -1 0	0 -1 -1 -1 -1 -1 0
1 2 2 0 -2 -2 -1	1 2 2 0 -2 -2 -1	-1 -2 -2 -2 -2 -2 -1	-1 -2 -2 -2 -2 -2 -1
1 2 3 0 -3 -2 -1	1 2 3 0 -3 -2 -1	-1 -2 -3 -3 -3 -2 -1	-1 -2 -3 -3 -3 -2 -1
1 2 3 0 -3 -2 -1	1 2 3 1 -3 -2 -1	0 0 0 0 0 0 0	0 0 0 1 0 0 0
1 2 3 0 -3 -2 -1	1 2 3 0 -3 -2 -1	1 2 3 3 3 2 1	1 2 3 3 3 2 1
1 2 2 0 -2 -2 -1	1 2 2 0 -2 -2 -1	1 2 2 2 2 2 1	1 2 2 2 2 2 1
0 1 1 0 -1 -1 0	0 1 1 0 -1 -1 0	0 1 1 1 1 1 0	0 1 1 1 1 1 0

定義済みのラプラシアンカーネル

次の表は、定義済みのラプラシアンカーネルをリストしたものです。

表 A-5 ラプラシアン 3×3

#0 輪郭 4	#1 + 画像× 1	#2 + 画像× 2
0 -1 0	0 -1 0	0 -1 0
-1 4 -1	-1 5 -1	-1 6 -1
0 -1 0	0 -1 0	0 -1 0
#3 輪郭 8	#4 + 画像× 1	#5 + 画像× 2
-1 -1 -1	-1 -1 -1	-1 -1 -1
-1 8 -1	-1 9 -1	-1 10 -1
-1 -1 -1	-1 -1 -1	-1 -1 -1
#6 輪郭 12	#7 + 画像× 1	
-1 -2 -1	-1 -2 -1	
-2 12 -2	-2 13 -2	
-1 -2 -1	-1 -2 -1	

表 A-6 ラプラシアン 5×5

#0 輪郭 24	#1 + 画像× 1
-1 -1 -1 -1 -1	-1 -1 -1 -1 -1
-1 -1 -1 -1 -1	-1 -1 -1 -1 -1
-1 -1 24 -1 -1	-1 -1 25 -1 -1
-1 -1 -1 -1 -1	-1 -1 -1 -1 -1
-1 -1 -1 -1 -1	-1 -1 -1 -1 -1

表 A-7 ラプラシアン 7×7

#0 輪郭 48	#1 + 画像× 1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
-1 -1 -1 48 -1 -1 -1	-1 -1 -1 49 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1

定義済みの平滑化カーネル

次の表は、定義済みの平滑化カーネルをリストしたものです。

表 A-8 平滑化 3×3

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

表 A-9 平滑化 5×5

1 1 1 1 1	1 1 1 1 1
1 1 1 1 1	1 1 1 1 1
1 1 0 1 1	1 1 1 1 1
1 1 1 1 1	1 1 1 1 1
1 1 1 1 1	1 1 1 1 1

表 A-10 平滑化 7×7

1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 0 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1

定義済みのガウシアンカーネル

次の表は、定義済みのガウシアンカーネルをリストしたものです。

表 A-11 ガウシアン 3×3

0	1	0	0	1	0	1	1	1
1	2	1	1	4	1	1	2	1
0	1	0	0	1	0	1	1	1
1	1	1	1	2	1	1	4	1
1	4	1	2	4	2	4	16	4
1	1	1	1	2	1	1	4	1

表 A-12 ガウシアン 5×5

1	2	4	2	1
2	4	8	4	2
4	8	16	8	4
2	4	8	4	2
1	2	4	2	1

表 A-13 ガウシアン 7×7

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

技術サポートおよび プロフェッショナルサービス

技術サポートおよびプロフェッショナルサービスについては、ナショナルインスツルメンツのウェブサイト (ni.com/jp) の下記のセクションを参照してください。

- **サポート**：オンライン技術サポートには以下のリソースがあります。
 - **セルフヘルプリソース**：解答やソリューションがすぐに必要な場合は、技術サポートリソースの広範なライブラリ (ni.com/support/ja) をご利用いただけます。それらのリソースは、日本語の他に英語とスペイン語でも提供しています。登録ユーザの方ならほとんどの製品で無償でご利用いただくことができ、ソフトウェアドライバおよびアップデート、技術サポートデータベース、製品マニュアル、トラブルシューティングウィザード、ハードウェアリファレンスデータベース、サンプルプログラム、チュートリアルおよびアプリケーションノート、計測器ドライバ、ディスカッションフォーラム、計測用語集などが含まれています。
 - **技術者によるサポートオプション**：弊社のエンジニアや計測 / オートメーション技術者へお問い合わせいただく場合は、ni.com/support/ja にアクセスしてください。オンラインシステムを利用することにより、質問内容が明確になり、電話、ディスカッションフォーラム、またはEメールにより弊社技術者に直接お問い合わせいただくことができます。
- **トレーニング**：自習形式のチュートリアルおよび対話式 CD（英語）については、ni.com/jp/training にアクセスしてください。また、インストラクタによる実践コースも世界各地で開催しております。
- **システムインテグレーション**：時間的に制約がある場合や、社内の技術リソースが限られている場合、その他プロジェクトで問題がある場合などは、ナショナルインスツルメンツのアライアンスプログラムのメンバーによるサービスをご利用いただけます。詳しくは、最寄りのナショナルインスツルメンツ営業所にお電話いただくか、ni.com/jp/alliance をご覧ください。

ナショナルインスツルメンツのウェブサイト (ni.com/jp または ni.com) で解答が見つからなかった場合は、最寄りの営業所またはナ

ショナルインスツルメンツ本社にお問い合わせください。世界各地の弊社営業所の連絡先は、本書の巻頭に掲載されています。また、弊社ウェブサイトの Worldwide Offices セクション (ni.com/niglobal) から各支社のウェブサイトにもアクセスすることもできます。各支社のサイトでは、お問い合わせ先、サポート電話番号、Eメールアドレス、イベント等に関する最新情報を提供しています。

用語集

接頭語	意味	値
p-	ピコ	10^{-12}
n-	ナノ	10^{-9}
μ -	マイクロ	10^{-6}
m-	ミリ	10^{-3}
k-	キロ	10^3
M-	メガ	10^6
G-	ギガ	10^9
t-	テラ	10^{12}

数値 / 記号

1D	一次元。
1/Y 乗関数	対数関数に似ていますが、効果は弱くなります。
2D	二次元。
3D	三次元。

A

AIPD 複素画像や画像に関連するキャリブレーション情報を保存するための、ナショナルインスツルメンツの内部画像形式（拡張子 APD）。

B

b	ビット。1つの二進数で、0または1。
B	バイト。8つの関連ビットのデータ、つまり8ビットバイナリ数。1バイトのデータを保存するのに必要なメモリの量のことも示します。
BMP	ビットマップ。8ビットおよびカラー画像に対して一般的に使用される画像ファイル形式（拡張子 BMP）。

C

CLUT カラーlookupアップテーブル。画像内のピクセルの値を赤、緑、および青 (RGB) の輝度に変換するテーブル。

D

dB デシベル。ボルト単位の信号に対して、2つの信号レベルの比率の対数測定を表すための単位： $\text{dB} = 20 \log_{10} V1/V2$ 。

F

FFT 高速フーリエ変換。画像のフーリエ変換の計算に使用する手法。

ft フィート。

H

h 時間。

HSI 色相、彩度、および輝度によるカラーエンコードスキーム。

HSL 色相、彩度、および明度情報を使ったカラーエンコードスキーム。画像内の各ピクセルは、色相に8ビット、彩度に8ビット、輝度に8ビット、未使用の8ビット、計32ビットを使用してコード化されます。

HSV 色相、彩度、および明度によるカラーエンコードスキーム。

Hz ヘルツ。1/秒単位の周波数。

I

I/O 入力 / 出力。通信チャンネル、オペレータインタフェースデバイス、データ集録および制御インタフェースなどを使用した、コンピュータシステムとのデータの送受信。

IMAQ 画像集録。

in. インチ。

INL 積分非線形性。アナログ I/O 回路の A/D または D/A 転送の理想的特性からの最悪ケースの偏差を LSB で表した尺度。

IRE 測定値の相対的単位 (Institute of Radio Engineers =無線技術学会により命名)。0 IRE はビデオ信号のブランキングレベルに相当し、100 IRE は白レベルに相当します。CCIR/PAL ビデオでは、黒レベルはブランキングレベルつまり 0 IRE に等しく、RS-170/NTSC ビデオでは黒レベルは 7.5 IRE です。

J

JPEG Joint Photographic Experts Group の略。8 ビットおよびカラー画像を保存するための、高減衰圧縮による画像ファイル形式 (拡張子 JPG)。

K

k キロ。ボルト、ヘルツ、メートルなどの測定単位とともに使用する、1,000 または 10^3 の標準メトリック接頭辞。

K キロ。データやコンピュータメモリを定量化するために B とともに使用する、1,024 または 2^{10} の接頭辞。

L

LabVIEW Laboratory Virtual Instrument Engineering Workbench の略。テストおよび計測アプリケーションで一般的に使用されるプログラミング言語 G に基づく、プログラム開発環境アプリケーション。

LSB 最下位ビット。

LUT ルックアップテーブル。画像のグレイレベル値の変換に使用される値を含む表。画像内の各グレイレベル値に対し、対応する新規の値をルックアップテーブルから取得します。

L スケルトン関数 スケルトン関数で L 型の構造要素を使用します。

M

m メートル。

M (1) メガの略。ボルト、ヘルツなどの測定単位とともに使用する、100 万または 10^6 の標準メトリック接頭辞。(2) メガの略。データやコンピュータメモリを定量化するために B とともに使用する、1,048,576 または 2^{20} の接頭辞。

