Freiform-Gesten in einem Natural User Interface mit Microsoft Kinect

Dominik Feininger
dominik.feininger@gmail.com
University of Applied Science Mannheim
2013

Abstract

Das Projekt „Earth Explorer“ handelt es sich um die Steuerung von Google Earth mithilfe von Frei Form Gesten. In dem Projekt wird eine Microsoft Kinect zur Gestenerkennung genutzt, dadurch ergibt sich eine Steuerung im stehend im freien Raum. In diesem Dokument wurden Erfahrungen über die Definition von Frei form Gesten aus dem MMI Projekt „Earth Explorer“ und der Weiterentwicklung zusammengefasst. Die Erfahrungen im MMI Projekt kommen eher aus der praktischen Richtung. Die Definition ist noch nicht funktionstüchtig implementiert und ist daher eine Zusammenfassung einer Internet Recherche. Die Entwicklung stützt sich wie viele Beispiele in anderen Kapitelten explizit auf eine Zoomgeste[1].

Kennenlernen der Menschlichen Motorik

Um zu verstehen, wie Gesten erkannt werden können muss verstanden werden, wie der Benutzer eine Bewegung ausführt. Dazu gehört auch welche Körperteile der Benutzer simultan mitbewegt, wenn er eine bestimmte Geste ausführt. So wird offensichtlich beim Bewegen der Hände zu einer Geste die Ellenbogen, Schultern usw. auch mitbewegt.

Klassifizierung der Gesten und des Persönlichen Raumes

Relativer Input / Fixer Output beschreibt ein Format bei dem der User frei in der Eingabe in der NUI ist, allerdings das Ausgabemedium fest installiert ist. Ein Display an der Wand ist immer fix und an der gleichen Stelle, unabhängig vom Nutzer. Der Benutzer muss sich im Raum ausrichten. Die Gesten bewegen sich im persönlichen und persönlichen Randraum. D.h. direkt am Körper und Reichweite der Arme.

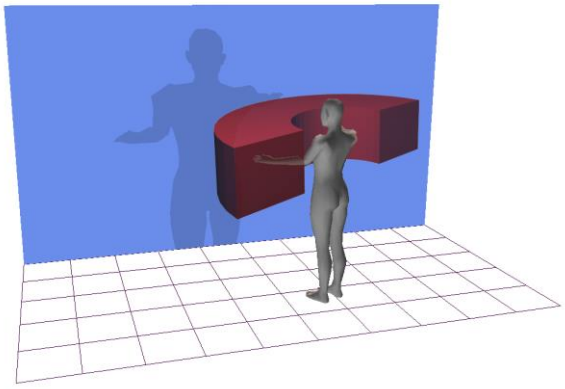

Abbildung 1 - persönlicher Randraum (rot)[2]

Definieren von Gesten

Definieren der Gesten kann überraschend schwierig sein. Die einfachsten Gesten sind diejenigen, die zu einem einzigen Zeitpunkt geschehen, und daher nicht auf vergangene Positionen der Gelenke zurückgreifen müssen. Zum Beispiel, wenn der Benutzer seine Hand über seinen Kopf hebt. Dies kann in einem einzelnen Frame überprüft werden. Komplizierte Gesten brauchen eine Zeit um als solche erkannt zu werden. Für eine zoom Geste, ist es nicht möglich aus einem einzigen Frame zu erkennen, ob eine Person zoomt oder einfach nur die Hand vor den Körper hält.

Das Programm muss in der Lage, relevante Informationen aus der Vergangenheit zu speichern, nur welche Informationen sind relevant? Sind die letzten 30 Frames genug Information? 30 Bilder bekommt man in nur einer Sekunde, vielleicht 60 Frames, also 2 Sekunden? Oder 5 Sekunden, also 300 Bilder? Menschen bewegen sich nicht so schnell. Vielleicht könnte man jeden fünften Frame speichern, dadurch würde die Länge bei 5 Sekundenwieder auf 60 Frames reduziert werden.

---

1 Geste mit beiden Armen, die einem vertikalen Pinch ähnelt.

2 2 Body-Centric Interaction Techniques for Very Large Wall Displays

Eine bessere Idee wäre die relevanten Informationen aus den Frames zu wählen. Für eine zoom Geste der Hände, die aktuelle Geschwindigkeit, wie lange sich bewegt wurde, wie weit wurden die Hände verschoben, etc. könnten nützliche Informationen sein.

Nachdem das definiert wurde, speichert man alle Informationen die im Zusammenhang mit der Geste stehen. Doch wie definiert man diese Informationen in Zahlen? Zoomen könnte eine bestimmte Mindestgeschwindigkeit, oder eine Richtung (links / rechts nach außen), oder eine Dauer erfordern.

Hier sind jedoch nicht die definierten 5 Sekunden von Interesse. Diese Dauer beschreibt das absolute Minimum[3] das benötigt wird um davon auszugehen, dass der Benutzer zoomt. Wie bereits erwähnt, kann das nicht anhand eines einzelnen Frames bestimmt werden. Es sollte nicht allein die Implementierte Dauer bestimmen, ob eine zoom Geste als solche erkannt wird. Da die Zeitspanne in der die Geste durchgeführt wird für jeden Benutzer unterschiedlich ist, da jeder Benutzer sich unterschiedlich schnell bewegt.

