

XG-335x

Qt+OpenCV の PC 用アプリケーション開発

Rev1.0 2014/11/12

目 次


1. 概要	1
1.1 はじめに.....	1
1.2 構成	1
2. 環境構築	2
2.1 前提環境.....	2
2.2 Qt インストール.....	2
2.3 OpenCV インストール.....	3
3. プログラミング	6
3.1 Qt Creator の起動	6
3.2 プロジェクトの新規作成.....	7
3.3 プロジェクトの設定変更.....	10
3.4 画像を表示する	11
3.5 画像処理を行う	15

表記

●コマンドラインの表記


本ドキュメントには、コマンドラインで入力する操作手順が記載されております。
操作は PC で行います。記述について以下に記載します。

プロンプトは、『\$』で記載します。
実際のプロンプトには、カレントディレクトリ等が表示されますが、本ドキュメントでは省略します。
なお、省略時には、コマンドプロンプトの前に、**省略** と表記します。


本ドキュメント中での入力では、以下のように表現し、入力の最後には、 があります。

例：make コマンドを実行する場合の表記

```
省略 $ make 
```

コマンドによっては 1 つのコマンドが複数行で記載されている場合もあります。
その場合には、2 行目以降の入力では ENTER キーを押さずに続けて入力し、 の表記がある行の最後で ENTER キーを入力してそのコマンドを実行してください。

例：2 行続いてコマンド入力がある表記

```
省略 $ sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake pkg-config  
yasm libjpeg-dev 
```

●ソースコードの表記

ソースコードを表示して説明する場合があります。以下の表記になります。

例：

```
#include <opencv2/opencv.hpp>
```

1. 概要

1.1 はじめに

本ドキュメントでは、Qt と OpenCV を利用した PC 上で実行するアプリケーションの開発方法について解説します。

- ・環境構築
- ・プログラミング例

PC 用の Qt+OpenCV アプリケーションは、クロス開発環境で XG シリーズ用に容易に移植することができます。XG シリーズ用アプリケーションの事前検証として開発することも有用です。そのため、本ドキュメントでは同じバージョン構成を想定して解説しています。



本ドキュメントでは、WindowsPC に VirtualBox で稼動する Ubuntu が導入されていることが前提となっています。『Linux 開発 インストールマニュアル for XG-335x』に手順の記載がありますので、先に設定を行ってください。

1.2 構成

ソフトウェア構成

Ubuntu 12.04 LTS
Qt Creator 2.4.1
Qt 4.8.1
OpenCV 2.4.9

2. 環境構築

2.1 前提環境

本章では、次の環境の上に構築する手順を説明しますので、事前にご準備ください。

- ・ Ubuntu 12.04 LTS ※
- ・ ネットワーク設定 : インターネット接続設定。ファイルダウンロードを行うために必要となります。

※ Ubuntu 導入の詳細は「Linux 開発インストールマニュアル for XG-335x」を参照してください。

2.2 Qt インストール

Qt および Qt Creator のインストールは、apt-get コマンドにて行います。
ターミナルを開き、以下のコマンドを実行します。

① Qt インストール前にパッケージを更新します。

```
省略 $ sudo apt-get update ←入力  
省略 $ sudo apt-get upgrade ←入力
```

次の様な問い合わせがありますので、Enter キー または、Y + Enter キーで続行してください。

```
続行しますか [Y/n]? ←入力
```



upgrade の実行時に、**GRUB (grub-pc)**の設定問い合わせがでる場合があります。
ご使用の環境にもよりますが本ドキュメントでの前提環境では GRUB は不要ですので、GRUB パッケージをインストールせずに続行してください。(インストールデバイスの選択画面ではデバイスを選択せずに進んでください。)

② Qt Creator パッケージをインストールします。

```
省略 $ sudo apt-get install qtcreator ←入力
```

このコマンドにより、Qt 4.8.1 と Qt Creator 2.4.1 がインストールされます。

2.3 OpenCV インストール

sourceforge.net から OpenCV 2.4.9 のソースファイルをダウンロードしてビルド&インストールする手順を説明します。



XG シリーズ開発キットの OpenCV は 2.4.9 です。
事前検証などの用途で PC 用アプリケーションのビルド&実行を行う場合には、PC 動作の環境も開発キットと同じ 2.4.9 をインストールします。



