|  |
| --- |
| HAW Landshut |
| Faktorisierungsalgorithmus zur Affinen Rekonstruktion von Kameras und 3d-Punkten |
| Studienarbeit im Fach Bildverstehen |

|  |
| --- |
| Tobias Weiden  12.7.2019  Prof. Siebert |

# Einleitung

Eine Rekonstruktion von 3-dimensionalen Punkten aus 2D-Bildern ist ein weitgehend erforschtes Feld. Hierbei gibt es gerade in einer immer weiter automatisierten Welt immer mehr Use-Cases. Ob selbstfahrende Autos, Roboter, die mit Menschen interagieren, oder einfache Objekt-/ oder Personen-Analysen; Für immer mehr Nutzfelder ist es interessant ein dreidimensionales Modell bspw. der Umwelt zu konstruieren. Hierbei werden verschiedenste Präferenzen gesetzt: Zum einen soll ein Modell möglichst genau oder möglichst schnell erstellt werden, um darauf zu reagieren.

Die meisten Methoden zur Erstellung eines 3D-Modells werden durch eine Verarbeitung von mehreren Bildern aus unterschiedlichen Perspektiven, auf denen bestimmte und bekannte Bildpunkte vorhanden sind.

Hierbei werden Kamera-Matrizen erstellt, die 3D-Punkte in 2D-Punkte wandeln und umgekehrt.

Im Folgenden wird eine Spezialisierung von Kamera-Modellen (Affine Kamera) erläutert, mit dessen Hilfe ein einfacher Algorithmus zur Erstellung von Kamera-Matrizen durchgeführt wird. Dies ist der Faktorisierungsalgorithmus zur Affinen Rekonstruktion von Hartley & Zisserman.

Dieser wird dann mit qualitativ und zeitlich mit der Standard-Kalibrierung von OpenCV verglichen. Hierzu wird zum einen der direkte Reprojektions-Fehler verglichen, welcher die durchschnittliche Differenz zwischen den Original-Punkten und den reprojezierten Punkten darstellt, und der Reprojektions-Fehler bei verschobenen 3D-Punkten. Dies wird mit selbstaufgenommenen Bildern aus verschiedenen Distanzen verglichen.

Die Implementierung erfolgte mittels Python und auch für die OpenCV-Kalibrierung wurde die offizielle Python-Bibliothek genutzt.

# Kameras und Projektion

Durch Projektion kreiert eine projektive Kamera ein flaches Bild aus dreidimensionalen Informationen. Es werden also 3D-Punkte in 2D-Punkte umgewandelt. [1]

## Perspektivische Projektion

Um die Tiefe des dargestellten Raumes zu begreifen, vergleichen wir bei einem zweidimensionalen Bild kameranahe Objekte mit kamerafernen Objekten (im Hintergrund). Objekte mit gleicher Größe in der realen Welt werden je nach Nähe zur Kamera größer (nah der Kamera) oder kleiner (fern der Kamera) dargestellt.

Liegt nun ein Objekt „entlang“ der Tiefe, so wird dieses der Tiefe hin kleiner. Im folgenden Beispiel wird die Bande kleiner, je weiter sie von der Kamera entfernt ist:



Abbildung : Bande, 5 Meter entfernt der rechten senkrechten Spange  
  
Ein Bild entlang einer Bande. Der linke Teil war näher der Kamera und wird somit größer dargestellt als der rechte Teil. Dadurch lässt sich die Tiefe für den Betrachter erahnen.

In diesem Bild tritt auch die Perspektivische Projektion auf: Linien, die in der echten Welt parallel verlaufen, konvergieren im Bild. Diese laufen im Fluchtpunkt zusammen. Dies kann man am Beispiel an dem oberen und unteren Ende der Bande nachvollziehen, wie im Folgenden verdeutlicht: [1]