Die Chancen stehen gut, dass an einem gewissen Punkt innerhalb dieser 5 Sekunden die Geste mit der Definition übereinstimmt und ein zoomen erkannt wird. Es gibt aber ein weiteres Problem, wie definiert man den Start und Endpunkt einer Geste? Die Geste kann während des Ausführens langsamer und schneller werden oder gar pausieren. Fängt die Geste nach der Pause wieder von Anfang an?

Der Punkt, den ich versuche hier klar zu machen ist, es gibt keine einfache Möglichkeit Gesten zu erkennen. Als Entwickler muss man sich in die Geste hinein denken und eine Reihe von gemeinsamen Positionen der Skelett-Daten über eine Zeit hinweg als eine Geste zu definieren.

Benutzererwartung

Der Benutzer sollten nicht gezwungen werden eine Bewegung für einen bestimmten Zeitraum zu wiederholen. Es ist überraschend anstrengend und nicht einfach. Der Benutzer erwartet eher wie beim Computer eine Art „Point-and-Click“, sobald geklickt wird, wird eine Reaktion des Systems erwartet. Es ist sehr unnatürlich einen Mausklick (z.B. 5sekunden) halten zu müssen bevor eine Anwendung startet.

Wiederholen einer Geste über einen Zeitraum ist okay, wenn es sich um eine stetige Aktion handelt und stetig Rückmeldung vom System gegeben wird. Somit versteht der Benutzer, dass für diese Aktion eine kontinuierliche Ausführung erforderlich ist.

3 Minimum Zeitdauer zum Erkennen einer Geste

Fakten

Dem Benutzer ist es nur begrenzt möglich seine Hände still zu halten. Beim Versuch die Hände still zu halten erkennt die Kinect Bewegungsänderungen in der dritten Nachkomastelle. Man kann von einer gewollten Bewegung sprechen, fall sich werte an der zweiten Nachkommastelle über längere Zeit ändern[3]. Das Koordinatensystem, ist folgendermaßen angeordnet:

X-Achse = rechts/ links
Y-Achse = hoch/ runter
Z-Achse = vor/ zurück (Tiefe)

Wobei die X-Achse sich relativ zum Körper des Benutzers verhält, die Z-Achse sich jedoch nicht verschiebt, da der Ursprung der Z-Achse in der Kinect liegt.

Berechnungen

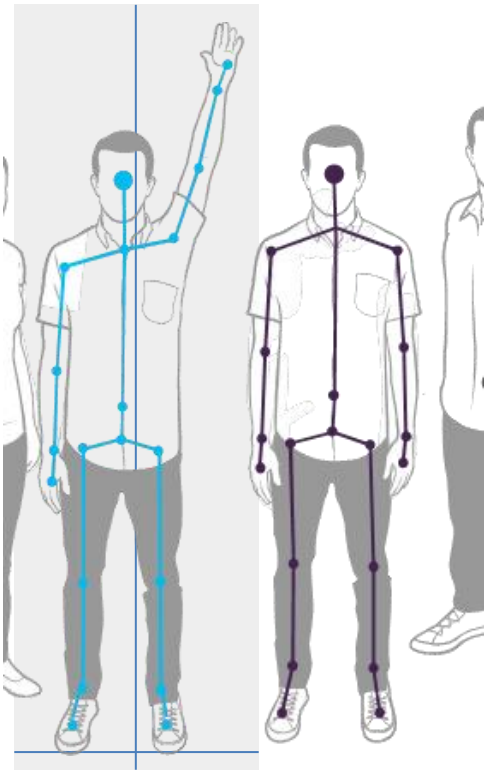

Abbildung 2 - Kinect User Skeleton[4]

Ausgegangen vom Körpermittelpunkt als Y-Achse befinden sich die linke Hand im negativen Bereich und die rechte Hand im positiven Bereich. Somit muss bei der Berechnung von Bewegungen jeweils Werte addiert bzw. subtrahiert werden. Der Bewegungsraum einer Hand wird sich im normalen Fall und bei optimaler Entfernung des Benutzers zur Kinect zwischen 0 und $^{+}/_{-}$ 0,8 befinden.

Übersetzten von Gesten

Eine wichtige Größe bei der Übersetzung von Gesten auf ein beliebig großes Display oder Leinwand bringt einige

4http://flurfunk.sdx-ag.de/2013/03/bewegungsdrang-teil-5-die-kinect-geht.html

Komplikationen mit sich. Es besteht ein Übersetzungsverhältnis von Geste zum erkennenden Gerät und zum Display. Die Bewegung die der User ausführt steht in einem bestimmten direkt proportionalen Verhältnis zu der Aktion auf dem Display. Um dieses Verhältnis genau zu bestimmen und in verschiedenen Umgebungen gleich zu halten muss eine Berechnungsmethode gefunden werden die die beteiligten Parameter verbindet und daraus eine Proportionale Umgebung schafft. Parameter dieser Gleichung sind z.B. Abstand von User zum Erkennungsgerät, Größe des Displays und Auflösung des Displays.

