

# **Pflichtenheft**

**Bachelor-Arbeit  
Christoph Gnip**

**Entwicklung eines 3D-Scanners zur berührungslosen  
Abtastung geometrischer Patientenkenndaten mittels  
bildgebender Verfahren**

**Projekttitel: 3D-Scan**

## Inhaltsverzeichnis

1 Allgemeines.....	4
1.1 Inhalt.....	4
2 Zielbestimmungen.....	5
2.1 Musskriterien.....	5
2.2 Wunschkriterien.....	5
2.3 Abgrenzungskriterien.....	6
2.4 Anwendungsbereiche.....	6
2.5 Zielgruppen.....	6
3 Systemkenndaten.....	7
3.1 Algorithmus.....	7
3.1.1 Simulation des Algorithmus.....	7
3.1.2 Verifizierung des Algorithmus und Vermessung der Szene.....	7
3.2 Hardware.....	8
3.2.1 Minicluster (Kameraanordnung und Beleuchtung).....	8
3.2.1.1 Szenenbeleuchtung und Schattenerzeugung.....	8
3.2.3 Hardwareplattform.....	8
3.3 Software.....	9
3.3.1 Übertragung zum PC.....	9
3.3.1.1 Serverprogramm.....	9
3.3.1.2 Clientprogramm.....	10
3.3.2 Simulationsdaten.....	10
4 Benutzeroberfläche.....	11
4.1 Benuztermodell.....	11
4.2 Kommunikationstrategie.....	11
4.3 Kommunikationsaufbau.....	11
4.4 Benuzterdokumentation.....	11

5. Qualitätsziele.....	11
5.1 Recherchen und Stand der Technik.....	11
5.2 Dokumentation.....	12
6 Entwicklungswerkzeuge.....	12
6.1. Hardware-Entwicklungswerkzeuge.....	12
6.2 Software-Entwicklungswerkzeuge.....	12
7. Projektmanagement.....	13
8. Anhänge.....	14
8.1 Abkürzungen.....	14
8.2 Glossar.....	14
8.3 Literaturquellen.....	14

# 1 Allgemeines

Dieses Dokument enthält die ausführliche Beschreibung der Bachelorarbeit von Christoph Gnip. Das Dokument ist in Form eines Pflichtenheftes aufgebaut und basiert auf dem Lastenheft DokNr.: 012-437-0301-06.

## 1.1 Inhalt

Bildgebende Verfahren stellen in der modernen Medizin ein wesentliches Werkzeug zur Diagnostik und Kenndatenaufzeichnung dar. Während bildgebende Systeme wie Magnet-Resonanz-Tomografie oder Ultraschall-Geräte zur Diagnostik dienen, werden 2D-Scanner oder 3D-Scanner verwendet, um Kenndaten wie geometrische Dimensionen, Position oder Lage von Patienten oder Instrumenten zu ermitteln. Scanner werden zum Beispiel mit Digitalkameras und bewegten Laserpunkten realisiert.

Mit der Bachelorarbeit von Herrn Gnip soll ein Beitrag zur Entwicklung eines neuartigen 3D-Scanners geleistet werden. Der Scanner besteht aus einer Vielzahl von Bildsensoren, die zu einer Multisensorkamera bzw. einem Cluster zusammengeschaltet sind. Jeder Cluster betrachtet eine Szene, in welcher sich beispielsweise Instrumente oder Patienten oder beide befinden und geometrische Kenndaten der Szene ermittelt werden müssen. Jeder einzelne Bildsensor betrachtet hierzu einen Teilausschnitt der Szene, die sich mit der Teilszene des benachbarten Sensors überschneidet.

## 2 Zielbestimmungen

Die nachfolgenden Formulierungen der Zielbestimmungen beschreiben die Aufgabenstellung aus Sicht des Auftragnehmer.

### 2.1 Musskriterien

Als Entwicklungsplattform für das FPGA ist die Entwicklungsumgebung Quartus II des Hersteller ALTERA zu verwenden. Die Verifikation des Algorithmen soll unter Quartus II durchgeführt werden. Hierzu ist ein Tool zu entwickeln, das Bitmap-Bilddateien als Stimulidaten für den Quartus-Simulator erzeugt und umgekehrt aus den Simulationsergebnissen entsprechende Bilder im Bitmap-Format erzeugt. Die Algorithmen sind so auszuführen, dass die konform sind, mit den Software-Modul-Schnittstellen (SMS) der Entwicklungsumgebung MiniQuadrix.

