

Geometrische Transformationen 2D

Transformationen starrer Körper

Körper: Menge von Punkten $\{P_i\}$ $P_i \in \mathbb{R}^2, \in \mathbb{R}^3$

Translation:

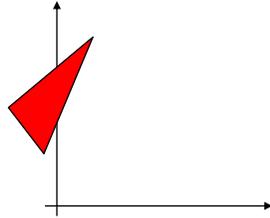
$$\forall P_i : P_i' = P_i + T$$



$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} \in \mathbb{R}^2$$

Transformationen starrer Körper (2)

Rotation:

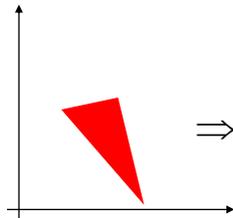


$$\forall P_i : P_i' = R \cdot P_i$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Um das Rotationszentrum

(x_r, y_r)



$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \end{bmatrix} + \begin{bmatrix} x_r \\ y_r \end{bmatrix}$$

Transformationen (3)

Skalierung:

$$\forall P_i : P_i' = S \cdot P_i$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scherung:

(kein starrer Körper)

$$\forall P_i : P_i' = S_H \cdot P_i$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & s_{hx} \\ s_{hy} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformationen (4)

Allgemein:

$\forall P_i :$

$$P_i' = P_i + T$$

$$P_i' = R \cdot P_i$$

$$P_i' = S \cdot P_i$$

$$P_i' = S_H \cdot P_i$$

Die Addition bei der Translation lässt aus hintereinander geschachtelten Transformationen, oft komplizierte Ausdrücke entstehen.

$$P_i' = P_i + T$$

$$P_i' = S \cdot (R \cdot (P_i + T))$$

Homogene Koordinaten

In Homogenen Koordinaten können alle Transformationen durch Multiplikationen ersetzt werden.

Ein algebraisches Objekt \mathbf{e} heisst homogen, falls für ein beliebiges $\lambda \neq 0$ das algebraische Objekt $\lambda \mathbf{e}$ das gleiche geometrische Element e repräsentiert.

$$e(\mathbf{e}) \equiv e(\lambda \mathbf{e}) \quad \forall \lambda \neq 0$$

2D Kartesische Koordinaten \rightarrow Homogenen Koordinaten

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ W \end{bmatrix}, \quad \text{wobei} \quad \begin{bmatrix} x \\ y \\ W \end{bmatrix} \quad \text{und} \quad k * \begin{bmatrix} x \\ y \\ W \end{bmatrix}$$

$$\text{den selben Punkt } P = \begin{bmatrix} x \\ y \end{bmatrix} \text{ beschreiben, also auch } \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Translation in Homogene Koordinaten

Alle Transformationen werden zu Multiplikationen!

$$P' = P + T(d_x, d_y) \quad \rightarrow \quad P' = T(d_x, d_y) \cdot P$$

$$\text{mit } T(d_x, d_y) = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \Leftrightarrow \quad P' = T(d_x, d_y) \cdot P$$

Translation in Homogenen Koordinaten (2)

Eigenschaften:

$$\text{sei } P' = T(d_{x1}, d_{y1}) \cdot P \quad \text{und} \quad P'' = T(d_{x2}, d_{y2}) \cdot P'$$

$$\text{?????} = P'' = T(d_{x1} + d_{x2}, d_{y1} + d_{y2}) \cdot P$$

$$P'' = T(d_{x2}, d_{y2}) \cdot (T(d_{x1}, d_{y1}) \cdot P) = (T(d_{x2}, d_{y2}) \cdot T(d_{x1}, d_{y1})) \cdot P$$

$$\begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y1} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow T(d_{x1} + d_{x2}, d_{y1} + d_{y2}) = T(d_{x1}, d_{y1}) \cdot T(d_{x2}, d_{y2})$$

Skalierung in Homogenen Koordinaten

Eigenschaften:

$$\begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1} \cdot s_{x2} & 0 & 0 \\ 0 & s_{y1} \cdot s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow S(s_{x1}, s_{y1}) \cdot S(s_{x2}, s_{y2}) = S(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2})$$

Rotation in Homogenen Koordinaten

Eigenschaften:

$$\begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$r_{11} = \cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta$$

$$r_{12} = ??$$

....

$$\Rightarrow R(\alpha) \cdot R(\beta) = R(\alpha + \beta)$$

Rechenregeln für Matrizen

$$A + B = B + A$$

$$A + O = O + A = A$$

$$A(B + C) = AB + AC$$

$$A(BC) = (AB)C := ABC$$

$$(AB)^T = (B^T A^T)$$

$$(AB)^{-1} = (B^{-1} A^{-1})$$

ACHTUNG !!!!

