

Mobiltelefone als Brücke zwischen der realen und der virtuellen Welt

Maria Husmann

ETH Zürich

Übersicht. Das Mobiltelefon bietet sich als Bindeglied zwischen der virtuellen und der realen Welt an. In dieser Arbeit werden drei optische Technologien (1D-Codes, 2D-Codes und Bilderkennung) vorgestellt, die es Mobiltelefonen ermöglichen, Objekte der realen Welt zu erkennen. Neben der Diskussion der Vor- und Nachteile werden zu jeder vorgestellten Technologie ein oder zwei Anwendungsbeispiele gegeben. Des Weiteren wird auf die sich ergebenden Probleme eingegangen. Diese Arbeit kann somit als eine Zusammenfassung der Papers [1] bis [5] betrachtet werden, die dafür als Quellen gedient haben.

1 Einleitung

Die Möglichkeit, Informationen über physische Objekte in unserer Umwelt abrufen zu können, verbindet die reale mit der virtuellen Welt und ermöglicht eine Vielzahl an Anwendungen. Es gibt verschiedene Methoden, wie dies umgesetzt werden kann. Beispielsweise können diese Objekte mit RFID-Tags¹ versehen werden. Es gibt mobile Geräte, die RFID-Tags auslesen können (beispielsweise gewisse Mobiltelefone). Diese sind jedoch noch nicht so stark verbreitet wie Mobiltelefone mit eingebauter Digitalkamera. Da Mobiltelefone meist in Griffnähe des Benutzers sind, macht es Sinn, sie für das automatische Erkennen von Objekten einzusetzen.

2 Mobile Tagging

In diesem Dokument bezeichnet mobile tagging die Erkennung von physischen Objekten per Mobiltelefon. „Tagging“ heisst übersetzt „markieren“ oder „beschildern“. Dabei gibt es verschiedene Arten von Markierungen. Es können die verbreiteten 1D-Strichcodes benutzt werden. Es gibt aber auch diverse unterschiedliche 2D-Codes. Eine weitere Methode zur Erkennung von Objekten kommt ganz ohne aufgedruckte Markierungen aus, sondern erkennt die Objekte auf Grund ihrer äusseren Erscheinung. Abhängig von der gewünschten Funktion der Anwendung, für welche die Objekte erkannt werden, hat jede Methode Vor- und Nachteile.

¹ Siehe http://en.wikipedia.org/wiki/Rfid_tag

2.1 1D-Codes

Heute tragen nahezu alle Produkte einen Strichcode. Ein weit verbreiteter Standard ist die EAN13 (European Article Number). Die so codierte Nummer ist weltweit eindeutig. Es gibt kommerzielle sowie nicht kommerzielle Datenbanken, die Informationen zu den Produkten sammeln. Beispiele dafür sind SINFOS², Codecheck³ und Wiki-Food⁴. Diese Informationen können umfassender sein, als die Angaben auf der Verpackung des Produktes. Kennt man also die EAN13, ist es einfach das gewünschte Produkt in der Datenbank zu finden.

Die 1D-Strichcodes lassen sich aber nicht nur zur Erkennung einer Produktnummer nutzen. Es ist auch möglich, die Position des Mobiltelefons im Verhältnis zum Code zu bestimmen. So kann das Telefon selbst als Steuergerät genutzt werden. Beispielsweise wählt ein Benutzer einen Menüpunkt durch Drehen des Telefons statt durch Drücken der Tasten aus. Oder er kann ein Spiel durch Bewegen des Telefons bedienen.

Der grosse Vorteil dieser 1D-Strichcodes gegenüber den anderen Möglichkeiten, liegt darin, dass diese Strichcodes schon auf den meisten Produkten aufgedruckt sind. Zu dem gibt es im Gegensatz zu den 2D-Codes weltweite Standards.



Bild 1. EAN13-Strichcode

2.1.1 Beispielapplikation: Allergy Assistant [1]

Der „Allergy Assistant“ ist eine Anwendung, die dem Benutzer mitteilt, ob ein Produkt eine für ihn gefährliche Substanz enthält. Der Benutzer legt dazu ein Profil an, das die Substanzen auflistet, auf die er allergisch reagiert.

