



Computer Vision

Martin Kleinstaubert, Julian Wörmann

1. Juni 2016

Modul 2

Im zweiten Modul sollen mittels Korrelationsverfahren Punktkorrespondenzen zwischen zwei Bildern berechnet werden. Alle vorgegebenen Aufgaben müssen in MATLAB implementiert werden. Hierzu dürfen mathematische Operationen und Routinen, jedoch keine Routinen der Computer Vision System Toolbox oder fertige Bildanalysefunktionen verwendet werden. Die Aufgaben können in Gruppen mit bis zu fünf Mitgliedern bearbeitet werden. Bitte verwenden Sie für jede Funktion die vorgegebenen Dateien und kommentieren Sie Ihre Schritte ausführlich.

Erweitern Sie das vorgegebene MATLAB Skript `CVHA2.m` so, dass alle nötigen Schritte vom Laden der mitgelieferten Bilder über den Aufruf Ihrer Funktionen bis hin zum Anzeigen des Resultats mit einem Aufruf dieses Skriptes ausgeführt werden. Geben Sie nur *funktionierenden Code* ab und verwenden Sie nur *relative Pfade*.

Fügen Sie am Anfang des Skriptes `CVHA2.m` als Kommentar die Namen aller Gruppenmitglieder und Ihre Gruppennummer hinzu. Stellen Sie sicher, dass nur die beteiligten Personen in Ihrer Moodle-Gruppe eingetragen sind. Komprimieren Sie Ihre Abgabe in einem Archiv (z.B. zip, rar, tar.gz) und geben Sie diese Datei auf Moodle für Ihre Gruppe ab.

Abgabeschluss ist Mo, 13.06.2016, 23:55 Uhr.

Korrespondenzschätzung

- Schreiben Sie eine MATLAB-Funktion

[Korrespondenzen] = punkt_korrespondenzen(I1,I2,Mpt1,Mpt2,varargin)

der zwei Grauwert-Bilder `I1,I2`, zwei Merkmalspunktmatrizen `Mpt1,Mpt2` sowie optional die Parameter `window_length` und `min_corr` übergeben werden können. Die Funktion soll die Regionen um die mitgelieferten Merkmalspunkte miteinander abgleichen, um Korrespondenzpunktpaare für weitere Stereo-Algorithmen zu gewinnen. Die Merkmalspunkte sollen mit dem in Modul 1 entwickelten Harris-Detektor extrahiert werden¹.

Für jeden Merkmalspunkt \mathbf{p}_1 aus `Mpt1` wird der Punkt \mathbf{p}_2 aus `Mpt2` gesucht, der die *Normalized Cross Correlation* (NCC) aller Pixel innerhalb eines quadratischen Fensters der Länge `window_length` um \mathbf{p}_1 und \mathbf{p}_2 maximiert und für den gilt $\text{NCC} > \text{min_corr}$. **Eine Rotationsnormierung ist nicht erforderlich.** Wird kein Punkt \mathbf{p}_2 mit $\text{NCC} > \text{min_corr}$ gefunden, so wird \mathbf{p}_1 verworfen. Wählen Sie geeignete Default-Werte für die optionalen Parameter.

Bedenken Sie bei der Implementierung der NCC, dass sich das Skalarprodukt zweier Matrizen $\text{tr}(\mathbf{A}^\top \mathbf{B})$ effizient als vektorielles Skalarprodukt $\mathbf{A}^s \top \mathbf{B}^s$ realisieren lässt, wobei \mathbf{A}^s und \mathbf{B}^s die vektorisierten Bildsegmente sind.

¹Sie können hierfür Ihre eigene (funktionierende!) Implementierung oder die vorgegebene Version verwenden

Gefundene Korrespondenzpunktpaare $\mathbf{p}_1/\mathbf{p}_2$ sollen in der Matrix **Korrespondenzen** gespeichert werden, so dass ihre jeweiligen Koordinaten in einer gemeinsamen Spalte aufgeführt sind:

$$\begin{bmatrix} x_{\mathbf{p}_1} & \dots \\ y_{\mathbf{p}_1} & \dots \\ x_{\mathbf{p}_2} & \dots \\ y_{\mathbf{p}_2} & \dots \end{bmatrix}$$

- Messen Sie die Zeit, die Ihr Punktmatcher benötigt, und geben Sie diese sowie die Anzahl der gefundenen Korrespondenzen auf der MATLAB-Konsole aus.
- Finden Sie eine geeignete Möglichkeit, um korrespondierende Punkte in den Bildern **I1** und **I2** möglichst eindeutig zu visualisieren (z.B. Farben, Zahlen, Pfeile etc.), und implementieren Sie diese. Über einen weiteren optionalen Parameter `do_plot` soll einstellbar sein, ob gefundene Korrespondenzpaare angezeigt werden oder nicht.