

C++/CLI & OpenCV

画像処理 GUIプログラミング 【第2版】

リッチなユーザーインターフェースを持つ画像処理プログラムの実現

北山洋幸◎著



■ サンプルファイルのダウンロードについて

本書掲載のサンプルファイルは、一部を除いてインターネット上のダウンロードサービスからダウンロードすることができます。詳しい手順については、本書の巻末にある袋とじの内容をご覧ください。

なお、ダウンロードサービスのご利用にはユーザー登録と袋とじ内に記されている番号が必要です。そのため、本書を中古書店から購入されたり、他者から貸与、譲渡された場合にはサービスをご利用いただけないことがあります。あらかじめご承知おきください。

- 本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送またはE-mail）でお送りください。
- 電話によるお問い合わせはお受けできません。
- 本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
- 匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

はじめに

OpenCV (Open Source Computer Vision Library) は、膨大な関数を用意した画像処理ライブラリ集です。一般的な二次元の画像処理、ヒストグラム処理、ポリゴン処理、プレートマッチング、オプティカルフロー、および顔認識など多様なアプリケーションを開発できる関数群を用意しています。これらの関数のリファレンス、ならびに大量のサンプルプログラムは、OpenCV のサイトに紹介されています。また、同サイトには、サンプルコードと共に、それに対する丁寧な説明文まで提供されています。このため、本書は、OpenCV 自体の説明は必要最低限に留めます。

空間フィルタ処理やヒストグラムなど、およびアフィン変換などについては、.NET Framework などでも、十分なクラスが用意されているため、OpenCV を使用することなく比較的容易に開発できます。しかし、OpenCV を使用すると、もっと高度なプレートマッチング、オプティカルフロー、DFT 処理などのプログラムを開発できます。

このように豊富な関数揃えているため、OpenCV を用いると、20 行程度のコードを記述するだけで、簡単な画像処理プログラムを開発できます。ところが、良い話ばかりではなく OpenCV が用意したユーザーインターフェース用の関数は最低限のものしか備えていません。このため、せっかく高度な画像処理ができて、ユーザーインターフェースや見栄えは陳腐なものになりがちです。

そこで、本書は C++/CLI を使用することによって、リッチな UI と高度な画像処理を融合させることに挑戦してみます。このような方法を採用すると、高度な画像処理と使いやすいユーザーインターフェース、そして素晴らしい GUI をもつアプリケーションを開発できます。

対象読者

- OpenCV の概略を知りたい人
- リッチなユーザーインターフェースを持った画像処理プログラムを開発したい人
- 画像処理プログラミング入門者

謝辞

出版にあたり、お世話になった株式会社カットシステムの石塚勝敏氏に深く感謝いたします。

2019 年盛夏 玉川上水駅前のコーヒーショップにて 北山洋幸

参考文献、参考サイト、参考資料

1. OpenCV ドキュメントなどのサイト (<http://docs.opencv.org/>)
2. OpenCV 4.1.1-dev documentation サイト (<https://docs.opencv.org/master/>)
3. 北山洋幸 著、『C# と OpenCV の融合プログラミング』、株式会社カットシステム
4. 北山洋幸 著、『OpenCV4 基本プログラミング』、株式会社カットシステム

本書の使用にあたって

開発環境、および、実行環境の説明を行います。

■プラットフォーム

OpenCV はマルチプラットフォームをサポートしています。今回は、C++/CLI を使用し、リッチなユーザーインターフェースを実現することを目的とするため、Windows を開発・実行環境とします。

■ Windows バージョン

Windows のバージョンへ依存するとは思えませんが、開発・実行を確認したのは Windows 10 Home Edition (64 ビット) です。

■ Visual C++ のバージョンとエディション

無償の Visual Studio Community 2019 を使用します。Visual Studio Community 2017 や Visual Studio Community 2015 などを使用しても問題ないと思われませんが確認は行っていません。

