

Besondere Lernleistung im Fach
Informatik

Thema:

Verfahren zur Erkennung präsentierter Objekte

Verfasser: Joa Diego Ebert
Kurs: Informatik (inf)
Kursleiter: Herr Dr. Carl-Heinz Tillmans
Betreuer: Herr Prof. Dr.-Ing. Franz Kummert
Abgabetermin: 05.04.2006

Abgabe am:

.....

(Unterschrift des Stufenleiters)

Inhaltsverzeichnis

1	Vorwort	3
1.1	Aufgabenstellung	3
2	Verfahren	3
2.1	Bildverarbeitung	3
2.2	Bildverstehen	3
3	Bitmaps und Farben	4
3.1	Binäre Bilder	4
3.2	Graustufen	4
4	Techniken	5
4.1	Filter	5
4.1.1	Dilation	5
4.1.2	Erosion	5
4.1.3	Closing	6
4.1.4	Neighbour	6
4.1.5	Retinex	7
4.1.6	Blobcounter	8
4.2	Künstliche neuronale Netze	8
4.2.1	Aufbau	9
4.2.2	Nutzen	9
5	Durchführung	10
5.1	Hardware/Aufbau	10
5.2	Vorbereitung	10
5.3	Verfahren zur Erkennung von Objekten	10
5.4	Erkenntnis	11

1 Vorwort

Die von mir angefertigte Arbeit im Fach Informatik behandelt die Bildverarbeitung am Computer mit dem Ziel der Objekterkennung. Bei dieser komplexen Thematik ist es nicht möglich alle Lösungsansätze zu verfolgen. Im theoretischen Teil werde ich daher die Grundlagen abhandeln, die notwendig sind, um die praktische Arbeit von mir nachzuvollziehen.

1.1 Aufgabenstellung

Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, welches Gegenstände verschiedener Form und Farbe erkennt. Dabei darf jeweils eine Eigenschaft von zwei Gegenständen identisch sein. Das heißt, die Farbe von zwei Gegenständen kann gleich sein, wenn sie eine unterschiedliche Form haben. Weiterhin sollen Gegenstände beliebig gedreht werden können.

Es stehen dem Benutzer mehrere Objekte zur Auswahl. Es wird immer das untersucht, welches bewegt (präsentiert) wurde.

2 Verfahren

Die gestellte Aufgabe umfasst sowohl die Bereiche der Bildverarbeitung als auch die des Bildverstehens.

2.1 Bildverarbeitung

Bei der Bildverarbeitung geht es hauptsächlich um das Aufbereiten von visuellen Informationen. Diese werden dann weiter analysiert (s. Bildverstehen). Thematik der Bildverarbeitung ist nicht die Objekterkennung, sondern das Anwenden von Filtern, um zum Beispiel Störungen aus einem Bild zu entfernen.¹

2.2 Bildverstehen

Bildverstehen bedeutet nicht viel mehr als die Detektion von Objekten beziehungsweise das Beschreiben ihrer Eigenschaften. Aufgrund dieser Erkenntnisse lassen sich dann Entscheidungen treffen.²

¹vgl.: <http://de.wikipedia.org/wiki/Bildverarbeitung>

²vgl.: <http://de.wikipedia.org/wiki/Bildverstehen>

3 Bitmaps und Farben

Der Begriff Bitmap oder auch Rastergrafik beschreibt eine $m \times n$ Matrix P mit $P = (p_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ ³, wobei jedes Element p wie folgt definiert sein kann:

Binär	$p_{ij} \in \{1; 0\}$
Graustufen	$p_{ij} \in \{0; \dots; 255\}$
RGB	$p_{ij} = (r_{ij}; g_{ij}; b_{ij})$ wobei r_{ij}, g_{ij} und $b_{ij} \in \{0; \dots; 255\}$

Es gibt noch mehr Farbformate wie zum Beispiel YUV⁴, CMYK⁵ oder auch HVC⁶. Diese werden hier jedoch nicht näher aufgeführt, da ich lediglich die genannten drei verwende.

3.1 Binäre Bilder

Binäre Bilder sind unerlässlich für viele morphologische⁷ Filter. Rechnet man Graustufen oder RGB-Werte in binäre Werte um, so muss man einen Schwellenwert θ einführen. Dieser ist eine Konstante, die angibt, ab welcher Helligkeit ein Element der resultierenden Matrix auf 1 gesetzt wird.

