

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. Ergänzend zur Challenge wird eine Klausur geschrieben. Beide Bewertungesergebnisse werden 50:50 in die Endnote gewichtet. 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.

Eine Benutzeroberfläche wurde zusätzlich zur vorgeschriebenen Aufgabenstellung angefertigt.

# Aufgabenstellung

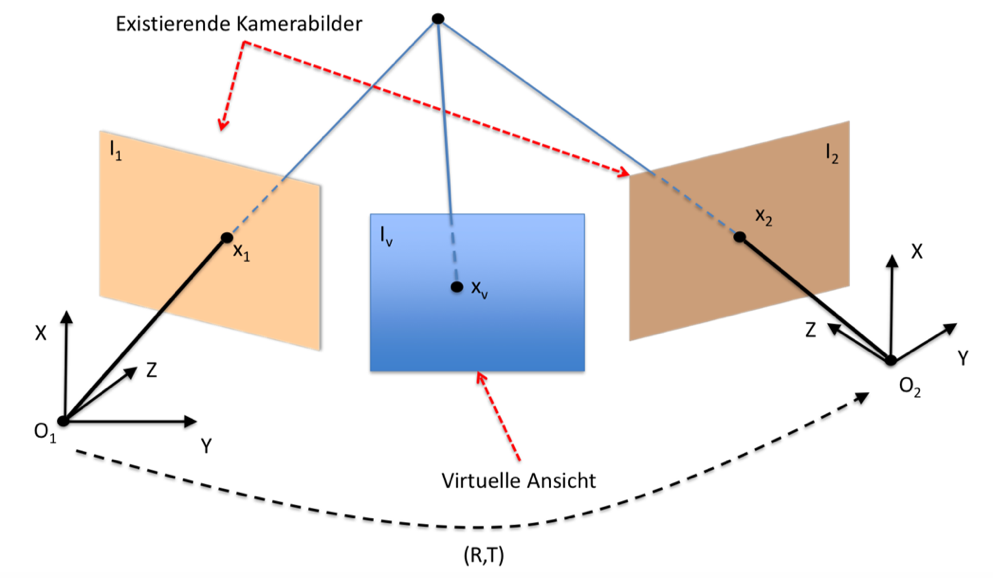
Die Challenge besteht darin, aus einem Stereo-Bild-Paar eine dritte virtuelle Ansicht zu generieren. Dabei soll der Blickwinkel der virtuellen Ansicht zwischen den beiden realen Ansichten liegen und durch einen Prozentwert frei bestimmbar sein. Das Programm soll in Matlab ohne die Hilfe spezialisierter Toolboxen erstellt werden. Als Grundlage liegen diverse Papers, die übers Semester verteilt, freiwillig zu bearbeiteten Hausaufgaben 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 erstellt und mittels des zu erstellenden Programm übereinandergelegt, sodass ein drittes Bild erstellt 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 : img/L1.jpg

Abbildung : img/R1.jpg

# Idee und Umsetzung

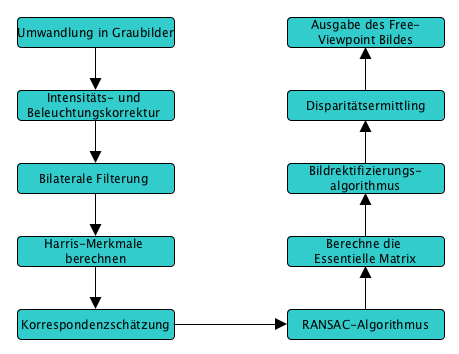


Abbildung : Projektverlauf

Abbildung 4 zeigt den Verlauf unserer Zusammensetzung unseres Projektes. Nach der Umwandlung der Bilder in Graubilder wird eine Intensitäts- und Beleuchtungskorrektur durchgeführt. Mit der bilateralen Filterung

## 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 einzelner 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 erhalten, wird mit der Funktion downsample.m die Auflösung auf, in unserem Falle eine viertel reduziert und mit der Funktion upsample.m wieder aufgebessert. Für das reine Auge stellt sich kein Unterschied ein, dennoch kann die Berechnungszeit um einiges verkürzt werden.

