



Computer Vision II (WS 2010/2011)

Institut für Neuroinformatik, Universität Ulm
Prof. Dr. Heiko Neumann, Jan Bouecke und Fabian Groh

Blatt 3

Ausgabe: 12.11.2010 :: Abgabetermin: Do, 25.11.2009, bis 12:00 Uhr (Raum 437)).

Abgabe: Quellcode per Email an Fabian Groh sowie den Quellcode und die Grafikausgabe gedruckt und geheftet an Jan Bouecke (Raum 437).
Bearbeitung der Blätter möglichst in **Zweier**-Gruppen.

Bewegungsdetektion mit regionenbasiertem Census-Matching (10 Pkt.)

In dieser Übungsaufgabe soll ein Bewegungsdetektor implementiert werden, welcher in zwei aufeinander folgenden Bildern jeweils Regionen mit konstantem Censuswert bestimmt und einander zuordnet. Der so detektierte spärliche Bildfluss soll visualisiert und die Ergebnisse diskutiert werden.

- a) Implementieren Sie zunächst eine Censustransformation als Funktion `calc_census(I)`, die die Censuswerte eines kompletten Bildes berechnet und in Form einer 2-D Matrix im Format `uint32` zurück liefert. Verwenden Sie hierbei ein 3-wertiges Schema für die Kodierung der Census-Werte (01 für positive Werte, 00 für Werte $\pm \epsilon$ ($\epsilon = 0.005$; diese Schwelle bezieht sich auf Grauwerte im Wertebereich (0, 1)) und 10 für negative Werte; vgl. Skript CVII, Teil II, Abschnitt 6). Vor der Berechnung der Censuswerte sollen die Eingabebilder mit einem Gaußfilter der Standardabweichung $\sigma = 1$ geglättet werden.
- b) Machen Sie sich mit der Funktion `region(F)` (siehe Webseite) vertraut, welche als Eingabe eine 2-D Matrix im `uint32` Format erwartet. Diese Funktion bestimmt zusammenhängende Regionen, in denen die Homogenität durch konstante Census-Werte in der Eingabematrix definiert ist. Das Ergebnis dieser Segmentierung wird, zusammen mit den Schwerpunkten und Größen der Segmente, zurückgeliefert (siehe `help region`). Sie erhalten somit nach der Vorverarbeitung für jedes Einzelbild eine symbolhafte Repräsentation der Bildstrukturen (Nr. der Region, Schwerpunkt (x, y), Census-Wert) (vgl. Skript CVII, Teil II, Abschnitt 6).
Verwenden Sie diese Merkmale ihrer Ergebnisrepräsentation, um Regionen im Bild zu unterdrücken, die eine bestimmte Mindestgröße nicht überschreiten (z.B. 3 Pixel). Testen Sie dies für verschiedene Schwellen auf den Bildern `yos_clouds_01.png`, `hamburg_taxi_01.png` und `exampleA_1.png`.
- c) Nachdem homogenen Regionen (homogen bezüglich der Censuswerte) bestimmt wurden, müssen Korrespondenzen zwischen den Regionen jedes Bildpaars hergestellt werden. Gehen Sie hier davon aus, dass eine gültige Korrespondenz und eine valide Bewegung dann vorliegt, wenn
 - der Censuswert zwischen den Bildern erhalten bleibt,
 - sich die Fläche einer Region um weniger als 75% von einem Bild zum nächsten hin ändert,
 - nur Regionen betrachtet werden, deren Fläche mindestens 3 Pixel beträgt, und
 - Verschiebungen von mehr als 10 Pixeln pro Frame ausgeschlossen sind.

Die Korrespondenzfindung soll durch den paarweisen Vergleich von Regionen in Bild 1 mit den Regionen in Bild 2 erfolgen, ohne jede mögliche Kombination testen zu müssen. Eine Methode besteht darin, die nach dem Selektionsschritt verbliebenen Regionen (Fläche > 3 Pixel) nach ihrem Censuswert zu sortieren. So kann in zwei sortierten Listen (jeweils eine für jedes Bild) nach passenden Korrespondenzen gesucht werden (siehe nachfolgender Algorithmus in Pseudocode-Notation):

- (1) Nimm die erste Region in Liste von Bild 1
- (2) Vergleiche Region mit allen Regionen mit gleichem Censuswert aus Liste B.

- ```
/* diese liegen hier ja alle hintereinander sortiert vor */
```
- (3) Falls mindestens eine passende Region, welche obigen Kriterien entspricht, dann  
    wähle die aus, welche die ähnlichste Flächengröße hat,  
    sonst  
        nimm eine Null-Bewegung an.
  - (4) Fahre mit nächster Region aus der Liste A fort (2), bis die Liste abgearbeitet ist.
- d) Nachdem im Aufgabenteil c) die korrespondierenden Regionen gefunden wurden, soll jedem Pixel die Verschiebung der Schwerpunkte korrespondierender Regionen zugeordnet werden. Falls es zu einem Pixel oder zu einer Region keine korrespondierende Region im zweiten Bild gibt, so soll eine Null-Bewegung (Geschwindigkeit = 0) angenommen werden. Visualisieren Sie das Ergebnis der Zuordnungen in Form eines Vektorfeldes. Verwenden Sie dazu die Funktion `quiver()`.
- e) Verwenden Sie zur lokalen Bildbeschreibung alternativ die Ergebnisse der Rangtransformation (diese können Sie durch Summation der Bits einfach bestimmen) und speichern Sie die Ergebnisse in einer `uint32` Matrix. Verwenden Sie ansonsten den gleichen Algorithmus zur Detektion des Bildflusses.
- f) Diskutieren Sie kurz die unterschiedlichen Ergebnisse der verschiedenen Methoden zur Detektion von Strukturen in Bildern (Census vs Rang) für alle 3 Sequenzen, `yos_clouds`, `hamburg_taxi`, `exampleA`.

Nützliche Befehle:

```
fspecial('gaussian'), wshift, bitset, zeros(h,w,'uint32').
```

Viel Spaß!