

I -04 画像解析

予習レポートおよび本実験の準備

- (1) 本実験では、個人が所有するノートパソコン(OSはWindows 2000/XP/Vista/7 日本語版のみ)を用いる。実験当日は必ずノートパソコンを持参すること。
- (2) 実験(その1)1~3 および 実験(その2)1~3を予習レポートとしてまとめること。
- (3) 実験開始時まで、本実験の最後のページにある実験メモをコピー、または下記に示すホームページの講義のページよりダウンロードして印刷しておくこと。
<http://sensor.eng.shizuoka.ac.jp/~yamasita/index-j.html>
- (4) 実験開始時まで、実験メモに学籍番号、班、氏名、実験日等を記入しておくこと。
- (5) 実験準備(画像ファイルやプログラムのダウンロード等、本実験資料の最後の内容)を予習レポート作成時に一緒に行っておくこと。

(その1) 粒子の計数

1. 目的

画像解析は、「見つける」「数える」「形を測る」「識別する」「記号・文字を読む」など複雑かつ多様な作業を画像処理により実現する技術であり、自然観測・生産現場・医療をはじめ様々な分野で利用されている。実験(その1)では、顕微鏡で見た粒子画像において粒子の数を「数える」ことを目的とする。

2. 画像処理手法

粒子の顕微鏡画像(図1)において、粒子数を計測する。本実験では、図2に示す手順により、画像処理技術を用いて粒子の自動計数を実施する。

実験(その1)では、2値化処理を用いて粒子と背景を分離し、粒子のある場所の個数を数えることにより粒子の計数を行う。粒子と背景の分離を行うための前処理として、モノクロ化、シェーディングの補正を行う。また、粒子の個数を数える際の誤差を小さくするために、ノイズ除去と粒子同士の分離を行う。

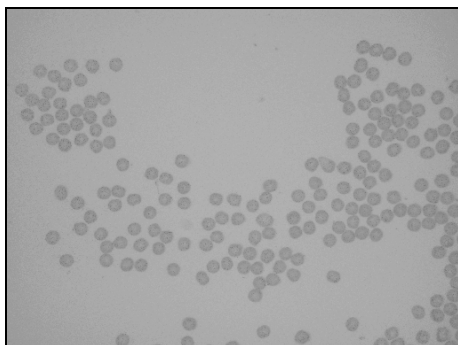


図1 粒子の顕微鏡画像の例

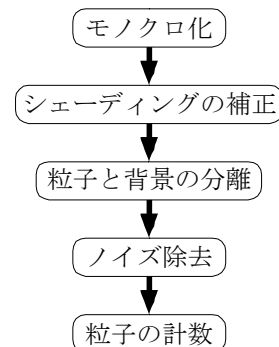


図2 粒子計数のフローチャート

① モノクロ化

デジタル画像は、画像が格子状に分割され、格子点毎に数値として表現された 2 次元平面上の濃淡分布の情報として扱われる。1 つ 1 つの格子点は画素 (Pixel) と呼ばれ、各画素の濃淡値は数値化されている。モノクロ画像では、各画素の濃淡値は 0 から $2^n - 1$ まであり、 $n=8$ のとき 256 階調となる。カラー画像は、赤 (R)・緑 (G)・青 (B) の RGB 成分から構成されており、各色が 256 階調で表現されている場合、最大表示色は $1677216 (=256^3)$ 色となる。

ここでは、カラー画像で与えられた粒子の顕微鏡画像のモノクロ化を行う。モノクロ画像の濃淡値 M は、カラー画像の R, G, B の値から、(1)式で算出される。

$$M = 0.299R + 0.587G + 0.114B \quad (1)$$

② シェーディングの補正

レンズを用いた撮像系では、輝度が一様なシーンを撮影しても画像の中心から離れるに従って明るさが低下すること (周辺減光) があり、画像中の場所によって明るさが均一でないことが多い。この不均一性をシェーディングと呼ぶ。シェーディングの補正を行うことにより、画像中の明るさは均一に近くなる。ここでは、(2)式に示すレンズのシェーディングのコサイン 4 乗則により、補正を行う。

$$E = L \frac{\pi}{4} \left(\frac{d}{f} \right) \cos^4 \alpha \quad (2)$$

ただし、 E は単位面積当たりの明るさ、 L は単位立方体当たりの輝度エネルギー、 d はレンズの直径、 f はレンズ中心から画像面までの距離 (像距離と呼ぶ)、 α は入射光がレンズの光軸となす角度である。ちなみに、 d/f はレンズがどれだけ明るく光を通すかを表し、その逆数は F ナンバーと呼ばれる。

