

Geometrische Transformationen

- Definition und Anwendungen
- Basisalgorithmus
- Populäre Transformationen
- Bestimmung von Transformationsparametern
- Piecewise Warping
- Bildregistrierung
- Morphing

Was ist geometrische Bildtransformation (1)

Bildverbesserungsoperatoren: Transformation der Grauwerte

Was ist geometrische Bildtransformation (1)

Bildverbesserungsoperatoren: Transformation der Grauwerte

Geometrische Transformationen (Warping): Transformation der Positionen

Ziel: geometrische Abbildung von f nach g

$$g(r, c) = f(r^* = u(r, c), c^* = v(r, c))$$

Folgende Transformationen sind besonders einfach zu realisieren:

- Pixelweise Translation
- Rotation von 90° , 180° und 270°
- horizontale/vertikale/diagonale Spiegelung

Was ist geometrische Bildtransformation (2)

Beispiel:

● Translation:

$$\begin{cases} r^* &= r + d_y \\ c^* &= c + d_x \end{cases}$$

● Skalierung:

$$\begin{cases} r^* &= s_y \cdot r \\ c^* &= s_x \cdot c \end{cases}$$

● Rotation:

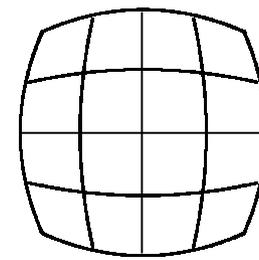
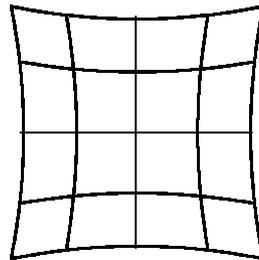
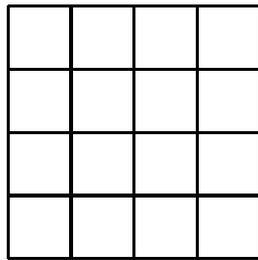
$$\begin{cases} r^* &= \cos \alpha \cdot r + \sin \alpha \cdot c \\ c^* &= -\sin \alpha \cdot r + \cos \alpha \cdot c \end{cases}$$

● Scherung:

$$\begin{cases} r^* &= r + a \cdot c \\ c^* &= c \end{cases} \quad \text{oder} \quad \begin{cases} r^* &= r \\ c^* &= b \cdot r + c \end{cases}$$

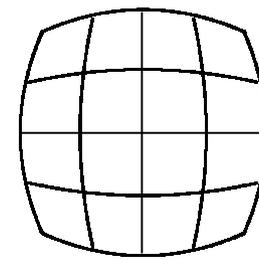
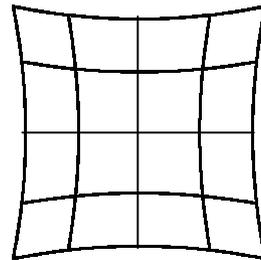
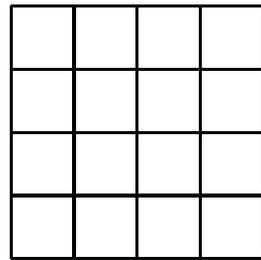
Anwendungen (1)

- Geometrische Entzerrung
Positive (Mitte) und negative (rechts) Verzeichnungen (an Bildrändern
Störungen am stärksten)



Anwendungen (1)

- Geometrische Entzerrung
Positive (Mitte) und negative (rechts) Verzeichnungen (an Bildrändern Störungen am stärksten)



- Korrektur von Skew und Slant in Dokumentbildern

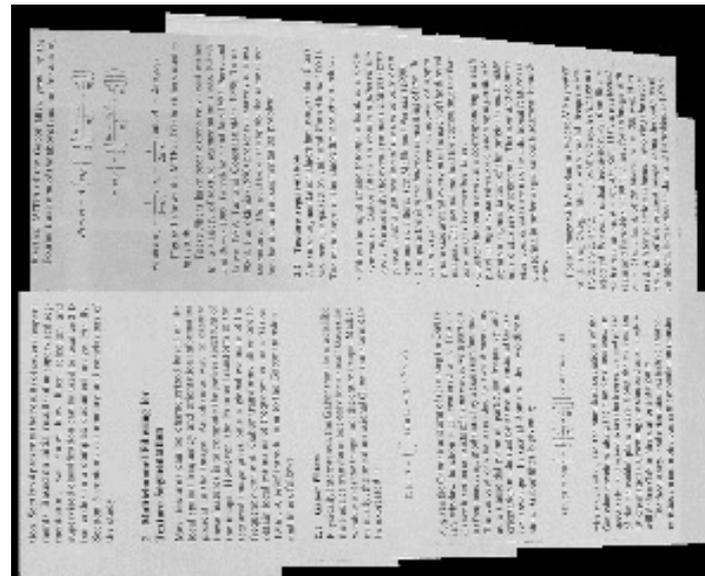
The film version of Miss Shelagh Delaney's

Nach Slant-Korrektur (Bestimmung des Slantwinkels und Durchführung einer Scherung)

The film version of Miss Shelagh Delaney's

Anwendungen (2)

- Mosaicing: Zusammenfügen von Teilbildern zum Gesamtbild



Anwendungen (3)

- Registrierung von Bildern:
 - zu unterschiedlichen Zeitpunkten aufgenommen
 - verschiedene Bildmodalitäten

Bildtransformation nötig, damit die Bilder pixelweise miteinander vergleichbar. Z.B. Vergleich medizinischer Befunde.