Die Bedienung ist simpel. Der Benutzer startet die Anwendung und hält das Mobiltelefon über den Strichcode des zu prüfenden Artikels. Das Gerät erkennt den Code und fordert beim Server die Produktdetails an. Das geschieht alles im Hintergrund und erfordert keine Interaktion mit dem Benutzer. Die Verbindung zum Server wird beim Starten der Anwendung hergestellt. Daher dauert es normalerweise nur kurz, um die gewünschten Informationen zu erhalten. Das erlaubt eine Echtzeitanwendung: Die Rückmeldung zum Produkt wird auf dem Bildschirm mit dem Hintergrund überlagert eingeblendet (siehe Bild 2). Der Benutzer muss folglich nicht ein Bild machen und auf eine Antwort im Browser warten.

² Siehe www.sinfos.com

³ Siehe www.codecheck.ch

⁴ Siehe www.wikifood.lu



Bild 2. Allergy Assistant (Vorgestellt in [1])

2.1.2 Beispielapplikation: Orientation Sensitive Menu [1]

Mit der „Orientation Sensitive Menu“-Anwendung wählt der Benutzer Menüeinträge durch Drehen des Mobiltelefons über dem Strichcode aus. Ein mögliches Anwendungsszenario sieht folgendermassen aus: Der Benutzer hält das Mobiltelefon über den Strichcode einer Musik-CD. Hält er es gerade über den Strichcode, erfährt er den aktuellen Preis der CD. Dreht er nach rechts, fügt er die CD seiner elektronischen Musiksammlung hinzu. Dreht er nach links, kann er sie bei eBay verkaufen.

2.2 2D-Codes

Die 1D-Codes wurden nicht dafür entwickelt, um mit dem Mobiltelefon erkannt zu werden. Als Alternative dazu wurden die 2D-Codes geschaffen, die für den Zweck charakteristische Eigenschaften aufweisen. So lassen sich mit 2D-Codes mehr Daten codieren, sie sind robuster zu lesen und teilweise ästhetischer. Es ist nicht wahrscheinlich, dass die etablierten 1D-Strichcodes auf Produkten von 2D-Codes abgelöst werden, aufgrund verschiedener Faktoren wie etwa den Kosten für eine solche Umstellung oder der aufkommenden RFID-Technologie. Daher empfiehlt sich die Anwendung von 2D-Codes bei Dingen, die sowieso gedruckt werden müssen, wie Poster oder Zeitschriften.

Bei den 2D-Codes hat sich nicht eine einzige Art von Code durchgesetzt. Es gibt viele verschiedene Codes und nicht alle sind standardisiert. Beispiel für 2D-Codes sind Data Matrix, QR-Codes, BeeTag oder der von Michael Rohs an der ETH entwickelte Visual Code.

Es gibt Applikationen für den Visual Code, die mehrere Codes auf einem Bild erkennen. Ausserdem spannt der Code ein Koordinatensystem auf und erlaubt so komplexere Interaktion. Man erkennt, ob das Telefon über dem Code gedreht und geneigt ist.



Bild 3. 2D-Codes. Visual code (links), DataMatrix (Mitte), BeeTag (rechts)

2.2.1 Beispielapplikation: Visual Code Dialer [2]

Für den „Visual Code Dialer“ codiert man eine Telefonnummer als Visual Code und druckt diesen aus, beispielsweise auf eine Visitenkarte. Der „Visual Code Dialer“ erkennt die Telefonnummer und zeigt sie auf dem Display an, sodass sie sofort angerufen werden kann. Erkennt er mehrere Nummern, so wird diejenige angezeigt, die am nächsten der Bildmitte ist.

2.2.2 Beispielapplikation: Visual Code Profile [2]

Die „Visual Code Profile“-Anwendung dient dem Auswählen des Telefonprofils (Lautlos, Meeting, Pager etc.). Dazu wird nur ein einziger Visual Code benötigt. Die Wahl des Profils erfolgt, in dem man mit einem Fadenkreuz einen Eintrag in einer Liste anzielt (siehe Bild 4). Durch das vom Visual Code aufgespannte Koordinatensystem, erkennt die Applikation, was man ausgewählt hat.

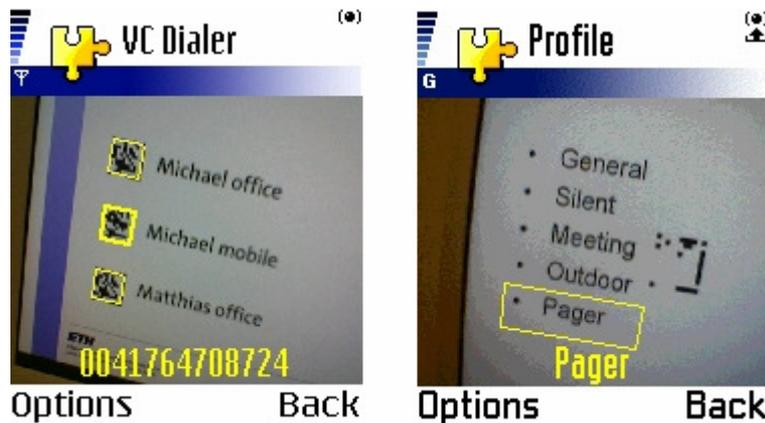


Bild 4. Visual Code Dialer (links), Visual Code Profile (rechts). (Beide vorgestellt in [2])

