

# **Implementierung einer Rückfahrkamera mit Einblendung von Trajektorien für das Rückwärtsfahren mit einachsigen Anhänger**

Studienarbeit  
im Studiengang Computervisualistik

vorgelegt von  
Martin Schönfeld

Betreuer: Dipl.-Inform. Uwe Berg, Institut für Softwaretechnik, Fachbereich 4  
Erstgutachter: Prof. Dr. Dieter Zöbel, Institut für Softwaretechnik, Fachbereich 4

Koblenz, im Oktober 2006

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ziel der Arbeit . . . . .	1
1.3	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Kinematik</b>	<b>3</b>
2.1	Kinematisches Modell eines Fahrzeugs mit einachsigen Anhänger . . . . .	3
2.2	Grenzen eines Lenkassistentensystems . . . . .	5
2.3	Arten von Trajektorien . . . . .	5
2.4	Visualisierung von Trajektorien . . . . .	8
<b>3</b>	<b>Lenkassistentensystem</b>	<b>9</b>
3.1	Sensorik im Versuchsfahrzeug . . . . .	9
3.2	Ausgangssituation . . . . .	10
3.3	Optisches Lenkassistentensystem unter Einsatz einer Rückfahrkamera . . . . .	12
<b>4</b>	<b>Technische Grundlagen</b>	<b>16</b>
4.1	Rückfahrkamera . . . . .	16
4.2	Softwarearchitektur . . . . .	16
4.3	IEEE1394 Firewire-Video . . . . .	19
4.4	Firewire-Video Bibliothek libdc1394 . . . . .	19
4.5	Graphische Benutzeroberfläche Qt . . . . .	20
<b>5</b>	<b>Implementierung</b>	<b>21</b>
5.1	Firewire-Installation unter Linux . . . . .	21
5.2	OpenGL Widget . . . . .	22
5.3	Videobild-Darstellung . . . . .	23
5.4	Projektion der Trajektorien . . . . .	24
5.5	Kalibrierung . . . . .	25
5.6	Benutzung der Lenkassistentenz . . . . .	28
<b>6</b>	<b>Schlusswort</b>	<b>32</b>
6.1	Zusammenfassung . . . . .	32
6.2	Nutzen der Lenkassistentenz . . . . .	32
6.3	Ausblick . . . . .	34

<b>A Anhang</b>	<b>35</b>
A.1 Technische Daten . . . . .	35
A.1.1 Barebone-PC . . . . .	35
A.1.2 Kamera . . . . .	37
A.2 Begriffe . . . . .	37
<b>Literaturangaben</b>	<b>40</b>

# 1 Einleitung

## 1.1 Motivation

Die Vorwärtsfahrt mit einem einachsigen Anhänger stellt für die meisten Fahrer keine große Schwierigkeit dar. Das Zugfahrzeug gibt die Richtung vor und der Anhänger wird mitgezogen. Jedoch ist das Rückwärtsfahren mit Anhänger für den ungeübten Fahrer oft sehr kompliziert, da die Lenkmanöver für eine bestimmte Richtungsänderung nicht intuitiv sind und ein „Umdenken“ erfordern.

Will man beispielsweise den Anhänger um eine Linkskurve fahren, so muss das Zugfahrzeug zunächst nach rechts gelenkt werden, damit der Anhänger linksherum geschoben wird. Schnell passieren Fehler und ein Unfall mit Sachschaden kann eine mögliche Folge sein. Besonders Fahrer, die nur gelegentlich mit einem Anhänger unterwegs sind, haben bei der Rückwärtsfahrt mit Anhänger Probleme. [siehe BZ05]

## 1.2 Ziel der Arbeit

Die Arbeitsgruppe Echtzeitsysteme der Universität Koblenz-Landau beschäftigt sich u.a. mit dem Schwerpunkt Mobile Systeme und hier insbesondere mit automobilen Systemen. Dabei wurde eine Reihe von Fahrerassistenzsystemen zur Unterstützung der Rückwärtsfahrt von Gespannen entwickelt.

Mit dieser Studienarbeit soll im Rahmen des Projekts EZlenk ein Fahrzeuggespann, bestehend aus einem Golf II und einem einachsigen Anhänger, mit einem optischen Lenkassistenzsystem ausgestattet werden, das den Fahrer bei rückwärtigen Fahrmanövern unterstützen soll. Hierzu soll das Gespann mit einer Rückfahrkamera versehen werden. Es handelt sich dabei um eine Firewire-Kamera, die am Heck des Anhängers zu befestigen ist. Das Bild der Rückfahrkamera soll auf einem Display auf dem Armaturenbrett des Fahrzeugs angezeigt werden. Außerdem soll die Möglichkeit geschaffen werden, eine oder mehrere Trajektorien perspektivisch korrekt in das Bild der Rückfahrkamera einzublenden. Zudem ist zu prüfen, ob Rahmenbedingungen wie z.B. die Fahrzeugbeladung berücksichtigt werden müssen. Die Implementierung verwendet die schon vorhandene

Softwarearchitektur EZauto, die den Rahmen für Automatisierungs- und Assistenzaufgaben bei Fahrzeugen, insbesondere bei Fahrzeugen im Gespann, setzt.

Die Kombination aus Rückfahrkamera und Trajektorieneinblendung erfordert die Beachtung besonderer ergonomischer und computergraphischer Gesichtspunkte. Es ist sicherzustellen, dass die Trajektorien im Bild so positioniert, skaliert und perspektivisch umgerechnet dargestellt werden, dass beim Fahrer der Eindruck entsteht, diese würden auf der Fahrbahn „liegen“. In diesem Aspekt spiegelt sich auch die Forderung nach der Unterstützung beim gezielten Fahren auf ein punktuelles Ziel hin wieder. Außerdem müssen sich die Trajektorien farblich vom Videobild abheben und deutlich zu erkennen sein.

### 1.3 Aufbau der Arbeit

Beginnend mit kinematischen Grundlagen werden in Kapitel 2 verschiedene Trajektorien und ihre Visualisierung vorgestellt.

Weiter in Kapitel 3 werden das vorhandene Lenkassistenzsystem und das Versuchsfahrzeug näher beschrieben. Der hintere Teil behandelt die Neuerung durch den Einsatz einer Rückfahrkamera.

Kapitel 4 beschreibt die technischen Grundlagen, die für diese Arbeit von Bedeutung sind, so dass in Kapitel 5 auf die konkrete Umsetzung und die Benutzung des Systems eingegangen werden kann.

Im Schlussteil der Arbeit wird im ersten Abschnitt ein Fazit gezogen und im Anschluss mögliche Erweiterungen für diese Arbeit aufgezeigt.



Wert und das gesamte Gespann bewegt sich auf konzentrischen Kreisbahnen. Führt das Zugfahrzeug anschließend mit dem gleichen Radlenkwinkel  $\alpha$  rückwärts, bleibt der Einknickwinkel  $\gamma$  des Gespanns erhalten. Somit wird eine Abhängigkeit zwischen dem Einknickwinkel  $\gamma$  und dem Radlenkwinkel  $\alpha$  deutlich, die sich funktional beschreiben lässt. Der Einknickwinkel  $\gamma$  wird in Abhängigkeit von den Fahrzeugabmessungen und dem Radlenkwinkel  $\alpha$  wie folgt berechnet:

$$rza(\alpha) = \frac{lza}{\tan(\alpha)}$$

$$\eta(\alpha) = \arctan\left(\frac{lzk - lza}{rza(\alpha)}\right)$$

$$rzk(\alpha) = \frac{rza(\alpha)}{\cos(\eta(\alpha))}$$

$$\beta(\alpha) = \arcsin\left(\frac{laa}{rzk(\alpha)}\right)$$

Der Einknickwinkel  $\gamma$  ergibt sich aus der Summe der beiden Winkel  $\beta$  und  $\eta$ :

$$\gamma(\alpha) = -(\beta(\alpha) + \eta(\alpha))$$

Umgekehrt lässt sich der Radlenkwinkel  $\alpha$  in Abhängigkeit von den Fahrzeugabmessungen und dem Einknickwinkel  $\gamma$  folgendermaßen berechnen:

$$laa2(\gamma) = \frac{lzk - lza}{\cos(\gamma)}$$

$$rza2(\gamma) = laa2(\gamma) * \sin(-\gamma)$$

$$rza(\gamma) = \frac{lza + laa2(\gamma)}{\sin(-\gamma)} - rza2(\gamma)$$

$$\alpha(\gamma) = \arctan\left(\frac{lza}{rza(\gamma)}\right)$$

Nach Einsetzen und Vereinfachen ergibt sich folgende Gleichung:

$$\alpha(\gamma) = \arctan \left( \frac{lza * \sin(-\gamma)}{laa + (lzk - lza) * \cos(\gamma)} \right)$$

## 2.2 Grenzen eines Lenkassistenzsystems

Bedingt durch die fahrzeugspezifische Beschränkung des Radlenkwinkels  $\alpha$  auf ein Intervall  $[-\alpha_{max}, \alpha_{max}]$ , gibt es zu jedem Gespann mit einachsigen Anhänger einen zu  $\alpha_{max}$  korrespondierenden Einknickwinkel  $\gamma_{crit} = |\gamma(\alpha_{max})|$ . Wird bei der Rückwärtsfahrt dieser charakteristische Einknickwinkel  $\gamma_{crit}$  überschritten, so kann das Gespann nur noch durch Vorwärtsfahren gerade ausgerichtet werden. Des Weiteren besteht die Gefahr, dass bei Überschreitung von  $\gamma_{crit}$  das Zugfahrzeug gegen die Deichsel des Anhängers fährt.

Der Wert  $\gamma_{crit}$  ist ein theoretischer Grenzwert, der in der Praxis deutlich unterschritten werden sollte: Je mehr man sich diesem Winkel nähert, desto länger wird die Fahrstrecke, die benötigt wird um das Gespann wieder in Rückwärtsfahrt gerade auszurichten. Laut [BZ05] hat sich die Faustformel  $|\gamma| < 0,8 * \gamma_{crit}$  als praxistauglich erwiesen.

## 2.3 Arten von Trajektorien

Das optische Lenkassistenzsystem zeigt dem Fahrer im Videobild der Rückfahrkamera Fahrkurven, so genannte Trajektorien, die als Ziel- und Lenkhilfe dienen sollen. Es werden folgende drei Arten von Trajektorien unterschieden (siehe [BZ06]):

- *Zugfahrzeugtrajektorie*

Die Zugfahrzeugtrajektorie beschreibt die Fahrkurve, die das Zugfahrzeug bei einem beliebigen aber bestimmten Radlenkwinkel  $\alpha$  fahren würde. Die Abbildung 2.2 zeigt die Zugfahrzeugtrajektorie als dunkelblaue Fahrkurve. In Abhängigkeit von den Abmessungen des Zugfahrzeugs und dem Radlenkwinkel  $\alpha$  lässt sich die Position der Mitte der hinteren Achse des Fahrzeugs mit der folgenden Gleichung beschreiben:

$$x^2 + y^2 = rza(\alpha)^2 \tag{2.1}$$

Ohne Beschränkung der Allgemeinheit wird der Bezugspunkt der Zugfahrzeugtrajektorie auf die Mitte der hinteren Zugfahrzeugachse gesetzt. Somit beschreibt die obere Gleichung die Zugfahrzeugtrajektorie.