Im Allgemeinen ist $AB \neq BA$

Verkettung von Transformationen

Rotation um Fixpunkt



(x_r, y_r)

$$R(x_r, y_r, \theta) =$$

$$T(x_r, y_r) \circ R(\theta) \circ T(-x_r, -y_r)$$

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos \theta & -\sin \theta & x_r(1 - \cos \theta) + y_r \sin \theta \\ \sin \theta & \cos \theta & y_r(1 - \cos \theta) - x_r \sin \theta \\ 0 & 0 & 1 \end{bmatrix}$$

Weitere Verkettungen

Skalierung um Fixpunkt

$$S(x_f, y_f, s_x, s_y) = T(x_f, y_f) \circ S(s_x, s_y) \circ T(-x_f, -y_f)$$

$$S(x_f, y_f, s_x, s_y) = \begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix}$$

Affine Transformationen

Eine Abbildung $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix}$ mit

$$x' = a_{xx}x + a_{xy}y + b_x$$

$$y' = a_{yx}x + a_{yy}y + b_y$$

heißt zweidimensionale affine Transformation!

Eigenschaften: Parallelen werden auf Parallelen abgebildet und endliche Punkte werden auf endliche Punkte abgebildet.

Eine affine Transformationen lässt sich immer in eine Rotation, eine Translation, eine Skalierung und eine Scherung zerlegen! **Leider nicht eindeutig!**

2D Transformationen und Rasterung

In Animationen mit aufwendigen Objekten können Rasteroperationen eine effiziente Alternative zur Berechnung affiner Transformationen sein!

Translation: Kopiere einen Pixelblock von einem Rastergebiet ins andere.

Rotation um $\mp 90^\circ$: Einfache Umsortierung der Blöcke. (umkehren der Pixelwerte jeder Zeile, dann Zeilen mit Spalten vertauschen)

Beliebige Rotationen und Skalierungen erfordern die Transformation der einzelnen Pixelfelder auf ihre Zielgebiete. (Intensitäten je nach Überlapp bestimmen.)

Warping & Morphing

Übersicht

Warping

- Geometrische Transformationen
- Forward and Inverse mapping
- Verschiebe Vektorfelder
- Schnelle Algorithmen für geom. Transformationen
- Probleme!

Morphing

- Mesh morphing
- Feature-based morphing
- Mehrdimensionales Morphing
- Morphing in drei und mehr Dimensionen

Warping: Transformationen der Bildgeometrie

Wofür:

- Korrektur von Linsenfehlern
- Vergrößern und verkleinern von Bildern
- Rotieren von Bildern
- Perspektivische Korrektur von Bildern
- Morphen!

z.B. Rotieren eines Bildes

Input [u,v]



Output [x,y]



Gesucht:

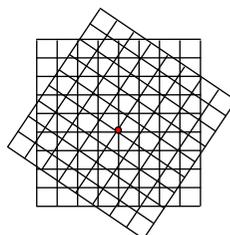
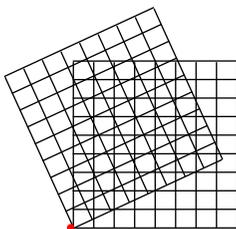
Eine Abbildung A die jedem Pixel im Output einen Farbwert aus der Vorlage zuweist.

$$\vec{x}' = A(\vec{x}) \quad \Rightarrow \quad \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A} \begin{bmatrix} u \\ v \end{bmatrix}$$

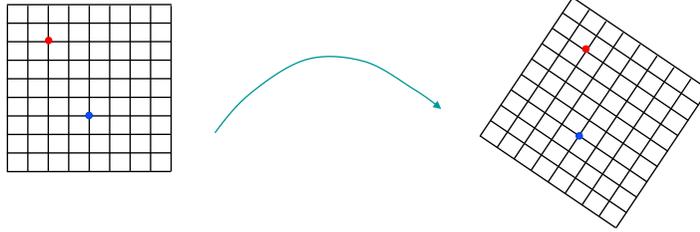
Rotation eines Bildes (2)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \end{bmatrix}$$



Rotation eines Bildes (3)



Gegeben zwei Bilder die sich nur durch eine Rotation unterscheiden!

