Dosuas - Die Symphonie des Sehens

Jugend Forscht 2018

Jonas Wanke und Yorick Zeschke

18. Januar 2018

Dosuas (**D**evice for **O**rientation in **S**pace **U**sing **A**udio **S**ignals) ist ein Gerät, welches blinden Menschen ermöglicht sich mithilfe von Tonsignalen im Raum zu orientieren und Objekte zu erkennen.

Das Projekt besteht aus zwei Unterprojekten, die beide bis zum Wettbewerb als Prototypen umgesetzt werden sollen. Einmal werden Bilder eines 3D Kamera mit einem Programm in Töne umgewandelt, die dann mit 3D-Audio Kopfhörern hörbar gemacht werden. Die andere Idee basiert darauf, so ähnlich wie eine Fledermaus Ultraschall Impulse zu senden und deren Reflektionen bzw. Echos hörbar zu machen, sodass man sich mit Klicklauten orientieren kann. Letzteres basiert auf der Technik der aktiven menschlichen Echoortung.

Inhaltsverzeichnis

1 Einleitung

Blinde Menschen haben schon immer Probleme damit, sich im Raum zu orientieren. Manche von ihnen, zum Beispiel *Daniel Kish* benutzen die Technik der *menschlichen Echoortung*, ein Verfahren, bei dem man regelmäßig mit dem Mund Klicklaute erzeugt und das Gehör darauf trainiert anhand der Reflektionen ein genaues Bild der Umgebung im Kopf zu erzeugen. Forscher haben herausgefunden, dass sich dabei die Struktur des Gehirns verändert und Signale von den Ohren im Sehzentrum verarbeitet werden. Mit genügend Übung schaffen es Blinde so zu „sehen“ und können Fahrrad fahren oder in den Bergen klettern.

Doch nicht jedem Blinden fällt es leicht und nicht jeder hat die Möglichkeit eine solche Technologie zu erlernen. Außerdem hat auch die menschliche Echoortung ihre Grenzen und ist ab einem bestimmten Punkt nicht mehr erweiterbar. Hier kommt die Technologie ins Spiel. Von Tag zu Tag ergeben sich neue Möglichkeiten mithilfe der verschiedensten technischen Hilfsmittel Menschen das Leben zu erleichtern. Geräte wie 3D-Sensoren oder Kameras können heutzutage schon oft sehr realistische und detaillierte Bilder aufnehmen, die dem menschlichen Sehen sehr nahe kommen.

Relativ neu ist zum Beispiel die Technologie der Retina Implantate, die sich momentan aber noch im Anfangsstadium der Entwicklung befinden. Mit ihnen soll es in Zukunft möglich sein, dass Blinde wie nicht sehbehinderte Menschen sehen, jedoch können die Kosten von 75.000 € aufwärts selten von den Blinden selbst getragen werden und werden nur manchmal von Krankenkassen übernommen. Auch gibt es zu viele blinde und sehbehinderte Menschen, als das es möglich wäre jeden mit einem so teuren Gerät zu versorgen.

Andere Firmen versuchen das Sehen technisch durch andere Sinne zu ersetzen. Ein berühmtes Beispiel dafür ist der „BrainPort V100“[1], welcher Kamerasignale in elektronische Impulse umwandelt, die auf der Zunge spürbar gemacht werden. Nachteile dieser Technologie sind vor allem lange Lernprozesse, die nur mit ärztlicher Unterstützung möglich sind, Probleme bei zu vielen Reizen oder große Ungenauigkeiten. Beispielsweise kann es passieren, dass ein Blinder beim betrachten des Geschehens auf einer großen Straße nichts mehr wahrnimmt, weil die der Tastsinn der zunge nicht für eine solche Reizüberlastung ausgelegt ist. Im Gegensatz dazu wird es vermutlich auch nicht möglich sein kleine oder komplexere Objekte zu erkennen, weil der Tastsinn der Zunge dazu wiederum nicht sensibel genug ist.

