

Praxis der Softwareentwicklung: 3D Reconstruction Framework from Multi-View Images (3D-MuVi)

Pflichtenheft

Gri Sha Nathanael Schneider Tim Brodbeck
Stefan Wolf Jens Manig Laurenz Thiel



WS 2015/16

Inhaltsverzeichnis

1	Einleitung	4
2	Zielbestimmung	5
2.1	Musskriterien	5
2.2	Kannkriterien	5
2.3	Abgrenzungskriterien	5
3	Produkteinsatz	6
3.1	Einsatzgebiet	6
3.2	Anwendungsbereich	6
3.3	Zielgruppe	6
3.4	Betriebsbedingungen	6
4	Produktumgebung	7
4.1	Software	7
4.2	Hardware	7
4.3	Schnittstellen	7
5	Produktfunktion	8
5.1	Grundfunktionen	8
5.2	Optionale Funktionen	8
6	Produktdaten	9
7	Systemmodell	10
8	Produktleistungen	11
9	Benutzeroberfläche	12
9.1	Hauptfenster	12
10	Globale Testfälle	13
10.1	Testfälle	13
10.2	Testszenarien	13
11	Qualitätszielbestimmungen	14
12	Entwicklungsumgebung	15
12.1	Allgemein	15

12.2 Implementierung	15
12.3 Validierung	15
12.4 ???	15
13 Glossar	16

1 Einleitung

Die Welt wird digital - und mit ihr viele Prozesse. Die Kommunikation läuft heutzutage größtenteils mit digitalen Hilfsmitteln. Auch Geschäftsprozesse werden immer mehr Digital abgewickelt, soweit möglich. Um dies zu bewerkstelligen, müssen die Daten aus der Realen Welt allerdings erstmal digitalisiert werden. Viel ist in dieser Richtung geschehen, wie z.B. Kameras, die Digitalisierung ganzer Bibliotheken und das Erfassen von Echtzeitdaten über Sensornetze sind nur ein Bruchteil. Doch das Einlesen komplexer Daten ist bis heute ein nicht-triviales Problem.

Viele Arbeitsbereiche erfordern eine Speicherung und Übertragung von Daten, wie sie im echten Leben vorzufinden sind - z.B. Dreidimensionale Bilder. In der Medizin werden sie benutzt, um Knochen und Skelette platzsparend verfügbar zu haben, in der Automobilindustrie benötigt man sie, um Prototypen schon vor dem ersten tatsächlichen Fertigungsschritt vor Augen zu haben. Besonders die Spielindustrie steigert ihre Anforderungen nach immer besseren und realistischeren Modellen.

Für viele dieser Anforderungen gibt es eine einfach erscheinende Lösung. Man nehme ein reales Objekt und digitalisiere seinen dreidimensionalen Aufbau. Doch leider ist das Problem nicht so einfach, wie es sich zuerst anhört. Es gibt viele verschiedene Verfahren um diese Aufgabenstellung und jedes einzelne bewährt sich besonders in einem spezifischen Problemfeld. Ein Beispiel ist Motion Capture, das besonders für die Rekonstruktion von Bewegungen im Dreidimensionalen Raum über die Zeit genutzt wird, Laserscanner können schnell und zuverlässig Personen vermessen. Dieses Projekt beschäftigt sich gezielt mit der Rekonstruktion von digitalen Repräsentationen der Objekte über zweidimensionale Aufnahmen des gewünschten Gegenstandes oder einer kompletten Szene.

Die für den Prozess verwendeten Algorithmen sind jedoch sehr spezialisiert und decken meist nur einen bestimmten Anwendungsfall ab. Zudem lassen sich für die einzelnen Schritte auch unterschiedliche Algorithmen kombinieren, was die Zusammenstellung für ein perfektes Ergebnis umso schwerer macht. So macht es z.B. einen Unterschied, ob man eine Szene aus einem Videoclip rekonstruieren möchte, ob man einzelne Fotos aus verschiedenen Blickrichtungen hat oder ob es sich um eine Luftaufnahme handelt.

Ziel des 3D-MuVi Projekts ist es, die Kombination von Algorithmen und das damit verbundene Suchen nach dem bestmöglichen Ergebnis zu vereinfachen. Der Benutzer bekommt die Möglichkeit, verschiedene Algorithmen für die einzelnen Schritte auszuwählen und sich die Ergebnisse anzusehen, um somit das bestmögliche Resultat zu erzielen.

2 Zielbestimmung

2.1 Musskriterien

- Standard 3-Workflow(Posenbestimmung, Tiefenschätzung, Fusion), mit verschiedenen Algorithmen pro Schritt
- Übergabe von Einzelbildern
- ...

2.2 Kannkriterien

- fünf verschiedene Workflows
- ...

2.3 Abgrenzungskriterien

- Algorithmen müssen nicht implementiert, nur eingebunden werden
- ...

