

ロボット工学実験 I

<表計算ソフトを使った
画像処理実験>

1. 画像処理とは

コンピュータ上のデジタル画像データを、目的に応じて処理することを画像処理(image processing)と呼ぶ。「画像処理」という言葉の狭義の意味は、人間の視覚にとって見やすい画像に変換することで、特にこれを画像変換と呼ぶ。広い意味での「画像処理」は、それ以外にも様々な機能を含む。

- (1)画像変換 人間に見やすいように加工する処理。雑音除去など
- (2)画像解析 画像からの特徴抽出(指の位置, 目鼻の位置, 特定な色の部分の座標など)
- (3)画像計測 画像から計測(物の個数, 物の大きさなど)
- (4)画像認識 特徴の判別など(道路標識の種類, 顔の表情, 手指の形状など)
- (5)画像理解 認識結果の理解(誰がどんな動作をしていてその意図や, どんな物体がどう動いているかなどを解釈)

(1)から(5)の順番で、扱う内容の抽象度が上がって難しくなる。例えば(5)画像理解を目標として実行したいとすると、それでも最初に(1)を行うことで処理をしやすいようにあらかじめ画像をきれいに加工しておき(注: (1)の画像変換を、一番初めに必要な処理として、特に「前処理」と呼ぶことも多い)、次に(2)や(3)で処理した結果から得られた情報、目や鼻の位置を元に、(4)によって人物の表情や特定人物の行動を取得し、最終的な(5)は、(4)の結果を利用しないと実現できない。

このため、最終的に(5)を実行する場合には、(1)から(5)の流れで順番に処理し、最終的にカメラに撮像されている周囲の状況を判断させることになる。このようにコンピュータに周囲の状況を理解させようという研究分野が、コンピュータビジョン(CV: Computer Vision)である。

コンピュータビジョンに含まれる分野としてさらにマシンビジョンとロボットビジョンというサブ分野もある。

(A) コンピュータビジョン コンピュータを利用した上記全般を指す分野。

(B) マシンビジョン (A)のうち工場のオートメーション(FA)に利用するための分野で、例えばベルトコンベアに流れてきた部品類の傷を検出するなどの検査の自動化に使用されてきた。特定の照明条件、固定角度の撮影条件など、決め打ちで処理可能なため、比較的難易度が低く、早くから実用化されている。

(C) ロボットビジョン (A)のうちロボットの視覚認識を実現するために使う分野で、動体に搭載されることが多い。このため実時間(リアルタイム)性能が要求されたり、様々な照明条件や撮影角度に対応したりする必要がある。視覚認識能力の向上にはあえてこだわらず、カメラだけでなく他の補助的なセンサー情報を利用して、様々な環境下でも高速で安定した認識性能を提供し、ロボットの動作目的を果たすことを第一に目標を設定している。

2. 本実験の目的

本実験では、特に狭義の意味の画像処理、つまり(1)から(4)までの画像処理を体験する。この実験の目的は、その仕組みを理解することである。

世の中には多くの画像処理ソフトが存在する。代表的なものでは Adobe 社の市販ソフトウェア

Photoshop や、フリーソフトの GIMP などが有名である。これらのソフトウェアは(1)の機能(一部(2)の機能)を提供してくれるソフトウェアであり、こうしたソフトウェアを道具として使用すれば、画像処理の詳細な内容を知らなくとも(1)や(2)の画像処理を行うことは可能である。しかし、画像処理の内容自体を学習したいとき、これらのソフトウェアでは具体的な処理内容を確認することが基本的にはできない。

画像処理の内容を確認し、仕組みを理解するためには、すでにコンピュータプログラミングができる人ならば、自分で画像処理プログラムを作成して試みるのが手っ取り早い。だが、プログラミングの習得には時間がかかるため、今からプログラミングを習得するとしたら画像処理の内容に到達するにはかなり時間を要してしまう。

そこで、本実験では、ビットマップ画像(3.を参照)を表計算ソフトに表データとして読み込み(画像の各ピクセルの値を各セルに読み込み)、表計算ソフト上で操作することで、プログラミング作業なしに、表計算ソフトの操作だけで画像データの概念と画像処理技法の基本の理解を目指す。

3. ビットマップ画像とは

ビットマップ画像は、画素(pixel)の集合体として表現される。画素を二次元に配列した状態で、配列された画素の数だけの幅と高さを持つ。

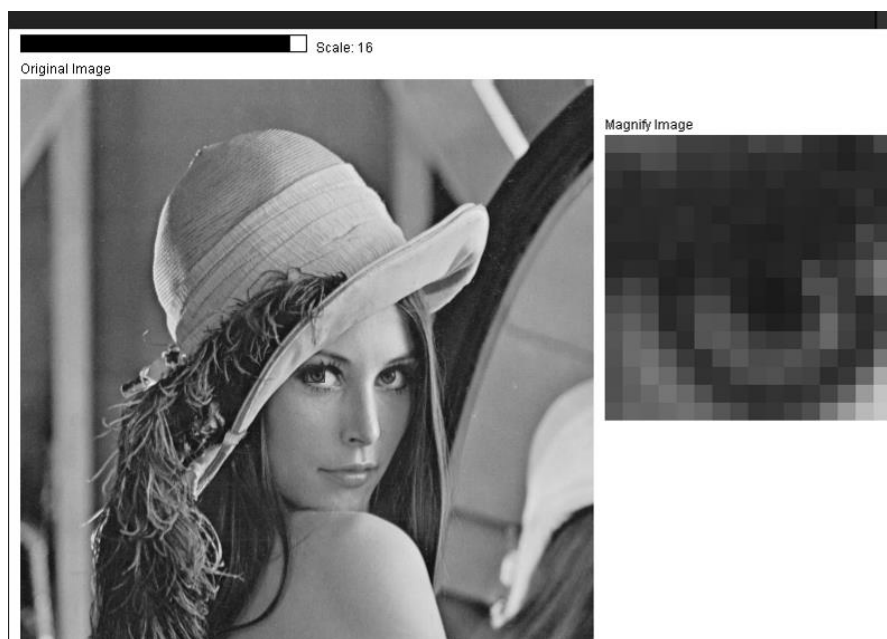


図 1 ビットマップ画像(グレースケール)