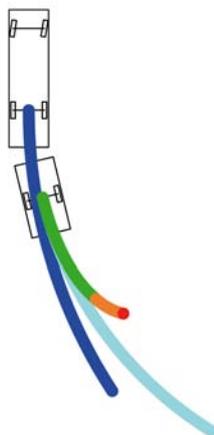


Abbildung 2.2: Aufsicht des Gespanns mit Zugfahrzeugtrajektorie als dunkelblaue Fahrkurve.

- *Anhängertrajektorie*

Die Anhängertrajektorie stellt die Fahrkurve des einachsigen Anhängers bei einem bestimmten Lenkwinkel  $\alpha$  und Einknickwinkel  $\gamma$  in Abhängigkeit von den Abmessungen des Gespanns dar. Die Abbildung 2.3 zeigt die Anhängertrajektorie als grüne Fahrkurve. Die Anhängertrajektorie wird indirekt berechnet. (siehe [ZB00]) Zunächst wird der Winkel  $v$  (siehe Abb. 2.1) in Abhängigkeit von  $u$  berechnet. Abhängig von  $v$  unterscheiden wir zwei Fälle: Falls die Bedingung  $|v| < 2 * \arctan\left(\frac{raa}{rzk-laa}\right)$  gültig ist, wird der Winkel  $v$  mit der folgenden Gleichung bestimmt:

$$v(u) = 2 * \arctan\left(\frac{raa}{rzk-laa} * \frac{e^{\frac{raa}{laa} * u} - 1}{e^{\frac{raa}{laa} * u} + 1}\right) \quad (2.2)$$

Falls die Bedingung  $|v| \geq 2 * \arctan\left(\frac{raa}{rzk-laa}\right)$  gilt, so wird der Winkel  $v$  mit dieser Gleichung berechnet:

$$v(u) = 2 * \arctan\left(\frac{raa}{rzk-laa} * \frac{e^{\frac{raa}{laa} * u} + 1}{e^{\frac{raa}{laa} * u} - 1}\right) \quad (2.3)$$

Beide Fälle resultieren aus der Initialposition des Zugfahrzeugs und des Anhängers. Die Gleichung 2.3 wird benutzt, falls der Anhänger sich innerhalb des Kreises befindet, welcher von der Trajektorie des Zugfahrzeugs beschrieben wird. Andernfalls wird die Gleichung 2.2 benutzt. Die Position des Anhängers kann in Abhängigkeit

von den Fahrzeugabmessungen und den beiden Winkeln  $v$  und  $u$  berechnet werden. Für einen bestimmten Bereich des Winkels  $u$  werden die korrespondierenden x- und y-Koordinaten der Mitte der Anhängerachse mit den folgenden Gleichungen berechnet:

$$x(u, v) = rzk * \cos(u) + laa * \cos(u - v) \quad (2.4)$$

$$y(u, v) = rzk * \sin(u) + laa * \sin(u - v) \quad (2.5)$$

Ohne Beschränkung der Allgemeinheit wird der Bezugspunkt der Anhängertrajektorie auf die Mitte der Anhängerachse festgelegt. Somit beschreiben die beiden oberen Gleichungen die Trajektorie des Anhängers.

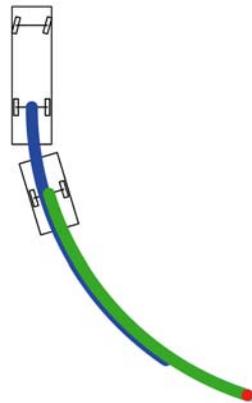


Abbildung 2.3: Aufsicht des Gespanns mit Anhängertrajektorie als grüne Fahrkurve.

- *Gespanntrajektorie*

Bei der Gespanntrajektorie handelt es sich um die Fahrkurve, die den Weg des Zugfahrzeugs und des Anhängers bei konstantem Einknickwinkel  $\gamma$  beschreibt. Die Abbildung 2.4 zeigt die Gespanntrajektorie als türkise Fahrkurve. Unter Beachtung der Abmessungen des Gespanns und des Einknickwinkels  $\gamma$  lässt sich die Mitte der Anhängerachse mit der folgenden Gleichung berechnen:

$$x^2 + y^2 = raa(\gamma)^2 \quad (2.6)$$

Ohne Beschränkung der Allgemeinheit wird der Bezugspunkt der Gespanntrajektorie auf die Mitte der Anhängerachse festgelegt. Somit beschreibt die obere Gleichung die Gespanntrajektorie.



Abbildung 2.4: Aufsicht des Gespanns mit Gespanntrajektorie als hellblaue Fahrkurve.

## 2.4 Visualisierung von Trajektorien

Bei der Fragestellung, welche der drei Trajektorienarten als Unterstützung für den Fahrer angezeigt werden sollen, spielt die Forderung an die Assistenz nach einer Lenk- und Zielhilfe eine entscheidende Rolle. Art bzw. Anzahl, Farbe und Sichtbarkeit der Trajektorien sollen so gewählt werden, dass ein ungeübter Fahrer gezielt, sicher und mit wenig Vorwissen das Gespann mit Hilfe der optischen Lenkassistenz rückwärts bewegen kann.

Zur Auswahl stehen drei Trajektorienarten: Zugfahrzeugtrajektorie, Anhängertrajektorie und Gespanntrajektorie. Es wurden alle drei Trajektorien implementiert mit der Möglichkeit, die Trajektorien einzeln ein- bzw. auszuschalten. Die Erfahrungen mit dem schon bestehenden System sowie zahlreiche Fahrversuche führten dann zu der Wahl der Anhänger- und der Gespanntrajektorie als Darstellungskombination, die den größten Nutzen für den Fahrer verspricht.

Die Farbwahl der Trajektorien spielt eine wichtige Rolle bei der Gestaltung der Mensch-Maschine-Schnittstelle. Die Farben müssen so gewählt werden, dass die Trajektorien sich gut voneinander unterscheiden lassen und sich vom Videobild der Rückfahrkamera optisch abheben. In der Lenkassistenz wurde Blau für die Gespanntrajektorie und Grün für die Anhängertrajektorie gewählt. Die Farbtöne sind satt und leuchtend und somit gut auf dem Display erkennbar.

# 3 Lenkassistentensystem

## 3.1 Sensorik im Versuchsfahrzeug

Um dem Fahrer eines Gespanns eine Lenkassistentz bieten zu können, muss zunächst der Zustand des Zugfahrzeugs und des Anhängers bekannt sein. Dieser Zustand kann anhand des Radlenkwinkels  $\alpha$  und des Einknickwinkels  $\gamma$  bestimmt werden. Speziell dazu wurde in einer früheren Studienarbeit von Jacek Schikora [Sch04] ein System entwickelt, welches über eine Infrarotkamera am Heck des Zugfahrzeugs (siehe Abb. 3.1) ein Fadenkreuz aus Leuchtdioden erfasst (siehe Abb. 3.2) und mit Mitteln der Bildverarbeitung den benötigten Einknickwinkel  $\gamma$  ermittelt.



Abbildung 3.1: Infrarotkamera am Heck des Zugfahrzeugs

Des Weiteren wird auch der Radlenkwinkel  $\alpha$  des Zugfahrzeugs benötigt, um die Anhängertrajektorie in Abhängigkeit des Lenkeinschlags darstellen zu können. Dazu wurde ein

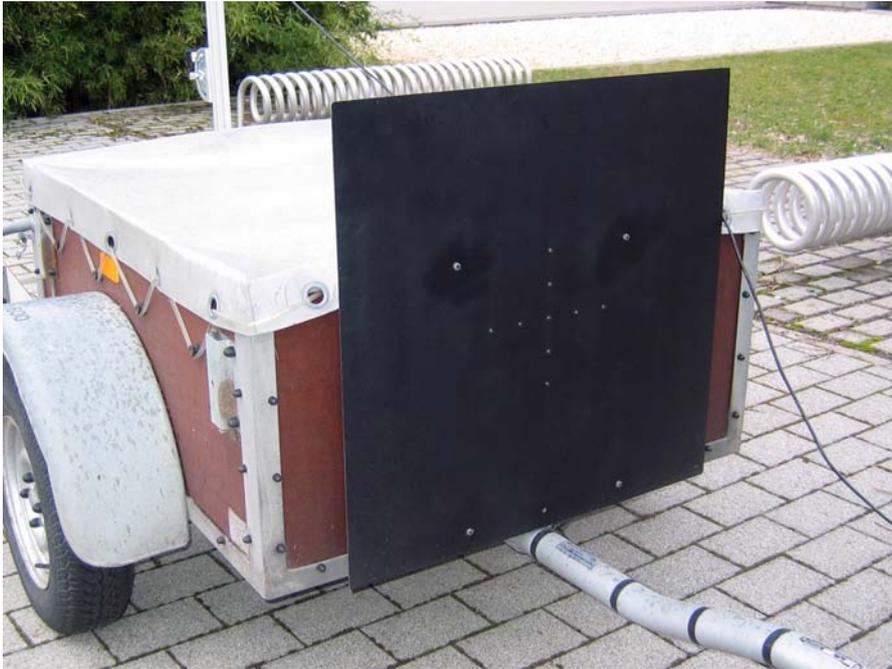


Abbildung 3.2: Fadenkreuz aus Infrarot-Leuchtdioden am Anhänger

Potentiometer an der Lenksäule angebracht (siehe Abbildung 3.3) und die Werte mittels eines Controllers zur Berechnung an den Barebone-PC weitergeleitet.

