

# Bachelorarbeit

Tobias Hahn

Techniken zur kamerabasierten Erkennung von  
frequenzcodierten Infrarotsendern für  
Indoorpositioningsysteme

Tobias Hahn

Techniken zur kamerabasierten Erkennung  
von frequenzcodierten Infrarotsendern  
für Indoorpositioningsysteme

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. -Ing. Franz Korf  
Zweitgutachter : Prof. Dr. -Ing. Andreas Meisel

Abgegeben am 16. September 2008

**Tobias Hahn**

**Thema der Bachelorarbeit**

Techniken zur kamerabasierten Erkennung von frequenzcodierten Infrarotsendern für Indoorpositioningsysteme.

**Stichworte**

Infrarot, Ambient Assisted Living, Ambient Intelligence, digitale Bildverarbeitung, IndoPosys, Punkterkennung, Position, Cyclic Redundancy Check

**Kurzzusammenfassung**

Gegenstand dieser Bachelorarbeit ist die Entwicklung und Implementierung eines Systems zur Erkennung und Unterscheidung von Infrarotsendern mittels einer Schwarz/Weiß Kamera. Das Infrarotsignal der tragbaren Sender muss über verschiedene Bildverarbeitungsschritte von der restlichen Umgebung klar hervorgehoben werden, um eine eindeutige Erkennung zu schaffen. Mit Hilfe von Blinkcodes wird eine Unterscheidung mehrerer Sender getroffen. In dieser Arbeit werden die Anforderungen und Techniken analysiert und daraus eine Realisierung entwickelt und umgesetzt. Die entworfenen Komponenten sind eine Teilaufgabe für ein späteres Indoorpositionssystem.

**Tobias Hahn**

**Title of the paper**

Techniques for camera-based detection of frequency-coded infrared transmitter for indoorpositioningsystems.

**Keywords**

Infrared, Ambient Assisted Living, Ambient Intelligence, digital image processing, IndoPosys, blob-detection, position, Cyclic Redundancy Check

**Abstract**

This bachelor's thesis is concerned with the development and implementation of a system to detect and to differ infrared transmitter by using a black/white camera. The infrared-signal of the portable transmitters must be highlighted from the background by using different digital image processing techniques, to create a clear detection. Then with the help of flashing-codes, a differentiation of several transmitters can be met. In this thesis the requirements and technologies are analyzed and out of it a realisation developed and moved. These modules are a subsystem designed for a future indoorpositioningsystem.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	7
<b>Abbildungsverzeichnis</b>	8
<b>1. Einführung</b>	9
1.1. Motivation .....	9
1.2. Ziel der Arbeit .....	10
1.3. Gliederung .....	11
<b>2. Grundlagen</b>	12
2.1. Infrarot.....	12
2.2. Position.....	13
2.2.1. physikalische Position .....	13
2.2.2. symbolische Position.....	13
2.2.3. absolute und relative Position .....	14
2.3. Positionierungssysteme .....	14
2.3.1. GPS.....	15
2.3.2. IMAPS.....	15
2.3.3. Active Badge .....	16
2.4. Bilderverarbeitung.....	17
2.4.1. digitales Bild .....	18
2.4.2. Punktoperation .....	18
2.4.3. lokale Operation .....	18
2.5. IC Imaging Control .....	19
<b>3. Analyse und Design</b>	21
3.1. Anforderungen .....	21
3.1.1. Funktionale Anforderungen .....	22
3.1.2. Nicht funktionale Anforderungen .....	22
3.2. Hardware .....	23
3.2.1. Infrarotsender .....	23
3.2.2. Kamera .....	23
3.2.3. Anschlussarten .....	24
3.2.4. Laborrechner .....	26
3.3. Software .....	26
3.3.1. Bildverarbeitung.....	27
3.3.1.1. Tiefpassfilter.....	27
3.3.1.2. Schwellwertverfahren.....	29
3.3.1.3. Differenzbild .....	29
3.3.2. Punkterkennung.....	29
3.3.3. Szenarien .....	31
3.3.4. Codierung .....	34
3.3.5. Systemarchitektur des Senders.....	36
3.3.6. Fehlererkennung in der Übertragung .....	38
3.4. Zusammenfassung.....	41

---

<b>4. Realisierung</b>	42
4.1. Sender.....	42
4.1.1. Frequenzgenerierung fürs Senden.....	42
4.1.2. CRC.....	43
4.1.3. Senden der Identifikation .....	43
4.2. Empfangssystem.....	45
4.2.1. Randbedingungen.....	46
4.2.2. Bildbearbeitung der Kamerabilder.....	47
4.2.3. Blob-Detector .....	49
4.3. Zusammenfassung.....	51
<b>5. Qualitätssicherung</b>	52
5.1. Bewegungen .....	52
5.2. Spiegelungen .....	53
5.3. Zeitmessungen.....	54
<b>6. Fazit</b>	55
6.1. Bewertung der Ergebnisse.....	55
6.2. Verbesserungsvorschläge und mögliche Änderungen .....	56
<b>Literaturverzeichnis</b>	57
<b>A. Spezifikation der Infrarot Diode</b>	60
<b>B. Inhalt der CD</b>	61

# Tabellenverzeichnis

3.1: Spezifikation DMK 21F04 .....	24
3.2: Vergleich von USB und FireWire.....	25
3.3: Vergleich der Szenarien .....	34
3.4: Vergleich der Architekturen für den Scheduler .....	38
3.5: zu sendende ID mit gerader und ungerader Parität .....	39
3.6: Vergleich der Fehlererkennungsmethoden .....	41
4.1: Symbole und deren Codierung .....	43
5.1: Daten für die Geschwindigkeitsberechnung .....	52
5.2: Zeitmessung der Funktionen .....	54
A.1.: Spezifikation Osram SFH48P .....	60

# Abbildungsverzeichnis

1.1: Überblick des groben Aufbaus des Systems - links eine Person mit Infrarotsender und rechts die Kamera und PC für Berechnungen .....	11
2.1: Wellenlängenbereiche der Strahlung [Perret 2008] .....	14
2.2: IMAPS: Empfangsmodul .....	18
2.3: Active Badges : - unten links die erste Generation - oben rechts die vierte Generation .....	19
2.4: 4er Nachbarschaft.....	21
2.5: 8er Nachbarschaft.....	21
3.1: Evolution-Board des Labors für Technische-Informatik der HAW.....	25
3.2: links verrauschtes Bild, rechts nach einem Tiefpassfilter [Wittman 2008] .....	29
3.3: Rechteckfilter .....	30
3.4: zwei Binominalfilter erstellt mit dem Pascalschen Dreiecks .....	30
3.5: unten das Differenzbild der beiden Aufnahmen [Jiang 2008] .....	32
3.6: Bewegung des Senders über einer Folge von 2 Bildern .....	35
3.7: Bitfehler bei unterschiedlichen Frequenzen.....	37
3.8: Berechnung der Checksumme vor dem Senden.....	42
3.9: Berechnung der Checksumme nach dem Empfangen.....	42
4.1: links das Bild, wie wir es sehen und rechts wie es im Puffer liegt .....	47
4.2: UML Klassendiagramm grobe Übersicht .....	48
4.3: Original Bild vor der Bearbeitung.....	50
4.4: Bild nach dem Schwellwertverfahren .....	50
4.5: Maske zum auffinden von Punkten(Blobs) Grün der min Bereich des Punktes .....	52



# 1. Einführung

## 1.1. Motivation

Heutzutage ist das Leben ohne technische Hilfsmittel kaum noch vorstellbar. Überall unterstützen uns elektronische Helferchen, um uns den Alltag zu erleichtern. In der Küche hat z.B. das Rührgerät den alten Schneebesen ersetzt. Und die Zähne werden mit einer elektrischen Zahnbürste, anstatt mit der Handbürste geputzt. Hörverluste bei älteren Menschen werden durch kleine kaum sichtbare Hörgeräte größtenteils ausgeglichen. Auch die Sicherheit des Menschen wird dadurch verbessert. So alarmieren z.B. Brandmelder die Bewohner bei Rauch- und Feuerentwicklung.

In den letzten Jahren stieg das Interesse der Forschung an tragbarer Elektronik (wearable computing). Ambient Intelligence (intelligente Umgebung) ist eins der Forschungsgebiete in diesem Bereich[TUK 2008]. Basis von Ambient Intelligence ist ein Netzwerk nahezu unsichtbarer, miteinander vernetzter rechnender Einheiten, welche über verschiedene Sensoren Informationen sammeln, diese verarbeiten und über Aktoren auf die Umgebung und damit auch auf den Menschen einwirken. So wird z.B. über Sensoren das Training von Sportlern überprüft und mit den Ergebnissen dieses Training weiter optimiert[TUK 2008].

Ein weiteres großes Forschungsgebiet in diesem Bereich ist Ambient Assisted Living (umgebungsunterstütztes Leben). In diesem Forschungsfeld geht es hauptsächlich um ältere und behinderte Menschen. Darunter werden Konzepte, Dienstleistungen und Produkte verstanden, die ein soziales Umfeld und neue Technologien verbinden. Älteren Menschen sollen diese Ergebnisse ein langes und selbst bestimmtes Leben in den eigenen vier Wänden ermöglichen[BBF 2008].

Ambient Assisted Living ist ein Innovationsfeld im Rahmenprogramm "Mikrosysteme" (2004 – 2009) des Bundesministeriums für Bildung und Forschung[BBF 2008]. Hierbei wird es aktiv von der VDI/VDE Innovation + Technik GmbH unterstützt. Auch für Europa ist Ambient Assisted Living ein neues Förderungsprogramm. 16 Länder nehmen zurzeit daran teil und gestalten dies in allen Aspekten selbst. Die Länder haben sich dazu verpflichtet, über die Laufzeit des Programms, verbindlich nationale Fördermittel bereit zu stellen.

In Bezug auf diese Forschungsgebiete entstand an der HAW Hamburg das Projekt „Indoor Positionierungssysteme (IndoPosys)“. Ziel dieses Projekts ist die Entwicklung eines robusten, leicht kalibrier- und konfigurierbaren Prototyps eines Indoor Positionierungssystems. Die Positionierung soll über tragbare Infrarotsender und fest installierten Kameras erfolgen. Die Kameras sind mit einem Rechner verbunden, der die Bildverarbeitung und Positionen errechnet. In Abbildung 1.1 ist ein grober Aufbau so eines Systems zu sehen.

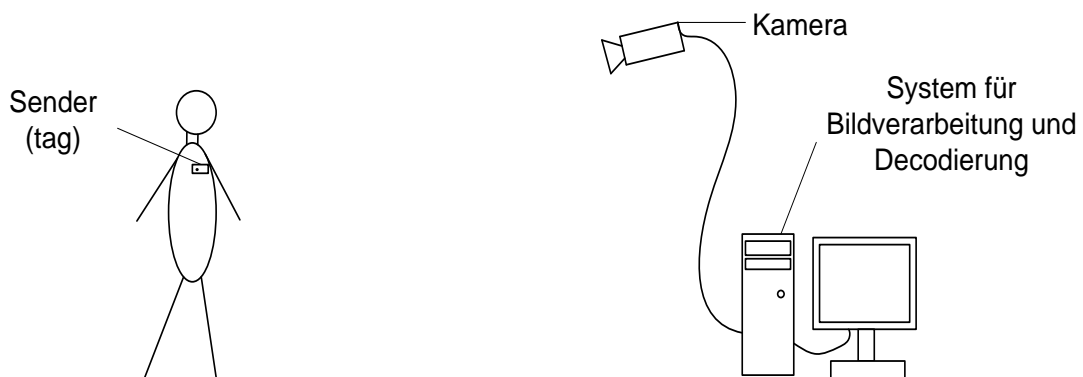


Abbildung 1.1: Überblick des groben Aufbaus des Systems - links eine Person mit Infrarotsender und rechts die Kamera und PC für Berechnungen

## 1.2. Ziel der Arbeit

Diese Bachelorarbeit befasst sich mit einer ersten Teilaufgabe des IndoPosys. Ziel dieses Teilsystem ist es, die in einem Kamerastream befindlichen und sich bewegende Infrarotsender zu erkennen und zu unterscheiden. Zusätzlich soll diese Arbeit einen Überblick und Hilfestellung für nachfolgende Arbeiten für das IndoPosys-Projekt geben.

## 1.3. Gliederung

Die Bachelorarbeit gliedert sich in folgende Kapitel

- **Kapitel 2 :**

Ein Überblick über verschiedene Begriffe und verwandte Themen und Systemen wird hier gegeben, um das Verständnis im Verlauf der Arbeit zu erleichtern.

- **Kapitel 3 :**

Zunächst werden die genaueren Anforderungen an das Teilsystem aufgezeigt. Danach werden Techniken beschrieben und ausgewählt, die zu einer Lösung dieser Anforderungen führen.

- **Kapitel 4 :**

Dieser Teil der Arbeit besteht aus der Beschreibung der Realisierung des Teilsystems. Hier wird die Umsetzung der zuvor beschriebenen Techniken aufgezeigt. Es wird auf Randbedingungen des bestehenden Versuchsaufbaus eingegangen.

- **Kapitel 5 :**

Qualitätssicherung der Entwicklung.  
Hier wird der prinzipielle Arbeitsablauf getestet.

- **Kapitel 6 :**

Zum Schluss wird ein Fazit gegeben und die Ergebnisse beurteilt.

## 2. Grundlagen

In diesem Kapitel wird ein Einblick über Techniken und Technologien gegeben, die in dieser Arbeit zum Einsatz kommen. Vergleichbare und ähnliche Positionssysteme und Arbeiten werden hier vorgestellt und deren Funktionsweise erklärt. Dies soll das spätere Verständnis meiner Arbeit erleichtern.

