

Projekt zum Abschluss des Studiums

Analyse von Sportereignissen mit künstlichem Sehen.

Pablo Saura Pérez

pablo.saura@um.es

48742017A

Ausbilder: Alberto Ruiz García

13. September 2017

Index

1. Zusammenfassung.	6
2. Erweiterte Zusammenfassung.	10
3. Einleitung.	14
3.1. Aufbau des Dokuments	15
4. Stand der Technik.	16
4.1. Computer Vision.....	16
4.2. Farbige Darstellung	17
4.2.1. RGB-Darstellung	18
4.2.2. Graustufen	18
4.3. Bildverarbeitungsbibliotheken	19
4.4. Vorhersagesysteme	19
4.5. Visionen für den Sport	20
4.5.1. Verfolgung von Anwendungen.....	20
4.5.2. Verhaltensanalyse	21
4.5.3. Entscheidungsfindung	21
5. Analyse der Ziele und der Mak	22
5.1. Aufgabenstellung	22
5.2. Mak	22
5.3. Zu erreichende Ziele	24
5.4. Technologien und Werkzeuge	24
6. Entwurf und Lösung der durchgeführten Arbeiten.	25
6.1. Suche nach dem zu untersuchenden System	25
6.2. Bildsegmentierung	27
6.3. Nachverfolgung.....	32
6.4. Vorhersage	36
6.5. System in der Umsetzung	45
7. Schlussfolgerungen und zukünftige Ruga	47
7.1. Schlussfolgerungen	47
7.2. Mögliche künftige Arbeiten	47

8. Anhang.	49
8.1. Gligo	49
8.2. Links zu Videos	55
9. Hilfe	55

Unterzeichnete Erklärung über die Originalität des Werks.

Ich, Pablo Saura Permit dem Personalausweis 48742017-A, Student des Studiengangs Computertechnik an der Fakultät für Informatik der Universität Murcia, erkläre, dass die von mir zur Bewertung vorgelegte Diplomarbeit ein Original und eine persönliche Ausarbeitung ist.

1. Resumen.

In Zeiten des ständigen Fortschritts und Wandels ~~w~~ie künstliche Intelligenz zu einer der Disziplinen, in die am meisten Zeit und Geld investiert wird. Eine Ära, in der massive Daten, Statistiken, Verhaltensmuster und generell alle Daten, die das Potenzial haben, genutzt zu werden, gesucht und analysiert werden. Die Verbesserung der Leistungsfähigkeit von Geräten und Computern im Allgemeinen ermöglicht es, die künstliche Intelligenz so weit voranzutreiben, dass Aufgaben, die früher nur von Menschen ausgeführt wurden, zunehmend von Maschinen übernommen werden.

Künstliches Sehen ist ein Zweig der künstlichen Intelligenz, der diesen Fortschritt erfährt, weil immer mehr Aufgaben automatisiert werden müssen, und eine Möglichkeit, sie zu überwachen, sind Kameras. Manchmal ist es nicht möglich, einen Sensor zu installieren, der die gesuchten Daten speichert, aber wir sind in der Lage, die Szene zu erfassen, sie zu analysieren und die gesuchten Daten zu erhalten. Die Analyse von Verhaltensweisen oder Situationen, die Automatisierung industrieller Aufgaben, die automatische Navigation (sowohl bei Robotern als auch bei ~~Flug~~) sind einige der Aufgaben, die in den letzten ~~Jahren~~ einen großen Wandel erfahren haben.

Im Abschnitt "Stand der Technik" geben wir einen kurzen Überblick über den aktuellen Stand dieser Disziplin, von ihrer Definition über die verschiedenen Disziplinen, die sie nutzen, bis hin zu einem Schema, das bei der Durchführung von Projekten dieser Art zu beachten ist. Wir werden uns die verschiedenen ~~Flüge~~ ansehen, die für Projekte zum künstlichen Sehen verwendet werden können. Abschließend werden wir verschiedene bestehende Anwendungen erörtern, die mit dieser Arbeit verwandt sind oder von denen wir Ideen für die Durchführung dieser Arbeit übernehmen können.

In dieser Arbeit widmen wir uns der Untersuchung des künstlichen Sehens im Bereich des Sports. Ziel ist es, eine Verbindung zwischen den Techniken des künstlichen Sehens und einigen Problemen in der Welt des Sports herzustellen, wobei die uns heute zur Verfügung stehenden Mittel genutzt werden. Eine Eigenschaft des Sports ist, dass er eine Leidenschaft weckt, die in Form von Sportübertragungen mit Kameras umgesetzt wird. Aus diesem Grund gibt es heute ~~se~~ Material in Form von Videos zu jeder Sportart, und die Möglichkeit, diese zu analysieren und Daten daraus zu extrahieren, beschäftigt viele Unternehmen in der Welt. Man ~~kön~~sagen, dass ~~letzte~~ bei jeder Fernsehübertragung einer beliebigen Sportart irgendeine Art von Anwendung gezeigt wird, bei der künstliches Sehen zum Einsatz kommt (Berechnung der zurückgelegten Strecke, der Flugbahnen usw.).

Analyse unter Verwendung einer relativ einfachen Architektur.

Die zu verwendenden Ressourcen sind: eine Kamera und eine Bildverarbeitungsbibliothek. Eine Kamera für die Aufnahme der zu analysierenden Szenen, aber es wurde beschlossen, sie für die Arbeit zu nehmen. Die Kamera wird die eines Mobiltelefons sein, so dass die Arbeit leicht reproduzierbar sein wird. Die verwendete Bildverarbeitungsbibliothek ist OpenCV in Verbindung mit der Sprache Python. OpenCV ist ~~keine~~ sehr leistungsfähige und weit verbreitete Bibliothek, so dass es eine Menge Informationen gibt, die Sie anleiten können.

Sobald wir über die Ressourcen verfügen, nehmen wir die Videos mit der Kamera auf, die wir analysieren wollen und auf die wir die Visualisierungstechniken anwenden wollen. Es wurde beschlossen, einige Videos vom Ballwerfen auszuwählen, da sie verschiedene Elemente enthalten, mit denen wir uns in dieser Arbeit beschäftigen wollen. Die Segmentierung des Objekts und seine Verfolgung werden einen großen Teil dieser Arbeit ausmachen. Nachdem die Techniken und Verfahren auf diese Videos angewandt wurden, werden sie auf ähnliche Situationen im Sport übertragen und schließlich auf eine reale Situation angewendet.

Zusammen mit der Segmentierung und Verfolgung des Objekts in den Videos wurde beschlossen, dass es auch interessant ~~wäre~~ eine Flugbahnvorhersage durchzuführen, da sie in einem skalierbaren Modell wie der ballistischen Flugbahn enthalten ist.

Zum jetzigen Zeitpunkt haben wir 4 Hauptmodule, auf die sich unsere Arbeit konzentrieren wird.

Das erste Modul ist die Segmentierung und Isolierung des Elements von Interesse. In dieser Phase führen wir eine Analyse der verschiedenen Elemente durch, aus denen unsere Szene besteht. Je nach dem Verhalten dieser Elemente wird es notwendig sein, die eine oder andere Technik anzuwenden. Da die allgemeinen Techniken, die normalerweise untersucht werden, nicht direkt angewendet werden können, müssen sie an die Bedürfnisse der verschiedenen Probleme angepasst werden. Es gibt nicht für jedes Problem eine ~~best~~ Art von Technik, sondern es wird eine Reihe von Tests mit den verschiedenen Techniken durchgeführt, bis ein zufriedenstellendes Ergebnis erzielt wird.

In den Videos, die in dieser Arbeit behandelt werden, gibt es ein Hauptelement, nämlich den Ball, den wir von der Szene isolieren wollen; dies wird unser Hauptelement von Interesse sein. Dazu müssen wir auch andere Elemente wie den Hintergrund oder den Boden in unserer Szene berücksichtigen. Nach der Analyse wird schließlich beschlossen, einen statischen Partikel-Eliminator wie BackgroundSubtractorMOG2 zu verwenden, wobei Bereiche wie der Boden, der Himmel oder das Spielfeld, sofern vorhanden, nicht berücksichtigt werden. Der Ball, die Spieler, d. h. die sich bewegenden Elemente, ~~bleiben dagegen~~ in der Szene. Dann erreichen wir durch ~~morphologische Transformationstechniken wie Erosion und Dilatation, dass unsere zu~~

isolierenden Elemente in der Szene berücksichtigt werden.

sind in unserer Szene besser definiert.

Die Segmentierung kann sich von Anwendung zu Anwendung stark unterscheiden, ist also ein wichtiger Teil, und die Suche nach dem "optimalen" Weg ist nicht trivial, weshalb die Segmentierung bei der Implementierung einer Anwendung viel Zeit in Anspruch nehmen kann.

Das zweite Modul ist die Verfolgung des Elements von Interesse. In diesem Teil wenden wir die Techniken auf das zuvor isolierte Element von Interesse an, um es zu verfolgen. Diese Verfolgung liefert uns die Daten über die Position in unserer Szene, die wir für die anschließende Vorhersage verwenden werden. Die Verarbeitung von Tracking-Techniken kann wiederum für andere Projekte verwendet werden, wo dies erforderlich ist, nicht nur bei Projekten zum künstlichen Sehen im Sport.

Die Nachverfolgung ist eines der größten Probleme, die bei einer Anwendung dieser Art auftreten können. Dank Techniken wie Mean-Shift, insbesondere der Camshift-Technik, wird bei dieser Arbeit ein hervorragendes Ergebnis erzielt. Durch die Zusammenführung der Segmentierung, bei der wir den Ball korrekt vom Rest der Elemente trennen, wird die Verfolgung auf zufriedenstellende Weise erreicht. Diese Camshift-Technik verwendet ein Farbmodell, indem sie ein Histogramm des zu suchenden Elements erstellt und mit der Szene vergleicht. Diese Technik sucht nach den Bereichen mit der höchsten Pixeldichte, die das Histogramm mit der Szene gemeinsam hat. Einmal gefunden, kann es diesen Punkt höchster Dichte in den folgenden Bildern verfolgen. Dieser Punkt höchster Dichte entspricht dem Element von Interesse in unserer Szene.

Wenn es uns gelungen ist, die Positionen des Zinselements zu speichern, betrachten wir dieses Modul als abgeschlossen.

Das dritte Modul dient der Vorhersage des Zustands des Schnittstellenelements. Bei den aufgenommenen Videos handelt es sich um Szenen, in denen sich das interessierende Element in einer vertikalen Ebene senkrecht zur optischen Achse der Kamera bewegt. Diese Einschränkung bei der Aufnahme der Videos wurde vorgenommen, weil dies die Anwendung von 2D-Vorhersagetechniken ermöglicht. Wäre dies nicht der Fall, ~~wäre~~ ~~und~~ der Einsatz dieser 2D-Techniken nicht die erwarteten Ergebnisse ~~liefern~~, ~~sondern~~ wir müssten auf 3D-Techniken zurückgreifen. Herkömmliche geometrische 3D-Ansichten erfordern mehrere Ansichten, d. h. verschiedene Schalen. Der Einsatz dieser Kameras ~~ist~~ ~~unser~~ Ziel der Arbeit, nämlich die Verwendung einer einzigen Kamera und von Lichtquellen, übersteigen.

Für diese Phase haben wir einen Kalman-Filter implementiert, der diese Vorhersage in 2D-Systemen ermöglicht. Da wir ein lineares dynamisches System wie die ~~haben~~ ~~haben~~, ist dieser Filter mit zufriedenstellenden Ergebnissen anwendbar. Dank dieser Technik können wir

Eigenschaften erhalten, die

auf die wir nicht zugreifen können, zusammen mit einer Unsicherheit über diesen Wert. Dies wird uns helfen, zukünftige Positionen unseres Elements von Interesse zusammen mit der erwähnten Unsicherheit vorherzusagen. Diese Unsicherheit kann als Wahrscheinlichkeitswert für die Schätzung der Position verwendet werden; je höher der Wert, desto weniger müssen wir "glauben", dass sich das Element in einer zukünftigen Situation in dieser Position befinden wird. Bei einem niedrigen Unsicherheitswert können wir jedoch mit größerer Sicherheit davon ausgehen, dass er vorhanden oder sehr nahe daran ist. Dies ist der Grundgedanke, auf dem dieser Prädiktor beruhen wird.

In diesem Teil wird auch erörtert, wie das Problem bei einem nichtlinearen System mit Hilfe eines UKF-Filters gelöst werden kann.

Das vierte und letzte Modul ist die Erstellung einer sportbezogenen Anwendung. In dieser Phase wird das gesamte bis dahin vermittelte Wissen zusammengeführt und auf eine reale berufliche Anwendung übertragen. Die Anwendung besteht aus einem Basketball-Schussvorhersageprogramm. Auf diese Weise nutzen wir die Module Segmentierung, Verfolgung und Vorhersage, die wir in unserer Arbeit entwickelt haben.

Da dies das letzte Ziel der Arbeit ist, sind die Forschung und die Umsetzung an diesem Punkt abgeschlossen, und es bleiben nur noch die endgültige Bilanz der Arbeit und mögliche Fortsetzungen.

Die Anwendung desselben Systems auf andere Sportarten mit ähnlichen Wurf- und Flugbahnen, wie z. B. Speer- oder Diskuswerfen; Die Anwendung desselben Systems auf andere Sportarten mit ähnlichen Wurf- und Flugbahnen, wie z. B. Speer- oder Diskuswerfen; Die Verbesserung der Anwendung durch die Erhöhung der Ressourcen, wie z. B. mehr Kameras; Die Berechnung der 3D-Flugbahn durch die Verwendung anderer Algorithmen oder mehrerer Kameras, wie oben erwähnt; Die Verwendung der Grundlage dieser Arbeit für Sportanwendungen, die keine ballistischen Flugbahnen beinhalten, bei denen jedoch die Segmentierungs- und Verfolgungsphasen ebenfalls erforderlich sind. Der Code der durchgeführten Anwendungen ist ebenfalls Teil dieses Dokuments, um die Reproduktion der Experimente und mögliche Fortsetzungen der Arbeit zu erleichtern.