## 3.2 Ausgangssituation

Zu Beginn der Studienarbeit wurden der Radlenkwinkel  $\alpha$  des Zugfahrzeugs und der Einknickwinkel  $\gamma$  zuverlässig erfasst und auf einem Display, das auf dem Armaturenbrett montiert ist, dargestellt. Dabei wurde eine maßstabsgetreue Draufsicht von Zugfahrzeug und Anhänger auf einem Raster gezeichnet und die Lenkbewegung sowie das Einknicken des Gespanns animiert. Die Trajektorien des Anhängers und des gesamten Gespanns wurden ebenfalls in Abhängigkeit von Lenk- und Einknickwinkel dargestellt. Die Abbildung 3.4 zeigt die beschriebene grafische Darstellung der Assistenz vor der Erweiterung. Ein ungeübter Fahrer konnte mit Unterstützung dieser Lenkassistenz das Gespann geradeaus rückwärts sowie auf einer Kreisbahn sicher bewegen. Dazu musste er beim Rückwärtsfahren durch die Lenkraddrehung die Anhängertrajektorie (grüne Fahrkurve in der Abbildung 3.4) in die gewünschte Position bringen, welche unmittelbar auf die Lenkraddrehung reagierte. Die Änderung der Gespanntrajektorie (hellblaue Fahrkurve in der Abbildung 3.4) wurde dann im Zuge der Rückwärtsfahrt deutlich. Hatte die Gespanntrajektorie die gewünschte Krümmung angenommen, so musste die grüne



Abbildung 3.3: Potentiometer an der Lenksäule des Versuchsfahrzeugs.

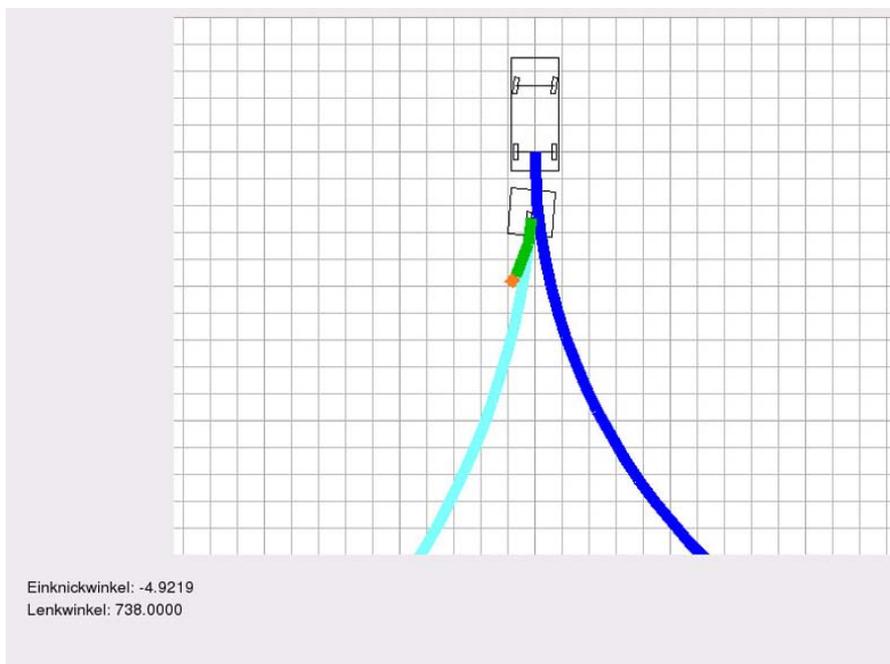


Abbildung 3.4: Animierte Darstellung des Gespanns.

Anhängertrajektorie deckungsgleich über der hellblauen Gespanntrajektorie durch korrigierende Lenkradbewegungen gehalten werden.

Nachteilig bei der Lenkassistenz war die Fahrerablenkung. Der Fahrer musste seinen Blick nahezu permanent auf das Display richten und schaute somit seltener in die Außenspiegel. Er hat die unmittelbare Umgebung hinter dem Gespann nicht wahrgenommen und stellte durch die fehlenden Informationen ein Sicherheitsrisiko dar. Ein weiterer Nachteil war die fehlende Möglichkeit mit dem Gespann ein Fahrziel anzuvisieren, denn der Bezug der Trajektorien zum Fahrzeugumfeld war nicht gegeben.

### 3.3 Optisches Lenkassistenzsystem unter Einsatz einer Rückfahrkamera

Eine Neuerung im bestehenden Lenkassistenzsystem liegt im Einsatz einer Rückfahrkamera am Heck des Anhängers. Dazu wurde eine mobile Halterung für die Kamera gebaut, die am Heck des Anhängers angebracht wird. Die Abbildung 3.5 zeigt die montierte Halterung inklusive Kamera am Anhänger.

Die montierte Kamera läßt sich stufenlos neigen in einem Bereich von circa  $160^\circ$ , ist in der Höhe variierbar und ist mittig zum Anhänger positioniert. Die Halterung ist durch den Einsatz von Aluminiumstangen leicht und einfach zu befestigen. Die Kamera wurde in einer Höhe von 107 cm fest montiert, was in etwa der Positionierung an einem LKW-Anhänger entsprechen würde. Die Neigung der Kamera bezüglich der Fahrbahnebene wurde auf  $9,5^\circ$  festgelegt. Bei der eingesetzten Farbbild Kamera handelt es sich um eine Firewirekamera von Imaging Source mit  $640 \times 480$  Pixel Auflösung (siehe Abbildung 3.6) mit einem Weitwinkelobjektiv. Die Kamera ist über ein 5m Firewire-Kabel mit der aktiven Firewire-Schnittstelle des Barebone-PCs verbunden. Durch die aktive Schnittstelle wird die Kamera mit Strom versorgt und es wird somit kein externes Netzteil für den Betrieb benötigt. Auf dem Display auf dem Armaturenbrett des Versuchsfahrzeugs wird das Videobild der Rückfahrkamera und die Trajektorien des Gespanns und des Anhängers perspektivisch korrekt im Bild angezeigt. Um das Navigieren mit dem Gespann zu erleichtern hat sich die Spiegelung des Kamerabildes als vorteilhaft erwiesen. Das Rückwärtsfahren mit Hilfe des Displays ist so intuitiver, da die Richtung der Lenkbewegung mit der Richtungsänderung der Trajektorien im Display übereinstimmt.

Mit den oben genannten Komponenten bietet das optische Lenkassistenzsystem (siehe Abbildung 3.7) dem Fahrer eine sehr gute Hilfe, das Gespann zielsicher zu manövrieren. Die Drehung am Lenkrad führt zu einer Bewegung der Anhängertrajektorie in eine bestimmte Richtung. Durch die Ablenkung der Anhängertrajektorie in die gewünschte Richtung wird beim Fahren die Fahrkurve des gesamten Gespanns auch in diese Rich-



Abbildung 3.5: Kamerahalterung am Heck des Anhängers.

tung bewegt. Die fortwährende Änderung der Gespanntrajektorie kann dann durch das Lenken der Anhängertrajektorie in Gegenrichtung verlangsamt und schließlich durch das deckungsgleiche Übereinanderlegen gestoppt werden und führt dazu, dass sich Zugfahrzeug und der Anhänger auf konzentrischen Kreisbahnen bewegen. Das System reagiert in Echtzeit und gibt somit unmittelbar eine Rückmeldung über die Trajektorienänderung. Der Fahrer kann so mit der Gespanntrajektorie ein Ziel anvisieren und das Gespann auf Kurs halten. Die Anhängertrajektorie dient dabei zum „Ziehen“ der Gespannkurve in eine bestimmte Richtung.



Abbildung 3.6: Firewire Kamera mit Weitwinkel-Objektiv.

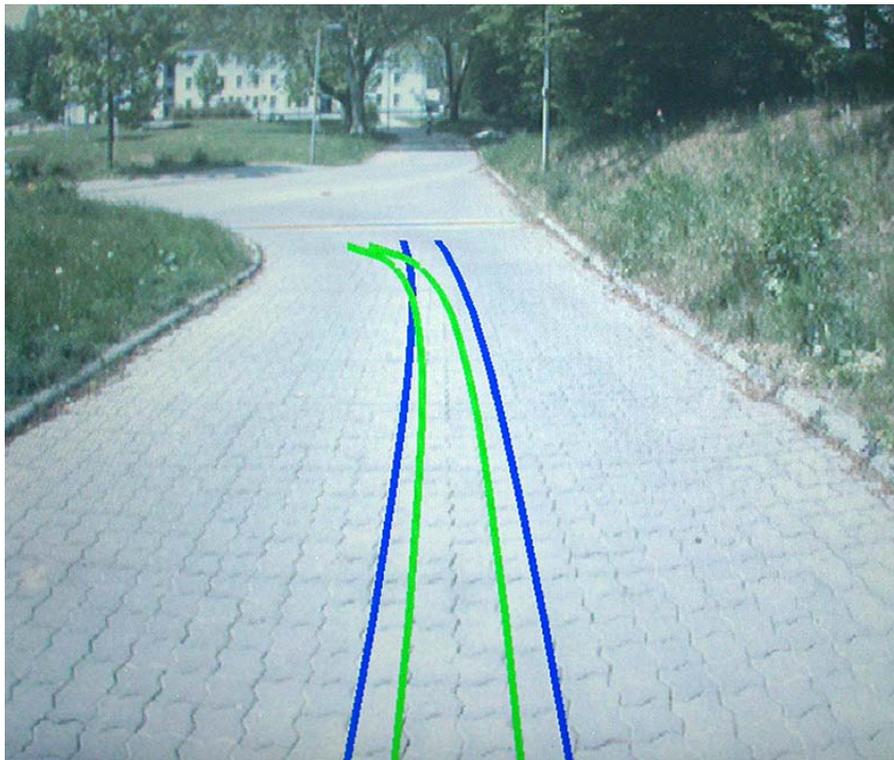


Abbildung 3.7: Anzeige im Display des optischen Lenkassistenzsystems: Gespanntrajektorie (Blau) und Anhängertrajektorie (Grün) im Bild der Rückfahrkamera.

# 4 Technische Grundlagen

## 4.1 Rückfahrkamera

Die perspektivische Darstellung der Trajektorien im Videobild der Rückfahrkamera erfordert Informationen über die Kamera und ihre Position in der realen Welt. Ziel ist es, die computergenerierten Trajektorien so in das Bild zu zeichnen, so dass der Eindruck entsteht, diese würden auf der Fahrbahn liegen. Darüber hinaus müssen auch die Proportionen der vorberechneten Trajektorien mit den tatsächlichen Fahrkurven, die das Gespann fährt, deckungsgleich sein. Nur dann ist auch ein gezieltes Rangieren mit dem Gespann möglich.

Angefangen bei den so genannten äußeren Kameraparametern, wird die genaue Position und der Winkel der Kamera bezüglich der Anhängerachse bzw. der Fahrbahnebene bestimmt. Die Abbildung 4.1 zeigt die Neigung der Kamera bezogen auf die Bildmitte und die Horizontale, die gleichzeitig eine Parallele zur Fahrbahnebene ist. Die Kamera ist um  $9,5^\circ$  nach unten geneigt, so dass der obere Bildrand  $5^\circ$  über den Horizont ragt. Dieser Wert hat sich in der Praxis als günstig erwiesen, denn so wird ein großer Bereich hinter dem Anhänger von der Kamera erfasst ohne die Fernsicht nach hinten zu sehr einzuschränken. Die Höhe in der die Kamera befestigt ist beträgt 107cm.