Wie viele Punktkorrespondenzen zwischen den Bildern sind notwendig um die Rotation eindeutig zu bestimmen?

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \end{bmatrix}$$

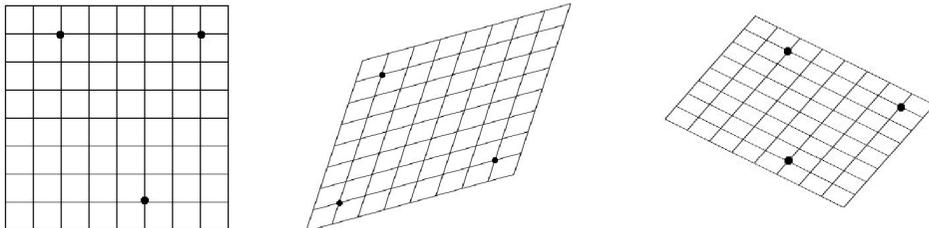
2 Punktpaare bestimmen die Rotation eindeutig!

Rotationen sind winkeltreu!

Affine Transformation

Eine affine Abbildung ist eine allgemeine lineare Abbildung die sich aus Translation, Rotation, Scherung, Skalierung und Streckung zusammensetzt.

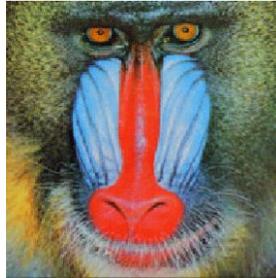
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \end{bmatrix}$$



3 Punktpaare bestimmen eine affine Abbildung eindeutig!

Bei affinen Abbildungen bleiben Parallelen parallel!

Affine Transformation ?



Eine perspektivische Abbildung ist **keine** affine Abbildung!!

Lässt sich ein planares perspektivisch verzerrtes Bild entzerren?

Wie viele Punktkorrespondenzen bestimmen eine perspektivische Abbildung?

Perspektivische Transformation

Die perspektivische Abbildung ist eine nicht lineare Funktion.

$$\begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + 1}$$

$$y = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + 1}$$

Geraden werden auf Geraden abgebildet!

Im allgemeinen bleibt weder die Parallelität von Geraden noch bleiben die Winkel zwischen Geraden erhalten.

Punkte können nach Unendlich abgebildet werden.

Wie viele Korrespondenzpaare bestimmen die Abbildung eindeutig?

Homogene Koordinaten (Einschub)

2D Kartesische Koordinaten \rightarrow Homogene Koordinaten

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} \quad \text{mit} \quad \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} \quad \forall w \neq 0,$$

wobei $\begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$ und $k \cdot \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$

den selben Punkt $P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ beschreiben, also auch $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x'}{w} \\ \frac{y'}{w} \\ 1 \end{bmatrix}$.

Perspektivische Transformation

Wir suchen die Abbildung $\begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix}$

In Homogenen Koordinaten

$$\begin{bmatrix} \tilde{w} \cdot x \\ \tilde{w} \cdot y \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} & a'_{23} \\ a'_{31} & a'_{32} & a'_{33} \end{bmatrix} \begin{bmatrix} w' \cdot u \\ w' \cdot v \\ w' \end{bmatrix} \quad \left| \begin{array}{l} \cdot \frac{1}{w'} \\ \cdot \frac{1}{a'_{33}} \end{array} \right.$$

$$\Rightarrow \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$x = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + 1}$$

$$y = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + 1}$$

Perspektivische Transformation

Lösen der Gleichungen mit 4 Punktkorrespondenzen. $(u_i, v_i), (x_i, y_i) \quad i = 0, 1, 2, 3$

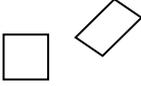
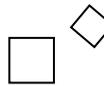
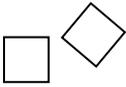
$$x = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + 1} \quad \Rightarrow \quad a_{11}u + a_{12}v + a_{13} - xa_{31}u - xa_{32}v = x$$

$$y = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + 1} \quad \Rightarrow \quad a_{21}u + a_{22}v + a_{23} - ya_{31}u - ya_{32}v = y$$