■ OpenCV のバージョン

OpenCV 4.1.1 を使用しました。本書の執筆時点では、OpenCV 4.1.1 には 32 ビットのバイナリは含まれていませんでしたので、x64 を使用します。もし、32 ビットの OpenCV を使用したい場合は付録 E を参照し、自身で 32 ビット用のバイナリを生成してください。なお、OpenCV 4.1.0 でも大部分の開発や実行を確認しています。OpenCV は頻りにバージョンアップがなされますので、大幅な変更がない限り最新のバージョンを使用することを勧めます。

■ x64 か x86

本書では x64 (64 ビット) を採用しました。理由は、OpenCV 4.1.1 には x86 (Win32 / 32 ビット) のバイナリは含まれていないためです。32 ビットの OpenCV を使用したい場合は付録 E を参照してください。

■ 例外処理

OpenCV の関数に間違った引数を指定すると、プログラムが異常終了することがあります。ただ、本書は説明を簡単にするために、例外処理は組み込んでいません。プログラムの異常終了を避けたいなら、try ~ catch で例外を捕捉してください。

■ URL

URL の記載がありますが、執筆時点のものであり、変更される可能性もあります。リンク先が存在しない場合、キーワードなどから自分で検索してください。

ライセンス

OpenCV も、一般的なオープンソースに漏れず、下記に示すライセンスが示されています。下記の内容に同意できない場合、OpenCV をダウンロード、コピー、インストール、および使用しないでください。

他のオープンソースのライセンス同様、難しい使用条件は付帯されていません。本ライセンスが OpenCV 使用の制限になることは通常考えられませんが、必ず全文にご自身で目を通し、内容を確認して使用してください。OpenCV のライセンスについては、パッケージ内の `opencv\LICENSE.txt` に記述されています。

By downloading, copying, installing or using the software you agree to this license.
If you do not agree to this license, do not download, install,
copy or use the software.

License Agreement
For Open Source Computer Vision Library
(3-clause BSD License)

Copyright (C) 2000-2019, Intel Corporation, all rights reserved.
Copyright (C) 2009-2011, Willow Garage Inc., all rights reserved.
Copyright (C) 2009-2016, NVIDIA Corporation, all rights reserved.
Copyright (C) 2010-2013, Advanced Micro Devices, Inc., all rights reserved.
Copyright (C) 2015-2016, OpenCV Foundation, all rights reserved.
Copyright (C) 2015-2016, Itseez Inc., all rights reserved.
Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the names of the copyright holders nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall copyright holders or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

用語

用語の使用に関して説明を行います。

■カタカナ語の長音表記

「メモリー」や「フォルダー」など、最近は語尾の「ー」を付けるのが一般的になっていますので、なるべく「ー」を付けるようにします。ただ、開発環境やドキュメントなどに従来の用語を使用している場合も多いため、本書では、語尾の「ー」は統一していません。なるべく統一するよう心がけましたが、参考資料などでも混在して使用していることもあり、困難でした。このため、従来の表現と最近の表現が混在しています。

■クラスとオブジェクト

本来はインスタンス化しているためオブジェクトと表現した方がよい場所でも、クラスと表現する場合があります。これは文脈から読み取ってください。

■InputArray と OutputArray

OpenCV 3.0以降の関数プロトタイプ宣言は、画像（行列）を InputArray と OutputArray で記述している場合があります。そこで、これにならい Mat などと記述した方が適切な場合も、InputArray と OutputArray を使用します。

■ユーザーインターフェース

ユーザーインターフェースを UI や GUI と省略する場合があります。

■コントロール

フォームデザイン時などに使用するコントロール、例えば PictureBox コントロールを、PictureBox と表現し、コントロールを省く場合があります。

■フォームとウィンドウ

GUI をデザイン時にフォームと呼び、実行時にウィンドウと呼ぶことがあります、どちらも同じものを指します。

■「/」と「¥」

ソースコードを記述する際に、「/」と「¥」のどちらでも構わない場合、「/」を採用しています。なお、本文中でも「¥」が適している場合でも、ソースコードが「/」を使用している場合、「/」で表現しています。