ここで、画素の画像中心からの距離を r とすると、 α は(3)式で表される。

$$\alpha = \tan^{-1} \frac{r}{f} \quad (3)$$

画像には(2)式の影響があるとし、その影響を打ち消すことによりシェーディングの補正を行う。

③ 粒子と背景の分離

濃淡値に基準値 (閾値^{しきい}と呼ぶ) を設け、各画素の濃淡値が閾値よりも明るい、暗いかによってその画像を白 (0) または黒 (1) の 2 値へ変換することを 2 値化処理と呼ぶ (図 3)。入力画像 $f(i, j)$ の閾値 t による 2 値化画像 $g_t(i, j)$ は以下の(4)式の操作によって得られる (i, j は画像中の座標値)。

$$g_t(i, j) = \begin{cases} 1, & f(i, j) \leq t \text{ のとき} \\ 0, & f(i, j) > t \text{ のとき} \end{cases} \quad (4)$$

画像の 2 値化処理は、物体と背景とを分離するのに最も広く使用されている手法の 1 つである。計測する対象の濃淡分布が、背景の濃淡分布よりも十分に高い (もしくは低い) 場合、2 値化処理により背景から対象を分離することができる。

閾値は、濃淡ヒストグラムを調べることで決定する。ヒストグラムでは、濃淡値が横軸、出現画素数が縦軸にとられており、本実験で用いる粒子画像では、図 4 に示すように谷の部分の濃淡値を閾値とすることができる (モード法)。

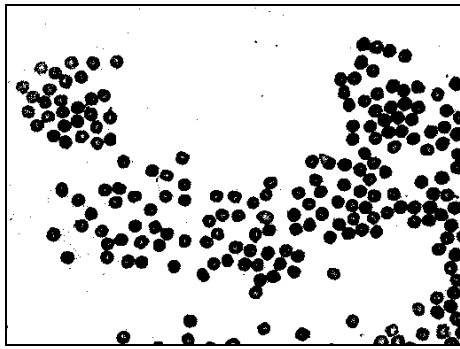


図3 2値化画像

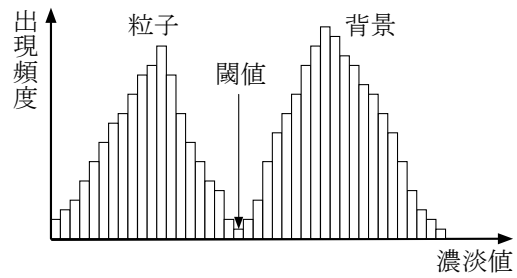


図4 ヒストグラムと閾値

④ ノイズ除去

2段階の処理によりノイズを除去する。

まず、2値化画像の孤立点除去によるノイズ除去を行う。ここでは、1画素だけ孤立して存在する画素は細かいノイズであるとみなし、除去する(図5)。具体的には、黒画素(粒子部分)の8-近傍がすべて白画素の場合には、その黒画素を白画素に変更する。また、白画素(背景部分)の4-近傍が黒画素の場合には、黒画素に変更する。ここで、4-近傍とは着目画素の上下左右にある4つの画素のことであり、8-近傍とは着目画素の周囲にある8つの画素のことである。ちなみに、着目画素とその4-近傍画素との連結関係を4-連結と呼び、着目画素とその8-近傍画素との連結関係を8-連結と呼ぶ。

上記の処理を行っても、粒子内部に白画素領域が残ることがある。そこで次に、白画素領域のうち面積が小さい部分を粒子領域に統合することによりノイズを除去する(小面積白画素領域の粒子部分への統合)。ここでは、白画素のうち統合可能な面積の閾値を設定し、閾値以下の面積を持つ白画素領域は黒画素に統合する。

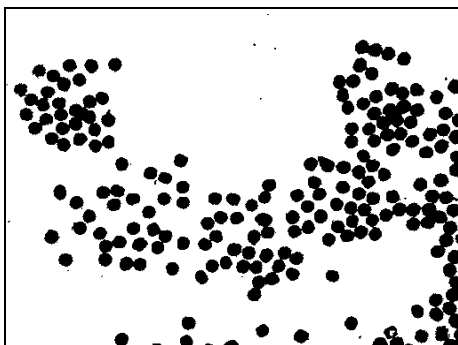


図5 ノイズ除去画像

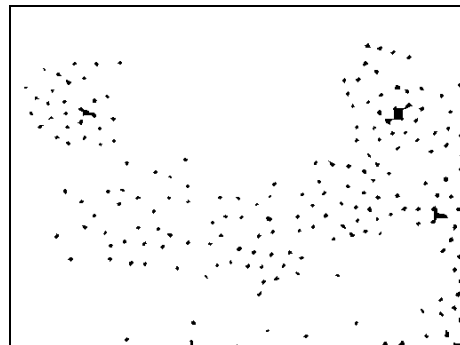


図6 収縮処理を複数回行った画像

⑤ 粒子の計数

黒画素領域について収縮処理(図6)を行った後、ラベリング処理を行って黒画素の連結領域(同じ色で繋がっている領域)の数を数えることにより、粒子を計数する。

収縮処理とは、同じ色が連結している領域の輪郭画素を取り除いて1層分小さくする処理のことである(図7)。画像 $f(i, j)$ を収縮した画像 $g(i, j)$ は式(5)の手順で得られる。

$$g(i, j) = \begin{cases} 0, & f(i, j) \text{ あるいはその4-近傍 (または8-近傍) のいずれかが0のとき} \\ 1, & \text{その他のとき} \end{cases} \quad (5)$$

