Computer Vision Challenge 2019

G23
Dhaouadi Oussema, Lin Kang, Patsch Constantin, Sukianto Tobias, Yan Kevin Tong

19. Juni - 19. Juli

1 Problemstellung und Methodik

In dieser Challenge soll eine Disparitäten Schätzung durchgeführt werden für eine vorliegende Stereoszene. Hierbei geht es darum für jedes Pixel im linken Stereobild I0, das korrespondierende Pixel im rechten Stereobild I1 zu finden (oder umgekehrt). Mit Hilfe der Disparity Map kann dadurch die Tiefe ermittelt werden. Unser Algorithmus für die Schätzung der Disparity Map basiert grundlegend auf *Block-Matching* (BM) [**?** 1, 6]. Gleichzeitig versuchen wir dabei möglichst viele Elemente dieser Lehrveranstaltung wieder zu verwenden. Unsere Methode wird getestet am Middebury und ETH Zürich Stereo Dataset [5, 4] durch Vergleich mit der Ground-Truth. Als Maß für die Qualität dient das Peak-Signal-to-Noise-Ratio(PSNR). Auch haben wir eine GUI realisiert, welche die Bedienung erleichtern soll.

2 Berechnung der Rotation und der Translation

Für die Berechnung der Rotationsmatrix $\mathbf{R}$ und des Translationsvektors T benötigen wird die Kalibrierungsmatrix $\mathbf{K}$ der Kamera(s) und Korrespondenzpunktpaare zwischen den Stereobildern. Dazu berechnen wir in beiden Bilder zuerst Merkmalspunkte mithilfe des Harris-Eckendetektors. Anschließend werden über Normalized Cross Correlation (NCC) Korrespondenzpunktpaare zwischen den Merkmalen gefunden. Danach wenden wir den RANSAC Algorithmus an, um *robuste* Korrepondenzen zu extrahieren. Mit Hilfe des normalisierten Achtpunktalgorithmus [2] berechnen wir im RANSAC-Schritt die Fundamentalmatrix und anschließend die essentielle Matrix $\mathbf{E}$. Dies liefert jeweils zwei Lösugen für $\mathbf{R}$ und T. Die einzig geometrisch logische Transformation finden wir heraus, indem die Tiefen der robusten Korrespondenzpunktpaare geschätzt werden und wählen dasjenige ($\mathbf{R}$,T) Paar mit der meisten positiven Tiefeninformation. Ein Blockdagramm hierfür ist zusehen in Abb. 1. Für unseren Algorithmus können die robusten Korrespondenzpunktpaare wiederverwendet weden um eine Schätzung der maximalen Disparität zu liefern.

3 Disparitäten Schätzung

Der Algorithmus, wie zu sehen im Blockdiagramm in Abb 2, kann aufgeteilt werden in drei wesentliche Schritte. Im ersten Schritt werden die Stereobilder in einen Preprocessing-Step in Grauwertbilder umgewandelt, danach runterskaliert, falls eine festgelegte Bildgröße überschritten wird, und anschließend optional entweder eine Gradientenberechnung per Sobelfilter, eine Normalisierung der Stereobilder oder eine Histogram-Equalization für Kontrasterhebung durchgeführt.

Im zweiten Schritt erfolgt die eigentliche Disparitätenschätzung durch Block-Matching. Hierbei wird für jedes Pixel im linken Stereobild ein Bildsegment ausgewählt, sodass das betrachte Pixel im Zentrum dieses Segments liegt. Im rechten Stereobild wird nach jenem Bildsegment gesucht, dass dem des linken Bildsegments „am meisten gleicht“. Das Pixel im Zentrum des rechten Bildsegmentes wird somit als das korrespondierende Pixel zum Pixel des linken Segments zugewiesen. Welches Segment „am meisten gleicht“ wird formell über das Minimum einer Kostenfunktion bestimmt, die in unserer Methode als Summe der quadrierten Differenzen oder kurz SSD (engl. Sum of Squared Differences) definiert ist. Für ein linkes Bildpixel mit Koordinaten (x, y), quadratische Segment mit Kantenlänge $s + 1$ und betrachtete Disparität d kann die SSD wie folgt ausgerechnet werden:

$$\mathrm{SSD}(x, y, d) = \sum_{i=x-s}^{x+s} \sum_{j=y-s}^{y+s} (\mathrm{I0}(i, j) - \mathrm{I1}(i - d, j))^2 \tag{1}$$