■ソースリストとソースコード

基本的に同じものを指しますが、ソースリストと表現する場合はソース全体を、ソースコードと表現する場合はソースの一部を指す場合が多いです。

■オブジェクト

インスタンスと表現した方が良い場合でも、オブジェクトと表現している場合があります。両方を、厳密に使い分けていませんので、文脈から判断してください。あるいは、物体を指す場合もあります。

■映像とフレーム

カメラから取得した画像を映像と表現する場合とフレームと表現する場合が混在しますが、同じものを指します。これらは文脈から判断してください。

■関数とメソッド

本来ならメソッドと表現した方が良さそうな場合でも、従来の名残か関数という表現がオリジナルドキュメントで採用されているときがあります。本書も、それに倣って関数という表現を多用します。

■動画、映像、画像、フレーム

これらは混在して使用しています。動画は画像の集合です。ある瞬間では、動画も画像です。このため、動画を画像と表現した方が適切な場合があります。また、動画はフレームに分割できますので、フレームと表現する場合があります。映像とフレームは、ほぼ同様の意味ですが、文章の流れからフレームと表現した方が良い場合、フレームを採用します。

目次

| | |
|------------|-----|
| はじめに | iii |
|------------|-----|

第 1 章 はじめてのプログラム 1

| | |
|---|-----------|
| 1.1 C++ 言語で開発 | 2 |
| C++ コンソールプロジェクト作成..... | 3 |
| 実行例..... | 9 |
| 関数の説明 | 11 |
| 1.2 C++/CLI でフォームアプリケーション開発..... | 14 |
| プロジェクト作成..... | 14 |
| ビルド..... | 17 |
| 1.3 C++/CLI で Windows フォームアプリケーションと OpenCV | 21 |
| 完成したフォーム..... | 25 |
| コンストラクター..... | 29 |
| mat2bmp メソッド..... | 30 |
| mat2bmp メソッドの拡張..... | 32 |
| readFile メソッド..... | 34 |
| FileMenuItem_Click メソッド..... | 35 |
| FileMenuItem_Click メソッド..... | 35 |
| 実行例..... | 35 |
| 1.4 テンプレートを使う..... | 36 |

第 2 章 アフィン変換 47

| | |
|---|-----------|
| 2.1 フリップ..... | 48 |
| 実行例..... | 54 |
| 関数の説明 | 55 |
| 関数プロトタイプの InputArray や OutputArray について | 57 |

| | | |
|-----|-------------|----|
| 2.2 | リサイズ | 58 |
| | 関数の説明 | 59 |
| 2.3 | 回転 | 60 |
| | 関数の説明 | 62 |
| 2.4 | 透視投影 | 64 |
| | 関数の説明 | 66 |

第3章 グラフィックス 69

| | | |
|-----|---------------------|----|
| 3.1 | 画像の上に円を描く | 70 |
| | 実行例 | 75 |
| | 関数の説明 | 77 |
| 3.2 | 画像の上に線を描く | 78 |
| | 関数の説明 | 79 |
| 3.3 | 画像の上に四角形を描く | 80 |
| | 関数の説明 | 81 |
| 3.4 | 画像の上に楕円や円弧を描く | 82 |
| | 関数の説明 | 84 |
| 3.5 | 画像の上に文字を描く | 85 |
| | 関数の説明 | 86 |

第4章 色の処理 89

| | | |
|-----|--------------------------|-----|
| 4.1 | グレースケール | 90 |
| | 実行例 | 95 |
| | mat2bmp メソッドの変更 | 96 |
| | 関数の説明 | 98 |
| 4.2 | 輝度平滑化 | 99 |
| | 関数の説明 | 100 |
| 4.3 | 閾値処理 (スレッシュホールド処理) | 101 |
| | 関数の説明 | 102 |