2.3 Bildererkennung

Eine dritte Variante, ein Objekt zu identifizieren, kommt ganz ohne aufgebrachte Markierungen aus. Dabei wird ein Foto des Objekts geschossen und an einen Server gesendet, der die Bildererkennung vornimmt. Die Bildererkennung kann auch lokal auf dem Mobiltelefon vorgenommen werden, aber dafür sind oft die Ressourcen zu knapp.

Der Erkennungsprozess besteht aus mehreren Schritten. Er basiert auf lokal sichtbaren Merkmalen, der globalen Geometrie und teilweise auch Metadaten wie der Position. Diese kann über GPS oder über die Identifikation der aktuellen Antenne ermittelt werden. Die Position alleine würde aber nicht reichen, um das Objekt zu bestimmen, da viele Objekte am gleichen Ort sein können oder die Position ungenau sein kann.

Diese Methode eignet sich sehr gut für Objekte, bei denen es schwierig ist eine Markierung anzubringen. Beispielsweise bei Gebäuden, die von verschiedenen Seiten oder aus unterschiedlichen Distanzen betrachtet werden, oder in einem überfüllten Museum, wo es schwierig wäre, bis zur Markierung vorzudringen.

2.3.1 Beispielapplikation: Cityguide [4]



Bild 5. Cityguide (vorgestellt in [4])

Eine in dem Paper von Quack, Bay und Van Gool [4] vorgestellte Beispielanwendung, die auf dieser Technologie basiert, ist der sogenannte „Cityguide“. Der „Cityguide“ verbindet Gebäude mit digitalen Informationen. Ein Tourist macht beispielsweise ein Foto vom Kolosseum in Rom und erhält als Antwort einen Wikipedia-Link dazu. Da es sehr viele Sehenswürdigkeiten gibt, ist die Datenbank, auf der gesucht wird, sehr gross, wenn man den Service weltweit anbieten will. Um Skalierbarkeit zu erhalten, wird das zu durchsuchende Gebiet eingeschränkt: Es wird nur nach Objekten gesucht, die geographisch in der Nähe des Benutzers liegen.

Der „Cityguide“ setzt sich aus einer Software auf der Client-Seite und einer auf der Server-Seite zusammen. Der Server umfasst eine relationale Datenbank, in der Metadaten (GPS Position, aktuelle Antenne) und Informationen zu der Sehenswürdigkeit gespeichert sind. Die Bildererkennung ist in C++ implementiert und kann per HTTP aufgerufen werden. Die Client-Seite wurde sowohl in Symbian C++ als auch in Java

implementiert. Die gesamte Bilderkennung erfolgt aber auf dem Server, somit wird das ganze Bild an den Server geschickt. Es existiert aber eine Middleware in PHP und Ruby, die mittels der Metadaten (Position) falls nötig die Anfrage vorverarbeiten kann.

3 Probleme und Lösungen

Je nach angewandter Methode treten verschiedene Probleme auf. Da die Objekterkennung nicht in einer kontrollierten Umgebung, sondern unter realistischen Bedingungen erfolgt, gibt es diverse Schwierigkeiten wie unscharfe Bilder, schlechte Beleuchtung oder Spiegelungen. Eine weitere Schwierigkeit besteht darin, dass es viele verschiedene Mobiltelefone mit unterschiedlichen Fähigkeiten und Software-Plattformen gibt. Daher ist es aufwändig, einen Grossteil der Geräte zu unterstützen. Zudem verfügen Mobiltelefone über weniger Rechenleistung und Speicher als ein herkömmlicher PC und die eingebauten Kameras sind oft nicht optimal.

3.1 Probleme bei 1D-Codes

Auf dem Strichcode müssen bis zu vier verschiedene Dicken von weissen und schwarzen Balken unterschieden werden. Um mit einer herkömmlichen Mobiltelefonkamera ein scharfes Bild von einem Strichcode zu schiessen, braucht man genug Abstand. Dann ist aber der Code im Bild sehr klein. Abhilfe schaffen einerseits Kameras mit Autofokus und andererseits kleine Makrolinsen, die vor der Kamera angebracht werden. Aber auch dies löst nicht alle Probleme. Die Makrolinse verhilft nur in einem bestimmten Abstand zu scharfen Bildern und der Autofokus reagiert manchmal verzögert.

