Projektseminar: Gestensteuerung einer 3D-Anwendung mittels Kinect

Mario Janke
Peter Lindner
Patrick Stäblein

Inhaltsverzeichnis

1 Rahmenbedingungen

1.1 Grundlagen & Technik

Gegeben ist eine bereits vorhandene 3D-Anwendung, die zu Demonstrationszwecken genutzt wird. Innerhalb der Anwendung ist es möglich,

- Objekte zu laden und damit anzeigen zu lassen sowie
- die Kamera (bzw. Kameras) zu manipulieren, d. h. zu bewegen, zu rotieren und zu zoomen,

zusätzlich geplant ist später

- geladene Objekte manipulieren, in diesem Falle skalieren oder löschen zu können.

Das Programm rendert dabei zwei Ausgabefenster, in denen die Szene dargestellt ist, wobei die Kameras 3D-Aufbau bilden.
Die so beschriebene Ausgabe wird über zwei Projektoren von hinten auf eine Projektionsfläche geworfen – ein Projektor für die linke Kamera und einer für die rechte. Wird die „Leinwand“ von vorne durch eine Shutterbrille betrachtet, entsteht der 3D-Eindruck.
Die Steuerung der Anwendung erfolgt über Tastatur und Maus bzw. Präsentationspointer.

1.2 Aufgabenstellung

Ziel des Projektseminars ist es, die Steuerung der Anwendung hinsichtlich einer Präsentation vor einer Zuschauergruppe zu erleichtern und intuitiv zu gestalten, sodass parallel an der Universität vorhandene (und bislang ungenutzte) Technik verwendet und präsentiert werden kann. In diesem Sinne geeignet und vorgeschlagen sind

- ein professionelles Trackingsystem zum Tracken von Raumpunkten und
- die Verwendung einer Microsoft Kinect 2 zur Gestenerkennung.

Das damit entwickelte Programm soll Folgendes leisten:

- Es soll in der Lage zu sein, sämtliche Steuerung und Manipulation, die oben beschrieben wurde durchzuführen.

- Die Bedienung soll sehr intuitiv und einfach sein, d. h. etwaige Gesten müssen bezüglich der ihnen zugeordneten Aktion einleuchtend und leicht auszuführen sein.
- Das Programm soll möglichst einfach eingebunden und wiederverwendet werden können.

1.3 Eigenschaften der Kinect

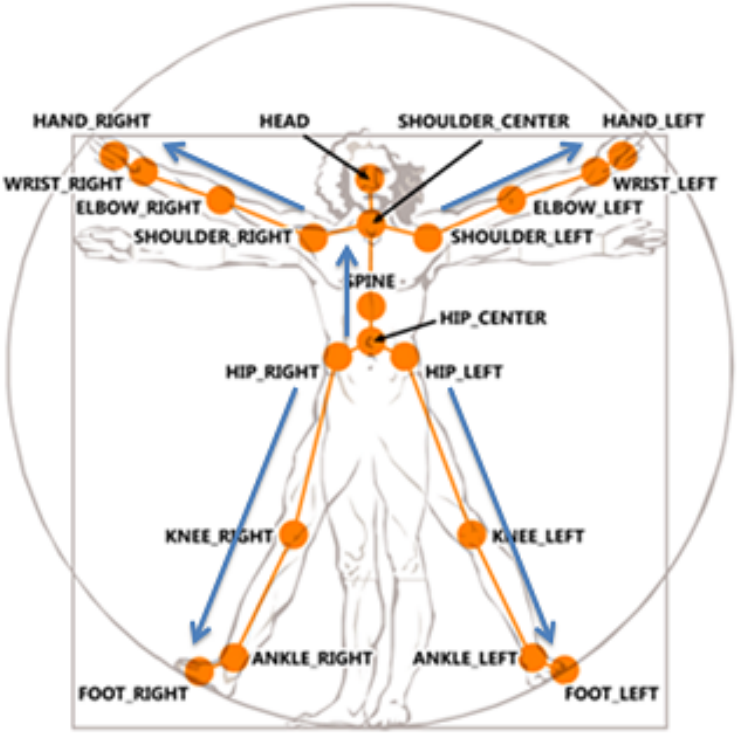


Abbildung 1: Gelenkpunkte die mit der Kinect getrackt werden können

[1] Wir stellen in diesem Abschnitt nur die für uns interessanten Eigenschaften und Möglichkeiten der Kinect vor (hinsichtlich unserer Aufgabe und der Rahmenbedingungen). Die Kinect erkennt visuell den 3D-Raum vor sich. Dabei werden Personen als solche detektiert und konfidenzbasiert mit einem primitiven und grobgranularen Skelett ausgestattet. Dieses Tracking ist für bis zu sechs Personen zeitgleich möglich. Weiterhin wird für beide Hände einer getrackten Person ein „Handzustand“ erkannt,

nämlich ob die Hand offen oder geschlossen ist, oder die sogenannte Lassogeste gebildet wird (etwa nur zwei Finger ausgestreckt). Kann einer Hand keiner dieser Zustände zugeordnet werden, ist ihr Status unbekannt. Diese Daten (Skelett und Status pro getrackter Person) können unter Verwendung der USB-Schnittstelle und des Kinect-SDKs abgegriffen werden. Sie werden dafür 30 mal in der Sekunde zur Verfügung gestellt.