第5章 フィルタ処理……105

| | | |
|-----|-------------|-----|
| 5.1 | フィルタ処理..... | 106 |
| | 実行例..... | 120 |
| | 関数の説明..... | 122 |

第6章 オブジェクトの処理……131

| | | |
|-----|--------------------------|-----|
| 6.1 | コーナー検出とノイズ除去..... | 132 |
| | 実行例..... | 137 |
| | 関数の説明..... | 139 |
| 6.2 | オブジェクト除去..... | 141 |
| | 実行例..... | 141 |
| 6.3 | 複数個所のオブジェクト除去..... | 151 |
| | 実行例..... | 157 |
| 6.4 | オブジェクト検出..... | 159 |
| | フォーム..... | 159 |
| | 実行例..... | 164 |
| | 関数の説明..... | 166 |
| 6.5 | 透視投影..... | 168 |
| | 実行例..... | 178 |
| | 関数の説明..... | 181 |
| 6.6 | 透視投影（マウスで頂点を指定）..... | 186 |
| | 実行例..... | 192 |
| 6.7 | 透視投影（ズームに対応）..... | 194 |
| | 実行例..... | 200 |
| 6.8 | オブジェクトのサイズ変更（自動認識）..... | 204 |
| | フォーム..... | 204 |
| | 実行例..... | 211 |
| 6.9 | マウスでオブジェクトを指定しサイズ変更..... | 214 |
| | フォーム..... | 214 |
| | 実行例..... | 219 |
| | 関数の説明..... | 221 |

| | | |
|------|----------------|-----|
| 6.10 | オブジェクト交換 | 223 |
| | フォーム | 224 |
| | 実行例 | 232 |

第7章 応用 235

| | | |
|-----|----------------------|-----|
| 7.1 | 特徴点検出 | 236 |
| | フォーム | 236 |
| | 実行例 | 242 |
| 7.2 | パノラマ | 245 |
| | 実行例 | 248 |
| | 関数などの説明 | 250 |
| 7.3 | DFT | 251 |
| | 実行例 | 260 |
| | 関数の説明 | 261 |
| 7.4 | DFT と IDFT | 265 |
| | 実行例 | 268 |
| | 関数の説明 | 269 |
| 7.5 | マウスで DFT の成分表示 | 270 |
| | 実行例 | 274 |
| 7.6 | マウスで指定した帯域を阻止 | 277 |
| | 実行例 | 278 |
| 7.7 | マウスで指定した帯域を通過 | 280 |
| | 実行例 | 286 |

第8章 動画処理 289

| | | |
|-----|----------------|-----|
| 8.1 | カメラ表示 | 290 |
| | 実行例 | 295 |
| | 関数の説明 | 295 |
| 8.2 | 動画ファイル表示 | 297 |
| | 実行例 | 302 |

| | | |
|------------|-----------------------------|------------|
| 8.3 | キャプチャー | 304 |
| | 実行例..... | 308 |
| | 関数の説明..... | 309 |
| 8.4 | 動画にフィルタ処理 | 311 |
| | 実行例..... | 317 |
| 8.5 | 動画のオブジェクト検出 | 319 |
| | フォーム..... | 319 |
| | 実行例..... | 325 |
| 8.6 | 動画のオブジェクトサイズ変更 | 326 |
| | フォーム..... | 326 |
| | 実行例..... | 332 |

付 録 **335**

| | | |
|------|----------------------------|-----|
| 付録 A | Visual Studio のインストール..... | 336 |
| 付録 B | OpenCV のインストール..... | 341 |
| 付録 C | 環境の設定..... | 345 |
| 付録 D | CMake のインストール..... | 351 |
| 付録 E | OpenCV をビルド..... | 355 |
| 索引 | | 363 |

1

はじめてのプログラム

C++/CLI から OpenCV を使用する基本を説明します。OpenCV の画像を Bitmap オブジェクトへ変換する方法も示します。高機能なことはいませんが、本章で C++/CLI から OpenCV を使用する基本を解説します。C++ 言語で開発したプログラムと比較するため、C++ と C++/CLI で同じ機能のプログラムを開発し、比較も行います。

