Karlsruher Institut für Technologie

Hochperformante Humanoide Technologien

Robotics I: Introduction to Robotics

Excercise 6 – Image Processing

Christian Marzi, Tamim Asfour

http://www.humanoids.kit.edu

Content

- **Task 1**

  Color Representation

- **Task 2**

  Camera Model

- **Task 3**

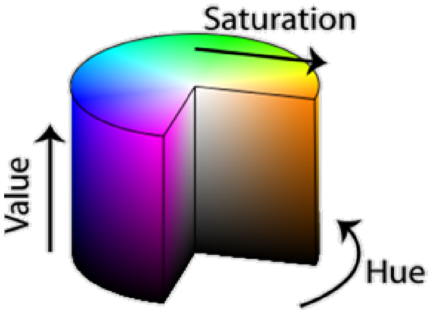
  Filters in Image Processing

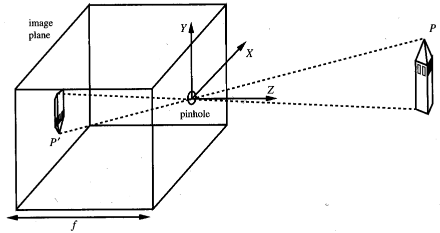
- **Task 4**

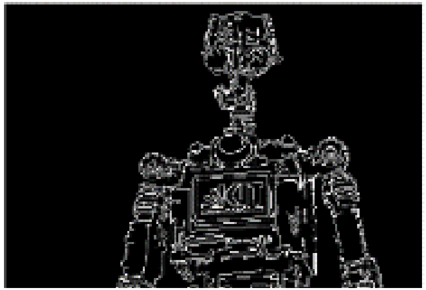
  Segmentation

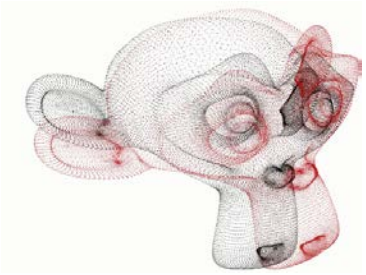
- **Task 5**

  Iterative Closest Point (ICP) Algorithm









Inhalt

- **Task 1**

  Color Representation

- **Task 2**

  Camera Model

- **Task 3**

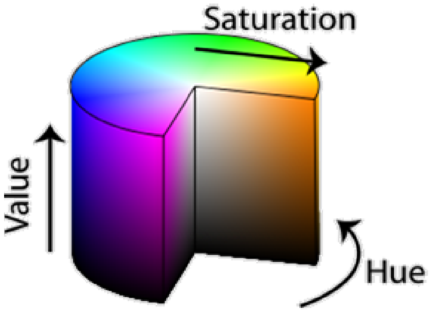
  Filters in Image Processing

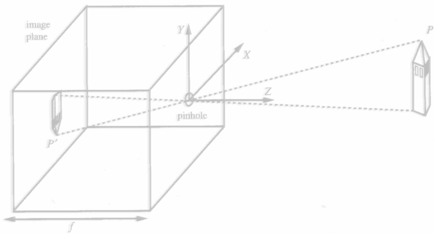
- **Task 4**

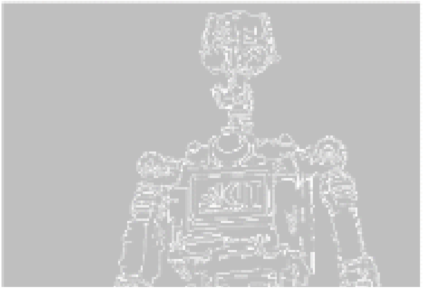
  Segmentation

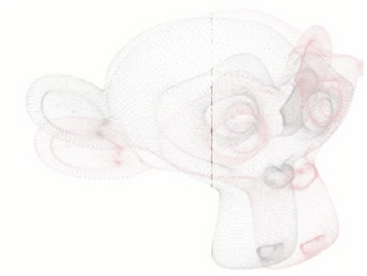
- **Task 5**

  Iterative Closest Point (ICP) Algorithm









Task 1

1. Transform the following colors specified in the RGB model into the HSI format.
   1. (120, 80, 210)
   2. (0, 150, 130)

2. A robot observes at a pack of cereal in a windowless laboratory. Someone turns the light a little brighter.
   1. Do the R, G or B values of the cereal package change?
   2. Do the H, S or I values of the cereal package change?

3. The robot observes the pack in a kitchen with windows. Outside, a cloud pushes in front of the sun.
   1. Do the R, G or B values of the cereal package change?
   2. Do the H, S or I values of the cereal package change?

HSI Color Representation

- *Transform the following colors specified in the RGB model into the HSI format.*

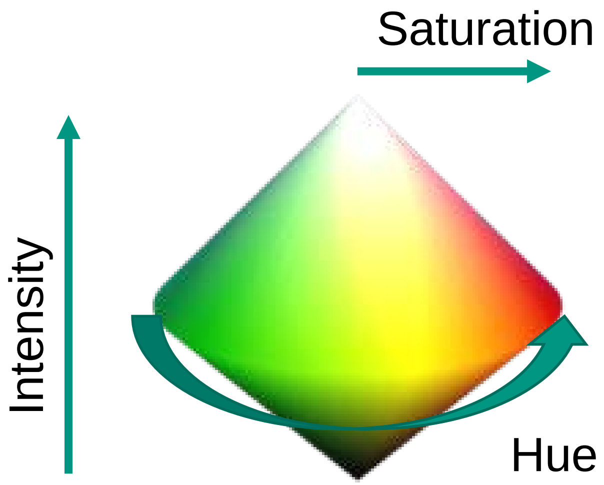
RGB to HSI Transformation:

$$H = \begin{cases} \theta, & \text{if } B < G \\ 360 - \theta, & \text{else} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R + G + B)$$



Lecture Chapter 9 Slide 22

Calculating Hue

- R:120, G: 80, B: 210

$$B < G\,?$$

$$120 \nless 80 \Rightarrow H = 360 - \theta$$

$$H = \begin{cases} \theta, & \text{if } B < G \\ 360 - \theta, & \text{else} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R + G + B)$$

R:120, G: 80, B: 210

Calculating Hue

$$H = 360 - \theta$$

- $\theta = \cos^{-1}\left(\frac{2R - G - B}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$
- $\theta = \cos^{-1}\left(\frac{240-80-210}{2\sqrt{(120-80)^2+(120-210)(80-210)}}\right)$
- $\theta = \cos^{-1}\left(\frac{-50}{2\sqrt{1600+11700}}\right)$
- $\theta \approx \cos^{-1}(-0.2168)$
- $\theta \approx 102.5°$
- $\Rightarrow H = 360° - \theta \approx \mathbf{257.5}°$

$$H = \begin{cases} \theta, & \text{if } B < G \\ 360 - \theta, & \text{else} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R+G+B)$$