Theoretisch kann der Eingaberaum, in dem sich der User bewegt sehr groß sein, da keine Physische Nähre gefordert ist. Hier limitiert jedoch der Sensor zum Aufnehmen den Bewegungen den Maximalen Bereich. Weiterhin ist zu beachten, dass die Eingabe ohne eine Berührungen einer Festen Oberfläche deutlich langsamer und behäbiger abläuft als bei Interaktionen mit einer festen Oberfläche.

Erfahrungen

Zu allen Überlegungen aus Kapitel 0 kommen in der tatsächlichen Umsetzung weitere Konstanten die für bestimmte Gesten sinnvoll sein können. Zusätzlich zu dem Minimum an Distanz die die Hand zwischen zwei gemessenen Frames zurücklegen muss kann ein Wert, der ein Minimum für die absolute gesamte Bewegungsdistanz festlegt sinnvoll sein. Damit können kleinere Zuckungen des Benutzers und smoothe Endpunkte der Gestenrückmeldung erzeugt werden.

Grundsätzlich erzeugt die Steuerung über die Kinect eine Zeitverzögerung vom Ausführen der Geste, bis zur Anzeige auf dem Bildschirm von ca. einer viertel Sekunde[5] pro Frame. Das kann, falls eine zu geringe Frame Rate benutzt wird zu Störungen der direkten Systemrückmeldung führen da sich die Summe aus Lack und gemessenem Frame Abstand sich auf bis zu einer halben Sekunden belaufen können.

In Abbildung 3 wir veranschaulicht, wie sich der Aktionsradius des Nutzers auf den Erkennungsbereich der Kinect auswirkt. Um bei Ausgestreckten Armen einen Abstand vom Benutzer zur Kinect in Z-Richtung angeben zu können muss bekannt sein, in welche Richtung der Nutzer die Arme streckt. Es sollte, rein für die Geste des Zoomens auf ein 2D Koordinatensystem zurückgegriffen werden. Dieses Koordinatensystem, das sich in Z- und X-Richtung erstreckt kann beliebig unterteilt werden.

Implementierung

Liste der Attribute:

```
//30frames per second
private static int totalFramesInInterval =
120;//4sec
//saves the skeletons
private Skeleton[] SkeletonFrames = new
Skeleton[totalFrames + 1];
//frame interval
private int frameInterval = 3;
private float minMovementPerFrame = 0.01F;
private float minMovementInTotal = 0.03F;
private float tollerance = 0.01F;
//calc from screensize and distance
private float proportion;
private float screenDPI;
//generic value to steer the Google API
private float zoomspeed = 3;
//in cm
private float distanceZUserToSensor = 200;
private float screenWidth;
```

Da aktuell nur die Zoomgeste bestand der Versuche ist ist der Wert `zoomspeed` der einzige Wert in der Liste der sich direkt auf eine Geste bezieht. Dieser Wert steuert die Geschwindigkeit mit der sich die Karten auf dem Display bewegt.

Um eine passende Beziehung zwischen `proportion`, `screenDPI`, `distanceZUserToSensor` und `screenWidth` herzustellen sind verschiedene Test nötig.

Mapping, Geste – visuelle Rückmeldung

Entscheidend für das sichere Gefühl für den Nutzer entsteht unteranderem auch daraus, dass die Bewegung wie erwartet umgesetzt wird und die Rückmeldung auf dem Display ein passendes Mapping wiederspiegelt. Dieses Mapping wird von verschiedenen Parametern bestimmt. Parameter, die von dem Setup der Umgebung abhängen sind z.B. die Entfernung des Nutzers zum Display, die PPI[6] des Displays und die Größe des Displays.

Interaktionsformen für NUIs im Kontext von Earth Explorer

- Ganz Körper. Hauptinteraktion über ganze Arme, Kopfbewegung und Füße.

- Interaktion unterstützt durch Finger.

- "dumme Tangibles" wie einfache Gegenstände, die als Inputparameter erkannt werden.

- "intelligente Tangibles" wie Sphero oder z.B. Infrarot LEDs

- Body Centered Design, der Körper wird als Sensitive Oberfläche verwendet.

- Interaktion unterstützt durch Gesichtsmimik

- Interaktionsunterstützung durch Schatten[7]

Quellen

5 Gemessen durch Benutzung

6 Pixel per Inch

7 4 Videoplace – an artificial reality

1. Julie Wagner, Mathieu Nancel, Sean Gustafson, St´ephane Huot, Wendy E. Mackay - Body-centric Design Space for Multi-surface Interaction - CHI 2013: Changing Perspectives, Paris, France
2. Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, Kellogg S. Booth - Body-Centric Interaction Techniques for Very Large Wall Displays - NordiCHI (2010),
3. Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, Wendy Mackay - Mid-air Pan-and-Zoom on Wall-sized Displays - May 7-12, 2011 Vancouver, BC, Canada
4. M. W. Krueger, T. Gionfriddo, K. Hinrichsen.- Videoplace – an artificial reality - CHI ’85, 1985