- Spezialeffekte



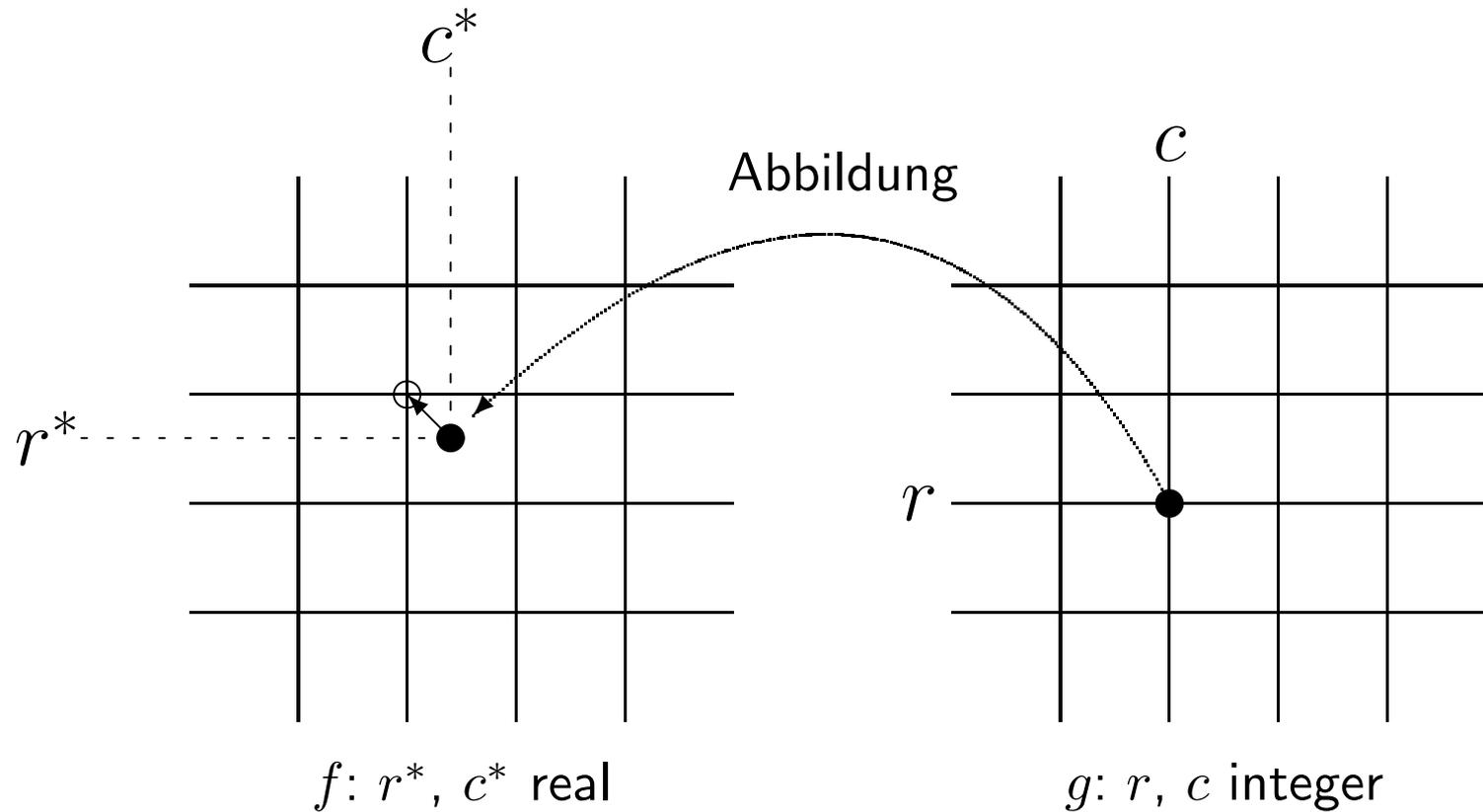
Algorithmus

Folgender Algorithmus anwendbar für beliebige Bildtransformation

```
for ( $r = 1; r \leq M; r++$ )  
  for ( $c = 1; c \leq N; c++$ ) {  
    berechne  $r^*$  und  $c^*$ ;  
    if ( $(r^* < 1)$  or ( $r^* > M$ ) or ( $c^* < 1$ ) or ( $c^* > N$ ))  
       $g(r, c) = 0$ ;  
    else  
      /* i.a. Interpolation nötig */  
       $g(r, c) = f(r^*, c^*)$ ;  
  }
```

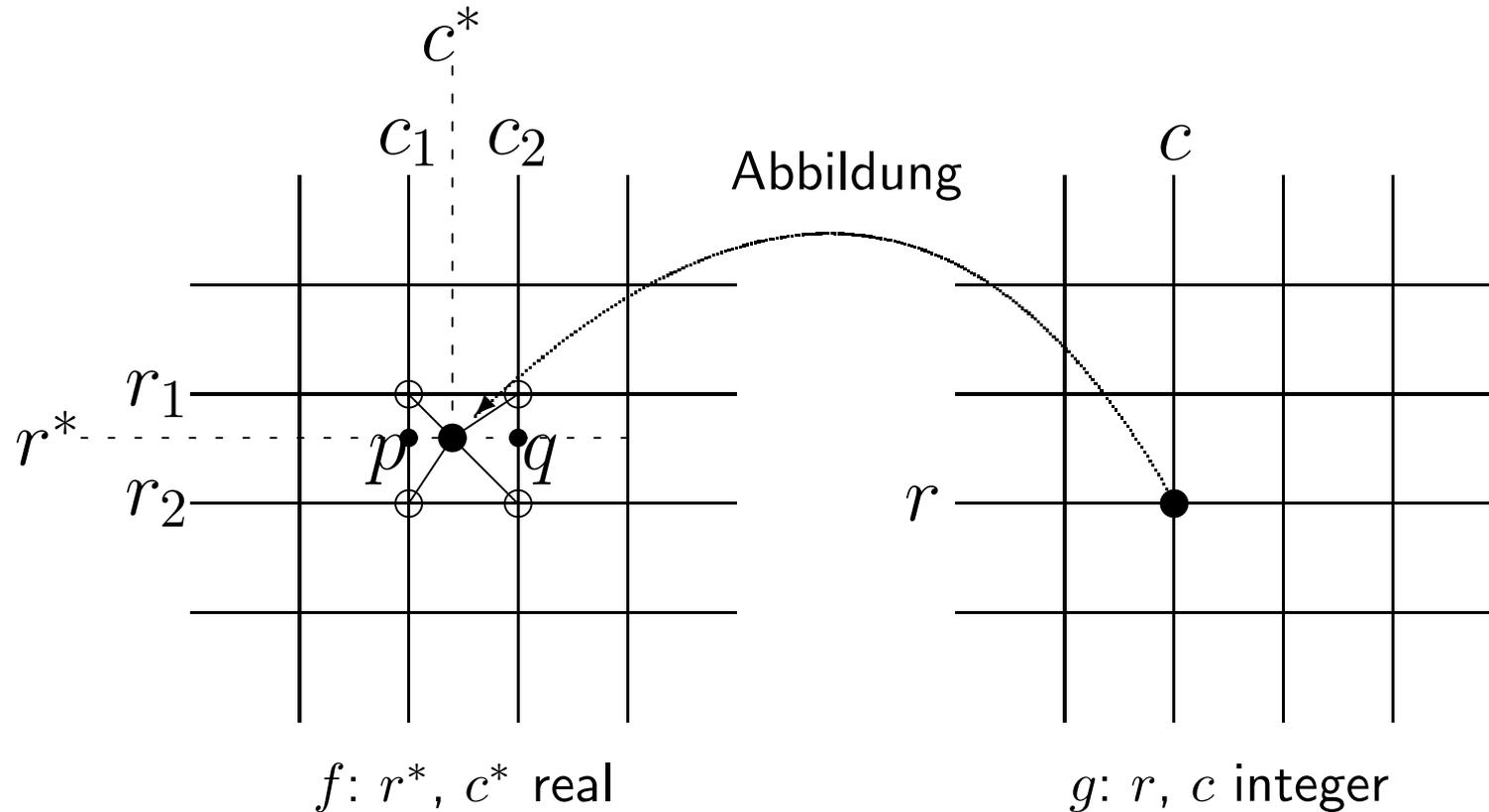
Interpolation (1)

- Nearest-neighbor Interpolation (zero-order)



Interpolation (2)

● Bilineare Interpolation (first-order)



$$\begin{aligned}f(p) &= f(r_1, c_1) + (r^* - r_1) \cdot (f(r_2, c_1) - f(r_1, c_1)) \\f(q) &= f(r_1, c_2) + (r^* - r_1) \cdot (f(r_2, c_2) - f(r_1, c_2)) \\g(r, c) &= f(r^*, c^*) = f(p) + (c^* - c_1) \cdot (f(q) - f(p))\end{aligned}$$

Interpolation (3)

Beispiel: 450% Vergrößerung

- Nearest-neighbor Interpolation: blockhaft



- Bilineare Interpolation:



Interpolationsfunktion höherer Ordnung möglich (Berücksichtigung größerer Nachbarschaft). In vielen Situationen liefert bilineare Interpolation bereits ausreichende Qualität.