Die Produkte, auf welche die Strichcodes aufgedruckt sind, weisen verschieden Eigenschaften auf, die das Erkennen des Codes erschweren. Sie können z.B. klein, rund oder aus Glas sein. Dies resultiert in Bildern mit Reflektionen, verzerrten Codes und schwierigen Lichtbedingungen. Ausserdem sind beispielsweise bei der Verwendung im Supermarkt weitere Produkte im Hintergrund zu sehen, welche die Bildanalyse beeinträchtigen können.

Will man zusätzlich die Anwendungen unterstützen wie das in 2.1.2 beschriebene „Orientation Sensitive Menu“, muss man auch mit verschwommenen Bildern umgehen können, da der Benutzer dabei das Mobiltelefon bewegt. Durch die Bewegung kann der Code auch aus der Bildmitte verschwinden. In der beschriebenen Anwendung wurde folgende Lösung verwendet:

Es wird mit einem Bild von 640x480 Pixel gearbeitet. Zuerst wird ein Algorithmus zur Kantendetektion ausgeführt. Dadurch entsteht ein schwarz-weiss-Bild. Ein Pixel ist schwarz, wenn der Unterschied zum vorherigen Pixel (in x- oder y-Richtung) einen bestimmten Grenzwert überschreitet. Auf dem entstehenden Bild wird eine Hough-Transformation ausgeführt. Die Hough-Transformation [8] ist eine Standardmethode in der Bilderkennung und hilft gerade Linien, Kreise und Ellipsen zu finden. Hier wird sie verwendet um die Strichcodelinien zu finden. Hat man die parallelen Linien gefunden, lässt sich die Ausrichtung des Codes bestimmen.

3.2 Probleme bei 2D-Codes

Je nach dem welche Codes verwendet werden und wo sie aufgedruckt werden, ergeben sich bei 2D-Codes die gleichen Probleme wie bei den 1D-Codes.

Zugleich gibt es viele verschiedene Codes und es fehlen Standards. Daher gibt es kaum Anwendungen, die alle Codes lesen können. Für den QR-Code und die Data Matrix gibt es jedoch schon ISO-Standards.

Verschiedene Anwendungen haben unterschiedliche Anforderungen an die verwendeten Codes, die schwierig in einer einzigen Art von Code umzusetzen sind:

- Datenmenge: Es sollen viele Daten in einem Code untergebracht werden. QR-Codes beispielsweise verschlüsseln maximal 2953 Bytes [6], im Gegensatz zu nur 284 Bits beim Visual Code [7].
- Ästhetik: Auf Werbepostern beispielsweise soll der Code auch hübsch anzusehen sein. Es werden farbige Codes, integrierte Logos oder BeeTags verwendet. Farbige Codes können aber bei ungünstigen Lichtbedingungen schwieriger zu erkennen sein.
- Zusätzliche Parameter: Bei den im Abschnitt 2.2 beschriebenen Anwendungen reicht die Entschlüsselung des Codes alleine nicht. Es werden weitere Parameter wie die Orientierung des Mobiltelefons relativ zum Code verwendet. Der Visual Code ist für diesen Zweck entwickelt worden. Er verfügt über sog. „Guide Bars“ – Streifen am Rand des Codes – und Elementen zur Korrektur der Verzerrung. Dank einem grossen und einem kleinen „Guide Bar“ können sogar stark geneigte Codes erkannt werden.

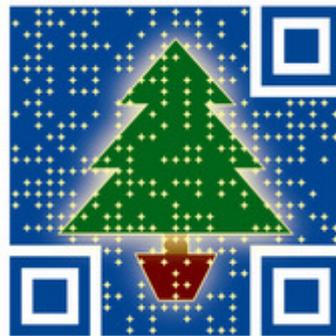


Bild 6. Farbiger QR-Code

3.3 Probleme bei der Bilderkennung

Will man ein Objekt identifizieren, muss das davon gemachte Bild mit einer oft sehr grossen Datenbank verglichen werden. Gleichzeitig soll die Bilderkennung unter realistischen Bedingungen erfolgreich sein. Das kann bedeuten, dass ein kleiner Ausschnitt vom Objekt fotografiert wurde oder dass die Lichtbedingungen nicht ideal sind. Vielleicht wurde das Bild aus der Ferne oder einem seltsamen Winkel aufge-

nommen. Der Hintergrund ist möglicherweise ungünstig. All dies erfordert robuste und rechenaufwändige Algorithmen.

