Geometrische Operationen

Bisher haben wir Operationen betrachtet, die die Intensität, Farbe etc. von Pixeln verändert haben.

Jetzt betrachten wir Operationen, die die Position von Pixeln verändern.

Original

Verzerrte Bilder

Geometrische Operationen

Eine **geometrische Operation** ist eine Abbildung (Transformation)

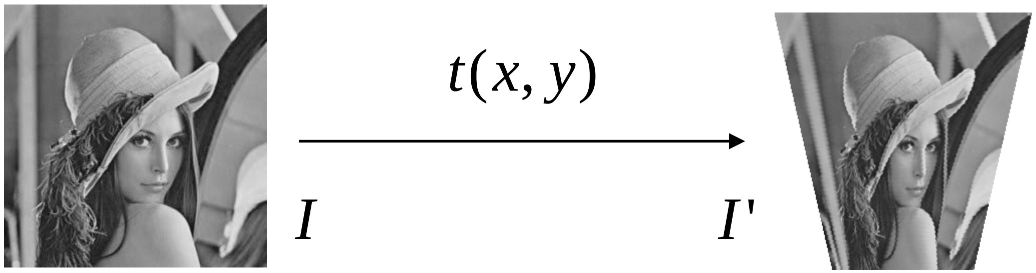
$$t : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}$$

die jedem Pixel (x,y) von Bild I eine neue Pixelposition (x',y') von Bild I' zuweist.

Beispiel: Die Abbildung

$$t(x, y) = (0.5 \cdot \tfrac{y}{H}(x - \tfrac{B}{2}) + \tfrac{B}{2}, y)$$

erzeugt eine perspektivische Transformation:



Schreibweisen

Geometrische Operationen setzen sich also aus zwei Funktionen zusammen:

$$x' = t_x(x, y)$$

$$y' = t_y(x, y)$$

In der Regel verwenden wir für geometrische Operationen Vektorschreibweise, d.h. wir fassen die Pixelpositionen als zweidimensionale Vektoren auf:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = t\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x(x, y) \\ t_y(x, y) \end{pmatrix}$$

oder kurz

$$\vec{x}' = t(\vec{x})$$

Ziel-nach-Quelle Transformation I

Problem: Wenn wir die Grauwerte von I mit einer Funktion ins Bild I' abbilden, liegen die neuen Positionen (x',y') in der Regel nicht auf dem Pixelraster

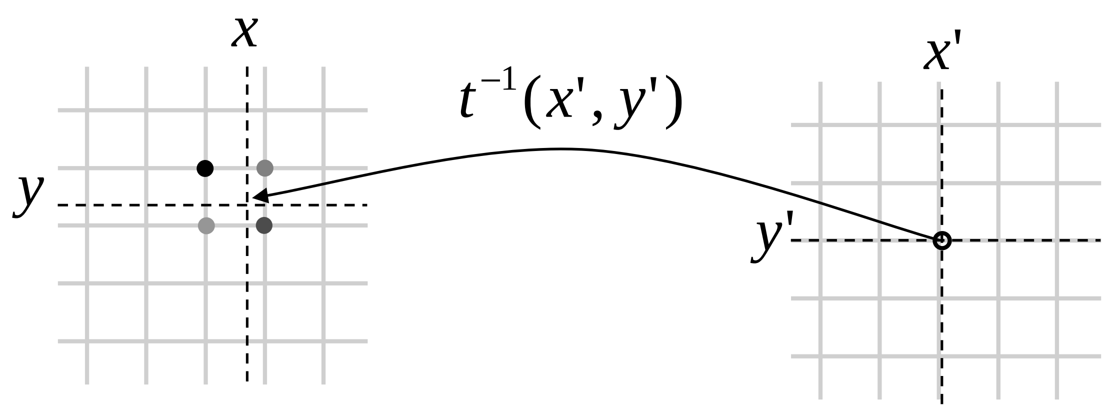
$$I \xrightarrow{t(x,y)} I'$$

Lösung: Für jede Pixelposition (x',y') im (Ziel-)Bild I': Berechne mit der Umkehrabbildung t^{-1} von t die ursprünglichen Koordinaten (x,y) in I

