FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Prof. Dipl.-Inf. Ingrid Scholl

Kathrin Goffart, B.Sc.

Lirije Tahiri

OpenCV Handbuch

Bachelor Informatik

WS 2017/18

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument dient als Einstieg in OpenCV mit Qt und dem QtCreator, analog zu den Foliensätzen für Matlab. Auch, wenn Sie das Dokument vollständig gelesen und nachvollzogen haben garantiert Ihnen das kein ausreichendes Wissen für die Projekte im Praktikum. Das Dokument dient dazu, die Grundlagen zu vermitteln. Studieren Sie zusätzlich das Tutorial und setzen Sie die Aufgaben dort um, diese geben Ihnen das nötige Wissen und die grundlegende Routine für die anschließenden Projekte.

In erster Linie war die eigene Nutzung von OpenCV für Ubuntu gedacht. Im Ilias finden Sie aber eine .zip Datei, die Ihnen die Installation für Windows mit minGW liefert. Diese Version ist nicht mit VisualStudio kompatibel. Beachten Sie bitte, dass dieses Dokument auf Qt 5.7.0, Qt Creator 4.1.0 und OpenCV 3.2.0 basiert. Alle Beispiele und Screenshots wurden über Ubuntu erstellt, sollten Sie mit anderen Betriebssystemen oder Versionen arbeiten, kann es zu Inkompatibilitäten kommen. Nutzen Sie dazu Hilfestellungen im Web.

Nach den Installationshinweisen beginnt dieses Handbuch mit einem Überblick über Qt, wie man ein Projekt aufsetzt und was in den Projektdateien zu beachten ist und ein paar Hinweise zur GUI-Entwicklung. Auf weitere Details, wie z. B. Datenstrukturen von Qt (z. B. QMap) wird hier nicht weiter eingegangen, eine ausführliche Dokumentation dazu finden Sie auf der Qt-Homepage. Es folgt eine Beschreibung zu OpenCV und den grundlegenden Befehlen zur Bildverarbeitung, wie sie Ihnen auch bei Matlab zur Verfügung gestellt werden. Weitere Hinweise und Tipps finden sie online auf in den Tutorials, die von OpenCV zur Verfügung gestellt werden.

2 Installation

OpenCV und Qt werden Ihnen auf den Praktikumsrechnern unter Windows zur Verfügung stehen. Um aber jederzeit auch von zu Hause aus an den Projekten arbeiten zu können, bietet sich eine Installation auf ihrem eigenen Rechner an. Da es sich hierbei komplett um Open Source Software handelt, können Sie diese natürlich gerne auch auf ihren eigenen Rechnern installieren.

Wenn Sie Ubuntu als Betriebssystem wählen, nutzen Sie (wenn möglich) keine Ubuntu-VM, sondern Ubuntu als Betriebssystem.

2.1 Qt und QtCreator Installation

Um die Software auf ihren eigenen Rechnern zu installieren, müssen sie zunächst Qt und QtCreator installieren. Einen Download der aktuellen QtCreator Version finden Sie unter `https://www.qt.io/download-open-source/`. Gehen Sie oben auf „QtCreator“und wählen sie die gewünschte Version.

Starten Sie sowohl unter Ubuntu als auch unter Windows die Installation und folgen Sie dem Setup.

2.2 OpenCV Installation

Im Anschluss installieren Sie OpenCV, laden Sie sich dazu das Paket für OpenCV 3.2.0 unter Linux runter `https://www.opencv.org/opencv-3-2.html`. Installiert wird dies unter Ubuntu mit den folgenden Befehlen:

```
1 Terminal öffnen (Strg+Alt+T)
2 cd /home/$<$yourDownloadPlaceOpenCV$>$
3 mkdir build
4 cmake ../
5 make -j 4
6 sudo make install
```

Um Ihnen eine Installation unter Windows zu erleichtern finden Sie im Ilias eine .zip Datei. Laden Sie sich diese herunter und entpacken Sie diese am gewünschten Ort. Damit haben Sie unter Windows die Installation abgeschlossen.

3 QTCreator

QtCreator ist eine Entwicklungsumgebung für Qt, die Visual Studio sehr ähnlich ist. Qt selbst verfügt über viele Datenstrukturen und Funktionalitäten, die verwendet werden können. Da sie aber hier nicht im Fokus stehen, wird darauf nicht weiter eingegangen. Wir nutzen den QtCreator für die OpenCV Projekte und daher wird an dieser Stelle nur die integrierte Funktionalität von QtCreator um GUI's zu erstellen ausführlich beschrieben. In diesem Kapitel werden Ihnen die wichtigsten Befehle und Funktionen vorgestellt.