Die Ergebnisse der Arbeit sollen geeignet sein, einen beliebig großen Cluster  $n \times m$   $\{n, m = 1 \dots 216\}$  zu betreiben. Das bedeutet, in den Algorithmen sind ggf. Nachbarschaftsverhältnisse anderer Bildsensoren zu berücksichtigen.

Zur Softwareentwicklung sind ausschließlich die bereits vorhandenen Hardwaredreiber der Baugruppe zu verwenden. Sollten diese nicht genügen ist Kontakt zu Herrn Brinker auszunehmen.

### 2.2 Wunschkriterien

Zur Veranschaulichung der Rechenergebnisse ist eine 3D- Darstellung der Ergebnisse in Falschfarben auf einem Display hilfreich. Sofern es der Zeitrahmen ermöglicht, ist vom Auftragnehmer eine entsprechende Software

zu entwickeln, um die Daten über den VGA- Anschluss des MiniQuadrix- Moduls oder via TCP/ IP auf einem PC darzustellen. Als Datenformat auf dem PC wäre zudem ein STL-CAD-Format wünschenswert.

## **2.3 Abgrenzungskriterien**

Algorithmen und Testanordnungen welche bereits in der Arbeitsgruppe BuS entwickelt wurden sind einzusetzen und nicht neu zu erfinden. Es dürfen/ sollen alle bereits erbrachten Arbeitsergebnisse der Arbeitsgruppe BuS genutzt werden und in der Bachelorarbeit von Herrn Gnip mit entsprechender Quellenangabe verwendet werden.

## **2.4 Anwendungsbereiche**

Die Baugruppe des Miniclusters ist lediglich so zu konzipieren, dass sie ausschließlich für den Laborbetrieb geeignet ist.

## **2.5 Zielgruppen**

Die Versuchsanordnung und Algorithmen sind für Teammitglieder der Arbeitsgruppe BuS auszuführen.

## **3 Systemkenndaten**

### **3.1 Algorithmus**

Der in dieser Arbeit verwendete Algorithmus zur Gewinnung der 3D-Szeneninformation ist das sog. Shape from Shadows- Verfahren (auch Shape from Darkness bezeichnet). Es ermöglicht, in Hardware ausgeführt, die Akquirierung der 3D-Daten in Echtzeit. Der Algorithmus wird mit Blick auf ein mögliches größeres Sensorcluster implementiert. Das bedeutet, dass Nachbarschaftsverhältnisse und Überlappungen des Sichtbereichs berücksichtigt werden. Um viele Informationen aus den Daten gewinnen zu können, wird die Lichtquelle zur Schattenerzeugung entlang einer kreisförmigen Trajektorie „bewegt“. Technisch wird die Bewegung von mehreren ortsfesten LED' s umgesetzt um die Szene zu beleuchten. Das Ergebnis der Rekonstruktion ist von der gewählten Trajektorie abhängig. Eine optimale Bahn ist daher zu finden, entsprechend werden die LED' s am Sensorcluster angeordnet.

#### **3.1.1 Simulation des Algorithmus**

Für die Simulation des Hardwarealgorithmus wird das Simulationswerkzeug der Firma ALTERA eingesetzt. Es wird entsprechend in Betrieb genommen und beschleunigt die Entwicklung der Hardwarealgorithmen.

#### **3.1.2 Verifizierung des Algorithmus und Vermessung der Szene**

In den ersten Entwicklungsschritten werden die Ergebnisse der Simulation des Hardwarealgorithmus mittels Matlab R2010a zusätzlich

verifiziert und die Szene vermessen. Mit diesem Werkzeug ist es sehr einfach die Szenendaten zu visualisieren und im Anschluss zu Vermessen.

## **3.2 Hardware**

Die Hardware dient dazu die Bilddaten zu verarbeiten und mit dem gewählten Algorithmus auszuwerten. Es werden im Versuchsaufbau zunächst vier Kameras verwendet. Die Beleuchtung der Szene übernehmen mehrere Leuchtdioden.

### **3.2.1 Minicluster (Kameraanordnung und Beleuchtung)**

Für das Minicluster werden vier Kameramodule des Typs: Kamerakopf KK 752 so angeordnet, dass sie die Szeneninformationen aufnehmen können und eine Weiterverarbeitung möglich ist. Die Anordnung entspricht einem 2x2 Cluster. Am Cluster sind zusätzlich, gemäß der Forderung im Lastenheft, Leuchtdioden angebracht.

