

Markerloses Tracking mithilfe Analyse durch Synthese auf Basis eines Partikelfilters

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science (B.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Maximilian Luzius

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)
Zweitgutachter: Anna Katharina Hebborn, M.Sc.
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im November 2014

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden. ☐ ☐

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ☐ ☐

.....
(Ort, Datum)

.....
(Unterschrift)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufbau der Arbeit	1
1.2	Stand der Technik	2
1.3	Markerbasiertes Tracking	2
1.4	Markerloses/Modellbasiertes Tracking	2
1.5	Kantenbasierte Bildvergleiche?	3
2	Grundlagen	3
2.1	Analyse durch Synthese	3
2.2	Partikelfilter	4
2.3	Sobelfilter	4
3	Implementierung	4
3.1	Software/Hardware	4
3.2	Bibliotheken	5
3.3	Anforderungen	6
3.4	Umsetzungsansatz	6
3.5	Details der Implementierung	6
4	Ergebnisse	6
4.1	GUI?	6
4.2	Performance	6
5	Ausblick und Fazit	6
5.1	Verbesserungen	6
5.2	Fazit	6
6	Anhang	6
7	Fragen	6
7.1	1.1	6
7.2	1.2	7

1 Einleitung

Im Bereich der Augmented Reality(kurz AR) weist Tracking, besonders markerloses Tracking, eine der größten Schwierigkeiten auf. Unter Augmented Reality wird verstanden, dass ein Kamerabild durch synthetische Informationen in Echtzeit überlagert oder auch ergänzt wird, so dass die Realität erweitert wird. Damit diese synthetischen Informationen an den richtigen Stellen eingeblendet werden können, muss aus der Sicht der realen Welt bestimmt werden, wo sich die Kamera befindet(Position) und wo sie hin blickt(Orientierung), die sogenannte Kamerapose. Im Bereich der AR gibt es zwei Herangehensweisen, die in markerloses und markerbasiertes Tracking unterteilt werden, jedoch meist beide nur in speziell vorgesehenen Bereichen funktionieren. Außerdem werden diese häufig durch 3D-Modelle der Umgebung und der Objekte unterstützt. Beim zweiten Verfahren werden in die reale Welt Marker(Bit-Codes z.B. QR-Code) eingefügt, so dass deren Position und Größe bekannt ist und diese von der Kamera erkannt werden können und folglich die Kamerapose zurückgerechnet werden kann. Das Problem dieses Verfahrens sind die Marker, welche vorher in der Welt angebracht werden und in dieser auch gut sichtbar sein müssen. Somit wird das markerlose Trackingverfahren immer wichtiger und sollte, soweit möglich das markerbasierte ersetzen. Da dieses Trackingverfahren für die Arbeit von Bedeutung ist, werden dessen Grundlagen in gleichnamigen Kapitel detailliert erklärt.

1.1 Aufbau der Arbeit

Im Laufe der Arbeit sollen zuerst im Kapitel „Einleitung“ verschiedene Trackingverfahren vorgestellt werden um das verwendete in einen Kontext einzuordnen. Des Weiteren werden die verschiedenen verwendeten Grundtechniken mithilfe ihrer Algorithmen und Begriffe im Detail im Kapitel „Grundlagen“ erklärt.

Im Hauptkapitel „Implementierung“ wird zunächst auf die gegebene Hardware und verwendete Software, sowie Bibliotheken eingegangen. Daraufhin wird der Ansatz der Konzeptphase erläutert und es wird auf Einzelheiten der Implementierung eingegangen. Im nächsten Kapitel wird das Ergebnis der Implementierung präsentiert und mithilfe verschiedener Videos und Einstellungen auf Performance getestet und ausgewertet.

Ein Ausblick bildet den Abschluss der Arbeit. In diesem werden Stärken und Schwächen der Implementierung diskutiert, sowie Verbesserungsvorschläge erwähnt und in einem Fazit zusammengefasst.

1.2 Stand der Technik

In diesem Kapitel wird die Arbeit in den Kontext schon existierender Arbeiten eingeordnet und auf mit in ihrer verwendeten, verwandten Techniken eingegangen. Zunächst wird durch die verschiedenen Grundtechniken des Trackings, das in dieser Arbeit verwendete genauer eingeordnet und in diesem Bereich auf die bereits existierenden verschiedenen und auch relevanten Algorithmen eingegangen.

1.3 Markerbasiertes Tracking

Das Markerbasierte Trackingverfahren ist die einfachste und am schnellsten umsetzbare Form des Trackings. Wie bereits in der Einleitung angeschnitten, müssen diese Marker(siehe Abb., Bit-Codes) in der Welt angebracht werden und deren Position, also die genaue Position der vier Eckpunkte, in der Welt bekannt sein, damit relativ zu diesen z.B. ein Würfel an der richtigen Stelle in der Welt gerendert werden kann. Soll z.B. dieser Würfel direkt auf dem Marker gerendert werden, so ist dies natürlich nicht nötig.