1.1

C++ 言語で開発

単に画像を表示するプログラムを C++ で開発します。同じようなプログラムを、次節では C++/CLI で開発します。本章の目的は、C++ 言語と C++/CLI 言語で開発したときの違いを示すことです。特に C++/CLI と OpenCV との相互運用を説明します。C++ 言語で開発したプログラムを示すのは、単にリファレンスとするためです。

プログラムの機能を簡単に図で説明します。ファイルから読み込んだ画像を表示するだけの単純なプログラムです。

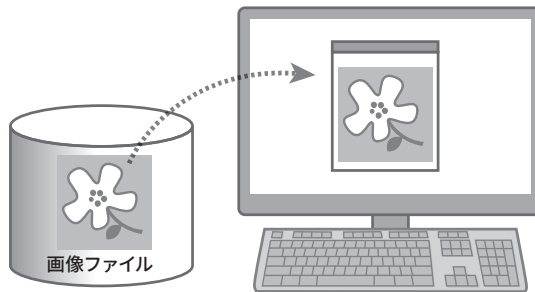


図1.1●プログラムの機能

まず、C++ 言語で開発したソースリストを示します。

リスト1.1●displImageCpp.cpp (01displImageCpp)

```
#include <iostream>

#include <opencv2/opencv.hpp>           // OpenCVヘッダ

#ifdef _DEBUG                           // Debugモードの場合
#pragma comment(lib,"opencv_world411d.lib")
#else                                    // Releaseモードの場合
#pragma comment(lib,"opencv_world411.lib")
#endif

int main(int argc, char* argv[])
{
```



```
//std::cout << "Hello World!\n";
const std::string windowName = "画像";

if (argc < 2)
{
    std::cerr << "error: no input file!!!" << std::endl;
    return -1;
}

cv::Mat img = cv::imread(argv[1]); // read image

cv::imshow(windowName, img);      // show image

cv::waitKey(0);                   // wait

return 0;
}
```

本プログラムは、コマンドラインに与えられた画像ファイルをウィンドウに表示するだけの単純なコンソールプログラムです。OpenCV は強力な画像処理ライブラリ群を用意していますが、ユーザーインターフェースに関する機能は貧弱です。強力な画像処理機能を持つ OpenCV ですが、使いやすいアプリケーションに上げるためのユーザーインターフェース機能は、豊富とは言い難いです。

Visual C++ のインクルードファイルとライブラリのパスが設定されていれば、OpenCV に関係するインクルードファイルやライブラリを、標準のヘッダやライブラリのようにファイル名を記述するだけで指定できます。ライブラリはプロジェクトのプロパティでも設定できますが、忘れることが多いため #pragma で指定します。_DEBUG によって Release 用と Debug 用の DLL を切り替えて使用するように、ライブラリ名の指定を切り替えます。

処理は、ごく普通ですのでソースリストを参照してください。特に説明が必要と思われる処理は行っていません。

■ C++ コンソールプロジェクト作成 ■

Visual C++ のコンソールプロジェクトの作成について簡単に説明します。ここでは Visual Studio 2019 Community で開発する例を示します。

1

2

3

4

5

6

7

8

付録

- ① Visual Studio 2019 Community を起動します。Visual Studio 2019 の画面が現れますので「新しいプロジェクトの作成」を選びます。



図1.2●新しいプロジェクトの作成

「新しいプロジェクトの作成」を選ばず「コード無しで続行」を選び、あらためて Visual Studio の [ファイル▶新規作成▶プロジェクト] を選び、プロジェクトを新規に作成しても構いません。