2 Vorüberlegungen

Für unser Vorgehen zentral sind die folgenden beiden Bereiche:

1. die technische Umsetzung, d. h.
    - das korrekte Erkennen und Werten von Gesten einer ausgezeichneten getrackten Person
    - das korrekte Berechnen notwendiger Bewegungsparameter
    - die Einbindung in die bestehende Applikation
2. die Interaktion mit dem Benutzer, d. h.
    - das Entwerfen intuitiver und eingängiger Gesten für die verschiedenen Zwecke
    - das Auszeichnen einer getrackten Person als „Master“, der das Programm steuert

Wir stellen in diesem Abschnitt die zentralen unmittelbaren Beobachtungen vor, die sich aus der Aufgabenstellung und dem Versuchsaufbau ziehen lassen.

Ausgehend von der Aufgabenstellung kann man abstrahierend zwischen zwei primitiven Steuerungsmodi unterscheiden:

- einem Modus, in dem die Kamera verschoben und rotiert werden kann &
- einem Modus, in welchem Objektmanipulationen möglich sind.

Der Benutzer sollte sich zu jedem Zeitpunkt nur in maximal einem dieser Modi aufhalten, d. h. gleichzeitige Kamera- und Objektmanipulation wird ausgeschlossen. Diese Vereinfachung treffen wir, da damit weniger komplexe Gesten benötigt werden und eine

solche simultane Manipulation keine praktische Relevanz besitzt. Für Manipulationen, die man sowohl für die Kamera, als auch für Objekte haben will, bietet dies zudem eine geeignete Kapselung, da z. B. Rotationsparameter berechnet werden und dann nur entschieden werden muss, ob sie auf die Kamera oder ein Objekt angewendet werden, je nach Modus. Dies reduziert die Gesamtzahl nötiger Gesten.

Die Kinect ermöglicht ein Tracking des gesamten Körpers für mehrere (genauer sechs) Personen. Wir beschränken uns aus naheliegenden Gründen jedoch auf einen Teil dieses Spektrums:

- Wir benötigen nur eine Person, die die Anwendung (möglichst ungestört) steuert. Eine genauere Auswertung der restlichen Personen, ihrer Skelette etc. ist unnötig.
- Die in unserem Anwendungsfall intuitiven Gesten werden ausschließlich mit den Händen (bzw. Armen) durchgeführt.

Primitive Erkennungsmöglichkeiten eines Masters kann man etwa aus der Entfernung der getrackten Personen zur Kamera und der Position der Personen im Raum gewinnen. Genauere Erklärungen folgen weiter unten.
Intuitive Gesten für Verschiebungen imitieren das Verschieben eines großen Gegenstands, etwa einer imaginären Box, sodass hier etwa ein Verschieben der flachen Hand in der Luft naheliegt. Für eine intuitive Drehgeste eignet sich die Vorstellung eines imaginären Lenkrads, genauer gesagt einer Lenkkugel, bei der die Rotation um eine Raumachse nach dem Lenkradprinzip erfolgt. Eine intuitive Geste zur Objektauswahl ist offenbar eine Greifgeste.

3 Entwurfsentscheidungen

3.1 Der Master

Der Master ist die Person (unter den getrackten Personen), der es obliegt, die Anwendung zu steuern, d. h. in unserem Anwendungsfall der Präsentation ist der Master der Präsentierende.
Es muss gewährleistet werden, dass nur der Master das Programm steuert und dabei von weiteren Personen im Raum nicht (bzw. nicht ohne weiteres) gestört werden kann. Die Erkennung muss robust gegen Jittering der Kinectdaten sein.

3.2 Gesten und ihre Wirkung

Wie oben erwähnt, entscheidet ein „globaler“ Modus, ob wir uns bei der Kamera- oder der Objektmanipulation befinden. Daher können wir ein und dieselbe Geste für das Verschieben der Kamera bzw. eines Objekts verwenden (Rotation analog). Die hier angegebenen Gestenbezeichner werden so auch im Quellcode verwendet.