R:120, G: 80, B: 210

Calculating Saturation und Intensity

- $I = \frac{R+G+B}{3}$
- $I = \frac{410}{3}$
- $I \approx 136.7$

- $S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$
- $S = 1 - \frac{3}{120+80+210} \cdot 80$
- $S = 1 - \frac{240}{410}$
- $S \approx 0.415$

$$H = \begin{cases} \theta, & \text{if } B < G \\ 360 - \theta, & \text{else} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R+G+B)$$

R:120, G: 80, B: 210

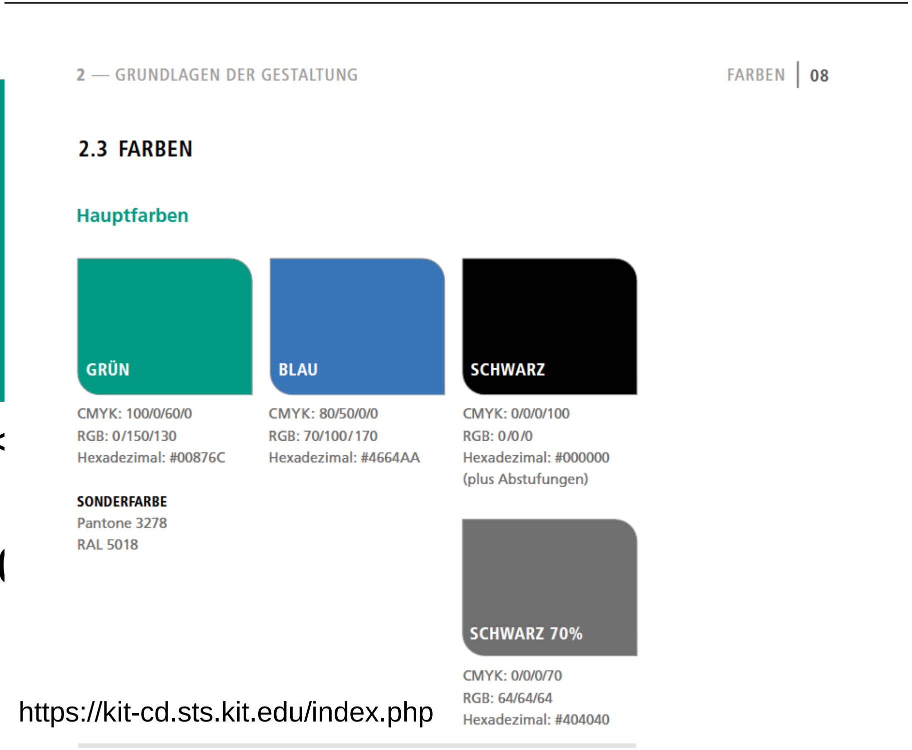
Color in HIS:
H: 257.5°, S: 0.415 , I: 136.7

Calculating Hue

- R:0, G: 150, B: 130

$B <$

$130 < 15$


2 — GRUNDLAGEN DER GESTALTUNG

FARBEN | 08

2.3 FARBEN

Hauptfarben

GRÜN

CMYK: 100/0/60/0
RGB: 0/150/130
Hexadezimal: #00876C

BLAU

CMYK: 80/50/0/0
RGB: 70/100/170
Hexadezimal: #4664AA

SCHWARZ

CMYK: 0/0/0/100
RGB: 0/0/0
Hexadezimal: #000000
(plus Abstufungen)

SONDERFARBE
Pantone 3278
RAL 5018

SCHWARZ 70%

CMYK: 0/0/0/70
RGB: 64/64/64
Hexadezimal: #404040

https://kit-cd.sts.kit.edu/index.php

Calculating Hue

$$H = \theta$$

- $\theta = \cos^{-1}\left(\frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}\right)$
- $\theta = \cos^{-1}\left(\frac{-150-130}{2\sqrt{(-150)^2 + (0-130)(150-130)}}\right)$
- $\theta = \cos^{-1}\left(\frac{-280}{2\sqrt{22500-2600}}\right)$
- $\theta \approx \cos^{-1}(-0.9924)$
- $\theta \approx 172.9°$
- $\rightarrow H = \theta \approx 172.9°$

$$H = \begin{cases} \theta, & \text{falls } B < G \\ 360 - \theta, & \text{sonst} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R + G + B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R + G + B)$$

R:0, G: 150, B: 130

Saturation und Intensity

- $I = \frac{R+G+B}{3}$
- $I = \frac{280}{3}$
- $I \approx 93.3$

- $S = 1 - \frac{3}{R+G+B} \min(R, G, B)$
- $S = 1$

$$H = \begin{cases} \theta, & \text{falls } B < G \\ 360 - \theta, & \text{sonst} \end{cases}$$

$$\theta = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B} \cdot \min(R, G, B)$$

$$I = \frac{1}{3}(R + G + B)$$

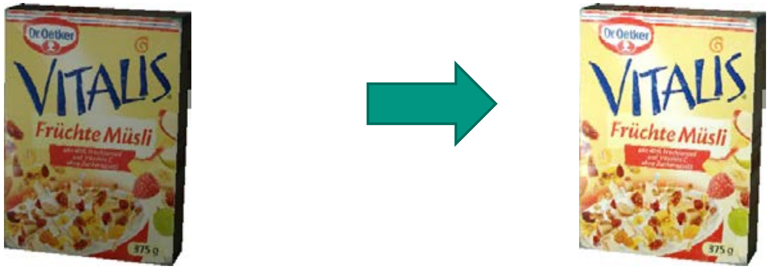
R:0, G: 150, B: 130

Color in HIS:
H: 257.5°, S: 1 , I: 93.3

Color Value Changes

A robot observes at a pack of cereal in a windowless laboratory. Someone turns the light a little brighter.

1. Do the R, G or B values of the cereal package change?
2. Do the H, S or I values of the cereal package change?



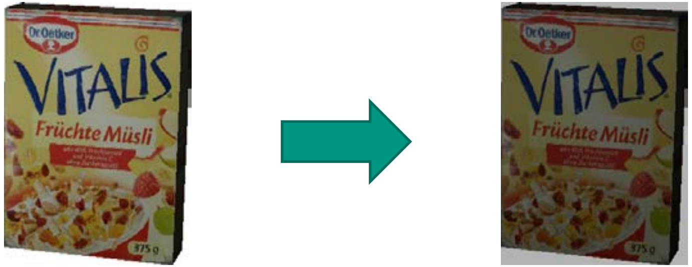
R, G and B change evenly, i.e. increase or decrease by the same factor.

H and S remain unchanged, I changes according to the change in brightness.

Color Value Changes

The robot observes the pack in a kitchen with windows. Outside, a cloud pushes in front of the sun.

