

# Robotertechnik und künstliche Intelligenz

## Teil 3: Softwaretechnologien in der Robotik



### Software als Grundlage für Flexibilität

Mit der Entwicklung von softwarebasierten Systemen nahm die Robotertechnik einen entscheidenden Aufschwung. Fest verdrahtete Einheiten ließen praktisch keinen Spielraum für flexible Anwendungen und Systeme. Schon die ersten funktionsfähigen Roboter arbeiteten bereits mit analogen oder sogar digitalen Schaltungen, sie waren jedoch nur für spezielle Anwendungen geeignet. Sobald geringste Veränderungen erforderlich wurden, mussten einzelne Komponenten oder ganze Module ausgetauscht oder ersetzt werden. Diese Situation änderte sich erst mit der Verfügbarkeit von programmierbaren Controllern und Prozessoren. Nun konnte durch einfaches Aufspielen einer neuen Software ein Roboter auch neue, bislang nicht beherrschbare Aufgaben übernehmen. Zudem bot die Softwaresteuerung bislang ungeahnte Möglichkeiten. Waren mit einfachen elektronischen Komponenten lediglich relativ simple Steuerregelungen möglich, bot sich nun ein ganzes Spektrum von neuen Applikationen. Von einfachen Sensoren bis hin

zu komplexen Bilderkennungsverfahren wurden Methoden verfügbar, die den Rahmen reiner Hardwareanwendungen bei Weitem sprengen würden. Bereits mit einfachen Programmen erzielen Roboter heute Leistungen, die mit rein elektronischen Lösungen nicht denkbar wären. Komplexe Softwarepakete erlauben es inzwischen sogar, dass Maschinen Aufgaben übernehmen, die vor wenigen Jahren nur von Menschen ausgeführt werden konnten. Dass aber für die Programmierung von Robotern nicht immer umfangreiche Fachkenntnisse erforderlich sind, zeigen grafische Anwendungen, die es bereits Kindern erlauben, einfache Aufgaben zu lösen.

### Programmieren für Kids mit Scratch

Der Bau und die Programmierung von Robotern hat für Kinder und Jugendliche eine ganz eigene Faszination. Während rein elektronische Schaltungen und Geräte für Einsteiger oft schwer zu durchschauen sind, bieten Roboter mit ihren mechanischen Komponenten ein im wahrsten Sinne des Wortes begreifbares System. Robots, wie etwa die bekannten Humanoiden Asimo oder NAO, üben vor allem auf Kinder eine große Anziehungskraft aus. Immer wenn diese humanoiden Gesellen auf Messen, Vorträgen oder Wissenschaftstagen vorgestellt werden, stehen sie im Mittelpunkt des Interesses von Kindern und Jugendlichen.

Um auch den Jüngeren die Möglichkeit zum Einstieg in modernste Robotertechnik zu bieten, wurde im Jahr 2007 eine grafikbasierte Programmiersprache veröffentlicht. Ein Team am Massachusetts Institute of Technology (MIT) in Cambridge (USA) entwickelte „Scratch“ als besonders für Kinder und Jugendliche geeignete Programmierumgebung. Das Hauptziel ist es, den Anwender mit den Grundkonzepten der Programmierung vertraut zu machen. Die drei Schritte Ausdenken, Entwickeln und Teilen sollen die kreative Erstellung eigener Programme ermöglichen. Scratch ist kostenlos und werbefrei und wird von einer internationalen Online-Community unterstützt. Der Name Scratch leitet sich von der Scratchtechnik der Diskjockeys der 80er- und 90er-Jahre ab. Beim klassischen Scratchen können aus bekannten Musikstücken neue Klänge erzeugt werden. Genauso entstehen beim Programmieren in Scratch aus interaktiven Objekten, Grafiken und Tönen neue Anwendungen. So kommen auch Einsteiger schnell zu Erfolgserlebnissen und werden motiviert, sich weiter mit der Materie zu beschäftigen.

Im Bereich der Roboterprogrammierung hat sich Scratch besonders schnell und intensiv etabliert, da die Raspberry Pi Foundation die zugehörige IDE mit in das Betriebssystem „Raspbian“ integriert hat. Damit steht mit dem Raspberry Pi eine weit verbreitete Hardwareplattform auch für Robotikanwendungen zur Verfügung. [Bild 1](#) zeigt ein Scratch-Anwendungsbeispiel zum Steuern von Motoren.

### Python – die angesagte Universalsprache für Robotik und KI

Für den Einstieg in die Welt der Softwareentwicklung ist Scratch sicherlich gut geeignet. Einfache Roboter lassen sich damit schnell und leicht



programmieren. Allerdings stößt man auch rasch an Grenzen, wenn es um komplexere Aufgaben geht. So kann ein simpler Linienverfolgungsalgorithmus noch rasch und übersichtlich in Scratch umgesetzt werden. Soll dagegen ein PID-Regler implementiert werden, ist Scratch nicht mehr die optimale Programmierumgebung. Hier steigt man besser auf die textbasierten Programmiersprachen um.

