

**Fachhochschule Gelsenkirchen  
Neidenburger Straße 43  
45897 Gelsenkirchen**

## **Bachelor-Arbeit**

**Christoph Gnip**

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

**Projekttitel: 3D-Scan**

**Abgabedatum: 2010-08-23**

Erstprüfer: Prof. Dr.-Ing Lothar Howah

Zweitprüfer: Prof. Dr.-Ing Christian Schröder

## Erklärung

Ich versichere an Eides Statt, dass ich diese Examensarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

.....  
Datum                      Unterschrift

## Kurzreferat

Das vorliegende Dokument enthält die Beschreibung zum Projekt 3D-Scan. Das Projekt ist die Abschlussarbeit des Autors in dem Studienfach Medizintechnik des Fachbereiches Physikalische Technik an der Fachhochschule Gelsenkirchen. Es beschreibt die Umsetzungsdetails und Probleme der im Pflichtenheft [GNI10-7] vereinbarten Punkte.

Ziel dieser Abschlussarbeit war es zum einen eine Miniaturversion eines Sensorclusters zu erstellen und zum anderen einen Algorithmus für diesen Sensorcluster (Multisensor-Kamera) zu entwickeln. Der Algorithmus soll in weiteren Schritten auf einer Hardwareplattform ausgeführt werden. Diese Hardwareplattform besteht aus programmierbaren Logikschaltkreisen (FPGA) und ist eine Entwicklung des Labors für Bauelemente und Schaltungen. Die Baugruppe des Sensors soll es ermöglichen, geometrische Patientenkenndaten zu erfassen. Die Erfassung erfolgt dabei berührungslos. Dazu werden die Bilddaten der Kameras ausgewertet.

Die geometrischen Informationen werden aus Schatten gewonnen, welche in der Szene vorkommen. Die Schatten werden durch eine entsprechende Beleuchtung künstlich erzeugt. Die für die Informationsgewinnung aus den Schatten notwendige Theorie wird in diesem Dokument ebenfalls behandelt.

Zum Schluss des Dokumentes werden die Ergebnisse der Arbeit präsentiert und ausführlich diskutiert. Verbesserungsvorschläge und Erfahrungen aus dieser Arbeit geben einen Ausblick für anschließende Arbeiten.

## Anerkennung

An dieser Stelle soll denjenigen Dank ausgesprochen werden, die mich während der Zeit an diesem Projekt begleitet, unterstützt, beraten und betreut haben.

Die Bedingungen in der Arbeitsgruppe BuS möchte ich dabei hervorheben. Vielen Dank also an Daniel und Marco. Auch Herrn Prof. Dr. Howah möchte ich hiermit danken, dafür dass Sie es mir ermöglichen ein Teil der Arbeitsgruppe zu sein und soviel wertvolle Erfahrung zu gewinnen.

Mom und Gerry, meinen besonderen Dank möchte ich euch aussprechen. Ihr seid hier zwar zuletzt genannt, jedoch nicht weniger wichtig. Nicht nur durch eure Korrekturlesungen habt ihr wahrscheinlich sogar den wichtigsten Anteil daran, dass diese Arbeit nun gebunden im Regal stehen kann.

## Inhalt

Verzeichnis der Bilder .....	7
Verzeichnis der Tabellen .....	9
Verzeichnis der Anhänge .....	9
Liste der Symbole .....	10
Liste der Abkürzungen .....	12
1 Einleitung .....	13
1.1 Motivation .....	13
1.4 Zielsetzung .....	13
1.5 Aufbau der Arbeit .....	14
2 Zielbestimmungen .....	14
2.1 Musskriterien .....	14
2.2 Wunschkriterien .....	14
2.3 Abgrenzungskriterien .....	15
2.4 Anwendungsbereiche .....	15
2.5 Zielgruppen .....	15
3 Shape from Shadows Algorithmus .....	15
3.1 Einleitung .....	16
3.2 Informationen aus den Schatten .....	16
3.2 Präsentation der Daten .....	21
3.4 Abgrenzung zu anderen Umsetzungen .....	22
3.4 Eignung für eine hardwaregestützte Bildverarbeitung .....	22
3.5 Anpassungen des Algorithmus an die Hardware .....	23
4 Hardware .....	24
4.1 MiniQuadrix II .....	25
4.2 Kamerasensoren .....	26
4.3 MiniCluster .....	28
4.4 Szenenbeleuchtung und Schattenerzeugung .....	31
4.5 Optik .....	31
4.6 Sichtbereiche und Überlappungen .....	34
4.7 Montageplatte .....	37
4.8 Kalibrierung .....	37
4.9 Objekte .....	38
5 Software .....	38
5.1 Umgebung .....	39
5.2 Software der MQ II- 3dscan_s .....	39
5.2.1 Übertragung zum PC .....	39

5.4 Algorithmus in Software .....	40
5.5 Portierung des Algorithmus auf die Hardware .....	40
6 Auswertung der Bilddaten .....	41
7 Ergebnisse.....	46
7.1 MiniCluster .....	46
7.2 3D-Szenenrekonstruktion.....	47
7.3 Probleme.....	50
8. Abschluss .....	52
8.1 Reflektion und Diskussion.....	52
8.2.Ausblick.....	53
9. Appendix.....	56
9.1 Begriffserläuterungen .....	56
9.2 Eingesetzte Software .....	60
9.3 Glossar.....	61
10 CD-Inhalt .....	81
Stichwortverzeichnis .....	82
Literaturverzeichnis .....	83

## Verzeichnis der Bilder

Abbildung 1: Shape from Shadows- 2D-Darstellung einer Struktur mit Lichteinfall ..	17
Abbildung 2: [DAU98-15] 2D Shadowgram .....	19
Abbildung 3: Kugelkoordinaten.....	19
Abbildung 4: [DAU98-15] a) Sandwich Cross Section b) Tunnel Cross Section .....	20
Abbildung 5: a) Höhenfeld 2D Darstellung b) gerenderte Ansicht des Höhenfeldes.	22
Abbildung 6: Bearbeitungsorte der Funktionsgruppen; Schema.....	24
Abbildung 7: MiniQuadrix II.....	25
Abbildung 8: Bildsensor, Typ KK 752.....	26
Abbildung 9: Typische Quantum Effizienz für IR.....	27
Abbildung 10: IR-Filter, Montage im Objektiv.....	27
Abbildung 11: Kameraanordnung; Darstellung der Sichtbereiche mit Überlagerungen .....	30
Abbildung 12: Nachbarschaftsverhältnisse.....	34
Abbildung 13: Allgemeiner Fall für einen quadratischen Sichtbereich .....	35
Abbildung 14: Änderung des Beleuchtungswinkels .....	36
Abbildung 15: Originalszenen der in dieser Arbeit präsentierten Bildserien 5 und 7 a) sowie 16 b) .....	38
Abbildung 16: Zusammenführung der Einzelbilder; Schema .....	43
Abbildung 17: Vereinigung der Zusammengeführten Einzelbilder zu einem Gesamtbild .....	44
Abbildung 18: Versuchsanordnung.....	45
Abbildung 19: Baugruppe, Unteransicht mit markierten Komponenten .....	46
Abbildung 20: Baugruppe, Seitenansicht mit montierter MQ II und Bildsensoren.....	47

Abbildung 21: Dauer der Rekonstruktion einer Szene; Zeitangaben in ms .....	48
Abbildung 22: Ergebnisse in 3D-Darstellung .....	49
Abbildung 23: Datenaufbereitung durch Schwellwertfilterung vor der Datenanalyse	51
Abbildung 24: Durch Rauschen verursachter Fehler .....	52
Abbildung 25: Kantenerkennung der Szene mit Sobel-Operator .....	54
Abbildung 26: [BAL06-14] Vereinigung von verschiedenen Bildinformationen .....	55
Abbildung 27: [SIL05-21] Gewinnung der Information durch konzentrische Anordnung der Kameras .....	56
Abbildung 28: FPGA-Cyclone II [WIK10-23] .....	57
Abbildung 29: Sensorcluster .....	58
Abbildung 30: Wellenlängenbereich des Sichtbaren Lichtes für den Menschen.....	60



## Verzeichnis der Tabellen

Tabelle 1:Einteilung Funktionsgruppen	23
Tabelle 2: Auswahl Objektabstand	32
Tabelle 3: Berechnung für die Brennweitenauswahl (Auszug)	33

## Verzeichnis der Anhänge

Anhang A: Ablaufplan 3dscan_s- Severstart	62
Anhang B: Ablaufplan 3dscan_s- Livemodus	63
Anhang C: Ablaufplan 3dscan_s- Automatische Rekonstruktion	64
Anhang D: Ablaufplan 3dscan_s- Automatische Kalibrierung	65
Anhang E: Kalibrierung- Schema	66
Anhang F: Kalibrierung- Schablone	67
Anhang G: Vollständige Optikberechnungen	68
Anhang H: Rückenplatte	69
Anhang I: Kamerastativ KAISER RS 1	70
Anhang J: Leistungsmerkmale MQ II	71
Anhang K: Schaltpläne	72

## Liste der Symbole

Sofern lokal nicht anders beschrieben, haben folgende Symbole für das gesamte Dokument Gültigkeit und folgende Bedeutungen.

a	Anzahl der Sichtbereiche für eine Szene; Horizontal
A	Bildpunktematrix; Bilddaten in 2D Darstellung
b	Anzahl der Sichtbereiche für eine Szene; Vertikal
B	Objekthöhe/ -breite
h	Höhe eines Punktes
L	Vektor des einfallenden Lichtstrahls auf die Szene
n	Anzahl der Kameras eines Clusters; Horizontal
m	Anzahl der Kameras eines Clusters; Vertikal
x, y, z	Dimensionen in kartesischen Koordinaten
$\Delta x$	Länge eines Schattenwurfes
$\Delta h$	Höhendifferenz zwischen dem Start und dem Ende des Schattens
Z	Wegauflösung pro Pixel
Y	Winkeländerung pro Pixel

## Indizes

0	Anfangspunkt
1	Endpunkt
hor	Horizontal
vert	Vertikal
i	Zählvariable; für Kameraindex

## Griechische Symbole

$\alpha$	Winkel zwischen der X-Achsennormale und dem Vektor L
$\Omega$	Bahn der Funktion $h(x)$
$\Theta$	Polarwinkel
$\varphi$	Azimuthwinkel
$\lambda$	Wellenlänge
$\Delta\lambda$	Wellenlängenspektrum der eingesetzten LEDs

## Liste der Abkürzungen

Erläuterungen zu in dieser Arbeit verwendeten Abkürzungen.

API	Schnittstelle zur Anwendungsprogrammierung (engl. Application Programming Interface)
SMS	Software-Modul-Schnittstelle (Hardware API)
FPGA	Field Programmable Gate Array
TCP/ IP	Transmission Control Protocol/ Internet Protocol
NFS	Network-File-System
FH-Ge	Fachhochschule Gelsenkirchen
BuS	Bauelemente und Schaltungstechnik
IR	Infrarot
CCD	Charge Coupled Device (Lichtempfindliches elektronisches Bauteil)
MQ II	MiniQuadrix II
PC	Personal Computer
CAD	Computer Aided Design
ROI	Region of Interest (Bildregion, die ausgewertet wird)
QE	Quantum Effizienz (quantum efficiency) des CCD-Chips
PCB	Printed Circuit Board
VHDL	Very (High Speed Integrated Circuits) Hardware Description Language
LVDS	Low Voltage Differential Signaling
i.d.R.	In der Regel
VGA	Video Graphics Array (Grafikkartenstandard)

## 1 Einleitung

### 1.1 Motivation

Bildgebende Verfahren stellen in der modernen Medizin ein wesentliches Werkzeug zur Diagnostik und Kenndatenaufzeichnung dar. Während bildgebende Systeme wie Magnet-Resonanz-Tomografen oder Ultraschall-Geräte zur Diagnostik dienen, werden 2D-Scanner oder 3D-Scanner verwendet, um Kenndaten wie geometrische Dimensionen, Position und Lage von Patienten oder Instrumenten zu ermitteln. Es wird heutzutage immer wichtiger diese Informationen automatisch und in möglichst kurzer Zeit zu gewinnen. Die automatische, berührungslose Erfassung von dreidimensionalen Patientendaten hat ein breites Anwendungsfeld. Es ist vorstellbar, dass diese Informationen (besonders Lage von Instrumenten und Patient) wichtige Orientierungshilfen für den Arzt während einer Operation darstellen könnten. Ein weiterer Anwendungsfall für den Sensor könnte sein, bei der Erstellung eines Zahnprofils zu assistieren und dem Dentaltechniker unmittelbar geeignete CAD-Daten zur Verfügung zu stellen. So könnte schnellstmöglich eine Zahnprothese angefertigt werden.

### 1.4 Zielsetzung

Mit dieser Bachelorarbeit 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, 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.

Die Ergebnisse der Arbeit sollen geeignet sein, einen beliebig großen Cluster  $n \times m$  ( $n, m=1 \dots 216$ ) zu betreiben.

Das bedeutet, es sind Nachbarschaftsverhältnisse der Sensoren untereinander zu berücksichtigen.

## 1.5 Aufbau der Arbeit

Dieses Dokument behandelt zunächst die zur Bearbeitung des Themas notwendigen, theoretischen Grundlagen. Anschließend folgt die Beschreibung der Umsetzung. Sofern es Abweichungen zu den im Pflichtenheft [GNI10-7] vereinbarten Vorgaben gibt, werden in diesem Teil die Gründe dafür erörtert. Die Präsentation der Ergebnisse wird im anschließenden Teil durchgeführt. Durch eine Diskussion der Ergebnisse, einer kritischer Stellungnahme und Vorschläge zur Verbesserung der Lösungswege wird die Arbeit abgeschlossen.

## 2 Zielbestimmungen

Die nachfolgenden Formulierungen der Zielbestimmungen beschreiben ausführlich die zu bearbeitende Aufgabenstellung, die Musskriterien sowie die geltenden Rahmenbedingungen [HOW10-11].

### 2.1 Musskriterien

Als Entwicklungsplattform für das FPGA ist die Entwicklungsumgebung Quartus II des Hersteller ALTERA zu verwenden. Die Verifikation des Algorithmus 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 sie konform mit den SMS der Entwicklungsumgebung MiniQuadrix II sind.

Zur Softwareentwicklung sind ausschließlich die bereits vorhandenen Hardware-treiber der Baugruppe zu verwenden. Sollten diese nicht genügen ist Kontakt zu Herrn Marco Brinker (FH-Ge) aufzunehmen.

### 2.2 Wunschkriterien

Zur Veranschaulichung der Rechenergebnisse ist eine 3D-Darstellung der Ergeb-

nisse in Falschfarben auf einem Display hilfreich. Sofern es der Zeitrahmen ermöglicht, wird eine entsprechende Software entwickelt, um die Daten über den VGA- Anschluss des MiniQuadrix-Moduls oder über eine Netzwerkverbindung auf einem PC darzustellen. Als Datenformat auf dem PC wäre zudem ein CAD-Format wünschenswert.

## **2.3 Abgrenzungskriterien**

Algorithmen und Testanordnungen, welche bereits in der Arbeitsgruppe BuS entwickelt wurden, sollen eingesetzt und nicht gänzlich neu erfunden werden. Es sollen alle bereits erbrachten Arbeitsergebnisse der Arbeitsgruppe BuS genutzt werden und in dieser Bachelorarbeit mit entsprechender Quellenangabe verwendet werden.

## **2.4 Anwendungsbereiche**

Die Baugruppe des Miniclusters ist so konzipiert, dass sie ausschließlich für den Laborbetrieb geeignet ist.

## **2.5 Zielgruppen**

Die Versuchsanordnung und Algorithmen werden für Teammitglieder der Arbeitsgruppe BuS ausgeführt.

# **3 Shape from Shadows Algorithmus**

Es folgt eine Beschreibung des in dieser Bachelorarbeit angewendeten Verfahrens zur Gewinnung von 3D-Szeneninformation. Außerdem wird auf Implementierungsdetails und Besonderheiten eingegangen, bei der die Ausführung des Algorithmus auf der Hardware der MQ II eine besondere Rolle spielt.

### 3.1 Einleitung

Der in dieser Arbeit verwendete Algorithmus zur Gewinnung der 3D-Szeneninformation ist das Shape from Shadows-Verfahren (auch als Shape from Darkness bezeichnet). Es wurde bereits in mehreren Veröffentlichungen beschrieben und technisch umgesetzt, z.B. [YU & T. 04-13], [BAL06-14], [DAU98-15]. Diese haben gezeigt, dass es möglich ist, Höheninformation einer Szene allein aus der Kenntnis der Beleuchtungsverhältnisse bzw. durch die Auswertung von in der betrachteten Szene befindlichen Schatten zu erhalten.

Die Idee, aus dem Schatten auf die Höhe eines Objektes zu schließen, liegt nahezu auf der Hand. Zum einen folgt dieser Sachverhalt sehr einfachen mathematischen Regeln (vergl. (1) & (2)) und zum anderen sind die benötigten Parameter offensichtlich leicht zugänglich.

Zur Veranschaulichung stelle man sich eine Person vor, die an einem sonnigen Tag auf einer offenen Fläche steht und einen Schatten wirft. Vermisst man die Länge des Schattens und ermittelt den Winkel der Sonne zur Person, kann man relativ einfach auf die Größe der Person schließen. Bei einer einzelnen Person ist der Sachverhalt trivial nimmt jedoch schnell an Komplexität zu.

Dazu stellen wir uns vor, man möchte Informationen über die Höhe und geometrische Beschaffenheit eines Gebäudes (z.B. den Gebäuden der Fachhochschule Gelsenkirchen) ermitteln und hat Zugriff auf Luftaufnahmen dieses Gebäudes, bestenfalls zu unterschiedlichen Tageszeiten. So muss es, analog zu den Überlegungen die für die Person gelten, möglich sein, aus den Aufnahmen diese Daten zu ermitteln.

Nun gilt es die benötigten Parameter, nämlich Schattenlänge und Beleuchtungswinkel, zu bestimmen. Die Bestimmung der Höhe eines Punktes aus der Länge des Schattens wird im folgenden beschrieben, die Beleuchtungswinkel ergeben sich aus der Bauweise des MiniClusters bedingt und sind im Kapitel 4 beschrieben bzw. hergeleitet.

### 3.2 Informationen aus den Schatten

Verallgemeinern wir nun die in der Einleitung dargestellten Überlegungen.



Zunächst wird das Prinzip der Gewinnung der Höheninformation anhand einer 2D-Darstellung erläutert und anschließend auf 3D-Fälle übertragen. Weiterhin werden Möglichkeiten zur Darstellung der Daten präsentiert.

Die Gewinnung von detaillierten Höheninformationen über eine Szene ist nicht durch die Auswertung eines Bildes möglich. Der Grund dafür ist die Natur der Bilder oder besser der Szene. Diese zeigt sich hochgradig unterbestimmt, d.h. es gibt keine eindeutige Lösung für das betrachtete Objekt solange der Fall nicht trivial ist.

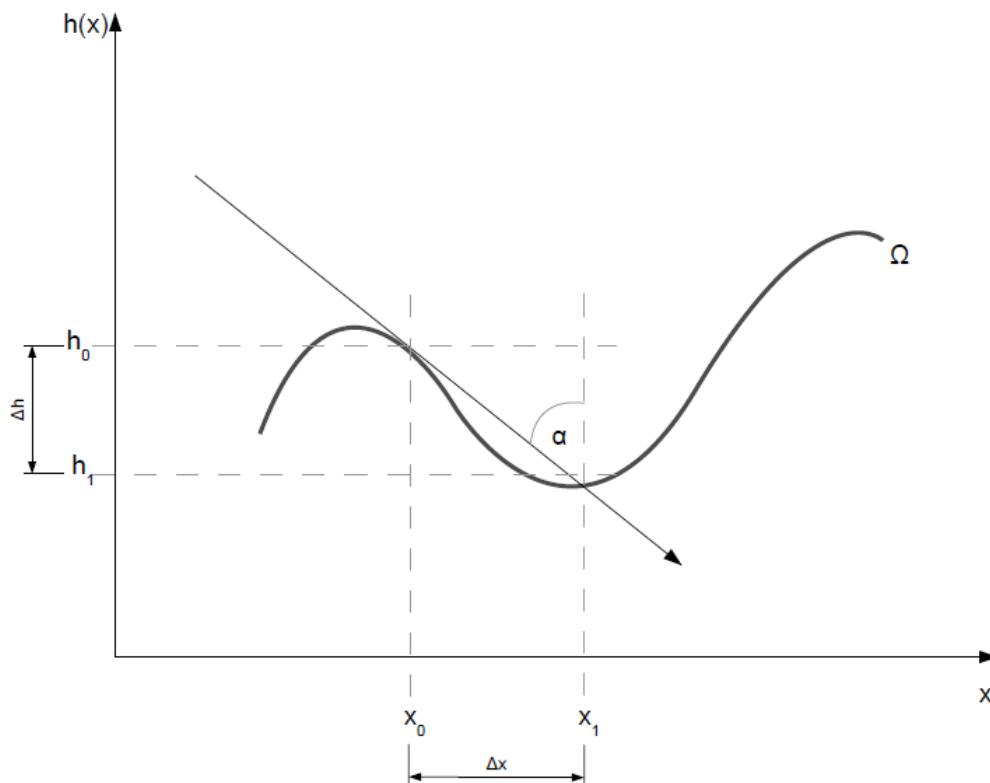


Abbildung 1: Shape from Shadows- 2D-Darstellung einer Struktur mit Lichteinfall

Aus Überlegungen über Schatten-Licht Beziehungen kann man folgende, einfache Zusammenhänge aus *Abbildung 1* ableiten:

$$\Delta h_{01} = h_0 - h_1 = \Delta x_{01} / \tan(\alpha) = (x_0 - x_1) / \tan(\alpha) \quad (1)$$

$$h(x) \leq h(x_0) - (x_0 - x_1) / \tan(\alpha), \quad \forall x \in [x_0, x_1] \quad (2)$$

(1) und (2) werden von nun an Schattenbedingungen genannt. Aus Gleichung (2) folgt außerdem, dass ein Bildpunkt, der nach einem Hell-Dunkel-Übergang tiefer oder auf gleicher Höhe liegen muss. Mit diesen Überlegungen kann eine Gleichung für die aus der Schattenlänge resultierende Höhe eines Bildpunktes, relativ zu seiner Umgebung, abgeleitet werden:

$$h(x_1) = h(x_0) - (x_0 - x_1) / \tan(\alpha) \quad (3)$$

Gleichung (3) wird in der späteren Implementierung benutzt, um das Bild zu rekonstruieren. So lässt sich von jedem ermittelten Schatten seine eigene Höhe bestimmen.

Nun betrachten wir den Fall einer Lichtquelle, die sich über die Szene bewegt. Die folgenden Ausführungen gelten unter der Voraussetzung, dass es sich bei der betrachteten Szene um eine geschlossene, geländeähnliche Oberfläche handelt.

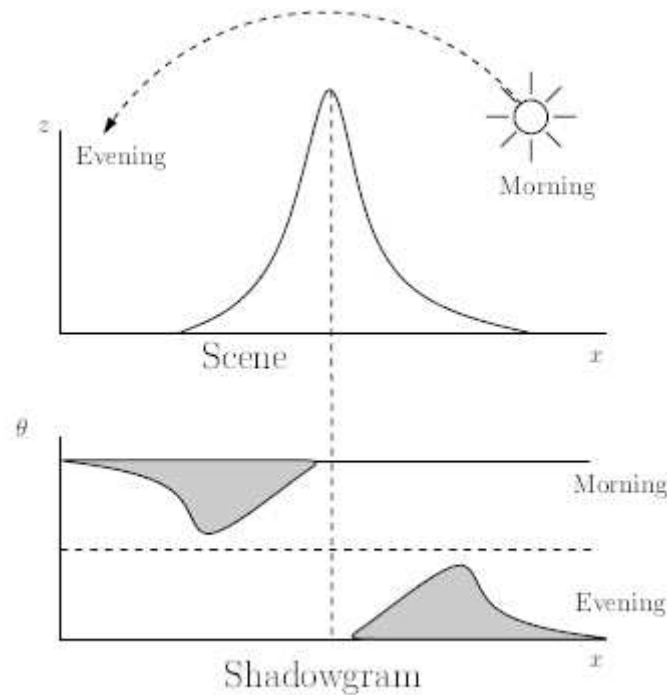


Abbildung 2: [DAU98-15] 2D Shadowgram

Abbildung 2 beschreibt den Zusammenhang zwischen einer sich im zweidimensionalen Raum bewegend Lichtquelle und dem Schatten der in der Szene entsteht. Für jedem Winkel  $\theta$  wird ein Bild aufgenommen und die jeweiligen Schattenverhältnisse ausgewertet. Werden die Daten aus den Einzelbildern zusammengeführt ergeben sie das Shadowgram. Es enthält damit die Informationen über die Illumination aufgetragen über Beleuchtungswinkel ( $\theta$ ) und die Bildpunkte ( $x$ ). [DAU98-15] beschreiben dieses Shadowgram als binäre Funktion. Es ist damit die Grundlage der Rekonstruktion der Szene, da es alle benötigten Informationen aus den Bildern enthält.