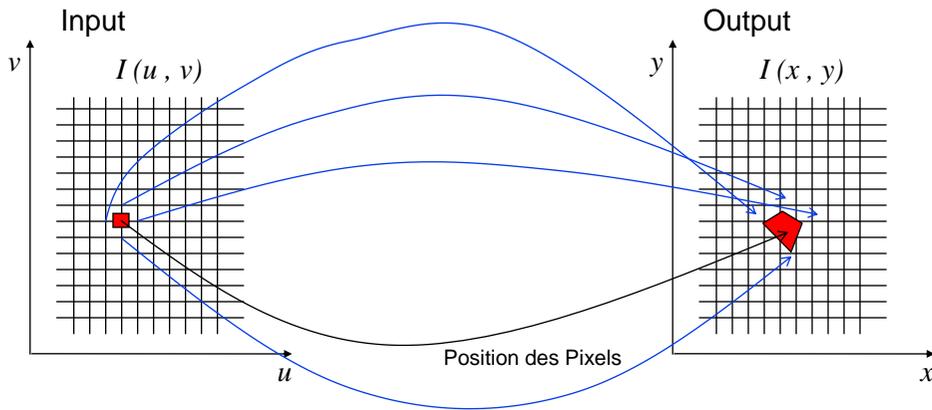
$$\Rightarrow \begin{pmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Planare Transformationen (Übersicht)

Multiple View Geometry in
Computer Vision
Hartley & Zissermann
Cambridge Press p23

Name	Matrix	Wirkung	Invariant
Projektiv	$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}$		Geraden, Schnittpunkte von Geraden,
Affine	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelen, Flächenverhältnisse, Linearkombinationen von Vektoren, ...
Ähnlich	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Längenverhältnisse & Winkel, ...
Euklidisch	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Längen & Flächen

„Forward Mapping“



$$x = u + \Delta u$$

$$y = v + \Delta v$$

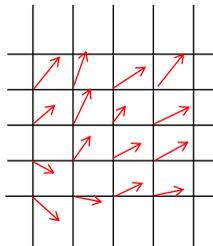
Problem: x, y sind nicht ganzzahlig!
Deshalb müssen die Nachbarpunkte in die Berechnung mit eingeschlossen werden.

„Forward Mapping (2)“

$$x_i = u_i + \underline{\Delta u_i}$$

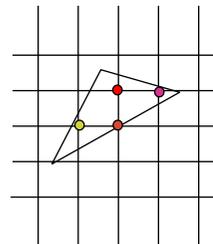
$$y_j = v_j + \underline{\Delta v_j}$$

Bezeichnung: $\Delta u, \Delta v$ wird auch mit Verschiebe-, Vektor- oder Korrespondenzfeld bezeichnet (*Displacement vector field*).



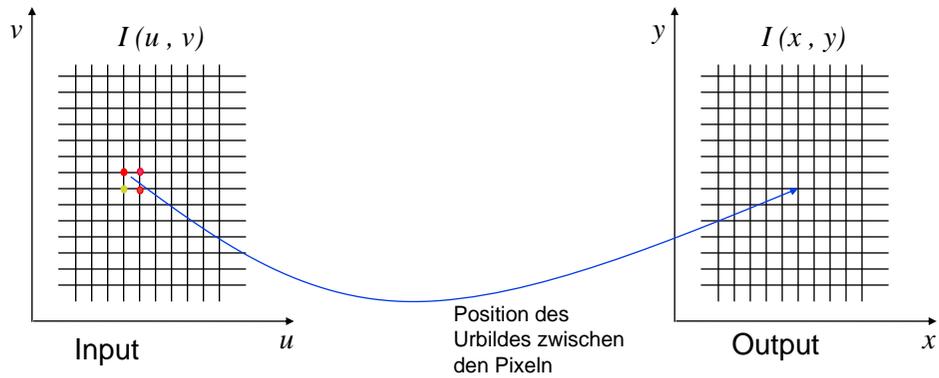
Vorgehensweise zur Implementierung:

1. Es werden mehrere Nachbarpunkte gleichzeitig betrachtet, meist Dreiecke.
2. Die Fläche zwischen diesen Punkten wird dann gerastert.



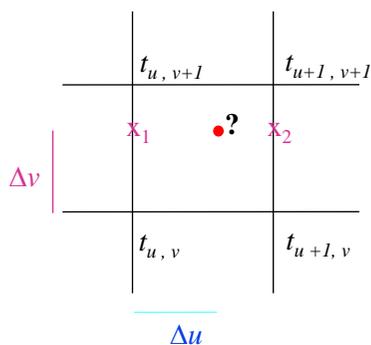
„Backward Mapping“

Kann die Inverse Abbildung zu $x = u + \Delta u$ bestimmt werden?
 $y = v + \Delta v$



Diese Abbildung wäre viel leichter auszuwerten! Denn es müsst ja nur zwischen den 4 umliegenden Pixeln interpoliert werden, und es könnten alle Pixel des Outputs der Reihe nach abgearbeitet werden. .