Ubuntu 12.04 LTS の apt-get コマンドによってインストールされる OpenCV パッケージは 2.3.1 です。

OpenCV 2.4.9 のインストール

- ① ホームディレクトリに OpenCV という名前の作業ディレクトリを作成します。

```
省略 $ cd
省略 $ mkdir OpenCV
省略 $ cd OpenCV
省略 $ pwd
/home/guest/OpenCV
```

- ② インストール済みの ffmpeg と x264 パッケージを削除します。

```
省略 $ sudo apt-get -qq remove ffmpeg x264 libx264-dev
```

- ③ OpenCV パッケージのビルドと実行に必要なパッケージを追加します。

```
省略 $ sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake pkg-co
nfig yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc
1394-22-dev libxine-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-d
ev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-d
ev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-
dev x264 v4l-utils ffmpeg
```

- ④ OpenCV 2.4.9 のソースを「sourceforge.net」からダウンロードします。

```
省略 $ wget -O OpenCV-2.4.9.zip http://sourceforge.net/projects/opencvlibrary/files/ope
ncv-unix/2.4.9/opencv-2.4.9.zip/download
```

- ⑤ ソースの zip を解凍します。

```
省略 $ unzip OpenCV-2.4.9.zip
```

- ⑥ 解凍したディレクトリに build ディレクトリを作成します。

```
省略 $ cd opencv-2.4.9
省略 $ mkdir build
省略 $ cd build
省略 $ pwd
/home/guest/OpenCV/opencv-2.4.9/build
```

⑦ cmake コマンドでビルドファイルを作成します。

```
省略 $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB
=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_P
YTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON ..
```

⑧ make コマンドでビルドを実行します。(実行環境によりますが、ビルドには時間がかかります)

```
省略 $ make -j2
```

(多くの cc1plus の警告が表示されますが、そのまま続行してください。)

⑨ checkinstall コマンドでインストールします。

```
省略 $ sudo checkinstall
```

途中で問い合わせがありますので、以下のように回答して進めます。

```
The package documentation directory ./doc-pak does not exist.
Should I create a default set of package docs? [y]: n
```

```
このパッケージの説明を書いてください
説明の末尾は空行かEOFにしてください。
>>
```

```
*****
*** Debian package creation selected ***
*****
```

このパッケージは以下の内容で構成されます:

```
0 - Maintainer: [ guest@guest-VirtualBox ]
1 - Summary: [ Package created with checkinstall 1.6.2 ]
2 - Name: [ build ]
3 - Version: [ 20141028 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ i386 ]
8 - Source location: [ build ]
9 - Alternate source location: [ ]
10 - Requires: [ ]
11 - Provides: [ build ]
12 - Conflicts: [ ]
13 - Replaces: [ ]
```

変更するものの番号を入力してください。Enterで続行します:

```
You probably don't want them to be included in the package.
```

```
それらを表示しますか? [n]:
```

```
それらをパッケージから除外しますか? (yesと答えることをおすすめします) [n]: y
```

⑩ ldconfig を実行して共有ライブラリの依存関係情報を更新します。

```
省略 $ sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf' ←入力  
省略 $ sudo ldconfig ←入力
```

以上で OpneCV 2.4.9 のインストールは完了です。

ファイルは次のディレクトリにインストールされています。

ライブラリファイル	/usr/local/lib
インクルードファイル	/usr/local/include

OpenCV 2.3 のアンインストール

「OpenCV 2.4.9 のインストール」の③で行った libopencv-dev パッケージのインストールによって、OpenCV 2.3.1 のパッケージがインストールされています。これを削除します。

```
省略 $ sudo apt-get remove libopencv-core2.3 ←入力
```

opencv-core に依存する他のパッケージも削除されます。

3. プログラミング

Qt Creator で OpenCV ライブラリを利用するアプリケーションを開発する手順を説明します。

3.1 Qt Creator の起動

- ① Dash ホームアイコンをクリックします。(Dash ホームは、Ubuntu 画面の左側にあるパネルの一番上のアイコンです。) 検索の文字入力部に“Qt”と入力すると、“Qt Creator”が表示されます。