# Graphische Benutzeroberfläche (GUI)

Die graphische Benutzeroberfläche dient für den Bediener zur Steuerung des Programmes. Auf der Oberfläche befinden sich nur die wichtigsten Operatoren.

Abbildung 5: Benutzeroberfläche vor der Berechnung

Nach einer kleinen Information, kann der Blickwinkel, Parameter *P*, von 0 bis 1 eingestellt werden. Die zur Verfügung stehenden Bilder werden über die Buttons *Linke Ansicht laden* und *Rechte Ansicht laden* ausgewählt und ins Programm geladen werden. Nach Abschluss des Ladevorgangs werden die Bilder jeweils unter den Buttons dargestellt. Der Pfad der Bilder ist direkt darunter einsehbar. Das 3 Bild, kann über den Button *3. Ansicht berechnen* über das dahinter entworfene Programm berechnet und angezeigt werden. Die Berechnungszeit, die das Programm benötigt, wird unterhalb des Bildes in Minuten angezeigt.

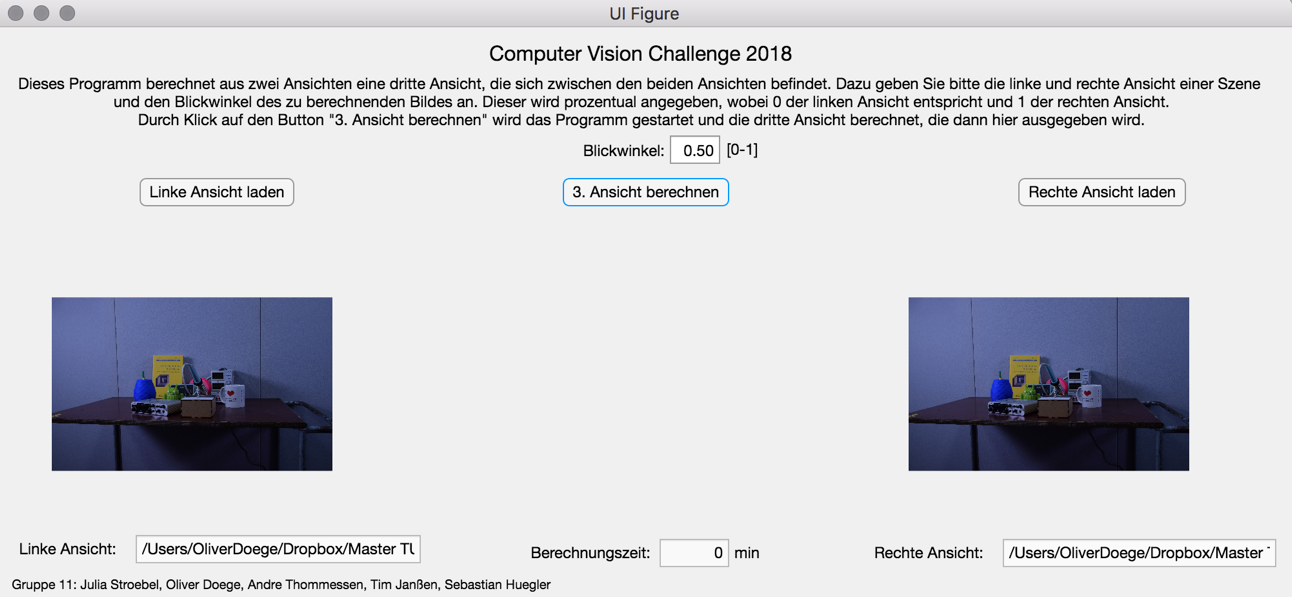


Abbildung 6: Benutzeroberfläche mit geladenen Bildern

# 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).