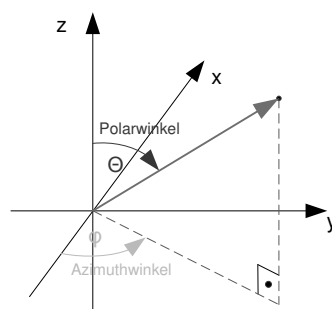


Abbildung 3: Kugelkoordinaten

Das 2D-Shadowgram lässt sich in einen dreidimensionalen Fall durch die Einführung eines zusätzlichen Freiheitsgrades der Lichtquelle überführen. Die Lichtquelle bewegt sich nun auf der von Winkel  $\theta$  und Winkel  $\varphi$  beschriebenen Halbkugel entlang einer Bahn (Trajektorie). Die betrachtete Szene befindet sich in der  $xy$ -Ebene.

Genau wie in dem Fall der 2D-Szene nehmen wir nun für jede Beleuchtungsrichtung ein Bild der Szene auf und erstellen eine Shadowgram aus den Daten. Dabei ergeben sich die in *Abbildung 4* dargestellten und von [DAU98-15] beschriebenen Graphen.

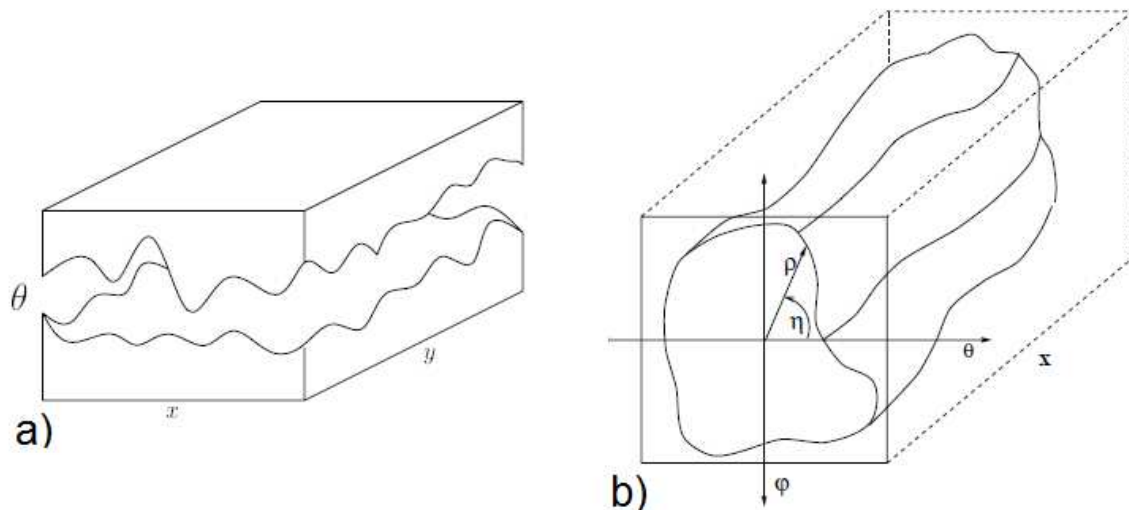


Abbildung 4: [DAU98-15] a) Sandwich Cross Section b) Tunnel Cross Section

Diese Darstellungsform ist die Visualisierung aller aus den Schatten gewonnener Szeneninformationen. Die Szene ist aus diesen Daten vollständig rekonstruierbar. Dabei ist diese Form der Daten rein theoretischer Natur und lediglich mit unendlich vielen Bildern und Beleuchtungswinkeln vollständig zu gewinnen. Das ist in der Praxis nicht praktisch umsetzbar.

Der von hier an weiter verfolgte Ansatz ist eine Diskretisierung der Lichtquellen-Trajektorie. Damit wird versucht auch mit wenigen Rohdaten in Form von Bildern auszukommen und dennoch eine vollständige Rekonstruktion der Szene zu ermöglichen. Kapitel 4 beschreibt die technische Umsetzung der Diskretisierung.

Es ist möglich, in Echtzeit die Akquirierung der 3D-Daten durchzuführen. Dazu soll der Algorithmus in der Hardware ausgeführt werden.

Der Algorithmus wird mit Blick auf ein größeres Sensorcluster implementiert, in dem mehrere Kameras die Szene beobachten. Das bedeutet, dass Nachbarschaftsverhältnisse berücksichtigt werden müssen, da es zu Überlagerungen der Blickfelder kommt. Aus den Überlagerungen dieser Sichtfelder wird es möglich sein, auch mit weniger Daten auszukommen. So kann die Rekonstruktion mit einer höheren Sicherheit durchgeführt werden, wenn es sich um einen Teil handelt, der von mehreren Sensoren gesehen wird.

### **3.2 Präsentation der Daten**

Das Ziel, die Präsentation der Informationen über Höhe und geometrische Beschaffenheit, lässt sich allgemein als Höhenfeld formulieren. Dies ist eine zweidimensionale, skalare Darstellung eines Höhenprofils. Dabei wird die Information über die Höhe, also der 3. Dimension, in dem Wert eines jeden Eintrags in diesem Feldes kodiert. [WIK10-3] *Abbildung 5* zeigt in a) eine entsprechende 2D Darstellung des Oberflächenreliefs und in b) eine gerenderte Ansicht dieses Höhenfeldes. Die gerenderte Ansicht veranschaulicht, dass es einfach ist, ein Höhenfeld zu visualisieren und weiterzuverarbeiten. Darüberhinaus ist es einfach auf der Basis eines 2D-Höhenfeldes Daten für ein CAD-Programm zu generieren. Viele CAD-Programme verfügen sogar über geeignete Import-Funktionen.

Da auch die vom Sensor gelieferten Bilddaten sich auf diese Formulierung reduzieren lassen, ist die eben beschriebene Speichermethode die Grundlage für die Entwicklung des Algorithmus. Auch werden in dieser Art die Ergebnisse der Rekonstruktion gespeichert.

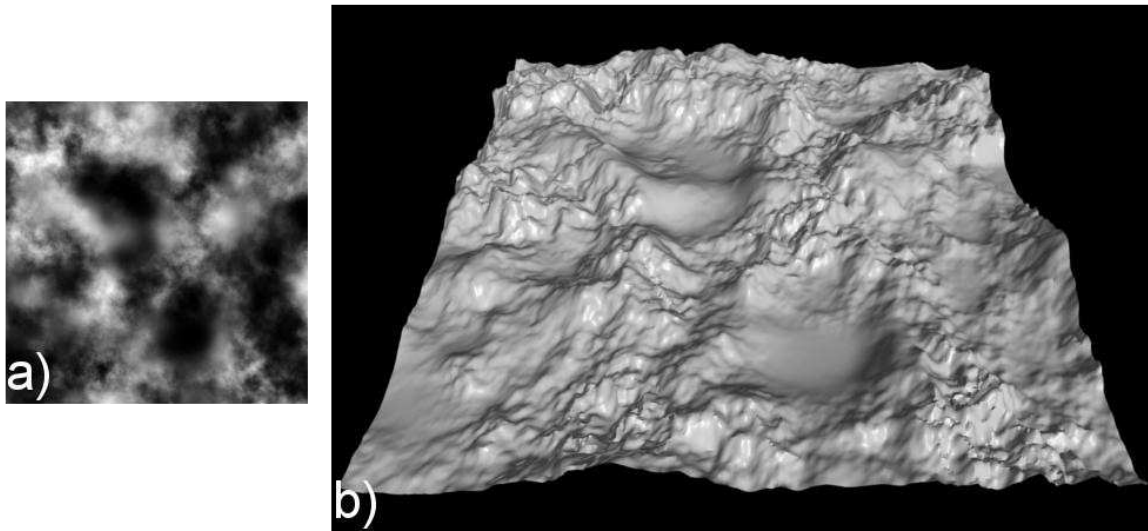


Abbildung 5: a) Höhenfeld 2D Darstellung b) gerenderte Ansicht des Höhenfeldes

### 3.4 Abgrenzung zu anderen Umsetzungen

Viele bisherigen Umsetzungen [YU & T. 04-13], [DAU98-15] stellen ein iteratives Verfahren vor (genauer eine Rekursion zur Minimierung des Fehlers), um die Informationen aus dem Shadowgram eines Bildpunktes zu entwickeln. Diese setzen die Speicherung einer Vielzahl von Bildern voraus, die anschließend parallel verarbeitet werden. Eine solche Vorgehensweise kann aufgrund der Umsetzung in der Hardware der MiniQuadrix II nicht in Betracht gezogen werden. Wie später beschrieben, wird die Bearbeitung eines Bildes unmittelbar nach der Aufnahme begonnen. Informationen werden also parallel zur Aufnahme für jedes Einzelbild gemacht und in einem abschließenden Arbeitsschritt zusammengeführt.

### 3.4 Eignung für eine hardwaregestützte Bildverarbeitung

Zu Beginn der Arbeit galt es herauszufinden, ob es überhaupt möglich ist, diesen Ansatz in Hardware auszuführen und wenn ja, wie dieser umzusetzen ist. Es wurde eine Methode entwickelt, mit der man die Akquirierung der Bilder mit der Szenenrekonstruktion parallel ablaufen lassen kann und somit die Stärke der Logikbaugruppe MQ II nutzt.

Eine einfache Portierung der in anderen Publikationen beschriebenen Vorgehensweisen ist nicht möglich, da diese zunächst, wie bereits erwähnt, eine vollständige Bildserie als Datengrundlage voraussetzen. Weiterhin ist das beschriebene Verfahren (z.B. das von [DAU98-15]) iterativ, was die Umsetzung zusätzlich erschwert. Diese Punkte zusammengekommen erlauben die direkte Umsetzung dieser Algorithmen auf Hardware nicht.

### 3.5 Anpassungen des Algorithmus an die Hardware

Ausgehend von dieser Überlegung, werden nun die Aufgaben des Algorithmus in Funktionsgruppen aufgeteilt und anschließend auf Umsetzbarkeit auf der Hardware geprüft und Teilen der Hardware zugewiesen. Die Funktionsgruppen sind:

Nr.	Beschreibung
1	Gewinnung der Bilddaten
2	Informationsverdichtung/ Filterung der Bilder
3	Informationsgewinnung aus Schatten des Einzelbilds
4	Bilddaten zusammenführen
5	Ergebnisdarstellung

Tabelle 1:Einteilung Funktionsgruppen

Ob eine Funktionsgruppe in der Hardware ausgeführt werden kann, muss einzeln geprüft werden. Es zeigt sich, dass die Unterteilung der Funktionsgruppen das vorher beschriebene Problem der iterativen Informationsgewinnung löst. So ist lediglich die Zusammenführung der Bilddaten (Funktionsgruppe 4) ein von der Hardware nicht zu bewältigender Schritt. *Abbildung 6* stellt die Bearbeitungsorte mit den in *Tabelle 1*

eingeteilten Funktionsgruppen schematisch dar.

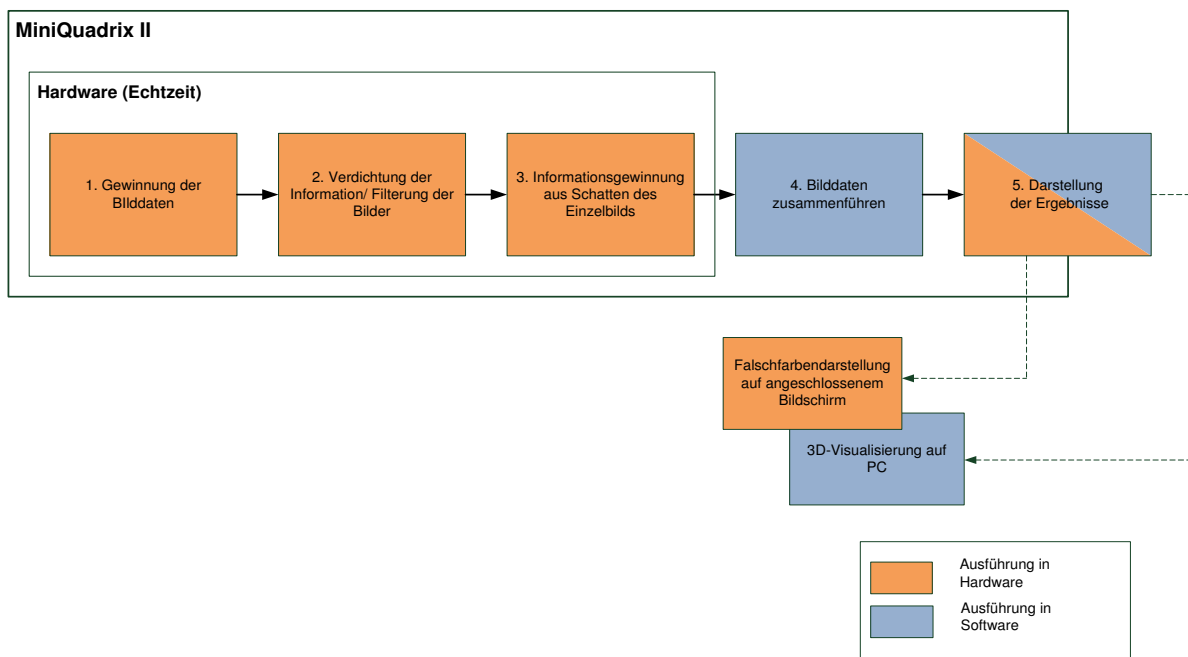


Abbildung 6: Bearbeitungsorte der Funktionsgruppen; Schema

Die vorgestellte Aufteilung gilt es nun in Hardware umzusetzen. Es wird dabei so vorgegangen, dass die Funktionsgruppe 3 zunächst in einer Softwarelösung erstellt und verifiziert wird und anschließend auf die MQ II portiert. Kapitel 5.5 beschreibt dabei welche Möglichkeiten prinzipiell dafür zur Verfügung stehen und welche genutzt wurden.

## 4 Hardware

In diesem Kapitel wird ausführlich die in dieser Arbeit entwickelte und eingesetzte Hardware beschrieben. Dazu gehören im Einzelnen:

- die entwickelte Platine des MiniClusters
- die Logikbaugruppe MiniQuadrix II,
- eine Rückenplatte zur Aufnahme der MQ II und des MiniClusters und



- das Kamerastativ.

## 4.1 MiniQuadrix II

Die Plattform zur Bildanalyse, die in dieser Arbeit eingesetzt wurde, ist die Baugruppe MiniQuadrix II. Das Modul der MQ II ist eine Entwicklung der Arbeitsgruppe BuS an der FH-Ge. Es ist speziell für die schnelle Bildverarbeitung ausgelegt. Die Plattform MiniQuadrix II ist sehr vielseitig. Sie verfügt über eine Add-On-Platine, die unterschiedlich bestückt werden kann. In *Abbildung 7 a)* ist eine Add-On-Platine mit einem Netzwerkanschluss und einem VGA-Ausgang gezeigt. In dieser Arbeit wird, zusätzlich zu den in der Abbildung gezeigten Anschlüssen, die Add-On-Platine über einen zusätzlichen Port verfügen, über den sich der LED-Treiber ansteuern lässt (o. A.).

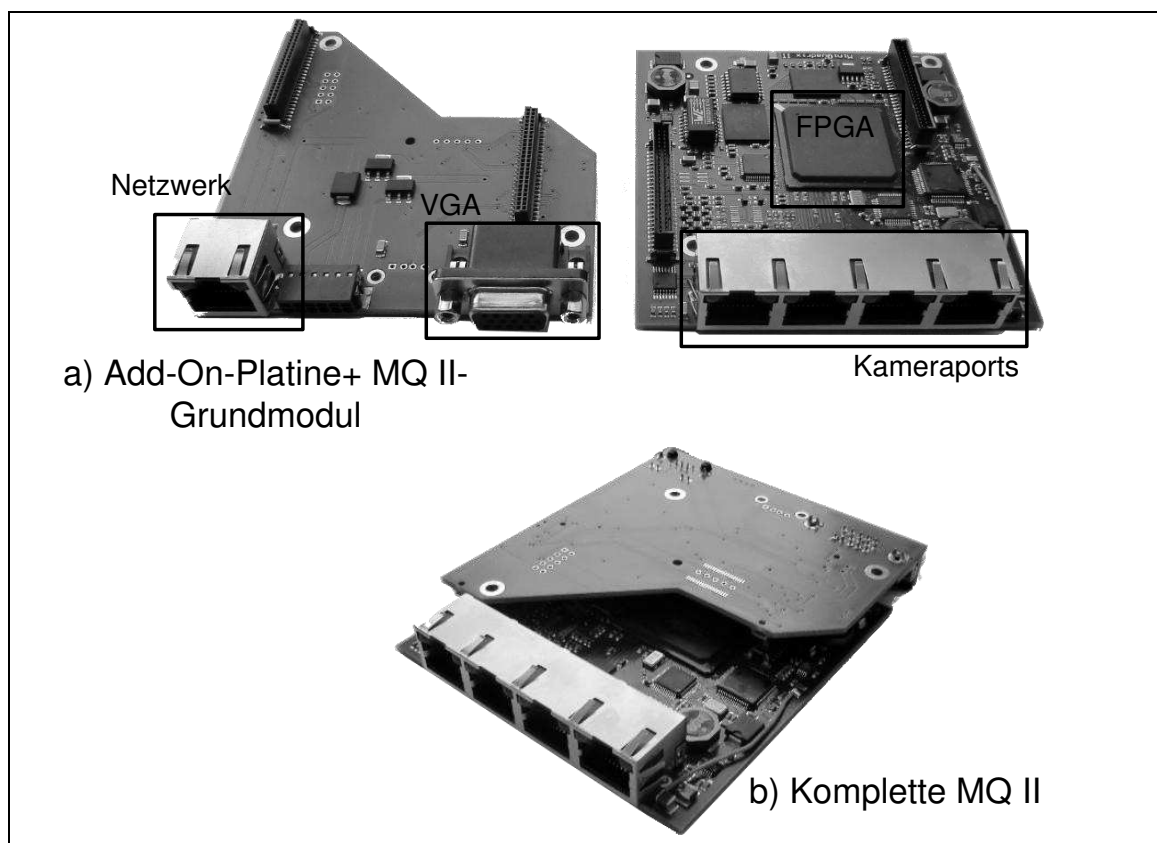


Abbildung 7: MiniQuadrix II

Eine Auflistung der Leistungsmerkmale ist im Anhang J zu finden.

Weiterhin konnte auf Erfahrungen mit der MQ II zurückgegriffen werden, die während der Tätigkeit in anderen Projekten in der Arbeitsgruppe BuS gesammelt werden konnten.

## 4.2 Kamerasensoren

Die eingesetzten Bildsensoren sind Module des Typs KK 752. Wie die MQ II sind auch sie Entwicklungen der Arbeitsgruppe BuS. Sie stehen in verschiedenen Ausführungen zur Verfügung, in dieser Arbeit werden die Sensoren ohne Einhausung mit Objektivhaltern eingesetzt. Abbildung 8 zeigt die Sensoren. Die Sensoren haben einen CCD-Chip [MIC-20] von Aptina (früher Micron) verbaut. Mit diesen Sensoren wurden bereits viele Projekte realisiert und daher stellen sie eine geeignete Grundlage für die Bildaufnahme in diesem Projekt dar.

Mit dem Sensor lassen sich Bilder mit einer maximalen Auflösung von 752H x 480V Pixeln (~0,36 Megapixel) aufnehmen.

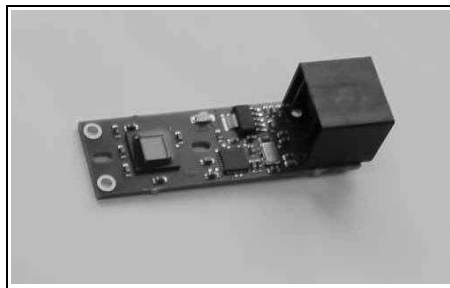


Abbildung 8: Bildsensor, Typ KK 752

Abbildung 9 zeigt, die QE der verwendeten Sensoren (MT9V024). Sie weisen eine im IR-Bereich gute Effizienz aus, verglichen z.B. mit Fotofilm der eine geringere Effizienz aufweist. Es wird ein geeignetes Schmalbandfilter eingebaut, das für das IR-Spektrum durchlässig ist. Und die Beleuchtung wird mit IR-Leuchtelementen realisiert.

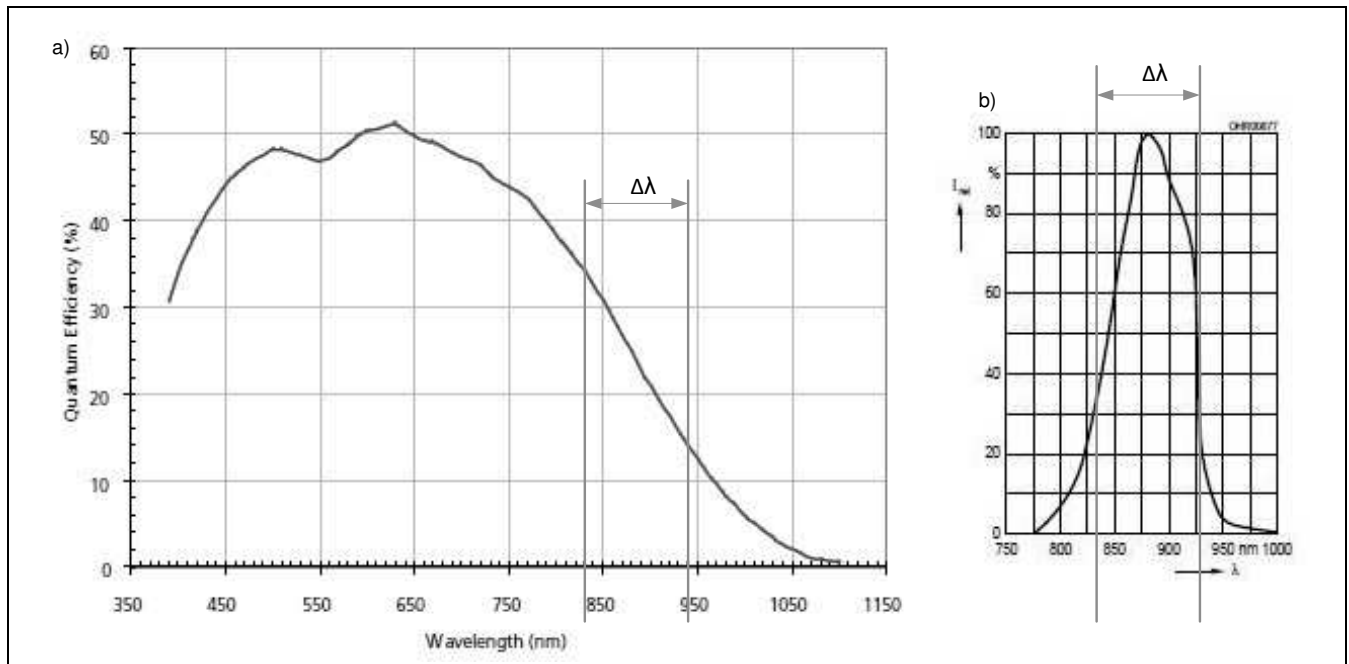


Abbildung 9: Typische Quantum Effizienz für IR

a) MT9V024 [MIC-20] b) Spektrum der LED [OSR09-18]

Die Montage des IR-Filters ist in *Abbildung 10* gezeigt.