#### **3.2.1.1 Szenenbeleuchtung und Schattenerzeugung**

Die Leuchtdioden dienen der gezielten Schattengenerierung für die 3D-Rekonstruktion. Sie werden in einer Anordnung mit den Kameras platziert. Die Ansteuerung wird manuell erfolgen. Um ein gutes Rekonstruktionsergebnis zu erzielen sollen mit der Anordnung verschiedene Trajektorien der Lichtquelle möglich sein. Aufgrund der Forderung nach einem hellen um homogenen Lichtkegel werden Power- LED' s verwendet.

#### **3.2.3 Hardwareplattform**

Als Modul für die Auswertung wird das System MiniQuadrix II



eingesetzt. Es zeichnet sich durch einen Leistungsstarken FPGA der Cyclone Familie aus. Auf dem Modul werden die Hardwarealgorithmen implementiert. Auf der MiniQuadrix II steht ein Betriebssystem (ucLinux) und eine API zu Verfügung, daher ist es sehr gut als Projektplattform geeignet. Sie ist in der Lage die vier Kameras des Sensorclusters direkt zu betreiben.

### **3.3 Software**

Für die Anforderungen müssen mehrere kleinere Programme entwickelt werden. Die Programmiersprache ist für alle Programmteile des Embedded-Systems C und konform zum C99-Standard. Die Clientprogramme werden in C++. Ein Win32 Client wird voraussichtlich mit C# und der .Net Umgebung realisiert. Für die Versionsverwaltung wird ein git-Verzeichnis angelegt. In diesem werden alle Programmteile gesammelt.

#### **3.3.1 Übertragung zum PC**

Gemäß den Wunschkriterien wird ein Mechanismus implementiert, der die gewonnen Szeneninformationen auf einen PC überträgt und dort darstellt. Auf dem PC wird zur Darstellung ein beliebiges 3D-Modellierungswerkzeug verwendet werden (z. B. Solidworks, 3ds Max etc.). Die Übertragenen Szeneninformationen werden in einem dazu kompatiblen Format (voraussichtlich Heightmap siehe [WIKI01]) gespeichert.

##### **3.3.1.1 Serverprogramm**

Auf dem MiniQuadrix- Modul wird es einen Serverdienst geben. Er kann die gewonnenen Informationen an einen PC auf Anfrage übertragen. Der

Server wird über ein einfaches Protokoll verfügen, das es möglich macht die Szeneninformationen zu einem PC zu übertragen. Das Programm wird abgeleitet aus dem Projekt: MiniQuadrix-Server, das bereits in der Arbeitsgruppe BuS erstellt wurde. Das MiniQuadrix- Protokoll verwendet TCP/ IP zur sicheren Übertragung. Nähere Angaben zum Protokoll und Funktionen des Servers sind der Dokumentation des MiniQuadrix- Servers [MQSDOKU] zu entnehmen. Der Server ermöglicht außerdem die Parametrierung des Moduls, wie die Einstellung der Kameras und Zugriff auf Register des Moduls von der Clientsoftware aus.

### **3.3.1.2 Clientprogramm**

Im ersten Schritt wird ein Linux Client erstellt. Wie bereits der Server wird es von dem Client des MiniQuadrix-Server Projektes abgeleitet, das beschleunigt die Entwicklung.

Wenn es die Entwicklungszeit zulässt wird ein Win32-Client erstellt. Die Entwicklungsumgebung für diesen Client ist das Visual Studio 2008.

### **3.3.2 Simulationsdaten**

Für die Erstellung des Simulationsdaten aus einem Bild im Bitmap Format wird es ein zusätzliches Programm geben. Voraussichtlich wird es durch eine Win32 Anwendung realisiert. Die Entwicklungsumgebung ist auch hierfür das Visual Studio 2008.

## **4 Benutzeroberfläche**

### **4.1 Benutztermodell**

Es liegen keine Anforderungen vor.

### **4.2 Kommunikationstrategie**

Um die Daten vom Modul zu einem PC zu übertragen wird das MiniQuadrix- Protokoll eingesetzt, siehe Dokumentation [MQSDOKU].

### **4.3 Kommunikationsaufbau**

Über den Aufbau der Kommunikation zwischen Client und Server gibt die Dokumentation [MQSDOKU] Auskunft.

### **4.4 Benutzterdokumentation**

Die Dokumentation erfolgt in Form der schriftlichen Bachelorarbeit nach den Maßgaben in Kap. 5.2.

