Ptouch

A paper touch interface

PROJET

MARIO SITZ ANDREAS RUPPEN

Mai 2008

Supervisé par :

Denis LALANE
et
Bruno DUMAS
Groupe Diva

UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Department für Informatik
Universität Freiburg (Schweiz)

UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

Das Projekt PTouch wurde im Rahmen der Vorlesung Multimodale Interface realisiert. Ziel des Projektes war es sich mit den Konzepten der multimodalität, auseinander zu setzen.

Beim Projekt handelt es sich um PTouch, eine Präsentations-Sotware. Normalerweise bringt jeder Referent bei einer Präsentation sein eigenes Laptop mit. Die Probleme sind manigfaltig : der Beamer wird nicht erkannt, das Latop hat keine Batterie mehr oder stürtzt ab. Um diese Probleme zu umgehen haben wir PTouch entwickelt. PToch besthet aus zwei Modulen : einmen Moodul dass es erlaubt PDF-Dokumente auf dem Server abzulgen und eine getaggte Kopie zurückzugeben und ein Präsentations-Modul.

Ein Referent bring somit nur noch einen Ausdruck des getaggten Dokuments mit sich. Die Installation erlaubt es dann das Dokument zu erkennen und die richtige Datei am Beamer anzuzeigen.

Die nächsten Kapitel werden im Detail die Funktionsweise der jeweiligen Module erklären. Zuerst wird die genaue Funktionweise und das Zusammenspiel der Kompomenent erklärt. Dann wird im Detail die Implementation diskutiert.

2 Realisation

Dises Kaptiel widmet sich der Realisation und dem Aufbau der Module aus denen PTouch besteht. Das erste Unterkapitel zeigt die Kommunikation der verschiedenen Module. Das nächste Unterkapitel widmet sich dann der konkreten Implementation.

2.1 Funktionsweise

Wie wir schon ein der Einführunge gesehen haben besteht Ptouch aus mehreren Modulen. Man kann das Projekt einerseit aufteilen in ein Module welches sich um das Taggen und ablegen der Dokumente kümmert und ein Modul welches das getaggte Dokumnet erkennt und das entsprechende PDF am Beamer anzeigt. Beim näheren hinschauen aber besteht das zweite Modul wiederum aus zwei Modulen. Das Eine kümmert sich um die erkennung des Papiers, während das Andere sich um die Identifikation des Dokuments kümmert. Im weiteren Verlauf werden nur die zwei letzen Module disuktiert da nur dise Implementiert wurden. Die Abbildung 2.1 zeigt das Zusammenspiel der verschiedenen Komponenten während einer Präsentation. Der