### 2.1. Infrarot

Zur Erkennung der Position und Übertragung von Daten in dem IndoPosys kommt infrarotes Licht zum Einsatz. Dieses Licht ist für das menschliche Auge nicht sichtbar und somit findet die Übertragung für den Menschen unbemerkt statt. Infrarotes Licht liegt in dem Wellenbereich von 780 nm bis hin zu 1 mm. Das sichtbare Licht liegt in dem Bereich von 380 nm bis 780 nm, wie in Abbildung 2.1 zu sehen. [BFS 2008]

Die eingesetzte Kamera, näher beschrieben in Abschnitt 3.2.2., kann jedoch dieses Licht aufnehmen und stellt es als eine sehr helle Lichtquelle dar.

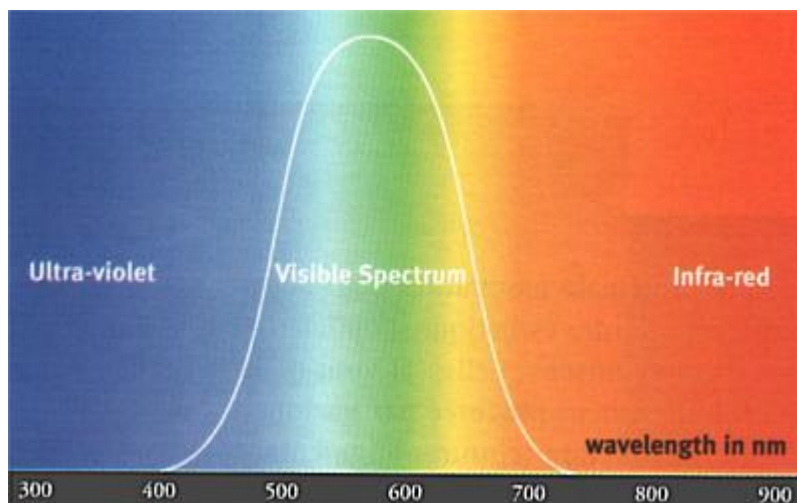


Abbildung 2.1: Wellenlängenbereiche der Strahlung [Perret 2008]

## 2.2. Position

In einem Positionierungssystem dreht sich alles um die Position eines Objektes. Egal ob es ein System für den Außenbereich oder in Gebäuden genutzt wird. Dabei kann es verschiedene Anforderungen, wie z.B. die Genauigkeit und die Art der Beschreibung, an die zu bestimmende Position geben. Denn es gibt verschiedene Arten von Positionen, die in dem folgenden Abschnitt beschrieben werden.

### 2.2.1. Physikalische Position

Die physikalische Position verwendet ein Koordinatensystem. Um die Position von einer Stadt in Deutschland anzugeben, kann z.B. das Gradnetz der Erde benutzt werden. Die Position wird in Längen und Breitengraden angegeben.

[Borriello 2001]

Beispiel :

physikalische Position der Stadt Hamburg im Gradnetz der Erde :

10° 00' Ost

53° 33' Nord

### 2.2.2. Symbolische Position

Bei der symbolischen Position handelt es sich nicht um eine Angabe mittels eines Koordinatensystems, sondern es wird viel mehr der Ort beschrieben wo sich das Objekt befindet [Borriello 2001]. Die Angaben können z.B. Räume, Straßen oder auch Städtenamen sein. Diese Form der Angabe ist oftmals zwar unpräzise, aber für den Menschen besser verständlich. Diese Angabe ist wohl die meist benutzte im täglichen Leben. Symbolische Positionen können auch oft hierarchisch sein.

Beispiel : Hamburg → Rathaus → Raum 12

### 2.2.3. Absolute und relative Position

Ist eine Positionsangabe absolut, so wurde sie mit Hilfe eines Referenzrasters ermittelt, die die Basis für alle zu lokalisierenden Objekte zugrunde liegt. Objekte die in diesem Raster, die sich an der gleichen Stelle in einem Raum befinden, liefern/besitzen auch die gleichen Positionswerte [Borriello 2001].

Bei einer relativen Positionsangabe wird die Position des Objektes mit Hilfe eines eigenen relativen Bezugspunktes beschrieben [Borriello 2001]. Stehen z.B. 3 Personen und ein Stuhl in einem Raum, so hat der Stuhl für jede Person im Raum eine andere Position, wenn die Position relativ zu sich selber angesehen wird.

Eine absolute Position kann in eine relative Position überführt werden. Dies wird möglich wenn eine zweite absolute Position verfügbar ist [Borriello 2001]. Ist also die absolute Position des Stuhls und dazu noch die absolute Position einer Person im Raum bekannt, so kann die Position des Stuhls relativ zu dieser Person angegeben werden.

Relative Positionen können nur in eine absolute Position überführt werden, wenn die absoluten Positionswerte der Referenzpunkte der relativen Positionsangaben bekannt sind. Ist die absolute Position einer Person im Raum bekannt und dazu noch die relative Position des Stuhls, kann nun die absolute Position des Stuhls ermittelt werden.

## 2.3. Positionierungssysteme

Da diese Arbeit in einem Zusammenhang zu einem Positionierungssystem steht, werden nun im folgenden Abschnitt diese Systeme zunächst grundlegend beschrieben. Im Laufe des Abschnitts werden drei Positionierungssysteme vorgestellt. Das erste, GPS, ist ein sehr bekanntes Positionierungssystem für den Außenbereich. Folgend wird ein System vorgestellt, das hier an der HAW Hamburg entwickelt wurde. Das letzte hier vorgestellte System bedient sich, wie in dieser Arbeit, der Infrarottechnik.

Positionierungssysteme sind technische Geräte, mit deren Hilfe eine Ortsbestimmung durchgeführt werden kann. Es wird durch Erfassung und Messung von Umgebungsparametern über einen Algorithmus die Position innerhalb eines definierten Koordinatensystems ermittelt. Die Messung bzw. Erfassung erfolgt mit Sensoren für

elektromagnetische Wellen, insbesondere Licht, Schallwellen oder anderen ubiquitären Medien. Die Messwerte liefern Entfernung und/oder Richtung einer bekannten Marke, so dass mit mehreren Messungen und mathematischen Berechnungen die Position relativ zu den Marken errechnet werden kann. Die errechnete Position wird in Koordinaten angegeben, wobei es unterschiedliche Koordinatensysteme gibt. Im folgenden Kapitel werden ein paar bereits bestehende Positionierungssysteme vorgestellt (und deren Techniken aufgezeigt). [Borriello 2001]

### **2.3.1. GPS**

Das bekannteste und meist benutzte Positionierungssystem ist wohl das Global Positioning System (GPS). 29 geostationäre Satelliten sind für dieses System im Einsatz. Diese Satelliten senden ständig ein synchronisiertes Signal aus. Die Daten in dem Signal bestehen aus der Position, der Kennung und aus dem Sendezeitpunkt des Satelliten. Der Empfänger kann mittels drei verschiedener Satelliten seine Position berechnen. Dies geschieht, indem die Laufzeit der gesendeten Signale ermittelt wird. Aus dieser Laufzeit wird die Entfernung zum Sender errechnet und aus den drei Entfernungen seine Position. Meistens wird ein vierter Satellit für die Zeit benötigt, da der Empfänger oft keine Uhr besitzt die genau genug ist. Die Position des Empfängers ist nur dem selbigen bekannt, womit Privatsphäre besteht.

Durch die geringe Sendeleistung ist die Positionierung in Gebäuden noch nicht möglich. Zusätzlich verhindert die Genauigkeit im Meterbereich das einsetzen der Technologie in Gebäuden unbrauchbar.

[Bauer 2003]

### **2.3.2. IMAPS**

Dieses System wurde 2006 von Sebastian Gregor an der HAW entwickelt und bedient sich einer anderen Technik als die in dieser Arbeit. In diesem System werden Funk und Ultraschallsignale von an der Decke befindlichen Modulen gesendet. Diese Signale werden zur gleichen Zeit losgeschickt. Die Entfernung des Empfängers (Abbildung 2.2) zu den Sendern, errechnet sich der Empfänger aus dem zeitlichen Unterschied der

eintreffenden Signale. Um die physikalische Position des Empfängers, mittels Multilateration zu berechnen, werden 4 Entfernungen zu verschiedenen Sendern benötigt. Die Positionsangaben in IMAPS sind sowohl physikalisch als auch symbolisch.

Bei diesem System wurde auf die Wahrung der Privatsphäre geachtet und die Positionsbestimmung dem Empfänger überlassen. Hohe Skalierbarkeit und Langlebigkeit des Systems waren weitere Ziele dieser Arbeit. Hinzu sollte es einen geringen Installationsaufwand bieten. [Gregor 2006]

Nachteil dieses Systems sind Reflektionen der Schallwellen von Wänden und Gegenständen und der damit verbundenen Verfälschung der Messergebnisse.

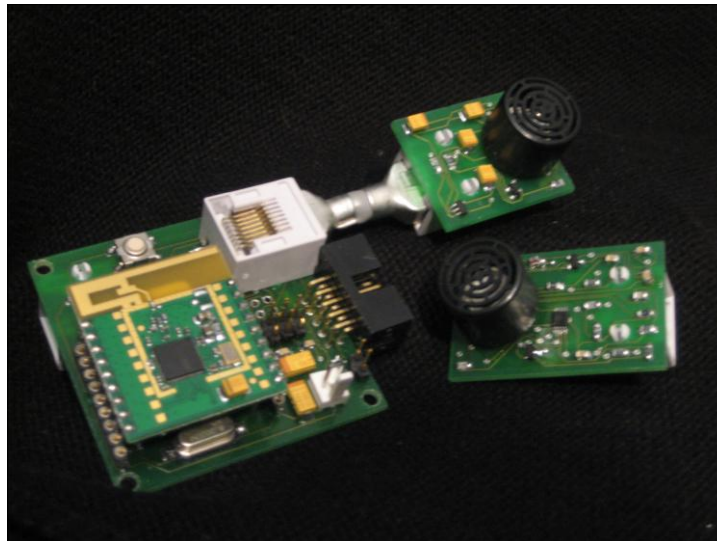


Abbildung 2.2: IMAPS: Empfangsmodul

### 2.3.3. Active Badge

Das Active Badge ist das erste System, welches für eine Positionsbestimmung in Räumen entwickelt wurde. Dieses System wurde 1992 in den Olivetti Labs in Cambridge entwickelt. Ähnlich wie in dieser Bachelorarbeit, werden bei dem Active Badge System Infrarotsender zur Positionierung verwendet.

Die Gegenstellen bilden aber in diesem System Infrarotempfänger, die in den Räumen installiert sind. Jede Person oder Objekt, welche zu lokalisieren sind, trägt solch einen



Infrarotsender (Abbildung 2.3), die sogenannten „Badges“. Solch ein Badge sendet in einem Intervall, z.B. 10s, seine eindeutige Identifikation. Diese wird von dem im Raum installierten Empfänger empfangen und zu einem zentralen Server gesendet. Dieser speichert die absolute symbolische Position aller Sender in einer Datenbank.

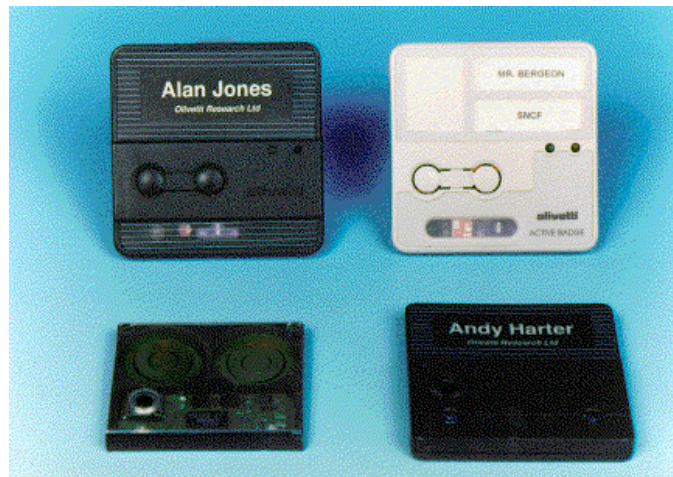


Abbildung 2.3: Active Badges : - unten links die erste Generation  
- oben rechts die vierte Generation

## 2.4. Bilderverarbeitung

Durch das Auge können wir viele Eigenschaften von Objekten, wie z.B. die Größe, Farbe und Bewegung erkennen. Dies geschieht aber viel mehr über die Erfahrung von Bildern und Situationen, als durch das Sehen selbst. Die Kombination aus Auge und Gehirn ist sehr leistungsstark. Über 10 Millionen verschiedene Farben können die Augen erkennen.

Die Technologie ist heutzutage so weit fortgeschritten, dass Digitalkameras schon manches mehr können als das Auge. In unserem Fall bezieht sich das auf die Erkennung von infrarotem Licht. Doch die Schwierigkeit liegt darin, das Gehirn nachzustellen, welche z.B. Objekte und Entfernungen erkennt. Bei der Bildverarbeitung und Objekterkennung muss genau analysiert werden, was erkannt werden soll. Mit diesen Vorinformationen werden die konkreten Fälle in die Software umgesetzt und somit das kleine Gehirn hinter der Kamera geschaffen.

In den nachfolgenden Abschnitten wird beschrieben, wie ein digitales Bild aufgebaut ist und wie man es manipulieren und analysieren kann.