Abschließend noch ein kurzer Rückblick auf die erreichten Ziele. Definition einer zu verfolgenden Architektur und Erfüllung der Ziele mit ihr. Zufriedenstellender Abschluss der Beschaffung der zu verwendenden Ressourcen, im Falle dieser Arbeit der zu analysierenden Videos. Zufriedenstellende Anwendung der verschiedenen in der Analysephase vorgeschlagenen Techniken des künstlichen Sehens. Entwicklung einer Anwendung, die alle vorherigen Ziele vereint und somit alle festgelegten Ziele erfüllt.

Man kann sagen, dass das maschinelle Sehen und der Sport zwei große Bereiche sind und dass darüber hinaus die

Es hat sich gezeigt, dass ihr gemeinsamer Betrieb zu sehr nützlichen Anwendungen führt, die heute viele Möglichkeiten bieten.

2. Erweiterte Zusammenfassung.

In diesen Zeiten des ständigen Fortschritts und Wandels wird die künstliche Intelligenz zu einer der Disziplinen, in die immer mehr Zeit und Geld investiert wird. Eine Ära, in der Massendaten, Statistiken, Verhaltensmuster und generell alle Daten, die sich auswerten lassen, durchsucht und analysiert werden. Die zunehmende Leistungsfähigkeit von Geräten und Computern im Allgemeinen ermöglicht es der künstlichen Intelligenz, in Aufgaben vorzudringen, die bisher nur von Menschen ausgeführt werden konnten und nun zunehmend von Maschinen übernommen werden sollen.

Computervision ist ein Teilgebiet der künstlichen Intelligenz, das diesen Fortschritt erfährt, weil immer mehr Aufgaben automatisiert werden müssen, und eine Möglichkeit, sie zu überwachen, sind Kameras. Manchmal ist es nicht möglich, einen Sensor zu installieren, der die gesuchten Daten speichert, aber wir sind in der Lage, die Szene zu erfassen, sie zu analysieren und die gesuchten Daten zu erhalten. Verhaltens- oder Situationsanalyse, Automatisierung von industriellen Aufgaben, automatische Navigation (sowohl bei Robotern als auch bei Fahrzeugen) sind einige der Aufgaben, die sich in den letzten Jahren stark verändert haben.

Im State of the Art wird ein kurzer Überblick über den aktuellen Stand dieser Disziplin gegeben, von ihrer Definition über die verschiedenen Disziplinen, die sich ihrer bedienen, bis hin zu einem Schema, das bei der Realisierung von Projekten dieser Art zu beachten ist. Wir werden uns die verschiedenen Bibliotheken ansehen, die für Projekte zum künstlichen Sehen verwendet werden können. Abschließend werden wir verschiedene Anwendungen diskutieren, die damit in Zusammenhang stehen oder aus denen Ideen für die Entwicklung dieser Arbeit übernommen werden können.

In dieser Arbeit befassen wir uns mit der Untersuchung des künstlichen Sehens im Bereich des Sports. Ziel ist es, eine Verbindung zwischen den Techniken des künstlichen Sehens und einigen Problemen in der Welt des Sports herzustellen und dabei die uns heute zur Verfügung stehenden Mittel zu nutzen. Eine Eigenschaft des Sports ist, dass er ein Hobby ist, und dieses Hobby wird in Sportübertragungen umgesetzt, die mit Kameras gemacht werden. Deshalb gibt es heutzutage eine Menge Material in Form von Videos über jede Sportart, und die Möglichkeit, diese zu analysieren und Daten daraus zu gewinnen, beschäftigt viele Unternehmen in der Welt. Man könnte sagen, dass es heutzutage

bei jeder Fernsehübertragung einer Sportart zahlreiche Anwendungen gibt, bei denen die Computervision zum Einsatz kommt: Berechnung der zurückgelegten Strecke, der Flugbahnen usw.

Im Gegensatz zu anderen Arbeiten oder Ansätzen werden wir uns hier auf die Analyse konzentrieren, indem wir eine recht einfache Architektur verwenden.

Die zu verwendenden Ressourcen sind: eine Kamera und eine Bildverarbeitungsbibliothek. Eine Kamera für die Aufnahme der zu analysierenden Szenen. Man könnte auch nach bereits gemachten Videos suchen, aber es wurde beschlossen, sie speziell für diese Arbeit aufzunehmen. Die Kamera wird die eines Mobiltelefons sein, so dass die Berechnungen leicht reproduzierbar sein werden. Die verwendete Bibliothek für künstliches Sehen ist OpenCV zusammen mit der Sprache Python. OpenCV ist heute eine sehr leistungsfähige und weit verbreitete Bibliothek, so dass es eine Menge Informationen gibt, die Sie anleiten.

Sobald wir die Ressourcen haben, nehmen wir mit der Kamera die Videos auf, die wir analysieren wollen und in denen wir mit der Durchführung der Vision-Techniken beginnen wollen. Wir haben Videos von Ballwürfen gewählt, weil sie verschiedene Elemente enthalten, mit denen wir uns in der Arbeit beschäftigen wollen. Die Segmentierung des Objekts und seine Überwachung werden einen großen Teil dieser Arbeit ausmachen. Sobald die Techniken und Verfahren auf diese Videos angewandt wurden, werden sie auf ähnliche Situationen im Sport übertragen und schließlich auf eine reale Situation angewendet.

Zusammen mit der Segmentierung und Verfolgung des Objekts in den Videos wurde beschlossen, dass es auch interessant wäre, eine Vorhersage der Flugbahn zu machen, da sie innerhalb eines skalierbaren Modells wie der ballistischen Flugbahn liegt.

Zum jetzigen Zeitpunkt haben wir 4 Hauptmodule, auf die sich unsere Arbeit konzentrieren wird.

Das erste Modul ist die Segmentierung und Isolierung des Elements von Interesse. In dieser Phase führen wir eine Analyse der verschiedenen Elemente durch, aus denen sich unsere Szene zusammensetzt. Je nach dem Verhalten dieser Elemente wird es notwendig sein, die eine oder andere Technik anzuwenden. Da die in der Regel untersuchten generischen Techniken nicht direkt angewendet werden können, müssen sie an die Bedürfnisse der verschiedenen Probleme angepasst werden. Es gibt nicht für jedes Problem eine spezifische Technik. Stattdessen gibt es eine Reihe von Tests mit den verschiedenen Techniken, um ein zufriedenstellendes Ergebnis zu erzielen.

In den Videos, die in dieser Arbeit behandelt werden, gibt es ein Hauptelement, nämlich den Ball, den wir von der Szene isolieren wollen; dies wird unser Hauptelement von Interesse sein. Dazu müssen wir auch andere Elemente wie den Hintergrund oder den Boden in unserer Szene berücksichtigen. Schließlich wird nach der Analyse beschlossen, einen statischen Teileliminator wie BackgroundSubtractorMOG2 zu

verwenden, da Bereiche wie der Boden, der Himmel und die Neigung, falls vorhanden, nicht berücksichtigt werden. Stattdessen würden der Ball, die Spieler, d. h. die sich bewegenden Elemente in der Szene bleiben. Anschließend werden durch morphologische Transformationstechniken wie Erosion und Dilatation die zu isolierenden Elemente in unserer Szene besser definiert.

Die Segmentierung kann sich von Anwendung zu Anwendung stark unterscheiden, daher ist es ein

Diese Segmentierung ist ein wichtiger Bestandteil und die Suche nach dem besten Weg ist nicht trivial, weshalb die Erstellung eines Antrags sehr viel Zeit in Anspruch nehmen kann.

Das zweite Modul ist die Verfolgung des Elements von Interesse. In diesem Teil wenden wir die Techniken auf das zuvor isolierte Element von Interesse an, um es weiter zu verfolgen. Diese Nachverfolgung wird uns Daten über die Position in unserer Szene liefern, die für die anschließende Vorhersage nützlich sind. Gleichzeitig können Tracking-Techniken auch bei anderen Projekten eingesetzt werden, wo dies erforderlich ist, nicht nur bei Projekten zum künstlichen Sehen im Sport.

Die Nachverfolgung ist eines der größten Probleme, die bei einer Anwendung dieser Art auftreten können. Dank Techniken wie Mean-Shift, insbesondere der Camshift-Technik, kann bei dieser Arbeit ein gutes Ergebnis erzielt werden. Durch die Verbindung mit dem Segmentierungsteil, bei dem es uns gelingt, den Ball korrekt von den übrigen Elementen zu trennen, wird die Verfolgung in zufriedenstellender Weise erreicht. Diese Camshift-Technik verwendet ein Farbmodell, indem sie ein Histogramm des gesuchten Objekts erstellt und es mit der Szene vergleicht. Bei dieser Technik wird nach den Bereichen mit der höchsten Pixeldichte gesucht, die das Histogramm mit der Szene gemeinsam hat. Einmal gefunden, kann es diesem Punkt größerer Dichte entlang der folgenden Bilder folgen. Dieser Punkt größerer Dichte entspricht dem Element von Interesse in unserer Szene.

Sobald wir die Positionen des Elements von Interesse gespeichert haben, schließen wir dieses Modul ab.

Das dritte Modul ist die Vorhersage des Zustands des Elements von Interesse. Bei den aufgenommenen Videos handelt es sich um Szenen, in denen sich das betreffende Element in einer vertikalen Ebene senkrecht zur optischen Achse der Kamera bewegt. Diese Einschränkung bei der Aufnahme des Videos wird gemacht, weil wir so 2D-Vorhersagetechniken anwenden können. Wäre dies nicht der Fall, würde die Anwendung dieser 2D-Techniken nicht die erwarteten Ergebnisse liefern, sondern wir müssten auf 3D-Techniken zurückgreifen. Traditionelle geometrische 3D-Vision-Techniken erfordern mehrere Ansichten, d. h. verschiedene Kameras. Der Einsatz dieser Kameras würde über das Ziel unserer Arbeit hinausgehen, das in der Verwendung einer einzigen Kamera und von Lichtressourcen besteht.

Für diese Phase wurde ein Kalman-Filter implementiert, der es uns ermöglicht, diese Vorhersage in 2D-Systemen zu verwenden. Bei einem linearen dynamischen System wie der

Ballistik ist dieser Filter mit zufriedenstellenden Ergebnissen anwendbar. Dank dieser Technik können wir Eigenschaften, auf die wir nicht zugreifen können, zusammen mit einer Unsicherheit über diesen Wert schätzen. Dies wird uns helfen, zukünftige Positionen unseres Elements von Interesse zusammen mit dieser Unsicherheit, die wir diskutiert haben, vorherzusagen. Wir können diese Unsicherheit als einen Wahrscheinlichkeitswert für die Schätzung der Position verwenden, je größer

der Wert, den wir haben, um "glauben uns weniger "als in einer zukünftigen Situation das Element in dieser Position sein wird. Aber mit einer Unsicherheit von geringem Wert können wir sicherer sein, dass es knapp wird. Das ist im Grunde die Idee, auf der dieser Prädiktor basieren wird.

In diesem Teil haben wir auch erörtert, wie das Problem bei einem nichtlinearen System durch die Verwendung eines UKF-Filters gelöst werden kann.

Das vierte und letzte Modul ist die Erstellung einer sportbezogenen Anwendung. In dieser Phase werden alle bis dahin behandelten Kenntnisse zusammengeführt und in Bezug auf die Arbeit in eine reale Anwendung gebracht. Die Anwendung besteht aus einer Basketballkorb-Wurfvorhersage. Auf diese Weise nutzen wir die Module Segmentierung, Verfolgung und Vorhersage, die wir in unserer Arbeit entwickelt haben. Wir haben verschiedene Videos von Basketballwürfen aufgenommen, damit wir sehen konnten, wie es in verschiedenen Situationen funktioniert, von Bällen, die reingehen, bis zu solchen, die nicht reingehen. Wir hätten gerne implementiert, dass das Programm, während der Ball unterwegs ist, einen gewissen Prozentsatz der Eintrittswahrscheinlichkeit angibt, aber das lassen wir aus Zeitgründen für die Zukunft.

Da dies das eigentliche Ziel der Arbeit ist, sind die Forschung und die Umsetzung abgeschlossen, und wir haben nur noch die endgültige Bilanz der Arbeit und die möglichen Folgemaßnahmen.

Wir sind sehr zufrieden mit der Art und Weise, wie sich die Arbeit und die Ergebnisse, die wir erzielt haben, entwickelt haben. Es ist wahr, dass die endgültige Anwendung recht einfach ist, aber es wurde versucht, das Maximum zu verallgemeinern und die Möglichkeit zu geben, auf andere Projekte zu übertragen. Ich für meinen Teil schließe diese Arbeit mit einigen Kenntnissen über Computer Vision ab, die ich sonst nicht erworben hätte, und mit dem Wunsch, andere Arbeiten in diesem Bereich fortzusetzen. Die Anwendung der Computervision bei der Entwicklung autonomer Autos oder die Nutzung von Techniken zur Erkennung möglicher Gefahren sind zweifellos die Bereiche, die für mich am interessantesten sind. Auch die Einführung neuer Techniken und Anwendungen in diesem Sport wird die Art und Weise, wie wir sie kennen, verändern, denke ich. Wir erwarten in naher Zukunft viele Anwendungen dieser Art im professionellen Wettbewerb.

Verschiedene Themen werden als künftige Arbeitsbereiche angesprochen. Die Anwendung desselben Systems auf andere Sportarten mit ähnlichen Wurf- und Flugbahnen,

wie z. B. Speer- oder Diskuswerfen. Die Verbesserung der Anwendung durch mehr Ressourcen, z. B. mehr Kameras. Die Berechnung der 3D-Trajektorie mit Hilfe anderer Algorithmen oder mehrerer Kameras, wie wir bereits erörtert haben. Die Nutzung der Grundlagen dieser Arbeit für Sportanwendungen, bei denen es nicht um ballistische Flugbahnen geht, bei denen aber auch die Phasen der Segmentierung und Überwachung erforderlich sind. Der Code der gestellten Anträge ist ebenfalls Teil dieses Dokuments, um die Reproduktion zu erleichtern

der Experimente und die mögliche Kontinuität der Arbeit.

Abschließend noch ein kurzer Überblick über die durchgeführten Aufgaben. Definieren Sie eine zu fördernde Architektur und erfüllen Sie damit die Ziele. Die Beschaffung der zu verwendenden Ressourcen, im Falle dieser Arbeit die zu analysierenden Videos, in zufriedenstellender Weise durchzuführen. die verschiedenen in der Analysephase vorgeschlagenen Techniken des künstlichen Sehens in zufriedenstellender Weise anzuwenden. Erstellen Sie einen Antrag, der alle oben genannten Ziele vereint und somit alle festgelegten Ziele erfüllt.