$$I \xleftarrow{t^{-1}(x',y')} I'$$

Ziel-nach-Quelle Transformation II

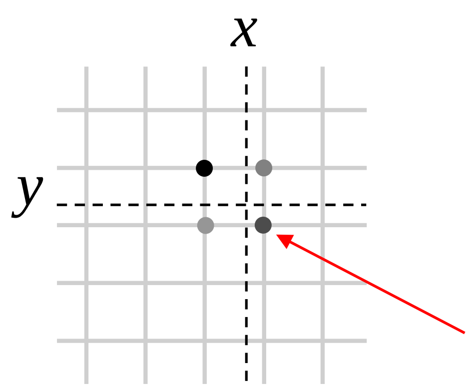
Die Position (x,y) in I wird ebenfalls i.d.R. nicht im Pixelraster liegen. Hier jedoch können wir interpolieren:



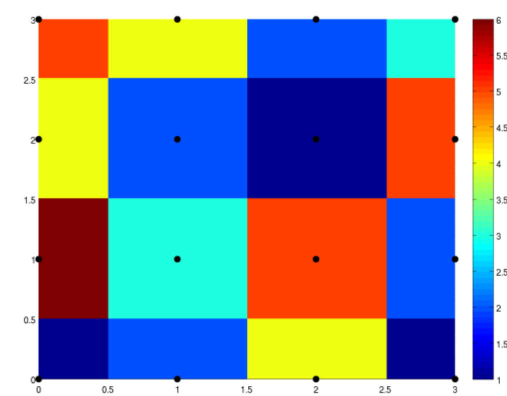
Wir bestimmen also den Grauwert von (x',y') aus den Nachbarpixeln von (x,y) !

Nächster Nachbar

Die einfachste Methode ist die **Nächster-Nachbar-Methode**: Wir suchen diejenige Pixelposition, die am nächsten an (x,y) liegt und verwenden deren Grauwert:



Diese Methode wird selten eingesetzt…



http://www.codecogs.com/library/maths/approximation/interpolation/multivariate.php

Bilineare Interpolation I

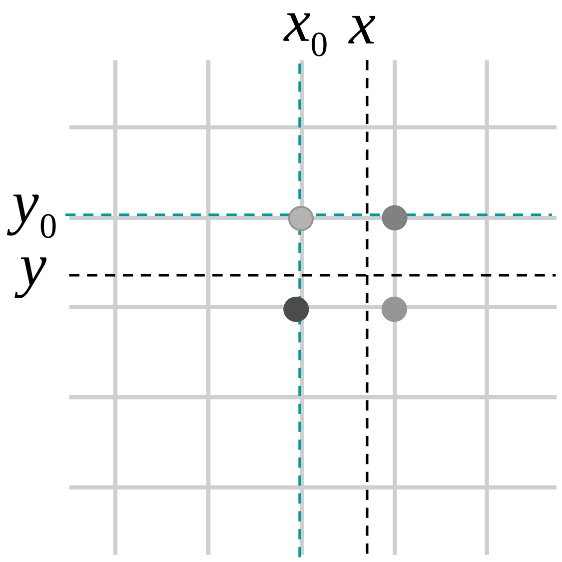
Kapitel 6

Geom. Operationen

Transformationen

- affin
- projektiv
- nicht-linear

Die gängigste Methode ist die der **bilinearen Interpolation**:



Sei $x_0 = \lfloor x \rfloor$ und $y_0 = \lfloor y \rfloor$

und

$\delta_x = x - x_0$ und $\delta_y = y - y_0$

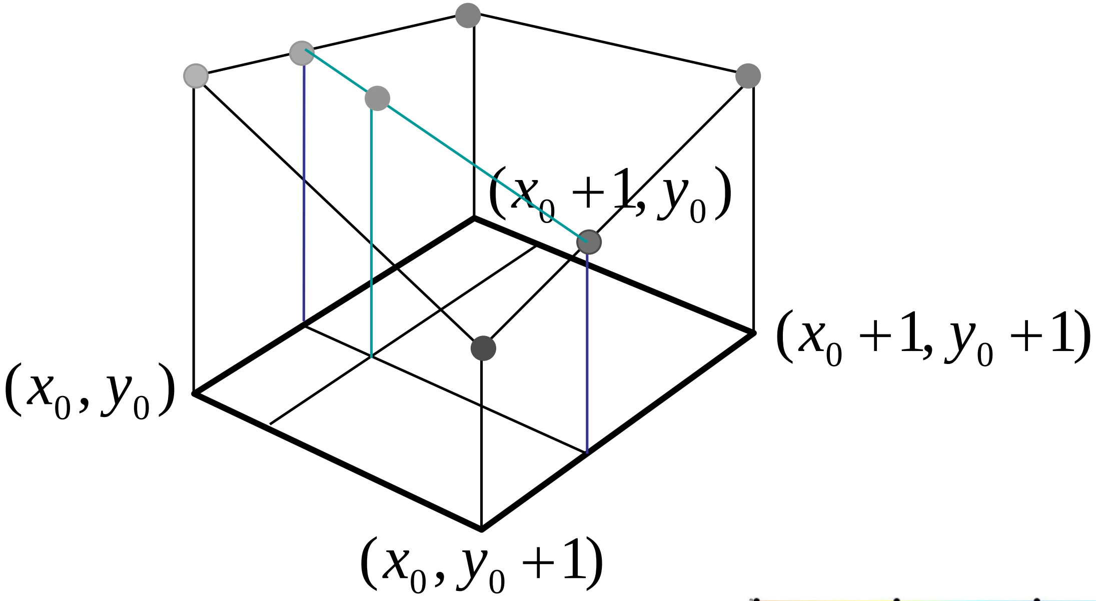
dann ist die bilineare Interpolation gegeben durch:

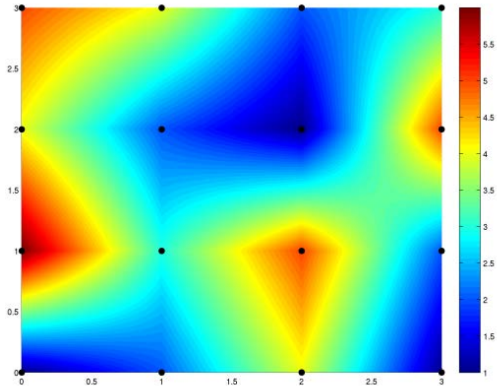
$$
\begin{aligned}
I(x,y) &= I(x_0+\delta_x, y_0+\delta_y) \\
&= (1-\delta_x)(1-\delta_y) I(x_0, y_0) \\
&+ \delta_x(1-\delta_y) I(x_0+1, y_0) \\
&+ (1-\delta_x)\delta_y I(x_0, y_0+1) \\
&+ \delta_x \delta_y I(x_0+1, y_0+1)
\end{aligned}
$$

Astrid Laubenheimer
Stand 28.06.2018

Bilineare Interpolation II

Die bilineare Interpolation ist die Bildung des gewichteten Mittelwerts der benachbarten Grauwerte.





http://www.codecogs.com/library/maths/approximation/interpolation/multivariate.php

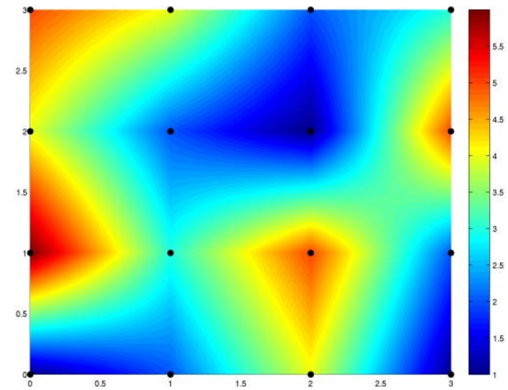
Astrid Laubenheimer
Stand 28.06.2018

Bikubische Interpolation

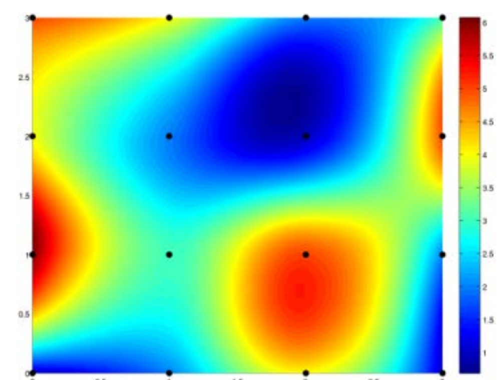
Alternativ kann auch bikubisch interpoliert werden. Dabei wird auch die Steigung (1. Ableitung) berücksichtigt.