Einschub: Interpolation von Pixeln



Meist wird bilineare Interpolation benutzt!

$$x_1 = t_{u, v} + \Delta v (t_{u, v+1} - t_{u, v})$$

$$x_2 = t_{u+1, v} + \Delta v (t_{u+1, v+1} - t_{u+1, v})$$

$$\bullet = x_1 + \Delta u (x_2 - x_1)$$

„Backward Mapping“ (2)

Ob sich eine Abbildung invertieren lässt ist von der Abbildung selbst abhängig (z.B. Rotationen lassen sich invertieren).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{falls } \mathbf{A}^{-1} \text{ existiert}$$

$$\Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{array}{l} u = x + \Delta x \\ v = y + \Delta y \end{array}$$

Angenommen es ist nur ein Vektorfeld $\Delta u, \Delta v$ gegeben, kann damit ein „Backward mapping“ durchgeführt werden?

Das heißt:
kann aus $x = u + \Delta u$, $u = x + \Delta x$ berechnet werden
 $y = v + \Delta v$, $v = y + \Delta y$

„Backward Mapping“ (3)

Das heißt:
kann aus $x = u + \Delta u$, $u = x + \Delta x$ berechnet werden
 $y = v + \Delta v$, $v = y + \Delta y$

~~Ganz einfach:~~

~~$$x = u + \Delta u \Rightarrow u = x - \Delta u$$~~

~~$$y = v + \Delta v \Rightarrow v = y - \Delta v$$~~

Dies ist noch nicht die gesuchte Lösung!!!

Denn $\Delta u(u, v)$, $\Delta v(u, v)$ sind immer noch eine Funktion von u, v . Danach sollte aber jedoch aufgelöst werden.

Das Umkehren eines Korrespondenzfeldes kann nur über „forward warping“ erfolgen.

Schnelle Algorithmen für geom. Transformation

Für bestimmte Transformationen lassen sich spezielle optimierte Algorithmen implementieren.

Klassisches Beispiel: Separable Abbildungen

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= A \begin{bmatrix} u \\ v \end{bmatrix} \\ \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} &= B \circ C \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

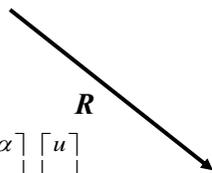
z.B. Die Rotation:

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tan \alpha & 1/\cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ 0 & 1 \end{bmatrix}$$

2-Pass Rotation



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tan \alpha & 1/\cos \alpha \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

3-Pass Rotation

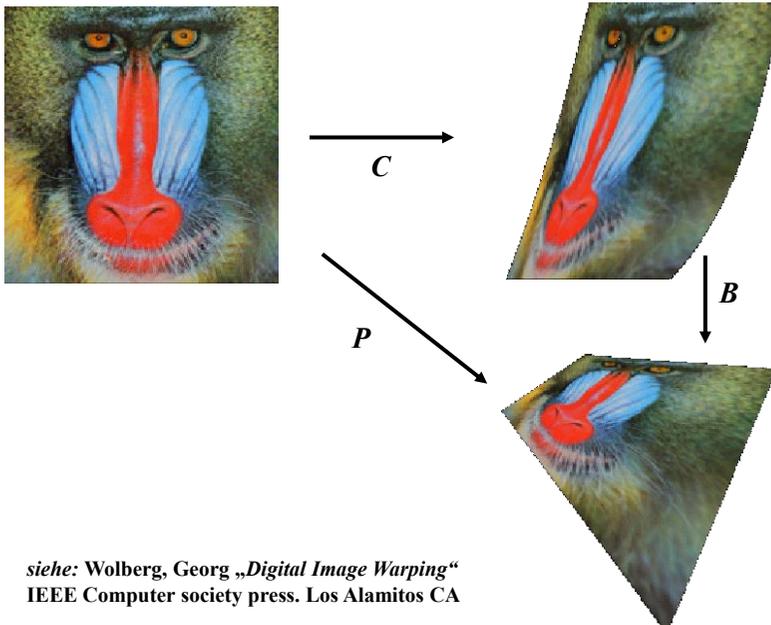
2- Schritt Rotation: Durch die Skalierung kann Information verloren gehen

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tan \alpha & 1/\cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ 0 & 1 \end{bmatrix}$$

Durch ein 3- Schritt Verfahren kann dies vermieden werden,
Es sind nur noch eindimensionale Scherungen nötig.