Weil unser Gehirn sehr anpassungsfähig ist und beeindruckende Leistungen im Finden von Regelmäßigkeiten oder Mustern erbringt, ist der Ansatz andere Sinne zu verwenden eine vielversprechende Strategie. Darauf setzt auch unser Projekt, Dosuas, welches den Hörsinn verwenden möchte um Blinden eine Hilfe für Orientierung und Erkennung der Umwelt zu geben.

[1] https://www.wicab.com/brainport-v100

2 Echoortungsstrategie

2.1 Funktionsweise

2.2 Fazit

3 3D Kamera Strategie

3.1 Funktionsweise

In diesem Teilprojekt werden die Daten einer 3D Kamera als Töne kodiert, die der Träger des Geräts dann verwenden kann um ein Gefühl für den ihn umgebenden Raum zu bekommen.

3.1.1 Verwendete Technologien

Der wichtigste Teil dieses Projekts ist eine ToF (Time of Flight) Kamera, die neben normalen Fotos auch sogenannte Tiefenbilder aufnehmen kann. In einem Tiefenbild bekommt jeder Pixel einen Wert, der die Entfernung zur Kameralinse in mm angibt. Der von uns verwendete „Cube Eye MDC500C“[2] Sensor hat eine Reichweite von 0.8 bis 5.3 Metern und einer Auflösung von 320x240 Tiefenpixeln. Time of Flight Kameras messen die Entfernung mit Infrarotlicht. Deshalb funktioniert der Sensor auch im Dunkeln und wird von normalen Lichtreflektionen nicht gestört. Trotzdem hat der Sensor Probleme beim Erkennen von lichtdurchlässigen oder reflektierenden Objekten (z.B. Glasscheiben oder Spiegel). Für einen Prototypen ist das aber kein großes Problem. Wir haben diese Kamera ausgewählt, weil sie uns von einem Familienmitglied[3] empfohlen wurde.

Einen weiterer wichtiger Teil des Projekts stellen 3D-Audio Kopfhörer dar. Diese können den Eindruck erzeugen, dass sich eine Tonquelle im dreidimensionalen Raum befindet, bzw. sich bewegt. Dieses Verfahren benutzen wir um dem Träger des Geräts einen Eindruck davon zu geben in welcher Richtung sich ein Objekt befindet.

Die dritte Komponente ist ein enquoteRaspberry Pi, ein Einplatinencomputer auf dem ein Linux Betriebssystem läuft. Dieser ist mit Kopfhörern und ToF Sensor verbunden und führt unser Programm aus. Wir verwenden den Raspberry Pi, weil er klein, mobil und stromsparend ist.

Zusammen ergeben die drei Bestandteile (und eine mobile Stromquelle) einen Prototypen, den Sie hier in der Abbildung sehen können.

TODO: Bild hier

3.1.2 Softwarestruktur

Das Programm ist in C++ geschrieben, weil die API des ToF Sensors C++ erfordert und C++ eine schnelle Sprache mit vielen Möglichkeiten ist. Wir verwenden folgende Libraries:

1. *MTF API* - eine Schnittstelle mit der man den Cube Eye Sensor ansteuern kann
2. *PCL* - eine Bibliothek um mit Punktwolken[4] zu arbeiten, wir benutzen sie für Bildverarbeitung der 3D Daten
3. *SFML* - eine einfache Multimedia Bibliothek, die wir für das Abspielen von 3D Sounds verwenden

[2] http://www.cube-eye.co.kr/en/#/spec/product_MDC500d.html

[3] Jan Nicklisch, Vater von Yorick Zeschke, arbeitet in der Firma „DILAX“, die diese Sensoren für Personenzählsysteme verwendet

[4] Punktwolken sind (ggf. geordnete) Mengen von Punkten im dreidimensionalen Raum, wobei jeder Punkt eine x, y und z Koordinate und einen Index bekommt

Die Software selbst besteht im Moment aus 3 Modulen, die sich aber bis zur Ausstellung noch verändern können.