Neben klassischen Sprachen wie C oder Bascom hat sich im Robotikbereich vor allem Python durchgesetzt. Mit dem Aufkommen der Raspberry-Pi-Plattform war der Erfolg von Python auch in nicht-professionellen Anwendungsbereichen nicht mehr aufzuhalten. Da der Raspberry Python standardmäßig mit an Bord hat, wurde sehr schnell eine riesige Menge von Applikationen verfügbar. Insbesondere im Roboterbereich hat der Pi weite Verbreitung gefunden und dieses hochinteressante Gebiet auch für Einsteiger zugänglich gemacht.

Python wurde mit dem Ziel größtmöglicher Einfachheit und Übersichtlichkeit entworfen. Die Sprache kommt mit relativ wenigen Schlüsselwörtern aus und die zugehörige Syntax ist für maximale Übersichtlichkeit optimiert. Python ist eine universelle, meist interpretierte und vollständige höhere Programmiersprache. Die inhärente Übersichtlichkeit unterstützt einen gut lesbaren, knappen Programmierstil. Programmblöcke werden beispielsweise nicht durch geschweifte Klammern, sondern durch obligatorische Einrückungen strukturiert. Wegen ihrer klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen und Python-Programme fallen oftmals wesentlich kompakter aus als ein vergleichbarer Code in anderen Sprachen.

Der Name „Python“ geht im Übrigen nicht auf die gleichnamige Schlangengattung zurück, sondern bezog sich ursprünglich auf die englische Komikertruppe Monty Python. Dies spiegelt sich auch an verschiedenen Stellen in Bezeichnungen und Dokumentationen wider.

Aufgrund der großen Akzeptanz und Verbreitung entstanden für Python verschiedene GUI-Toolkits, wie z. B. Tkinter. Damit lassen sich ohne großen Aufwand auch eigene grafische Bedienoberflächen programmieren. Darüber hinaus können mit verschiedenen Softwaremodulen für Sensoren, Aktoren oder Kameras etc. problemlos auch komplexe Aufgaben gelöst werden. Insbesondere auf der Raspberry-Pi-Plattform lassen sich so Robotersysteme entwickeln, deren Funktionsbereich von einfacher Hindernisumfahrung bis zur Gesichtserkennung oder Sprachsteuerung reicht.

Bild 2 zeigt z. B. den Blick durch eine Fahrzeugkamera. In das Gesichtsfeld werden dabei die Bedienelemente mit eingeblendet, sodass das Fahrzeug via WLAN bequem und effizient über einen PC-, Laptop- oder Tablet-Bildschirm gesteuert werden kann. Ein Antippen oder Anklicken der Steuerelemente genügt, um das Fahrzeug zu lenken, zu beschleunigen, abzubremsen oder zu stoppen.

Derartige interaktive Steuerungen sind mit Python schnell und einfach realisierbar, sodass sich auch fortgeschrittene Projekte leicht umsetzen lassen. Die Einbindung von Kameras, Motorsteuerungen oder Funksystemen wie WLAN oder Bluetooth stellt dank umfangreicher Bi-

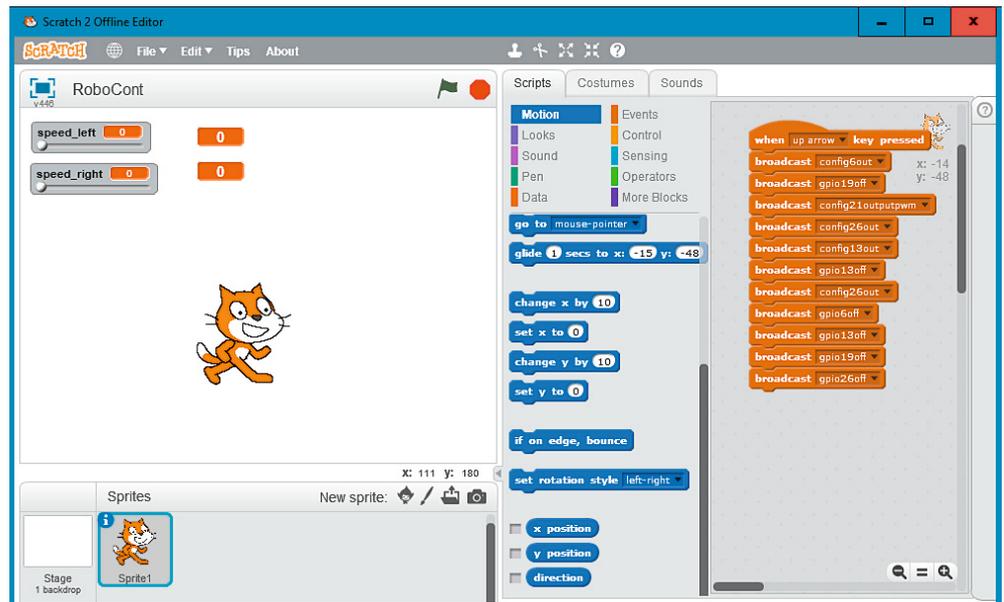


Bild 1: Robotersteuerung mit Scratch

bibliotheken kein Problem dar. So lässt sich etwa ein autonomes Fahrzeug entwickeln, das seine Steuerbefehle über WLAN erhält.

