

Computer Vision Challenge

Computer Vision

im Studiengang Master Elektro- und Informationstechnik

Sommersemester 2018

Gruppe 11:

Tim Janßen

Andre Thommessen

Oliver Doege

Julia Ströbel

Sebastian Hügler

Abgabe: 12.09.2018

Prüfer: [Prof. Dr.-Ing. Klaus Diepold](http://www.ldv.ei.tum.de/team/ordinarius/klaus-diepold/) / M.Sc. Stefan Röhrl

Inhaltsverzeichnis

[Einleitung 3](#_Toc523077643)

[Aufgabenstellung 3](#_Toc523077644)

[Idee und Umsetzung 5](#_Toc523077645)

[Umwandlung der Bilder in Graubilder 5](#_Toc523077646)

[Intensitäts- und Beleuchtungskorrektur 5](#_Toc523077647)

[Bilaterale Filterung 5](#_Toc523077648)

[Harris-Merkmale berechnen 6](#_Toc523077649)

[Korrespondenzschätzung 6](#_Toc523077650)

[Finde robuste Korrespondenzpunktpaare mit Hilfe des RANSAC-Algorithmus 6](#_Toc523077651)

[Berechne die Essentielle Matrix 6](#_Toc523077652)

[Bildrektifizierungsalgorithmus 6](#_Toc523077653)

[Disparitätsermittlung 6](#_Toc523077654)

[Ausgabe des Free-Viewpoint Bildes 6](#_Toc523077655)

[Verkürzung der Bearbeitungszeit 7](#_Toc523077656)

[Graphische Benutzeroberfläche (GUI) 8](#_Toc523077657)

[Abbildungsverzeichnis 9](#_Toc523077658)

# Einleitung

Die Computer Vision Challenge ist ein Bestandteil zur Bewertung der Leistungen der Studenten über ein Semester im Fach Computer Vision. Dieses Jahr besteht die Challenge darin, aus einem Stereo-Bild-Paar eine dritte virtuelle Ansicht zu generieren. Die Computer Vision Challenge ist eine Programmierarbeit, die in Gruppen bestehend aus drei bis fünf Personen durchgeführt wird. Es wird vom Lehrstuhl ein Skript vorgegeben, dessen Vorgaben zu erweitern sind. Einzelne Schritte sind zu kommentieren und Quellen auszuweisen.

Zusatzpunkte sind über eine optimierte Laufzeit und über eine GUI zu erreichen. Ein optimierter Code mit einer relativ kurzen Ausführungszeit wird positiv angerechnet. Eine graphische Benutzeroberfläche(GUI) wirkt sich ebenfalls positiv aus. Diese sollte zum Laden von Stereobildern, zur Einstellung des Blickwinkels der virtuellen Ansicht und zur Ausgabe der virtuellen Ansicht genutzt werden können.

# Aufgabenstellung

Die Challenge besteht darin, aus einem Stereo-Bild-Paar eine dritte virtuelle Ansicht zu generieren. Dabei soll der Blickwinkel der virtuellen Ansicht zwischen den beiden realen Ansichten liegen und durch einen Prozentwert frei bestimmbar sein. Das Programm soll in Matlab ohne die Hilfe spezialisierter Toolboxen erstellt werden. Als Grundlage liegen diverse Papers vor, die übers Semester verteilt wurden, dazu die Folien aus den Vorlesungen sowie die eigene Internetrecherche.