1. *sensorReader.cpp* - ein Modul, das die Schnittstelle zum ToF Sensor benutzt um Daten zu lesen und in eine Struktur für die weitere Verarbeitung zu bringen
2. *imageProcessor.cpp* - ein Modul zum Umwandeln der 3D Daten in eine Punktwolke und Vorbereiten bzw. Verarbeiten der Punktwolke
3. *audioPlayer.cpp* - ein Modul, welches die Vorbereiteten Daten in Töne umrechnet und diese dann abspielt

3.1.3 Programmablauf

Die Software läuft kontinuierlich in einer Schleife, bis sie beendet wird und macht alle 6 Sekunden ein Tiefenbild. Dieses wird vom Sensor als 2D Matrix von 320x240 (=76.800) natürlichen Zahlen dargestellt. In diesem Tiefenbild gibt jeder Pixel den Abstand zur nächsten lichtreflektierenden Region in mm an. Dadurch erscheint eine grade Fläche direkt vor dem Sensor jedoch auf dem Bild als nach außen gekrümmt, weil das vom Sensor gesendete Infrarotlicht zu den Seiten der Fläche (und zurück) ein bisschen länger braucht, als zur Mitte. Wegen der kugelförmigen Ausbreitung der Lichtstrahlen von der Kameralinse hat das Bild außerdem ein Polarkoordinatensystem[5], mit dem wir nicht gut weiterarbeiten können. Um das Problem der Verzerrung zu beheben und das Bild in ein kartesisches Koordinatensystem umzurechnen verwenden wir einen Undistortion[6] Algorithmus. Danach wird das Bild in eine Punktwolke mit kartesischem Koordinatensystem umgerechnet.

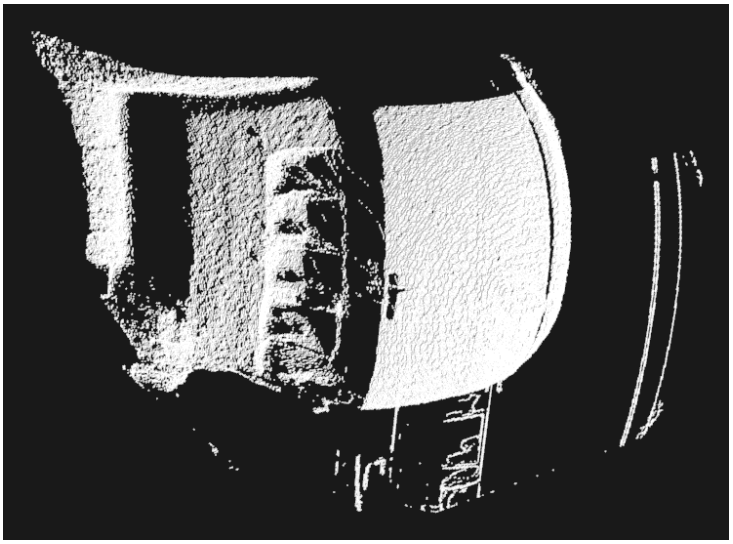

Abbildung 1: verzerrte Punktwolke mit Messfehlern, vor dem Undistortion-Algorithmus

TODO: Bild vom selben Zimmer! Wie in Abb. 1 zu sehen, wirkt ein ebene Fläche, in diesem Fall eine Wand, wie ein Teil der Oberfläche einer Kugel. In der nächsten Abbildung (Abb. 2) sieht man die Aufnahme des selben Zimmers, in der die Wand grade erscheint. Hier sind auch Schrank (rechts in der Ecke) und Decke besser zu erkennen.

[5] jeder Pixel gibt eigentlich einen Winkel zwischen 0° und 75° an, weil der Sensor ein Sichtfeld von 75° hat

[6] in unserem Fall einen vom Hersteller mitgelieferten, den man direkt in der Sensorkonfiguration einstellen kann