Interpolation (4)

Beispiel: Vergleich Nearest-neighbor / bilinear / bikubisch

- Fall 1: Rotation um 10 Grad



Original



Nearest-neighbor



bilinear

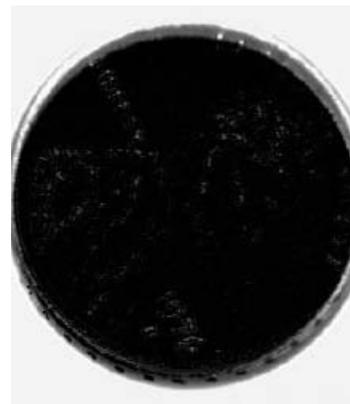


bikubisch

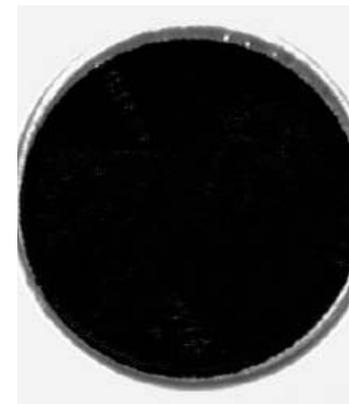
- Fall 2: 36 mal Rotation um 10 Grad; Differenz mit dem Original



Nearest-neighbor



bilinear



bikubisch

Populäre Transformationen (1)

- Affine Transformation

$$r^* = a_1 r + a_2 c + a_3$$

$$c^* = b_1 r + b_2 c + b_3$$

Translation, Skalierung, Rotation und Scherung sind Spezialfälle!
(Quadrat \implies beliebiges Parallelogramm)

- Bilineare Transformation

$$r^* = a_1 r c + a_2 r + a_3 c + a_4$$

$$c^* = b_1 r c + b_2 r + b_3 c + b_4$$

Populäre Transformationen (2)

● Polynomiale Transformation

$$r^* = \sum_{i=0}^m \sum_{j=0}^{m-i} a_{ij} r^i c^j, \quad c^* = \sum_{i=0}^m \sum_{j=0}^{m-i} b_{ij} r^i c^j$$

● Fall $m = 1$: affine Transformation

● Fall $m = 2$:

$$r^* = a_1 r^2 + a_2 c^2 + a_3 r c + a_4 r + a_5 c + a_6$$

$$c^* = b_1 r^2 + b_2 c^2 + b_3 r c + b_4 r + b_5 c + b_6$$

● Perspektivische Abbildung

$$r^* = \frac{a_1 r + a_2 c + a_3}{a_7 r + a_8 c + 1}, \quad c^* = \frac{a_4 r + a_5 c + a_6}{a_7 r + a_8 c + 1}$$

Quadrat \implies beliebiges Viereck

Bestimmung von Transformationsparametern (1)

Wahl von k korrespondierenden Punktpaaren (meistens manuell):

$$g : p_i(r_i, c_i) \longleftrightarrow f : q_i(r_i^*, c_i^*), \quad 1 \leq i \leq k$$

Bestimmung optimaler Transformationsparameter durch Minimierung der Fehler:

$$\min_u \sum_{i=1}^k [u(r_i, c_i) - r_i^*]^2$$
$$\min_v \sum_{i=1}^k [v(r_i, c_i) - c_i^*]^2$$

Bestimmung von Transformationsparametern (1)

Wahl von k korrespondierenden Punktpaaren (meistens manuell):

$$g : p_i(r_i, c_i) \longleftrightarrow f : q_i(r_i^*, c_i^*), \quad 1 \leq i \leq k$$

Bestimmung optimaler Transformationsparameter durch Minimierung der Fehler:

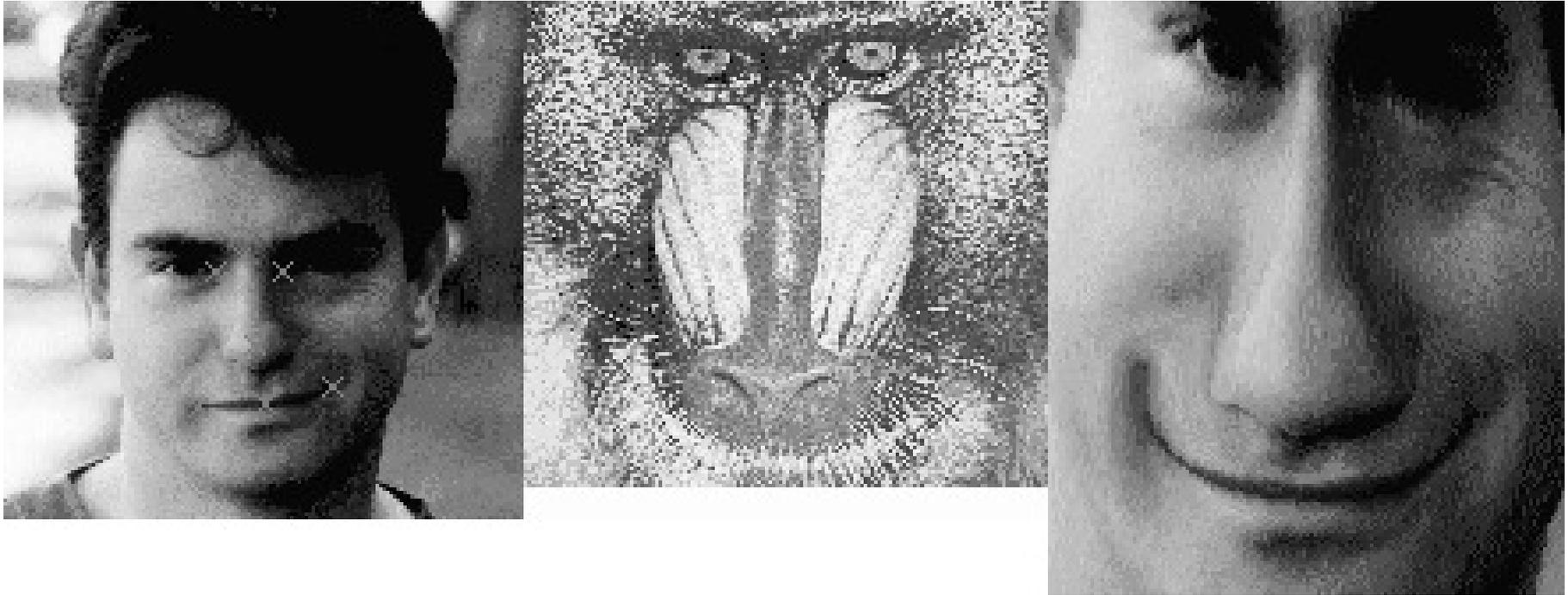
$$\min_u \sum_{i=1}^k [u(r_i, c_i) - r_i^*]^2$$
$$\min_v \sum_{i=1}^k [v(r_i, c_i) - c_i^*]^2$$

Beispiel:



Bestimmung von Transformationsparametern (2)

Beispiel:



Bestimmung von Transformationsparametern (3)

Illustration: affine Transformation

Gegeben k korrespondierende Punktpaare

$$g : p_i(r_i, c_i) \longleftrightarrow f : q_i(r_i^*, c_i^*), \quad 1 \leq i \leq k$$