また、ラベリング処理とは、同じ連結成分に属する領域毎に番号(ラベル)を付ける処理のことである。ここで、連結成分とは4-連結あるいは8-連結で繋がっている1-画素成分(対象物画素)あるいは0-画素成分(背景画素)の集まりである。

粒子同士が接触している場合は、収縮処理を繰り返すことにより粒子が分離できる。ただし、繰り返し数が多すぎると、粒子が消滅してしまう。そこで、1回収縮処理を行う度に連結領域数と画像の変化を見て、粒子数を推定する。

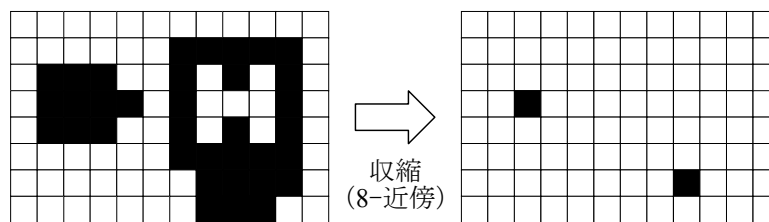


図7 収縮処理

3. 実験

次に示す手順で実験を行う。各ステップにおける処理結果は、画面への出力、及び生成された画像ファイルやテキストファイルを画面上に表示することにより確認する。

① モノクロ化

原画像(粒子の顕微鏡画像、カラー)をモノクロ化する。

プログラム: grayscale.exe

キーボード入力: カラー原画像ファイル名, モノクロ画像ファイル名

必要な画像ファイル: カラー原画像ファイル

生成される画像ファイル: モノクロ画像ファイル

※ ファイル名の最後3文字のことを拡張子と呼ぶ。画像ファイルの拡張子はbmp, テキストファイルの拡張子はtxtとする。例) 画像ファイルの場合 gazo.bmp, テキストファイルの場合 bunsho.txt

② シェーディングの補正

2-1: 光量参照領域の選択

モノクロ画像ファイルを開き、背景成分のみからなる矩形領域を2つ選択して、それぞれの端点座標対を求める(ペイントのカーソル座標を読み取る)。この2つの領域のうち、1つはなるべく画像中心に近い部分を、もう1つはなるべく周辺の部分を選択する。

プログラム: Windows のアクセサリのペイント

※ 画面中央部分の座標値①②, 画面端部(左上, 左下, 右上, 右下のいずれでも良い)の座標値③④を順番に調べる(図8)。

※ 画像ファイルをダブルクリックしても Windows のペイントが起動しない場合、「スタート」→「すべてのプログラム」→「アクセサリ」→「ペイント」を選択することで直接ペイントを起動することができる。

2-2: 像距離の算出

選択した2つの矩形領域の平均画素レベルをそれぞれ求め、像距離 f を算出する。

プログラム: focus.exe

キーボード入力: モノクロ画像ファイル名, 2つの矩形領域の端点座標対(左上と右下)

矩形領域を色表示した画像ファイル名

必要な画像ファイル: モノクロ画像ファイル

生成される画像ファイル: 矩形領域を色表示した画像ファイル

画面への出力: 像距離 f の値

※ 点の座標値は x 座標と y 座標をスペースで区切って入力する。

例) 座標値(113, 238)のとき: 113 238(Enter)