Mit SSD werden die einzelnen Pixel Intensitäten beider Bildsegmente voneinander abgezogen, quadriert, und anschließend akkumuliert. Das Segment des korrespondierenden Pixel im rechten Bild weist somit die kleinste SSD auf. Mit der Annahme das die Bilder parallele Epipolarlinien auf der selben Höhe haben, und ein gegebenes Disparitäten-Intervall $[d_{\min}, d_{\max}]$ gegeben ist, kann man die Suche auf diesem Intervall beschränken und musst nicht entlang der kompletten horizontalen Achse suchen. Die geschätzte Disparität $\hat{d}$ des linken Pixels $p = (x, y)$ is damit

$$\hat{d} = \arg\min_{d} \quad \mathrm{SSD}(x, y, d). \tag{2}$$

Diese Berechnungen können sehr rechenaufwendig sein, besonders für große Bilder. Für ein quadratisches Segment mit Kantenlänge S und Bilddimension $M \times N$, wobei M die Höhe in Pixel und N die Breite in Pixel entspricht, errechnet sich die Rechenaufwand zu etwa $\mathcal{O}(S^2MN(d_{\max} - d_{\min}))$. Um den Rechenaufwand zu verringern wird daher BM auf das runterskalierte Bild angewendet. Da sich die Segmente überlagern, können die Kosten rekursiv aktualisiert werden und somit effizent berechnet werden. Der Rechenaufwand reduziert sich damit auf $\mathcal{O}(SMN(d_{\max} - d_{\min}))$ und steigt nur noch linear mit der Größe des Segments. Die Disparity Map nach diesen Schritt schaut etwa so aus wie in Abb.

Im dritten Schritt wird ein Postprocessing-Step oder Refinement-Step durchgeführt. Da die gewonnene Disparity Map durch reines Block-Matching fehlerbehaftet ist, sollen hier diese korrigiert werden. Dazu wird erstmal ein Uniquness-Check durchgeführt, der als Maß für die Qualität der geschätzten Disprität eines Pixels dienen soll. Wenn $\tilde{d}(x, y)$ die Disparität zur zweit kleinsten SSD eines Pixels p korrespondiert, dann ist die geschätzte Disparität *unique* für ein Uniquness-Threshold τ falls gilt

$$\mathrm{SSD}(x, y, \tilde{d}) < (1 + \tau)\mathrm{SSD}(x, y, \hat{d}). \tag{3}$$

Idealerweise sollte für eine Disparität d des Pixels p im linken Bild gelten, dass das korrespondierende Pixel im rechten Bild dieselbe Dispartiät aufweist. Diese Eigenschaft nennen wir *consistent* und soll für die ganze Disparity Map überprüft werden. Für ein Distance-Threshold τ_d ist die Disparität eines Pixels consistent falls gilt

$$\mathrm{D0}(x, y) - \mathrm{D1}(x - d, y) < \tau_d, \tag{4}$$

wobei D0 die Disparity Map des linken Bildes entspricht und D1 die des rechten Bildes. Die Disparitäten der Pixel, welche nicht consistent sind werden als *Occlusion* klassifiziert. Occlusion sind solche Merkmale in einem Stereobild, welche verdeckt sind im anderen Bild.
Um die Disparity Map auf einem Subpixel-Level zu erhalten, wird für jedes Pixel mit zugehörigen Disparität $\hat{d}$ und seine zwei benachbarten Disparitäten und Kosten, $\mathrm{SSD}(x, y, \hat{d}-1)$ und $\mathrm{SSD}(x, y, \hat{d}+1)$,

parabolisch interpoliert,

$$\hat{d} \leftarrow \hat{d} + \frac{\mathrm{SSD}(x, y, \hat{d}+1) - \mathrm{SSD}(x, y, \hat{d}-1)}{2 \cdot (\mathrm{SSD}(x, y, \hat{d}-1) - 2\mathrm{SSD}(x, y, \hat{d}) - \mathrm{SSD}(x, y, \hat{d}+1))}. \tag{5}$$