Konsequenz: Es müssen noch Nachbarpixel berücksichtigt werden.

Bilinear



Bikubisch



http://www.codecogs.com/library/maths/approximation/interpolation/multivariate.php

Affine Transformationen I

Kapitel 6

Geom. Operationen

Transformationen

- affin
- projektiv
- nicht-linear

Affine Transformationen mit $\det A \neq 0$ heißen **Affinität**.

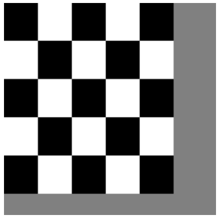
Die einfachsten geometrischen Operationen sind die **affinen Transformationen**, also diejenigen Operationen, die sich als Matrixmultiplikation und / oder Vektoraddition

$$\vec{x}' = A\vec{x} + \vec{b} \quad \text{bzw} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$

schreiben lassen. Zu diesen Transformationen gehören

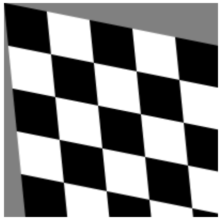
- **Skalierung** längs der x- bzw. y-Achse um einen Faktor s_x bzw. s_y:

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



- **Scherung** längs der x- bzw. y-Achse um Faktor b_x bzw. b_y:

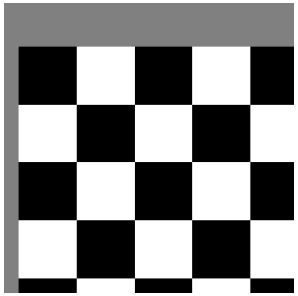
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Affine Transformationen II

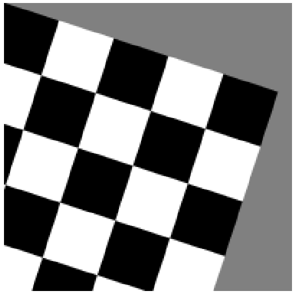
- **Translation** um einen Vektor $\vec{b}$:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$



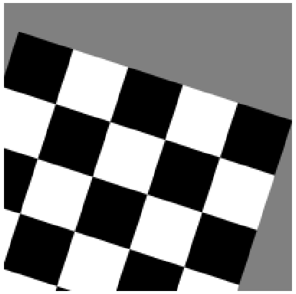
- **Drehung** um den Winkel α

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



- oder als **Bewegung** (erst Drehung, dann Translation)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$



Astrid Laubenheimer
Stand 28.06.2018

Homogene Koordinaten

Affine Transformationen lassen sich linearisieren, wenn man sie von kartesischen in **homogene Koordinaten** überführt:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_x \\ a_{21} & a_{22} & b_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Allgemeiner sind affine Transformationen genau diejenigen Abbildungen, die in homogenen Koordinaten wie folgt darstellbar sind:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Drehung um einen Punkt

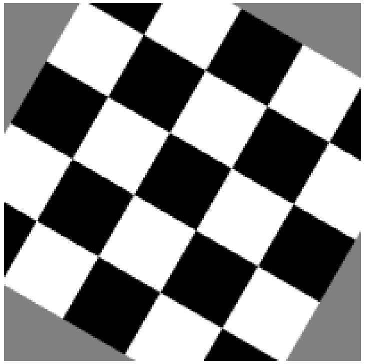
In dieser Schreibweise wird eine Drehung um die Pixelposition (x_c, y_c)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Translation um } (x_c,\, y_c)} \underbrace{\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Drehung um } \alpha} \underbrace{\begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Translation um } (-x_c,\, -y_c)} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

zu der folgenden Vorschrift

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & -x_c\cos\alpha + y_c\sin\alpha + x_c \\ \sin\alpha & \cos\alpha & -x_c\sin\alpha - y_c\cos\alpha + y_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} x_c + (x - x_c)\cos\alpha - (y - y_c)\sin\alpha \\ y_c + (x - x_c)\sin\alpha + (y - y_c)\cos\alpha \\ 1 \end{pmatrix}$$



Eigenschaften affiner Transformationen

Affine Transformationen haben folgende Eigenschaften:

- Die **Verkettung** zweier affiner Transformationen ist wieder eine affine Transformation (z.B. darstellbar als Multiplikation der entsprechenden homogenen Matrizen)
- Affine Transformationen bilden
  - Geraden auf Geraden,
  - Dreiecke auf Dreiecke,
  - Parallele Geraden auf parallele Geraden und
  - Rechtecke auf Parallelogramme

  ab.
- Das Abstandsverhältnis von Punkten, die auf einer Geraden liegen, bleibt bei affinen Transformationen erhalten.