MB	メモリのメガバイト。
MMX	Multimedia Extensions の略。インテルのチップをベースにした、整数の並列処理を可能にする技術で、8 ビット画像の高速処理ができます。
MSB	最上位ビット。
M スケルトン関数	スケルトン関数で M 型の構造要素を使用します。

N

NH-IMAQ	ナショナルインスツルメンツの IMAQ ハードウェア用ドライバソフトウェア。
N 次フィルタ	非線形フィルタを使用して画像をフィルタ処理します。このフィルタは、処理対象のピクセル周辺のピクセルの値を並べ（分類し）ます。処理対象のピクセルは、N 番目のピクセル値に設定されます。ここで N はフィルタの次数です。

P

PNG	Portable Network Graphic の略。8 ビット、16 ビット、およびカラー画像を保存するための、非高減衰圧縮による画像ファイル形式（拡張子 PNG）。
pts	ポイント。

R

RGB	赤、緑、および青（RGB）カラー情報を使用したカラーエンコードスキーム。カラー画像内の各ピクセルは、赤に 8 ビット、緑に 8 ビット、青に 8 ビット、アルファ値（未使用）に 8 ビットの 32 ビットでコード化されます。
ROI	関心領域。(1) 画像を表示しているウィンドウからグラフィックに選択された画像の領域。この領域に集中して処理を進めることができます。(2) 集録ウィンドウの、ハードウェアによるプログラムが可能な四角形の部分。
ROI ツール	LabVIEW ツールパレットにある各ツールの集まりで、画像から関心領域を選択することができます。それらのツールを使用すると、ポイントやライン、多角形、四角形、楕円領域、およびフリーハンドラインや領域を選択することができます。

S

s 秒。

T

TIFF Tagged Image File Format の略。8 ビット、16 ビット、およびカラー画像をエンコードするのに一般的に使用される画像形式（拡張子 TIF）。

V

V ボルト。

VI

仮想計測器。(1) 一般に PC とともに使用する、ハードウェアおよびソフトウェア要素を組み合わせたもので、従来のスタンドアロン計測器と同じ機能を果たします。(2) フロントパネルというユーザインタフェースとブロックダイアグラムというプログラムからなる、LabVIEW のソフトウェアモジュール (VI)。

Y

Y 累乗関数 「指数関数」を参照。

あ

穴埋め関数 バイナリ画像内に存在するオブジェクトの穴をすべて埋めます。

アライメント マシンビジョンアプリケーションで検査対象部分の位置、向き、およびスケールを測定する処理。

アルファチャンネル ガンマ補正など、カラー画像に関するその他の情報をコーディングするのに使用するチャンネル。アルファチャンネルは、RGB ピクセルの 4 バイト表記の最初のバイトとして格納されます。

い

イメージング 画像を集録および表示して画像データを解析するプロセス。

う

ウォーターシェッド 画像を複数の領域にセグメント化するテクニック。

内側傾斜 オブジェクトの内側境界線を検出します。

え

エッジ 画像内またはピクセル配列上における、ピクセル輝度の急激な変化（遷移）によって定義されます。

エッジ輝度差 エッジの傾斜または遷移領域に対応するピクセルの数。

エッジ検出 画像内のオブジェクトのエッジを識別する複数のテクニックのいずれか。

エッジコントラスト エッジの前と後の平均ピクセル輝度の差。

エッジヒステリシス 立ち上がりエッジと立ち下がりエッジ間のしきい値レベルの差。

円形関数 バイナリ画像で円形オブジェクトを検出します。

演算子 画像のマスク処理、結合、および比較を行います。IMAQ Vision では、算術および論理演算子を使用できます。

エントロピー 画像の任意性の尺度。高エントロピーの画像は、低エントロピーの画像よりピクセル値が変化に富んでいます。

お

オフセット 他の画像の基点を配置する、画像内の座標位置。オフセットの設定は、マスク操作を実行する際に便利です。

オープン 収縮の後に膨張を行うこと。オープン処理により、画像内の小さなオブジェクトが取り除かれ、オブジェクトの境界線が滑らかになります。

か

カーネル ピクセルとその近接ピクセルとの関係を表す構造。関係は、各近接ピクセルの重心係数によって指定されます。

解像度 ピクセルの行と列の数。 m 行 n 列で構成される画像は、解像度 $m \times n$ で表されます。

回転シフト	基準画像に対する、1つの画像の回転の程度。この回転は、画像の中心に対して計算されます。
回転に影響されないマッチング	テスト画像内でどのような向きに位置していても、どの角度で回転していても、基準パターンを検出できるパターンマッチングテクニック。
海里	航海および航空にて使用される長さの国際単位で、1,076.115 フィート、つまり 1,853 メートルに対応します。「マイル」も参照。
ガウシアンフィルタ	スムージングフィルタに似たフィルタですが、フィルタ処理でガウシアンカーネルを使用します。ガウシアンフィルタでは、平滑化フィルタよりもぼやけの効果が弱くなっています。
確率関数	画像内のピクセルが特定のグレーレベル値を持つ確率を定義します。
画素	デジタル画像の要素。ピクセルともいいます。
画像	二次元輝度関数 $f(x, y)$ のことで、 x と y は空間座標、任意のポイント (x, y) での値 f は、そのポイントでの輝度に比例しています。
画像アスペクト比	アクティブなピクセル領域の、アクティブなライン領域に対する比率。RS-170 または CCIR のような標準ビデオ信号の場合、フルサイズの画像アスペクト比は通常 4/3 (1.33) です。
画像強調	センサから集録した画像の質を、SN 比、画像コントラスト、エッジ鮮明度などの点で向上させるプロセス。
画像形式	画像をファイルに保存する方法を定義します。通常ヘッダとピクセルデータからなります。
画像処理	画像に適用可能なさまざまな処理および解析関数が含まれます。
画像の視覚化	画像（画像データ）をユーザに対して提示（表示）すること。
画像の鮮明度	「ピクセル深度」を参照。
画像バッファ	画像の保存に使用するメモリ位置。
画像パレット	画像を画面に表示するための色の階調で、通常カラーlookupアップテーブルで定義されます。
画像ファイル	画像に関するピクセルデータとその他の情報を含むファイル。
画像ブラウザ	ビジョンアプリケーションで解析または処理する画像のサムネールを含む画像。

画像ボーダー	画像を取り囲むユーザ定義のピクセル領域。ピクセルの近接ピクセルの値に基づいてピクセルを処理する関数では、画像のボーダーが必要です。
画像マスク	さらに処理を行うため、ソース画像の一部を分離したバイナリ画像。ソース画像内のピクセルは、その対応するマスクピクセルの値が0以外の場合に処理されます。対応するマスクピクセルの値が0の場合、ソースピクセルはそのままで変更されません。
画像理解	画像の内容を、ピクセルレベルでなく記号レベルで解釈するテクニック。
カラー画像	通常 RGB 形式にコード化された、カラー情報を含む画像。
カラースペース	色の算術的表現。たとえば、色は赤と緑と青、色相と彩度と明度、または色相と彩度と輝度によって表現することができます。
関数	単一のコード行で実行される一連のソフトウェア命令で、入力や出力がある場合があります、実行すると値が返されます。
ガンマ	ビデオ信号の輝度レベルとその輝度を生成するために必要な電圧レベルの差における非線形変化。

き

基準パターン	マシンビジョンアプリケーションが画像内の部分の位置と向きを検出するために使用する、その部分上の基準となるパターン。
輝度	(1) 色のデコーディング処理の際に、カラーピクセルの赤、緑、および青の各成分に追加される定数。(2) 白いオブジェクトとグレーのオブジェクト、および明るいオブジェクトと暗いオブジェクトを区別する知覚。(3) 赤、緑、青の三原色の合計を3で割ったもの。(赤 + 緑 + 青)/3
輝度キャリブレーション	光学密度や集光などのユーザ定義の数値を、画像のグレーレベル値に割り当てること。
輝度しきい値	オブジェクトのグレーレベル値の範囲に基づいて、オブジェクトの特性を示します。オブジェクトの輝度範囲がユーザ指定の範囲に入る場合、それをオブジェクトとみなします。それ以外の場合は、背景の一部とみなしません。
輝度範囲	画像内のオブジェクトのグレーレベル値の範囲を定義します。
輝度プロファイル	画像の ROI 上のピクセルのグレーレベル分布。

キャリパ	(1) エッジ検出、粒子解析、図心、および検索関数によって求められた位置に基づいて、距離、角度、円形フィット、重心などを計算する、IMAQ Vision Builder の関数。(2) 画像内の指定したパス上で一組のエッジを検出する計測関数。この関数はエッジの抽出を行い、次に、立ち上がりエッジと立ち下がりエッジ間の距離やエッジのコントラストなど指定の基準に基づいて一組のエッジを検出します。
距離関数	オブジェクトの各ピクセルに、オブジェクトの縁からの最短ユークリッド距離に等しいグレーレベル値を割り当てます。
距離キャリブレーション	画像内のラインの物理的寸法を定義することにより、ピクセルの物理的寸法を決めること。
近接演算	画像内のポイントに対する演算で、そのポイントに近接するピクセルの値を考慮します。
近接ピクセル	画像処理の際、その値が近隣のピクセルの値に影響するピクセル。あるピクセルの近接ピクセルは、カーネルまたは構造要素によって定義されません。
<	
空間キャリブレーション	画像内のピクセル領域に物理的寸法を割り当てること。
空間フィルタ	近接ピクセルの輝度変化に対応して、ピクセルの輝度を変更します。このフィルタは、エッジ検出、画像強調、ノイズ削減、平滑化などに使用できます。
空間分解能	画像の行と列の数で表した画像内のピクセル数。
クラスタ化	画像内で認識される位相の数に対応する離散クラス数に、画像をソートするテクニック。各クラスに対してグレー値と重心が決まります。このプロセスは、各位相またはクラスの重心を表す値が得られるまで繰り返されます。
グレースケール画像	白黒情報を含む画像。
グレースケールモフォロジー	グレーレベル画像に対してモフォロジー操作を実行する関数。
グレーレベル	画像内のピクセルの輝度。
グレーレベル収縮	画像内で、より輝度の低いピクセルに囲まれたピクセルの輝度を下げます。

