

# Windows API を用いた画像処理アプリケーションの構築

谷口真一

## 1 はじめに

近年、情報技術の発達に伴い、コンピュータによる画像処理技術も大きく進歩した。画像処理技術は、医療におけるレントゲン写真の解析やセキュリティ分野における顔認証など、様々な分野で利用されている。本研究室のタンパク質グループにおいても、計算機シミュレーションによるタンパク質の立体構造予測を行った際、求められた立体構造の可視化を行うために画像処理技術を用いている。しかし、画像処理技術はアルゴリズムが複雑であり、習得は容易ではない。そのため今回の演習では、画像処理技術の初歩について学習し、画像処理アプリケーションを C++ 言語、及び Windows API を用いて実装した。

## 2 画像処理技術

### 2.1 画像処理とは

画像処理<sup>1)</sup>とは、コンピュータを用いて画像を解析、加工することである。主な画像処理としては、モノクロ画像処理、閾値処理などが挙げられる。コンピュータを用いて画像を扱う際には、画像をデジタル画像化し、数値として表現しなければならない。そのために、画像に対し、空間的な格子上的の標本化・量子化を行った上で、画像処理アルゴリズムを用いて画像の加工を行う。標本化とは、連続的なアナログ信号を離散的なデジタル信号に変換する処理のことであり、より簡単に述べると、画像の濃淡値を画素に分割する処理のことである。また、量子化とは、画素に分割された画像の濃淡値を一定の単位の整数値で近似する処理のことであり、例えば、256 階調のデジタル画像に変換する場合であれば、0~255 の値のいずれかの近似値に当てはめることである。

### 2.2 モノクロ画像処理

モノクロ画像処理とは、カラー画像をモノクロ画像に変換する処理のことである。アルゴリズムとしては、カラー画像の RGB 成分毎に標本化・量子化を行い、各画素毎に下記の式 (1) 式を適用する。

$$f(i, j) = 0.299 * R + 0.587 * G + 0.114 * B \quad (1)$$

モノクロ画像に変換する際に式 (1) 式を適用する理由としては、YIQ 表色系が挙げられる。YIQ 表色系と RGB 表色系との変換公式は下記の式 (2) 式である。式 (2) 式の Y の値が輝度を表しているため、これを用いて白黒濃淡画像を作成する。

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

### 2.3 閾値処理

閾値処理とは、与えられたモノクロ画像に対して 2 値化を行う処理のことである。アルゴリズムとしては、閾値  $t$  を定め、各画素毎に下記の式 (3) 式を適用する。

$$f_t(i, j) = \begin{cases} 255 & ; f(i, j) \geq t \\ 0 & ; f(i, j) < t \end{cases} \quad (3)$$

上記の式 (3) 式は、定めた閾値  $t$  と各画素の値を比較し、閾値  $t$  よりも画素の値が大きければ 255、小さければ 0 の値を入力するものである。また、閾値  $t$  の閾値選択に関しては、今回作成したアプリケーションでは閾値  $t$  を直接指定する手法を用いたが、その他の手法として、p-タイル法やモード法などの手法がある。

## 3 Windows API

Windows API<sup>2)</sup>とは、Windows が OS としてアプリケーションに提供する関数セットのことである。Windows API を用いることで、Windows OS で用意されたライブラリ関数を利用することができ、容易に GUI(Graphical User Interface) の描画などを行うことができる。Windows API の目的はプラットフォームの統一であり、以前はキー入力や画面の操作などといったハードウェアを直接操作することはアプリケーションのプログラムコード中で行う必要があった。また、そのプログラムはデバイスに依存しているために異なるコンピュータでは実行できないという問題があったが、Windows API を用いることにより、プログラマが Windows アプリケーションを作成することを容易にし、かつ、作成したアプリケーションが同じ Windows OS がインストールされたコンピュータであれば実行が可能となった。しかし、Windows API は Windows OS と密接に結びついているため、Windows OS の影響を直接的に受けてしまい、プログラマは OS のバージョンごとに API の違いに注意して開発しなければならないという新たな問題が生じ、その問題の解決のために、利用目的ごとに API をまとめた MFC(Microsoft Foundation Class) というクラス・ライブラリが開発され、MFC を用いることで、OS のバージョン間の違いを意識することなく開発が行えるようになった。現在では、.NET Framework というフレームワークが開発され、.NET Framework が実装された環境であれば、開発・実行が可能となっている。