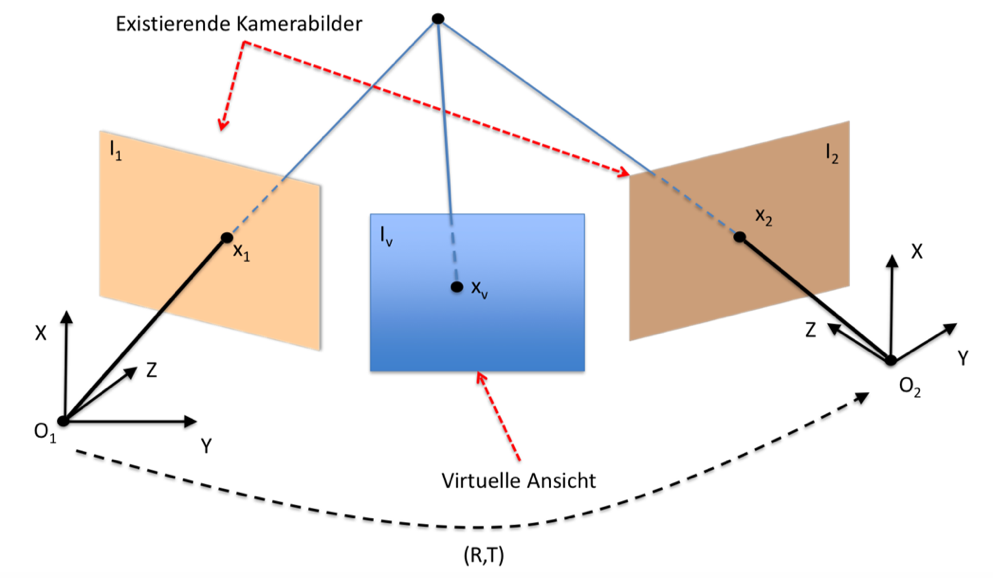


Abbildung 1: Geometrischer Zusammenhang zwischen den realen Bildern und einer künstlich generierten Ansicht

Abbildung 1 zeigt das zu untersuchende Szenario. Es werden zwei Bilder vom selben Motiv vorgegeben und mittels des zu erstellenden Programm übereinandergelegt, sodass ein drittes Bild berechnet wird.

Das vorgegebene Skript *challange.m* soll so erweitert werden, dass die Farbbilder *img/L1.jp*g und *img/R1.jpg* geladen und der Funktion *free\_viewpoint* übergeben werden. Die Funktion *free\_viewpoint* soll dann anhand des Parameters *p* eine virtuelle Ansicht zwischen den beiden realen Ansichten generieren.

Der Parameter *p* soll standardmäßig auf 50% stehen. Die Rechenzeit, die die Funktion *free\_viewpoint* benötigt hat, soll in der Variable *elapsed\_time* abgespeichert werden.



Abbildung 2: img/R1.jpg



Abbildung 2: img/L1.jpg

# Idee und Umsetzung

In diesem Kapitel werden zunächst die Idee und der Ablauf des Programms skizziert. Danach werden die Aufgaben der einzelnen Programmbestandteile erläutert.

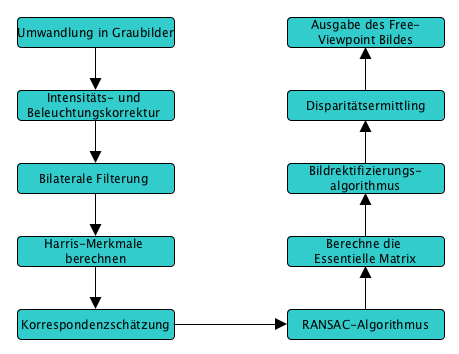


Abbildung 4: Ablauf des Programmes

Die Abbildung 4 zeigt den Aufbau und den Ablauf unseres Programms. Zuerst werden die beiden eingelesenen Bilder in Graubilder umgewandelt, um einen geringeren Aufwand bei den späteren Berechnungen zu erzielen. Danach wird eine Intensitäts- und Beleuchtungskorrektur bei beiden Bildern durchgeführt, um den vollen Intensitätsbereich ausnutzen und damit spätere Eckendetektionen zu erleichtern. Dieselbe Aufgabe verfolgt die darauffolgende bilaterale Filterung, bei der das Bild zwar weichgezeichnet wird, jedoch Kanten und Ecken erhalten bleiben. Im Anschluss erfolgt die Kanten- und Eckendetektion mit Hilfe des Harris-Detektors. Mit den detektierten Ecken aus den beiden Bildern kann eine Korrespondenzschätzung durchgeführt werden, um die Eckpunkte aus dem einen Bild den jeweiligen Eckpunkten aus dem anderen Bild zu zuordnen. Mit dem RANSAC-Algorithmus werden aus den geschätzten Korrespondenzen die robusten Korrespondenzpunktpaare bestimmt, um höhere Genauigkeit bei der Bestimmung der essentiellen Matrix zu erzielen. Diese erfolgt direkt im Anschluss mit Hilfe des Achtpunktalgorithmus. Durch die essentielle Matrix können die Rotation und Translation zwischen den beiden Ansichten berechnet werden. Diese werden zur Berechnung der dritten Ansicht benötigt.