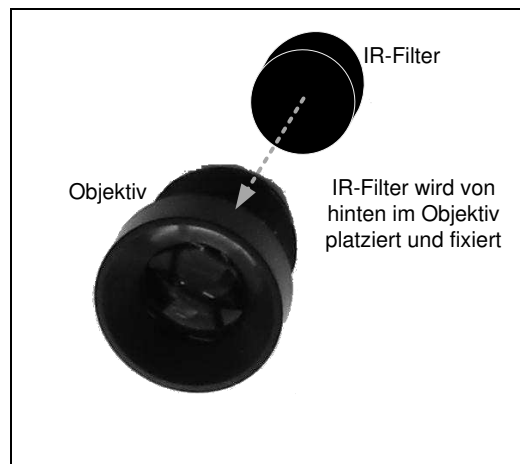


Abbildung 10: IR-Filter, Montage im Objektiv

Durch die Filterung kann ein gutes Ausgangsbild für weitere Verarbeitung gewährleistet werden, da es nicht sehr anfällig für Änderungen in der Umgebungsbeleuchtung ist. Vorausgesetzt, das Umgebungslicht hat einen nicht zu großen IR-Anteil (z.B. stark einstrahlendes Sonnenlicht).

### 4.3 MiniCluster

Für den in dieser Arbeit erstellten MiniCluster wurden vier Kameramodule des Typs: Kamerakopf KK 752 so angeordnet, dass sich ein 2 x 2 Cluster ergibt. Sie werden auf der Platinenrückseite montiert und in Abstrahlrichtung der LEDs ausgerichtet.

Am Cluster sind zusätzlich, gemäß Vereinbarung im Pflichtenheft [GNI10-7], Leuchtdioden angebracht. Wie im Kapitel 3 bereits beschrieben, ist es das Ziel der Beleuchtung einen gerichteten Schatten zu generieren, verschiedene Beleuchtungsrichtungen zu ermöglichen und damit die von der Rekonstruktion vorausgesetzte Trajektorie umzusetzen. Abweichend zu den Forderungen im Pflichtenheft, werden in dieser Arbeit keine Power-LEDs eingesetzt. Der Grund dafür ist, dass solche LEDs für den IR-Bereich nicht verfügbar sind oder ihre Anschaffung preislich den Rahmen dieser Arbeit überstiegen hätte.

Um dennoch der Forderung nach einem möglichst hellen und homogenen Lichtkegel zu entsprechen, fiel die Wahl auf LEDs des Typs SFH 4680 des Herstellers OSRAM [OSR09-18]. LEDs dieses Typs emittieren IR-Licht und haben sich bereits in mehreren anderen Projekten der Arbeitsgruppe BuS bewährt.

Aus dem Datenblatt [OSR09-18] kann man entnehmen, dass die LEDs sich kurzfristig im Hochleistungsbereich betreiben lassen. Dabei beträgt der maximale Stoßstrom 1 A, bei einer Pulsdauer von 300  $\mu$ s.

Desweiteren weisen die LED eine sehr gute Abstrahlcharakteristik im Bezug auf Abstrahlwinkel und Wellenlänge auf. Die Leuchtdioden werden pulsartig betrieben. Auch bei kurzem, pulsartigem Betrieb der LEDs wird eine hohe Verlustleistung umgesetzt, es entsteht entsprechend viel Wärme. Es ist also ein geeignetes Wärmemanagement nötig. Wie im Datenblatt vorgeschlagen, wurde die Padgröße daraufhin angepasst, um eine möglichst große Fläche zu realisieren. Das macht die Anbringung von externen Kühlkörpern unnötig.

Die Ansteuerung der LEDs erfolgt über einen geeigneten Treiberbaustein auf der Platine des MiniClusters. Der Treiber wird unmittelbar über einen entsprechenden Port der MQ II angesteuert. Diese Art der Ansteuerung ermöglicht eine größtmögliche Freiheit in Bezug auf die Beleuchtungsparameter. Es lassen sich der LED-Strom (damit die Intensität der LEDs) und die Pulsdauer einstellen. Das macht diesen

Treiber, verglichen mit anderen Ansteuerungsmöglichkeiten, überlegen.

Zusätzlich verkürzte sich die Entwicklungszeit, da eine entsprechende Lösung der Arbeitsgruppe BuS mit projektspezifischen Anpassungen schnell erstellt und umgesetzt werden konnte. Das Grunddesign der Schaltung wurde von Herrn Dipl.-Ing. (FH) Marco Brinker (FH-Ge) entwickelt. Die vorgenommenen Anpassungen ermöglichen den Betrieb von vier Reihen mit je 10 LEDs.

Das MiniCluster ist quadratisch und hat eine Kantenlänge von 16 cm. Der Abstand von einer LED-Reihe zur gegenüberliegenden beträgt 15 cm.

Der Abstand der Kameras beträgt 40 mm horizontal und 40 mm vertikal.

*Abbildung 19* zeigt die fertige Baugruppe.

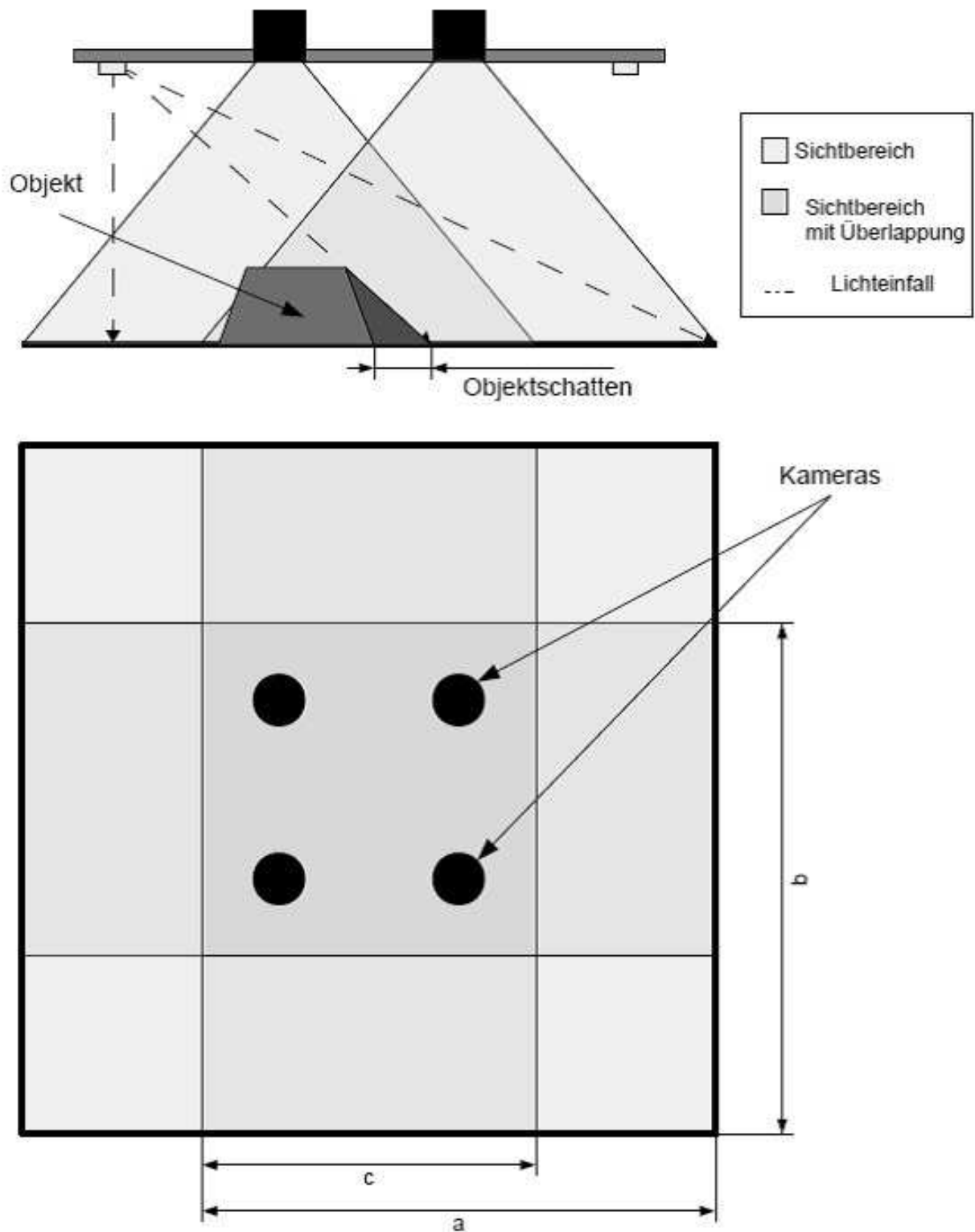


Abbildung 11: Kameraanordnung; Darstellung der Sichtbereiche mit Überlagerungen

Die Leiterkarte des MiniCluster wurde im Labor für IC-Design an der FH-Ge auf einer Fräsmaschine gefertigt. Dieser Schritt war möglich, da die Platine mit einem zweilagigen Layout (Top-Layer und Bottom-Layer) auskommt. Bestückt wurden die

Teile von Hand im BuS-Labor.

#### 4.4 Szenenbeleuchtung und Schattenerzeugung

Auf der Platine sind vier LED Anordnungen angebracht. Jede Anordnung enthält 10 LEDs. Die Leuchtdioden dienen der gezielten Schattengenerierung für die 3D-Rekonstruktion. Die Ansteuerung der Baugruppe erfolgt über einen eigenen Port der MQ II. Die Schaltung wurde daraufhin entwickelt, verschiedene Beleuchtungseinstellungen umsetzen zu können. Intensität und Pulsdauer der LEDs lassen sich von einer Software aus einstellen. Um ein gutes Rekonstruktionsergebnis zu erzielen, sollen mit der Anordnung verschiedene Trajektorien der Lichtquelle möglich sein. Damit lassen sich verschiedene Beleuchtungsrichtungen einstellen.

#### 4.5 Optik

Die Optik dient dazu, eine geeignete Datengrundlage zu erzeugen. Es werden einiger Forderungen an die Optik gestellt. Zunächst muss die Szene auf der gesamten Breite ausgeleuchtet werden. Nur dadurch kann man sicher Informationen aus den Schatten gewinnen. Daher muss die Entfernung von Szene zum MiniCluster so gewählt werden, dass der Abstrahlwinkel von  $\pm 20^\circ$  ausreicht, um die Szene vollständig auszuleuchten. Einfache trigonometrische Überlegungen ergeben dabei:

$$\tan(\alpha_{max}) = i_{max} / D \quad (4)$$

Aus (4) lässt sich  $i_{max}$  berechnen zu:

$$i_{max} = \tan(\alpha_{max}) \cdot D \quad (5)$$

<b>D [mm]</b>	200	250	250	...	400	450
<b>i<sub>max</sub> [mm]</b>	73	91	110	...	146	164

Tabelle 2: Auswahl Objektabstand

Nach *Tabelle 2* wurden 450 mm für den minimalen Objektabstand nach diesen Überlegungen festgelegt. In Abstimmung mit der ROI der Kameras wurde dann die benötigte Brennweite der Objektive bestimmt.

Desweiteren muss eine geeignete Brennweite ausgewählt werden. Die Brennweite lässt sich aus dem Zusammenhang zwischen Sensorgröße, Abstand zum Objekt und Objekthöhe ermitteln. Er lautet:

$$f = (b \cdot D) / B \quad (6)$$

, mit

b := Sensorbreite

B := Objekthöhe

D := Objektentfernung

Aus dem Datenblatt [MIC-20] ermittelte Größe des Chips ist: 2,88 mm • 4,512 mm. Aus der vollständigen Tabelle *Anhang G* kann man entnehmen, dass einige Objekthöhen ausprobiert wurden. Die Festlegung für den ausgeführten Messaufbau in dieser Arbeit ist: Brennweite 16 mm und ein Abstandsintervall zur Szene von [445, 450] mm.



Idx	Brennweite [mm]	Abstand [mm]	Objekthöhe [mm]	Sensorbreite
...	...	...	...	...
8	15,84	440	80	2,88
9	16,02	445	80	2,88
10	16,2	450	80	2,88
11	16,128	560	100	2,88
...	...	...	...	...

Tabelle 3: Berechnung für die Brennweitenauswahl (Auszug)

Aus der Auswahl des Objektivs ergibt sich für jede Kamera ein Sichtfenster von 80 mm x 80 mm.

Horizontale und Vertikale Auflösung  $Z$  einer Kamera lassen sich bestimmen zu:

$$Z = Z_{hor} = Z_{vert} \quad (6)$$

$$Z = B_{hor}/pxs_{hor} = B_{vert}/pxs_{vert} = 80 \text{ mm}/480 = \underline{\underline{0,166 \text{ mm}/pxs}} \quad (7)$$

Aus der Bauform des MiniClusters ergeben sich damit folgende, in Kapitel 4.6 dargestellte Sichtbereiche respektive Nachbarschaftsverhältnisse.



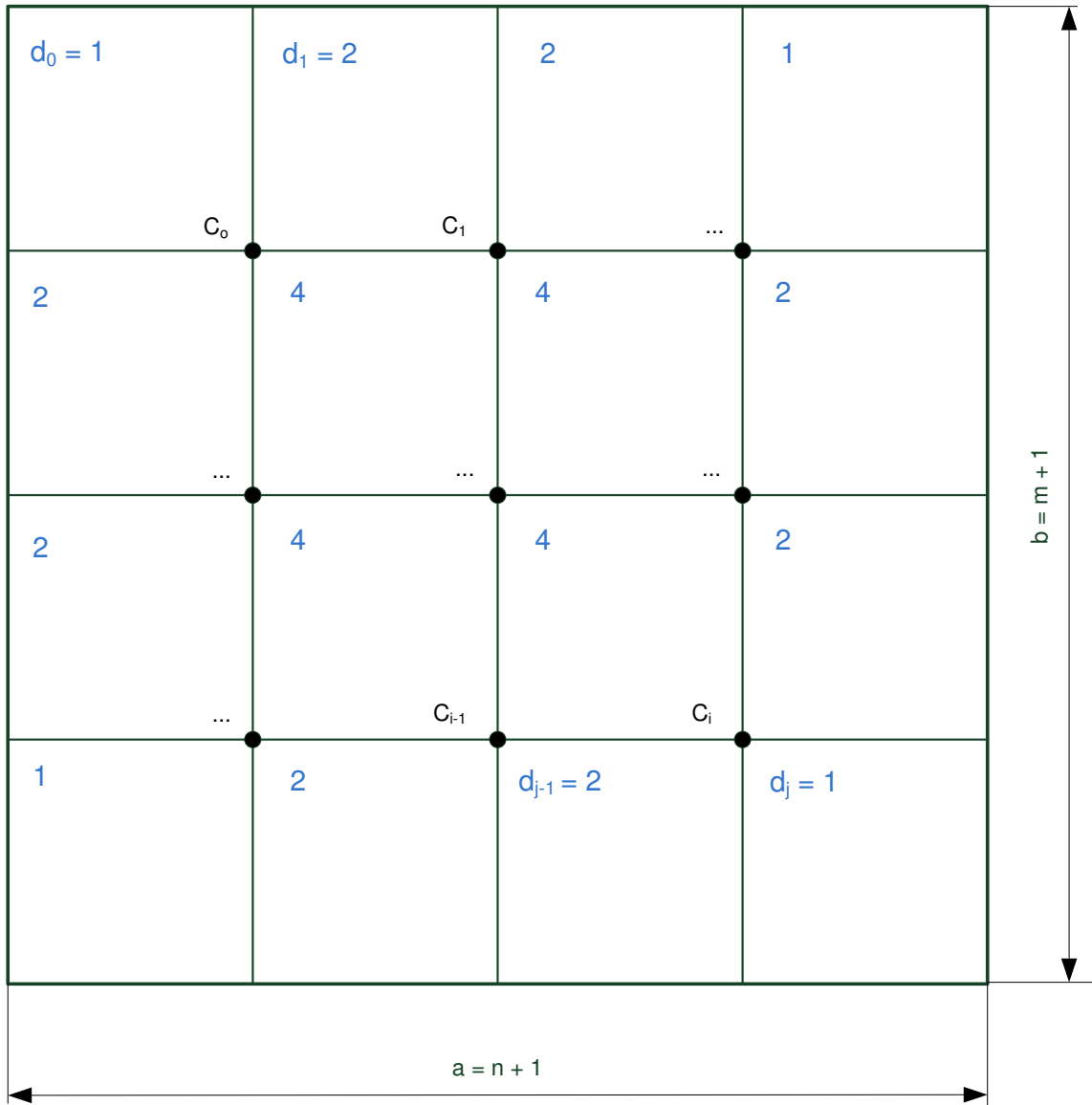


Abbildung 13: Allgemeiner Fall für einen quadratischen Sichtbereich

Abbildung 13 zeigt die Verhältnisse bei einem größeren Cluster. Es ist gelten dabei dieselben geometrischen Voraussetzungen, wie bei dem MiniCluster. Es treten auch hier wieder drei unterschiedliche Bereiche auf.

Die Anzahl der einzelnen Bereiche ergibt auch aus der Beziehung:

$$(a \cdot b) = (n + 1) \cdot (m + 1) = B_{ges} \quad (8)$$

Es ergeben sich folgende Zusammenhänge:

$$\text{Anzahl der Bereiche die 4 Kameras sehen} = (a - 1) \cdot (b - 1) \quad (9)$$

$$\text{Anzahl der Bereiche die 2 Kameras sehen} = 2(a - 1) + 2(b - 1) \quad (10)$$

$$\text{Anzahl der Bereiche die 1 Kamera sieht} = 4 \quad (11)$$

Für rechteckige Sichtbereiche der Kameras kann es dazu kommen, dass Bereiche entstehen, die von mehr als vier Kameras gesehen werden. Für diese Fälle gelten ähnliche Überlegungen und sind hier nicht weiter erwähnt.

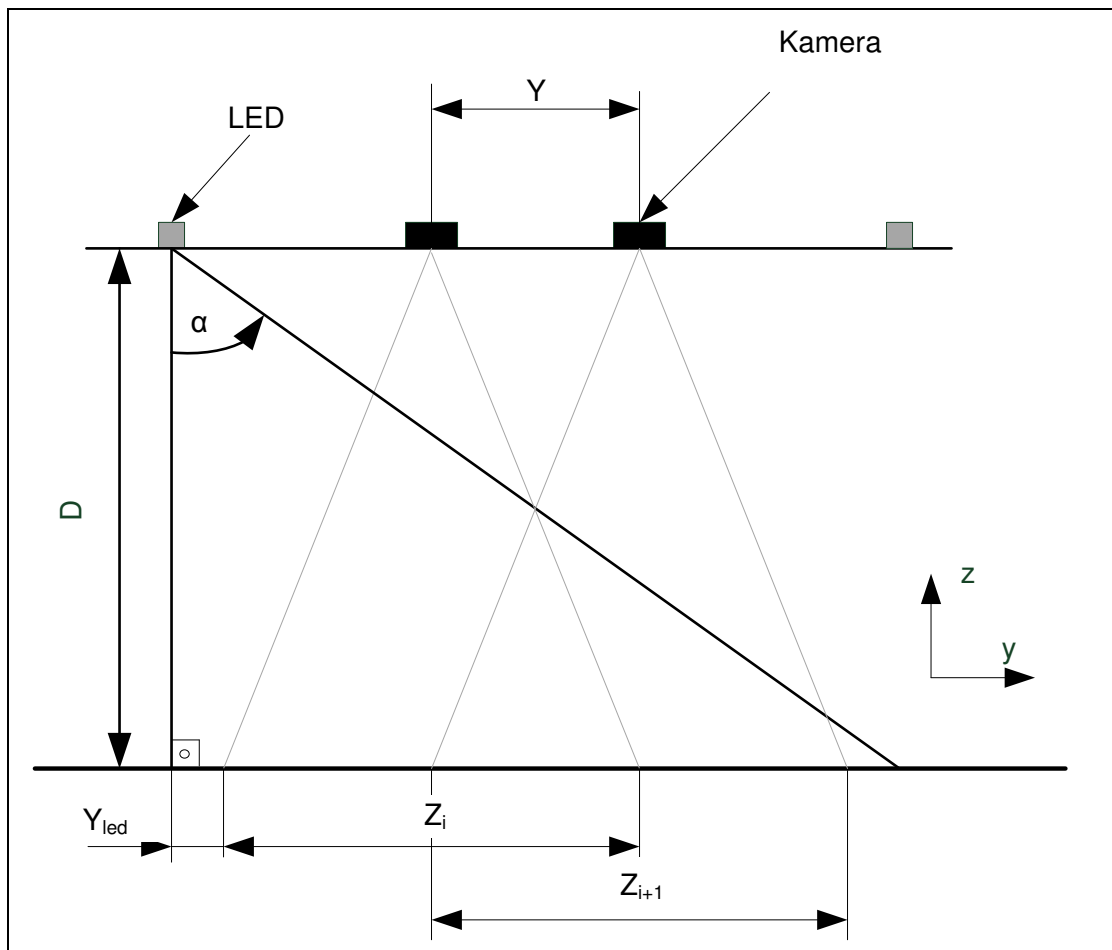


Abbildung 14: Änderung des Beleuchtungswinkels

Wie in Gleichung (2) beschrieben, lässt sich die Höheninformation aus der Schattenlänge mithilfe des Winkels  $\alpha$  berechnen. Da die Lichtquelle aufgrund des geringen Objektabstandes  $D$  nicht als im unendlichen befindlich angenommen werden kann, muss die Änderung des Winkels über den Sichtbereich einer Kamera berechnet werden. Dabei wird die Winkelauflösung  $Y$  pro Pixel bezogen, damit fällt die Auswertung leichter.

Die Winkeländerung  $Y$  beträgt: 
$$\underline{\underline{Y = 0,02 \text{ } ^\circ / \text{Pixel}}}$$

*Abbildung 14* zeigt dabei den Zusammenhang zwischen in der zy-Ebene. Da das MiniCluster quadratisch ist, sind auch diese Beziehungen für andere Ebenen identisch.

## 4.7 Montageplatte

Um die Baugruppe an das Kamerastativ RS 1 montieren zu können, wurde eine simple Montageplatte erdacht. Sie ist mit einfachen Funktionen ausgestattet und soll ein Modul der MQ II und das MiniCluster aufnehmen können. Zusätzlich soll es Zugentlastungen für entsprechende Kabelführungen bieten.

Datenblatt des Kamerastativs und technische Zeichnung der Montageplatte befinden sich im *Anhang H* bzw. *Anhang I*

## 4.8 Kalibrierung

Es ist nicht möglich, die Sichtbereich mechanisch aufeinander abzugleichen. Die Sichtbereiche werden grob von Hand aufeinander ausgerichtet und eine Softwarelösung (3dscan\_s) ermittelt im Anschluss automatisch die Kalibrierung. Dabei wird die Lage des x-Offset sowie des y-Offset für eine Kamera ermittelt. Im *Anhang E* ist die Kalibrierung näher beschrieben. Zur Kalibrierung wurde während dieser Arbeit eine entsprechende Schablone erstellt, sie ist im *Anhang F* zu finden.

## 4.9 Objekte

Für die ersten Entwicklungsschritte wurde eine Reihe von Bilddaten aufgenommen. Die abgebildeten Objekte sind Anordnungen aus LEGO-Steinen.