## **5. Qualitätsziele**

### **5.1 Recherchen und Stand der Technik**

Die Daten, die zur Recherche herangezogen werden, sollten möglichst aktuell sein. Der Konzeptentwurf sollte mit Hilfe modernster Notation ansprechend gestaltet werden. Die Umsetzung sollte zuverlässig sein. Die genutzten Softwarewerkzeuge sollen zukunftsorientiert ausgewählt werden.

## **5.2 Dokumentation**

Sämtliche Textdokumente sind kompatibel zu Microsoft Word 2003 oder Word 2007 zu erstellen. Als Formatvorlage ist das Dokument [How10-2] zu verwenden.

## **6 Entwicklungswerkzeuge**

Die Werkzeuge mit denen die Hard- und Software entwickelt wird werden so ausgewählt das sie zukunftssicher sind oder mindestens ein einfache Portierung ermöglichen. Alle unten aufgeführten Programme liegen entsprechend in der Arbeitsgruppe vor.

### **6.1. Hardware-Entwicklungswerkzeuge**

Für der FPGA- Hardwarealgorithmen werden die in der Arbeitsgruppe BuS verwendeten Programme des Herstellers ALTERA benutzt. Auch der verwendete Simulator ist Teil dieser Programmsuite. Das für die Verifikation des Algorithmus vorgesehene Programm Matlab liegt in der Version 7.10 (R2010a) vor.

### **6.2 Software-Entwicklungswerkzeuge**

Für die Entwicklung des Embedded-Programmcodes in C wird der Open-Source-Editor Kate verwendet. Als Compiler dient der von ALTERA bereitgestellte gcc. Das Linux Clientprogramm wird mit dem selben Editor entwickelt und entsprechen kompiliert. Für die Entwicklung des Win32- wird das Microsoft Visual Studio in der Version 2008 verwendet.

## 7. Projektmanagement

Projektlaufzeit: 12 Wochen.

Beginn: 20. KW

Vorgänge	Ressource	KW											
		20	21	22	23	24	25	26	27	28	29	30	31
Recherche		X	X										
Bestellung LEDs	FH-Ge		X										
Schaltungsplanung	FH-Ge		X	X									
Schaltungsanordnung				X									
Algorithmuserstellung				X	X	X							
Programmieren (Stimulidaten)				X	X								
Algorithmusverifikation						X	X						
Programmieren (Server)							X	X					
Programmieren (Client/Linux)							X	X					
Programmieren (Client/Win32)									X	X			
Dokumentation									X	X	X	X	X
Präsentation	Beamer												X

Tabelle 1: Projektablaufplan

## 8. Anhänge

### 8.1 Abkürzungen

KW	Kalenderwoche
FPGA	Field Programmable Gate Array
TCP/ IP	Transmission Control Protocol/ Internet Protocol
FH- Ge	Fachhochschule Gelsenkirchen
BuS	Bauelemente und Schaltungstechnik

### 8.2 Glossar

Wikipedia	Freie Enzyklopädie
Matlab	Software zur Lösung mathematischer Probleme und zur grafischen Darstellung
.Net	Laufzeitumgebung (Framework) von Microsoft
git	Freie Software zur verteilten Versionsverwaltung von Dateien (insbesondere gut geeignet für Quellcode)

### 8.3 Literaturquellen

[MQSDOKU] Gnip, C.; Dokumentation Miniquadrix Protokoll;  
DokNR.: 012-605-03-01\_13\_dokumentation-miniquadrix\_protokoll.pdf

[How10-2] Howah, L.; Hausinterne Word-Vorlage zur Erstellung von Examensarbeiten; DokNr. 012-400-2902-00 Vorlage Wochenbericht.doc

[WIKI01] Wikipedia, Heightmap/ Höhenfeld  
<http://de.wikipedia.org/w/index.php?title=Höhenfeld&oldid=57498257>

## Alphabetischer Index

### A

Abgrenzungskriterien.....	2, 6
Algorithmus.....	2, 7f., 12f.

### M

Minicluster.....	2, 6, 8
MiniQuadrix.....	5f., 8ff.
Musskriterien.....	2, 5

### P

Programm.....	2, 9f., 12ff.
Protokoll.....	10f., 14

### S

Shape from Shadows.....	7
Software.....	2f., 5, 9ff., 14

### W

Wunschkriterien.....	2, 5, 9
----------------------	---------