Kommt dann noch eine WLAN-Kamera hinzu, verfügt man über ein Roboter-Fahrzeug, mit dem man die eigenen vier Wände effizient im Auge behalten kann. Bei einer Anbindung an das Internet kann man so etwa die eigene Wohnung überwachen, während man z. B. auf Dienstreise in den USA oder im Urlaub in Australien ist.

## Drahtlose Steuerung über Webinterfaces und WLAN

Um ein Roboterfahrzeug vollständig drahtlos zu kontrollieren, wird keine aufwendige Funkfernsteuerung mehr benötigt. Das in den meisten Haushalten ohnehin vorhandene WLAN ist vollkommen ausreichend. Die eigentliche Steuerung kann über einen Arduino mit WLAN-Shield oder aber über einen Raspberry Pi ab Version 3 erfolgen, da die neueren Raspberry-Versionen bereits über ein integriertes WLAN-Modul verfügen.



Bild 2: Gesichtsfeld mit eingeblendeten Steuerelementen



Bild 3: Roboterfahrzeug mit WLAN-Kamera

Für die Bildübertragung kann die Raspberry-PiCam oder eine externe WLAN-Kamera zum Einsatz kommen. **Bild 3** zeigt ein Roboterfahrzeug, welches mit einer universellen WLAN-Heimkamera ausgestattet ist.

Die Bildübertragung erfolgt direkt über das WLAN-Modul der Kamera. Für die Steuerung des Fahrzeugs kommt ein Arduino mit WLAN-Shield zum Einsatz. Das Fahrzeug kann so vollständig über ein WEB-Interface gesteuert werden. Alternativ könnte hier auch ein Raspberry Pi mit PiCam verwendet werden. In diesem Fall würde das Kamerabild über den gleichen WLAN-Kanal wie die Steuerbefehle übertragen werden.

Eine gewisse Problematik der Websteuerung ergibt sich, wenn die Internetverbindung abbricht. Insbesondere beim Betrieb über das World Wide Web kann es immer wieder zu Verbindungsabbrüchen kommen, vor allem wenn man sich in Gebieten mit schlechter Internet-Infrastruktur befindet.

In diesen Fällen wäre das Kamerafahrzeug gefährdet, da es stur den letzten Befehl, z. B. „Volle Fahrt voraus“ ausführen würde. Eine im Weg befindliche Wand oder gar ein Treppenabsatz könnten hier katastrophale Folgen für den Roboter haben.

Es ist deshalb sinnvoll, zusätzlich zur WLAN-Steuerung auch Sensoren vorzusehen, die das System im Fall der Fälle autonom steuern können und so Schlimmeres verhindern. Hier sind Antikollisionsmechanismen auf Infrarot- oder Ultraschallbasis das Mittel der Wahl. Abstürze an Treppenabsätzen oder Ähnlichem können mit nach unten gerichteten optischen Sensoren verhindert werden.

Zusammen mit entsprechenden Software-Algorithmen und Regelsystemen kann sich das Fahrzeug dann auch bei Ausfall der WLAN-Steuerung sicher in der heimischen Wohnung oder in Büroumgebungen bewegen.

### Die hohe Kunst der Regelungstechnik: autonome Fahrzeuge, selbstbalancierende Roboter und Kalman-Filter

Bereits im letzten Artikel (Sensoren und Aktoren in der Roboterentwicklung, ELV Journal 5/2018) wurde ein einfacher Regelkreis vorgestellt. So wurde darge-

legt, wie ein autonomes Fahrzeug mithilfe optischer Sensoren einer auf dem Boden aufgemalten Linie folgen kann. Im einfachsten Fall kann diese Aufgabe bereits mit einem sogenannten Zweipunktregler gelöst werden.

Ein vereinfachter C-Code für die Arduino-Programmierungsumgebung könnte dazu so aussehen:

```
// lineFollower

int leftSensor;
int rightSensor;

void setup()
{ pinMode(6, OUTPUT); // front LED
  digitalWrite(6, HIGH); // front LED ON
}

void loop()
{ leftSensor = analogRead(2);
  rightSensor = analogRead(3);
  setMotorDirection (1, 1);
  if(leftSensor-rightSensor > 0) setMotorSpeed(80,40);
  else setMotorSpeed(40,80);
}
```

Das Programm erfasst die Helligkeitswerte der beiden Liniensensoren. Diese sind an der Unterseite des Fahrzeugs angebracht und erfassen den Bereich auf der rechten bzw. linken Seite der Leitlinie. Wird eine Helligkeitsdifferenz detektiert, reduziert das Programm die Geschwindigkeit eines Antriebsmotors. Droht das Fahrzeug nach links auszubrechen, wird der rechte Motor gebremst, bei einem Ausbruch nach rechts der linke. Mit dieser einfachen Methode kann das Fahrzeug bereits einer Linie folgen. Allerdings wird schnell klar, dass das Fahrverhalten nicht optimal ist. Der autonom fahrende Roboter zeigt ein zitterndes Pendelverhalten rund um die Bodenlinie und verliert häufig die Führung.