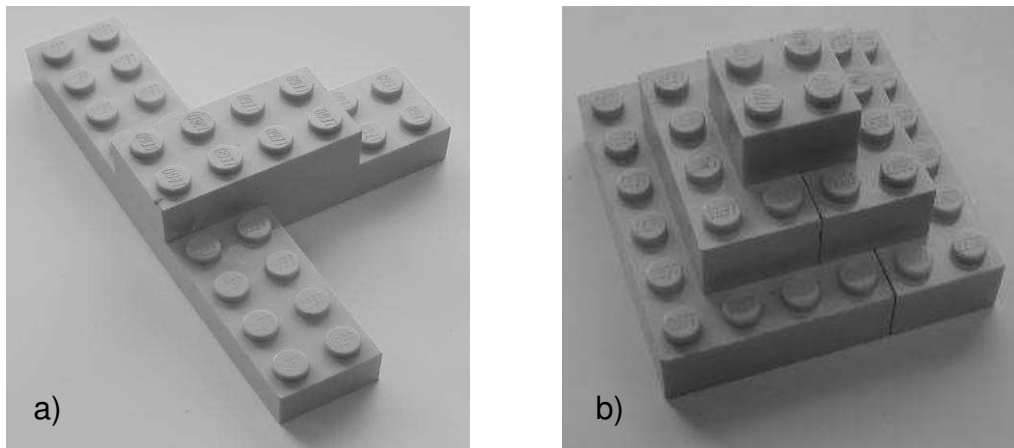


Abbildung 15: Originalszenen der in dieser Arbeit präsentierten Bildserien 5 und 7 a) sowie 16 b)

Obwohl der vornämliche Anwendungsbereich für diese Elemente Spielwaren sind, eignen sie sich für die Verwendung in dieser Arbeit sehr gut.

Sie ermöglichen eine große Flexibilität, da es leicht fällt die neu anzuordnen. Obwohl es sich um Spielwaren handelt werden diese Blöcke mit einer höhen Toleranz von wenigen Mikrometern gefertigt.

## 5 Software

Es folgt eine Beschreibung der Software, die im Laufe des Projektes erstellt wurde. Eine ausführliche Dokumentation der Software liegt separat vor und ist auf dem Datenträger digital zu finden.

## 5.1 Umgebung

Auf der MQ II steht als Betriebssystem  $\mu$ Linux zur Verfügung. Als Prozessor wurde im FPGA-Design der NIOS II von Altera umgesetzt. Die Softwareprojekte wurden mit dem entsprechenden von Altera bereitgestellten Compiler nios2-gcc übersetzt. Alle Programme für die MQ II sind in der Programmiersprache C geschrieben und konform zum C99 Standard. Die Softwareumsetzung des Algorithmus ist in C++ geschrieben und bedient sich neben den Standardbibliotheken der OpenCV-Library der Firma Intel in der Version 2.0. Die Dokumentation aller Software wird mithilfe von Doxygen automatisch erstellt, dazu sind die Kommentare im Quellcode entsprechend umgesetzt.

## 5.2 Software der MQ II- 3dscan\_s

Kern der Rekonstruktion ist das während dieser Arbeit entwickelte Werkzeug 3dscan\_s. Grundfunktionen dieses Tools sind es, die Hardware zu parametrisieren, die Rekonstruktion zu starten, die automatischen Kalibrierungsparameter zu bestimmen und die Ergebnisse auf einen PC zu übertragen.

Es benutzt die auf der MQ II zur Verfügung stehende API [BRI10-24]

Das Tool 3dscan\_s ist in einer weiteren ausführlicheren Art Dokumentiert [GNI10-25].

### 5.2.1 Übertragung zum PC

Alle Daten, die während der Rekonstruktion anfallen sowie das Ergebnis, werden über der NFS-Protokoll vom 3dscan\_s-Tool auf einen Clientrechner übertragen. Dazu wird der Ordner in dem System der MQ II eingebunden und der Ordnerpfad beim Programmaufruf übergeben.

## 5.3 Clientprogramm

Ein Programm, welches Daten auf einem PC empfängt, konnte mangels Zeit nicht programmiert werden. Auf einem Linux Betriebssystem können die Daten unmittelbar über ein in dem System der MQ II eingebundenen NFS-Ordner übertragen werden.

## 5.4 Algorithmus in Software

Zunächst wurde der Algorithmus in einer Softwarelösung erstellt. Die Software entspricht benutzt dabei der vorgenommenen Aufteilung in die Funktionsgruppen. So ist eine einfache Auslagerung einzelner Blöcke in Hardware möglich.

## 5.5 Portierung des Algorithmus auf die Hardware

Um den in Software umgesetzten Algorithmus auf die Hardware zu portieren, gibt es verschieden Möglichkeiten. Einige davon sind:

- Beschreibung eines Hardwareblockes mit VHDL
- Beschreibung eines Hardwareblocks in Verilog
- Erstellung eines Hardwareblockes im Quartus-Designer
- Erstellung einer Custom-Instruction mit einem C++-Dialekt (SystemC)

Die genannten Punkte sind im Wesentlichen, die Methoden, die in der Arbeitsgruppe BuS zur Umsetzung von Algorithmen benutzt werden. Die Entwicklung eines solchen Blocks ist sehr aufwändig, nach jeder Änderung muss das vollständige Design Compiliert werden. Die Dauer dieses Vorgangs beträgt i.d.R. mehrere Stunden (bei der MQ II ca. 3 Stunden).

Da der Algorithmus in den ersten Entwicklungsschritten in der Programmiersprache C/ C++ Verfasst wird, liegt die Umsetzung des Algorithmus als eine Custom-Instruction nahe. Eine Custom-Instruction ist eine Anweisung, die im System der MQ II eingebunden werden kann und in Hardware ausgeführt wird. Dabei wird die Programmiersprache SystemC verwendet um die Hardware zu beschreiben. Ein von Altera gelieferter Compiler übersetzt die Instruction für den NIOS II-Prozessor.

Bei SystemC handelt es sich um C++ (Klassen-) Bibliotheken und Makros, die eine hardwarenahe Entwicklung und Simulation möglich machen. Im Gegensatz zu VHDL oder Verilog läuft die Entwicklung dabei auf einer höheren Abstraktionsebene ab. Damit ist eine schnellere Entwicklung als mit konventionellen Hardwarebeschreibungssprachen möglich.



## 6 Auswertung der Bilddaten

Die Auswertung der Schatten in der Szene ist in folgende Punkte unterteilt:

1. Suche nach einem Schatten (Hell-Dunkel-Wechsel)
2. Ermittlung der Schattenlänge
3. Berechnung der Höhe des zum Schatten korrespondierenden Punktes
4. Nachbearbeitung der Szene (Anpassung der Höhen durch gewonnene Informationen)
5. Zusammenführung zu Gesamtbild einer Kamera
- 6 Vereinigung der Daten aller Kameras

Die Punkte 1-3 sind der Ergebnis des entsprechenden Hardware-Funktionsblocks, vorher als Funktionsblock 3 beschrieben. Die Suche nach einem Schatten ist einfach als Linescan realisiert, d.h. es wird innerhalb einer Bildzeile gesucht. Stellt der Algorithmus einen Hell-Dunkel-Wechsel fest, beginnt die Ermittlung der Schattenlänge. Aus der Länge dieses Schattens und der Position, an welcher der Schatten gefunden wurde, wird die Höhe des Punktes berechnet. Es handelt sich dabei um die relative Höhe bezogen auf seine Umgebung. Jedoch muss die Umgebungshöhe entsprechend angepasst werden, um an die absolute Höhe der Szene zu gelangen.

Dieser Vorgang wurde, wie bereits beschrieben, von anderen Umsetzungen stets iterativ gelöst. In der Nachbearbeitung des Bildes nach diesem Ansatz werden die gewonnenen Informationen aus den bekannten, ermittelten Schatten auf den Rest der Szene übertragen. So ergibt sich bereits ein Höhenprofil der Szene in jedem Einzelbild (siehe *Abbildung 16 a*)).

Dieses Bild ist naturgemäß mit einer extrem hohen Unsicherheit behaftet. Daher werden im Anschluss die Bilder einer Kamera, also eins für jede Beleuchtungsrichtung, zusammengeführt. Das ist eine simple Operation, die die Sicherheit der Aussage einer Kamera erhöht (siehe *Abbildung 16 b*)). Dabei werden

die Bilder miteinander verglichen und nur die Daten übernommen, die in allen Szenen vorhanden sind. Das Ergebnis der Einzelbildzusammenführung wird anschließend zu einem Gesamtbild vereint.

In diesem Schritt werden die Informationen als sicher betrachtet, die in allem Überlagerungsbereichen vorkommen (*Abbildung 17 b*). In dem Blickfeld von vier Kameras ist das Ergebnis der Rekonstruktion sicherer als dort, wo nur eine Kamera die Szene betrachtet.

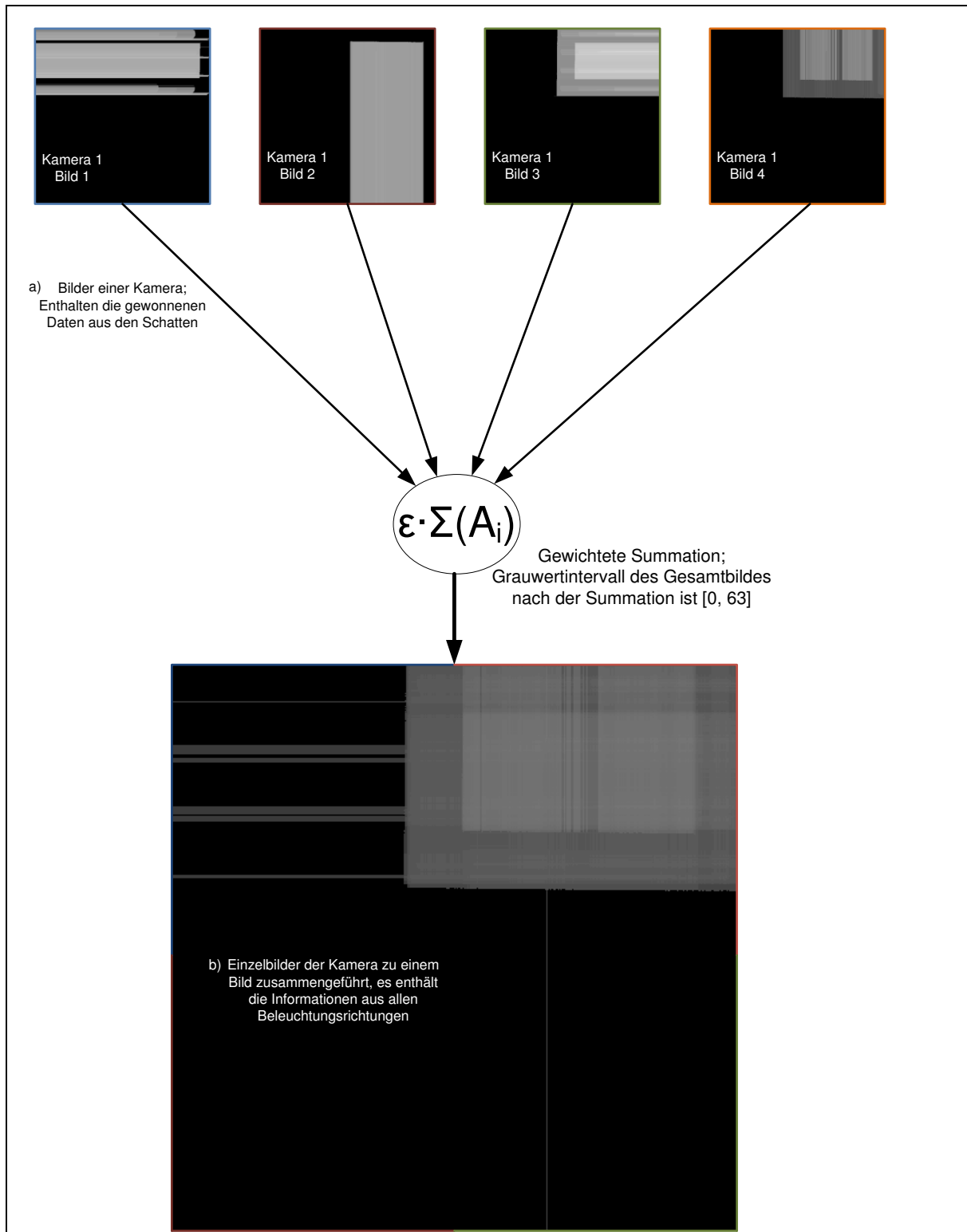


Abbildung 16: Zusammenführung der Einzelbilder; Schema

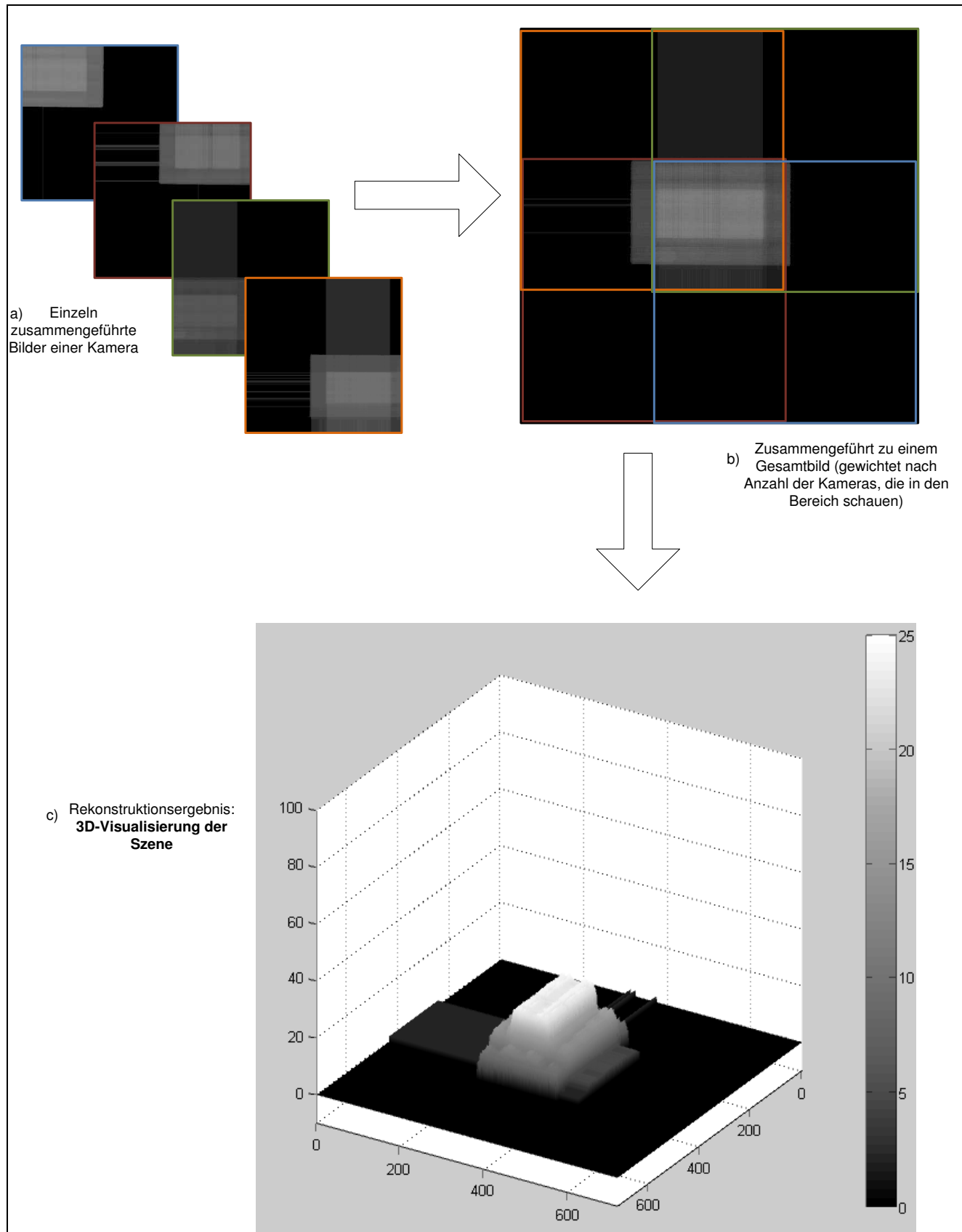


Abbildung 17: Vereinigung der Zusammengeführten Einzelbilder zu einem Gesamtbild

c) Darstellung des Rekonstruktionsergebnisses in Matlab

Um eine gute Reproduzierbarkeit der Bildserien zu gewährleisten, wurden alle Bilder mit derselben Versuchsanordnung aufgenommen und die Bildparameter dokumentiert. Eine Auflistung aller Bildserien und ihren Bildparameter ist auf dem Datenträger zu finden, der dieser Arbeit beiliegt.

Eine vollständige Darstellung des Versuchsaufbaus zeigt die *Abbildung 18*.

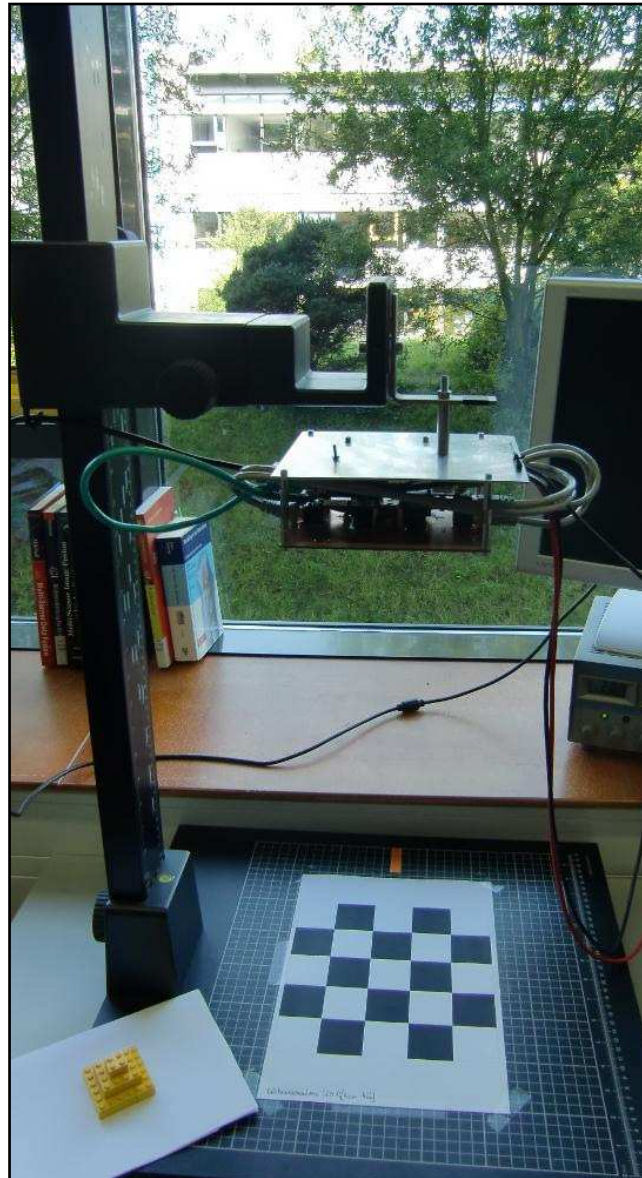


Abbildung 18: Versuchsanordnung

## 7 Ergebnisse

Auf den folgenden Seiten werden die Ergebnisse dieser Arbeit präsentiert. Offene Probleme werden ebenfalls vorgestellt.

### 7.1 MiniCluster

Abbildung 19 und Abbildung 20 zeigen die fertig montierte Baugruppe. Die entsprechenden Baugruppen

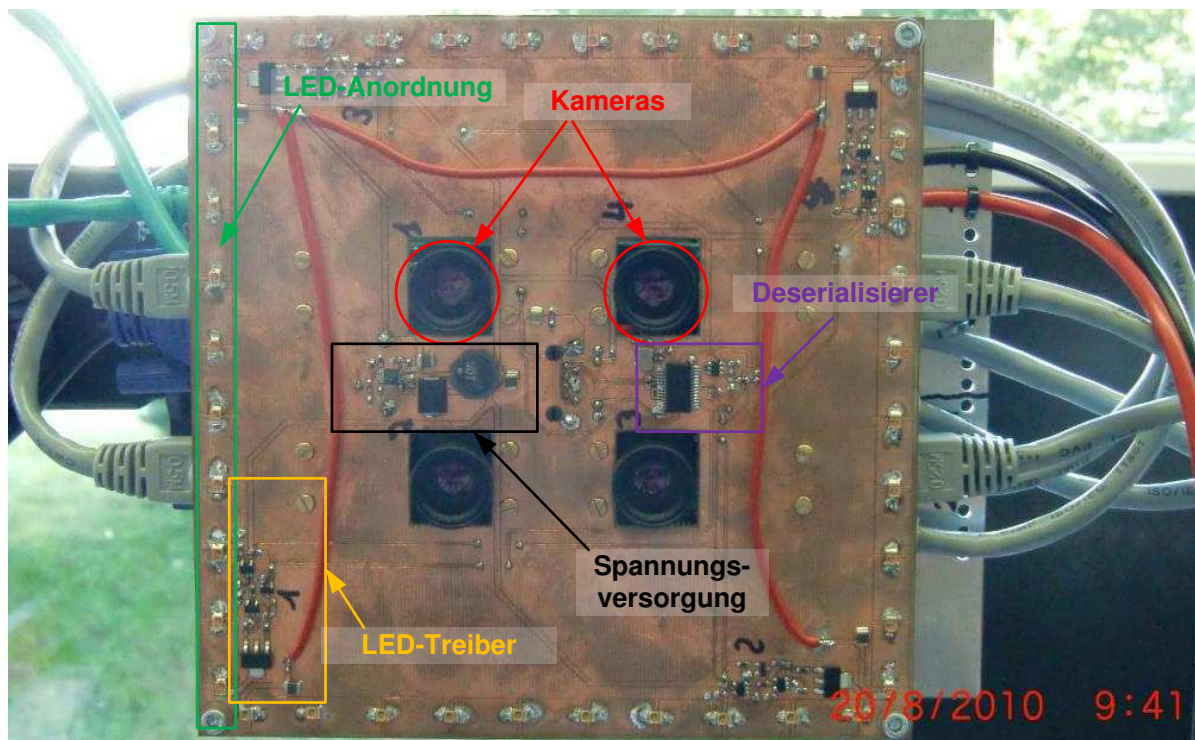


Abbildung 19: Baugruppe, Unteransicht mit markierten Komponenten

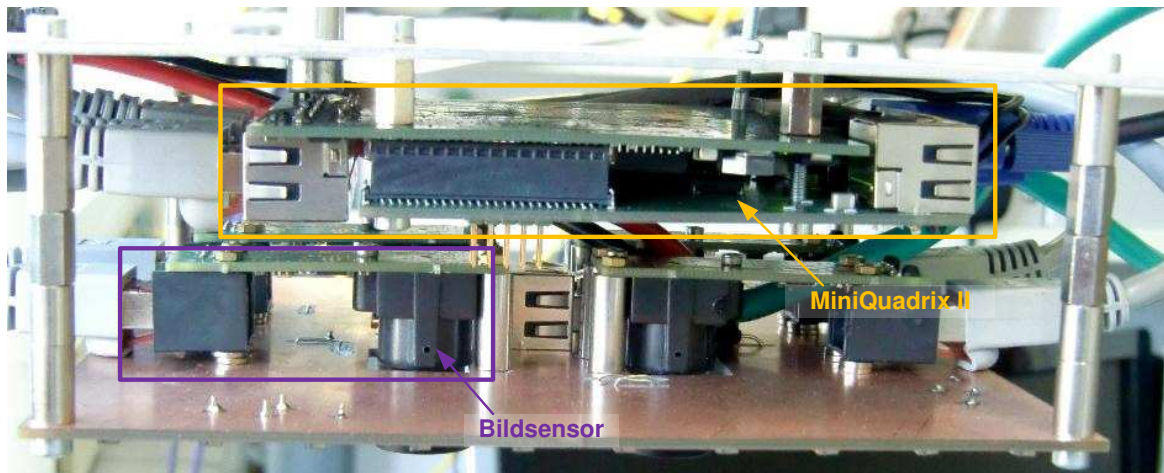


Abbildung 20: Baugruppe, Seitenansicht mit montierter MQ II und Bildsensoren

Im *Anhang H* befinden sich die technischen Zeichnungen für die gesamte Baugruppe, die Schaltpläne der Schaltungen sowie die PCB-Layouts.