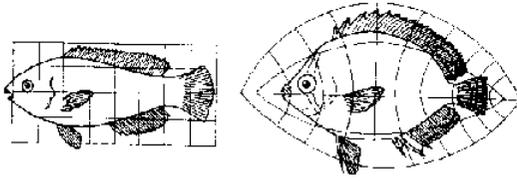
$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & \tan(\alpha/2) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sin \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & \tan(\alpha/2) \\ 0 & 1 \end{bmatrix}$$

2-Pass für perspektivische Transformation



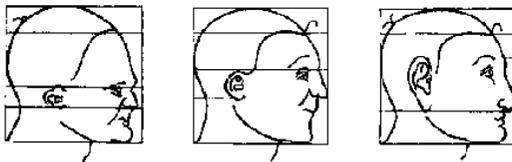
Warping zur Beschreibung einer Methamorphose

Morphing

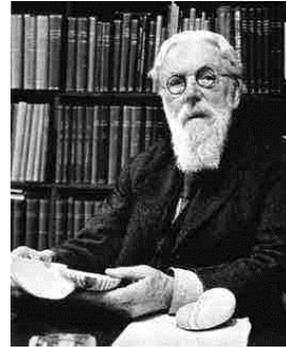


Scarus sp.

Pomacanthus.



D'Arcy Thompson (after Albrecht Dürer)



D'Arcy Thompson 1860 – 1948
Mathematiker aus Schottland

D'Arcy Thompson (2)

Choose picture: Argyrolepecus olfersi

Show grid

$F(x, y) = (p(x, y), q(x, y))$ where Identity map

$p(x, y) = \frac{0.000}{\downarrow \uparrow} x^2 + \frac{0.000}{\downarrow \uparrow} xy + \frac{0.000}{\downarrow \uparrow} y^2 + \frac{1.000}{\downarrow \uparrow} x + \frac{0.500}{\downarrow \uparrow} y$ and Clear

$q(x, y) = \frac{0.000}{\downarrow \uparrow} x^2 + \frac{0.000}{\downarrow \uparrow} xy + \frac{0.000}{\downarrow \uparrow} y^2 + \frac{0.000}{\downarrow \uparrow} x + \frac{1.400}{\downarrow \uparrow} y$

x-range: -3 to 3 y-range: -3 to 3
 Change both ranges: ↔ Default ranges: ↔
 Rate of change: 20 Line width: ↔

Morphing : Addition von Bilder

$\frac{1}{2}$  + $\frac{1}{2}$  =

Keine Korrespondenz 

Korrespondenz 

Korrespondenz ist die Punkt zu Punkt Zuordnung zwischen zwei Bildern und kann durch ein Korrespondenzfeld beschrieben werden.

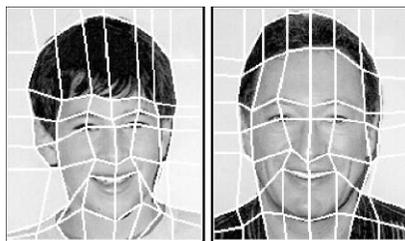
Erstellen von Korrespondenzfeldern

Das Hauptproblem beim Morphing ist das Erstellen von Korrespondenzfeldern (Verschiebefeldern).

Manuelle Verfahren sind die Regel: Meist wird die Korrespondenz an wenigen Stellen manuell bestimmt und für die restlichen Pixel interpoliert.

Die Verfahren unterscheiden sich nur in der Interpolationsmethode.

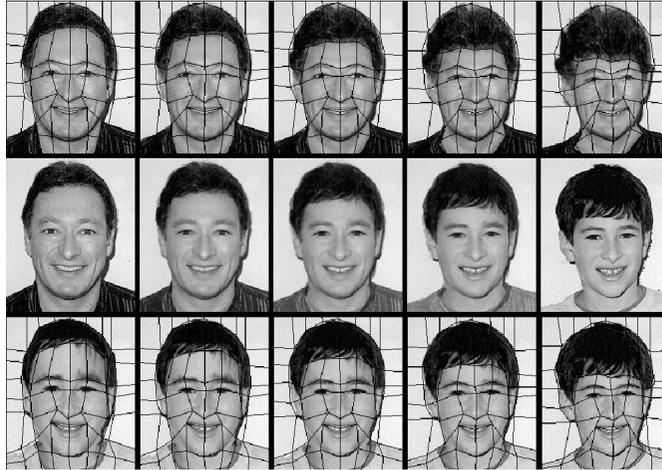
1. Mesh Warping.