Die Parameter ergeben sich aus Lösen der beiden Gleichungssysteme:

$$\begin{cases} a_1 r_1 + a_2 c_1 + a_3 = r_1^* \\ a_1 r_2 + a_2 c_2 + a_3 = r_2^* \\ \dots \dots \dots \dots \dots = \dots \\ a_1 r_k + a_2 c_k + a_3 = r_k^* \end{cases} \Rightarrow \underbrace{\begin{bmatrix} r_1 & c_1 & 1 \\ r_2 & c_2 & 1 \\ \dots & \dots & \dots \\ r_k & c_k & 1 \end{bmatrix}}_D \cdot \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} r_1^* \\ r_2^* \\ \dots \\ r_k^* \end{bmatrix}}_R$$
$$\begin{cases} b_1 r_1 + b_2 c_1 + b_3 = c_1^* \\ b_1 r_2 + b_2 c_2 + b_3 = c_2^* \\ \dots \dots \dots \dots \dots = \dots \\ b_1 r_k + b_2 c_k + b_3 = c_k^* \end{cases} \Rightarrow \underbrace{\begin{bmatrix} r_1 & c_1 & 1 \\ r_2 & c_2 & 1 \\ \dots & \dots & \dots \\ r_k & c_k & 1 \end{bmatrix}}_D \cdot \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_B = \underbrace{\begin{bmatrix} c_1^* \\ c_2^* \\ \dots \\ c_k^* \end{bmatrix}}_C$$

Bestimmung von Transformationsparametern (4)

Illustration: affine Transformation (Fort.)

Mittels der Methode der kleinsten Quadrate (least-square method):

$$A = (D^t D)^{-1} D^t R$$

$$B = (D^t D)^{-1} D^t C$$

Hierbei wird der Fehler

$$\epsilon_r = \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*)^2$$

bzw.

$$\epsilon_c = \sum_{i=1}^k (b_1 r_i + b_2 c_i + b_3 - c_i^*)^2$$

minimiert

Bestimmung von Transformationsparametern (5)

Herleitung:

● Schritt 1:
$$\frac{\partial \epsilon_r}{\partial a_1} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot r_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_2} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot c_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_3} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) = 0$$

Bestimmung von Transformationsparametern (5)

Herleitung:

● Schritt 1:
$$\frac{\partial \epsilon_r}{\partial a_1} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot r_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_2} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot c_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_3} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) = 0$$

● Schritt 2:

$$\underbrace{\begin{bmatrix} \sum_{i=1}^k r_i^2 & \sum_{i=1}^k r_i c_i & \sum_{i=1}^k r_i \\ \sum_{i=1}^k r_i c_i & \sum_{i=1}^k c_i^2 & \sum_{i=1}^k c_i \\ \sum_{i=1}^k r_i & \sum_{i=1}^k c_i & k \end{bmatrix}}_{D^t D} \cdot \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} \sum_{i=1}^k r_i r_i^* \\ \sum_{i=1}^k c_i r_i^* \\ \sum_{i=1}^k r_i^* \end{bmatrix}}_{D^t R}$$

Bestimmung von Transformationsparametern (5)

Herleitung:

● Schritt 1:
$$\frac{\partial \epsilon_r}{\partial a_1} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot r_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_2} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) \cdot c_i = 0$$

$$\frac{\partial \epsilon_r}{\partial a_3} = 2 \cdot \sum_{i=1}^k (a_1 r_i + a_2 c_i + a_3 - r_i^*) = 0$$

● Schritt 2:

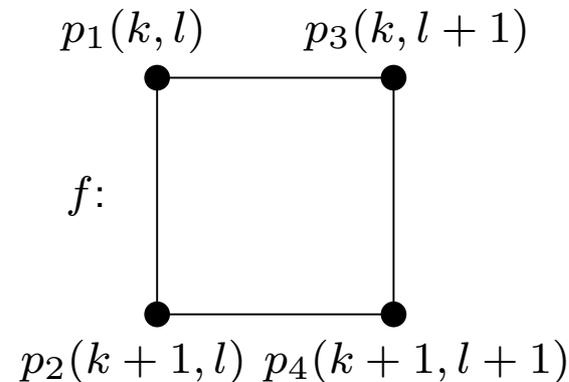
$$\underbrace{\begin{bmatrix} \sum_{i=1}^k r_i^2 & \sum_{i=1}^k r_i c_i & \sum_{i=1}^k r_i \\ \sum_{i=1}^k r_i c_i & \sum_{i=1}^k c_i^2 & \sum_{i=1}^k c_i \\ \sum_{i=1}^k r_i & \sum_{i=1}^k c_i & k \end{bmatrix}}_{D^t D} \cdot \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} \sum_{i=1}^k r_i r_i^* \\ \sum_{i=1}^k c_i r_i^* \\ \sum_{i=1}^k r_i^* \end{bmatrix}}_{D^t R}$$

● Schritt 3: $A = (D^t D)^{-1} D^t R$

Bildvergrößerung

Vergrößerung (Faktor 2; andere Werte analog):

$$\begin{cases} r^* & = & \frac{r}{2} \\ c^* & = & \frac{c}{2} \end{cases}$$



● Fall 1: $r = 2k, c = 2l$

$$g(r, c) = f(k, l) = f(p_1)$$

● Fall 2: $r = 2k + 1, c = 2l$

$$g(r, c) = f(k + 0.5, l) = \frac{1}{2} \cdot (f(p_1) + f(p_2))$$

● Fall 3: $r = 2k, c = 2l + 1$

$$g(r, c) = f(k, l + 0.5) = \frac{1}{2} \cdot (f(p_1) + f(p_3))$$

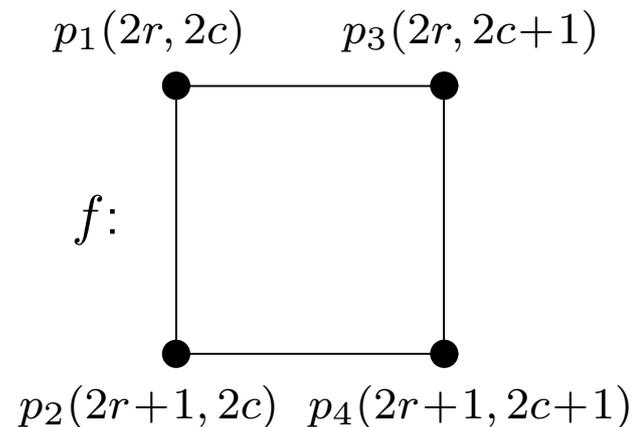
● Fall 4: $r = 2k + 1, c = 2l + 1$

$$g(r, c) = f(k + 0.5, l + 0.5) = \frac{1}{4} \cdot (f(p_1) + f(p_2) + f(p_3) + f(p_4))$$

Bildverkleinerung

Verkleinerung (Faktor 2; andere Werte analog):
$$\begin{cases} r^* & = & 2r \\ c^* & = & 2c \end{cases}$$

