

LiesMich

WICHTIG !

Diese Software funktioniert nur zusammen mit einer (vorher installierten) Vision Command Kamera (oder einer Logitech Kamera) und installiertem Spirit.OCX von Lego !!

Wegen des Spirit.OCX kommen nur serielle Infrarot Sender/Empfänger in Frage !! Diese Software funktioniert (noch?) nicht mit einem IR Tower für die USB Schnittstelle !!

Das Spirit.OCX befindet sich auf den RIS 1.0 und RIS 1.5 CDs. Nachdem es in den Windows Ordner (C:\Windows\) kopiert wurde, muß es bei dem System angemeldet werden. Dies funktioniert folgendermaßen:

START→Ausführen... Hier “regsvr32 c:\windows\spirit.ocx” eingeben. Das System meldet die erfolgreiche Anmeldung mit “DllRegisterServer in c:\windows\spirit.ocx succeeded”

Motivation

Bereits seit 1986 forscht das renommierte Forschungszentrum Massachusetts Institute of Technologies (MIT) an einem LEGO-Stein kompatiblen Mikrocomputer, dem “MIT Programmable Brick”. Dieser Mikrocomputer hat LEGO inspiriert, selbst einen Mikrocomputer zu entwickeln, der auf der Forschungsarbeit des MIT basiert. Dieser LEGO-Stein wird als RCX im LEGO MINDSTORMS Robotics Invention System vertrieben.

Seit Ende 2000 gibt es von LEGO unter dem Namen Vision Command auch eine Kamera, die zusammen mit dem RCX für Robotik Aufgaben benutzt werden kann.

Die Idee zu dieser Arbeit entstand, nachdem ich mit Kommilitonen an einem Wettbewerb im Rahmen einer Werbeveranstaltung teilgenommen habe, bei dem es darum ging, die LEGO Vision Command Kamera zusammen mit mehreren LEGO RCX für Robotik Aufgaben einzusetzen. Die zum Lieferumfang der Kamera

gehörende Software ist in der Lage, auf Bewegung sowie auf angelernte Farben zu reagieren. Als Reaktion werden Kommandos an den zu einem Roboter gehörenden RCX übermittelt. Diese Aufgaben erfüllte die LEGO Software allerdings nicht zu unserer Zufriedenheit, da nur noch die Kommandos der Vision Command Software auf dem RCX ausgeführt werden. Eine gesonderte Programmierung des RCX ist nicht mehr möglich, selbst die an den RCX angeschlossenen Sensoren können nicht ausgelesen werden. Diese Unzufriedenheit führte zu der Idee eine eigene Software für diese Kamera und den RCX zu schreiben, die im Rahmen des Robot–Labs der Hochschule für Angewandte Wissenschaften Hamburg eingesetzt werden kann.

Software

Die Software ist für eine ebene, weiße Spielfläche gedacht, über der an einem Stativ hängend die Kamera angebracht ist. Die Beleuchtung sollte homogen, von oben und möglichst ohne Schattenwurf erfolgen. Tageslicht sollte nach Möglichkeit vermieden werden. Diese Bedingungen vereinfachen die digitale Bildverarbeitung und ermöglichen ein recht zuverlässiges Bildverstehen im Bezug auf die verschiedenen Farbflächen.