3.1 Anlegen und Öffnen eines Projekts im QtCreator

Öffnen Sie den QtCreator. Um ein bestehendes Projekt zu öffnen gehen Sie auf „Open Project“ oder wählen Sie es direkt unter „Recent Project“ aus. Gehen Sie auf „New Project“ um ein neues Projekt zu erstellen (siehe Abbildung 1).

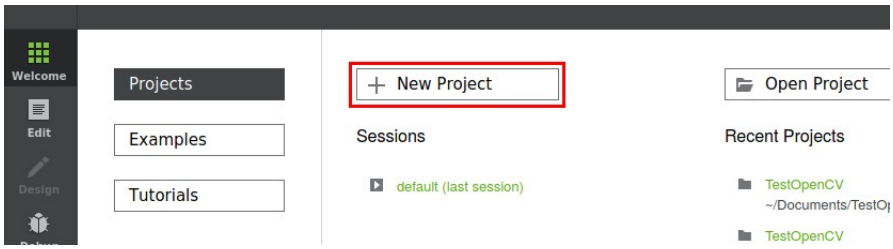

Abbildung 1: QTCreator Startseite - neues Projekt über Button anlegen

Es öffnet sich ein Fenster wie in Abbildung 2 unter dem Sie die Art ihres Projektes wählen.

Auf der linken Seite finden Sie verschiedene Arten von Projekten, die Qt bereits zur Verfügung stellt. Wenn Sie diese anklicken, öffnen sich in der Mitte verschiedene Varianten. Klicken Sie eine Variante an und es erscheint im rechten Teil eine Kurzbeschreibung. Wählen Sie hier die gewünschte Klasse aus und klicken Sie dann auf „Choose“. Als Beispiel wird hier eine „QT Widgets Application“ gewählt, da diese für die GUI-Entwicklung genutzt wird und die meisten Aufgaben im Rahmen dieses Praktikums diese verwenden.

In den nachfolgenden Schritten legen Sie zuerst den Namen des Projekts und den Speicherort fest. Danach folgen Kit Selections, diese können unverändert bleiben. Zuletzt gibt man den Namen der ersten Klasse an, die automatisch in das Projekt eingefügt wird. Bei einem Widget Projekt ist das die Klasse des Hauptfensters. Damit sind alle Einstellungen vorgenommen und es folgt eine kleine Übersicht der Dateien, die angelegt werden. Mit „Finish“ wird das Projekt erstellt.

Die Projektstruktur sieht wie in Abbildung 3 aus, die einzelnen Dateien sind bereits vorkonfi-

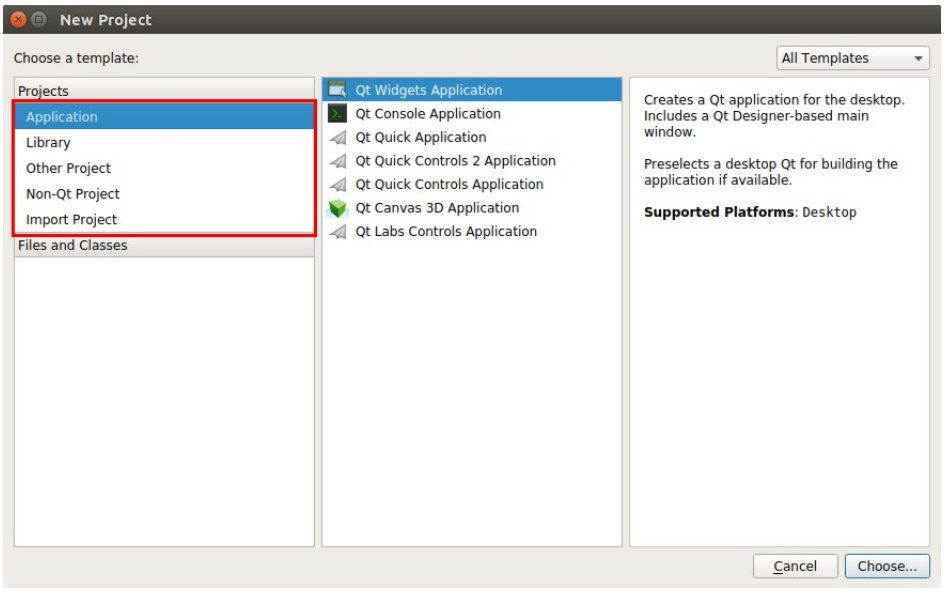

Abbildung 2: Template Auswahl für ein neues Projekt

guriert (vererben, initialisieren, etc.). Das Projekt besteht zu Beginn aus einer main.cpp und drei Dateien zum „exampleWindow“ (.h, .cpp und .ui). Die Header und Source Datei sollten bekannt sein, die .ui Datei enthält die GUI. Nähere Erläuterungen dazu folgen später.