グレーレベル膨張	画像内で、より輝度の高いピクセルに囲まれたピクセルの輝度を高めません。
クローズ	膨張のあとに収縮を行うこと。クローズ処理は、オブジェクト内の小さな穴を埋め、オブジェクトの境界線を滑らかにします。
黒基準レベル	画像において可能な最も暗いレベル。「白基準レベル」も参照。
クロマ	ビデオ信号におけるカラー情報。
クロミナンス	「クロマ」を参照。

け

傾斜コンボリューションフィルタ	「傾斜フィルタ」を参照。
傾斜フィルタ	グレーレベル値で輪郭を抽出（エッジ検出）します。傾斜フィルタには、プレウィットフィルタとソーベルフィルタが含まれます。
形状マッチング	形状テンプレートで指定したオブジェクトの形状と一致するオブジェクトを画像内から検出します。マッチングのプロセスは回転に影響されず、またオブジェクトのスケールにも影響されないように設定することができます。
ゲー징	オブジェクトまたはオブジェクト間の距離の測定。
計測器ドライバ	特定のプラグインコンピュータボードを制御する、NI-IMAQ などの高レベルソフトウェア関数のセット。計測器ドライバは、プログラミング言語から呼び出し可能な関数から、LabVIEW の仮想計測器（VI）にいたるまで、いくつかの形式があります。
検査	部品の欠損や部品表面のひびなどの単純な欠陥を検出するため、部品をテストするプロセス。
検査関数	画像内のピクセルのグループを解析して、サイズ、形状、位置、およびピクセルの連結性についての情報を返します。一般的なアプリケーションには、部品の品質、欠陥解析、オブジェクトの検出、オブジェクトの仕分けなどがあります。

こ

光学表現	FFT 変換後の画像で、中央部分に低周波数情報、端部分に高周波数情報が含まれます。
------	---

高減衰圧縮	解凍したときに元の画像に似ているがまったく同じにはならない圧縮方法。
構造要素	ほとんどのモフォロジー操作で使用するバイナリマスク。構造要素は、どの近接ピクセルが操作に影響するかを決定します。
コントラスト	画像内の明るい輝度値と黒の輝度値との差。
コンボリューション	「線形フィルタ」を参照。
コンボリューションカーネル	フィルタ処理において、フィルタを表すのに使用する二次元行列（またはテンプレート）。これらのカーネルの内容は、カーネルが表すフィルタのインパルス応答の離散二次元表記です。

さ

彩度	純色に加えられた白の量。彩度は、色の鮮やかさと関係があります。彩度 0 は、まったく白の足されていない純色に対応します。ピンクは、赤の彩度を落としたものです。
サブピクセル精度	エッジ座標の位置をピクセル以下の精度で検出します。
三次元表示	画像の光の輝度を三次元座標系に表示します。この座標系では、画像の空間座標が二次元を形成し、光の輝度が残りの一次元を形成します。
算術演算子	乗算、除算、加算、減算、および剰余などの画像演算。

し

しきい値範囲	2つのパラメータ、グレーレベルしきい値の上の値と下の値。
色相	ピクセルの主調色を表します。色相関数は、R、G、Bの三原色を使用して生成可能なすべての色をカバーする連続的な関数です。「RGB」も参照。
色相オフセット角	解析の際に不連続が関心値の範囲外で発生するようにするため、すべての色相値に追加する値。
色度	色相と彩度を組み合わせたもの。色度と輝度の関係により色の特徴が決まります。
シグマフィルタ	エッジを明確に描くハイパスフィルタ。
二乗関数	「指数関数」を参照。

指数関数	画像の明るい領域で輝度を低減しコントラストを高めて、画像の暗い領域ではコントラストを下げます。
指数関数とガンマ補正	画像内の高グレーレベル情報を拡張し、低グレーレベル情報を削減します。
自動メディアン関数	オープン処理とクローズ処理の組み合わせを使用して、オブジェクトの境界線を滑らかにする関数。
シフトに影響されないマッチング	テスト画像内で基準パターンがどの位置にあっても検出されますが、回転やサイズの変更には影響を受けるパターンマッチングテクニック。
収縮	境界線に沿ってオブジェクトのサイズを縮小し、画像内の孤立した点を取り除きます。
重心	(1) オブジェクト上で、いずれの軸に対してもオブジェクトの最初のモーメントを変えることなく、オブジェクトの全質量を集めることのできるポイント。(2) 画像ヒストグラムで、画像のグレースケール値範囲の図心を表すグレースケール値。
周波数フィルタ	周波数領域で、空間フィルタに対応するもの。画像の場合、周波数情報は空間周波数の形式になっています。
樹状突起	オブジェクトのスケルトンの枝状部分。
白基準レベル	特定のビデオシステムにおいて何を白とするかを定義するレベル。「黒基準レベル」も参照。
真理値表	処理に使用される規則を記述した、論理演算子に関連する表。

す

スキップ関数	各オブジェクトを他のオブジェクトから分離し、それらのオブジェクトから等距離にあるラインを、画像内に取得します。
スケールに影響されないマッチング	テスト画像で基準パターンがどのようなサイズになっていても検出できる、パターンマッチングテクニックです。
スケルトン関数	幅が1ピクセルになるまで、オブジェクトに対し細線化操作を繰り返し適用します。
スコア	集録した画像がテンプレート画像にどの程度近いかを示す0から1000までの数。1000というスコアは完全一致を示します。0というスコアはまったく一致していないことを示します。
スムージングフィルタ	ピクセルの近接ピクセルでの輝度変化を減衰することにより、画像を不鮮明にします。

せ

セグメント化関数	ラベル付けされたバイナリ画像を、それぞれが固有のオブジェクトを含む、重なりのないセグメントに完全分割します。
線形フィルタ	ピクセルの値とその近接ピクセルの値に基づいて、ピクセル値を計算する特殊なアルゴリズム。この計算で得られた合計値を行列の要素の合計で割り、新規のピクセル値を求めます。
鮮明度	1つのピクセルが取れる値の数、つまり画像に表示される色または階調の数のこと。

そ

相互相関	パターンマッチングを実行するための最も一般的な方法。
相対精度	ADCの精度をLSBで表した度数。すべての非線形および定量化エラーが含まれますが、ADCに送り込まれる回路のオフセットおよび増幅エラーは含まれません。
ソーベルフィルタ	3x3のフィルタカーネルを使用して、グレースケール値により輪郭を抽出(エッジ検出)します。
ゾーン	表示されている画像の中で、ユーザによるクリックに反応する領域。ゾーンを使用して、イベントを制御し、それをLabVIEW内で解釈することができます。
外側傾斜	オブジェクトの外側境界線を検出します。

た

対数関数演算子	画像の暗い領域で輝度とコントラストを高め、画像の明るい領域でコントラストを下げます。
対数と逆ガンマ補正	画像内の低グレースケール情報を拡張し、高グレースケール情報を圧縮します。
ダイナミックレンジ	回路が処理可能な最大信号レベルの、同じく最小信号レベル(通常ノイズレベル)に対する比率で、通常デシベルで表記します。
ダニエルソン関数	距離関数に似ていますが、より正確な結果を得ることができます。

ち

中心エネルギー グレースケール画像の重心。「重心」を参照。

つ

ツールパレット 関心領域の変更、ズームイン / アウト、画像パレットの変更などを行うためのツールの集合。

て

定量解析 画像内のオブジェクトのさまざまな測定値を取得します。

適正オープン オープンとクローズの連続的操作を組み合わせたもので、小さな粒子を取り除いたりオブジェクトの境界線を滑らかにしたりするのに使用します。

適性クローズ クローズとオープンの連続的操作を組み合わせたもので、小さな穴を埋めたりオブジェクトの境界線を滑らかにしたりするのに使用します。

デジタル画像 離散ピクセル数に変換された画像 $f(x, y)$ 。空間座標と輝度の両方が指定されます。

デフォルト設定 ドライバに記録されているデフォルトパラメータ値。多くの場合、制御器のデフォルト入力は特定の値になっています（たいていは 0）。

デンシトメトリ 光学または写真濃度を定めること。

テンプレート カラーマッチング、形状マッチング、またはパターンマッチング関数を使用して、画像内で一致を検出しようとしている色、形状、またはパターン。テンプレートは、画像から選択された領域の場合と、画像全体の場合があります。

と

凸関数 バイナリ画像でオブジェクトの凸領域を計算します。

凸閉包 粒子を囲むことのできる最小の凸多角形。

ドライバ IMAQ や DAQ デバイスなど、特定のハードウェアデバイスを制御するソフトウェア。

に

二値化 指定範囲内の輝度を持つすべてのピクセルをオブジェクトに、それ以外のピクセルを背景に割り当てることにより、オブジェクトを背景から分離します。生成されるバイナリ画像では、オブジェクトは 255 のピクセル輝度で表示され、背景は 0 に設定されます。

は

バイナリ画像 通常ピクセル輝度 1（または 255）で表すオブジェクトと 0 で表す背景を含む画像。

バイナリ二値化 画像ピクセルの輝度に基づいて、画像を関心オブジェクト（ピクセル値 1 を割り当て）と背景（ピクセル値 0 を割り当て）に分離すること。

バイナリモフォロジー バイナリ画像に対してモフォロジー操作を実行する関数。

ハイパス FFT フィルタ 画像の FFT 領域に存在する低周波数を削除または減衰します。

ハイパス減衰 ローパス減衰の逆。

ハイパス周波数フィルタ 画像の周波数領域に存在する低周波数を削除または減衰（トランケート）します。ハイパス周波数フィルタは、空間画像での輝度の緩い変化に関連する情報を抑圧します。