各画素には、色情報が格納される。一般的なカラー画像では光の三原色に対応して赤、緑、青をそれぞれ 0 から 255 までの 256 段階で示す値が格納される。図 1 に示したようなグレースケール画像では、色情報の代わりに明暗の度合いをおなじく 256 段階で示す値一つだけが格納される。

本実験では図 1 のようなグレースケール画像を使用し、表計算ソフトのセルひとつひとつに各画素の明度値を格納させる。

4. 実験準備

4.1 USB メモリの準備

この実験はコンピュータ上でを行い、実行結果のデータを持ち帰ってレポートを作成するために、**USB メモリを持参**すること。(スマートフォンなどメモリ代わりになる機器でも構わない。その場合接続ケーブルを忘れないこと。)

なお、自宅の PC で実験内容をやり直したり、追加実験したりすることも可能である。その方法は最後に添付してある付録を参照すること。

4.1 PC の起動

まずは実験室の PC を起動するとユーザ名とパスワードの入力欄が現れるが、次のように入力する。

ユーザ名: ISE_USER

パスワード: (※実験中に指示)

起動後に、自分の USB メモリを挿入すると、デスクトップ上に自動認識された USB メモリのアイコンと、USB 内容のフォルダウィンドウが現れる。

4.1 必要データのダウンロード

この実験では、Microsoft 社の表計算ソフト Excel を使用する。Excel のような表計算ソフトには、画像ファイルをそのままシートに貼り付けて表示する機能はあるものの、画像ファイルのデータをセルのデータとして読み込む機能はない。そこで、セルに画素データを読み込みや保存を行うための画像処理実験用マクロを自作した。この画像処理実験用マクロ機能付のブックデータファイルと、実験に使用する画像データは、Web ページからダウンロード可能となっている。

まずは自分の **USB メモリ**に図 2 のデータを各自ダウンロードする。もしも実験中にデータが破損した場合にも、何度でもダウンロードしなおすことができる。

(1) ダウンロードページ

<http://www3.nit.ac.jp/~tamura/imageprocessing/>

(2) ダウンロードするデータ

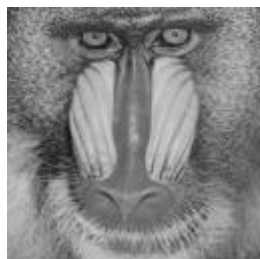
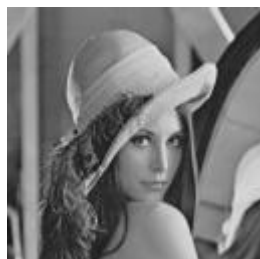




図 2 画像処理分野でよく使用されているテスト画像データ 8 種類

ブックデータ:ImageProcessing.xls

画像データ: 図2に示した 8 種類 (girl_s.bmp, lenna_s.bmp, mandrill_s.bmp, barbara_s.bmp, boats_s.bmp, bridge_s.bmp, cameraman_s.bmp, house_s.bmp)

4.2 ブックデータの使い方

コピー(またはダウンロード)したブックデータ ImageProcessing.xls を、ダブルクリックして起動すると、ダウンロードしたファイルに関する警告画面が表示される(図 3)。そこで「編集を有効化する」を選択して、承認する。続いてマクロの警告画面が表示されるので「コンテンツの有効化」を選択する。

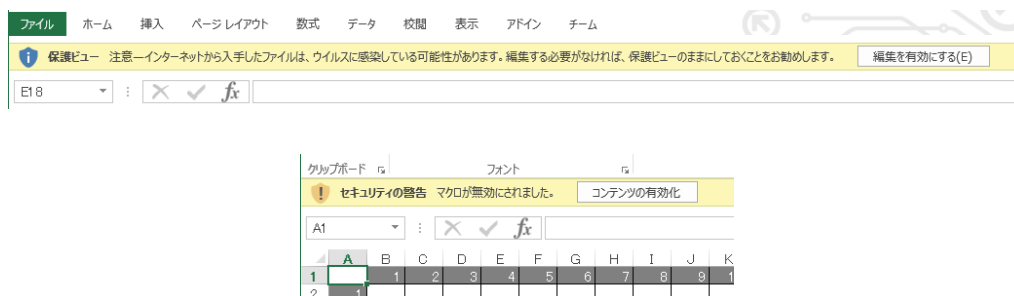


図 3 ダウンロードファイルの警告画面

すると図 4 に示すようなメニュー画面が表示されるので、これらを実験中に必要に応じて利用する。

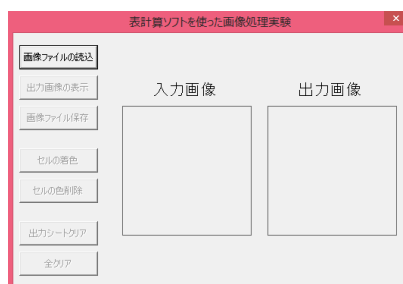


図 4 メニュー画面

- ① 画像ファイル読み込み BMP 形式の画像ファイルを読み込んで入力シートに展開。
- ② 出力画像の表示 出力シートのセルの値にしたがって、出力される画像を表示。
- ③ 画像ファイル保存 出力シート内容を BMP 形式の画像として保存。
- ④ セルの着色 現在のシートのセルの値にしたがって、着色。