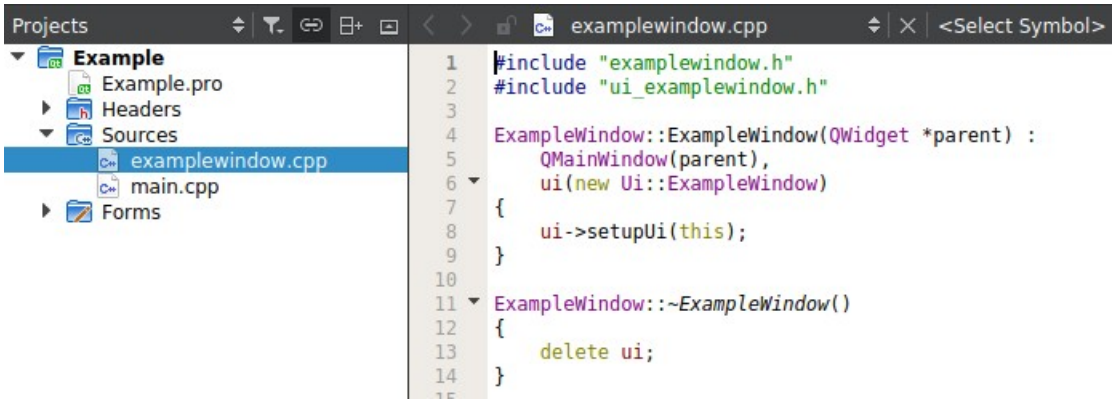

Abbildung 3: Beispiel Projekt nach der ersten Initialisierung in Qt

Öffnen Sie als erstes einmal die .pro Datei. Hier finden Sie grundlegende Eigenschaften des Projekts. Standardmäßig werden hier bereits die Projektdateien eingefügt. Nach der Erstellung ihres Projekts sieht sie wie in Abbildung 4 aus.

Um nun mit OpenCV arbeiten zu können muss in diese Datei der Pfad und die Bibliotheken von allen verwendeten Libraries eingetragen werden. Ein Beispiel hierfür kann wie in Abbildung 5 aussehen (dieses Beispiel finden Sie zum Download im Ilias). Sie müssen hier aber selbst schauen welche Libs Sie für Ihr Projekt benötigen. Es kann sein, dass hier überflüssige Pakete enthalten sind oder benötigte fehlen. Des Weiteren müssen auch die Pfade geprüft werden,

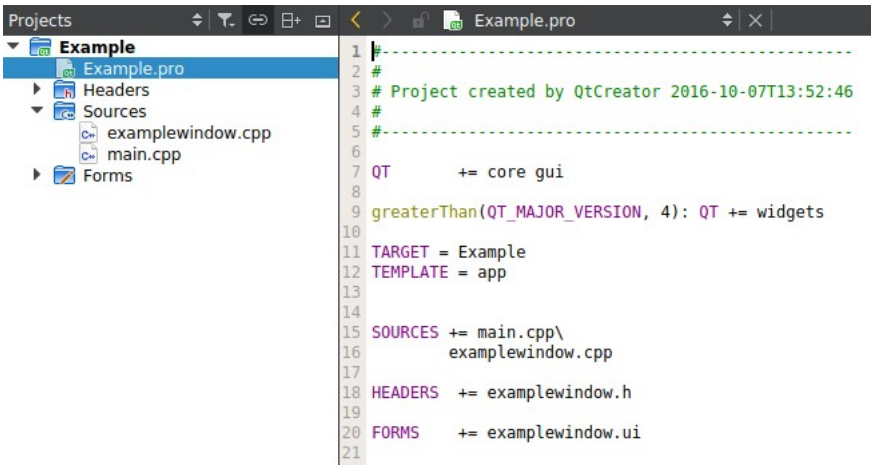

Abbildung 4: Projekt File eines Beispiel Prijekts.

sollten Sie OpenCV woanders installiert haben, so müssen diese Bereiche angepasst werden. Das gilt insbesondere für Windows Nutzer.