Geradentreue
Parallelentreue
Teilverhältnis-
treue

Charakterisierung affiner Transformationen

Eine affine Transformationen ist durch <u>drei</u> Paare von Punkten (Vektoren)

$$\vec{x}_i \rightarrow \vec{x}'_i \quad \text{für} \quad i = 1,2,3$$

eindeutig bestimmt, vorausgesetzt, die Vektoren $\vec{x}_1, \vec{x}_2$ und $\vec{x}_3$ bilden ein echtes Dreieck, liegen also nicht auf einer Geraden.

Die affine Transformation kann durch Lösung eines linearen Gleichungssystems mit <u>sechs</u> Gleichungen bestimmt werden, denn jedes Punktepaar liefert zwei Gleichungen:

$$x'_i = a_{11}x_i + a_{12}y_i + a_{13}$$
$$y'_i = a_{21}x_i + a_{22}y_i + a_{23}$$

Eine geschlossene Form der Lösung findet man z.B. in [Burger & Burge 2005], siehe Literaturliste zur Vorlesung.

Projektive Transformationen

Eine **Homographie** (**Kollineation, Vier-Punkt-Transformation**) ist eine Abbildungen der Form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{w} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

i.d.R $\neq 1$

Im Gegensatz zu den affinen Transformationen muss hier nach der Matrix-Vektor-Multiplikation normiert werden, d.h. das (zweidimensionale) Ergebnis einer **projektiven Transformation** ist:

$$x' = \frac{\hat{x}}{\hat{w}} \quad \text{und} \quad y' = \frac{\hat{y}}{\hat{w}}$$

Multiplikation der Matrix mit einer Zahl $s \neq 0$ ändert nichts am Ergebnis!

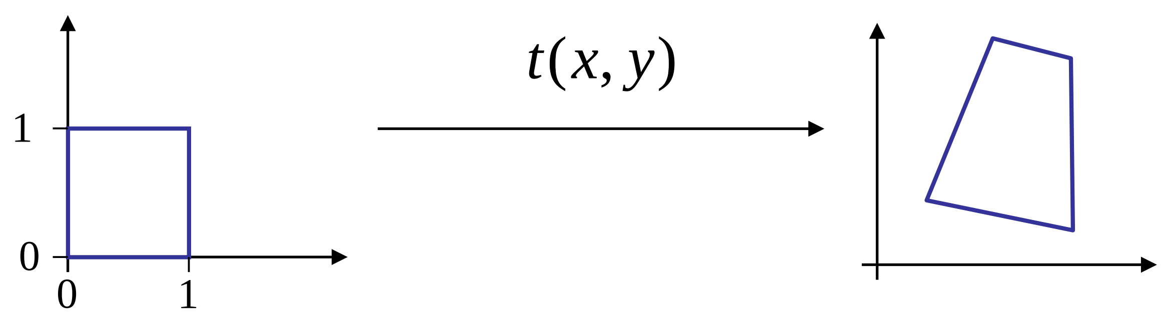
Homographien spielen in der Bildverarbeitung eine große Rolle, insbesondere, wenn es um 3D-Bildverarbeitung geht!!!

Eigenschaften projektiver Transformationen

Projektive Transformationen haben folgende Eigenschaften:

- Die **Verkettung** zweier projektiver Transformationen ist wieder eine projektive Transformation
- Projektive Transformationen bilden
  - Geraden auf Geraden und
  - Rechtecke auf Vierecke ab.
- Abstandsverhältnisse und Parallelität bleiben i.A. nicht erhalten!
- Mit einer projektiven Transformationen lässt sich jedes Viereck auf jedes andere Viereck abbilden:

Geradentreue



Charakterisierung projektiver Transformationen

Eine projektive Transformation ist durch <u>vier</u> Paare von Punkten

$$\vec{x}_i \rightarrow \vec{x}'_i \quad \text{für} \quad i = 1,..,4$$

(bis auf einen Faktor s) eindeutig bestimmt, vorausgesetzt, die Vektoren $\vec{x}_1, \vec{x}_2, \vec{x}_3$ und $\vec{x}_4$ bilden ein nicht degeneriertes Viereck.