## 7.2 3D-Szenenrekonstruktion

Die Rekonstruktion der Szene läuft in den in Kapitel 6 beschriebenen Schritten ab. Eine Auswahl von 3D-Szenen, die in dieser Arbeit erstellt wurden, befindet sich in *Abbildung 22*.

Die Dauer der Rekonstruktion einer Szene ist *Abbildung 21* in beschrieben. Diese Zeit setzt sich aus folgenden Teilzeiten zusammen.

- Aufnahmedauer einer Bildserie
- Verarbeitung der Schatteninformation
- Zusammenführung der Bilddaten nach der Aufnahme aller Bildserien

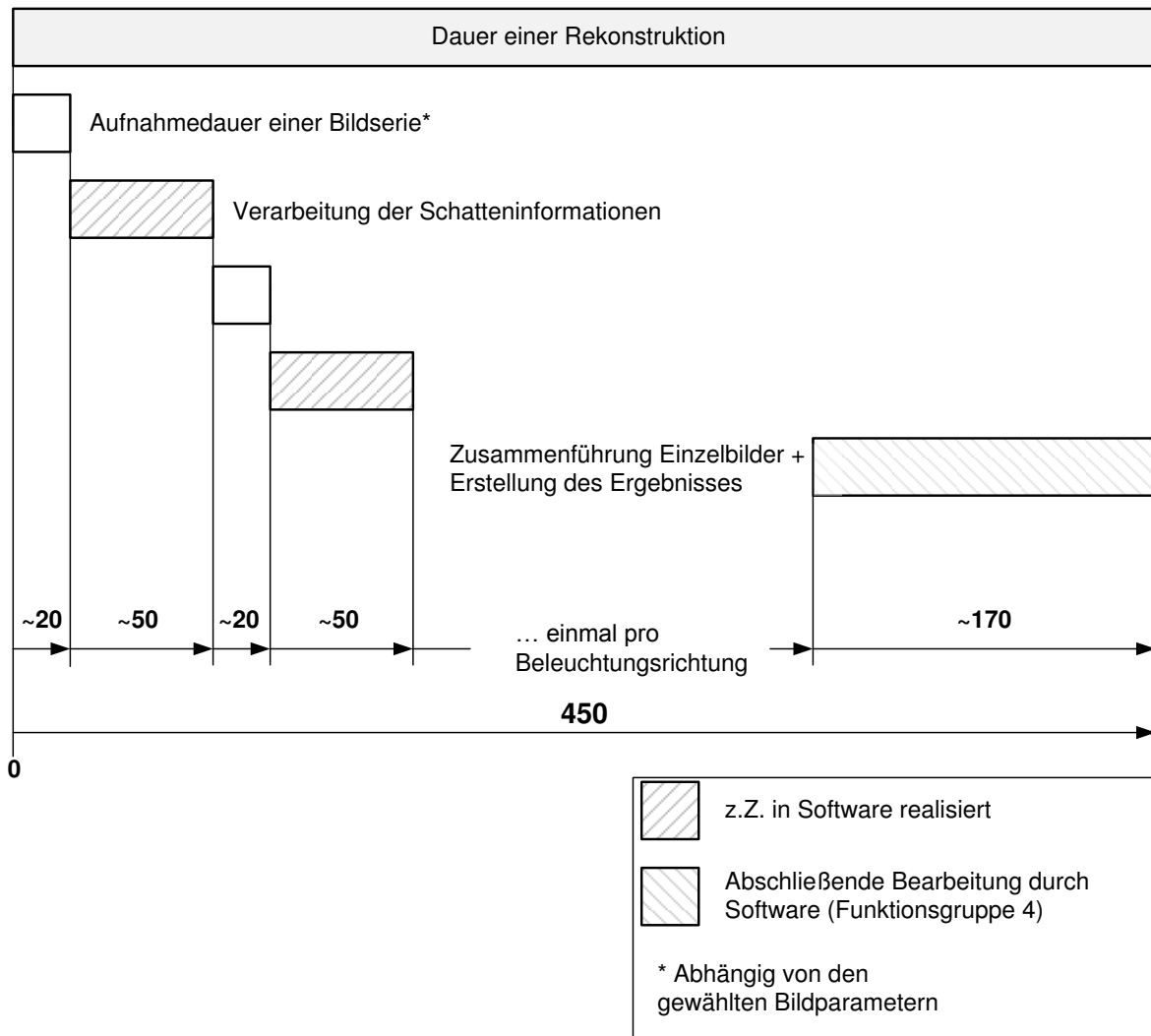


Abbildung 21: Dauer der Rekonstruktion einer Szene; Zeitangaben in ms

Die Zeit, die für eine Rekonstruktion benötigt wird, ist um ein vielfaches länger als ursprünglich angestrebt. Das liegt daran, dass es bis zum Ende der Arbeit nicht gelungen ist die beschriebene Aufteilung der Funktionsgruppen auch umzusetzen. Der Ansatz, mit SystemC eine Custom-Instruction zu erstellen, ist nicht erfolgreich abgeschlossen worden. Das bedeutet, die Rekonstruktion läuft ab der Funktionsgruppe 3 vollständig in Software. Dadurch ist die Zeit, die für eine Verarbeitung der Schatten benötigt wird, viel zu hoch. Wäre es gelungen, diesen Schritt erfolgreich abzuschließen, wäre eine theoretische Aquirierungsrate (Framerate) von vier bis fünf Bildern pro Sekunde möglich.

Der Grund für die nicht erfolgreiche Umsetzung ist die Unerfahrenheit in der Entwicklung mit SystemC.



Die Ergebnisse zeigen, dass die Rekonstruktion in den Grundzügen gut funktioniert. Es ist möglich, einfache Objekte bis Objekte mittlerer Komplexität zu erfassen.

Allerdings weisen die Ergebnisse einen hohen Fehler auf. Eine quantitative Fehlerbetrachtung des Algorithmus fällt schwer, da er nicht daraufhin entwickelt, eine genaue Vermessung der Geometrie zu ermöglichen. In *Abbildung 22 a) und b)* werden die Rekonstruktionsergebnisse der Bilddatensets 7 und 8 dargestellt. Dort werden sie mit den Ausgangsobjekten verglichen.

Dabei wird deutlich, dass es zu teilweise hohen Abweichungen in den Details kommt. Hingegen werden die groben Grundzüge der in der Szene befindlichen Objekte gut rekonstruiert.

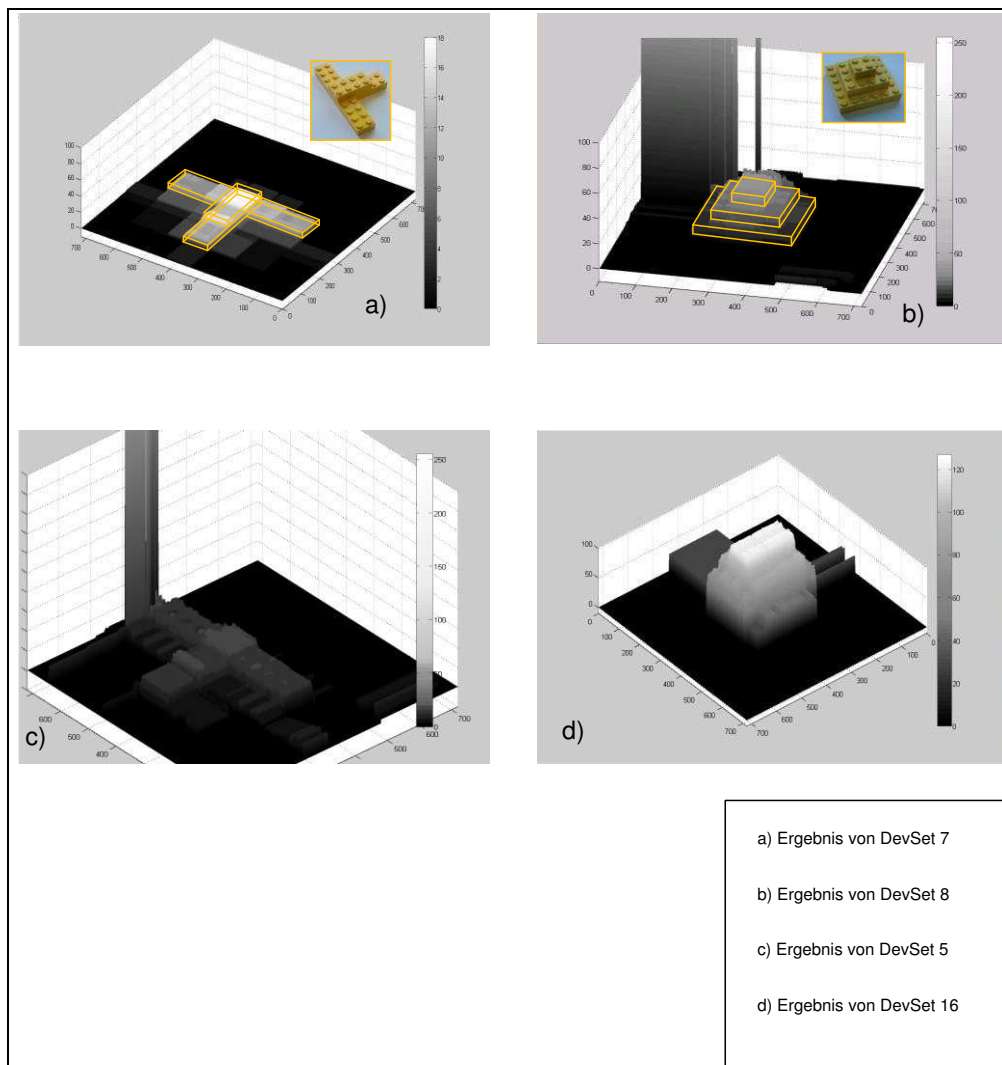


Abbildung 22: Ergebnisse in 3D-Darstellung

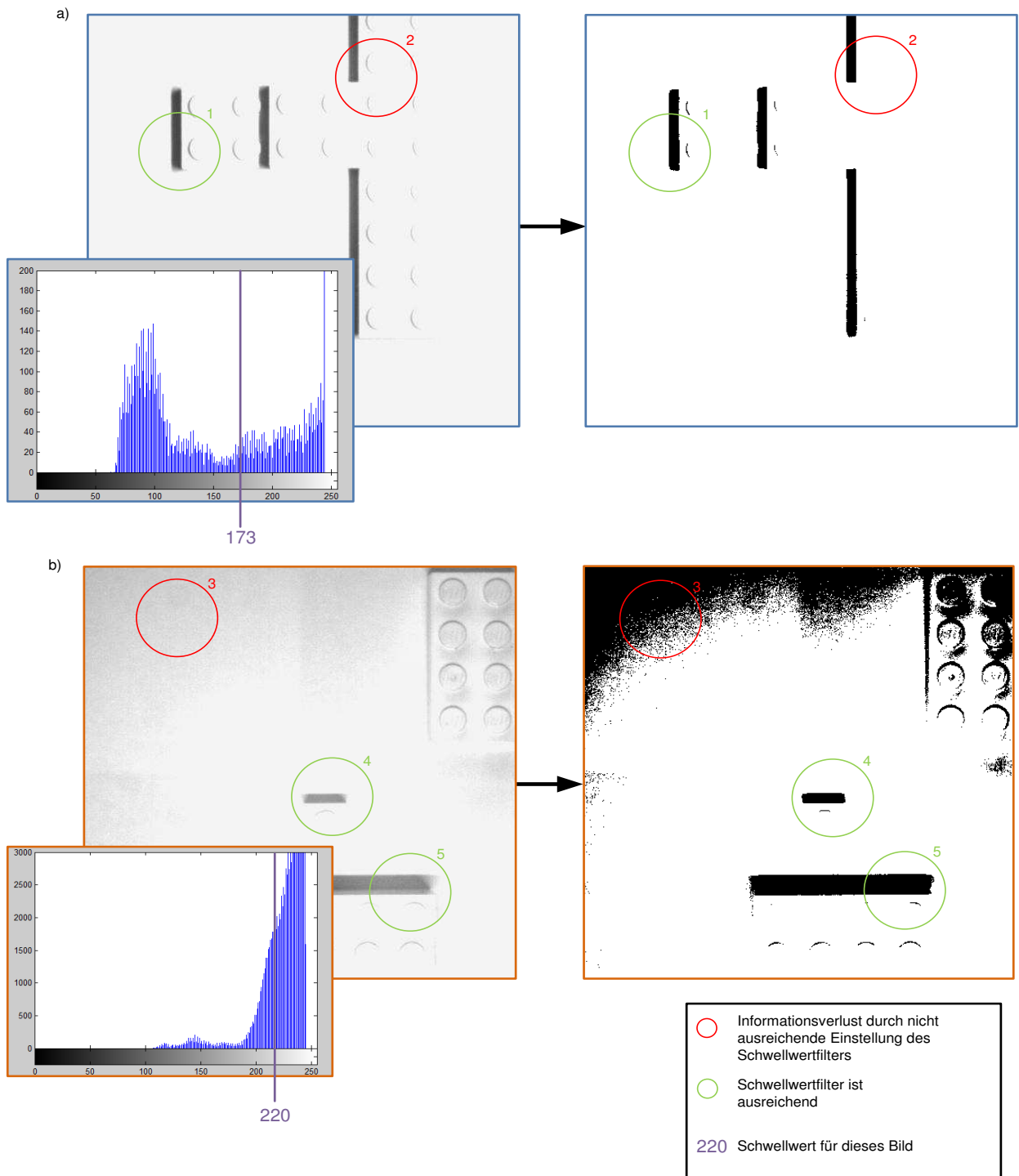
### 7.3 Probleme

Neben den Problemen mit der Umsetzung des Algorithmus in Hardware traten bei der Umsetzung noch andere Komplikationen auf.

Das Hauptproblem ist die geeignete Aufbereitung der Datengrundlage.

Wie in *Abbildung 23* dargestellt ist es sehr wichtig einen guten Schwellwert zu finden. Dabei ist in *a)* gezeigt, wie es zu Informationsverlusten durch einen nicht zureichenden Schwellwert kommt. Hingegen *b)* weist kaum Informationsverluste auf. Das der Auswertung übergebene Schwellwertbild ist allerdings als komplett ungeeignet zu betrachten, da zu viel vom Rauschen in dem Bild enthalten ist.

Zu welchen Fehlinterpretationen es führen kann ein solches Bild in der Rekonstruktion zu führen zeigt *Abbildung 24*.



Bilder aus DevSet: a) 3\_1\_1.bmp; b) 5\_3\_0.bmp

Abbildung 23: Datenaufbereitung durch Schwellwertfilterung vor der Datenanalyse

Histogramm für das jeweilige Bild ist angegeben mit entsprechenden Schwellwert; a) zeigt: gutes Ergebnis, relativ viel Datenverlust; b) zeigt: Ergebnis nahezu unbrauchbar; keine Daten verloren, dafür keine Verdichtung

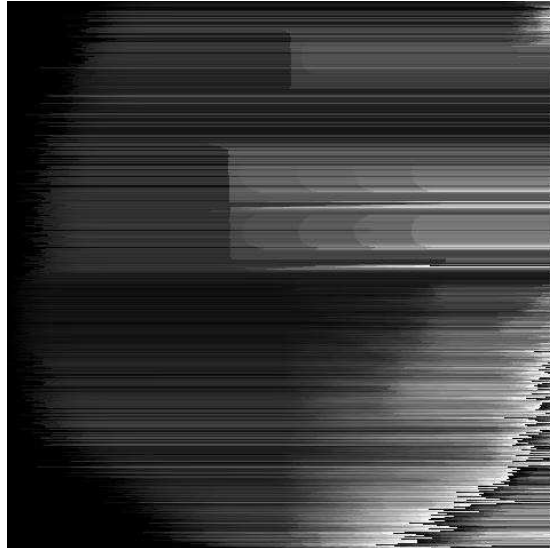


Abbildung 24: Durch Rauschen verursachter Fehler

*Abbildung 24* macht deutlich, wie wichtig eine verlässliche Filterung ist. Sind in den nach der Schwellwertfilterung zur Rekonstruktion benutzten Daten in der Art fehlerträchtig, ist ein gutes Ergebnis faktisch auszuschließen.

## 8. Abschluss

### 8.1 Reflektion und Diskussion

Betrachtet man die Ergebnisse, die in dieser Arbeit erreicht wurden zeigt sich, dass es sich bei der Umsetzung des Shape from Shadows-Verfahrens, auf eine Hardwareplattform, nicht um eine Trivialität handelt.

Durch die kurze Bearbeitungszeit, die für dieses Projekt zur Verfügung stand, sind der Abschluss und die Ergebnisse nicht vollkommend zufriedenstellend ausgefallen. Es gibt eine Vielzahl von Punkten, die gezielt in anschließenden Arbeiten verbessert werden können. Im Rahmen dieser Bachelorarbeit konnten viele, detaillierte Erkenntnisse gewonnen werden, die in diesem Dokument nicht explizit erwähnt werden, aber in den erbrachten Ergebnissen (z.B. Quellcodekommentaren) auftauchen.

Die Bildgebung ist, wie in den Ergebnissen deutlich wurde, noch Gegenstand für

Verbesserungen. Die in dieser Arbeit verwendete Methode die gewonnenen Bild-daten aufzuarbeiten (Schwellwert) erwies sich als diffizil. Es wurde zu viel Zeit darauf verwendet, eine manuelle Schwellwerteinstellung zu finden (vergl. *Abbildung 23*). Es müssen daher Maßnahmen gefunden werden, geeignete Einstellungen des Schwellwertfilters automatisch vorzunehmen.

Das Rekonstruktionsergebnis selber ist mit weiteren Problemen behaftet, die zum einen in der Natur der Schatten liegen (Stichwort: hochgradige Unbestimmtheit) als auch in der Beleuchtung, die in dieser Arbeit umgesetzt wurde. Um ein gutes Rekonstruktionsergebnis zu erhalten, ist die Kenntnis über die grobe Objektgeometrie notwendig. Diese Geometrie ist es aber, die wir erhalten möchten. Das schränkt die Allgemeingültigkeit des Verfahrens weiter ein.

Wie sich zeigte, war die Datengrundlage nicht ausreichend. Dabei geht es weniger um die Güte als schlicht um die Menge an aussagekräftigen Rohdaten. Die vorgenommene Diskretisierung des Shadowgrams auf vier Beleuchtungsrichtungen erwies sich als zu gering. Zu erwarten ist, dass sich die Rekonstruktion um ein Vielfaches vereinfacht, wenn man die Anzahl an Beleuchtungsrichtungen erhöht. Dadurch lässt sich auch eine bessere Allgemeingültigkeit des Ergebnisses herstellen.

Aus den Ausführungen in Kapitel 7.2 geht hervor, dass die Ausführungsgeschwindigkeit des Algorithmus nicht optimal ist. Der Grund dafür ist, dass die Bearbeitungszeit es nicht erlaubte, die in den Funktionsgruppen der Hardware zugeordneten Blöcke, in Hardware umzusetzen.

## 8.2.Ausblick

Als wesentliches Problem bei der Rekonstruktion stellte sich die geringe Informationsdichte heraus. Wie die Ergebnisse zeigen, kann nur für eine kleine Zahl der Objekte mit zufriedenstellenden Ergebnissen gerechnet werden.

Für viele Objekte können die aus den Bildern und Schatten gewonnenen Informationen keine Eindeutigkeit herstellen. Um die Szene möglichst allgemeingültig zu rekonstruieren gilt: je größer der Ausgangsdatenbestand desto besser. In dem in dieser Arbeit beschriebenen Ansatz wurden nur vier diskrete Beleuchtungswinkel realisiert. Um an mehr Informationen zu gelangen ist die Erweiterung der Baugruppe des MiniClusters oder eines entsprechenden größeren Clusters um weitere LED-

Anordnungen denkbar.

Ist der Datenbestand jedoch gering, wirken sich kleinste Fehler bei der Informationsverdichtung stark auf der Ergebnis aus. Daher ist es sehr wichtig, diese Aufbereitung solide durchzuführen. Dahingehend kann der Algorithmus erweitert und damit verbessert werden. Zum Beispiel kann man eine automatische Ermittlung des Schwellwertes anhand eines Referenzbildes durchführen, um anschließend alle Bilder der Serie mit diesem Schwellwert zu filtern. Das wäre zu erreichen, indem man vor der Schwellwertfilterung ein Histogramm der Szene erstellt, es geeignet teilt und somit das Ergebnis optimiert (vergl. hierzu *Abbildung 23*- jeweiliges Histogramm). Das würde die Aufbereitung der Daten um ein vielfaches Robuster machen.

Nach den Erfahrungen dieser Arbeit kann eine weitere Filterung der Daten das Ergebnis der Rekonstruktion verbessern. Vorstellbar wäre eine Verknüpfung der Schwellwertfilterung mit einer Kantenerkennung der Originalszene. Dazu eignen sich Kantendetektoren wie der Canny-Algorithmus oder der Sobel-Operator (*Abbildung 25*).

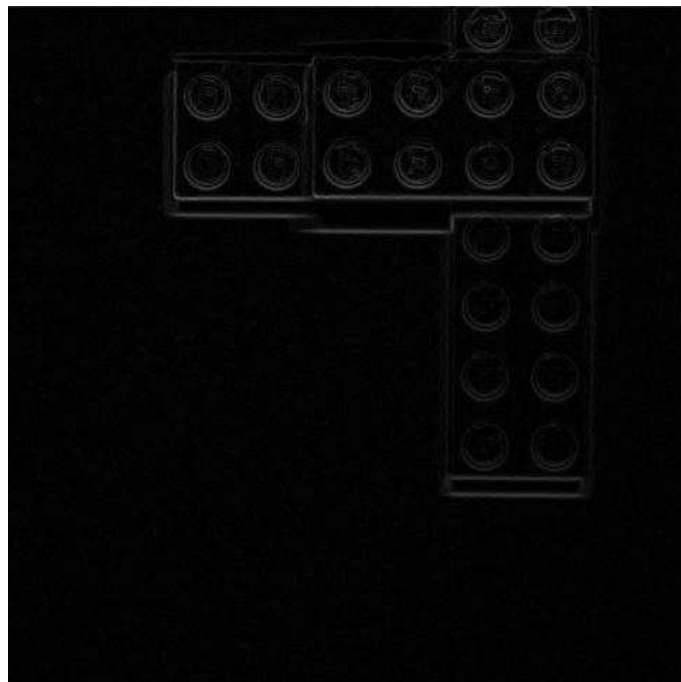


Abbildung 25: Kantenerkennung der Szene mit Sobel-Operator

Ferner kann man überlegen, ob es überhaupt der günstigste Weg ist, allein aus den

Schatten der Szene die 3D-Daten zu gewinnen. Schatten sind nicht die einzigen Eigenschaften, die bei geeigneter Beleuchtung die in einer Szene befindlichen Objekte beschreiben. Neben den aus den Schatten gewonnenen Informationen können noch andere Bild-/ bzw. Beleuchtungseigenschaften dazu genutzt werden, ein viel besseres 3D-Rekonstruktionsergebnisse zu erhalten. Es ist zum Beispiel denkbar, die Schatteninformationen mit der Silhouette des Objektes oder der Verteilung der Intensitäten zu verknüpfen. Auch zu dieser Art der Rekonstruktion gibt es bereits Arbeiten mit entsprechenden Ergebnissen [BAL06-14]. *Abbildung 26* ist aus dieser Veröffentlichung entnommen, dort wurde dieser Ansatz verfolgt. Zurzeit gibt es keine Aussage darüber, ob eine technische Umsetzung mit der MiniQuadrix II möglich wäre.