Ergebnisbild g aus jeder 2. Zeile / Spalte des Eingabebildes f



Informationsverlust: nur p_1 genutzt, p_2 , p_3 und p_4 ungenutzt; Lösung:

$$g(r, c) = \frac{1}{4} \cdot (f(p_1) + f(p_2) + f(p_3) + f(p_4))$$

statt

$$g(r, c) = f(2r, 2c) = f(p_1)$$

Bildverkleinerung: Pyramide

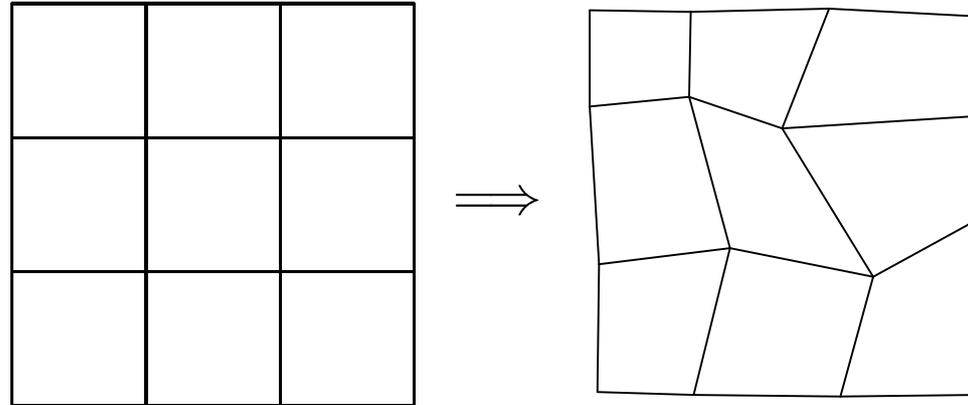
Pyramide: multiresolution representation



Viele Aufgaben in der Bildanalyse mit einer Pyramide effizienter lösbar

Piecewise Warping (1)

Annahme: Kontrollgitter mit erwünschter Deformation



Separate Bearbeitung der Bildteile (Vierecke); häufig bilineare Transformation; größere Flexibilität, z.B. einen Bildteil unverändert lassen

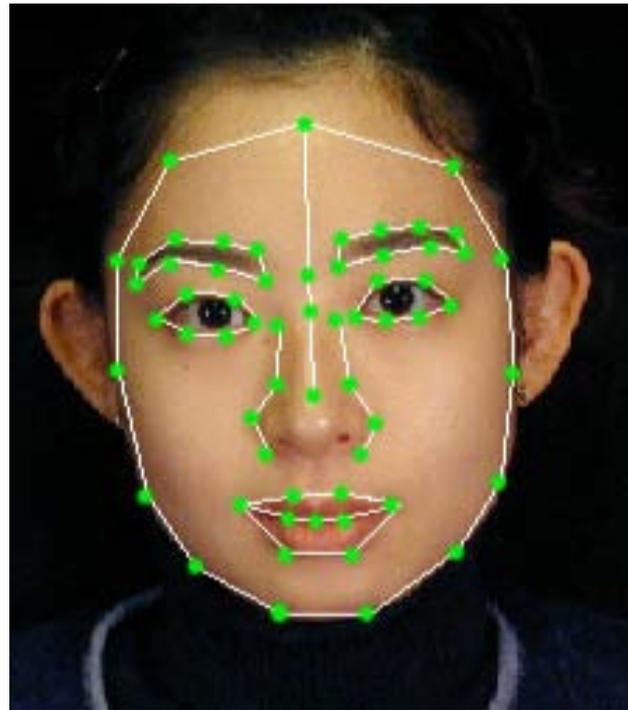


Andere Aufteilungsschemen der Bildebene, z.B. in Dreiecke, möglich

Piecewise Warping (2)

Beispiel: Simulation von Gesichtsausdrücken

Grundlage: Gesichtspaar mit (manuell) markierten Punkten \implies
Bewegungsvektoren der Punkte



Zu bearbeitendes Bild: Markierte Punkte \implies Dreiecksnetz
Bewegungsvektoren \implies neues Dreiecksnetz (durch Warping erreicht)

Piecewise Warping (3)

Beispiel: Simulation von Gesichtsausdrücken (Fort.)



Warping ist nur 2D Transformation!

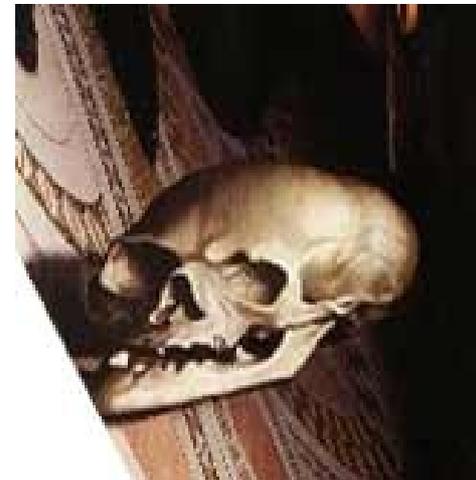
Gewisse 3D Effekte, z.B. Abbildung einer Ebene auf eine andere, möglich
Verschiedene Tiefenebenen \implies realitätsverzerrende Wirkungen



Anamorphose

Anamorphose: Bilder, die nur unter einem bestimmten Blickwinkel bzw. mittels eines speziellen Spiegels oder Prismensystems erkennbar sind

“Die Gesandten”: Gemälde von Hans Holbein dem Jüngeren, entstanden im Frühjahr 1533 in London. Originalformat 206 x 209 cm. Es hängt im Raum vier der National Gallery in London.



Bildregistrierung

Ausgangslage: Zwei Bilder desselben Inhaltes; unter unterschiedlichen Bedingungen entstanden

Ziel: das eine Bild ins andere Bild zu transformieren, so dass sie nachher *geometrisch* zueinander passen

Verschiedene Formen der Bildregistrierung:

- Different viewpoints (multiview analysis)
- Different times (multitemporal analysis)
- Different sensors (multimodal analysis)

In vielen Anwendungen treten diese Faktoren auch gleichzeitig auf

Bildregistrierung

Ausgangslage: Zwei Bilder desselben Inhaltes; unter unterschiedlichen Bedingungen entstanden

Ziel: das eine Bild ins andere Bild zu transformieren, so dass sie nachher *geometrisch* zueinander passen

Verschiedene Formen der Bildregistrierung:

- Different viewpoints (multiview analysis)
- Different times (multitemporal analysis)
- Different sensors (multimodal analysis)

In vielen Anwendungen treten diese Faktoren auch gleichzeitig auf

Unzählige Verfahren bekannt (1993–2003 > 1000 Forschungsartikel)

- merkmalsbasiertes Vorgehen
- bildbasiertes Vorgehen

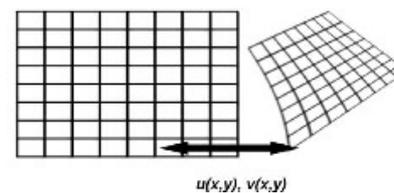
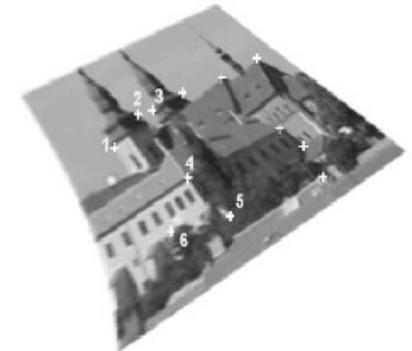
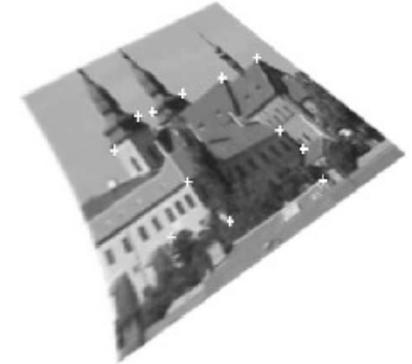
Übersicht: B. Zitova and J. Flusser, Image registration methods: A survey, Image and Vision Computing, 21: 977–1000, 2003.