2-3: シェーディングの補正

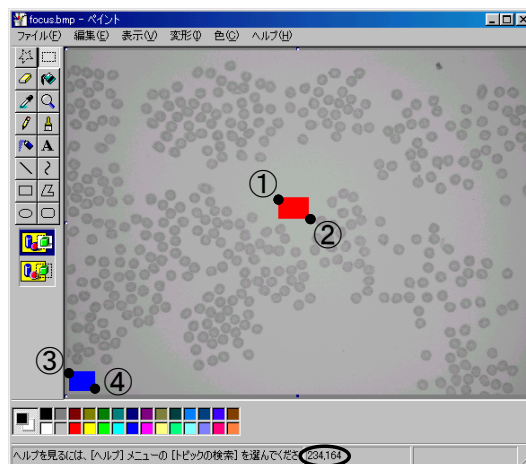
計算した像距離 f を用いて、シェーディングを補正する。

プログラム: shading.exe

キーボード入力: モノクロ画像ファイル名, 推定像距離, 補正モノクロ画像ファイル名

必要な画像ファイル: モノクロ画像ファイル

生成される画像ファイル: 補正モノクロ画像



マウス位置の座標値がここに表示される

図 8 Windows のペイントで画像ファイルを開いた様子

③ 粒子と背景の分離

シェーディング補正を行ったモノクロ画像を2値化して、背景部分と粒子部分に分離する。

3-1: 濃淡ヒストグラムの計測

シェーディング補正を行ったモノクロ画像のヒストグラムを求める。

プログラム: histogram.exe

キーボード入力: 補正モノクロ画像ファイル名, ヒストグラムテキストファイル名

ヒストグラム画像ファイル名

必要な画像ファイル: 補正モノクロ画像ファイル

生成される画像ファイル: ヒストグラム画像ファイル

生成されるテキストファイル:ヒストグラムのテキストファイル

※シェーディング補正をしていない画像に対してヒストグラムをとり比較してみる(検討(2)).

3-2:2 値化

ヒストグラムの画像およびヒストグラムのテキストファイル(Windows のアクセサリのメモ帳で開く)を参考に2値化の閾値を決定して,補正モノクロ画像の2値画像を得る.

プログラム:binarization.exe

キーボード入力:補正モノクロ画像ファイル名,2値化閾値,2値画像ファイル名

必要な画像ファイル:補正モノクロ画像ファイル

生成される画像ファイル:2値画像ファイル

④ ノイズ除去

4-1:孤立点除去

孤立点を除去することにより2値化画像からノイズ除去を行う.

プログラム:isolation.exe

キーボード入力:2値画像ファイル名,孤立点除去画像ファイル名

必要な画像ファイル:2値画像ファイル

生成される画像ファイル:孤立点除去画像ファイル

4-2:白画素領域のラベリング

白画素領域のラベリングにより,白画素の連結領域と各領域の面積を求める.

プログラム:labeling.exe

キーボード入力:孤立点除去画像ファイル名,ラベリングモード(1=白画素)

白画素ラベリング画像ファイル名,ラベリング領域面積のテキストファイル名

必要な画像ファイル:孤立点除去画像ファイル名

生成される画像ファイル:白画素ラベリング画像ファイル

生成されるテキストファイル:ラベリング領域面積のテキストファイル

4-3:小面積白画素領域の粒子部分への統合

粒子内部に存在する小面積の白画素領域を粒子部分に統合する.白画素ラベリング領域面積リストファイルの内容を参考にする(Windows のアクセサリのメモ帳で開く).

プログラム:segmentation.exe

キーボード入力:白画素ラベリング画像ファイル名,面積閾値,白画素領域統合画像ファイル名

必要な画像ファイル:白画素ラベリング画像ファイル

生成される画像ファイル:白画素領域統合画像ファイル

⑤ 粒子の計数

以下の手順を繰り返すことにより,粒子の計数を行う.

5-1:画像の収縮

背景粒子分離画像の黒画素領域について収縮処理を行う.

プログラム: erosion.exe

キーボード入力: 被収縮画像ファイル名, 収縮画像ファイル名, 収縮処理の回数

必要な画像ファイル: 被収縮画像ファイル

生成される画像ファイル: 収縮画像ファイル

5-2: 黒画素の連結領域の計数

ラベリング処理により, 収縮した画像における黒画素の連結領域数を数える.

プログラム: labeling.exe

キーボード入力: 収縮画像ファイル名, ラベリングモード (0 = 黒画素)

黒画素ラベリング画像ファイル名, ラベリング領域面積のテキストファイル名

必要な画像ファイル: 収縮画像ファイル

算出される画像ファイル: 黒画素ラベリング画像ファイル

算出されるテキストファイル: ラベリング領域面積のテキストファイル

画面への表示: ラベル数

4. 結果と考察

- (1) 実験結果をまとめ, 考察を行え. レポートには, 画像処理前の原画像と実験結果の画像を必ず付けること. 必要に応じてコマンドプロンプトから入力した値や, 途中経過の画像なども付けること.
- (2) シェーディングの補正時に, 中心から r_1 の距離での明るさが E_1 , r_2 の距離での明るさが E_2 であったとする. r_1 , E_1 , r_2 , E_2 を用い, (2)式, (3)式から像距離 f を求めよ.
- (3) シェーディングの補正を行わないと粒子と背景の分離結果がどうなるか考察せよ.
- (4) カラー画像の粒子数を各自数え, 実験で得られた個数との比較を行え. また, 結果が異なる場合は, その原因を考えよ.