ハイパストランケーション ローパストランケーションの逆。

ハイパスフィルタ 画像の輝度変化を強調し、エッジ（またはオブジェクトの境界線）を検出して、画像の詳細部分を強調します。

配列 順序付けおよび指標付けされた同じタイプのデータ要素のセット。

パターンマッチング グレースケール画像内でグレースケールテンプレートをすばやく見つけるためのテクニック。

バッファ 集録したデータの一時的な格納場所。

パレット 画像を画面に表示するための色の階調で、通常カラーlookupアップテーブルで定義されます。

反転関数 画像のピクセル値を反転し、画像のフォトネガティブを生成します。

ひ

ピクセル	画素。ビデオスキャンラインを構成する最小部分。コンピュータのモニターに表示する場合、ピクセルの最適寸法は正方形（アスペクト比 1:1、つまり幅と高さが同じ）です。
ピクセルアスペクト比	ピクセルが占める領域の実際の横サイズと縦サイズ間の比率。集録したピクセルは正方形であることが望ましいため、最適値は 1.0 ですが、通常はカメラの質によって 0.95 から 1.05 までのばらつきがあります。
ピクセル深度	ピクセルのグレーレベルを表すのに使用するビット数。
ピクセル補正	画像内のピクセルの物理的寸法を直接補正すること。
非高減衰圧縮	解凍したときに元の画像とまったく同じになる圧縮方法。
ヒストグラフ	LabVIEW で、直接グラフに配線可能なヒストグラム。
ヒストグラム	画像のピクセルの各グレーレベル値ごとの量的分布を示します。
ヒストグラム反転	画像のフォトネガティブを生成します。反転した画像のヒストグラムは、元の画像のヒストグラムを中心に対して水平に反転させたものと同じです。
ヒストグラム平均化	画像のピクセルのグレーレベル値を、ヒストグラムの全範囲（8 ビット画像で 0 から 255）を占めるように変換します。
非線形傾斜フィルタ	特に垂直エッジを抽出する、ハイパスエッジ抽出フィルタ。
非線形ソーベルフィルタ	二次元傾斜情報に基づいた、ハイパスエッジ抽出フィルタ。このフィルタは、傾斜演算によるノイズの強調を抑える平滑化効果があります。
非線形フィルタ	各ピクセル値をその周辺ピクセルの非線形関数に置き換えます。
非線形プレウィットフィルタ	二次元傾斜情報に基づいた、ハイパスエッジ抽出フィルタ。
ビット深度	ピクセルの値をコード化するのに使用するビット数 (n)。特定の n に対し、ピクセルは 2^n 通りの値をとることができます。たとえば、 n が 8 ビットの場合、ピクセルは、0 ~ 255 の間の 256 の値を取ることができます。 n が 16 ビットの場合、ピクセルは 0 ~ 65,536 または -32,768 ~ 32,767 の間の 65,535 の値を取ることができます。
ヒットミス関数	構造要素で定義したパターンに似たオブジェクトを画像から検出します。
微分フィルタ	グレーレベルで輪郭を抽出（エッジ検出）します。

標準表現 FFT 変換後の画像で、端部分に低周波数情報、中央分に高周波数情報が含まれます。

ピラミッドマッチング 画像をサブサンプルしたものを基準パターンと比較することにより、パターンマッチングアルゴリズムの速度を向上させるテクニック。

ふ

フーリエスペクトル 画像のフーリエ変換の振幅情報。

フーリエ変換 空間領域から周波数領域への画像の変換。

フォーム プログラムのユーザインタフェースを作成するために、制御器や表示器を配置する画面上のウィンドウまたは領域。

複素画像 画像の FFT から得られた情報を格納します。FFT プレーンを形成する複素数は、実数部に 32 ビット、虚数部に 32 ビットの 64 ビット浮動小数点値でコード化されています。

太線化 構造要素で指定したパターンと一致するオブジェクトにある部分を加えることによって、オブジェクトの形状を変更します。

プレウィットフィルタ 3×3 のフィルタカーネルを使用して、グレイレベル値により輪郭を抽出（エッジ検出）します。

分離関数 細い部分によって接触しているオブジェクトを切り離します。

へ

平均化関数 「ヒストグラム平均化」を参照。

平方根関数 「対数関数」を参照。

平面の数（画像内） 画像を構成するピクセルの配列数。グレイレベル画像や擬似カラー画像は、1つの平面で構成されていますが、RGB 画像は、赤、青、緑の各要素に対する3つの平面で構成されています。

ほ

ボーダー関数 バイナリ画像で、画像の縁に接触しているオブジェクト（または粒子）を削除します。

膨張 境界線に沿ってオブジェクトのサイズを拡大し、オブジェクト内の小さな穴を取り除きます。

法定マイル	「マイル」を参照。
補間	画像またはピクセル配列を再サンプリングする際に、既知の値間の値を検出するのに使用するテクニック。
細線化	構造要素で指定したパターンと一致するオブジェクトからある部分を取り除くことによって、オブジェクトの形状を変更します。
ぼやけ	画像の詳細部分の量を少なくします。ぼやけは、一般的にカメラの焦点不良によって起こります。画像を意図的にぼやけさせるには、ローパス周波数フィルタを適用します。

ま

マイル	5,280 フィート、つまり 1,609.344 メートルに相当する長さの単位。海里と区別するため、法廷マイルともいいます。「海里」も参照。
マシンビジョン	ビジョン検査タスクを実行する自動アプリケーション。
マスク FFT フィルタ	ユーザが指定したマスク（範囲）に含まれる周波数を削除します。

み

密度関数	線形ヒストグラムの各グレーレベルに対し、この関数は画像内で同じグレーレベルを持つピクセルの数を生成します。
------	---

め

明度	(1) カラーピクセルのグレースケール輝度で、そのピクセルの赤、緑、および青の最大値と最小値の平均として計算します。(2) 「ルマ」を参照。
メディアンフィルタ	近接ピクセルのメディアン値を各ピクセルに割り当てるローパスフィルタ。このフィルタは、オブジェクトの輪郭を不鮮明にすることなく、孤立したピクセルを取り除くことができます。
メモリバッファ	「バッファ」を参照。

も

文字照合	印刷された文字の質を検査するマシンビジョンアプリケーション。
文字認識	人間が読むテキストを機械によって読み取る能力。

元の画像	元の入力画像。
モフォロジー変換	画像内のオブジェクトの構造を抽出して変更します。そのような変換によって、オブジェクトの拡大（膨張）や縮小（収縮）、穴埋め、内包部の塗りつぶしやボータの平滑化などを行うことができます。主に、オブジェクトの輪郭を描画したり、数量検査解析用に画像を変換するために使用されます。

ゆ

ユークリッド距離	デカルト系の 2 ポイント間の最短距離。
----------	----------------------

ら

ラインゲージ	画像内のラインに沿って、非常に正確なサブピクセル精度によって選択したエッジ間の距離を測定します。たとえば、この関数はポイント間やエッジ間の距離の測定に使用することができます。また、この関数は画像全体で測定をステップ処理したり繰り返すこともできます。
ラインプロファイル	画像内のピクセルラインに沿ったグレーレベル分布を表します。
ラプラシアンフィルタ	ピクセル周辺の輝度変化をハイライトすることにより、画像内のオブジェクトの輪郭を抽出します。
ラベリング	バイナリ画像の各オブジェクトに固有の値を割り当てるプロセス。このプロセスは、画像内のオブジェクト数を確認すると、各オブジェクトを固有に識別するのに便利です。

り

リアルタイム	データを蓄積して後に処理するのではなく、データを集録しながら処理するイベントまたはシステムの特徴。
粒子	画像内の連結された領域またはピクセルのまとまりで、すべてのピクセルが同じ輝度レベルのもの。
粒子解析	画像内の粒子に関する情報を生成する一連の処理操作および解析関数。
領域	ソフトウェアが制御および定義する、集録ウィンドウまたはフレームの四角形部分。
領域しきい値	ユーザ指定範囲に入るオブジェクトをサイズによって検出します。

る

ルマ ビデオ画像における輝度情報。ルマ信号の振幅は、ビデオ信号の輝度に応じて変化し、白黒画像には完全に対応します。

れ

連結性 あるピクセルの周辺ピクセルの中で、近接ピクセルを形成するピクセルを定義します。

連結性 4 水平および垂直方向に隣接するピクセルのみが近接ピクセルとみなされません。

連結性 8 すべての隣接するピクセルが近接ピクセルとみなされます。

ろ

ローパス FFT フィルタ 画像の FFT 領域に存在する高周波数を削除または減衰します。

ローパス減衰 画像の周波数に線形減衰を適用します。この場合、最低周波数には減衰を適用せず、最高周波数には完全な減衰を適用します。

ローパス周波数フィルタ 画像の周波数領域に存在する高周波数を減衰します。ローパス周波数フィルタは、空間画像での輝度の急激な変化に関連する情報を抑圧します。

ローパストランケーション 一定の周波数より上の周波数をすべて削除します。

ローパスフィルタ 画像の輝度変化を減衰します。このようなフィルタを使用すると、詳細を取り除いてエッジをぼかすことにより、画像を滑らかにすることができます。

ロバーツフィルタ グレーレベルで輪郭、特に斜めのエッジを抽出（エッジ検出）します。

論理演算子 AND、NAND、OR、XOR、NOR、XNOR、差、マスク、平均、最大値、および最大値の各画像演算処理。

索引

数値

16ビット画像表示のマッピング方法、2-3
16ビット画像表示、マッピング方法、2-3

A

AIPD (ナショナルインスツルメンツの画像ファイル形式)、1-6

B

bitmap (BMP) ファイル形式、1-5
BMP (bitmap) ファイル形式、1-5

C

CIE-Lab カラースペース
RGB を CIE $L^*a^*b^*$ に変換する、1-20
概要、1-17
CMY カラースペース
RGB を CMY へ変換する、1-21
説明、1-18