## 4 画像処理アプリケーション

本アプリケーションは、ファイル名を指定したカラー画像をモノクロ画像に変換、そして、2値化画像への変換を行う GUI アプリケーションである。実行画面を Fig. 1 に示した。

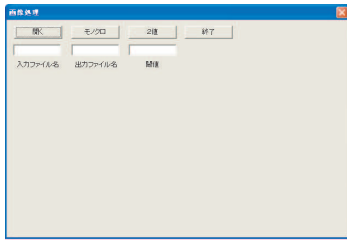


Fig.1 画像処理アプリケーション (出典：自作)

### 4.1 実行結果

モノクロ画像変換の実行結果を Fig. 2 に示す。また、2値化画像変換の実行結果を Fig. 3 に示す。Fig. 3 の閾値は 100 とする。

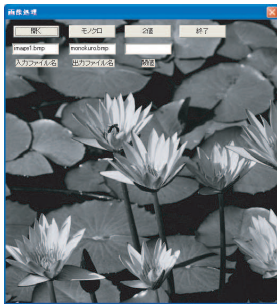


Fig.2 モノクロ画像変換 (出典：自作)



Fig.3 2値化画像変換 (出典：自作)

## 5 画像検索アプリケーション

本アプリケーションは、ある画像ファイルとその画像から切り出した部分画像ファイルを読み込み、元画像に対する部分画像の存在位置を検索するアプリケーションである。実行画面を Fig. 4 に示した。

### 5.1 実行結果

切り出した画像ファイルを Fig. 5 に示す。また、検索の実行結果を Fig. 6 に示す。

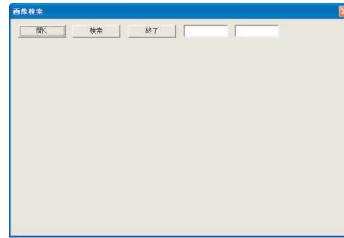


Fig.4 画像検索アプリケーション (出典：自作)



Fig.5 切り出した画像ファイル (出典：自作)

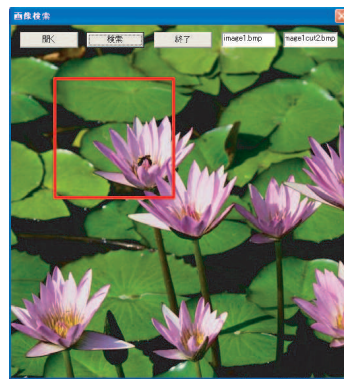


Fig.6 検索結果 (出典：自作)

## 6 まとめ

今回の演習では、C++と Windows API を用いて画像処理を行うアプリケーションを構築した。構築に用いた Windows API は、Windows OS と直接やり取りをするため、普段見ることのない形式の変数が数多く現れた。このことにより、OS は数多くの特殊な形式の変数により、管理・実行されていることがわかった。また、今回は Windows API を用いたが、次回このようなプログラムを行う際には、目的毎に複数の API をまとめ、短いプログラムコードを用いて間接的に呼び出す機能を持った MFC(Microsoft Foundation Class) を用いればさらに GUI を発展させることが出来るのではないかと考えられる。

### 参考文献

- 1) 井上誠喜, 他共著: C 言語で学ぶ実践画像処理, オーム社, 2002.
- 2) 山本信雄, プログラマ養成入門講座 Visual C++ はじめての Windows プログラミング, 2004.