2. Field Warping.



Mesh Warping



aus: Wolberg, Georg „*Image Morphing: A Survey*“
Department of Computer Science City College of New York

Field Warping

Beim „Mesh Warping“ entstehen immer dann Bildfehler, wenn die Gitterkanten nicht mit den Bildkonturen übereinstimmen.

Ausweg: Man markiert nur noch Konturen in den Bildern (manuell). Durch das Abbilden der Konturen auf einander, wird dann für einige Punkte die Korrespondenz bestimmt, die dann auf die anderen Bildpunkte ausgedehnt wird.

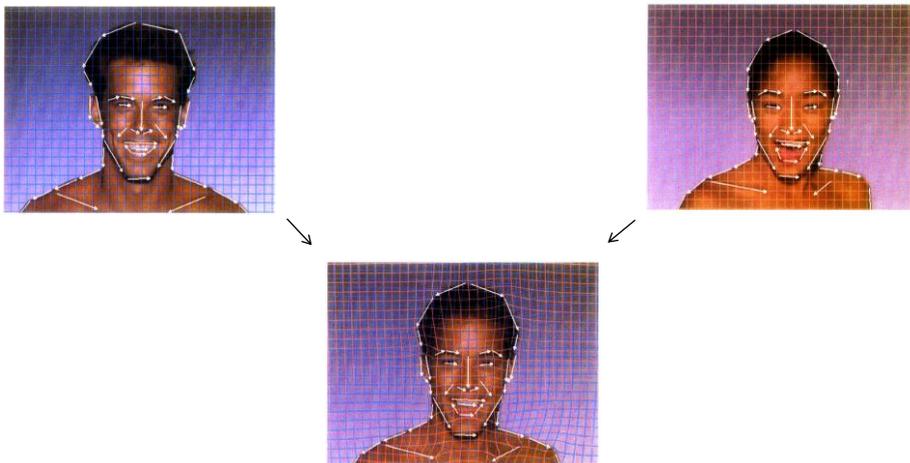
Merkmalsbasiertes Morphing (Field Morphing)



aus: S. Lee, G. Wolberg, K. Chwa and S. Y. Shin
Image Metamorphosis with Scatter Feature Constraints
IEEE Transactions on Visualization and Computer Graphics 2:4 1996

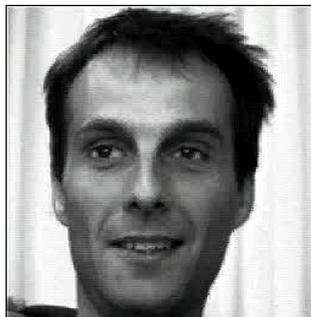
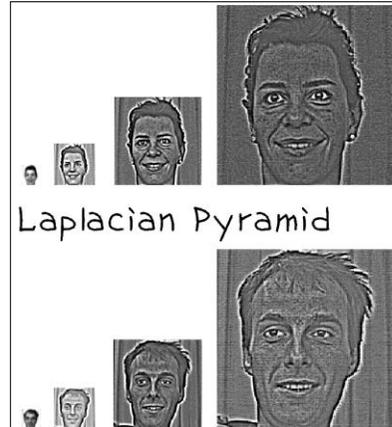
Most Famous "Black or White" in 1991

"Feature-Based Image Metamorphosis",
by T. Beier and S. Neely, Siggraph 1992



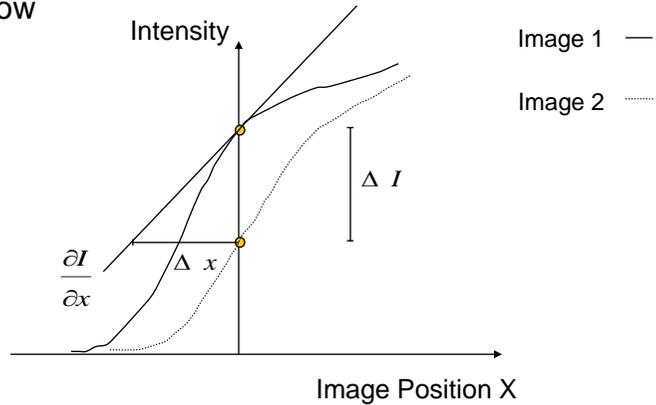
Gradienten Verfahren zur Optischen Fluss Berechnung

$$\sum_{\text{pixel}} \left(\Delta I - \frac{\partial I}{\partial x} \Delta x - \frac{\partial I}{\partial y} \Delta y \right)^2 = \min!$$