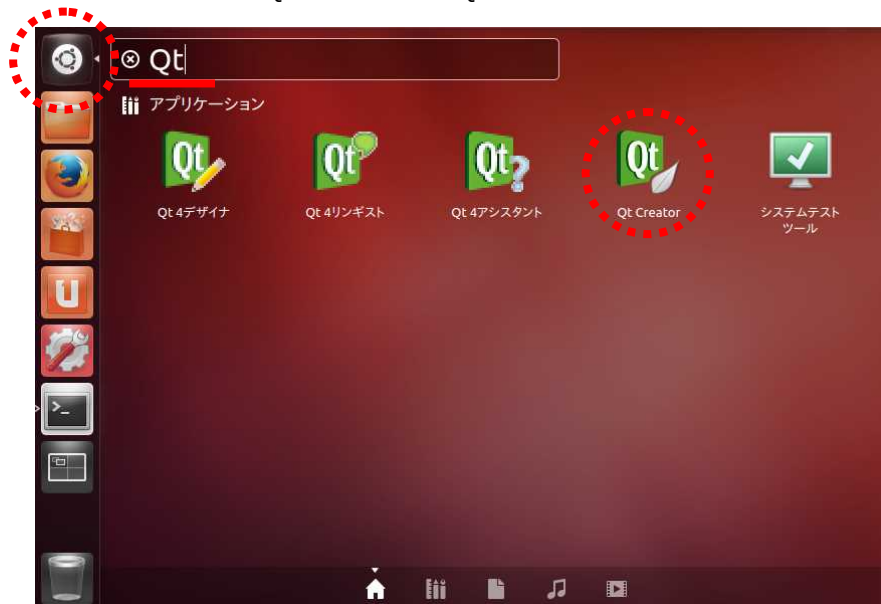


Fig 3.1-1 Qt Creator 検索

- ② Qt Creator をクリックします。
Qt Creator が起動し、次のような画面が表示されます。

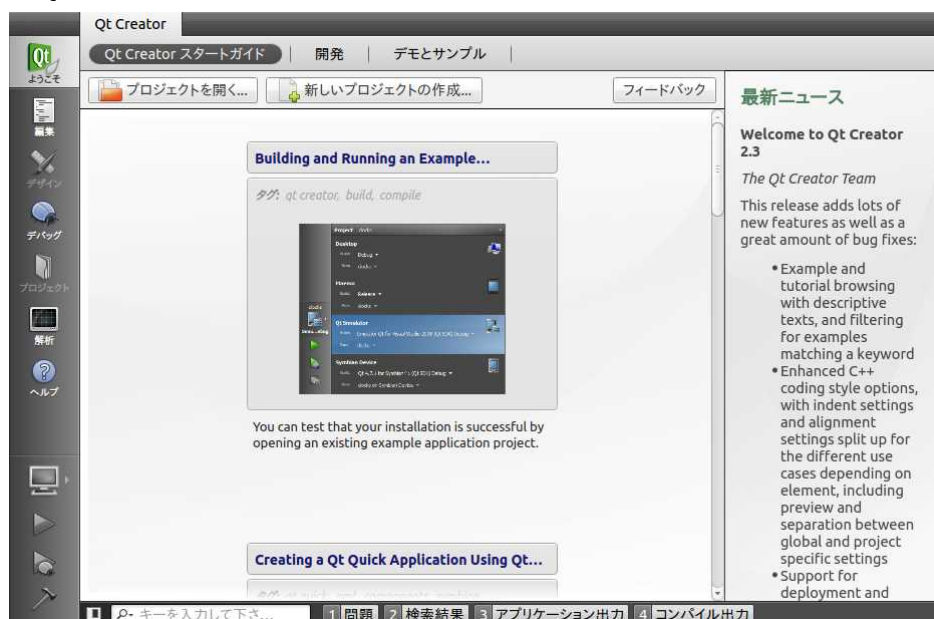


Fig 3.1-2 Qt Creator 画面



次回より検索せずに起動するために、パネルにアイコンを登録しておく便利です。
Qt Creator 起動するとパネルに Qt Creator のアイコンが表示されます。これを常に表示するために、右クリックして「ランチャーに固定」にチェックをいれます。