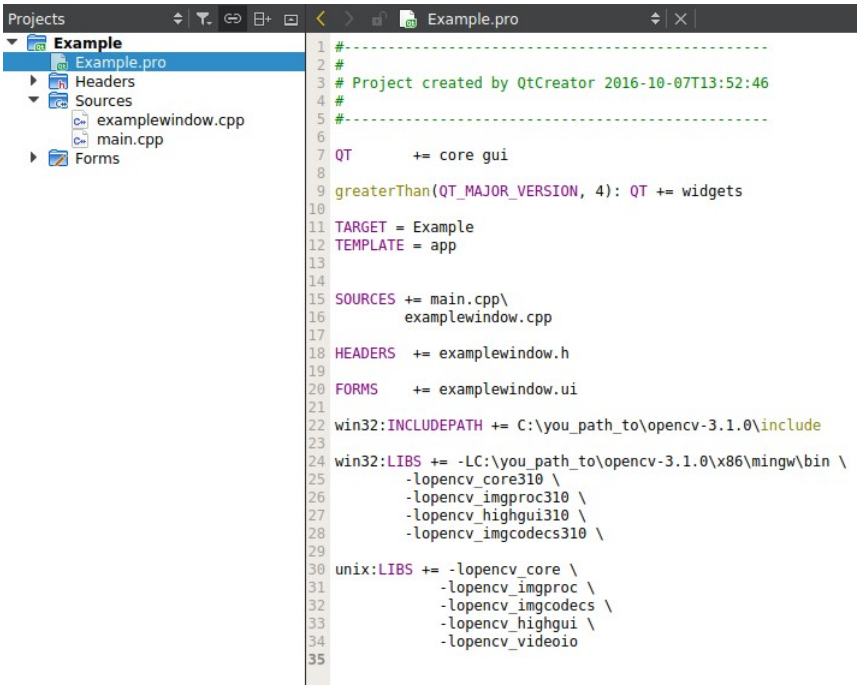

Abbildung 5: .pro File Beispiel mit einigen OpenCV Libraries

3.2 Einbinden einer neuen Klasse in ein bestehendes Projekt

Um eine weitere Klasse (Funktion) zum Projekt hinzu zu fügen, kann man mit Rechtsklick auf das Projekt „Add new“ auswählen. Es öffnet sich ähnlich wie beim Anlegen eines neuen Projekts ein Menü (siehe Abbildung 6). Je nach Bedarf kann man z. B. eine C++ Klasse oder

eine Qt-Klasse hinzufügen. Dies ist davon abhängig, ob Sie z. B. einen Dialog hinzufügen wollen (QDialog Class), ein komplett neues Fenster geöffnet werden soll (QWidget Class) oder Sie nur eine Klasse für verschiedene Funktionen und Algorithmen brauchen (C++ Class).