Wir können sagen, dass das künstliche Sehen und der Sport zwei große Bereiche sind, und es hat sich auch gezeigt, dass ihr gemeinsamer Einsatz heute sehr nützliche Anwendungen mit vielen Möglichkeiten hervorbringt.

3. Einführung n.

Wir leben in einer Welt, in der sich die Computertechnologie sprunghaft weiterentwickelt. Jedes ~~kommen~~ neue, leistungsfähigere Prozessoren zu niedrigeren Preisen auf den Markt, und auch die Kameras werden immer besser und bieten mehr Möglichkeiten. Es ist nicht ~~kl~~ass sich das maschinelle Sehen ~~ke~~iner der fortschrittlichsten Disziplinen entwickelt. Die maschinelle Bildverarbeitung braucht diese beiden Säulen, und immer leistungsfähigere Maschinen und Kameras tragen dazu bei, dass immer mehr Anwendungen, die auf die maschinelle Bildverarbeitung angewiesen sind, unterstützt werden.

Gleichzeitig führt die zunehmende Leistungsfähigkeit der ~~Teil~~ dazu, dass auch Bereiche, in denen sie bisher nicht eingesetzt wurde, von ihr Gebrauch zu machen beginnen. Sport hat es schon immer gegeben, aber heutzutage ~~und~~ diese beiden Disziplinen aufgrund des hohen Wettbewerbsniveaus, des großen Geldvolumens und der oben erwähnten großen technologischen Macht zusammen.

In dieser Arbeit wird versucht, dies zu vereinen, indem eine Anwendung für künstliches Sehen geschaffen wird, die im Sport eingesetzt werden kann. Ein wichtiger Teil davon ist die Erstellung der Anwendung mit Mitteln, die später reproduzierbar sind und keinen großen finanziellen Aufwand erfordern.

Ziel ist es, die Bewegung von Elementen von Interesse in einer Sportszene (Ball, Spieler...) zu erkennen und zu analysieren. In dieser Arbeit werden wir die verschiedenen Probleme untersuchen, die auftreten können, um das Ziel zu erreichen, und wir werden eine Lösung für

einfache Fälle vorschlagen, die als Ausgangspunkt dienen wird. Diese Lösung soll wiederum mit einem kostengünstigen Ansatz realisiert werden, bei dem wir nur eine einzige Kamera und einen Computer benötigen.

mit einer ~~Real~~künstliches Sehen.

Das Erreichen einer praktikablen Lösung mit solch begrenzten Ressourcen wird uns helfen, uns vorzustellen, wie leistungsfähig das künstliche Sehen mit mehr Ressourcen sein ~~könnte~~

Gleichzeitig werden im Rahmen der Arbeit verschiedene Probleme, die bei der Entwicklung einer Bildverarbeitungsanwendung auftreten können, wie die Segmentierung und Verfolgung von Objekten in Videos, erläutert und gelöst. Schließlich wird ein Prädiktor für zukünftige Zustände entwickelt und auf die Sportanwendung angewendet.

3.1. Aufbau des Dokuments .

Dieses Dokument ist wie folgt gegliedert: Eine kurze Zusammenfassung des aktuellen Stands der Technik des Rahmens, in den diese Arbeit eingebettet ist. Im nächsten Kapitel wird eine Analyse vorgenommen und es werden die Ziele festgelegt, die mit dieser Arbeit erreicht werden sollen. Es folgt die Entwurfs- und Umsetzungsphase, und im letzten Kapitel werden mögliche zukünftige ~~Ergebnisse~~ untersucht und die allgemeinen Schlussfolgerungen der Arbeit gezogen. Die blau markierten Wörter sind mit einem Link versehen, dem Sie folgen können, um weitere Informationen zu erhalten oder der zu Demonstrationsvideos führt.

4. Auf dem neuesten Stand der Technik .

4.1. Vision durch Computer.

Das maschinelle Sehen ist ein Teilgebiet der künstlichen Intelligenz, das sich mit der Erfassung von Informationen durch die Analyse von Bildern mit Hilfe eines Computers befasst. Die Informationen, die ein Mensch über den Sehsinn aus seiner Umgebung wahrnehmen kann, sind keine einfache Aufgabe für einen Computer. Das künstliche Sehen versucht, ein digitales Bild, d. h. eine Reihe von numerischen Daten, in verwertbare Informationen umzuwandeln, so wie es unser Gehirn tun würde.

Die britische Machine Vision Association und die Gesellschaft für Mustererkennung (Society for Pattern Recognition) haben sich zurückgezogen.

Der Begriff des künstlichen Sehens wird wie folgt definiert:


"Die Computer Vision befasst sich mit der automatischen Extraktion, Analyse und dem Verständnis nützlicher Informationen aus einem einzelnen Bild oder einer Bildfolge. Es geht um die Entwicklung einer theoretischen und algorithmischen Grundlage für ein automatisches visuelles Verständnis"[5].

Die Übersetzung ~~wäre~~ lauten: "Computer Vision befasst sich mit der automatischen Extraktion, Analyse und dem Verständnis nützlicher Informationen aus einem einzelnen Bild oder einer Bildfolge. Es geht um die Entwicklung einer theoretischen und ~~algorithmischen~~ Grundlage für ein automatisches visuelles Verständnis".

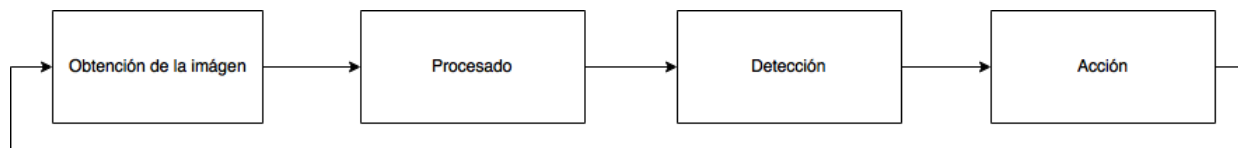
Aus dieser Definition ergibt sich, dass sie zum Bereich der künstlichen Intelligenz gehört. Konzepte wie die automatische Extraktion, Analyse und das Verstehen von Informationen mit Hilfe von Algorithmen spiegeln diesen Teil der Informatik wider und haben das maschinelle Sehen zu einer der am schnellsten wachsenden Disziplinen im Informationszeitalter gemacht.

Heute können wir feststellen, dass die maschinelle Bildverarbeitung in vielen verschiedenen Bereichen und für viele verschiedene Aufgaben eingesetzt werden kann. Einige dieser Aufgaben, an denen die maschinelle Bildverarbeitung beteiligt ist, sind die folgenden:

- Robotik: Sowohl Navigations- als auch Prozesssteuerungsaufgaben in der Industrie.
- Medizin: Bildverarbeitung für die Diagnose.
- ~~Topf~~ Durch Oberflächenmodellierung anhand von Bildmaterial (z. B. Google Maps).
- Klassifikatoren: Muster- und Formerkennung bei Objekten (z. B. biometrische Scanner).

- Navigation: Einsatz in selbstfahrenden Autos oder unbemannten 
- Ereigniserkennung: Durch Variationen in den Bildern (z.B. Personenzählung, Ein- und Ausreisekontrolle).
- Automatisches Lernen.
- Viele andere...

Um all diese Aufgaben erfüllen zu können, müssen jedoch zunächst die zu verarbeitenden Informationen beschafft werden. Zu diesem Zweck werden wir kurz erläutern, wie diese Informationen gewonnen und anschließend analysiert werden.



Eine Möglichkeit, diesen Prozess zu verstehen, ist in der Abbildung dargestellt. Jede Phase wird im Folgenden ausführlicher behandelt.

- Bildaufnahme: In dieser Phase wird das Bild oder die Bildsequenz, die analysiert werden soll, mit einer Kamera aufgenommen.
- Verarbeitung: Diese Phase besteht aus der Anwendung von Techniken oder Anweisungen zur Veränderung des Bildes, um ein neues Bild zu erhalten, in dem die gewünschte Erkennung leichter möglich ist. Zu diesen Techniken gehören: Glättung, Farbmodellierung, Dehnung, Filterung und viele andere Techniken.
- Erkennung: Sobald das Bild verarbeitet wurde, werden Erkennungstechniken eingesetzt, um die Informationen im Bild zu finden, die für uns von Wert sind (z. B. die Erkennung eines Objekts).
- Aktion: Mit dem Ergebnis der Erkennung können wir die Aktion anwenden, für die wir unser System entwickelt haben (z. B. das Verfolgen des zuvor erkannten Objekts).

4.2. Farbdarstellung .

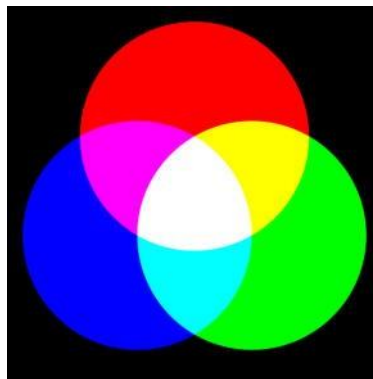
Farbdarstellungen sind mathematische Modelle, mit denen Farben durch Zahlenwerte dargestellt werden können. Da ein Computer nicht die gleiche Farbwahrnehmung hat wie ein Mensch, müssen Farben kodiert und durch Werte dargestellt werden, die Computer verstehen können.

Die gebräuchlichste Form der Darstellung sind numerische Vektoren, und die verschiedenen in dieser Arbeit verwendeten Typen werden im Folgenden erörtert.

4.2.1. RGB-Darstellung.

In diesem Modell werden die Farben durch Tripel von normalerweise 8-Bit-Ganzzahlen dargestellt, Ganzzahlwerte zwischen 0 und 255. Daher haben wir einen Vektor der Form (R, G, B) , wobei R, G, B die Intensität jeder Farbe (Rot, Grün und Blau) darstellen und das Tripel eine Farbe durch Addition dieser drei darstellt. Einige Beispiele sind:

- $(255, 0, 0)$ Farbe Rot.
- $(0, 0, 0)$ Keine Farbe, d. h. schwarz.
- $(255, 255, 255)$ Weiß.



Es ist zu beachten, dass das Modell die verschiedenen Farben nicht genau definiert. Je nach Gerät, das dieses Modell verwendet, kann derselbe RGB-Wert anders aussehen. Das Farbmodell ist das gleiche, aber der Farbraum kann variieren.

4.2.2. Graustufen.

Ein Graustufenbild ist nichts anderes als ein Wert für jedes \mathbf{p} in 8 Bits von 0 bis 255, wobei jeder Wert eine Art von Grau ist. Graustufen werden wegen ihrer Einfachheit bei der Anwendung von Computer-Vision-Techniken häufig für die Bildverarbeitung verwendet.

Rein rechnerisch ist ein Graustufenbild eine Matrix aus Breite und Höhe, wobei jede Position ein \mathbf{p} mit dem Grauwert ist.

Zum Abschluss dieses Abschnitts sei noch erwähnt, dass es neben diesen auch andere Farbdarstellungen gibt: HSL, YUV, LAB, LUV,...

4.3. Bildverarbeitungsbibliotheken .

Wenn wir eine Anwendung erstellen, in der wir Computer Vision verwenden, müssen wir wählen, welche der vorhandenen Bibliotheken wir verwenden wollen. In diesem Abschnitt werden wir die verschiedenen ~~Bib~~besprechen, die ~~la~~für die Erstellung dieser Art von Anwendungen existieren.

- [Accord.NET](#) ist ein .Net Framework für maschinelles Lernen, kombiniert mit Audio- und Bildverarbeitung.
- [Ilastik](#) Wie sie sich selbst definieren: Benutzerfreundliches und einfaches Werkzeug für die Klassifizierung, Segmentierung und Analyse von Bildern.
- [Matlab-Bibliotheken](#) Matlab bietet eine Reihe von Bildverarbeitungsfunktionen und -algorithmen zur Verwendung in seiner Plattform.
- [OpenCV](#) ist die ~~Bib~~ die wir in dieser Arbeit verwenden und die sowohl für den akademischen als auch für den kommerziellen Gebrauch kostenlos ist. Es enthält Schnittstellen für C++, C, Python und Java. Es wurde so konzipiert, dass es für den Einsatz in Echtzeitanwendungen rechnerisch effizient ist. Mit über 47 Millionen Nutzern ist sie ~~ih~~rer Website eine der größten Communities.
- [SimpleCV](#) Eine ~~Bib~~künstliches Sehen für Leute, die die Leistung von OpenCV nicht benötigen, und zwar auf eine einfachere Art und Weise, so definieren sie sich auf ihrer Website.

4.4. Vorhersagesysteme n.

Zu den Anwendungen des maschinellen Sehens gehören prädiktive Anwendungen mit Kameras.

Ein Beispiel sind selbstfahrende Fahrzeuge, die derzeit entwickelt werden. Diese Fahrzeuge verwenden Kameras, um die Umgebung zu modellieren, und versuchen gleichzeitig, das Verhalten anderer Fahrzeuge auf der Straße vorherzusagen, um Entscheidungen vorzuszusehen.

Ein weiteres Beispiel für eine Anwendung ist die von der Universität der Algarve entwickelte Anwendung Pool Live Aid. Diese Anwendung hilft Billardanfängern, ihre Fähigkeiten zu verbessern. Es verwendet eine Kamera, um die Flugbahn des Balls auf dem Tisch zu berechnen.



Bild von der Website des Unternehmens

4.5. Visio'n angewandt auf Sport .

Die große Leidenschaft, mit der ~~les~~ sportliche Wettkämpfe ausgetragen werden, und die mitunter komplizierten Entscheidungen über die Regeln haben zum Einsatz von Technologie im Sport geführt.

Die Vision ist einer der Notfallbereiche, wenn es darum geht, die Regeln der verschiedenen Spiele anzuwenden, aber die Vision ist auch bei der Vorbereitung von Spielen, beim Training oder bei der Analyse von bereits stattgefundenen Spielen präsent. In einem Sektor, in dem am ~~h~~ viel Geld bewegt wird, wird die Einführung von ~~Technologie~~ wie dem künstlichen Sehen zu einer besseren Vorbereitung und faireren Spielen führen.

