**Fallstudie - Freiform-Gesten mit Microsoft Kinect**

Dominik Feininger

dominik.feininger@gmail.com

University of Applied Science Mannheim

2013

# Abstract

Bei dem Projekt „Earth Explorer“6 handelt es sich um die Steuerung von Google Earth mit Hilfe von Freiform Gesten. In dem Projekt wird eine Microsoft Kinect zur Gestenerkennung genutzt, dadurch ergibt sich eine Steuerung stehend im freien Raum. In diesem Dokument wurden Erfahrungen über die Definition von Freiform-Gesten aus dem MMI Projekt „Earth Explorer“ und der Weiterentwicklung zusammengefasst. Die Entwicklung stützt sich wie viele Beispiele in anderen Kapiteln, explizit auf eine Zoom-Geste[[1]](#footnote-1).

# Kennenlernen der Menschlichen Motorik

Um zu verstehen, wie Gesten erkannt und als Eingabequelle 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 werden zum Beispiel beim Bewegen der Hände zum Darstellen einer Geste die Ellenbogen, Schultern usw. auch mitbewegt.

# Klassifizierung der Gesten und des Persönlichen Raumes

Relativer Input / Fixer Output beschreibt ein Format, bei dem sich der User frei in der NUI[[2]](#footnote-2) bewegen kann, das Display (Ausgabemedium) allerdings fest installiert ist. Ein Display an der Wand ist immer fix, unabhängig von den Bewegungen des Nutzers. Der Benutzer muss sich im Raum anhand des Displays ausrichten. Die Gesten bewegen sich im persönlichen Raum und persönlichen Randraum. D.h. direkt am Körper und Reichweite der Arme.



Abbildung 1 - persönlicher Randraum (rot)[[3]](#footnote-3)

# 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 sein, 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 150 Bilder? Menschen bewegen sich nicht so schnell. Vielleicht könnte man jeden fünften Frame speichern, dadurch würde die Länge bei 5 Sekunden wieder auf 30 Frames reduziert werden.

Eine bessere Idee wäre die relevanten Informationen aus den Frames zu wählen. Zum erkennen einer Zoom-Geste der Hände können Parameter wie die aktuelle Geschwindigkeit, wie lange sich bewegt wurde, wie weit wurden die Hände verschoben, etc. nützliche Informationen sein um verschiedene Reaktionen des Systems auszulösen.

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

Hier sind jedoch nicht die definierten 5 Sekunden von Interesse. Diese Dauer beschreibt das absolute Minimum[[4]](#footnote-4) 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 und 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?

Den Punkt, den ich versuche hier klar zu machen ist dass es gibt keine eindeutige Möglichkeit Gesten zu erkennen gibt. Als Entwickler muss man sich in die Geste hinein denken und eine Reihe von Positionen der Skelett-Daten (Position der Gelenke) über eine Zeit hinweg beobachten um eine Bewegung als eine Geste definieren zu können.

# Benutzererwartung

Der Benutzer sollten nicht gezwungen werden eine Bewegung für einen bestimmten Zeitraum zu wiederholen. Denn es ist überraschend anstrengend und nicht einfach, stattdessen ermüdend und frustrierend. 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. 5 Sekunden) 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.

# Theoretische Umsetzung

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. In diesem Wertebereich kann man nicht von einer gewollten Bewegung sprechen. Erst sobald sich Werte ab der zweiten Nachkommastelle über längere Zeit ändern[3] kann in diesem Beispiel von einer Konkret gewollten Bewegung gesprochen werden. Das Koordinatensystem, in dem sich der Benutzer aufhält, ist folgendermaßen definiert:

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 im Kinect Sensor liegt.

# Berechnungen

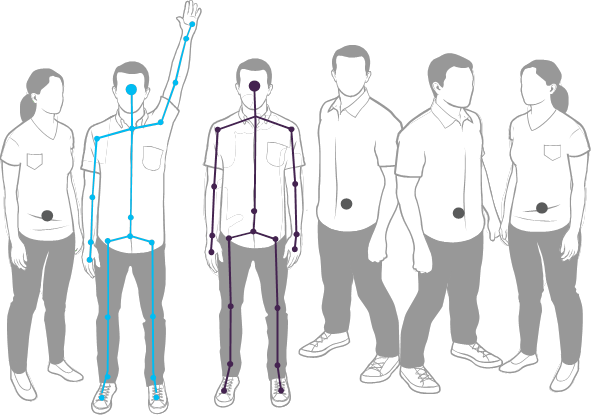


Abbildung 2 - Kinect User Skeleton[[5]](#footnote-5)

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 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 zur Rückmeldung 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. Dieser Parameter gleicht die vorhandenen Größen und Distanzen aneinander an, sodass eine vergleichbare Basis in verschiedenen Umgebungen zu Stande kommt. Parameter dieser Gleichung sind z.B. Abstand vom User zur Kinect, Größe des Displays und Auflösung des Displays.