Anschließend füllen wir die Occlusion mit jeweils der nächstliegenden linken Disparität. Zuletzt soll noch Rauschen herausgefiltert werden, hierfür benutzen wir den Medianfilter, der auch Salt-and-Pepper Filter genannt wird. Damit die Kanten nicht zu sehr verwischen beim rauschfiltern, kategorisieren wir zuerst mit Hilfe des Harrisdetektors jedes Pixel entweder als Ecke, Kante oder Fläche. Hierfür wird der Harrisdetektor genutzt. Somit kann man z.B auf Flächen größere Filter benutzt, während auf Kanten kleinere verwendet. Bevor die Disparity Map ausgegeben wird, wird diese noch zurück auf die Originalgröße skaliert.

4 3D Rekonstruktion der Szene

Für die 3D Rekonstruktion brauchen wir die korrespondierenden Welktkoordinaten (X, Y, Z) für jeden Pixel (x, y). Um aus der Disparität d in Pixelkoordinaten die Tiefen Z zu schätzen, benutzen wir die Formel in [7]

$$Z = \frac{\mathrm{baseline} \cdot f}{d + d_{\mathrm{offs}}},$$

Z ist dabei in mm, die baseline ist die Basislinie der Kameras in mm, f die Brennweite in Pixel und d_{offs} der x-Unterschied zwischen den Brennpunkten der Kameras in Pixel. Um (X, Y) zu berechnen, gebrauchen wir folgende Umrechnungen:

$$X \sim \mathbf{K}^{-1}x \cdot \frac{Z}{f}, \quad Y \sim \mathbf{K}^{-1}y \cdot \frac{Z}{f}. \tag{6}$$

Da die Brennweite nur in Pixel gegeben ist und kein Aufschluss zur exakten Sensorgröße gefunden wurde, stimmt (6) nur bis auf Skalierung. Eine Beispielrekonstruktion anhand einer Ground Truth is zusehen in Abb. 4.

5 Peak-Signal-To-Noise-Ratio

Um das Peak-Signal-To-Noise-Ratio (PSNR) zu berechnen, normieren wir zuerst die Ground-Truth D_{truth} und die geschätzte Disparity Map D auf den Wertebereich $[0, 255]$. Danach berechnen wir den Mean-Square-Error (MSE) mit Hilfe der Formel

$$\mathrm{MSE} = \frac{\sum_{x=1,y=1}^{M,N}[D(x, y) - D_{\mathrm{truth}}(x, y)]^2}{MN}, \tag{7}$$

wobei M und N die Dimensionen der Matrix sind. Dieses verwenden wir um das PSNR wie folgt auszurechnen

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{8}$$

wobei wir 255 in die Formel einsetzen, da uint8 verwendet wird.

6 Graphical User Interface

Das Design der GUI wurde mit Hilfe von „GUIDE“ [3] erstellt. Zuerst wird ein Ordner mit zwei Ansichten (.png), einer Tiefenkartendatei (.pmf) und der Datei „calib.txt“ geöffnet. Anschließend stellt der Benutzer die Parameter entsprechend der Szene ein. Danach gibt ein Mausklick auf die Schaltfläche ≪Disparity Map≫nach wenigen Sekunden die Tiefenkarte, den Translationsvektor (Baseline), die Rotationsmatrix, die Laufzeit und den PSNR-Wert aus. Zudem wurde eine Konsole für Debug-Zwecke hinzugefügt.
Unteranderem sollte beachtet werden, dass jene den Typ des angezeigten Bildes ändern kann. Tatsächlich könnte die Stereoansicht entweder in Farbe oder in Rot/Cyan dargestellt werden. Die Disparitätskarte kann in Graustufen, in Farben oder im 3D-Raum mit den realen Farben der Pixel dargestellt werden.
Ein Beispiel für die GUI ist in der Abbildung 5 dargestellt.