Es gibt zahlreiche Anwendungen von künstlichem Sehen im Sport, wir werden einige der häufigsten im Alltagssport auflisten und kommentieren.

4.5.1. Verfolgung von Anwendungen.

Man kann sagen, dass der Sport das Zeitalter der Massendaten erlebt hat, bei jedem Spiel werden wir mit zahlreichen Statistiken und Daten über die Mannschaften oder Teilnehmer überschwemmt. Die Entwicklung von Anwendungen, die ein aufgezeichnetes Spiel analysieren können, hat es möglich gemacht, zahlreiche Statistiken zu erstellen, die von den Mannschaften selbst, den gegnerischen Teams, den Fernsehsendern und sogar den Wettenden genutzt werden.

Ein bekanntes System ist das System zur Messung der von den Spielern bei Fußballspielen zurückgelegten Strecke. Die FIFA (Federation Internationale de Football Association) verwendet bei ihren Wettbewerben die **MATRICS-Technologie** mit Hilfe von Kameras im Stadion.

ist es möglich, die Position der Spieler zu jedem Zeitpunkt zu berechnen und somit ihre Bewegung und die zurückgelegte Strecke während des gesamten Spiels zu ermitteln.

4.5.2. Verhaltensanalyse.

SECOND SPECTRUM ist ein Unternehmen, das ein Kamerasystem zur Erfassung und Analyse des Spielerverhaltens beim Basketball einsetzt.

Dieses System analysiert zum Beispiel die verschiedenen Schüsse eines Spielers und prognostiziert die Wahrscheinlichkeit eines Schusses in Abhängigkeit von der Zone, in der er sich befindet, was dem Spieler helfen kann, von Positionen aus zu trainieren, in denen er nicht so erfolgreich ist.

4.5.3. Entscheidungsfindung.

Der bekannteste Sportrichter der Welt ist zweifellos das **HawkEye**. Dieses System wurde 2005 im Tennis eingeführt und ~~besteht~~ aus einer Reihe von strategisch platzierten Kameras, die mittels Triangulation und Geschwindigkeitsberechnung die Flugbahn des Balls berechneten. Anhand des Spielmodells und der Bilder, die mit einer Hochgeschwindigkeitskamera aufgenommen werden, wird die 3D-Position des Balls berechnet, so dass man weiß, ob ein Ball getroffen wurde oder nicht. Dieses System wurde von den Ingenieuren der Roke Manor Research Limited entworfen, aber das Projekt wurde später von **Hawk-Eye Innovations Ltd. übernommen**.

Im Jahr 2012 wurde das Hawk's Eye dann in eine der meistgespielten Sportarten der Welt, den Fußball, aufgenommen. In diesem Fall werden 7 Kameras verwendet, die im ~~Tor~~ platziert sind. Diese Kameras nehmen Bilder auf, die verarbeitet werden, um die 3D-Position zusammen mit einem Mikrochip im Inneren des Balls wiederherzustellen. Auf diese Weise ist es möglich zu erkennen, ob ein Ball vollständig im ~~Tor~~ gelandet ist und somit ein Tor auf der Anzeigetafel erscheint. Das System warnt den Schiedsrichter in dem Moment, in dem ein Tor gegeben werden sollte.

Beide Systeme haben dazu beigetragen, die Politik, die in diesen Sportarten aufgrund ihrer großen Anhängerschaft manchmal aufkommt, zu verringern.

5. Analyse der Ziele und der Methodik .

5.1. Stellenbeschreibung .

Ziel dieser Arbeit ist die Realisierung einer Anwendung, die das künstliche Sehen nutzt, um die große Leistungsfähigkeit dieses Bereichs der künstlichen Intelligenz zu testen, ohne dabei zahlreiche Ressourcen einsetzen zu müssen.

Die Arbeit besteht aus mehreren Teilen, zum einen aus der Untersuchung von Bildverarbeitungstechniken wie Segmentierung oder Verfolgung. Zweitens: Vorhersagetechniken, die auf reale Modelle angewandt werden, wie z. B. das ~~ML~~Modell. Schließlich die Anwendung der Schlussfolgerungen aus den ersten beiden Teilen auf Anwendungen im Bereich des Sports, insbesondere auf die Vorhersage von Flugbahnen bei Wurfsportarten.

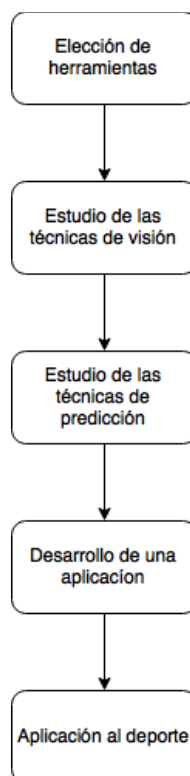
Für die Durchführung der Arbeiten sollen leicht zugängliche Ressourcen verwendet werden, für deren Reproduktion keine große Infrastruktur erforderlich ist. Es wird ~~also~~nur eine einzige Kamera verwendet, im Falle dieser Arbeit die eines Mobiltelefons, um die zu analysierenden Bilder aufzunehmen. Gleichzeitig werden frei verfügbare Computerwerkzeuge verwendet, die von jedermann heruntergeladen und genutzt werden können. Auf diese Weise wird die Arbeit einfach und leicht reproduzierbar sein, mit Blick auf mögliche Verbesserungen oder Weiterführungen in der Zukunft.

5.2. Methodik .

Sobald die Arbeit definiert ist, legen wir die ~~Methode~~Methode fest, mit der sie durchgeführt werden soll. Wir werden nun die einzelnen Teile definieren:

- Wahl der Werkzeuge: Bei der Durchführung eines Bildverarbeitungsprojekts müssen die verschiedenen Werkzeuge berücksichtigt werden, die für die Beschaffung und Verarbeitung von Ressourcen zur Verfügung stehen. Einerseits müssen wir Bilder erhalten, also brauchen wir entweder Kameras, um die Bilder selbst aufzunehmen, oder wir müssen Bilder von einem Bildspeicherdienst erhalten. In unserem Fall werden wir die zu analysierenden Bilder oder Videos selbst aufnehmen und dafür die Kamera eines mobilen Geräts verwenden. Sobald wir die zu verarbeitenden Daten erhalten haben, benötigen wir eine ~~ML~~ML für künstliches Sehen. Im Falle dieser Arbeit werden wir OpenCV verwenden, das wir im Abschnitt über ~~Tools~~Tools und Werkzeuge erläutern werden. Bei der Auswahl einer ~~ML~~ML ist es immer besser, eine Programmiersprache zu wählen, mit der man sich wohlfühlt; in diesem Fall werden wir Python verwenden. Sobald wir all dies haben, können wir mit der Bearbeitung der Bilder beginnen.

- **Untersuchung der Bildverarbeitungstechniken:** Sobald wir die Bilder haben, müssen wir die Techniken durchführen, um uns auf die wichtigen Teile zu konzentrieren. Im Fall dieses Projekts ist es notwendig, mit Hilfe dieser Techniken das Objekt zu finden, für das die Vorhersage gemacht werden soll. Mit Hilfe von Segmentierungs- und Verfolgungstechniken lässt sich feststellen, an welcher Stelle sich das Objekt zu einem bestimmten Zeitpunkt befindet.
- **Untersuchung der Vorhersagetechniken:** Dank der im vorherigen Teil erhaltenen Positionsdaten konnte die Vorhersage der Position in den folgenden Augenblicken durchgeführt werden. In diesem Teil sollte das für die Aufgabe am besten geeignete Vorhersageverfahren angepasst werden.
- **Entwicklung einer Anwendung:** In diesem Teil werden wir alle in den vorangegangenen Schritten erzielten Fortschritte zusammenführen, um eine Anwendung zu erstellen, die zur Vorhersage der Flugbahn eines Objekts verwendet werden kann, und wir werden die Ergebnisse überprüfen, um zu sehen, ob sie für das, was wir suchen, geeignet sind.
- **Anwendung auf den Sport:** Schließlich werden wir nach Anwendungen des oben entwickelten Systems auf den Sportsektor suchen.



5.3. Ziele auf erfüllen.

Man ~~lässt~~ sagen, dass es einige Zwischenziele gibt, die wir im Abschnitt über ~~den~~ näher erläutert haben, aber das endgültige Ziel der Arbeit besteht darin, zu prüfen, ob es möglich ist, ein System zu schaffen, das eine Flugbahn vorhersagen kann, und dieses System gleichzeitig auf eine Sportart anzuwenden, wobei vernünftige Daten gewonnen werden, und es wird auch geschätzt, dass eine große Anzahl von Kameras nicht notwendig ist, wie in anderen bestehenden Anwendungen.

5.4. Technologien und Werkzeuge .

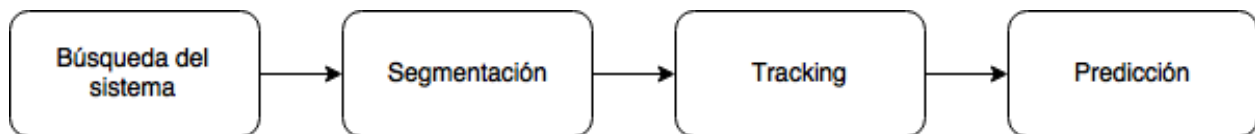
- Kamera: Die Bilder und Videos werden mit der Kamera eines Iphone 7 aufgenommen. Es wurde beschlossen, die Kamera eines mobilen Geräts zu verwenden, um nicht viel Geld für spezielle Hochgeschwindigkeitskameras ausgeben zu müssen, die in anderen Bildverarbeitungsprojekten verwendet werden. Insbesondere die Kamera dieses Geräts ermöglicht die Produktion von Videos in hoher Qualität und die Möglichkeit, eine hohe Anzahl von Bildern pro Sekunde zu wählen, die eine bessere Analyse durch mehr Bilder pro Sekunde ermöglichen.
- ~~Bild~~ und Programmiersprache: OpenCV wurde unter den oben genannten ~~Bild~~ ausgewählt, weil es sehr leistungsfähig ist und es eine große Anzahl von Benutzern und Informationen darüber gibt. Gleichzeitig wurde versucht, das im Fachgebiet Künstliches Sehen erworbene Wissen in irgendeiner Weise anzuwenden und fortzuführen.

Obwohl es Versionen von OpenCV für die Programmierung in C++ und Java gibt, wird in dieser Arbeit Python verwendet werden. Die Lernkurve dieser Sprache ist niedriger, da es sich um eine Sprache handelt, die diesen Zweck erfüllt und mit der man sich während des Kurses beschäftigt hat. Python leidet manchmal unter Geschwindigkeitsproblemen, wurde aber bevorzugt, da es eine leichter zu erlernende Sprache ist.

6. Entwurf und Lösung der durchgeführten Arbeiten .

In diesem Abschnitt werden wir den Design- und Implementierungsteil des Projekts entwickeln. Wie wir bereits erwähnt haben, ist diese Arbeit auf einige Zwischenziele ausgerichtet, die weiter unten entwickelt werden.

Diese Zwischenziele sind so miteinander verbunden, dass ihre Verwirklichung von den vorangegangenen Zielen abhängt. In der folgenden Abbildung sind die Schritte dargestellt, die für die Implementierung durchgeführt werden müssen:



6.1. Suche nach dem zu analysierenden System .

Bei der Durchführung eines Projekts zum künstlichen Sehen ist es sehr wichtig, zunächst das System zu analysieren, auf dem es ausgeführt werden soll. Da Kameras zur Aufzeichnung der auszuführenden Aktionen verwendet werden müssen, ist es notwendig, ihre Platzierung zu untersuchen, um die Informationen unter Berücksichtigung möglicher Einschränkungen optimal zu erfassen.

Dieses Projekt befasst sich mit der Vorhersage von ~~der~~ Flugbahnen. Diese Art von System wurde gewählt, weil es in Sportanwendungen eingesetzt werden kann, z. B. bei der Analyse von Würfeln im Basketball, Fußball, Volleyball, Kugelstoßen, Speerwurf oder Hammerwurf in anderen Sportarten, in denen ebenfalls Würfe durchgeführt werden, die der Flugbahn der Kugel entsprechen.

"Die ~~der~~ Flugbahn ist die Flugbahn eines Geschosses, das nur seiner eigenen Trägheit und den Kräften des Mediums unterliegt, in dem es sich bewegt, hauptsächlich der Gravitationskraft.

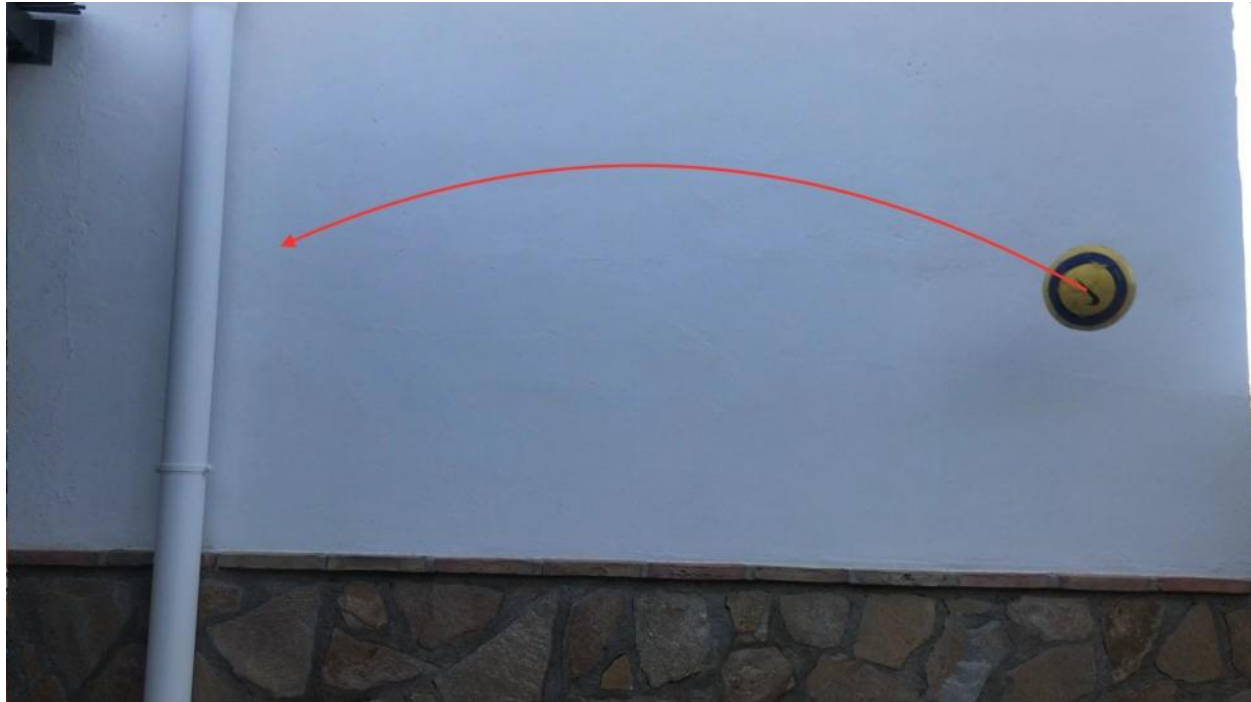
Wenn nur die Schwerkraft auf das Projektil ~~w~~ ist die ~~der~~ Flugbahn eine Kugel. Durch das Vorhandensein anderer Kräfte wie aerodynamischer (atmosphärischer) Luftwiderstand, Auftriebskraft, Corioliskraft (Wirkung der Erdrotation) usw. weicht die tatsächliche Flugbahn jedoch etwas von einer Parabel ab. " [7]

In dieser Arbeit lohnt es sich nicht, ~~nur~~ die Schwerkraft zu berücksichtigen, da andere Kräfte (Reibung usw.) als kleine Störungen behandelt werden können.