In meiner Arbeit verwende ich nur die Umrechnung von Graustufen zu binären Bildern. Die Funktion, welche von Graustufen zu binären Bildern umrechnet, bietet sich an, da in den Graustufen nur die Helligkeit steckt und keine weitere Farbinformation.

Sei $A = (a_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ mit $a_{ij} \in \{0; \dots; 255\}$ und $B = (b_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ mit $b_{ij} \in \{0; 1\}$. Dann ist $t(i, j) : A \rightarrow B$ mit

$$t(i, j) := \begin{cases} 1 & \text{für } a_{ij} \geq \theta; \\ 0 & \end{cases}$$

3.2 Graustufen

Für die Erkennung von Bewegung werden keine Farbbilder (RGB), sondern Graustufen verwendet. Graustufen sind Farbneutral und jedes Element p_{ij} aus P wobei $p_{ij} \in \{0; \dots; 255\}$ gibt nur noch die Helligkeit Y_{ij} wieder.

Sei $A = (a_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ mit $a_{ij} = (r_{ij}; g_{ij}; b_{ij})$. So ergibt sich für $B = (b_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ mit $b_{ij} \in \{0; \dots; 255\}$ die Funktion $g(i, j) : A \rightarrow B$ mit

$$g(i, j) := 0.3 \cdot r_{ij} + 0.59 \cdot g_{ij} + 0.11 \cdot b_{ij}$$

³vgl.: http://www.uni-regensburg.de/EDV/Misc/CompGrafik/Script_1.html#Kap1.1

⁴Weiterführend <http://de.wikipedia.org/wiki/YUV>

⁵Weiterführend <http://de.wikipedia.org/wiki/CMYK>

⁶Weiterführend <http://www.photoshop-weblog.de/index.php?p=204>

⁷die Gestalt oder Form betreffend

Die Farbwerte werden jeweils unterschiedlich gewichtet, da das menschliche Auge sehr empfindlich auf gleiche Farbbeträge reagiert. Bei gleichen Farbbeträgen für Blau und Grün würde das Grün heller aussehen. Daher würde die Helligkeit des Bildes in Graustufen von der des Farbbildes abweichen. Das gewichtete Graustufenbild tut dies jedoch nicht.⁸

4 Techniken

4.1 Filter

Als Filter werden Algorithmen zur Bildverarbeitung klassifiziert. Diese lassen sich noch in weitere Gruppen unterteilen. So gibt es morphologische Filter und Störungsfilter um zwei zu nennen. Im Folgenden werden angewandte Filter erläutert.

4.1.1 Dilation



Dilation (Ausdehnung) ist ein morphologischer Filter. Die geometrische Fläche eines Objekts wird durch diesen Filter in einem binären Bild erweitert. Dilation ist definiert als der Verband aller Elemente a aus dem Objekt A mit allen Elementen b aus der Strukturierungsfunktion B .⁹ Sei $R = (r_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$ mit $r_{ij} \in \{0; 1\}$.

$$A \oplus B = \{t \in R : t = a + b, a \in A, b \in B\}$$

Dieser Filter wird lediglich für den in 4.1.3 aufgeführten Filter verwendet.

4.1.2 Erosion

Auch Erosion (Abtragung) ist ein morphologischer Filter. Dieser ist komplementär zum Dilation-Filter. Die geometrische Fläche eines Objekts wird verkleinert. Erosion

⁸vgl.: <http://home.arcor.de/ulile/node54.html>

⁹H. R. Myler und A. R., The pocket handbook of image processing algorithms in C, Prentice-Hall, 1993, S. 59



wird als das komplementäre Ergebnis des Dilation-Filters definiert.¹⁰

$$A \ominus B = (A^c \oplus B)^c$$

Dieser Filter wird lediglich für den in 4.1.3 aufgeführten Filter verwendet.

4.1.3 Closing



Closing (Schließen) ist definiert als die Anwendung einer Dilation auf A gefolgt von einer Erosion auf A . Für die beiden Filter wird die Strukturierungsfunktion B verwendet.¹¹

$$c(A, B) := (A \oplus B) \ominus B$$

Den Closing-Filter habe ich nach der Beschreibung auf Seite 40 in „The pocket handbook of image processing algorithms in C“¹¹ implementiert. Er wird benutzt, um eventuelle Lücken in den binären Bildern zu verkleinern und zu schließen.