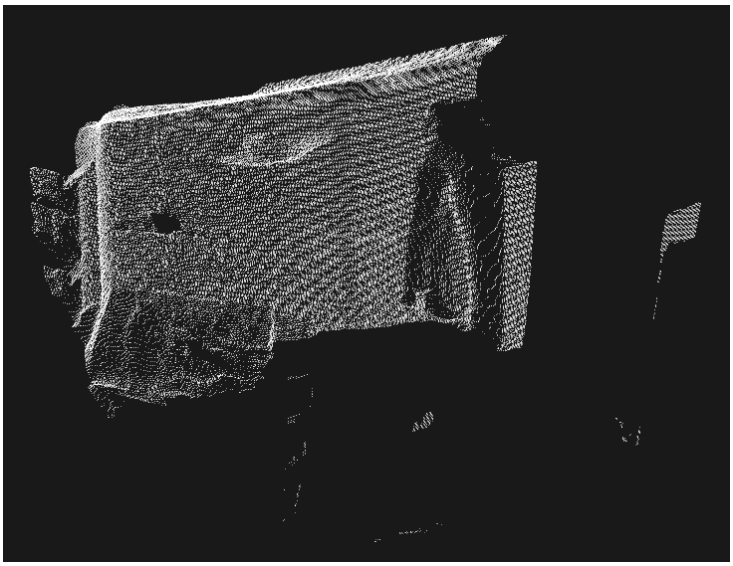

Abbildung 2: algorithmisch korrigierte Punktwolke ohne Verzerrung

Nachdem die Punktwolke korrigiert wurde, werden Boden und Decke entfernt. Momentan filtern wir einfach alle Voxel heraus, deren y-Koordinate kleiner als 40 oder größer als 200 ist[7]. Diese naive Methode soll später durch das Entfernen von Boden und Decke mit dem Flächenfindungsalgorithmus RANSAC geschehen. Hier sehen Sie eine Punktwolke nach der Filterung. Diese enthält nur noch ungefähr 62800 Punkte.

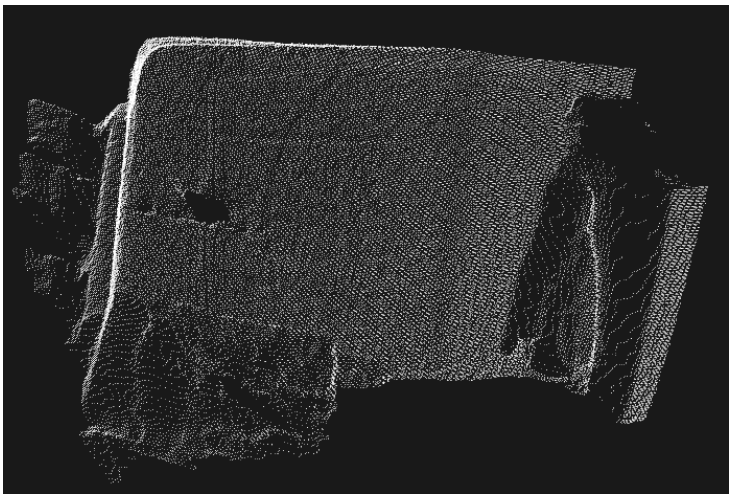

Abbildung 3: Punktwolke nach Entfernen von Decke bzw. Boden

Einmal wird die Punktwolke dann noch gefiltert, um ihre Größe zu reduzieren. Diese Art von Filterung nennt man Downsampling. Wir verwenden dafür den „VoxelGrid“ Algorithmus der „**P**oint **C**loud **L**ibrary“. Weil VoxelGrid die Punktwolke mithilfe eines Durchschnittsverfahrens reduziert, werden statistische Anomalien und Extremwerte ausgefiltert. Ein Bild mit genauer Auflösung eignet sich für unseren Zweck nicht so gut, wie ein niedriger auflösendes, denn bei hoher Auflösung können kleine oder ungewöhnlich geformte Regionen (z.B. Messfehler, kleine Gegenstände, Kanten, Ecken, etc.) beim Hören

[7]Koordinaten auf der y-Achse gehen von 0 (oberer Bildrand) bis 240 (unterer Bildrand)

einen ungewollten Eindruck erzeugen. Das Geschieht, weil unser Ohr sehr stark auf Änderungen in einem regelmäßigen Ton, wie wir ihn im Normalfall hören wollen, reagiert. Hier sehen Sie wieder den selben Raum, nur das die Punktwolke diesmal nur noch aus ungefähr 15650 (also ca. ein Fünftel der ursprünglichen Größe) Punkten besteht.