TRANSLATE_GESTURE Der Benutzer hat beide Hände geöffnet, mit den Handflächen zur Kamera (wichtig ist nur, dass die Kinect beide Hände als offen erkennt, die genaue Haltung ist dabei egal). Ein paralleles Verschieben der beiden Hände in eine Richtung bewirkt ein zur Bewegungsgeschwindigkeit proportionales Verschieben der Kamera bzw. des Objekts in diese Richtung.

ROTATE_GESTURE Der Benutzer hat beide Fäuste geballt. Dann bewirkt eine gleichzeitige Bewegung der Hände auf einer Kreisbahn eine Rotation der Kamera bzw. des Objekts um die Senkrechte des zugehörigen Kreises.

GRAB_GESTURE Der Benutzer schließt eine Hand und behält die andere geöffnet.

UNKNOWN Dies enthält alles, was als keine der anderen Gesten erkannt wird.

Tests mit der Kinect haben ergeben, dass es notwendig ist, bei derartig selbst implementierten Gesten auch eigene Robustheitsmechanismen einzubauen, die die Gestenerkennung gegen Schwankungen der Kinecterkennung (etwa des Status einer Hand) abhärten.

3.3 Zustandsmaschine

Durch die Grundmodi „Kameramanipulation“ und „Objektmanipulation“ werden zwei Superzustände definiert, innerhalb deren die erkannte Geste die Aktionen bestimmt. Da je erkannter Geste andere Arbeit geleistet wird, bietet es sich an, diese wiederum in Zustände zu kapseln. Von außen folgt unser Tool daher dem folgenden groben Schema:

- Aufruf aus Hauptprogramm
- Auswertung der Kinectdaten
- Gestenerkennung

- Berechnung im aktuellen Zustand
- Etwaiger Zustandswechsel

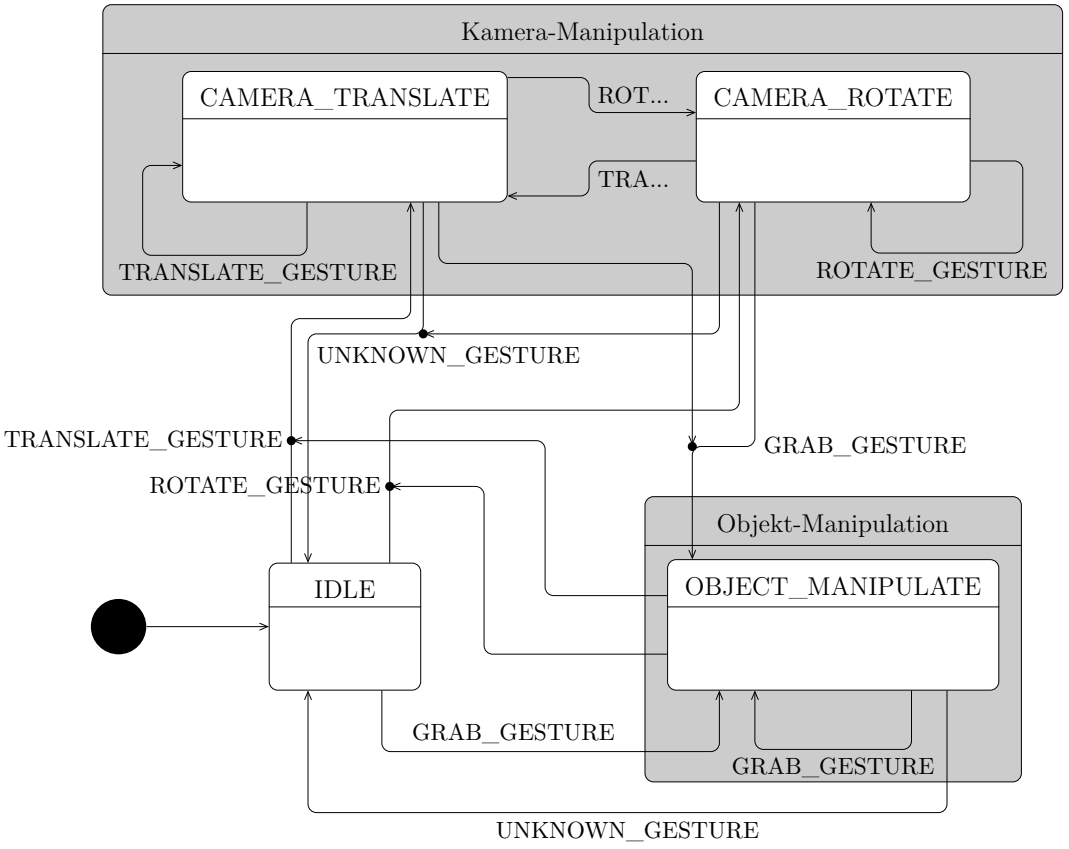
Die letztendlichen Zustände und das dazugehörige Zusammenspiel sind ein Ergebnis unserer Überlegungen zusammen mit dem Feedback, das wir aus unseren vielen Tests gewonnen haben. Ursprüunglich war es angedacht, lediglich zwischen einem Kamera- und einem Objektzustand zu unterscheiden, also einem Zustand zur Kamera- und einem zur Objektmanipulation, wobei dann die entsprechende Geste bestimmt, wie manipuliert wird. Die oben genannten Subzustände haben sich dann aus den Betrachtungen zur Einfachheit und Intuitivität ergeben. Im Folgenden stellen wir die Zustände unserer Zustandsmaschine vor und geben dabei an, was jeweils berechnet wird und einen Zustandswechsel herbeiführt.