- ⑤ セルの色削除 着色されたシートの色を削除.
- ⑥ 出力シートクリア 出力シートのセルの値を削除.
- ⑦ 全クリア 全てのシートのセルの値を削除.

このブックデータは、あらかじめ次の4つのシート(表)が用意されている(図 5). ウィンドウ左下にあるシート名のタブをクリックすることで切り替えられる. (※必要なら自分で追加してもよい. 追加はシート名のところで右クリックして「シートを挿入」.)

28	27								
29	28								
30	29								
31	30								

入力 作業用 作業用2 出力

図 5 4つのシート

5. 実験課題

次の実験手順で作成した画像データと作成したシートをそれぞれ自分の USB メモリへ保存せよ. これらのデータはレポート作成の際に必要な. なお, このデータを用いて自宅の PC で実験内容をやり直したり, 追加実験したりすることも可能である. その方法は最後に添付してある付録を参照すること.

5.1 実験手順

(1) 画像データの読み込み

「画像ファイル読み込み」メニューを使って, 先ほどダウンロードしておいた好きな画像データをロードする(図 6).

読み込みには時間がかかることがあるので, 128×128 のセルすべてにちゃんと値が読み込まれたことを確認してから, 次のステップへ進む.

(2) 画像処理の例

では, 最初に「画像変換」の例である「白黒反転」を行ってみよう.

「入力」シートに読み込んだデータはそのままにして, 「出力」シートを開き, 「出力シート」の各セル(128×128 の範囲)に「=255 - 入力!セル記号」という式を次の図 7 のように入力してみよ. セル記号には, その式を入力するセルの位置を示す記号(例えば B2 や C7 など)を書き込む.

例えば B2 セルの場合には, 「=255 - 入力!B2」, C7 セルならば「=255 - 入力!C7」のように, セル位置を示す記号を, 入力するセル位置に応じて変化させることを意味する.

しかし, 表計算ソフトではどこかひとつのセルに正しい式を入力して, それをコピーし, 別のセルにペーストすると自動的にセル位置に合わせた変化を起こしてくれるので, 単純に必要な場所全部にコピー&ペーストすればよく, それほど入力操作には手間はかからない.