- ② 「新しいプロジェクトの作成」画面が現れますので「コンソール アプリ」を選びます。

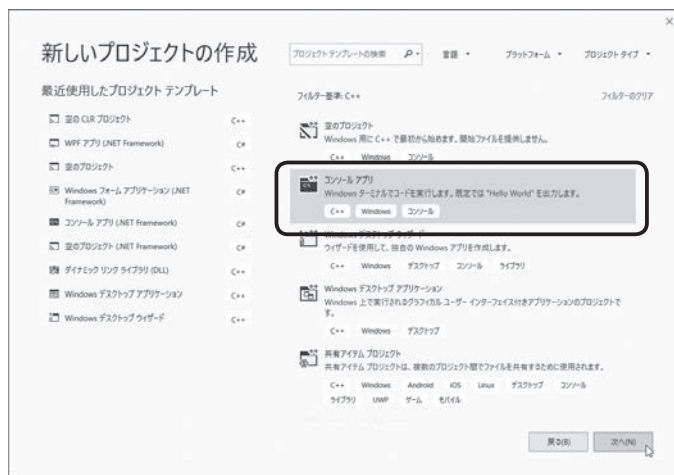


図1.3●「新しいプロジェクトの作成」画面

③ 「新しいプロジェクトを構成します」画面が現れますので「プロジェクト名」や「場所」などを指定します。

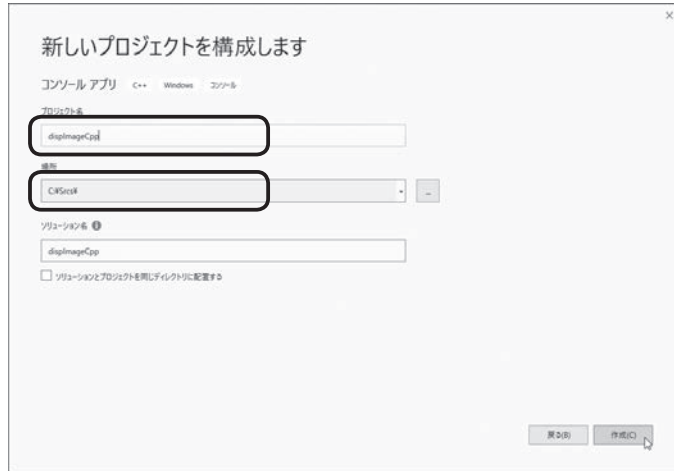


図1.4●「新しいプロジェクトを構成します」画面

④ これでプロジェクトが作成されます。このプロジェクトには、デフォルト状態のソースファイルが含まれます。

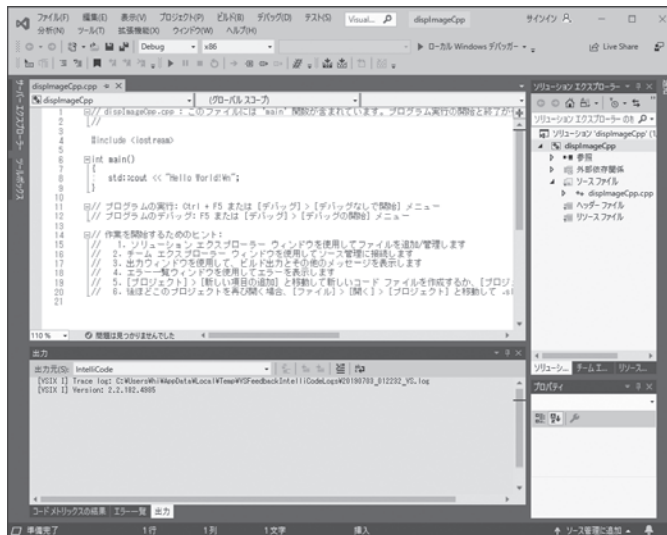


図1.5●プロジェクト完成

1

2

3

4

5

6

7

8

付録

⑤ソースファイルを書き換えた様子を示します。プロジェクトのプロパティなどを変更していないため、ソースコードの下部に赤い線が引かれインクルードなどが正常に処理されていないのが分かります。