Sobald das zu lösende Problem ausgewählt ist, müssen wir die Proben für das Experiment

nehmen.

In unserem Fall handelt es sich um Videos von ~~Flug~~ Es wurden Videos von Kugeln ausgewählt, die sich in einer Ebene senkrecht zur optischen Achse der Kamera bewegen. Diese Einschränkung bei der Aufnahme von Videos ermöglicht die Anwendung von 2D-Vorhersagetechniken, so dass wir nur eine Kamera und ein Video benötigen, um das Ziel zu erreichen. Dies war eines der Ziele dieser Arbeit, nämlich zu versuchen, die Flugbahn nur mit Hilfe einer Kamera vorherzusagen. Für die Verwendung herkömmlicher geometrischer 3D-Vision-Techniken wären mehrere Ansichten erforderlich.



Ein weiteres wichtiges Detail ist die Frage, wie viele Daten benötigt werden, um die Berechnungen korrekt durchführen zu können. Bei einer Anwendung des künstlichen Sehens, bei der wir Videos verwenden, werden die Bilder einzeln analysiert.

Bei einer normalen Videoaufnahme werden 24 bis 30 Bilder pro Sekunde aufgenommen. Die leistungsfähigeren Kameras, die wir ~~haben~~ haben, können Videos mit 60 Bildern pro Sekunde aufzeichnen, was uns bei unserer Analyse sehr ~~helfen~~ hilft. Da die Flugbahn eines Abschusses von der Geschwindigkeit abhängt, mit der er erfolgt, ~~kann~~ vorkommen, dass er aus der Kabinenhaube herausfliegt oder auf dem Boden aufschlägt, wo er abprallt und seine Flugbahn ~~ändert~~ ändert. In diesen Fällen, in denen die Datenerfassungszeit kürzer ist, können wir umso mehr Bilder pro Sekunde analysieren, ~~je~~ für dem besten Ergebnis kommen.

Um die Auswirkungen der Bilder pro Sekunde, mit denen das Video aufgezeichnet wird, zu vergleichen, werden wir verschiedene Videos mit 60 und 120 Bildern pro Sekunde aufnehmen, da das Video mit 60 und 120 Bildern pro Sekunde aufgezeichnet wird.

Das Gerät, das wir verwenden werden, bietet uns diese Möglichkeit.

Ein weiterer wichtiger Aspekt bei der Erstellung von Videos ist der Abstand zum zu filmenden Objekt.

Je weiter entfernt wir analysieren wollen, desto weniger Pixel werden in den Bildern erfasst und desto mehr ~~Bild~~ Bildern erfasst.

Das erste Bild ist für die Verfolgung des Balls komplizierter als das zweite, da es kleiner ist, aber der Ball wird die Segmentierung und Verfolgung durchführen, die wir in den nächsten Abschnitten sehen werden. In der folgenden Abbildung haben wir einen Vergleich, wie sich dies auf die oben genannten Punkte auswirkt. Das erste Bild ist komplizierter, den Ball zu verfolgen als das zweite, da es kleiner ist, aber gleichzeitig hat der Ball einen längeren Flug und wir können die Flugbahn besser vorhersagen.



Die Suche nach dem optimalen Gleichgewicht in Abhängigkeit von unserer Anwendung wäre ein wichtiger Faktor, der vor Beginn der Implementierung berücksichtigt werden sollte.

6.2. Bildsegmentierung .

Sobald wir das System und das zu analysierende Material haben, beginnen wir mit der Segmentierungsphase. In unserem Fall wird der Segmentierungsprozess an den Videos durchgeführt, die mit dem Ziel aufgenommen wurden, den zu verfolgenden Ball zu lokalisieren.

"Das Ziel der Segmentierung ist es, die Darstellung eines Bildes zu vereinfachen und/oder zu verändern, so dass es aussagekräftiger und leichter zu analysieren ist. Die Segmentierung wird sowohl zur Lokalisierung von Objekten als auch zur Ermittlung ihrer ~~Gm~~ innerhalb eines Bildes verwendet. Genauer gesagt ist die Bildsegmentierung ein Prozess, bei dem jedem ~~Bild~~ Bildes ein Etikett zugewiesen wird, so dass ~~Es~~ die dasselbe Etikett haben, auch bestimmte ähnliche visuelle Merkmale aufweisen. [8]

Obwohl es in der Bildverarbeitung zahlreiche Segmentierungstechniken gibt, konzentrieren wir uns in diesem Abschnitt auf die Kommentierung und Erläuterung der für dieses Projekt verwendeten Techniken.

Um das Objekt zu verfolgen, werden wir ein Farbmodell dieses Objekts verwenden. In diesem Abschnitt müssen wir Segmentierungstechniken anwenden, um die Erstellung dieses Farbmodells zu erleichtern. Dazu müssen wir das sich bewegende Objekt in irgendeiner Weise isolieren, damit andere Objekte oder Szenarien nicht stören.

Als ein Objekt, das sich zwischen Bildern bewegt¹ können wir diesen Unterschied zwischen einem Bild und einem Objekt, das sich zwischen Bildern bewegt, nutzen.

Auf diese Weise können wir statische Objekte oder Hintergründe ausschließen, indem wir das gesuchte Objekt eingrenzen. Auf diese Weise können wir statische Objekte oder Hintergründe ausschließen, indem wir das gesuchte Objekt eingrenzen. [Beispiel im Video](#).

Die für diese Entfernung verwendete Technik ist BackgroundSubtractorMOG2, die in der OpenCV-Bibliothek enthalten ist. Diese Hintergrundentferner sind ein wichtiger Bestandteil vieler Computer-Vision-Anwendungen.

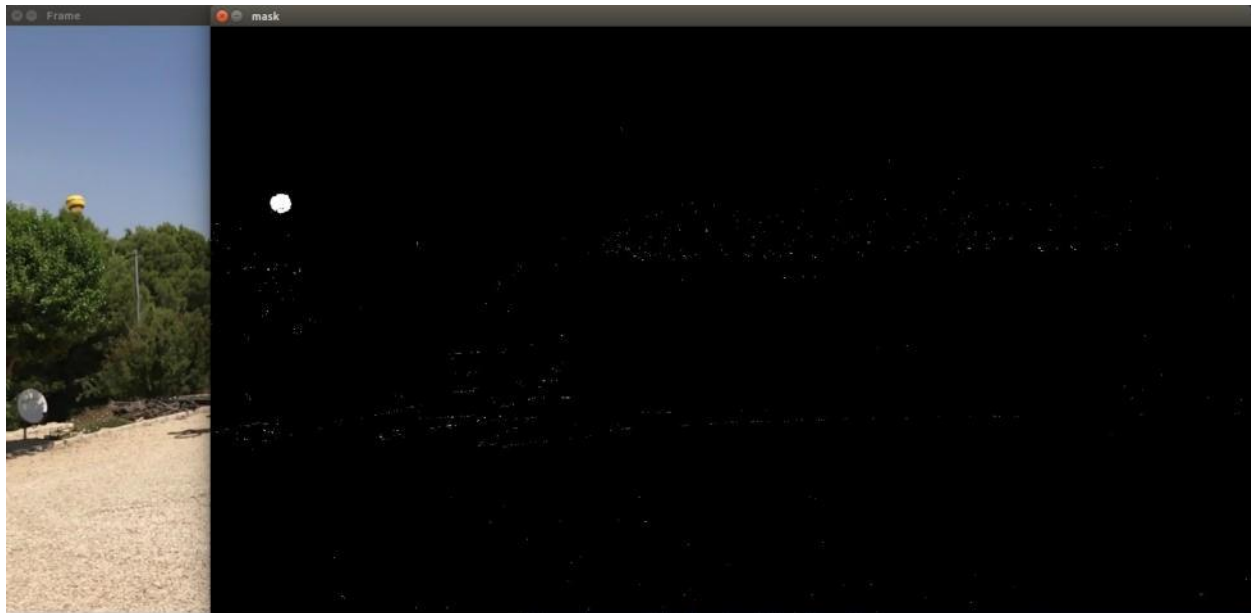
Das Beispiel, das immer wieder zur Erläuterung herangezogen wird, ist eine Anwendung, die mit Hilfe einer statischen Kamera Besucher zählt, die einen Ort betreten, oder Fahrzeuge, die eine Straße entlangfahren. Zur Durchführung der Zählung ist es zunächst erforderlich, den Besucher oder das Fahrzeug, die die beweglichen Elemente darstellen, vom statischen Hintergrund zu isolieren. Wenn wir einen einheitlichen Hintergrund hätten, wäre es sehr einfach, ihn zu entfernen, aber in realen Anwendungen sind Hintergründe nicht so. Diese Techniken sind dafür zuständig, die Unterschiede und damit die Objekte zu finden, die sich bewegt haben, indem ein Bild vom vorherigen abgezogen wird. [9][10]

Ein Problem, das auftreten kann, ist die Behandlung von Objektschatten. Der Schatten eines Objekts bewegt sich mit ihm und tritt daher in unserer Anwendung als falsches Positiv erscheinen.

Die Verwendung von BackgroundSubtractorMOG2 führt zu einer besseren Anpassung an Lichtveränderungen als BackgroundSubtractorMOG, und es enthält auch eine Option zur Schattenerkennung, die zwar die Verarbeitungsgeschwindigkeit verlangsamt, aber für unsere Anwendung sehr nützlich ist. Der Unterschied zum MOG-Algorithmus besteht darin, dass er ein Verfahren verwendet, bei dem jedes Hintergrundpixel durch eine Mischung von Gaußschen K-Verteilungen ($K = 3$ bis 5) modelliert wird. Die Mischungsgewichte stellen die Zeitanteile dar, die diese Farben in der Szene verbleiben. Die wahrscheinlichen Hintergrundfarben sind diejenigen, die am längsten und am statischsten bleiben. MOG2 hingegen wählt für jedes Pixel die entsprechende Anzahl von Gaußverteilungen aus.

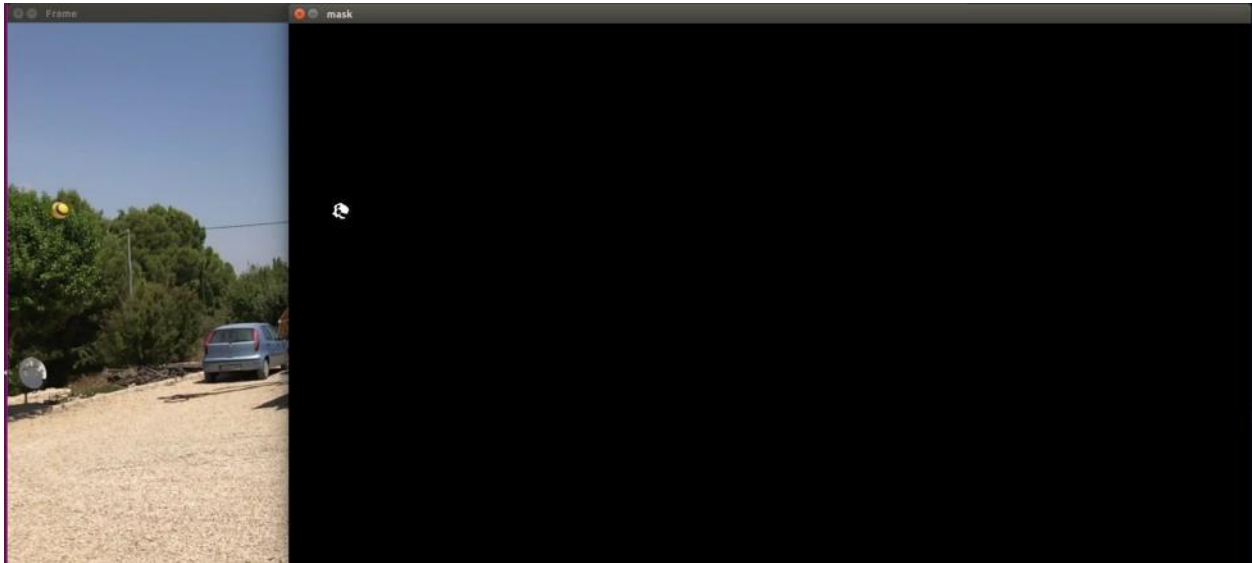
In der folgenden Abbildung sehen wir, wie die Verwendung dieser Technik in unserer Anwendung aussieht.

¹Ein Einzelbild ist jedes der Bilder, aus denen ein Video besteht.