(その2) 粒子形状特徴の計測

1. 目的

実験(その2)では, 顕微鏡で見た粒子画像において粒子の「形を測る」ことを目的とする.

2. 画像処理手法

粒子の顕微鏡画像(図9)において, 粒子形状を計測する. 本実験では, 図10に示す手順により, 画像処理技術を用いて粒子の形状特徴を自動的に計測する.

実験(その2)では, 実験(その1)での①~④を行った後, 粒子の形状を整形するために収縮・膨張処理を繰り返す. その後, 輪郭検出を行い, 輪郭を2次曲線の方程式に当てはめて形状特徴を計測するとともに, 粒子の周囲長・面積からも形状特徴を求める.

① モノクロ化, ② シェーディングの補正, ③ 粒子と背景の分離, ④ ノイズ除去

実験(その1)と同様.

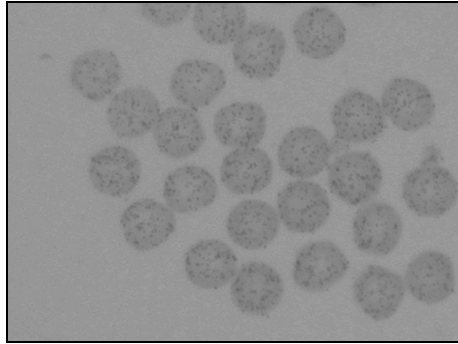


図9 粒子の顕微鏡画像の例

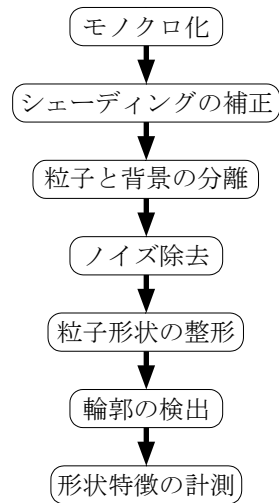


図10 粒子特徴計測のフローチャート

⑤ 粒子形状の整形

収縮・膨張処理を行うことにより、粒子形状の整形を行う(図 11, 図 12). なお、収縮・膨張処理には、画像中の小さな孤立領域や細い線分を除去するとともに、連結している粒子を分離する効果もある。

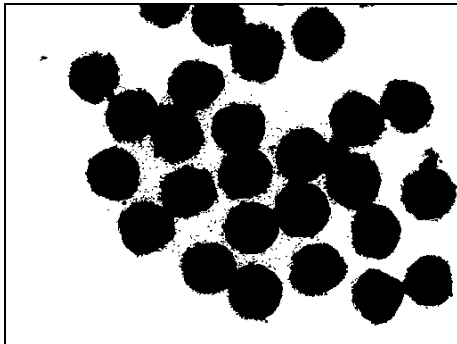


図11 収縮・膨張前の画像

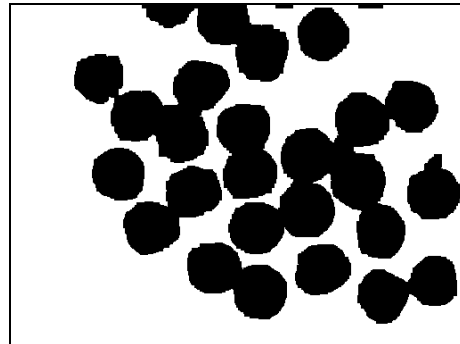


図12 収縮・膨張後の画像

ここで、膨張処理とは、実験(その 1)の⑤で触れた収縮処理と逆の処理で、同じ色が連結している領域の輪郭画素を外側に 1 層分増やし、厚くする処理のことである(図 13). 画像 $f(i, j)$ を膨張した画像 $g(i, j)$ は式(6)の手順で得られる。

$$g(i, j) = \begin{cases} 1, & f(i, j) \text{ あるいはその4-近傍 (または8-近傍) のいずれかが1のとき} \\ 0, & \text{その他のとき} \end{cases} \quad (6)$$

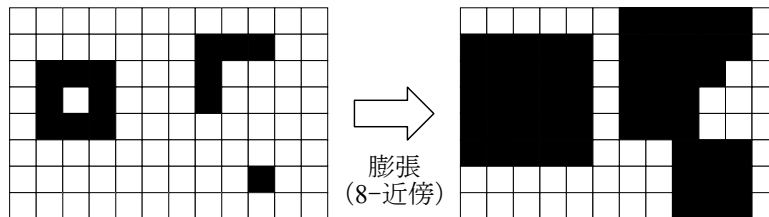


図13 膨張処理

⑥ 輪郭の検出

粒子の輪郭(エッジ)の検出を行う(図 14).