### **2.4.1. Das digitale Bild**

Da Computer keine kontinuierlichen Bilder verarbeiten können, müssen diese Bilder als zweidimensionale Punktfelder abgespeichert werden. Solch ein Punkt wird als Pixel (picture element) bezeichnet. Pixel repräsentieren die Belichtungsstärke, gemessen in Grauwerten, an der zugehörigen Gitterposition. Die übliche Angabe der Position des Pixels ist die Matrixnotation. Der Zeilenindex wird mit  $m$  und der Spaltenindex mit  $n$  beschrieben. Daraus folgt die  $M \times N$  Matrix. [Jähne 2002]

### **2.4.2. Punktoperation**

Punktoperationen verändern die Grauwerte einzelner Bildpunkte nur in Abhängigkeit vom Grauwert selbst und eventuell von der Position des Pixels. Spielt die Position keine Rolle, so wird von einer homogenen Punktoperation gesprochen, andernfalls sind es inhomogene Operationen. Mehrere verschiedene Grauwerte können auf einen Grauwert abgebildet, dies ist unumkehrbar und deshalb geht dadurch ein Teil des Bildinhaltes unwiederbringlich verloren. [Jähne 2002]

### **2.4.3. Lokale Operation**

Lokale Operationen manipulieren den Grauwert in Abhängigkeit einer Nachbarschaft oder einer örtlich begrenzten Region um einen Pixel. Nachbarschaftsbeziehungen sind bedeutende Eigenschaften von diskreten Bildern. Sie definieren, was wir als zusammenhängende Region und damit als digitales Objekt betrachten. Bei einem zweidimensionalen quadratischen Gitter gibt es zwei Möglichkeiten benachbarte Bildpunkte zu definieren. Pixel können benachbart betrachtet werden, wenn sie mindestens eine gemeinsame Ecke oder eine gemeinsame Kante aufweisen können.

Daraus folgt, dass es eine 4er-Nachbarschaft (Abbildung 2.4) oder eine 8er-Nachbarschaft (Abbildung 2.5) gibt. [Jähne 2002]

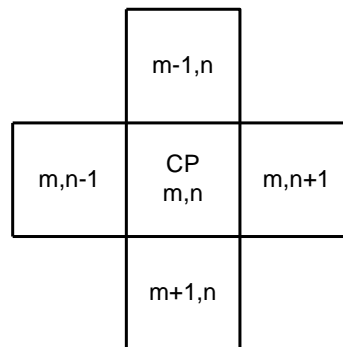


Abbildung 2.4: 4er Nachbarschaft

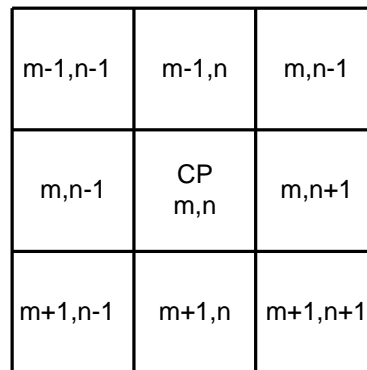


Abbildung 2.5: 8er Nachbarschaft

CP = betrachteter Pixel

## 2.5. IC Imaging Control

IC Imaging Control ist eine von der Firma The Imaging Source Europe GmbH zur Verfügung stehende API für Visual Studio .Net, die in dieser Arbeit zur Entwicklung eingesetzt wird [Image 2008]. Sie wird mit der später beschriebenen Kamera (Kap. 3.2.2.) geliefert. Aber auch das Arbeiten mit Kameras anderer Hersteller ist möglich. Diese API kapselt die Aspekte von DirectShow, eine Programmierschnittstelle für multimedia Anwendungen der Windows-Plattform, die zur Verwaltung, Kontrolle und

zum Zugang von Bild-Geräten und Daten nötig sind, um dem Anwender eine Hardwareunabhängige Programmierschnittstelle zu gewährleisten.

IC Imaging Control stellt eine Klasse Grabber bereit, der wie ein konventioneller Framegrabber, die Aufgaben der Steuerung der Eingänge, Video-Formate und den eigentlichen Grabbing-Prozess übernimmt. Zusätzlich wird noch ein Grabber-Listener, der bei fertiger Übertragung des Bildes aufgerufen wird, benötigt. Ist dies geschehen, liegt das digitale Bild zur Weiterverarbeitung bereit. Somit erleichtert diese API die Entwicklung der Software für das Empfangssystem und der Entwickler kann sich auf die eigentliche Bildverarbeitung und die anderen Komponenten konzentrieren.

## **3. Analyse und Design**

In diesem Kapitel werden zunächst die Anforderungen beschrieben, die zur Umsetzung der Ziele aus Kapitel 1.2. notwendig sind. Nachfolgend wird die Hardware aufgezeigt, die für diese Arbeit von der HAW Hamburg zur Verfügung steht. Verschiedene Techniken zur späteren Realisierung (Kap. 4.) von dem zu sendenden Systems, wie auch des zu empfangenen Systems werden hier aufgeführt und erläutert. Dazu wird auf 2 Szenarien eingegangen, die eine mögliche Erfüllung der Ziele aufzeigt. Abschließend wird auf die Systemarchitektur des Senders und einer Fehlererkennung eingegangen.

### **3.1. Anforderungen**

Ziel dieser Arbeit ist es, ein System zu entwickeln, dass mittels einer Schwarzweißkamera mehrere Personen und Objekte gleichzeitig finden und unterscheiden soll. Dies soll mittels eines kleinen tags(engl.), die diese Personen und Objekte an sich tragen, geschehen.

#### **Infrarotsender**

Der Sender soll später z.B. auf einem Brustausweis oder einem Headset integriert werden. Daher wird er aus einem kleinen Mikrocontroller bestehen und somit wenig Ressourcen zu Verfügung stellen können. Die Software für den Sender muss aus diesem Grund klein ausfallen.

## **Kamerasystem**

Das Kamerasystem hat die Aufgabe, in den aufgenommen Bildern, die Sender aus dem Hintergrund heraus zu filtern. Über eine Folge mehrerer Bilder müssen die Sender mit ihrer Codierung unterschieden werden. Bei der Übertragung dieser Codes kann es natürlich auch zu Fehlern kommen. Hierfür muss eine Methode zur Verifikation dieser Codes gefunden werden. Die Daten, wie z.B. die Position im Bild und deren Kennung, sollen für das spätere Positionssystem zur Verfügung gestellt werden.

Bei einer Kamera, die ca. 30 Bilder pro Sekunde aufnimmt, liegt die maximale Zeit für Berechnungen bei ca. 33ms. Daher sollten die eingesetzten Techniken möglichst einfach sein.

### **3.1.1. Funktionale Anforderungen**

#### **Sender**

- Mitteilung seiner Position über die Infrarotdiode

#### **Empfangssystem**

- Auffinden von Sendern
- Unterscheidung mehrerer Sender
- Verifizieren der Sender
- Daten der Sender bereitstellen

### **3.1.2. Nicht funktionale Anforderungen**

#### **Sender**

- Geringer Verbrauch von Ressourcen des Mikrocontrollers
- Geringer Strombedarf

#### **Empfangssystem**

- Einfache Methoden
- kurze Bearbeitungszeiten

## 3.2. Hardware

### 3.2.1. Infrarotsender

Für die Entwicklung der Software des Infrarotsenders stand das Evolution-Board (Abbildung 3.1) mit dem AVR AT90CAN128 zur Verfügung. Dieses Board ist eine gute Basis um Schaltungen vorläufig zu realisieren und zu testen, da alle Port und Schnittstellen nach Außen geführt sind und leicht zu bestecken sind. Für diese Arbeit ist dieses Board mehr als ausreichend, da für die Funktion des Senders lediglich ein Port mit einem Pin für die Diode benötigt.

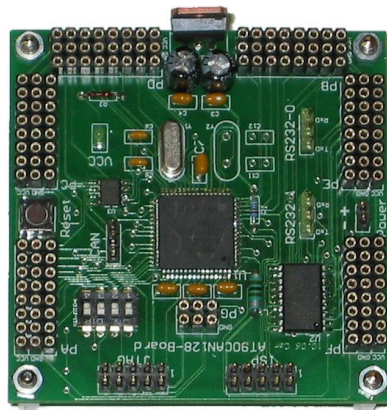


Abbildung 3.1: Evolution-Board des Labors für Technische-Informatik der HAW

Als Infrarotdiode ist im Labor der HAW nur die Osram SFH485P [Osram 2008] im Sortiment. Um die Diode auf dem Board zu betreiben, wird noch ein Vorwiderstand zur Strombegrenzung in Reihenschaltung benötigt. Einige Spezifikationen dieser Diode sind im Anhang A. zu finden.

### 3.2.2. Kamera

Die für diese Arbeit bereitstehende Kamera DMK 21F04 ist von der Firma The Imaging Source [Image 2008]. Diese Firma vertreibt viele verschiedene Kameraversion. Auflösungen von 640 x 480 bis 1280 x 960 Pixel sind dort verfügbar. Auch verschiedene Bildwiederholungsfrequenzen werden dort angeboten. Diese Kameras gibt es jeweils in zwei verschiedenen Anschlussarten. Zum einen als USB-Kamera oder mit der Verbindungsart FireWire.

Anschlussart	FireWire IEEE 1394
Allgemeines Verhalten: Video-Formate @ Bildrate	640 x 480 Y800 @ 30, 15, 7.5, 3.75 fps
Empfindlichkeit	4 lx bei 1/30s, Verstärkung 0 dB
Einstellungen (man): Belichtungszeit	1/3300 bis 1/30 s

Tabelle 3.1: Spezifikation DMK 21F04

### 3.2.3. Anschlussarten

Hindernisse oder Bewegungen von Person können den Sichtkontakt zum tag behindern oder ganz abreißen lassen. Deswegen werden für ein späteres robustes Positionierungssystem mehrere Kameras, die im Raum verteilt angebracht sind, benötigt. Bei der steigenden Anzahl von Kameras im System, steigt natürlich auch die zu übertragene Datenmenge stark an. Nicht alle Anschlussarten von Kameras sind dann noch geeignet, diese Menge zu bewältigen. In diesem Abschnitt werden zwei Typen zur einer möglichen Realisierung, einer Verbindung der Kameras mit einem Endsystem, aufgezeigt.

## USB

USB gibt es in verschiedenen Versionen mit unterschiedlichen Übertragungsraten. USB 1.1 bietet zwei Geschwindigkeiten. Low-Speed überträgt die Daten mit 1,5 Mbit/s und Full-Speed mit 12 Mbit/s. Die Erweiterung USB 2.0 erlaubt es Daten mit High-Speed 480 Mbit/s zu transferieren. USB 2.0 ist im Moment noch die aktuellste Version, doch USB 3.0 ist schon länger in der Entwicklung. Die Version soll mit 4,8 Gigabit/s übertragen können.

Die maximale Leitungslänge für eine einfache Verbindung von einem Host oder eines Hubs zu einem Gerät oder einem weiteren Hub beträgt 5m. Eine maximale Entfernung eines Gerätes zum Host, durch einsetzen mehrerer Hubs, beträgt 30m.

[Kelm 2001] [Intel 2008]



## FireWire

Auch von FireWire wurden bis heute mehrere Versionen entwickelt. So gibt es FireWire 400 ( IEEE 1394a ) mit einer Übertragungsrate von 100, 200 und 400 Mbit/s. Mit der Weiterentwicklung FireWire 800 ( IEEE 1394b ) ist eine Geschwindigkeit von 800 Mbit/s und 1600 Mbit/s möglich.

Eine einzelne Verbindung zwischen einem Host/Hubs zum einem Gerät darf hier bis zu 4,5m lang sein. Mit dem hintereinanderschalten von Hubs darf hier eine maximale Entfernung eines Gerätes zum Host von 72m erreicht werden. [Anderson 1999]

	Geschwindigkeit	Teilnehmer	Länge
USB 1.1	1,5 Mbit/s 12 Mbit/s	127 pro USB-Host	5m zwischen 2 Geräten 30m max.
USB 2.0	480 Mbit/s	127 pro USB-Host	5m zwischen 2 Geräten 30m max.
FireWire 400	100 Mbit/s 200 Mbit/s 400 Mbit/s	64 pro Bus	4,5m zwischen 2 Geräten 72m max.
FireWire 800	800 Mbit/s 1600 Mbit/s	64 pro Bus	4,5m zwischen 2 Geräten 72m max.

Tabelle 3.2: Vergleich von USB und FireWire

### Maximale Teilnehmerzahl bei FireWire 800:

Kameraauflösung :  $640 \times 480 = 307200$  Bytes

Bildwiederholungsfrequenz :  $307200 \text{ Bytes} \times 30\text{fps} = 9216 \text{ KBytes/s} = 73,728\text{Mbit/s}$

max. Teilnehmer pro Bus :  $1600\text{Mbit} / 73,728\text{Mbit} = \mathbf{21}$

Mit Hilfe der Übertragungsraten aus der Tabelle 3.2, kann die maximale Anzahl an Kameras berechnet werden. Bei der in dieser Arbeit verwendeten Kamera (Kap. 3.2.2.) und der Geschwindigkeit von 1600 Mbit/s kann eine maximale Anzahl von 21 Aufnahmegeräten pro Bus erreicht werden.

### **3.2.4. Laborrechner**

In dem Bereich Video- und Bildverarbeitung entstehen große Datenmenge und viele Operation. Deshalb spielt der Prozessor eine wichtige Rolle. In den letzten Jahren stieg die Leistung der Prozessoren rasant an, womit die digitale Bildverarbeitung in viele Bereiche vorstoßen konnte.

Der Laborrechner der für diese Arbeit zur Verfügung steht, ist ein Quad Core Prozessor mit 4 x 2400 mhz. Dieser Prozessor liefert die nötige Leistung und Reserven, um diese und die weiterführende Arbeiten zu verwirklichen, da in vergleichbaren Arbeiten sogar langsamere Rechner deren Aufgaben erfüllen konnten. Für die Verbindung zu den Kameras wurde der Rechner mit einer FireWire-Karte mit 3 Ports ausgestattet. Das darauf befindliche Betriebssystem ist Windows XP.

## **3.3. Software**

Der Abschnitt Software zeigt zunächst Techniken auf, die für die Implementierung des Empfangssystems mit der Kamera in Frage kommen. Als erstes werden Methoden zur Manipulation des Bildes beschrieben, die für eine erfolgreiche Erkennung des Senders im Bild nötig sein können. Danach wird mittels zwei gezeigter Szenarien zunächst eine Auswahl der Codierung, zur Unterscheidung der tags, getroffen und dessen Umsetzung und Decodierung auf der Empfangsseite nachfolgend beschrieben.