7 Abbildungen

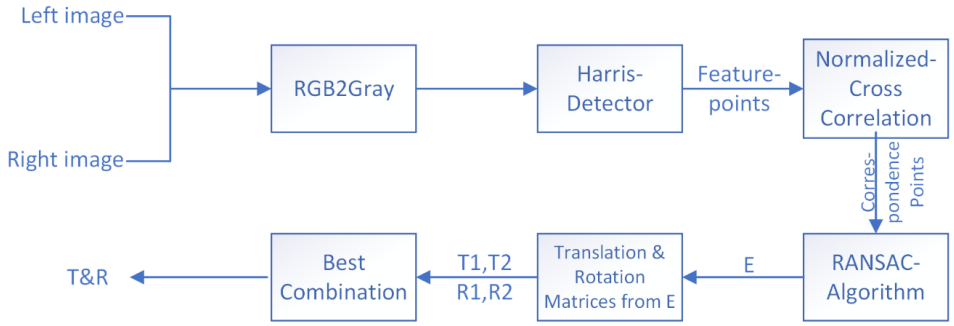


Abbildung 1: Blockdiagramm zur Berechnung der Rotation und Translation

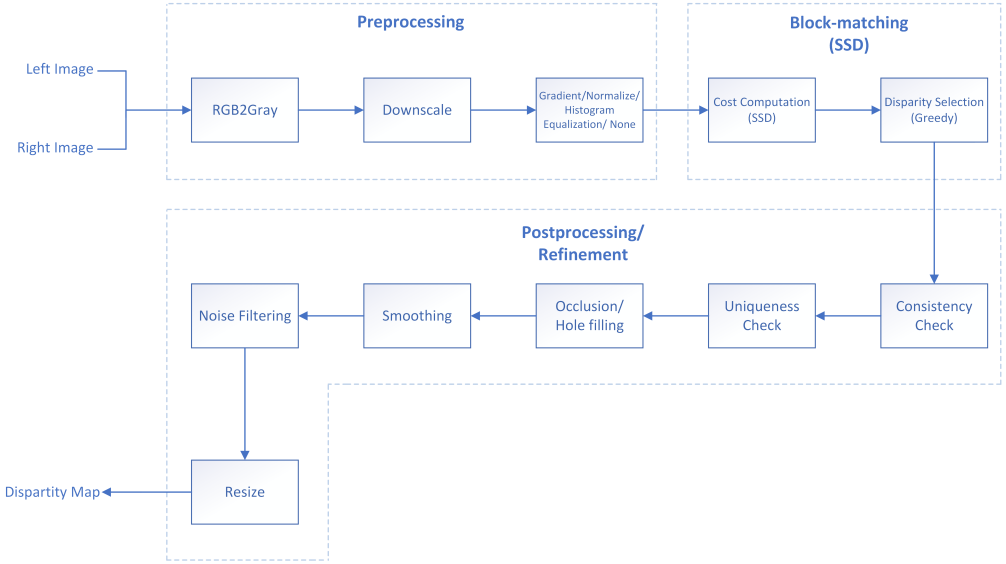
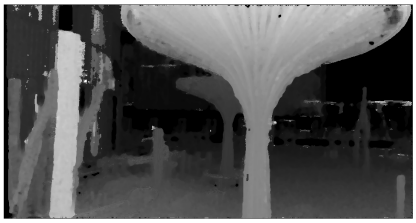