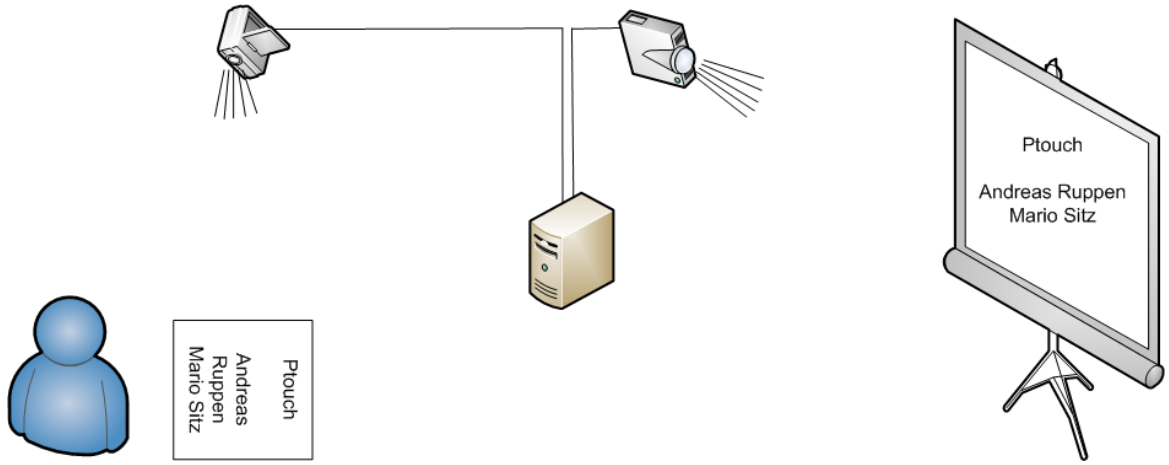

FIG. 2.1: Komponenten von PTouch

Referat kommt mit einer ausgedruckten und getaggten Version von seiner Präsentation. Diese wird von einer Kamera erkannt und auf den Bilschirm angezeigt. Jedes Dokument wird eindeuting durch einen Strichcode identifiziert (2.2.1). Der Strichcode enthält zwei Blöcke aus Zahlen. Der erste Block enthält die Nummer des Dokuments und der Zweite enthält die Seitenzahl. Mit diesem Strichcode ist es der Maschiene möglich, dem erkannten Dokument ein auf dem Server abgeletes PDF zuzuordnen. Damit der Strichcode erkannt werden kann muss zuerst die Position des Blattes bestimmt werden. Diese Aufgabe wird von eimen unabhängigen Module ausgeführt(2.2.2).

Nebem dem Erkennen von Dokumenten ist es auch möglich einen den Mauszeiger auf dem Bilschirm zu erkennen. Möglich wird das durch eine Fingerhut mit einem roten Punkt.

2.2 Implementation

Dises Unterkapitel widmet sich der Implementation der Vorgestellten Module. Das Modul Barcode (2.2.1) kümmert sich um die Interpretation von Strichcodes. Des weiteren sichert es die Kommuikation mit dem PDF Betrachter. Das Modul PTouch (2.2.2) ist für die Erkennung verantwortlich. Beide Teile basieren auf verschiednen Bibliotheken aus der Open-source Welt. Die Kommunikation zwischen den zwei Modulen und mit dem Betriebssystem erfolgt durch das Schnittstellensystem DBus. Aus Performance Gründen wurde die ganze Application in C und C++ geschrieben.

Die folgende Diskussion befasst sich weniger mit Quellcode. Das Ziel der folgenden Kapitel ist es zu verstehen wie die einzelnen Module funktionnieren und zusammenarbeiten. Der interresierte Leser findet jedoch beiliegend einen reichlich dokumentierten Quellcode.

2.2.1 Barcode

Die Strichcodeerkennung wurde mit der Open Source Bibliothek Zebra (0.3 Revision 52) realisiert. Das Programm bekommt eine Bilddatei vom getaggten Dokument. In diesem Dokument wird dann ein Strichcode gesucht. Falls einer gefunden wird, wird dieser interpretiert. Ein Strichcode enthält zwei Informationen, den Dokumentennamen (damit der Strichcode nicht zu lang wird, wurde eine Dokumentennummer eingeführt.) und die aktuelle Seitenzahl. Beide Informationen sind durch ein Minuszeichen getrennt. Beide Informationen werden an das Präsentationsprogramm weitergeleitet. Dies ermöglicht es auch das Dokument während der Präsentation zu wechseln. Weiterhin erlaubt dies, innerhalb eines Dokuments beliebig hin und her zu springen.