Ein weiterer Teil dieses Abschnitts ist die mögliche Systemarchitektur der Software des tags. Zum Ende dieses Abschnittes werden Methoden zur Fehlererkennung bei der Übertragung aufgezeigt.

### 3.3.1. Bildverarbeitung

Um das Auffinden des Senders im Bild zu erleichtern, muss der Hintergrund unterdrückt oder gar ganz aus dem Bild gelöscht werden. Dieser Abschnitt zeigt einige Methoden ein digitales Bild in diese Richtung hin zu manipulieren.

#### 3.3.1.1. Tiefpassfilter

Tiefpassfilter sind lokale Operationen(Kap. 2.4.3.) und werden zum glätten und verwischen eines Bildes benutzt(Abbildung 3.2). Bei einer Glättung des digitalen Bildes kann störendes Rauschen unterdrückt werden. Kleine Bildpunkte werden herausgefiltert, wobei grobe Strukturen erhalten bleiben. Glättungsfiler können mit verschiedenen Verfahren verwirklicht werden. [Jähne 2002]



Abbildung 3.2: links verrauschtes Bild, rechts nach einem Tiefpassfilter

[Wittman 2008]

## Rechteckfilter

Eine Methode ist der Rechteckfilter (Abbildung 3.3). Bei diesem Filter wird aus mehreren Bildpunkten ein Mittelwert gebildet. Alle neun Pixel, der betrachtete Pixel und die acht Nachbarn, fließen mit gleicher Gewichtung in den betrachteten Pixel ein. Diese Faltungsmaske ist separierbar, somit ersetzbar durch zwei eindimensionale Filterungen realisierbar. Zudem ist dieser Filter richtungsabhängig ( anisotrop ).  
[Meisel 2007]

$$\text{Pixel} = 1/9 \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Abbildung 3.3: Rechteckfilter

## Binominalfilter

Eine weitere Methode ist der Binominalfilter, wie in Abbildung 3.4 zu sehen. Hier wird nicht der Mittelwert gebildet, sondern die einzelnen Pixel gehen mit verschiedenen Wertungen ein. Kanten werden hier nicht so stark verwischt. Die Filtermatrix wird mittels des Pascalschen Dreiecks erstellt. Der Binominalfilter ist isotrop, daher nicht richtungsabhängig. Auch dieser Filter ist separierbar.[Meisel 2007]

$$\text{Pixel} = 1/16 \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad \text{Pixel} = 1/256 \times \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

Abbildung 3.4: zwei Binominalfilter erstellt mit dem Pascalschen Dreiecks

### 3.3.1.2. Schwellwertverfahren

Durch ein Schwellwertverfahren wird das digitale Bild segmentiert. Die Zugehörigkeit eines Bildpunktes zu einem Segment wird durch das Vergleichen des Grauwertes mit einem Schwellwert entschieden. Da diese Operation für jeden Pixel einzeln angewendet wird, ist das Schwellwertverfahren eine homogene Punktoperation (Kap. 2.4.2.). Diese Methode benötigt wenig Berechnung und ist dadurch sehr schnell.

[Jähne 2002]

### 3.3.1.3. Differenzbild

In manchen Fällen, wie z.B. bei Überwachungssystemen, möchte der Anwender Bewegungen und Veränderungen im Bild erkennen (Abbildung 3.5). Dies wird über ein Differenzbild verwirklicht. Dabei werden zwei aufeinander folgende Bilder voneinander subtrahiert. Hat sich nun im Bild etwas geändert, entsteht ein Helligkeitsunterschied zwischen den zu subtrahierenden Pixeln. Nur diese Pixel werden sichtbar und die Bewegung im Bild erkennbar. In der Abbildung 3.4 wird klar ersichtlich, dass nur noch die fahrenden Autos im Differenzbild vorkommen. Rauschen im Bild sollte vorher so gut wie möglich gefiltert werden. [Jähne 2002]

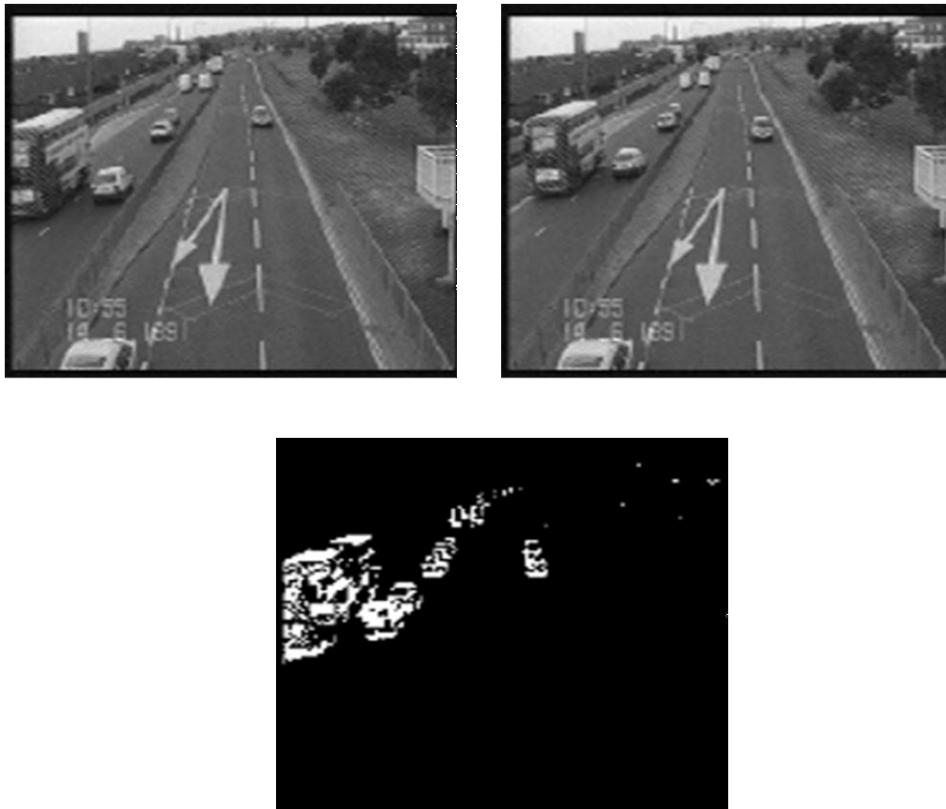


Abbildung 3.5: unten das Differenzbild der beiden Aufnahmen [Jiang 2008]

### **Zwischenfazit der Bildverarbeitung für spätere Umsetzung**

Da das Infrarotlicht als sehr helle Quelle im Bild dargestellt wird, kommt zunächst das Schwellwertverfahren(Kap. 3.3.1.2.) zum Einsatz. Hier wird ein sehr hoher Schwellwert gewählt, um alles Dunklere als das Infrarotlicht aus dem Bild zu löschen. Um kleineres Rauschen im Bild zu verhindern, kann je nach Bildqualität vorher ein Tiefpassfilter(Kap. 3.3.1.1.) zur Glättung mit in die Filterung aufgenommen werden. Wird jetzt noch das Differenzbild(Kap. 3.3.1.3.) erzeugt, wird weitestgehend nur noch der blinkende Sender im Bild auftauchen. Da aber das Sonnenlicht auch Infrarotlicht beinhaltet, kann es trotzdem noch zu einigen störenden Bildflecken kommen. Diese Probleme werden mit der im nächsten Abschnitt(Kap. 3.3.2.) beschriebenen Punkt-erkennung überwunden.

### 3.3.2. Punkterkennung

Nachdem der größte Teil des Hintergrunds im Bild herausgefiltert ist, wird mit der Punkterkennung der nächste Schritt vorgenommen. Diese Erkennung sucht aus den restlichen Störungen im Bild die Signale der Infrarotsender heraus. Da die Entfernung eines Senders zur Kamera variieren kann, muss die Punkterkennung verschieden große Punkte auffinden können. Verwirklicht werden kann dies durch eine Suchmaske [Ming 2007]. Die Auswahl der Maske entscheidet sich nach der maximalen und minimalen Größe von Punkten, die noch als Punkt erkannt werden sollen und können.

Eine weitere Anforderung an die Punkterkennung sind Bewegungen. Da die Untersuchung des ganzen Bildes zeitintensiv ist, kann bei gefundenen Punkten ein so genanntes Window-Of-Interest (WOI) [Wischy 2001] um den Punkt gebildet werden. In diesem kleinem Fenster wird nur noch nach Veränderung des Punktes geschaut. Dies verkleinert den Rechenaufwand erheblich. Bei der Punkterkennung für diese Arbeit ist dies jedoch nicht möglich, da zu jeder Zeit ein neuer Sender im Bild erscheinen könnte. Deshalb muss jedes Mal das gesamte Bild nach neuen Infrarotsendern durchsucht werden. Aus diesem Grund ist das Bilden eines WOI überflüssig.

Um nun das Verfolgen von Bewegungen der Infrarotsender umzusetzen, werden die Koordinaten eines gefundenen Punktes mit den Koordinaten eines in einem früheren Bild bereits gefundenen Punktes verglichen. Sind diese neuen Koordinaten in einem festgelegten Radius eines alten Punktes, so wird angenommen, dass dies der selbige ist und die Koordinaten des Punktes aktualisiert. Sollte dies für keine älteren Punkt zutreffen, wird dieser Punkt als neuer Sender in eine Liste übernommen.

### 3.3.3. Szenarien

Mit diesen vorgestellten Techniken lässt sich nun die Erkennung eines Infrarotsignals, als ein Punkt im Bild, realisieren. Jetzt muss nur noch sichergestellt werden, dass es sich auch um einen zu lokalisierenden Infrarotsender handelt. Zu dieser Komponente zählt auch das Unterscheiden zwischen mehreren Sendern. Der Sender muss also eine Identifikationsnummer(ID) senden. Hier werden nun zwei Szenarien für verschiedene Realisierungen dieser ID vorgestellt und deren Vor- und Nachteile aufgezeigt.

## Szenario 1

In Szenario 1 sendet ein Infrarotsender dauerhaft eine ihm zugeteilte Frequenz (z.B. 20 Hz). Hier kommt die Technik des Differenzbilds (Kap. 3.3.1.3.) zum Einsatz. Zuvor wird der Hintergrund grob herausgefiltert, sodass Bewegungen von Personen nicht im Differenzbild vorkommen. Da jetzt nur noch die Bewegungen des Senders im Bild sichtbar werden, erscheint ein Flankenwechsel des Senders von An auf Aus und andersherum klar im Bild. Dadurch entfällt eine Synchronisation des Empfängersystems auf den Sender. Das Empfängersystem kann zu jedem Zeitpunkt mit einer Decodierung des Senders beginnen. Über eine Sekunde gesehene Signale des Senders werden gezählt und so einfach die Frequenz des Senders ermittelt. Die Punkterkennung muss hier nur noch die Flankenwechsel zählen. Die Unterscheidung der Infrarotsender läuft über die zugeteilten verschiedenen Frequenzen.

Vorteile sind hier die einfache und schnelle Decodierung der Sender. Diese Methode bringt aber auch einige Nachteile mit sich.

Ein kleinerer Nachteil ist die Anzahl der möglichen Sender im System. Das liegt daran, dass die hier verwendete Kamera 30 Bilder pro Sekunde aufnimmt. Dies begrenzt die Frequenz nach oben auf maximal 30 Hz. Nach unten ist die Frequenzbreite durch die Sollzeit der Decodierung begrenzt. Ist diese z.B. auf eine Sekunde beschränkt, können schon keine Frequenz unter 1 Hz mehr erkannt werden. Die Anzahl wird hier also stark von der Kamera bestimmt.

Zweiter und größerer Nachteil sind Bewegungen eines Senders. Sobald ein Sender ein Infrarotsignal (Diode AN) über zwei Bilder sendet, und in der Zeit zwischen dem ersten und dem zweiten Bild seine Position verändert (Abbildung 3.6). Dadurch erscheinen im Differenzbild zwei Punkte, da es zwei Änderungen gibt. Der eine Punkt entsteht durch das fehlende Infrarotlicht an der alten Position und der zweite durch das neue Vorkommen des Infrarotlichts an der neuen Position.



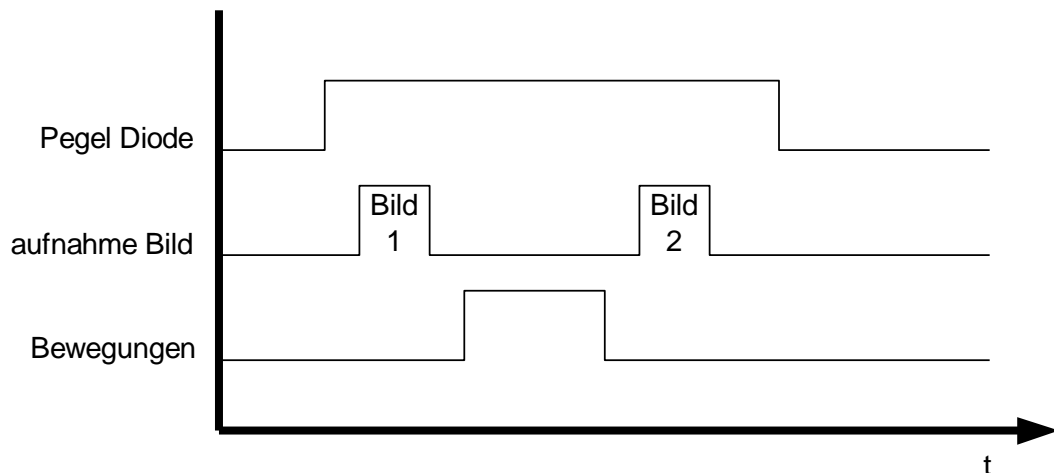


Abbildung 3.6: Bewegung des Senders über einer Folge von 2 Bildern

## Szenario 2

Für das Szenario 2 wird eine andere Codierung gewählt. Hier wird nicht mittels einer gesendeten Frequenz unterschieden, sondern jeder Infrarotsender bekommt einen eigenen Code zur Identifikation. Sender 1 bekommt z.B. die Codierung 10101 zugewiesen. Da die Kommunikation zwischen Sender und Empfänger nur in eine Richtung läuft, kann dort keine Synchronisation stattfinden. Der Infrarotsender kann nicht einfach seinen Code mehrmals hintereinander senden. Das Empfangssystem kann den Anfang und das Ende des Codes nicht erkennen. Deshalb muss das Empfangssystem auf einen definierten Start und auf ein definiertes Stopp warten.