CAMERA_IDLE Zweck dieses Zustands ist es, eine Art Default-Zustand bereitzustellen, in dem keine Kamera- und auch keine Objektmanipulation vorgenommen wird. Der Zustand wird betreten, wenn keine der vordefinierten Gesten sicher genug erkannt wurde. Durch Ausführung der entsprechenden Gesten gelangt man zurück in die anderen Zustände.

CAMERA_TRANSLATE In diesem Zustand werden gemäß der oben erklärten Geste die Parameter zur Kamerabewegung bestimmt. Wir berechnen dazu aus den gepufferten Positionswerten von linker und rechter Hand die diskrete Ableitung, die uns ein Maß für die Geschwindigkeit der Bewegung liefert. Ebenso erhält man daraus die Richtung, in die die Hände bewegt wurden. Aus diesen Größen berechnen wir Translationsparameter für die x-, y- und z-Richtungen, die für diesen Zustand unsere `motionParameters` definieren.

CAMERA_ROTATE Analog wird nun in diesem Zustand die Rotation vorbereitet.

Genaueres zum Aussehen der State-Machine als Datenstruktur ist in Abschnitt ???? zu finden.


Kamera-Manipulation
CAMERA_TRANSLATE
ROT...
CAMERA_ROTATE
TRA...
TRANSLATE_GESTURE
ROTATE_GESTURE
UNKNOWN_GESTURE
TRANSLATE_GESTURE
GRAB_GESTURE
ROTATE_GESTURE
Objekt-Manipulation
IDLE
OBJECT_MANIPULATE
GRAB_GESTURE
GRAB_GESTURE
UNKNOWN_GESTURE

3.4 Robustheit und Pufferung

Wie wir vorangegangen festgestellt haben, sind einige der Mechanismen, die wir implementieren wollen anfällig gegenüber qualitativ niedrigwertigen Kinectdaten. Tests mit der Kinect haben folgende kritische Situtationen ergeben:

- Gelenke und Skelettbestandteile in der Nähe von Objekten und anderen Personen. Diese können falsch oder verzerrt erkannt werden. So kann etwa die erkannte Handposition zwischen zwei Kinectframes Raumunterschiede von mehreren Metern aufweisen und zurückspringen.
- Status der Hände. Auch bei durchgängiger Aufrechterhaltung eines Handzustands kann es passieren, dass die Kinect vereinzelt falsche Zuweisungen trifft oder keine Zuweisung möglich ist.

Beide Situationen lassen sich behandeln, indem Entscheidungen unseres Programms, nicht nur vom augenblicklichen Rückgabewert der Kinect abhängen, sondern auch einige vergangene Werte mit einbeziehen. So kann ermittelt werden, ob der aktuelle Wert (mit hoher Wahrscheinlichkeit) ein zu ignorierender Ausreißer ist. Dazu wird ein Ringpuffer verwendet und an den entsprechenden Stellen im Quellcode ein gewichtetes Mittel über den Pufferinhalt gebildet, wobei neuere Einträge mit deutlich größerem Gewicht eingehen. Für Gesten kann dann mit einer bestimmten Zuverlässigkeit eine Zuordnung getroffen werden, für Raumpositionen stellt dieses Mittel eine Glättung dar. Dies hat den positiven Nebeneffekt, dass die endgültige Anwendung der errechneten Parameter auf die Kamera bzw. das Objekt ebenfalls geschmeidiger werden.

4 Bemerkungen zum Quellcode

4.1 Wichtige Datenstrukturen, Variablen und Funktionen

4.2 Details zum Zusammenspiel

4.3 Einbinden

Literatur

[1] Microsoft Developer Network. *Skeletal Tracking*. `https://msdn.microsoft.com/en-us/library/hh973074.aspx`. [Online; accessed 25-April-2017]. 2017.