Die Transformation kann durch Lösung eines linearen Gleichungssystems mit <u>acht</u> Gleichungen bestimmt werden, denn wieder liefert jedes Punktepaar zwei Gleichungen:

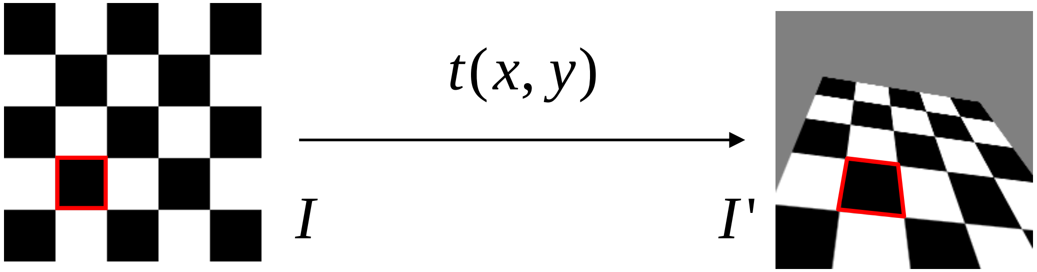
$$x'_i = a_{11}x_i + a_{12}y_i + a_{13} - a_{31}x_i x'_i - a_{32}y_i x'_i$$

$$y'_i = a_{y1}x_i + a_{y2}y_i + a_{y3} - a_{31}x_i y'_i - a_{32}y_i y'_i$$

Das Gleichungssystem wird i.d.R. mit Standard-Methoden zur Lösung linearer Gleichungssysteme wie z.B. DLT (direkte lineare Transformation) gelöst.

Ein OpenCV-Beispiel

Beispiel: Eine projektive Transformation erzeugt durch vier Paare von Pixelpositionen:



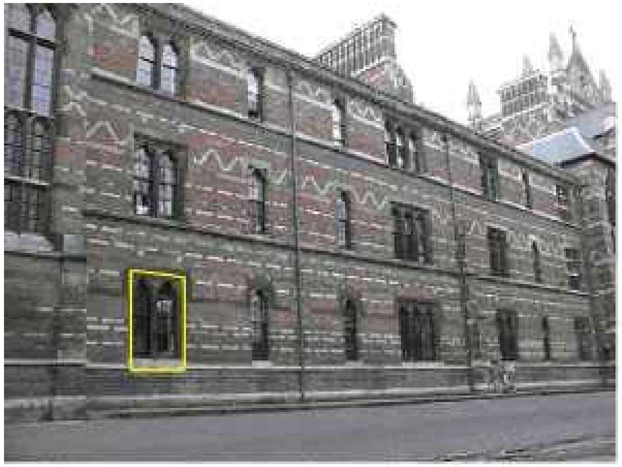

```
CvMat* homo=cvCreateMat(3,3,CV_32FC1);
CvPoint2D32f *c1 = new CvPoint2D32f[4];  // source points
CvPoint2D32f *c2 = new CvPoint2D32f[4];  // dest points
c1[0].x = 40.0;    c1[0].y = 159.0;
c1[1].x = 80.0;    c1[1].y = 159.0;
  ⋮

homo = cvGetPerspectiveTransform(c1, c2, homo);
cvWarpPerspective(I, IStrich, homo, 1+8, cvScalarAll(128));

delete[] c1;  delete[] c2; cvReleaseMat( &homo );
```

Beispiel Gebäuderekonstruktion

Beispiel: Eine projektive Transformation zur Herstellung einer Frontalansicht:



Quelle: http://plus.maths.org/content/getting-picture

Beispiel Geokodierung I

Kapitel 6

Geom. Operationen

Transformationen

- affin
- **projektiv**
- nicht-linear

Für die Geokodierung wird eine Transformation von einem aktuellen Bild I auf eine Karte oder (wie hier) auf ein Orthophoto I‘ gesucht:

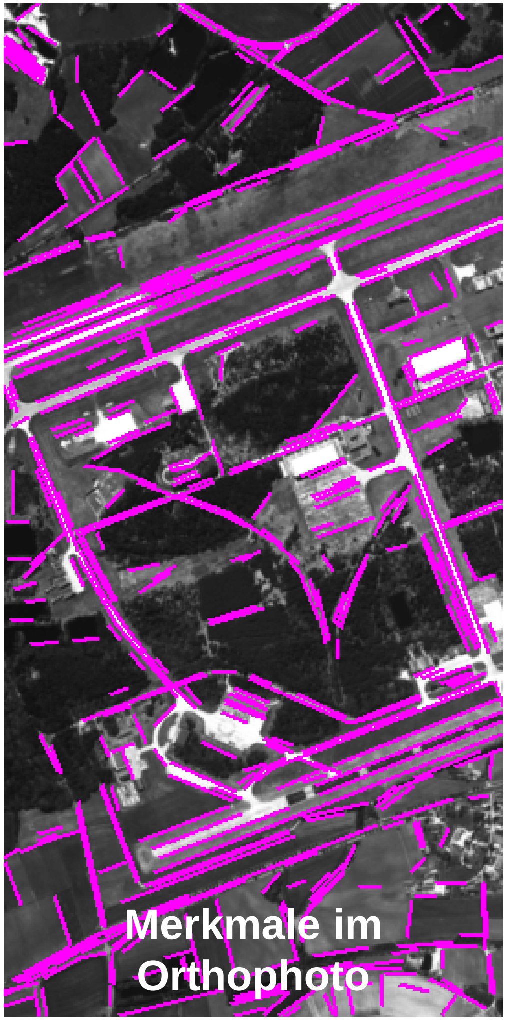
Referenzierung t

Vogel-perspektive »bird‘s eye view«

Astrid Laubenheimer

Beispiel Geokodierung II

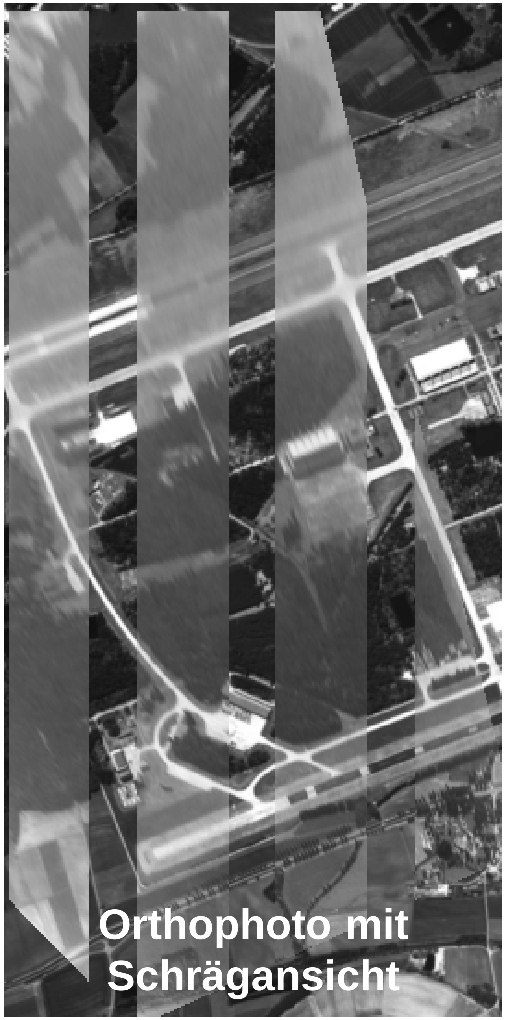
Für die Referenzierung werden aus beiden Bildern Merkmale extrahiert, hier z.B. Strecken:

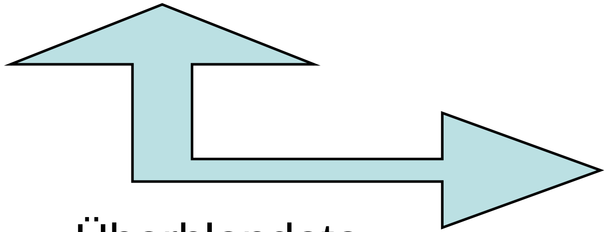


Bestimme eine projektive Abbildung, die die Merkmale der Schrägansicht optimal auf die Merkmale des Orthophotos abbildet.