Merkmalsbasierte Registrierung

Grundsätzliches Vorgehen:

- Detektion von Merkmalen: Punkte, Konturen, Regionen, etc.
- Zuordnung von Merkmalen
- Bestimmung der Transformation aus den zugeordneten Merkmalen
- Geometrische Bildtransformation inkl. Interpolation
- Nachbearbeitung (Zusammenfügen transformierter Bilder)

Beispiel: Mosaicing



Bildbasierte Registrierung (1)

Keine Merkmale explizit detektiert; explizite Annahme von Transformationsform

Der Transformationsraum wird abgesucht; die optimale Transformation bringt das eine Bild in beste Übereinstimmung mit dem anderen Bild (Übereinstimmung zweier Bilder durch Bildvergleich).

Vergleich zweier Bilder f und g :

- sum of squared differences (SSD):

$$d(f, g) = \sum_i \sum_j (f(i, j) - g(i, j))^2$$

- cross correlation:

$$d(f, g) = \sum_i \sum_j f(i, j)g(i, j)$$

Bildbasierte Registrierung (2)

Vergleich zweier Bilder f und g (Fort.):

- normalized correlation coefficient (NCC):

$$d(f, g) = \frac{\sum_i \sum_j (f(i, j) - \bar{f})(g(i, j) - \bar{g})}{\sqrt{\sum_i \sum_j (f(i, j) - \bar{f})^2} \sqrt{\sum_i \sum_j (g(i, j) - \bar{g})^2}}$$

\bar{f} : Durchschnittsgrauwert von f , \bar{g} : Durchschnittsgrauwert von g

- Mutual Information (multimodal analysis):

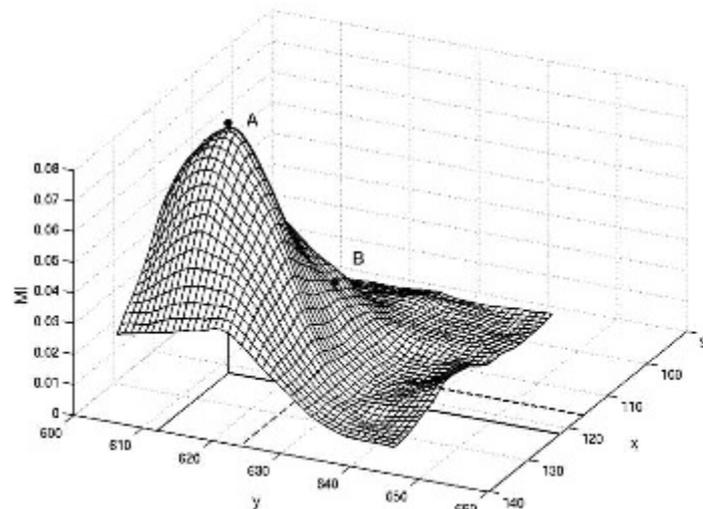
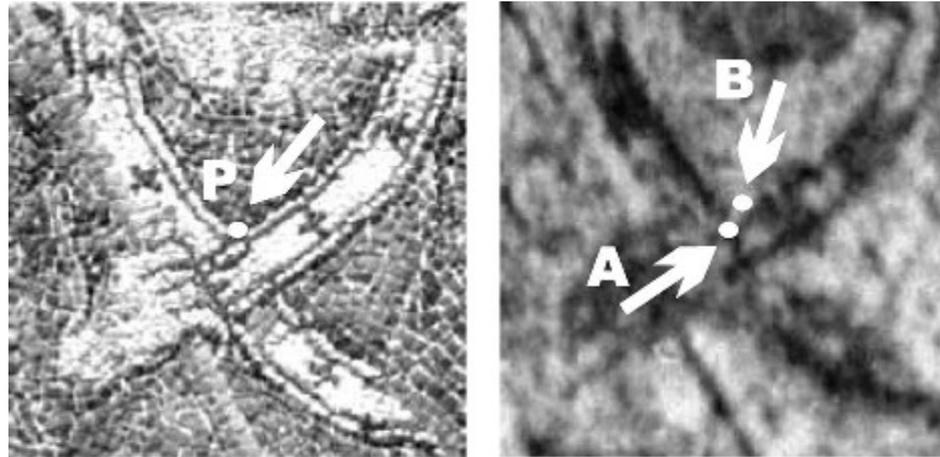
$$MI(f, g) = \sum_{x=0}^{255} \sum_{y=0}^{255} p_{fg}(x, y) \log \frac{p_{fg}(x, y)}{p_f(x)p_g(y)}$$

Typischerweise wird ein (iteratives) numerisches Verfahren eingesetzt, um die optimale Transformation zu finden

Bildbasierte Registrierung (3)

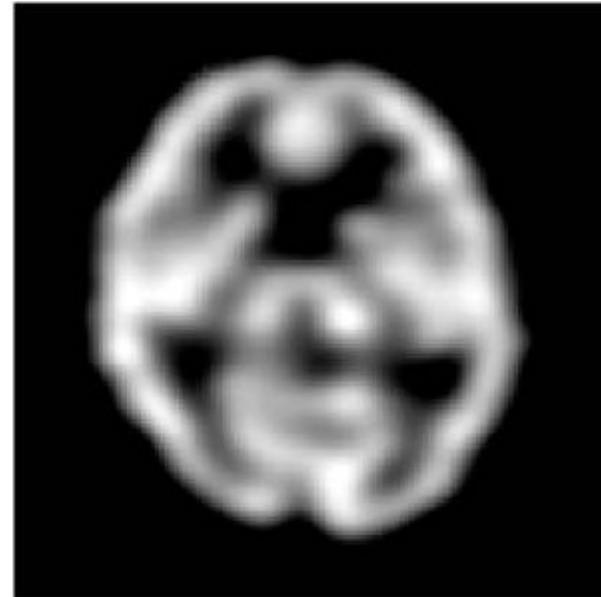
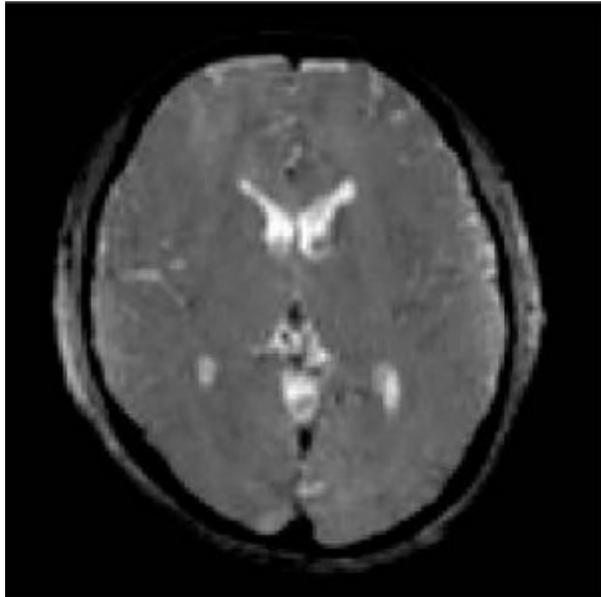
Beispiel: Mutual Information

Vergleich einer Nachbarschaft von P mit verschiedenen Teilbildern derselben Größe aus dem zweiten Bild. A: korrespondierender Punkt zu P .



