



Computer Vision II (WS 2010/2011)

Institut für Neuroinformatik, Universität Ulm

Prof. Dr. Heiko Neumann, Jan Bouecke und Fabian Groh

Blatt 2

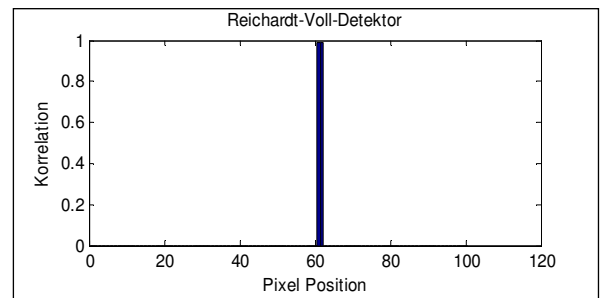
Ausgabe: 29.10.2010 :: Abgabetermin: Do, 11.11.2009, bis 12:00 Uhr (Raum 437)).

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.

1. Reichardt-Halb- und Volldetektor (4 Pkt.)

Implementieren Sie einen Reichardt-Detektor, welcher zwei Bildpaare auf eine Verschiebung von $\Delta x=5$ Pixeln (nach rechts) hin untersucht. Verwenden Sie dazu die Bilder *image1.png* und *image2.png*, welche zwei aufeinander folgende Zeitschritte darstellen.

- Implementieren Sie einen einfachen Halbdetektor, welcher die Helligkeit des Pixels x im Bild 1 mit der Helligkeit des Pixels $x+5$ im zweiten Bild vergleicht (siehe Skript, Teil III, Folie 31).
- Wenden Sie den so implementierten Halb-detektor an jedem Pixel $x=(1:100)$ des Bildes an und stellen Sie das Antwortverhalten in Abhängigkeit von x in einem Graphen dar.
- Erweitern Sie ihren Halb-Detektor zu einem Voll-Detektor, der die Bewegung nach rechts und nach links betrachtet. Der Voll-Detektor soll dabei für die Bewegung nach rechts korrekt sein. Wenden Sie diesen erneut auf das Bildpaar an und visualisieren Sie die das Antwortverhalten.
- Erklären Sie kurz das unterschiedliche Antwortverhalten.



Nützliche Befehle: `imread`, `wshift`

2. Reichardt-Halb-Detektor zur Bewegungsschätzung (6 Pkt.)

In dieser Aufgabe wird Bewegung aus Bildsequenzen mit Hilfe des Reichardt- Halbdetektors bestimmt (siehe 2b). Als Eingabe (R_1 und R_2 , siehe Skript, Teil III) werden hier **Laplace-gefilterte** Versionen der Einzelbilder einer Sequenz verwendet.

Die Ergebnisse sollen, wie unten beschrieben, für folgende Bildpaare ausgewertet werden ($k \in \{1,2\}$):

- `sprintf('yos_clouds_%02d.tif',k);`
- `sprintf('motionC%d.tif',k);`
- `sprintf('exampleA_%d.tif',k);`

Schritte der Bearbeitung:

- Berechnen Sie die Laplace-gefilterten Eingabebilder R_1 und R_2 ($R_i = \text{Bild}_i * \text{Laplace}$). Dabei kann der Laplacefilter (Laplace) durch die Funktion `fspecial` generiert werden.
- Für die Berechnung der Korrelation (Ähnlichkeit) zwischen zwei Bildern, wird der im Skript beschriebene Halbdetektor um eine Glättung erweitert, um stabilere Ergebnisse zu erhalten (vgl. Abb. rechts):
Das Ergebnis der Multiplikation M der beiden Eingaben wird, wie im Folgenden beschrieben, räumlich integriert (geglättet) und bildet erst danach die Ausgabe A des so erweiterten Halbdetektors.
Dabei bezeichnet F_1 das Eingabebild zum Zeitpunkt t ($F_1(y,x)=R_1(y,x)$) und F_2 das um $(\Delta x, \Delta y)$ verschobene Eingabebild R_2 zum Zeitpunkt $(t+\Delta t)$, $F_2(y,x) = \text{wshift}('2D', R_2(y,x), (\Delta y, \Delta x))$.