Durch einfache Kantendetektionsalgorithmen, können diese durch ihre einmaligen nicht in der Natur vorkommenden Muster sehr gut getrackt und voneinander unterschieden werden. Dieser Vorteil, was das Verfahren sehr robust und zuverlässig macht, ist zugleich auch der große Nachteil dieses Verfahrens, da die Marker in dem angebrachten Raum stark auffallend sind, was je nach Szenario unerwünscht oder auch nicht möglich ist. Alternativ und immer populärer wird deshalb das markerlose Tracking eingesetzt.

BILD eines MARKERS (z.B. artoolkit)

1.4 Markerloses/Modellbasiertes Tracking

Wie der Name des Verfahrens schon zu erkennen gibt, wird beim Markerlosen Tracking kein Marker verwendet. Was zur Folge hat, dass über andere Wege die Kamerapose bestimmt werden muss um an der richtigen Stelle rendern zu können. Im markerlosen Bereich werden dazu häufig verschiedene Formen von Sensorik verwendet. Seien es Positionssensoren, wie GPS oder in diesem Sinne auch WLAN-Stärke, Bewegungssensoren, wie ein Gyroskop oder ein Kompass, oder andere Sensoren, wie Ultraschall, so dass der Abstand zu einem Hindernis ermittelt werden kann.

Das modellbasierte Verfahren setzt voraus, dass es von dem zu Trackendem Objekt oder einer Umgebung ein exaktes Modell gibt, so dass es in irgendeiner Form zu einem Abgleich dieses Modells und des Kamerabilds kommen kann. Das genauer verwendete Verfahren wird im Kapitel der „Grundlagen „ vorgestellt.

1.5 Kantenbasierte Bildvergleiche?

Canny, Sobel, Hough Vergleichsarten -> recherche, Sebastian Kowalzy Houghtransformation war nix

Definition des Ähnlichkeitsmaßes(Schwellwertes),

2 Grundlagen

In diesem Kapitel werden die verschiedenen Grundlagen, welche für das Verständnis dieser Arbeit notwendig sind, genau erklärt. Dabei sollen sowohl die verwendeten Algorithmen als auch die Begriffe des Trackings und der Bildverarbeitung vorgestellt werden.

2.1 Analyse durch Synthese

Diese Methode setzt voraus, dass ein exaktes 3D-Modell der Umgebung vorhanden ist. Des Weiteren ist eine initiale Pose nötig, so dass der Algorithmus zu Beginn eine Kamerapose im Weltkoordinatensystem hat, von welcher aus er starten kann. Auf dessen Basis wird das aktuelle Kamerabild errechnet, indem die Pose des vorherigen bekannt ist und in deren nächster Umgebung(z.B. mithilfe des Partikelfilters) neue synthetische Bilder gerendert werden. Diese werden mit dem aktuellen Bild entweder merkmals- oder ähnlichkeitsbasiert abgeglichen, so dass das synthetische Bild mit der höchsten Ähnlichkeit die neue Kamerapose ist.