## Umwandlung der Bilder in Graubilder

## Intensitäts- und Beleuchtungskorrektur

Die Bilder werden heller gestellt um die Kanten besser erkennen zu können.

## Bilaterale Filterung

Bei der Bilateralen Filterung werden die Oberflächen weicher aber die Kanten bleiben erhalten. Dies erleichtert die Suche nach Merkmalspunkten und unnötige Störgrößen werden elminiert.

## Harris-Merkmale berechnen

## Korrespondenzschätzung

## Finde robuste Korrespondenzpunktpaare mit Hilfe des RANSAC-Algorithmus

## Berechne die Essentielle Matrix

## Bildrektifizierungsalgorithmus

Ziel ist es die Epipole beider gegebenen Punkte gegen unendlich laufen zu lassen. Hier werden Bildpunkte der einzelnen Objekte jeweils der verschiedenen Bilder zugeordnet.

Bilder werden so gedreht, dass sie Parallel zueinander sind. (Paper Overview (erstes))

Parallele Epipolarlinien. 🡪 Suche auf den Linien nach gleichen Merkmalen (NNC-SAD)

Die Funktion sad\_scanline.m nnc\_scanline.m sorgt dafür, dass um ein Pixel ein gewisser Bereich (Window) gelegt wird, welches mit einem Bereich des anderem Bildes verglichen wird. Den Bereichen, die die geringste Abweichung aufweisen wird die gleiche Zugehörigkeit zugeordnet. Formel 1-7 Paper s.1025 , Berechnung Abstand, ... , Tiefenkarte Resultat, dadurch sind die Punkte im Raum bekannt und diese können auf die dritte, neue Fläche projiziert werden.

## Disparitätsermittlung

Vorstufe der Tiefenkarte

## Ausgabe des Free-Viewpoint Bildes

Die Bildgröße des dritten, berechneten Bildes wird auf die Bildgröße der gegebenen Bilder angepasst.

## Verkürzung der Bearbeitungszeit

Um eine schnellere Berechnung des Bildes zu erzielen, wird mit der Funktion *downsample.m* die Auflösung reduziert, in unserem Falle um einen Viertel, und mit der Funktion *upsample.m* wieder auf die ursprüngliche Auflösung hochgerechnet. Für das reine Auge stellt sich kein Unterschied ein, dennoch kann die Berechnungszeit um einiges verkürzt werden.

# Graphische Benutzeroberfläche (GUI)

Die graphische Benutzeroberfläche dient dem Bediener zur Steuerung des Programmes. Für ein schnelles zurechtfinden befinden sich auf der Oberfläche nur die wichtigsten Operatoren.



Abbildung 5: Benutzeroberfläche vor der Berechnung

Nach einer kurzen Erklärung, was dieses Programm leistet und wie es zu bedienen ist, kann der Blickwinkel, Parameter *p*, von 0 bis 1 eingestellt werden. Die richtige Eingabe wird durch eine minimale und maximale Grenze in dem Eingabefeld sowie einem Hinweis hinter dem Feld gewährleistet. Die gewünschte linke und rechte Ansicht wird jeweils über die Buttons *Linke Ansicht laden* und *Rechte Ansicht laden* ausgewählt. Dazu wird eine Explorer-Ansicht geöffnet, um die Bildauswahl für den Benutzer zu vereinfachen. Nach der Auswahl werden die Bilder in das Programm geladen und dem Benutzer zur Bestätigung jeweils unter den Buttons angezeigt. Zusätzlich ist der Pfad der Bilder direkt darunter einsehbar, damit der Benutzer kontrollieren kann, ob die richtigen Bilder geladen wurden und er sich nicht verklickt hat. Das dritte Bild kann dann über den Button *3. Ansicht berechnen* über das dahinter entworfene Programm berechnet werden. Nach der Berechnung wird das erzeugte Bild zwischen beiden zuvor geladenen Bildern angezeigt, um einen guten Vergleich zu den äußeren Ansichten zu geben. Die Berechnungszeit, die das Programm für die Berechnung der mittleren Ansicht benötigt, wird unterhalb des Bildes in Minuten angezeigt.