Deutlich verbessert wird die Situation durch die Anwendung eines sogenannten PID-Reglers. Dieser ermöglicht durch proportionale, integrale und differentielle Regelungsanteile eine optimale Anpassung des Robotersystems an die geforderte Aufgabe. Der Regler errechnet genau wie bei der Zweipunktversion aus der Differenz der beiden Liniensensoren eine Stellgröße für die Motoren. Die Störgröße, in diesem Fall ein Abweichen von der Bodenlinie, bewirkt eine Veränderung der Regelgröße, die vom Regler kompensiert werden muss.

Der proportionale Anteil multipliziert die Regelabweichung mit einem konstanten Verstärkungsfaktor und gibt das Ergebnis unverzüglich weiter. Dieser Anteil arbeitet relativ schnell. Allerdings verbleibt stets eine gewisse Regelabweichung, da die Schleifenverstärkung des Kreises nicht beliebig groß werden kann. Regelabweichungen werden daher nur bis zu einem gewissen Maß kompensiert.

Der integral wirkende Anteil summiert dagegen die Regelabweichung über der Zeit auf und multipliziert diese Summe mit einem Faktor. Der I-Anteil wird also umso größer, je länger eine Regelabweichung anliegt. Dieser Reglerzweig ist zwar langsam, hat aber den Vorteil, dass er sehr exakt ist, da er prinzipiell alle Abweichungen vollständig eliminieren kann.

Der differential wirkende Anteil bewertet Signaländerungen. Er reagiert damit sozusagen bereits auf die Ankündigung einer Regelabweichung. Ein Regler mit D-Anteil kann dadurch sehr schnell werden.

Ist das Regelsystem optimal abgestimmt, kann das autonome Roboterfahrzeug auch komplizierten Linienverläufen exakt folgen. Der Roboter ist dann in der Lage, mit hohen Geschwindigkeiten zu agieren, die Lenkbewegungen sind wohldosiert und es kommt kaum mehr zum Führungsverlust. Mit justierten PID-Reglern kann die Geschwindigkeit und Präzision von Roboterfahrzeugen die Fähigkeiten menschlicher Fahrer sogar deutlich übersteigen.

Die Linienverfolgung ist eine der einfachsten Aufgaben in der Robotik. In den letzten Jahren wurden auch wesentlich aufwendigere und komplexere Probleme erfolgreich gelöst. Dazu gehören etwa der auf-



rechte Gang auf zwei Beinen oder das Fangen von Bällen mittels eines servogesteuerten Roboterarms. Auch hier ist immer ein komplexes Netzwerk von Regelalgorithmen am Werk.

Auch die Steuerung physikalisch instabiler Systeme ist möglich. Beispiele hierfür sind:

- Invertierte Pendel
- Selbstbalancierende Roboterfahrzeuge
- Die Lagestabilisierung von RC-Helikoptern oder Drohnen

Neben schnellen Regelungen sind für diese Aufgaben auch weitere Methoden für die Verarbeitung von Sensordaten erforderlich. Eines der am häufigsten eingesetzten Verfahren ist die Kalman-Filterung. Hierbei handelt es sich um ein spezielles mathematisches Verfahren, das dazu dient, Fehler in Sensorwerten zu minimieren. Zudem liefert die Methode Schätzungen für nicht messbare Größen. Die gesuchten Werte müssen dabei jedoch durch ein mathematisches Modell beschrieben werden können. Ein Vorteil des Verfahrens ist, dass seine spezielle mathematische Struktur sich besonders für den Einsatz in Echtzeitsystemen eignet. Kalman-Filter spielten bereits im Apollo-Programm eine zentrale Rolle und brachten letztlich Neil Armstrong und seine Kollegen heil zum Mond und wieder zurück.

Weite Verbreitung hat das Kalman-Filter in der Inertialnavigation beispielsweise von Flugzeugen oder Drohnen gefunden. Die Signale verschiedener Sensoren werden hierzu verknüpft, um eine optimale Schätzung der aktuellen Position und Orientierung zu erhalten.

Selbstbalancierende Roboter (Bild 4) haben weltweit das Interesse vieler Forscher, Studenten und Hobbyanwender geweckt. Prinzipiell betrachtet handelt es sich um umgekehrte Pendel auf Rädern. Das invertierte Pendel ist aufgrund seiner instabilen Natur ein klassisches Problem der Regelungstechnik. Der bekannte Segway PT (Personal Transporter) war einer der Auslöser für die Popularität selbstbalancierender Roboter. Diese personentragenden Roboterfahrzeuge erfreuten sich aufgrund ihrer Manövrierfähigkeit, insbesondere ihres geringen Wendradius, sehr schnell großer Beliebtheit. Inzwischen hat sich der Segway in vielen Bereichen etabliert. Er wird sowohl von Touristen im Stadtpark als auch von polizeilichen Einsatzkräften und sogar als Krankentransportwagen genutzt. In jüngster Zeit hat eine Weiterentwicklung des Segways, das sogenannte Hoverboard, wieder die Aufmerksamkeit vieler Menschen auf die dahinter stehende Technik gelenkt. Darüber hinaus kommen Kalman-Filter auch in anderen Technikgebieten zum Einsatz. So nutzen PLL-Filter in Radio- und Funkgeräten spezielle Varianten der Kalman-Filtermethoden.