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

# Erfahrungen

Zu allen Überlegungen aus dem letzten Kapitel 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 wird 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. Das kann, falls eine zu geringe Frame Rate benutzt wird, zu Störungen der direkten Systemrückmeldung führen, da sich die Summe aus Verzögerung und gemessenem Frame Abstand auf bis zu einer halben Sekunden belaufen kann.

In Abbildung 2 - Kinect User Skeleton wird 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 in beliebig große Raster unterteilt werden.

# Implementierung

Liste der Attribute:

//Verzögerung, die minimal Bewegung, mit der der Effekt der Geste einsetzt

public float minMovementFrame = 0.01F;

//die Geschwindigkeit der Geste im Verhältnis zur Geschwindigkeit mit der sich das Objektes vergrößert/verkleinert

public float zoomspeed = 5;

//das Verhältnis zwischen Gestengröße und Effektgröße (Vergrößerung/Verkleinerung des Objektes)

public float proportionParam = 1;

//Aktivitätsradius/-feld der Geste (Aktionsbereich des Nutzers)

public float handMovingZTollerance = 3;

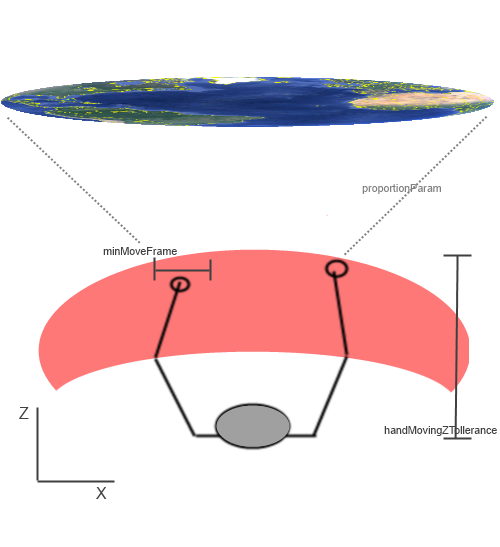


Abbildung - Veranschaulichung der Parameter

Da aktuell nur die Zoom-Geste bestand der Versuche 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 (Objekt) auf dem Display bewegen. Dieser Wert wird als einziger an das Google Earth Framework weitergegeben und verarbeitet.

Der Wert proportionParam beeinflusst die Reaktion des Systems auf die Entfernung der Hände des Nutzers zueinander. Dieser Parameter bildet die Größe von Objekten ab. Umso weiter die Hände auseinander sind, desto langsamer (0<x<1) / schneller (x>1) bewegt sich das Objekt.

Der Parameter minMovementFrame bedeutet, wie weit sich eine Hand mindestens bewegen muss um als Geste erkannt werden zu können.

Um des Nutzers Aktionsradius zu bestimmen, wird der Parameter handMovingZTollerance genutzt. Dieser beschreibt die minimale Entfernung zwischen dem Körper des Nutzers und dem Aktionsradius seiner Hände in Richtung des Kinect Sensors (Z-Achse). Sobald sich beide Hände außerhalb dieser Mindestentfernung befinden wird die Gestensteuerung aktiviert.

# User Interface der Testapplikation



Abbildung 4 - User Interface

Der Kern der Applikation ist das große obere Feld welches die Weltkugel beinhaltet. In der unteren Zeile sind von links nach rechts folgenden Felder zu sehen, eine live Darstellung des Skelettes des Nutzers, das Icon der Applikation, der aktuelle Status der Applikation und die Eingabefelder für Testwerte der Applikation.

# Mapping, Geste – visuelle Rückmeldung

Entscheidend für das sichere Gefühl des Nutzers ist unteranderem auch, 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]](#footnote-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]](#footnote-7)

# Usability Study

Zur Durchführung der Usability Study wird ein zweites digitales Referenzobjekt zum Testen bereitgestellt. Es handelt sich hier um ein einfarbiges 2D Objekt, das mithilfe der gleichen Gesten wie die Weltkugel vergrößert und verkleinert wird. Dieses zweite Testobjekt soll eventuelle Einflüsse die von der Oberfläche des zu bewegenden Objektes stammen entkoppeln und das Ergebnis verifizieren. Ausgetauscht werden die beiden Objekte durch tauschen der HTML Seite. Die im Userinterface () dargestellt wird.

