

Demoapplikation Serviceroboter fürs Kinderzimmer – Integration von Bildverarbeitung, Handlungsplanung und Navigation

Diplomarbeit von Oliver Sachse



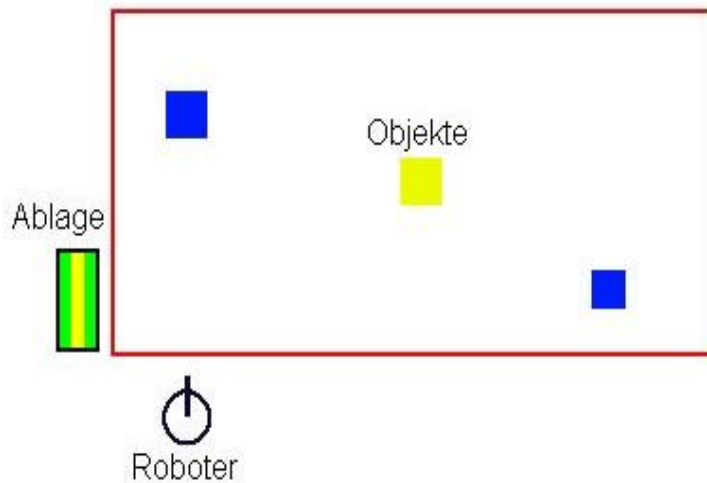
Aufgabenstellung

Zielstellung des Themas ist die Entwicklung einer vollständigen Demoapplikation mit einem realen Roboter für folgendes Szenario: Auf einem Teppichuntergrund liegt verstreut farbiges Spielzeug, das durch einen Roboter eingesammelt und in einer Sammelstelle abgelegt werden soll.

Die Objekte sollen mittels optischer Objekterkennung lokalisiert und gezielt aufgenommen werden. Für die Lösung der Aufgabe ist eine intelligente Verknüpfung von Bildverarbeitung, Handlungsplanung und Navigationsleistungen notwendig.

Besonderes Augenmerk ist auf die Robustheit und die „Gutmütigkeit“ des Systems zu legen.

Das Szenario



Teppich:

- Rechteckbildende Linien
- Farbe *ROT*
- Größe: ca. 1.3x3 m

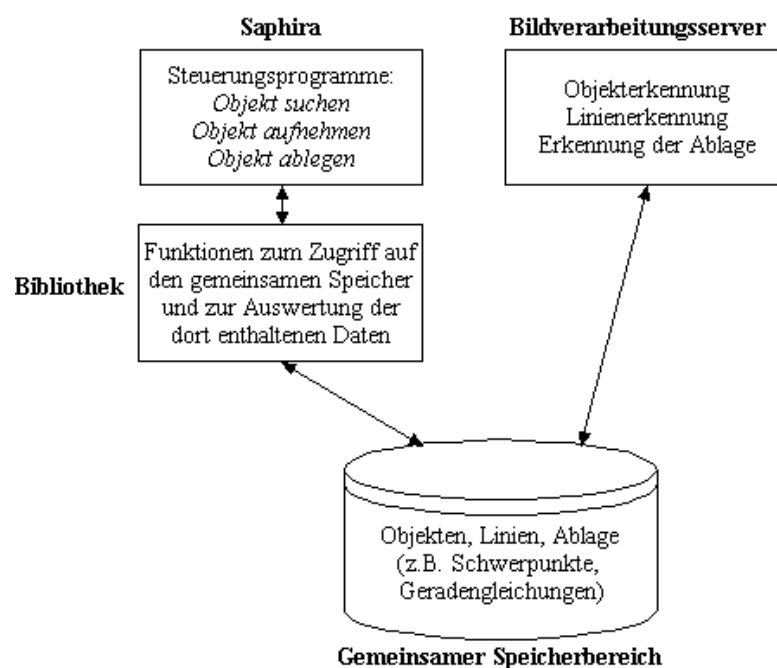
Objekte:

- LEGO-Duplo-Bausteine
- Größe: 4x4x3 cm

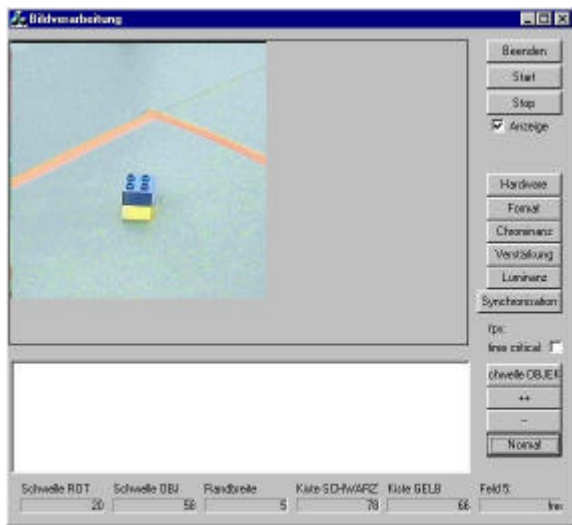
Ablage:

- Farblich markiert
- Größe: 58x23x5 cm

Systemarchitektur



Bildverarbeitung



- Bilder von PCMCIA-Framegrabber
- Parallel zu Saphira
- Thread für Bildverarbeitung
- Einstellung von Schwellwerten
- Anzeige der Zwischenschritte

Aufgaben:

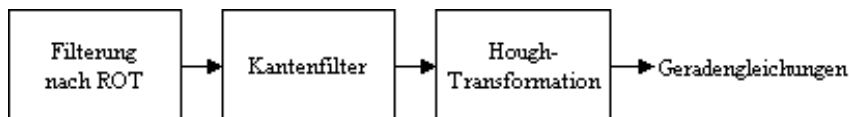
- Erkennung der Teppichbegrenzungen
- Erkennung der Objekte
- Erkennung der Markierung der Ablage
- Daten im gemeinsamen Speicher ablegen

Erkennung der Teppichbegrenzung

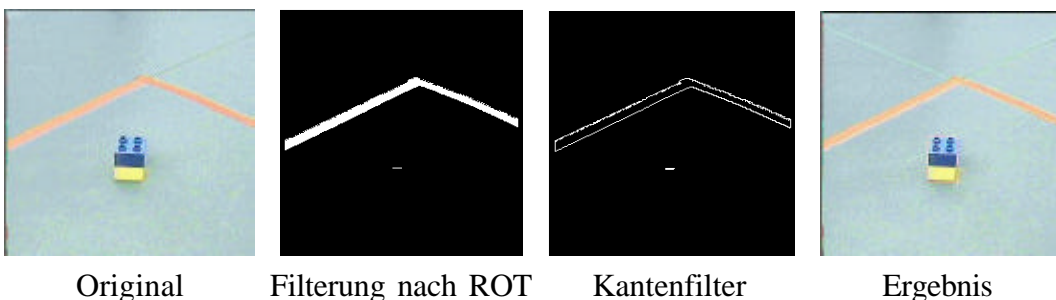
Eigenschaften

Farbe ROT, rechteckbildende Linien

Verarbeitungsschritte



Beispiel

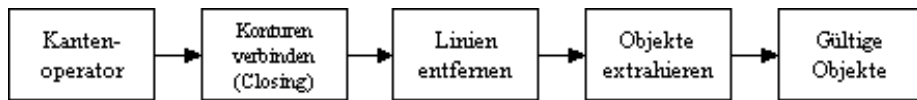


Erkennung der Objekte

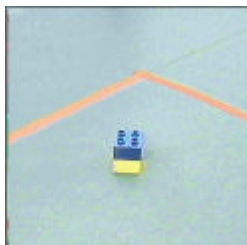
Eigenschaft

Heben sich farblich vom Untergrund ab

Verarbeitungsschritte



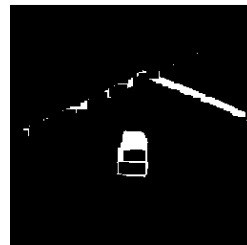
Beispiel



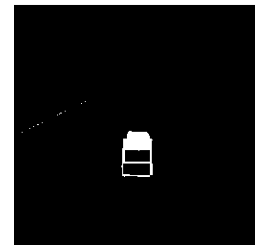
Original



Kantenfilter



Closing



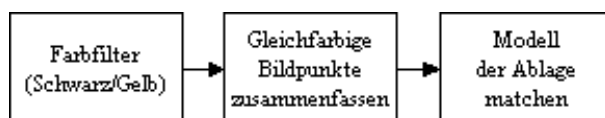
Linien entfernt

Erkennung der Ablage

Eigenschaft

Schwarz-Gelb-Schwarze Markierung an der Teppichseite

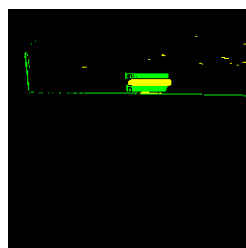
Verarbeitungsschritte



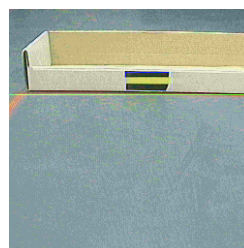
Beispiel



Original



Kantenfilter



Markierung eingetragen

Robotersteuerung

Zur Lösung der Aufgabe Weltwissen notwendig, z.B.
Objekt gegriffen, Roboterposition, Punkt-Artefakte, Suche beendet?