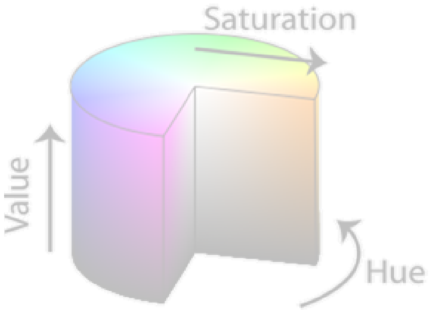
1. Do the R, G or B values of the cereal package change?
2. Do the H, S or I values of the cereal package change?

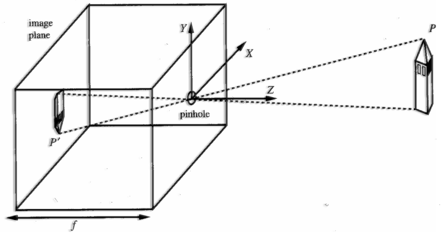


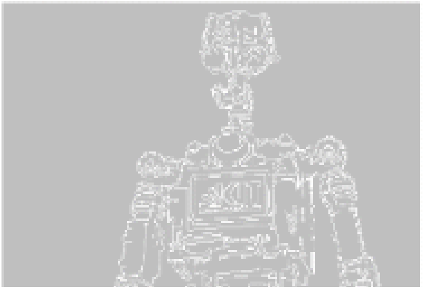
As not only the light intensity but also the color spectrum of the light changes here, all values, including H and S, change - although in most cases significantly less than the RGB values.

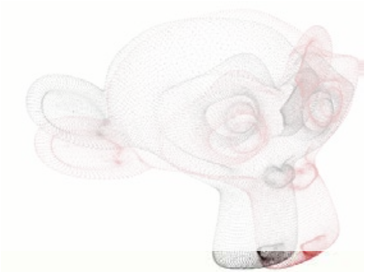
Inhalt

- Task 1
  Color Representation
- **Task 2**
  Camera Model
- Task 3
  Filters in Image Processing
- Task 4
  Segmentation
- Task 5
  Iterative Closest Point (ICP) Algorithm









Task 2

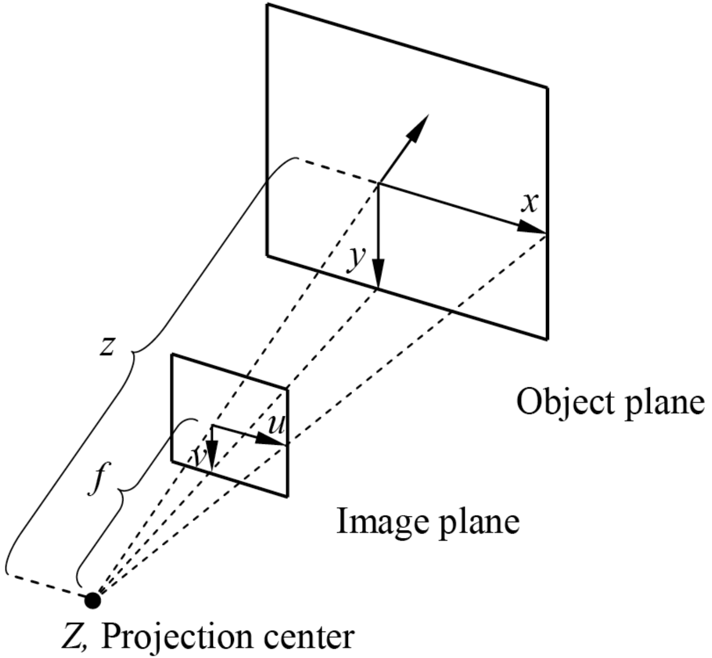
- Given a pinhole camera in positive position with focal length $f = 20\ \mathrm{mm}$
  - You use it to photograph a building from a distance of $350\ \mathrm{m}$. In the picture, the building is $0.8\ \mathrm{mm}$ high. How high is the actual building?
  - You are taking a photo of Cologne Cathedral, which is known to be $100\frac{\pi}{2}\ \mathrm{m}$ high, from the opposite bank of the Rhine from a distance of $800\ \mathrm{m}$. In your photo, the cathedral is 314 pixels high. How many pixels per millimeter does the camera have?

Camera Model

- Given a pinhole camera in positive position with focal length $f = 20\ \mathrm{mm}$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z} \begin{pmatrix} x \\ y \end{pmatrix}$$

Lecture Chapter 9 Slide 36



Camera Model

You use it to photograph a building from a distance of $350\ \mathrm{m}$. In the image, the building is $0.8\ \mathrm{mm}$ high. How high is the actual building?

$$\binom{u}{v} = \frac{f}{z}\binom{x}{y}$$

- $v = \frac{f}{z} y$
- $y = v \frac{z}{f}$
- $y = 0.8\ \mathrm{mm} \cdot \frac{350\ \mathrm{m}}{20\ \mathrm{mm}}$
- $y = 0.8 \cdot \frac{350}{20}\ \mathrm{m}$
- $y = 0.4 \cdot 35\ \mathrm{m} = 14\ \mathrm{m}$

Camera Model

You are taking a photo of Cologne Cathedral, which is known to be $100\frac{\pi}{2}$ m high, from the opposite bank of the Rhine from a distance of 800 m. In your photo, the cathedral is 314 pixels high. How many pixels per millimeter does the camera have?

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z}\begin{pmatrix} x \\ y \end{pmatrix}$$

Camera Model

You are taking a photo of Cologne Cathedral, which is known to be $100\frac{\pi}{2}$ m high, from the opposite bank of the Rhine from a distance of 800 m. In your photo, the cathedral is 314 pixels high. How many pixels per millimeter does the camera have?

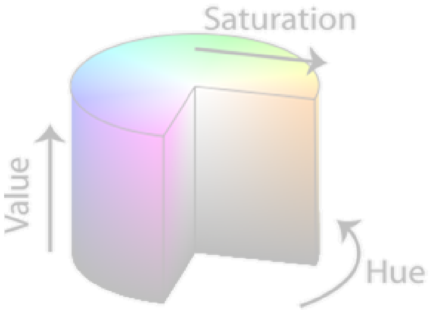
- $y = 100 \cdot \frac{\pi}{2}\ m \approx 157\ m$, $z = 800\ m$, v corresponds to 314 px
- $v = \frac{f}{z} y = \frac{20 \text{ mm}}{800 \text{ m}} 100 \cdot \frac{\pi}{2} m = \frac{100\ \pi}{80}\ mm$
- $314 \text{ px} \triangleq \frac{314}{80}\ mm$
- $1 \text{ px} \triangleq \frac{1}{80}\ mm$
- $80 \text{ px} \triangleq 1 \text{ mm}$
- The camera has 80 $\frac{px}{mm}$