2.2.2 PTouch

Damit die Strichcodeerkennung zuervlässig funktionniert muss man wissen wo und wie das Blatt im Bild der Kamera liegt. Zudem muss das PTouch die Dimension de Papiers in der Kamera kennen um zu ermittlen wo der Mauszeiger ist. Damit PTouch die Position und Orientierung des Blattes erkennt, ist das Blatt mit zwei Tags angereichert. Dieses sind, ein grünes und ein gelbes Viereck, die je in einer Ecke das Blattes sind. Jedes Viereck ist schwarz umrandet um den Kontrast zum umliegenden Blatt zu erhöhen. Da Drehungen des Blattes erlaubt sind ist es wichtig zwei verschiedene Tags zu haben. Somit markiert das grüne Viereck die linke untere Ecke, und das gelbe Vierck die rechte obere Ecke des Blattes. Die Farben selber werden nur zur Identifikation eines erkannten Virecks gebraucht, jedoch nicht zur Erkennung selbst. Diese Technik erlaubt es das Blatt zu verschieben und zu drehen. Um die Vierecke zu erkennen wird jedes Bild von der Kamera durch mehrere Filter geschickt. Die Abbildugn 2.2 zeigt die Kette von Filtern an die jedes Bild durchläuft.

Der Referat trägt einen Fingerhut der oben einen roten Kreis hat und unten einen Blauen. Der rote Kreis wird vom Programm erfasst. Da das Programm die Beamer-auflösung, die Dimension und Ausrichtung des Blattes kennt, kann es mithilfe einer affinen Transformation die Koordinationen dieses Punktes auf dem Beamer berechnen. Dieser Punkt entrspricht dann der Stelle wo der Mauszeiger ist. Somit ist es möglich mit dem Finger auf dem Blatt etwas zu zeigen so dass die Zuhörer das Geschehen am Beamer verfolgen können. Alternativ dazu könnte man auch einen handelsüblichen Laser-pointer nehmen. Ist der Finger nun auf einem Link kann man

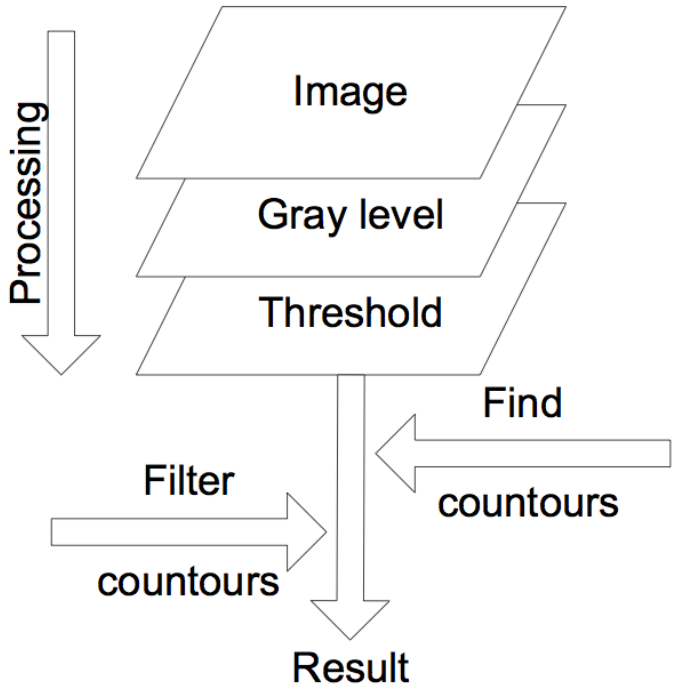

FIG. 2.2: Bild aufbearbeitung in PTouch

den Fingerhut umdrehen, so dass der blaue Kreis nach oben schaut. Erkennt das Program nun den blauen Kreis führt es einen Klick an dieser Stelle aus. Damit kann man zum Beispiel, in einer Präsentation integriertes Videos starten. Die click funktion ist in der aktuellen Version von PTouch noch nicht implementiert. Die Filterkette welche den roten Punkt erfasst ist ähnlich wie in Abbildung 2.2.

Sobald PTouch in einem Bild von der Kamera ein Blatt erkennt, bereitet es dieses auf, speichert es ab und sendet via DBus eine Meldung an Barcode damit dieser dann das Bild weiter verarbeiten kann.