4.1.4 Neighbour

Um einen schnellen Filter zum Entfernen der Störung aus einem binären Bild zu haben, habe ich diesen Filter selbst geschrieben. Dabei werden für alle Elemente b_{ij} , wobei $B = (b_{ij})_{i \in \{1, \dots, m\}; j \in \{1, \dots, n\}}$, die benachbarten Elemente gemäß einer Strukturierungsfunktion S überprüft.

¹⁰H. R. Myler und A. R., The pocket handbook of image processing algorithms in C, Prentice-Hall, 1993, S. 76

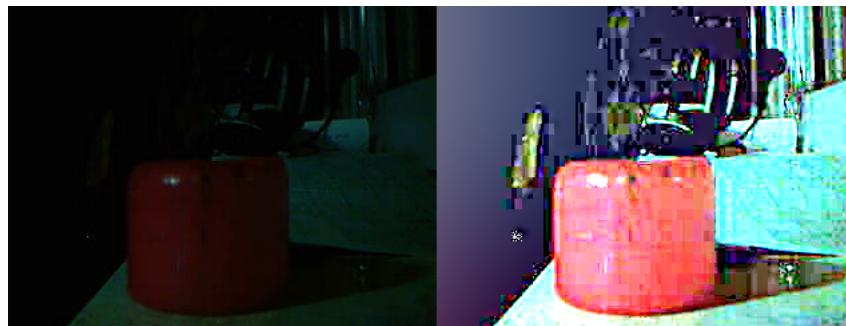
¹¹H. R. Myler und A. R., The pocket handbook of image processing algorithms in C, Prentice-Hall, 1993, S. 40



Sei $S = (s_{kl})_{k \in \{1, \dots, m\}; l \in \{1, \dots, n\}}$ mit $s_{kl} \in \{0; 1\}$ und $l \bmod 2 = k \bmod 2 \wedge k \bmod 2 = 1$. Wähle $\theta \in \{0; \dots; k \cdot l\}$.

$$n(i, j) := \begin{cases} 1 & \text{für } \sum_{m=0}^{k-1} \sum_{n=0}^{l-1} b_{i+(m-\frac{k-1}{2}); j+(n-\frac{l-1}{2})} \cdot s_{m+1; n+1} \geq \theta; \\ 0 & \end{cases}$$

4.1.5 Retinex



Der Retinex-Filter dient dazu, das von der Kamera aufgenommene Bild mehr dem der Realität anzugleichen. Ziel des Filters ist es, ein Bild A in die Helligkeitswerte H und Reflektionswerte R aufzuteilen, so dass $a_{ij} = r_{ij} \cdot l_{ij}$. Dadurch hat man die Möglichkeit eine Korrektur der Helligkeit und Farben vorzunehmen. So werden die Farben pro Objekt reduziert und es vereinfacht die Erkennung eben dieser¹². Ein weiterer Vorteil ist, dass man auch noch nachts mit einer billigen Kamera arbeiten kann.

Ich habe mit Hilfe des gimp¹³ Quelltexts¹⁴ den vorhandenen Retinex-Filter für meine Bedürfnisse angepasst und modifiziert, da die im gimp implementierte Version stark optimiert ist (als Beispiel rekursives Gausssmoothing).

Im allgemeinen wird der Retinex-Filter definiert als die Ausgabe R des Bildes I

¹²vgl.: http://cs.uni-muenster.de/u/lammers/EDU/ss03/Robotfussball/Ausarbeitungen/VisuelleWahrnehmung/RobotVision_Ausarbeitung.pdf

¹³GNU Image Manipulation Program <http://gimp.org>

¹⁴Stable 2.3.7

auf einer einfachen Pixel-Basis¹⁵.

$$R(x, y) = \log\left(\frac{I(x, y)}{I(x, y) * M(x, y)}\right)$$

wobei $M(x, y) = \exp\left(\frac{x^2+y^2}{\sigma^2}\right)$. σ ist dabei eine Konstante, die die Größe von M reguliert. * repräsentiert in diesem Fall eine Convolution-Matrix.

4.1.6 Blobcounter



Der Blobcounter¹⁶ ist ein Filter, welcher verbundene Pixel einfärbt. Hier werden von jedem Pixel p_{ij} die benachbarten Pixel überprüft. Sind benachbarte Pixel noch nicht mit einer Farbe versehen, so wird eine neue Farbe zugewiesen. Andernfalls wird dem Pixel p_{ij} die Farbe der benachbarten Pixel zugewiesen¹⁷.