Architektur:

- Vordefinierter Plan
- Reaktive Anteile (z.B. Objekt greifen)

Implementierung:

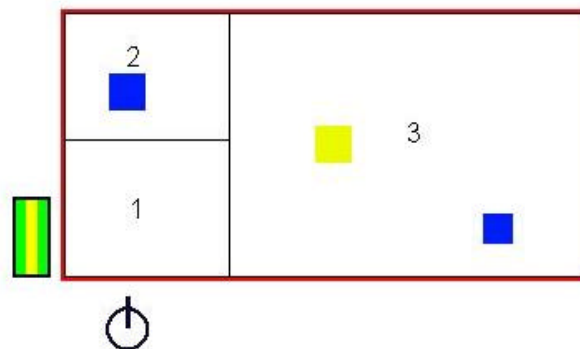
- Colbert-Activities
- Bibliotheken (Zugriff auf gemeinsamen Speicher, Kamera und Greifer)

3 Haupt-Phasen:

1. Nächstes Objekt suchen
2. Objekt aufnehmen
3. Objekt ablegen

1. Objektsuche

Aufteilung des Teppichbereiches in 3 Suchgebiete



- Kamerabewegungen
(hauptsächlich während der Fahrt)
- Bibliotheksfunktionen zur Suche im Bild
- Überwachung der Begrenzungslinien

2. Objekt aufnehmen

Vorgehensweise:

- Langsames Zufahren auf das Objekt
- Objekt zentriert im Bild (Kamera- und Roboterbewegung)
- Abschliessende Vorwärtsbewegung mit Orientierung an Greiferlichtschranken

Probleme:

- Zuverlässigkeit der Lichtschranken
- Feststellen, ob Objekt im Greifer

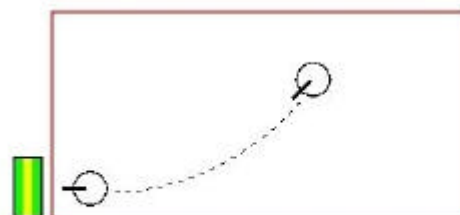
3. Objekt ablegen

Ablauf:

- Zu einem Punkt vor der Ablage navigieren
- Objekt ablegen

2 Ansätze zu Navigation zur Ablage:

- **Odometriebasiert:**
 - Über gespeicherte Punkte (Artefakte)
 - Ausrichten an Ecklinien
- **Bildverarbeitungs basiert:**
 - optische Erkennung der Ablage
 - P-Regler; Ziel: gekrümmte Bahn



Probleme bei der Implementierung

- Bild-Qualität
- Zuverlässigkeit der Lichtschranken
- rotate – Befehl
- Status von Activities

- **Aktualisierung der globalen Roboterkoordinaten**

Robustheit

Definition?

- Toleranz gegenüber Störungen
- Planinherente Störungen: Was kann alles schiefgehen?
- Systeminherente Störungen: Was geht zusätzlich schief? (Feststellen durch Tests)

Beispiele:

- Berücksichtigung mehrerer Sensorquellen (Zeit, Odometrie, Bildverarbeitung)
- „Genauer Hinschauen“
- Kurzzeitige Bildstörungen
- Ausrichten an Linien (während Suche, vor Ablage)

Testergebnisse:

- Bsp.: 2x8 Testläufe mit jeweils 10 Objekten: 100% u. 75% erfolgreich
- Fehlerursachen:
 - Beschaffenheit der Objekte (Farbe, Form, Material)
 - Begrenzungslinien nicht gefunden (Bildstörung?)
 - Verzögerte Befehlsausführung (Kamera)
- Kaum gefährliche Ausfälle

Zusammenfassung und Ausblick

Was wurde erreicht?

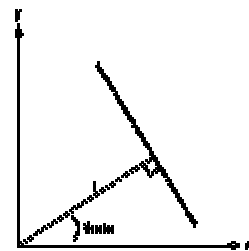
- Vollständige Demoapplikation entwickelt
- Hohe Zuverlässigkeit (Tests);
Voraussetzung: eingehaltene Rahmenbedingungen

Erweiterungsmöglichkeiten

- Gesamtgeschwindigkeit des Systems
- Erstellung und Benutzung einer Karte
- Bildverarbeitung (z.B. Klassifizierung von Objekten)

Hough-Transformation

- Geradengleichungen in Hessescher Normalform
- Parameterraum (Akkumulator)
- Ablauf:
Für jeden Bildpunkt, der auf einer Kante liegt, berechne alle möglichen durch ihn laufenden Geraden und erhöhe die zugehörigen Akkumulatorzellen um 1
- Ergebnis:
Liegen mehrere Bildpunkte auf einer Geraden, dann wurde der Zähler für die zugehörigen Geradenparameter entsprechend oft erhöht



$$r = x * \cos(?) + y * \sin(?)$$



Hough-Akkumulator
nach Transformation