### Von der Kollisionsvermeidung zur Gesichtserkennung

Neben der Linienverfolgung spielt die Kollisionsvermeidung eine entscheidende Rolle in der Entwicklung von robotergesteuerten Fahrzeugen. Die Linienverfolgung kommt vorzugsweise in speziellen Umgebungen, wie etwa Lagerhallen oder Logistikzentren, zum Einsatz. Kollisionsvermeidung dagegen ist eine für alle autonomen Fahrzeuge unverzichtbare Voraussetzung. Selbstfahrende Autos müssen über eine zuverlässige Methode zur Vermeidung von Zusammenstößen mit anderen Fahrzeugen, Menschen, Tieren oder festen Hindernissen verfügen. Neben Ultraschallsystemen kommen für diese Aufgabe sowohl Laserscanner, LIDAR- als auch Bilderkennungsverfahren zum Einsatz. Der Vorteil von Laserscannern und LIDAR (light detection and ranging) liegt in der Erfassung von dreidimensionalen Umgebungsdaten. Anders als bei einer reinen Bilderkennung liefern Laserscanner auch Distanzwerte. Der Laserscanner erlaubt es also, eine detaillierte 3D-Karte der Umgebung zu erzeugen. Das heißt, ein Hindernis wird nicht nur als solches erkannt, sondern auch die Distanz zwischen Objekt und Fahrzeug wird ermittelt. Größere Bekanntheit erlangten Laserscanner durch das Google-Auto, bei welchem eine entsprechende Einheit auf dem Fahrzeugdach montiert war.

Die Daten der LIDAR- und Kamerasensoren müssen in Echtzeit verarbeitet werden. Dies stellt eine enorme Herausforderung an die Rechen-

leistung und die Software der eingesetzten Steuercomputer dar. Die klassischen Programmierverfahren stoßen hier schnell an ihre Grenzen. Die reale Welt ist zu komplex, um sie mit konventionellen, regelbasierten Programmen zu erfassen.

Daher setzen die Entwickler zunehmend auf die Methoden der Künstlichen Intelligenz (kurz KI). Dieses Forschungsgebiet hat sich die Übertragung menschlicher Intelligenzleistungen auf Computersysteme zum Ziel gesetzt. Das sogenannte „Deep Learning“ wird dabei eine zentrale Rolle spielen.

Menschen können problemlos Objekte erkennen und entsprechende Beobachtungen in ein Modell ihrer unmittelbaren Umgebung umsetzen. Selbst die leistungsfähigsten Supercomputer sind mit dieser Aufgabe bis heute überfordert. Erst Deep-Learning-Verfahren brachten einen gewissen Durchbruch. Diese zentrale Lernmethode der Künstlichen Intelligenz basiert auf Optimierungsverfahren künstlicher neuronaler Netze mit zahlreichen Zwischenschichten. Im Verlauf eines Lernprozesses werden die Netze dabei mit umfangreichen Datensätzen trainiert. Durch Deep Learning können Computer, Roboter oder auch autonome Fahrzeuge lernen, ihre Umwelt regelrecht zu verstehen. Aktuell sind sich die Experten einig, dass vollständig autonom fahrende Autos nur mithilfe von künstlichen neuronalen Netzen realisierbar sein werden.

Das autonome Fahren ist jedoch nur eine der Anwendungen, die durch den Einsatz von künstlichen neuronalen Netzen revolutioniert wurden. Auch auf anderen Gebieten gelangen entscheidende Durchbrüche. Zu nennen sind hier:

- Text- und Handschriftenerkennung, Texterzeugung
- Spracherkennung und Sprachsteuerung (Siri, Alexa etc.)
- Suchmaschinen wie Google
- Steuerung von Robotern
- Maschinelle Sprachübersetzung (Google Translator, DeepL)
- Bild- und Gesichts(wieder)erkennung

