

11B0033 AuV: Meilenstein 06

Videokomprimierung, Spracherkennung

Julius Schöning
j.schoening@hs-osnabrueck.de

Abgabedatum: 07. Januar 2021 23:59:59

In diesem Meilenstein werden Sie sich mit der Interframe-Kodierung von Videosignalen, der Objektbasierten Kodierung und der Spracherkennung mit künstlichen neuronalen Netzwerken beschäftigen. Alle Aufgaben dieses Aufgabenblattes können Sie wahlweise mit Matlab oder der Scriptssprache Python bearbeiten.

Matlab steht Ihnen als Total Headcount Lizenz der Hochschule Osnabrück zur Verfügung. Installationsinformationen finden Sie unter <https://wiki.hs-osnabrueck.de/pages/viewpage.action?pageId=8749400>.

Gruppen

Der Meilenstein *Videokomprimierung, Spracherkennung* wird in Gruppenarbeit mit den bereits bekannten Mitgliedern umgesetzt. Dieser Meilenstein ist so ausgelegt, dass Sie diesen an Ihrem eigenen Computer bearbeiten können.

Eine Veränderung der Gruppenzusammensetzung ist nicht mehr möglich.

Abgabe

Die Ergebnisse aller Teilaufgaben sind schriftlich in einem strukturierten Praktikumsbericht im PDF-Format zusammenzufassen. Ergänzen Sie Ihre Ausführungen durch aussagekräftige Screenshots, Abbildungen, Plots und Tabellen. Mögliche Word und L^AT_EX Vorlagen für Praktikumsberichte finden Sie im Downloadbericht dieses Praktikum.

Den strukturierten Praktikumsbericht und alle Quellcodedateien, die Sie in diesem Meilenstein erzeugen oder auf die Sie sich in Ihrem Praktikumsbericht beziehen müssen in Ihrem Abgabearchiv vorhanden sein.

Für die Abgabe komprimieren Sie alle erstellen Matlab bzw. Python Quellcodedateien sowie Ihren strukturierten Praktikumsberichts als PDF in ein zip-Archive mit dem Namen *06_UserName1_UserName2_UserName3.zip*. Laden Sie das Zip-File bis spätestens zum 07. Januar 2021 23:59:59 im Abgabebereich AuV Praktikum im OSCA als hoch.

Testat

Bereiten Sie sich für ein Testat mit ggf. schriftlichen Kurztest von ca. 10 Minuten vor. Inhalt dieses Testats werden die Themen der Meilensteine 05 und 06 sein. Ihre Gruppe erhält einen persönlichen Termin für die KW02/2021 zugeteilt.

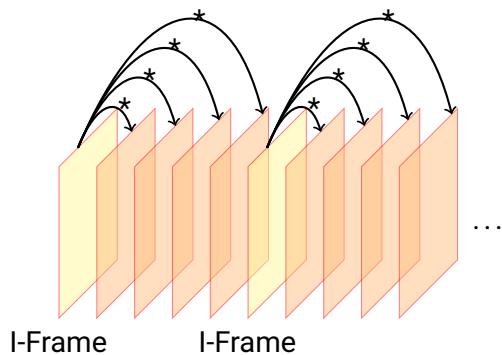
1 Aufgabe: Interframe-Kodierung mit Forward Motion Estimation

In Videos ähneln sich viele der Frames. Videos können effektiver komprimiert werden, wenn diese Ähnlichkeiten erkannt und anstelle eines kompletten Bilds gespeichert werden.

- a) Erstellen Sie ein Matlab- oder Python-Script, das das Video *car.mp4* mittels Interframe-Kodierung mit Forward Motion Estimation kodiert. Ein aus I- und P-Frames bestehendes Video ist als Ergebnis ausreichend. Der codierte Bitstream muss nicht in einer separaten Datei gespeichert werden.

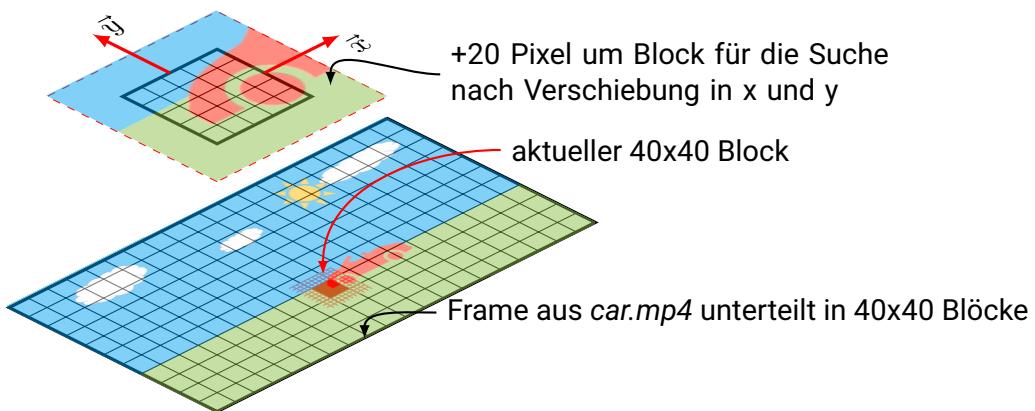
Gehen Sie dabei folgendermaßen vor:

- Lesen Sie nacheinander die Frames des Videos *car.mp4* ein. Jeder fünfte Frame wird als I-Frame genutzt.



I-Frames gefolgt von vier P-Frames; * Forward Motion Estimation

- Berechnen Sie nun mit *imabsdiff* welche Pixel sich vom letzten I-Frame zum aktuellen Frame ändern.
- Teilen Sie das Bild in 40x40 Pixel Blöcke ein. Berechnen Sie für jeden Block des Differenzbilds die durchschnittliche Abweichung.



Schema Motion Estimation eines Blockes

- Für jeden Block des aktuellen Frames der sich zum I-Frame geändert hat, suchen Sie nun den im Umkreis von 20 Pixel die bestmöglichste Verschiebung des Blockes vom I-Frame in \vec{x} und \vec{y} . Erstellen Sie aus den I-Frames und den verschoben Blöcke des I-Frames ein Ergebnisvideo ein. Als Bewertungskriterium für die Qualität Ihres Ergebnisvideos können Sie mit *immse* den Mean Square Error berechnen.

- Jeder Frame Ihres Ergebnisvideos sollte jetzt ausschließlich aus Blöcken der I-Frames bestehen.

Speichern Sie Ihr Script unter `auv_06_Aufgabe_1.m` bzw. `auv_06_Aufgabe_1.py`.

- Betrachten Sie nun das Ergebnisvideo. Welche Maßnahmen können ergriffen werden, um die Qualität des Videos zu verbessern?
- Moderne Video-Codecs nutzen Forward und Backward Motion Estimation sowie zusätzlich zu I- und P-Frames auch B-Frames. Beschreiben Sie die Funktion der B-Frames und diskutieren Sie die Vor- und Nachteile gerade in Hinsicht auf Bildqualität und Implementation.

2 Aufgabe: Objektbasierte Kodierung

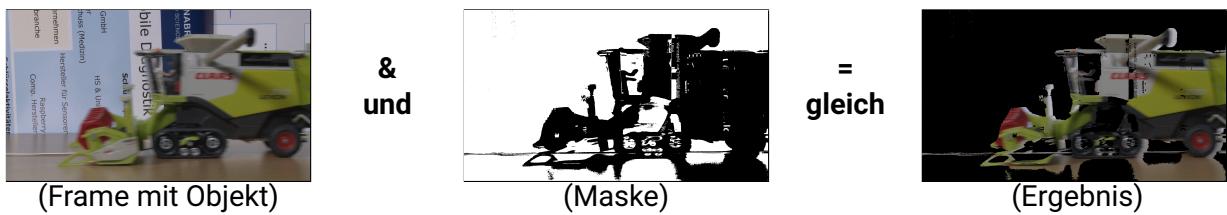
Der Vorder- und Hintergrund von Videos kann unterschiedlich komprimiert werden. Der Hintergrund ist nicht so wichtig und kann stärker komprimiert werden, was eine bessere Kompressionsrate ermöglicht.

- Um Vorder- und Hintergrund unterschiedlich komprimieren zu können, muss erkannt werden, was im Vordergrund steht. Betrachten Sie dazu das Matlab-Script `initialBackgroundRemoval.m`. Beschreiben Sie im Projektbericht die Funktion jeder Programmzeile.
- Modifizieren Sie das initiale Script nun so, dass anstelle des einfarbigen Hintergrunds ein unruhiger Hintergrund entfernt werden kann. Der Hintergrund soll mittels Subtraktion zweier Bilder bestimmt werden. Passen Sie die Variable `Threshold` an, um ein besseres Ergebnis zu erhalten. Testen Sie ihr Script mit dem Video `combineColoredBackground.mp4`.

Erstellen der Maske aus Frame mit und ohne Objekt.



Anwenden der Maske um Objekt zu Segmentieren.



- Nun soll der Hintergrund stärker komprimiert werden. Wenden Sie dafür mit `imfilter` und `fspecial` einen Filter vom Typ `average` mit den Größen 1, 8, und 16 auf den Hintergrund und speichern Ihre Ergebnisvideo im H.264 Format.

Speichern Sie Ihr Script unter `auv_06_Aufgabe_2.m` bzw. `auv_06_Aufgabe_2.py`.

- Vergleichen Sie die Videoqualität und Dateigrößen des Originalvideos und ihrer komprimierten Version.

3 Aufgabe: Spracherkennung mit künstlichen neuronalen Netzwerken

Ein weiterer Weg Wörter zu erkennen ist sie mit einem künstlichen neuronalen Netzwerk zu klassifizieren.

- a) Befolgen Sie das Matlab Tutorium zur Spracherkennung mit einem neuronalen Netzwerk (<https://de.mathworks.com/help/deeplearning/ug/deep-learning-speech-recognition.html>).

Installieren Sie dazu die notwendigen Toolboxen. Die Audio Toolbox kann als 30 Tage Trial-Version installiert werden.

Testen Sie die Implementation mit selbst aufgenommenen Wörtern.

Hinweis: Sie brauchen das Tutorium nur bis zum Kapitel “Load Speech Commands Data Set” für diese Praktikumsaufgabe auszuprobieren, da das Training eines künstlichen neuronalen Netz sehr sehr viel Rechenleistung beansprucht. Wenn Sie mögen können Sie gerne auch das neuronale Netz lernen.

Speichern Sie Ihr Script unter *auv_06_Aufgabe_3.m* bzw. *auv_06_Aufgabe_3.py*.

- b) Informieren Sie sich über neuronale Netzwerke. Wie unterscheidet sich diese Art der Erkennung von der Spracherkennung im letzten Meilenstein?
- c) Diskutieren Sie die Vor- und Nachteile der Spracherkennung und Audio-Komprimierung unter Zuhilfenahme von künstlichen neuronalen Netzen.