



# Dokumentation Computer Vision Challenge

Abgabedatum: 12.09.2018

Gruppe: G07

Gruppenmitglieder:  
Moritz Eckhoff  
Tobias Betz  
Victor van Eetvelt  
Fabian Uhl  
Christian Geiger



## Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>2</b>
<b>2</b>	<b>Programmablauf</b>	<b>2</b>
<b>3</b>	<b>Rektifizierung</b>	<b>2</b>
<b>4</b>	<b>Disparity Map</b>	<b>4</b>
<b>5</b>	<b>Erstellen des virtuellen Bildes</b>	<b>5</b>
<b>6</b>	<b>Nachbearbeiten des virtuellen Bildes</b>	<b>6</b>
<b>7</b>	<b>Zusatzpunkte</b>	<b>6</b>
7.1	Graphische Benutzeroberfläche . . . . .	6
7.1.1	Hauptmenu . . . . .	8
7.1.2	Rectify . . . . .	8
7.1.3	Disparity Map . . . . .	8
7.2	Laufzeit . . . . .	8
<b>8</b>	<b>Generierte Ansichten</b>	<b>9</b>

## 1 Aufgabenstellung

Aus einem Stereo-Bildpaar soll eine dritte virtuelle Ansicht generiert werden. Der Blickwinkel der virtuellen Ansicht soll zwischen den beiden realen Ansichten liegen und durch einen Prozentwert frei bestimbar sein. Das Programm soll in Matlab ohne die Hilfe spezialisierter Toolboxen erstellt werden. Abbildung 1 zeigt die schematisch den geometrischen Zusammenhang der existierenden Kamerabilder und des virtuell erstellten Bildes.

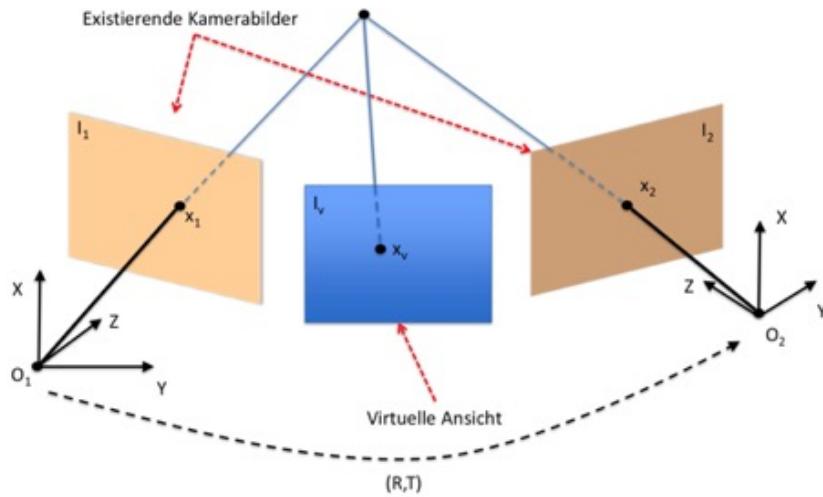


Abbildung 1: Geometrischer Zusammenhang zwischen den realen Bildern und einer künstlich generierten Ansicht

## 2 Programmablauf

In Abbildung 2 ist der Programmablauf zu sehen. Als erstes werden die beiden existierenden Kamerastereoaufnahmen geladen. Im nächsten Schritt müssen die beiden Bilder rektifiziert werden. Dabei werden die beiden Bilder so verändert, dass die Epipolarlinien parallel verlaufen. Dieser Schritt wird in einem folgenden Kapitel genauer erläutert. Als nächstes wird die sogenannte Disparity Map erstellt. In dieser wird dargestellt, wie sich jeder einzelne Pixel von einem zum anderen Bild verschiebt. Aus diesen Informationen kann anschließend das virtuelle Zwischenbild generiert werden. In einem nachfolgenden Schritt werden nicht berechnete Pixel korrigiert.