D

dltd 傾斜フィルタ
定義済みのカーネル
ソーベルフィルタ、A-2
プレウィットフィルタ、A-1

E

Equalize 関数
基本概念、5-8
例、5-9

H

HSL カラースペース
RGB を HSL カラースペースにマッピングする、1-19
カラースペクトルを生成する、14-1
基本概念、1-17

I

IMAQ Vision 画像の内部表示形式、1-6

J

JPEG (Joint Photographic Experts Group) 形式、1-6

L

LCD 関数
アルゴリズムの制約、15-3
目的と使用方法、15-2
Lスケルトン関数、9-26

M

Mスケルトン関数、9-27

N

N次フィルタ
基本概念、5-31
数学的概念、5-35
例 (表)、5-31

P

PNG (portable network graphics) ファイル形式、1-5

R

RGB カラースペース
カラースペースを変換する
RGB からグレースケールへ、1-18
RGB および HSL、1-19
RGB および CIE $L^*a^*b^*$ 、1-20
RGB キューブ (図)、1-16
カラースペースを変換する
RGB および CMY、1-21
RGB および YIQ、1-21
基本概念、1-15
ROI。「関心領域」を参照

S

SKIZ 関数

基本概念、9-27

セグメント化関数との比較、9-29

Spoke 関数、11-12

T

tagged image file format (TIFF)、1-5

TIFF (tagged image file format)、1-5

X

X 軸の二乗の和、Y 軸の二乗の和、X 軸と Y 軸
の和パラメータ、粒子測定、10-8

X 軸の和パラメータ、粒子測定、10-8

Y

YIQ カラースペース

RGB を YIQ へ変換する、1-21

説明、1-18

Y 軸の和パラメータ、粒子測定、10-8

あ

穴埋め関数、バイナリモフォロジー、9-23

穴パラメータ

デジタル粒子の長さ、10-2

デジタル粒子の面積、10-1

アライメントアプリケーション

エッジ検出、11-4

カラーパターンマッチング、14-21

パターンマッチング、12-1

い

一次バイナリモフォロジー関数。「バイナリ
モフォロジー」を参照

色の 3 原色説、1-14

う

ウェブ

技術サポート、B-1

プロフェッショナルサービス、B-1

内側エッジ関数、バイナリモフォロ

ジー、9-14

内側傾斜関数、バイナリモフォロジー、9-14

右辺 X、底辺 Y パラメータ、デジタル粒
子、10-3

え

エッジ検出

エッジ検出の方法

サブピクセル精度、11-9

上級、11-8

シンプル、11-7

エッジの定義、11-4

エッジの特徴

一般的なモデル (図)、11-5

エッジの位置パラメータ、11-6

エッジの強度パラメータ、11-6

エッジの極パラメータ、11-6

エッジの長さパラメータ、11-6

概要、11-1

使用目的

アライメント、11-4

ゲーシング、11-2

検出、11-3

寸法測定、13-9

二次元検索領域

スポーク、11-12

同心レーキ関数、11-13

レーキ、11-11

パターンマッチングにおける、12-7

エッジベースの座標系関数

2つの検索領域、13-6

単一検索領域、13-5

円形関数、バイナリモフォロジー、9-30

円形検出関数、寸法測定、13-10

演算子

基本概念、6-1

算術、6-2

使用目的、6-1

論理および比較、6-2

エントロピーテクニク、自動二値化

概要、8-5

詳細の説明、8-7

お

オープン関数

グレースケールモフォロジー

説明、5-38

例、5-39

バイナリモフォロジー

基本概念、9-13

例、9-13

温度パレット、2-6

オンライン技術サポート、B-1

か

カーネルの定義

ガウシアンフィルタ、5-26

傾斜フィルタ、5-16

平滑化フィルタ、5-24

ラプラシアンフィルタ、5-21

解像度

視野、3-4

センササイズとセンサのピクセル数、3-5

定義、3-3

ピクセル解像度を決定する、3-3

ガウシアンフィルタ

カーネルの定義、5-26

定義済みのカーネル、A-7

例、5-26

加算演算子 (表)、6-2

カスタマー

技術サポート、B-1

トレーニング、B-1

プロフェッショナルサービス、B-1

画素、1-1

画像。「デジタル画像」を参照

IMAQ Vision 画像の内部表示形式、1-6

カラー、1-5

定義、1-1

ピクセル深度、1-2

平面の数、1-3

画像解析

輝度測定、4-6

ヒストグラム

解説、4-4

カラー画像のヒストグラム、4-5

基本概念、4-2

使用目的、4-1

線形ヒストグラム、4-3

ヒストグラムの解釈、4-4

累積ヒストグラム、4-3

ラインプロファイル、4-6

画像修正、キャリブレーションにおける、3-14

画像処理

空間フィルタ

詳細の説明、5-32

非線形フィルタ、5-15、5-27

分類のまとめ (表)、5-14

空間フィルタ処理

使用目的、5-13

グレースケールモフォロジー関数

オープン、5-38

概念および数学的演算、5-41

基本概念、5-36

クローズ、5-39

自動メディアン、5-40

収縮、5-37

使用目的、5-36

適正オープン、5-40

適正クローズ、5-40

膨張、5-37

コンポリューションカーネル、5-10

定義、2-1

ルックアップテーブル

Equalize、5-8

基本概念、5-1

指数およびガンマ補正、5-6

使用目的、5-1

対数と逆ガンマ補正、5-4

定義済み、5-3

ルックアップテーブル変換、5-1

画像タイプ

1ピクセル当たりのバイト数 (表)、1-3

カラー画像、1-5

グレースケール画像、1-4

複素画像、1-5

画像の解析。「画像解析」を参照

画像の空間解像度、1-2

画像の視覚化、定義、2-1

画像の鮮明度 (ビット深度)、1-2

画像の面積パラメータ、10-1

画像表示

16ビット画像表示のマッピング方法、2-3

- 基本概念、2-1
- 使用目的、2-1
- 表示モード、2-2
- 画像ファイルと形式、1-5
- 画像ボーダー
 - 定義、1-8
 - ピクセル値を指定する、1-8
 - ボーダーのサイズ、1-8
- 画像マスク
 - 画像マスクを使用する、1-11
 - 画像マスクを制限するためのオフセット、1-11
 - 異なるオフセットで適用する (図)、1-12
 - 定義、1-10
 - マスクの効果 (図)、1-11
 - 利用目的、1-10
- 画像理解、パターンマッチング、12-6
- カラー画像
 - 1 ピクセル当たりのバイト数 (表)、1-3
 - エンコード、1-5
 - カラー画像のヒストグラム、4-5
- カラー検査、14-7
- カラー識別
 - オブジェクトを分類する、14-12
 - 目的と使用方法、14-6、14-12
 - 例、14-7
- カラースペース
 - CIE-Lab カラースペース、1-17
 - CMY カラースペース、1-18
 - HSL カラースペース、1-17
 - RGB カラースペース、1-15
 - YIQ カラースペース、1-18
 - 一般的なカラースペースのタイプ、1-13
 - カラースペクトルを生成する、14-1
 - 色彩感覚、1-14
 - 定義、1-13
 - 変換
 - RGB および CIE $L^*a^*b^*$ 、1-20
 - RGB および HSL、1-19
 - RGB および CMY、1-21
 - RGB および YIQ、1-21
 - RGB からグレースケールへ、1-18
 - 利用目的、1-13
- カラースペースを変更する。「カラースペース」を参照
- カラースペクトル
 - HSL カラースペース、14-1
 - 概要、14-1
 - 生成する、14-3
- カラー二値化
 - 使用目的、8-9
 - 範囲
 - HSL 画像 (図)、8-11
 - RGB 画像 (図)、8-10
- カラーの位置検出
 - オブジェクトの識別、14-12
 - オブジェクトを分類する、14-12
 - 概要、14-10
 - 機能
 - 周辺光状態、14-15
 - パターンの向きおよび複数のインスタンス、14-14
 - ぼやけとノイズの状態、14-15
 - 基本概念、14-15
 - 検査、14-11
 - 使用目的、14-10
- カラーパターンマッチング
 - 機能
 - 周辺光状態、14-22
 - パターンの向きおよび複数のインスタンス、14-21
 - ぼやけとノイズの状態、14-22
 - 基本概念
 - カラーの位置検出とグレースケールパターンマッチング、14-24
 - カラーマッチングとカラーの位置検出、14-23
 - グレースケールパターンマッチング、14-23
 - 使用目的
 - アライメント、14-21
 - ゲー징、14-20
 - 検査、14-20
- カラー分布
 - 学習する、14-9
 - 比較する、14-9
- カラーマッチング
 - 概要、14-6
 - カラー分布を学習する、14-9
 - カラー分布を比較する、14-9
 - 基本概念、14-9

使用目的
 カラー検査、14-7
 カラー識別、14-6

関心領域
 関数、2-1
 キャリブレーション ROI、3-12
 キャリブレーションの修正領域、3-15
 使用目的、2-9
 定義する、2-10
 輪郭のタイプ (表)、2-10

慣性 I_{xx} , I_{yy} , I_{xy} のモーメントパラメータ、10-7

き

幾何学測定、13-13
 技術サポート、B-1
 技術サポートデータベース、B-1
 基準パターン
 一般的な基準パターンの例 (図)、12-2
 定義、12-1