3 Produkteinsatz

3.1 Einsatzgebiet

3.2 Anwendungsbereich

3.3 Zielgruppe

3.4 Betriebsbedingungen

4 Produktumgebung

4.1 Software

Das Programm soll auf linuxbasierten Betriebssystemen laufen, da es mit QT entwickelt und möglichst Plattform unabhängig entwickelt wird, kann es sein, dass es auch unter anderen Betriebssystemen läuft.

4.2 Hardware

Als Hardware wird ein einfacher PC benötigt, welcher die Softwarebedingungen erfüllt.

4.3 Schnittstellen

5 Produktfunktion

Die Beschreibung der Funktionalität gliedert sich in Grundfunktionen und optionale Funktionen.

5.1 Grundfunktionen

5.1.1 Funktional

\F1010\ Auswahl der Algorithmen Blablabla.

5.1.2 Nichtunktionalität

\F2010\ Blub Blaaa

5.2 Optionale Funktionen

\F3010\ ??? Blaaa

6 Produktdaten

Es werden folgende Daten gespeichert:

- \PD1010\ Zwischenergebnisse
- \PD1020\ Endergebnis
- \PD1030\ Parameter der einzelnen Algorithmen
- \PD1040\ Globale Einstellungen
- \PD1050\ Konfiguration des Workflows

7 Systemmodell

8 Produktleistungen

9 Benutzeroberfläche

9.1 Hauptfenster

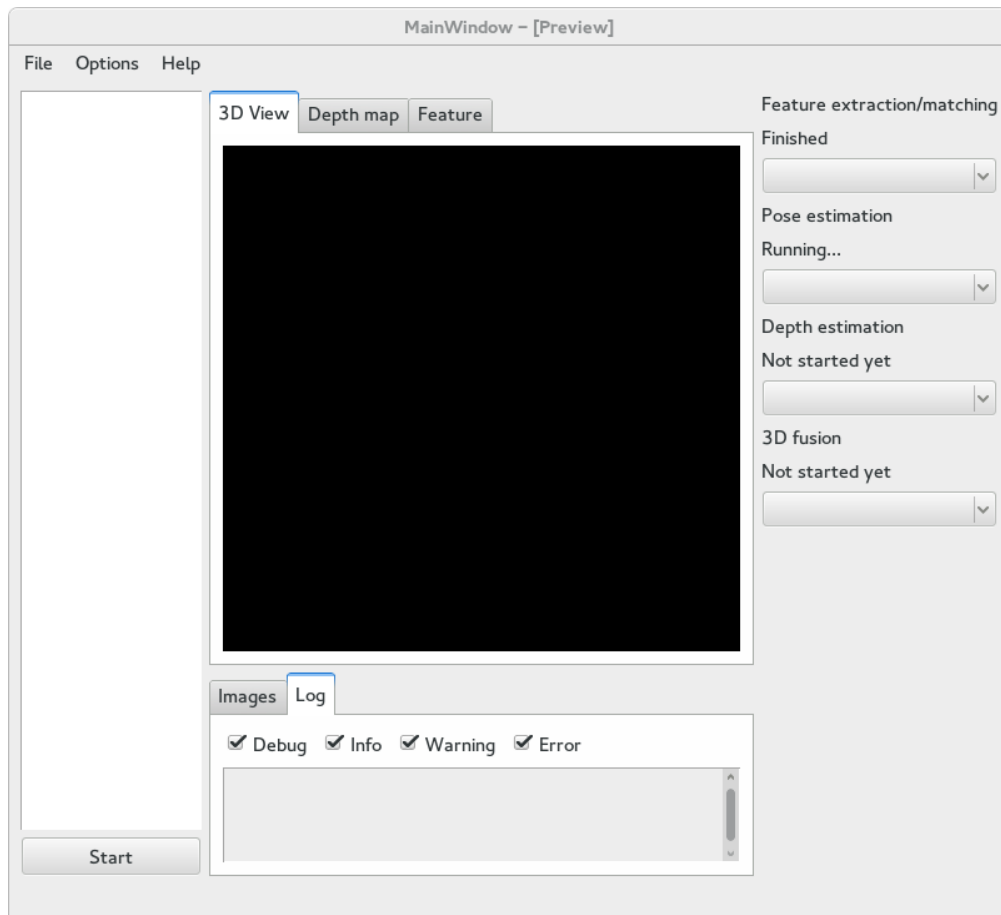


Abbildung 9.1: Hauptfenster

Algorithmuswahl Algorithmen können dem Workflow entsprechend aus ComboBoxen gewählt werden.

Bilderinput Eingabebilder werden als Thumbnails angezeigt.

10 Globale Testfälle

10.1 Testfälle

\T1010\ Start des Programms
\T1020\ ...

10.2 Testszenarien

- Testszenario 1
Blub

11 Qualitätszielbestimmungen

12 Entwicklungsumgebung

12.1 Allgemein

- L^AT_EX
- Versionskontrolle (Git)

12.2 Implementierung

- Qt Creator
- ...

12.3 Validierung

- QtUnit

12.4 ???

- Bla

13 Glossar

Point cloud Menge von Punkten eines Vektorraums.