Einleitung

Die Fähigkeit zur Pose Estimation in Gebäuden verschafft im Bauwesen eine Reihe von Anwendugnen wie z.B. automatsiche Baufortschritterfassung oder Facility-Management und Navigation über Augmented Reality.

Es gibt Lokalisierungsansätze in Gebäuden mit unterschiedlichen Technologien wie Lidar Wireless Access Points Bluethooth Beacons etc. worin ein visuelle Lokalisierungsverfahren über eine mobile Kamera die kostengünstigste, da zum einen heute jeder einen Smartphone bei sich trägt und zum anderen eine flächendeckende Hardware installation erspart bleibt.

Visuelle Lokalisierungsänstze wie visuelle Odometrie oder Simultaneos-Localization-and-Mapping sind relativ zum Ausgangspunkt. So eine Lösung für das Kidnapped-Robot-Problem bzw. die absolute Bestimmung der Pose über einen Anfragebild ist möglich durch das Suchen eines korrespondierenden Bildes in einer Bildergalerie oder das regressieren der Pose über Bild-Features. Allerdings benötigen diese Verfahren gelabelte Daten, welche wiederum auch zeit und kostenlastig sind.

Dafür versuchen Ansätze über künstliche neuronale Netzwerke wie z.B. PoseNet eine Lösung zu finden, worin das Ermitteln der Ground-Truth-Daten über SfM-Methoden genügt, sprich man nimmt nur einen Video der Zone auf, worin man es quasi überquert. Allerdings sind SfM-Methoden sehr fehleranfällig, sprich dreht man sich zu schnell um die Ecke wird die Verbindung zum vorherigen Frame verloren etc. und können sehr sehr viel Zeit in anspruch nehmen.

Grundlagen

Daher versuchen Acharya et al. Statt über SfM-Methoden über die Simulation eines Gebäuden Daten zu gewinnen

## CNN Architektur

Adad

Methodik

In dieser Abbildung ist noch einmal der Ansatz als Pipeline dargestellt. PoseNet wird mit den Gradientenbilder der synth. Daten trainiert und anschließend mit den Gradientenbilder der realen Daten evaluiert. Vorest wird PoseNet mit den Gewichten eines Models initalisiert, welches auf der GoogLeNet Architektur mit den Places Datensatz trainiert wurde. Die Position und die Orientierung als Quaternion ist das Output des Netzwerks.

Das Kapitel Methodik ist folgend aufgebaut. Zuerst wird die Erhebung der realen Daten die Generierung der synth. Daten beschrieben. Danach wird die Verarbeitung der Daten dargestellt. Im Anschluss werden die erstellten Datensätze und die Trainingsparameter angegeben.

## Erhebung der realen Daten

In der Literatur wurden beliebige Kameras für die Aufnahme der realen Daten verwendet und anschließend SfM-Methoden eingesetzt, um die Ground-Truth Daten zu ermitteln. SfM-Methoden können echt nervig werden, weil diese sehr fehleranfällig sind. Wir haben zwei Kameras der Intel Realsense Reihe verwendet. Einmal die Tracking Kamera 265 und die Depth-Kamera 435. Die T265 versichtere bei gegebenen Bestkonditionen die Pose mit einem Drift von 1% zu ermitteln. Wir konnten die Bestkonditonen nicht gewährleisten, s.d. eine größere Abweichung enstanden ist. Dazu später mehr. Die t265 wurde über die d435 montiert, s.d. wir über ROS die Kameras zeitgleich ansprechen und den Datenfluss synchronisieren könnten. Letzendlich hatten wir von der T265 die relative Pose zum Ausgangspunkt und stereo Fischaugenbilder. Von der D435 hatten wir einen RGB-Bild, Tiefenbild und eine Punkwolke der Szene. Für uns waren nur die Pose der T265 und die RGB-Bilder der D435 wichtig.