## 3 Rektifizierung

Die Rektifizierung ist ein wichtiger Schritt bei der Erstellung eines virtuellen Bildes. Das Ziel der Rektifizierung ist, dass die Epipolarlinien der Bilder hori-

## Programmablauf Challenge

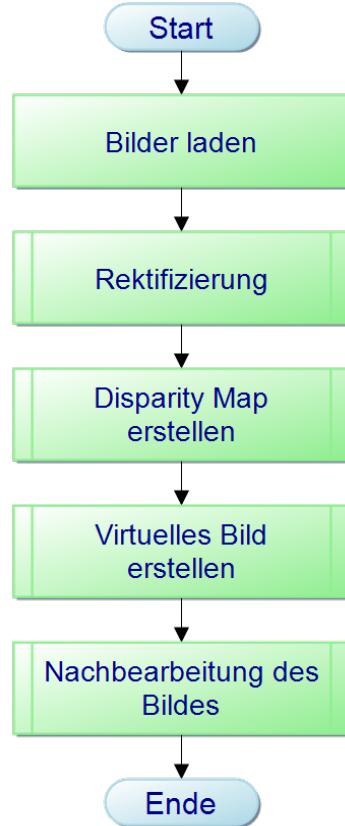


Abbildung 2: Programmablauf Challenge

zontal zueinander verlaufen. Korrespondenzpunkte von beiden gegebenen Bildern besitzen nach dem Bearbeitungsschritt die gleichen vertikalen Koordinaten. Dadurch kann die Disparity Map einfacher erstellt werden, da nur auf einer horizontalen Linie nach Korrespondenzpunkten gesucht werden muss. Außerdem bringt es Vorteile bei der Rekonstruktion des virtuellen Bildes. Die Abbildung 3 zeigt ein Bild nach einer Rektifizierung. Das obere Bildpaar zeigt die unrektifizierten Bilder und im unteren Bildpaar sind die rektifizierten Bilder zu sehen.

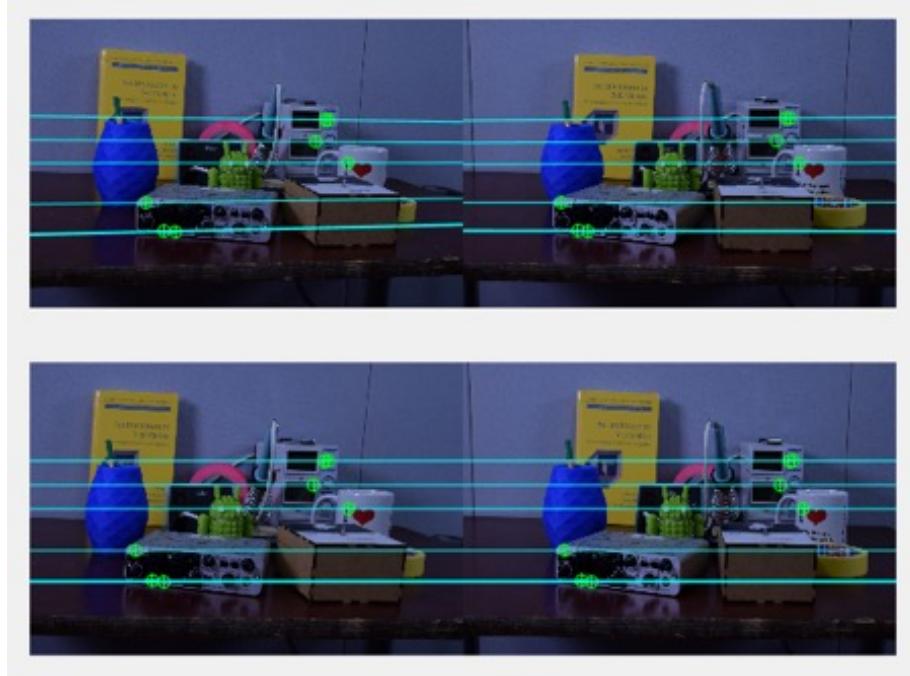


Abbildung 3: Ergebnis der Rektifizierung

## 4 Disparity Map

In einem ersten Versuch sollte eine Tiefenkarte mit Hilfe von Korrespondenzpunkten erstellt werden. Dabei wurden die in der Vorlesung und in den Hausaufgaben implementierten Funktionen verwendet und so angepasst, dass diese die für die gegebene Aufgabe geforderte Funktion erfüllen. Jedoch ist hier ein weiteres Problem aufgetreten. In den gegebenen Bildern existieren zu viele homogene Flächen, bei welchen die Suche nach Korrespondenzpunkten mit den bereits implementierten Funktionen, nicht möglich ist. In Abbildung 4 ist zu sehen, wie eine Tiefenkarte aussieht, die mit den gefundenen Korrespondenzpunkten erzeugt wurde. Es sind nur die Umrisse der Gegenstände auf dem Tisch und die Kante in der Wand zu erkennen.