Korrespondenz zwischen zwei Bildern

Gradient-based Optical Flow

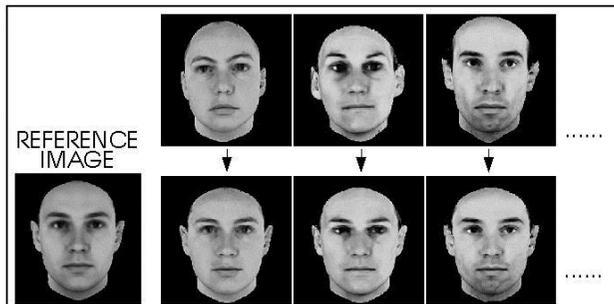


$$\Delta I = \frac{\partial I}{\partial x} \Delta x$$

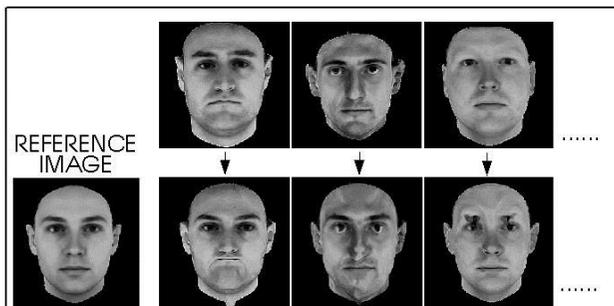
$$\sum_{\text{pixel}} \left(\Delta I - \frac{\partial I}{\partial x} \Delta x - \frac{\partial I}{\partial y} \Delta y \right)^2 = \min!$$

Grenzen des ,Optischen Fluss' Verfahrens

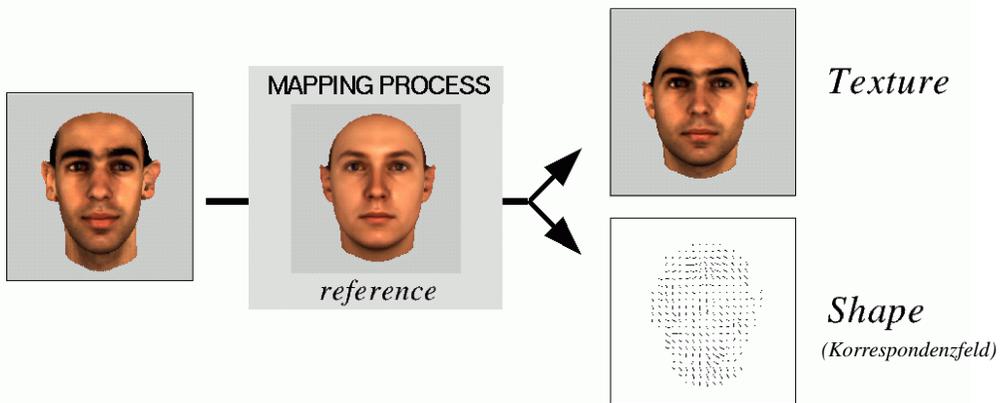
Erfolgreich



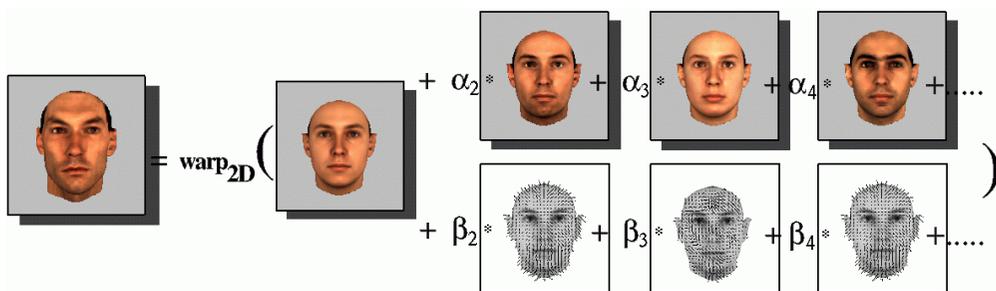
Mit Fehlern



Die Korrespondenz beschreibt den Formunterschied



Mehrdimensionales Morphen



Morphing in mehreren Dimensionen.

Morphing oder Interpolations Verfahren werden heute auf eine Vielzahl unterschiedlicher Datenstrukturen zur Bilderzeugung verwendet.

3D Objekte

Animation

.....

