



## Computer Vision II (WS 2010/2011)

Institut für Neuroinformatik, Universität Ulm

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

### Blatt 7

Ausgabe: 21.1.2011; Abgabetermin: Do, 3.2.2011, bis 18:00 Uhr (Raum 426).

**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 oder Dreier-Gruppen**.

### Tracking: Das Lucas-Kanade Verfahren (10 Pkt.)

In dieser Aufgabe soll zunächst eine einfache *Warping*-Funktion für eine bildbasierte Verfolgungsaufgabe untersucht werden. In einem weiteren Schritt soll mittels dieser (einfachen) *Warping*-Funktion das Lucas-Kanade Verfahren für das *Matching* eines Bildausschnittes in zeitlich aufeinander folgenden Bildern verwendet werden.

- a) **Warping:** Vorbereitend untersuchen Sie folgende *Warping*-Funktion für Bilder und implementieren diese:

$$\mathbf{w}(\mathbf{x}; \mathbf{p}) = \begin{pmatrix} s & 0 & t_x \\ 0 & s & t_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Welche Dimension hat der Parametervektor  $\mathbf{p}$  und welche Bedeutungen haben die einzelnen Parameter?

Testen Sie ihre Funktion mit dem Eingabebild `box-00.pgm` für

- eine Skalierung um den Faktor 2, wodurch sich die Fläche der Box vervierfacht; dabei soll eine möglichst einfache Parametrisierung verwendet werden, bei der auch eine zusätzliche Translation erlaubt ist.
- eine Skalierung um den Faktor 2, wobei das Zentrum der Box an der gleichen Position verbleiben soll sowie
- eine Verschiebung um (20,-30) in Bildpixeln, bei unveränderter Größe.

Wie sind für jede dieser Warpings jeweils Werte des Parametervektors  $\mathbf{p}$  zu wählen?

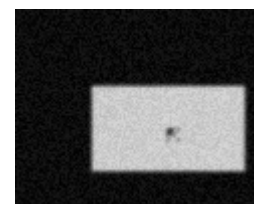
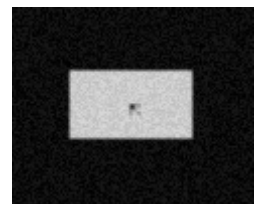
**Hilfreiche Funktionen:** `ndgrid`, `interp2`. **Beachte:** Wie werden Bereiche im Bild aufgefüllt, die nicht interpoliert werden können?

- b) **Tracking mittels Lucas-Kanade Ansatz:** Implementieren Sie den in der Vorlesung vorgestellten Algorithmus (Skript, Kap. VII, S. 15ff) und *warp*en damit den Bildinhalt von `box-00.pgm` mittels eines iterativen Gradientenabstiegs für den Parametervektor  $\mathbf{p}$  auf den entsprechenden Bildausschnitt in `box-01.pgm`.

Zunächst müssen Sie dazu die Jakobi-Matrix<sup>1</sup>  $\mathbf{J}$  zur gegebenen Transformation  $w$  bestimmen. Diese ändert sich nicht in jeder Iteration und muss daher nur einmal bestimmt werden. Dagegen müssen die Ortsableitungen im gewarpten Bild nach jeder Aktualisierung von  $\mathbf{p}$  neu bestimmt werden. Mit dem ortsabhängigen Spaltenvektor

$\mathbf{N} = \mathbf{J}^T \cdot \nabla I_w$ , dem ersten Bild  $I_1$  und dem gewarpten Bild  $I_w$  lässt

sich die Iterationsvorschrift dann wie folgt formulieren:



---

<sup>1</sup>  $\mathbf{J} = \frac{\partial w(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}}$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \left( \sum_{x,y} \mathbf{N} \cdot \mathbf{N}^T \right)^{-1} \cdot \sum_{x,y} \mathbf{N} \cdot (I_2 - I_w)$$

Zeichnen Sie dabei über die Iterationen folgende Werte auf:

- Die euklidische Norm des Parametervektors  $\|\mathbf{p}\|$ .
- Den quadratischen Fehler

$$E = \sum_{x,y} (\text{imwarp}(I_1) - I_2)^2.$$

Der Algorithmus soll nach max. 40 Iterationen stoppen oder früher, falls  $\|\Delta\mathbf{p}\| < 10^{-5}$ .

Zeigen Sie die Differenz zwischen dem gewarppten ersten und dem zweiten Bild. Zusätzlich sollen in jeweils einem Graphen die Werte für  $\|\mathbf{p}\|$  und  $E$  über die Iterationen angezeigt werden. Gestartet werden soll das iterative Verfahren mit einem Parametervektor  $\mathbf{p}$ , welcher die *Warping*-Funktion zur Identität macht.

Hinweise: Eine Umsetzung ist am einfachsten (nicht am schnellsten), wenn über jeden Punkt des Bildes iteriert wird um den dortigen Vektor  $\mathbf{N}$  zu berechnen.

- c) **Tracking ohne Skalierung:** Wenden Sie den von Ihnen in b) entwickelten Algorithmus auf das Bildpaar `box-00.pgm` und `box-02.pgm` an. Erläutern Sie das Resultat und den dabei ermittelten Parametervektor  $\mathbf{p}$  kurz in eigenen Worten.

### Tracking: Bonusaufgabe

Wenden Sie Ihren Algorithmus auf das Bildpaar `img_035.pgm` und `img_036.pgm` an um das Vorfahrt achten Schild zu verfolgen. Für die Auswertung der Ähnlichkeiten bzw. des Ähnlichkeitsgradienten soll dabei nur die lokale Nachbarschaft um das Schild herum in die Berechnung einfließen. Die Beiträge der übrigen Pixel des Bildes sollen mithilfe einer Maske ausgeblendet werden (siehe `img_035_mask.pgm`). Somit muss in jedem Iterationsschritt zusätzlich zum *Warping* des Bildes die zugehörige Maske auf die gleiche Weise transformiert werden.



Viel Spaß!