## Generierung der synth. Daten

Für die Generierung der synth. Daten wurden zuerst die Gebäudemodelle aus den BIMs der Gebäuden extrahiert und in Blender 2.79b simuliert. Die Aufnahmestrecken der realen Daten wurden bestmöglich imitiert und als Nurbs-Pfade rekonstruiert. Auf den Nurbs-Pfaden wurden inn 5cm intervallen mit einer +- Neigung von 10° in der y- und z- Achse Bilder gerendert. Die intrinsichen Daten der Kameras wurden an die virtuellen Kameras übergeben. Die Auflösung der synth. Bider wurden aus Performancegründen halbiert, da diese viel skalierter ins Netzwerk eingegeben werden, dazu aber später mehr. Insgesamt wurden 3 Typen pro Strecke erzeugt. Hier ist auch ein Ausschnitt des Plugings, wo man auch händisch die Konotrollpunkte des NURBS anpassen konnte etc.

## Verarbeitung der Daten

Ziel war es mit Gradientenbildern das Netzwerk zu trainieren und zu evaluieren. Also wurden die realen sowie synth. Bilder in ihren Gradietenbilder umgewandelt. Die Auflösung realen Bilder wurden vorher auf die Größe der synth. Daten halbiet. Das führte bei den Gradientenbilder zu schärferen Kanten, vorallem bei durch Bewegung verschwommenen Bildern.

Durch die künstlichen Lichter der synth. Bilder entsandten Artefakte, vorallem bei dem cartoon-Datensatz, welche mit einem Treshold-Verfahren unterdruckt wurden, Konsistenzhalber wurde dieser Tresholdverfahren bei allen Datentypen verwendet

## Datensätze

Ziel war es den Ansatz von Acharya auf längeren und in mehreen Richtungen verlaufende sowie auf mehrere Etagenebenen erstreckende Strecken zu untersuchen. Es wurden Daten aus der nördlichen Hälfte des 6. Stockwerks des IC-Gebäudes und der Seminargebäude der Hochschule Bochum erhoben.

Die Simulationen der Gebäuden unterschieden sich im Simulationsdetail. Während die Simulation der IC-Gebäude minimalistisch war, enthielt die Simulation von der Hochschule Objekte der techschnichen Gebäudeausrüstung wie z.B. Feuermelder etc. weiterhin bestand die IC-Simulation aus sich wiederholenden Strukturen, Hingegen hatte die HS-Simulationen einen eindeutigen Treppenhaus, weshalb ein Datensatz aus dem IC erhoben wurde und 3 aus der Hochschule.

Der aus dem IC-Gebäude erhobene Datensatz wurde IC-loop bennant und bildet eine geschlossene Scheife. Die HS-Gamma Strecke ausgehend aus einem Zimmer verläuft in eine Schlaufe und übergeht zu einem optisch ähnlichen Flur. Die Strecken HS-stairs-up und down erstrecken sich ab –und hochwärts auf einer Treppe naheligend der Schlaufe im HS-gamma.

Grün ist die Strecke die in den Simulationen rekonstruiert wurde. Die Blaue Strecke wurde von der T265 ermittelt und die schwarzen Umrisse der Kameras geben die Aufnahmerichtung an.

## Trainingsparameter

Nach dem die Datensätze vorhanden gewesen sind, musste noch das PoseNet Netzwerk trainiert werden. So ein künstliches neuronales Netzwerk besteht aus zwei Parameterarten. Es gibt zumal die Parameter, eher bekannt als Gewichte, die im Trainingsprozess mit den Trainingsdaten optimiert werden und solche, auch Hyperparameter genannt, die das Netzwerk konfiguieren und z.B. das Lernverhalten angeben. Damit die Untersuchung von den Daten abhänig bleiben, wurden die Hyperparameter übernommen bzw. gleichermaßen bestimmt oder im selben Verhältnis zum Datensatz gewählt.

Der einzige Hyperparameter der durch ein Grid-Search Verfahren bestimmt wurde, ist der Hyperparameter beta von der PoseNet Kostenfunktion. Dabei wurde mit den realen Daten trainiert und evaluiert. Die dabei ermittelte Akkuratesse der Netzwerke sollte zudem auch als Referenzwerte dienen.

Hier sieht man einige der Hyperparemeter. Es macht gerade aus Zeitgründen keinen Sinn auf jeden einzelnen Hyperparameter einzugehen.