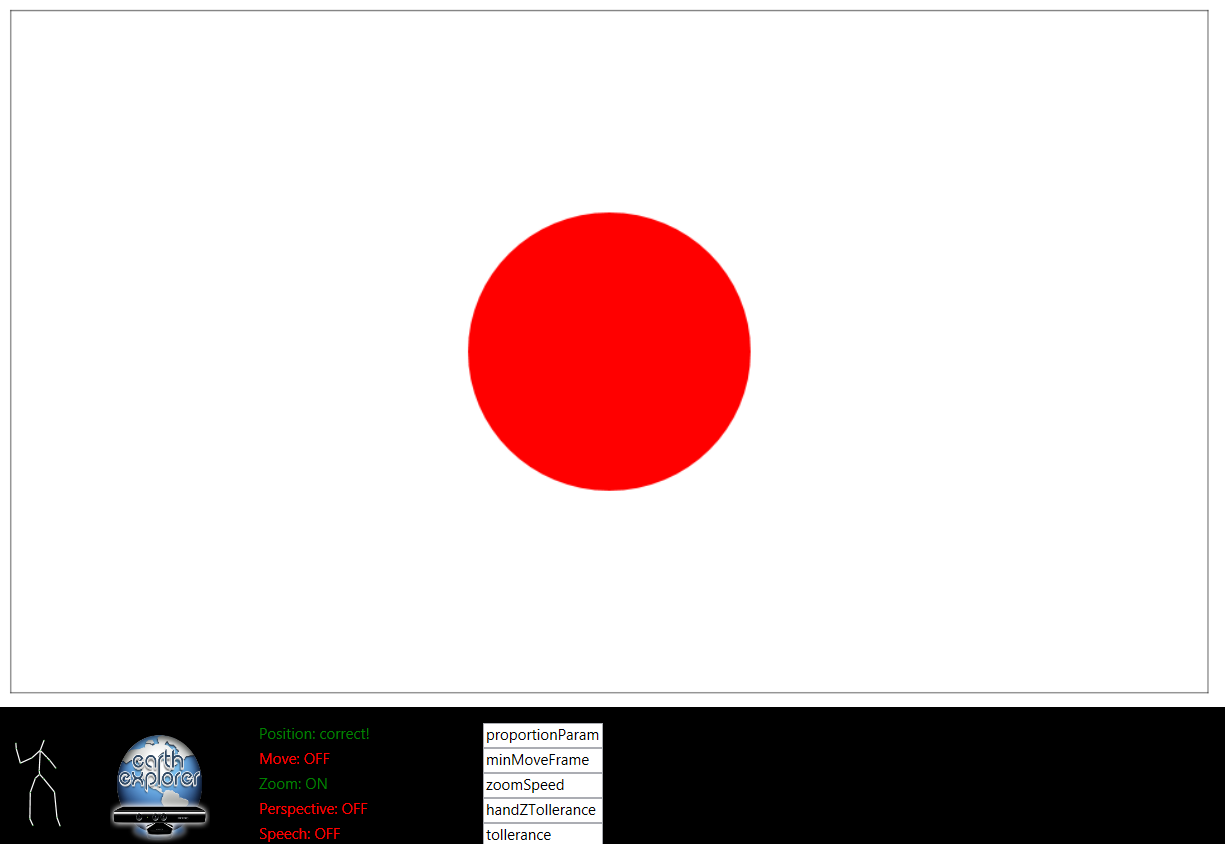


Abbildung - User Interface (Usability Study)

# Quellen

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
5. Thinkmoto - http://www.thinkmoto.de/gestures/
6. Earth Exlorer – R.Braun, V.Hermann, D.Feininger, University of Applied Sciences Mannheim, http://www.youtube.com/watch?v=rqMDSeDHWVg ,2013

1. Geste mit beiden Armen, die einem vertikalen Pinch ähnelt. [↑](#footnote-ref-1)
2. Natural User Interface [↑](#footnote-ref-2)
3. Body-Centric Interaction Techniques for Very Large Wall Displays [↑](#footnote-ref-3)
4. Minimum Zeitdauer zum Erkennen einer Geste [↑](#footnote-ref-4)
5. http://flurfunk.sdx-ag.de/2013/03/bewegungsdrang-teil-5-die-kinect-geht.html [↑](#footnote-ref-5)
6. Pixel per Inch [↑](#footnote-ref-6)
7. Videoplace – an artificial reality [↑](#footnote-ref-7)