Bildbasierte Registrierung (4)

Beispiel: Mutual Information (Registrierung multimodaler Bilder in der Medizin)



Bildbasierte Registrierung (5)

Elastische Registrierung:

- Keine explizite Annahme von Transformationsform
- Die Bewegung pro Bildpunkt wird geschätzt

Bildbasierte Registrierung (5)

Elastische Registrierung:

- Keine explizite Annahme von Transformationsform
- Die Bewegung pro Bildpunkt wird geschätzt

Anwendung:

Bewegungsdetektion und -korrektur in PET (Positron Emission Tomography)

- lange Aufnahmezeit \Rightarrow Bewegung (Herz, Lunge)
- Quantifizierung der Organe Herz/Lunge schwierig
- Kleinste Strukturen z.B. Plaque kaum sichtbar

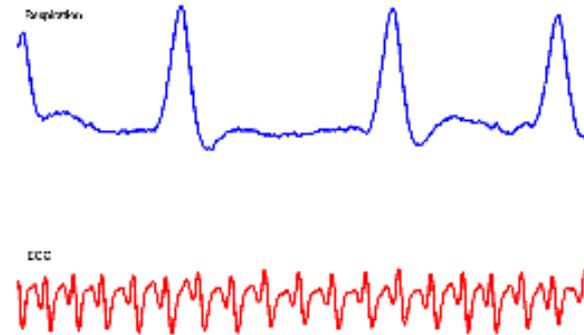


Bildbasierte Registrierung (6)

Ziel: Zwischenzustände durch Atemgating isolieren und auf Referenzzustand abbilden; Rekonstruktion mit allen transformierten Zuständen

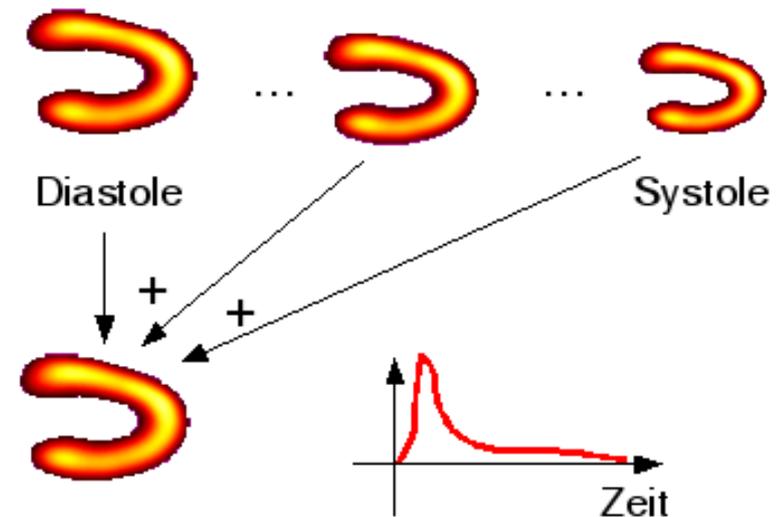
- **Bewegungsdetektion**

EKG- und Atem-Triggerung



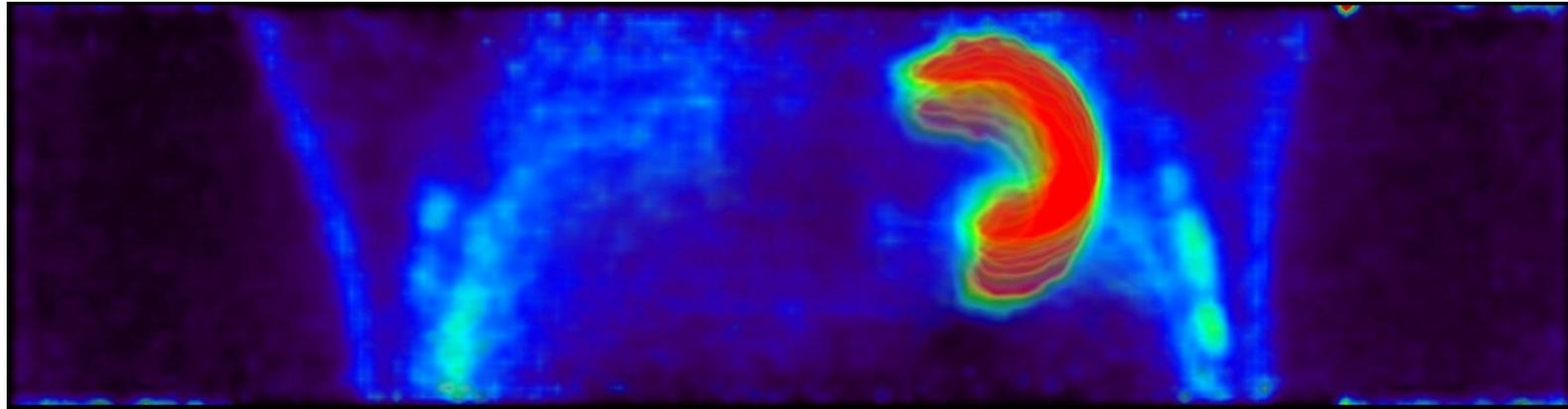
- **Bewegungskorrektur**

Mathematische 3D-Deformation

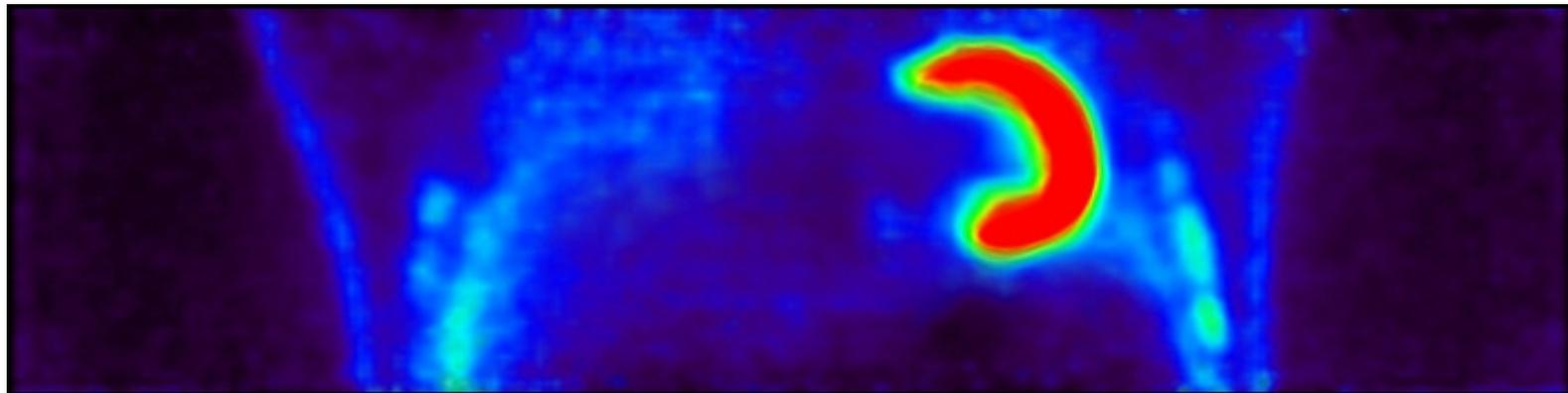


Bildbasierte Registrierung (7)

Überlagerung von acht Zuständen:



Ergebnis nach der Korrektur:



Morphing (1)

Ziel: Glatte Überführung eines Startbildes I_s in ein Zielbild I_t durch Erzeugung einer Reihe von Zwischenbildern. Glattheit sowohl in Bildgeometrie (Warping) als auch in Bildintensität (Überblenden der Grauwerte bzw. Farben). Startbild mit zunehmender Dauer stufenweise ausgeblendet / Zielbild simultan eingeblendet