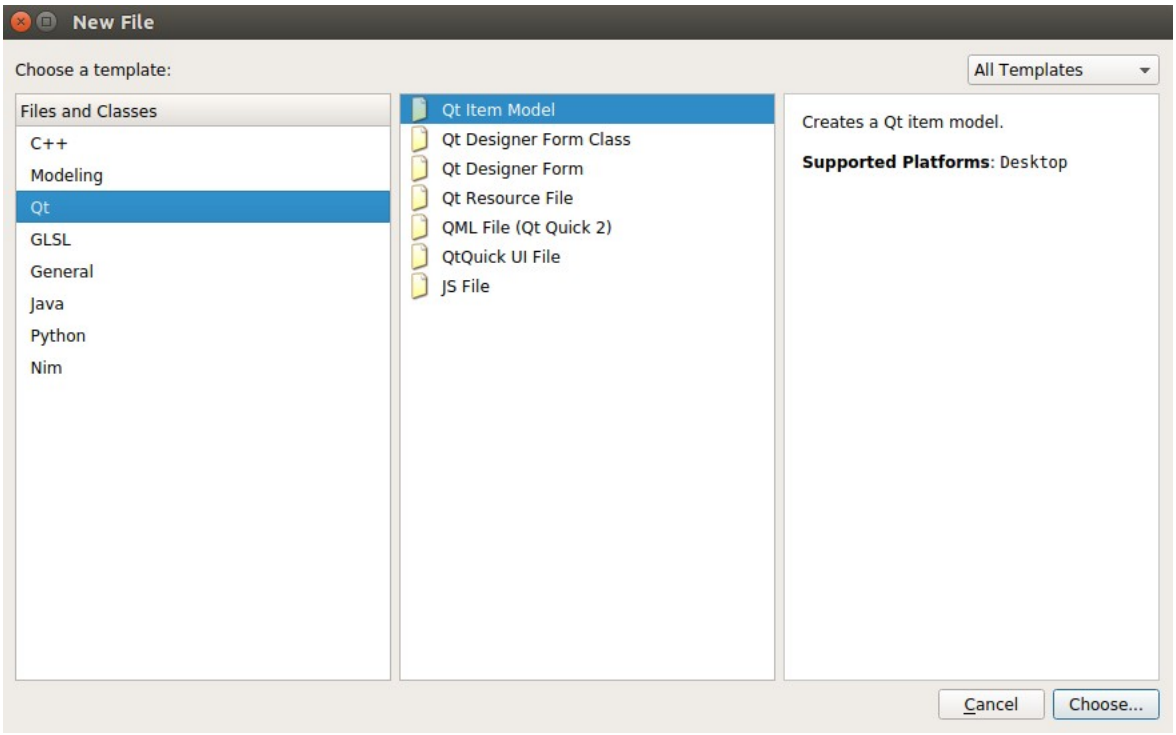

Abbildung 6: Menü zum Einbinden von Dateien

Wählen Sie die gewünschte Klasse aus und klicken Sie „Choose“. Im Anschluss müssen Sie die Klasse noch benennen und können Einstellungen zur Vererbung vornehmen. Das Menü ist dabei ähnlich wie beim Anlegen eines Projektes. Sind alle Einstellungen vorgenommen, klicken Sie „finish“.

Qt sollte automatisch alle nötigen Eintragungen vornehmen. Schauen Sie noch einmal in die eben schon erwähnte Projektdatei (.pro), Sie sollten nun sehen, dass die Dateien ihrer neuen Klasse hier ergänzt wurden. Die Dateien werden mit ein paar Standardzeilen von Qt automatisch gefüllt, diese unterstützen die Konfiguration und erleichtern Ihnen die Arbeit.

Um nun weitere Grundlagen zu erlernen, können Sie Tutorials von Qt selber nutzen. Unter `www.qt-project.org` finden Sie verschiedene Themen, die sie nach Bedarf testen können. Sie können im Projektverlauf hilfreich sein. Falls Sie es nicht bereits getan haben, bearbeiten Sie einmal unter „Getting Started“ das Tutorial zu den Widget Projects.

3.3 GUI-Entwicklung mit QT

Neben dem von Qt bereitgestellten Tutorial, werden hier einige Grundlagen vermittelt, die zumindest für den ersten Einstieg helfen. Das QT-GUI Tool ist leicht zu verstehen und zu handhaben. Um mit einer GUI zu arbeiten müssen Sie als Projektart ein „Widget Project“ wählen. Die GUI Dateien sind unter „Forms“ aufgelistet und haben .ui als Endung.

Zu jeder eingefügten QtWidget oder Dialog Klasse im Projekt gehört eine .ui Datei. Bei Erstellung eines Widget Projekts enthält die zu Beginn angelegte Klasse den Startbildschirm. Sie kann ebenfalls editiert werden. Durch Doppelklick öffnet sich der Qt Designer wie in Abbildung 7. Hier finden Sie links verschiedene Elemente, die Sie zu Ihrer GUI hinzufügen können und rechts oben die aktuell vorhandenen Objekte sowie rechts unten die Eigenschaften des aktuell ausgewählten Objekts.