Die Algorithmen können auf dem Mobiltelefon oder auf dem Server implementiert werden. Die Mobiltelefone verfügen aber über weniger Rechenleistung und Speicher. Die Bilderkennung dauert folglich länger. Bei einer Erkennung auf dem Mobiltelefon hätte der Benutzer aber den Vorteil von mehr Datenschutz (da nicht das ganze Bild übertragen wird) und direktem Feedback bei einem ungünstigen Bild, sodass er ein neues schießen kann. Werden die gesamten Bilddaten an den Server geschickt, sind die Kosten für die Datenübermittlung grösser. Zudem skaliert diese Methode schlechter. Je mehr Mobiltelefone sich mit dem Server verbinden, desto stärker wird er ausgelastet. So kann es wiederum zu Verzögerungen kommen.

Zusammenfassung und Schlussfolgerung

Die drei Methoden – 1D-Codes, 2D-Codes und Bilderkennung – zur Verbindung der virtuellen mit der realen Welt empfehlen sich für unterschiedliche Einsatzgebiete. Die 1D-Strichcodes sind vorteilhaft bei Anwendungen mit kommerziellen Produkten, wo sie ohnehin schon aufgedruckt sind. Die 2D-Codes sind sinnvoll bei Objekten, die sowieso neu gedruckt werden wie beispielsweise Poster und Magazine. Die direkte Bilderkennung ohne Markierung findet Anwendung bei dem Erkennen von Objekten, bei denen es schwierig ist, eine geeignete Stelle für einen Marker zu finden, wie beispielsweise Sehenswürdigkeiten oder Strassenschilder.

Probleme ergeben sich aus der geringen Rechenleistung, Speicherkapazität und Kameraqualität von Mobiltelefonen, fehlenden Standards bei 2D-Codes und den vielen unterschiedlichen Ansprüchen. Weitere Schwierigkeiten entstehen durch die Anwendung ausserhalb einer kontrollierten Umgebung (Lichtverhältnisse etc.).

Der Einsatz von Mobiltelefonen als Brücke zwischen der realen und der virtuellen Welt bietet grundsätzlich viel Potential. Das Leben kann für einzelne Personengruppen viel einfacher gemacht werden mit Anwendungen wie dem „Allergy Assistant“. Es entstehen neue Arten der Interaktion mit dem Mobiltelefon durch das Erkennen der Kameraposition relativ zum Code. Da viele Personen ihr Mobiltelefon ständig auf sich tragen, kann man sich unzählige Anwendungen für den Alltag vorstellen.

Referenzen

1. Robert Adelman. Mobile Phone Based Interaction with Everyday Products - On the Go. *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*.
2. Michael Rohs, Beat Gfeller. Using Camera-Equipped Mobile Phones for Interaction with Real-World Objects. In: *Alois Ferscha, Horst Hoertner, Gabriele Kotsis (Eds.): Advances in Pervasive Computing, Austrian Computer Society (OCC), ISBN 3-85403-176-9, pp. 265-271, Wien, Österreich, April 2004*.
3. Michael Rohs. Real-World Interaction with Camera-Phones. *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004). Lecture Notes in*

Computer Science (LNCS) No. 3598, Springer-Verlag, pp. 74-89, Tokyo, Japan, November 2004.

4. Till Quack, Herbert Bay, und Luc Van Gool. Object Recognition for the Internet of Things. *Internet of Things 2008, Zurich, Switzerland*
5. Christof Roduner und Michael Rohs. Practical Issues in Physical Sign Recognition with Mobile Devices. In: *Thomas Strang, Vinny Cahill, Aaron Quigley (Eds.): Pervasive 2006 Workshop Proceedings (Workshop on Pervasive Mobile Interaction Devices, PERMID 2006). pp. 297-304, Dublin, Ireland, May 2006.*
6. QR Code. (2008, March 16). In Wikipedia, The Free Encyclopedia. Abgerufen um 09:51 am 16. März 2008, unter http://en.wikipedia.org/w/index.php?title=QR_Code&oldid=198596219
7. Michael Rohs. Linking Physical and Virtual Worlds with Visual Markers and Handheld Devices. PHD talk an der ETH Zürich, August 2005. <http://www.vs.inf.ethz.ch/res/papers/rohs-phd-2005.talk.pdf>
8. P.Hough. Methods and means for recognizing complex patterns. U.S. Patent 3,069,654, 1962