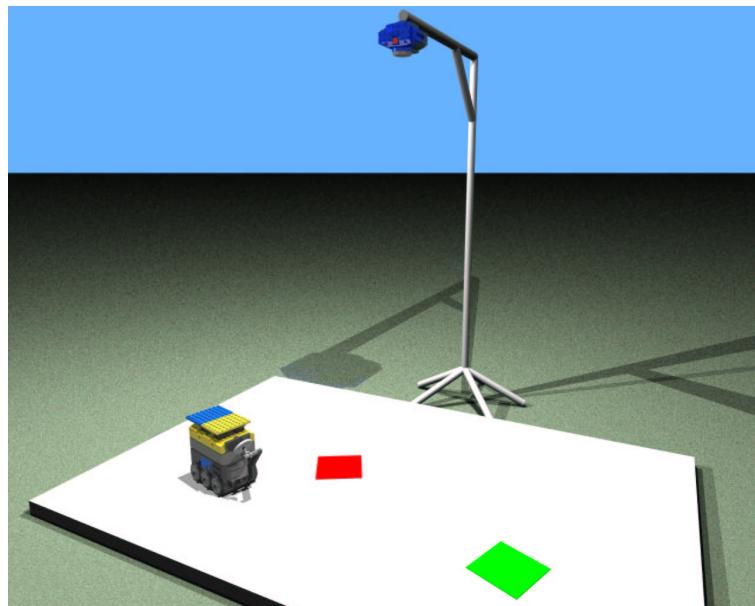


Abbildung 1: Prinzipieller Aufbau des Szenarios

Die Software soll es ermöglichen, ein von der LEGO Vision Command Kamera aufgenommenes Bild so in Informationen zu wandeln, daß sie über die In-

frarotschnittstelle an den RCX geschickt werden können.

Hierbei wird jeweils ein Objekt einer bestimmten Farbe beobachtet. Die zu erkennenden Objekte können von einer der sechs Farben Rot, Grün, Blau, Gelb, Cyan oder Magenta sein.

Flächen dieser Farben werden bestimmt und ihre Größe sowie ihr Schwerpunkt berechnet. Die Größe wird benötigt, um die größte Fläche einer Farbe zu bestimmen, falls mehrere Flächen einer Farbe vorhanden sind. Da nur ein Koordinatenpaar pro Farbe verwaltet wird, werden die Informationen einer kleineren Fläche der gleichen Farbe verworfen.

Die Koordinatenpaare der einzelnen Farbflächen werden bezüglich der Auflösung der Kamera ermittelt und auf Anforderung, per Infrarotkommunikation an den RCX des Roboters geschickt.

In einem RCX mit der Firmware 2.0 gibt es zweiunddreißig persistente Variablen. Dies sind fest adressierte, vorzeichenbehaftete 16 bit Integer und werden von dem auf dem RCX laufenden Programm der Reihe nach vergeben. Diese Besonderheit macht sich die Kamerasoftware beim Speichern der relevanten Größen zunutze.

Eine Notwendigkeit des Programmes, das auf dem RCX läuft ist, daß die ersten dreizehn Variablen eine bestimmte Datenstruktur aufweisen.

In ihnen werden nacheinander die X- und Y-Koordinaten der sechs Farben in der Reihenfolge Blau, Grün, Cyan, Rot, Magenta und Gelb gespeichert. Die dreizehnte Variable dient zur Synchronisation der Kamerasoftware mit der RCX Software.

Diese Variable ermöglicht es, das Senden der Daten zu steuern. So sendet die Kamerasoftware nicht kontinuierlich Daten, sondern nur auf Anforderung. Diese Synchronisation ist notwendig, da das Senden der Daten via Spirit Control bis zu einerhalb Sekunden dauern kann. Fährt der mobile Roboter in dieser Zeit, entsprechen die empfangenen Daten nicht mehr der aktuellen Position. Mit Hilfe der Synchronisations-Variablen hat man die Möglichkeit, den Roboter zu stoppen, die Daten anzufordern und erst nach Ende der Übertragung von der Bildverarbeitungssoftware den Weg (neu berechnet) fortzusetzen. Diese Anforderung obliegt dem RCX.

Die folgende Funktion stammt aus dem Programm, das auf dem LEGO Roboter des Szenarios verwendet wird.

```
void cameracontrol() {  
    Wait (wait_time); // Kamerabild aktualisieren lassen  
  
    blau_x = 0; // Alle Werte auf Null, damit Werte  
    ... // auf alle Fälle neu sind !  
    ...  
    gelb_y = 0 ;
```

```

sync = 1;           // Kontrolle an Kamerasoftware übergeben

while (sync);      // nichts tun solange Kamerasoftware die Kontrolle hat !
}

```

(Das komplette NQC Programm für den Roboter ist unter <http://www.informatik.fh-hamburg.de/~lego> zu finden.)