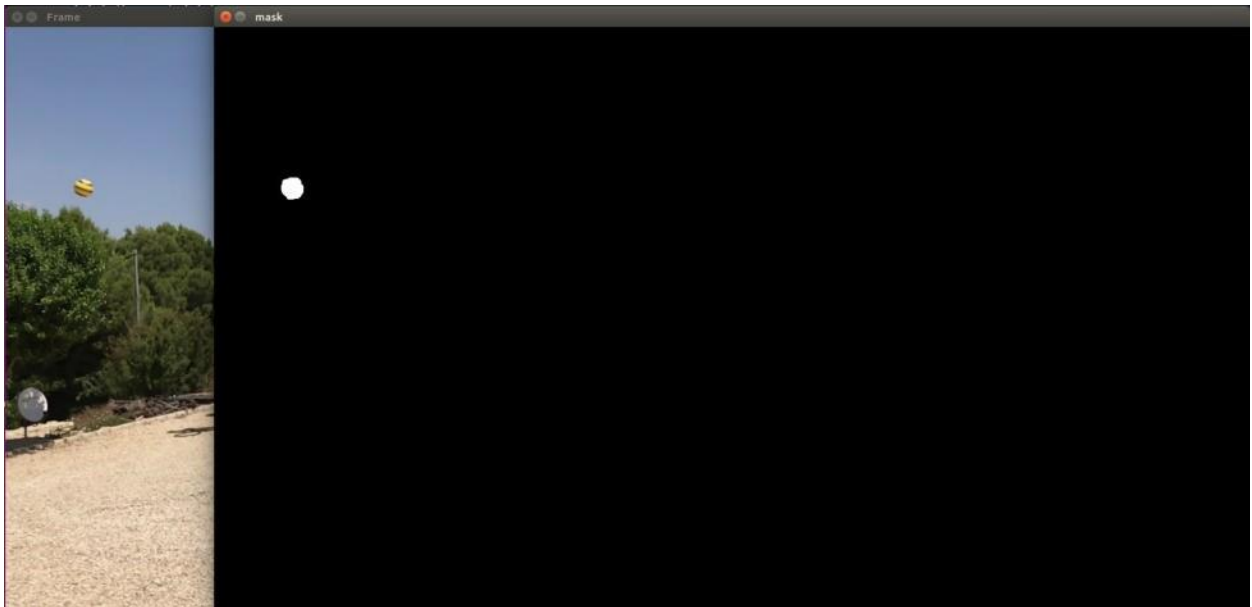


Es ist uns gelungen, den Ball so zu isolieren, wie wir gesucht haben, aber wir haben auch eine Reihe von Punkten, die auf kleine Abweichungen oder Rauschen zwischen den Bildern zurückzuführen sind. Diese kleinen Schwankungen können zum Beispiel durch den Wind verursacht werden, der die Blätter bewegt, und der Algorithmus erkennt dies. Der nächste Schritt besteht darin, diese Variationen so weit wie möglich zu eliminieren, so dass der Ball das wichtigste Objekt ist. Dazu werden wir drei Schritte durchführen und drei OpenCV-Funktionen verwenden:

- Aushöhlen: Entfernt die äußeren Pixel von Objekten. [11][12]
- MedianBlur: Nimmt eine Reihe von Pixeln und mittelt und ersetzt sie. Dies ist nützlich, um weiße Punkte in schwarzen Bereichen zu entfernen. [13]
- Dilatieren: Im Gegensatz zum Erodieren werden die Ränder der Kugel vergrößert, entweder weil sie bei den anderen beiden Techniken verloren gegangen sind oder weil wir diesen Teil größer machen wollen. [12] Die Kugel wird größer.



Erode.

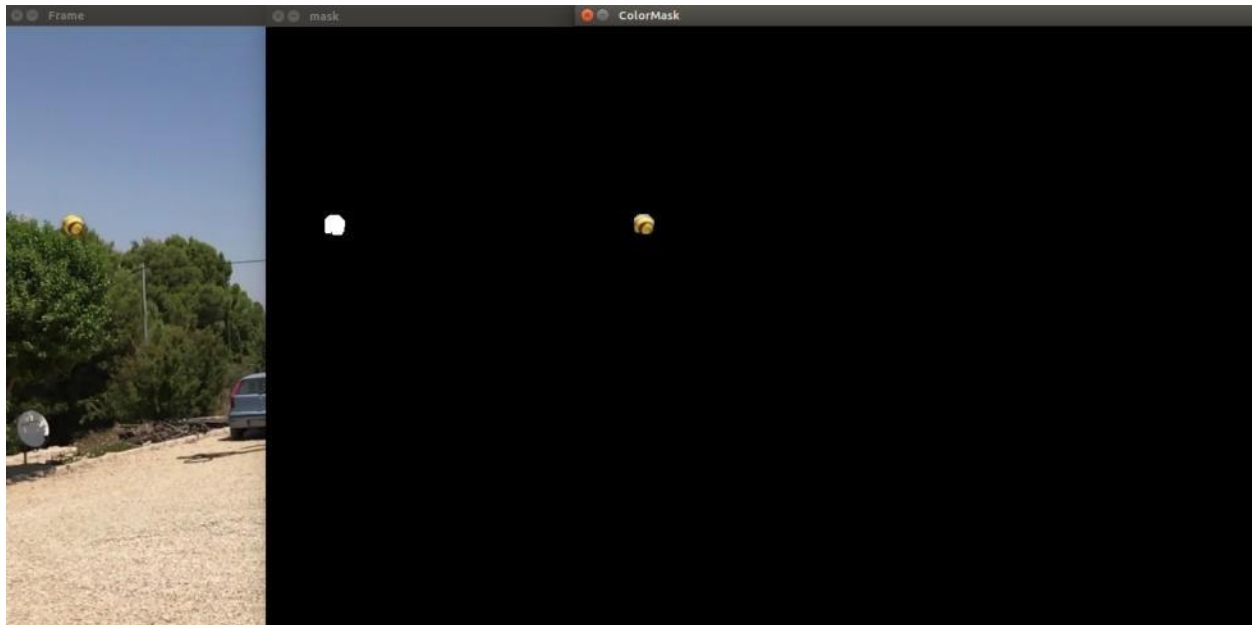


Median + Dilatation.

Der Unterschied ist nach der Durchführung dieser Schritte spürbar. Schließlich wandeln wir das vorherige Ergebnis in eine Maske um (eine Maske ist ein Bild, bei dem der Wert der Pixel ein wahrer oder falscher Wert ist).

Wendet man diese Maske auf das Originalbild an, werden die Bereiche, die vorher weiß waren in der Maske einen echten Wert haben. Das Ergebnis

ist der farbige Ball zusammen mit dem schwarzen Hintergrund. Das Ergebnis können Sie auch in diesem [Beispielvideo](#) sehen.



Original, Mascara, Anwendung des Mascara auf das Original

Wir haben es geschafft, das Objekt, das wir für unsere Anwendung benötigen, auf vernünftige Weise zu isolieren, wie wir es in diesem Abschnitt [von](#)

6.3. T Regale.

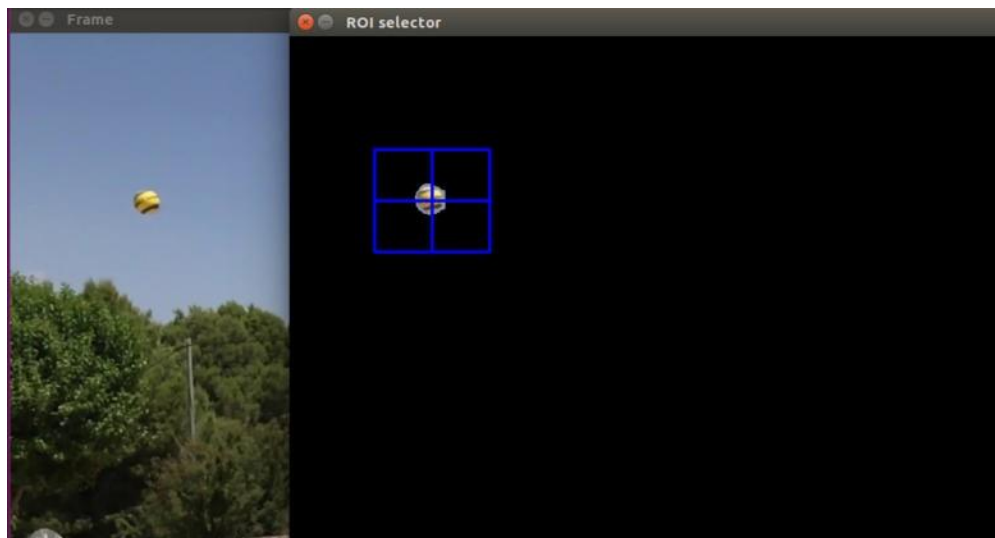
Angenommen, wir wollen die Position eines sich bewegenden Objekts in einer Bildfolge erkennen. Wenn das Objekt mehr oder weniger kompakt ist und eine ~~alle~~ Farbe hat, können wir eine ähnliche Technik anwenden. Aus dem Likelihood-Bild des Modells (der Reprojektion des Histogramms) können wir die Ellipse der Ungewissheit erhalten, die den größten Teil des Objekts umfasst, und mit ihr eine Region von Interesse mit der geeigneten Größe definieren, mit der wir im nächsten Bild arbeiten können.

Dies ist die Grundlage für die **Mean-Shift- und Camshift-Techniken**. Der Unterschied zwischen ihnen besteht darin, dass Mean Shift die Suche immer mit der gleichen Größe durchführt und Camshift (Continuously Adaptive Mean Shift) die Größe des Suchfensters bei jeder Ausführung variiert. Es kann vorkommen, dass sich ein Objekt näher an die Kamera heran- oder von ihr wegbewegt und dadurch die Anzahl der ~~Es~~ größer oder kleiner wird. In ~~der~~ Fall funktioniert Camshift besser als Mean Shift.

Aus diesem Grund wurde Camshift[14][15] gewählt, um unser Objekt zu verfolgen.

Um die Verfolgung durchzuführen, müssen wir zunächst das Farbmodell des Objekts erstellen, dem die Camshift-Funktion folgen soll. Dieses Modell übergeben wir der Funktion zusammen mit der Position des Bildes, an der sie mit der Suche beginnen soll.

Das Modell sind in unserem Fall die Farbhistogrammwerte des Balls. Durch die Auswahl der interessierenden Region und die Verwendung der OpenCV-Funktionen zur Berechnung von Histogrammen erstellen wir das Modell.



Nach der Erstellung des Modells und mit der durch die Auswahl der Region erhaltenen Position müssen wir

verwenden Sie die Funktion mit den erforderlichen Parametern. Da die Funktion nach dem Bereich mit der höchsten Pixeldichte für das erstellte Histogramm sucht, ist es notwendig, die Reprojektion des Histogramms zu erstellen. Diese Reprojektion zusammen mit der anfänglichen Suchposition und einem Abbruchkriterium (Anzahl der Iterationen oder Bewegung des Objekts) sind die Parameter, die die Funktion übernimmt. Wenn sich das Objekt bewegt, bewegt sich auch die Reprojektion des Histogramms mit, was den Algorithmus dazu veranlasst, das Suchfenster zur nächsthöheren Dichteposition zu verschieben (siehe folgendes Bild).



Das Ergebnis der Ausführung gibt das neue Fenster, in dem gesucht werden soll, zusammen mit dem Ergebnis der Suche zurück. In Grün der Bereich, den die Funktion erkannt hat, und in Blau der Bereich, in dem sie bei der nächsten Iteration weiter suchen wird.



Schließlich wollen wir mit dem Tracking eine Reihe von Daten gewinnen, die für den letzten Zweck, die Vorhersage, verwendet werden können. Dank Camshift können wir die Positionen, in denen sich unser Objekt in jedem Frame befindet, speichern. Diese Positionen dienen uns zur Berechnung der geschätzten Position zu einem späteren Zeitpunkt, indem wir das mathematische Modell anwenden, das sich in jedem Fall auswirkt, im Fall dieser Arbeit das ~~1a~~ Modell.

— Je mehr Daten uns für die Vorhersage zur Verfügung stehen, desto zuverlässiger wird die

Vorhersage sein. Hier kommt der von uns verwendete Aufnahmemodus ins Spiel, insbesondere die verwendeten Bilder.

pro Sekunde zu denen, die wir aufzeichnen.

In den folgenden Bildern sehen wir den Vergleich zwischen einem Video, das mit 60 Bildern pro Sekunde aufgenommen wurde, und einem anderen mit 120 Bildern pro Sekunde (wir können uns auch vorstellen, wie es in Videos aussehen würde, die mit 30 oder 240 Bildern pro Sekunde aufgenommen wurden).



Videoaufnahme mit 60fps



Videoaufzeichnung mit 120fps

~~Durch die doppelte Anzahl von Bildern pro Sekunde können wir doppelt so viele Messungen~~

vornehmen.

Dies ist eine wichtige Tatsache, die bei einem solchen Antrag berücksichtigt werden muss. Zum Abschluss dieses Abschnitts noch ein Beispiel, bei dem das Objekt aus der Szene verschwindet und wieder auftaucht.



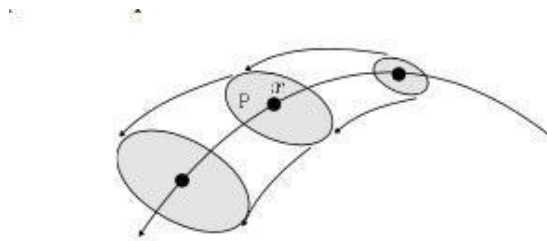
Ein vollständiges Video der Aufführung finden Sie unter [folgendem Link](#) für weitere Einzelheiten.

6.4. Vorhersagen.

Sobald es möglich ist, das Ziel zu isolieren, es zu verfolgen und Daten über seine Position zu erhalten, können wir diese verwenden, um eine Vorhersage über einen zukünftigen Zustand zu machen. Bei der Suche nach einem Prädiktor müssen wir die Art des Systems, an dem wir arbeiten, berücksichtigen. Das System dieser Arbeit ist die Vorhersage von ~~der~~ Flugbahnen, also ein lineares dynamisches System, das sich nach einem dynamischen Gesetz entwickelt. Bei dieser Art von System kann der Kalman-Filter verwendet werden [16].