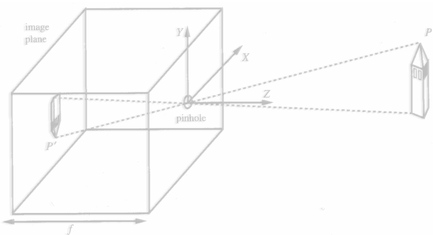
$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z} \begin{pmatrix} x \\ y \end{pmatrix}$$

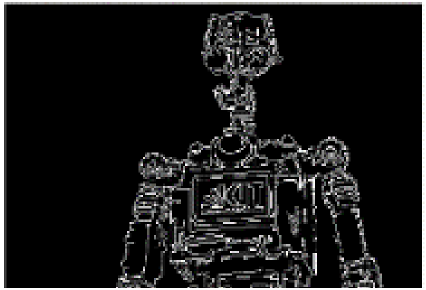
- As a camera spec this is also often denoted as inverse value:
  - Pixel pitch $\frac{1}{80px/mm} = 12{,}5\mu m$

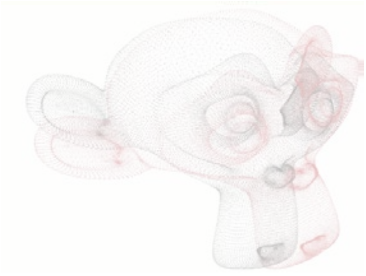
Content

- **Task 1**
  Color Representation
- **Task 2**
  Camera Model
- **Task 3**
  Filters in Image Processing
- **Task 4**
  Segmentation
- **Task 5**
  Iterative Closest Point (ICP) Algorithm





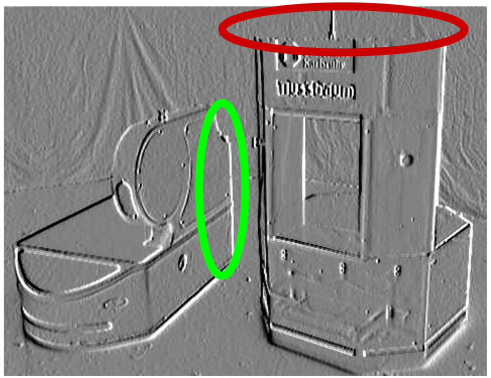


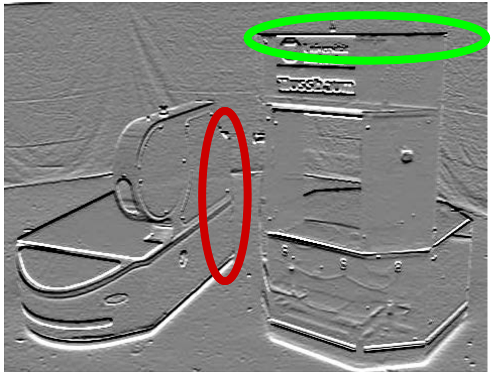


Filtering – Prewitt

Lecture Chapter 9 Slide 83

- **Prewitt-X Filter** $P_x = \frac{\partial g(x,y)}{\partial x}$
  - Approximated by $p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$
- **Prewitt-Y Filter** $P_y = \frac{\partial g(x,y)}{\partial y}$
  - Approximated by $p_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$
- **Properties**:
  - Good results for the detections of vertical or horizontal edges





Prewitt Filter

Given the following gray-scale image B

$$B = \begin{pmatrix} 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 20 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 20 & 20 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \end{pmatrix}$$

- Calculate the result of filtering B with a Prewitt-X Filter
- Calculate the result of filtering B with a Prewitt-Y Filter

Application of a Filter

$w(x,y)$

-1	0	1
-2	0	2
-1	0	1

$*$

$f(x,y)$

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y

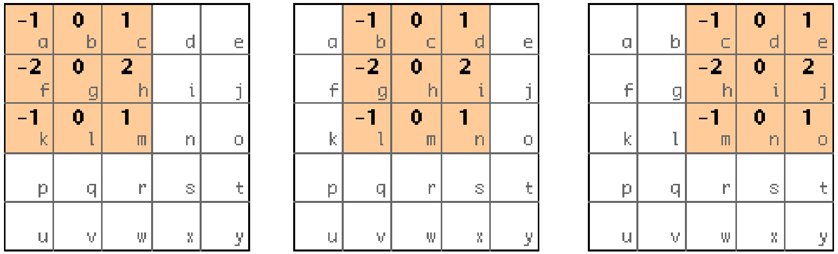
$=$

$g(x,y)$

	g'	h'	i'	
	l'	m'	n'	
	q'	r'	s'	

$$m' = -g + i - 2l + 2n - q + s$$

Filter as correlation



g' = -a+c-2f+2h-k+m

h' = (d+2i+n)-(b+2g+l)

i' = (e+2j+o)-(c+2h+m)

Prewitt Filter

$$B = \begin{pmatrix} 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 20 & 20 & 20 & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 20 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\ 20 & 20 & 20 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \end{pmatrix}$$

- Prewitt Edge Detektor in x-direction:

$$p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow B_x = \begin{pmatrix} \; & \; & \; & \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; & \; & \; & \; \end{pmatrix}$$

Prewitt Filter

- $B_x(2,2) = -1\,B(1{,}1) + 0\,B(1{,}2) + 1\,B(1{,}3) - 1B(2{,}1) + 0\,B(2{,}2)$
  $+1\,B(2,3) - 1\,B(3,1) + 0\,B(3{,}2) + 1\,B(3{,}3)$

  $= -20 + 0 + 20 - 20 + 0 + 20 - 20 + 0 + 20$
  $= 0$

- $B_x(2,3) = -20 + 0 + 20 - 20 + 0 + 20 - 20 + 0 + 20 = 0$

- $B_x(2,4) = -20 + 0 + 40 - 20 + 0 + 40 - 20 + 0 + 40 = 60$

Prewitt Filter

Karlsruher Institut für Technologie

$$B = \begin{pmatrix}
20_{*(-1)} & 20_{*0} & 20_{*1} & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\
20_{*(-1)} & 20_{*0} & 20_{*1} & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\
20_{*(-1)} & 20_{*0} & 20_{*1} & 20 & 40 & 40 & 40 & 40 & 30 & 30 & 30 \\
50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\
20 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\
20 & 20 & 50 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20 \\
20 & 20 & 20 & 50 & 50 & 50 & 50 & 50 & 20 & 20 & 20
\end{pmatrix}$$

- Prewitt Edge Detektor in x-direction:

$$p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow B_x = \begin{pmatrix}
0 & 0 & 60 & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{pmatrix}$$