Die räumliche Integration des Produktes des einfachen Halbdetektors kann in Matlab bequem durch Glättung mit einem gaussförmige Faltungskern W umgesetzt werden. Dessen Standardabweichung σ definiert dabei die zur Integration verwendete Nachbarschaft.

In dieser Teilaufgabe soll nun ein solcher Reichardt(halb)detektor für $\Delta x, \Delta y \in \{-6, -5, \dots, 5, 6\}$ und einem Gaussfilter mit der Größe 9×9 und $\sigma=2$ implementiert werden.

Hinweis:

Es bietet sich an, die Korrelations-Ergebnisse in Matlab in einem 4D Array zu speichern, wobei die 1. und 2. Dimension den Ort (x,y) angeben und die 3. und 4. Dimension den Geschwindigkeitsvektor ($\Delta x, \Delta y$) (Verschiebungsvektor von $(-6,-6)$ bis $(+6,+6)$).

```
...
Correlation = zeros(frameheight,framewidth,13,13);
for dx = -6:6      % dx entspricht  $\Delta x$  in pixel/frame
    for dy = -6:6  % dy entspricht  $\Delta y$  in pixel/frame
        ...
        Correlation(:, :, dy+7, dx+7) = ...;
    end
end
...
```

- c) Visualisieren Sie die Antworten eines vollständigen Satzes von Reichardt-Halbdetektoren in Form einer Ähnlichkeitslandschaft (ims-

how(squeeze(Correlation(y,x,:,:),[])) an folgenden Positionen und beschreiben Sie jeweils in einem kurzen Satz das Gesehene:

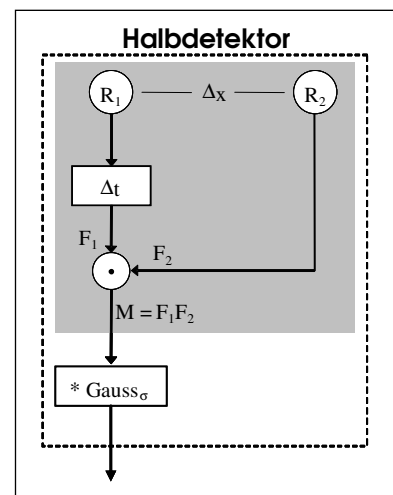
yos_clouds: x=100, y=100
 exampleA: x=20, y=20 und x=20, y=40
 image: x=50, y=50

- d) Für die Visualisierung eines Feldes von Bewegungsvektoren, wird an jedem Ort im Bild der Eintrag mit der maximalen Korrelation gesucht und der entsprechend zu deren Berechnung verwendete Verschiebungsvektor als Vektorpfeil im Bild angezeigt.

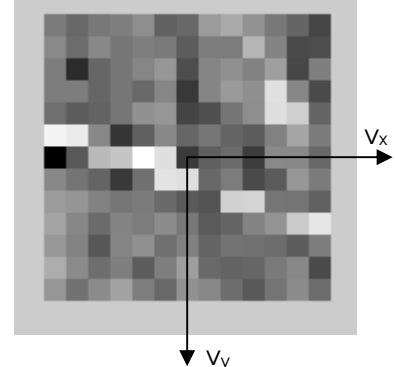
Visualisieren Sie nun die Ergebnisse des von Ihnen in b) implementierten Reichardt(halb)-detektors mithilfe der zur Verfügung gestellten Matlab-funktion `plot_flow_corr_vec(C, Dx, Dy, rate)`. Diese Funktion erwartet das 4-dimensionale Array mit den Korrelationsantworten (aus Aufgabenteil b)), sowie zwei Vektoren, welche die verwendeten Verschiebungen beinhalten $(-6,6)$ und einen Parameter `rate`, der die Anzahl der gezeichneten Pfeile reduziert. Für den Parameter `rate` sollten Sie für Bildpaar `yos_cloud` den Wert 8, für `image` 3, für `example` 5 und für `motion` 8 verwenden.

Nützliche Befehle:

`fspecial('laplacian'), sprintf, imread, wshift, squeeze, imfilter.`



Korrelationsantwort an einer Position für 2D-Verschiebungsvektoren



Viel Spaß!