"Der Kalman-Filter ist ein 1960 von Rudolf E. Kalman entwickelter Algorithmus zur Ermittlung des verborgenen (nicht messbaren) Zustands eines linearen dynamischen Systems, der auch dann nützlich ist, wenn das System additivem Rauschen ausgesetzt ist. Da der Kalman-Filter ein rekursiver Algorithmus ist, kann er in Echtzeit ausgeführt werden, wobei nur die aktuellen Eingangsmessungen, der zuvor berechnete Zustand und seine Unsicherheitsmatrix verwendet werden und keine zusätzlichen Informationen erforderlich sind. Der Kalman-Filter hat zahlreiche Anwendungen in der ~~Technik~~ Eine häufige Anwendung ist die ~~Flug~~ Navigation und Steuerung von ~~Flugzeugen~~ insbesondere von Raumfahrzeugen. Darüber hinaus wird der Filter häufig in Bereichen wie ~~Industrie~~ und ~~Öl~~ eingesetzt." [17]

Wir haben also ein System, das sich nach einem dynamischen Gesetz entwickelt $x_{t+1} = F(x_t)$. Unsere Informationen über den Zustand sind mit Unsicherheit behaftet, die mit der Zeit zunimmt:



Angenommen, wir haben eine Eigenschaft $z = H(x)$, die wir beobachten können. Im Idealfall sollten wir in der Lage sein, diesen Zustand zu messen. In diesem Fall wäre $z = x$ und das Problem wäre trivial. Da es sich um ein Vorhersageproblem handelt, ist es normal, dass wir nicht immer über diese Informationen verfügen und außerdem Rauschen auftreten kann. Dieses z wird eine Eigenschaft sein, die von einigen Komponenten von x abhängt. Wenn wir zum Beispiel die Flugbahn eines sich bewegenden Objekts mit Hilfe einer Kamera schätzen wollen, enthält der Zustand Positions- und Geschwindigkeitskoordinaten, aber die Beobachtung zeichnet nur Positionsinformationen auf (wie in unserem Fall).

Obwohl das Problem aufgrund des Mangels an Informationen unlösbar scheint, ~~nur~~ wir eine Art inverses Modell von H erstellen, das den Zustand aus Teilinformationen rekonstruiert. Es ist unmöglich, ein solches inverses Modell zu erhalten, aber mit Hilfe von Beobachtungsinformationen können wir die Unsicherheit über den vollständigen Zustand verringern. In der Praxis werden wir über eine Reihe von Beobachtungen z_t verfügen, die während der gesamten Entwicklung des Systems gemacht werden und mit denen wir eine Reihe von Vorhersage-Korrekturschritten durchführen, um den Zustand x_t anzupassen. Die Verringerung der Unsicherheit (bei unvollständiger Information) erfolgt, weil das dynamische Evolutionsgesetz eine Abhängigkeit zwischen den verschiedenen Zustandsvariablen, die durch Beobachtung gewonnen werden, und den latenten Variablen herstellt.

In unserem Beispiel von Position und Geschwindigkeit erhält die Beobachtung nur die Position zu einem bestimmten Zeitpunkt, aber die Geschwindigkeit kann durch die Abhängigkeit der Position von früheren Positionen und Geschwindigkeiten geschätzt werden. Es ist interessant zu sehen, wie diese Geschwindigkeitsschätzung automatisch ermittelt wird, ohne dass man sie ~~g~~konstruieren muss.

Im einfachsten Fall, wenn sowohl das dynamische Gesetz F als auch das Beobachtungsmodell H lineare Funktionen sind und die Unsicherheit über den Zustand, das Systemrauschen und das Beobachtungsrauschen additiv und gaußförmig sind, kann der Kalman-Filter-Algorithmus angewendet werden.

Der Algorithmus sieht folgendermaßen aus: Bei gegebenen Transformationen F und H , bei denen das Rauschen durch die Kovarianzmatrizen S bzw. R modelliert wird, wird die Information über x in Form von Mittelwert μ und Kovarianzmatrix P mit einer neuen Beobachtung z wie folgt aktualisiert:

- $\mu' \leftarrow F\mu$
- $P' \leftarrow FPF^T + S$
(Entwicklung des Staates)
- $\varrho \leftarrow z - H\mu'$
(Diskrepanz zwischen der Beobachtung und ihrer Vorhersage)
- $K \leftarrow P'H^T(HP'H^T + R)^{-1}$
(Kalman-Verstärkung)
- $\mu \leftarrow \mu' + K\varrho$

■ $P \leftarrow (I - KH)P'$ (Berichtigung)

Nachdem wir das theoretische Modell des Kalman-Filters erläutert haben, werden wir es in unserer Arbeit in die Praxis umsetzen. In diesem Fall werden wir versuchen, anhand der früheren Messungen die Position zu späteren Zeitpunkten zusammen mit der Unsicherheit vorherzusagen. Die zu verwendende Funktion ist diejenige, die von den Professoren der Fakultät innerhalb des umucv-Funktionspakets für OpenCV entwickelt wurde und die unter dem folgenden [Link](#) zu finden ist.

```

1 numpy als np importieren
  numpy.linalg as la
  importieren

def kalman(mu,P,F,Q,B,u,z,H,R):
5     # mu, P : aktueller Zustand und seine
      Unsicherheit. # F, Q: dynamisches System
      und sein Rauschen.
      # B, u : Kontrollmodell und Eingabe.
      #      z:Beobachtung.
      # H, R : Beobachtungsmodell und sein
      Rauschen
      mup = F @ mu + B @ u      #Vorausgesagter Zustand ohne
      pp = F @ P @ F.T + Q;      Beobachtung.
                                  #Ungewissheit, wenn es keine Beobachtung
                                  gibt.
      zp = H @ mup              #Vorhersage über das Modell.

      # Wenn es keine Beobachtungen gibt, machen wir
      nur Vorhersagen.
      wenn z keine ist:
          Rückgabe mup, pp, zp

      epsilon = z - zp          #Diskrepanz zwischen der Beobachtung
                                und ihrer
                                Vorhersage.

21     k = pp @ H.T @ la.inv(H @ pp @ H.T +R) #Kalman Gain.

      new_mu = mup + k @ epsilon; #Neuer aktueller Zustand.
25     new_P = (np.eye(len(P))-k @ H) @ pp; #Neue Unsicherheit.
      return new_mu, new_P, zp

```

Die 2D-Bewegungsgleichungen lauten wie folgt:

$$\mathbf{v}' \leftarrow \mathbf{v} + \mathbf{a} \Delta t$$

$$v'_{yy} \leftarrow v_{yy} + a_y \Delta t$$

$$x' \leftarrow x + v_x \Delta t + \frac{1}{2} a_x (\Delta t)^2$$

$$y' \leftarrow y + v_y \Delta t + \frac{1}{2} a_y (\Delta t)^2$$

wobei x', y', v'_x, v'_{xy} stellen die Zustandsvariablen zum nächsten Zeitpunkt dar.

Die geeignete Matrixform für die Entwicklung des allgemeinen Kalman-Filters lautet wie folgt:

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Die Matrix F implementiert den Teil des dynamischen Gesetzes, in dem die Beschleunigungsterme nicht vorkommen. Diese Begriffe werden mit der Matrix B, der Einwirkung auf das System, \mathbf{u} Es ist nicht möglich, alles in einer einzigen Matrix zusammenzufassen, da die Aktionsmatrix ein additiver Term ist (sie multipliziert kein Element des Zustands) und dies bedeutet ~~wie~~ dass die Dynamik des Systems nicht rein linear wäre.

$$B = \begin{bmatrix} \Delta t^2 / 2 & 0 \\ 0 & \Delta t^2 / 2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

H ist die Matrixform, die ausdrückt, dass wir die Position (aber nicht die Geschwindigkeit) beobachten.

wobei γ der Pixel-Meter-Umrechnungsfaktor für den Objektabstand ist.

Nun ist es an der Zeit, die Segmentierungs- und Tracking-Teile mit der Filterimplementierung zu verbinden. In jedem Rahmen, in dem es möglich ist, einen Beobachtungspunkt durch **Camshift** zu erhalten, wird dieser an den Kalman-Filter-Algorithmus weitergeleitet, um seinen Zustand unter Unsicherheit zu aktualisieren.

Liegt keine Beobachtung vor, erzeugt der Filter einen neuen geschätzten Zustand bei aus den vorherigen Daten. Im Gegenzug, was durchgeführt werden wird, ist eine Vorhersage von Staaten nach dem aktuellen, werden wir sehen, dass wir eine geschätzte Position und eine Unsicherheit zu erhalten. Je weiter wir vom aktuellen Zustand entfernt sind, desto größer ist die Unsicherheit, und wenn wir eine größere Anzahl von Beobachtungen haben, ist die Unsicherheit kleiner.

Wenn das Objekt aus der Szene verschwindet, wird die aktuelle Vorhersage beibehalten, bis eine neue Beobachtung gemacht wird, die sich auf die Vorhersage auswirken kann.

Wir werden uns eine Reihe von Arbeitsbeispielen ansehen:



Im ersten Bild ist die Unsicherheit der Vorhersage sehr groß, da es nur wenige Messungen gibt, aber sie ist nicht sehr konkret.



In der zweiten Phase, mit mehr Messungen, ist es in der Lage, die Position des Objekts in der Zukunft zu schätzen.



Auf dem letzten Bild sehen wir, dass die Vorhersage richtig war und das Objekt der vorhergesagten Flugbahn folgte.

Wir haben die gleichen Schritte durchgeführt, aber jetzt mit einem Video, in dem der Ball verloren geht, um zu sehen, ob er in der Lage ist, richtig vorherzusagen:



Wir sehen, dass die Flugbahn weitgehend mit der Vorhersage übereinstimmt. Wir haben auch die Erkennung von Abprallern und die Berechnung der neuen Flugbahn implementiert, wie auf dem letzten Bild zu sehen ist. [Link zur Videodemonstration.](#)

Das Ziel dieses Abschnitts, die vorangegangenen Teile miteinander zu verbinden und die Vorhersage zu erreichen, wurde erfolgreich verfolgt. Der nächste und letzte Schritt besteht darin, die Implementierung in eine sportbezogene Anwendung zu bringen.

Wenn wir die 3D-Trajektorienberechnung als nichtlineares Beobachtungsmodell durchführen wollten, ~~keine~~ eine Art Filter namens **UKF**[16] verwendet werden; dies bleibt jedoch eine mögliche zukünftige Aufgabe.

6.5. System in der Umsetzung n.

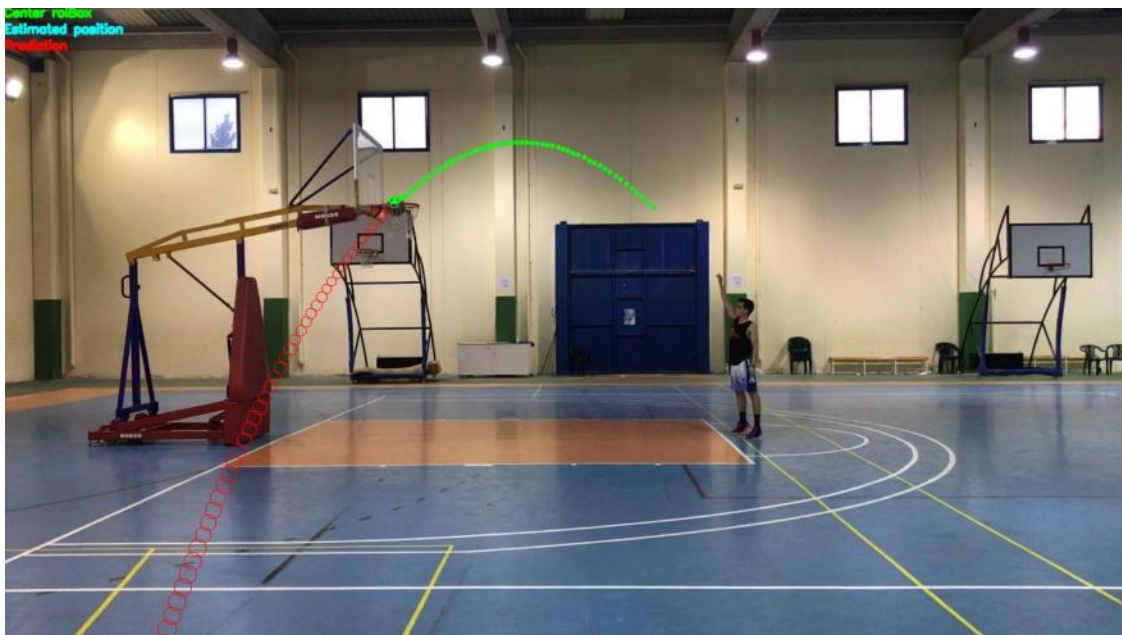
Um das letzte Ziel, die Entwicklung einer sportorientierten Anwendung, zu erreichen, haben wir beschlossen, die vorangegangenen Schritte für die Erstellung eines Basketball-Trajektorie-Vorhersageprogramms zu verwenden.

Da wir bereits die zu verwendenden Techniken gesehen haben, werden wir nun das System in der Ausführung zeigen:

Am Anfang ist die Ungewissheit noch recht groß, aber allmählich wird klar, wohin der Ball fliegen wird.



Wir sehen, dass der Ball am Ende die vorhergesagte Flugbahn einhält.





Werfen an die Tafel.



Pitching out.

Wie in den anderen Abschnitten ~~keine~~ **Video-Demonstration** der verschiedenen Starts ~~h~~
Das Ziel, eine auf den Sport ausgerichtete Anwendung des künstlichen Sehens zu schaffen, ist erreicht und damit das Ende der Arbeit.

7. Schlussfolgerungen und künftige Arbeit

7.1. Conclusions.

In der Analysephase wurden die Ziele für die Arbeit festgelegt. Das erste Ziel lag in der Untersuchung und Anwendung von Techniken des maschinellen Sehens und der Bildanalyse. Dieses Ziel wurde in den Bereichen Segmentierung und Verfolgung zufriedenstellend erfüllt, obwohl, wie bereits erwähnt, jede Anwendung und jede Szene anders sein kann und dieselben Techniken entweder auf andere Weise oder anders angewendet werden müssen. Bei einer Bildverarbeitungsanwendung muss man das Szenario sehr gut studieren, um ein gutes Ergebnis zu erzielen. Aber im Großen und Ganzen sind wir mit dem in diesem Teil erzielten Ergebnis recht zufrieden.

Das nächste Ziel war die Anwendung von Prognosetechniken. Wir haben gezeigt, dass durch die Verwendung eines Kalman-Filters eine Vorhersage zukünftiger Zustände mit einer gewissen Unsicherheit erreicht werden kann. Dies war ein schwieriger Teil der Arbeit, und wir können auch hier feststellen, dass er zufriedenstellend gelöst wurde.