Diese Funktion wird auf dem RCX jedesmal aufgerufen, wenn neue Daten von der Kamerasoftware angefordert werden.

Zuerst werden die zwölf Variablen, die die Koordinaten aufnehmen, auf Null gesetzt. Dies garantiert, daß die anschließend in diesen Variablen gespeicherten Werte aktuelle Werte sind. Die hier verwendete Variable `sync` entspricht der Synchronisations-Variablen. Der RCX setzt sie auf Eins und wartet darauf, daß die Kamerasoftware diese Variable zurücksetzt.

Grafische Benutzer-Schnittstelle

Die grafische Benutzer-Schnittstelle¹ dient als einzige Möglichkeit des Benutzers zur Interaktion. Microsoft Visual C++ stellt eine Vielzahl an grafischen Elementen zur Erstellung solcher Schnittstellen zur Verfügung. Mit Hilfe dieser Elemente ist der in Abbildung 2 abgebildete Dialog erstellt worden. Im Hinblick auf eine eventuelle Benutzung der Software durch Mitglieder der internationalen LUG-NET Gemeinde, ist dieser Dialog in englischer Sprache verfaßt worden.

Die wichtigsten Elemente des Dialoges sind die sechs Bilder. Durch sie kann der Fortschritt der Bildverarbeitung nachvollzogen werden.

In der Ecke oben links wurde das Benutzer-Interface des VideoPortals platziert. In ihm wird das Echtzeitbild dargestellt. Außerdem kann man über dieses Interface die Kamera betreffende Einstellungen vornehmen.

Rechts neben diesem Interface wurde das Referenzbild angeordnet, das für die Referezbild-Division benötigt wird. Der Knopf unterhalb des Referenzbildes ermöglicht es, ein neues Referenzbild aufzunehmen. Dieses Referenzbild soll der ausgeleuchteten leeren Spielfläche entsprechen.

Das Bild rechts oben stellt das mittels Referenzbild-Division optimierte Bild dar. Der Schieberegler mit der Bezeichnung “normalize factor” entspricht einem Faktor, der notwendig ist, um das korrigierte Bild wieder in Integerzahlen darzustellen. Betrachtet man, wie im Fall dieser Arbeit, eine Szene die Licht absorbiert, liegt dieser Wert nahe dem Höchstwert für das Pixel.