⑦ 形状特徴の計測

対象の形状がどの程度円に近いかを表す円形度を用い, 2通りの方法で形状特徴を計測する. 円形度 e は(7)式で定義される.

$$e = \frac{4\pi S}{L^2} \quad (7)$$

ただし, S は対象の面積, L は周囲長である. 真円のとき e は 1 となり, 複雑な形状となるほどその値は小さくなる.

1 つ目の形状特徴計測方法では, 得られた輪郭の 2 次曲線への当てはめを行う(図 15). 一般に, 2 次曲線の方程式は(8)式で表される.

$$px^2 + qxy + ry^2 + sx + ty + u = 0 \quad (8)$$

(8)式において係数は 6 個であるが, 5 係数は残りの 1 係数の定数倍とすることができるので, ある係数を 1 とおき, 残りの 5 係数を求めれば良い. 従って, 輪郭上の 5 点の (x, y) 座標を画像から求め, (8)式に代入し, 連立方程式を解くことで, 各係数を求めることができる. 得られた 2 次曲線の係数を用いて周囲長・面積を求め, 粒子の円形度を求める.

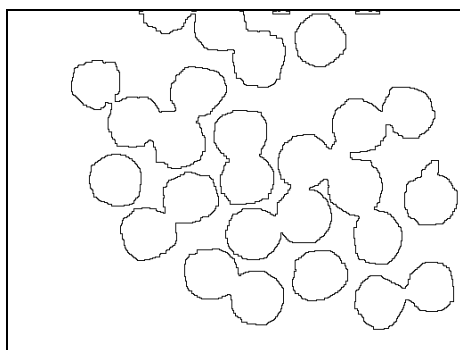


図 14 輪郭検出結果

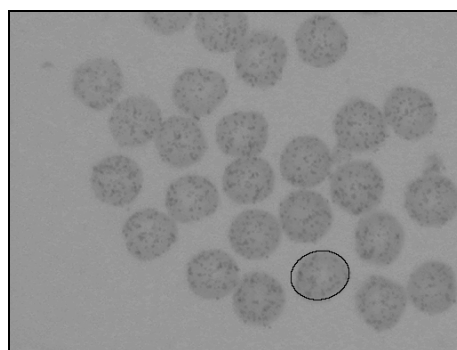


図 15 2次曲線の当てはめ

2 つ目の形状特徴計測方法では, 2 次曲線への当てはめを行わずに, 粒子 1 個の全画素数を数え, 画素数を面積とみなす. また, 粒子の輪郭の画素数を数え, 周囲長とみなす. 求めた面積と周囲長を用いて, 円形度を求める.

3. 実験

① モノクロ化, ② シェーディングの補正, ③ 粒子と背景の分離, ④ ノイズ除去
実験(その 1)と同様.

⑤ 粒子形状の整形

手順 5-1 の収縮処理を n_1 回行った後, 手順 5-2 の膨張処理を n_2 回行い, 粒子の輪郭を鮮明化する.

5-1: 画像の収縮

実験(その1)と同様.

5-2: 画像の膨張

背景粒子分離画像の黒画素領域について膨張処理を行う.

プログラム: dilation.exe

キーボード入力: 被膨張画像ファイル(5-1 で収縮を n_1 回行った画像ファイル)名

膨張処理の回数, 膨張画像ファイル名

必要な画像ファイル: 被膨張画像ファイル(5-1 で収縮を n_1 回行った画像ファイル)

生成される画像ファイル: 膨張画像ファイル

※収縮・膨張処理の回数を増やして(n_1, n_2 を大きくして)結果を比較する(検討(2)).

⑥ 輪郭の検出

6-1: 輪郭の検出

輪郭(エッジ)の検出を行う.

プログラム: edge.exe

キーボード入力: 収縮・膨張処理画像ファイル名, 輪郭検出画像ファイル名

必要な画像ファイル: 収縮・膨張処理画像ファイル

生成される画像ファイル: 輪郭検出画像ファイル

6-2: 輪郭点位置の指定

輪郭検出済画像ファイルを開き, 1つの粒子について, 輪郭部分の座標を5箇所大まかに求め, 座標値を得る. また, このとき粒子内部の点の座標(7-3 で利用)を1箇所調べる.

プログラム: Windows のアクセサリのペイント

6-3: 輪郭点の位置自動補正

大まかに求めた座標値を用い, 輪郭上の位置を正確に自動検出する.

プログラム: fitting.exe

キーボード入力: 輪郭検出画像ファイル名, 輪郭点の座標, 検出確認用画像ファイル名

必要な画像ファイル: 輪郭検出画像ファイル

生成される画像ファイル: 検出確認用画像ファイル

画面への出力: 正確な輪郭点の座標

⑦ 形状特徴の計測

円形度を算出するため, 2通りの方法で粒子の周囲長と面積を求める. まず, 手順7-1と手順7-2より, 2次曲線への当てはめを行うことで方程式の係数を求める. その結果を用いて楕円の長軸と短軸の長さを計算することで, 周囲長と面積を求める. 一方, 手順7-3より, 2次曲線への当てはめを行わずに粒子の輪郭から直接周囲長と面積を求める.