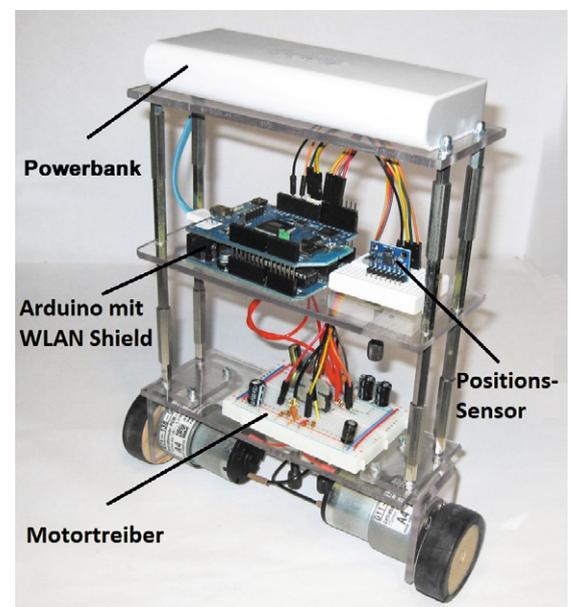


Bild 4: Selbstbalancierender Roboter

Insbesondere die Gesichtserkennung kann bereits auf kleinen Rechneinheiten, wie etwa dem Raspberry Pi, umgesetzt werden. Bei entsprechenden Verfahren ist zwischen der Lokalisation eines Gesichts im Bild und der Zuordnung des Gesichts zu einer bestimmten Person zu unterscheiden. Im ersten Fall wird geprüft, ob und wo ein Gesicht zu sehen ist, im zweiten geht es um den deutlich anspruchsvolleren Vorgang einer eindeutigen Identifizierung.

Die Gesichtserkennung zählt zu den biometrischen Verfahren. Sie wird in der Sicherheitstechnik zum Zweck der Identifikation von natürlichen Personen eingesetzt. Meist dient die technische, computergestützte Gesichtserkennung der Zutrittskontrolle zu sicherheitsrelevanten Bereichen. Aber auch die Suche nach Doppeleinträgen in Datenbanken, beispielsweise in Melderegistern zur Vermeidung von Identitätsdiebstahl oder Sozialbetrug, hat in den letzten Jahren eine besondere Bedeutung erlangt.

Einfache Gesichtserkennungsverfahren verwenden die zweidimensionale Vermessung besonderer Merkmale wie Augen, Nase oder Mund. Hierbei wird deren Position, Abstand und Lage zueinander bestimmt. Moderne Verfahren setzen dagegen komplexe Berechnungen wie die Waveletanalyse, Fourier- oder Gabor-Transformation bzw. Hauptkomponentenanalysen ein. Vor wenigen Jahren wurden noch vier von fünf Personen nicht korrekt erkannt. Inzwischen konnte die Fehlerrate auf unter 1 % reduziert werden. Damit liegt sie in der gleichen Größenordnung wie Fingerabdruck- oder Iriserkennungsverfahren. Die besten Systeme übertreffen jetzt sogar die Fähigkeiten der menschlichen Gesichtserkennung, sodass sie zur automatischen Personenkontrolle an Grenzen und Flughäfen eingesetzt werden können.

Aktuelle Verfahren beruhen wiederum auf maschinellem Lernen und können auch Strukturen aller Art erkennen, etwa Verkehrszeichen für das autonome Fahren. Inzwischen sind sogar frei verfügbare Programmsysteme zur Gesichts- und Objekterkennung verfügbar. Das bekannte OpenCV (für Open Computer Vision) läuft bereits auf der Raspberry-Pi-Plattform und liefert überraschend gute Ergebnisse.

## OpenCV

Die Erkennung von Gesichtern hat erhebliches Zukunftspotenzial – sowohl im positiven als auch im negativen Sinne. Es kann daher nicht schaden, wenn



Bild 5: Gesichtserkennung mit OpenCV

man eigene Erfahrungen auf diesem Gebiet sammelt. In diesem Fall ist man mit der quelloffenen Bibliothek OpenCV gut bedient. Das modulare Softwarepaket verfügt über Algorithmen für unterschiedlichste Aufgaben, unter anderem:

- Identifizierung von Objekten und Aktionen in Fotos und Videos
- Tracking von Objekten
- 3D-Visualisierung von Stereokamera-Streams
- Erstellen von Fotografien aus kleineren Einzelaufnahmen (Stitching)
- Bildervergleich
- Gesichtserkennung

Eine zentrale Komponente von OpenCV ist das Machine Learning Modul. Darin finden sich die passenden Algorithmen für viele der genannten Aufgaben. Die Software ist damit auch in der Lage zu lernen, was ein Gesicht ist und zu welcher Person es gehört. Gesichter in einer Fotografie zu identifizieren ist vergleichsweise simpel. OpenCV liefert dazu ein Python-Programm, das Gesichter oder Augen in Webcam-Aufnahmen erkennt und markiert. Bild 5 zeigt ein Beispiel dazu.

Neben der Gesichtserkennung eignet sich OpenCV auch für die Realisierung weiterer hochinteressanter Bildverarbeitungsprojekte. Eine Anwendung ist das zeitaufgelöste Erfassen von Stromverbrauchswerten. In vielen Haushalten finden sich immer noch Stromzähler mit mechanischem Zählwerk. Diese sogenannten Ferraris-Zähler arbeiten nach dem Induktionsprinzip. Hierbei wird in einer Aluminiumscheibe ein magnetisches Drehfeld induziert, welches die Scheibe durch Wirbelströme in Rotation versetzt. Die Anzahl der Scheibenumdrehungen wird durch einen Zählwert aufgezeichnet und dient als Maß für die verbrauchte Strommenge. Dieser Zählertyp besitzt keine direkte, genormte Schnittstelle für das Auslesen der verbrauchten elektrischen Energie. Über die optische Erfassung des Zählerstands mit einer Kamera und der Bilderkennungssoftware von OpenCV kann man dennoch den Zählerstand elektronisch erfassen. Ein entsprechendes Projekt kann mit einem Raspberry Pi umgesetzt werden.