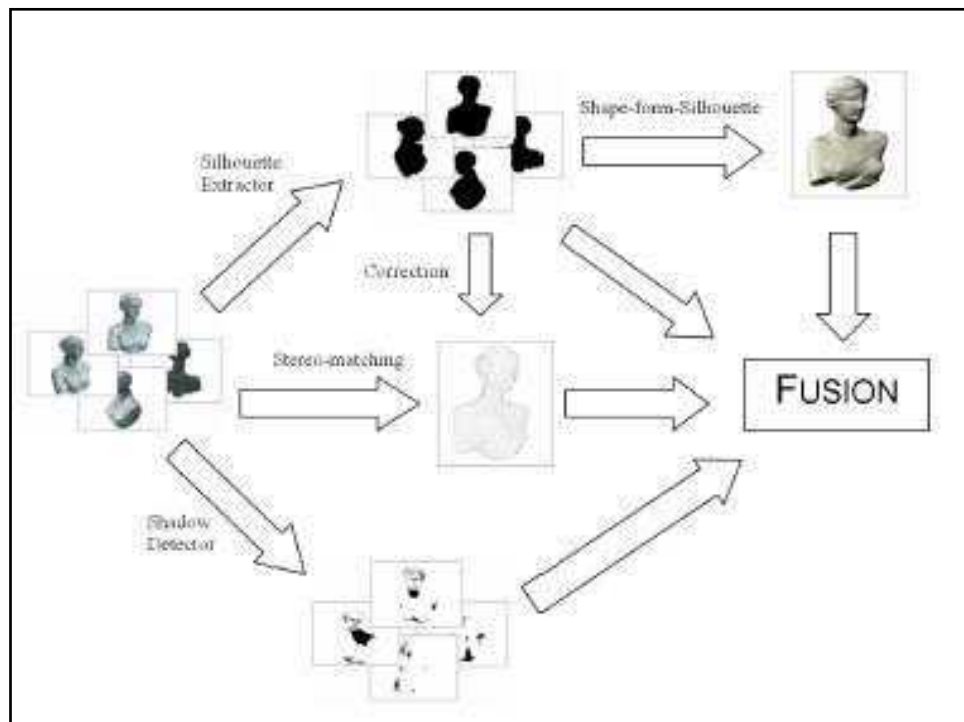


Abbildung 26: [BAL06-14] Vereinigung von verschiedenen Bildinformationen

Auch kann die Anordnung der Kameras überdacht werden, die Flexibilität der MQ II würde die notwendigen Anpassungen möglich machen. Dabei wäre es denkbar, die Kameras nicht in einer Ebene anzuordnen, wie es mit dem MiniCluster umgesetzt wurde, sondern konzentrisch um den Körper herum. Ziel ist eine Verdichtung der Information schon aus der Anordnung der Sensoren heraus.

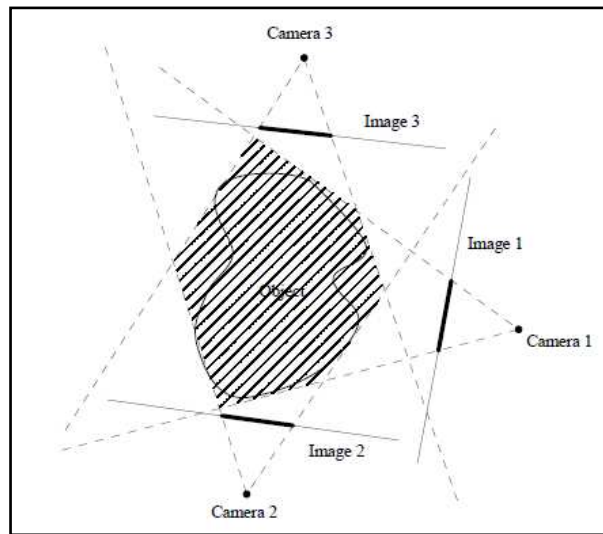


Abbildung 27: [SIL05-21] Gewinnung der Information durch konzentrische Anordnung der Kameras

Die Abbildung 27 zeigt die Möglichkeit eine anordnungsbedingte Verdichtung der Information vorzunehmen. Dabei ist die Schnittmenge der Bildbereich relevant für die Rekonstruktion.

## 9. Appendix

### 9.1 Begriffserläuterungen

#### Algorithmus

Der Begriff des Algorithmus wird in der Literatur wie folgt definiert:

*Ein Algorithmus (auch Lösungsverfahren) ist eine formale Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten. [WIK10-2]*

Ein wesentliches Ziel dieser Arbeit war es einen Algorithmus zu erstellen, der in Hardware ausgeführt wird.



## FPGA

*Ein Field Programmable Gate Array (kurz: FPGA) ist ein Integrierter Schaltkreis (IC) der Digitaltechnik, in den eine logische Schaltung programmiert werden kann. [WIK10-1]*

Durch das Funktionsprinzip eines FPGAs ist dieser Baustein aus der Digitaltechnik nicht mehr wegzudenken. Es ist möglich, die internen Strukturen zu konfigurieren, um verschiedene Verschaltungen von Logikgattern zu ermöglichen. Dabei machen sie es sowohl möglich Schaltungen geringer Komplexität (z.B. Zähler) als auch sehr komplexe Schaltungen (z.B. Mikroprozessoren) zu realisieren. Ein weiterer Vorteil ist, dass Vorgänge parallel ablaufen können. Das stellt die Grundlage für eine schnelle Signal-/ Datenverarbeitung dar. Sinnvoll lassen sich FPGAs dort einsetzen, wo es notwendig ist, nachträgliche Veränderungen an einer Implementierung vorzunehmen. Dies ist vor allem in der Entwicklung von Algorithmen zur digitalen Bildverarbeitung der Fall.



Abbildung 28: FPGA-Cyclone II [WIK10-23]

## MiniCluster

Ein Teilziel dieser Arbeit ist die Entwicklung eines MiniClusters, daher wird dieser Begriff häufig genannt. Dabei handelt es sich um einen Zusammenschluss von vier Kameras zu einem Funktionsverbund sowie einer entsprechend entwickelten Schaltung zur Szenenbeleuchtung. Abgeleitet ist dieser Verbund von einem Cluster, der mit einer größeren Anzahl an Kameras ausgestattet ist.

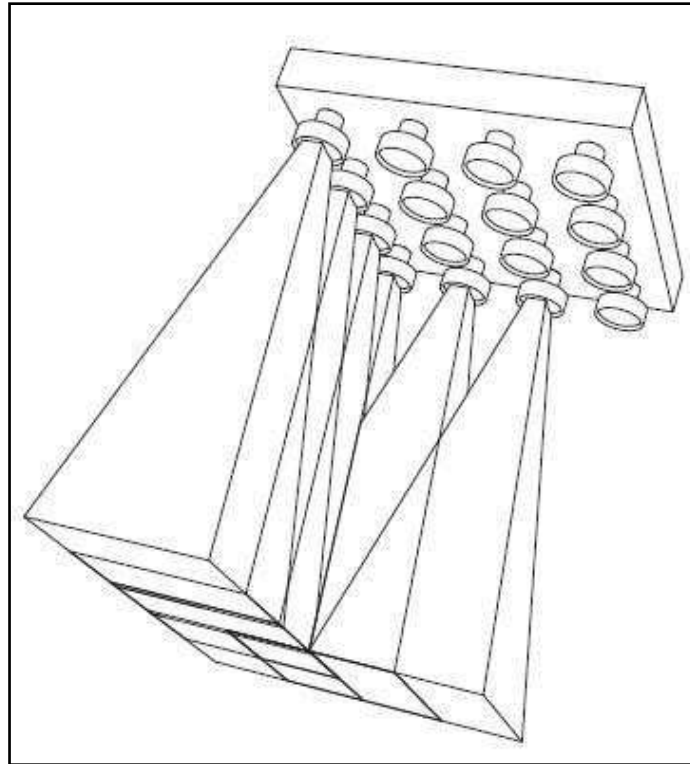


Abbildung 29: Sensorcluster

Abbildung 29 zeigt ein Sensorcluster mit 4 x 4 Kamerasensoren, wie es im Pflichtenheft zu dieser Arbeit beschrieben wird. Ein Prototyp eines größeren Clusters ist in der Arbeitsgruppe BuS entwickelt worden. In der Abbildung gut erkennbar ist dabei die Überlagerung der Bildbereiche einzelner Sensoren.

## Digitales Filter

*Ein digitales Filter ist ein elektronisches Filter zur Manipulation eines Signals [...] Der Unterschied zum Analogfilter liegt in der Realisierung: Analoge Filter werden mit passiven elektronischen Bauelementen wie Kondensatoren, Spulen, Widerständen oder aktiv mit Operationsverstärkern aufgebaut. Digitale Filter werden mit Logikbausteinen wie ASICs, FPGAs oder in Form eines sequentiellen Programmes mit einem Signalprozessor realisiert. [WIK10-16]*

Verschiedene Digitale Filter (z.B. Schwellwertfilter) werden in der Bearbeitung dieser Bachelorarbeit verwendet und ihr Einfluss auf die Bildrekonstruktion beschrieben.

## Echtzeit

Echtzeitverhalten ist ein wesentliches Merkmal eines Algorithmus, der in Hardware ausgeführt wird. Echtzeit ist wie folgt definiert:

*In der Informatik spricht man von Echtzeit (englisch: real-time, deutsch (veraltet): Realzeit), um sie von einer Modellzeit zu unterscheiden. Echtzeit bedeutet dabei die Zeit, die Abläufe in der realen Welt verbrauchen. Modellzeit hingegen bedeutet die von einer Software selbstverwaltete Laufzeit. [WIK10-8]*

Zeigt sich die Modellzeit synchron zur Echtzeit spricht man von echtzeitfähigen Systemen. Weiterhin wird bei der Definition der Echtzeit zwischen weicher und harter Echtzeit differenziert:

*Hartes Echtzeitverhalten bedeutet [...], dass anhand von Hardwarespezifikationen und Modellrechnungen eine beweisbare obere Grenze für die Dauer eines Vorgangs angegeben werden kann. [WIK10-8]*

Typischer Weise zeichnen sich Hardwarealgorithmen durch hartes Echtzeitverhalten aus. Man kann also seine Laufzeit genau vorherbestimmen bzw. mit geeigneten Werkzeugen messen.

## **Infrarot**

Infrarot bezeichnet einen Spektralbereich elektromagnetischer Wellenlängen zwischen sichtbarem Licht und Terahertzstrahlung. Typische Wellenlängen liegen also zwischen 720 nm und 1 mm für den IR-Bereich.

*Abbildung 30* zeigt einen Vergleich, der für den Menschen wahrnehmbaren Wellenlänge des Lichtes und der von den Leuchtdioden emittierten Strahlung.

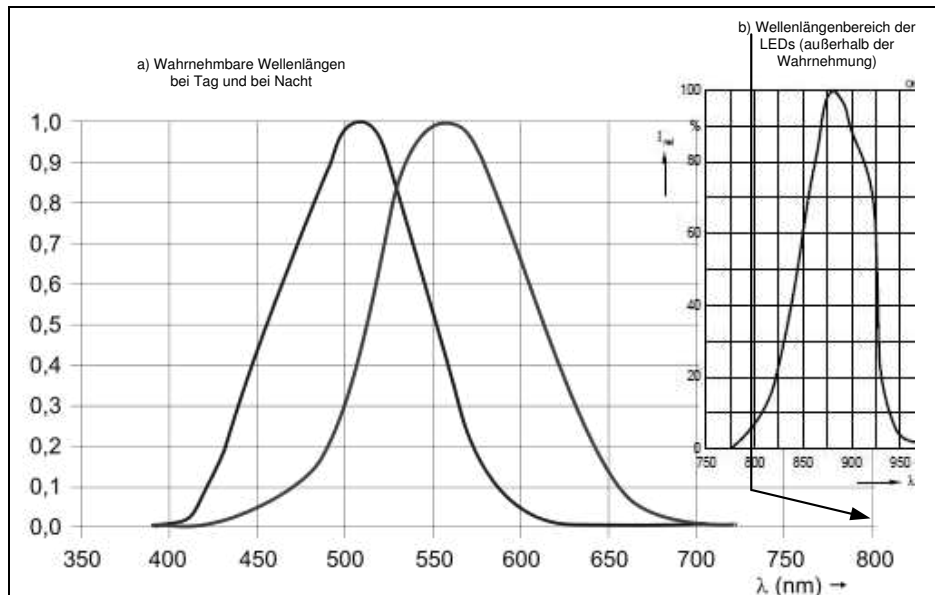


Abbildung 30: Wellenlängenbereich des Sichtbaren Lichtes für den Menschen

Die Darstellung zeigt, dass IR nicht sichtbar für den Menschen ist. Technisch kann man das Spektrum jedoch nutzen, z.B. sind CCDs-Chips in der Lage das IR-Spektrum aufzunehmen.

## Offset

Unter einem Offset ist ein konstanter additiver oder subtraktiver Faktor zu verstehen, der bei einer Messung i.d.R. immer in den Ergebnissen zu finden ist. Ein Offset ist somit ein systematischer Fehler und kann durch geeignete Maßnahmen (Stichwort: Kalibrierung) ermittelt werden. Ist der Offset bekannt, ist der Fehler nicht länger relevant und damit die Messung sicher.

## 9.2 Eingesetzte Software

In dieser Arbeit wurde folgende Software eingesetzt. Die Angabe erfolgt mit Versionsnummer.

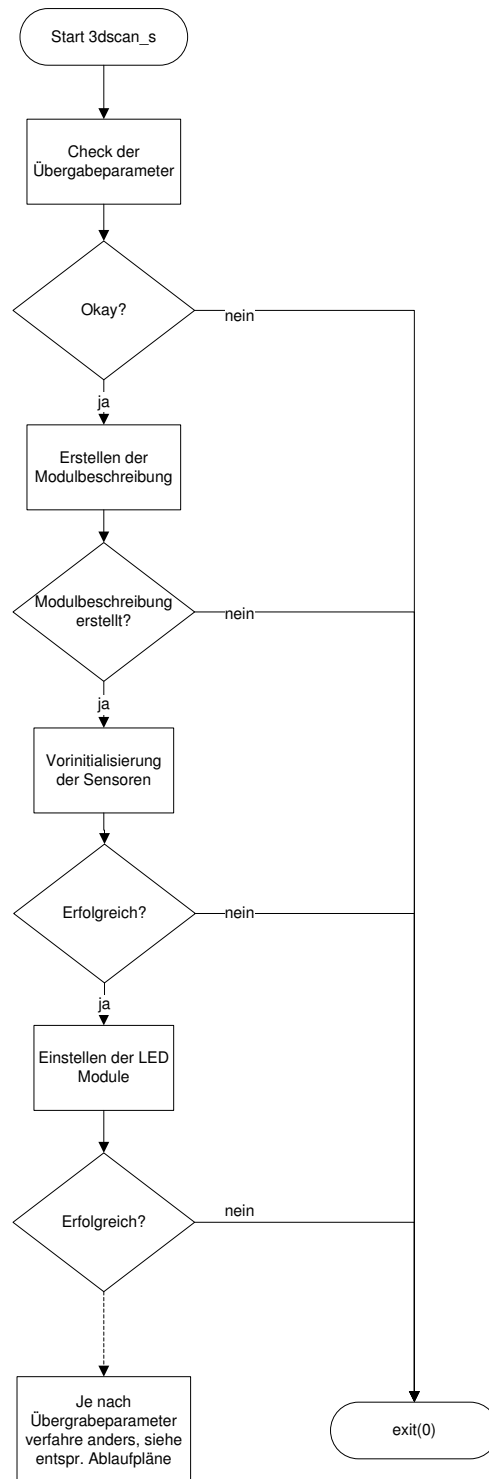
OpenCV	Open Computer Vision; Quellcodeoffene -Bibliothek der Firma Intel zur Bildverarbeitung (Version 2.0; Linux; GTK+)
--------	-------------------------------------------------------------------------------------------------------------------

nios2-gcc	Compiler der Herstellers Altera für den Nios II Prozessor
Matlab	Programm zur Lösung mathematischer Problemstellung der Firma The MathWorks, Inc. (Version 7.10 R2010a)
Doxygen	Software zur automatisierten Dokumentation von Quellcode (Version 1.7.1)
MQ II-API	Programmierschnittstelle für die Hardware das MQ II (Version 0.1)
GIMP	Quellcodeoffenes Bildbearbeitungsprogramm (Version 2)

### 9.3 Glossar

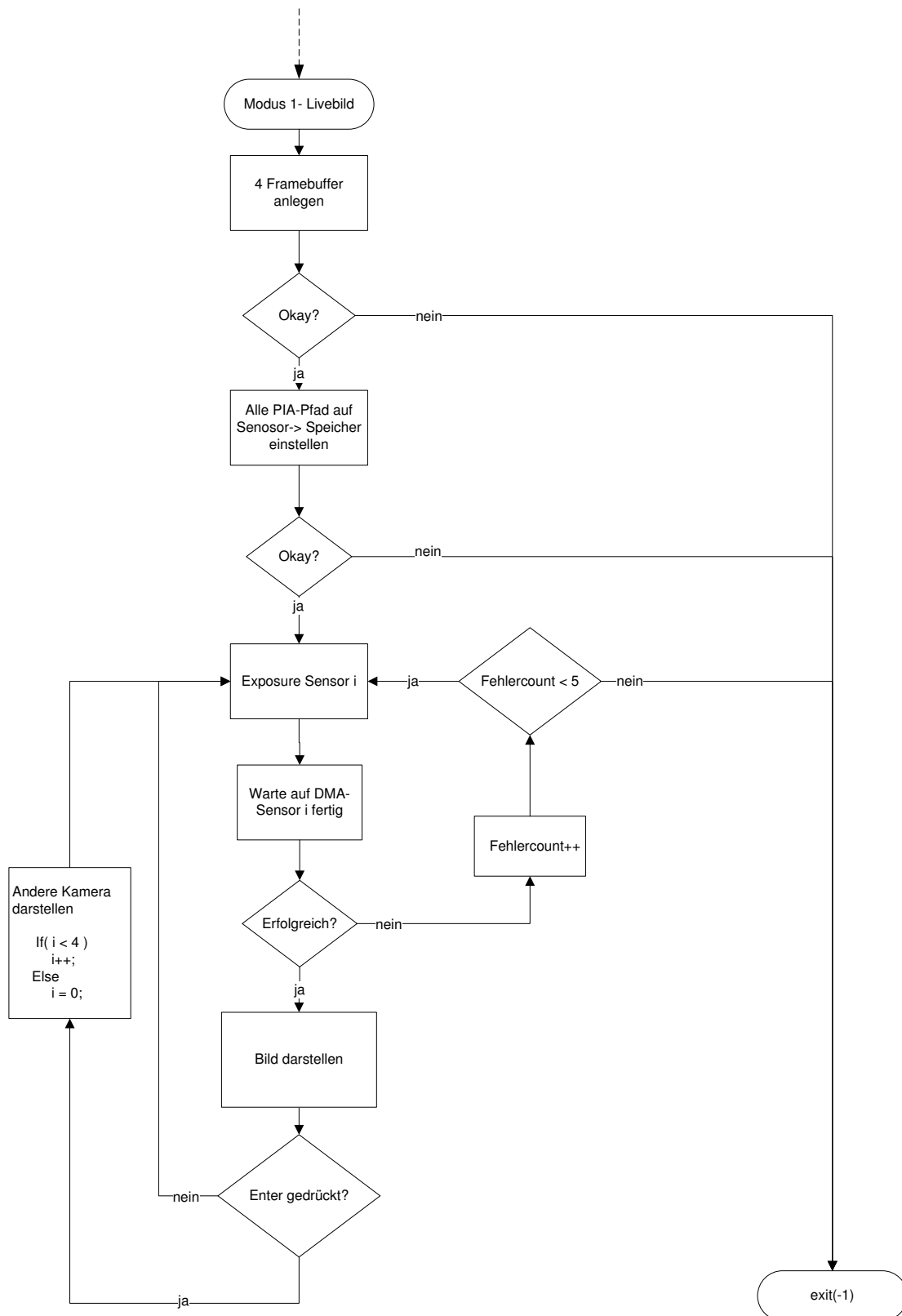
OpenSource	Quellcodeoffene Software, Der Quellcode kann frei heruntergeladen und bearbeitet werden
Wikipedia	Freie Enzyklopädie
µLinux	(Mikro)Linux- Quelloffenes Betriebssystem, speziell geeignet für Embedded-Systeme

## Anhang A: Ablaufplan 3dscan\_s- Severstart

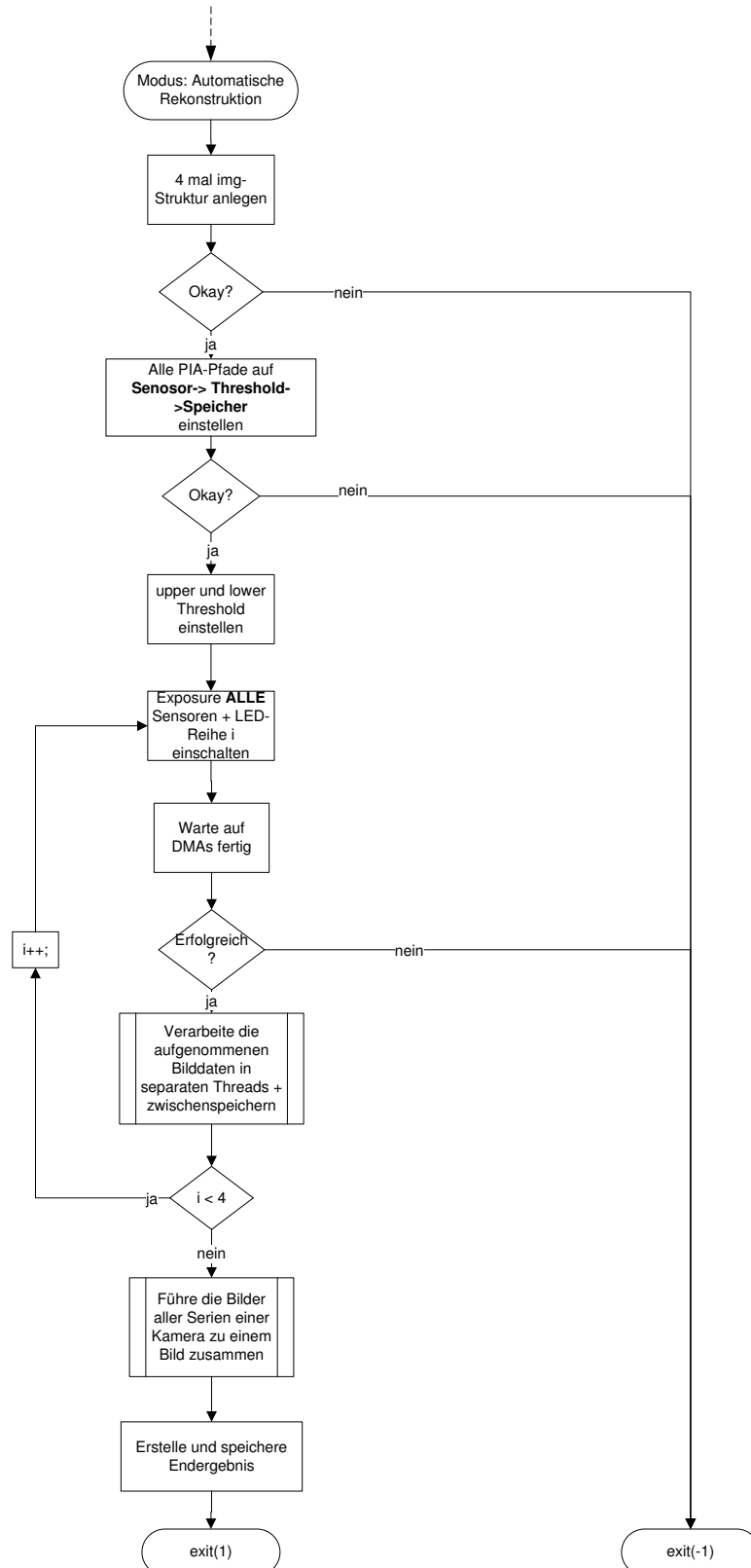


## Anhang B: Ablaufplan 3dscan\_s- Livemodus

Zeige Livebild auf Monitor am VGA-Anschluss an.