Zu den inneren Kameraparametern zählen unter anderen der horizontale und der vertikale Kameraöffnungswinkel. Diese Werte sind Winkel, die zwischen der Bildmitte und der linken und rechten beziehungsweise der oberen und unteren Bildkante gemessen werden. Da die Herstellerangaben fehlerhaft sind, mussten die Kameraöffnungswinkel von Hand vermessen werden. Der horizontale Kameraöffnungswinkel beträgt somit  $55^\circ$  und der vertikale  $40^\circ$ .

## 4.2 Softwarearchitektur

Die Implementierung des optischen Lenkassistenzsystems wird in ein vorhandenes Framework eingebettet. Bei dem Framework handelt es sich um eine modular aufgebaute Softwarearchitektur für das assistierte Fahren, deren Einsatz auf verschiedenen Systemen

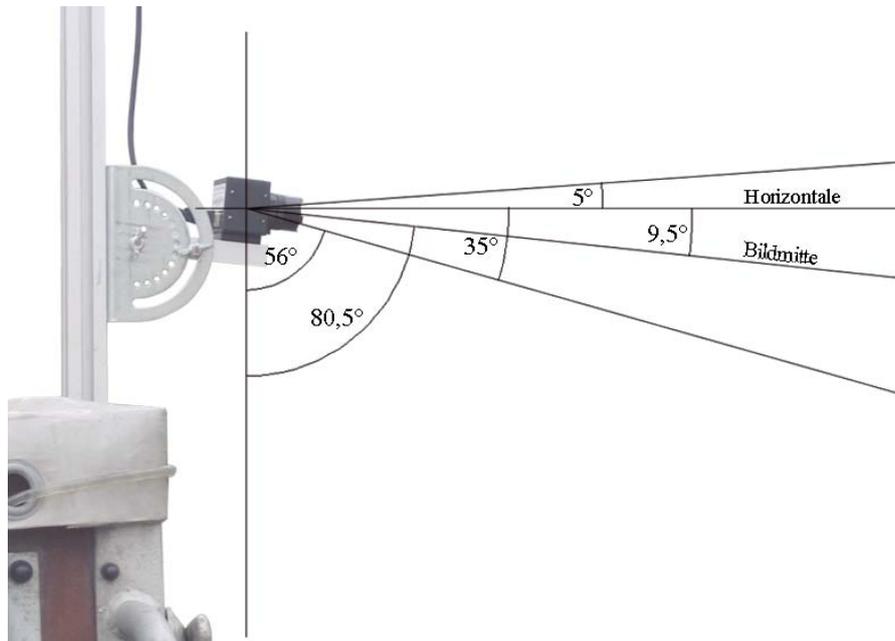


Abbildung 4.1: Neigung der Kamera bezüglich der Horizontalen und der Bildmitte.

durch geringe Modifikation möglich ist. Die bisherige Lenkassistentz wurde als Modul in das Framework eingebunden und wird durch die optische Komponente dieser Studiearbeit erweitert.

Im Datenflussdiagramm (siehe Abbildung 4.2) ist die Erweiterung im linken Teil dargestellt. Die Videodaten der Firewirekamera werden über den Capturing-Prozess als Einzelbilder an den OpenGL Trajektorienprojizierungsprozess weitergeleitet. Dieser ist für die Darstellung des Videobildes und der Projektion der Trajektorien zuständig. Der rechte Teil des Datenflussdiagramms beinhaltet die Bestimmung des Einknickwinkels  $\gamma$  und des Radlenkwinkels  $\alpha$ . Der Einknickwinkelprozess berechnet den Einknickwinkel  $\gamma$  aus den Bilddaten der Infrarotkamera mittels eines Bildsegmentierungsverfahrens. Der Radlenkwinkel  $\alpha$  wird aus dem diskreten Wert des Potentiometers an der Lenksäule, der über die RS232 Schnittstelle übertragen wird, bestimmt. Der Hardware-Prozess stellt die beiden Werte weiteren Prozessen zur Verfügung. Mit diesen Werten als Grundlage werden dann die Trajektorien berechnet und können im OpenGL Trajektorienprojizierungsprozess gezeichnet und schließlich über die GUI zusammen mit dem Rückfahrkamerabild ausgegeben werden.

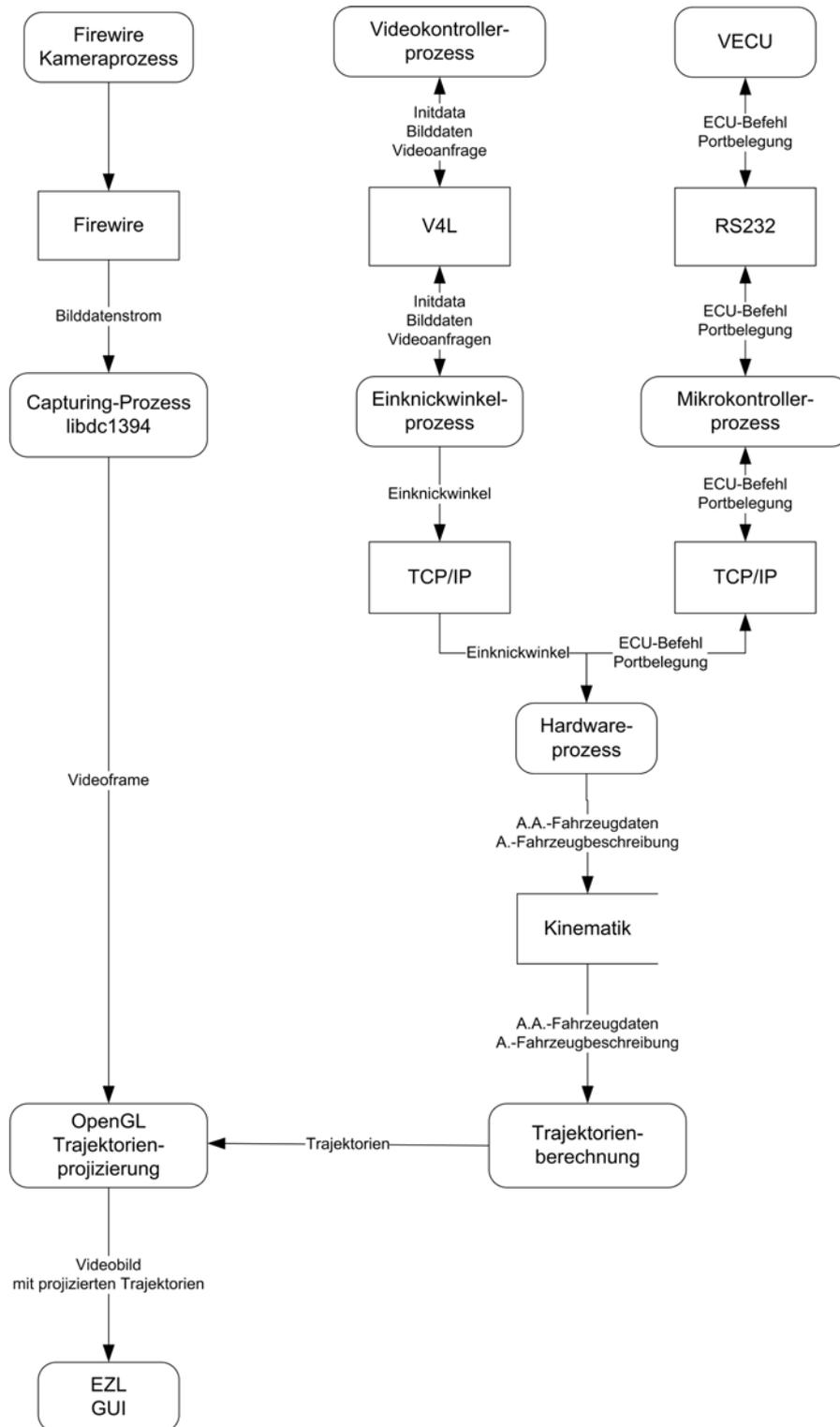


Abbildung 4.2: Datenflussdiagramm des optischen Lenkassistenzsystems.

### 4.3 IEEE1394 Firewire-Video

Firewire, auch als IEEE1394 bekannt, ist ein von Apple entwickeltes Multimedia Bussystem. Es handelt sich dabei um ein Peer-To-Peer System, d.h. es benötigt keinen Host-Rechner. Die Datenübertragungsrate beträgt bis zu 400Mbit/s. Der 6-Pol. Stecker liefert bis zu 1,5A Strom. Somit kommen die meisten Firewire-Geräte ohne zusätzliche Stromspeisung aus. Das Bussystem kann mit maximal 63 Knoten pro Bus mit 4,5 m Abstand betrieben werden. Theoretisch sind dabei bis zu 1023 Busse möglich. [Zel05]

Das Firewire-Bussystem bietet zwei Übertragungsmodi: asynchron und isochron. Im asynchronen Modus gibt es drei Befehle: Read, Write und Lock. Die Adresszugriffe werden dabei als Befehle interpretiert. Der isochrone Modus ist für die Audio/Video-Datenübertragung konzipiert, da der Datentransfer mit garantierter Bandbreite stattfindet und die Daten im Kanal übertragen werden. Allerdings ist beim isochronen Modus keine Fehlerkontrolle vorgesehen. [Zel05]

Unter Linux ist Firewire im Kernel fest implementiert. Der Zugriff erfolgt über so genannte Lowlevel bzw. Highlevel Treiber, die Befehle auf unterschiedlich stark abstrahierten Ebenen anbieten. Highlevel Treiber besitzen keine komplizierten Befehle auf der Protokoll-Ebene und erleichtern somit die Programmierung von Firewire Anwendungen. Das Modul `ieee1394` verwaltet diese Lowlevel und Highlevel Treiber.

Zwei für das Projekt wichtige Highlevel Treiber sind `raw1394` und `video1394`. `Raw1394` bietet direkten Zugriff auf den IEEE1394 Bus über das Device `/dev/raw1394` und `video1394` ermöglicht den isochronen Transfer über DMA.

Des Weiteren werden die Bibliotheken `libraw1394` und `libdc1394` eingesetzt. `Libraw1394` bietet eine einheitliche Schnittstelle zum IEEE1394 Bus und `libdc1394` die Programmierschnittstelle für Firewire Kameras. Letztere wird im folgenden Kapitel näher behandelt.