Abbildung 2: Blockdiagramm zur Berechnung der Disparity Map

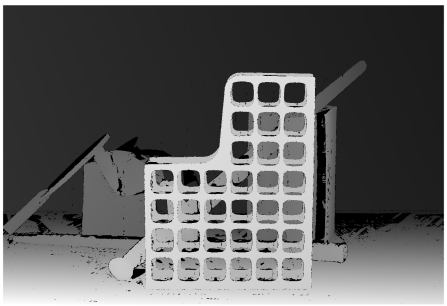
(a) Playground Ground Truth



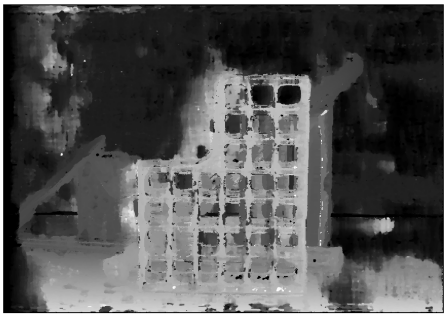
(b) Playground PSNR = 12.83 dB

(c) Terrace Ground Truth

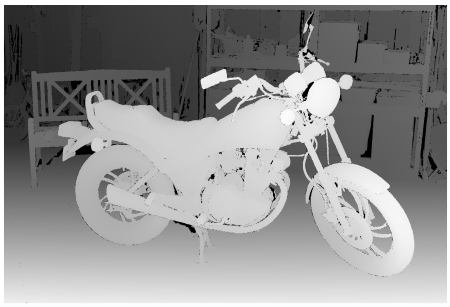
(d) Terrace PSNR = 9.3 dB



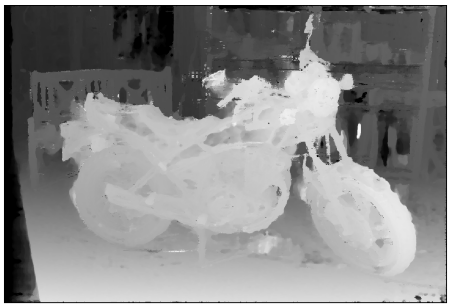
(e) Sword Ground Truth



(f) Sword PSNR = 13.25 dB



(g) Motorcycle Ground Truth



(h) Motorcycle PSNR = 15.43 dB

Abbildung 3: Disparity Map Schätzung mittels unserer Methode (rechts) vs Ground Truth (links) angewendet am Middlebury und ETH Zürich Dataset

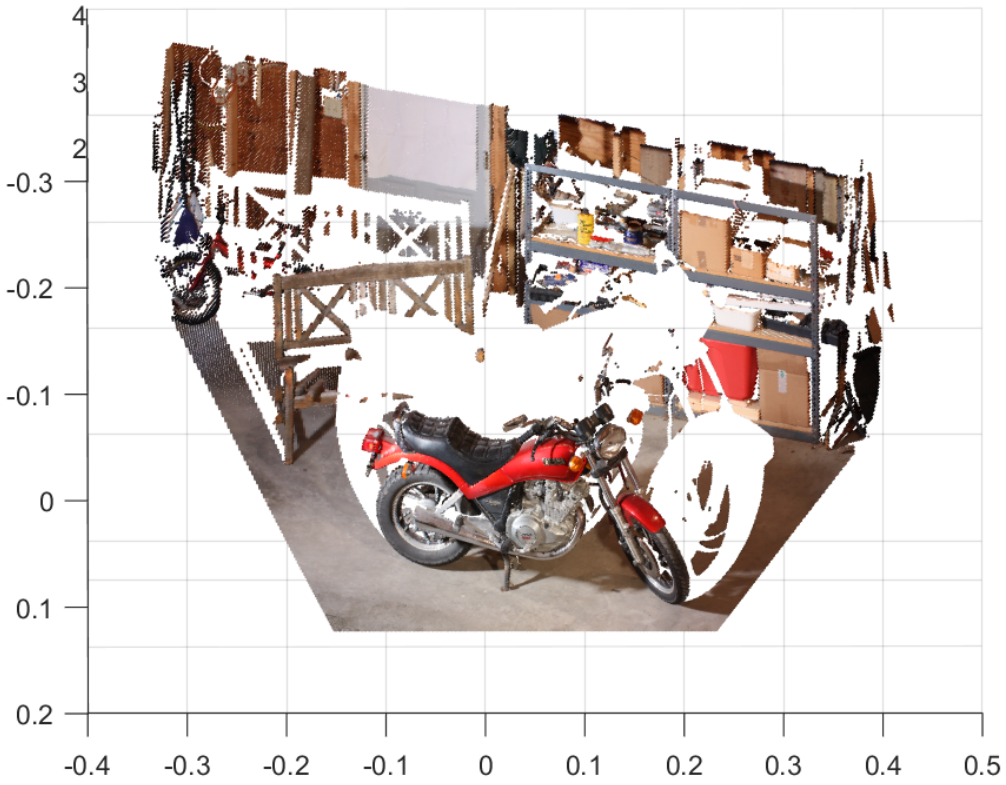


Abbildung 4: 3D Rekonstruktion der Motorcycle Szene mittels Gorund Truth

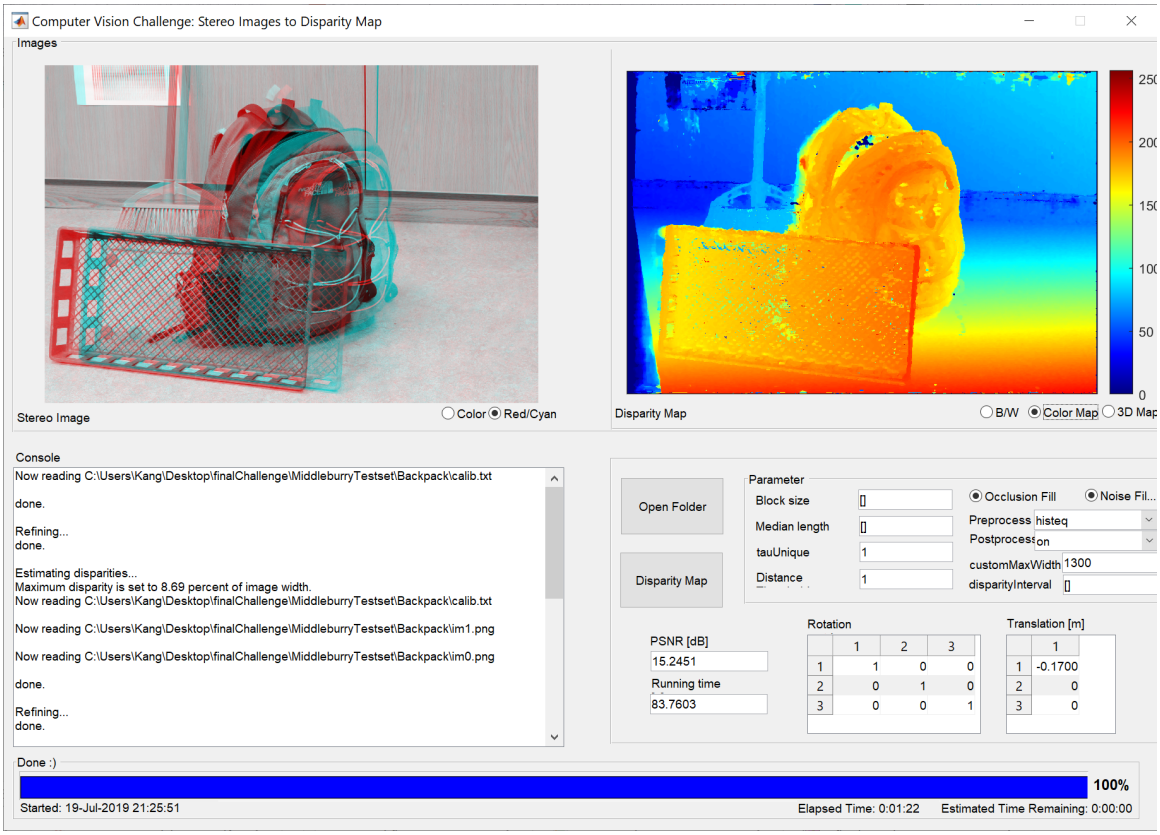


Abbildung 5: Graphical User Interface

Literatur

[1] Rostam Affendi Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016, 2016.

[2] Richard I Hartley. In defence of the 8-point algorithm. In *Proceedings of IEEE international conference on computer vision*, pages 1064–1070. IEEE, 1995.

[3] Mathworks. guide. `https://ch.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html`.

[4] Thomas Schöps; Torsten Sattler; Marc Pollefeys. Eth stereo dataset. `https://www.eth3d.net/datasets`.

[5] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[6] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[7] Daniel Scharstein; Richard Szeliski. Middleburry stereo dataset. `http://vision.middlebury.edu/stereo/data/scenes2014/`.