¹üblicherweise als GUI (Graphical User Interface) bezeichnet

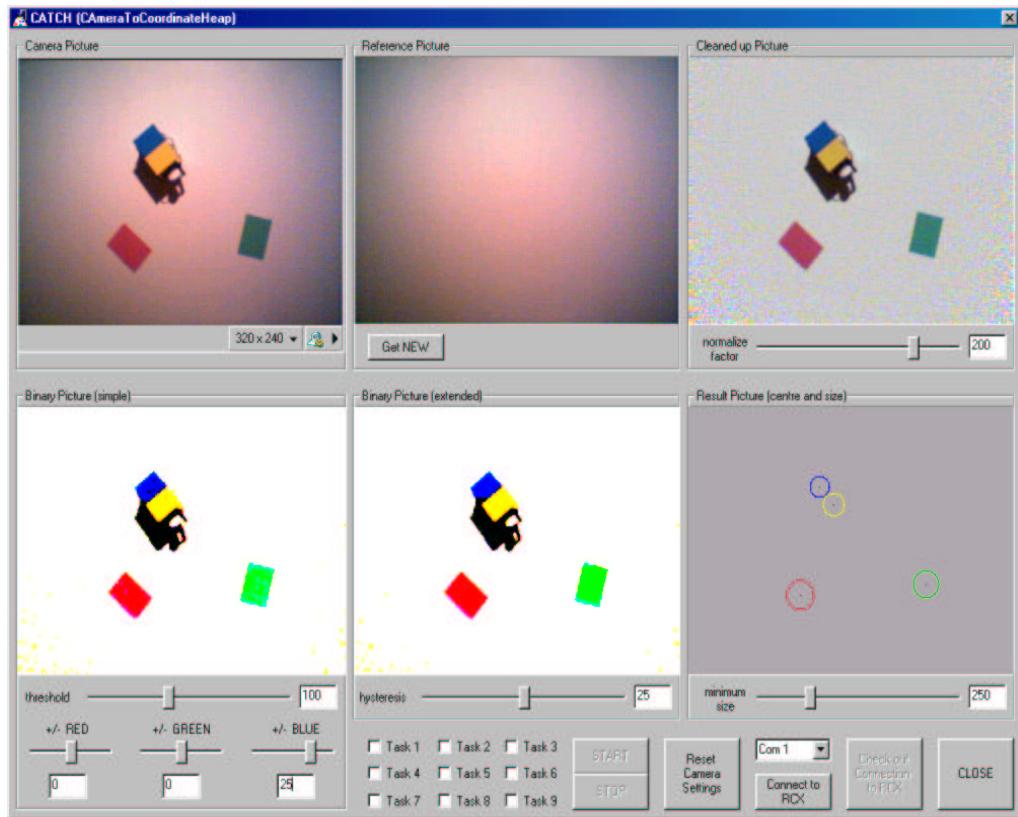


Abbildung 2: Das Programm CATCH (CAmera To Coordinate Heap)

Unten links wurde das Ergebnisbild einer einfachen Binarisierung angebracht. Der zu diesem Bild gehörende Schieberegler "threshold" entspricht dem Schwellwert des verwendeten Verfahrens zur Binarisierung von Grauwertbildern. Die drei unter ihm dargestellten Schieberegler ermöglichen es, die eingestellte Schwelle individuell, für jeden einzelnen der drei Farbkanäle, anzupassen.

In dem Bild in der Mitte unten wird ein erweitert binarisches Bild dargestellt. Über den Schieberegler "hysteresis" kann die Größe der Hysterese des erweiterten Verfahrens der Binarisierung eingestellt werden. Die notwendigen Schwellen entsprechen denen der einfachen Binarisierung. Der einzige neue Parameter ist die Hysterese. Es mag überflüssig erscheinen beide Verfahren der Binarisierung einzusetzen, allerdings hat sich gezeigt, daß sich der Schwellwert am besten mit dem Verfahren der einfachen Binarisierung einstellen läßt.

Das Ergebnis der Merkmalsextraktion ist schließlich in dem Bild unten rechts dargestellt. Hierbei entspricht der Mittelpunkt des Kreises den Koordinaten des Schwerpunktes und die Größe des Kreises ist ein Maß für die Größe der Fläche. Mit Hilfe des Schiebereglers "minimum size" kann eine Mindestgröße der

Farbfläche eingestellt werden, so daß Farbflächen die erkannt werden sollen, eine Mindestanzahl Pixel haben müssen.

Die Schaltflächen unterhalb der genannten Bilder sind in erster Linie für die Kommunikation mit dem RCX zuständig. Über sie läßt sich, falls beim Starten des Programmes keine Verbindung zum Infrarotsender und/oder zum RCX hergestellt worden ist, diese Verbindung herstellen. Die Verbindung zum RCX läßt sich prüfen oder abbauen. Es lassen sich einzelne oder mehrerer Tasks auf dem RCX starten oder stoppen. Eine Schaltfläche dient zum Rekalibrieren der Kameraeinstellungen, eine weitere zum Beenden des Programmes.

Kontakt

Für Fragen stehe ich gerne zur Verfügung.

Entweder an

balze2@gmx.net

oder aber (sogar besser) an

Question.about.CATCH@gmx.net