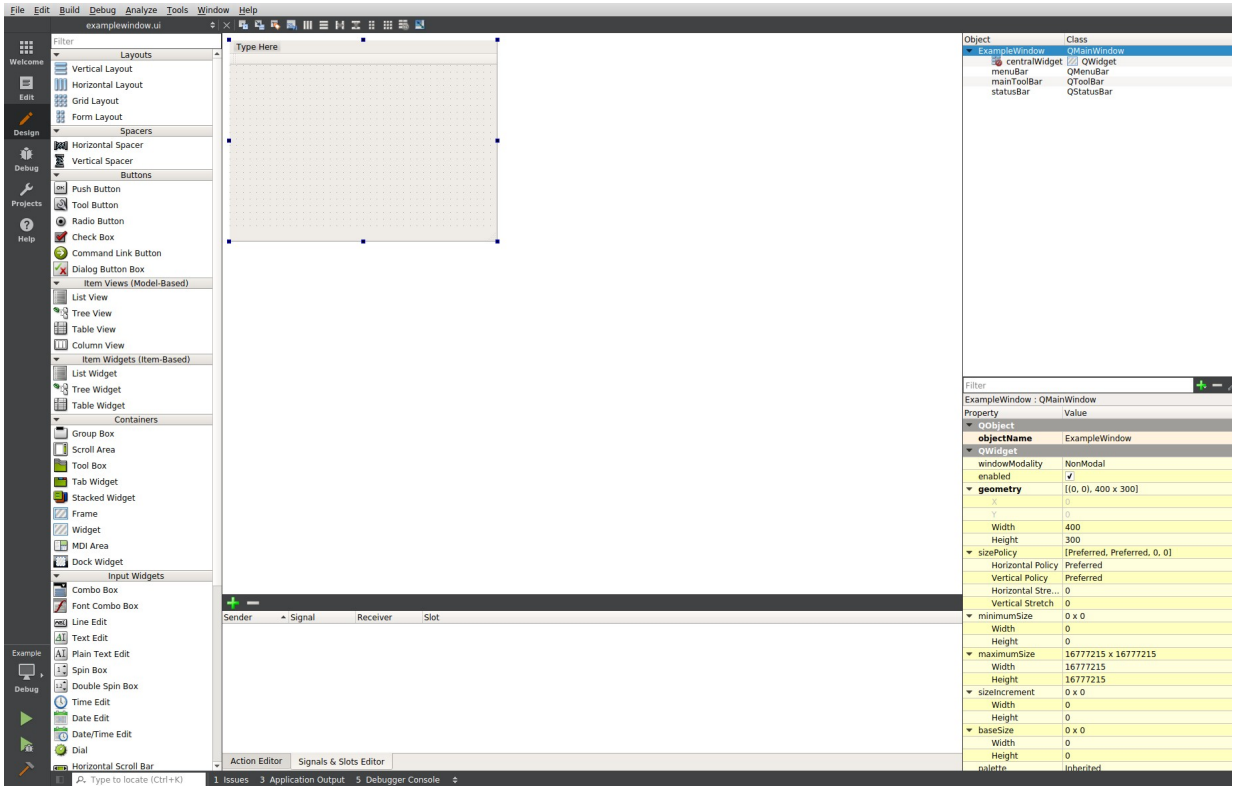

Abbildung 7: UI Designer von QT

Schauen wir uns das Prinzip wie Qt mit GUIs arbeitet einmal am Beispiel eines Buttons und einer einfachen Texteingabe/-ausgabe an. Dazu müssen zunächst ein Button und die Textfelder hinzugefügt werden. Benennen Sie die Elemente mit sinnvollen Namen um Sie später auseinander halten zu können (z.B. pb_Name für einen Button und le_name für ein Edit-Feld).

Ihre GUI könnte wie in Abbildung 8 aussehen.

Um den Button mit Funktionen zu hinterlegen, gehen Sie mit einem Rechtsklick auf den But-

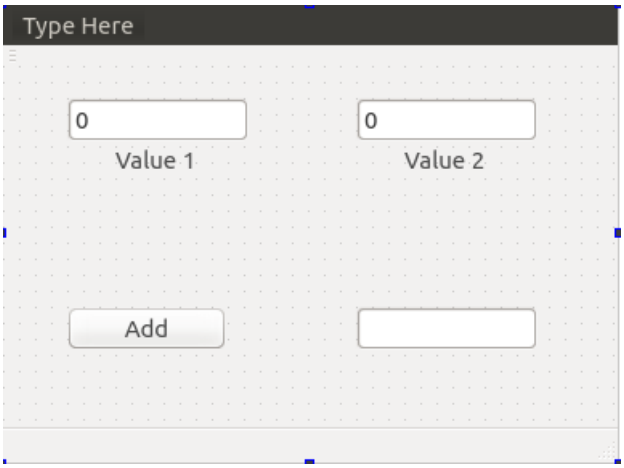

Abbildung 8: Beispiel GUI in QT

ton und wählen Sie „Go to Slot“aus. Sie bekommen eine Auswahl möglicher Slots angezeigt, aus denen Sie den von Ihnen gewünschten wählen. In dem Beispiel wird „clicked()“ gewählt. Klicken Sie auf „OK“ und Sie wechseln automatisch in Ihre zugehörige Source-Datei. Dort wird direkt die entsprechende Funktion angelegt. In der Header-Datei wird die Funktion ebenfalls automatisch eingetragen.