H2T

Prewitt Filter

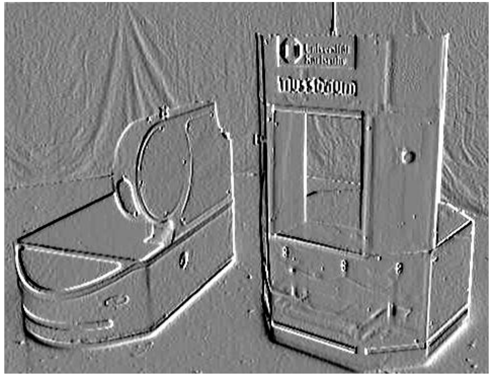
- $B_x = \begin{pmatrix} 0 & 0 & 60 & 60 & 0 & 0 & -30 & -30 & 0 \\ 0 & 0 & 40 & 40 & 0 & 0 & -50 & -50 & 0 \\ 30 & 0 & 20 & 20 & 0 & 0 & -70 & -70 & 0 \\ 60 & 30 & 0 & 0 & 0 & 0 & -90 & -90 & 0 \\ 60 & 60 & 30 & 0 & 0 & 0 & -90 & -90 & 0 \end{pmatrix}$
- $B_y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 90 & 90 & 70 & 50 & 30 & 30 & 10 & -10 & -30 \\ 60 & 90 & 70 & 50 & 30 & 30 & 10 & -10 & -30 \\ -60 & -30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -60 & -60 & -30 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Prewitt Filter

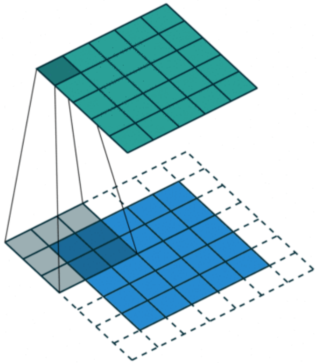
- $B_x = \begin{pmatrix} 0 & 0 & 60 & 60 & 0 & 0 & -30 & -30 & 0 \\ 0 & 0 & 40 & 40 & 0 & 0 & -50 & -50 & 0 \\ 30 & 0 & 20 & 20 & 0 & 0 & -70 & -70 & 0 \\ 60 & 30 & 0 & 0 & 0 & 0 & -90 & -90 & 0 \\ 60 & 60 & 30 & 0 & 0 & 0 & -90 & -90 & 0 \end{pmatrix}$

- Extra Questions:
  - Which Color is *0*?
  - Which Color is *-90*?

Not a regular color space but a representation of the edges!
But can be transformed in an image



Filter Operations – Image Egdes



How can I keep the resulting image the same size?
→Padding

- **Constant** value (e.g. 0):
  Image is extended
  by value 0 pixels

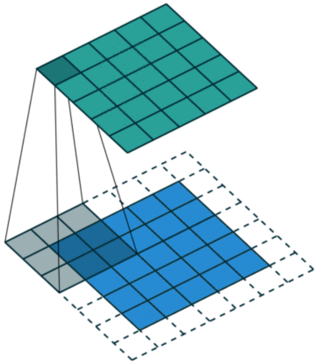
- **Wrap**:
  Image is repeated

Constant

Wrap

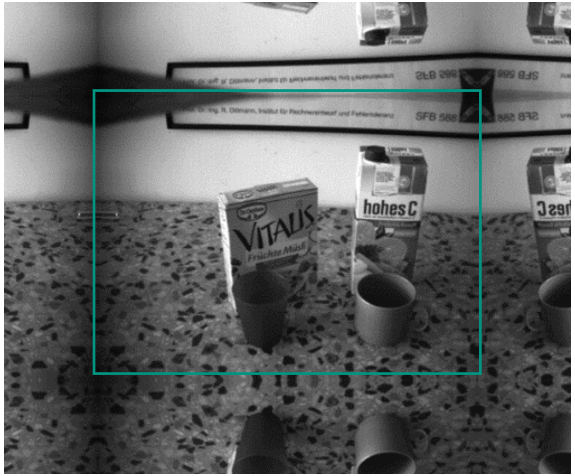
Filter Operations – Image Egdes

How can I keep the resulting image the same size?
→Padding



- **Mirror**, **Reflect**:
  Image is mirrored at the edges

- **Clamp**, **Replicate**:
  Take the last value



Mirror

Clamp

Lecture Chapter 9 Slide 65

Filter Operations – Image Egdes

Padding

- Example: Mean filter using:

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Input image

$$\begin{pmatrix} 30 & 60 & 90 \\ 120 & 0 & 150 \\ 180 & 210 & 240 \end{pmatrix}$$

Filter Operations – Image Egdes

Constant:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 60 & 90 & 0 \\ 0 & 120 & 0 & 150 & 0 \\ 0 & 180 & 210 & 240 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Result:

$$\frac{1}{9}\begin{pmatrix} 210 & 450 & 300 \\ 600 & 1080 & 750 \\ 510 & 900 & 600 \end{pmatrix}$$

Filter Operations – Image Egdes

Wrap:

$$\begin{pmatrix} 240 & 180 & 210 & 240 & 180 \\ 90 & 30 & 60 & 90 & 30 \\ 150 & 120 & 0 & 150 & 120 \\ 240 & 180 & 210 & 240 & 180 \\ 90 & 30 & 60 & 90 & 30 \end{pmatrix}$$

Result:

$$\frac{1}{9}\begin{pmatrix} 1080 & 1080 & 1080 \\ 1080 & 1080 & 1080 \\ 1080 & 1080 & 1080 \end{pmatrix}$$

Filter Operations – Image Egdes

Mirror/Clamp:

$$\begin{pmatrix} 30 & 30 & 60 & 90 & 90 \\ 30 & 30 & 60 & 90 & 90 \\ 120 & 120 & 0 & 150 & 150 \\ 180 & 180 & 210 & 240 & 240 \\ 180 & 180 & 210 & 240 & 240 \end{pmatrix}$$

Result:

$$\frac{1}{9}\begin{pmatrix} 480 & 630 & 780 \\ 930 & 1080 & 1230 \\ 1380 & 1530 & 1680 \end{pmatrix}$$

Filter Operations – Image Egdes

No padding:

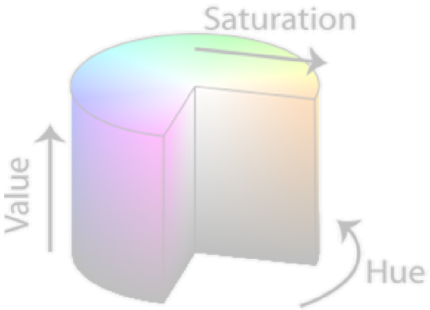
$$\begin{pmatrix} 30 & 60 & 90 \\ 120 & 0 & 150 \\ 180 & 210 & 240 \end{pmatrix}$$