Schließlich die Entwicklung und Anwendung auf eine reale Situation. Obwohl es sich um eine einfache Anwendung handelt, gibt sie uns ein Bild von der Art der Anwendungen, die möglich sind und daher betrachten wir das Ziel auch als erreicht.

Im Allgemeinen ist die Zufriedenheit mit der Arbeit recht hoch, da sie sich mit einem sehr interessanten Thema wie dem künstlichen Sehen befasst, ein echtes Problem behandelt und Kenntnisse vermittelt hat, die über die Kompetenzen des Abschlusses hinausgehen.

7.2. Mögliche künftige Arbeiten .

Für die künftige Arbeit haben wir verschiedene Aufgaben, die wir in Angriff nehmen können. Mit den Ergebnissen dieser Arbeit können wir sie auf andere ähnliche Systeme anwenden, z. B. auf verschiedene Arten von Würfeln, vom Speer- oder Diskuswurf bis hin zur Berechnung der Flugbahnen bei Würfeln im American Football, Rugby oder jeder anderen Sportart, bei der die Bedingungen ähnlich sind wie die, die in dieser Arbeit behandelt wurden.

Als Fortsetzung dieser Arbeit könnte man auch nach Möglichkeiten suchen, eine rechnerisch schnellere Analyse durch andere Techniken zu erreichen. Diese Arbeit konzentrierte sich auf die Videoanalyse, könnte aber auch für den Einsatz in Echtzeit angepasst werden.

Eine weitere Möglichkeit wäre, dieses Projekt unter dem Gesichtspunkt der 3D-Bahnberechnung fortzusetzen, entweder durch die Verwendung verschiedener Ansichten und Rahmen oder durch die Anwendung einer speziellen Technik für nichtlineare Systeme.

Diese Arbeit kann auch als Ausgangspunkt für die Entwicklung anderer Anwendungen dienen, die sich nicht mit der Berechnung von Flugbahnen befassen, sondern mit Hilfe der Segmentierungs- und Tracking-Teile für die Sportanalyse genutzt werden können. Eine mögliche Anwendung könnte darin bestehen, die Position von Spielern auf einem Spielfeld zu verfolgen.

8. A nexus.

8.1. Code .

```
1 importiere cv2 als cv
2 import numpy as np
   from scipy.ndimage.filters import gaussian_filter
4 aus umucv.kalman importieren kalman, ukf
   umucv.htrans as ht importieren

   REDU = 8
8
   def rgbh(xs,mask):
10 def   normhist(x): return x / np.sum(x)

12 def   h(rgb):
       return cv.calcHist([rgb])
14                               , [0, 1, 2]
                               , imCropMask
16                               , [256//REDU, 256//REDU, 256//REDU,
256//REDU].
                               , [0, 256] + [0, 256] + [0, 256]
18                               )

20 return   normhist(sum(sum(map(h, xs))))

22 def smooth(s,x):
       return gaussian_filter(x,s,mode='constant')

       bgsub = cv.createBackgroundSubtractorMOG2(500, 60, True)
       #Schwellenwert kann variieren (60)
26 cap = cv.VideoCapture("Videos/11.MOV")
       key = 0

       Kernel = np.ones((3,3),np.uint8)
30 crop = Falsch
       camshift =
       Falsch

       Beendigung = (cv.TERM_CRITERIA_EPS | cv.TERM_CRITERIA_COUNT, 10, 1)
```



```
Schriftart = cv.FONT_HERSHEY_SIMPLEX

pause= Falsch
38 ##### Initial Kalman #####

40 # angeben, dass Kalman aktualisiert. Dies ist der Anfangswert
    degree = np.pi/180
42 a = np.array([0, 900])

    #fps = 60
46 fps = 120 dt
    = 1/fps
48 t = np.arange(0,2.01,dt)
    noise = 3

50
    F = np.array(
52     [1, 0, dt, 0,
        0, 1, 0, dt,

        0, 0, 1, 0,
560     , 0, 0, 0, 0, 1 ] ).reshape(4,4)

58 B = np.array(
        [dt**2/2, 0,
          600,dt**2/2,
            dt,0,
620            ,dt] ).reshape(4,2)

    H = np.array(
66     [1,0,0,0,
        0,1,0,0] ).reshape(2,4)

70

        # x, y, vx, vy
72 mu = np.array([0,0,0,0,0,0])
        # ihre Unsicherheiten
74 P = np.diag([1000,1000,1000,1000,1000])**2
```

```
#Res = [(mu,P,mu)]
76 res=[]
    N = 15 # um eine anfängliche Streckung vorzunehmen und zu sehen, was
        passiert, wenn die Beobachtung dann verloren geht
78
    sigmaM = 0,0001# Modellrauschen
80 sigmaZ = 3*Rauschen # sollte gleich dem durchschnittlichen Rauschen
        der Bildverarbeitung sein. 10 Pixel pje.

82 Q = sigmaM**2 * np.eye(4)
    R = sigmaZ**2 *
        np.eye(2)

    listCenterX=[]
86 listCenterY=[]
    listpoints=[]
88 while(True):
    Taste = cv.waitKey(1) & 0xFF
90 if key== ord("c"): crop = True
    if key== ord("p"): P = np.diag([100,100,100,100,100])**2
92 if Schlüssel==27: break
    if key==ord(" "): pause =not pause
94 if (pause): weiter

96 ret , frame = cap.read()
    #frame=cv.resize(frame,(800,600))
98 frame=cv .resize(frame,(1366,768))

100 bgs = bgsub.apply(frame)

102 bgs = cv.erode(bgs,kernel,iterations = 1)
    bgs = cv.medianBlur(bgs,3)
104 bgs = cv.dilate(bgs,kernel,iterations=2)

106 bgs = (bgs > 200).astype(np.uint8)*255
    colorMask = cv.bitwise_and(frame,frame,mask = bgs)
108

110 if (crop):
    fromCenter=
        False
```

112img

= colourMask

```
114     r = cv.selectROI(img, fromCenter)
    imCrop = img[int(int(r[1]):int(r[1]+r[3]),
    int(r[0]):int(r[0]+r[2])]
    Ernte = Falsch
    camshift = True
    imCropMask = cv.cvtColor(imCrop, cv.COLOR_BGR2GRAY)
    ret,imCropMask =
    cv.threshold(imCropMask,30,255,cv.THRESH_BINARY)
    his = smooth(1,rgbh([imCrop],imCropMask))

    roiBox = (int(r[0]), int(r[1]),int(r[2]),
    int(r[3])) cv.destroyWindow("ROI selector")

124 if(camshift):
```

```
cv.putText(frame, 'Center roiBox', (0,10), font,
0.5, (0,255,0), 2, cv
.LINE_AA)
cv.putText(frame, 'Geschätzte Position', (0,30),
font, 0.5, (255,255,0), 2, cv.LINE_AA)
cv.putText(frame, 'Vorhersage', (0,50), font, 0.5, (0,0,255), 2, cv.
.LINE_AA)

130
rgbr = np.floor_divide( colourMask ,
REDU) r,g,b = rgbr.transpose(2,0,1)
l =
134
sein[r,g,b]
maxl =
l.max()
136
aa=np.clip((1*l/maxl*255),0,255).astype(np.uint8)
#cv.imshow("Rückprojektion", cv.resize(aa, (400,250)))
#
Rückprojektion

140

# Camshift anwenden und die Punkte auf den Bildschirm
zeichnen (rb, roiBox) = cv.CamShift(l, roiBox,
termination) cv.ellipse(frame, rb, (0, 255, 0), 2)

#####Kalman-Filter#####
146

xo=int(roiBox[0]+roiBox[2]/2)
148
yo=int(roiBox[1]+roiBox[3]/2)
error=(roiBox[3])
#Berechnungszentrum des Roibix
```

```

# print(me, error)

150
    if (me < Fehler oder bgs.sum() < 50 ):
152mu        , P, pred = kalman(mu, P, F, Q, B, a, None, H, R)
            m = "None".

154mm = Falsch

156Ansonsten    :
            mu, P, pred = kalman(mu, P, F, Q, B, a, np.array([xo, yo]), H, R)
158m =          "normal"
            mm = True

160

    if (mm) :
162listCenterX    .append(xo)
                listCenterY.append(yo)

164

    listpoints.append((xo, yo, m))
166res    += [(mu, P)]

168##### Vorhersage ##### mu2
    = mu
170P2    = P res2
    = []

    for _ in range(fps*2):
174mu2        , P2, pred2 = kalman(mu2, P2, F, Q, B, a, None, H, R)
            res2 += [(mu2, P2)].

    xe = [mu[0] for mu, _ in res]
178xu    = [2*np.sqrt(P[0,0]) for _, P in res]
    ye = [mu[1] for mu, _ in res]
180yu    = [2*np.sqrt(P[1,1]) for _, P in res]

182

184xp = [mu2[0] für mu2, _ in res2]
    und yp = [mu2[1] für mu2, _ in
res2].

186xpu    = [2*np.sqrt(P[0,0]) for _, P in res2]
    ypu = [2*np.sqrt(P[1,1]) for _, P in
res2]

```

188

```
190     for n in range(len(listCenterX)): # Zentrum der Roibox
        cv.circle(frame, (int(listCenterX[n]), int(listCenterY[n])), 3,
            (0, 255, 0), -1)
192
        for n in [-1]: #range(len(xe)): # xe und ye geschätzt
194             #uncertainty = (xu[n]*yu[n])
            #cv.circle(frame, (int(xe[n]), int(ye[n])), int(uncertainty),
                (255, 255, 0), -1)
            Unsicherheit=(xu[n]+yu[n])/2
            cv.circle(frame, (int(xe[n]), int(ye[n])), int(Unsicherheit),
                (255, 255, 0), 1)
198
        for n in range(len(xp)): # x und y vorhergesagt
200             uncertaintyP=(xpu[n]+ypu[n])/2
            cv.circle(frame, (int(xp[n]), int(yp[n])), int(uncertaintyP),
                (0, 0, 255))
202
        print("Liste der Punkte")
204         for n in range(len(listpoints)):
            print(listenpunkte[n])

        if(len(listCenterY)>4):
208             if((listCenterY[-5] < listCenterY[-4]) und (listCenterY[-4] <
                listCenterY[-3]) und (listCenterY[-3] > listCenterY[-2])
                und
                (listCenterY[-2] > listCenterY[-1])):
                print("REBOTE")
210                 listCenterY=[]
                listCenterX=[]
212                 listpoints=[]
                res=[]
214                 mu = np.array([0,0,0,0,0,0])
                P = np.diag([100,100,100,100,100])**2
216

218 cv.imshow('FarbMaske', colourMask)
        #cv.imshow('FarbMaske', cv.resize(FarbMaske, (800, 600)))
220 cv.imshow('mask', bgs)
        #cv.imshow('Frame', cv.resize(frame, (800, 600)))
222 cv.imshow('Rahmen', Rahmen)
```

8.2. Links zu Videos.

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

<https://www.youtube.com/watch?v=Mg40mE0zwBA>

https://www.youtube.com/watch?v=65Tq_nlP788

https://www.youtube.com/watch?v=65Tq_nlP788

<https://www.youtube.com/watch?v=YlPOTxYvt6U>

<https://www.youtube.com/watch?v=YlPOTxYvt6U>

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

9. ~~Bibli.~~

Referenzen

- [1] Richard Szeliski (2010). Computer Vision: Algorithmen und Anwendungen. Springer-Verlag. ISBN 978-1848829343.
- [2] Alberto Ruiz, Anmerkungen Thema Visio'n visio'n,
<http://dis.um.es/profesores/alberto/material/percep.pdf>
- [3] Rafael C. Gonzalez, Richard E. Woods, **Digitale Bildverarbeitung.**
- [4] Dokumentation OpenCV, http://docs.opencv.org/3.3.0/d6/d00/tutorial_py_root.html
- [5] The British Machine Vision Association and Society for Pattern Recognition Abgerufen am 20. Februar 2017, <http://www.bmva.org/visionoverview>
- [6] Draw.io, Online-Diagramm-Generator, <https://www.draw.io>
- [7] Wikipedia, Trayectoria Balística

,[https://es.wikipedia.org/wiki/ Trayectoria_bal%C3%ADstica](https://es.wikipedia.org/wiki/Trayectoria_bal%C3%ADstica)

- [8] Wikipedia, Segmentierung, [https://es.wikipedia.org/wiki/Segmentierung_\(Bild_Verarbeitung\)](https://es.wikipedia.org/wiki/Segmentierung_(Bild_Verarbeitung))
- [9] OpenCV, Hintergrundsubtraktion, http://docs.opencv.org/master/db/d5c/tutorial_py_bg_subtraction.html#gsc.tab=0
- [10] Hanzi Wang und David Suter, A RE-EVALUATION OF MIXTURE-OF-GAUSSIAN BACKGROUND MODELING, Monash University, Victoria, Australien
- [11] OpenCV, Erode und Dilate, http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html
- [12] Jean Serra, Bildanalyse und mathematische Morphologie ISBN 0-12-637240-3 (1982)
- [13] OpenCV, MedianBlur, http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html
- [14] Wikipedia, Camshift, <https://fr.wikipedia.org/wiki/Camshift>
- [15] Bradski, G.R., Echtzeit-Gesichts- und Objektverfolgung als Bestandteil einer wahrnehmungsorientierten Benutzerschnittstelle,. Anwendungen der Computer Vision, 1998. WACV '98. Proceedings, Fourth IEEE Workshop on , vol., no., pp.214,219, 19-21 Oct 1998.
- [16] Kalman, R.E. (1960). ^{Ein} neuer Ansatz für lineare Filterung und Vorhersageprobleme"(PDF). Zeitschrift für Basic Engineering. 82 (1): 35-45. doi:10.1115/1.3662552
- [17] Wikipedia, Kalman-Filter, https://es.wikipedia.org/wiki/Filtro_de_Kalman
- [18] Julier, Simon J.; Uhlmann, Jeffrey K. (1997). . ^{Eine} neue Erweiterung des Kalman-Filters auf nichtlineare Systeme"(PDF). Int. Symp. Luft- und Raumfahrt/Verteidigung Sensing, Simul. and Controls. Signalverarbeitung, Sensorfusion und Zielerfassung VI. 3: 182. bibco- de:1997SPIE.3068..182J. doi:10.1117/12.280797.