### 4.4 Firewire-Video Bibliothek `libdc1394`

`Libdc1394` ist eine auf `libraw1394` aufgesetzte API, die zur Steuerung von IEEE1394 Kameras dient. Die Bibliothek wurde in einem Open-Source Projekt für Linux entwickelt und ist bei der GNU als Lesser General Public License (LGPL) lizenziert. [Pro06]

Mit den `libdc1394` Funktionen kann man u.a. Kameras anweisen, mit der Datenübertragung zu beginnen bzw. sie zu beenden und die Kamerafeatures auszulesen. Außerdem bietet diese API die Möglichkeit, die Übertragungsgeschwindigkeit (100, 200 oder 400 MBit/s), die Framerate der Kamera (15 oder 30 Bilder/s) und den Bildmodus (RGB,

YUV411, YUV422, ...) auszuwählen.

Libdc1394 ist unter Linux ab der Kernelversion 2.4.21 oder jünger lauffähig, da bei früheren Versionen die Firewire Unterstützung instabil oder gar nicht funktionierte. Die Bibliothek bietet mehrere Möglichkeiten der Videoaufnahme, die sich im Bildformat und der Datenübertragungsart unterscheiden. Für die Ansteuerung der Rückfahrkamera wird die schnelle DMA-Datenübertragung im RGB-Bildmodus mit einer Auflösung von 640x480 Pixel pro Bild und einer Bildwiederholfrequenz von 15 Bildern pro Sekunde benutzt.

## 4.5 Graphische Benutzeroberfläche Qt

Qt ist eine Klassenbibliothek für die plattformübergreifende Programmierung graphischer Benutzeroberflächen unter C++.

Qt wird von der norwegischen Firma Trolltech entwickelt, und ist für verschiedene Betriebssysteme erhältlich. Die Klassenbibliothek steht sowohl unter der GNU General Public License (GPL) als auch unter einer kommerziellen Lizenz, welche allerdings nur benötigt wird, falls mit der Umgebung Produkte entwickelt werden sollen, die nicht wiederum unter einer freien Lizenz stehen.

Qt arbeitet mit sogenannten Widgets. Dieser Begriff wird unter Unix für Fenster verwendet. Nahezu alles, was man auf einer in Qt geschriebenen Oberfläche sieht, ist ein Widget: Buttons, Laufbalken, Dialogboxen, ... . Widgets können ihrerseits wieder Subwidgets enthalten. In Qt ist ein Widget ein Objekt einer Klasse, die von der Klasse QWidget abgeleitet ist. (vgl. [Her04])

Neben der Entwicklung von graphischen Benutzeroberflächen bietet Qt die Unterstützung von OpenGL an. OpenGL (Open Graphics Library) ist eine Spezifikation für eine plattform- und programmiersprachenunabhängige Programmierschnittstelle zur Entwicklung von 3D-Computergrafik, die in der optischen Lenkassistentz zur perspektivischen Darstellung der Trajektorien verwendet wird.

# 5 Implementierung

## 5.1 Firewire-Installation unter Linux

Für den Betrieb einer Firewire-Kamera wird eine IEEE1394-Schnittstelle im System benötigt. Üblich ist eine Onboard-Lösung oder eine zusätzliche Firewirekarte auf dem Mainboard. Unter Linux wird die Benutzung der Firewireschnittstelle ab der Kernelversion 2.4 unterstützt.

Folgende Installationsschritte sind unter Linux durchzuführen:

1. Libraw1394 herunterladen:

Da sich die Bibliotheken für die IEEE1394-Schnittstelle stets weiterentwickeln, sollte man darauf achten die neueste Version herunterzuladen. Libraw1394 ist unter der folgenden Internetadresse zu bekommen:

<http://sourceforge.net/projects/libraw1394>

2. Libraw1394 entpacken und kompilieren:

```
cd /mein_verzeichnis/libraw1394
tar xvfz libraw1394-1.2.0.tar.gz
cd libraw1394-1.2.0
./configure
```

```
//warten bis configure die make Dateien erzeugt
```

```
make
```

```
//Dateien werden kompiliert
```

```
su
```

```
make install
```

```
//Als Superuser anmelden und Installation beenden
```

### 3. Module laden:

Damit auf die Firewire-Schnittstelle zugegriffen werden kann, müssen noch vorher die benötigten Module geladen werden. Dazu ist ebenfalls die Anmeldung als Superuser erforderlich.

```
su
modprobe raw1394
modprobe ohci1394
modprobe ieee1394
```

Nach erfolgreichem Laden der Module, sollten diese nach dem Aufruf von `dmesg` angezeigt werden. Durch das Eintragen der oberen `modprobe ...` Anweisungen in die Datei `/etc/rc.local` werden die Module bei jedem Systemstart automatisch geladen und müssen nicht erneut von Hand eingegeben werden.

### 4. Testen mit testlibraw:

Das mitgelieferte Programm `libraw1394-1.2.0/src/testlibraw` testet alle erkannten IEEE1394 Adapter und meldet Fehler bzw. listet nach erfolgreicher Installation die Anzahl der angeschlossenen Geräte sowie die Ergebnisse einer versuchten Datenübertragung.

## 5.2 OpenGL Widget

OpenGL ist ein Application Programmer's Interface (API) zur 2D und 3D Grafik Programmierung, entwickelt von SGI [SG06] und auf allen gängigen Plattformen verfügbar. Genormt werden nur die Grafikkommandos. Die Fenster- und Eingabegeräteverwaltung wird dem jeweiligen Windowsystem überlassen.

OpenGL ist ein Standard, keine Implementierung. Die OpenGL-Befehle werden an die Grafikkarte weiter gegeben oder auf der CPU ausführt, wenn die Grafikkarte den entsprechenden Befehl nicht kennt. Aufgrund seiner Plattformunabhängigkeit gehört OpenGL im 3D-Grafik-Bereich mittlerweile zum Industriestandard. Die Qt-Klassenbibliothek, die in dem Projekt zur Fensterdarstellung eingesetzt wird, bietet eine direkte Unterstützung für OpenGL.

Fenster einer graphischen Benutzeroberfläche werden in Qt als Widgets bezeichnet. So existiert für OpenGL ein eigenes Widget, das `QGLWidget`. Von dieser Klasse wird das `EZOpenGLWidget` abgeleitet und die virtuellen Methoden `initializeGL()`, `resizeGL()` und `paintGL()` reimplementiert. Die folgende Auflistung erläutert diese Methoden näher:

- `initializeGL()` - Diese QT spezifische virtuelle Methode wird vor den `resizeGL()` und `paintGL()` Methoden aufgerufen. In dieser Methode werden spezifische Benutzereinstellungen wie z.B. Farbeinstellung der Grafikenster eingestellt oder grafische Primitive erzeugt.
- `resizeGL()` - Diese QT-spezifische virtuelle Methode wird aufgerufen, wenn die Fenstergröße verändert und dadurch der Inhalt des Grafikensters neu gezeichnet werden muss.
- `paintGL()` - Diese QT spezifische virtuelle Methode ist für das Zeichnen der Grafikobjekte zuständig. Sie wird immer dann ausgeführt, wenn das Grafikenster aktualisiert oder die Fenstergröße verändert wurde.

Innerhalb der `initializeGL()` Methode wird die Datenübertragung der Firewire Kamera über `libdc1394` initialisiert und gestartet. Das Aufnehmen und Anzeigen des aktuellen Bildes (Frame) in OpenGL wird in `paintGL()` realisiert und im nächsten Kapitel näher behandelt. Des Weiteren werden in dieser Methode die Trajektorien gezeichnet. Die Befehle, mit denen die Trajektorie gezeichnet wird, werden speziell in der Klasse `EZOpenGLZeichenflaeche` mit OpenGL-Zeichenroutinen implementiert. Bei den Zeichenbefehlen handelt es sich um grundlegende Operationen, wie beispielsweise das Zeichnen eines Polygons in einer bestimmten Farbe und Strichdicke. Die Softwarearchitektur nutzt diese grundlegenden Operationen und ist somit unabhängig vom Ausgabemedium. So müssen nur die tatsächlichen Zeichenbefehle für das jeweilige System implementiert werden.

### 5.3 Videobild-Darstellung

Die Videobilder der Firewire Kamera werden innerhalb der Klasse des OpenGL-Widgets ausgelesen und in OpenGL dargestellt. Dazu wird zunächst die Kamera in der OpenGL Methode `initializeGL()` initialisiert und anschließend die Datenübertragung gestartet. Die Initialisierung der Kamera erfordert unter anderem die Angabe der Bildauflösung, des Bildformates sowie der Bildwiederholrate. Diese Angaben hängen von der eingesetzten Kamera ab.

Nach dem Start der Datenübertragung werden permanent Bilder von der Kamera in den DMA-Ringbuffer geladen und können angezeigt werden. Dazu wird bei jedem Aufruf der OpenGL-Methode `paintGL()` vor dem Zeichnen der Trajektorien das aktuelle Videobild aus dem Ringbuffer geholt und in der OpenGL 3D-Welt dargestellt. Dies geschieht, angetrieben durch einen Timer, alle 30ms. Da die Kamera eine Bildwiederholrate von 15 Bildern/Sekunde liefert, wird ca. alle 60ms ein neues Bild zur Verfügung gestellt. Nach

dem Abtasttheorem (siehe [Sha]) folgt, dass das Bild doppelt so oft angefordert werden muss wie es von der Kamera geliefert werden kann, um die gewünschte Bildwiederholfrequenz zu erreichen. Die Methode `dc1394_dma_single_capture()` der Bibliothek `libdc1394` stellt dabei das aktuellste Bild aus dem Speicher zur Verfügung (vgl. [Sch06]). Da es sich dabei um ein zweidimensionales Bild handelt wird dieses, vergleichbar mit einem Hintergrundbild, hinter die 3D-Szene auf einer 2D-Ebene gezeichnet. Dazu wird die OpenGL-Methode `void glRasterPos4f()` eingesetzt, die eine 3D-Position, die so genannte Rasterposition, in Fenstekoordinaten abspeichert. Die Rasterposition wird zur Positionierung von Pixel- und Bitmap-Schreibvorgängen genutzt. Die OpenGL-Methode `glDrawPixels()` liest die von `dc1394_dma_single_capture()` bereitgestellten Pixel-daten aus dem Speicher und schreibt diese in den Framebuffer relativ zur aktuellen Rasterposition. Gemeinsam mit den projizierten Trajektorien wird dann das aktuelle Videobild im OpenGL-Widget gezeichnet.

### 5.4 Projektion der Trajektorien

Die Trajektorien werden in einer virtuellen dreidimensionalen Welt gezeichnet, die von einer virtuellen Kamera beobachtet wird. Diese virtuelle Kamera muss die gleichen inneren und äußeren Parameter besitzen wie die reale Kamera. Kameraöffnungswinkel, Position und Neigung der virtuellen Kamera sind also identisch mit den Werten der realen Kamera. Die betrachtete virtuelle dreidimensionale Szene muss dann mittels der perspektivischen Projektion auf einer zweidimensionalen Ebene umgerechnet werden, die dann in das Videobild der Rückfahrkamera gezeichnet werden kann.