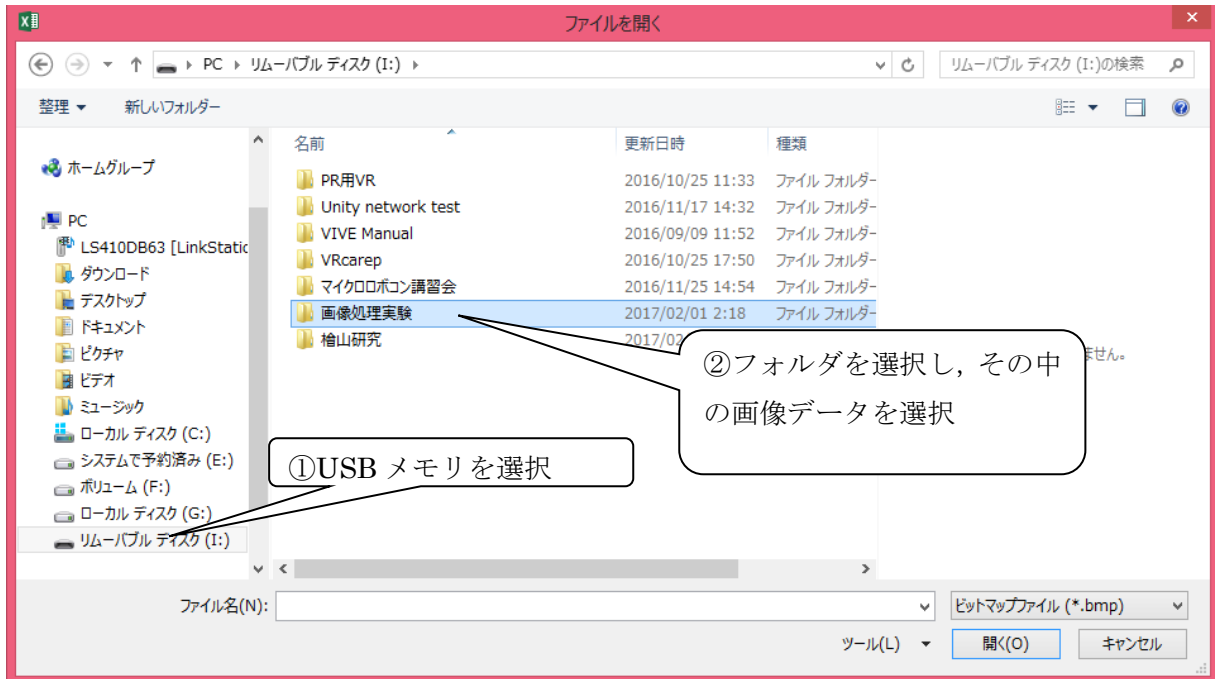


図 6 画像読み込み画面

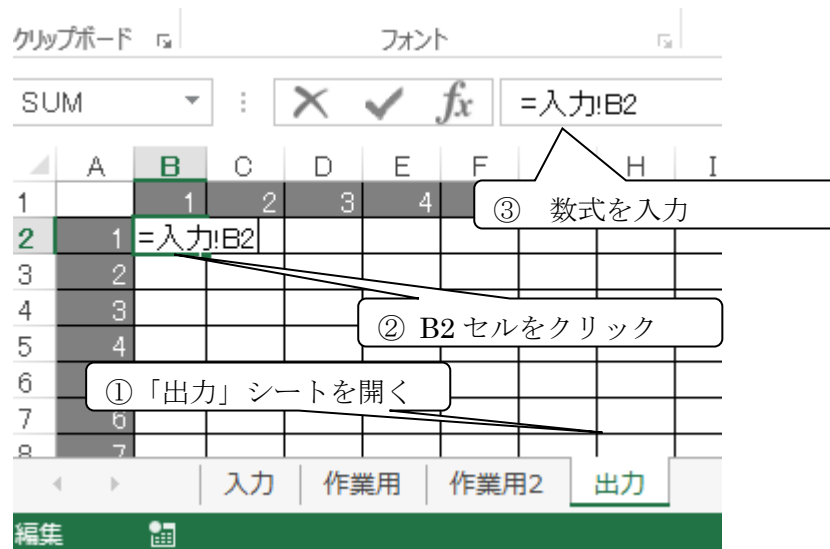


図 7 数式の入力

128×128 セル全部に式を入力し終わって「出力」シートが完成したら、一度「出力画像の表示」メニューを使って期待した画像が出来ているかを確認し、「画像ファイル保存」メニューを使って「出力」シートのデータを画像ファイルとして自分の USB に保存する。ファイル名は例えば「白黒反転」のように各自でわかりやすいものにしておくこと。画像ファイルの拡張子は自動で「.bmp」となるため特に入力する必要はない(図 8)。

保存した画像は、ダブルクリックして開き、中身を閲覧することで確認できる。

この時点でのブックデータそのものを「白黒反転.xls」と名前をつけて保存しておく。保存方法は、通常の表計算ソフトのデータ保存と同じである。これ以降の画像処理は、入力する数式が異なるだけで手順としては同様に行えばよい。



図 8 画像保存

(3) 自分で撮影したデータの利用

次に、自分で撮影したデータを利用してみよう。撮影は、自分の携帯やスマホのカメラを利用せよ。持っていない人は、同じ班の人に借りて撮影すること。

撮影の対象物はなんでもよい(例えば窓の外や、友達顔や、自分の持っている小物類などでもよい)。ただし、あとの実験で必要になるため、画面の一部に何か数字が写っている写真にしよう。例えば図 9 のようなものである。

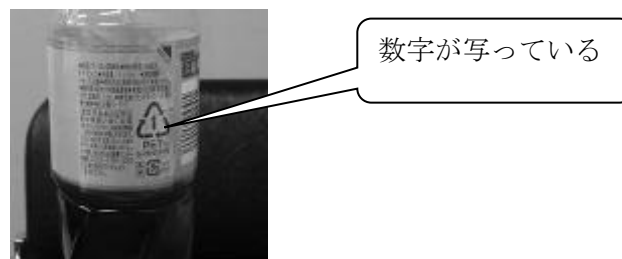


図 9 撮影写真

撮影した写真データを自分の大学のメールアドレス(または Gmail や Yahoo メールなどの自分が持っている Web メールアドレス)へ送信せよ。ただし写真サイズが大きいと送信できないことがあるため注意。送信できない場合サイズを縮小してから送信しなすこと。

Web メールを PC で開き、今送信した写真データを開いて PC に保存する。次にペイントを起動する。スタートメニューの「アプリケーション」⇒「Windows アクセサリ」⇒「ペイント」と選択することで起動できる。起動したら「ファイル」から「開く」で先ほどダウンロードした写真を開く。メニューの「ホーム」内にある「サイズ変更」を選択し、単位を「ピクセル」にした後、「128×128」画素になるように調整する(図)。このとき「縦横比を維持する」のチェックボックスを外すこと。

サイズ変更ができたなら「ファイル」メニューの「名前を付けて保存」で「ファイルの種類」を「24 ビット ビットマップ(*.bmp,*.dib)」形式を選択する(図)。ファイル名は各自のわかりやすいもので構わない。