Vorteil von diesem Szenario ist die Anzahl der Sender im System. Die Anzahl der Sender ist nur auf die Länge der Codes beschränkt. Mit einer Länge von 6, wie in dem Beispiel zuvor, können bereits 64 verschiedene Infrarotsender im System betrieben werden. Diese Länge ist nur durch die Solldauer der Decodierung beschränkt. In diesem Szenario ist es auch möglich die Sender zu priorisieren. Personen mit einer hohen Priorität haben einen kurzen Code und weniger priorisierte bekommen einen längeren Code. Dadurch werden wichtige Personen schneller erkannt. Durch den Wegfall des Differenzbildes in diesem Szenario führen Bewegungen des Senders zu keinen Problemen mehr.

Nachteil hier ist die Dauer der Decodierung des Codes. Umso mehr Teilnehmer sich im System befinden, umso länger wird der Code und somit auch die Decodierungszeit. Mit dem Start- und Stoppsignal verlängert sich das Decodieren nochmals.

	Vorteile	Nachteile
Szenario 1: senden von Frequenzen z.B. 20 Hz	- schnelle und einfache Codierung/Decodierung	- Bewegungen führen zu verfälschten Sendern - Anzahl der Teilnehmer sehr beschränkt
Szenario 2: senden von Codes z. B. 10101	- Anzahl der Teilnehmer kaum beschränkt - Bewegungen sind möglich	- Zeit der Decodierung kann bedingt ansteigen

Tabelle 3.3: Vergleich der Szenarien

### Wahl der Codierung

Wegen der Probleme bei Bewegungen scheidet der Lösungsansatz aus Szenario 1 aus. Die Vorteile können hier die Nachteile keineswegs ausgleichen. Szenario 2 hingegen erfüllt die Anforderung und dessen Nachteil ist nur bei sehr großen Systemen relevant.

#### 3.3.4. Codierung

In diesem Abschnitt wird erklärt, wie die Codierung aufgebaut sein muss, die der Infrarotsender bezogen auf die Bildwiederholungsfrequenz sendet.

Im Idealfall würde der Infrarotsender mit einer Frequenz von 30 Hz arbeiten und so seine Bits Bild für Bild übertragen können. Dies würde voraussetzen, dass alle

Taktgeber im System perfekt synchronisiert sind. Da aber keine bidirektionale Kommunikation stattfinden kann, ist dies nicht möglich.

Deshalb muss der Infrarotsender mit einer leicht verkleinerten Frequenz arbeiten. Die verlangsamte Frequenz bewirkt, dass regelmäßig Bits doppelt empfangen werden (Abbildung 3.7). Um diese Fehler zu kompensieren, muss eine eigene Codierung erstellt werden.

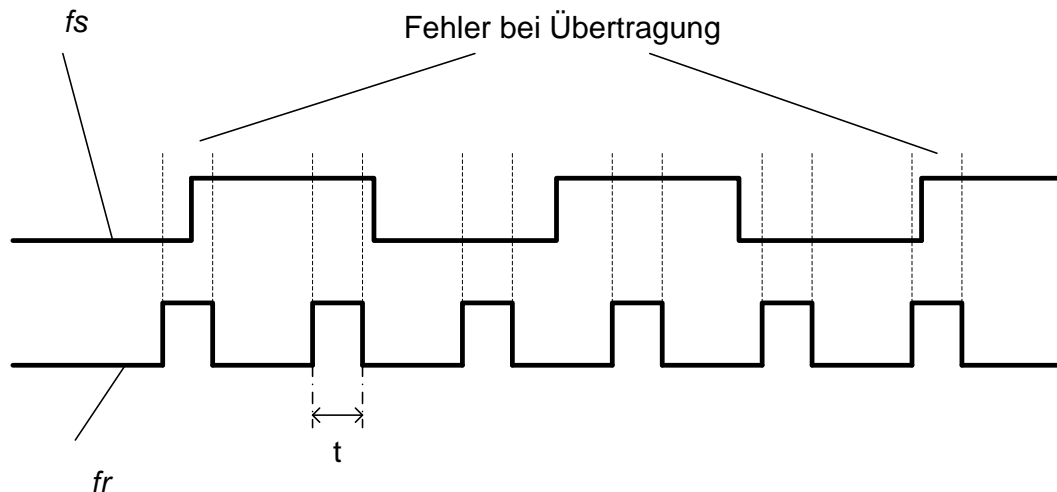


Abbildung 3.7: Bitfehler bei unterschiedlichen Frequenzen

Frequenz der Kamera :  $f_r = 30 \text{ Hz}$

Frequenz des Senders :  $f_s = 24 \text{ Hz}$

Belichtungszeit der Kamera:  $t$

Um diese Fehler in der Übertragung nun umgehen zu können, werden die Bits symbolisch übertragen [Mühlhäuser 2003]. Soll nun das Bit '0' übertragen werden, wird dies mit einer Bitfolge von '10' getan. Bei einem Fehler kommt es zu einer Folge aus '110'. Beim Decodieren wird diese Kombination nun auch als '0' angesehen. Möchte der Sender aber eine '1' übertragen, tut er dies mit dem Symbol bestehend aus '1110'. Als drittes Symbol muss nun noch das START/STOP Signal, auf das der Empfänger warten muss, codiert werden. Logischerweise wird dies mit der Folge aus '111110' realisiert, da ein Bitfehler einer '1' als '11110' empfangen wird.

Mit dieser symbolischen Codierung kann nun erfolgreich das Empfangssystem die Kennung von Infrarotsendern ermitteln.

### 3.3.5. Systemarchitektur des Senders

Die hier aufgeführten Punkte beschreiben zwei mögliche Systemarchitektur des Senders. Es werden verschiedene Lösungen für eine Realisierung des Schedulers aufgezeigt und nach ihren Vor- und Nachteilen ausgewählt. Der Scheduler hat die Aufgabe die Diode in bestimmter Zeit ein oder auszuschalten, um die Identifikation zu senden. Diese ID wird ständig von dem Sender ausgestrahlt.

#### Scheduler

Die Anforderung hier an den Scheduler ist die Realisierung eines Real-Time Systems. Es gibt verschiedene Stufen eines Real-Time Systems.

- **Hard Real-Time:** Bei einem Hard-Real-Time System hat die Nichteinhaltung aufgestellter Zeitgrenzen(Deadlines) eine kritische Auswirkung auf das ganze System. So ist z. B. das Nichteinhalten der Auslösezeit bei einem Air-bag im Auto nicht akzeptabel.
- **Firm Real-Time:** Kommt es zu einer Verletzung der Zeitgrenzen in einem firmen Real-Time Systems, werden die Ergebnisse zwar unbrauchbar, aber das laufende System wird nicht beschädigt.
- **Soft Real-Time:** In einem soften Real-Time System ist das Einhalten von Zeitgrenzen, z.B. aus Performance gründen, wünschenswert. Deren Nichteinhaltung hat so gut wie keine Folgen für das System.

In unserem Fall haben wir es mit einem Firm Real-Time System zu tun. Die hauptsächliche Aufgabe des Senders ist das dauerhafte Senden seiner Identifikation über die Diode. Kommt es dort zu einer Nichteinhaltung einer Zeitgrenze, ist die gesendete Identifikation meist nicht mehr zu gebrauchen. Dies hat keinen großen Einfluss auf das laufende System, da die Identifikation fortlaufend immer wieder gesendet wird.

## **Super Loop**

Die Super Loop ist die wohl einfachste Architektur um Tasks regelmäßig auszuführen. Mit dieser Methodik werden alle abzuarbeitenden Tasks hintereinander in einer großen Programmschleife abgearbeitet. Um nun Tasks zu einer bestimmten Zeit laufen zu lassen, werden Zeitverzögerungen (delays) zwischen die Tasks eingebaut.

Vorteile der Super Loop sind ihre einfache Programmierung und ihre geringe Forderung an Ressourcen. Jedoch hat diese Methode auch große Nachteile. Um zeitkritische Aufgaben bewältigen zu können, in denen Tasks zu einem bestimmten Zeitpunkt bearbeitet werden müssen, müssen alle Laufzeiten der Tasks genau bekannt sein und dürfen auch nicht variieren.

## **Scheduling mittels Hardware-Timer**

Eine weitere und bessere Lösung für einen Scheduler ist die Ausführung der Tasks in Interrupts Service Routinen (ISR). Bei diesem Ansatz bekommt jede Task eine eigene ISR und jeweils einen Timer zugewiesen. Damit ist es möglich die einzelnen Tasks zu einem exakt vorher bestimmten Zeitpunkt auszuführen. In dieser Lösung dürfen die Tasks nun auch unterschiedliche Ausführungszeiten besitzen, ohne dass es sich auf den Startzeitpunkt der nächsten Task auswirkt.

Auch dieser Ansatz bringt einige Nachteile mit sich. Zum einen braucht jede Task einen eigenen Timer, womit die Anzahl der Task durch die Anzahl der Timer begrenzt ist. Bei einer Erweiterung eines Systems mit einer Vollbelegung der Timer kommt nur ein Hardwarewechsel in Frage. Ein weiterer Nachteil ist das gleichzeitige Auftreten mehrerer Interrupts oder das Auslösen eines Interrupts bei noch laufendem Task.

Diese Probleme werden aber beseitigt, wenn es zu einem Einsatz eines einzigen Timers kommt. Alle Tasks im System werden vorher nach ihrer Ausführungszeit in eine Liste sortiert. In der einzig übergebliebenen ISR des Schedulers wird nun entschieden, welche Task dran ist und diese eine Task ausgeführt. Dieser Scheduler ist zusätzlich noch sehr gut erweiterbar, da es kaum einen Unterschied macht, einen oder z.B. zehn Tasks im System zu haben

	Super Loop	Hardware-Timer
Vorteil	- sehr einfache Umsetzung - wenig Ressourcen nötig	- einfache Umsetzung - exakte Ausführungszeiten - sehr gut erweiterbar
Nachteil	- ungenaue Ausführungszeiten	- mehr Ressourcen (Hardware-Timer) werden benötigt

Tabelle 3.4: Vergleich der Architekturen für den Scheduler

Da die Statusänderung der Infrarotdiode immer zum selben Zeit geschehen muss, und die Auswahl des Status variieren kann, wird die Architektur mit dem Hardware-Timer zum Einsatz kommen. Im Moment wird es zwar nur eine Task geben, dies kann sich aber mit späteren Erweiterungen in anderen Abschlussarbeiten ändern. Sollte dies der Fall sein, ist der Scheduler leicht zu erweitern.

### 3.3.6. Fehlererkennung in der Übertragung

Bei der Übertragung der Identifikation des tags, kann es zu Fehlern kommen. Zum Beispiel kann der Sichtkontakt mitten in der Übertragung der ID unterbrochen werden und somit die ID unbrauchbar sein. Hierzu werden im folgenden Fehlererkennungsverfahren aufgezeigt.

#### Paritätsbit

Das Paritätsbit dient zur Überprüfung von eventuell auftretenden Übertragungsfehlern. Die Idee dahinter ist die Ergänzung der übertragenen Bits durch ein zusätzliches Bit, das die Zahl der Einsen auf einen geraden oder einen ungeraden Wert bringt. Im ersten Fall ist es die gerade, im zweiten die ungerade Parität.

Beispiele:

ID	ID + ungerade Parität	ID + gerader Parität
000	0001	0000
001	0010	0011
011	0111	0110
111	1110	1111

Tabelle 3.5: zu sendende ID mit gerader und ungerader Parität

Die Verwendung eines Paritätsbits zur Fehlererkennung ist jedoch nur in wenigen Fällen sinnvoll, da nur Einzelfehler erkannt werden. Denn wenn zwei Bits kippen bleibt die Parität erhalten und der Fehler wird nicht erkannt. Zusätzlich wird das komplette Fehlen von Nullen oder das Fehlen von mehreren Einsen nicht erkannt.

[Sweeney 1992]

## Cyclic Redundancy Check

Besser als die lineare Methode des Paritätsbits ist die zyklische Redundanzprüfung CRC. Hier wird nicht nur ein Bit zur Fehlererkennung benutzt, sondern eine Checksumme, die eine gewünschte Länge annehmen kann, verwendet. Vor Beginn der Übertragung der Daten wird diese Checksumme mittels eines Generatorpolynoms berechnet (Abbildung 3.8). Dieses Generatorpolynom wird Bitweise mit den Daten XOR verknüpft bis ein Rest (Checksumme) übrig bleibt. Das verwendete Generatorpolynom ist auch dem Empfänger bekannt. Die Checksumme wird an die Daten angehängt und übertragen. Nach abgeschlossener Übertragung werden die Daten + Checksumme erneut dem CRC unterzogen (Abbildung 3.9). Diese Prüfung muss als Checksumme 0 ergeben.