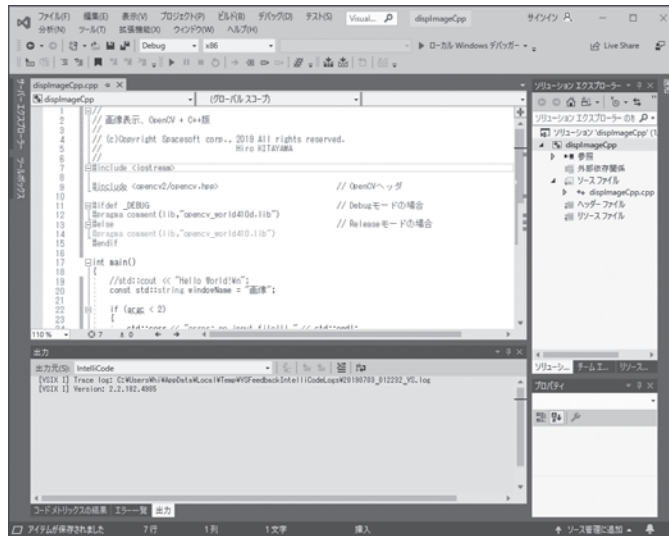


図1.6 ●ソースファイルを書き換える

Visual Studio や OpenCV のインストール、そして環境の設定については、該当する付録を参照してください。その解説に従って Visual C++ のインクルードファイルとライブラリのパスが設定されていれば、OpenCV に関するインクルードファイルやライブラリを、標準のヘッダやライブラリのように参照できます。ライブラリはプロジェクトのプロパティでも設定できますが、忘れないように #pragma で指定します。

プログラムの先頭で、引数の数が適切かチェックし、正常なら cv::imread 関数で入力画像ファイルを cv::Mat に読み込みます。次に、cv::imshow 関数で画像表示します。

cv::waitKey 関数でキー入力を監視し、何か入力があるまで表示を続けます。画像を表示しているウィンドウに対し、何かキー入力が行われるとプログラムは終了します。

⑥ソースコードの入力は完了しましたが、いくつかのオプションを設定する必要があります。OpenCV をインストールした場合、バージョン 3 あたりから 32 ビットのバイナリやライブラリは含まれなくなっています。CMake を使用すれば、各種バージョンの OpenCV をビルドできますが、ここでは標準で用意されているバイナリを使用することとします。そこで、まず

プロジェクトを 64 ビットへ変更します。

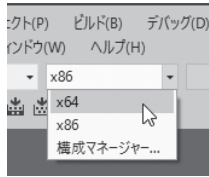


図1.7 ● 64ビット (x64) へ変更

⑦次に、プロパティページを表示させ、インクルードファイルとライブラリファイルの存在場所を設定します。プロジェクトを選択した状態で [プロジェクト ▶ プロパティ] メニューを選択するか、ソリューションエクスプローラーでプロジェクトを選択し、マウスの右ボタンを押します。すると、メニューが現れますので、[プロパティ] メニューを選択します。最初に構成を「すべての構成」へ変更します。そして、「C/C++」→「全般」→「追加のインクルードディレクトリ」へ「C:\opencv\build\include;」を、「リンカー」→「追加のライブラリディレクトリ」へ「C:\opencv\build\x64\vc15\lib;」を入力します。これで、OpenCV のプログラムをビルドする設定が完了です。