これで、自分の撮影したデータを(1)でロードしたのと同じように読み込むことができるようになる。これ以降の実験は、自分の撮影したデータを使用せよ。

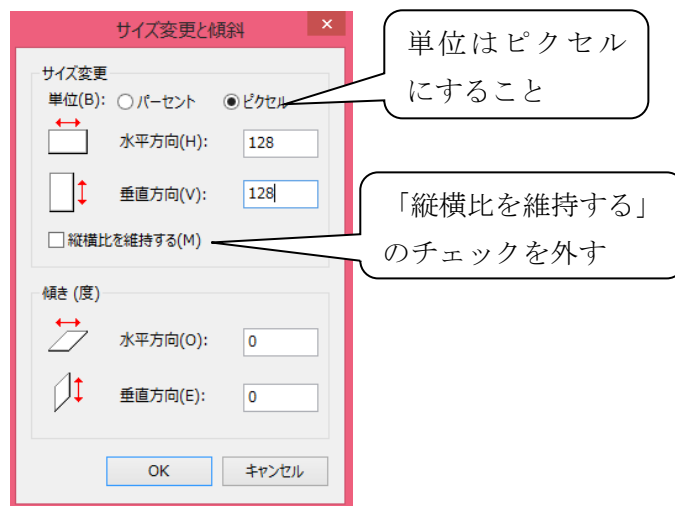


図 10 画像の拡大・縮小

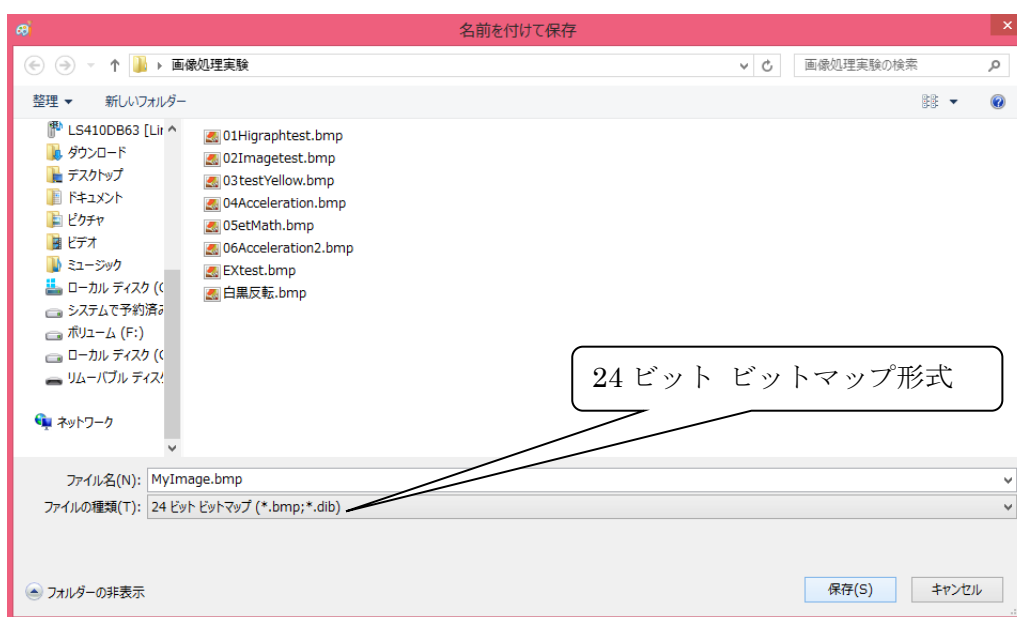


図 11 名前を付けて保存

(4) 自分の画像で白黒反転実験

では、自分の撮影したデータを用いて、画像処理実験の本番を行う。この(4)から(9)までが一週目の予定で、二週目は(11)だけの予定である。

先ほど保存したブックデータ「白黒反転.xls」を開き、「画像ファイルの読込」を行う。画像をロードしなおすだけで「出力」シートの中身も更新される。なぜならば、出力シートには画像データではなく、画像処理の数式だけ入力されていたため、数式が参照している入力シートの中身が更新されると、自動的に出力シートの数式が再計算されるためである。「出力画像の表示」で出力画像を確認し、改めてその画像を「白黒反転」として「画像ファイル保存」から保存して、あとで自分の実験レポートに掲載せよ(図 12)。



図 12 白黒反転結果の例

考察(1) ここでの実験は画像変換に対応する部分の処理である。得られた画像はどんなもので、なぜその画像が数式「=255 - 入力!セル記号」で得られるのか考えよ。(ヒント: 入力シートの値が 255 だったとき、0 だったとき、40 や 60 だったときなど、具体例で考えてみよ。)

(5) 画像の 2 値化実験

先の(4)のブックデータの保存をあらためて済ませ、その上で出力シートの数式を、次のものに変更する。

「=if(入力!セル記号 > 127, 255, 0)」

同様に完成した出力シートを画像ファイル「2 値化」として保存し、確認せよ。また、この時点のブックデータも名前をつけて保存しておくこと。以下、いくつか「画像変換」処理の実験を行う。結果の例を図 3 に示す。

考察(2) ここでの実験も画像変換に対応する部分の処理である。得られた画像はどんなもので、なぜその画像がこの数式で得られるのか考えよ。また、数式の 127 の部分の値を 60 や 200 などに変更した場合(128×128 全部で)、どんな出力画像になるのか、予想した上で実際どうなるか確認し、結果を考察してみよ。



図 13 2 値化の例

(6) 画像の γ (ガンマ)変換実験

「出力」シートの各セルに γ 変換の式を入力し、 γ 値を0.5や2.0に変えてみてそれぞれの場合の結果画像を確認してみよ。この式でoutは出力シートに格納する値、inは、入力シートのセルの値を意味する。入力画像と γ 値を0.5と2.0にした結果の例を図94に示す。

$$out = 255 \times \left(\frac{in}{255} \right)^{\frac{1}{\gamma}} \quad (\gamma \text{ 変換の式})$$

※ヒント：表計算ソフトで X^y のようなべき乗は、power(X, y)で計算できる。



図 94 ガンマ変換の結果例(左:入力, 中央: $\gamma=0.5$, 右: $\gamma=2.0$)