Bild 6 zeigt, wie die PiCam zunächst ein fotografisches Bild des Zählerstands aufnimmt (oben links). Dieses wird dann mit verschiedenen Methoden wie etwa Kantendetektion und Kontrastverstärkung vorverarbeitet (oben rechts). In einem dritten Schritt erfolgt mithilfe von Mustererkennungsverfahren die Erzeugung eines numerischen Werts (unten). Man erkennt, dass für die Nachkommastelle keine eindeutige Zuordnung erfolgen konnte, da die Ziffern nur teilweise erkennbar sind. In diesem Fall muss man sich beispielsweise durch Interpolation aus mehreren Aufnahmen behelfen.

Die gewonnenen numerischen Daten können nun stündlich oder sogar minutlich aufgezeichnet werden. Im Laufe der Zeit kann so ein Energieverbrauchsdiagramm erstellt werden, welches es erlaubt, besonders stromintensive Perioden zu erfassen. Eine genaue Analyse erlaubt es dann, Geräte mit hohem Stromverbrauch mit preisgünstigem Nachtstrom zu betreiben oder andere energieeinsparende Maßnahmen zu ergreifen.

## Maschinenbewusstsein?

Die Umsetzung von Objektbeobachtungen in ein Umgebungsmodell kann bereits als ein erster Schritt in Richtung einer bewussten Wahrnehmung angesehen werden. Oft wird sogar davon gesprochen, dass sich bereits ein Rasenmäher- oder Staubsaugerroboter über seine Umgebung bewusst sein muss, da er seine Aufgabe sonst nicht korrekt ausführen könnte. Letztendlich kann man daher sogar abstrakte Begriffe wie eben das Bewusstsein durch Aspekte der Regelungstechnik beschreiben.

Die niedrigste Stufe ist ein einfacher Regelkreis. So ermöglicht es etwa ein Heizkörperthermostat, die Temperatur in einem Raum weitgehend konstant zu halten. Er sammelt permanent Umweltdaten und ist sich somit zumindest über einen Parameter seiner Umgebung „bewusst“. Darüber hinaus kann er diesen Parameter beeinflussen und so seinen „Wünschen“ anpassen.



Einzeller wie etwa Bakterien verfügen bereits über eine Vielzahl von Rückkopplungsschleifen und können deshalb etwa Gebiete mit ausreichendem Nahrungsangebot aufsuchen. Selbst die einfachsten Pflanzen haben schon eine zweistellige Anzahl von Regelungen implementiert. Sie können beispielsweise die Richtung der Schwerkraft bestimmen und so nach oben wachsen. Farne und Moose können ihre interne Feuchtigkeit regeln und ihre Blätter bei Trockenheit einrollen. Die meisten Pflanzen sind in der Lage, ihre Blätter oder Blüten nach dem Sonnenstand auszurichten etc. Sie sind sich also in diesem Sinne über verschiedene Eigenschaften ihrer Umgebung „bewusst“. In diesem Sinne kann man hier von einem Bewusstsein der Stufe I sprechen.

Bewegungsfähige Organismen benötigen bereits ein deutlich weiterentwickeltes Regelsystem. Viele Funktionen werden schon über einen gemeinsamen „Bus“, das sogenannte zentrale Nervensystem gesteuert. Hier kommen schnell hundert oder mehr Regelungssysteme zusammen. Beispiele reichen von einfachen Fadenwürmern bis hin zu den Reptilien. Diese Lebewesen kann man in die Bewusstseinskategorie II einordnen.

Die nächste Kategorie III beschreibt Tiere, die nicht nur ihre räumliche Position erfassen können, sondern auch andere Lebewesen als solche erkennen. Die Entscheidung, ob es sich um Fressfeinde, Beutetiere oder Paarungspartner handelt, ist für diese Lebewesen von entscheidender Bedeutung. Die Anzahl der sensorischen Informationen nimmt nun rasant zu. Die Verarbeitung dieser „Daten“ erfordert bereits eine erhebliche Rechenleistung. Eine zentrale Verarbeitungseinheit, das Gehirn, muss nicht nur über eine ausgezeichnete Bildverarbeitung verfügen, sondern auch in der Lage sein, rasch und effizient lebenswichtige Entscheidungen zu treffen.