Beispiel:

Generatorpolynom: 10011

Nutzdaten: 10101100101

Nutzdaten mit Anhang : 101011001010000

der Anhang hat immer die Länge Generatorpolynom-1

$$\begin{array}{r}
 10101100100000 \\
 \underline{10011} \\
 11010 \\
 \underline{10011} \\
 10010 \\
 \underline{10011} \\
 11000 \\
 \underline{10011} \\
 10110 \\
 \underline{10011} \\
 1010 \quad \text{Rest}
 \end{array}$$

Abbildung 3.8: Berechnung der Checksumme vor dem Senden

1010 ist der Rest der als Checksumme mit an die Nutzdaten gehängt wird. Somit würden die in diesem Fall zu senden Daten wie folgt aussehen: 10101100101010

Sind die Daten übertragen wird wieder der CRC durchgeführt.

$$\begin{array}{r}
 10101100101010 \\
 \underline{10011} \\
 11010 \\
 \underline{10011} \\
 10010 \\
 \underline{10011} \\
 11010 \\
 \underline{10011} \\
 10011 \\
 \underline{10011} \\
 0000 \quad \text{Rest}
 \end{array}$$

Abbildung 3.9: Berechnung der Checksumme nach dem Empfangen



Ist hier nun der Rest ungleich Null, hat ein Fehler in der Übertragung stattgefunden. Mit Hilfe dieser Methodik können mehrfache Bitfehler erkannt werden. Auch das komplette Fehlen von Bits wird durch das Bilden der CRC-Checksumme als Fehler festgestellt. Somit ist diese Fehlererkennung gut geeignet für unsere Anforderungen.

	Paritätsbit	CRC
Vorteile	<ul style="list-style-type: none"><li>- leicht umsetzbar</li><li>- nur ein Bit zusätzlich</li></ul>	<ul style="list-style-type: none"><li>- bemerken von mehreren Bitfehler</li><li>- bemerken von fehlenden Bit</li></ul>
Nachteile	<ul style="list-style-type: none"><li>- nur ein Bitfehler zu erkennen</li></ul>	<ul style="list-style-type: none"><li>- etwas aufwendiger in der Implementation</li><li>- Datenlänge erhöht sich</li></ul>

Tabelle 3.6: Vergleich der Fehlererkennungsmethoden

### 3.3. Zusammenfassung

Das Kapitel Analyse hat uns die Anforderung, die bestehenden Hardwarekomponenten, sowie eine Reihe von möglichen für Methoden und Techniken für eine Realisierung des Empfangssystems, wie auch für den Sender, aufgezeigt. Durch abwägen der Vor- und Nachteile, wurden die einzelnen Methoden und Techniken ausgesucht.

## 4. Realisierung

Das Kapitel Realisierung beschreibt die Umsetzung der ausgewählten Techniken und Methoden aus dem Kapitel Analyse(3.). Im ersten Teil wird die Realisierung des Infrarotsenders aufgezeigt. Der Aufbau und das Senden des Codes wird hier detailliert beschrieben. Mit dem Überblick was ein Tag sendet, wird nun im nächsten Abschnitt auf das Empfangssystem eingegangen. Als erstes werden einige Randbedingungen aufgezeigt, die eine Realisierung erst ermöglichen. Danach werden die implementierten Methoden beschrieben.

### 4.1. Sender

Als Programmierumgebung wurde AVR Studio der Firma Atmel verwendet [Atmel 2008]. Mit ihr ist das Kompilieren des Codes sowie das Aufspielen des Programms auf den Mikrocontroller möglich. Die Entwicklung der Software für den Sender wurde in C vorgenommen.

#### 4.1.1. Frequenzgenerierung fürs Senden

Für das frequenzierte Scheduling der Infrarotdiode wurde die Methode mit dem Hardware-Timer (Kap. 3.3.5.) gewählt. Da für das Senden eine sehr langsame frequenzierte Aufrufung der ISR benötigt wird ( $< 30\text{Hz}$ ), und der Mikrocontroller mit einem Taktgeber(Quarz) von 16 MHz betrieben wird, fiel die Wahl auf einen 16 Bit Counter.

Der Counter wurde auf fast-pwm und clear on compare match eingestellt. Über das Output Compare Register wurde die gewünschte Zeit eingestellt. Um bei einem Quarz

von 16 mhz nun auf eine kleine Frequenz zu kommen, musste der Prescaler auf den höchsten Wert (1024) eingestellt werden.

Da bei diesem Sender nur eine Task, nämlich das Ein- oder Ausschalten der Diode, vorhanden ist, entfällt die Implementation eines Dispatchers. Die Task wird direkt in der ISR bearbeitet.

Da der 16 Bit Tmer3 des AVR's benutzt werden muss, kann der Sleepmodus nicht eingebaut werden. Bei dem AT90CAN128 kann nur der Timer2 den Mikrocontroller aus dem Sleepmodus holen.

#### 4.1.1. CRC

Zur Fehlererkennung wird die zu sendende Identifikationsnummer einem CRC unterzogen. Da die Übertragungsdauer mit jedem zusätzlichen Bit um mehrere Bilder verzögert wird, wurde nur ein 4 Bit langes CRC-Polynom gewählt. Somit verlängern sich die ID nur um 3 Bit.

Zunächst wird in der Funktion *crc(uint32\_t data\_to\_send)* die Daten(ID) hinten um 3 Bit erweitert. Im zweiten Schritt wird nun immer das höchstgradige Bit mit der Wertigkeit '1' in den Daten gesucht. Sobald gefunden, wird das CRC-Polynom auf den gleichen Grad der Daten gebracht und die XOR-Verknüpfung vorgenommen. Dies wird solange wiederholt bis das Ende der Daten erreicht ist. Am Schluss wird nun der CRC-Rest hinten an die ID angefügt.

#### 4.1.2. Senden der Identifikation

Sind die zu senden Daten( ID + CRC-Summe ) vollständig, müssen nun die einzelnen Bits in eine symbolische Codierung umgewandelt werden. Diese Umwandlung wird in der Funktion *encode\_tx\_data(uint32\_t data)* vorgenommen. Die symbolische Codierung wurde wie in Abschnitt 3.3.4. beschrieben gewählt:

Bit/Symbol	Codierung
START/STOP	111110
1	1110
0	10

Tabelle 4.1: Symbole und deren Codierung

Beispiel:

ID      CRC      Daten  
 10101 + 011 = 10101011

Umgewandelt:

1110 10 1110 10 1110 10 1110 1110

Diese Folge wird in eine Variable *encoded\_tx\_data* geschrieben und deren Länge dabei aufgezeichnet. Somit kann die Länge der ID variieren und es können kurze IDs, deren Übertragung dadurch verkürzt wird, für höher priorisierte Personen oder Objekte vergeben werden. Die Länge der IDs wird durch die Wahl der Datentypen der Variablen begrenzt. In diesem Fall wurde eine 64 Bit Variable für *encoded\_tx\_data* gewählt. Mit dieser Größe sind maximal 65535 Sender programmierbar.

Ist die Umwandlung der ID abgeschlossen, wird der Scheduler gestartet. Welchen Status die Diode annehmen soll wird in der ISR des Schedulers entschieden. Über eine einfache Schiebeoperation werden die einzelnen Bits der Daten an die Diode weiter gegeben. Bevor die Daten gesendet werden, wird jedes Mal das START/STOP-Symbol (Tabelle 4.1) Bit für Bit abgearbeitet. Ist das letzte Bit der Daten an die Diode gelegt worden, fängt dieser Vorgang sofort von vorne an. Bei einer 5-Stelligen ID beträgt die maximale Übertragungsdauer 42 Bilder. Hinzu kommen Bitfehler durch die unsynchronen Frequenzen(Kap.3.3.4.) der Kamera und des Senders.

## 4.2. Empfangssystem

Für die Entwicklung der Software des Empfangssystems, bestehend aus der Kamera(Abschnitt 3.2.2.) und dem Laborrechner(Abschnitt 3.2.4.), kommt die API IC Imaging Control(Abschnitt 2.5.) zum Einsatz. Diese API kann in Verbindung mit der Entwicklungsumgebung Visual Studio 2003, die den Studierenden an der HAW Hamburg zur Verfügung steht, genutzt werden. In der vorliegenden Version 3.0 von IC Imaging Control, kann die Entwicklung wahlweise unter C++ oder C# geschehen [Image 2008]. Da ich im Verlauf des Studiums es schon mit C++ zu tun hatte, fiel die Wahl der Programmiersprache auf diese.

Über die nun vorhandene Projektvorlage IC Dialog Application in Visual Studio werden die nötigsten Klassen vorgeneriert. Die Klasse Grabber ist die Hauptklasse, die zur Verfügung gestellt wird. Eine Objekt-Instanz dieser Klasse führt die eigentliche Bilderfassung durch und stellt die Bilddaten bereit. Objekte der auch vorgenerierten Klasse CListener können diesem Grabber übergeben werden. Jedes Mal, wenn jetzt ein Bild der Kamera in den Puffer des Grabbers übertragen wurde, führt dieser die Callback-Funktion frameReady im Listener aus. Das Bild liegt nun in einem eindimensionalen Array vor. Der Pixel mit dem Index 0 liegt im Bild ganz unten links (Abbildung 4.1).

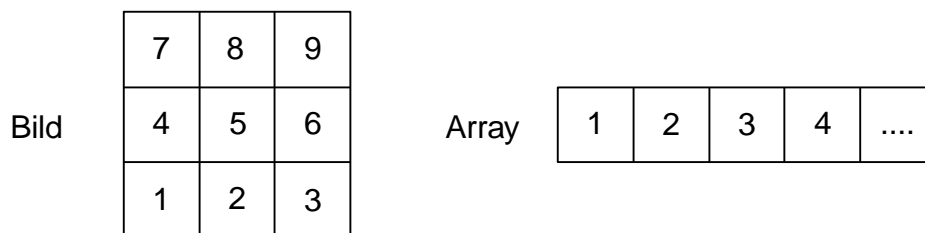


Abbildung 4.1: links das Bild, wie wir es sehen und rechts wie es im Puffer liegt

An diesem Punkt wird nun die weitere Verarbeitung angesetzt. Zuerst wird an der Stelle mittels der entworfenen Klasse BackgroundFilter die Bildverarbeitung vorgenommen. Danach wird in der zweiten von mir entworfenen Klasse, BlobDetector, die restliche Erkennung des Senders vorgenommen. Diese zwei Klassen werden im Laufe dieses Abschnittes weiter beschrieben.

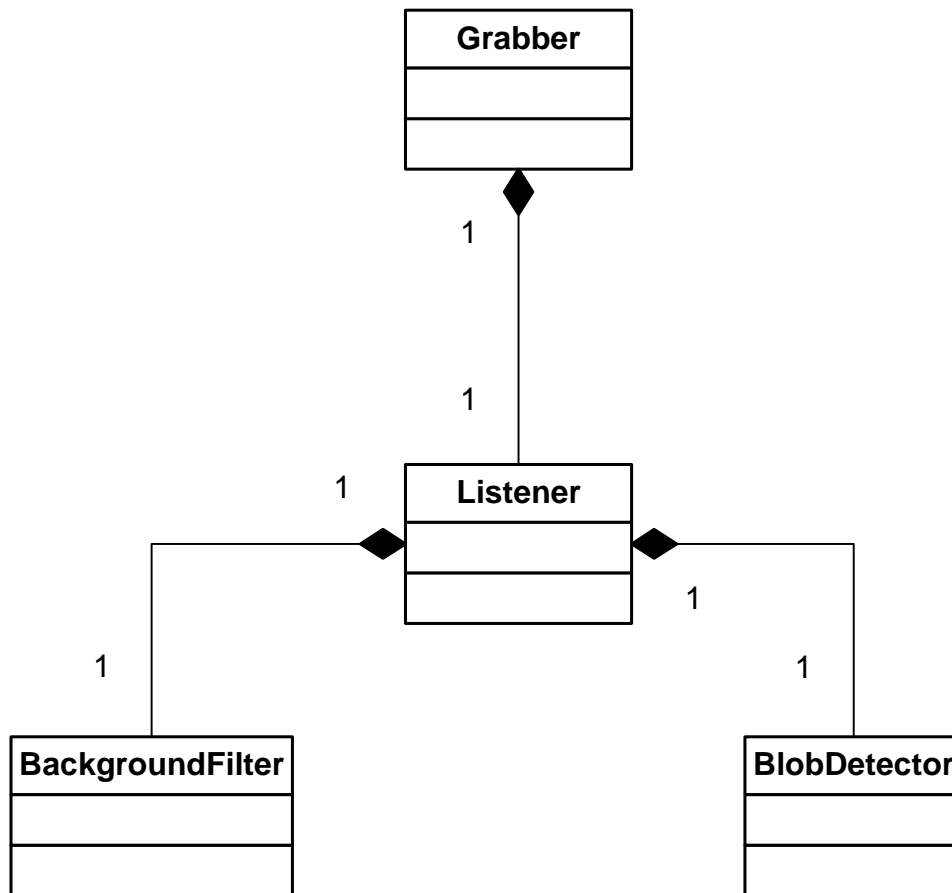


Abbildung 4.2: UML Klassendiagramm grobe Übersicht

### 4.2.1. Randbedingungen

Die Realisierung des Empfangssystems unterliegt einigen Randbedingungen.

