

olto3d

OL Karte ins 3D Format

R E V 0 . 1

Flavian Kaufmann
13. Dezember 2018

Maturaarbeit
Betreuungsperson:
Beni Keller

Kantonsschule Zug
2019

Inhaltsverzeichnis

1	Einleitung	3
2	Computer Vision	4
2.1	OpenCV	4
2.2	Anwendungen	4
2.2.1	Bilder entschärfen (Image smoothing)	4
2.2.2	Kantenerkennung (Edge detection)	5
3	Daten	6
3.1	Datenbeschaffung	6
3.2	Aufbereitungsprozesse	6
3.3	Datenverarbeitung	6
4	3D	7
4.1	Transformationen	7
4.2	Collada	7
5	Resultate	8
5.1	Probleme	8
5.2	Diskussion	8
5.3	Wie weiter?	8

1 Einleitung

2 Computer Vision

Wir Menschen haben kaum Schwierigkeiten die Welt um uns herum zu Verarbeiten. Sei es ein Haus, ein Auto oder eine OL-Karte, wir erkennen ohne Problem, um was es sich dabei handelt, oder wie wir etwas zu interpretieren haben. Für einen Computer ist diese Aufgabe jedoch nicht selbstverständlich. Es wurde jedoch eine Vielfalt von Verfahren entwickelt, damit ein Computer fähig ist, anhand von digitalen Bildern, Entscheidungen zu treffen, dreidimensionale Räume zu rekonstruieren oder auch Bilder wieder weiterzuverarbeiten. Diese Prozesse sind aber nicht immer einfach zu verstehen. In der Computersprache werden Bilder als Zahlenraster repräsentiert. Dabei entsprechen die einzelnen Werte den Farbtönen der dazugehörigen Pixels (Siehe Abbildung 1). [2,5] Computer Vision beruht zum Beispiel darauf, diese Raster bzw. Matrizen mithilfe von mathematischen Operationen, oft Matrixmultiplikationen, weiterzuverarbeiten.

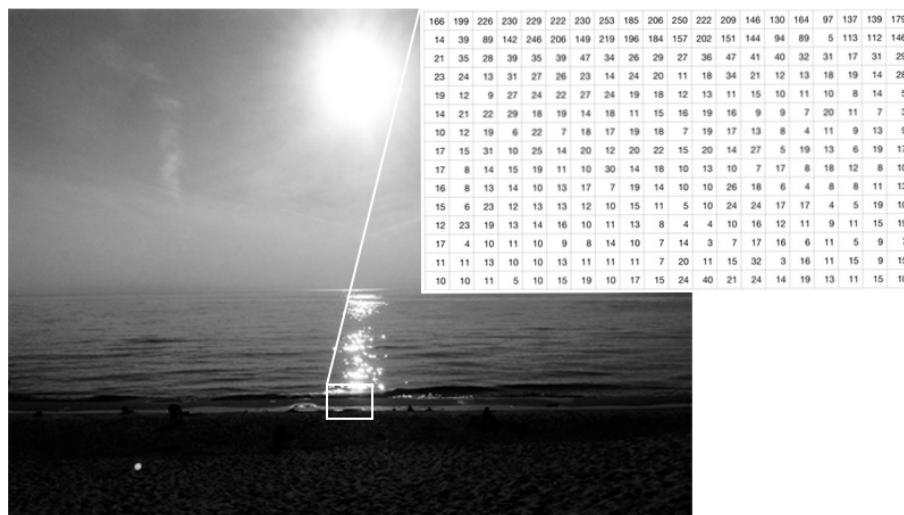


Abbildung 1: Rasterbild

2.1 OpenCV

Die Verarbeitung von visuellen Daten am Computer ist weit verbreitet und findet in vielen Sparten der Informatik seine Anwendungen. Um Projekte, welche das Verlangen von Computer Vision haben, zu realisieren, ist eine Mehrzahl von Bibliotheken vorhanden, eine davon OpenCV [3]. OpenCV ist eine open source Bibliothek, die primär in C und C++ geschrieben wurde. Es sind jedoch APIs auf verschiedenen Entwicklungsplattformen wie iOS oder Windows erhältlich. Zudem kann die Bibliothek auch mit den meisten aller häufigst gebrauchten Programmiersprachen wie Python, Java, etc. verwendet werden. [5]

2.2 Anwendungen

2.2.1 Bilder entschärfen (Image smoothing)

Zu den häufigst angewendeten Bildbearbeitungsalgorithmen gehören Entschärfungen. Diese werden vor allem zum Entfernen von Bildrauschen angewendet. Unter

Bildrauschen versteht man Abweichungen und Störungen der Farbwerte der Pixels eines digitalen Bildes im Vergleich zur Realität. Diese Störungen entstehen hauptsächlich bei Aufnahmen unterbelichteten Bilder oder bei der digitalen Kompression eines Bildes, um dessen Dateigrösse zu vermindern. Um diese Farbabweichungen wieder herzustellen, dh. zu interpolieren, wird jedes neu zu berechnende Pixel $g(i, j)$ mit einem gewichteten Durchschnitt des alten Pixels und dessen darumlegenden berechnet. Dies wird auch als linearer Filter (1) bezeichnet. Die Koeffizienten der Gewichtungen werden mit einer Matrizze (auch Kernel genannt) $h(k, l)$ beschrieben. Ein solcher Kernel kann wie folgt (2) aussehen. Dies ist jedoch einer der einfachsten linearen Filter und würde lediglich das arithmetische Mittel der benachbarten Pixel berechnen. [4, 5]

$$g(i, j) = \sum_{k, l} f(i + k, j + l) h(k, l) \quad (1)$$

[4]

$$h(k, l) = \frac{1}{k \cdot l} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad (2)$$

[4]

In der Praxis gibt es geeignetere Filter, um eine gewünschte Bildrausch Reduzierung zu erlangen. Einer der bekannteren wäre der Gaussian Blur, welcher die statistische Normalverteilung bzw. Gauss-Verteilung verwendet, um ein Mittel zu berechnen, benannt nach dem Mathematiker Carl Friedrich Gauss. [1]

2.2.2 Kantenerkennung (Edge detection)

3 Daten

3.1 Datenbeschaffung

3.2 Aufbereitungsprozesse

3.3 Datenverarbeitung

4 3D

4.1 Transformationen

4.2 Collada

5 Resultate

5.1 Probleme

5.2 Diskussion

5.3 Wie weiter?

Literatur

- [1] Larry Freeman. Carl friedrich gauss. <http://fermatslasttheorem.blogspot.com/2005/06/carl-friedrich-gauss.html>, Jun 2005. Besucht am 3. Sep 2018.
- [2] Richard Szeliski. *Computer Vision - Algorithms and Application*, pages 3–15. Springer, 2011.
- [3] OpenCV Team. Opencv. <https://opencv.org>, Sep 2018. Besucht am 3. Sep 2018.
- [4] OpenCV Team. Smoothing images. https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html, Sep 2018. Besucht am 3. Sep 2018.
- [5] Gary Bradski und Adrian Kaehler. *Learning OpenCV - Computer Vision with the OpenCV Library*. O'REILLY, 2008.

Abbildungsverzeichnis

1	Rasterbild	4
---	----------------------	---