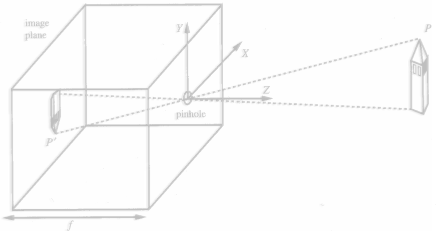
Result:

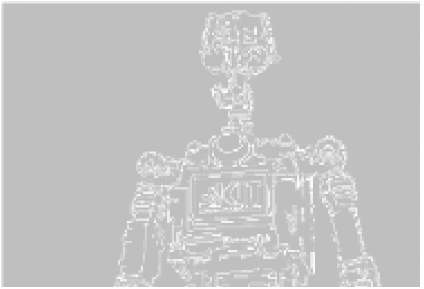
$$\frac{1}{9}\begin{pmatrix} X & X & X \\ X & 1080 & X \\ X & X & X \end{pmatrix}$$

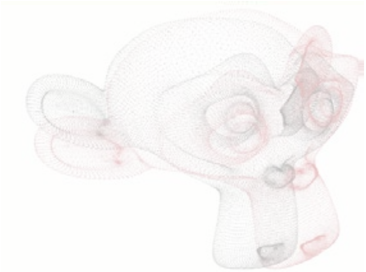
Content

- **Task 1**
  Color Representation
- **Task 2**
  Camera Model
- **Task 3**
  Filter in der Image Processing
- **Task 4**
  Segmentation
- **Task 5**
  Iterative Closest Point (ICP) Algorithm









Threshold Segmentation

In the image in Figure 2 an object should be detected. The image has already been converted to a greyscale image and the intensity values (0 ... 255) are denoted on the pixels. For segmentation, threshold segmentation is applied with a threshold of $T = 51$.

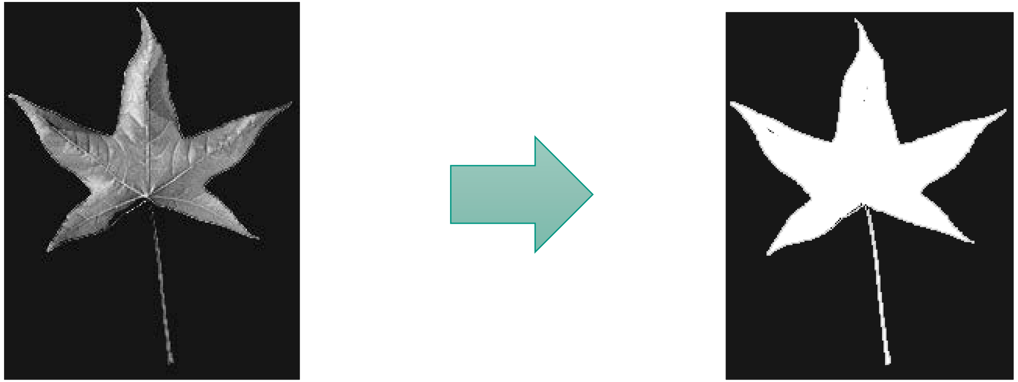
255	212	74	181	176	176	196	171	156
181	219	23	135	163	56	46	69	56
186	148	79	186	230	163	64	54	84
166	237	69	179	166	69	120	112	245
48	194	194	179	107	117	99	87	43
222	5	186	163	115	77	105	46	71
207	186	10	13	41	48	43	54	201
189	196	89	64	128	71	89	74	54

255	212	74	181	176	176	196	171	156
181	219	23	135	163	56	46	69	56
186	148	79	186	230	163	64	54	84
166	237	69	179	166	69	120	112	245
48	194	194	179	107	117	99	87	43
222	5	186	163	115	77	105	46	71
207	186	10	13	41	48	43	54	201
189	196	89	64	128	71	89	74	54

Segmentation – Threshold Filtering

- Threshold filtering to convert a grayscale image into a binary image
- Intensity of each pixel (u,v) is compared to a predefined threshold T

$$Img'(u,v) = \begin{cases} 255, & \text{if } Img(u,v) > T \\ 0, & \text{else} \end{cases}$$



Threshold Segmentation

In the image in Figure 2 an object should be detected. The image has already been converted to a greyscale image and the intensity values (0 ... 255) are denoted on the pixels. For segmentation, threshold segmentation is applied with a threshold of $T = 51$.

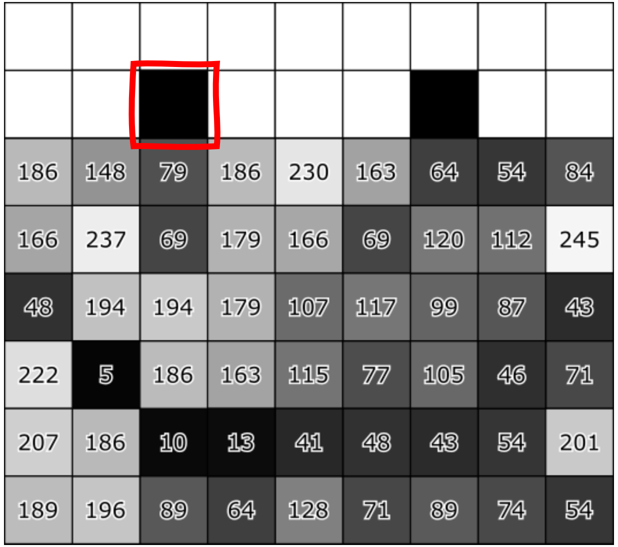
181	219	23	135	163	56	46	69	56
186	148	79	186	230	163	64	54	84
166	237	69	179	166	69	120	112	245
48	194	194	179	107	117	99	87	43
222	5	186	163	115	77	105	46	71
207	186	10	13	41	48	43	54	201
189	196	89	64	128	71	89	74	54

$$Img'(u,v) = \begin{cases} 255, & \text{if } Img(u,v) > T \\ 0, & \text{else} \end{cases}$$

- $Img(1,1) = 255 \rightarrow Img(1,1) = 255$
- $Img(1,2) = 212 \rightarrow Img(1,2) = 255$
- $Img(1,3) = 74 \rightarrow Img(1,3) = 255$
- ...

Threshold Segmentation

In the image in Figure 2 an object should be detected. The image has already been converted to a greyscale image and the intensity values (0 ... 255) are denoted on the pixels. For segmentation, threshold segmentation is applied with a threshold of $T = 51$.