考察(3) 画像変換の三番目の処理である。得られた画像はどんなもので、なぜその画像がこの数式で得られるのか考えよ。(ヒント： γ が0.5のとき、入力inが0から255まで連続して変化するとき、outがどのように変化するか、表計算ソフトで計算してグラフを描いてみる。)

グラフを作成するには、表計算ソフトの新規シートに次の表1のような計算を行う。入力値を1から255まで用意して、 γ 変換の式の計算結果をまとめる。この表1を元に、図5のような散布図を作成すればよい。図5のような画像の変換式による入力値と出力値の散布図のことをトーンカーブと呼ぶ。図5は一番単純な「出力値=入力値」という式の例であり、実際の計算式にあてはめた場合、結果のカーブはいろいろな曲線を示す。

そのグラフを使って、考察を書く。 γ が0.5の時のトーンカーブと2.0のときのトーンカーブがどう変わり、それがどういうことを示すのか考える。「カーブがこうなった」だけでなく、具体的に「128のあたりでこうなった。それはこういう理由で、出力画像のこの場所をみると確かにその理屈にあったように変化が表れている。」などのように書いてほしい。

表 1 グラフ元のデータ

入力	出力
1	入力が1のときの式の計算結果
2	入力が2のときの式の計算結果
3	入力が3のときの式の計算結果
...	...
255	入力が255のときの式の計算結果

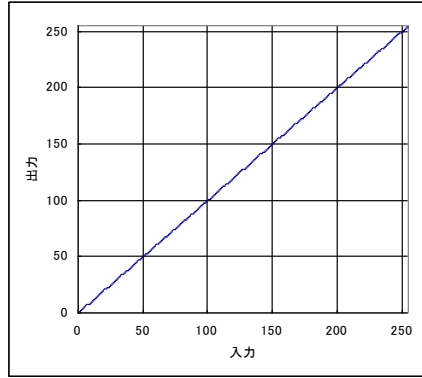


図 15 トーンカーブ例

(7) 画像の S 字カーブ変換

「出力」シートの各セルに S 字変換の式を入力し、a 値を 0.01 や 0.03 に変更してそれぞれの場合の結果画像を確認してみよ。結果の例を図 106 に示す。

$$out = \frac{255}{1 + e^{a(-in+127)}} \quad (\text{S 次カーブ変換の式})$$

※ヒント: e は自然対数の底の 2.71... のことで、 e^x は表計算ソフトでは $\exp(x)$ で計算できる。



図 106 S 字カーブ変換の結果例(左:入力, 中央:a=0.01, 右:a=0.03)

考察(4) 画像変換の四番目の処理である。得られた画像はどんなもので、なぜその画像がこの数式で得られるのか考えよ。ヒント:a が 0.01 のとき、入力 in が 0 から 255 まで連続して変化するとき、out がどのように変化するか、表計算ソフトで計算してグラフを描いてみる。グラフ作成の方法はγ変換のときと同じである。a が 0.03 のときも同様。

(8) Sobel フィルタ

次は、「画像解析」処理の例として輪郭線抽出を行ってみよう。「作業用」シートに中間結果として縦方向微分、「作業用 2」シートに横方向微分を計算する式を入力し、「出力」シートには両方向の微分値を 2 乗した和の平方根値を格納してみよ。

例えば、(左上でなく)入力.C3 セルにおける縦微分値を求めるには、作業用シートの C3 の位置で、「 = 入力!D2 + 入力!D3*2 +入力!D4 -入力!B2 -入力!B3*2 -入力!B4」を計算すればよい。

「入力」	マスク
B2 C2 D2	-1 0 1
B3 C3 D3	-2 0 2
B4 C4 D4	-1 0 1

図 117 縦方向微分マスク

これは、図 117 に示す C3 を中心とする上下左右 9 セルを使って計算するもので、C3 から見て右側縦一列から、左側縦一列を引き算する式となっている。これにより、画像の中の C3 セルの位置に、縦方向に明度差の段や、何かの縦線があった場合に、一番大きな値となる。横微分はマスクの向きを変えて横方向を計算すればよい。

そして最後に、「出力」シートの C3 セルで、「=sqrt(作業用!C3*作業用!C3+ 作業用 2!C3*作業用 2!C3) 」を計算する。sqrt 関数はルートを計算する関数である。これで両方向の微分値の 2 乗和の平方根、つまり、ベクトルの大きさを計算することになり、それを最終結果とする。

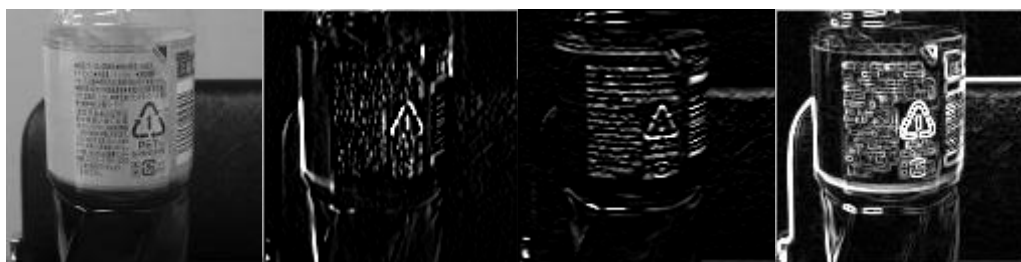


図 18 Sobel フィルタの結果例(左から入力, 縦微分, 横微分, 最終結果)

