FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Erkennung sich wiederholender Ereignisse in Langzeitvideoaufnahmen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Achim Sedelmaier Bakk.techn.
Matrikelnummer 0505921

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Univ.Prof. Dipl.-Ing. Dr.techn. Christian Breiteneder
Mitwirkung: Dipl.-Ing. Dr. Matthias Zeppelzauer

Wien, 07.03.2016

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Erklärung zur Verfassung der Arbeit

Achim Sedelmaier, Bakk.techn.
Krichbaumgasse 19/15, 1120 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 07.03.2016, Unterschrift: 

Abstract

Long-time video recordings, created by animal monitoring or surveillance, can easily span hundreds of hours and thus take up a long time in the post-processing. Often it is necessary to view the entire video to assure to gather all occurrences of interest. However, plenty of times the interest focuses only on specific events that occur repeatedly (e.g. typical actions of people or animals). In this case, viewing the entire video is very inefficient.

This problem can be reduced by presenting the relevant events in a condensed form. It is the intention in this thesis to automatically extract re-occurring events from long time video recordings. Therefore image processing algorithms and statistical algorithms are used to identify recurring events. Motion and visual appearance of objects are captured and described in the form of basic feature patterns. If a feature pattern is repeated at a later point in time, a repeated event is detected.

The processing occurs as follows: First the footage is scanned for moving objects by use of temporal and local image segmentation. At every video frame an attempt is being made to find these objects in the previous frame, to obtain a temporal tracking of the objects. After objects are tracked over time, motion features and color features are extracted from every object. As a result every object together with its motion pattern is described by a set of features. After the prior extracted features are clustered, an alphabet is generated to describe higher-level events related to the object in terms of strings. Similar events are detected by a further clustering, which proves the similarity of the strings. String matching is applied to detect repeated events.

We evaluate our method on long-time surveillance video recordings of animal enclosures. Our experiments show that re-occurring events can be detected robustly by the proposed method. This thesis describes our algorithm, the performed experiments, presents results and discusses open topics at the end.

Kurzfassung

Langzeitvideoaufnahmen, wie bei Tierbeobachtungen oder bei der Sicherheitsüberwachung, können über mehrere hundert Stunden lang sein und somit viel Zeit in der Nachbearbeitung in Anspruch nehmen. Oft ist es notwendig das gesamte Videomaterial zu sichten, um alle Vorkommnisse des Interesses zu erfassen. Des weiteren liegt das Interesse oft auf bestimmten sich wiederholenden Ereignissen (z.B. typische Handlungen von Personen oder Tieren). Eine vollständige Durchsicht ist in diesem Fall ineffizient.

Dieses Problem kann verringert werden, indem relevante Ereignisse in zusammengefasster Form dargestellt werden. Diese Arbeit beschäftigt sich mit der automatischen Extraktion sich wiederholender Ereignisse von Langzeitvideoaufnahmen. Hierbei kommen Algorithmen aus der Bildverarbeitung und der Statistik zum Einsatz, um diese zu identifizieren. Bewegung und Aussehen von Objekten werden erfasst und als elementare Merkmalsmuster beschrieben. Wiederholt sich ein Eigenschaftsmuster zu einem späteren Zeitpunkt, wird ein sich wiederholendes Ereignis erkannt.

Der Prozess läuft wie folgt ab: Zunächst wird das Material mittels zeitlicher und örtlicher Bildsegmentierung nach bewegten Objekten durchsucht. Zu jedem Frame des Videos wird versucht diese Objekte im vorherigen Frame wiederzufinden, um Objekte zeitlich zu verfolgen. Anschließend werden Bewegungsmerkmale und Farbmerkmale für jedes Objekt errechnet. Dadurch wird jedes Bewegungsmuster als eine Folge von Merkmalen beschrieben. Die zuvor extrahierten Merkmale werden einem Clustering unterzogen, wodurch ein Alphabet zur Beschreibung von komplexen Ereignissen als Symbolketten entsteht. Durch ein erneutes Clustering, welches die Symbolketten miteinander vergleicht, werden ähnliche Ereignisse erkannt.

Wir evaluierten unsere Methode anhand von Langzeitbeobachtungen von Tiergehegen. Unsere Experimente haben gezeigt, dass sich wiederholende Ereignisse mit der vorgeschlagenen Methode erkennen lassen. Diese Arbeit beschreibt unseren Algorithmus, sowie die Ausführung der Experimente. Zum Schluss werden Ergebnisse gezeigt und offene Themen diskutiert.

Danksagung

Für die Unterstützung bei dieser Arbeit möchte ich meinem Mitbetreuer Matthias Zeppelzauer meinen besten Dank aussprechen. Ein spezielles Dankeschön geht an meine Eltern Andrea und Wolfgang Sedelmaier, die mir dieses Studium ermöglicht haben. Weiters danke ich dem Tierpark Schönbrunn für die Zurverfügungstellung von Videomaterialien, sowie meinen Studienkollegen Christian Goldstein und Thomas Hahn für die Unterstützung und Unterhaltung in leidvollen Studienphasen.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation und Problemstellung

Langzeitvideoaufnahmen, wie bei Tierbeobachtungen oder bei der Sicherheitsüberwachung, können viel Zeit in der Nachbearbeitung in Anspruch nehmen. Um alle Vorkommnisse erfassen zu können, ist es meist notwendig das gesamte Videomaterial zu sichten. Wenn das Interesse bei der Videoanalyse auf bestimmten Ereignissen oder Akteuren liegt, ist eine vollständige Durchsicht der Langzeitvideoaufnahmen nicht notwendig. Vor der Durchsicht ist jedoch nicht bekannt, an welchen Stellen sich Ereignisse von Interesse befinden. Diese Arbeit beschäftigt sich mit der automatischen Extraktion wiederkehrender Ereignisse in diesen Videoaufnahmen. Hierbei werden Algorithmen aus der Bildverarbeitung und der Statistik angewendet, um vollautomatisch Ereignisse zu identifizieren und wiederzuerkennen.

Langzeitvideoaufnahmen sollen in zusammengefasster Form dargestellt werden, indem wiederkehrende Ereignisse mit ähnlicher Bewegung und Objekten erkannt werden. Dadurch können häufig auftretende Ereignisse zeitsparend gesichtet werden. Dem Betrachter dieses Materials soll es mit Hilfe dieser Methode möglich sein, in kurzer Zeit die typischen und häufig vorkommenden Ereignisse aus dem Videomaterial herauszufiltern, während alle anderen Ereignisse bzw. Videoabschnitte in denen keine Bewegung stattfindet, ausgeblendet werden. Das System soll bereits beim Erstellen des Videomaterials laufen, sodass in Echtzeit mit dem Ergebnis gearbeitet werden kann.

Vor allem im Bereich der Überwachung oder Dokumentation können Videoaufnahmen von großer Länge entstehen. Diese Arbeit untersucht, am Beispiel von Monitoring-Aufnahmen aus Tiergehegen, wie Sequenzen herausgefiltert werden können, in denen sich Tiere immer wieder gleich verhalten, um dadurch Auffälligkeiten oder ein Verhaltensmuster zu erkennen.

1.2 Beitrag zum Forschungsgebiet

In verwandten Arbeiten geht es darum Ereignisse in Videoaufnahmen wiederzuerkennen, mit dem Unterschied, dass oft Merkmale nicht von einzelnen Objekten extrahiert werden, sondern der ganze Bereich eines Frames dafür verwendet wird. [6][30] Dadurch ist es schwierig Ereignisse anhand von Bewegungen und Aussehen der Objekte festzumachen. Wenn man hingegen jede Bewegung im Bild als möglichen Objektkandidaten sieht und aus jeder Bewegung Merkmale ausliest, so kann jeder Bewegungsablauf und das Aussehen der Objekte mit allen anderen verglichen und auf Ähnlichkeit überprüft werden. In einigen Methoden [21] werden Objektbewegungen anhand von Trajektorien beschrieben. Hierbei werden die Standorte der Objekte ausgelesen und ihre Bewegungspfade gespeichert. Anschießend werden diese miteinander verglichen und auf Ausreißer überprüft, um ungewöhnliche Bewegungen zu detektieren.

Die hier vorgestellte Methode unterscheidet sich auch von anderen, indem ein zweistufiges Clustering genützt wird. In einem ersten Clustering werden sogenannte Low-Level-Features extrahiert. Hierbei werden Bewegungsrichtung, Ort und Farbe von bewegten Objekten als Merkmale aufgezeichnet, wobei eine Objektbewegung immer aus zwei aufeinanderfolgende Frames erkannt wird. Je nach Anwendungsbedarf werden die benötigten Merkmalskategorien gewählt und anschließend geclustert. Die einzelnen Cluster werden als beschreibendes Alphabet der Objekte und ihrer Bewegungen genützt. Um Ereignisse zu erkennen, werden Objekte über die Zeit verfolgt. Somit besteht ein Ereignis aus einer Folge von Objekten, die über eine Zeichenkette beschrieben werden kann. Danach werden diese Zeichenketten mittels eines Verfahrens, das ursprünglich aus der Spracherkennung kommt, miteinander verglichen und über ein weiteres Clustering gruppiert.

1.3 Anwendungen

Automatisierte Videoanalyse-Systeme sind in verschiedensten Anwendungsbereichen vertreten. In der Sicherheitsüberwachung kommen diese Systeme vermehrt zum Einsatz, um auf außergewöhnliche Ereignisse hinzuweisen, wie Anomalien oder Aktivitäten in sonst inaktiven Bereichen. Auch bestimmte Bewegungsmuster von Fahrzeugen, Passanten oder Kunden sind von bedeutendem Interesse. So können an öffentlichen Plätzen, wie Flughäfen, ungewöhnliche Verhaltensmuster automatisch erkannt werden, wie das Zurücklassen eines Gepäckstückes. Im Straßenverkehr können mit Hilfe dieser Systeme Verstöße gegen die Verkehrsordnung erkannt werden. Um diese aussergewöhnlichen Ereignisse zu erkennen, müssen die sich wiederholenden Ereignisse herausgefiltert werden. Unsere Methode kann daher auch als Grundlage für die zuvor erwähnten Anwendungsgebiete genutzt werden.

Auch in der Tierbeobachtung finden Videoanalyse-Systeme ihre Anwendung. Ein Beispiel hierzu ist das Kennzeichnen von Zeitpunkten in Langzeitvideoaufnahmen, in denen Aktivität stattgefunden hat. Somit können Videosequenzen ohne Aktivität bei der Durchsicht übersprungen werden. Eine Erweiterung dieser Funktion wäre es festzustellen, ob ein gewisses Ereignis öfter auftritt. Somit könnte beim Betrachten eines Ereignisses auf weitere Zeitpunkte, an denen ähnliche Ereignisse stattfanden, hingewiesen werden. Bestehende Systeme benötigen für eine solche Funktion vordefinierte Beschreibungen eines Ereignisses von Interesse.

1.4 Aufbau der Arbeit

In Kapitel 2 wird auf die theoretischen Grundlagen, auf die diese Methode aufbaut, näher eingegangen. Dabei werden Methoden aus der Bildverarbeitung, wie morphologische Filter und diverse Verfahren zur Hintergrundmodellierung beschrieben. Im Anschluss wird gezeigt, wie der optische Fluss zwischen zwei aufeinander folgenden Bildern einer Videosequenz bestimmt werden kann. Am Ende des Kapitels werden Methoden aus der Mustererken-

nung vorgestellt, die ihre Anwendung im Clustering und im String-Matching finden.

Kapitel 3 zeigt Arbeiten, die mit der hier vorgestellten Arbeit verwandt sind. Hierbei wird auf Funktionsweise, Unterschiede und Gemeinsamkeiten näher eingegangen.

In Kapitel 4 wird die neuentwickelte Methode beschrieben. Dabei wird zuerst ein Flussdiagramm präsentiert, das den wesentlichen Ablauf veranschaulicht. Diesem Diagramm wird prozedural in der gesamten Methodenbeschreibung gefolgt. Hierbei wird zunächst beschrieben, wie die Bewegungserkennung und Objektidentifizierung stattfindet. Anschließend wird erläutert, welche Merkmale aus den erkannten Objekten extrahiert und wie diese einem Clustering unterzogen werden. Es wird weiter gezeigt, wie Objekte als Symbolkette beschrieben werden können. Als nächstes wird beschrieben, wie das Distanzmaß zwischen den Symbolketten für die Erstellung einer Distanzmatrix berechnet wird. Zum Abschluss wird ein weiteres Clustering vorgestellt, welches die zuvor erkannten Objekte nach ihrem Aussehen, ihres Aufenthaltsortes und ihrer Bewegungsrichtung gruppiert und das Videomaterial in einer kompakteren Form ausgegeben werden kann.

In Kapitel 5 werden zuerst die verwendeten Daten, die zur Evaluierung dienen, beschrieben. Als nächstes wird der Evaluierungsaufbau gezeigt. Es werden hierzu synthetische sowie auch reale Videosequenzen genutzt. Es werden die verschiedene Ergebnisse der Methode präsentiert und analysiert. Die Evaluierung teilt sich hierzu in ein Proof of Concept und eine qualitative sowie eine quantitative Analyse.

In Kapitel 6 wird die hier vorgestellte Methode zusammengefasst und bewertet. Es wird gezeigt, welche Barrieren bei der Methodenentwicklung aufgetreten sind, inwiefern das gestellte Problem gelöst wurde und welche Möglichkeiten der Weiterentwicklung von bestehenden Systemen und der vorgestellten Methode vorhanden sind.

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen näher beschrieben, auf der die hier vorgestellte Methode basiert. Die hier beschriebenen Verfahren werden anhand der Reihenfolge der im praktischen Teil beschriebenen Methodenschritte behandelt. Als erstes werden Methoden aus der Bildverarbeitung, wie morphologische Filter und diverse Verfahren zur Hintergrundmodellierung beschrieben. Als nächster Punkt wird gezeigt, wie der optische Fluss zwischen zwei aufeinander folgenden Bildern einer Videosequenz bestimmt werden kann. Anschließend werden Methoden aus der Mustererkennung vorgestellt, beginnend mit Möglichkeiten der Merkmalsextraktion. Im Weiteren werden String-Matching-Algorithmen und grundlegende Clustering-Methoden vorgestellt.

2.1 Morphologische Filter

Morphologische Filter sind wichtige Werkzeuge für die Analyse räumlicher Strukturen in Bildern. Mit ihrer Hilfe lässt sich die Form von Objekten beschreiben, sowie auch Veränderungen der Form herbeiführen. Die zwei grundlegenden Operationen sind Dilatation und Erosion. Hierbei geht man meist von einem Binärbild aus, in dem Objekte mit Pixelwert 1 und Regionen ohne Objekt mit Pixelwert 0 festgelegt werden. Mittels eines definierten Strukturelementes können diese Objekte manipuliert werden, um gewünschte Effekte zu erzielen. Diese Filter können auch auf Grauwertbilder angewendet werden. [1]

2.1.1 Erosion und Dilatation

Bei der Erosion wird ein Objekt verkleinert. Es wird überprüft, welche Verschiebungen von einem Strukturelement in ein Objekt passt. Hierbei wird das Strukturelement pixelweise über das ganze Bild verschoben und zu jedem Pixel überprüft, ob das Strukturelement an dieser Stelle als Ganzes in ein Objekt passt. Ist dies der Fall, dann gehören diese Pixel an dieser

Stelle weiterhin zu diesem Objekt. Alle anderen Pixel werden nicht mehr zu diesem Objekt gezählt. Die Dilatation ist der Gegenspieler der Erosion und vergrößert somit Objekte. Es wird an jeder Pixelposition überprüft, ob das Strukturelement ein Objekt berührt. Ist dies der Fall wird die Pixelposition zum Objekt hinzugefügt und das Objekt dehnt sich aus. [1]

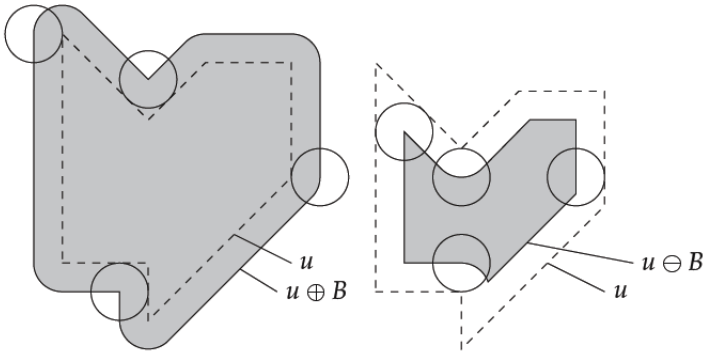


Abbildung 1: Dilatation (links) und Erosion (rechts) eines Objektes (gestrichelte Linie) mit einer Kreisscheibe als Strukturelement (Bild von [1]).

2.1.2 Opening und Closing

Wenn die beiden Operationen, Dilatation und Erosion, kombiniert werden, können weitere Effekte auf Binärbilder oder Grauwertbilder erzielt werden. Objekte, die kleiner als das Strukturelement sind, können mittels Erosion entfernt werden. Jedoch sind alle übrig gebliebenen Objekte um das Strukturelement verkleinert. Erfolgt anschließend eine Dilatation mit dem selben Strukturelement, vergrößern sich die Objekte wieder annähernd auf ihre ursprüngliche Größe. Diese Kombination der beiden Operationen nennt man Opening. Der umgekehrte Fall wird Closing genannt. Hierbei werden Löcher in Objekten eines Binärbildes mittels Dilatation gefüllt, die kleiner als das Strukturelement sind. Als Ergebnis erhält man Objekte, die um das Strukturelement größer sind. Erfolgt nun eine anschließende Erosion, wird diese Vergrößerung wieder rückgängig gemacht. Somit lassen sich Ungenauigkeiten der Objekterkennung mittels Schwellwertbildung glätten. [1]

2.2 Hintergrund Subtraktion

Dieses Verfahren ist eine weit verbreitete Methode, um sich bewegende Objekte zu ermitteln. Hierbei wird meist eine statische Kamera vorausgesetzt, um ein Video mit gleichbleibendem Blickwinkel zu erzeugen. Es gibt verschiedenste Methoden, die sich nach Geschwindigkeit, Speicherbedarf und Genauigkeit unterscheiden. Das Problem und der Lösungsansatz sind jedoch bei allen gleich. Man will den sich verändernden Vordergrund vom statischen Hintergrund trennen, um im Anschluss nur mehr den Vordergrund zu erhalten. Somit ist es möglich nur jene Objekte zu beobachten, die sich bewegen.

Hierzu benötigt man die Differenz zwischen dem aktuellem Videoframe und einem Referenzframe. Bei der Hintergrund Subtraktion ist dies das Hintergrundbild und repräsentiert eine Szene ohne sich bewegender Objekte.

Im Folgenden werden Methoden der Hintergrund Subtraktion vorgestellt, die sich in ihrer Komplexität, Geschwindigkeit, Leistungsfähigkeit und ihrem Speicherbedarf unterscheiden.

Abbildung 2: Hintergrund Subtraktion mit Hilfe des aktuellen Videoframes (links oben) und einem Hintergrundbild (rechts oben). Das Ergebnis ist ein Differenzbild (links unten) aus dem im Anschluss ein Schwellwertbild (rechts unten) generiert werden kann.

2.2.1 Mittelwertverfahren

Dieses Verfahren startet mit einer Initialisierungsphase, bei der das Hintergrundmodell berechnet wird. Es werden hierbei für jede Position eines Pixels im Bild die Mittelwerte über alle Bilder einer Bildsequenz berechnet. Das erreicht man am einfachsten, in dem man alle Bilder pixelweise addiert und durch die Größe der Bildsequenz dividiert. Das ergibt das Arithmetische Mittel für jede Pixelposition.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{1}$$

Weiters kann man auch den Median statt dem arithmetischen Mittel zur Mittelwert Berechnung nutzen. Hierbei wird eine Zahlenreihe nach ihren Werten sortiert und der Wert, der an mittlerer Stelle steht, als Repräsentant herangezogen. Diese Methode ist sehr Leistungs- und Speicherintensiv, da hier drei Farbwerte für jedes Pixel sortiert werden müssen. [20]

Nachdem das Hintergrundmodell erstellt wurde, wird jedes Bild mit dem Hintergrundbild verglichen, indem das Hintergrundbild vom aktuellen Videoframe subtrahiert wird. Bei Grauwertbildern ist dies der absolute Wert einer pixelweisen Subtraktion. Bei Farbbildern mit mehreren Farbwerten, wie im Lab-Farbmodell, kann der Abstand wie folgt berechnet werden. [23]

$$\Delta E = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2} \tag{2}$$

Differenz $d(x, y)$ zwischen den Bildern x und y:

$$d(x, y) = \sqrt{(x_L - y_L)^2 + (x_a - y_a)^2 + (x_b - y_b)^2} \tag{3}$$

2.2.2 Running Average

Der Nachteil der im vorigen Punkt beschriebenen Methode ist, dass sich das Hintergrundbild nicht auf Veränderungen der Umgebung einstellen kann. So kann es bei Außenaufnahmen vorkommen, dass es zuerst bewölkt und später sonnig ist. Die Helligkeit unterliegt hierbei großen Schwankungen. Das Hintergrundmodell ist somit für beide Wetterzustände nicht optimal. Es gibt die Möglichkeit das Hintergrundmodell mittels Running Average aktuell zu halten. Hierbei wird für jeden Frame der Hintergrund fließend angepasst.

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha F_t(x, y) \tag{4}$$

Die Variable α ist ein regulierender Wert, der zwischen 0 und 1 beträgt und bestimmt, mit welcher Intensität ein neuer Frame F ins Hintergrundbild B mit einfließen soll. Eine Initialisierungsphase ist hierbei nicht notwendig. Es kann aber ein gewisser Zeitabschnitt genützt werden in welchem ein Hintergrundbild erzeugt wird, aber keine Subtraktion der Frames stattfindet. Dadurch ist eine genauere Detektion des Vordergrundes bereits anfangs möglich. [28]

Die konstante Laufzeit $\mathcal{O}(1)$ und Speicherbedarf $\mathcal{O}(1)$ dieses Verfahrens können für Echtzeitsysteme ein Vorteil sein. Ein Nachteil kann sein, dass bei langsamen Bewegungen der Vordergrund zu stark ins Hintergrundbild mit einfließt. Objekte heben sich unter Umständen zu wenig vom Hintergrund ab und die Intensität des Subtraktionsbildes übersteigt nicht den Schwellwert, um als Vordergrund deklariert zu werden. Als Abhilfe kann man nur Pixel ins Hintergrundbild aufnehmen, die nicht als Objekt bzw. Vordergrund erkannt wurden. Jedoch bleiben Objekte, die sich zuerst bewegt haben und im Anschluss stehen bleiben, als Vordergrund erhalten. [28]

2.2.3 Running Gaussian Average

Dieses Verfahren basiert auf der Gauß-Funktion, welche auch als Normalverteilung bekannt ist. Hierbei wird für jede Pixelposition der Mittelwert und die Varianz einer Bildsequenz berechnet. Aus diesen zwei Werten kann eine Gauß-Funktion modelliert werden. Der Mittelwert wird wie im vorigen Kapitel aus einem Running Average errechnet: [20]

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \tag{5}$$

I gibt das laufende Pixel zum Zeitpunkt t und μ_{t-1} den vorhergehenden Mittwert an. Um den Einfluss auf das Hintergrundmodell des neuen Pixels zu steuern, wird α als regulierendes Gewicht verwendet. Diese Methode zur Hintergrund Subtraktion ist äußerst ressourcensparend, da nur Mittelwert und Varianz für jede Pixelposition gespeichert werden müssen. Damit ein Pixel als Vordergrund eingestuft werden kann, gilt folgendes Kriterium: [20]

$$|I_t - \mu_t| > k\sigma_t \tag{6}$$

Der Schwellwert k ist eine gewählte Konstante und reguliert, ab welcher Differenz ein Pixel als Vordergrund deklariert wird. Aus der Varianz kann die Standardabweichung σ errechnet werden.

2.3 Optischer Fluss

Der optische Fluss ist die sichtbare Bewegung von Objekten in Bildfolgen. Jeder Vektor gibt eine Geschwindigkeit und Richtung einer Pixelbewegung an. [12] Das bedeutet, dass jeder Wert im Vektorfeld anzeigt, wo sich die Pixel eines Bildes im folgenden Bild einer Bildsequenz hin bewegt haben. Somit bietet der optische Fluss auch die Möglichkeit Vordergrundobjekte in

Videosequenzen zu verfolgen. Ein Beispiel für das Vektorfeld eines optischen Flusses wird in Abbildung 3 gezeigt.

Damit Bewegungen in Bildsequenzen erkannt werden können, werden die Grauwerte der aufeinander folgenden Bilder miteinander verglichen. Hierbei geht man von einer konstanten Beleuchtung und Position des Beobachters bzw. der Kamera aus. Das tatsächliche Bewegungsfeld unterscheidet sich vom optischen Fluss, da Grauwerte nur die Veränderung der Helligkeitsintensität wieder geben. Eine drehende Kugel hat somit gleichbleibende Grauwerte, obwohl eine Bewegung stattfindet. Dieser Umstand wird jedoch für die Bestimmung des optischen Flusses vernachlässigt. [19]

Es gibt verschiedene Methoden um den optischen Fluss in Bildsequenzen zu errechnen. Man kann diesen mittels lokalen Methoden, wie etwa jene von Lucas und Kanade [17] und mittels globalen Methoden, wie in Horn und Schunck [12] beschrieben, errechnen. Meist sind lokale Methoden robuster gegen Störungsrauschen, während globale Methoden ein dichteres Strömungsfeld erzeugen. [3]

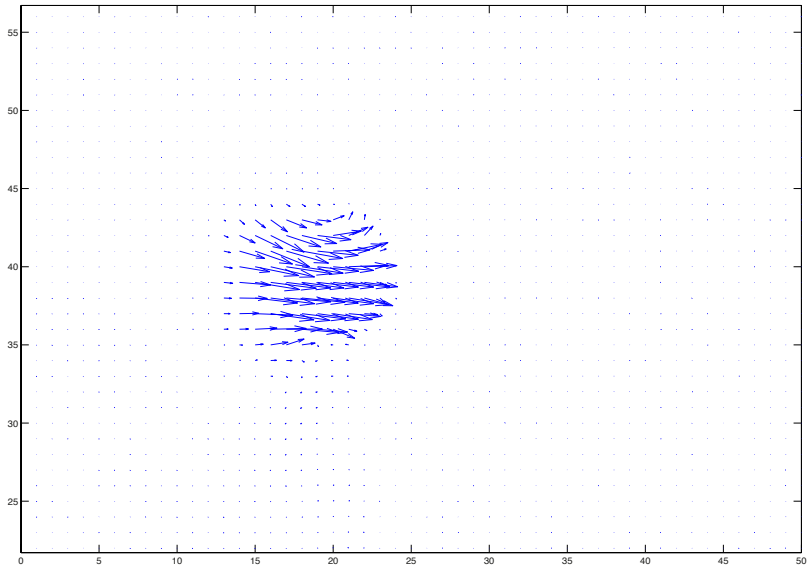


Abbildung 3: Vektorfeld des optischen Flusses.

2.4 Merkmalsextraktion

Ein wichtiger Schritt in der Objekterkennung ist die Merkmalsextraktion. Sie dient dazu Objekte anhand von diskriminativen Parametern unterscheiden zu können. Diese Parameter werden für jedes Objekt in einem hochdimensionalen Merkmalsvektor zusammengefasst. Je ähnlicher Objekte aussehen oder sich verhalten, umso näher liegen diese im Merkmalsraum beieinander. Solche Gruppierungen von ähnlichen Objekten in einem Merkmalsraum nennt man Cluster.

Welche Merkmale gewählt werden, um Objektklassen zu trennen, ist von der Anwendung abhängig. Zur Beschreibung von Objekten, sollen die Merkmale translations-, rotations-, größen- und spiegelinvariant. Dadurch verringert sich der Aufwand der Bildverarbeitung, da diese Parameter nicht orts- oder achsengebunden und die Objekte nicht gleich groß oder gleich ausgerichtet sein müssen. Ein guter Merkmalsvektor kann Objekte exakt zu ihrer zugehörigen Objektklasse zuweisen und erlaubt ein gutes Clustering im Merkmalsraum. [9]

Geometrische, translations- und rotationsinvariante, größenvariante Merkmale:

- Fläche - Die Anzahl der Pixel eines Objektes
- Masse - Summe der Grauwerte eines Objektes im Grauwertbild
- Umfang - Anzahl der Randpixel eines Objektes

Ein Beispiel für ein geometrisches größeninvariantes Merkmal ist der Schwerpunkt eines Objektes. Jedoch ist dieses Merkmal nur rotationsinvariant bei Rotation um den Schwerpunkt und ist nicht translationsinvariant. Weitere Beispiele für Merkmalsparameter sind, die Anzahl an Löchern innerhalb eines Objektes (Euler-Zahl), der Mittelwert der RGB-Farbwerte eines Objektes oder das Grauwert- oder Farbhistogramm. Diese Merkmale sind translations-, rotations-, größen- und spiegelinvariant, da ihre Parameter bei

Verschieben, Rotation, Skalierung und Spiegelung des Objekts konstant bleiben. Ein Grauwerthistogramm zeigt die Häufigkeitsverteilung der Grauwerte eines Bildes oder eines Bildausschnittes an. Hierbei wird für jeden annehmbaren Grauwert gezählt wie viele Pixel diesen Wert in einem Bild besitzen. Die örtliche Information der einzelnen Pixel geht hierbei verloren, sowie die Information über die Lage und Größe eines Objektes. Die Skala der Grauwerte kann verringert werden. Dadurch verringert sich die Dimensionalität des Histogramms, die Datenmenge verringert sich und kann daher leichter zur Merkmalsextraktion eingesetzt werden. Die Häufigkeitswerte sind somit in Intervalle bzw. Klassen eingeteilt. [9]

Auch von einem Farbbild kann ein Histogramm errechnet werden. Eine Möglichkeit ist, dass von jedem Farbkanal ein eigenes Histogramm angelegt wird. Im Fall eines RGB-Bildes wäre dies jeweils ein Histogramm für den Rot-, Grün- und Blau-Kanal. Es kann jedoch auch ein mehrdimensionales Histogramm erstellt werden. Bei einem RGB-Farbbild werden die Häufigkeiten der Farben in einem quadratischen Farbwürfel mit den drei Achsen für Rot, Grün und Blau eingetragen.

In der Abbildung 4 sieht man ein mehrdimensionales Farbhistogramm, das die Häufigkeit von 27 unterschiedlichen Basis-Farbwerten zeigen kann. Hierzu wird jedes Pixel des Bildes zu jenem Basis-Farbwert gezählt, zu dem sein Farbwert am nächsten ist.

Weitere Möglichkeiten um Merkmale aus Bildern zu erhalten sind:

- Lokale Merkmale - Kantenerkennung, Eckendetektion, Detektion von Interest Points
- Textur Merkmale - Kontrast, Grobheit, Orientierung
- Bewegungsmerkmale - Optischer Fluss

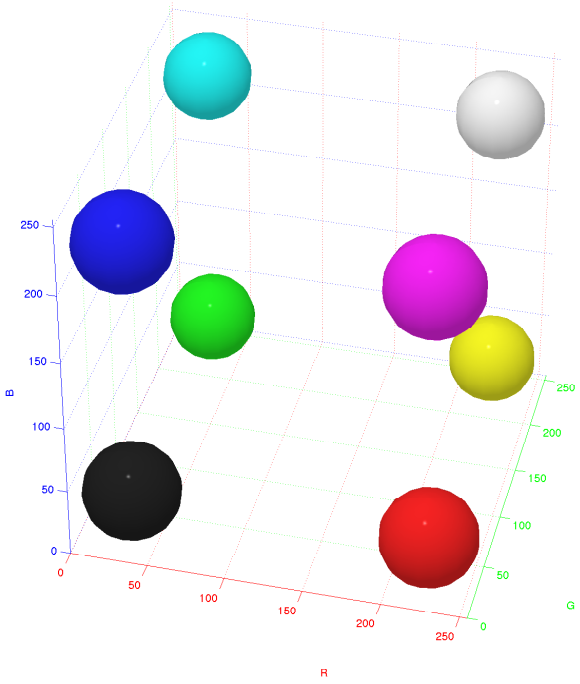


Abbildung 4: 3D Farbhistogramm (Bild erstellt mit [29]).

2.5 Dichtebasiertes Clustering

Mit Hilfe der Clusteranalyse lassen sich Strukturen in Merkmalsdaten erkennen. In vielen Anwendungen ist es nicht möglich oder zu aufwendig eine Klassenzugehörigkeit zu bestimmen. Beim Clustering wird versucht anhand der Merkmalsdaten in einem Merkmalsraum, Häufungspunkte zu erkennen und somit den Datensatz zu gruppieren. [22]

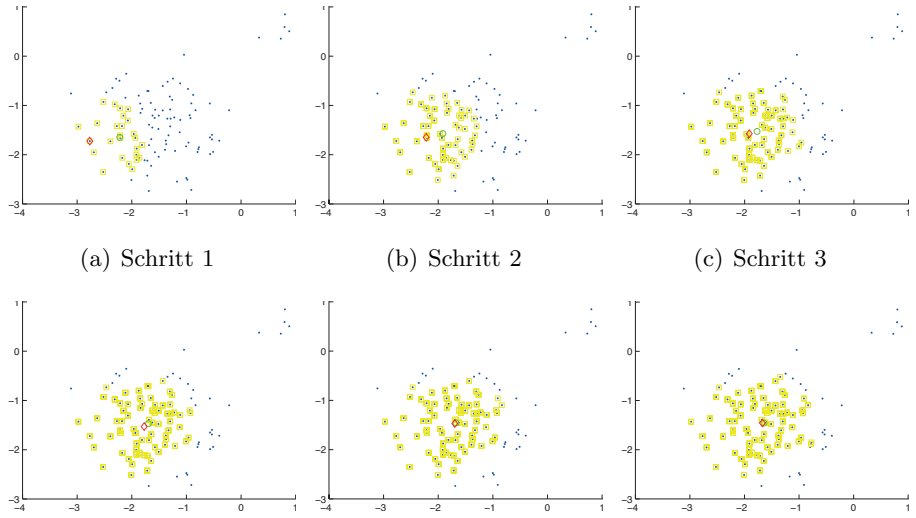
Dichtebasierte Clustering Methoden werden häufig genutzt, wenn zuvor nicht bekannt ist, wie viele Cluster gebildet werden. Hierzu kann die Berechnung der Dichte in einem Merkmalsraum genützt werden, wobei Cluster als Regionen mit hoher Konzentration von Datenpunkten betrachtet werden. Zwischen den Clustern befinden sich Regionen niedriger Dichte, die somit eine geringere Konzentrationen von Datenpunkten aufweisen. [26]

Sind sich Objekte sehr ähnlich, ist die Distanz zwischen den von den Objekten extrahierten Merkmalen gering. Das dichtebasierte Clusterverfahren DBSCAN [10] bestimmt die Dichte eines Merkmalraumes über die zwei Parameter ε und $minPts$. Die ε-Nachbarschaft von p beschreibt ein Beobachtungsfenster in Form einer Hyperkugel mit Radius ε um den Punkt p. Der Parameter $minPts$ gibt die minimale Anzahl von Punkten innerhalb des Beobachtungsfensters an, um einen Punkt als Kernpunkt zu deklarieren. Benachbarte Kernpunkte werden zu Clustern gruppiert. Randpunkte eines Clusters können von anderen Kernpunkten erreicht werden, besitzen jedoch zu wenige Nachbarpunkte innerhalb ihres Beobachtungsfensters. Ein Punkt wird als Clusterpunkt definiert, wenn dieser in der ε-Nachbarschaft eines Kernpunktes liegt. Somit lassen sich Cluster unterschiedlicher Größe und Form erkennen. [13]

2.5.1 Mean Shift Clustering

Eine effiziente Methode, die Maxima der Verteilungsdichte in einem Datensatz eines Merkmalraumes zu erkennen, ist Mean Shift. Andere Methoden, wie Mixture of Gaussians, sind für manche Anwendungen nicht stark genug um komplexe Verteilungen abzuschätzen. Mean Shift arbeitet mit einem Kernel Dichteschätzer, welcher beliebige Verteilungen schätzen kann und dabei keine Gradienten errechnen muss. Ein Merkmalsraum besteht aus einer endlichen Menge an Beobachtungen. Mean Shift verfolgt das Ziel, die Maxima der Dichte in einem Merkmalsraum zu finden. Der Kernel Dichteschätzer errechnet einen sogenannten Mean Shift Vektor. Dieser zeigt in Richtung des Gradienten. Nimmt man einen Ausschnitt eines Merkmalsraumes und errechnet den Mean Shift Vektor, so erhält man eine Angabe über die Richtung, in der die Dichte am stärksten steigt. Wird dieses Fenster nun um diesen Vektor verschoben und aus diesem neuen Bereich der Mean Shift Vektor errechnet, nähert sich das Beobachtungsfester immer weiter einem Maximum der Dichte. Werden diese Schritte nun wiederholt (Abbildung 5), konvergiert der Beobachtungsbereich zu einem der Maxima der Dichte. Al-

so jenen Stellen in einem Merkmalsraum, an denen der Mean Shift Vektor gleich Null ist. Vorteile von Mean Shift sind, dass keine Schrittgröße gewählt und kein Gradient ausgerechnet werden muss. Die Schrittgröße ergibt sich adaptiv aus dem Mean Shift Vektor. [31]



(a) Schritt 1 (b) Schritt 2 (c) Schritt 3

(d) Schritt 4 (e) Schritt 5 (f) Schritt 6

Abbildung 5: Der Mean Shift Vektor wird solange in Richtung des Gradienten verschoben bis ein Maximum der Dichte gefunden wurde. Die rote Raute kennzeichnet den Mean Shift Vektor des vorherigen Schrittes. Der grüne Kreis stellt den aktuellen Mean Shift Vektor dar. Gelbe Vierecke markieren Punkte, die zur Verktorberechnung verwendet wurden.

Um dieses Verfahren für Clustering zu nutzen wird für jeden Eintrag im Merkmalsraum ein Mean Shift Vektor errechnet. Die Beobachtungsfenster werden in Richtung der Gradienten und somit in Richtung der Maxima bei jedem Schritt um den Mean Shift Vektor verschoben. Wenn jeder Punkt den kompletten Pfad zu einem Maximum zurückgelegt hat, werden die Orte der Maxima als Clusterzentren festgelegt. Die Zugehörigkeit eines Merkmalseintrages zu einem Cluster wird über seinen Mean Shift Pfad bestimmt. Dieser führt direkt in ein jeweiliges Clusterzentrum. Alle Elemente, deren Mean Shift Pfad in einem Clusterzentrum zusammentreffen, gehören demselben Cluster an. [7]

Der Vorteil dieser Methode gegenüber anderen ist, dass keine Information über die Verteilung bekannt sein muss. Somit ergibt sich die Clusteranzahl automatisch und ist nur von der Größe des Beobachtungsfensters abhängig. [11]

Die Clusteranzahl wird jedoch von der Größe des Beobachtungsfensters beeinflusst. Wird eine fixierte Bandbreite festgelegt, wird für jeden Datenpunkt im Merkmalsraum der selbe Fensterradius genutzt. In [11] werden mehrere Methoden der adaptiven Bandbreite vorgestellt. Die Methode des Nearest Neighbour stellt hierbei die einfachste Variante dar. Es wird die Bandbreite zu jedem Punkt anhand der nächsten Nachbaren geschätzt. Somit können Cluster unterschiedlicher Dichte gefunden werden. Eine semi-adaptive Methode wird in [4] näher erläutert. Hierbei wird die Bandbreite schrittweise um maximal 10% verändert. [31]

2.6 String Matching

Neben der numerischen Darstellung von Merkmalsvektoren, können Daten auch symbolisch als Symbolketten, sogenannten Strings, dargestellt werden. Im Allgemeinen können diese Zeichenketten unterschiedlich lang sein. Daher sind spezielle Methoden für ihren Vergleich notwendig. Im Folgenden werden zwei Methoden vorgestellt, die ein Ähnlichkeitsmaß zwischen zwei Sequenzen unterschiedlicher Länge errechnen können.

2.6.1 Levenshtein-Distanz

Mit Hilfe der Levenshtein-Distanz können Zeichenketten unterschiedlicher Länge verglichen werden. Sie gibt an, wie viele Änderungsoperationen notwendig sind, um aus der einen Zeichenkette eine andere bestimmte Zeichenkette zu formen. Daher wird diese Methode auch als Editierdistanz bezeichnet, wobei die zulässigen Operationen Einfügen, Löschen, Ersetzen und Verschieben von Zeichen sind. Im Gegensatz zum Hamming-Abstand, können so Zeichenketten verglichen werden, die unterschiedlich lang sind.

Die Levenshtein-Distanz ist ein Wert, der die minimale Anzahl an Änderungsoperationen zwischen zwei Strings angibt. [8]

Vergleicht man die Wörter „Abend“ und „Robe“ so sind mindestens 4 Operationen notwendig, um eine Zeichenkette in die andere zu überführen. [24]

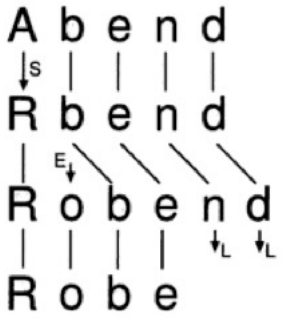


Abbildung 6: Editierdistanz zwischen „Robe“ und „Abend“; S=Substitution; E=Einfügung; L=Löschung (Bild von [24]).

In Abbildung 6 wird das „A“ durch „R“ ersetzt, ein „o“ wird eingefügt sowie „n“ und „d“ entfernt.

2.6.2 Dynamic-Time-Warping

Dynamic-Time-Warping (DTW) ist eine Technik zum Vergleich zweier zeitabhängiger Signale bzw. Sequenzen. DTW wurde ursprünglich für die Spracherkennung entwickelt. Die beiden Sequenzen müssen hierbei zeitdiskrete Signale, Merkmalssequenzen oder Symbolketten sein. Um diese Sequenzen vergleichen zu können, benötigt man eine lokale Kosten- oder Abstandsfunktion. Wenn sich die zwei betrachteten Sequenzen ähnlich sind, hat das Distanzmaß einen geringen Wert. Im Gegensatz dazu wird der Wert umso höher, je unterschiedlicher die Signale sind. Das Ziel ist eine angeglichene Anordnung zu finden, bei der die Kosten der Kostenfunktion am geringsten sind. [18]

Die Hauptprobleme hierbei sind, die zeitliche Verzerrung und die unterschiedliche Länge der Signale. Eine Zeitnormierung wird mit Hilfe des dyna-

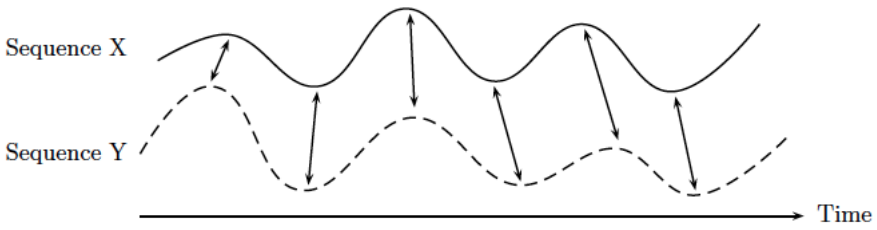


Abbildung 7: Zwei zeitabhängige Sequenzen entlang einer Zeitachse. Abgeglichene Punkte sind mit Pfeilen gekennzeichnet. (Bild von [18]).

mischen Programmierens erreicht, wobei die Wörter bzw. Symbolsequenzen gedehnt und gestaucht werden. Dies erfolgt jedoch nicht proportional, sondern unregelmäßig, sodass nur die Folge der Symbole und nicht deren hintereinander wiederholtes Auftreten ausschlaggebend für die Kostenfunktion ist. [5]

3 Verwandte Arbeiten

Die in dieser Arbeit vorgestellte Methode wurde bisher noch nicht näher in der Literatur beschrieben, jedoch gibt es wissenschaftliche Arbeiten, die ähnliche Ziele verfolgen. Die verwendeten Methoden kommen aus dem Bereich der Überwachung und dem Monitoring. Es wird versucht wiederkehrende Ereignisse in Videoaufnahmen zu finden. Hierzu wird als erstes definiert, wann ein Ereignis beginnt und endet. In weiterer Folge müssen geeignete Merkmale gefunden werden, um Ereignisse miteinander vergleichen zu können.

3.1 Sequenzsegmentierung zur Ereigniserkennung

Die Methode in [30] versucht mit Hilfe eines einfachen statistischen Distanzmaßes Videosequenzen miteinander zu vergleichen. Das Distanzmaß wird zum Isolieren und Clustern von Ereignissen in Langzeitvideoaufnahmen genutzt. Hierzu sind keine a priori Kenntnisse über die Art oder über die zeitliche Ausdehnung der Ereignisse notwendig. Nachdem das Video in Subsequenzen segmentiert wurde, erfolgt ein Clustering dieser, wobei eine Subsequenz als Ereignis betrachtet wird. Als Ergebnis des Clustering-Prozesses erhält man Gruppierungen von ähnlichen Ereignissen.

Ein Anwendungsbeispiel für dieses System ist das Wiedererkennen eines speziellen Ereignisses, wie zum Beispiel eine kurze Sequenz, in der ein Tennisaufschlag zu sehen ist. Das System kann zu einer ähnlichen Szene im Schnelllauf vorspulen.

Die Methode wird an verschiedenen Personenbewegungen angewandt, die mit Hilfe einer statischen Kamera aufgezeichnet wurden. Dabei trägt die Person unterschiedliche Kleidung und wurde von verschiedenen Perspektiven und Entfernungen aufgenommen. Ein weiteres Video enthält Aufnahmen von sportlichen Aktivitäten.

In der Arbeit [30] wird ein Video in immer kleinere zeitliche Stücke zerteilt, welche dann auf ihre Ähnlichkeit über ein Distanzmaß zwischen den

einzelnen Sequenzen geprüft werden. Hierbei sind keine Trainingsdaten erforderlich. Das System clustert die Sequenzen ohne Vorwissen. Hierbei wird ein statisches Distanzmaß zwischen den Videosequenzen anhand ihres Inhaltes errechnet. Dabei können die Sequenzen auch unterschiedliche Länge haben. Dieses nicht-parametrische Distanzmaß kann genutzt werden, Ereignisse in langen Videosequenzen zu clustern. Dabei wird davon ausgegangen, dass ähnliche Ereignisse auch ähnlich viel Zeit beanspruchen. Um Aktivitäten zu verschiedenen zeitlichen Auflösungen zu extrahieren, wird eine „zeitliche Pyramide“ des gesamten Videos mittels Weichzeichner und unterschiedlichen Abtastraten erzeugt. Dabei haben alle Sequenzen in der Pyramide die gleiche Frameanzahl. Eine Ebene in der Pyramide besitzt halb so viele Frames, wie in der nächst höher aufgelösten Stufe. Von jeder Sequenz wird der lokale Itensitätsgrad von jedem Punkt im Bild zu jeder Zeit bestimmt. Die Gradientenrichtung gibt an, in welche Richtung die stärkste Änderung der Intensität eines Pixels zu seinen Nachbarpixel aufweist. Diese ist vom lokalen Verhalten eines bewegten Objektes abhängig. [30]

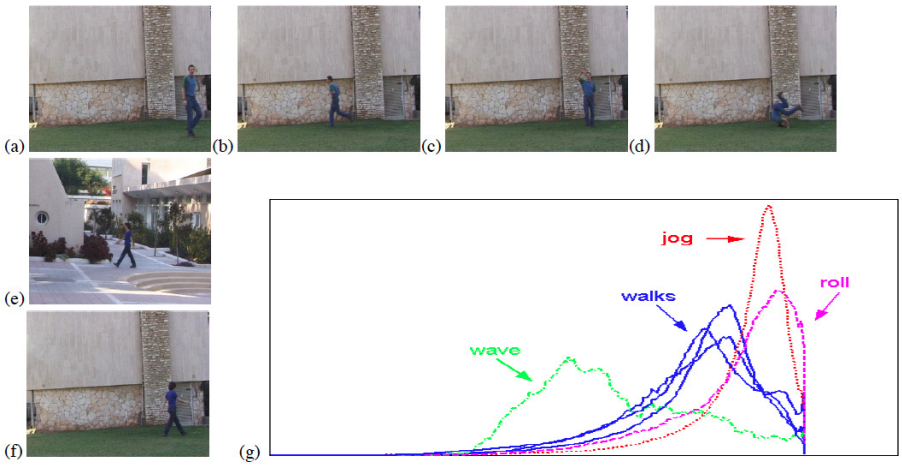


Abbildung 8: Die empirische Verteilung von 6 verschiedenen Videoclips (Bild von [30]).

Die Abbildung 8 zeigt die empirische Verteilung einer Merkmalskomponente von sechs Verschiedenen Videoclips. In drei der Clips geht eine Person mit unterschiedlicher Kleidung und vor unterschiedlichem Hintergrund, so-

wie aus verschiedenen Perspektiven und Entfernungen durch das Bild. In den anderen drei Clips führt diese Person verschiedene Aktivitäten aus, hat jedoch dieselbe Kleidung an. Der Hintergrund, die Perspektive und die Entfernung sind bei allen drei Aufnahmen ebenfalls gleich. Die Verteilung der „walks“ Clips liegt enger zusammen, als jene der drei Clips mit verschiedenen Aktivitäten. Dies zeigt, dass ähnliche Bewegungen erkannt werden können, unabhängig von Kleidung, Hintergrund, Perspektive oder Entfernung. [30]

Das Distanzmaß zwischen zwei Sequenzen wird berechnet, indem die empirische Verteilung von allen Merkmalen in den Auflösungsstufen der zwei Sequenzen auf einen einzelnen Distanzwert mittels Divergenz heruntergerechnet wird. Um die Empfindlichkeit bei kleineren Unterschieden des Histogramms zu senken, werden diese geglättet. Verschiedene Auflösungsstufen der Merkmale sollen ein besseres Distanzmaß ergeben. Dafür werden aber multidimensionale Histogramme benötigt, welche speicher- und rechenintensiv sind. Daher werden die Histogramme zu einem eindimensionalen Histogramm zusammengehängt. Nachdem ein Distanzmaß zwischen Frequenzen vorhanden ist, können diese Sequenzen auch auf Ähnlichkeiten überprüft werden. Mittels eines zeitlichen Fensters der Länge T, welches über das gesamte Video und somit auch über alle anderen Subsequezen der gleichen Länge geschoben wird, können diese miteinander verglichen werden. Hierbei entsteht eine Ähnlichkeitsmatrix, welche geclustert werden kann, wodurch wiederum die Sequenzen geclustert werden. [30]

Abbildung 9 zeigt Ausschnitte eines einzelnen Videoclips bei der eine Person mittels statischer Kamera beim Basketball aufgenommen wurde. Eine kurze Szene von einem Wurf auf den Basketballkorb wurde mit allen anderen Subsequenzen verglichen. Niedrigere Werte zeigen eine Ähnlichkeit zwischen den Subsequenzen an. Mit Hilfe eines Schwellwertes können somit gleiche Aktivitäten im Video gefunden werden. [30]

In der Arbeiten [30] geht es ebenfalls darum Ereignisse in Videoaufnahmen wiederzuerkennen, mit dem Unterschied, dass Merkmale nicht von einzelnen Objekten extrahiert werden, sondern der ganze Bereich eines Video-

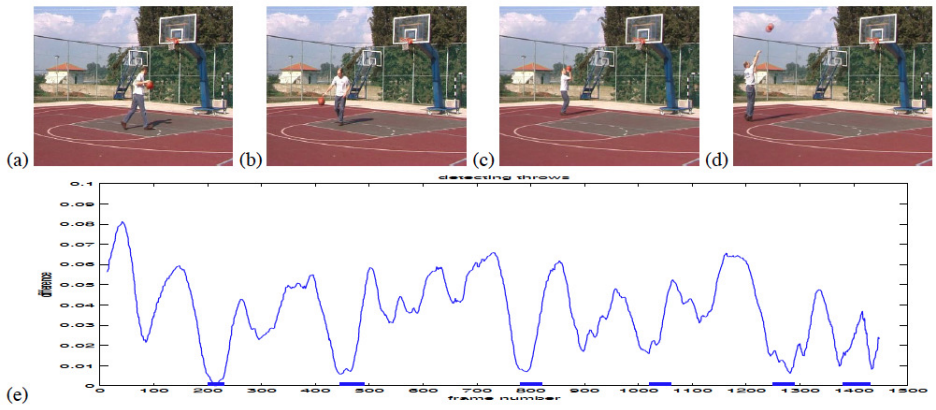


Abbildung 9: a) - d) Videofilmes aus einer Basketball Videosequenz. e) Distanzmaße zwischen einer Subsequenz und allen anderen Subsequenzen. (Bild von [30]).

Frames dafür verwendet wird. Dadurch ist es schwierig Ereignisse anhand von Bewegungen und Aussehen der Objekte festzumachen. Wenn man hingegen jede Bewegung im Bild als möglichen Objektkandidaten sieht und aus jeder Bewegung Merkmale ausliest, kann jeder Bewegungsablauf der Objekte mit allen anderen verglichen und auf Ähnlichkeit überprüft werden.

3.2 Energiedifferenzen zur Ereigniserkennung

Ziel der Methode aus [6] ist es, Ereignisse in Überwachungsvideos, die hochwertige Informationen enthalten, zu identifizieren und zu clustern. Der Grundgedanke der Methode ist, dass durch Einteilung in statische und dynamische Ereignisse, ein Überwachungsvideo schneller durchgesehen werden kann. Ereignisse werden über Energiedifferenzen zwischen Framebildern erkannt. Sind alle Ereignisse bestimmt worden, werden Merkmale extrahiert und anschließend geclustert. Anhand der Clusterstruktur wird dem Nutzer eine Zusammenfassung präsentiert. Damit dieses System funktioniert, muss das importierte Videomaterial über eine statische Kamera erstellt worden sein. Hierfür werden Überwachungsvideos aus einem Parkhaus verwendet.

In [6] werden Bewegungen mittels Differenzbildern erkannt. Hierzu wird ein festgelegtes und sich aktualisierendes Referenzbild von jedem Frame des Videos subtrahiert. Daraus erhält man ausschließlich Regionen, in denen

sich optisch etwas verändert hat. Aus dem Differenzbild kann nun ein Energiewert kalkuliert werden. Dieser Energiewert wird mittels arithmetischen Mittel aus allen Intensitätswerten des Differenzbildes errechnet. In einem nächsten Schritt werden jene Frames extrahiert und in Menge I (Frames von Interesse) gesammelt, deren Energiewert einen gewissen Schwellwert überschreitet und somit als Eventkandidat festgelegt wird. Als Referenzbild für die Differenzbildgenerierung wird immer jenes Frame genützt, dessen zugehöriger Energiewert diesen Schwellwert als letztes überschritten hat. Aus dieser Menge I werden nun jene Frames zusammengefasst, die nebeneinander liegen und deren Distanz $d(x_i, x_{i+1})$ kleiner als ein fixer Schwellwert ist. Die daraus resultierenden Framemengen S bestehen nun aus Eventkandidaten, die das folgende Kriterium erfüllen müssen um als Event klassifiziert zu werden. [6]

$$\frac{d^I(S_i')}{d^F(S_i')}, > d_{tr} \tag{7}$$

Diese zwei Längenmaße ergeben sich aus der Anzahl der Frames des gesamten Framesets des Videos zwischen erstem und letztem Frame des Eventkandidaten S_i und der Anzahl der Frames des Eventkandidaten im Frameset I. In Relation gesetzt muss dieser Wert wieder einen Schwellwert übersteigen um als Event zu gelten. Das bedeutet, je mehr Frames sich im Video zwischen zwei benachbarten Frames von Interesse befinden, umso geringer ist die Wahrscheinlichkeit, dass es als Event kategorisiert wird. [6]

Nachdem alle Ereignisse die von Interesse sind bestimmt wurden, werden aus jedem Eventframe Merkmale extrahiert. Dafür werden MPEG-7 Low-Level Merkmale genützt. Die Merkmale, Farbe, Textur und Kantenhistogramm, werden nach ihrer Relevanz gewichtet und dienen zur Berechnung eines Distanzmaßes zwischen zwei Frames. Wird von jedem möglichen Framepaar das Distanzmaß berechnet, so entsteht eine Ähnlichkeitsmatrix. Diese Matrix wird anschießend einem Spectral Clustering [25] unterzogen. Aus jedem Cluster werden Bilder herausgelesen, welche diesen repräsentieren, um

ein statisches Durchsehen zu ermöglichen. Weiters werden Videosequenzen, die einen Cluster repräsentieren, zur dynamischen Ausgabe genützt. [6]

Bei der Methode in [6] wird auch wieder, wie bei [30], die Veränderung des ganzen Bildes betrachtet und somit findet auch kein Clustering der Objekte, sondern der Frames statt. Low-Level Merkmale werden hingegen direkt aus einem MPEG-7 Video extrahiert, um eine Näherung der Bewegung zu erhalten.

3.3 Erkennung expliziter Ereignisse und Anomalien

In der Arbeit von Piciarelli [21] wird zwischen zwei Arten der Ereigniserkennung in Überwachungsvideos unterschieden: der Erkennung expliziter Ereignisse und der Erkennung von Anomalien.

Beide Methoden verwenden als ersten Schritt eine Objekterkennung, die über eine Hintergrundsubtraktion erreicht wird. Der Unterschied zu anderen Systemen und unserer Methode ist, dass ein Hintergrundbild auch aus schwenk-, zoom- und neigbaren Kameras generiert werden kann. Eine solche Kamera ist zwar an einem fixen Standpunkt befestigt, kann aber horizontal und vertikal geneigt werden, sowie den Bildausschnitt verkleinern und vergrößern. Es wird hierbei ein sogenanntes Hintergrundmosaik mit Hilfe von Überlagerungen und Matching-Points erstellt. Dadurch ergibt sich der Vorteil, dass ein größerer Bereich überwacht werden kann. [21]

Explizite Ereignisse werden durch Messwerte von Verknüpfungen zwischen Objekten erkannt. So können aus Folgen von kleineren Ereignissen, komplexere Ereignisse beschrieben werden. Sind die Atomic Events aus einer Überwachung eines Parkplatzes bekannt, wie „Fahrzeug erscheint“, „Fahrzeug stoppt“ und „Person erscheint neben Fahrzeug“, kann daraus ein komplexeres Ereignis, wie „Fahrzeug wird geparkt“ beschrieben werden. Hierfür ist jedoch ein a priori Wissen über die erkennbaren Ereignisse notwendig.

Nachdem ein Hintergrund von dem aktuellen Frame subtrahiert wurde, bleiben nur mehr Veränderungen zum Hintergrundbild übrig. Dieses Differenzbild gibt an, in welchem Bereich eine Bewegung stattgefunden hat.

Daraus kann man ganze Bewegungsfolgen eines Objektes ermitteln. Weiters werden „Low-Level Features“ ausgelesen. Diese Merkmale sind zum Beispiel der Standort, die Geschwindigkeit oder die Form des Objektes und werden später in der „High-Level Event“ Analyse eingesetzt. Nachdem diese Informationen für die Objekte in einem Ereignis bekannt sind, können auch Beziehungen zwischen Objekten erkannt werden. [21]

Sollen explizite Ereignisse herausgefiltert werden, so kann der Abstand zwischen den einzelnen Objekten gemessen werden. Anhand dieser Beziehungsmuster kann nach ungewöhnlichen Ereignissen gesucht werden. Hierfür wird zwischen „simple events“ und „complex events“ unterschieden. Das erstere ist ein Basisereignis, das bereits bei der Hintergrundsubtraktion erkannt wird. Zweiteres ist ein Ereignis, bei dem mehrere Akteure beteiligt sind, oder welches mit anderen Ereignissen verbunden ist und dadurch auf komplexere Weise beschrieben werden muss. Die „simple events“ werden in Typen eingeteilt wie Erscheinen, Verschwinden, Bewegungsstart, Bewegungsende, sowie Eintritts- und Austrittszone. Ein Akteur wird über seine Lage und einer Liste von „simple events“ beschrieben. Komplexe Ereignisse werden über einen „event matcher“ gebildet. Dies geschieht in drei Schritten. Für jedes „complex event“ werden beteiligte Akteure und andere „complex events“ miteinander verbunden und überprüft, ob diese das Ereignis eingeleitet haben. Danach werden die Variablen mit den Informationen der zuvor gefundenen Verknüpfungen gefüllt. Sind diese Variablen innerhalb eines gewissen Bereiches, werden diese Verknüpfungen als „complex Event“ gespeichert. [21]

Als Beispiel für ein explizites Ereignis wird in Abbildung 10 ein Szenario angeführt, in dem eine Person eine Tasche auf halbem Weg abstellt und eine andere Person diese zu einem späteren Zeitpunkt aufnimmt und anschließend den Beobachtungsbereich mit der Tasche verlässt. Die zwei Personen und die Tasche werden als Akteure erkannt. Durch das „Low-Level Processing“ und den „event matcher“ werden die Distanzen der Akteure zueinander

ausgewertet. Dadurch wird dieses „complex event“ beschrieben und es kann überprüft werden, ob es sich innerhalb gewisser Parameter befindet. [21]

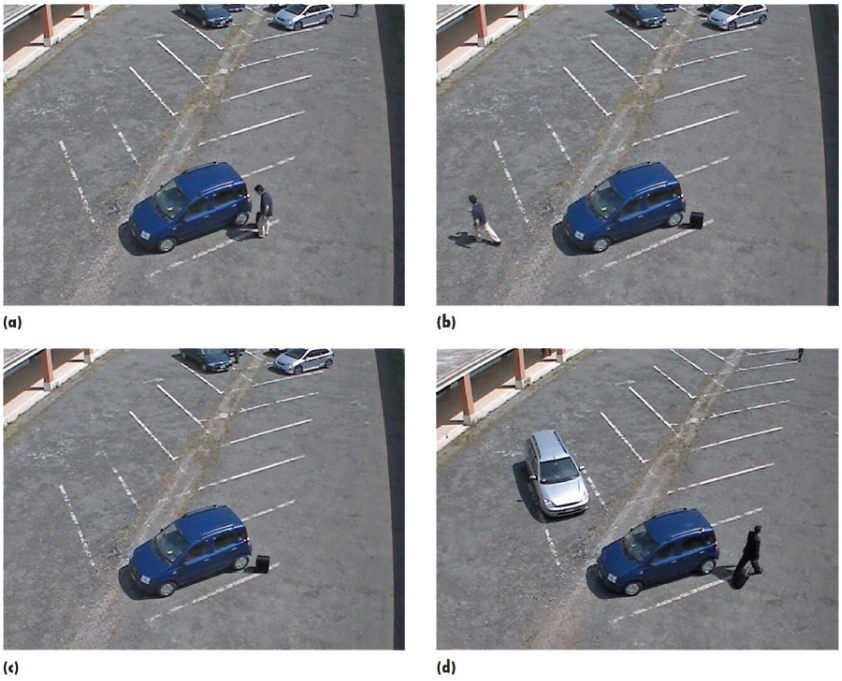


Abbildung 10: Beispielszenario für ein explizites Ereignis, das erkannt werden soll. a) Person stellt eine Tasche in der Nähe eines Autos ab. b) Person verlässt den Beobachtungsbereich. c) Die Tasche bleibt weiterhin neben dem Auto. d) Eine weitere Person nimmt die Tasche auf. (Bild von [21]).

Auch Anomalien können automatisch erkannt werden. Die Grundidee ist, dass nach einem Clustering von Ereignissen jene heraus gefiltert werden, die von den anderen Ereignissen in Aussehen oder Verhalten abweichen. Über ein Clustering werden seltene Ereignisse gefiltert, indem man Ausreißer erkennt. Eine solche Funktion wird oft bei Überwachungssystemen gewünscht. Anomalien können somit leichter bzw. automatisch erkannt werden. Ungewöhnliche Ereignisse definieren sich über ihr seltenes Auftreten und lassen sich daher mittels Clustering und dem Auffinden von Ausreißern gut herausfiltern. Damit dieses System ungewöhnliche Ereignisse in Videoaufnahmen erkennen kann, benötigt es keinerlei Information, wie gewöhnliche oder ungewöhnliche Ereignisse aussehen. [21]

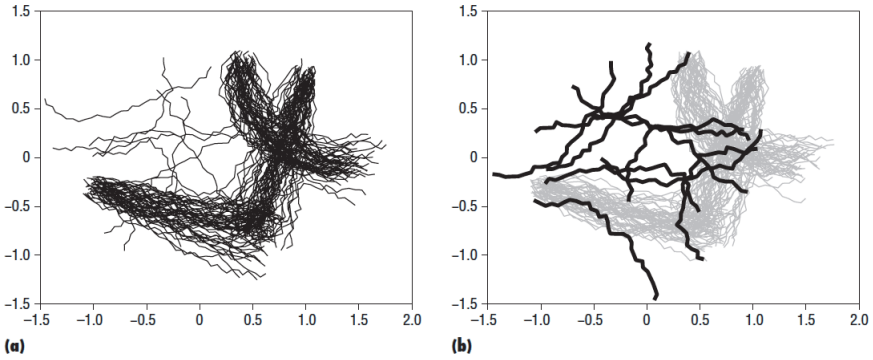


Abbildung 11: Trajektorien von Objektbewegungen und Erkennung von Ausreißern (Bild von [21]).

In Abbildung 11 sind verschiedene Bewegungspfade (Trajektorien) von Objekten eingezeichnet, wobei die dicken Linien Ausreißer im Clustering anzeigen. Als Clustermethode wurde eine Support Vector Machine (SVM) genützt. Diese ist aber nicht für Anomalienerkennung geeignet, da diese zuvor mit Daten trainiert werden muss. Damit die Daten jedoch geclustert werden können, wird eine single-class SVM genützt. Diese findet die größte Gruppe von Elementen im Datensatz und lässt sich dahingehend anpassen, sodass auch Ausreißer erkannt werden. Um die verschieden langen Trajektorien miteinander vergleichen zu können, müssen diese auf ein einheitliches Maß gebracht werden. Hierfür werden oft String Kernels [16], wie in der Textverarbeitung und Bioinformatik genutzt. Das in diesem Kapitel beschriebene System verwendet jedoch ein Subsampling, das alle Trajektorien auf gleiche Länge normalisiert. Nach dem Clustering befinden sich die stark von den anderen abweichenden Trajektorien ausserhalb des Clusters. Somit werden die zugehörigen Ereignisse als Anomalien erkannt. [21]

Als Problem wird das Bemerken und Festlegen eines Ereignisses gesehen, da ein Ereignis in der realen Welt in vielen verschiedenen Formen auftreten kann. Auch bei der Objekterkennung wurden Mängel festgestellt, da Objektbewegungen manchmal nicht durchgehend erkannt wurden, weil kurze Zeit keine Bewegung festgestellt werden konnte. Die größte Aufgabe wird

darin gesehen, die Flexibilität der „anomaly detection“ mit der Genauigkeit von „explicit-recognition“ zu verbinden. [21]

Die Erkennung von Anomalien arbeitet ebenso wie die Merkmalsextraktion unserer Methode mit Low-Level-Features und benötigt kein a priori Wissen über den Inhalt des Videomaterials. In [21] benötigen Explizite Ereignisse im Gegensatz zu den komplexen Ereignissen unserer Methode, Vorwissen über den Inhalt und setzen sich wie in unserer Methode aus einfacheren Ereignissen zusammen.

4 Methodisches Verfahren

Um einen Überblick über die Methode zu bekommen, wird im ersten Unterkapitel die gesamte Methode kurz vorgestellt. In den nächsten Unterkapiteln werden die einzelnen Verarbeitungsschritte der Methode beschrieben. Zunächst wird der Prozess der Bewegungserkennung erläutert, gefolgt von der Beschreibung der Objektidentifizierung und der anschließenden Merkmalsextraktion. Zum Schluss werden die zwei aufeinander aufbauenden Clustering-Methoden beschrieben, welche es ermöglichen „Atomic-Actions“ und komplexe Sequenzen von Atomic-Actions zu erkennen.

4.1 Aufbau der Methode

Die entwickelte Methode zur Erkennung sich wiederholender Ereignisse in Langzeitaufnahmen gliedert sich in zwei große Teilaufgaben: Das Detektieren von Objektbewegungen und Ereignissen in einer Videoaufnahme und das Clustern dieser Ereignisse nach Verhalten und Aussehen.

Hierzu wird jedes Video-Frame nach bewegten Objekten untersucht. Diese Objekte sollen anhand ihres Aussehens, Bewegung und Standort mit Hilfe eines Clusterings eingeteilt werden. Somit können Objekte nicht nur über ihre Merkmale beschrieben werden, sondern vereinfacht über ihre Clusterzugehörigkeit. Daraus ergibt sich ein Alphabet zur Objektbeschreibung. Gleichzeitig werden Objekte über mehrere Frames hinweg verfolgt und bilden Ereignisse. Diese Ereignisse können über eine Folge ihrer Objekte beschrieben werden und somit auch mit Hilfe des zuvor erstellten Alphabets als Zeichenkette. Die erkannten Ereignisse können somit über ein weiteres Clustering auf Ähnlichkeit überprüft werden. Das Ergebnis dieses Clusterings soll die wiederholt auftretenden Ereignisse kompakt dem Benutzer zur Durchsicht zur Verfügung gestellt werden.

Als erster Schritt wird ein Hintergrundbild durch Berechnung eines Mittelwertes von mehreren Frames generiert. Hierzu wird eine Videoaufnahme benötigt, welche über eine statische Kamera erstellt wurde. Durch Subtrak-

tion des gerade zu bearbeitenden Frames und des Hintergrundframes erhält man ein Differenzbild. Wenn nun ein Schwellwert festgelegt wird, bekommt man ein Schwarzweißbild. Weiße Flächen kennzeichnen nun Regionen in denen Bewegung statt findet. Anhand dieser Bewegungen im Bild kann darauf geschlossen werden, dass diese Bewegungen von Objekten hervorgehen.

Jedes gefundene Objekt bekommt eine eindeutige Nummer zugeordnet, um es identifizieren zu können. Hierbei wird eine Label-Maske erstellt, in der jedes Pixel eines Objektes mit einer Identifikationsnummer versehen ist. Damit ein Objekt über die Zeit identifiziert werden kann, wird der optischer Fluss des vorherigen Frames und des aktuellen Frames untersucht. Hiermit kann man errechnen, wo sich das Objekt im vorherigen Frame vermutlich befand. Für die errechnete Region wird anschließend in der Labelmaske des vorherigen Frames nachgesehen, welche ID am häufigsten vorkommt. Dem Objekt wird die entsprechende ID zugewiesen. Somit können Objekte über mehrere Frames verfolgt werden.

Nachdem die Objektbewegungen erfasst und somit Objekte gekennzeichnet wurden, werden Merkmale dieser Objekte berechnet. Hierzu werden immer zwei aufeinanderfolgende Frames zusammen genommen, um nach der Merkmalsextraktion sogenannte Atomic-Actions zu erhalten. Diese Atomic-Actions können Informationen über Bewegungsrichtung, Ort und Aussehen enthalten. Diese Informationen werden aus jedem Framepaar jedes Objektes berechnet. Somit wird ein Objekt durch eine Serie von Merkmalen beschrieben.

Auf Basis der zuvor extrahierten Merkmale werden nun Atomic-Actions einem Clustering unterzogen. Daraus ergibt sich eine neue Form, wie die Objekte beschrieben werden können. Es ergeben sich Action-Labels, die wie ein neues Alphabet die Objektbewegung, das Objektaussehen, sowie den Objektort darstellen können. Wobei je nach Anwendungsfall eine geeignete Auswahl der drei Merkmalskategorien gewählt wird. Dient rein die Bewegungsrichtung als Eingabe für das Clustering, so könnte die Bewegungsrichtung von links nach rechts mit dem Symbol A gekennzeichnet sein

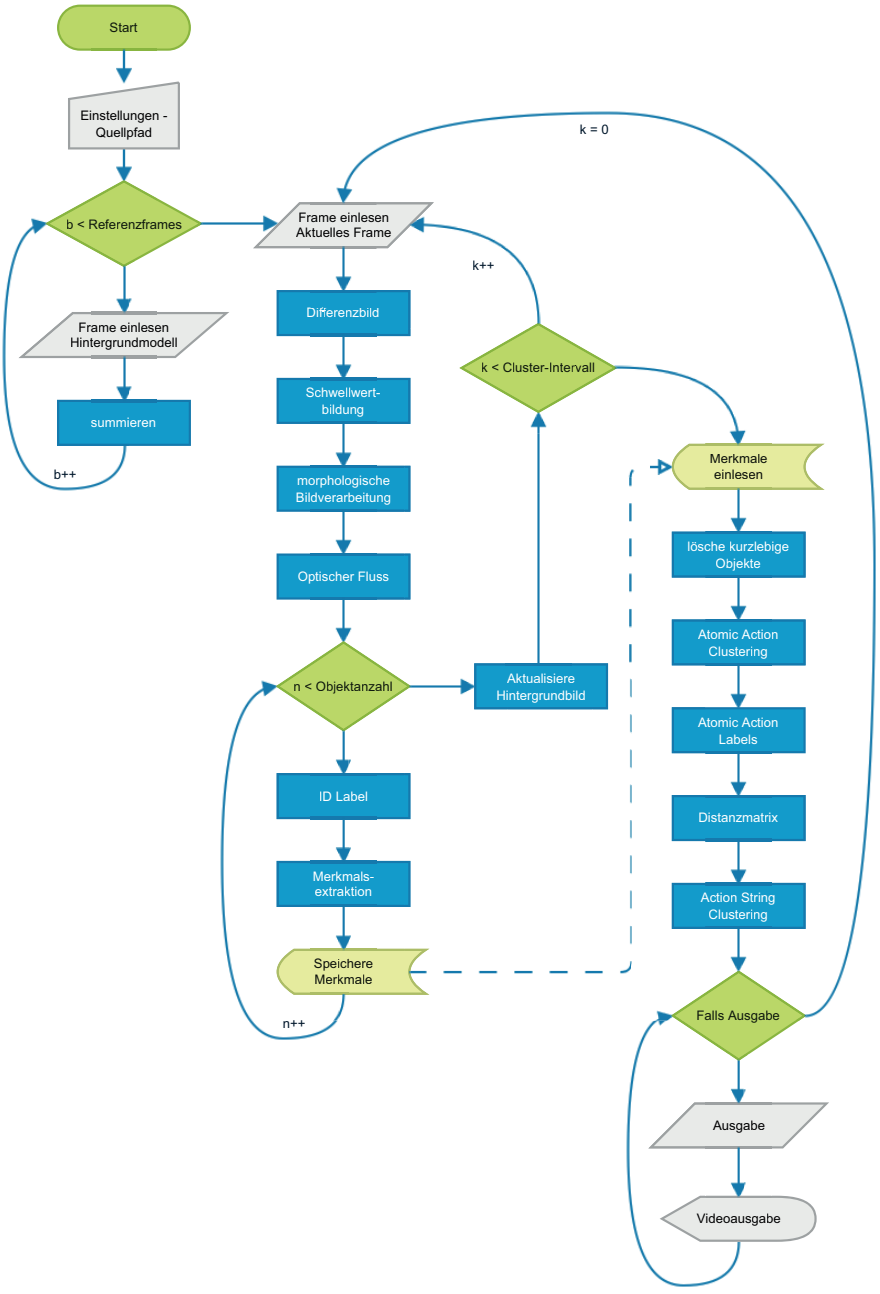


Abbildung 12: Flussdiagramm der hier vorgestellten Methode; b, n und k sind hierbei Zählervariablen der Schleifendurchgänge.

und die Bewegungsrichtung von rechts nach links mit dem Symbol B. Eine Person die nun von links nach rechts und anschließend von rechts nach links durch das Bild läuft, könnte wie folgt als String beschrieben werden: AAAAAAAABBBBBBBB.

Diese Strings können nun untereinander auf Ähnlichkeiten überprüft werden. Da Objekte zeitlich unterschiedlich lang existieren, sind auch ihre Strings unterschiedlich lang. Diese müssen auf ein einheitliches Maß übertragen werden. Hierfür wird die Ähnlichkeit zwischen allen Objekt-Strings errechnet und eine Distanzmatrix erstellt.

Nun erfolgt ein weiteres Clustering um ähnliche Ereignisse zu finden. Wurden alle drei Merkmalskategorien für das Clustering genutzt, beinhalten die entstandenen Cluster nun Ereignisse mit ähnlichem Aussehen, ähnlicher Bewegungsfolge und befinden sich in räumlicher Nähe. Aus der Information können Ereignisse die am häufigsten vorkommen im Video lokalisiert und hervorgehoben werden. Videosequenzen in denen keine wiederkehrende Ereignisse vorkommen, können nun auch ausgeblendet oder verkürzt dargestellt werden.

4.2 Bewegungserkennung

Die hier angewandte Technik um Bewegung zu erkennen ist die Hintergrund Subtraktion. Diese kann auf unterschiedliche Weise erreicht werden. In Kapitel 2.2 werden die Methoden Mittelwertverfahren und Running Average beschrieben, die hier ebenfalls verwendet wurden, da diese wenig Speicher und Rechenleistung in Anspruch nehmen.

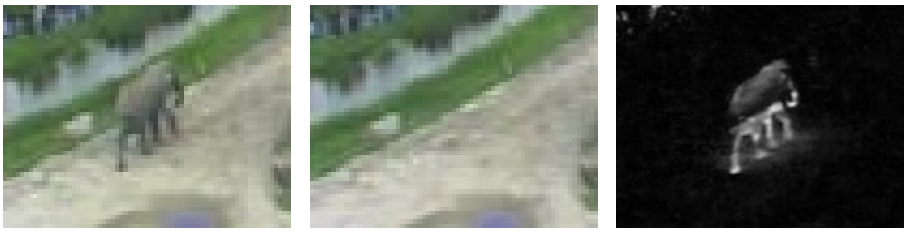
In einer ersten Phase wird eine Initialisierung eingeleitet, in der ein anfängliches Hintergrundmodell erstellt wird. Dies ist notwendig um von Anfang an Aussagen über den Vordergrund machen zu können. Das Hintergrundbild wird erstellt, indem eine Anfangssequenz von n Frames des Videomaterials als Referenz genützt wird. Aus diesen Bildern wird das arithmetische Mittel errechnet und stellt somit das Hintergrundmodell dar.

Nachdem das Hintergrundmodell erstellt wurde, wird jeder Frame des Videomaterials zeitlich nacheinander verarbeitet. Je nach Framerate des zugeführten Videos und der verwendeten Framerate nach der Datenreduktion, ist es möglich das System in Echtzeit während einer Videoaufnahme arbeiten zu lassen. Somit lässt sich zu jedem Zeitpunkt eine Ausgabe der Ergebnisse erstellen.

Mit dem jeweiligen aktuell bearbeiteten Bild der Videoaufnahme und dem Hintergrundbild findet nun eine pixelweise Subtraktion statt. Da es sich um Farbbilder handelt, muss die Distanz der drei Farbwerte an jeder Pixelposition errechnet werden. Dies geschieht mittels Euklidischer Distanz, wie in Kapitel 2.2.1 und in „Colorimetry: Understanding the CIE System“ [23] beschrieben. Daraus ergibt sich ein Differenzbild, welches Pixel die am unterschiedlichsten zueinander sind, am hellsten anzeigt (siehe Abbildung 13(c)). Ein zuvor gewählter Schwellwert bestimmt nun, ob ein Pixel zum statischen Hintergrund oder zum bewegten Vordergrund zählt und stellt sich als Schwarzweißbild, auch Schwellwertbild genannt, dar. Weiße Flächen kennzeichnen nun Regionen, in denen Bewegung stattfindet. Da die Kamera statisch ist, kann davon ausgegangen werden, dass diese Regionen Bewegung von Objekten kennzeichnen. Ein Beispiel für ein Schwellwertbild ist in Abbildung 13(d) zu sehen. Durch Beleuchtungsänderungen können jedoch unerwartete Artefakte fälschlich als Objekte erkannt werden.

Das Differenzbild, aber auch das Schwellwertbild, kann mit verschiedensten morphologischen Operationen der Bildverarbeitung bearbeitet werden. Kleinere Strukturen können hiermit entfernt werden, ohne größere Strukturen stark zu verändern. Dadurch kann der Vordergrund besser erkannt werden und kleinere Veränderungen, die unter Umständen nicht relevant sind, werden nicht in den Vordergrund mit eingerechnet. In Kapitel 2.1 werden hierzu die grundlegenden morphologischen Operationen näher erläutert. Abbildung 13 zeigt Effekte von morphologischen Operationen anhand eines Bildausschnittes einer Tierbeobachtung. Man kann beobachten, dass nach der Schwellwertbildung der Elefant aus mehreren von einander getrennten

Objekten gekennzeichnet wird. Nach dem Anwenden von Opening und Closing sind kleinere Objekte verschwunden und der Elefant wird durch ein Objekt gekennzeichnet. Als letzter Schritt werden noch Löcher innerhalb des Objektes gefüllt.



(a) Bildausschnitt Elefant (b) Hintergrundbild (c) Differenzbild

(d) Schwellwertbild (e) Opening und Closing (f) Füllen von Löchern

Abbildung 13: Effekte von morphologischen Operationen; (a) Bildausschnitt eines Elefanten; (b) Bildausschnitt des Hintergrundbildes; (c) Differnezbild von Ausschnitt mit Elefant und Hintergrund; (d) Schwellwertbild des Differenzbildes; (e) Binärbild nach Anwendung der morphologischen Filter Opening und Closing; (f) Binärbild nach dem Füllen von Löchern.

Um das Hintergrundbild robust gegenüber Veränderungen der Umgebung zu halten, wird jedes Bild, das bearbeitet wird, in das Hintergrundmodell miteingebracht. Dies wird mittels Running Average realisiert. Hierbei wird jedes Bild zu einem gewissen Bruchteil zum Hintergrundbild hinzugerechnet. Es gibt auch die Möglichkeit nur Pixel des aktuellen Bildes ins Hintergrundbild aufzunehmen, die nicht als Vordergrundbild identifiziert wurden. Dadurch wird das Hintergrundmodell stabiler. Jedoch werden Objekte, die sich zuerst bewegt haben und dann still stehen, nie in das Hintergrundmodell mit aufgenommen. Das kann problematisch werden, wenn Objekte

sich in das Bild bewegen und für längere Zeit stehen bleiben, wie bei dauerhaften Veränderungen der Umgebung durch bauliche Maßnahmen oder einparkende Autos. Für die hier vorgestellte Methode wird das Vordergrundbild nur zu einem geringen Anteil in das Hintergrundmodell aufgenommen, um eine bessere Objekterkennung und Objektidentifizierung zu erreichen.

4.3 Objektidentifizierung

Objekte sind die Bausteine eines Ereignisses. Im Folgenden sollen Objekte und ihre Bewegungen erkannt werden. Dabei kann ein Objekt auch eine Gruppe von Akteuren darstellen. Das kann an dieser Stelle des Algorithmus nicht unterschieden werden.

Nachdem eine Hintergrundsubtraktion und ein Schwellwertbild erstellt wurde, wird jede zusammenhängende Ansammlung von Vordergrundpixeln als ein einzelnes Objekt betrachtet. Jedes dieser Objekte bekommt eine eindeutige Nummer zugeordnet, um es identifizieren zu können. Hierbei wird eine Label-Maske erstellt, in der jedes Pixel eines Objektes mit einer Identifikationsnummer versehen ist. Damit ein Objekt über die Zeit, d.h. über mehrere Frames hinweg, identifiziert werden kann, wird der optische Fluss vom aktuellen Frame zum vorherigen Frame untersucht.

Der optische Fluss vergleicht zwei Bilder miteinander und beschreibt durch einen Vektor an jeder Pixelposition die Verschiebungen der Strukturen in diesen Bildern. Damit lässt sich die Bewegungsrichtung und Bewegungsgeschwindigkeit von sich bewegenden Objekten in zwei aufeinander folgenden Bildern einer Videosequenz beschreiben (siehe Abbildung 14). Dabei wird jeder Wert des optischen Flusses an den Pixelpositionen des Objektes zur Schätzung des vorherigen Standortes des Objektes im vorherigen Bild verwendet. Als Eingabebilder kann man zwei aufeinander folgende und unbearbeitete Bilder der Videoquelle nützen, oder aber auch nur den zuvor errechneten Vordergrund dieser zwei Bilder. Hierbei werden alle Pixelpositionen, die zum Hintergrund gehören, auf Null gesetzt. Somit kann eine schnellere Berechnung des optischen Flusses erreicht werden.

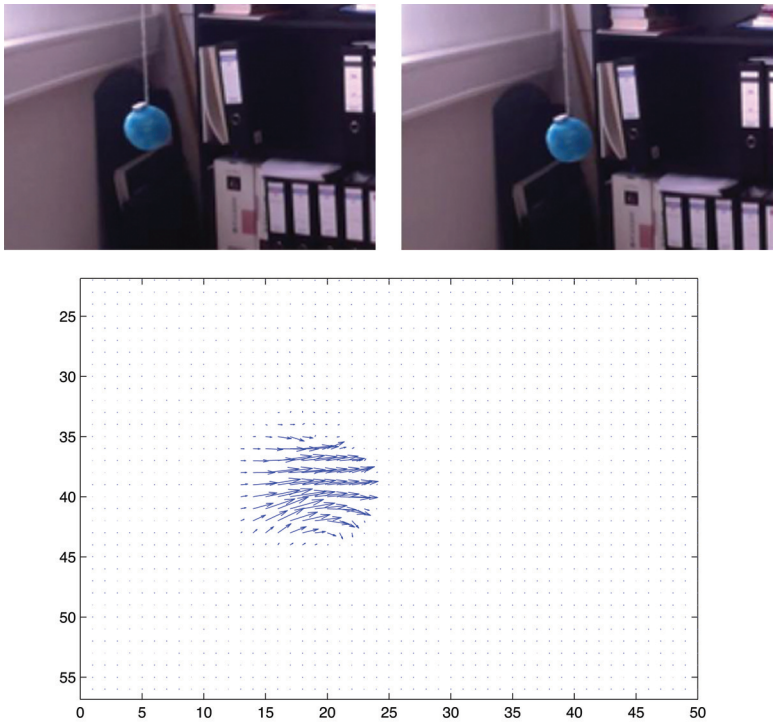


Abbildung 14: Zwei aufeinander folgende Bilder einer Videosequenz und der optische Fluss als Vektorfeld.

Die hier verwendete Methode um den optischen Fluss zu berechnen ist in „Beyond pixels: exploring new representations and applications for motion analysis“[15] von Ce Liu beschrieben und basiert zum größten Teil auf „Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods“[3] und „High accuracy optical flow estimation based on a theory for warping“[2]. Die Implementierung ist veröffentlicht unter [14].

Nach der Berechnung des optischen Flusses kann abgeschätzt werden, von wo sich das Objekt wegbewegt hat. In dieser vermuteten Region wird anschließend in der Label-Maske des vorherigen Frames nachgesehen, welche Identifikationsnummer (ID) in der vermuteten Region am häufigsten vorkommt und diese dem Objekt zugewiesen. Dadurch können Pixelregionen, die als Objekte erkannt wurden, über die Zeit als zusammenhängend erkannt und als ein Objekt gekennzeichnet werden. Wenn sich an diesem

vermuteten Standort keine Einträge in der Label-Maske des vorherigen Frames befinden, dann wird diese Pixelansammlung als neues Objekt erkannt und eine neue ID dem Objekt zugewiesen. Wenn Objekte miteinander verschmelzen, befinden sich mehrere Identifikationsnummern in der vermuteten Region. In diesem Fall wird die am häufigsten vorkommende ID dem Objekt zugewiesen. Nun kann es auch passieren, dass sich ein Objekt teilt. Hier muss entschieden werden, welches Objekt welche ID bekommt. In der hier vorstellten Methode führt das größte Teilobjekt die ID weiter, während die anderen Teilobjekte neue Identitäten bekommen. Eine weitere Möglichkeit ist, dass jedes Teilobjekt neu identifiziert wird.

Auch wenn sich in der vermuteten Region einige Pixel mit Identifikationsnummer befinden, muss das Verhältnis zwischen Pixel mit und ohne ID stimmen. Hierzu wird ein Schwellwert zur Regulierung eingesetzt. Dieser vergleicht die Anzahl der Objektpixel mit Label-Werten (IDs) mit der Anzahl derer ohne Label-Werte. Wenn folgendes zutrifft

$$H_A(0) * w > \sum_{i \in L} H_A(i) \tag{8}$$

wird dem Objekt eine neue ID zugewiesen. Wobei L die Labelmaske mit den Identifikationsnummern in der vermuteten Region A angibt. w dient zur Gewichtung des Verhältnisses. In dieser Methode wurde eine Gewichtung von $w = 2$ gewählt. Es müssen sich daher mehr als doppelt so viele Pixel ohne ID als mit ID in der vermuteten Region befinden, um das Objekt als neu zu deklarieren. Durch eine hohe Gewichtung ist die Wahrscheinlichkeit geringer, dass Trajektorien abreißen, jedoch können Objekte leichter falsch identifiziert werden.

Um die Rechenzeit des optischen Flusses zu reduzieren, kann man die zwei zu untersuchenden Bilder nochmals verkleinern und danach die Ergebniswerte hoch skalieren. Wie bei jeder Datenreduktion leidet hierbei die Genauigkeit der Ergebnisse. Weiters muss man beachten, dass wenn die Framerate zu gering gewählt wurde, Objekte nicht im vorherigen Bild gefunden werden, weil diese örtlich zu weit von einander entfernt sind. Die Methode

des optischen Flusses ist somit nicht mehr in der Lage, die zwei eigentlich zugehörigen Objekte miteinander zu verknüpfen und gibt daher keine Vektoren aus, die vom aktuellen zum vorherigen Objekt-Standort führen. Diese Vektoren führen meist in eine fehlerhaft bestimmte und somit auch leere Region in der Label-Matrix. Die Folge ist, dass das Tracking abreißt und ein neues Objekt mit neuer Identifikationsnummer erstellt wird.

Nachdem jedes Objekt örtlich und zeitlich identifiziert wurde, besitzt man auch die Information, wann jedes einzelne Ereignis anfängt und aufhört zu existieren und an welchem Ort es sich über diese Zeit befand.

4.4 Atomic-Actions

Um Objekte in einer Bildsequenz nach ihrem Aussehen und ihrer Bewegung clustern zu können, müssen verschiedene Parameter aus den Objekten extrahiert werden. Mit Hilfe dieser Parameter lassen sich die Objekte beschreiben und miteinander vergleichen. Ein Ereignis bestehen aus einer Folge von Objekten über mehrere Frames hinweg. Extrahiert man aus jedem einzelnen Objekt eines Ereignisses verschiedenste Parameter, lässt sich dieses Ereignis als eine Aneinanderreihung von Parametern beschreiben. Damit diese Datenmenge noch in Echtzeit geclustert werden kann, muss diese aber verringert werden. Daher erfolgt vor dem endgültigen Clustering der Ereignisse, ein Clustering der Bewegung, des Ortes und des Aussehens der Objekte.

In der Arbeit [21] bestehen komplexe Ereignisse aus einer Kombination von mehreren Atomic-Events, welche grundlegende kurze Ereignisse beschreiben. Dieser Ansatz wurde für unsere Methode ebenfalls aufgegriffen. Um die Beschreibung von Objektbewegung, -position und -aussehen möglichst elementar zu gestalten, werden Objektpaare zum Zeitpunkt t und $t-1$ gebildet. Für jedes Paar wird ein Merkmalsvektor, der Ort, Richtung und Aussehen darstellt, berechnet und beschreibt somit eine Atomic-Action. Die Positionen und das Aussehen der beiden Objekte einer Atomic-Action werden hierbei gemittelt. Bei der Objektidentifizierung werden bereits nützliche Daten errechnet, die in den Merkmalsvektor gespeichert wer-

den können. Daher werden die Merkmale der Objekte auch gleichzeitig mit der Objektidentifizierung extrahiert.

Folgende Merkmale werden bei dieser Methode extrahiert:

- Objektzentrum

  Die Position eines Objektes wird zweidimensional über die X-Achse und Y-Achse beschrieben. Hierzu wird das Objektzentrum berechnet, indem der geometrische Schwerpunkt errechnet wird. Dieser entspricht dem arithmetischen Mittel aller Pixelkoordinaten x und y eines Objektes. Die Ergebniswerte s_x und s_y sind die kartesischen Koordinaten des Objektzentrums der Objektregion A.

  $$s_x = \frac{\sum x}{A} \qquad s_y = \frac{\sum y}{A} \tag{9}$$

  In dieser Methode werden bei der Merkmalsextraktion alle Werte normiert, sodass sie im Intervall (0, 1) liegen. Daher werden die Ergebniswerte s_x mit der Framebreite und s_y mit der Framehöhe des Gesamtbildes dividiert.

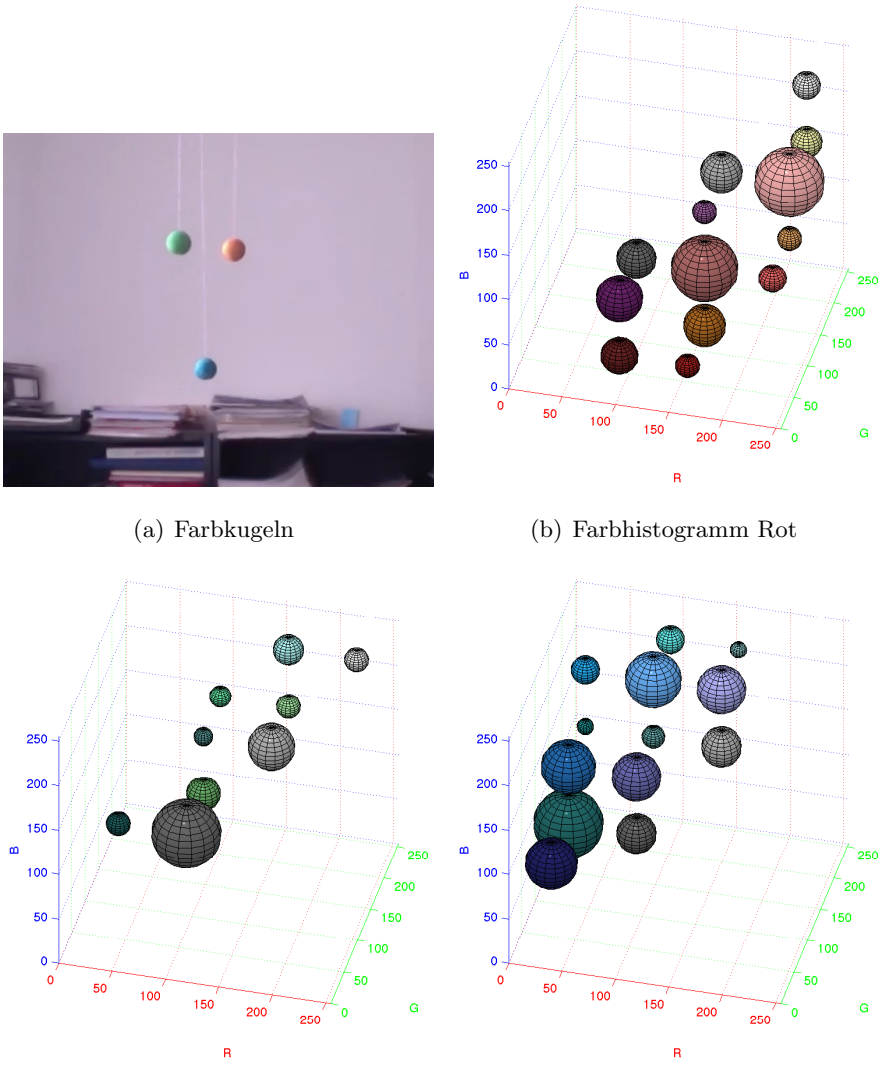
- Objektrichtung

  Zur Objektverfolgung wurde der optische Fluss des Bildes bzw. des Vordergrundes errechnet. Für jedes Pixel eines Objektes ist der optische Fluss als Richtungsvektor vorhanden. Aus allen Richtungsvektoren $\vec{v}$ der Pixel eines Objektes wird das arithmetische Mittel $\vec{a}$ errechnet und als Objektrichtung festgelegt. Damit man nur die Richtung ohne der Geschwindigkeit erhält, wird der Ergebnisvektor mit seiner Länge $|\vec{a}|$ dividiert und somit gleichzeitig auf die Länge 1 normiert.

  $$\vec{a} = \frac{\sum \vec{v}}{A} \qquad Normiert : \frac{\vec{a}}{|\vec{a}|} \tag{10}$$

- Farbhistogram

  Um das Aussehen eines Objektes zu beschreiben wird ein Farbhistgramm aus den Pixelfarbwerten der Objektregion errechnet. Anhand der Abbildung 15 kann man erkennen, dass die Häufigkeiten der Farbwerte in einen Farbwürfel eingezeichnet werden.



(a) Farbkugeln (b) Farbhistogramm Rot

(c) Farbhistogramm Grün (d) Farbhistogramm Blau

Abbildung 15: (a) Bild von drei kugelförmigen Objekten, die unterschiedliche Farbhistogramme aufweisen. (b)-(d) Einzelne Farbhistogramme der drei Kugeln (Bilder erstellt mit [29]).

Dieser Farbwürfel besteht aus drei Dimensionen mit den Achsen für Rot, Grün und Blau, wobei der Wertebereich auf 4 Werte pro Achse festgelegt wurde. Somit ergeben sich 64 Basiswerte. Der Farbwert jedes Pixels eines Objektes wird einem Basis-Farbwert, der ihm im Farbwürfel am nächsten ist, zugeteilt. Als Ergebnis erhält man die Häufigkeiten der Basis-Farbwerte und somit ein Farbhistogramm. Die Werte werden ebenfalls auf 0 bis 1 normiert und vom dreidimensionalen Raum auf einen eindimensionalen Raum verschoben, um diese in den Merkmalsvektor schreiben zu können.

4.5 Clustering von Atomic-Actions

Der Merkmalsvektor besteht aus den zuvor errechneten Parametern, wie Position, Bewegungsrichtung und einem Farbhistogramm der einzelnen Atomic-Actions. Zusätzlich werden die Identifikationsnummer des Objektes und die laufende Framenummer zu jedem Objekt gespeichert, um nach einem Clustering das Objekt bzw. Ereignis im Videomaterial wiederzufinden und zu markieren. Dadurch lassen sich nach dem Clustering Videos der einzelnen Cluster ausgeben.

Die Merkmalsvektoren werden einem Mean Shift Clustering unterzogen um die Atomic-Actions nach ihrer Bewegung, ihrem Ort und Aussehen zu sortieren. Dabei ergeben sich verschiedene Cluster, in denen sich jene Atomic-Actions befinden, die ähnlich sind. Die Clusteringmethode Mean Shift wurde für dieses Verfahren gewählt, da diese keine Information über die erwartete Clusteranzahl benötigt. [11]

Nachdem jeder Atomic-Action eine Cluster-ID zugewiesen wurde, wird zu jedem Videoframe eine Label-Matrix erstellt, in der jede Objektregion mit der zugehörigen Cluster-ID markiert wird. Da Objekte und somit auch Atomic-Actions durch Motiontracking miteinander verbunden sind, ergibt sich für eine Folge von zusammenhängenden Atomic-Actions, eine Label-Zeichenkette. Somit werden Ereignisse als Action-Strings beschrieben.

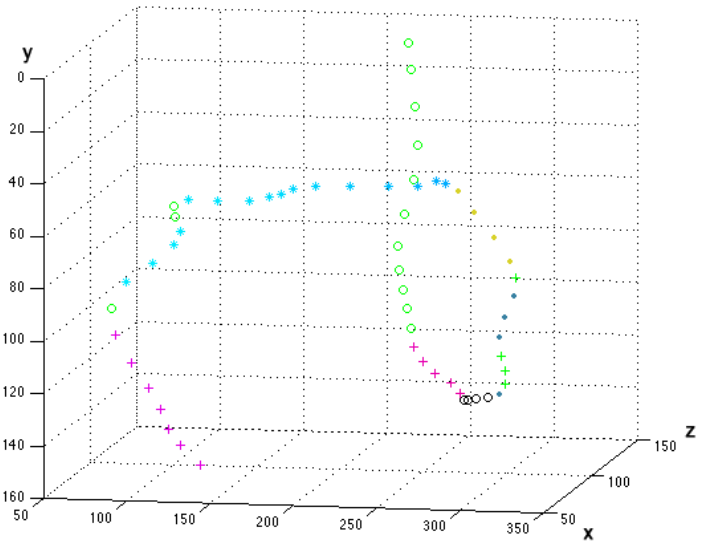


Abbildung 16: Action-String eines Ereignisses nach dem Atomic-Action Clustering, geclustert nach Bewegungsrichtung. Die unterschiedlichen Marker bezeichnen unterschiedliche Cluster der Atomic-Actions.

In Abbildung 16 sieht man den Action-String eines Ereignisses. Die X- und Y-Achse zeigen den jeweiligen Standort der Atomic-Action an. Die Z-Achse zeigt die Frame-Nummer an. Für diese Darstellung wurde ein Clustering nur nach Bewegungsrichtung erstellt und die verschiedenen Cluster mit farbigen Symbolen gekennzeichnet. Das Objekt bewegt sich spiralförmig von links unten nach oben und ändert stetig seine Richtung. Eine Bewegung von links nach rechts wird hier mit blauen Sternen eingezeichnet. Eine Bewegung von unten nach oben mit grünen Kreisen.

In Abbildung 17 wurden die Atomic-Actions nach ihrer Position geclustert. Die X- und Y-Achse zeigen den jeweiligen Standort des Objektes im Bild an. Die Cluster wurden mit verschiedenen Farben gekennzeichnet.

In Abbildung 18 sieht man die Richtungen von verschiedenen Atomic-Actions eingetragen. Als Distanz wurde 0.5 gewählt und um +0.5 in Richtung X- und Y-Achse verschoben. Somit sind alle Richtungsparameter im Wertebereich von 0 und 1. Die Farbe zeigt die Clusterzugehörigkeit an.

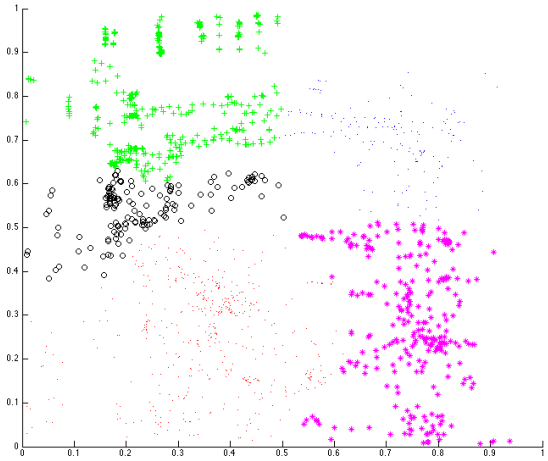


Abbildung 17: Action-Strings nach ihrer Position geclustert.

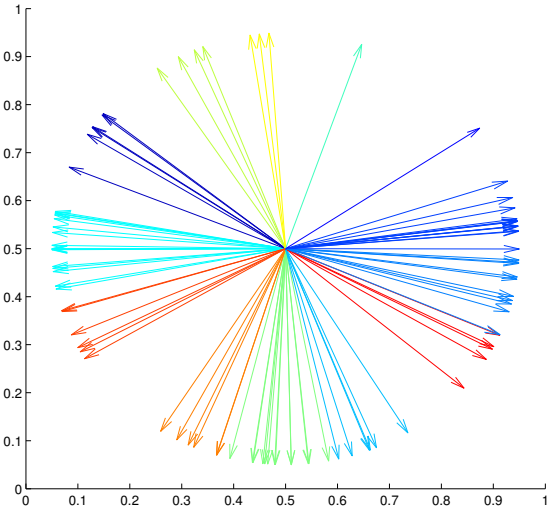


Abbildung 18: Bewegungsrichtungen von Objekten einer kurzen Videosequenz. Das Zentrum der Richtungsvektoren liegt bei [0.5,0.5].

Es kann auch ein Clustering erstellt werden, das nur mit den Farbhistogrammen arbeitet. In Abbildung 19 sieht man die durchschnittlichen Histogramme von drei Clustern. Diese drei Cluster sind das Resultat aus den drei verschieden farbigen kugelförmigen Objekten aus Abbildung 15(a).

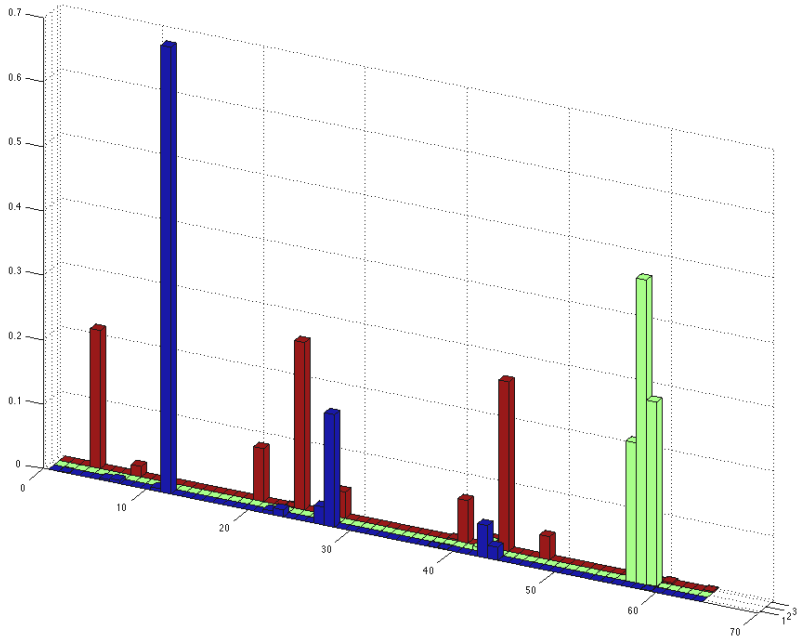


Abbildung 19: Eindimensionales Farbhistogramm der drei Farbkugeln.

Man kann deutlich den Unterschied dieser drei Cluster erkennen. Dass sich die Balken eines Clusters soweit auseinander befinden, obwohl die Objekte beinahe einfärbig sind, liegt an der Abbildung des dreidimensionalen Histogramms in einen eindimensionalen Darstellungsvektor.

4.6 Distanzmatrix

Nachdem jedes Ereignis als Action-String beschrieben wurde, ist es notwendig diese Symbolketten miteinander zu vergleichen, um ein Ähnlichkeitsmaß zwischen den Ereignissen zu finden. Hierfür eignen sich diverse String-Matching Verfahren die im Kapitel 2.6 näher beschrieben werden. Durch das Vergleichen von allen Strings untereinander, entsteht eine Distanzmatrix, welche als Eingabe für ein Clustering dient.

Dynamic-Time-Warping macht es möglich, zeitabhängige Sequenzen mit zeitlichen Verzerrungen und unterschiedlicher Länge miteinander zu vergleichen. [5] Diese Eigenschaft wird hier benötigt, da Objekte unterschiedlich lang existieren und daher ihre Action-Strings ebenfalls von unterschiedlicher Länge sein können. Es wird ebenfalls angenommen, dass Objektbewegungen sich gleichen können, obwohl diese in verschiedenen Geschwindigkeiten passieren.

Beim Dynamic-Time-Warping werden die Längen der zu vergleichenden Strings gemessen, um eine Matrix mit Breite und Höhe der Stringlängen zu erstellen. Mittels einer Kostenfunktion wird jedes Symbol der einen Symbolkette mit den Symbolen der anderen Kette auf Gleichheit überprüft. Diese Werte werden in die Matrix eingesetzt. Nun sucht sich der Algorithmus den kostengünstigsten Weg durch die zuvor erstellte Matrix und durchläuft somit beide Signale vom Ende bis zum Anfang. Dabei werden die Kosten zusammengezählt. Zur Bestimmung der Ähnlichkeit der beiden Symbolketten, kann der Algorithmus die Kostenfunktion sowie die Gesamtkosten in einem Durchgang ohne Rückverfolgung des Pfades ausgeben. Diese Methode hat den Vorteil, dass temporäre Verzögerungen und Beschleunigungen eines Objektes wenig Auswirkung auf die Ähnlichkeit untereinander haben. Dieser Algorithmus betrachtet nur Folgen von unterschiedlichen Symbolen, ohne die Wiederholung von gleichen Symbolen hintereinander zu werten. So ist es möglich, dass sich ein Objekt einmal schneller und einmal langsamer durch das Bild bewegt und beides trotzdem als sehr ähnlich betrachtet wird.

Mit Hilfe einer Doppelschleife wird jeder Action-String mit allen anderen Action-Strings durch Dynamic-Time-Warping verglichen, um so ein Abstandsmaß zwischen den Ereignissen zu errechnen. Bei der Implementierung wurde die Methode Dynamic-Time-Warping [27] von Quan Wang genutzt.

Nach Berechnung der Distanzen ergibt sich für jedes Ereignis ein Vektor. Die Länge des Vektors entspricht der Anzahl der vorkommenden Ereignisse im beobachteten Abschnitt der Videoaufnahme. Die Werte des Vektors beschreiben die Differenzen zu allen anderen Ereignissen und werden für

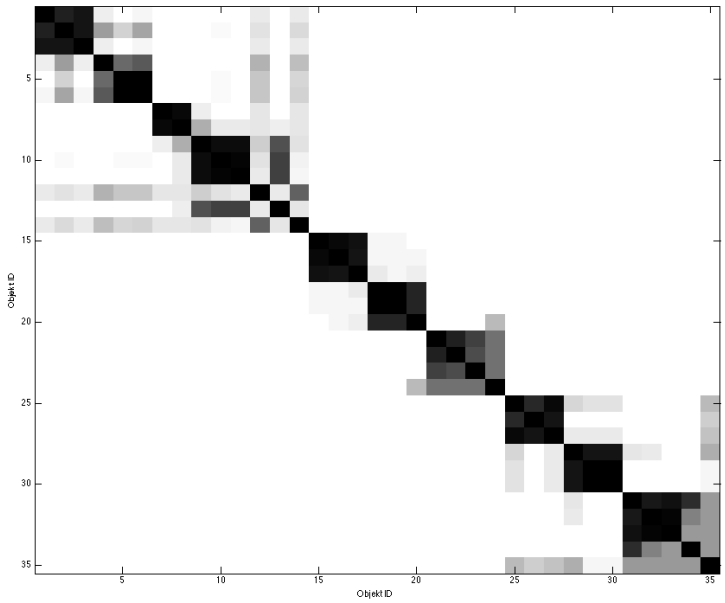


Abbildung 20: Distanzmatrix vor dem Clustering.

jedes Ereignis errechnet und gemeinsam in einer Matrix zusammengefasst. Daraus ergibt sich eine Distanzmatrix, die alle Abstände der Ereignisse zueinander beinhaltet. Da eine solche Distanzmatrix symmetrisch aufgebaut ist, kann die Rechenzeit um die Hälfte verkürzt werden. Das wird damit erreicht, indem man nur auf einer Hälfte der Matrix rechnet.

In Abbildung 20 wird eine Distanzmatrix von 35 Ereignissen als Grauwertbild dargestellt. Dunkle Regionen zeigen Ähnlichkeiten zwischen zwei Action-Strings an, wobei Zeilennummer und Spaltennummer die Identifikationsnummern aller 35 Ereignisse darstellen.

4.7 Clustering von Action-Strings

Nachdem eine Distanzmatrix erstellt wurde, welche die Ähnlichkeit zwischen unterschiedlichen Action-Strings darstellt, erfolgt ein erneutes Clustering. Die Distanzmatrix dient hierbei dem Mean Shift Algorithmus als Eingabe.

In Abbildung 21 ist eine nach Clustern sortierte Distanzmatrix zu sehen. Jedes Ereignis mit gleicher Clusterzugehörigkeit wurde mit der sel-

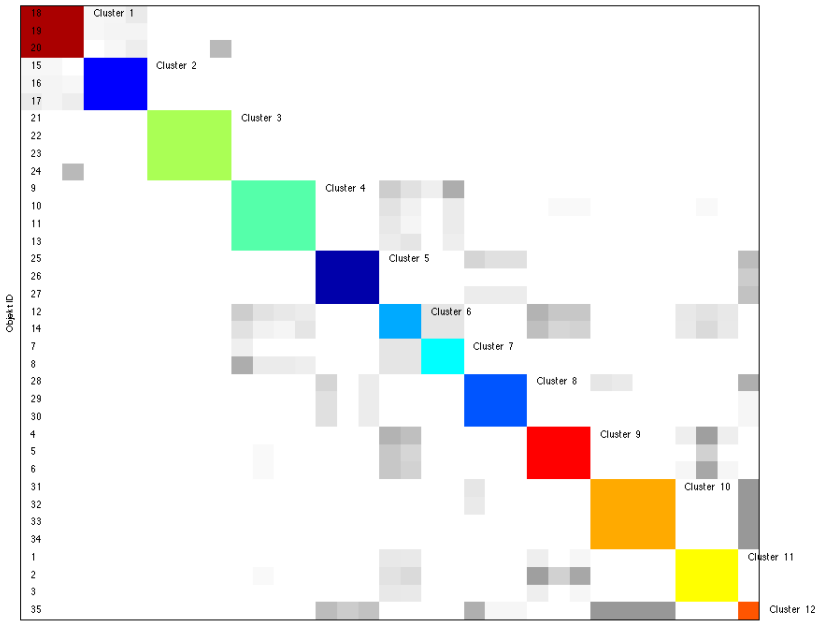


Abbildung 21: Sortierte Distanzmatrix nach dem Clustering nach Ort, Richtung und Farbe.

ben Farbe überdeckt. Man sieht, dass ähnliche Action-Strings Blöcke in der geclusterten Distanzmatrix erzeugen. Das bedeutet, dass ihre Folgen von Atomic-Actions auch ähnlich sind und somit ein ähnliches Aussehen und einen ähnlichen Bewegungsablauf haben, sowie sich in ähnlicher örtlicher Lage befinden.

Um das Clusterergebnis besser zu veranschaulichen, kann man den Ort, die Richtung und die farblich markierte Clusterzugehörigkeit in einer oder mehreren Grafiken ausgeben. Dabei wird jede Atomic-Action als Richtungspfeil eingezeichnet. Atomic-Actions haben nicht nur eine Richtung gespeichert, sondern auch einen Ort. Somit kann der Richtungspfeil an diesem Ort eingezeichnet werden. Diese Pfeile können nun je nach Clusterzugehörigkeit eingefärbt werden. Dadurch erhält man einen guten Überblick, ob das Clustering sinngemäß statt gefunden hat.

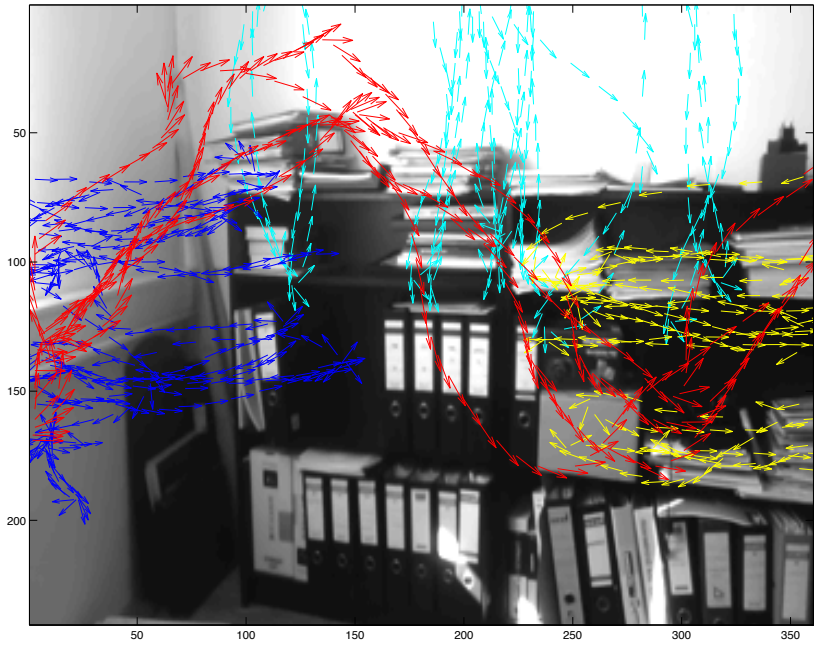


Abbildung 22: Bewegungspfade nach dem Action-String-Clustering nach Ort und Bewegungsrichtung.

In Abbildung 22 wurde ein Clustering nach Bewegungsrichtung und Ort durchgeführt. Dabei wurden Kugeln durch das Kamerabild mit verschiedenen Bewegungsabläufen an verschiedenen Orten im Bildbereich durchgeführt. Man kann eindeutig vier Cluster erkennen, die sich sowohl örtlich, als auch im Bewegungsablauf unterscheiden. Die blau gefärbten Objektpfade zeigen, dass sich Objekte vom linken Bildrand aus, zuerst nach rechts und dann nach links wieder hinaus aus dem Bild bewegten. Gegenteilig verhält es sich mit den gelb gefärbten Pfaden. Objekte die sich von oben herab und wieder hinauf bewegten wurden türkis eingezeichnet. Als vierten Cluster kann man die rot gefärbten Objektpfade erkennen, welche im zickzack das Bild durchlaufen.

Eine weitere Kontrollmöglichkeit ist die Clusterzugehörigkeit nach dem ersten Clustering, dem Atomic-Action-Custering, auszugeben. In Abbildung 23 wurden wieder Ort und Richtung in einer Grafik eingezeichnet. Diesmal wurden aber die Merkmale nicht nach ihren Symbolketten bzw. nicht

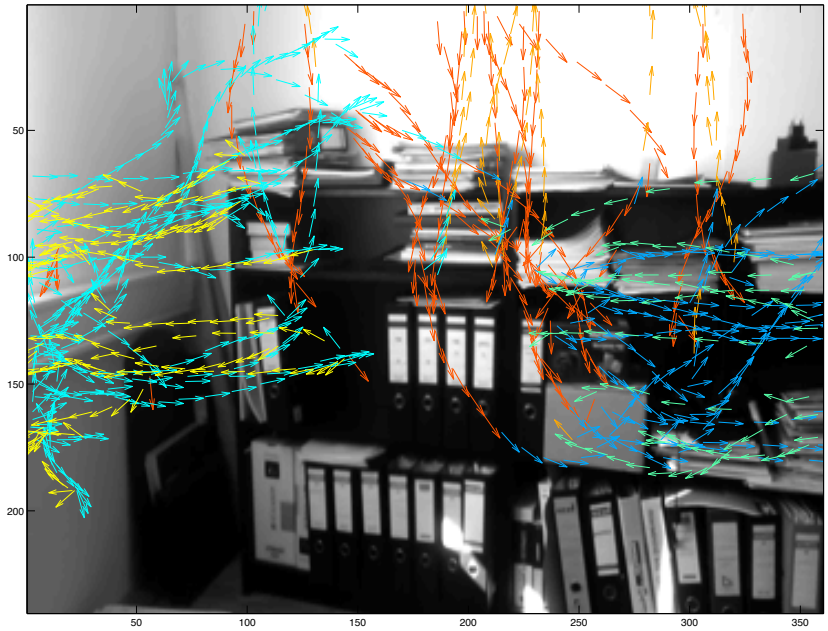


Abbildung 23: Bewegungspfade nach dem Atomic-Action-Clustering nach Ort und Bewegungsrichtung.

nach ihrer Folge von Arten der Bewegungsrichtung und des Ortes eingefärbt, sondern nach ihren Bewegungsrichtungen und Orten der einzelnen Atomic-Actions selbst. Daher sieht man links im Bild, türkise Objektpfade die nach rechts zeigen und gelbe die nach links zeigen. Sieht man sich nun wieder die Grafik aus Abbildung 22 an, sieht man, dass sich alle Objekte, die sich zuerst nach rechts (Abbildung 23 türkis) und dann nach links (Abbildung 23 gelb) bewegt haben in ein und dem selben Ereignistyp (Abbildung 22 blau) befinden.

4.8 Ausgabe sich wiederholender Ereignisse

Als Ergebnis des Action-String Clustering erhält man die Information, welche Action-Strings sich ähneln und sich somit im selben Cluster befinden. Sich wiederholende Ereignisse in einer Videosequenz können am besten über eine graphische Ausgabe visualisiert werden. Dabei bieten sich verschiedens-

te Möglichkeiten an. Eine Möglichkeit ist, ein Video auszugeben in dem nur jene Frames des Eingangsvideos ausgegeben werden, in denen auch Ereignisse vorkommen, die sich wiederholt haben. Die Information, wann ein Ereignis anfängt und wann es aufhört, kann aus den erhobenen Daten der Objektidentifizierung ausgelesen werden. Nimmt man für die Ausgabe nur Frames, in denen sich Objekte dieser Ereignisse befinden, entsteht ein kürzeres Video, das in kürzerer Zeit betrachtet werden kann. Alle anderen Frames werden hierbei vernachlässigt und nicht ausgegeben. Weiters kann man die Information über die Standorte der Objekte mit einfließen lassen. Während der Objektidentifizierung werden auch die Merkmale der Objekte erhoben. So ist es möglich, die Objekte die ein Ereignis auslösen, im Video zu markieren. Eine weitere Möglichkeit ist, entlang einer Zeitleiste die Intensität von Bewegungsaktivität einzuzeichnen. Wird eine Passage in der Zeitleiste markiert, wird diese als Video abgespielt. Zusätzlich werden in der Zeitleiste jene Passagen gekennzeichnet, in denen ähnliche Ereignisse vorkommen.

Abbildung 24: Beobachtung aus dem Tierpark Schönbrunn Wien.

In Abbildung 24 sieht man einen Frameausschnitt einer Beobachtung aus dem Tierpark Schönbrunn in Wien. Hierbei wurde das Elefantengehege mittels einer statischen Kamera gefilmt. Nach der Objektidentifizierung, Merkmalsextraktion, sowie dem hierarchischen Clustering, wurden Objekte, die an wiederholten Ereignissen beteiligt waren, farblich nach ihrem Ereignistyp markiert.

Es bietet sich auch an, für jeden Ereignistyp ein eigenes Video zu erstellen, um die Ereignisse getrennt von einander betrachten zu können. Dabei erzielt man bei der Betrachtung des Videos einen zeitlichen Vorteil, da statische Szenen und Szenen ohne wiederholten Ereignissen wegfallen.

Es besteht auch die Möglichkeit das Video bei Passagen, die nicht als wiederholende Ereignisse erkannt wurden, schneller abzuspielen, indem man die Frameanzahl bei diesen Abschnitten verringert. Passagen mit sich wiederholenden Ereignissen können trotzdem in normaler Geschwindigkeit ausgegeben werden.

5 Evaluierung

Die Evaluierung teilt sich in drei Teile. Als erstes wird eine Proof of Concept Studie durchgeführt. Hierbei wird die grundsätzliche Funktion der Methode überprüft. Hierzu wird eine synthetische Videosequenz genützt, die eindeutige Objekte und Objektbewegungen beinhaltet. Es erfolgt ein Clustering der einzelnen Merkmale und von Kombinationen von Merkmalen.

Der zweite Teil ist eine qualitative Evaluierung eines komplexeren Videos, mit realen Szenen einer Tierbeobachtung mit markanten Objektbewegungen. Abschließend erfolgt eine quantitative Evaluierung, bei der die Laufzeit der einzelnen Komponenten und der gesamten Methode analysiert werden, sowie die Erfolgsrate von erkannten sich wiederholenden Ereignissen.

5.1 Proof of Concept Evaluierung

Um die generelle Funktion der Methode zu überprüfen, wird ein synthetisches Video mit eindeutig unterscheidbaren Objekten und Objektbewegungen verwendet. Hierzu wurden drei farblich unterschiedliche Kugeln durch das Bild bewegt. In Abbildung 25 sieht man die drei unterschiedlich farbigen Kugeln, die als Testobjekte dienen.

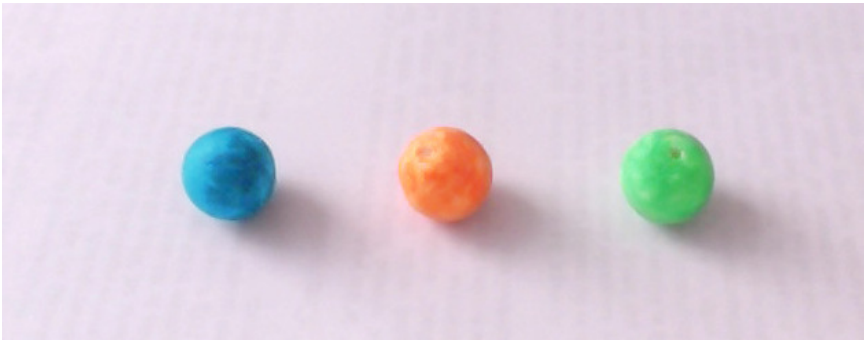


Abbildung 25: Verschieden farbige Kugeln aus der synthetischen Videosequenz.

Für das synthetische Video wurden die Kugeln von rechts nach links, von links nach rechts, von oben nach unten und im zickzack durch das Bild bewegt. Insgesamt beinhaltet das Video 30 Ereignisse. Dabei entstehen chronologisch als erstes 10 Ereignisse der blauen Kugel, danach 10 der orangen und zuletzt 10 der grünen Kugel. Mit jeder Kugel werden dieselben Bewegungsarten ausgeführt. Drei Mal bewegt sich die Kugel vom rechten Rand in das Bild und wieder hinaus. Danach bewegt sich die Kugel drei Mal vom oberen Rand hinein, wandert ein Stück ins Bild und verlässt das Bild wieder mit einer Aufwärtsbewegung. Als nächstes wird die Kugel drei Mal vom linken Rand in das Bild und wieder hinaus bewegt. Bei der vierten Bewegung wird die Kugel in einem Zickzack-Kurs durch das Bild geführt. Die vier Kategorien von Objektbewegungen werden in Abbildung 26 gezeigt. Eine Auflistung der enthaltenen Ereignisse ist in Tabelle 1 zu sehen.

ID	Farbe	Ort/Richtung	ID	Farbe	Ort/Richtung
1	blau	rechts	16	orange	oben mittig
2	blau	rechts	17	orange	links
3	blau	rechts	18	orange	links
4	blau	oben rechts	19	orange	links
5	blau	oben rechts	20	orange	zickzack
6	blau	oben links	21	grün	rechts
7	blau	links	22	grün	rechts
8	blau	links	23	grün	rechts
9	blau	links	24	grün	oben mittig
10	blau	zickzack	25	grün	oben mittig
11	orange	rechts	26	grün	oben mittig
12	orange	rechts	27	grün	links
13	orange	rechts	28	grün	links
14	orange	oben mittig	29	grün	links
15	orange	oben mittig	30	grün	zickzack

Tabelle 1: Auflistung der Ereignisse in der synthetischen Videosequenz.

Die Videosequenz hat eine Framerate von 30 Bildern pro Sekunde, welche für die Verarbeitung auf die Hälfte reduziert wurde. Weiters wurde die Framegröße auf 360 Pixel Breite und 240 Pixel Höhe verkleinert. Die Auflösung für die Berechnung des optischen Flusses wurde nochmals geviertelt.

Nach dem Clustering nach Bewegungsrichtung sollten sich die Objekte in vier verschiedenen Clustern befinden. Wenn der Ort oder die Farbe als Merkmal hinzukommt, sollte sich die Anzahl der Cluster weiter erhöhen. Dies ist abhängig von den Parametereinstellungen des Clusterings, welche auch die Clusteranzahl beeinflussen. Je niedriger die Bandbreite des Clusterings gewählt wird, desto differenzierter wird zwischen Arten von Atomic-Actions bzw. Ereignistypen unterschieden. Für diese Evaluierung wurde beim Atomic-Action-Clustering eine Bandbreite von zirka 0.1, beim Action-String-Clustering eine Bandbreite von zirka 1 gewählt.

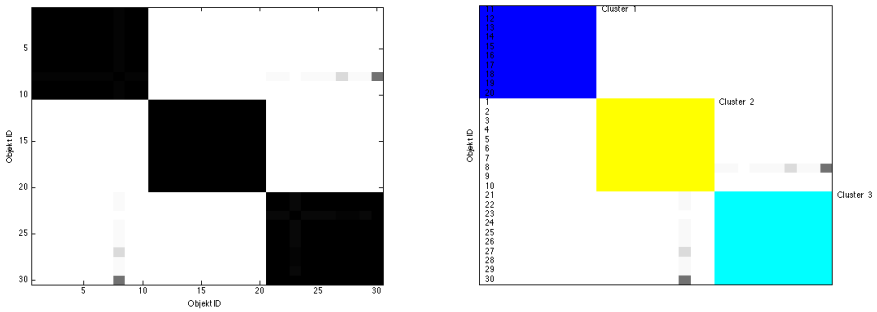
(a) Bewegungspfade (b) Bewegungspfad

Abbildung 26: Erwartete Klassen von Bewegungsabläufen der kugelförmigen Objekte nach einem Clustering nach Bewegungsrichtung.

5.1.1 Clustering nach Aussehen

Das Ergebnis des Clusterings nach Farbe zeigt, dass jedes Objekt richtig erkannt wird. Daraus ergeben sich drei Cluster, in denen sich jeweils die Ereignisse mit dem Objekt mit gleicher Farbe befinden. In Abbildung 27 wird die Distanzmatrix und die nach Cluster sortierte Distanzmatrix der Objekte gezeigt. Es sind deutlich drei große Regionen zu sehen. Daraus kann man ablesen, dass diese Objekte zu ein und demselben Cluster gehören. Auch

die Ausgabe eines Videos, mit nach Clusterzugehörigkeit eingefärbten Objekten, zeigt, dass jedes Objekt wie erwartet erkannt wird. Die Struktur der Distanzmatrix ergibt sich, da zuerst Ereignisse mit dem blauen Testobjekt, anschließend die selben Ereignistypen mit dem orangen und zuletzt mit dem grünen Testobjekt hintereinander zur Erzeugung der Testvideosequenz erstellt wurden.

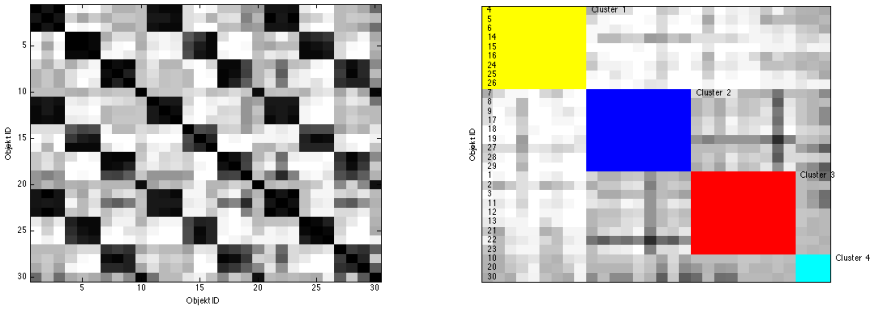


(a) Distanzmatrix (b) Sortierte Distanzmatrix

Abbildung 27: Distanzmatrix vor und nach dem Clustering nach Farbe. Ereignistypen sind nach dem Clustering farblich markiert.

5.1.2 Clustering nach Objektbewegung

Das Clustering nach Objektbewegung liefert ebenfalls das erwartete Ergebnis. Es wurden dabei vier Cluster gebildet, die jeweils die Ereignisse mit den Bewegungen aus Abbildung 26 beinhalten. Zwei ähnliche Ereignisse wurden Anfangs als eigene Cluster erkannt. Je niedriger der Parameter für die Bandbreite des Atomic-Action-Clusterings gewählt wird, desto größer die Anzahl der Cluster. Somit können bei einer zu niedrig gewählten Bandbreite mehr Cluster entstehen als gewünscht. Es wird dadurch detaillierter zwischen Bewegungsrichtungen unterschieden. Die Bandbreite des Action-String-Clusterings gibt an, wie detailliert zwischen ähnlichen Folgen von Bewegungsrichtungen unterschieden werden soll. Die Abbildung 28 zeigt die unsortierte und sortierte Distanzmatrix.



(a) Distanzmatrix (b) Sortierte Distanzmatrix

Abbildung 28: Distanzmatrix nach dem Clustering nach Bewegungsrichtung.

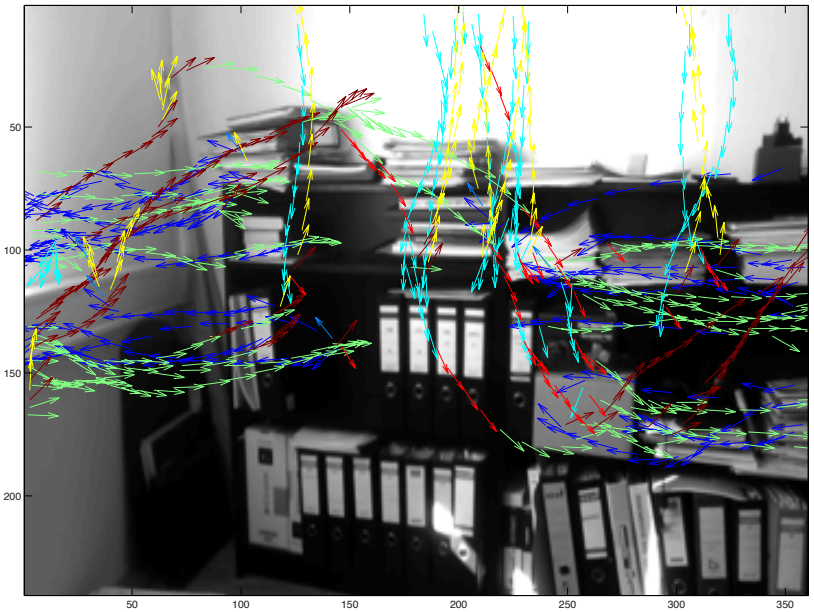


Abbildung 29: Bewegungspfade nach dem Atomic-Action-Clustering nach Bewegungsrichtung.

Als weiteres Analyseinstrument kann man sich die Bewegungsrichtungen der Atomic-Actions als Vektoren im Bild ausgeben lassen. Dabei werden Pfeile in Bewegungsrichtung am Ort der Atomic-Actions eingezeichnet

und somit der Bewegungspfad eines Objektes angezeigt. Diese Vektorpfeile werden je nach Cluster eingefärbt, um die Clusterzugehörigkeit anzuzeigen.

In Abbildung 29 werden die Bewegungspfade nach dem Atomic-Action-Clustering dargestellt. Man kann erkennen, dass Bewegungen nach rechts grün, nach links blau, nach oben gelb, nach unten türkis usw. eingefärbt sind.

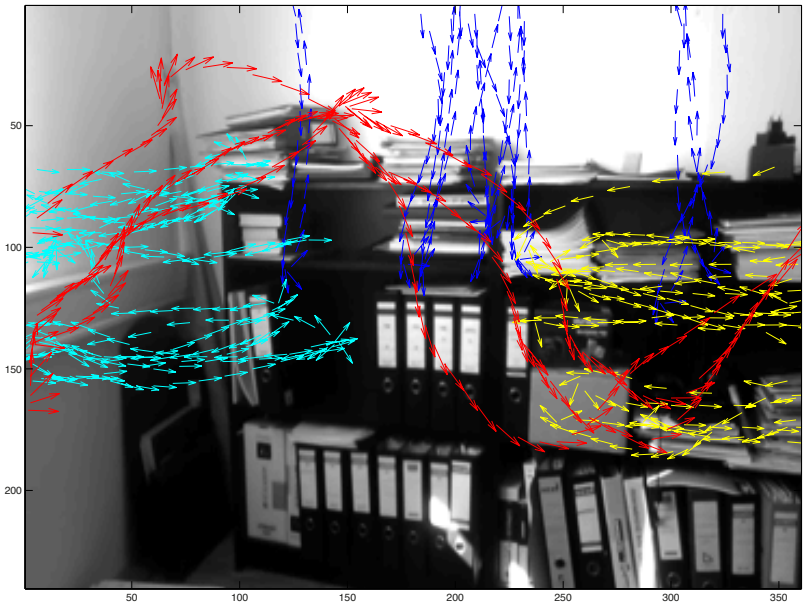
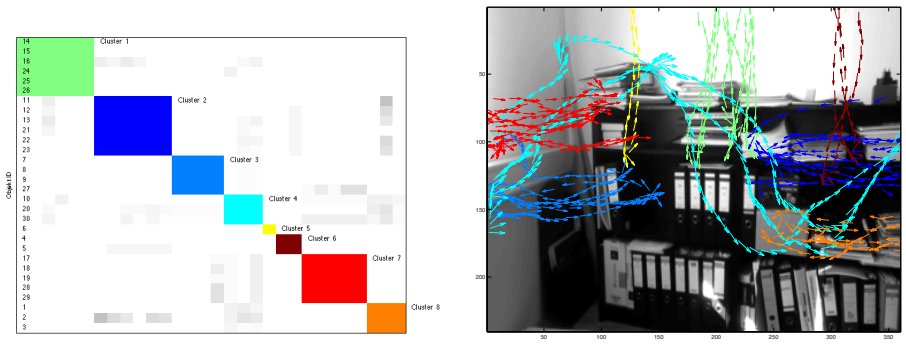


Abbildung 30: Bewegungspfade nach dem Action-String-Clustering nach Bewegungsrichtung.

In Abbildung 30 werden die Bewegungspfade nach dem Action-String-Clustering dargestellt. Vergleicht man diese Grafik mit der von Abbildung 29 kann man sehen, dass ähnliche Folgen von Bewegungsrichtungen im selben Cluster sind. Das bedeutet, dass Objekte nach ihren Folgen von Bewegungsrichtungen beschrieben werden können.

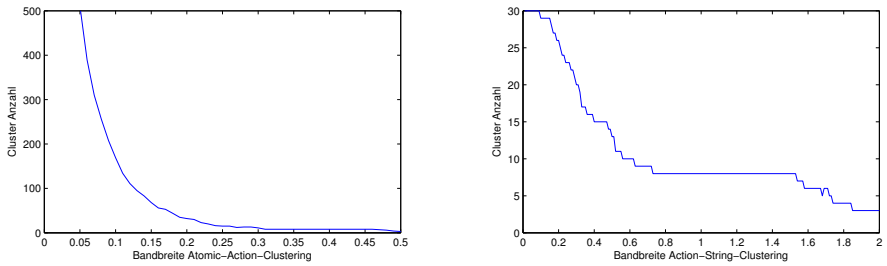
5.1.3 Clustering nach Objektbewegung und Ort

In weiterer Folge wird der Ort als weiteres Merkmal zur Richtung hinzugefügt, wodurch nicht nur zwischen Bewegungsrichtungen unterschieden wird, sondern auch in welchem Teil des Bildes dies passiert. Nach einem Clustering nach Ort und Richtung kann man in Abbildung 31 sehen, dass nun mehr Cluster als zuvor existieren. Man kann nun die örtliche Komponente deutlich erkennen. Je nachdem wie fein das Clustering erfolgen soll, muss dies über die Parameter der Bandbreiten der Clusterings festgelegt werden. Die Abbildung 32 zeigt den Zusammenhang von Bandbreite und Clusteranzahl.



(a) Sortierte Distanzmatrix (b) Bewegungspfade

Abbildung 31: Ergebnisse nach dem Action-String-Clustering nach Bewegungsrichtung und Ort.



(a) Bandbreite Atomic-Action-Clustering (b) Bandbreite Action-String-Clustering

Abbildung 32: Zusammenhang von Bandbreite und Clusteranzahl.

5.1.4 Clustering nach Objektbewegung, Ort und Aussehen

Nun lassen sich die Merkmale Farbe, Richtung und Ort auch zusammen in einem Clustering verwenden. Dadurch entsteht ein größerer Merkmalsraum und ein differenzierteres Clustering kann erfolgen. Wird ein Clustering nach Farbe und Richtung durchgeführt, werden 12 Cluster erwartet, da in der verwendeten Videosequenz zwischen 4 Bewegungsarten und 3 Farben unterschieden wird. In Abbildung 33 ist eine Distanzmatrix zu sehen, in der jede Farbe für einen Cluster steht. Man kann diese mit dem Objektkatalog von Tabelle 1 vergleichen. Die drei markanten Blöcke in der Struktur der Distanzmatrix entstehen durch die drei verschiedenen Farben der Testobjekte. Die Ereignisse mit gleichem Testobjekt ähneln sich trotz unterschiedlicher Bewegungsabläufe, können jedoch voneinander unterschieden werden. Tabelle 2 zeigt die Clusterzugehörigkeit der Objekte. Wird ein Clustering nach Farbe, Richtung und Ort durchgeführt, können sich weitere Cluster ergeben, da die Objektbewegungen einer Bewegungskategorie nicht exakt gleich für diese Videosequenz ausgeführt worden sind und somit der Ort variiert.

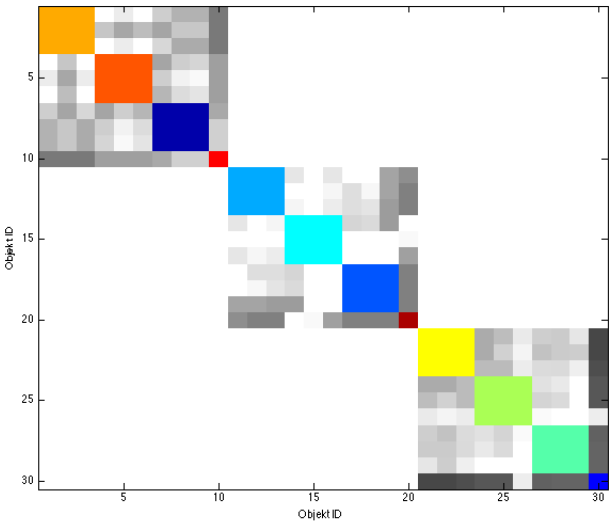


Abbildung 33: Unsortierte Distanzmatrix nach dem Clustering nach Farbe und Bewegungsrichtung.

Cluster	Farbe	Ort/Richtung	Ereignisse	korrekt
1	blau	rechts	1, 2, 3	✓
2	blau	oben	4, 5, 6	✓
3	blau	links	7, 8, 9	✓
4	blau	zickzack	10	✓
5	orange	rechts	11, 12, 13	✓
6	orange	oben	14, 15, 16	✓
7	orange	links	17, 18, 19	✓
8	orange	zickzack	20	✓
9	grün	rechts	21, 22, 23	✓
10	grün	oben	24, 25, 26	✓
11	grün	links	27, 28, 29	✓
12	grün	zickzack	30	✓

Tabelle 2: Ergebnistabelle nach dem Clustering nach Farbe und Bewegungsrichtung.

5.2 Qualitative Evaluierung

Für die qualitative Evaluierung wird ein zusammengeschnittenes Überwachungsvideo von einem Elefantengehege mit markanten Objektbewegungen von Elefanten und Tierpflegern ausgewählt. Ziel ist, dass Elefant von Mensch, sowie nach Ort und nach Bewegungsrichtung unterschieden wird. In Abbildung 34 kann man einen Eindruck der unterschiedlichen Akteure, Orte und Bewegungsrichtungen bekommen.

Für diese Videosequenz wurden verschiedene Szenen gewählt, die sich in Kategorien einteilen lassen. Elefanten und Pfleger gehen von links unten nach rechts oben hinauf, sowie von rechts oben nach links unten hinunter. In der rechten Bildhälfte gehen ebenfalls Elefanten hinauf und hinab, wobei diese kürzere und längere Strecken zurücklegen und öfter nebeneinander gehen. Das führt dazu, dass sich Objekte beim Objekttracking teilen oder Objekte verschmelzen. Es werden noch weitere Cluster erkannt wie zum Beispiel, Fußgänger die am Gehege vorbei gehen und Spiegelungen in Wasser-

(a) Elefant links hinauf

(b) Elefant links hinunter

(c) Elefant rechts hinauf

(d) Elefant rechts hinunter

(e) Pfleger hinauf

(f) Pfleger hinunter

Abbildung 34: Markante Akteure, Orte und Bewegungsrichtungen aus dem geschnittenen Video der Tierbeobachtung.

pfützen. Insgesamt werden 90 Objekte registriert, von denen die wichtigsten in Tabelle 3 angeführt sind. Tabelle 4 zeigt eine Auflistung der wichtigsten Ereignistypen der Tierbeobachtung.

ID	Akteur	Ort/Richtung
2	Elefant	links hinauf
4	Elefant	links hinauf
7	Elefant	oben
11	Elefant	oben
14	Elefant	links hinunter
16	Elefant	links hinunter
23	Elefant	links hinauf
32	Elefant	rechts hinauf - hinunter
35	Pfleger	links hinunter
38	Pfleger	links hinunter
41	Pfleger	links hinauf
46	Elefant	rechts unten
59	Elefanten	rechts unten
60	Pfleger	links hinunter
66	Elefanten	rechts hinauf - hinunter
83	Elefant	rechts hinauf - hinunter

Tabelle 3: Auflistung markanter Ereignisse der Tierbeobachtung.

Die Videosequenz hat eine Framerate von 10 Bildern pro Sekunde, welche für die Verarbeitung auf die Hälfte reduziert wurde. Weiters wurde die Framegröße auf 288 Pixel Breite und 240 Pixel Höhe verkleinert. Die Sequenz hat eine Gesamtlänge von zirka 10 Minuten und wurde aus einer Tierbeobachtung eines Elefantengeheges des Schönbrunner Tiergartens zusammengesetzt. Die Auflösung für die Berechnung des optischen Flusses wurde nochmals geviertelt.

Akteur	Ort/Richtung	Ereignisse ID
Elefant	links hinauf	2, 4, 23
Elefant	links hinunter	14, 16
Elefant	oben	7, 11
Elefant	rechts hinauf - hinunter	32, 66, 83
Elefant	rechts unten	46, 59
Pfleger	links hinauf	41
Pfleger	links hinunter	35, 38, 60

Tabelle 4: Ereignistypen der Tierbeobachtung.

5.2.1 Clustering nach Aussehen

Nach dem Clustering nach Farbhistogrammen, kann man in Abbildung 35 erkennen, dass sich in der linken Bildhälfte die Elefanten (hellblau) von den Pflegern (grün, blau, orange) unterscheiden. Die Pfleger transportieren verschiedene Objekte und tragen unterschiedlich farbige Kleidung. Der Pfleger mit einer grünen Schiebetruhe (blau) wird wiedererkannt. Ein paar Elefanten sind in separaten Clustern vorhanden. Elefanten mit Objekt-ID 66 wurden nicht wie erwartet erkannt, sondern sind mit Pfleger ID 38 in einem Cluster (grün).

5.2.2 Clustering nach Objektbewegung

Das Clustering nach Objektbewegung ergibt 3 wesentliche Cluster. In Abbildung 36 sind diese Bewegungen farblich eingezeichnet. Es wird unterschieden zwischen Bewegung nach links unten (orange), nach rechts oben (blau), sowie Bewegungen die zuerst hinauf und dann hinunter (grün) führen. Weitere Cluster sind Bewegung nach rechts unten (rot) und nach links oben (türkis).

5.2.3 Clustering nach Objektbewegung und Ort

Fügt man nun den Ort als weiteres Merkmal hinzu, werden weitere Cluster gebildet. In Abbildung 37 ist zu erkennen, dass sich Elefanten und Pfleger

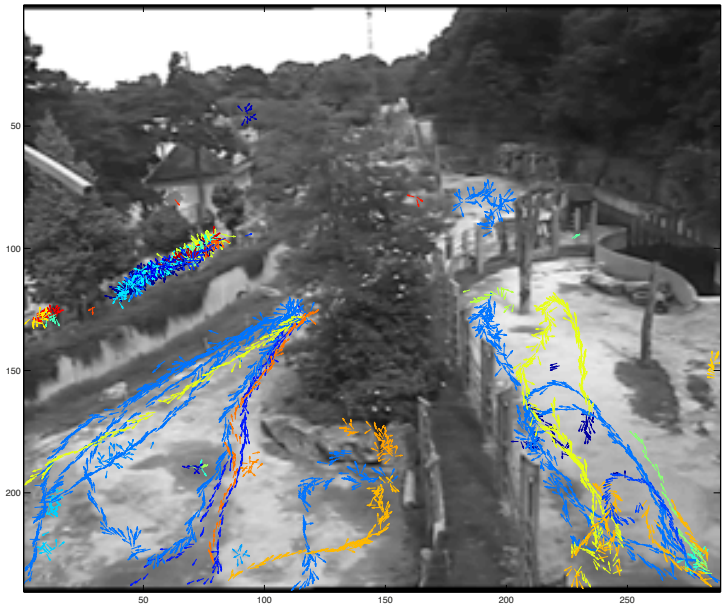


Abbildung 35: Bewegungspfade nach Farbhistogramme geclustert.

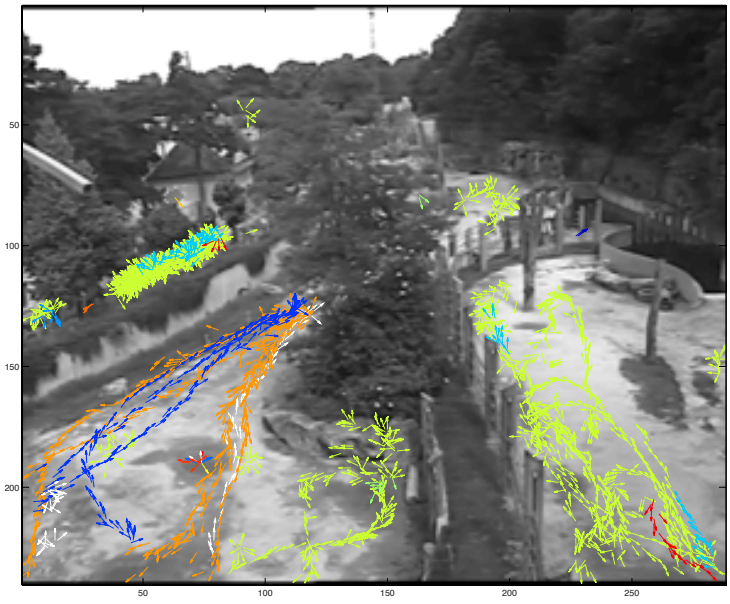


Abbildung 36: Bewegungspfade nach Bewegungsrichtung geclustert.

in der linken Bildhälfte hauptsächlich durch ihre Bewegungsrichtung unterscheiden und werden weniger durch die örtliche Komponente beeinflusst. Elefanten in der rechten Bildhälfte hingegen, werden nun auch örtlich gruppiert.

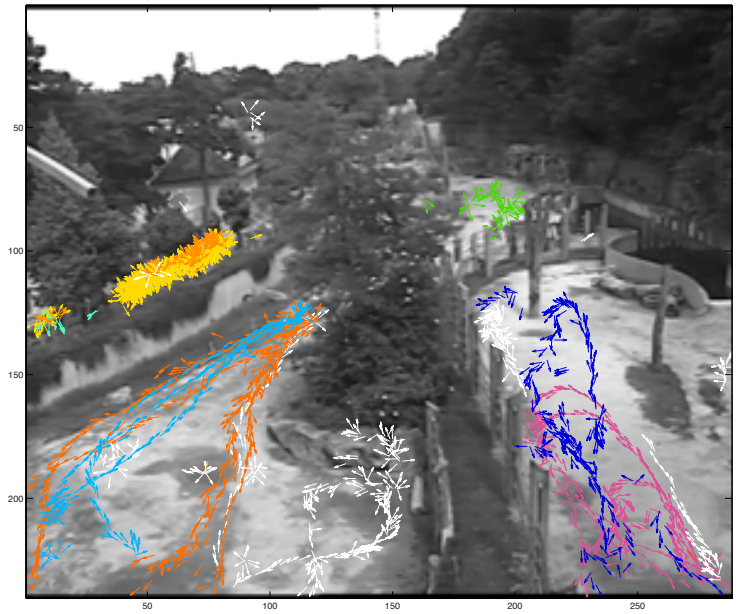


Abbildung 37: Bewegungspfade nach Bewegungsrichtung und Ort geclustert.

5.2.4 Clustering nach Objektbewegung, Ort und Aussehen

Werden nun alle drei Merkmale, Richtung, Ort und Aussehen, zu einem Merkmalsraum zusammengeführt, ergibt dies eine detaillierte Unterscheidung der Objekte. In Abbildung 38 sind wiederholte Bewegungspfade farblich gekennzeichnet. Dabei werden in der linken Bildhälfte wiederholte Ereignisse von den drei Elefanten (ID 2, 4, 23), die von links unten nach rechts oben gehen (hellgrün), erkannt. Die drei Pfleger (ID 35, 38, 60) die von rechts oben nach links unten gehen werden blau markiert. Unterschieden wird zwischen den zwei Elefanten (ID 14 und 16), die von rechts oben nach links unten gehen, da diese örtlich unterschiedliche Bewegungspfade besitzen. Auch der Pfleger (ID 41), der hinauf geht, unterscheidet sich von den

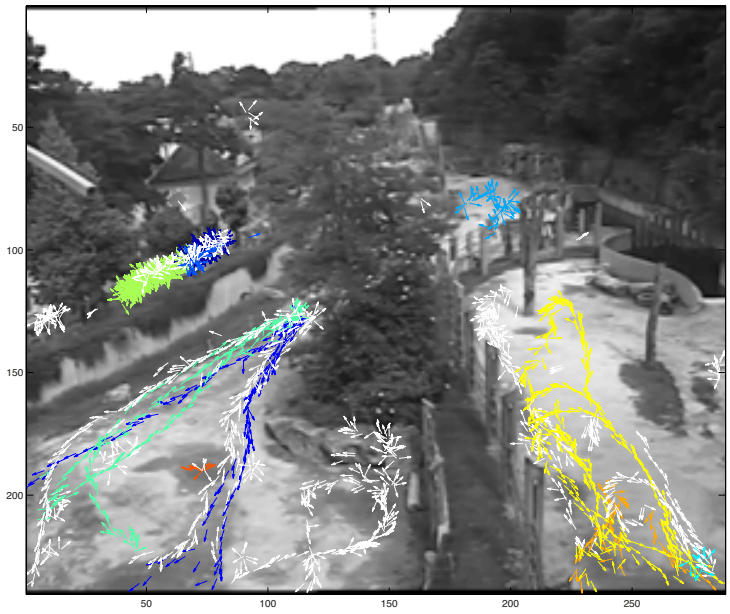


Abbildung 38: Bewegungspfade nach Bewegungsrichtung, Ort und Aussehen geclustert.

Elefanten, deren Bewegung von links unten nach rechts oben führt. In der rechten Bildhälfte werden die Elefanten (ID 32, 66, 83) als ähnlich erkannt und gelb eingezeichnet. Elefanten (ID 46 und 59), die nur kurz rechts unten zu sehen waren, werden orange markiert. Die Elefanten (ID 7 und 11) im oberen Teil des Bildes sind hellblau gekennzeichnet. In Abbildung 39 kann die Clusterzugehörigkeit anhand der, nach Cluster sortierte, Distanzmatrix abgelesen werden. In Tabelle 5 werden die Ergebnisse des Clusterings gezeigt und können mit der Auflistung der Ereignistypen in Tabelle 4 verglichen werden.

Das Ergebnis des Clusterings nach Objektbewegung, Ort und Aussehen ist durchaus nachvollziehbar. Es werden sich wiederholende Ereignisse erkannt und korrekt gekennzeichnet. Elefanten die links von oben nach unten gehen (ID 14, 16) werden nicht als ähnlich zueinander erkannt, da ihre örtliche Distanz mehr Gewicht hat, als vor dem Clustering erwartet.

Ein Problem für die Objektidentifizierung ist die Verschmelzung und Trennung von Objekten und wechselnde Lichtverhältnisse. Bei Aufnahmen

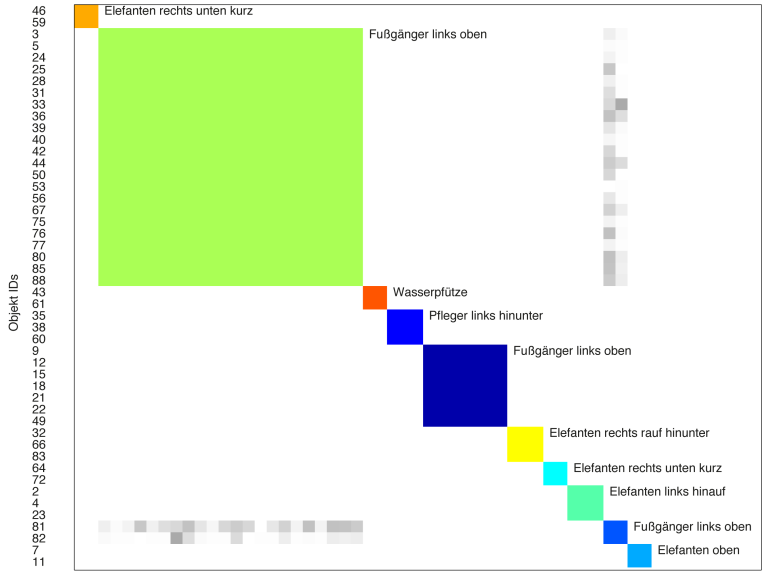


Abbildung 39: Sortierte Distanzmatrix - Cluster mit mehreren Objekten.

Akteur	Ort/Richtung	Ereignisse	korrekt
Elefant	links hinauf	2, 4, 23	✓
Elefant	links hinunter	-	✗
Elefant	oben	7, 11	✓
Elefant	rechts hinauf - hinunter	32, 66, 83	✓
Elefant	rechts unten	46, 59	✓
Pfleger	links hinunter	35, 38, 60	✓

Tabelle 5: Ergebnistabelle der sich wiederholenden Ereignisse nach dem Clustering nach Objektbewegung, Ort und Aussehen.

im Freien verändert sich die Helligkeit des Bildes deutlich, wenn die Sonne von einer vorbeiziehenden Wolke verdeckt wird. Dadurch ändern sich Schatten von statischen Objekten in der Umgebung, welche als Folge der wechselnden Lichtverhältnisse als Objekte erkannt werden. Diese falsch erkannten Objekte können mit anderen Objekten verschmelzen und das Objekttracking unterbrechen. Werden Objektbewegungen durch unregelmäßiges Tracking unterbrochen, werden diese Objektpfade auch nicht als ganzer Action-String gespeichert. Das hat zur Folge, dass manche Objekte nicht wie erwartet zuge-

ordnet werden können. Je nachdem mit welchem Faktor der aktuelle Frame zur Hintergrunderstellung mit einfließt, benötigt das Hintergrundmodell unterschiedlich viel Zeit, um den Hintergrund komplett zu aktualisieren, sodass Schatten nicht mehr als Objekte erkannt werden.

5.3 Quantitative Evaluierung

Bei der quantitativen Evaluierung wird festgestellt, ob das System in einem realen Umfeld funktioniert. Um ein Ergebnis bewerten zu können, erfolgt zuvor eine Prameterbestimmung, die am besten geeignete Parameter der Bandbreite für das Clustering finden soll. Diese Bandbreite legt die Größe des Beobachtungsfensters des Mean-Shift-Clusterings fest und beeinflusst die Clustergrößen. Im Anschluss wird das Ergebnis der Ereigniswiedererkennung anhand einer längeren Videoaufnahme einer Tierbeobachtung präsentiert. Als Nächstes wird die Laufzeit der einzelnen Komponenten analysiert und überprüft, ab welcher Datenreduktion des Bildmaterials die Bewegungserkennung und die Objektidentifizierung versagen. Weiters werden das Wachstum des Merkmalsraumes, sowie die Laufzeitveränderungen des Clusterings analysiert.

Für die quantitative Evaluierung wurde eine Videosequenz von 60 Minuten einer Beachtung eines Elefantengeheges genutzt. In Abbildung 34 werden die verschiedenen Ereignistypen beschrieben. Als Merkmale wurden Ort, Bewegungsrichtung und Farbe aus den Atomic-Actions extrahiert.

5.3.1 Parameterbestimmung

Um geeignete Parameter der Clustering-Bandbreite bestimmen zu können, muss festgelegt werden, welche Ereignisse miteinander Cluster bilden sollen. Hierzu wird eine Teilsequenz des Eingangsvideos manuell analysiert und vorkommende Ereignisse gruppiert. Nachdem das Soll des Clusteringergebnisses bestimmt wurde, wird das Clustering mit unterschiedlichen Parametern der Bandbreite ausgeführt.

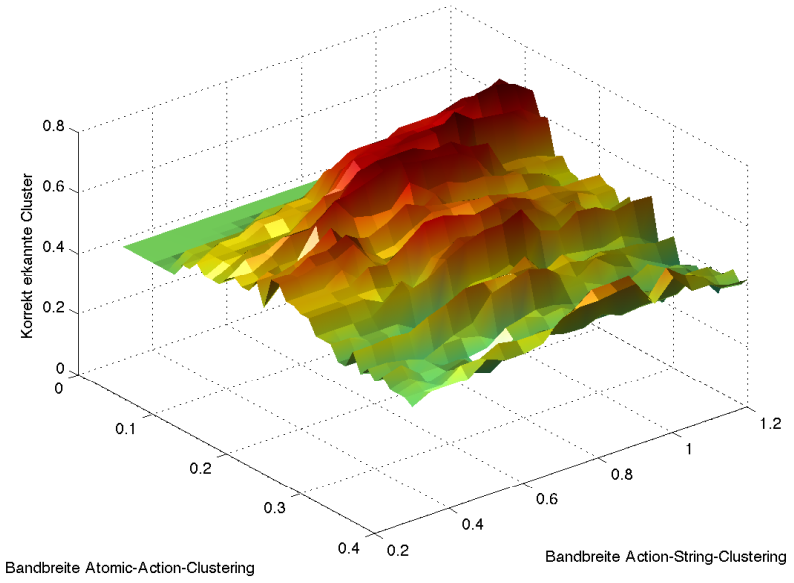


Abbildung 40: Bestimmung der Bandbreiten für Atomic-Action-Clustering und Action-String-Clustering. Ähnlichkeiten von Clusterergebnissen zu gewünschtem Clusterergebnis.

Für das Atomic-Action-Clustering und das Action-String-Clustering werden Wertebereiche festgelegt und Parameterkandidaten entnommen. Es erfolgt für jede Kombination der Parameter der zwei Kandidatenlisten ein Clustering. Jedes Clusteringergebnis wird mit dem gewünschten Clusterergebnis auf Ähnlichkeit überprüft. Hierzu wird für jeden Cluster des Sollergebnisses der ähnlichste Cluster aus dem Clusterergebnis gefunden und die Anzahl der übereinstimmenden Ereignisse gezählt und durch die Clustergröße normiert. Man erhält für jeden Cluster des gewünschten Clusterergebnisses einen Ähnlichkeitswert. Das arithmetische Mittel dieser Ähnlichkeitswerte beschreibt die Ähnlichkeit zwischen Sollergebnis und Clusterergebnis.

Nachdem alle Kombinationen von Parameterkandidaten für das Clustering als Parametereingabe verwendet wurden, können jene Parameterkandidaten ausgelesen werden, die das Clustering mit der höchsten Ähnlichkeit hervorgebracht haben. In Abbildung 40 sieht man die Ähnlichkeiten entlang der Achsen der beiden Clusterings eingezeichnet.

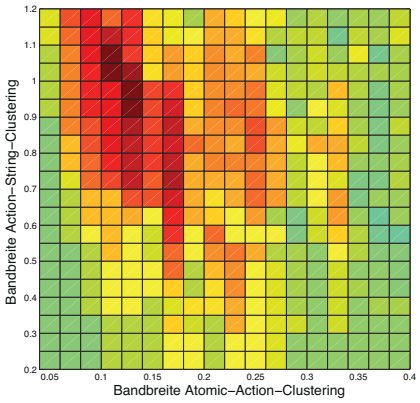


Abbildung 41: Bestimmung der Bandbreiten für Atomic-Action-Clustering und Action-String-Clustering. Dunkelrote Regionen zeigen Parametereinstellungen mit der geringsten Fehlerquote an.

5.3.2 Ergebnis der Ereigniswiedererkennung

In der Tierbeobachtung wurden 3295 Atomic-Actions und 98 einzelne Ereignisse erkannt. Die zuvor ausgeführte Parameterbestimmung ergab eine Bandbreite von 0.11 für das Atomic-Action-Clustering und 0.9 für das Action-String-Clustering. In Abbildung 41 können diese Werte anhand der dunkelroten Regionen abgelesen werden. Zur Auswertung des Ergebnisses wurden 10 Ereignistypen mit insgesamt 30 wiederkehrenden Ereignissen, sowie weitere Parameterwerte rund um die bestimmten Bandbreiten festgelegt.

Anhand der Ergebnistabelle 6 in Zeile 6 kann ablesen werden, dass bei einer Bandbreite von 0.11 für das Atomic-Action-Clustering und einer Bandbreite von 1.0 für das Action-String-Clustering 592 Atomic-Action-Cluster und 13 Action-String-Cluster gebildet wurden. Es wurden keine Ereignisse falsch zu Clustern geordnet, 2 potentielle wiederkehrende Ereignisse zu keinem Cluster zugeordnet, sowie 3 vermutete Cluster nicht gebildet. Daher ergibt diese Parametereinstellung die geringste Fehlerquote der 3 unterschiedlichen Fehlerquellen, die in Tabelle 6 am rechten Rand aufgelistet sind.

#	BB AA	BB AS	CL AA	CL AS	wk-Ereign.	falsch in Cluster	fehlt in Cluster	fehlende Cluster
1	0.09	0.8	774	7	15	0	3	4
2	0.09	0.9	774	12	25	0	3	4
3	0.09	1.0	774	14	30	0	5	3
4	0.11	0.8	592	12	30	0	3	3
5	0.11	0.9	592	13	32	0	3	3
6	0.11	1.0	592	13	34	0	2	3
7	0.13	0.8	439	9	28	0	1	4
8	0.13	0.9	439	12	36	1	1	4
9	0.13	1.0	439	16	47	6	8	1

Tabelle 6: Ergebnistabelle der quantitativen Evaluierung
Jede Tabellenzeile zeigt die Ergebnisse eines Methodendurchlaufs mit den definierten Parameter für das Atomic-Action-Clustering und Action-String-Clustering.
BB AA = Bandbreite Atomic-Action-Clustering;
BB AS = Bandbreite Action-String-Clustering;
CL AA = Anzahl der Atomic-Action-Cluster;
CL AS = Anzahl der Action-String-Cluster;
wk-Ereign. = Anzahl der erkannten wiederkehrenden Ereignisse.

Je höher die Bandbreite des Atomic-Action-Clusterings gewählt wird, desto geringer die Anzahl der Atomic-Action-Cluster. Das Alphabet zur Beschreibung von Ereignisabfolgen verkleinert sich, da weniger zwischen Atomic-Actions unterschieden wird. Als Folge, ergeben sich durch die geringere Unterscheidung mehr wiederkehrende Ereignisse. Bei einer zu hoch gewählten Bandbreite, können die aus den Atomic-Actions bestehenden Action-Strings zu wenig von einander unterschieden werden. Bei einer zu niedrig gewählten Bandbreite, können Ereignisse zu genau beschrieben werden und somit nicht mehr als ähnlich erkannt werden.

Je höher die Bandbreite des Action-String-Clusterings festgelegt wird, desto höher die Anzahl der Cluster in denen sich mehr als ein Ereignis befindet. An einer gewissen Grenze verringert sich die Anzahl der Action-String-Cluster bei steigender Bandbreite, da die Cluster immer mehr miteinander verschmelzen. Bei einer zu hoch gewählten Bandbreite werden größere Clus-

ter als erwartet gebildet, wodurch sich vermehrt Ereignisse in falschen Cluster befinden, bzw. können die gebildeten Cluster nicht den vordefinierten Ereignistypen zugewiesen werden. Wird die Bandbreite zu niedrig gewählt, wird zu genau zwischen den Action-Strings unterschieden und es entstehen vermehrt Cluster mit nur einem Ereignis als Mitglied. Diese Ereignisse werden somit nicht als sich wiederholend erkannt.

5.3.3 Laufzeitanalyse

Die vorgestellte Methode basiert auf einer Stapelverwaltung des Eingabevideos. Es wird jeder Frame dieser Videosequenz einzeln und hintereinander eingelesen. Dabei durchläuft jedes Bild die gleichen Prozesse. Das Laden des Bildes, die Verkleinerung, die Generierung eines Hintergrund- und Differenzbildes, sowie die morphologischen Filter benötigen eine Laufzeit von $O(N)$, wobei N die Anzahl der Frames im Video angibt. Die Objektidentifikation und die Merkmals Extraktion werden in einer Schleife abgearbeitet. Die Anzahl der Schleifendurchgänge ist gleich der Anzahl der erkannten Objekte bei der Bewegungserkennung. Dadurch beträgt die Laufzeit $O(N*m)$, wobei m die Anzahl der erkannten Objekte angibt. Es ergibt sich eine Gesamtlaufzeit von $O(N * m)$ für die Prozesse vor dem Clustering. Wobei zu beachten ist, dass die Objektidentifizierung und der optische Fluss nur berechnet werden muss, wenn $m > 0$ zutrifft. Wenn man sich die einzelnen Komponenten des Prozesses genauer ansieht, kann man Unterschiede der benötigten Zeit erkennen. In Tabelle 7 ist der Zeitaufwand vom Einlesen des Bildes bis zur Merkmals Extraktion aufgelistet. Für die Berechnung wurden die Videosequenz aus Kapitel 5.2 Qualitative Evaluierung genützt und Durchschnittswerte berechnet. Als Entwickungsumgebung wurde MATLAB2012b auf einem System mit 2.2 GHz Prozessor und 10 GB Arbeitsspeicher genützt. Die Framegröße des Eingabevideos beträgt hierbei eine Auflösung von 702x576 Pixel. Das Bild wird auf 288x240 Pixel verkleinert. Zur Berechnung des optischen Flusses wird die Auflösung nochmals reduziert. Die angegeben Werte

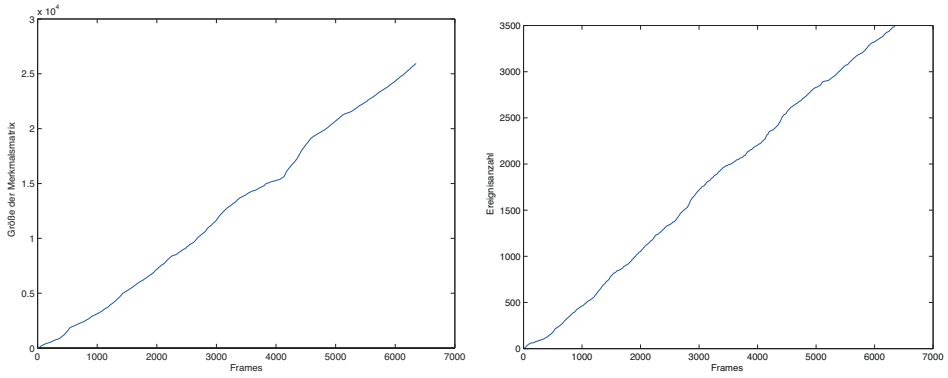
sind abhängig von der Leistung der verwendeten Hardware und dienen dazu, die einzelnen Prozesse untereinander vergleichen zu können.

Prozess	Sekunden
(Einlesen des Bildes von MPEG-4 Video	0.0574)
Einlesen des Bildes von JPEG Bilder	0.0141
Verkleinerung des Bildes	0.0083
Differnzbildgenerierung	0.0017
Hintergundbildgenerierung	0.0035
Morphologische Filter	0.0236
Optischer Fluss	0.0438
Objektidentifizierung	0.0156
Merkmals Extraktion Farbe	0.0212
Merkmals Extraktion Ort	0.0029
Merkmals Extraktion Richtung	0.0007
Gesamt	0.1354

Tabelle 7: Durchschnittlich benötigte Zeit der einzelnen Prozesse vor dem Clustering.

Anhand der gemessenen Zeiten kann man erkennen, dass sowohl das Einlesen des Bildes von einer Videoquelle, als auch die Berechnung des optischen Flusses die zeitaufwändigsten Prozesse sind. Insgesamt benötigen die Prozesse in etwa 0.14 Sekunden pro Bild. Das bedeutet in diesem Fall eine maximale Framerate von 7 Bildern pro Sekunde, damit die Methode in Echtzeit arbeiten kann. Der optische Fluss und die Merkmalsextraktion erfolgt jedoch nur dann, wenn im aktuellen Frame Objekte erkannt wurden. Für diese Abschnitte werden keine Berechnungen vorgenommen und können schneller abgearbeitet werden.

Mit der Länge des Videos steigt auch die Anzahl der erkannten Objekte und somit auch die Größe der Merkmalsmatrix. In Abbildung 42 kann man erkennen, dass sich die Ereignisanzahl und die Größe der Merkmalsmatrix linear entwickelt.

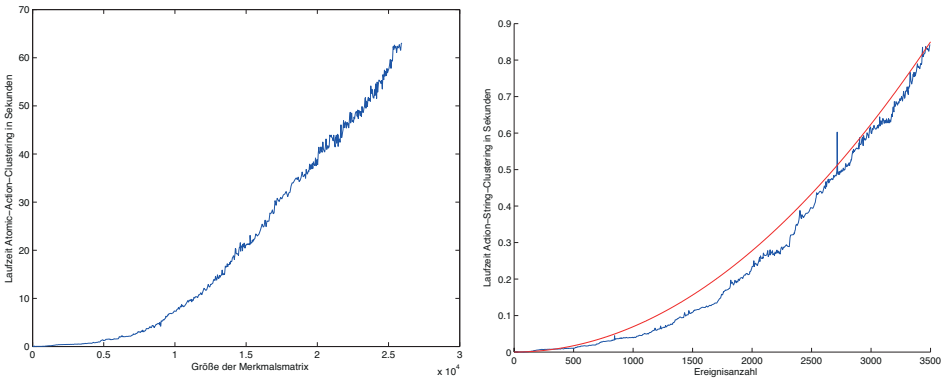


(a) Wachstum der Merkmalsmatrix (b) Wachstum der Ereignisanzahl

Abbildung 42: Wachstum der Merkmalsmatrix und der Ereignisanzahl.

Der zweite Teil der Methode besteht aus den zwei Clusterings, dem Atomic-Action-Clustering und dem Action-String-Clustering. Beide arbeiten mit dem Algorithmus Mean-Shift, welcher in Kapitel 2.5.1 näher erläutert wird. Der Vorteil dieses Clusterverfahrens ist, dass die Clusteranzahl nicht zuvor festgelegt werden muss, da es keine a-priori Informationen nach der Merkmalsextraktion gibt. Die Laufzeit des Mean-Shift beträgt $O(N^2)$, wobei N die Anzahlt der Merkmalsverktoren angibt. In Abbildung 43 wird die gemessene Laufzeit der Clusterings anhand der Größe der Merkmalsmatrix und der Anzahl der erkannten Ereignisse gezeigt. Hierfür wurde eine Videosequenz mit 6383 Videoframes, 25941 Merkmalseinträgen und 3495 Objekte verarbeitet.

Um die Laufzeit zu minimieren, ist es sinnvoll das Clustering in einem eigenen Prozess laufen zu lassen, während weiterhin die Objektidentifizierung und die Merkmalsextraktion parallel läuft. Eine weitere Möglichkeit ist das Action-String-Clustering zu stoppen, wenn eine gewisse Anzahl von Clustern erreicht ist, da es nur eine endliche und noch sinnvolle Anzahl von Ort, Richtung und Farbe geben kann. Ein Atomic-Action wird daraufhin zum Beispiel mittels Nearest-Neighbor-Algorithmus klassifiziert. Ein erneutes Clustering hätte somit erst dann zu erfolgen, wenn Cluster ineinander wachsen.



(a) Laufzeit Atomic-Action-Clustering (b) Laufzeit Action-String-Clustering

Abbildung 43: Laufzeit der Clusteringmethoden.

5.3.4 Zeitersparnis

Nach dem Clustering ist bekannt, an welchen Stellen sich wiederholende Ereignisse vorkommen. Somit lässt sich ein verkürztes Video ausgeben. Passagen, an denen keine relevanten Ereignisse vorkommen, werden ausgelassen. Um eine Zeitersparnis bei der Durchsicht zu ermitteln, wurde eine Videosequenz von 60 Minuten der Tierbeobachtung des Elefantengeheges verarbeitet. Insgesamt wurden 6383 Frames eingelesen. Die Anzahl der Frames mit sich wiederholenden Ereignissen beträgt 5144. Wenn man das Hauptaugenmerk auf Elefanten und Pfleger legt, verkürzt sich das Video auf 2589 Frames.

5.3.5 Datenreduktionspotential

Um die Laufzeit der Methode zu verringern, können die eingelesenen Videodaten in zeitlicher und räumlicher Dimension reduziert werden. Hierbei muss darauf geachtet werden, die Erkennungsrate der Objektidentifizierung und die Erkennung von sich wiederholenden Ereignissen nicht negativ zu beeinflussen.

Das Videomaterial lässt sich auf drei Ebenen reduzieren. Es lässt sich die Framerate und Framegröße des Eingangsvideos, sowie die Auflösung der Eingabebilder für die Berechnung des optischen Flusses, verringern. Alle

drei Datenreduktionsarten sind vom Inhalt des Eingabevideos abhängig. Die Reduktionsraten müssen für den speziellen Fall gewählt werden, um eine zuverlässige Objekterkennung und Objektidentifizierung gewährleisten zu können. Es kommt auf die Größe und auf die Geschwindigkeit der Objekte an, die erkannt werden sollen. Bei zu großem Abstand zwischen Objektstandort und dem Objektstandort im darauf folgenden Frame, erkennt die Methode des optischen Flusses das Objekt nicht wieder.

Im konkreten Fall des Elefantenvideos funktioniert die Objektidentifizierung bis einschlich 0,4 Frames pro Sekunde. Das Eingabevideo hat eine Framerate von 10 Frames pro Sekunde. Das entspricht einer Reduktion um das 25-fache.

Auch die Framegröße des Eingabevideos kann reduziert werden, wenn es der Inhalt zulässt. Das Eingabevideo hat eine Auflösung von 702x576 Pixel. Die Größe der Objekte wird in Abbildung 34 gezeigt. Auch bei einer Reduktion der Bildgröße auf ein Drittel, werden die Objekte noch zufriedenstellend erkannt und verfolgt. Die Bildgröße für den optischen Fluss kann bis zu einem Achtel reduziert werden, um den Prozess zu beschleunigen.

6 Schlussfolgerungen

Dieses Kapitel beinhaltet im Wesentlichen eine Zusammenfassung und Diskussion der Arbeit. Es wird erörtert, inwiefern das gestellte Problem gelöst werden konnte und welche Engpässe und Barrieren vorhanden sind. Des Weiteren werden Möglichkeiten der Weiterentwicklung von bestehenden Systemen aufgezeigt.

6.1 Zusammenfassung

Um eine Methode zur Erkennung sich wiederholender Ereignisse in Langzeitvideoaufnahmen zu entwickeln, erfolgte als erstes eine Auseinandersetzung mit den theoretischen Grundlagen und Arbeiten, die sich bereits mit ähnlichen Problemen beschäftigen. Anhand der gewonnenen Erkenntnisse wurde eine Methode entwickelt, die sich in mehreren aufeinander folgende Prozesse teilen lässt. In einem ersten Schritt erfolgt eine Objekterkennung über eine Hintergrundsubtraktion, um Bewegung und Objekte zu detektieren. Über eine Objektidentifikation werden Objekte über mehrere Frames hinweg verfolgt und zu Ereignissen zusammengefasst. Es werden aus jedem erkannten Ereignis, zu jedem Zeitpunkt, die Merkmale Ort, Bewegungsrichtung und Aussehen ausgelesen. Somit wird jedes Ereignis als Folge von Merkmalen beschrieben. Die gewonnenen Daten aus der Merkmalsextraktion werden geclustert und bilden dadurch ein Alphabet, welches nun zur Objektbeschreibung dient. Jedes Element des Alphabetes entspricht einem Cluster und somit auch einer Kategorie von Bewegung, Ort und Aussehen. Die Ereignisse können nun vereinfacht als Symbolkette (Action-Strings) beschrieben werden. Dadurch verringert sich die Dimensionalität der Ereignisbeschreibung und ein Vergleichen der Ereignisse über eine String Matching Methode wird ermöglicht. Nach einem Clustering der Action-Strings werden Ereignisse mit ähnlichem Bewegungsmuster und Aussehen gruppiert. Somit kann die Datenmenge des Videomaterials reduziert werden, indem häufig auftretende Ereignisse kompakt dargestellt werden können.

6.2 Reflexion und Möglichkeiten der Weiterentwicklung

Die Implementierung der Methode und die Evaluierung der Implementierung hat gezeigt, dass das Konzept, welches in dieser Arbeit ausgearbeitet wurde, zu den erwarteten Ergebnissen führt. Jedoch haben sich einige Hürden bemerkbar gemacht.

Bei der Objekterkennung werden bei wechselnden Lichtverhältnissen die Objekte nicht immer korrekt erkannt. Dadurch wird die Objektidentifizierung ungenau und beeinflusst somit auch das Clustering. Eine Verbesserung des Hintergrundmodells, indem starke Veränderungen der Lichtverhältnisse herausfiltert werden, kann die Objekterkennung verbessern. Weiters könnte zur Objektidentifikation nicht nur der vorangegangene Frame herangezogen werden, sondern auch der optische Fluss von weiter zurückliegenden Frames berechnet werden. Somit könnte das Abreissen von Trajektorien vermindert werden. Können alle Objekte korrekt verfolgt werden, erfolgt auch ein korrektes Clustering anhand ihres Verhaltens und Aussehens. Ähnliche Ereignisse, die durch die Objekte ausgelöst werden, werden somit wiedererkannt.

Die Schwierigkeit bei den Prozessen des Clusterings ist, die richtigen Parameter für die Bandbreite zu finden, wobei erschwerend hinzu kommt, dass die Ergebnisse des Atomic-Action-Clusterings jene des Action-String-Clusterings deutlich beeinflussen. Die Bandbreite gibt die Größe des Beobachtungsfester des Mean Shift Clusterings an und wirkt sich auf die Clusteranzahl aus. Je nachdem, wie detailliert Ereignisse auf Ähnlichkeit überprüft werden sollen, muss das Clustering zuvor kalibriert werden. Hierzu wird zur Parameterbestimmung eine Teilsequenz des Eingabevideos manuell analysiert und bestimmt, welche Ereignisse sich im selben Cluster befinden sollen. Anschließend wird ermittelt welche Parameter das gewünschte Ergebnis am ehesten erzeugen.

Die quantitative Evaluierung hat ergeben, dass sich die Anzahl der erkannten Objekte deutlich auf die Laufzeit des Clusterings auswirkt. Dieser Aspekt kann bei Langzeitvideoaufnahmen nicht außer Acht gelassen werden. Lösungsansätze hierfür wären, das Clustering in einem parallelen Prozess,

sowie nur in größeren Frameintervallen, ablaufen zu lassen. Eine weitere Möglichkeit ist das Action-String-Clustering zu einem gewissen Zeitpunkt mit einer Klassifizierung zu tauschen. Sind alle erforderlichen Objekttypen in verschiedene Cluster aufgeteilt, können neu erkannte Objekte mittels Nearest-Neighbor-Klassifikation einem vorhandenen Cluster zugeordnet werden. Ein Clustering der Atomic-Actions hätte somit erst zu erfolgen, wenn Cluster ineinander wachsen.

6.3 Fazit

Die Ausarbeitung der Methode hat ergeben, dass Objekte über Symbolketten beschrieben und auch auf Ähnlichkeit überprüft werden können. Synthetische Videos konnten hierbei zeigen, dass verschiedene Bewegungen, Standorte und Farben unterschieden und wiedererkannt werden können. Durch die Datenreduktion des Eingabevideos können die Prozesse von der Eingabe bis zur Merkmalesxtraktion in Echtzeit berechnet werden. Das Clustering hingegen skaliert schlecht mit der Länge des Videos und der Anzahl der Objekte und bedarf weiterer Optimierung. Mit Hilfe von Videoaufnahmen aus dem Tiergarten Schönbrunn konnte gezeigt werden, dass die Methode auch unter realen Verhältnissen zu guten Ergebnissen führt.

6.4 Ausblick

Mit Hilfe dieser Methode können bestehende Überwachungs- und Beobachtungssysteme in ihrem Funktionsumfang erweitert werden. Ein Überwachungssystem könnte dahingehend erweitert werden, indem ein erfolgtes Ereignis ausgewählt werden kann, um bei Wiederauftreten eines ähnlichen Ereignisses einen Alarm auszulösen. Auch bei der nachträglichen Durchsicht des Videomaterials, könnte diese Erweiterung nützlich sein, indem bestimmte Ereignisse markiert werden.

Das System könnte auch gegenteilig genützt werden, um Ereignisse, die von allen anderen abweichen, zu ermitteln. Hierbei werden alle Cluster mit

nur einem Ereignis als Ausreisser erkannt. Somit könnten Anomalien in Überwachungen erkannt werden.

Die Methode kann Objekte auch anhand ihres Aussehens wiedererkennen. Dies kann bei Tierbeobachtungen vorteilhaft sein, wenn das Hauptaugenmerk auf einer gewissen Tierart liegt. Somit könnten all jene Szenen herausgefiltert werden, die für den Beobachter relevant sind.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] Kristian Bredies and Dirk Lorenz. *Mathematische Bildverarbeitung: Einführung in Grundlagen und moderne Theorie*, volume 4. Springer, 2011.

[2] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

[3] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149 vol.2, 2000.

[5] Damir Culjat. Dynamische programmierung in der spracherkennung. Technical report, Technical report, FU Berlin, FB Informatik, 1999.

[6] U. Damnjanovic, V. Fernandez, E. Izquierdo, and J.M. Martinez. Event detection and clustering for surveillance video summarization. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, pages 63–66, May 2008.

[7] Konstantinos G Derpanis. Mean shift clustering. 2005. http://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf.

[8] Zoulfa El Jerroudi. *Eine interaktive Vorgehensweise für den Vergleich und die Integration von Ontologien*, volume 25. BoD–Books on Demand, 2010.

[9] Angelika Erhardt. *Einführung in die Digitale Bildverarbeitung*, volume 1. Vieweg+Teubner - Springer, 2008.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[11] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: a texture classification example. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 456–463 vol.1, Oct 2003.

[12] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[13] Peer Kröger. Neue herausforderungen fur dichtebasiertes clustering. *GI-Edition*, page 85.

[14] Ce Liu. Optical flow matlab/c++ code. http://people.csail.mit.edu/celiu/OpticalFlow/, last visited: November 2014.

[15] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009.

[16] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, March 2002.

[17] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[18] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[19] Bernd Neumann. Optical flow. *ACM SIGGRAPH Computer Graphics*, 18(1):17–19, 1984.

[20] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4, Oct 2004.

[21] Claudio Piciarelli and Gian Luca Foresti. Surveillance-oriented event detection in video streams. *IEEE intelligent systems*, 26(3):32–41, 2011.

[22] Thomas A. Runkler. *Data Mining*, volume 1. Vieweg+Teubner - Springer, 2010.

[23] J. Schanda. *Colorimetry: Understanding the CIE System*. Wiley, 2007.

[24] Ingo Schmitt. *Ähnlichkeitssuche in Multimedia-Datenbanken: Retrieval, Suchalgorithmen und Anfragebehandlung*. Oldenbourg Verlag, 2006.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, Aug 2000.

[26] F. Stompe. *Real-Time Data Mining:*. Diplomica Verlag, 2009.

[27] Quan Wang. Dynamic time warping (dtw). http://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping--dtw-, last visited: November 2014.

[28] Zheng Yi and Fan Liangzhong. Moving object detection based on running average background and temporal difference. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pages 270–272, Nov 2010.

[29] Jan Zatyik and Pavel Rajmic. 3d histogram of rgb image. http://www.mathworks.com/matlabcentral/fileexchange/38685-3d-histogram-of-rgb-image, last visited: October 2014.

[30] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–123–II–130 vol.2, 2001.

[31] Henrik Zimmer. Mean shift-tracking. http://ganymed.imib.rwth-aachen.de/deserno/seminare/bv_2005-07.pdf, last visited: November 2014.