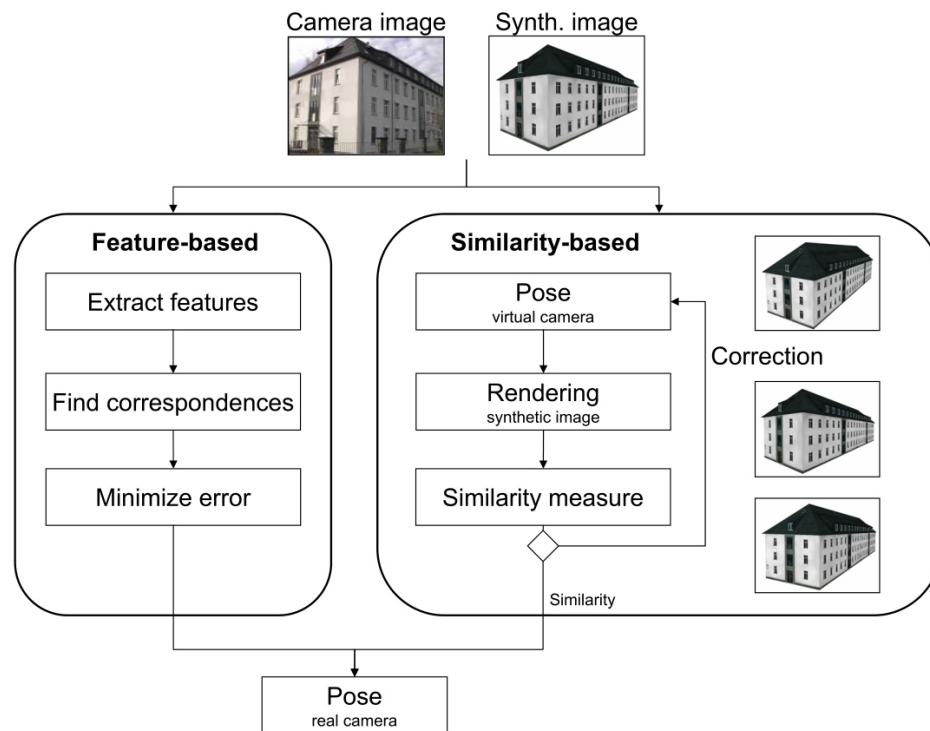


Abb.1: ADS Schema

2.2 Partikelfilter

Ein Partikelfilter ist allgemein ein Algorithmus mithilfe dessen sich der Zustand eines sich in Bewegung befindenden Systems schätzen lässt. In diesem Fall bedeutet das, dass sich durch diesen Algorithmus die Kamerapose schätzen lässt. Es funktioniert in der Weise, dass um die letzte bekannte Kameraposition eine Partikelwolke mithilfe der Normalverteilung aufgespannt wird.

2.3 Sobelfilter

3 Implementierung

3.1 Software/Hardware

Hardware:

- Laptop mit Windows 8 und Intel Core i5-3210M 2,5GHz CPU, 4GB Memory, GeForce GT 635M

Software:

- C++
Die Wahl von C++ ist mit der IDE einhergegangen.
- Qt Creator 3.1.2 Based on Qt 5.3.1(MSVC 2012, 32 bit)
Qt bringt die Standardfunktionen einer IDE mit und auch eine recht einfache und schnelle Funktion GUIs zu erstellen.
- Blender 2.68a
Mit Blender wurde das Model erstellt.
- FFmpeg
Ein Konsolentool um Videos zu Konvertieren, zu schneiden und in der Auflösung anzupassen.

3.2 Bibliotheken

- OpenCV
- Glew
- GLM
- Particle++ sollte benutzt werden, leider nicht möglich in dieser Kombination, da bestehendes projekt mit der opencv version nicht c++ 11 kann

3.3 Anforderungen

3.4 Umsetzungsansatz

3.5 Details der Implementierung

4 Ergebnisse

4.1 GUI? ist ja nicht von mir

4.2 Performance

5 Ausblick und Fazit

5.1 Verbesserungen

5.2 Fazit

6 Anhang

7 Fragen

-Grundlagen nach und nach ergänzen? Ja

-Der oder das Filter? Das Filter

-Nicht zu viel vorstellen(andere Methoden, besonders markerbasiert eher nicht), da nur BA-Arbeit? Verfahren wie Partikelfilter nicht so ausführlich wie MNohn? -Markerbasiert nicht, kein SLAM, Partikelfilter recht ausführlich

-Abb. aus Primärliteratur referenzieren?! Ja

-Partikelfilter in Aufbau nach ADS? Ja. Benutzt der Partikelfilter nur die letzte Pose oder auch die Bewegungstendenz der vorherigen? Beides, kommt auf die Implementierung an

-Was gehört zu ADS, auch Gauß oder Partikelfilter? ADS ist rendern und vergleich

-ADS sowohl merkmals- als auch ähnlichkeitsbasiert vorstellen? Ja. Umsetzung aber ähnlichkeitsbasiert(kanten) oder freie Wahl? Ja.

-Welches Betriebssystem? Windows

-Wann bist du in Singapur? August-Oktober

-Edge based Tracking blob anschauen

7.1 1.1

-lineare verfahren -> Merkmalskorrespondenzen

-nichtlineare verfahren -> Ähnlichkeitsbasiert

- Analyse auf Kantenbildern mithilfe der Teabox -> Haus/Stadt

- Schwarz-Weiß Eingabebild?

- Sobeloperator zur Kantendetektion -> sieht gut aus
- Cannyoperator nur ein Pixel breite Kante -> Dilatation verstärken?
- Hough-Transformation braucht parametrisierbare Form
- Webcam vom Laptop oder externe? Video -> Auflösung 800x600
- Qt und OpenCV? Ja

7.2 1.2

-erwähnen, dass die GUI, welche Jochen implementiert hat benutzt wurde...wie? wo? -die wahl der software war somit ja nicht frei?! erwähnen?
 -Particle++ sollte benutzt werden, leider nicht möglich in dieser Kombination, da bestehendes projekt mit der opencv version nicht c++ 11 kann....
 -rel. vergleich der synthetischen bilder, da sonst die box wegdriftet -ist der partikelfilter nur steuerung und auswahl des besten partikels? -im kapitel Implementierung die unterkapitel anforderungen und dann umsetzungsansatz? -wo genau den aufbau der arbeit, den entwurf einbringen? beim umsetzungsansatz?