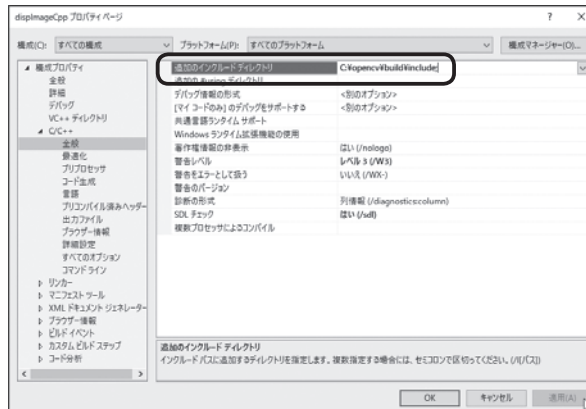


図1.8 ● インクルードディレクトリ設定

1

2

3

4

5

6

7

8

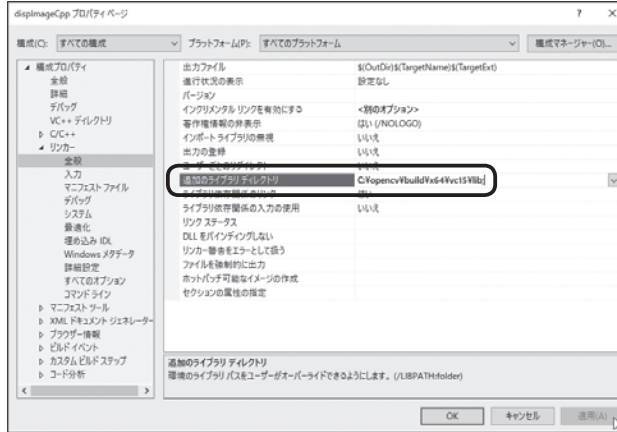


図1.9 ● ライブラリディレクトリ設定



OpenCV のディレクトリ

インクルードディレクトリやライブラリディレクトリに指定するディレクトリは OpenCV のバージョンや、OpenCV をインストールしたときに変わります。これらのディレクトリは自身の環境に合わせて変更してください。

⑧構成を Debug から Release に変更します。これを行わなくても Debug バージョンの実行形式ができるだけで特に問題はなく、Debug のままビルドしても構いません。

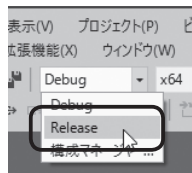


図1.10 ● DebugからReleaseに変更

これを行わないと、実行時に OpenCV のファイルが見つからず、実行時エラーが発生します。以降に、プログラムの実行例を示します。プログラム名に続き、画像ファイル名を指定します。

```
C:¥Srcs>dispImageCpp lenna.bmp
```

```
C:¥Srcs>
```

プログラムが起動すると、画像が表示されます。当然ですが、カレントフォルダーに画像ファイルを用意しておいてください。別のフォルダーに画像ファイルが存在する場合、フルパスで指定してください。

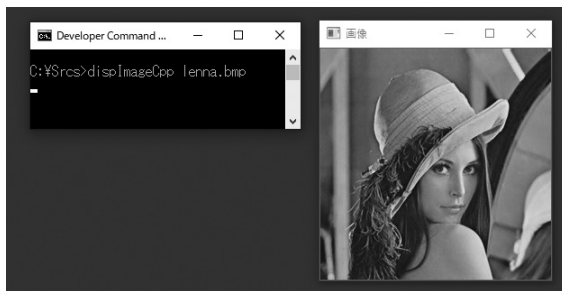


図1.12●プログラムを起動した様子

プログラムを終了させたい場合、画像表示ウィンドウにフォーカスを移動し、何かキーを押してください。コンソールにフォーカスをおいたままでは、キー入力を行ってもプログラムへキーコードが渡りませんのでプログラムは終了しません。しっかり画像を表示しているウィンドウに対しキーを送ってください。



Visual Studio のバージョン

これまでの説明は Visual Studio 2019 用の OpenCV を使用することを前提としています。このため、フォルダー名に「¥vc15」を使用しています。もし Visual Studio 2015 を使用するなら、「¥vc15」を「¥vc14」へ変更してください。ただし、使用中の Visual Studio が OpenCV を生成した Visual Studio のバージョンと整合性が取れていれば、これらのバージョンに違いがあっても構いません。OpenCV のバージョンによって提供されるバイナリが Visual Studio のすべてをサポートしているとはかぎりません。これらは Visual Studio のバージョンによって変更され