Diesen Filter habe ich nach „Connected components labeling - algorithms in Mathematica, Java, C# and C++“¹⁷ implementiert, um die Rechtecke der Blobs zu extrahieren und so Informationen über Lage und Größe der bewegten Objekte zu bekommen.

4.2 Künstliche neuronale Netze

Künstliche neuronale Netze (KNN) orientieren sich stark am biologischen Vorbild. Man hat es bisher geschafft Netze zu entwickeln, die aus identischen Einheiten (künstlichen Neuronen) bestehen und lernfähig sind. Die Neuronen sind dabei hochgradig verknüpft und können an Beispielen lernen¹⁸.

Da eine ausführliche Erklärung dieser Thematik den Rahmen dieser Arbeit sprengen würde, werde ich nur kurz die Grundprinzipien anreißen.

¹⁵vgl.: <http://dragon.larc.nasa.gov/background/pubabs/papers/slides.pdf>

¹⁶Blob = Binary large object

¹⁷vgl.: http://www.izbi.uni-leipzig.de/izbi/Publikationen/Publi_2004/IMS2004_JankowskiKuska.pdf

¹⁸vgl.: <http://www.grundstudium.info/neuro/node6.php>

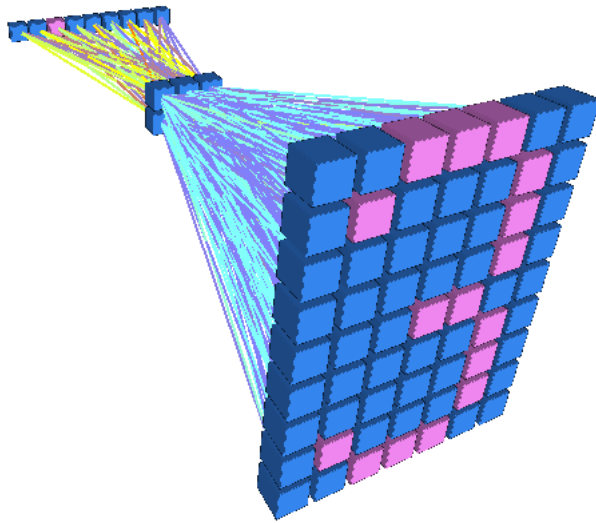


Abbildung 1: Ein $63 - 6 - 9$ Feed-Forward Netz zur Erkennung von Zahlen

4.2.1 Aufbau

Wie in der Natur werden Verbindungen zwischen Neuronen in ihrer Stärke unterschieden. Die Stärke einer Verbindung zwischen den Neuronen i und j wird mit w_{ij} bezeichnet. Alle empfangenen Informationen eines Neurons i werden mit net_i , wobei $net_i = \sum_{k=1}^n w_{ki} \cdot o_k$, bezeichnet. Desweiteren erhält ein Neuron die Ausgabe o_i , welche erfolgt, falls der Schwellenwert θ_i überschritten wird.

Den einfachsten Aufbau stellen Feed-Forward-Netze da, denn sie bestehen aus vollverknüpften Neuronen-Schichten, welche von links nach rechts durchlaufen werden, ohne dass eine Rückkopplung dabei stattfindet. Die Eingabe der Daten erfolgt dabei über den sogenannten Input-Layer. Differenziert werden die Informationen in n versteckten Ebenen (Hidden-Layer) und schließlich erfolgt die Ausgabe über den Output-Layer (EVA-Prinzip).

4.2.2 Nutzen

Mit Hilfe von künstlichen neuronalen Netzen lassen sich Muster wiedererkennen. Für mich ist vor allem die Fehlertoleranz ein wichtiger Aspekt, da selten zwei Muster, die man mit Hilfe der Bildverarbeitung aufbereitet, gleich aussehen, sondern meistens nur ähnlich.

5 Durchführung

5.1 Hardware/Aufbau

Es wurde eine Kamera mit 320×240 Pixeln Auflösung auf einer Tischplatte befestigt. Gegenstände wurden in einem Abstand von 50cm platziert. Lichtverhältnisse entsprachen dem Tageslicht. Das heißt, es wurde keine Veränderung vorgenommen.

5.2 Vorbereitung

Es wurde eine Umgebung geschaffen, die es ermöglicht jedes Bild, welches die Kamera empfängt zu verarbeiten. Die Kamera wurde dazu über DirectShow angesprochen und die Daten in einem 320×240 großen Array gespeichert.