輝度しきい値、8-2
 輝度測定
 使用目的、4-6
 デンシトメトリパラメータ、4-7

輝度、定義、1-17
 キャリブレーション。「空間キャリブレーション」を参照
 キャリブレーション関数のエラーマップ出力、3-12
 キャリブレーション関数の品質スコア出力、3-12
 キャリブレーション用視点アルゴリズム、3-11
 キャリブレーション用非線形アルゴリズム、3-11
 距離関数、バイナリモフォロジー、9-29
 距離測定、13-12
 近接ピクセル、定義、1-8

く

空間キャリブレーション
 アルゴリズム、3-11
 概要、3-7
 画像修正、3-14
 キャリブレーションプロセス、3-8
 座標系、3-9

座標系を再定義する、3-17
 修正領域、3-15
 使用目的、3-7
 シンプルキャリブレーション、3-16
 スケールモード、3-14
 定義、3-7
 品質情報、3-12
 空間キャリブレーションの品質情報、3-12
 空間周波数、7-1
 空間フィルタ
 カテゴリ、5-14
 線形フィルタ
 詳細の説明、5-32
 定義、5-14
 非線形フィルタ
 N 次フィルタ、5-31
 ガウシアンフィルタ、5-26
 傾斜フィルタ、5-15、5-29
 シグマフィルタ、5-30
 詳細の説明、5-33
 ソーベルフィルタ、5-28
 微分フィルタ、5-29
 プレウィットフィルタ、5-27
 平滑化フィルタ、5-24
 メディアンフィルタ、5-30
 ラプラシアンフィルタ、5-20
 ローパスフィルタ、5-30
 ロバーツフィルタ、5-29
 分類のまとめ (表)、5-14

空間フィルタ処理
 使用目的、5-13
 クラスタ化テクニック、自動二値化
 詳細の説明、8-7
 概要、8-3
 例、8-4

グレースケール画像
 1 ピクセル当たりのバイト数 (表)、1-3
 RGB からグレースケールへ変換する、1-18
 ピクセルエンコード、1-4

グレースケールパターンマッチング
 カラーの位置検出とグレースケールパターンマッチング、14-24
 メソッド、14-23

- グレースケールモフォロジー関数
 - オープン
 - オープンとクローズの例、5-39
 - 概要、5-38
 - 概念および数学的演算、5-41
 - 基本概念、5-36
 - クローズ
 - オープンとクローズの例、5-39
 - 概要、5-39
 - 自動メディアン
 - 概念および数学的演算、5-43
 - 概要、5-40
 - 収縮
 - 概念および数学的演算、5-41
 - 概要、5-37
 - 収縮と膨張の例、5-37
 - 使用目的、5-36
 - 適正オープン
 - 概念および数学的演算、5-42
 - 概要、5-40
 - 適正クローズ
 - 概念および数学的演算、5-42
 - 概要、5-40
 - 膨張
 - 概念および数学的演算、5-41
 - 概要、5-37
 - 収縮と膨張の例、5-37
 - グレーレベル値
 - バイナリパレット (表)、2-8
 - ピクセル、1-1
 - グレーレベルパレット、2-5
 - クローズ関数
 - グレースケールモフォロジー
 - 説明、5-39
 - 例、5-39
 - バイナリモフォロジー
 - 基本概念、9-13
 - 例、9-14
 - クロマ、定義、1-17
- け**
- 傾斜パレット、2-7
 - 傾斜フィルタ
 - 線形
 - エッジの抽出とエッジのハイライト、5-17
 - エッジの太さ、5-19
 - カーネルの定義、5-16
 - 定義、5-15
 - フィルタ軸および方向、5-16
 - 例、5-15
 - 非線形
 - 数学的概念、5-34
 - 定義、5-29
 - 傾斜フィルタでエッジの輪郭を明確にする
 - エッジの抽出とハイライト、5-17
 - エッジの太さ、5-19
 - 形状の等価、デジタル粒子
 - 楕円長軸、10-5
 - 四角形辺比率、10-7
 - 楕円短軸、10-6
 - 楕円率、10-6
 - 長方形の短辺、10-6
 - 長方形の長辺、10-6
 - 等価楕円短軸、10-5
 - 形状の特徴
 - Waddel ディスクの直径、10-8
 - 伸長因子、10-7
 - 水力半径、10-7
 - 分散度、10-7
 - ヘイウッド円形度、10-7
 - 形状マッチング
 - 基本概念、12-10
 - 使用目的、12-10
 - 寸法測定、13-12
 - 例、12-11
 - 計測器ドライバ、B-1
 - 計測器リーダ
 - LCD 関数、15-2
 - 使用目的、15-1
 - バーコード、15-3
 - メータ関数、15-1
 - ゲージングアプリケーション。「寸法測定」も参照
 - エッジ検出、11-2
 - カラーパターンマッチング、14-20
 - パターンマッチング、12-1
 - 検査
 - カラー検査、14-7
 - カラーの位置検出、14-11

カラーパターンマッチング、14-20
 パターンマッチング、12-1
 減算演算子 (表)、6-2
 減衰
 ハイパス FFT フィルタ、7-9
 ローパス FFT フィルタ、7-7
 弦と軸、デジタル粒子
 平均垂直切片、10-4
 弦 X の平均、10-4
 弦 Y の平均、10-4
 最長セグメント長、10-4
 粒子の方向、10-4
 最大切片、10-4
 弦平均 X および弦平均 Y パラメータ、デジタル粒子、10-4

こ

光学表現、FFT 表示、7-5
 構造要素
 値、9-3
 基本概念、9-2
 サイズ、9-2
 収縮関数の効果 (表)、9-12
 使用目的、9-1
 ピクセルフレームの形状、9-4
 膨張関数の効果 (表)、9-13
 高速フーリエ変換 (FFT)。「周波数フィルタ」も参照
 FFT 表現
 光学表現、7-5
 標準表現、7-3
 FFT 表示、7-13
 定義、7-1
 フーリエ変換の概念、7-12
 コントラスト
 コントラスト不足を判定するためのヒストグラム、4-2
 設定する、3-5
 コンボリューション
 種類、5-13
 定義、5-13
 コンボリューションカーネル。「線形フィルタ」も参照
 カーネルの例 (図)、5-11
 基本概念、5-10

サイズ、5-13
 使用目的、5-10
 フィルタ処理の仕組み (図)、5-11
 ボーダー部分のピクセルのフィルタ処理 (図)、5-12

さ

最小値演算子 (表)、6-3
 最大切片パラメータ、デジタル粒子、10-4
 最大値演算子 (表)、6-3
 最長セグメント左辺 X、上辺 Y パラメータ、デジタル粒子、10-3
 最長セグメントパラメータ、デジタル粒子、10-4
 彩度
 定義、1-17
 ヒストグラムで検出する、4-1
 作動距離、定義、3-3
 座標系
 空間キャリブレーション
 起点と角度 (図)、3-10
 再定義する、3-17
 軸方向 (図)、3-10
 寸法測定、13-3
 エッジベースの関数、13-5
 使用目的、13-3
 定義手順、13-4
 パターンマッチングベースの関数、13-7
 左辺 X、上辺 Y パラメータ、デジタル粒子、10-3
 サポート
 技術、B-1
 算術演算、6-2
 サンプルコード、B-1

し

四角形辺比率パラメータ、10-7
 色彩の感覚、1-14
 色相、定義、1-17
 軸。「弦と軸」を参照
 シグマフィルタ
 基本概念、5-30
 数学的概念、5-35

- 指数およびガンマ補正
 - 基本概念、5-6
- 指数関数とガンマ補正
 - まとめ (表)、5-3
 - 例、5-7
- システムインテグレーションサービス、B-1
- システムのセットアップ。「デジタル画像」も参照
 - 解像度、3-3
 - 基本概念、3-1
 - 基本パラメータ (図)、3-2
 - 高品質の画像を集録する、3-3
 - コントラスト、3-5
 - 視点、3-5
 - 被写界深度、3-5
 - 歪み、3-7
- 視点
 - オブジェクトに対するカメラアングル、3-6
 - 視点エラーと歪みエラー (図)、3-6
- 自動二値化
 - エントロピー
 - 概要、8-5
 - 詳細の説明、8-7
 - 概要、8-3
 - クラスタ化
 - 概要、8-3
 - 詳細の説明、8-7
 - 例、8-4
 - 相互分散
 - 概要、8-5
 - 詳細の説明、8-8
 - テクニク、8-6
 - メトリック
 - 概要、8-5
 - 詳細の説明、8-8
 - モーメント
 - 概要、8-5
 - 詳細の説明、8-9
- 自動メディアン関数
 - グレースケールモフォロジー
 - 概念および数学的演算、5-43
 - 概要、5-40
 - バイナリモフォロジー、9-22
- 視野
 - 定義、3-3
 - ピクセル解像度と視野の関係 (図)、3-4
- 周期パレット (図)、2-9
- 収縮関数
 - グレースケールモフォロジー
 - 概念および数学的演算、5-41
 - 目的と使用方法、5-37
 - 例、5-37
 - バイナリモフォロジー
 - 基本概念、9-11
 - 構造要素の効果 (表)、9-12
 - 例、9-12
- 重心 (X, Y)、デジタル粒子の座標、10-3
- 修正領域、キャリブレーションにおける、3-15
- 周波数処理、7-1
- 周波数フィルタ
 - FFT 表現
 - 光学表現、7-5
 - 標準表現、7-3
 - 概要、7-1
 - 高速フーリエ変換の概念
 - FFT 表現、7-3
 - FFT 表示、7-13
 - 概要、7-3
 - フーリエ変換、7-12
 - 使用目的、7-2
 - 定義、7-1
 - ハイパス FFT フィルタ
 - 概要、7-2
 - 減衰、7-9
 - トランケーション、7-10
 - 例、7-10
 - マスク FFT フィルタ
 - 概要、7-2
 - 目的と使用方法、7-11
 - ローパス FFT フィルタ
 - 概要、7-2
 - 減衰、7-7
 - トランケーション、7-7
 - 例、7-8
- 周辺光状態
 - カラーの位置検出ツール、14-15
 - カラーパターンマッチング、14-22
 - パターンマッチング、12-4