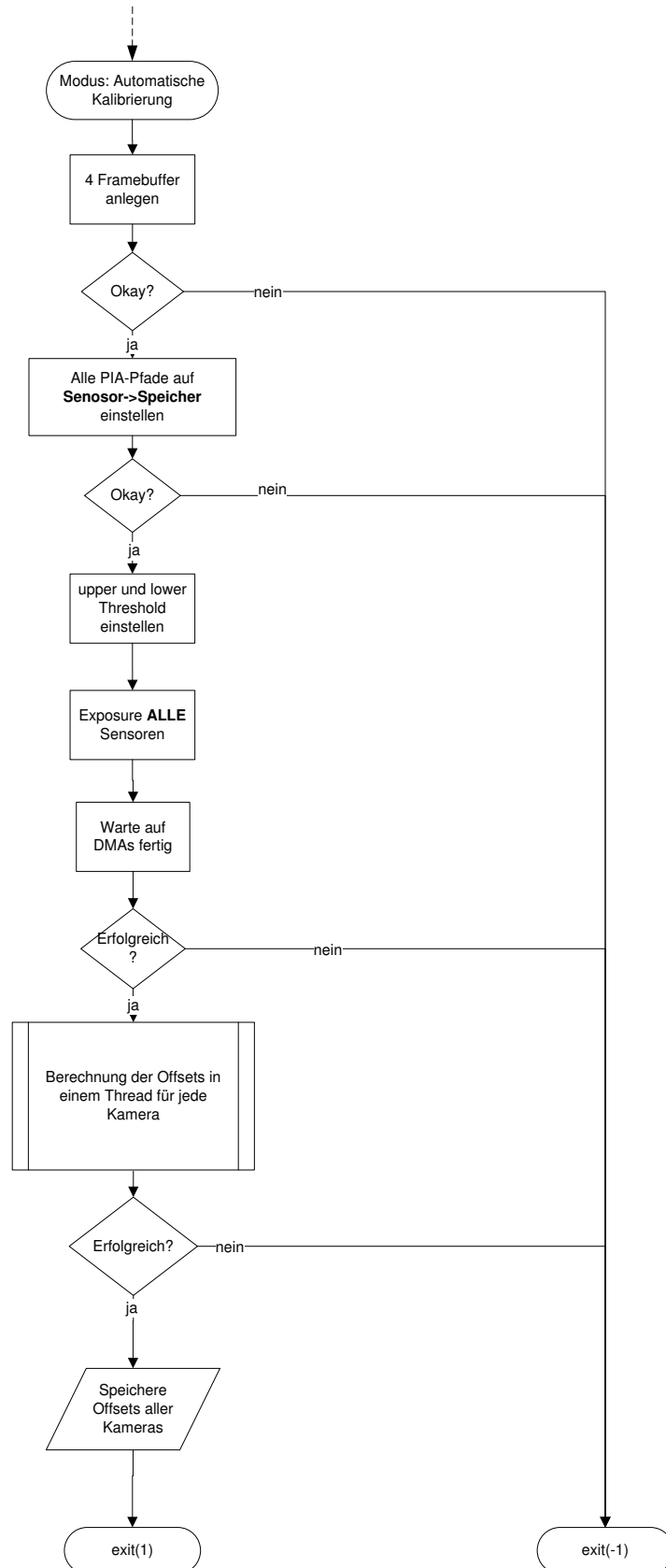


## Anhang C: Ablaufplan 3dscan\_s- Automatische Rekonstruktion

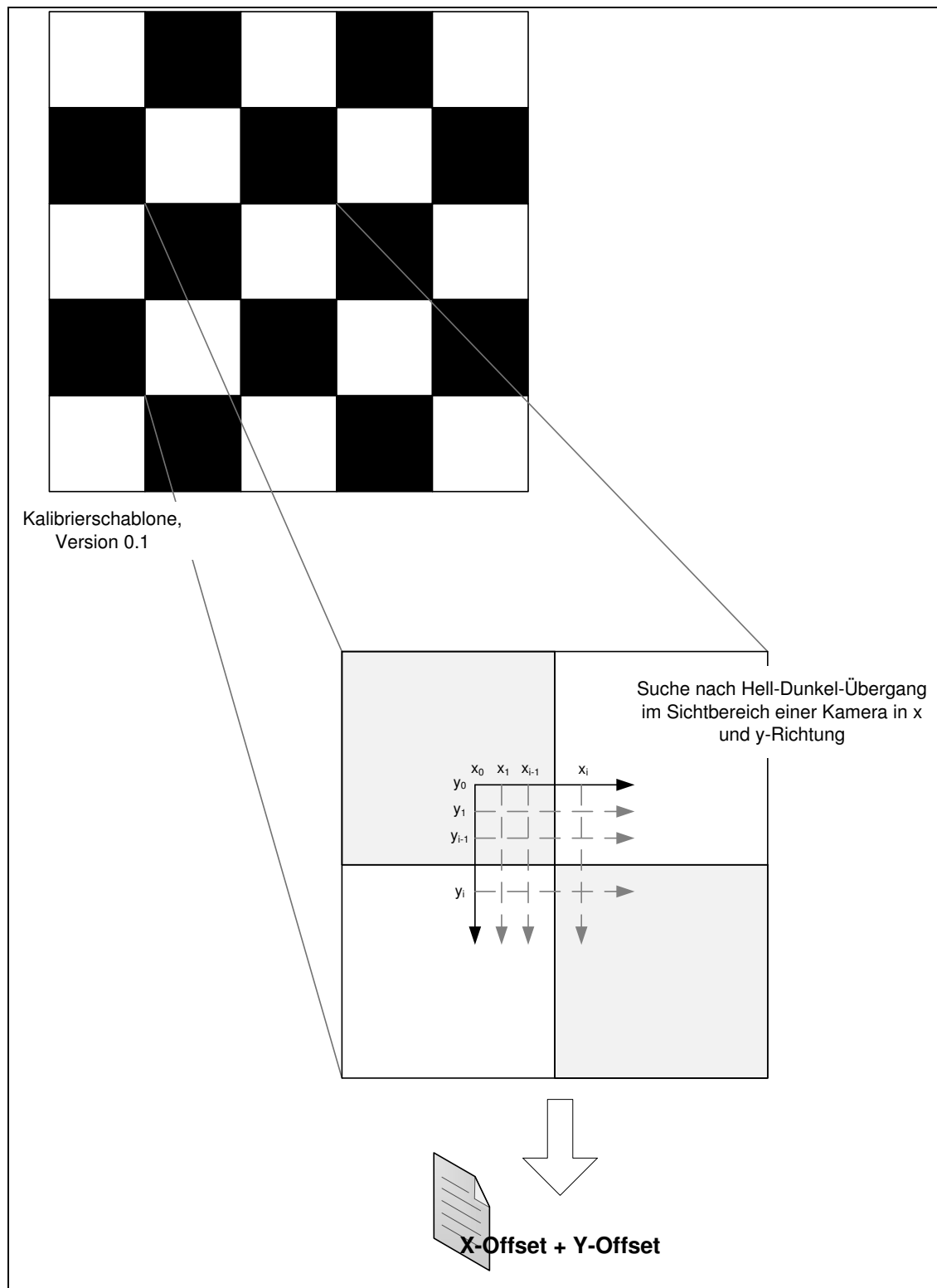




## Anhang D: Ablaufplan 3dscan\_s- Automatische Kalibrierung

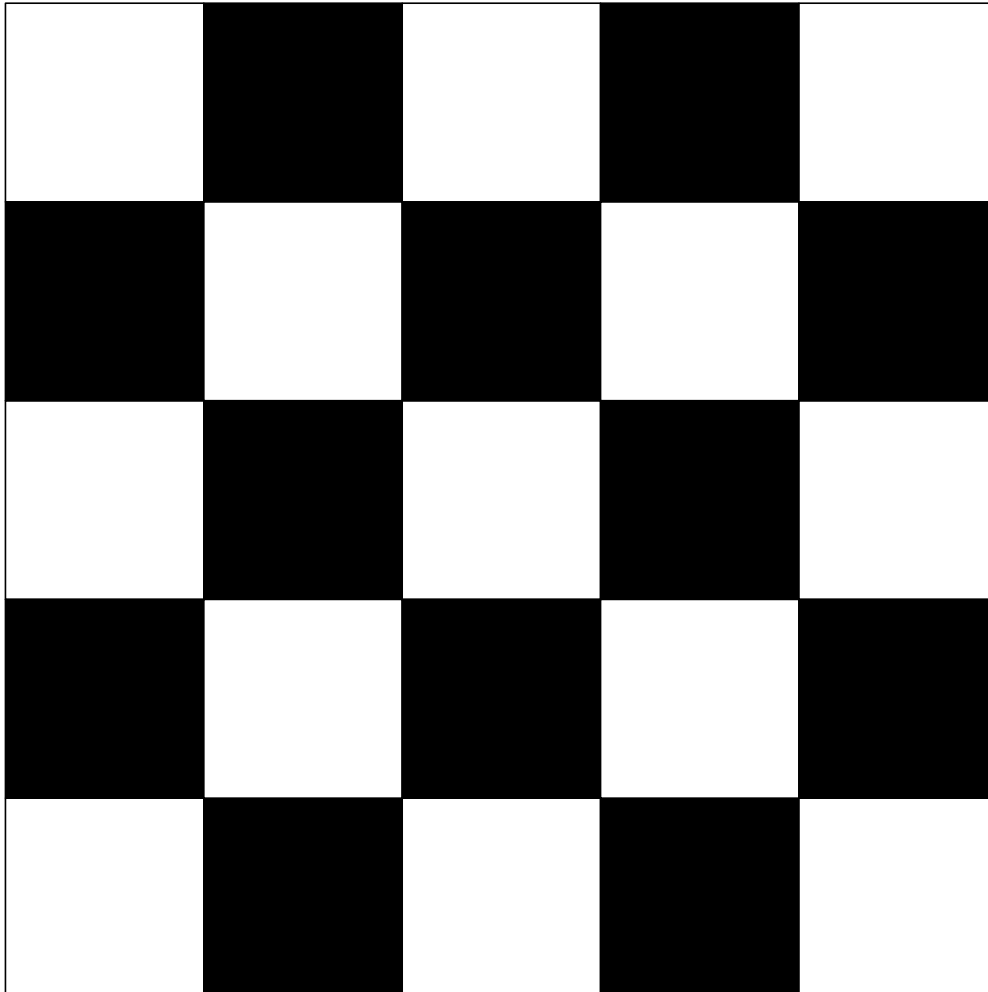


## Anhang E: Kalibrierung- Schema



Ablauf der Kalibrierung: Es wird ein Bild aufgenommen, danach wird im Bild nach der Position eines Hell-Dunkel-Übergangs (oder umgekehrt) gesucht. Laufweite  $i$  und Startposition können eingestellt werden. Kalibrierung wird in einer Datei ausgegeben.

## Anhang F: Kalibrierung- Schablone



Kalibrierschablone,  
Version 0.1

Verwendete Kalibrierschablone. Eignet sich auch um ein größeres Cluster nach demselben Schema zu kalibrieren.

## Anhang G: Vollständige Optikberechnungen

Berechnung der Brennweiten					
Idx	Brennweite [mm]	Abstand [mm]	Objekthöhe [mm]	Objektbreite [mm]	Sensor
1	8	300	180	140	1/3"
2	8	150	150	110	1/3"
3	16	500	150	110	1/3"
4	16	400	120	90	1/3"
5	8,4	350	200	150	1/3"
6	12	300	120	90	1/3"
7	8	400	240	180	1/3"
8	12	500	200	150	1/3"
9	16	450	130	100	1/3"
10	12	1000	400	300	1/3"

$$f = (b * D) / B$$

b= Sensorbreite

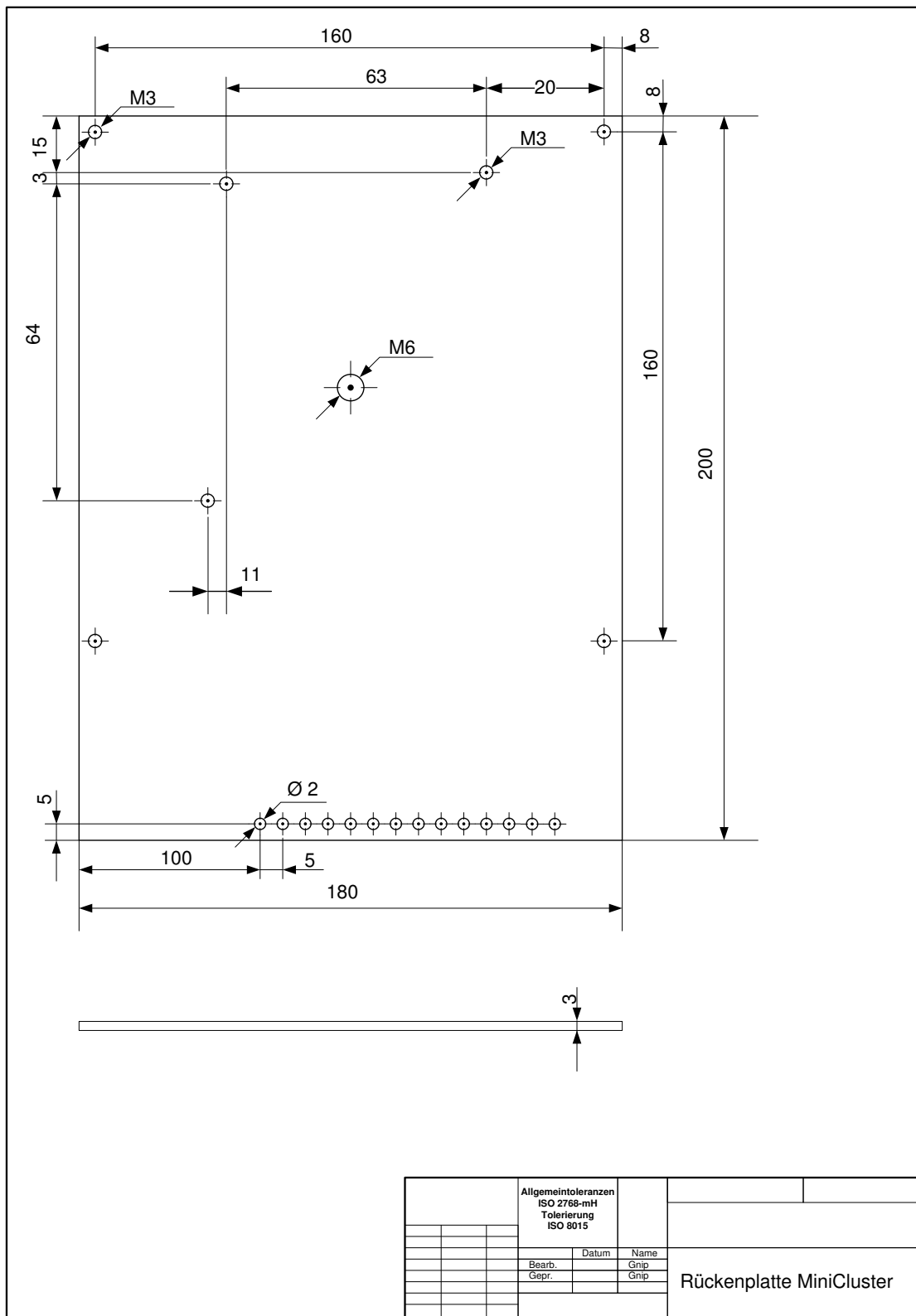
D= Objektentfernung

B= Objekthöhe

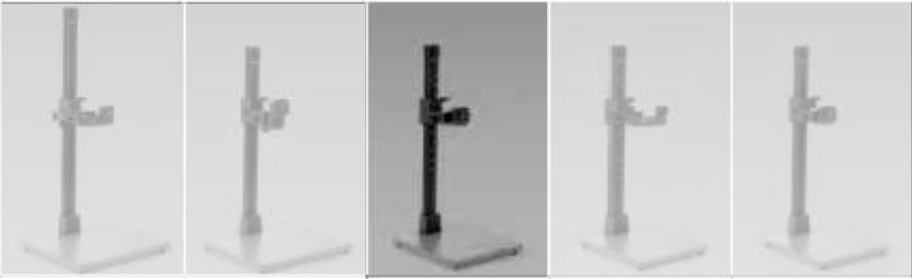














































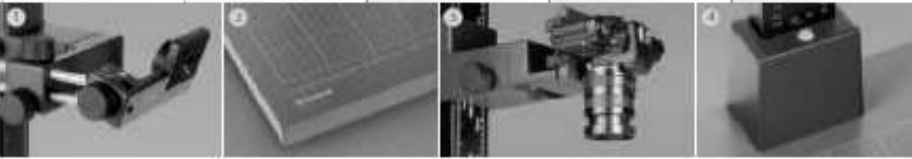
f= Brennweite

Idx	Brennweite [mm]	Abstand [mm]	Objekthöhe [mm]	Objektbreite [mm]	Sensor
1	3,6	100	80	80	2,88
2	7,2	200	80	80	2,88
3	9	250	80	80	2,88
4	10,8	300	80	80	2,88
5	12,6	350	80	80	2,88
6	14,4	400	80	80	2,88
7	15,12	420	80	80	2,88
8	15,84	440	80	80	2,88
9	16,02	445	80	80	2,88
10	16,2	450	80	80	2,88
11	16,128	560	100	80	2,88

# Anhang H: Rückenplatte



## Anhang I: Kamerastativ KAISER RS 1

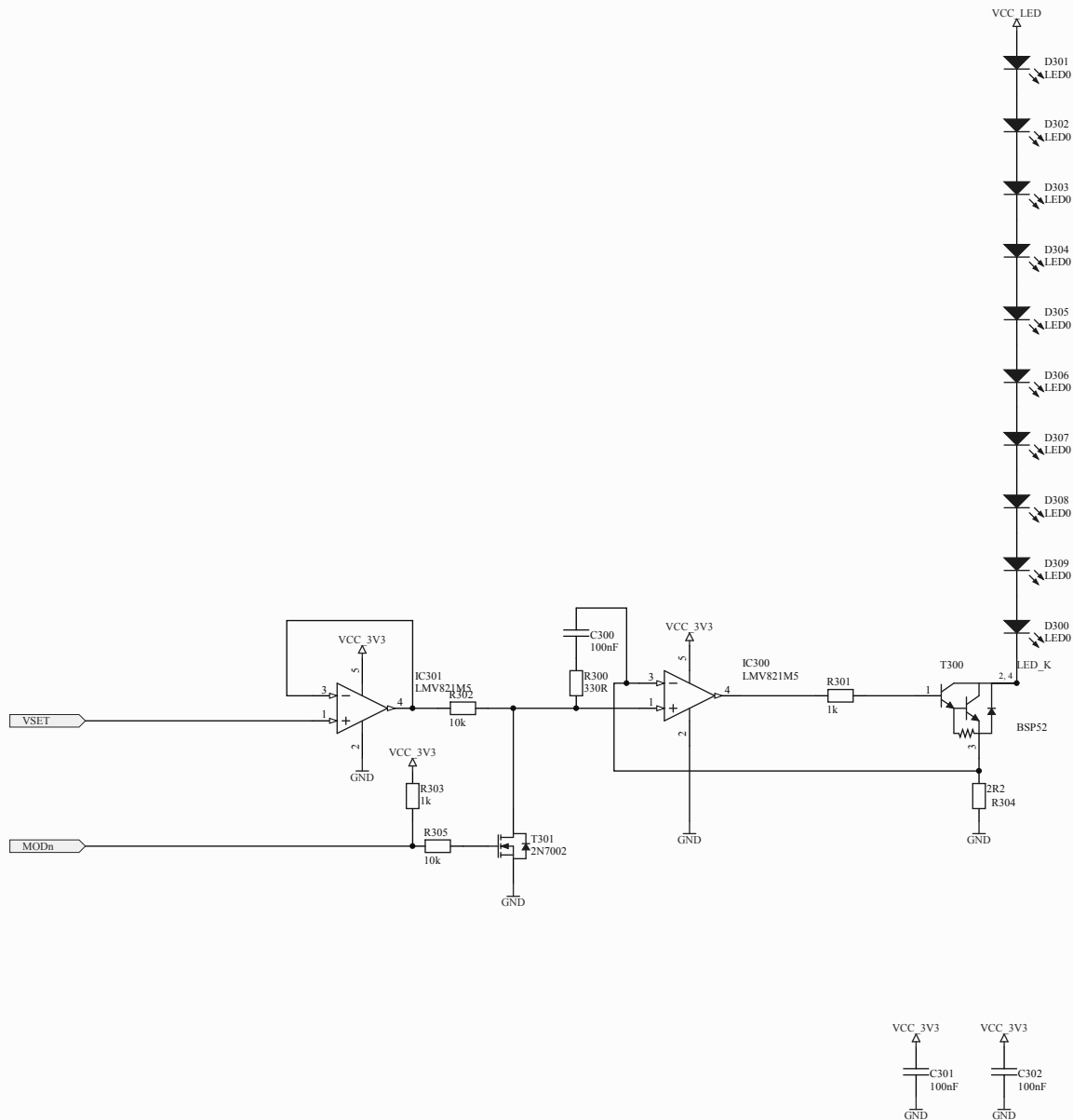
<b>Kaiser R1-System</b>  <b>Stative</b>					
	RSX	RS 10	RS 1	RS 1	RS 1 microdrive
<b>Artikelnummer</b>	5512	5513	5510	5511	5507
<b>STATIV</b>	Bestehend aus Grundbrett mit Säule und Kameraarm.				
<b>Grundbrett</b>	Mattgrau beschichtet, aufgedrucktes Feinraster, cm und inch-Skala  .				
 <b>Außenmaße</b>	60 x 50 cm	60 x 50 cm	45 x 50 cm	45 x 50 cm	45 x 50 cm
 <b>Stärke</b>	32 mm	32 mm	28 mm	28 mm	28 mm
 <b>Sockel mit Wasserwaage </b>					
 <b>Höhenverstellbare Füße</b>					
<b>Säule</b>	Mattschwarz eloxiertes Aluminium-Profil, 70 mm breit, vibrationsarm, cm- und inch-Skalen, Handkurbel für Höhenverstellung				
 <b>Höhe</b>	120 cm	100 cm	100 cm	100 cm	100 cm
 <b>Höhenverstellung mit Frictiontrieb</b>	-	-			-
 <b>Höhenverstellung mit Zahnstange</b>			-	-	
 <b>Gewichtsausgleich</b>					
 <b>Feintrieb</b>	-	-	-	-	
<b>Kameraarm</b>	Austauschbar				
 <b>Typ (s. Seite 11)</b>	RTX 	RTP	RA 1	RT 1	RA 1
 <b>Verstellbarkeit</b>	horizontal, schwenkbar	horizontal	-	horizontal	-
 <b>Kameragewinde</b>	B $\frac{7}{8}$ / B $\frac{1}{4}$ , wechselbar	B $\frac{7}{8}$ / B $\frac{1}{4}$ , wechselbar	B $\frac{7}{8}$ / B $\frac{1}{4}$ , wechselbar	B $\frac{7}{8}$ / B $\frac{1}{4}$ , wechselbar	B $\frac{7}{8}$ / B $\frac{1}{4}$ , wechselbar
 <b>Positionierhilfe </b>	-			-	
 <b>Blockierschraube</b>			-	-	-
 <b>Kameramontage</b>	innen/außen	außen	außen	innen hoch/quer	außen
	<b>a</b>	125,5 cm	105,5 cm	105 cm	105 cm
	<b>b</b>	18 - 31 cm	15,5 - 23,8	18,5	23 - 31,8
	<b>c</b>	60 x 43 cm	60 x 43 cm	45 x 43 cm	45 x 43 cm
	<b>M</b>	max. 3,5 kg	max. 4 kg	max. 4 kg	max. 3 kg
					