In diese Funktion tragen Sie nun ein was passieren soll, wenn der Button gedrückt wird. Im Beispiel sollen die beiden oberen LineEdit-Felder ausgelesen, die Werte daraus addiert und das Ergebnis in der unteren LineEdit-Zeile angezeigt werden. Unten können Sie die Befehle dem Beispielcode entnehmen. (Testen Sie es ruhig einmal selbst aus!)

```
1  void bvProject1::on_pb_calc_clicked()
2  {
3      double val1, val2, finVal;
4      QString finString;
5
6      //Auslesen der beiden Werte in den oberen LineEdit-Feldern
7      val1 = ui->le_value1->text().toDouble();
8      val2 = ui->le_value2->text().toDouble();
9
10     //Berechnung und Umwandlung in QString
11     finVal = val1 + val2;
12     finString = QString::number(finVal);
13
14     //Ausgabe des Ergebnisstrings in LineEdit
15     ui->le_final_val->setText(finString);
16 }
```

In den Praktikumsunterlagen finden Sie ein komplettes Qt-Projekt mit einer GUI-Funktion, die bereits einige OpenCV Elemente enthält. Diese ist ausreichend kommentiert, um die Grundlagen zu verstehen. Weitere Hilfestellungen, z.B. wie Sie in der GUI etwas ansprechen oder auslesen können, finden Sie in der Qt Dokumentation selbst. Sie können sich zudem jederzeit durch den Punkt-Operator mögliche Funktionen vorschlagen lassen, meist ist es logisch welche verwendet werden sollte. Für mehr Routine können Sie auch das Widget Tutorial nutzen.

4 OpenCV

Der Name OpenCV steht für Opensource Computer Vision. Wie leider häufig bei Open Source Software ist die Dokumentation teilweise lückenhaft, die meisten Sachen sind aber für unseren Zweck ausreichend unter `http://docs.opencv.org/3.2.0/index.html` beschrieben. Für weitere Hilfestellungen bietet sich Stackoverflow an, wo es ein breites Spektrum an Beiträgen zu OpenCV gibt (Google bietet hier auch viel Hilfe ;-)). Die grundlegenden Funktionen, die Sie für dieses Praktikum brauchen, werden in diesem Handbuch ausführlich erklärt.

OpenCV stellt selbst auch eine GUI zur Verfügung, diese wird im Praktikum jedoch nicht unterstützt. Im Kapitel zu Qt haben sie bereits alle GUI-Funktionalitäten erklärt bekommen.

4.1 Grundlagen

OpenCV stellt eine ausführliche Matrix Struktur zur Verfügung. Sie ist ähnlich zu mehrdimensionalen Arrays und erleichtert das Rechnen auf Bildern. Wir werden Matrizen hauptsächlich für die Arbeit mit Bildern verwenden. Matrizen kommen daher später noch einmal bei den Grundlagen zur Bildverarbeitung vor.

Die Datenstruktur cv::Mat enthält verschiedene Funktionen, wie z. B. size oder clone. Die Details dazu und welche weiteren Funktionen es gibt, finden Sie in der zugehörigen Dokumentation unter OpenCV: (`http://docs.opencv.org/3.2.0/d6/d6d/tutorial_mat_the_basic_image_container.html`)

Den Datentyp Vektor kennen Sie bereits aus GIP und ADS. In diesem Praktikum werden Sie mehrfach Vektoren brauchen, daher hier nochmal eine beispielhafte Initialisierung, wobei die Größe des Vektors optional angegeben werden kann.

```
1 //Aufbau der Initialisierung:
2 vector<type> v(size);
3
4 //ein Beispielvektor der Größe 20
5 vector<float> f(20);
```

OpenCV verfügt zudem über verschiedene Operationen, um z. B. eine elementweise Multiplikation durchführen zu können. Relevant werden hier vor allem folgende Funktionen:

```
1 /* Zunächst definieren wir dazu verschiedene Operanden, mit denen die
     Funktionen getestet werden:*/
2 Mat a = (Mat <double>(3,3) << 1, 2, 1, 3, 5, 3, 4, 1, 4);
3 Mat b = (Mat <double>(3,3) << 2, 2, 2, 1, 0, 1, 3, 2, 3);
```

```
4
5  //Ergebnis Matrix ohne Initialisierung:
6  Mat dest;
7
8  add(a, b, dest);
9  subtract(a, b, dest);
10 multiply(a, b, dest, scale);
11 divide(a, b, dest, scale);
```

Weitere Parameter für diese Funktionen können Sie in der OpenCV Dokumentation nachlesen. Weitere Matrix-Berechnungsmöglichkeiten finden Sie unter folgendem Link `http://docs.opencv.org/3.2.0/dc/d84/group__core__basic.html`.