In diesem Schritt werden die Koordinaten der rotierten und transformierten Trajektorien aus dem 3D-Raum auf eine 2D-Fläche projiziert. Die einzelnen Schritte der Rotation und Transformierung der Trajektorien werden im Kapitel 5.5 ausführlich behandelt.

Bei der Projektion kann man zwischen orthogonaler und perspektivischer Projektion unterscheiden. Bei der orthogonalen Projektion bleibt die Größe der Objekte unabhängig von der Entfernung erhalten. Dagegen ist die perspektivische Projektion realistischer und entspricht der Sicht der Rückfahrkamera. Bei ihr werden die Objekte mit zunehmender Entfernung kleiner, was durch einen pyramidenstumpfförmigen Ansichtsbereich erreicht wird. (siehe Abbildung 5.1)

Der Befehl, um in OpenGL eine perspektivische Projektionsansicht zu erstellen, ist `gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far);` Es wird dabei eine Matrix für eine symmetrische perspektivische Projektion erzeugt, die mit der aktuellen Matrix multipliziert wird. `Fovy` ist der vertikale Sichtwinkel in der `yz`-Ebene, dessen Wert im Bereich von 0.0 bis 180.0 Grad liegen muss. Diese entspricht

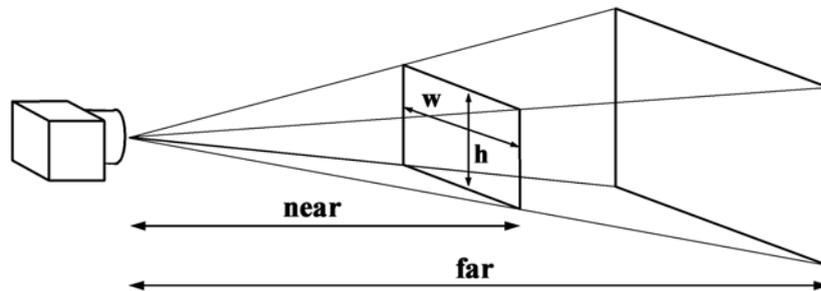


Abbildung 5.1: Pyramidenstumpfförmiger Ansichtsbereich der perspektivischen Projektion.

dem horizontalen Öffnungswinkel der Rückfahrkamera und beträgt 40.0 Grad. **Aspect** ist das Seitenverhältnis von Breite  $w$  zu Höhe  $h$  des Pyramidenstumpfes. **Near** und **far** bezeichnen den Abstand von der vorderen und hinteren Begrenzungsebene und müssen daher positiv sein. Alles was außerhalb der Begrenzungsebene ist, wird nicht gezeichnet. (siehe [Tre97])

## 5.5 Kalibrierung

Durch die Kalibrierung werden die äußeren und inneren Kameraparameter der Firewire-Rückfahrkamera auf die virtuelle Kamera übertragen. Dadurch soll erreicht werden, dass die computergenerierte Trajektorie scheinbar im zweidimensionalen Videobild der Rückfahrkamera auf der Fahrbahn erscheint. Des Weiteren sollen die Abmessungen der Trajektorien mit den Fahrkurven des Gespanns übereinstimmen.

Die virtuelle Kamera der dreidimensionalen Szene in OpenGL schaut stets entlang der  $z$ -Achse in die virtuelle Welt. Die Bildebene, in der das aktuelle Rückfahrkamerabild gezeichnet wird, ist so positioniert, dass es exakt im Sichtbereich der virtuellen Kamera liegt und somit auf dem Display die gesamte Fläche einnimmt. Es muss also mittig zur  $z$ -Achse und parallel zur  $x$ - $y$ -Ebene liegen. Außerdem darf sich die Videobildebene nicht mit den Trajektorien schneiden und sollte daher in entsprechender Entfernung zur virtuellen Kamera platziert werden (siehe Abbildung 5.2). Für die genaue Ausrichtung des Videobildes können in der Textdatei `opengl.dat` die drei Koordinaten der Bildmitte angegeben werden.

Die Trajektorien werden auf einem Raster mit dem unteren Rand mittig über dem Ur-

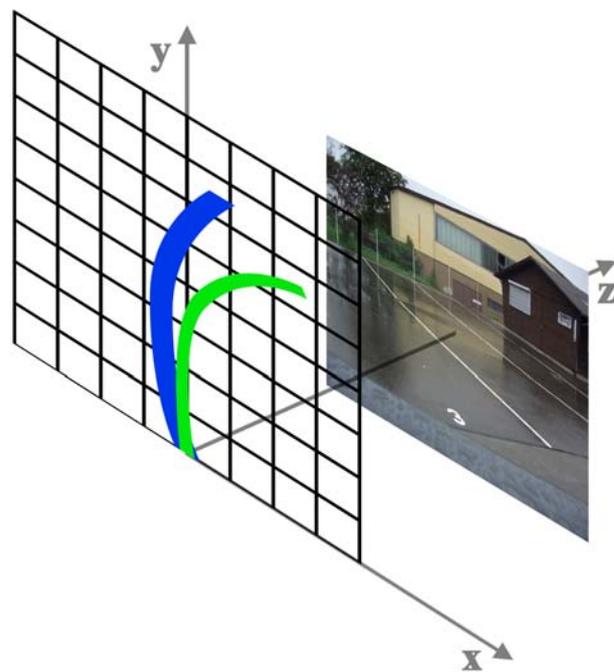


Abbildung 5.2: Videobildebene entlang der  $Z$ -Achse und Darstellung der Trajektorie in der  $X$ - $Y$ -Ebene.

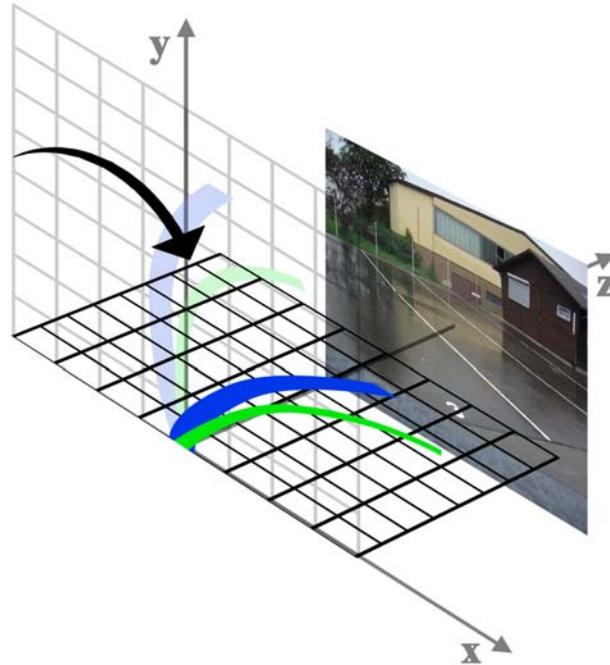


Abbildung 5.3: Rotation der Trajektorien um die X-Achse.

sprung in der x-y-Ebene gezeichnet. Es ist nicht die gesamte Trajektorie sichtbar, die ab der Hinterachse des Anhängers berechnet wird, sondern nur der Teil ab der Stelle, an der die Rückfahrkamera positioniert ist. Dazu wurde der Bezugspunkt der Zeichenfläche, in der die 10 Meter lange Trajektorie liegt, um 0,8 Meter verschoben. Dieser Wert entspricht dem Abstand der Rückfahrkamera zum Mittelpunkt der Anhängerachse.

Im nächsten Schritt wird die Trajektorie 80,5 Grad um die x-Achse rotiert, so dass sie parallel zur Fahrbahnebene des Rückfahrkamerabildes liegt. Der Rotationswinkel ergibt sich durch die Neigung der Rückfahrkamera um 9,5 Grad bezüglich der Fahrbahnebene. Die Abbildung 5.3 zeigt die rotierte Trajektorie.

Nun müssen die Trajektorien, die noch in der Bildmitte des Videobildes liegen, so weit entlang der y-Achse abgesenkt werden, so dass sie auf der Höhe der Fahrbahn erscheinen. Der Wert, um die die Trajektorien abgesenkt werden sollen, ist aber nicht bekannt und muss ermittelt werden. Dazu wird ein Stab mit einer Länge, die der Position der Rückfahrkamera über der Fahrbahn entspricht, hinter den Anhänger senkrecht zur Fahrbahn aufgestellt. Dabei soll die Oberkante des Stabs in der Bildmitte auf des Rückfahrkamerabildes auf Höhe der Trajektorie erscheinen. Durch das manuelle Absenken der Trajektorie über definierte Tasten auf der Tastatur wird diese in ihrer gesamten Länge

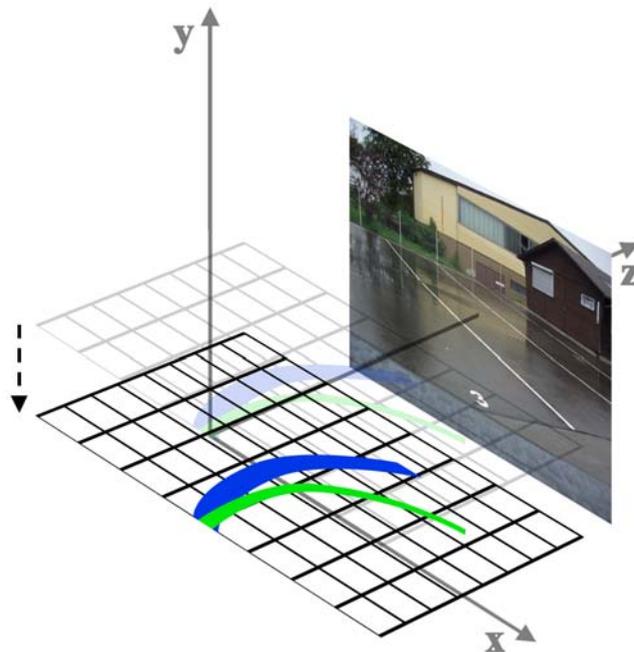


Abbildung 5.4: Verschiebung der Trajektorien in Richtung der negativen Y-Achse.

sichtbar und wird so lange nach unten verschoben bis das Ende über der unteren Kante des Stabes liegt. Die Verschiebung der Trajektorien ist in der Abbildung 5.4 zu sehen.