Durch einer Recherche im Internet wurde eine andere Methode (Block-Matching) gefunden, mit welcher das Erstellen einer Tiefenkarte bzw. Verschiebungskarte möglich ist. Mit Hilfe des Block Matching wurde daraufhin eine Verschiebungskarte (Disparity Map) erstellt. Dabei wird ein Bildausschnitt mit einer definierten Größe aus dem ersten Bild ausgeschnitten und dessen korrespondierender Bildausschnitt entlang der Epipolarlinie im zweiten Bild gesucht. Die Suche beschränkt sich dabei auf eine festgelegte, bildspezifische Maximalverschiebung (disparity range). Als Verschiebung wird die Entfernung in Pixeln zwischen den korrespondierenden Bildausschnitten in die Disparity Map einge-

tragen. Um die Blöcke einander zuordnen zu können wird die Sum of Squared differences benutzt. Die Anzahl und Qualität der Korrespondenzpunkte steigt im Vergleich zu unserer Version mit den Korrespondenzpunktpaaren, da nur auf einem bestimmten Abschnitt der Epipolarlinie gesucht wird. Außerdem wird das Programm dadurch deutlich schneller. Es wurde ein vorhandenes Programm von Chris McCormick verwendet. [MCC] In Abbildung 5 ist die Disparity Map zu sehen, die mit Hilfe des Block Matching Algorithmus erstellt worden ist. Wie vorallem am Hintergrund zu erkennen ist, gibt es noch einige Fehler, jedoch wird jedem Pixel ein Wert zugeordnet und größtenteils ist dieser Wert auch richtig. Beispielsweise sind die Tischkante, die Box, der Frequenzgenerator und die Android-Figur sehr gut zu erkennen.

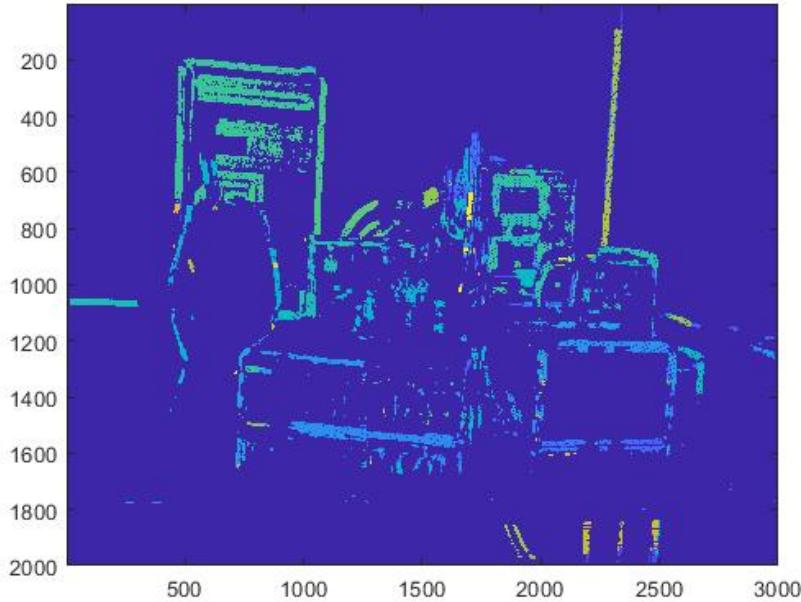


Abbildung 4: Tiefenkarte aus Korrespondenzpunkten

## 5 Erstellen des virtuelles Bildes

In der Funktion „Reconstruction 3D“ wird das virtuelle Bild erstellt. Dabei werden als erstes die Pixel nach der Entfernung zur Kamera sortiert, sodass weiter hinten liegende Gegenstände zuerst konstruiert werden. Das hat den Grund, dass wenn zwei oder mehr Bildpunkte auf denselben Pixel projiziert werden, der Pixel des weiter vorne liegenden Gegenstandes dargestellt wird. Dadurch soll verhindert werden, dass Hintergrund, welcher im ursprünglichen Bild zu se-