Für die Bewusstseinskategorie IV kommt noch eine weitere entscheidende Komponente hinzu. Nicht nur die räumliche Lage muss erfasst werden, auch die zeitliche Dimension bekommt eine wichtige Bedeutung. Das Konzept einer „Zukunft“ ist hierbei ganz wesentlich. Die Regelschleifen beziehen sich nicht mehr nur auf das Hier und Jetzt. In einem Bewusstsein der Stufe IV wird kontinuierlich eine mögliche Zukunft geplant. Die Regelungen erstrecken sich nun über erhebliche Zeiträume, wie Stunden, Tage, Monate oder sogar Jahre. Es kommen Fähigkeiten wie „Erfahrung“ und „Lernen“ hinzu. Letztendlich handelt es sich dabei nur um Regelkreise, die zusätzlich über einen großen Speicher verfügen. Wer etwa einmal zu schnell mit dem Fahrrad um eine Kurve gefahren ist, verfügt automatisch über eine neue Regelschleife, welche dafür sorgt, dass die Geschwindigkeit vor engen Radwegkurven „unbewusst“ reduziert wird. Die folgende Tabelle fasst diese Bewusstseinsstufen zusammen:

	Beispiele	Regelkreise	Anzahl von Regelungen
I	Amöben, Pflanzen	Schwerkraft, Licht, Temperatur	1 bis ca. 10
II	Würmer, Reptilien	Räumliche Orientierung	Dutzende bis Hunderte
III	Säugetiere	Bildverarbeitung, Freund-/Feinderkennung	Hunderte bis Tausende
IV	Menschenaffen, Menschen	Zeitliche Erfassung, Zukunftsplanung, lernfähige Regelungen	Tausende Regelung und Simulation

Man erkennt, dass auch Maschinen und Roboter immer mehr in Gebiete vordringen, die ursprünglich nur den Lebewesen oder sogar nur dem Menschen vorbehalten waren. Ob und wann Computer oder Roboter ein vollständiges Bewusstsein erlangen werden, wird sich erst in der Zukunft zeigen. Falls es allerdings tatsächlich dazu kommen sollte, dann könnte dies zu einer der größten Herausforderungen für die gesamte Menschheit werden.

## Ausblick

In diesem Artikel wurden verschiedene Softwaretechnologien und ihre Anwendungen in der Robotik vorgestellt. Von der einfachen Kollisionsvermeidung über ausgeklügelte Regelkreise bis hin zur Bilderkennung

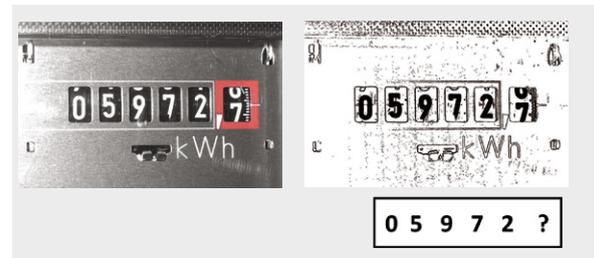


Bild 6: Machine Vision: Zeichenerkennung mit OpenCV

reichen die softwaretechnischen Gadgets, die modernen Robotern ihre oftmals bereits unglaublichen Fähigkeiten verleihen.

Im nächsten Artikel stehen verschiedene Anwendungen zu autonomen Fahrzeugen im Vordergrund. Fahrerlose Kraftfahrzeuge werden vermutlich die ersten Anwendungen Allgemeiner Künstlicher Intelligenz (AKI) sein. Bis vor Kurzem war die maschinelle Intelligenz immer nur auf sehr spezielle Aufgaben, wie die Lösung mathematischer Probleme oder das Schachspiel, begrenzt. Hier haben die Maschinen die menschlichen Fähigkeiten zum Teil bereits übertroffen. So schlug bereits im Jahr 1997 der Computer Deep Blue den damaligen Schachweltmeister Kasparow unter Turnierbedingungen.

Um ein Fahrzeug sicher und unfallfrei durch hektischen Stadtverkehr zu lenken ist dagegen eine breite, sogenannte Allgemeine Intelligenz notwendig. Hier müssen Hunderte von Sensorwerten innerhalb von Sekundenbruchteilen ausgewertet und korrekte Fahrbefehle, wie lenken oder bremsen, erzeugt werden. Nicht nur Abstands- und Geschwindigkeitsmessungen sind erforderlich, sondern auch Bilderkennung und -erfassung. Diese Aufgabe ist durch regelbasierte Technologien nicht mehr zu lösen. Hier sind selbstlernende Systeme erforderlich, die bereits weitgehend auf künstlicher Intelligenz basieren.

Neben grundlegenden Betrachtungen zum autonomen Fahren werden aber auch wieder Praxisanwendungen zu diesem Thema erläutert. So wird ein kleines autonom fahrendes System vorgestellt, das ohne großen Aufwand nachgebaut werden kann. Das Fahrzeug ist modular aufgebaut und erlaubt durch die Anwendung verschiedener Sensoren, erste Erfahrungen mit selbstlenkenden Fahrrobotern zu gewinnen.

Anhand verschiedener Beispiele wird erläutert, wie mobile Roboter Hindernisse erkennen und diesen ausweichen können. Neben einfachen Messsystemen werden auch Bild- und Mustererkennungsverfahren erläutert, welche die Grundlage für autonomes Fahren in öffentlichen Verkehrsräumen bilden. **ELV**

Empfohlene Produkte	Bestell-Nr.
Arex-Asuro-Miniroboter ARX-03	CY-09 73 14
Velleman-Roboterbausatz Allbot VR408	CY-12 26 15
Ultraschall-Abstandssensor für Minicomputer wie Raspberry Pi, Arduino und Co.	CY-12 21 21
JOY-iT Weitwinkel-Kamera für Raspberry Pi	CY-14 51 32