Im letzten Schritt der Kalibrierung werden die Trajektorien in der Länge und Breite skaliert. Auch hier ist der Skalierungsfaktor unbekannt und muss ermittelt werden. Hier wird das zuschaltbare Raster, auf dem die Trajektorien liegen, zur Hilfe genommen. Die quadratischen Rastersegmente entsprechen einer Kantenlänge von einem Meter bezogen auf die Abmessungen der Trajektorien. Die Zeichenfläche mit den Trajektorien wird nun in der Breite und Länge so weit vergrößert bzw. verkleinert bis sie im Videobild deckungsgleich erscheinen mit einer zuvor erstellten quadratischen Markierung in einem Meter Breite und Höhe auf der Fahrbahnoberfläche. Die Abbildung 5.5 visualisiert die Skalierung in der dreidimensionalen Welt. Die Kalibrierung ist mit diesem Schritt abgeschlossen.

## 5.6 Benutzung der Lenkassistentz

Folgende Schritte sind zum Starten der Lenkassistentz unter Linux erforderlich:

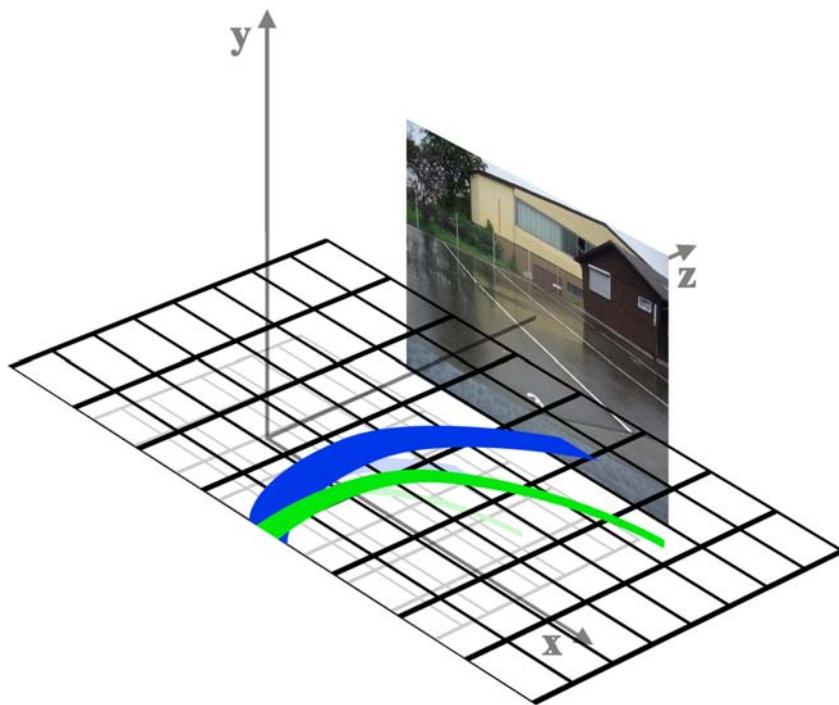


Abbildung 5.5: Skalierung der Trajektorien mit Hilfe der Rasterfläche.



Abbildung 5.6: Einsatz des optischen Lenkassistentensystems im Versuchsfahrzeug.

1. Anmelden als root mit `su root`
2. Laden des Firewire-Treibers mit `modprobe raw1394`
3. Anmelden als ezlenk2 mit `su ezlenk2`
4. Starten der Lenkassistentz mit `./starte`

Abbildung 5.6 zeigt die Sicht des Fahrers im Versuchsfahrzeug.

Nach dem Start sind diese Tastenkürzel verfügbar:

- 'U' Raster anzeigen/ausblenden
- 'A' Anhängerkurve anzeigen/ausblenden
- 'G' Gespannkurve anzeigen/ausblenden
- 'L' Zugfahrzeugkurve anzeigen/ausblenden
- 't' Trajektorienstil ändern (z.Z. 2 Möglichkeiten)
- 'r' Rotationswinkel verkleinern
- 'R' Rotationswinkel vergrößern

- 'X' Trajektorie in positive X-Richtung verschieben
- 'x' Trajektorie in negative X-Richtung verschieben
- 'Y' Trajektorie in positive Y-Richtung verschieben
- 'y' Trajektorie in negative Y-Richtung verschieben
- 'Z' Trajektorie in positive Z-Richtung verschieben
- 'z' Trajektorie in negative Z-Richtung verschieben
- 'q' Programm beenden

# 6 Schlusswort

## 6.1 Zusammenfassung

Mit Hilfe des optischen Lenkassistenzsystems hat der Fahrer die Möglichkeit ein Gespann mit einachsigen Anhänger zu kontrollieren und das zukünftige Verhalten vorzusehen. Die Besonderheit des Systems liegt in der Erkennung möglicher Lenkfehler bevor sich diese auf die Fahrtrichtung auswirken können. Dazu wird das zukünftige Verhalten des Gespanns in Echtzeit visualisiert. Zusätzlich kann der Fahrer ein Ziel, welches angesteuert werden soll, exakt anvisieren.

In der Evaluationsphase wurde das System erfolgreich getestet. Besonders unerfahrene Fahrer konnten das Gespann in alltagsähnlichen Situationen wie dem Rückwärtsfahren um die Kurve oder dem Rückwärtsinparken kontrolliert steuern. Dabei war die Eingewöhnungszeit für das System sehr kurz. Insgesamt führt die optische Lenkassistenz zu einer erheblichen Verbesserung der Fahrmanöver. Teilweise können sogar Schäden am Gespann verhindert werden.

## 6.2 Nutzen der Lenkassistenz

Nach der Implementierung der optischen Lenkassistenz wurde das System von 14 Personen evaluiert. Die Testpersonen gehörten einer heterogenen Gruppe beider Geschlechter im Alter von 21 bis 43 Jahren an. Die Testgruppe bestand aus Fahrerinnen und Fahrern mit und ohne Erfahrungen mit Gespannen, mit normaler Fahrerlaubnis sowie mit LKW Führerschein. Die Evaluation sollte den Nutzen und die Bedienbarkeit des optischen Lenkassistenzsystems prüfen. Dazu mussten die Testpersonen in einem Parcours (siehe Abbildung 6.1) typische Fahrmanöver durchführen:

- Rückwärtsfahrt geradeaus,
- Rückwärtsfahrt durch eine Links- oder Rechtskurve,
- Rückwärtsfahrt an eine Laderampe, oder



Abbildung 6.1: Rückwärtsfahrt geradeaus.

- rückwärts Einparken in verschieden große Parklücken.

Die Fahrmanöver waren sortiert nach dem Schwierigkeitsgrad. Zu Beginn wurden einfache Manöver gefordert und schrittweise die Schwierigkeit erhöht. Während der Testphase wurde die Herzfrequenz der Testfahrer aufgezeichnet. Der Test hat gezeigt, dass die Herzfrequenz und somit der Stress und die Anstrengung bei Benutzung der Lenkassistentz viel niedriger ist. Diese Beobachtung stimmt mit dem subjektiven Empfinden der meisten Fahrer überein.

Ohne die Benutzung der Lenkassistentz sind viele Fahrer in kritische Situationen geraten. Der Anhänger ist zu stark eingeknickt und es bestand die Gefahr den Anhänger oder das Zugfahrzeug zu beschädigen. Mit der Nutzung der Lenkassistentz ist die Anzahl kritischer Situationen deutlich gesunken. Mit der Lenkassistentz brauchten die meisten Testfahrer weniger Platz für die gestellten Fahrmanöver. Durch die Rückfahrkamera konnten die Testpersonen näher an Hindernisse wie eine Laderampe oder das anvisierte Ziel heranzufahren. Des Weiteren wurde die Anzahl der Versuche durch den Einsatz des Lenkassistentzsystems reduziert.

Insgesamt haben Testfahrer mit geringer Fahrerfahrung bessere Resultate erzielt als Fahrer mit langjähriger Fahrpraxis. Fahrer, die das Fahren mit einem Anhänger beherrschen, hatten Probleme ihre Fahrweise an die Lenkassistentz anzupassen. Ohne die Benutzung des Lenkassistentzsystems wurden viele Fahrmanöver abgebrochen, weil die Fahrer mit der Aufgabe überfordert waren. Nach der Aktivierung der Lenkassistentz wurden die Fahrmanöver mit einem guten oder sehr guten Ergebnis abgeschlossen.

### 6.3 Ausblick

Trotz der positiven Evaluation wurde der Wunsch geäußert mehr Informationen auf dem Display anzuzeigen. Beispielsweise wäre die Anzeige der Trajektorien in Form einer Hüllkurve in der Breite des Gespanns vorteilhaft. Dadurch wäre ein exaktes Rangieren zwischen zwei Hindernissen leichter. Hierfür werden Erweiterungen des Systems implementiert und getestet. Auch der Einsatz von Ultraschallsensoren am Heck des Anhängers würde noch mehr zur Sicherheit beim Rückwärtsfahren beitragen. Die Information über den Abstand zu einem Hindernis hinter dem Gespann sollte ebenfalls auf dem Display dargestellt werden. Auch die Ergänzung durch akustische Meldungen wie beispielsweise Warnungen bei kritischem Einknickwinkel oder die Unterstützung beim Rückwärtsfahren auf einer stabilen Kreisbahn sind vorgesehen. Eine automatisierte Kalibrierung des Systems würde die Portierbarkeit auf ein anderes Gespann bzw. den Einsatz einer anderen Rückfahrkamera vereinfachen. Ferner ist auch der Test des Systems bei professionellen LKW-Fahrern geplant. Zudem wäre eine Umsetzung des Systems auch auf zweiachsige Sattelschlepper wünschenswert.

# A Anhang

## A.1 Technische Daten

### A.1.1 Barebone-PC

Als Zentralrechner wird ein schneller Barebone-PC mit kleinen Abmessungen für Sockel-A-CPU's eingesetzt. Er ist im Kofferraum eingebaut und über mehrere verdeckt verlegte Kabel mit der Sensorik und Elektronik im Bereich des Armaturenbrettes verbunden. Der XPC SN41G2V3 wurde von Shuttle entwickelt, um Mittelpunkt des digitalen Lebens zu sein. Daher rührt auch die Wohnzimmeroptik des Gehäuses. Dieses Mini-PC-System bietet gleichermaßen Stil, Funktionalität und Geschwindigkeit. Ob zum Abspielen von digitaler Musik oder zur Videobearbeitung - der XPC SN41G2V3 hat den optimalen Mix aus Größe, Power und Stil. Der XPC SN41G2V3 ist ideal einsetzbar für digitale Medien (Musik, Fotos und Video), für Spiele- und Office-Anwendungen und ist ein kraftvoller Ersatz für Standard-PCs.