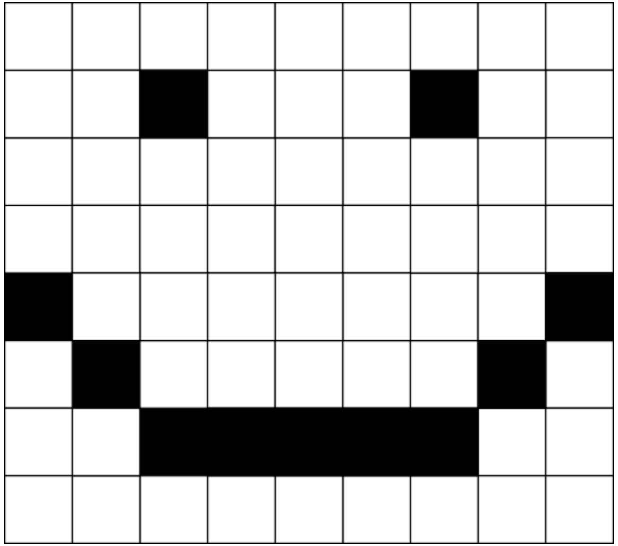


$$Img'(u,v) = \begin{cases} 255, & \text{if } Img(u,v) > T \\ 0, & \text{else} \end{cases}$$

- $Img(1{,}1) = 255 \rightarrow Img(1{,}1) = 255$
- $Img(1{,}2) = 212 \rightarrow Img(1{,}2) = 255$
- $Img(1{,}3) = 74 \rightarrow Img(1{,}3) = 255$
- ...
- $Img(2{,}1) = 181 \rightarrow Img(2{,}1) = 255$
- $Img(2{,}2) = 219 \rightarrow Img(2{,}2) = 255$
- $Img(2{,}3) = 23 \rightarrow Img(2{,}3) = 0$

Threshold Segmentation

In the image in Figure 2 an object should be detected. The image has already been converted to a greyscale image and the intensity values (0 ... 255) are denoted on the pixels. For segmentation, threshold segmentation is applied with a threshold of $T = 51$.



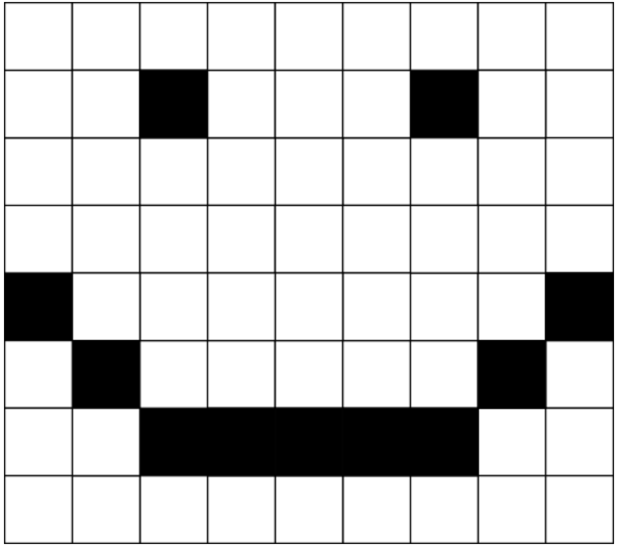
$$Img'(u,v) = \begin{cases} 255, & \text{if } Img(u,v) > T \\ 0, & \text{else} \end{cases}$$

- $Img(1{,}1) = 255 \rightarrow Img(1{,}1) = 255$
- $Img(1{,}2) = 212 \rightarrow Img(1{,}2) = 255$
- $Img(1{,}3) = 74 \rightarrow Img(1{,}3) = 255$
- ...
- $Img(2{,}1) = 181 \rightarrow Img(2{,}1) = 255$
- $Img(2{,}2) = 219 \rightarrow Img(2{,}2) = 255$
- $Img(2{,}3) = 23 \rightarrow Img(2{,}3) = 0$

Extra Question

As per our definition, what is the object that was segmented in the image?

- Emoji?
- Background?



Depends on the definition of segmentation

Here: 255, if $Img(u,v) \boxed{>} T$

→ Background was segmented

Masking

How can the color information of the segmented object be preserved after segmentation.

- Use the segmented image as mask for the original image.
  - Use the original image value for all segmented as 255

255	212	74	181	176	176	196	171	156
181	219		135	163	56		69	56
186	148	79	186	230	163	64	54	84
166	237	69	179	166	69	120	112	245
	194	194	179	107	117	99	87	
222		186	163	115	77	105		71
207	186						54	201
189	196	89	64	128	71	89	74	54

Content

- **Task 1**

  Color Representation

- **Task 2**

  Camera Model

- **Task 3**

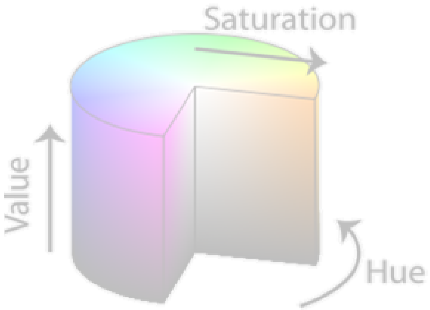
  Filter in der Image Processing

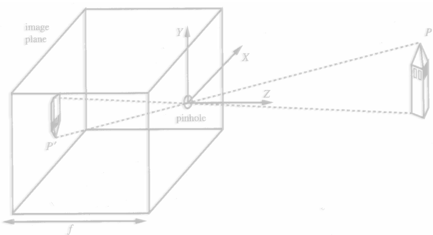
- **Task 4**

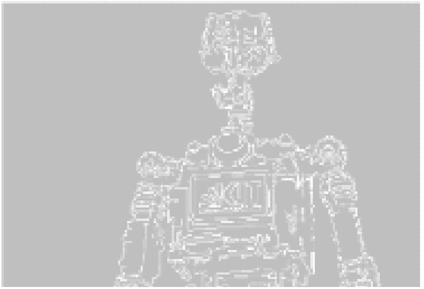
  Segmentation

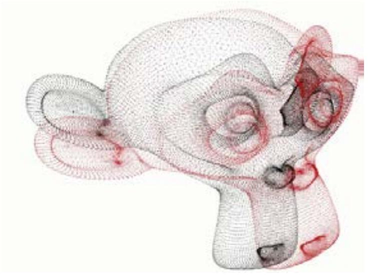
- **Task 5**

  Iterative Closest Point (ICP) Algorithm









Iterative Closest Point

- **Iterative Closest Point** (ICP) is a common algorithm for registering two sets A, B with a priori unknown assignment (Besl and McKay, 1992)

- **Example: Registration of two 3D point clouds**
  - For each iteration k:
    - For each point a_i from A, find point b_i from B that is closest to a_i
    - Calculate a transformation T_k such that D_k is minimal, e.g. with (Horn, 1987):