7-1: 2次曲線方程式の算出

検出された5つの輪郭点の座標により, 2次曲線の方程式の係数を求める.

プログラム:equation.exe

キーボード入力:輪郭点の座標

画面への出力:2次曲線の方程式の係数

7-2:2次曲線の重畳表示

得られた方程式の係数を入力して描画し, 原画像に重畳表示して, 一致度を確認する.

プログラム:ellipse.exe

キーボード入力:モノクロ画像ファイル名, 2次曲線の係数, 2次曲線重畳画像ファイル名

必要な画像ファイル:モノクロ画像ファイル

生成される画像ファイル:2次曲線重畳画像ファイル

※ 方程式の係数は, 各係数の値をスペースで区切って入力する.

例) 係数 $p=1.9, q=2.0, r=3.4, s=4.2, t=5.8, u=6.3$ のとき: 1.9 2.0 3.4 4.2 5.8 6.3 (Enter)

7-3:粒子の周囲長と面積の算出

2次曲線への当てはめを行わず, 画像から粒子の周囲長と面積を算出する.

プログラム:size.exe

キーボード入力:輪郭検出画像ファイル名, (6-3 で求めた) 正確な輪郭点の座標 (1 点)

(6-2 で求めた) 内部点の座標, 輪郭及び内部点確認用画像ファイル名

必要な画像ファイル:輪郭検出済画像ファイル

生成される画像ファイル:輪郭及び内部点確認用画像ファイル

画面への出力:粒子の周囲長と面積

4. 結果と考察

- (1) 実験結果をまとめ, 考察を行え. レポートには, 画像処理前の原画像と実験結果の画像を必ず付けること. 必要に応じてコマンドプロンプトから入力した値や, 途中経過の画像なども付けること.
- (2) 収縮・膨張処理の回数が多いと, どのような現象が起こるか考察せよ. またその原因は何か述べよ.
- (3) 輪郭当てはめを行った 2 次曲線 $px^2 + qxy + ry^2 + sx + ty + u = 0$ を角度 θ 回転させることにより, 楕円の形 $\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$ に変形せよ. また, 実験で得られた $p \sim u$ の数値を使って, 楕円の長軸と短軸の長さを求めよ.

ヒント: $\theta = \frac{1}{2} \tan^{-1} \frac{q}{r-p}$ (回転方向を時計回り・反時計回りのどちらと考えるかによって異なる)

- (4) 楕円の面積は πab , 周囲長は近似的に $2\pi\sqrt{\frac{a^2+b^2}{2}}$ となることから, (3)の結果を用い, 粒子の周囲長と面積を求め円形度を計算せよ. 同様に, 楕円の方程式を用いずに得られた粒子の周囲長と面積から円形度を計算せよ. また, それらの結果について比較・考察せよ. 結果が異なる場合には, その理由も考えて述べよ.

ヒント:size.exe によって算出される粒子の周囲長は, 粒子の輪郭を構成している画素の数を表している. また, 面積は粒子の輪郭と内部の画素の数の和を表している.

参考文献

- (1) 「ビジュアル情報処理－CG・画像処理入門」, CG-ARTS 協会, 2004 年(第 2 版 4 刷は 2009 年).
- (2) 「デジタル画像処理」, CG-ARTS 協会, 2004 年(第 2 版 4 刷は 2009 年).
- (3) 田村秀行 編著:「コンピュータ画像処理」, オーム社, 2002 年.

実験準備

(1) 実験用ディレクトリ(フォルダ)の作成

実験用のディレクトリ(フォルダ)を作成し, 使用する画像ファイルとプログラムをコピーする.

(1-1) フォルダの作成

実験用のフォルダを作成する. フォルダの名前や場所は自由に決めることができるが, 以下では標準例を示す.

例)Cドライブの直下に“ImageAnalysis”というフォルダを作成する.

(1-2) 画像ファイルのダウンロード

講義のホームページへ移動し, 画像ファイルをダウンロードする.

ファイル名の先頭が A の画像ファイルは実験(その 1)用, B の画像ファイルは実験(その 2)用であり, A と B でそれぞれ 1 つずつ, 合計 2 つのファイルを使用する. どの画像ファイルを使用するかは, 授業中に指示する.

例)授業中に 12 番を使用するように指示された場合, A012.bmp と B012.bmp をダウンロードする.

なお, 画像ファイルは講義のホームページから 1 個ずつ, もしくは全画像を ZIP 形式で圧縮して 1 つのファイルにまとめたものをダウンロード可能である. ZIP ファイルをダウンロードした場合には, ファイルを解凍する.