In Matlab gibt es den Doppelpunktoperator um Spalten oder Zeilen auszuwählen und Operationen darauf auszuführen. OpenCV stellt dies nicht direkt zur Verfügung, sondern bietet mehrere Funktionen dafür. Zu beachten ist aber, dass es für eine Spalte/Zeile eine andere Funktion als für einen Spalten-/Zeilenbereich gibt. Nachfolgend ein Codebeispiel, analog verhält es sich auch zu den anderen Funktionen. Weitere Informationen finden Sie auch hier in der OpenCV Dokumentation:

```
1  //Ein paar Operanden
2  Mat a = (Mat <double>(3,3) << 1, 2, 1, 3, 5, 3, 4, 1, 4);
3  Mat b = (Mat <double>(3,4) << 1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5);
4  vector<double> v(3);
5  for (size_t i = 0; i < vPoints.size(); ++i)
6      v.[i] = (float)(i*3);
7
8  //Die erste Spalte der Matrix soll nun mit dem Vektor multipliziert werden.
9  a.col(int x) * v;
10
11 //Multipliziere die Matrix a mit den ersten drei Spalten der Matrix b
12 a * b.colRange(0, 2);
```

In den Matlab Unterlagen sind auch die Funktionen zum Plotten von Graphen aufgelistet. Bei OpenCV gibt es solche Funktionen auch, da diese aber in den Praktika voraussichtlich nicht gebraucht werden, werden Sie hier nicht weiter vorgestellt. Die OpenCV eigenen Funktionen sind zudem viel komplexer als bei Matlab und ein einfacher Funktionsplot ist dadurch aufwändiger. Eine kleine Einführung finden Sie unter `http://docs.opencv.org/3.2.0/d3/d96/tutorial_basic_geometric_drawing.html`, weitere Informationen sind in anderen Kapiteln der Dokumentation zu finden.

4.2 Bilder und Farbtabellen erzeugen

Zum Einlesen von Bildern verwendet OpenCV ähnliche Funktionen wie Matlab. Ein großer Unterschied, den Sie dringend behalten sollten: OpenCV liest Bilder als BGR anstatt als RGB ein. Wenn Sie also Operationen auf den Farbkanälen machen wollen, beachten Sie die Reihenfolge oder transformieren Sie das Bild ggf. vorher.

Das Auslesen der Bildinformationen bedarf hier keiner Extra-Funktion sondern kann über die Member-Funktionen erfolgen. Beispiele dazu finden Sie im Code-Beispiel unten und unter `http://docs.opencv.org/3.2.0/db/deb/tutorial_display_image.html`.

Das Speichern von Bildern funktioniert wie bei Matlab mit imwrite.

In Tabelle 1 finden sie eine Übersicht über Dateiformate, die ein Bild haben kann und die Sie mit OpenCV einlesen können.

Format:	Formatname:	Varianten:	Bildtyp
bmp	Windows bitmap	unkomprimiert Bilder: 1-, 4-, 8-, 16-, 24-, 32-Bits; RLE (run length encoded) Bilder: 4-, 8-Bits	RGB, Indexbild
gif	Graphics interchange format	1- bis 8-Bit Bilder	Indexbild
jpg, jpeg	Joint Photographic Experts Group	Grauwertbilder mit 8- oder 12-Bit bei verlustbehafteter Kompression; 8-, 12-, 16-Bit Grauwertbilder mit verlustfreier Kompression	RGB, Grauwertbild
tif, tiff	Tagged Image File Format	1-, 8-, 24-Bit unkomprimierte Bilder; 1-, 8-, 24-Bit Bilder mit „packbits“ Kompression, 1-Bit Bilder mit CCITT Kompression, 16-Bit Grauwertbild, 16-Bit Indexbild, 48-Bit RGB-Bild	RGB, Indexbild, Grauwertbild
png	Portable Network graphics	1-, 2-, 4-, 8-, 16-Bit Grauwertbilder, 8- und 16-Bit Indexbilder und 24- oder 48-Bit RGB Bilder	RGB, Indexbild, Grauwertbild

Tabelle 1: Dateiformate

Zum Umwandeln der Bilder in ein Binärbild stellt OpenCV die Threshold Funktion zur Verfügung. Die genauen Eingaben dafür und auch noch weitere hilfreiche Funktionen dazu finden Sie unter: `http://docs.opencv.org/3.2.0/d7/da8/tutorial_table_of_content_imgproc.html`. Um ein Bild zu normieren müssen alle Werte auf den Bereich [0..1] reduziert werden, z. B. durch eine Division durch 255 bei einem Wertebereich von [0..255]. Hierzu können Sie die bereits bekannten Funktionen nutzen.

Die Tutorialseite und Dokumentation von OpenCV ist sehr ausführlich und hier werden nicht alle Funktionen im Detail vorgestellt. Dies soll Ihnen nur ein grundlegendes Wissen zu OpenCV

vermitteln und vor allem Ihnen zeigen, wo Sie die benötigten Informationen im Zweifelsfall finden.