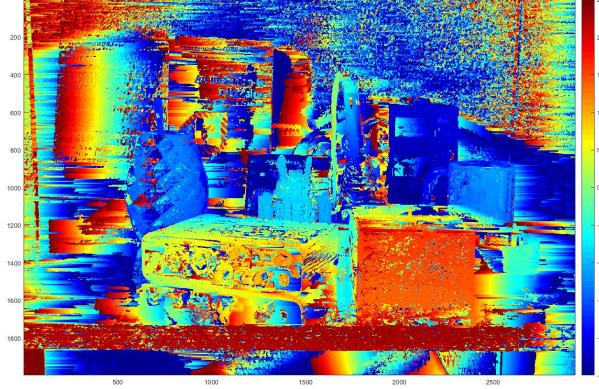


Abbildung 5: Disparity Map aus Block Matching

hen ist, aber im Zwischenbild nicht zu sehen sein soll, den Gegenstand verdrängt welcher dargestellt werden soll. Mit Hilfe der Disparity Map und der vom Nutzer gewünschten relativen Verschiebung, wird die neue x-Koordinate berechnet. Die Formel hierfür sieht wie folgt aus:

$$x_{\text{neu}} = \text{round}(\text{DisparityMap1}(y, x) \times \text{relativeVerschiebung} + x_{\text{alt}})$$

Wenn diese innerhalb des Bildes liegt, wird der Farbwerte vom gegebenen Bild in das virtuelle Bild übertragen. Für alle noch schwarzen Pixel wird der gleiche Vorgang mit dem anderen gegebenen Bild wiederholt. In Abbildung 6 ist das virtuelle Bild zu sehen, welches aus den beiden gegebenen Bildern, der Disparity Map und der vom Nutzer gewünschten relativen Verschiebung erzeugt wird. Es sind noch einige schwarze Punkte zu erkennen. Das ist auf die Ungenauigkeit der Disparity Map zurückzuführen.

## 6 Nachbearbeiten des virtuellen Bildes

In diesem Schritt werden die schwarzen Pixel eliminiert. Dafür wird jedes Pixel überprüft und die schwarzen Pixel detektiert. Es wird links und rechts von jedem Pixel das am nächsten liegende Pixel gesucht, welches einen Farbwert besitzt und dieser wird anschließend übernommen. In Abbildung 7 ist das virtuelle Bild zu sehen, nachdem die schwarzen Pixel aufgefüllt wurden.

## 7 Zusatzpunkte

### 7.1 Graphische Benutzeroberfläche

Die Graphische Benutzeroberfläche lässt sich über die gui.mlapp starten und basiert auf dem Appdesigner, welcher von Matlab mitgeliefert wird. Mithilfe die-



Abbildung 6: Virtuelles Bild nicht nachbearbeitet (30 % Verschiebung)



Abbildung 7: Virtuelles Bild nachbearbeitet ( 30 % Verschiebung)



ser Benutzeroberfläche können beliebig Bilder geladen werden und ein virtuelles Bild mit einem einstellbarem Blickwinkel erzeugt werden.

### 7.1.1 Hauptmenu

Über load left image und load right image lassen sich das linke und rechte Bild laden. Mit dem Displacement value lässt sich der Verschiebungsfaktor einstellen. Durch run compute wird eine Berechnung eines neuen Bildes basieren auf den geladenen Bildern, der geladenen Parameterdatei und dem eingestellten Verschiebungsfaktor einleiten. Unterhalb des run compute Kopfs wird der Status der Berechnung angezeigt. Weitere Parameterdateien können im Reiter Rectify und DisparityMap erzeugt werden und über load parameter geladen werden. Used Parameter zeigt die aktuelle Parameterdatei an. Falls im Status-Fenster die Fehlermeldung „rectified images will be <10% of originals“ angezeigt wird, muss das Programm nochmals gestartet werden, da schlechte Korrespondenzpunktpaare gefunden wurden.

### 7.1.2 Rectify

Hier lassen sich die einzelnen Schritte zur Bestimmung der Fundamentalmatrix und die Rectification nachvollziehen. So können die Parameter für den Harrisdetektor, die Korrespondenzpunktbestimmung und dem Ramsac-Algorithmus manipulieren und die resultierenden Korrespondenzpunktpaaren in jeder Instanz visualisiert werden. Die Fundamentalmatrix lässt sich mit Compute F basieren auf den Korrespondenzpunktpaaren berechnen und anzeigen. Der komplette Rectificationsalgorithmus lässt sich mit Rectify starten. Mit save parameter werden die Rectification-Parameter und die Parameter zur Bestimmung der Disparity-Map gespeichert.