Fig 3.1-3 ランチャーに固定

3.2 プロジェクトの新規作成

新規作成

メニューの **ファイル** -> **ファイル/プロジェクトの新規作成...** を選択します。

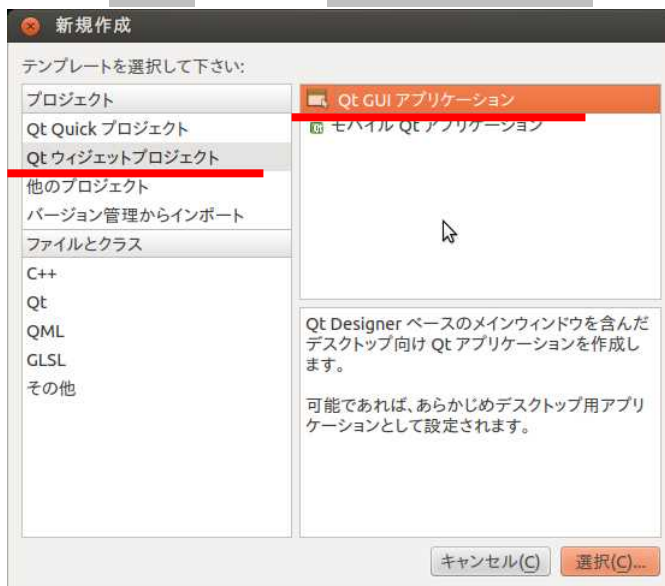


Fig 3.2-1 新規作成

ここでは「Qt ウィジェットプロジェクト」の「Qt GUI アプリケーション」を指定し、選択ボタンを押します。

プロジェクト名とパス



Fig 3.2-2 プロジェクト名とパス

プロジェクト名とパスを設定します。例えば、

- ・名前 : opencv01
- ・パス : /home/guest/projects/

と入力します。この場合、/home/guest/projects/opencv01/ フォルダが作成されて、フォルダ内にプロジェクトファイル群が入ります。パスには、存在するフォルダを指定してください。

ターゲット設定



Fig 3.2-3 ターゲット設定

既定値のまま[次へ]進みます。

クラス情報



Fig 3.2-4 クラス情報

基底クラス=QWidgetに変更します。[次へ]進みます。

プロジェクトの管理



Fig 3.2-5 プロジェクト管理

バージョン管理システムを設定できますが、今回は<なし>のままとします。
作成されるフォルダとファイル名の情報が表示されていますので、確認します。
[完了]ボタンを押します。

プロジェクトファイルとトップウィジェット (=ベースウィンドウ) となる Widget クラスが作成され、opencv01 プロジェクトを開いた画面に切り替わります。

3.3 プロジェクトの設定変更

Qt プロジェクトファイルに、OpenCV ライブラリのリンク設定を行います。

- ① プロジェクトファイル opencv01 / opencv01.pro をダブルクリックして開きます。



Fig 3.4-1 プロジェクトファイル

- ② OpenCV ライブラリリンクの定義を追加します。

```
LIBS += -lopencv_core
LIBS += -lopencv_imgproc
LIBS += -lopencv_highgui
```

以上の3つのライブラリを使用します。



OpenCV ライブラリの利用に応じて、プロジェクトファイルの LIBS 定義を変更してください。例えば、次のライブラリがあります。

```
LIBS += -lopencv_calib3d
LIBS += -lopencv_contrib
LIBS += -lopencv_features2d
LIBS += -lopencv_flann
LIBS += -lopencv_gpu
LIBS += -lopencv_legacy
LIBS += -lopencv_ml
LIBS += -lopencv_nonfree
LIBS += -lopencv_objdetect
LIBS += -lopencv_ocl
LIBS += -lopencv_photo
LIBS += -lopencv_stitching
LIBS += -lopencv_superres
LIBS += -lopencv_video
LIBS += -lopencv_videostab
```

3.4 画像を表示する

OpenCV 機能を利用した画像ファイルの読み込み／表示の手順を説明します。

ラベル定義 (ui ファイル編集)