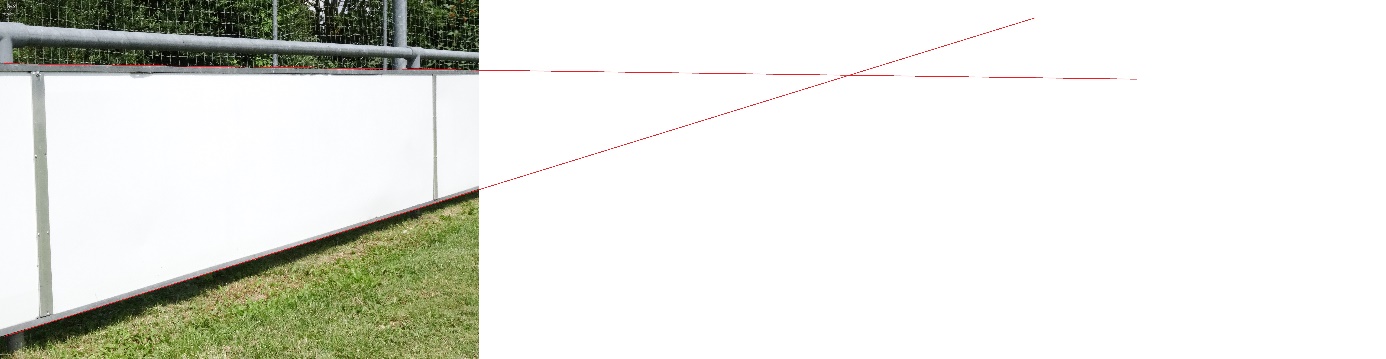


Abbildung : Bande mit eingezeichnetem Fluchtpunkt, 5 Meter Entfernung

## Affine Kamera

TODO: Brennweite erklären

Beachten wir nun die Auswirkungen, wenn die Brennweite und damit auch der Abstand zum Bildinhalt (soll dieser der gleiche bleiben bei gleicher Sensorgröße) erhöht werden:

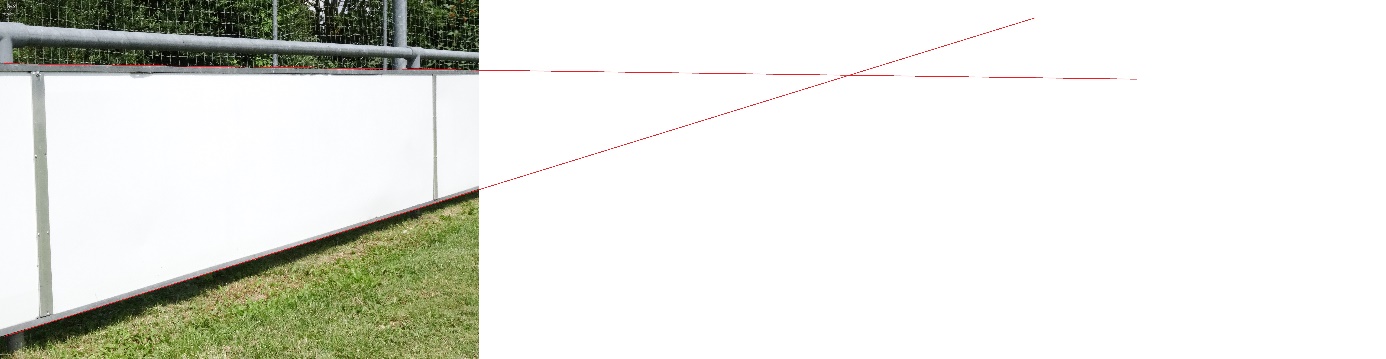


Abbildung : Bande mit Fluchtpunkt, 5 Meter Entfernung, 11mm Brennweite

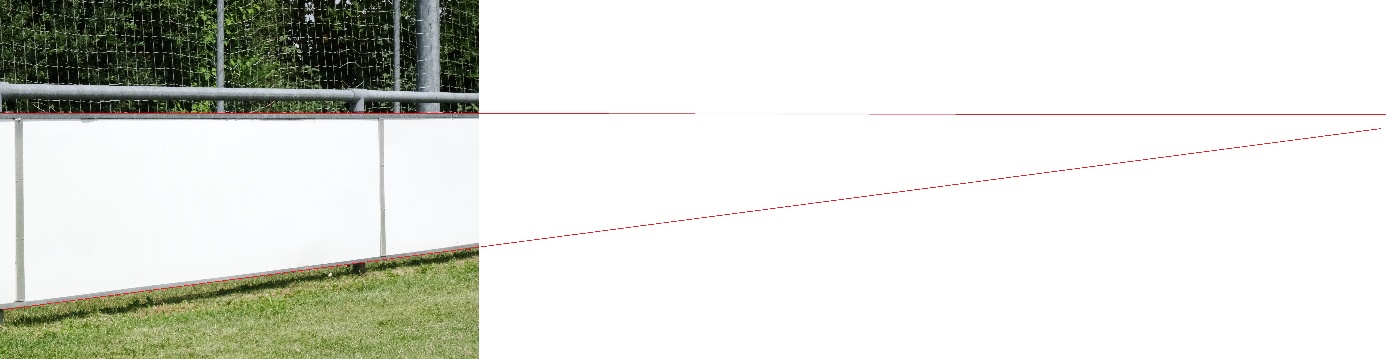


Abbildung : Bande mit Fluchtlinien, 10 Meter Entfernung, 26mm Brennweite

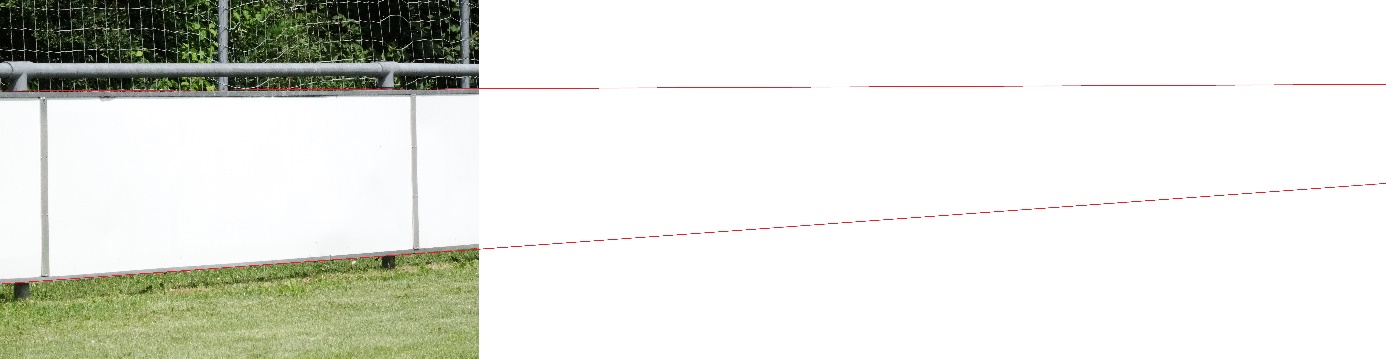


Abbildung : Bande mit Fluchtlinien, 20 Meter Entfernung, 55mm Brennweite

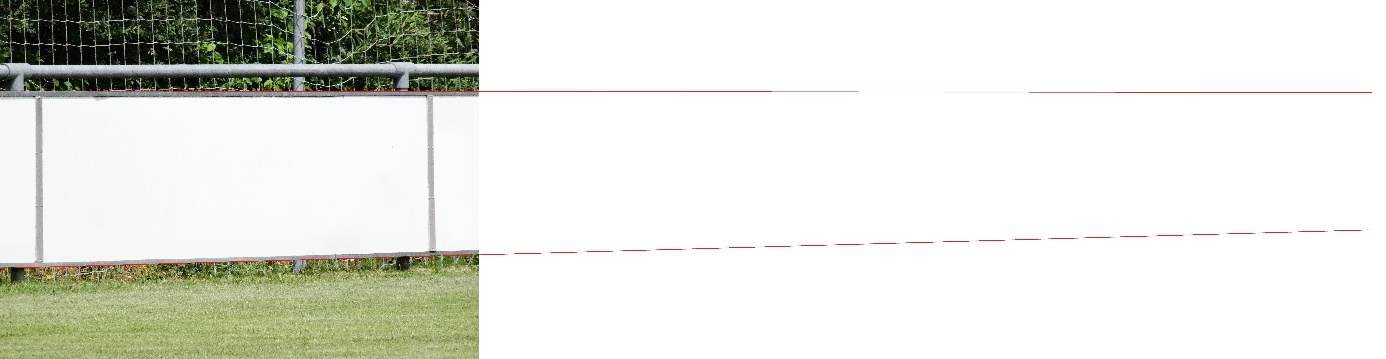


Abbildung : Bande mit Fluchtlinien, 50 Meter Entfernung, 124mm Brennweite

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | M. Obeysekera, *Affine Reconstruction from Multiple Views using Singular Value Decomposition,* The University of Western Australia, 2003. |