考察(5) 中間結果の縦方向微分と横方向微分, そして最終結果で, 得られた画像はどんなもので, なぜその画像がこの数式で得られるのか考えよ。

(9) 一週目の実験データの保存について

作成した画像とシートは自分の USB メモリなどに保存して持ち帰るか, 大学などの Web メールを使って自分宛に添付メールで送信することで自分からの PC からも参照できる。

一週目の予定はここまでであるが, ここまでの分のレポートを, 作業内容を忘れないうちに作成しておくことを強く勧める。

(10) 一週目の後片付け

データを保存後, USB メモリのとりはずしは, タスクバー上のアイコンを右クリックして, 「ハードウェアを安全に取り外してメディアを取り出す」を選択してから行う。いきなり取り外すとメモリのデータが破損する。

Windows はシャットダウン操作しないまま電源を落とすと破損するので, シャットダウンする。シャットダウンすると自動的に電源は落ちる。シャットダウンの確認が済んだら, 一週目の実験は終了である。

(11) 二週目の作業開始

引き続き二週目の作業を行う。一週目と同様に PC を起動し、自分の USB メモリから、ImageProcessing.xls に先週の自分が撮影した画像をロードしておく。

(12) テンプレートマッチング

引き続き二週目の作業を説明する。最後に「画像認識」処理の例として、テンプレートマッチングを行ってみよう。二週目はこのテンプレートマッチングだけであるが、やや複雑な手順となる。また、条件やテンプレートを変更して何回も出力画像を作成し、たくさんの考察を行って欲しい。

テンプレートマッチングは、画像の中からテンプレート画像に類似した部分を検出する手法である(図 191219)。



図 1912 テンプレート画像(右)の例と検出例(左の枠内)

テンプレート画像は、探索したい画像パターン、ここでは例えば数字の画像としよう。例えば「作業用」シートに数字の「1」の形状を表すような画像データを 4×4 画素として作成する(図 130)。

取った写真に写っている 1 に近くなるように、背景は 50%の灰色(=128)と 25%程度の暗い灰色(=50)で表現した例である。

実験に使用している画像データに含まれそうなテンプレート画像を作成してみよ。例えば数字の 5 が写っている画像なら、4×4 画素で表現可能な一部の特徴的なところで「∟」のパターンを使用してみるなど、各自で工夫してみよ。

	1	2	3	4	5					
1	128	50	128	128						
2	128	50	128	128						
3	128	50	128	128						
4	128	50	128	128						
5										
6										

図 130 手動で入力したテンプレート画像(「作業用」シート)

作業用 2 シートには、このテンプレート画像と、ある座標位置での類似度を計算させる。ある座標(x,y)での類似度は、単純なところでは SAD(Sum of Absolute Difference)あるいは SSD(Sum of Squared Difference)が用いられる。

$$R_{SAD} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |I(i, j) - T(i, j)| \quad R_{SSD} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (I(i, j) - T(i, j))^2$$

ただし、Iは入力像、Tはテンプレート画像、(i,j)はテンプレート画像内の座標を示す。T(i,j)は座標 i,j でのテンプレート画像の画素値を示す。I(i,j)は入力画像の座標(x+i,y+j)での画素値を示す。わかりやすくいえば、SADもSSDもテンプレート画像を入力画像の一部とを突き合わせて、画素の違いの合計を求めたものである。その大きさをもって、座標(x,y)における画像とテンプレートの「相違度」「距離」「類似度」を表している。

例えば SAD の計算結果を作業用 2 シートに格納するとすれば、その B2 セルでの数式は、

=ABS(入力!B2-作業用!\$B\$2)+ABS(入力!B3-作業用!\$B\$3)+ABS(入力!B4-作業用!\$B\$4)+ABS(入力!B5-作業用!\$B\$5)+ABS(入力!C2-作業用!\$C\$2)+ABS(入力!C3-作業用!\$C\$3)+ABS(入力!C4-作業用!\$C\$4)+ABS(入力!C5-作業用!\$C\$5)+ABS(入力!D2-作業用!\$D\$2)+ABS(入力!D3-作業用!\$D\$3)+ABS(入力!D4-作業用!\$D\$4)+ABS(入力!D5-作業用!\$D\$5)+ABS(入力!E2-作業用!\$E\$2)+ABS(入力!E3-作業用!\$E\$3)+ABS(入力!E4-作業用!\$E\$4)+ABS(入力!E5-作業用!\$E\$5)

となる。ABS は絶対値の関数である。\$B\$2 の表記は、セル B2 と同じものを表している。「B2」表記と「\$B\$2」表記の違いは、式をコピーして別のセルに張り付けたとき、その B2 のセル位置を自動的に変化させず、固定するための表記法で、絶対座標と呼ぶ。テンプレート画像側のセル位置はずらしたくないため絶対座標を使用する。

これによって SAD が作業用 2 に作成できたが、これは見てのとおり SAD は、単に差の合計のため値の範囲に基準がなく、大きな値になることが多い。このため作業用 2 シートを画像として保存しても意味がない。

最終的に出力シートに検出結果を作成する。画像として保存できるように、まず今回の SAD の最大値と最小値を求めておき、これを基準にして 0 から 255 の間の値に変換する。

	1	2	3	4	5
1	128	50	128	128	
2	128	50	128	128	
3	128	50	128	128	
4	128	50	128	128	
5					
6	MAX	1747			
7	MIN	315			

図 141 作業用シートへの SAD の最大値・最小値の計算式の追加

図 141 のように C7 セルに最大値の計算式、C8 セルに最小値の計算式を入力しておく。最大値は作業用シートにある 128×128 セル分の SAD の計算結果の最大値であり、つまり数式は「=MAX(作業用 2!B2:DY129)」となる。同様に最小値は「=MIN(作業用 2!B2:DY129)」である。