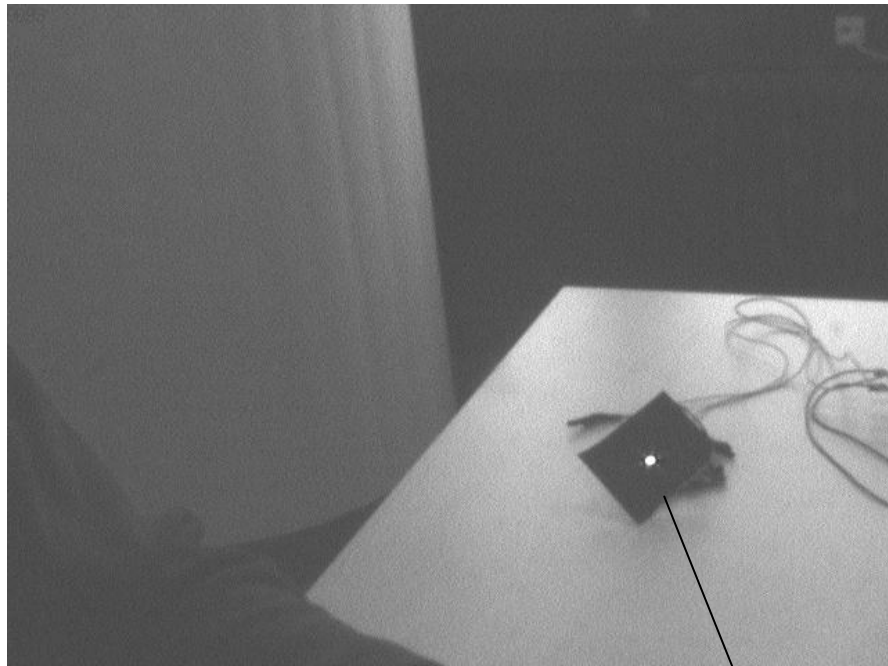
- Die Entfernung der Kamera zum Sender ist von der Auflösung abhängig. Bei einer Raumgröße oder zu beobachten Fläche von 6,4m x 4,8m würde, bei einer vorliegenden Kameraauflösung von 640x480 Pixel, ein Pixel 1cm repräsentieren. Da die Infrarotdiode nur wenige Millimeter breit ist, wird eine Erkennung bei dieser Konstellation so gut wie unmöglich. Um eine gute Erkennung des Senders zu schaffen, muss die Entfernung zur Kamera so verkürzt werden, dass das Signal der Diode eine Größe von mehreren Pixeln ergibt.

Alternativ können mehrere Dioden zu einem Kreis angeordnet oder die Fläche des abstrahlenden Infrarotlichtes mit einer Art Milchglasscheibe vergrößert werden.

- Die zweite Bedingung für eine erfolgreiche Realisierung ist die Montage der Infrarotdiode auf einem dunklen Hintergrund. Steht ein zu lokalisierendes Objekt nämlich in der Nähe einer hellen Lichtquelle, wie einem Fenster, so kann die Diode alleine nicht von dem hellen Hintergrund unterschieden werden. Ist die Diode aber auf einem dunklen Ausweis (5x5 cm) montiert, kann die Diode in dieser Fläche klar hervorstechen.
- Ähnlich wie die zweite Randbedingung, spielt die nächste Bedingung auch auf den Hintergrund an. Das Kameraobjektiv muss so eingestellt werden, dass das Bild unterbelichtet aufgenommen wird. Es ist so weit herab zu stellen, bis der Hintergrund sich von dem Infrarotlicht absetzt und dieses Licht sich im obersten Helligkeitsbereich der Kamera befindet. Wird die Kamera stark überbelichtet entsteht so gut wie kein Helligkeitsunterschied zwischen dem Infrarotlicht und dem Hintergrund. Womit eine erfolgreiche Erkennung kaum möglich wird.

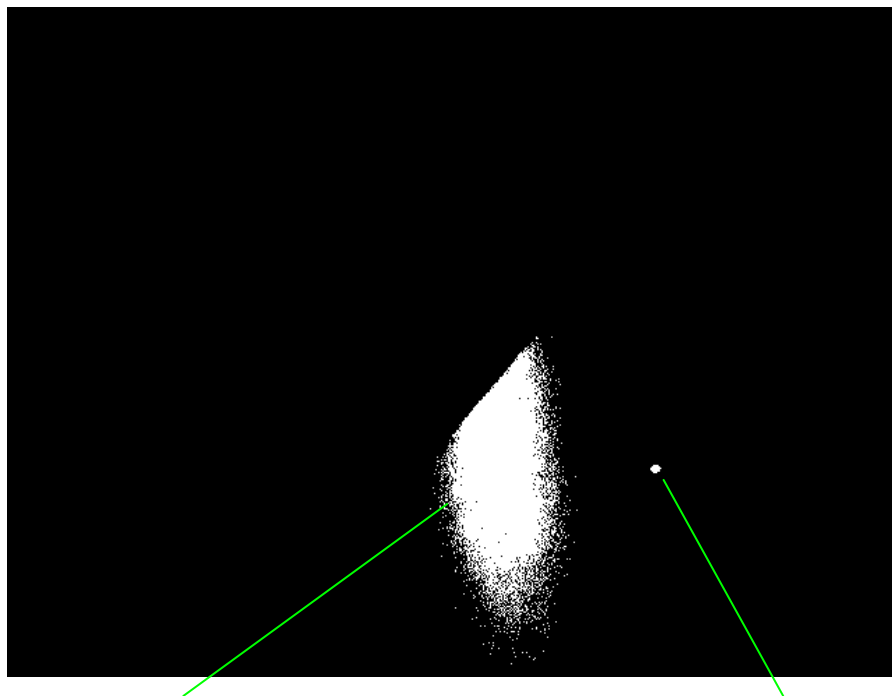
#### **4.2.2. Bildbearbeitung der Kamerabilder**

Die Bildbearbeitung, in der Klasse BackgroundFilter, ist kleiner ausgefallen als zunächst erwartet. Zum Einsatz kommt nur das Schwellwertverfahren(Kap. 3.3.1.2.). Mit Hilfe eines Schwellwertes der zur Laufzeit auch verändert werden kann, wird ein Binärbild erzeugt. Alles unter diesem Wert wird auf 0 gesetzt und alles drüber auf 255. Dies erleichtert später die Berechnungen mit den einzelnen Bildpunkten. In Abbildung 4.3 ist das unveränderte unterbelichtete Bild zu sehen und im zweiten(Abbildung 4.4) das Bild nach dem Schwellwertverfahren.



Infrarotsender (tag)

Abbildung 4.3: Original Bild vor der Bearbeitung



Reflektion von Sonnenlicht

Infrarotsender (tag)

Abbildung 4.4: Bild nach dem Schwellwertverfahren



Alternativ zur digitalen Bildverarbeitung lag mir ein physikalischer Lichtfilter für die Kamera vor. Dieser Filter ist nur im infraroten Bereich (Kap. 2.1.) lichtdurchlässig. Befindet sich die Kamera nun in einem Raum ohne weitere Infrarotlichtquelle (Sonnenlicht durchs Fenster), ist im aufgenommenen Bild nur die Infrarotdiode des Senders zu sehen. Dadurch kann die Bildverarbeitung in solchen Räumen entfallen.

### 4.2.3. Blob-Detector

Da nun der Punkt klar im Bild ersichtlich ist, muss dieser noch aufgefunden werden. Hier für wurde die Klasse BlobDetector erstellt. Das detektieren eines Punktes, in diesem Fall eines tags, geschieht über eine Maske (Abbildung 4.5) die aus zwei Teilen besteht. Der erste Teil ist nur zum Auffinden von Punkten einer Mindestgröße zuständig.

Nach und nach wird über das Bild eine 3x3 Matrix, in Abbildung 4.5 grün dargestellt, geschoben. Wird nun eine Fläche mit einem vorher eingestellten Mindestwert von Pixel gefunden, kommt der zweite Teil der Maske zum Einsatz. In diesem Teil wird in einem kleinen Abstand zur 3x3 Matrix das Umfeld der Fläche abgesucht. Ist dieser frei von hellen Pixeln, wird diese Fläche als Punkt erkannt.

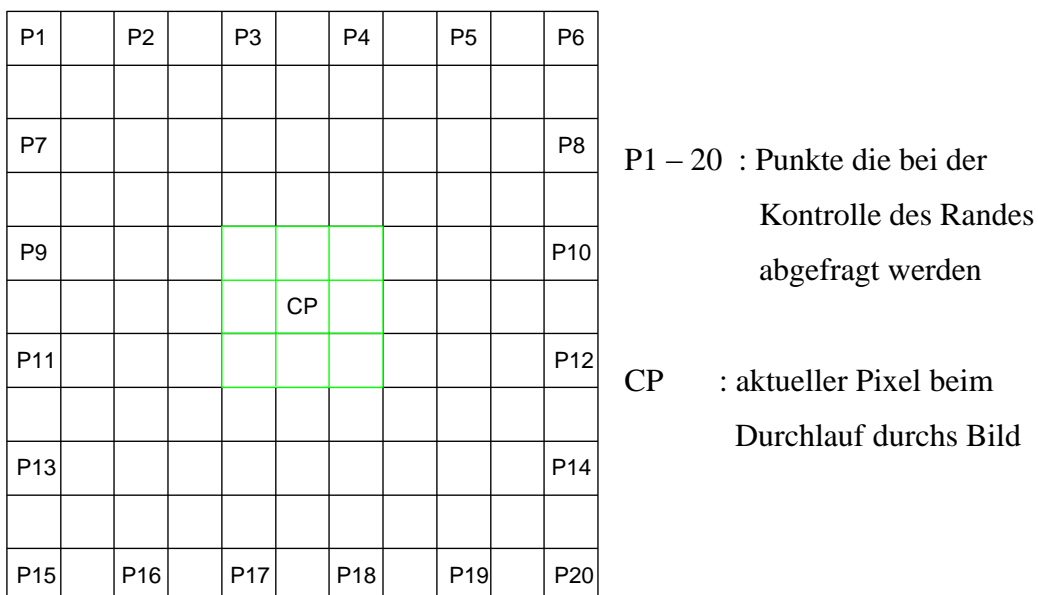


Abbildung 4.5: Maske zum auffinden von Punkten(Blobs)

Grün der min Bereich des Punktes

War die Erkennung eines Punktes erfolgreich, so wird der Index des Pixels in die NxM Matrixdarstellung(2.4.1.) umgerechnet. Danach muss nun noch entschieden werden, ob es sich um einen bereits gefundenen Sender, der sich z.B. bewegt hat, oder einen neuen Sender handelt. Dazu wird eine, für die gefundenen Sender, angelegte Liste durchsucht. Zunächst wird nach und nach die Position der schon eingetragenen Sender mit der gerade gefundenen Position verglichen. Hier erleichtert die Matrixdarstellung das Bilden einer sogenannten Region of Interest (ROI). Tritt eine Übereinstimmung auf, wird die Position des Senders aktualisiert. So ist auch die mögliche Bewegung der Sender realisiert. Ist dies für keinen in der Liste befindlichen Sender der Fall, so wird ein neuer Sender mit seiner Position in die Liste eingetragen.

### **Verifizierung des Senders**

Ist der Sender in der Liste eingetragen, ist noch nicht sichergestellt, dass es sich wirklich um einen Sender im Positionssystem handelt. Es könnte z.B. ja nur eine helle blinkende Lichtquelle sein. Um nun den Sender zu verifizieren, werden seine Signale über eine Bildfolge, in einer Variable, aufgezeichnet. Beim Auftauchen des Signals wird die Variable um eine Stelle nach links geschoben und eine 1 aufaddiert. Setzt das Signal aus, wird nur geschoben und somit eine '0' realisiert. Diese Variable wird nach jedem Bild auf das START/STOP-Symbol(Tabelle 4.1) hin überprüft. Wird dieses Symbol erkannt, wird nun die Variable auch auf das Symbol für die "1" und die "0"(Tabelle 4.1) abgefragt. Die erkannten Nullen und Einsen werden als zwischenzeitliche ID gesichert. Taucht nun das START/STOP-Symbol erneut auf, wird die zwischenzeitliche ID dem CRC, wie in Abschnitt 4.1.1. schon beschrieben, unterzogen. Bei erfolgreicher Überprüfung wird die ID beim Sender abgespeichert. Ist dies nicht der Fall, wird sie einfach verworfen.

### **Löschen eines inaktiven Senders**

Ist ein Sender z.B. ausgeschaltet worden, wird dieser, nach einer einstellbaren Zeit, aus der Liste gelöscht. Es wird die Framenummer des letzten Vorkommens des Senders mit der aktuellen Framenummer verglichen. Ist der Abstand zu groß, wird der Sender gelöscht und die Liste aktualisiert.

### **4.3. Zusammenfassung**

In diesem Kapitel wurde die Realisierung des Senders und des Empfangssystems beschrieben. Es wurde aufgezeigt wie die Codierung der symbolischen Übertragung im Sender abläuft und mit Hilfe einer Fehlererkennung erweitert wird.

Der Aufbau des Empfangssystems wurde dargelegt. Die Realisierung und das Ergebnis der Bildverarbeitungsmethode haben wir in diesem Kapitel beschrieben bekommen. Am Ende wurden die Umsetzungen der Erkennung und Decodierung, sowie die Verifikation des Senders aufgezeigt.

# 5. Qualitätssicherung

In diesem Kapitel wird die prinzipielle Arbeitsweise des Empfangssystems getestet. Dem Sender wurden keine größeren Tests unterzogen, da das Testboard(Kap. 3.2.1.) nicht einer Endversion entspricht.

## 5.3. Bewegungen

Wie in Abschnitt 4.2.3. beschrieben, werden Bewegungen mittels einer Region of Interest realisiert. Welche Geschwindigkeiten mit dem Sender möglich sind, wird über die Größe der ROI festgelegt. Diese kann beliebig groß gewählt werden. Dabei ist aber drauf zu achten, dass bei zu großen ROIs sich Sender gegenseitig stören. So werden zwei Sender die eine kleinere Entfernung als die Größe der ROI zu einander haben, als ein und derselbe Sender angesehen.

Um die Bewegungen zu testen, wurde hierzu der Sender, mit einer Entfernung von 1,5m zur Kamera, in verschiedenen Geschwindigkeiten nach links, rechts, oben und unten bewegt. Die Größe der ROI wurde von 10 x 10 bis hin zu 100 x 100 Pixel eingestellt.