5.3 Verfahren zur Erkennung von Objekten

Mit jedem Bild, das die Kamera empfängt, wird eine Funktion aufgerufen. In dieser Funktion wird die Erkennung gesteuert. Vorab wird bei jedem Bild die Bewegung getestet. Die Bewegungserkennung muss, wie später deutlich wird, keine Auskunft über die Stärke oder Platzierung der Bewegung geben. Es ist lediglich notwendig zu wissen, ob eine Bewegung stattgefunden hat oder nicht.

Um dies zu überprüfen wird das Differenzbild D mit $d_{ij} = |a0_{ij} - a1_{ij}|$ berechnet, wobei $a0$ und $a1$ die Graustufen des aktuellen und vorangegangenen Bildes sind. Ist die Summe aller Pixel im Differenzbild größer einer Konstante, so wurde eine Bewegung erkannt. Ist dies der Fall, so wird im nächsten Bild ohne Bewegung das vorher bewegte Objekt untersucht.

Findet keine Bewegung statt, so wird die Szene neu aufgebaut. Diese speichert das aktuelle Bild der Kamera als Hintergrund. Wird ein Objekt entfernt oder bewegt, so lässt sich über die Szene der veränderte Bildausschnitt ermitteln. Greift also ein Benutzer mit der Hand nach einem Gegenstand, so ist das vorangegangene Bild das ohne Bewegung, und dies wurde für die Szene als Hintergrund definiert. Solange bis der Benutzer die Hand wieder aus dem Bild nimmt, wird die Szene nicht neu gesetzt. Nimmt der Benutzer die Hand dann aus dem Bild, lässt sich ermitteln, welcher Gegenstand bewegt wurde, da zu dem alten Bild der Szene nun eine Differenz besteht.

Sobald also die Hand aus dem Bild genommen ist, wird die Objekterkennung durchgeführt. Dazu wird vom aktuellen Bild der alte Hintergrund entfernt und eine binäre Maske erstellt. Von diesem Binärbild wird dann mögliche Störung entfernt (Neighbour-Filter) und kleinere Lücken werden mit Hilfe des Closing-Filters geschlossen. Als nächstes werden die Blobs innerhalb der binären Maske analysiert. Ist wenigstens ein Blob vorhanden, so wurde ein Objekt bewegt. Andernfalls hat

der Benutzer nur die Hand in die Kamera gehalten. Wurde mindestens ein Blob gefunden, so wird die Bounding-Box¹⁹ des jeweiligen Blobs berechnet. Dabei wird die größte Bounding-Box ermittelt, kleinere werden als Störfaktoren ignoriert. Das mit dem Retinex-Filter bearbeitete Farbbild sowie die Bounding-Box und die durchschnittliche Farbe innerhalb dieser werden dann zur weiteren Analyse betrachtet.

Die binäre Maske des Objekts hat im Idealfall die Form des Gegenstands. Diese wird ausgeschnitten und skaliert in das neuronale Netz gespeist. Durchschnittliche Farbe und Größe müssen nicht weiter verarbeitet werden. Das Objekt mit den größten Übereinstimmungen in diesen drei Kategorien wird als erkannt gemeldet.

5.4 Erkenntnis

Der Versuch hat ergeben, dass die Bildererkennung stark mit der Kamera variiert. Für mich hat sich gezeigt, dass eine schlechte Kamera sehr viele Probleme bereitet. So kriegt man bei Tageslicht sehr viele störende Pixel und bei etwas weniger Licht schon wiederum fast gar kein Bild mehr. Doch mit Hilfe von sehr komplizierten Filtern lässt sich auch dieses Problem lösen. Problematisch ist es jedoch, wenn der Blobcounter durch Helligkeitsschwankungen ein falsches Bild bekommt, auf dem zum Beispiel noch eine Hand zu sehen ist oder der ganze Raum durch neue Lichtverhältnisse vom alten abweicht. Insgesamt bin ich mit dem Verfahren zufrieden, da die Erkennung fast in Echtzeit läuft und zudem meist richtige Ergebnisse liefert.

¹⁹Umschließendes Rechteck

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die besondere Lernleistung ohne fremde Hilfe erbracht habe, und nur die angeführten Quellen und Hilfsmittel benutzt habe.

Bielefeld, 5. April 2006

(Unterschrift des Schülers)