描画先のラベルを追加定義します。以下の手順で行います。

- ① opencv01 / フォームを展開し、widget.ui をダブルクリックします。



Fig 3.4-1 widget.ui ファイル

- ② 編集モードからデザインモードに移行し、widget.ui のデザイン画面が表示されます。

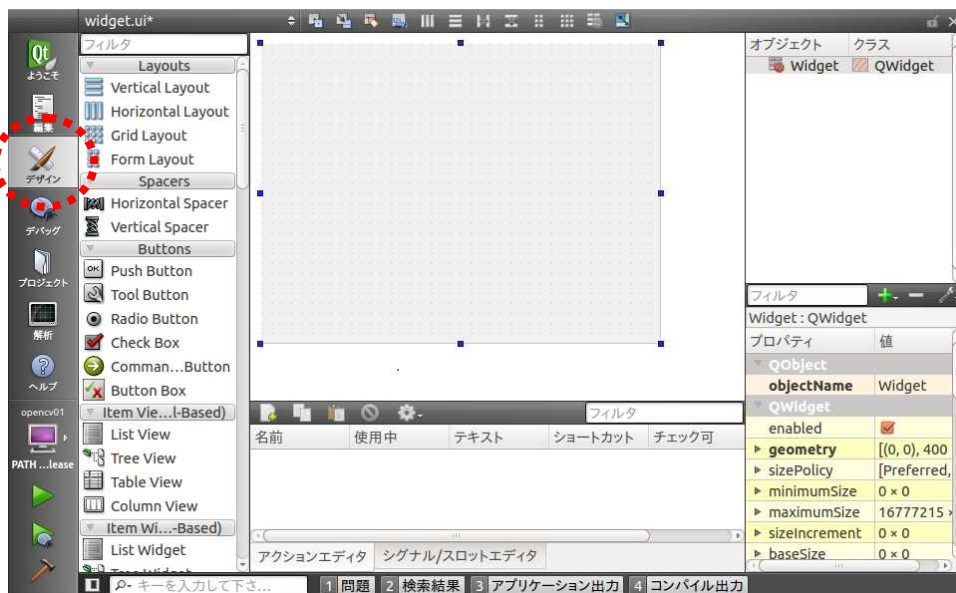


Fig 3.4-2 デザインモード

- ③ 左側のウィジェットボックス内の 'Label' をドラッグ&ドロップでウィンドウに配置します。
表示文字の "TextLabel" は不要ですので削除します。