#### *Technische Informationen*

- Anwendungen: Home, Media Center
- Gehäuse: G2-Gehäusetyp aus Aluminium, Farbe: silber Laufwerksschächte: 1 x 5,25“, 2 x 3,5“ (davon 1 intern) Abmessungen: 30 x 20 x 18,5 cm (LBH) Gewicht: 3,5 kg netto / 5,0 kg brutto
- Mainboard: Shuttle FN41U (S5011 v1.3), Shuttle-Format, spezielles Design für SN41G2V3 Chipsatz: NVIDIA nForce2 IGP / nForce2 RAID MCP Award V6.0PG BIOS, 4MBit Flash-Speicherchip mit Hardware-Überwachung und ACPI Power-Management Abmessungen: 25,4 x 18,5 cm
- Netzteil: Silent-X 250 Watt Mini-Netzteil, 115/230V kompatibel Anschlüsse: 20-Pol ATX, 4-Pol ATX12V
- Prozessor-Unterstützung: Sockel A (Socket 462) unterstützt AMD Athlon XP, Sempron und Duron Prozessoren mit 400, 333 oder 266 MHz FSB-Frequenz unterstützt Burn Proof Überhitzungsschutz für Athlon XP

- Prozessor-Kühlung: Shuttle I.C.E. Heatpipe-Kühlsystem, 80mm-Lüfter
- Speicher-Unterstützung: 2 x 184-Pin Steckplätze, Dual Channel unterstützt PC3200 DDR400 DDR SDRAM Speicher bis zu einer Gesamtkapazität von 2GB Achtung: DDR400 wird nur in Zusammenhang mit einer zusätzlichen (eingesteckten) AGP-Grafikkarte unterstützt unterstützt keine gepufferten oder ECC-Module
- Steckplätze: 1 x AGP 8X/4X (nur 1,5V) 1 x 32 Bit PCI (v2.2) Integrierte Grafikfunktion für 2 Monitore Northbridge mit integrierterm GeForce4 MX Grafikprozessor bis zu 128MB Shared Memory (UMA) Hochleistungs-256Bit-2D/3D-Beschleuniger nView Dual-Display-Architektur integrierter 300MHz D/A-Wandler max. 1920x1440 Auflösung mit TV-Ausgang 6-Kanal Sound Realtek ALC650 5.1 Kanal Soundfunktion konform mit AC97 v2.2 analog: Line-in/out (6-Kanal), Mikrofon, CD-in, AUX digital: optischer SPDIF Ein- und Ausgang, koaxialer SPDIF Ausgang
- Netzwerk Controller: Realtek 8201BL Fast Ethernet Netzwerk Controller unterstützt die Übertragungsraten 10 / 100 MBit/s unterstützt Wake-on-Lan-Funktion (WOL)
- Firewire-Controller: IEEE1394 Firewire Controller Realtek 8801 mit 2 Firewire-Anschlüssen (1x vorne und 1x hinten) unterstützt 400Mb/s, 200Mb/s, 100Mb/s Datentransferrate
- Anschlüsse Laufwerke: 2 x Serial ATA 150 unterstützt Raid 0+1 2 x IDE ATA 133 1 x Disketten-Laufwerk
- Anschlüsse Vorderseite: Mikrofon Kopfhörer (Line-out) Line-in 2 x USB 2.0 Firewire (4 pol.)
- Anschlüsse Rückseite: 2 x VGA Monitor-Anschlüsse TV-Ausgang (S-Video oder composite) 6-Kanal Audio Line-out (2x rear/front, bass/center) Line-in optischer SPDIF Ein- und Ausgang koaxialer SPDIF Ausgang 1 x Firewire (6 pol.) PS/2 für Maus und Tastatur Netzwerk-Anschluß (RJ45) 2 x USB 2.0 1 x serielle Schnittstelle optionales Wireless-LAN-Modul optionale parallele Schnittstelle
- Weitere Anschlüsse onboard: 4 x USB 2.0 (insgesamt 8x) 1 x CDROM Audio-Eingang 1 x Audio-Hilfseingang (Line-in) 1 x Infrarot-Schnittstelle 3 x Lüfter-Anschlüsse
- Konformität: Dieses Gerät wird als informationstechnische Einrichtung (ITE) der Klasse B eingestuft und ist hauptsächlich für den Betrieb im Wohn- und Bürobereich vorgesehen. Durch das CE-Zeichen wird die Konformität mit den folgenden EU-Richtlinien bestätigt: (1) EMV-Richtlinie 89/336/EWG Elektromagnetische Verträglichkeit (2) Niederspannungsrichtlinie 73/23/EWG Elektrische Betriebsmittel zur Verwendung innerhalb bestimmter Spannungsgrenzen

## A.1.2 Kamera

- Allgemeines Verhalten: Video-Formate @ Bildrate 640 x 480 UYVY @ 15, 7.5, 3.75 fps 640 x 480 Y411 @ 30, 15, 7.5, 3.75 fps 640 x 480 Y800 @ 30, 15, 7.5, 3.75 fps Empfindlichkeit 4 lx bei 1/30s, Verstärkung 0 dB Gamma 0.45, 1 Dynamikbereich ADC: 10 bit, Ausgang: 8 bit SNR ADC: 9 bit bei 25 °C, Verstärkung 0 dB
- Schnittstelle (optisch): Sensor-Spezifikation ICX098BQ [671,36 KB, PDF] Typ Progressive Scan Format 1/4 „, Auflösung H: 640, V: 480 Pixel-Grösse H: 5.6  $\mu\text{m}$ , V: 5.6  $\mu\text{m}$  Objektivanschluss C/CS-Mount
- Schnittstelle (elektrisch): Versorgungs-Spannung 8 bis 30 VDC Stromaufnahme ca. 90 mA bei 12 VDC
- Schnittstelle (mechanisch): Abmessungen H: 50.6 mm, B: 50.6 mm, L: 50 mm Gewicht 195 g
- Einstellungen (man): Belichtungszeit 1/3300 bis 1/30 s Verstärkung 0 bis 36 dB Offset 0 bis 511 Schärfe 0 bis 255 Gamma 0.451 Sättigung 0 bis 200 Weissabgleich -2 dB bis +6 dB
- Einstellungen (auto): Belichtungszeit 1/3300 bis 1/30 s Verstärkung 0 bis 36 dB Offset 0 bis 511 Weissabgleich -2 dB bis +6 dB

(siehe[Sou06])

## A.2 Begriffe

- *asynchrone Übertragung*: Die asynchronen Übertragungsverfahren werden auch als Start-Stop-Verfahren bezeichnet, da vor dem Senden einer Dateneinheit ein Start-Signal geschickt wird, und nach dem Senden ein Stop-Signal. Die Dateneinheit ist hier in der Regel nur ein Zeichen mit einer Länge, die zwischen 5 und 8 Bits beträgt, und die bitseriell über eine Leitung geschickt wird. Das Verfahren wird als asynchron bezeichnet, da während der Übertragung eines Zeichens Sender und Empfänger mit unabhängigen, wenn auch möglichst synchronen Takten arbeiten. (siehe [Kow06])
- *isochrone Übertragung*: Sonderform der asynchronen Übertragung
- *LowLevel Treiber*: Der Low-level-Treiber ist für die Ansteuerung der internen Hardwarechnittstelle zuständig.

- *HighLevel Treiber*: Der High-level-Treiber ist für einen Gerätetyp, z.B. eine Webcam, zuständig.
- *DMA*: Speicherdirektzugriff oder englisch Direct Memory Access
- *Kernel*: Ein Betriebssystemkern oder Systemkern ist der zentrale Bestandteil eines Betriebssystems. In ihm ist die Prozess- und Datenorganisation festgelegt, auf der alle weiteren Softwarebestandteile des Betriebssystems aufbauen.
- *API*: Abkürzung für engl. application programming interface, deutsch: Schnittstelle zur Anwendungsprogrammierung
- *OpenGL*: OpenGL (Open Graphics Library) ist eine Spezifikation für ein plattform- und programmiersprachenunabhängiges API zur Entwicklung von 3D-Computergrafik.
- *Framebuffer*: Der Bildspeicher bzw. Framebuffer ist Teil des Video-RAM und entspricht einer digitalen Kopie des Monitorbildes.
- *Ringbuffer*: Warteschlangen mit je einem Zeiger auf Anfang und Ende.

# Literaturverzeichnis

- [BZ05] Uwe Berg and Dieter Zöbel. Haptische Lenkassistentz zur Unterstützung der Rückwärtsfahrt von Fahrzeugen mit einachsigen Anhängern. *Fortschritts-Bericht VDI, Reihe 22, S.131 ff*, 2005.
- [BZ06] Uwe Berg and Dieter Zöbel. *Visual Steering Assistance for Backing-Up Vehicles with One-axle Trailer*. Department of Computer Science, University Koblenz-Landau, 2006.
- [Her04] Helmut Herold. *Das Qt-Buch: portable GUI-Programmierung unter Linux/Unix/Windows*. Nürnberg, SuSE-Press, 2004.
- [Kow06] Prof. Dr. W. Kowalk. Rechnernetze. <http://einstein.informatik.uni-oldenburg.de/rechnernetze>, Abruf: August 2006.
- [MW04] M. Mitschke and Wallentowitz. *Dynamik der Kraftfahrzeuge*. Springer Verlag, 2004.
- [Pro06] Open-Source Projekt. 1394-based DC Control Library. <http://sourceforge.net/projects/libdc1394/>, Abruf: Juni 2006.
- [Sch04] Jacek Schikora. *Winkelbestimmung zwischen Zugmaschine und Anhänger*. Studienarbeit, Universität Koblenz-Landau, Campus Koblenz, 2004.
- [Sch06] Johann Schoonees. libdc1394 FAQ. [http://kauri.auck.irl.cri.nz/~johanns/libdc1394/libdc1394\\_FAQ.html](http://kauri.auck.irl.cri.nz/~johanns/libdc1394/libdc1394_FAQ.html), Abruf: Februar 2006.
- [SG06] Inc. Silicon Graphics. OpenGL. <http://www.sgi.com/products/software/opengl/>, Abruf: Juni 2006.
- [Sha] Claude Elwood Shannon. Communication in the Presence of Noise. *Proc. IRE, Vol. 37, No. 1*.
- [Sou06] The Imaging Source. Produktspezifikation. <http://www.1394imaging.com/de/products>, Abruf: August 2006.
- [Tre97] Mary Treseler. *OpenGL Programming Guide*. Addison-Wesley Publishing Company), 1997.

- [ZB00] D. Zöbel and E. Balcerak. Präzise Fahrmanöver für Fahrzeuge im Gespann. *Autonome Mobile Systeme 2000 (S. 148-156)*, 2000.
- [Zel05] Dr. Rolf Zellmer. Technische Grundlagen von IEEE-1394 / Firewire. <http://www.firewire-infos.de>, Abruf: Oktober 2005.