2.3 Upload und Tagging

Es ein Webservice zur Verfügung gestellt, der eine PDF Datei entgegennimmt. Diese Datei wird auf dem Präsentationsserver gespeichert. Danach wird der Datei ein eindeutiger Namen zugeordnet. Mit Diesem Namen wird für jede Seite des Dokuments ein Strichcode mit Barbecue erstellt. Jede Seite wird mit dem dazugehörigen Strichcode und dem günen und gelben Viereck getaggt. Das Resultat wird dem Anwender zurückgegeben. Dieses getaggte Dokument wird für die Präsentation verwendet.

Die Abbildung 2.3 zeigt wie die Tags später auf dem Blatt liegen.

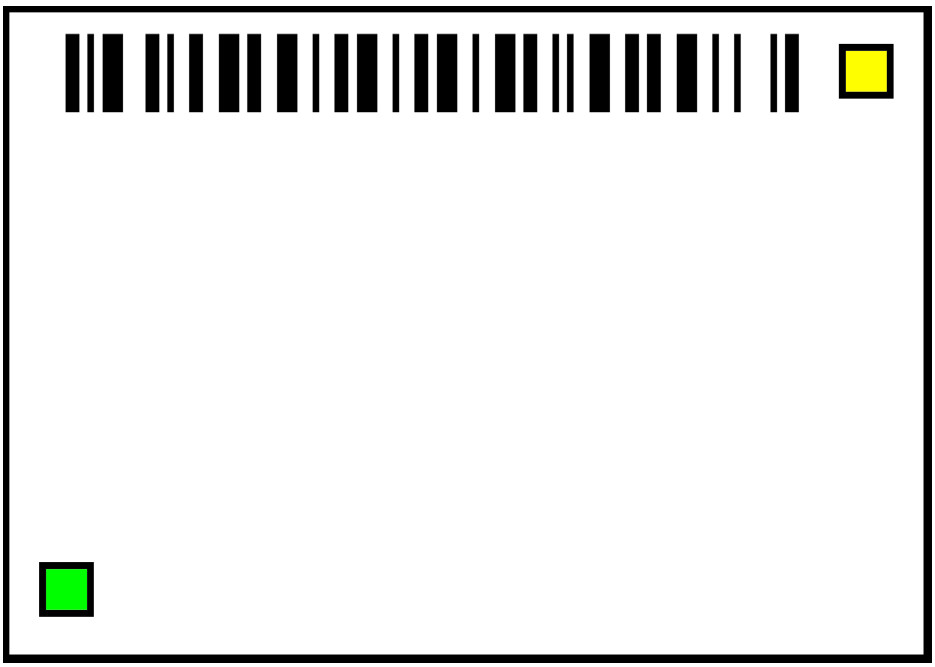

FIG. 2.3: Anordnung der Tags auf dem Blatt

3 Bibliotheken

3.1 Zebra

Zebra (zebra.sourceforge.net) ist eine Open source Bibliothek (LGPL) zu Erkennung von Strichcodes für Linux. Zur Zeit werden EAN/UPC und Code 128 Strichcodes erkannt. Die Bibliothek ist in C geschrieben und bietet zusätzlich einen C++ Wrapper. Als Input kann ein Videostream genutzt werden oder eine Bilddatei. Mit Hilfe des Entwicklers haben wir Zebra auch nach OS X portiert. Dies steht ab der nächsten Version zur Verfügung. Im Subversion ist es ab Revision 52 enthalten.

3.2 OpenCV

OpenCV — Open Computer Vision Library (sourceforge.net/projects/opencvlibrary) ist ein Open source Bibliothek (BSD) zur Bearbeitung von Bildern. Die benutzte Version ist 1.0. Die Bibliothek ist kompatibel mit Linux, OS X und Windows (32-bit). Ziel der Bibliothek ist es, wie ihr Name schon sagt, dem Computer die möglichkeit zu geben, Dinge zu erkennen. Die Bibliothek selber stellt jedoch nur Funktionen zur Bildbearbeitung (thresholding, fitEllipse, usw.) zur verfügung. Open CV enthält HighGUI. Mit Hilfe dieser Bibliothek können primitve Oberflächen programmiert werden, die entweder ein Bild von der Kamera oder Regler zur Einstellung von Werten anzeigen.