Überblenden (beide Farbwerte eines jeden Bildpunktes mit verschiedenen Gewichten linear interpoliert, z.B. 1, 0.75, 0.5, 0.25, 0 für Startbild und 0, 0.25, 0.5, 0.75, 1 für Zielbild)



Morphing (2)

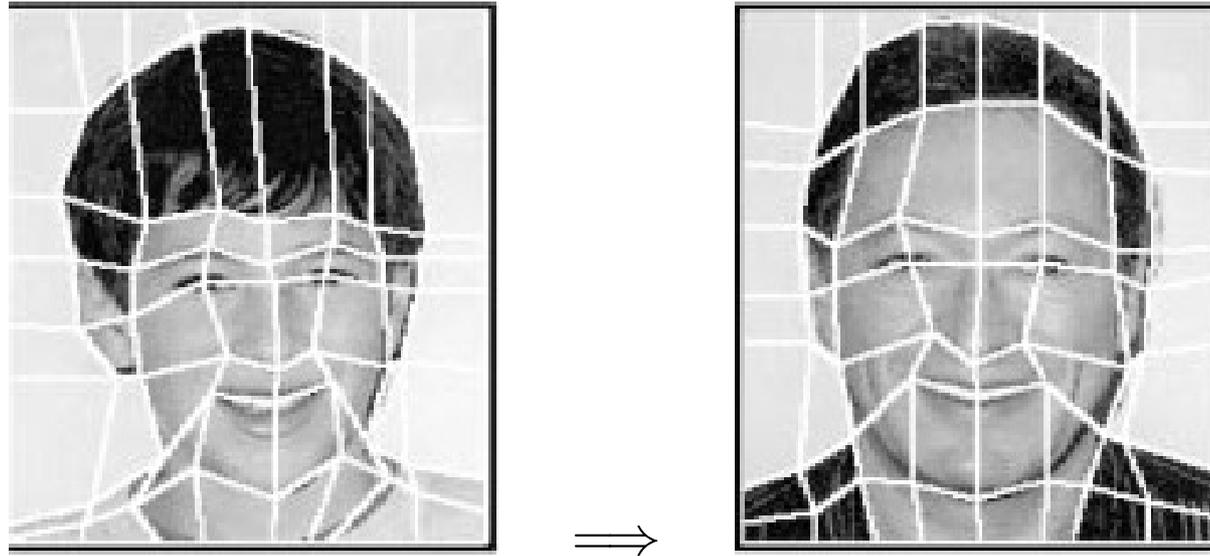
Morphing = Bildüberblendung + geometrische Bildtransformation



Verschiedene Techniken bekannt, z.B. Mesh-Warping, Field Morphing. Siehe: G. Wolberg, Image morphing: A survey, Visual Computer, 14: 360–372, 1998.

Mesh-Warping (1)

Grundlage: Gitterdarstellung des Start- und Zielbildes M_s / M_t



Algorithm Mesh-Warping

```
for (jedes Zwischenbild  $I_i$ ) do {  
    berechne ein Gitter  $M_i$  für  $I_i$  durch lineare Interpolation von  $M_s$  und  $M_t$ ;  
    transformiere  $I_s$  nach  $I_1$  unter Verwendung von  $M_s$  und  $M_i$ ;  
    transformiere  $I_t$  nach  $I_2$  unter Verwendung von  $M_t$  und  $M_i$ ;  
    berechne  $I_i$  durch lineare Interpolation von  $I_1$  und  $I_2$ ;  
}
```

Mesh-Warping (2)

Berechnung des Gitters M_i : Kontrollpunkt a in $M_s \implies$ Kontrollpunkt b in M_t

Entsprechender Kontrollpunkt in M_i : $a + t(b - a)$, $t \in [0, 1]$

Parameter t gibt indirekt die Geschwindigkeit an, wie schnell a im Laufe der Zeit nach b wandelt. Zwei Fälle:

- einheitliche Übergangsfunktion: konstante Geschwindigkeit



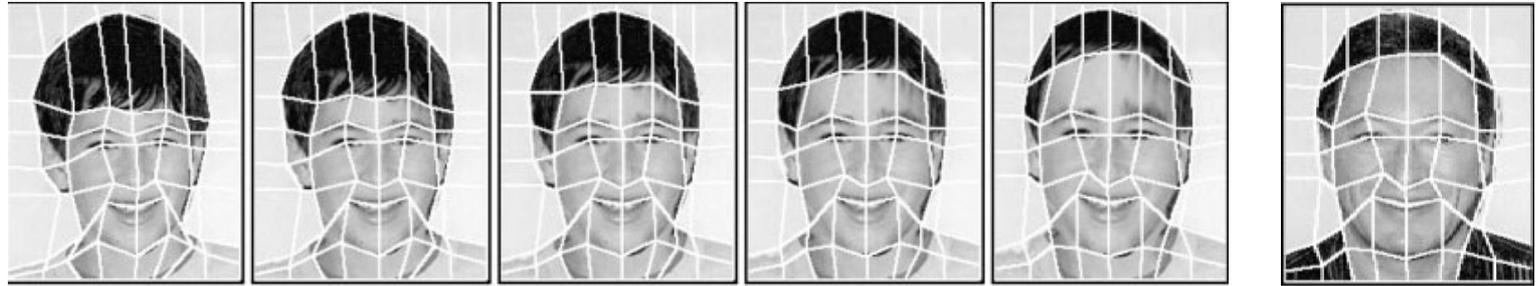
- uneinheitliche Übergangsfunktion: variable Geschwindigkeit an unterschiedlichen Kontrollpunkten



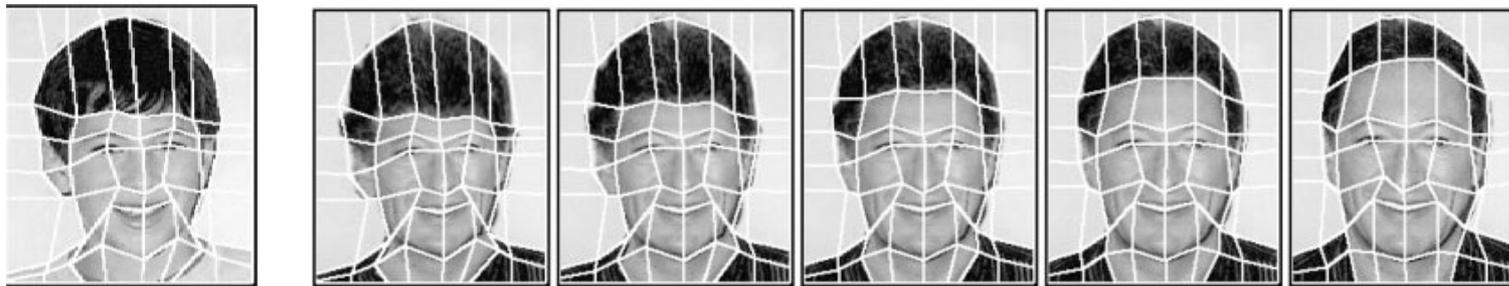
Mesh-Warping (3)

Beispiel:

Transformation des Startbildes: I_1



Transformation des Zielbildes: I_2



Morphing-Ergebnis (Überblendung: lineare Interpolation von I_1 und I_2):



Mesh-Warping (4)

Beispiel:

