

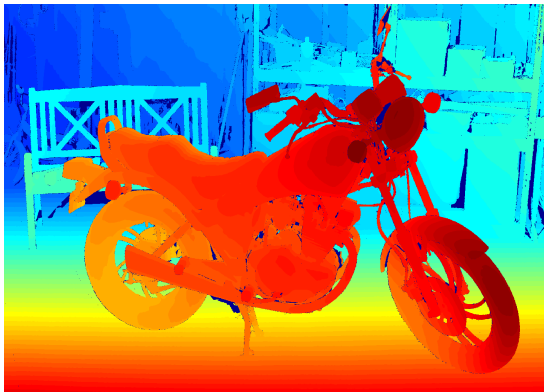
Computer Vision Challenge

Klaus Diepold

Stefan Röhl

Lehrstuhl für Datenverarbeitung
Technische Universität München

18. Juni 2019



Zusammenfassung

Die Computer Vision Challenge besteht in diesem Jahr darin, aus einem Stereobildpaar eine Disparity Map zu schätzen.

1 Datenset

Die zu verarbeitenden Daten entsprechen den Middlebury Stereo Datasets [1]. Ihnen werden vier verschiedene Szenen auf Moodle zur Verfügung gestellt. Jeder Ordner enthält zwei Dateien *im0.png* und *im1.png*, welche Farbbilder oder Grauwertbilder sein können. Zudem ist eine Datei *calib.txt* enthalten, welche die Kalibrierungsmatrix, die Baseline und zum Teil weitere Informationen enthält. Damit Sie die Ergebnisse Ihrer Arbeit testen können, liegt jedem Bildpaar eine Disparity Map bei, welche in der Datei *disp0.pfm* gespeichert ist. In Abbildung 1 ist der Inhalt eines Szenenordners exemplarisch dargestellt. Die genauen Spezifikationen der einzelnen Dateien finden Sie auf der Projektseite des [Middlebury Stereo Datasets](#).

Um Ihren Algorithmus robuster zu machen, können Sie auch selbst noch weitere Datensätze ausprobieren.

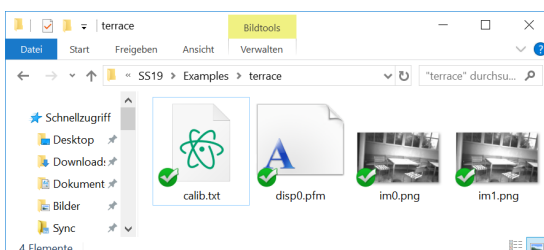


Abbildung 1: Exemplarischer Inhalt eines Szenenordners

2 Spezifikation

Im Folgenden finden Sie die detaillierte Aufgabenbeschreibung der Computer Vision Challenge 2019. Wir bitten Sie, diese genau zu lesen.

2.1 Aufgaben

Diese Aufgaben sind verpflichtend für alle Gruppen.

1. **Disparity Map:** Schreiben Sie eine Funktion $[D, R, T] = \text{disparity_map}(\text{scene_path})$, welche als Eingabeparameter den Pfad zu einem Szenenordner erhält. Die Funktion soll dann alle benötigten Quelldateien¹ selbstständig öffnen und daraus die Disparitäten D berechnen. Der Rückgabewert D soll vom Datentyp *uint8* im Wertebereich $[0, 255]$ sein. Zusätzlich soll die Funktion die Beschreibung der euklidischen Bewegung als Rotationsmatrix R und Verschiebungsvektor T zurückgeben. T soll dabei die Einheit Meter haben.
2. **Verification:** Schreiben Sie eine Funktion $p = \text{verify_dmap}(D, G)$, welche das Peak Signal-To-Noise Ratio (PSNR) zwischen einer geschätzten Disparity Map D und der Ground Truth G berechnet. Diese Funktion soll zwei Matrizen als Eingabeparameter erhalten. Stellen Sie hierbei zunächst sicher, dass die beiden Matrizen auf das Intervall $[0, 255]$ normiert sind. Das errechnete PSNR wird in der Einheit dB angegeben.
3. **Unittests:** Vervollständigen Sie die Datei *test.m*. Diese Datei soll folgende Eigenschaften anhand verschiedener Testcases prüfen. Nutzen Sie dazu die MATLAB Unittests.
 - (a) **check_toolboxes:** Überprüfen Sie für die Dateien *challenge.m*, *disparity_map.m* und *verify_dmap.m*, ob Sie auch wirklich keine Toolbox verwenden sondern nur grundlegende MATLAB Funktionen.
 - (b) **check_variables:** Überprüfen Sie, dass alle geforderten Variablen in der Datei *challenge.m* nicht leer bzw. größer als Null sind.
 - (c) **check_psnr:** Vergleichen Sie ihre eigene Implementierung des PSNR mit der in der Image Processing Toolbox und prüfen Sie, ob das Ergebnis innerhalb einer angemessenen Toleranz liegt.

¹Bitte beachten Sie, dass die *disp0.pfm* Dateien bei der Auswertung nicht vorliegen werden.

4. **Challenge:** Vervollständigen Sie die Datei *challenge.m* und achten Sie darauf, dass alle geforderten Variablen nach dem Aufruf gesetzt sind. Starten Sie den Ausführungstimer bevor Sie die Disparity Map berechnen und anschließend mit der Ground Truth vergleichen. Speichern Sie den Wert des Timers in der variable `elapsed_time`, nachdem die Berechnungen abgeschlossen sind. Geben Sie die Resultate für `R`, `T`, `p` und `elapsed_time` in der Commandline in angemessener Formatierung aus und zeigen Sie die gefundene Disparity Map `D` als Bild an.
5. **Plots:** Erstellen Sie die Disparity Maps für die bereit gestellten Szenenordner und fügen Sie diese in ihr Dokumentations-PDF ein.
6. **GUI:** Schreiben Sie eine grafische Benutzeroberfläche, welche es erlaubt, einen Szenenordner auszuwählen und die darin enthaltenen Bilder anzeigt. Per Button soll dann die Disparity Map berechnet und ebenfalls angezeigt werden. Zusätzlich sollen auch die Werte für `R`, `T` und `p` in der GUI dargestellt werden. Die GUI soll mit dem Ausführen der Datei *start.gui.m* gestartet werden können.

2.2 Zusatzfunktionen

Sie können einen eventuell fehlenden Punkt aus den vorherigen Aufgaben ausgleichen, indem Sie ihr Programm mit Zusatzfunktionen ausstatten. Im Folgenden geben wir ein paar Beispiele für Zusatzfunktionen an:

- **3D Plot:** Stellen Sie die Szene in einem 3D Plot dar und färben Sie die Raumpunkte entsprechend der Farbe der dazugehörigen Pixel ein.
- **Ausführungszeit:** Gestalten Sie ihren Code performant und erzielen Sie eine überdurchschnittlich kurze Ausführungszeit.
- ...

Weitere Ideen für Zusatzfunktionen können Sie gerne mit uns in der Fragestunde besprechen.

2.3 Report

Jede Abgabe wird komplettiert durch die Dokumentation der Software, sowie die Darstellung der geforderten Plots in einem PDF.

- **Readme:** Fügen Sie ihrem Code eine *Readme.txt* Datei bei, in der Sie beschreiben, wie ihre Software auszuführen ist und weisen Sie explizit auf Zusatzfunktionen hin.
- **Dokumentation:** Erstellen Sie ein PDF-Dokument *doku-GXX.pdf*, indem Sie die Funktionsweise ihres Programms kurz erklären und legen Sie auch die Quellen dar, die sie für Ihren Ansatz konsultiert haben. Nutzen Sie dazu gerne auch mathematische Beschreibungen, 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. Setzen sie auch die Namen aller am Projekt beteiligten Personen an eine gut sichtbare Stelle im Dokument.

- **Kommentare:** Kommentieren Sie Ihre Schritte ausführlich. Die Sprache des Programms und der Kommentare kann Deutsch oder Englisch sein.

3 Auswertung

Ihr Code wird nach der Abgabe auf seine Komplettheit und Ausführbarkeit auf den bereits bekannten als auch auf unbekannten Szenen überprüft. Weitere Bewertungsaspekte sind die Erfüllung aller im Abschnitt 2.1 definierten Funktionen, sowie die Vollständigkeit der Ausgabe und die geforderten Variablen. Des Weiteren hat die Qualität und Vollständigkeit der abgegebenen Dokumente, wie sie in Abschnitt 2.3 beschrieben sind, Einfluss auf diese Prüfungsleistung. Fehler in den Pflichtanteilen können Sie durch Erfüllen von Zusatzfunktionen ausgleichen. Hierbei kann maximal ein Punkt ausgeglichen werden.

4 Gruppen

Die Computer Vision Challenge ist eine Programmierarbeit, die in Gruppen bestehend aus vier bis fünf Personen durchgeführt wird. Tragen Sie sich als Gruppe über das Modul [Gruppenwahl](#) in Moodle ein und stellen Sie sicher, dass nur die beteiligten Personen in dieser Gruppe eingetragen sind. Tragen Sie die Gruppennummer, die Namen der Teilnehmer und die entsprechenden Email-Adressen aus Moodle in die dafür vorgesehenen Variablen im Skript *challenge.m* ein, da Ihnen sonst keine Note zugeordnet werden kann. Ihre eindeutige Email-Adresse finden Sie in Ihrem Moodle-Profil. Vergleichen Sie hierzu Abbildung 2.