そのうえで、この最大値、最小値を使って、

$$(\text{ある SAD の値} - \text{最小値}) / (\text{最大値} - \text{最小値})$$

を計算すれば、ある「SAD の値」が全体の中でどのくらいの大きさを表しているか 0.0 から 1.0 の間の比率で示すことができる。0.0 は最小値と同じ、1.0 なら最大値と同じであることを表す。これを画像にして一目で見るためには、この式を 255 倍して、見やすくなるように白黒反転を行

考察(6) テンプレートマッチングを行った結果、検出した座標はちゃんと目的の場所かどうか報告せよ。また、誤検出の場所は何処か? なぜその場所で誤検出されたか考えよ。誤検出を避けるにはどうすればよいか考えよ。

テンプレート画像が 4×4 では、実際には小さすぎるため、十分わかりやすいテンプレートを表現できないが、テンプレート画像を次のように色々変えて試してみよ。

- ① 図 19120 のように手動で見つけたいパターンを指定したもの(数種類試してみよ。)
- ② そもそも入力画像に含まれないパターンを指定した場合
- ③ 実際に入力画像中に存在する、特徴的な箇所を 4×4 画素分抜き出したもの
- ④ (時間があれば) 8×8 画素に拡大して①か②か③と同様のもの(頑張って「=ABS(...)... 8×8 個分の式.....」を作成することになる)

(13) 応用課題

実験がかなり早く終わった人、画像処理に興味がある人、いい点が欲しい人、将来は情報系に就職したい人などは、この応用課題にも挑戦してみよう。これらは自宅でも作業できる。

応用課題① GIMP は代表的なフリーソフトの画像処理ソフトであり、本実験で行った画像処理をすべて行うことができる。

GIMP の使い方をインターネットなどで調べて、本実験で行った画像処理と同じことをして、その操作方法を報告せよ。また、処理結果は本実験のものと一致したか確認せよ。

応用課題② 例えば、「平滑化フィルタ」や「鮮鋭化フィルタ」という画像処理を自分で調べて、本実験と同様に表計算ソフトを使って実際に試してみよ。自分で調べた別の画像処理でもよい。

応用課題③ Sobel フィルタの画像を、白黒反転し、さらに二値化してみよ。写真を線画にしたような画像になる。もっときれいにするためには、どうすればよいかいろいろ試してみよ。

応用課題④ Sobel フィルタの画像を、テンプレートマッチングしてみたらどうなるか。普通の画像を使うよりも結果が良くなるだろうか。 8×8 のテンプレートとして例えば人の目などを検出できるだろうか試してみよう。

応用課題⑤ 「画像の膨張・収縮」について調査してみよ。どのように使用されるのか、どういった画像処理結果が得られるのかまとめよ。

③と④のヒント: 入力画像や途中の画像をガンマ変換や S 字変換で調整してみたり、二値化するときに 127 でなく調整した値にしてみたりいろいろありえる。また「雑音除去」など別の画像処理テクニックを試してみることも考えられる。

(14) 二週目の片付け

一週目と同様に、必要なデータ、画像を USB メモリに保存し、機材を片付けて終了である。

6. レポートの提出に関して

レポートは三週目までに作成し、次に従って提出すること。三週目の時間中にチェックされた後、その場で修正してもらう。

6.1 提出方法

- (1) 締切り: 三週目の実験日
- (2) 提出方法: 三週目にチェックを受けた後、レポート表紙に検印をもらう。修正すべき点は修正した後、事務室前 BOX に完成レポートを提出すること。

6.2 レポート内容について

指定の表紙を使用すること。この実験のレポートは、画像を多数含むためワープロでの作成を推奨する。ただし友人のレポートをコピーせず、必ず自分で作成すること。画像処理の対象の写真は自分が撮影したものなので、当然結果画像は他人とは異なるはずである。

また、次の目次にしたがうこと。

1. 目的 (実験の目的について書く。このテキスト参照。)
 2. 実験内容 (どういう機材を使ってどういう課題でどういうことを行ったか。この実験の場合には、特に表計算ソフトのセルにどういう式を入力したのか詳しく)
 4. 実験結果 (この実験の場合にはどういう画像を得られたか画像を掲載、説明文も必要)
 5. 考察 (少なくとも実験手順の間に書かれている考察(1)から考察(6)までを書く事。さらに応用課題について、その他自分の考えを自由に書き足してよい。)
 6. まとめ (この実験でなにをしてどんな結果になって何が分かったのか簡潔にまとめる)
- 参考文献 (ここには章番号は付けない。参考にした画像処理の本は何か書く)

6.3 レポート作成上の注意

次の注意事項をよく読んで作成せよ。

- (1) 章の見出しを必ずつける。
- (2) 図や表の見出しを必ずつける。 図は図の下に。表は表の上に。
- (3) ページ番号をつけること。
- (4) 参考文献は次のように書く

[番号] 著者名, 書名, 出版社, 出版年

例: [1] 田村仁, 「レポートの書き方」, 日工大出版, 2010 年

付録. 自宅で追加作業するには

自宅の PC を使う場合であっても、本実験で使用する画像読み込み・保存のために使用していたマクロプログラムは、Microsoft Office Excel がインストールされていればダウンロードページからダウンロードし、起動するだけで作業ができるはずである。