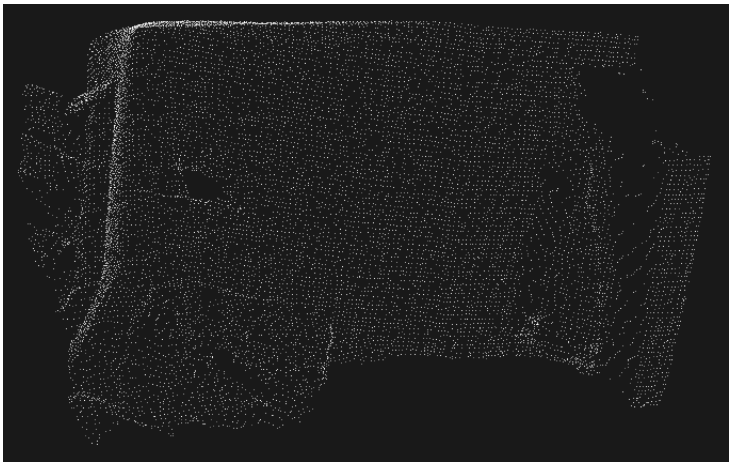

Abbildung 4: Punktwolke nach Downsampling, geringere Auflösung

Nach der Filterung wird das Bild in 320 vertikale Spalten unterteilt, die jeweils 240 Pixel hoch sind. Für jede Spalte wird aus den 5 „nächsten“ (auf den den Träger bezogen) Voxels [8] ein Voxel berechnet, dessen z-Koordinate (Tiefe) der Durchschnitt der 5 anderen Voxel ist. Seine x-Koordinate entspricht der Spaltennummer und die y-Koordinate ist ebensfalls der Durchschnitt der anderen 5 y-Koordinaten. 320 dieser Voxel ergeben einen radarscannähnlichen Streifen (horizontal) mit Tiefen- und Breiteninformationen. Dieser wird für das Tonabspielen benutzt.

Ein Ton, wir nennen ihn „Radar Swipe“, weil er einem Radarscann durch das ganze Bild ähnelt, bewegt sich immer vom linken zum rechten Rand des Sicht-, bwz. Hörfeldes und ändert dabei (meistens) fortwährend seine Frequenz. Durch diese wird eine Entfernung angegeben[9]. Zusammen mit der Position[10], welche man über die 3D-Audio Kopfhörer mitbekommt kann man sich mit etwas Übung ein gutes Bild der Umgebung und ihrer Beschaffenheit machen. Zum Beispiel könnte ein Ton, der in der linken Bildhälfte langsam tiefer wird und in der rechten Bildhälfte gleich bleibt eine schräge Wand darstellen, die in einiger Entfernung in eine zum Nutzer parallele Wand übergeht. So ein einfaches Beispiel kommt zwar selten vor, und meistens gibt es noch eine Menge Störgeräusche, aber näheres dazu im Abschnitt 3.2.

Momentan beträgt die Dauer zum Abspielen eines Bildes fünf Sekunden, gefolgt von einer Sekunde Pause. Bis zum Wettbewerb wollen wir die Dauer noch verkürzen, jedoch erfordert das ein gewisses Training, weil der Mensch üben muss, mehr Informationen in kürzerer Zeit zu verarbeiten. Zusammen mit einer Verkürzung der Pausen zwischen Aufnahmen hoffen wir die Zeit für einen Programmzyklus auf ungefähr eineinhalb bis drei Sekunden zu reduzieren. Im folgenden Programmablaufplan wird der Ablauf etwas präziser beschrieben.**TODO: PaP**

[8] so bezeichnet man einen dreidimensionalen Pixel mit x, y und z-Koordinaten
[9] tiefe Töne entsprechen großer Entfernung und hohe Töne einem nahen Objekt
[10] diese ist virtuell, hört sich aber wegen der Kopfhörer sehr realistisch an

3.2 Praxistest und Ergebnisse

4 Diskussion

4.1 Entwicklung

4.2 Ausblick

4.3 Nutzen und Fazit

5 Anhang