Bildbreite bei einer Entfernung von ca 1,5m	ca 100cm
Auflösung der Kamera	640 x 480 Pixel
ROI-Einstellung	50 x50 Pixel
Bilder pro Sekunde	30fps

Tabelle 5.1: Daten für die Geschwindigkeitsberechnung

mögliche Geschwindigkeit(aus Tabelle 5.1) :

$$\frac{100\text{cm} \times 25 \text{ Pixel} \times 30 \text{ fps}}{640 \text{ Pixel}} = 1,17\text{m/s}$$

Gehgeschwindigkeiten in Hamburg = 1,47 m/s [Morgenroth 2003]

Ab einer Größe von 50 x 50 Pixel wurden, bezogen auf die Entfernung zu Kamera, akzeptable Geschwindigkeiten möglich. Die letztendlich beste Größe muss je nach den Gegebenheiten vor Ort gewählt werden.

Weiterhin wurde der Winkel des Senders zur Kamera geändert. Dazu wurden die Kamera und der Sender auf eine Höhe gelegt. Mittels eines nebenstehenden Winkelmessers wurden Werte ermittelt. Wie zu erwarten wurden nur akzeptable Ergebnisse leicht unterhalb des Abstrahlwinkels, der Infrarotdiode, von  $\pm 40^\circ$  erreicht. Auch bei der Entfernung zur Kamera ist die Diode der entscheidende Faktor. So produziert das Restlicht der Diode, bei starker Reduzierung der Entfernung, noch einen Punkt(Blob) im Bild. Dadurch kommt es zu Fehlern bei der Sollübertragung einer '0'. Deshalb musste die Codierung der Symbole(Tabelle 4.1)hinten mit einer weiteren '0' erweitert werden. Grund hierfür ist wohl eine schlechte Ausschaltzeit der Diode.

## 5.2. Spiegelungen

Treten Spiegelungen der Sender im Bild auf, kommt es zu einer Duplizierung des Senders. Es kann zwischen zufälligen und dauerhaften Spiegelungen unterscheiden. Zufälligen Spiegelungen treten zu einem unbestimmten Zeitpunkt und Ort auf. Meist aber nur für eine kurze Zeit. Diese Spiegelungen lassen sich nur mit großem Aufwand verhindern. Dauerhafte Spiegelungen, z.B. durch Reflektoren von Leuchten, können über eine feste Maske für die jeweilige Kamera verhindert werden [Mühlhäuser 2003]. In dieser Maske ist jeder feste Spiegelbereich im Bild eingetragen. Erscheint nun ein Sender in diesem Bereich wird er als Spiegelung angesehen und ignoriert.

In diesem System ist keine Maßnahme gegen Spiegelungen vorgenommen worden. Bei zufälligen Spiegelungen wird der duplizierte Sender mit in die Liste aufgenommen. Nachdem die Spiegelung verschwunden ist, wird der Sender nach einiger Zeit automatisch gelöscht. Masken für die dauerhaften Spiegelungen müssen Vorort angefertigt werden.

## 5.2. Zeitmessungen

Eine wichtige Anforderung ist die Laufzeit der eingesetzten Methoden und Techniken. Da die Kamera 30 Bilder pro Sekunde aufnimmt, bleiben zur Bearbeitung des Bildes und für die Bildbearbeitung, Erkennung und Unterscheidung der Sender nur eine Bearbeitungszeit von ca 33ms.

Mit den Funktionen *QueryPerformanceFrequency* und *QueryPerformanceCounter* wurden nun die Zeiten auf dem in Abschnitt 3.2.4. System gemessen. Die Zeiten der Funktion *filterBackground* der Klasse *BackgroundFilter* und der Funktion *detectBlobs* der Klasse *BlobDetector* wurden unabhängig voneinander gemessen.

Funktion	Zeiten ohne Sender	Zeiten mit 2 Sendern
<i>filterBackground</i>	1,4ms - 1,5ms	1,4ms - 1,5ms
<i>detectBlobs</i>	4,9ms - 5,0 ms	4,9ms - 5,0 ms

Tabelle 5.2: Zeitmessung der Funktionen

Die Zeiten wurden einmal mit 2 Sendern und einmal mit keinem Sender im Bild ermittelt, um die Laufzeit der Verwaltung und Decodierung der Sender zu überprüfen. Die Zeiten wurden über eine Folge von 50 Bildern 20-mal aufgezeichnet. Wie in Tabelle 5.2 zu sehen ist, ist die Einhaltung der 33ms pro Bild deutlich gegeben. Zusätzlich lässt sich aussagen, dass die Laufzeit sich bei eingesetzten Sendern kaum verändert hat.

## **6. Fazit**

In dieser Arbeit konnte erfolgreich ein Teilsystem für das IndoPosys realisiert werden. Ziel war es tragbare Infrarotsender mittels einer Kamera zu erkennen und zu unterscheiden. Im Laufe der Arbeit wurde ein Basissystem für den Sender entworfen, mit dem es möglich ist verschiedene IDs zu senden. Die hauptsächliche Arbeit bestand jedoch darin ein Empfangssystem zu entwickeln, das Infrarotsignale im Bild mit möglichst einfachen Mitteln herausfiltert und diese Signale decodiert.

Dieses System kann nun als Basis für folgende Arbeiten an dem IndoPosys verwendet werden.

### **6.1. Bewertung der Ergebnisse**

Die Unterscheidung und Verfolgung der Sender läuft erfolgreich. Mit der gefundenen Codierung für die Übertragung der Identifikation lassen sich sehr viele Sender im Gesamtsystem betreiben. Zusätzlich ist sogar eine gewisse Priorisierung der zu positionierenden Objekte möglich.

Auch die Anforderung an die Laufzeit wurde erfolgreich umgesetzt. Mit der derzeitigen Gesamtbearbeitungszeit von ca 6,5ms, bleibt in diesem Fall ca. 80% der Zeit für weitere Maßnahmen übrig.

Mit dem entwickelten Teilsystem lässt sich schon jetzt in wenigen Schritten ein einfaches Positionierungssystem aufbauen. Die gefundenen Sender können der Kamera zugeordnet werden. Über die symbolische Position(Abschnitt 2.2.2.) der Kamera kann nun eine Aussage in Bezug auf die symbolische Position der Sender getroffen werden.

## 6.2. Verbesserungsvorschläge und mögliche Änderungen

Der erste Punkt der verbessert werden sollte, ist die Infrarotdiode. Hier sollte nach einem besseren Ersatz gesucht werden. Diese sollte möglichst einen großen Abstrahlwinkel sowie eine schnelle Ein- und Ausschaltzeit besitzen. Dies würde den Erkennungsbereich des Empfangssystems erhöhen und somit eine Ersparnis von Kameras einbringen. Auch der Leistungsverbrauch sollte beachtet werden, damit später ein langlebiger Betrieb mit Batterien möglich ist.

Die Entwicklung und Fertigung der Hardware eines ersten tags ist ein zweiter Punkt für die Zukunft. Hier muss auf Kosten und Energieeffizienz geachtet werden. Da das Programm des Senders kaum Ressourcen benötigt, kann ein sehr kleiner Mikrocontroller zum Einsatz kommen. Es wird lediglich ein Port und ein Timer benötigt. Hier kann also z.B. ein Atmel ATtiny zum Einsatz kommen[Atmel 2008]. Mit einem kleineren Quarz kann ein anderer Timer gewählt werden und somit der Sleepmode des Mikrocontrollers, um Energy zu sparen, genutzt werden.



# Literaturverzeichnis

- [Anderson 1999] Anderson, Don, 1999 : FireWire System Architecture  
Version 1999 ; Verlag: Addison-Wesley Longman ; Auflage 2nd edition  
ISBN 0-201-48535-4
- [Atmel 2008] Atmel : Hersteller von Mikrocontrollern  
<http://www.atmel.com/>  
Version 2008 ; letzter Abruf 08.2008
- [Bauer 2003] Bauer, Manfred, 2003 : Vermessung und Ortung mit Satelliten  
Version 2003 ; Verlag: Herbert Wichmann ; Auflage 5 ; ISBN 3-87907-360-0
- [BBF 2008] Bundesministeriums für Bildung u. Forschung : Ambient Assisted Living  
<http://www.aal-deutschland.de/>  
Version 2008 ; letzter Abruf : 08.2008
- [BFS 2008] Bundesamt für Strahlenschutz : <http://www.bfs.de>  
Version 2008 ; letzter Abruf: 07.2008
- [Borriello 2001] Borriello, Gaetano ; Hightower, Jeffrey : *Location Systems for Ubiquitous Computing*  
[http://www.seattle.intel-research.net/pubs/062120021154\\_45.pdf](http://www.seattle.intel-research.net/pubs/062120021154_45.pdf)  
Version 2001 ; letzter Abruf 09.2008
- [Gregor 2006] Gregor, Sebastian : Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden  
<http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf>  
Version 2006 ; letzter Abruf 09.2008
- [Image 2008] The Image Source : Hersteller von Bildverarbeitungs-Komponenten  
<http://www.theimagingsource.com/de/>  
Version 2008: Abruf : 07.2008
- [Intel 2008] Intel : Entwicklung des USBs  
<http://www.intel.com>  
Version 2008 ; Abruf : 07.2008
- [Jähne 2002] Jähne, Prof. Dr. Bernd, 2002 : Digitale Bildverarbeitung  
Version 2002 ; Verlag: Springer ; Auflage 5 ; ISBN 3-540-41260-3

- [Jiang 2008] Jiang, Prof. Xiaoyi ; Rothaus, Kai : Vorlesung Computer Vision  
<http://cvpr.uni-muenster.de>  
Version 2008 ; letzter Abruf : 08.2008
- [Kelm 2001] Kelm, Dipl.-Ing Hans Joachim, 2001 : USB 2.0  
Version 2001 ; Verlag: Franzis ; Auflage 1 ; ISBN 3-7723-7965-6
- [Lang 1997] Lang, Hans Werner ; FH Flensburg : CRC-Verfahren  
<http://www.inf.fh-flensburg.de/lang/algorithmen/code/crc/crc.htm>  
Version 1997; letzter Abruf : 08.2008
- [Meisel 2007] Meisel, Prof. Dr.-Ing. Andreas : Robotvision  
[http://www.informatik.haw-Hamburg.de/fileadmin/Homepages/ProfMeisel/Vorlesungen/WP\\_RobotVision/V/RV03.pdf](http://www.informatik.haw-Hamburg.de/fileadmin/Homepages/ProfMeisel/Vorlesungen/WP_RobotVision/V/RV03.pdf)  
Version 2007 ; letzter Abruf 08.2008
- [Ming 2007] Ming, Anlong ; Ma, Huadong : A blob detector in color images  
[http://portal.acm.org/ft\\_gateway.cfm?id=1282335&type=pdf&coll=GUIDE&dl=GUIDE&CFID=2849811&CFTOKEN=42097989](http://portal.acm.org/ft_gateway.cfm?id=1282335&type=pdf&coll=GUIDE&dl=GUIDE&CFID=2849811&CFTOKEN=42097989)  
Version 2007 ; letzter Abruf 09.2008
- [Morgenroth 2003] Morgenroth, Dr. Olaf ; TU Chemnitz : Gehgeschwindigkeiten  
<http://www.tu-chemnitz.de/tu/presse/2003/06.17-11.06.html>  
Version 2003; letzter Abruf 09.2008
- [Mühlhäuser 2003] Mühlhäuser, Max ; Aitenbichler, Erwin :  
An IR Local Positioning System for Smart Items and Devices  
<http://ieeexplore.ieee.org/iel5/8560/27094/01203576.pdf?tp=&isnumber=&ar number=1203576>  
Version 2003 ; letzter Abruf 09.2008
- [Osram 2008] Osram : Infrarotdiode  
[www.osram.de](http://www.osram.de)  
Version 2008; letzter Abruf : 07.2008
- [Perret 2008] perret optic : OZON - SONNE - AUGEN  
[http://www.perret-optic.ch/Soleil\\_ozone\\_yeux/soleil\\_ozone\\_yeux\\_d.htm](http://www.perret-optic.ch/Soleil_ozone_yeux/soleil_ozone_yeux_d.htm)  
Version 2008; Abruf 07.2008
- [Pont 2001] Pont, Michael J.: Patterns for time-triggered embedded systems  
Version 2001 ; Verlag: Addison-Wesley Longma ; ISBN 0-201-33138-1

- [Sweeney 1992] Sweeney, Peter : Codierung zur Fehlererkennung und Fehlerkorrektur  
Übersetzt von Andreas Höfler  
Version 1992 ; Verlag: Hanser ; ISBN 3-446-16439-1
- [TUK 2008] TU Kaiserslautern : Ambient Intelligence  
<http://www.eit.uni-kl.de/ami/frame.html?de>  
Version 2008 ; letzter Abruf 08.2008
- [Wischy 2001] Wissen, Michael ; Wischy, Markus Alexander ; Ziegler, Jürgen :  
Realisierung einer laserbasierten Interaktionstechnik für Projektionswände  
<http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2001/V26.pdf>  
Version 2001 ; letzter Abruf 09.2008
- [Wittman 2008] Wittman, Todd : Fundamentals of Image Processing  
[http://www.math.umn.edu/~wittman/hw1\\_p7b.html](http://www.math.umn.edu/~wittman/hw1_p7b.html)  
Version 2008; letzter Abruf 08.2008

## A. Spezifikation der Infrarot Diode

Wellenlänge	880nm
Durchlaßstrom $I_f$	100mA
Spannung $U_f$ max	3V
Abstrahlwinkel	$\pm 40^\circ$
Strahlungsintensität	
max	6.3 mW/Sr
min	3.15 mW/Sr

Tabelle A.1.: Spezifikation Osram SFH48P

## **B. Inhalt der CD**

Diese CD beinhaltet diese Arbeit in Form eines PDFs und den Quellcode des Empfangssystems sowie den Quellcode des Senders. Zusätzlich befindet sich IC Imaging Control 3.0 auf der CD.

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 16. September 2008

---

Ort, Datum

---

Unterschrift