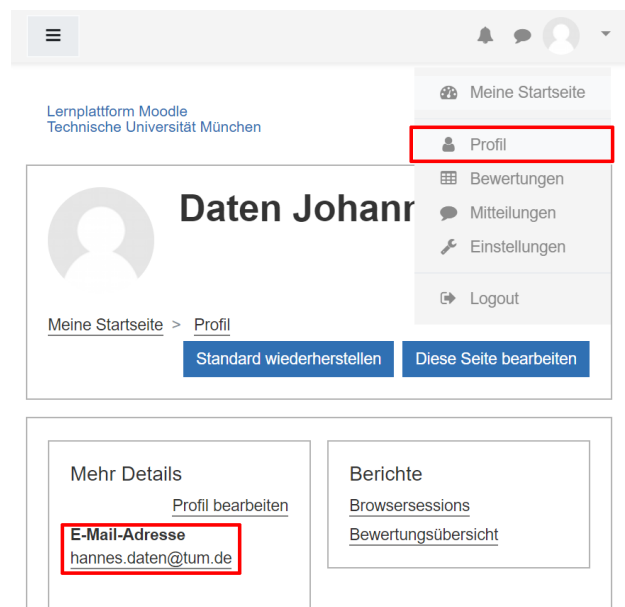


Abbildung 2: Eindeutige Email-Adresse im Moodle-Profil

5 Abgabe

Geben Sie nur funktionierenden Code ab und verwenden Sie nur relative Pfade, da sonst keine Bewertung vorgenommen werden kann. Testen Sie ihr Programm am besten vorher auf einem unbekannten Rechner z.B. im Eikon, ob dieses dort auch ausführbar ist und zum gewünschten Ergebnis führt. Zum Testen wird die Matlab Version 2019a verwendet. Wenn Sie mit einer stark abweichenden Matlab Version arbeiten, vergewissern Sie sich, dass keine grundlegenden Funktionen im Vergleich zu dieser Version geändert wurden. Komprimieren Sie Ihre Abgabe in einem *.zip-Archiv und geben Sie diese Datei auf Moodle für Ihre Gruppe ab. Abbildung 3 zeigt, wie der Inhalt Ihrer Abgabe beispielsweise aussehen könnte.

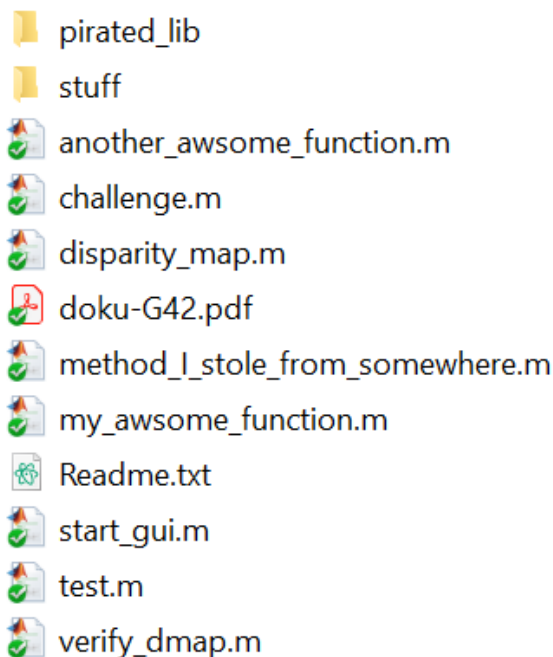


Abbildung 3: Beispiel eines abgegebenen Archivs

Abgabedetails:

- **Abgabefrist:** Die Abgabefrist ist der **19.07.2019** um **23:55 Uhr**. Nachträgliche Abgaben können nicht berücksichtigt werden!
- **Abgabeformat:** Sie können nur *.zip-Dateien abgeben. Achten Sie auf eine eindeutige Benennung nach dem Schema *GXX.zip* wobei *XX* für Ihre Gruppennummer steht. Sind Sie also in Gruppe 42 geben Sie eine Datei ab, die *G42.zip* heißt. Die Abgabe von Gruppe 2 heißt entsprechend *G02.zip*.
- **Ordner:** Stellen Sie sicher, dass beim Entpacken der *.zip-Datei der Inhalt in genau einem Unterordner landet. Sie können dies z.B. beim Entpacken auf Linux mit dem Befehl `unzip` prüfen. Eine exemplarisch entpackte Abgabe sehen Sie in Abbildung 4.
- **Bilder:** Ihre Abgabe sollte keine übermäßig großen Dateien oder gar Szenenordner enthalten, diese haben wir bereits selbst vorliegen. Die Uploadgröße ist auf 5MB beschränkt.

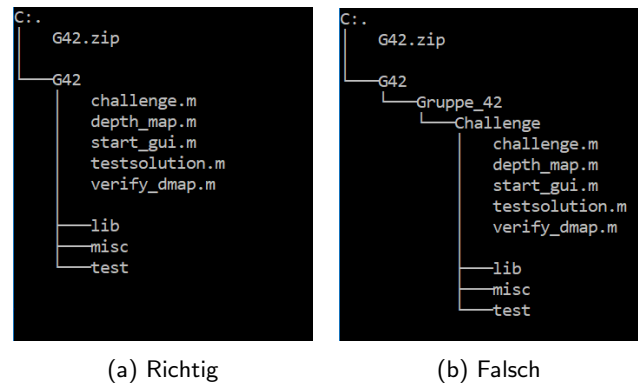


Abbildung 4: Ordnerstruktur der Abgabe nach dem Entpacken der *.zip-Datei.

Literatur

- [1] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Kraethwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Pattern Recognition* (X. Jiang, J. Hornegger, and R. Koch, eds.), (Cham), pp. 31–42, Springer International Publishing, 2014.