上級バイナリモフォロジー関数。「バイナリモフォロジー」を参照
 乗算演算子 (表)、6-2
 照明状態、パターンマッチング、12-4
 除算演算子 (表)、6-2
 診断リソース、B-1
 伸長因子パラメータ、10-7
 真理値表、6-4

す

水力半径パラメータ、10-7
 スケールモード、キャリブレーションにおける、3-14
 スケールに影響されないマッチング、12-5
 スケルトン関数
 L スケルトン、9-26
 M スケルトン、9-27
 SKIZ、9-27
 セグメント化関数と SKIZ 関数の比較、9-29

寸法測定

概要、13-1
 関心領域で画像の部分を検出する、13-2
 座標系
 エッジベースの関数、13-5
 使用目的、13-3
 定義手順、13-4
 パターンマッチングベースの関数、13-7
 使用目的、13-1
 測定値を判定する、13-3
 測定を行う
 解析幾何学、13-13
 概要、13-2
 距離測定、13-12
 ラインフィット、13-14
 特徴または測定ポイントを検出するエッジベースの特徴、13-9
 概要、13-2
 形状ベースの特徴、13-12
 ラインおよび円形の特徴、13-10

せ

正方形ピクセルフレーム、9-6
 世界各地での技術サポート、B-2

セグメント化関数
 SKIZ 関数との比較、9-29
 基本概念、9-28
 絶対差分演算子 (表)、6-2
 線形ヒストグラム、4-3
 線形フィルタ
 ガウシアンフィルタ
 定義済みのカーネル、A-7
 傾斜フィルタ
 定義済みのカーネル空間フィルタ
 非線形フィルタ
 定義済みのカーネル、A-1
 平滑化フィルタ
 定義済みのカーネル、A-6
 ラプラシアンフィルタ
 定義済みのカーネル、A-5
 センササイズ
 センサのピクセル数、3-5
 定義、3-3

そ

相互相関、パターンマッチング
 詳細の説明、12-8
 関連の手順 (図)、12-9
 相互相関、パターンマッチング。「カラーパターンマッチング、エッジ検出」も参照
 概要、12-5
 相互分散テクニック、自動二値化、8-8
 ソーベルフィルタ
 基本概念、5-28
 数学的概念、5-33
 定義済みのカーネル、A-2
 例、5-28
 外側エッジ関数、バイナリモフォロジー、9-14
 外側傾斜関数、バイナリモフォロジー、9-14
 ソフトウェアドライバ、B-1

た

対称軸、傾斜フィルタ、5-16
 対数と逆ガンマ補正
 基本概念、5-4
 まとめ (表)、5-3
 例、5-4

楕円長軸パラメータ、10-5
 楕円率パラメータ、10-6
 高さパラメータ、デジタル粒子、10-2
 ダニエルソン関数、9-30
 多様な測定値、デジタル粒子、10-8

ち

長方形の短辺パラメータ、10-6
 長方形の長辺パラメータ、10-6

て

定義済みのカーネル
 ガウシアンカーネル、A-7
 傾斜カーネル
 ソーベルフィルタ、A-2
 プレウィットフィルタ、A-1
 平滑化カーネル、A-6
 ラプラシアンカーネル、A-5
 定義済みのルックアップテーブル、5-3
 適正オープン関数
 グレースケールモフォロジー
 概念および数学的演算、5-42
 概要、5-40
 バイナリモフォロジー、9-21
 適正クローズ関数
 グレースケールモフォロジー
 概念および数学的演算、5-42
 概要、5-40
 バイナリモフォロジー、9-21
 デジタル画像
 IMAQ Vision 画像の内部表示形式、1-6
 画像タイプ
 1ピクセル当たりのバイト数
 (表)、1-3
 グレースケール画像、1-4
 複素画像、1-5
 画像ファイル形式、1-5
 画像ボーダー、1-8
 画像マスク、1-10
 カラースペース
 CIE-Lab カラースペース、1-17
 CMY カラースペース、1-18
 HSL カラースペース、1-17
 RGB カラースペース、1-15
 YIQ カラースペース、1-18

一般的なカラースペースのタイプ、1-13
 色彩感覚、1-14
 定義、1-13
 変換

 RGB および CIE L*a*b*、1-20
 RGB および CMY、1-21
 RGB および HSL、1-19
 RGB および YIQ、1-21
 RGB からグレースケール
 へ、1-18

 利用目的、1-13

定義、1-1

プロパティ

 概要、1-2

 画像の解像度、1-2

 画像の鮮明度、1-2

 平面の数、1-3

 画像ボーダー、1-8

デジタル画像処理、定義、1-1

デジタル粒子

 一次測定値の定義、10-8

 形状の等価、10-5

 形状の特徴、10-7

 座標、10-2

 使用目的、10-1

 多様な測定値、10-8

 長さ、10-2

 面積、10-1

 誘導測定値 (表)、10-9

 弦と軸、10-4

デジタル粒子の座標

 右辺 X、底辺 Y、10-3

 最長セグメント X および最長セグメント
 Y、10-3

 左辺 X、上辺 Y、10-3

 重心 X と重心 Y、10-3

デジタル粒子の面積

 穴の数、10-1

 穴の総面積、10-1

 画像の面積、10-1

 ピクセル数、10-1

 比率、10-1

 粒子の総面積、10-1

 粒子の面積、10-1

デジタル粒子の誘導測定値 (表)、10-9
 デンシトメトリパラメータ、4-7
 電話による技術サポート、B-2

と

等価楕円短軸パラメータ、10-5
 同心レーキ関数、11-13
 ドキュメント
 オンラインライブラリ、B-1
 特徴、パターンマッチング、12-1
 凸関数、バイナリモフォロジー、9-31
 ドライブ
 計測器、B-1
 ソフトウェア、B-1
 トラブルシューティングリソース、B-1
 トランケーション
 ハイパス FFT フィルタ、7-10
 ローパス FFT フィルタ、7-7
 トレーニング、B-1
 カスタマー、B-1

な

長さパラメータ、デジタル粒子
 穴の周囲、10-2
 高さ、10-2
 幅、10-2
 粒子の周囲、10-2
 ナショナルインスツルメンツ
 カスタマーレーニング、B-1
 技術サポート、B-1
 システムインテグレーションサービ
 ス、B-1
 世界各地の営業所、B-2
 プロフェッショナルサービス、B-1
 ナショナルインスツルメンツの画像ファイル
 形式 (AIPD)、1-6
 ナショナルインスツルメンツへのお問合わせ、B-2

に

虹パレット、2-7
 二値化
 概要、8-1
 カラー、8-9

輝度しきい値、8-2
 自動
 エントロピー、8-5、8-7
 クラスタ化、8-3、8-7
 詳細の説明、8-6
 相互分散、8-5、8-8
 テクニック、8-6
 メトリック、8-5、8-8
 モーメント、8-5、8-9
 使用目的、8-1
 例、8-2

の

ノイズ。「ぼやけとノイズの状態」を参照

は

バーコードリーダー
 アルゴリズムの制約、15-4
 目的と使用方法、15-3
 排他的論理和演算子 (表)、6-3
 バイナリ形状マッチング、12-10
 バイナリパレット
 グレーレベル値 (表)、2-8
 周期パレット (図)、2-9
 バイナリモフォロジー
 一次モフォロジー関数
 内側傾斜、9-14
 オープンおよびクローズ、9-13
 自動メディアン、9-22
 収縮および膨張、9-11
 使用目的、9-10
 外側傾斜、9-14
 適正オープン、9-21
 適正クローズ、9-21
 ヒットミス、9-15
 太線化、9-19
 細線化、9-17
 概要、9-1
 構造要素
 値、9-3
 基本概念、9-2
 サイズ、9-2
 使用目的、9-1
 ピクセルフレームの形状、9-4

上級バイナリ関数

- 穴埋め、9-23
- 円形、9-30
- 基本概念、9-23
- 距離、9-29
- 使用目的、9-22
- スケルトン、9-26
- セグメント化、9-28
- ダニエルソン、9-30
- 凸、9-31
- 分離、9-25
- ボーダー、9-23
- ラベリング、9-23
- ローパスフィルタとハイパスフィルタ、9-24

ピクセルフレームの形状

- 概要、9-4
- 正方形フレーム、9-6
- 例 (図)、9-4
- 六角形フレーム、9-7

連結性

- 基本概念と例、9-7
- 詳細の説明、9-9
- 使用目的、9-7
- 連結性 4、9-9
- 連結性 8、9-9

ハイパス周波数 (FFT) フィルタ

- 概要、7-2
- 減衰、7-9
- トランケーション、7-10
- 例、7-10

ハイパスフィルタ

- 定義、5-14
- バイナリモフォロジー
 - 基本概念、9-24
 - 効果 (表)、9-24
 - 例、9-25
- 分類 (表)、5-14

パターンの向きおよび複数のインスタンス

- カラーの位置検出ツール、14-14
- カラーパターンマッチング、14-21
- パターンマッチング、12-3

パターンマッチング。「カラーパターンマッチング、エッジ検出」も参照

- 新しいテクニック、12-6
- 概要、12-1

機能

- 周辺光状態、12-4
- パターンの向きおよび複数のインスタンス、12-3
- ぼやけとノイズの状態、12-4
- グレースケールパターンマッチング
 - カラーの位置検出とグレースケールパターンマッチング、14-24
 - メソッド、14-23
- 形状マッチング、12-10
- 従来からのテクニック、12-5
- 使用目的、12-1
- スケールに影響されないマッチング、12-5
- 寸法測定に使用する座標系、13-7
- 相互相関
 - 概要、12-5
 - 詳細の説明、12-8
- バイナリ形状マッチング、12-5、12-10
- ピラミッドマッチング、12-5
- 幅パラメータ、デジタル粒子、10-2
- パレット
 - 温度パレット、2-6
 - 基本概念、2-5
 - グレイレベルパレット、2-5
 - 傾斜パレット、2-7
 - 使用目的、2-4
 - 定義、2-4
 - 虹パレット、2-7
 - バイナリパレット、2-8