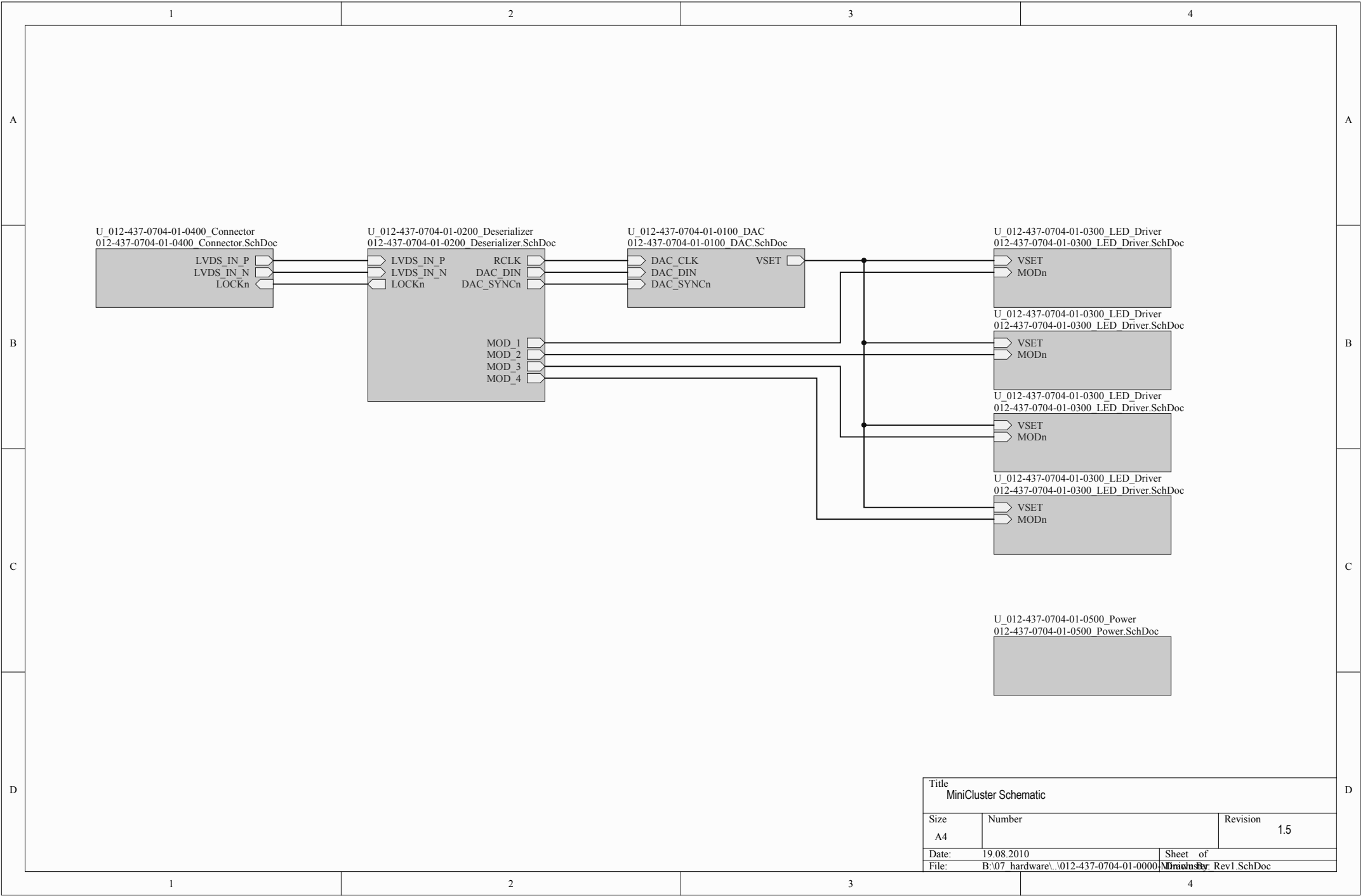
## **Anhang J: Leistungsmerkmale MQ II**

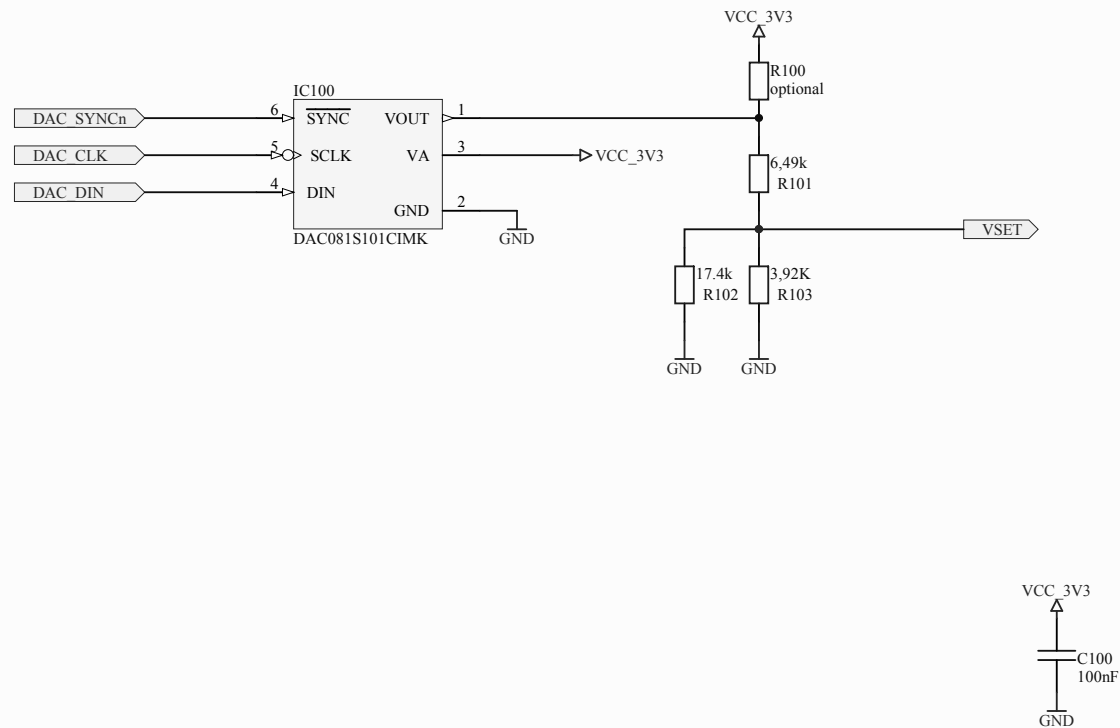
- FPGA, Cyclone III
- 4 Input-Ports für den Anschluss von Kamerasensoren des Typs KK 752
- LED Treiberausgang (RJ 45)
- Integrierte Hardwareblöcke für die Bildanalyse (u. A. Kantenerkennung, Schwellwertfilter etc.)
- VGA Ausgang
- Betriebssystem- µcLinux
- Netzwerkfähigkeit
- NIOS II- Prozessor

**Anhang K: Schaltpläne**

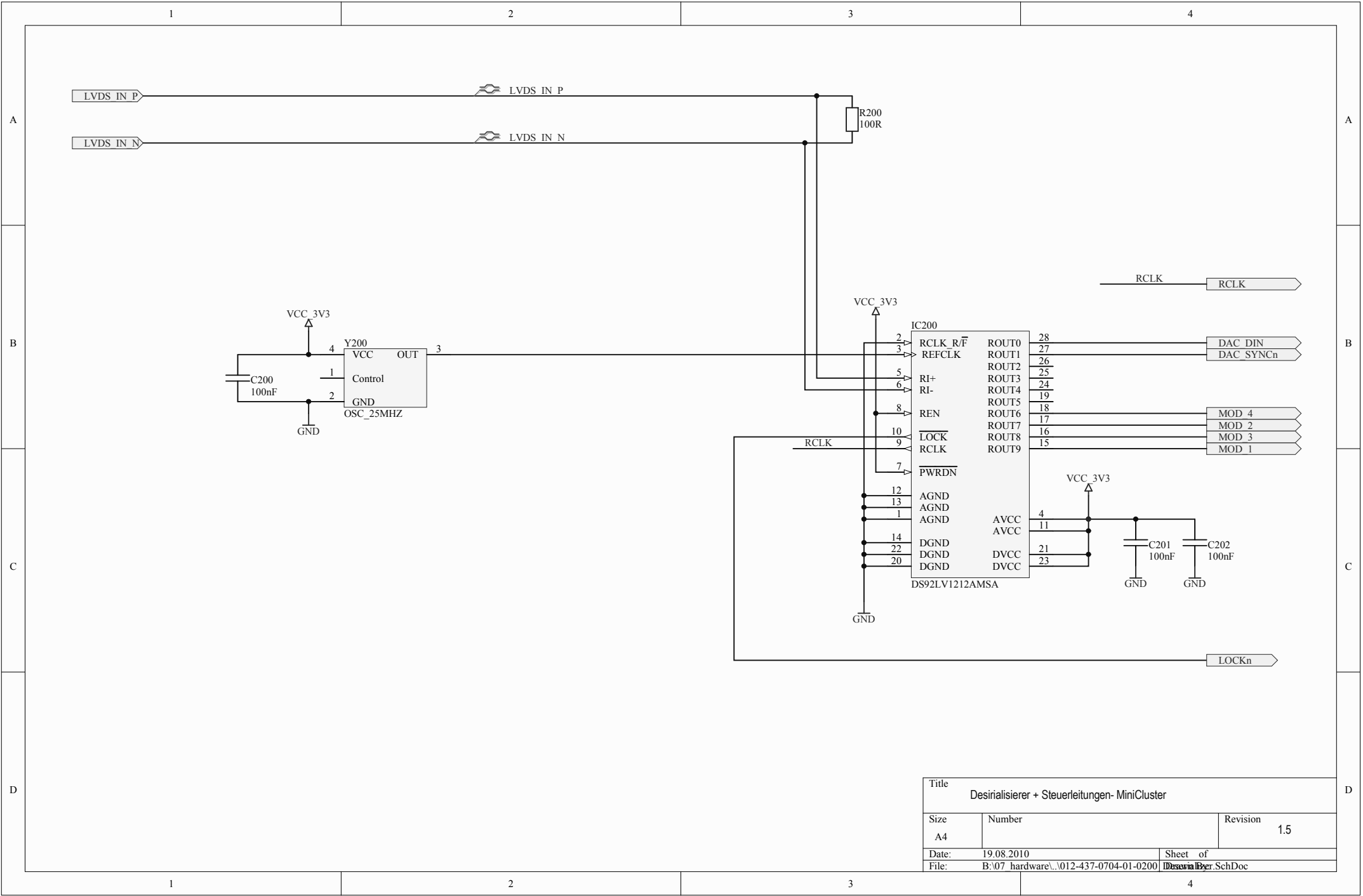


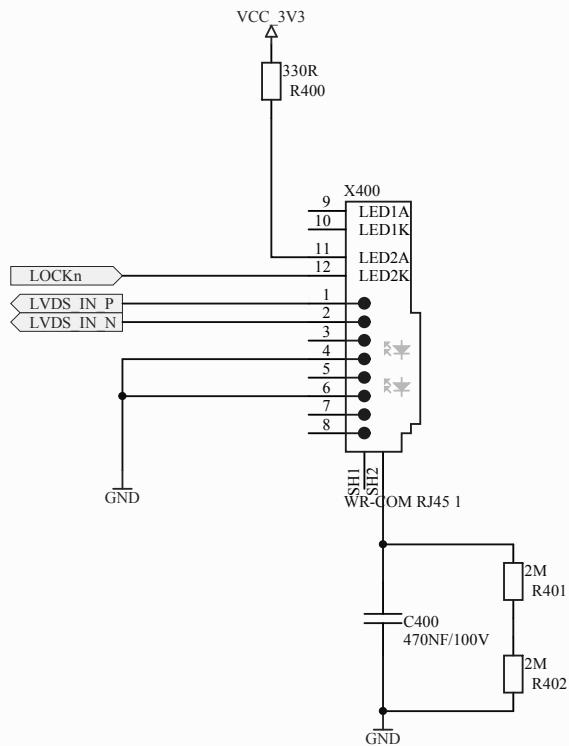




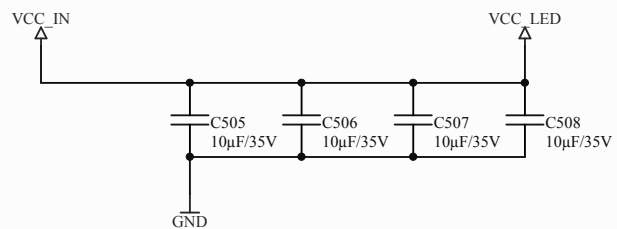
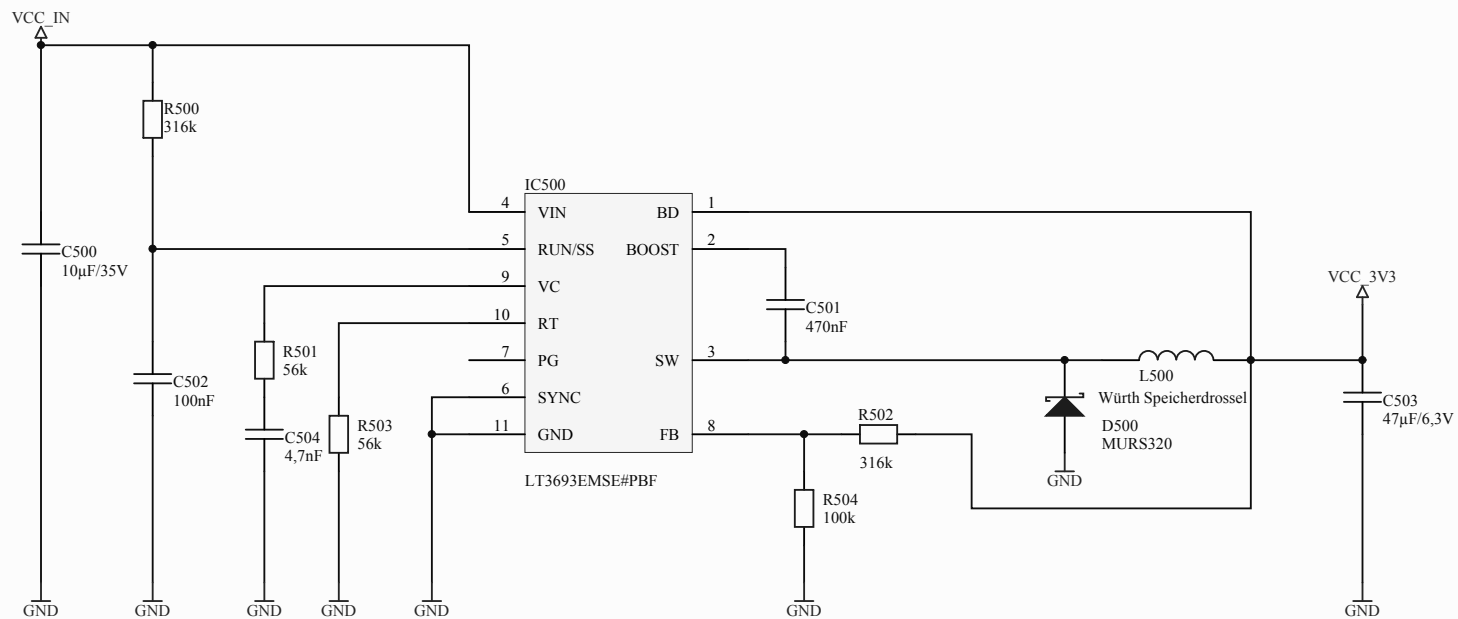


Title DAC- MiniCluster			
Size A4	Number		Revision 1.5
Date:	19.08.2010		Sheet of
File:	B:\07 hardware\...\012-437-0704-01-0100.DWG		Doc

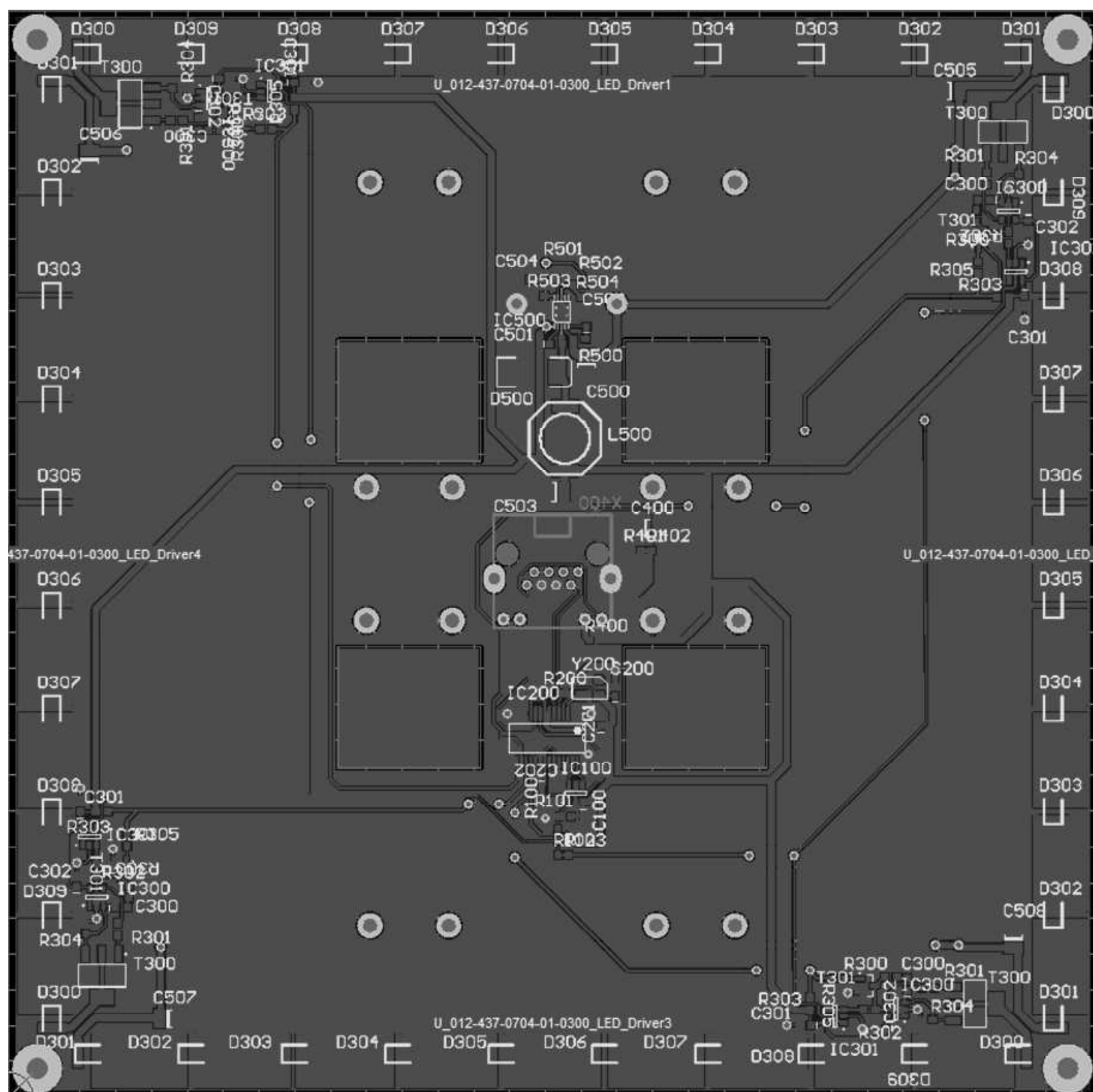




Title Connection 2 MQ_IL- MiniCluster		
Size A4	Number	Revision 1.5
Date:	19.08.2010	Sheet of
File:	B:\07 hardware\...\012-437-0704-01-0400	Drawn By: SchDoc



Title Power Supply- MiniCluster		
Size A4	Number	Revision 1.5
Date:	19.08.2010	Sheet of
File:	B:\07 hardware\...\012-437-0704-01-0500	Drawn By Doc



## 10 CD-Inhalt

Inhaltsverzeichnis des diesem Dokument beiliegenden Datenträgers.

1	Wochenberichte	7-06	Datenblätter
1-01	Vorlage	7-07	MiniQuadrix 2
1-02	Berichte	8	Algorithmus
2	Lastenheft	8-01	Skizzen
3	Pflichtenheft	8-02	Bildsets
4	Recherche	8-03	Ergebnisse
5	Gesrächsprotokolle	9	Software
6	Organisation	9-01	Matlab
7	Hardware	9-02	gnip_3dscan (Projektornder)
7-01	Berechnungen	9-03	Dokumentationen
7-02	Skizzen	10	Projektdokumentation
7-03	Bauteilbibliotheken	10-0	Thesis
7-04	Layouts	10-01	Abbildung
7-05	PCB-Projekt	10-02	Citavl



## Stichwortverzeichnis

Abgrenzung zu anderen Umsetzungen	22	Kalibrierung	37
Abschluss	52	Kamerasensoren	26
Algorithmus	56	KK 752	26
Appendix	56	MiniCluster	28, 57
Ausblick	53	MiniQuadrix II	25
Begriffserläuterungen	56	Montageplatte	37
Bildverarbeitung	25	Offset	60
CD-Inhalt	78	Optik	31
Digitales Filter	58	Präsentation der Daten	21
Echtzeit	59	Reflektion	52
Eingesetzte Software	60	Schwellwert	53
FPGA	57	SFH 4680	28
Hardware	24	Shadowgram	19, 22
Höhenfeld	21	Shape from Shadows	15
Höhenprofils	21	Software	38
Infrarot	59	Szenenbeleuchtung	31
		Trajektorien	31

## Literaturverzeichnis

- [BAL06-14] Ballan, L.: Multimodal 3D Shape Recovery from Texture, Silhouette and Shadow Information, 02.08.2010.
- [BRI10-24] Brinker: MiniQuadrix II API Dokumentation, 2010.
- [DAU98-15] Daum M., D. G.: On 3-D Surface Reconstruction Using Shape from Shadows, 02.08.2010.
- [GNI10-7] Gnip, C.: Pflichtenheft Bachelorarbeit, C. Gnip. 3D-Scan, 2010.
- [GNI10-25] Gnip, C.: Dokumentation 3dscan\_s, c.Gnip 3D-Scan, 2010.
- [HOW10-11] Howah, L.: Lastenheft Bachelor-Arbeit Christoph Gnip. Entwicklung eines 3D-Scanners zur berührungslosen Abtastung geometrischer Patientenkenndaten mittels bildgebender Verfahren. 3D-Scan, 02.08.2010.
- [MIC-20] Micron: Aptina/ Micron- Datasheet. 1/3-Inch Wide-VGA CMOS Digital Image Sensor- MT9V024, o.J.
- [OSR09-18] Osram: Datenblatt, Engwinklige LED im MIDLED-Gehäuse (880 nm). SFH 4680, SFH 4685. Lead (Pb) Free Product - RoHS Compliant.
- [SIL05-21] Silvio Savarese: Shape Reconstruction from Shadows and Reflections, 2005.
- [WIK10-1] Wikipedia: FPGA- Wikipedia. Wiki.  
[http://de.wikipedia.org/w/index.php?title=Field\\_Programmable\\_Gate\\_Array&oldid=77004545](http://de.wikipedia.org/w/index.php?title=Field_Programmable_Gate_Array&oldid=77004545).
- [WIK10-16] Wikipedia: Digitales Filter.  
[http://de.wikipedia.org/w/index.php?title=Digitales\\_Filter&oldid=76719464](http://de.wikipedia.org/w/index.php?title=Digitales_Filter&oldid=76719464).
- [WIK10-2] Wikipedia: Algorithmus Wikipedia. Wiki.  
<http://de.wikipedia.org/w/index.php?title=Algorithmus&oldid=77118908>.
- [WIK10-23] Wikipedia: Altera- Wikipedia.  
<http://de.wikipedia.org/w/index.php?title=Altera&oldid=70277822>.
- [WIK10-3] Wikipedia: Heightmap/ Höhenfeld Wikipedia. Wiki.  
<http://de.wikipedia.org/w/index.php?title=Höhenfeld&oldid=57498257>.
- [WIK10-8] Wikipedia: Echtzeit Wikipedia.  
<http://de.wikipedia.org/w/index.php?title=Echtzeit&oldid=74819208>.
- [YU & T. 04-13] YU, Y.; T. CHANG, J.: Shadow Graphs and 3D Texture Reconstruction, 02.08.2010.