3.3 QT

QT Cross-Platform Application Framework (trolltech.com/products/qt) ist ein Plattform übergreifendes Framework. QT ist verfügbar für Linux, OS X, Windows und andere. Qt stellt den Modulen Barcode und PTouch Zugriff aufs Betriebssystem zur Verfügung. PTouch braucht dieses Feature insbesondrer um die Bildschirmauflösung auszulesen und den Mauszeiger zu bewegen. Daneben braucht Barcode QT's Schnittstelle zu DBus (freedesktop.org/wiki/Software/dbus) um die Präsentationssoftware zu steuern. Benutzt wurde im Projekt die Version 4.4 von QT.

3.4 D-Bus

D-Bus ist ein message-passing Protokol. Dieses erlaubt eine schnelle Kommunikation zwischen Applikationen. D-Bus basiert auf IPC (Inter-Process Communication) und RPC (Remote Procedure Call).

3.5 Barbecue

Babecue (barbecue.sourceforge.net) ist eine Open source Bibliothek in Java welche Strichcodes erstellt. Die Bibliothek bietet einen Webservice über JavaEE. Über diesen Webservice können Bilder von Strichcodes heruntergeladen werden.

3.6 PDFTK

Die Tags für die Dokumente wurden mit Gimp (www.gimp.org) erstellt. PDFTK (www.accesspdf.com/pdftk) erlaubt PDF Dokumente einfach zu bearbeiten.

4 Fazit

Das Projekt funktionniert und die Algorithmen sind korrekt. Jedoch leidet das Projekt noch an einigen Schwächen. Eine der Rahmenbedingungen war, dass die Kamera von der Decke aus und ohne Zoom das Blatt filmen muss. Diese Bedingung schliesst die Benutzung von AR-Toolkit aus, weil dieses Framework Tags aus einer so grossen entfernung nicht mehr erkennen kann. OpenCV ist als Framework zwar mächtiger als ARToolkit, stellt aber auf der anderen Seite nur Funktionen zur Bildverarbeitung dar. Die Erkennung der Tags muss hier von Hand implementiert werden. Auch erkennt OpenCV nicht die Drehung der Tags. Deshalb muss dieser Algorithmus auch programmiert werden.

Auf der anderen Seite braucht es auch Hardware von sehr hoher Qualität, vor allem was die Kamera angeht. Je weiter die Kamera vom Blatt entfernt ist, desto mehr Auflösung muss die Kamera zur Verfügung stellen damit eine fehlerfreie Erkennung der Tag und vor allem des Strichcodes möglich ist. Linux schwächelt hier ein bisschen. Webcams werden nur bis zu einer Auflösung von 640x480 Pixel unterstüzt, was viel zu wenig ist um den Strichcode aus über einem Meter richtig zu erkennen. Die andere Möglichkeit ist eine Kamera über Firewire zu benutzen. Jedoch gibt es auch hier Probleme : zum Einen ist die Auflösung einer normalen DV-Kamera kaum höher als 640x480 (es bräuchte also schon eine Kamera mit HD Auflösung) zum Anderen funktionnierte unsere Kamera nicht über Firewire. Da die zebra Bibliothek jetzt unter OS X funktionniert wäre eine andere Lösung das Projekt komplett in OS X aufzuziehen. Jedoch müsste das D-Bus message system durch etwas anderes ersetzt werden, da unter OS X kein Programm D-Bus unterstützt. Ein erfahrener Programmierer sollte mit wenig Aufwand ein Präsentiertool, basierend auf dem PDF Toolkit und Cocoa, schreiben können. Somit hätte das Projekt nur noch die Abhängigkeit nach Zebra, OpenCV und Cocoa.

Da das Tagging der Dokumente nicht unentbehrlich ist für das System (die Dokumente wurde einzel, per Hand getaggt) haben wir diesen Teil aus Zeitgründen nicht implementiert.