ひ

比較演算。「論理および比較演算」を参照

ピクセル

- 画像ボーダーの値、1-8
- 近接ピクセル、1-8
- 空間座標、1-1
- グレイレベル値、1-1
- ピクセル数
 - センサの、3-5
 - 粒子の面積で、10-1

ピクセル解像度

- 決定する、3-3
- 視野との関係、3-4
- 定義、3-3

- ピクセル深度、1-2
- ピクセルフレームの形状
 - 概要、9-4
 - 正方形フレーム、9-6
 - 例 (図)、9-4
 - 六角形フレーム、9-7
- 被写界深度
 - 設定する、3-5
 - 定義、3-3
- ヒストグラム
 - 解釈、4-4
 - カラー画像のヒストグラム、4-5
 - 基本概念、4-2
 - 使用目的、4-1
 - 線形ヒストグラム、4-3
 - 定義、4-1
 - ヒストグラムの解釈、4-4
 - 累積ヒストグラム、4-3
- ヒストグラムの解釈、4-4
- ヒストグラムの目盛、4-4
- 歪み
 - 視点エラーと歪みエラー (図)、3-6
 - 説明、3-7
- 非線形傾斜フィルタ
 - 定義、5-29
- 非線形フィルタ
 - N次フィルタ
 - 概要、5-31
 - 効果 (表)、5-31
 - 数学的概念、5-35
 - 概要、5-15
- ガウシアンフィルタ
 - カーネルの定義、5-26
 - 例、5-26
- 傾斜フィルタ
 - エッジの抽出とエッジのハイライト、5-17
 - エッジの太さ、5-19
 - カーネルの定義、5-16
 - 概要、5-29
 - 数学的概念、5-34
 - フィルタ軸および方向、5-16
 - 例、5-15
- シグマフィルタ
 - 概要、5-30
 - 数学的概念、5-35
- 詳細の説明、5-32、5-33
- ソーベルフィルタ
 - 数学的概念、5-33
 - 説明、5-28
 - 定義済みのカーネル、A-2
 - 例、5-28
- 微分フィルタ
 - 概要、5-29
 - 数学的概念、5-34
- プレウィットフィルタ
 - 数学的概念、5-33
 - 説明、5-27
 - 定義済みのカーネル、A-1
 - 例、5-28
- 分類 (表)、5-14
- 平滑化フィルタ
 - カーネルの定義、5-24
 - 例、5-24
- メディアンフィルタ
 - 概要、5-30
 - 数学的概念、5-35
- ラプラシアンフィルタ
 - カーネルの定義、5-21
 - 輪郭の抽出とハイライト、5-21
 - 輪郭の太さ、5-23
 - 例、5-20
- ローパスフィルタ
 - 概要、5-30
 - 数学的概念、5-35
- ロバーツフィルタ
 - 概要、5-29
 - 数学的概念、5-34
- ビット深度 (画像の鮮明度)、1-2
- ヒットミス関数、バイナリモフォロジー
 - 基本概念、9-15
 - 使用の方法 (表)、9-17
 - 例、9-16
- 否定論理積演算子 (表)、6-3
- 否定論理和演算子 (表)、6-3
- 非破壊的オーバーレイ
 - 基本概念、2-11
 - 使用目的、2-11
- 微分フィルタ
 - 数学的概念、5-34
 - 定義、5-29

表示

画像表示

- 16ビット画像表示のマッピング方法、2-3
- 基本概念、2-1
- 使用目的、2-1
- 表示モード、2-2

関心領域

- 使用目的、2-9
- 定義する、2-10
- 輪郭のタイプ (表)、2-10

パレット

- 温度パレット、2-6
- 基本概念、2-5
- グレーレベルパレット、2-5
- 傾斜パレット、2-7
- 使用目的、2-4
- 虹パレット、2-7
- バイナリパレット、2-8

非破壊的オーバーレイ、2-11

標準表現、FFT 表示、7-3

ピラミッドマッチング、12-5

比率パラメータ、10-1

ふ

フィルタ。「コンボリューションカーネル」、「周波数フィルタ」、「空間フィルタ」を参照。

フーリエ変換、7-12

フーリエ変換。「高速フーリエ変換 (FFT)」も参照

複素画像

- 1ピクセル当たりのバイト数 (表)、1-3
- 定義、1-5

太線化関数、バイナリモフォロジー

- 基本概念、9-19
- 例、9-20

プレウィットフィルタ

- 基本概念、5-27
- 数学的概念、5-33
- 定義済みのカーネル、A-1
- 例、5-28

プログラミング例、B-1

プロフェッショナルサービス、B-1

分散度パラメータ、10-7

分離関数、バイナリモフォロジー、9-25

へ

ヘイウッド円形度パラメータ、10-7

平滑化フィルタ

- カーネルの定義、5-24
- 定義済みのカーネル、A-6
- 例、5-24

平均演算子 (表)、6-3

平均垂直切片、デジタル粒子、10-4

平面の数、1-3

平面、画像における数、1-3

ヘルプ

- 技術サポート、B-1
- プロフェッショナルサービス、B-1

ほ

方向、傾斜フィルタ、5-16

膨張関数

- グレースケールモフォロジー
- 概念および数学的演算、5-41
- 目的と使用方法、5-37
- 例、5-37

バイナリモフォロジー

- 基本概念、9-11
- 構造要素の効果 (表)、9-13
- 例、9-12

ボーダ。「画像ボーダー」を参照

ボーダー関数、バイナリモフォロジー、9-23

補正投影 X パラメータ、粒子測定、10-8

補正投影 Y パラメータ、粒子測定、10-8

細線化関数、バイナリモフォロジー

- 基本概念、9-17
- 例、9-18

ぼやけとノイズの状態

- カラーの位置検出ツール、14-15
- カラーパターンマッチング、14-22
- パターンマッチング、12-4

ま

マスク。「画像マスク;要素を構成する」を参照

マスク FFT フィルタ

- 概要、7-2
- 目的と使用方法、7-11

マスク演算子 (表)、6-3

マッピング方法、16ビット画像表示の、2-3

め

- メータ関数
 - アルゴリズムの制約、15-2
 - 目的と使用方法、15-1
- メディアンフィルタ
 - 基本概念、5-30
 - 数学的概念、5-35
- メトリックテクニック、自動二値化
 - 詳細の説明、8-8
 - 概要、8-5

も

- モーメントテクニック、自動二値化
 - 概要、8-5
 - 詳細の説明、8-9
- モジュロ演算子 (表)、6-2
- モフォロジー関数。「バイナリモフォロジー、グレースケールモフォロジー関数」を参照

よ

- よくある質問、B-1

ら

- ライン検出関数、寸法測定、13-10
- ラインフィット関数、寸法測定
 - データセットとフィットライン (図)、13-15
 - 平均二乗距離の計算 (図)、13-16
 - 最も強いラインフィット (図)、13-17
- ラインプロファイル、4-6
 - 使用目的、4-6
- ラプシアンフィルタ
 - カーネルの定義、5-21
 - 定義済みのカーネル、A-5
 - 輪郭の抽出とハイライト、5-21
 - 輪郭の太さ、5-23
 - 例、5-20
- ラベリング関数、バイナリモフォロジー、9-23

り

- 粒子解析
 - 基本概念、III-2
 - 使用目的、III-2
 - パラメータ、III-3
- 粒子の周囲パラメータ、10-2
- 粒子の総面積パラメータ、10-1
- 粒子の方向パラメータ、10-4
- 粒子面積パラメータ、10-1
- 粒子、定義、III-1
- 輪郭
 - 抽出とハイライト、5-21
 - 太さ、5-23

る

- 累積ヒストグラム、4-3
- ルックアップテーブル
 - Equalize、5-8
 - 指数関数とガンマ補正、5-6
 - 対数と逆ガンマ補正、5-4
 - 定義済みのルックアップテーブル、5-3
- ルックアップテーブル変換
 - 使用目的、5-1
 - 基本概念、5-1
 - 例、5-2

れ

- レーキ関数、11-11
- 連結性
 - 基本概念と例、9-7
 - 詳細の説明、9-9
 - 使用目的、9-7
 - 連結性 4、9-9
 - 連結性 8、9-9
- レンズの焦点距離、設定する、3-5

ろ

- ローパス周波数 (FFT) フィルタ
 - 概要、7-2
 - 減衰、7-7
 - トランケーション、7-7
 - 例、7-8

- ローパスフィルタ
 - 定義、5-14
 - バイナリモフォロジー
 - 基本概念、9-24
 - 効果 (表)、9-24
 - 例、9-25
 - 非線形
 - 基本概念、5-30
 - 数学的概念、5-35
 - 分類 (表)、5-14
- 六角形ピクセルフレーム、9-7
- ロバーツフィルタ
 - 数学的概念、5-34
 - 定義、5-29
- 論理および比較演算
 - 演算子リスト (表)、6-3
 - 真理値表、6-4
 - バイナリ画像マスクで使用する (表)、6-3
 - 目的と使用方法、6-2
 - 例、6-5
- 論理差分演算子 (表)、6-3
- 論理積演算子 (表)、6-3
- 論理和演算子 (表)、6-3