Beispiel Geokodierung III

Wenn die Parameter der Transformation bekannt sind, können die bekannten Geokoordinaten der Karte oder des Orthophotos auf das aktuelle Bild transformiert werden.





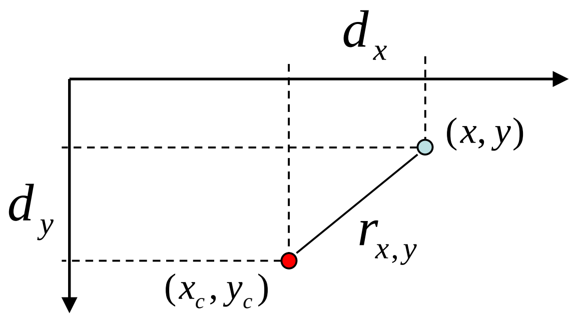
Überblendete
Ergebnisse der Referenzierung

Radiale Transformationen

Für eine **radiale Transformation** wird ein Ankerpunkt (x_c, y_c) festgelegt und für jedes Pixel (x, y) die folgenden Werte (x', y') berechnet

$$d_x = x - x_c \quad \text{und} \quad d_y = y - y_c$$

$$r = \sqrt{d_x^2 + d_y^2}$$



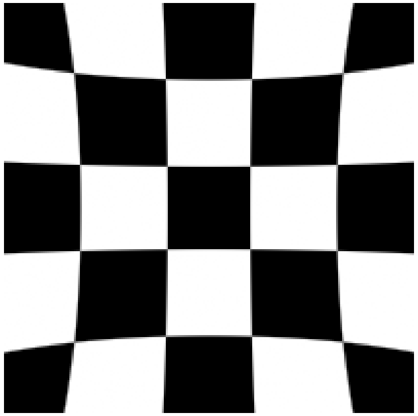
Je nach gewünschtem Grad $2n$ wird die Transformation durch eine Reihe von Parametern $k_2, k_4, \ldots k_{2n}$ bestimmt. Sie ist gegeben durch

$$x' = x + d_x\,(1 + k_2 r^2 + k_4 r^4 + \ldots + k_{2n} r^{2n})$$

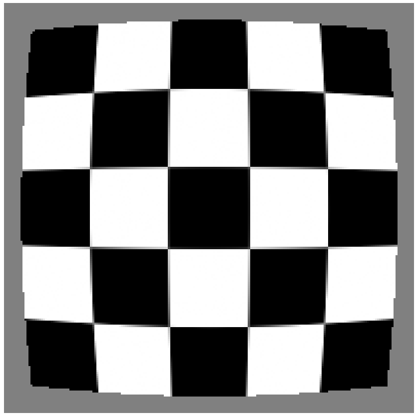
$$y' = y + d_y\,(1 + k_2 r^2 + k_4 r^4 + \ldots + k_{2n} r^{2n})$$

Beispiel Radiale Transformation

Beispiel: Eine radiale Transformation vom Grad 2 (n=1) bezogen auf die Bildmitte als Ankerpunkt



Kissenförmige Verzerrung
k_2 = -0.00001



Tonnenförmige Verzerrung
k_2 = 0.00001

Astrid Laubenheimer
Stand 28.06.2018

Bilineare Transformationen

Bilineare Transformationen sind Polynome ersten Grades mit zwei Veränderlichen x und y: bezogen auf den Ankerpunkt (x_c, y_c)

$$x' = a_1 d_x + a_2 d_y + a_3 d_x d_y + a_4 + x_c$$
$$y' = b_1 d_x + b_2 d_y + b_3 d_x d_y + b_4 + y_c$$

mit $d_x = x - x_c$ und $d_y = y - y_c$

Beispiel: Mit der Bildmitte als Ankerpunkt und

$$a_1 = 1, a_2 = 0, a_3 = 0.002, a_4 = 0$$
$$b_1 = 0, b_2 = 1, b_3 = -0.005, b_4 = 0$$