Ergebnisse

Wie bereits erwähnt besteht das Output des PoseNet aus zwei Komponenten. Deshalb gibt in dieser Arbeit ein Evaluationsergebnis die Abweichung der Position in Meter und den Orientierungsfehler in Grad an. Diese werden anhand der Positionsfehler verglichen. Die Akkuratesse der Netzwerke ist der Median der Evaluationsergebnisse. Außerdem wurden die zuvor mit den synth. Daten trainierten Netzwerke zweimal evaluiert. Die erste Evaluation folgte mit den korrespondierenden synth. Evaluationsdaten. Die zweite Evaluation folgte mit den realen Evaluationsdaten. Ferner sollten die bei der Bestimmung des Beta Hyperparameters ermitteln Akkuratessen als referenzwerte dienen.

## IC-loop

Auf der IC-loop Strecke konnte eine durchschnittliche Akkuratesse von 1.80m, 8.05° mit den synth. Daten erzielt werden. Hingegen konnte mit realen Daten 16.68, 73.25 erreicht werden. Betrachtet man nun das beste Netzwerk mit realen Daten, stellt man fest, dass dieses Netzwerk alle Evaluationsdaten in einem ca. 5x30m großen Teilbereich bestimmt hat. Ebenso wurde die Orientierung als die Aufnahme Richtung der unteren horizontalen Strecke bestimmt.

## HS-gamma

Auf der HS-gamma Strecke konnte eine durchschnittliche Akkuratesse von 0.85m, 8.07° mit synth. Daten erreicht werden. Mit den realen Daten konnte durchschnittlich 9.67m, 32.10° erzielt werden. Bei genauem Hinsehen des Netzwerks mit den besten Ergebnissen bei realen Daten stellt man fest, dass das Netzwerk die Evaluationsdaten in einem ca. 5x20m großen Teilbereich lokalisiert hat, ebenso wurde die Orientierung als die Aufnahmerichtung der horizontalen Strecken bestimmt.

## HS-stairs-up

Auf der HS-stairs-up Strecke konnte mit den synth. Daten eine durchschnittliche Akkuratesse von 0.85, 8.07° erreicht werden. Mit den realen Daten hingegen nur 4.75m, 56.15° und beim genauen Betrachten der Evaluation des grad-edge Netzwerks, sieht man das die Evaluationsdaten zwischen dem oberen und unteren Treppenlauf lokalisiert wurden. Es gibt abwechselnd immer größere Positionsfehler, die größten sind auf dem Treppenpodest zu erkennen. Ebenso konnte man auch erkennen, dass die Orientierung hin und wiedermal in der entgegensetzten, sprich 180° falsch bestimmt wurden.

## HS-stairs-down

Die Ergebnisse von HS-stairs-down sind den Ergebnissen von HS-stairs-up ähnlich. Es konnte mit synth. Daten eine Akkuratesse von 0.93m, 8.03° erreicht werden, und mit den realen Daten 5.01m, 49.29°. Hierbei wurden die Evaluationsdaten ebenso zwischen dem oberen und unteren Treppenlauf lokalisiert. Abwechselnde Positionsfehler sind hierbei sichtbarer, vor allem die der entgegengesetzten Orientierungen.

## Zusammenfassung

Nochmal zusammengefasst war auf den hier erhobenen Datensätzen eine Akkuratesse von 1.17m, 7.34° als Referenzwert gesetzt. Diese wurden bei der Bestimmung des Hyperparameters ermittelt. Weiterhin konnte bei der Evaluation der zuvor mit synth. Daten trainierten Netzwerken eine Akkuratesse von 1.19m, 8.35° erreicht werden, welche ziemlich nah an den Referenzwerten sind. Bei der domänenübergreifenden Evaluation, sprich die Evaluation mit den realen Daten, konnte eine durchschnittliche Akkuratesse von 10.95, 49,69° erzielt werden.

Diskussion

Nach dem die Ergebnisse sachlich vorgestellt wurden, möchte ich diesen nun eine Bedeutung verleihen. Die Diskussion ist in 3 Abschnitte geteilt. Zuerst möchte ich die Defizite der angewandten Methodik auflisten, s.d. man diese im Hinterkopf behält. Danach werden die Ergebnisse interpretiert und mit den Ergebnissen der Literatur verglichen. Im Anschluss möchte ich noch Empfehlungen für weiterführende Forschungen geben.

## D. der angewandten Methodik

Die in dieser Arbeit verwendetet Intel Realsense T265 versprach bei Bestkonditionen eine Drift von 1%. Wir konnten die Bestkonditionen nicht erfüllen s.d. eine Abweichung bis zu 5% entstanden ist. Trotzdem konnte eine Korrespondenz der Ground-Truth-Daten zwischen der Erhobenen und von der Simulation verschafft werden. Allerdings ist ein negativer Einfluss auf die Akkuratesse der domänenübergreifenden Evaluationen nicht auszuschließen bzw. denkbar.

Ein Weiterer Punkt ist, dass die Akkuratesse eines KNNS in dieser Arbeit durch das stochastische Gradientenabstiegsverfahren im Trainingsprozess vom Zufall abhängt. Die bestmögliche Akkuratesse zu finden würde den Rahmen einer Bachelorarbeit sprengen. Deshalb wurden je Datentyp die Trainingsprozesse 5-mal wiederholt. Daher könnte bei weiteren Trainingsprozesse bessere Ergebnisse erzielt werden.

Zudem wurden die Hyperparameter von Acharya et al. übernommen und könnten auf deren Datensatz optimiert sein bzw. mit deren Datensatz besser harmonieren. Infolgedessen könnten bei einer anderen Belegung der Hyperparameter ebenso bessere Ergebnisse erzielt werden.

## D. der Ergebnisse

In dieser Arbeit konnte auf die erhobenen Datensätze bei einer domänenübergreifenden Evaluation eine durchschnittliche Akkuratesse von 10.95m, 49.69° erzielt werden. Ferner sollten die bei der Hyperparameterbestimmung erhaltenen Akkuratesse von 1.17m, 7.34° als Referenzwerte dienen. Dieses konnte die durch die Evaluationen mit Daten der gleichen

Domäne mit 1.19m, 8.35° angenähert werden. Allerdings wurde ein domänenübergreifender Ansatz gestrebt und diese ist bei den betroffenen Gebäuden mit einer Positionsakkuratesse von ca. 11m nicht denkbar.

Bei den HS-stairs-down und up Strecken konnte keine Generalisierungsfähigkeit der Netzwerke erkannt werden, vielmehr konnte eine zufällige Zuordnung interpretiert werden. Naja, so eine Treppe weist eben von Natur aus wiederholenden Strukturen.

Deshalb könnten die Unfähigkeiten der Netzwerke mit perceptual-aliasing begründet werden. Perceptual-aliasing ist eine der größten Herausforderungen visueller Lokalisierungsverfahren und wird in diesem Kontext als das optisch gleiche Erscheinen zweier verschiedenen Stellen eines Gebäudes definiert.

Weiterhin wurden die Evaluationsdaten der Strecken IC-loop und HS-gamma in einem

ca. 5m breiten und 20m bis 30m langen Teilbereich lokalisiert. Ebenso konnte man auch erkennen, dass die Netzwerke nur eine Richtung erlernt haben. Während das Lokalisieren in einem Teilbereich bei der IC-Loop Strecke auf perceptual-aliasing geführt werden könnte, da sowohl die vertikalen als auch die horizontalen Strecken optisch gleich aussehen, unterscheiden die Evaluationsdaten der vertikalen und entlang der Schlaufe Strecke sich von den horizontalen Strecken. Nichtdestotrotz wurden diese nur in einem Teilbereich der linken horizontalen Strecke lokalisiert. Interessanterweise stellen diese Ergebnisse Parallelen zudem Datensatz bzw. Ergebnissen von Acharya et al. her.

Die schlechten Ergebnisse auf den HS Datensätzen könnten auf das hohe Level-of-Detail der Simulationen zurückgeführt werden, da Achaya et al. überraschenderweise feststellten, dass die Zunahme des Level-of-Details zu einer Abnahme der Akkuratesse führte. Widersprüchlicher weise empfahlen die Autoren für eine besseres Akkuratesse ein hohes Level-of-Detail der Simulationen. Naja, die Ergebnisse der HS-gamma Strecke waren weit über einem vorstellbaren Einfluss des hohen Level-of-Detail. Es sollte auch nicht außer

Acht gelassen werden, dass die Ergebnisse von HS-Gamma Gemeinsamkeiten zu den Ergebnissen von IC-loop und den Ergebnissen von Acharya et al. aufweisen.

Was so eigentlich schon fraglich erscheint. PoseNet ist grundsätzlich nicht begrenzt auf eine Teilzone von ca. 5m breite und 30m lange. In dieser Arbeit konnte mit Daten der gleichen

Domäne eine Positionsakkuratesse von ca. 1m erreicht werden. Ferner konnten Wlach et al. auf einen größeren Datensatz PoseNet erfolgreich anwenden.

Daher liegt die Schlussfolgerung nahe, dass PoseNet bei domänenübergreifender Anwendung nur in einem kleinen Bereich in einer Richtung trainiert werden kann.

## Empfehlung für weiterführende Forschungen

Acharya et al. erzielten mit den edge-Datensätzen die besten Ergebnisse. In dieser Arbeit konnte mit den Gradientenbildern der cartoon-Datensätzen häufiger bessere Ergebnisse erzielt werden. Dennoch kann auf diesen Tatsachen kein bestes synth. Datentyp festgelegt werden, da hierzu die bestmögliche Akkuratesse nicht gewährleistet werden können, weil diese wie bereits erwähnt vom Zufall abhängig sind. Man könnte daher bei gleichen Hyperparametern die Anzahl der Trainigsprozesse erhöhen, um bessere Ergebnisse zu erzielen oder auszuschließen. Infolgedessen könnte der Zusammenhang der Datentypen und Ergebnissen untersucht werden.

Wie vorhin erwähnt wurden die Hyperparameter übernommen oder im gleichen Verhältnis bestimmt. Eine Optimierung der Hyperparameter sollte zu besseren Ergebnissen führen und

ist daher ebenso eine Empfehlung für weiterführende Forschung

Es gibt Nachfolger von PoseNet die das perceptual-aliasing Problem behandeln oder auch die Ungewissheit der Ergebnisse mitliefern, sodass man zu der bestimmten Pose noch einen

Maß mitgeliefert bekommt, die angeben soll, inwieweit man diesen Ergebnissen vertrauen kann. Daher macht es Sinn die im Rahmen dieser Bachelorarbeit erhobenen Datensätze bei den Nachfolgern anzuwenden.

Fazit

So, um jetzt alles nochmal abzurunden und abzuschließen. Ziel dieser Bachelorarbeit war es den Ansatz von Acharya in größeren Gebäuden auf längeren Strecken zu untersuchen. In wurde dieser Ansatz mit 2 Gebäuden auf 4 Strecken untersucht.

Zusammenfassend konnte festgestellt werden, dass eine Lokalisierungsverfahren durch eine domänenübergreifende Anwendung von PoseNet auf die hier erhobenen Datensätze nicht möglich gewesen ist. Es konnte bei Daten der gleichen Domäne eine Akkuratesse von ca. 1m in der Position und 8° in der Orientierung erzielt werden. Bei der domänenübergreifenden Anwendung konnte eine Akkuratesse von 10.95m in der Position und 49.69° in der Orientierung erreicht werden. Hier konnte PoseNet nur in einem Teilbereich und in einer Richtung trainiert werden. Angesichts der Parallelen zu den Acharya et al. Datensätzen lag die Schlussfolgerung nahe, dass der Ansatz auf die erwähnten Eigenschaften begrenzt ist.

Ein Lokalisierungsverfahren mit einer Positionsakkuratesse von ca. 11m ist undenkbar. Sogar die potenziell mögliche Akkuratesse von 1m wäre im direkten Gebrauch für z.B. Anwendungen mit Augemented Reality nicht brauchbar, allerdings durch Kaskadeneffekt korrigierbar.

Jetzt kommt so ein gröberer Ausblick zu der domänenübergreifenden Anwendung.

Die Unzureichende Akkuratesse der domänenübergreifenden Evaluation liegt den Simulationsdefiziten wie das Fehlen von Objekten oder domänenspezifsiche Artefakte wie z.B. durch Licht oder Reflektieren von Gläsern etc. zurück. Daher ist es lohnenswert in diesem Zusammenhang zu untersuchen, ob die Diskrepanzminimierung zwischen der

synth. Daten und der realen Daten z.B. durch GANS zu besseren Ergebnissen führt. Ferner könnte man die möglichen Posen im Trainingsprozesses so einschränken, dass diese immer im begehbaren Flächen oder tatsächlich möglichen Bereichen liegen. Die dafür nötigen Informationen kann man ja durch die Simulation bzw. BIM der Gebäuden gewinnen.