(1-3) 画像ファイルの移動

ダウンロードした画像ファイルを(1-1)で作成したフォルダに移動させる. 必ず作成したフォルダの直下に画像ファイルを移動させること.

例)A012.bmp と B012.bmp を C:\ImageAnalysis に移動させる.

(1-4) プログラムのダウンロード

画像ファイルと同様に, 実験で用いる 15 個のプログラムのファイルを講義のホームページから 1 個ずつもしくは一括してダウンロードする. 一括してダウンロードした場合には, ファイルを解凍する.

(1-5) プログラムの移動

ダウンロードしたプログラムのファイルを(1-1)で作成したフォルダに移動させる. 必ず作成したフォルダの直下に移動させること.

例)15 個のプログラムのファイルを C:\ImageAnalysis に移動させる.

(2) コマンドプロンプトでのディレクトリ移動

「スタート」→「すべてのプログラム」→「アクセサリ」→「コマンドプロンプト」を選択し、コマンドプロンプト画面中において自分が作成したディレクトリ(フォルダ)に移動する。

例)コマンドプロンプトの画面で `cd C:\ImageAnalysis (Enter)` と入力する。

レポートについて

- (1) 実験(その1)と実験(その2)の「4. 結果と考察」にある課題問題は全問正解となるまで再レポートとなるので注意すること。なお、以下のホームページの講義のページに本実験に関する資料がある。課題問題が分からないときには参考のこと。

<http://sensor.eng.shizuoka.ac.jp/~yamasita/index-j.html>

- (2) 実験の結果を適切に整理して書くと同時に、考察も書くこと。
- (3) レポートには画像処理前の原画像と実験結果の画像を付けること。必要に応じて途中経過も付ける。画像はカラー印刷でも、モノクロ印刷でも良い。また、実験メモは必ずレポートと一緒に綴じて提出すること。ただし、実験メモはあくまでメモであるため、必要なデータはまとめ直してレポートに書くこと。
- (4) 実験の感想を書くこと。

shading	モノクロ画像ファイル名 (g) <u>(b)と同じ</u> 像距離の値 (h) <u>(f)と同じ</u> 補正モノクロ画像ファイル名 (i) _____
histogram	補正モノクロ画像ファイル名 (j) <u>(i)と同じ</u> ヒストグラムテキストファイル名 (k) _____ ヒストグラム画像ファイル名 (l) _____
binarization	補正モノクロ画像ファイル名 (m) <u>(i)と同じ</u> 2 値化閾値 (n) _____ 2 値画像ファイル名 (o) _____
isolation	2 値画像ファイル名 (p) <u>(o)と同じ</u> 孤立点除去画像ファイル名 (q) _____
labeling	孤立点除去画像ファイル名 (r) <u>(q)と同じ</u> ラベリング領域面積のテキストファイル名 (s) _____ 白画素ラベリング画像ファイル名 (t) _____
segmentation	白画素ラベリング画像ファイル名 (u) <u>(t)と同じ</u> 面積閾値 (v) _____ 白画素領域結合画像ファイル名 (w) _____
erosion	被収縮画像ファイル名 (x) <u>(w)と同じ</u> 収縮処理の回数 (y) _____ 収縮画像ファイル名 (z) _____
dilation	被膨張画像ファイル名 (A) <u>(z)と同じ</u> 膨張処理の回数 (B) _____ 膨張画像ファイル名 (C) _____
edge	収縮・膨張処理画像ファイル名 (D) <u>(C)と同じ</u> 輪郭検出画像ファイル名 (E) _____
fitting	輪郭検出画像ファイル名 (F) <u>(E)と同じ</u> 輪郭点座標 (G) (____, ____) (____, ____) (____, ____) (____, ____) (____, ____) 検出確認用画像ファイル名 (H) _____ 正確な座標 (I) (____, ____) (____, ____) (____, ____) (____, ____) (____, ____)
equation	正確な輪郭点の座標 (J) <u>(I)と同じ</u> 2 次方程式の係数 (K) $p =$ _____, $q =$ _____, $r =$ _____ $s =$ _____, $t =$ _____, $u =$ _____
ellipse	モノクロ画像ファイル名 (L) <u>(b)と同じ</u> 2 次曲線の係数 (M) <u>(K)と同じ</u> 2 次曲線の重畳画像ファイル名 (N) _____
size	輪郭検出画像ファイル名 (O) <u>(E)と同じ</u> 正確な輪郭点の座標 (P) <u>(I)と同じ(1点のみ)</u> 内部点の座標 (Q) (____, ____) 輪郭及び内部点確認用画像ファイル名 (R) _____ 粒子の面積 (S) _____ 粒子の周囲長 (T) _____