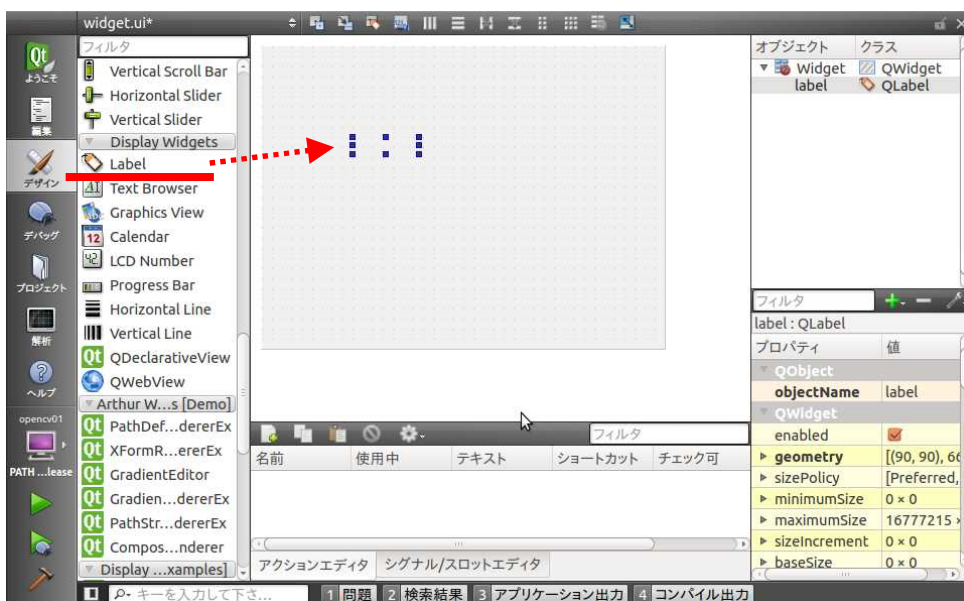


Fig 3.4-3 ラベル配置

- ④ ベースウィジェットの "Widget" を選択します。
上部のアイコンリストの中の「水平に並べる」アイコンをクリックしてレイアウト設定します。

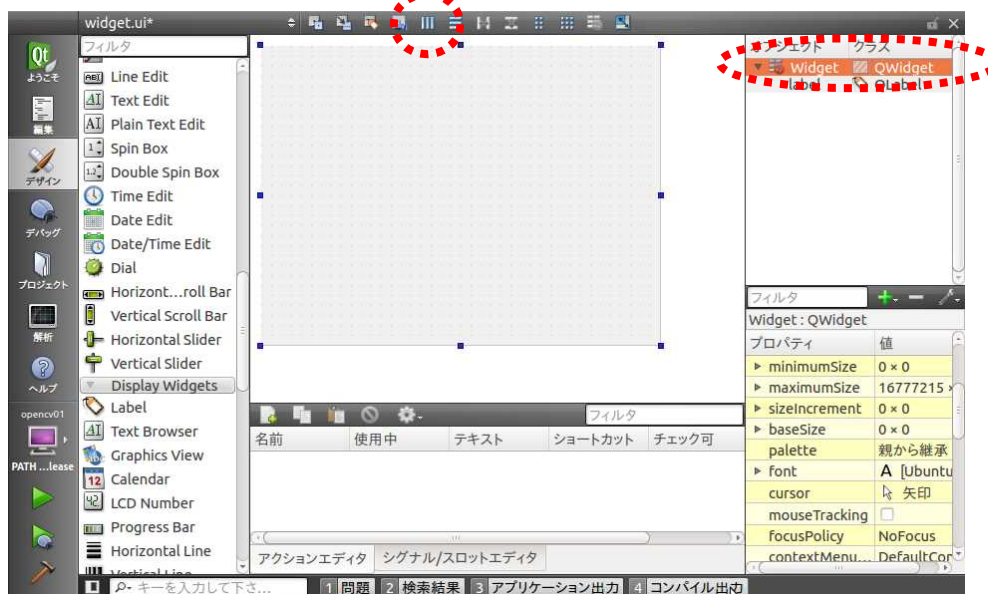


Fig 3.4-4 レイアウト設定



Fig 3.4-5 「水平に並べる」アイコン

- ⑤ Label を選択して、レイアウトの変更を確認します。
下図のようなサイズに変更されています。

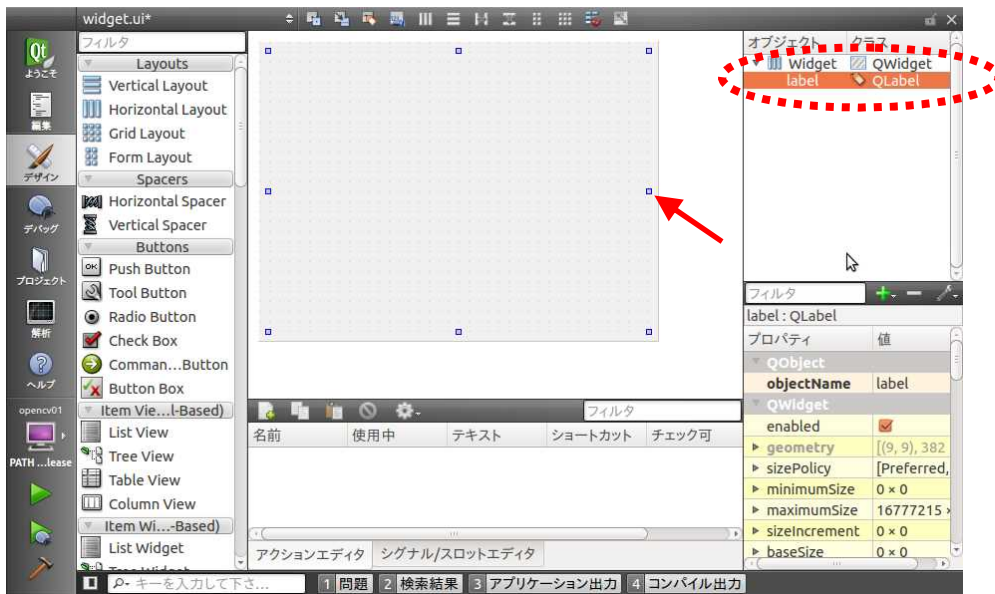


Fig 3.4-6 レイアウト確認

コード編集

画像ファイル読み込／表示の処理を定義します。widget.cpp ファイルを以下のように変更します。

```
#include "widget.h"
#include "ui_widget.h"

#include <opencv2/opencv.hpp>

Widget::Widget(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::Widget)
{
    ui->setupUi(this);

    const char* filename="/usr/local/share/OpenCV/samples/c/fruits.jpg";
    cv::Mat img = cv::imread(filename, 1);

    cv::cvtColor(img, img, CV_BGR2RGB);
    QImage qimg(img.data, img.cols, img.rows,
                img.step, QImage::Format_RGB888);
    ui->label->setPixmap(QPixmap::fromImage(qimg));
}

Widget::~Widget()
{
    delete ui;
}
```