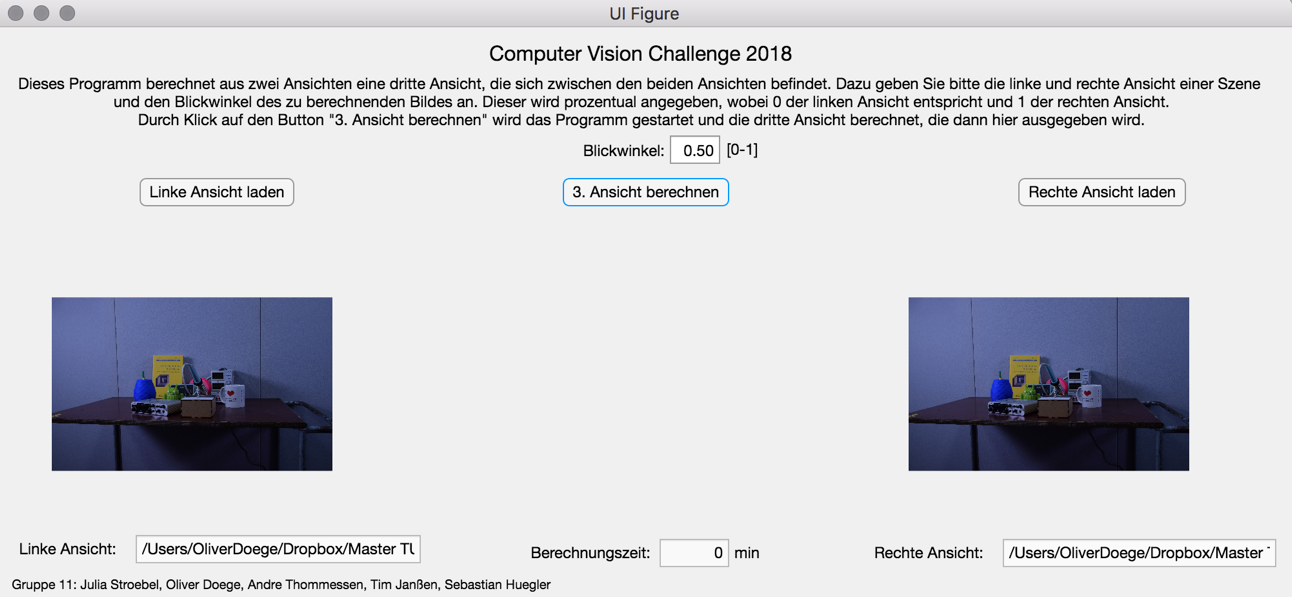


Abbildung 6: Benutzeroberfläche mit geladenen Bildern

# Abbildungsverzeichnis

[Abbildung 1: Geometrischer Zusammenhang zwischen den realen Bildern und einer künstlich generierten Ansicht 3](file://localhost/Users/OliverDoege/Desktop/Doku_Gr11_Challenge.docx#_Toc522452448)

[Abbildung 2: img/L1.jpg 4](file://localhost/Users/OliverDoege/Desktop/Doku_Gr11_Challenge.docx#_Toc522452449)

[Abbildung 2: img/R1.jpg 4](file://localhost/Users/OliverDoege/Desktop/Doku_Gr11_Challenge.docx#_Toc522452450)

[Abbildung 4: Projektverlauf 5](file://localhost/Users/OliverDoege/Desktop/Doku_Gr11_Challenge.docx#_Toc522452451)

**1.1 Mindestanforderungen**

* **Dokumentation:** Erstellen Sie ein Dokument (\*.pdf, \*.docx), indem Sie die Funkti- onsweise ihres Programms kurz erklären und legen Sie auch die Quellen dar, die sie für Ihren Ansatz konsultiert haben. Nutzen Sie dazu gerne auch mathematische Be- schreibungen, Skizzen und Blockdiagramme. Fügen Sie in dieses Dokument auch die geforderten Werte, die Sie errechnet haben, sowie Plots, die Sie erstellt haben ein, und legen Sie das Dokument Ihrem Abgabearchiv bei. Weisen Sie in diesem Dokument auch auf eventuelle Zusatzfeatures hin, die Sie implementiert haben (sie- he Zusatzpunkte).