### 7.1.3 Disparity Map

In diesem Reiter lassen sich die Parameter zur Bestimmung der Disparity Map manipulieren und der Einfluss von unterschiedlichen Algorithmen visualisieren. Auch lässt sich hier der Einfluss des scaling value zeigen.

## 7.2 Laufzeit

Um die Laufzeit zu optimieren haben wir die Berechnung der einzelnen Programmschritte optimiert. Speziell bei der Bestimmung der Disparity Map haben wir mehrere Implementierungsansätze realisiert und diese unter dem Aspekt Zeit und Qualität gegeneinander antreten lassen. Mithilfe des Skalierungsfaktors werden die Bilder vor der Berechnung der Disparity Map runterskaliert und anschließen hochskaliert. So lässt sich der Rechenaufwand für die Disparity Map drastisch Reduzieren und das Qualität des Resultats steuern.

## 8 Generierte Ansichten

Das implementierte Programm verwendet zur Rektifizierung einen von uns programmierten Code. Jedoch ist das finden robuster Korrespondenzpunkte schwierig. Deshalb ist unsere Fundamentalmatrix nicht sehr gut. Aus diesem Grund haben wir zur Generierung der nachfolgenden Bilder auf eine vorhandene Toolbox von A. Fusiello von der DPIA Udine zurückgegriffen. [Fus] Dieser Code liefert ein sehr gutes rektifiziertes Bild. Dadurch wird auch die Rekonstruktion besser.



Abbildung 8: Virtuelles Bild nicht nachbearbeitet (20% Verschiebung)



Abbildung 9: Virtuelles Bild (45% Verschiebung)



Abbildung 10: Virtuelles Bild (70% Verschiebung)



Abbildung 11: Virtuelles Bild (100% Verschiebung)



Abbildung 12: Virtuelles Bild (20% Verschiebung)

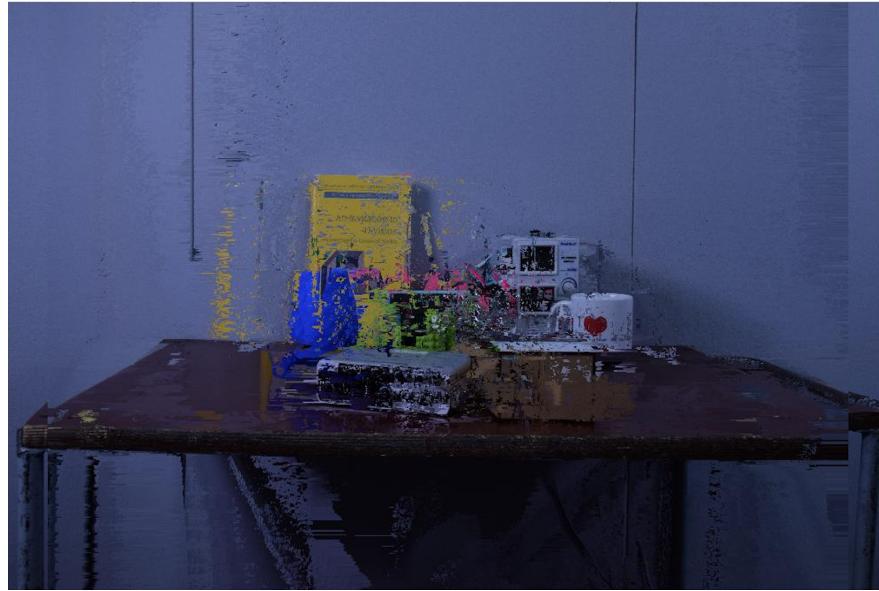


Abbildung 13: Virtuelles Bild (45% Verschiebung)



Abbildung 14: Virtuelles Bild (70% Verschiebung)



Abbildung 15: Virtuelles Bild (100% Verschiebung)



## Literatur

[Fus] FUSIELLO ANDREA (Hrsg.): *Epipolar rectification.*  
<http://www.diegm.uniud.it/fusiello/demo/rect/>, Abruf:  
09. September. 2018

[MCC] MCCORMICK CHRIS (Hrsg.): *Stereo Vision Tutorial - Part 1.*  
<http://mccormickml.com/2014/01/10/stereo-vision-tutorial-part-i/>,  
Abruf: 09. September. 2018