実行

- ① Qt Creator 左下の実行ボタン(緑の三角アイコン)を押して実行スタートします。
初回の実行では、ビルドしてから実行されます。



Fig 3.4-7 「実行」ボタン

- ② PC(Ubuntu)上でプログラムが起動します。
下図のように、“Widget”ウィンドウのラベルに fruits.jpg 画像が表示されます。

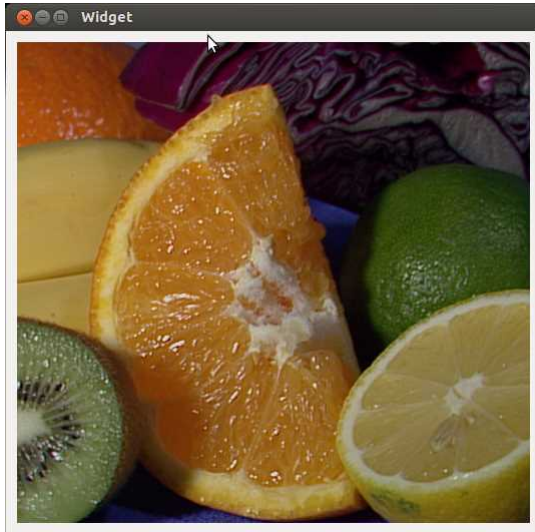


Fig 3.4-8 画像表示結果

ui ファイルのラベルサイズ/ウィンドウサイズは小さいですが、画像サイズに合わせて大きく表示されます。

解説

- ・ 画像ファイル fruits.jpg は OpenCV サンプルフォルダにあるファイルです。
- ・ imread()は 画像ファイルを読み込んで、cv::Mat 型の画像データにしています。
- ・ cvtColor()は画像データの変換を行うメソッドです。CV_RGB2BGR を指定してデータの並びを BGR から RGB の順にスワップしています。これは、OpenCV が BGR の並びになっているのに対し、表示に使う Qt は RGB であるためです。
- ・ QImage は Qt の画像データクラスです。コンストラクタで cv::Mat のデータを渡して作成しています。

3.5 画像処理を行う

次は、読み込画像に対して、ガウシアンフィルタを用いたぼかし(平滑化)と Canny 法のエッジ検出を行います。本節では、前節の画像表示処理からの変更部分のみを記載します。

コード編集

widget.cpp のコンストラクタ部分を次のように変更します。

```
{
    ui->setupUi(this);

    const char* filename="/usr/local/share/OpenCV/samples/c/fruits.jpg";
    cv::Mat img = cv::imread(filename, 1);

    cv::Mat gray;
    cv::cvtColor(img, gray, CV_BGR2GRAY);
    cv::GaussianBlur(gray, gray, cv::Size(7, 7), 1.5);
    cv::Canny(gray, gray, 0, 50);
    cv::cvtColor(gray, img, CV_GRAY2BGR);

    cv::cvtColor(img, img, CV_BGR2RGB);
    QImage qimg(img.data, img.cols, img.rows,
                img.step, QImage::Format_RGB888);
    ui->label->setPixmap(QPixmap::fromImage(qimg));
}
```

実行

実行ボタンを押して起動します。下図のようなエッジ画像が表示されます。



Fig 3.5-1 エッジ検出結果

解説

GaussianBlur() でぼかし処理をしています。

Canny()で Canny 法のエッジ検出を行っています。

Canny()は 8 ビットシングルチャンネルの入力画像である必要があるため、cvtColor()でグレースケール変換しています。

ご注意

- ・本文書の著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書に記載されている内容およびサンプルプログラムについての技術サポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。
- ・本文書の内容については、万全を期して作成いたしました。が、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書の内容は、将来予告なしに変更されることがあります。

商標について

- ・VirtualBox は、Oracle Corporation の登録商標、商標または商品名称です。
- ・Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標です。
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・Microsoft、Windows、Windows NT は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。

- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 834
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp