

Prof. Dr.-Ing.habil.
Dirk Joachim Lehmann
Data Science in IoT
Fakultät für Informatik
di.lehmann@ostfalia.de
www.dirk-lehmann.de

3D Flight Scanner System - mit Quadrocoptern im Outdoor Bereich

Hintergrund

Eine häufig auftretende Aufgabe - beispielsweise im Rahmen von Vermessungsaufgaben, im industriellen Design, in der Kartografie (z.B. bzgl. Google Maps), in der Architektur und der Bauplanung, im Bereich von FX- und Spezialeffekten, im Marketing und im Medium Film/Serie - ist es 3D Modelle von realen Objekten zu generieren.

Mit 3D-Modellen bezeichnen wir eine virtuelle Repräsentation eines (oft realen) Objektes, wie beispielsweise einem Tisch, einem Haus oder eines Autos etc. Ein solches Model besteht (in der rudimentärsten Beschreibungsform) in vielen Fällen aus Dreiecksnetzen – also aus einer Menge von aneinander angrenzenden Dreiecken – um die Oberflächenform des Objekts im Rechner virtuell zu modellieren. In der Abbildung sehen sie einige Beispiele für Dreiecksnetze von 3D-Modellen:

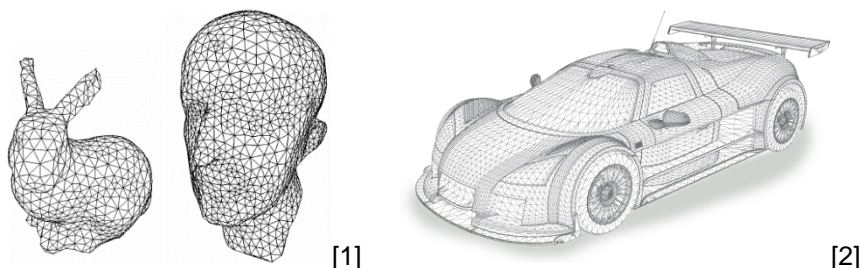


Abb.1: Beispielhafte Dreiecksnetze (sogenannte Wireframes) von virtuellen 3D-Modellen

Betrachten wir die Verbindungskanten der Dreiecke nicht, erhalten wir eine Menge von 3D Punkten – man spricht von einer 3D-Punktwolke (aller Dreieckseckpunkte) - als rudimentäre Repräsentation des virtuellen 3D-Modells eines (realen) Objektes, wie in folgender Abbildung dargestellt.

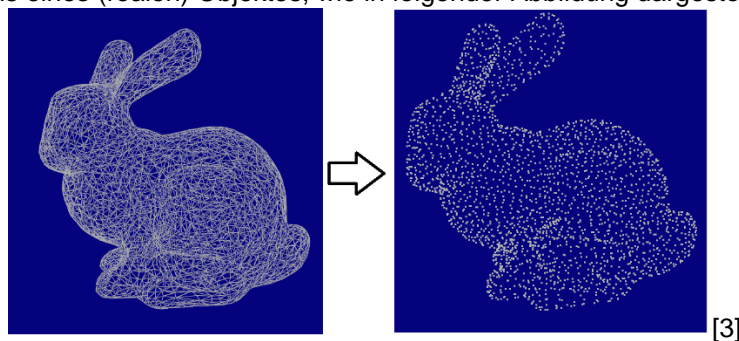


Abb. 2: Dreiecksnetz und 3D Punktwolke eines virtuellen 3D Modells

Um solche 3D-Modelle zu erstellen ist es möglich die Geometrien manuell zu modellieren, z.B. über Systeme aus dem Bereich *Computer Aided Design*. Dies ist ein zeitaufwendiger Prozess, daher wurden zusätzliche Systeme entwickelt, um 3D-Modelle bzw. 3D-Punktwolken (semi-) automatisch und

zeiteffizient zu generieren. Oftmals handelt es sich dabei um optische Scan-Verfahren, welche beispielsweise Laufzeiten von (reflektierten) Laserlicht erfassen (*Time of Flight*), um derart die virtuellen 3D-Punktwolken eines realen Objektes zu ermitteln.

Structure from Motion (SfM)

Eine alternative Möglichkeit aus dem Bereich der Computer Vision, mit dem Ziel 3D-Punktwolken zu erzeugen, ist es diese aus einer Serie von 2D Bildern selbst zu ermitteln. Ein diesbezüglicher Ansatz ist das *Structure from Motion*:

Die Kernidee ist, mithilfe von 2D Kameras, Bildaufnahmen des Objektes aus unterschiedlichen Positionen aufzunehmen und aus diesen die räumlichen 3D Koordinaten korrespondierender Bildpunkte zu errechnen (zu triangulieren) und auf diese Art die 3D-Punktwolke des 3D-Modells zu generieren. Die folgende Abbildung stellt diesen Prozess schematisch dar.

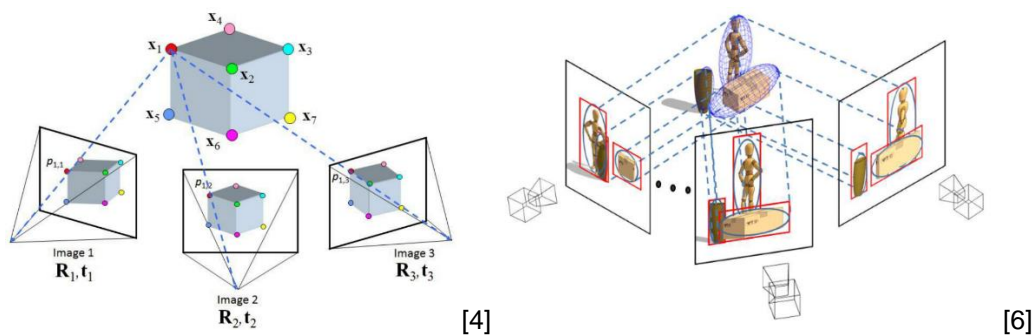


Abb 3. Triangulierung/Ermittlung von 3D Punktwolken korrespondierender Bildpunkte im *Structure from Motion*

Dadurch ist es möglich, eine 3D-Punktwolke aus einer Serie von Bildern zu generieren, wie in der folgenden Abbildung dargestellt:

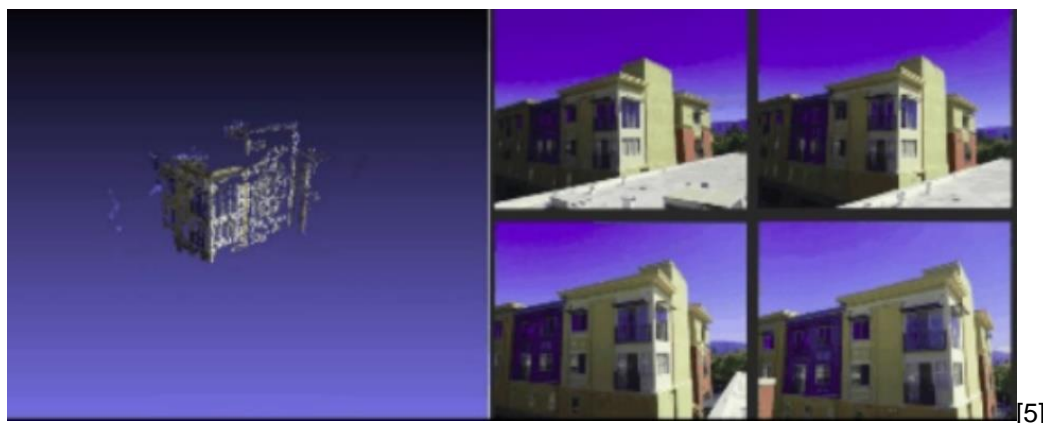


Abb 4. (links) Resultierende 3D Punktwolke (rechts) generiert aus der vorliegenden Bildserie im *Structure from Motion*.

Der Vorteil dieses Verfahrens ist zudem, dass kein komplexer Versuchsaufbau benötigt wird, um eine 3D-Punktwolke zu generieren. Eine mobile 2D Mono-Kamera und die Nutzung des Umgebungslichts genügt, um mithilfe eines handelsüblichen Arbeitsrechners die Triangulationen erfolgreich zu berechnen. Zur Ergänzung: Der Begriff *Structure from Motion* selbst ergibt sich daraus, dass eine 3D-Punktwolke ermittelt wird (= *Structure*) aus einer Bildserie unterschiedlicher Positionen, welche z.B. dadurch gewonnen werden kann, indem sich die Kamera um das Objekt herumbewegt (= *Motion*), während die Bildaufnahme läuft, sodass die benötigte Bildserie beispielsweise als separates Video vorliegt.

Einsatz von Quadrocopter zur Erzeugung von Bildserien

Im Projekt dienen Quadrocopter dazu, die notwendigen Bildserien zu generieren. Ein Quadrocopter ist ein kleines batteriebetriebenes und im Vergleich kostengünstiges Fluggerät, welches mit einer 2D Mono-Kamera bestückt ist und welches über Funksignale in seiner Bewegung gesteuert werden kann. Derart sind schwer erreichbare Orte für eine visuelle Inspektion zugänglich (Dach, Pylonen von Strommasten, verwinkelte Brückensegmente etc.). Zudem kann ein Quadrocopter im Außenbereich (=Outdoor) eingesetzt werden. Dadurch ist es möglich, die benötigten Bildserien auf pragmatischem Wege zu erzeugen. Die folgende Abbildung zeigt beispielhaft einige erwerbbaare Quadrocopter:



DJI Tello Mini Drohne [7]



DJI Mini 2 Fly Drohne [8]

Angebot

Im Rahmen ihres *Praxisprojektes*, *Masterseminars*, *Masterprojektes*, ihrer *Bachelorarbeit*, *Masterarbeit* oder ähnlichen Studienleistungen - wie z. B. einem *interdisziplinären Digitalisierungsprojekt* - können sie gerne diese Aufgabe bearbeiten

Melden Sie sich gerne bei mir unter: di.lehmann@ostfalia.de

Aufgaben/Arbeitspakete

Das Projekt geht von Aufgaben, wie der Ansteuerung der Drohne, über die Auswertung der Bilddaten und der Erzeugung der Punktwolken und Geometrien, hin zur Missionsplanung und dem User-Interface Design – überstreicht somit eine breite Themenvielfalt der Informatik – und besteht aus klar voneinander abgrenzbaren Arbeitspaketen (AP):

- **AP1) Structure from Motion-Komponente in Python**
 - o **Eingabe:** Bildseriendaten konkreter Bildformate, intrinsische und extrinsische Kameraparameter, Korrekturfaktoren radialer- und tangentialer Verzeichnungen
 - o **Ausgabe:** 3D Punktwolken in standardisierten Formaten (z.B. las, xyz, e57, rcp, etc), Farb-/Texturinformationen der assoziierten 3D Punkte

Beispiele für in diesem Paket umzusetzende Features:

 - In Abhängigkeit der Linsen- und Shutter-Eigenschaften der Kamera: korrigieren von radialen- und tangentialen Verzeichnungen
 - Finden korrespondierender Punkte in den Bildern der Bildserie (= Key Points/ Feature Points)
 - Triangulation von 3D Raumkoordinaten, basierend auf korrespondierenden 2D Punkten in der Bildserie
 - Berücksichtigung von z.B. Eigenschaften des optischen Flusses, um Fehler in den Key Points automatisiert zu korrigieren.
 - ...
- **AP2) Geometrierzeugung-Komponente in Python**
 - o **Eingabe:** 3D Punktwolken, Farb- und Texturinformationen
 - o **Ausgabe:** diskrete Geometrien wie Dreiecksnetze oder analytische Geometrien wie NURBS in persistent speicherbaren Datenformaten

Beispiele für in diesem Paket umzusetzende Features:

- Registrieren von Punktwolken
- Interaktives Hinzufügen/Entfernen von 3D Punkten
- Automatische Methoden um Rauschen aus den Punktwolken zu entfernen
- Automatische Methoden um Dreiecksnetze (oder polygonale Netze) zu berechnen
- Automatische Methoden um Geometrien in NURBs umzuwandeln (Non-uniform rational B-Splines)
- Interaktives reparieren und bereinigen (Cleaning) von korrupten Dreiecksnetzen
- Laden/Speichern von 3D Punktwolken
- ...

- AP3) Missionssteuerung-Komponente in Python

- [Eingabe](#): ID-Quadrocopter, Wegpunkte in 3D (z.B. in GPS-Koordinaten oder in Bezug auf ein konkretes Weltkoordinatensystem), Kameraden des Quadrocopter, ein zu virtualisierendes Objekt von Interesse in der realen Welt.
- [Ausgabe](#): Kommunikation, Kommunikationsüberwachung, und Steuerbefehle für Quadrocopter der ID zum kollisionsfreien Abfliegen der Wegpunkte und speichern der Bildserien aus der Kamera.

Beispiele für in diesem Paket umzusetzende Features:

- Kollisionserkennung und Kollisionshandling/-vermeidung
- Erkennung des Objekts von Interesse, damit sich die Drohne für optimale Bildaufnahmen korrekt ausrichten kann
- Autonome Exitstrategien bei Kommunikationsausfall und Reichweitenüberschreitung
- Autonome Exitstrategien bei schwachen Batterien (Missionsabbruch, automatische Rückkehr zu definiertem Heimatpunkt, übergehen in einen sicheren Systemzustand)
- Bilddatenübertragung und persistente Speicherung in einem zuvor definierten Datenablagensystem
- Höhenkontrolle/Geschwindigkeitskontrolle/Flugzustandsüberwachung
- Abbilden vereinfachter Flugbefehle wie FlyTo3DPoint(x,y,z) in Regelbefehle
- Übertragung Regelbefehle und Missionsdurchführung des Quadrocopter der ID
- Self-Awareness (das System kann eigenständig beurteilen, ob es einen sicheren Betriebszustand ermöglichen kann, oder nicht, z.B. wegen externer Faktoren: Wind, Regen, Nebel, Dunkelheit; oder interner Faktoren: beschädigte Rotoren und ähnliches. Falls nicht, wird Mission nicht ausgeführt/vom System verweigert)
- Erfassen/Ermitteln externer Faktoren zur Betriebssicherheit (Regen, Wind, Nebel, Helligkeit, ...)
- Erfassen/Ermitteln interner Faktoren zur Betriebssicherheit (Batteriestatus, Kommunikationsstatus, funktionsfähige Aktuatorik, etc.)
- ...

- AP4) Datenmanagement-Komponente in Python

- [Eingabe](#): Stream von Bilddaten aus Quadrocoptern, Meta-Daten, Avionik-Daten während Mission (cf. AP 3 Missionssteuerung-Komponente), 3D Punktwolken und Geometrien aus Modellberechnungen (cf. AP 2 Geometrieerzeugungs-Komponente).
- [Ausgabe](#): Persistente Datenstrukturen, Datenablage, Kataloge abgelegter Daten (Bildserien, durchgeführte Mission, visualisierte Objekte, Meta-Informationen etc.), Retrieval Möglichkeiten, Datensicherungsoptionen

Beispiele für in diesem Paket umzusetzende Features:

- Konzepte für einen skalierbaren Datensee
- Authentifizierungskonzepte für Zugriffssteuerung auf Daten
- Anonymisierungskonzepte für Bilddaten (einhalten geltender Rechtsnormen, wie DSGVO zu personenbezogenen Daten)
- Datenablage und diesbezügliche (invertierte) Kataloge der Bildserien
- Datenablage und diesbezügliche (invertierte) Kataloge der Avionik- und Metadaten
- Datenablage und diesbezügliche (invertierte) Kataloge für virtuell Modelle
- Session und Missions-Archiv
- Bildbasierte- (Content-based Image Retrieval) und Text-basierte Retrieval Möglichkeiten
- Datenkonsistenzprüfungen und Sicherung der Datenqualität (Erkennen und Umgang mit korrupten Daten, Missed Values, etc.)
- Eine Reset-Möglichkeit des Daten-Management-Systems um ihn in einen früheren konsistenten Datenzustand zu überführen (z.B. bei zunehmend auftretenden Inkonsistenzen)
- ...

- AP 5) Quadrocopter-Flottenmanagement-Komponente in Python

- [Eingabe](#): Eine Anzahl unterschiedlicher Quadrocopter, auf welche im Rahmen der Missionsplanung zurückgegriffen werden können, um Missionen koordiniert und synchronisiert effizient durchzuführen
- [Ausgabe](#): Für die Missionsplanung im System registrierte bzw. angemeldete Quadrocopter

Beispiele für in diesem Paket umzusetzende Features:

- Manuelles Anmelden und Abmelden von Quadrocopter im 3D Flight Scanner System
- Übersicht der Betriebszustände aller verfügbaren (angemeldeten, registrierten) Quadrocopter
- Automatisches Abmelden von Quadrocopter aus Flotte, wenn nicht mehr betriebsicher.
- ...

- AP 6) Automatische Missionsplanungs-Komponente in Python

- **Eingabe:** Objekt von Interesse der realen Welt (z.B. als segmentierte Pixelmaske des Wunschobjektes über ein PAD u.Ä.)
- **Ausgabe:** Abzufliegende Wegpunkte und Aufteilung der Wegpunkte über verfügbare Quadrocopter-Flotte (cf. AP 5) für eine optimale optische Objektvermessung (optimal= wenigste Anzahl an Wegpunkte bzw. Gesamtwegstrecke bei maximaler Abdeckungsrate des zu vermessenden Objektes), Schätzung/Prädiktion angenommener Abdeckungsrate des Objektes bzgl. dieser Wegpunkte

Beispiele für in diesem Paket umzusetzende Features:

- Manuelles Anmelden und Abmelden von Quadrocopter im 3D Flight Scanner System
- Übersicht der Betriebszustände aller verfügbaren (angemeldeten, registrierten) Quadrocopter
- Mission für Objekt von Interesse planen
- Geplante Mission für Objekt von Interesse mit Quadrocopter-Flotte ausführen (und dabei überwachen)
- Mission bei Problemen beenden/abbrechen und flottenweit einen sicheren Systemzustand herbeiführen
- ...

- AP 7) User-Interface

- **Eingabe:** Schnittstellen zu allen APs
- **Ausgabe:** interaktive User-Interface Systeme um die Parametrisierungen, Monitoring aufgaben, und Missionen durchführen zu können. (jedes AP erhält einen eigenen Tap zur besseren Trennung der UI)

Beispiele für in diesem Paket umzusetzende Features:

- User Interface für AP1 *Structure from Motion*-Komponente
 - Bildserien Viewer: Bildserien anzeigen, durch-scrollen, navigieren, etc.
 - UI Elemente, um korrespondierende Punkte in unterschiedlichen Bildern des gleichen Objektes zu berechnen und anzuzeigen (Feature Points)
 - Interaktives korrigieren von fehlerberechneten korrespondierenden Punkten
 - Interaktives Löschen/Hinzufügen von korrespondierenden Punkten in den Bildserien
 - Anzeigen der resultierenden 3D-Punktvolke und interaktives Navigieren in dieser
 - Interaktive Eingabe von Kameraparametern (intrinsisch, extrinsisch, Korrekturparameter)
 - ...
- User Interface für AP2 Geometrieerzeugungs-Komponente
 - 3D-Punktvolken Viewer
 - UI Elemente hinzufügen für interaktive Steuerung von Operationen wie entrauschen der 3D-Punktvolken, Geometrien berechnen, etc.
 - 3D-Geometrie Viewer
 - UI Elemente um 3D Geometrien zu manipulieren
 - ...
- User Interface für AP3 Missionssteuerung-Komponente
 - Quadrocopter Kamera Viewer (aktuelle Sicht der Kamera eines Quadrocopters im Einsatz darstellen)
 - Abzufliegende Wegpunkte des Quadrocopter darstellen
 - Statusanzeige (Einsatzzeiten, Batterie, etc.)
 - ...
- User Interface für AP4 Datenmanagement-Komponente
 - Dateien Viewer/Explorer (Übersicht aller Daten)
 - UI Elemente für parametrisierte Suche nach Dateien (Datum, Volltextsuche, Bildbasierte Suche, etc.)
 - UI Elemente zur Darstellung von Statistiken von Dateien (Nutzungsdauer, Frequenz, etc.)
 - UI für Anmeldung und Authentifizierungen
 - ...
- User Interface für AP5 Quadrocopter-Flottenmanagement
 - Hinweis: UI AP5 ist assoziiert mit UI von AP3 / Faktisch ist UI AP5 eine Übersichts-UI über Eigenschaften der Flotte und UI AP3 eine Detailsicht auf einen einzelnen Quadrocopter

- Quadrocopter-Flotten Viewer
- Informationen zu Betriebssicherheit der Flotte und Umgebung (extern, intern)
- User Interface für AP6 Automatische Missionsplanungs-Komponente
 - UI Elemente für die Missionsplanung und Missionsausführung
 - ...

Zu den Arbeitspaketen hinzu kommen notwendige Recherchetätigkeiten, Make-Or-Buy-Entscheidungen, Beachtung von Lizenzfragen, Aspekte der Continuous Integration und des Code-Managements, Dokumentationsaufgaben, Fragen zum Aufsetzen/Deployment und der Migration von Entwicklersystemen (und zum Projektmanagement), wie sie in der Softwareentwicklung üblich sind.

Im Rahmen einer Projektbearbeitung wird nicht erwartet, dass unmittelbar alle APs bearbeitet werden können. Je nach Umfang Ihrer zu erbringenden Studierendenleistung können Teilaspekte einzelner APs bearbeitet werden. Den konkreten, jeweiligen Umfang stimmen wir im Vorfeld gerne gemeinsam ab.

Kommentar zum Projekt:

Projekte für Befliegungen mit Drohnen finden Sie mittlerweile in unterschiedlichen Unternehmen der Industrie und an unterschiedlichen Hochschulen und Universitäten. Der USP dieses Projektes ist die Kombinationen aus Befliegung und Flottenmanagement und der spätere weitere Ausbau der Funktionalitäten (z.B. autonome Waldbranderkennung u.Ä.) innerhalb eines System-Frameworks.

Referenzen

[1] Ivrisimtzis, Ioannis and Jeong, Won-Ki and Seidel, H.-P, 2003, Using Growing Cell Structures for Surface Reconstruction

[2] <https://www.affluent-tech.com/engineering/automotive/crash-and-safety/sports-car-wireframe/>
Stand: 7.7.2022

[3] <https://www.codeproject.com/Articles/839389/Fast-Point-Cloud-Viewer-with-Csharp-and-OpenGL>
Stand: 7.7.2022

[4] Yilmaz, Ozgur and Karakus, Fatih, 2013, Stereo and kinect fusion for continuous 3D reconstruction and visual odometry, isbn 978-1-4799-3343-3

[5] <https://realitybytes.blog/2018/05/10/computer-vision-perception-structure-from-motion/>
Stand: 7.7.2022

[6] Structure from Motion with Objects, Marco Crocco, Cosimo Rubino, Alessio Del Bue, CVPR, https://openaccess.thecvf.com/content_cvpr_2016/papers/Crocco_Structure_From_Motion_CVPR_2016_paper.pdf

[7] <https://drohnenwissen.com/mini-drohne-dji-ryze-tello-im-testbericht/>
Stand: 9.7.2022

[8] <https://i.ebayimg.com/images/g/0zUAAOSwWLtiZnKv/s-l640.jpg>
Stand: 9.7.2022

Vorkenntnisse

Es ist hilfreich – aber keine Voraussetzung – wenn Sie Vorkenntnisse/Interesse mitbringen in

- Softwareentwicklung
 - Python
 - Grundkonzepte der Bildverarbeitung / Computervision
 - Data Engineering
-

Vorarbeiten/Voraussetzungen

Hier finden sie – zur weiteren Information - Videos zum *Structure from Motion* auf YouTube:

<https://www.youtube.com/watch?v=i7ierVkXYa8>
<https://www.youtube.com/watch?v=JIOzyyhk1v0>

Eine Implementation für das *Structure from Motion* in Python finden Sie hier:

<https://github.com/Ashok93/Structure-From-Motion-SFM->

Testbild-Daten für das *Structure from Motion* von der ETH Zürich finden Sie hier:

<https://www.eth3d.net/datasets>

Es stehen zunächst zwei Drohnen zum Aufsetzen des Projektes zur Verfügung:

- o DJI Trello Mini Drohne
 - Max. 30 km/h
 - Max. 12 min Flugzeit
 - 20-50 m Reichweite effektiv (100 m laut Herstellerangabe unter absoluten Idealbedingungen)
 - Max. 10m Flughöhe
 - 720 dpi Kamerabilder welche per UDP übertragen werden (z.B. an einen Laptop)
 - 5 MP Kamera und kann Videos mit 720p und 30 Bildern pro Sekunde aufnehmen.
 - Wir liegen in der „kleiner 250g“ Drohnen-Klasse

Je nach Verlauf und Bedürfnisse im Projekt können weitere Anschaffungen wie PADs, Rechner, Drohnen etc. getätigt werden.

Zur Kommunikation und Python-basierter Regelung/Steuerung der Drohnen über Funk steht die Python-Bibliothek *tello-python* zur Verfügung (pip install tello-python):

<https://pypi.org/project/tello-python/>

Beispiele und Tutorials, wie diese Bibliothek angewandt wird:

<https://www.youtube.com/watch?v=hahOMH4Y74w>

<http://46.38.235.241/webpage/dirfiles/misc/DJITelloDrohneProgrammieren/Tello-SDK-2.0-User-Guide-2018.11.pdf>

<http://46.38.235.241/webpage/dirfiles/misc/DJITelloDrohneProgrammieren/DJI-Tello-mit-Python-3-programmieren.pdf>

<http://46.38.235.241/webpage/dirfiles/misc/DJITelloDrohneProgrammieren/pythonBespTelloDrohneFlugbefehle.py>

<http://46.38.235.241/webpage/dirfiles/misc/DJITelloDrohneProgrammieren/pythonBespTelloDrohneKameraAuslesen.py>

Kurzeinführung in Python:

https://www.youtube.com/watch?v=x_kYpwi1L1k

OnboardingProzess

Alle wichtigen Zugänge einrichten um die Arbeit am Projekt aufnehmen zu können:

<http://46.38.235.241/webpage/dirfiles/misc/onboarding/OnboardingProzess.pdf>

Organisatorisches

Projektmanagement per Scrum in Trello:

<https://trello.com/b/bncilBU5/3d-flight-scanner>

Projektcode-Management per GitHub:

<https://github.com/DirkJLehmann/3DFlightScanner.git>

Projektdokumentations-Management:

https://docs.google.com/document/d/1jgI0qrJ96WdNo8PdWLzFGGe_IKOeUfjhEeBrWBCETzUg/edit?usp=sharing

Bei Interesse melden Sie sich bitte unter:

Prof. Dr.-Ing.habil.
Dirk Joachim Lehmann
Data Science in IoT
Fakultät für Informatik
di.lehmann@ostfalia.de