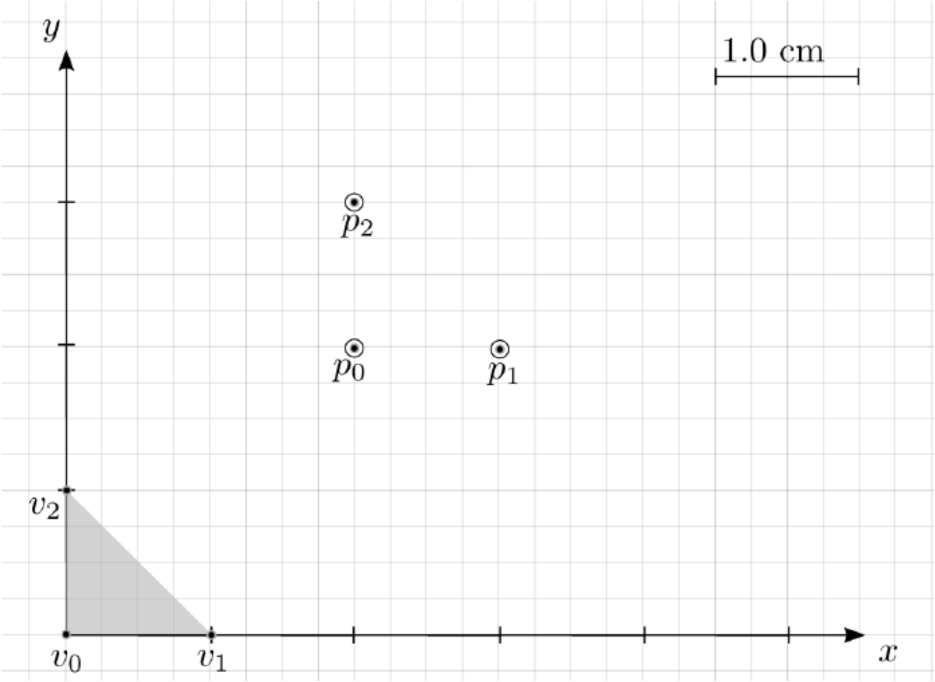
$$D_k = \sum_i \| a_i - T_k \cdot b_i \|^2$$

  - D_k combines translation and rotation
  - Apply transformation T_k to all points from B (update)
  - Termination:
    - Threshold for the difference $(D_{k-1} - D_k)$ or
    - Maximum number of iterations reached

Lecture Chapter 9 Slide 137

Task 5

Given: Triangle $V = (\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2})$ and point cloud $P = (\boldsymbol{p_0}, \boldsymbol{p_1}, \boldsymbol{p_2})$



Task 5

Given: Triangle $V = (\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2})$ and point cloud $P = (\boldsymbol{p_0}, \boldsymbol{p_1}, \boldsymbol{p_2})$

- Set up the error function F_T for the ICP algorithm

- To solve the problem the ICP-algorithm is to be applied using the steepest gradient approach. Determine the simplified error function:

$$F'_T = \|\boldsymbol{v_0} - \boldsymbol{p_0}\|^2$$

- For the ICP algorithm from the second part of the task, specify the function that approximates V to P with a step size α. Draw the first two iterations for $\alpha = 0.25$ in Figure 3

Error Function

- Set up the error function F_T for the ICP algorithm

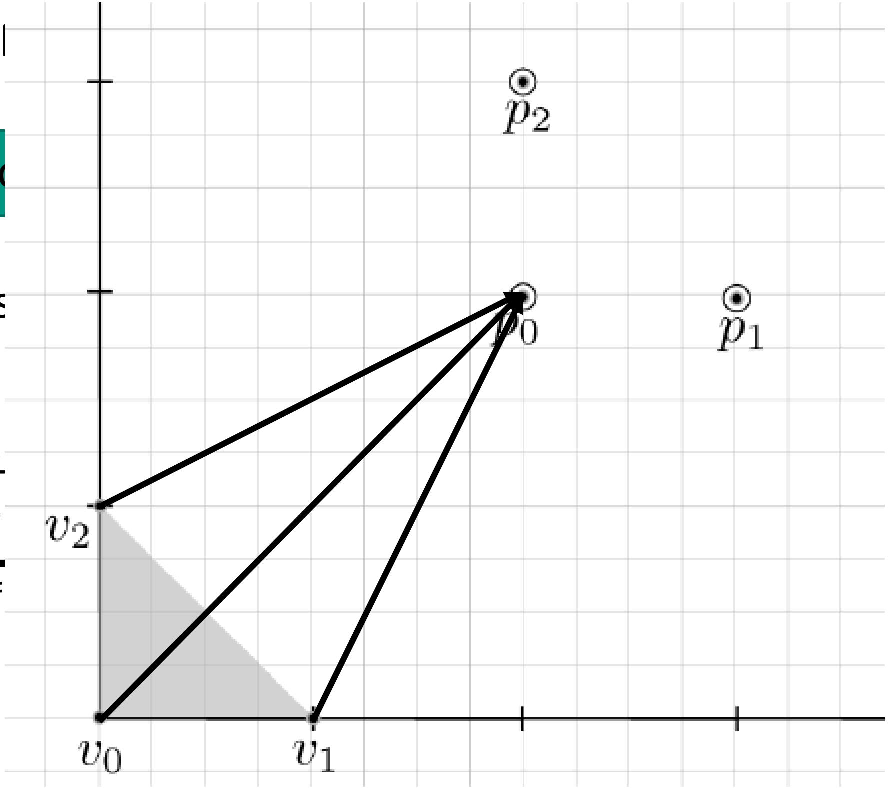
Error Function

- Set up the error function F_T for the ICI

For each point a_i from A, find point b_i fr

- The error function F_T with the corners
point cloud $P = (\boldsymbol{p_0}, \boldsymbol{p_1}, \boldsymbol{p_2})$ yields:

$$F_T(V) = \sum_{j=}^{2}$$



Gradient of F_t'

To solve the problem the ICP-algorithm is to be applied using the steepest gradient approach. Determine the simplified error function:

$$F_T' = \|\boldsymbol{v}_0 - \boldsymbol{p}_0\|^2$$

- Gradient of F_T'

Gradient of F_t'

To solve the problem the ICP-algorithm is to be applied using the steepest gradient approach. Determine the simplified error function:

$$F_T' = \|\boldsymbol{v_0} - \boldsymbol{p_0}\|^2$$

- Gradient of F_T'

$$F_T' = \left(\sqrt{\left(v_{0,x} - p_{0,x}\right)^2 + \left(v_{0,y} - p_{0,y}\right)^2}\right)^2 = \left(v_{0,x} - p_{0,x}\right)^2 + \left(v_{0,y} - p_{0,y}\right)^2$$

$$\nabla F_T' = \begin{pmatrix} \frac{d}{dx} F_T' \\ \frac{d}{dy} F_T' \end{pmatrix} = 2\begin{pmatrix} v_{0,x} - p_{0,x} \\ v_{0,y} - p_{0,y} \end{pmatrix} = 2(\boldsymbol{v_0} - \boldsymbol{p_0})$$

1. + 2. Iteration

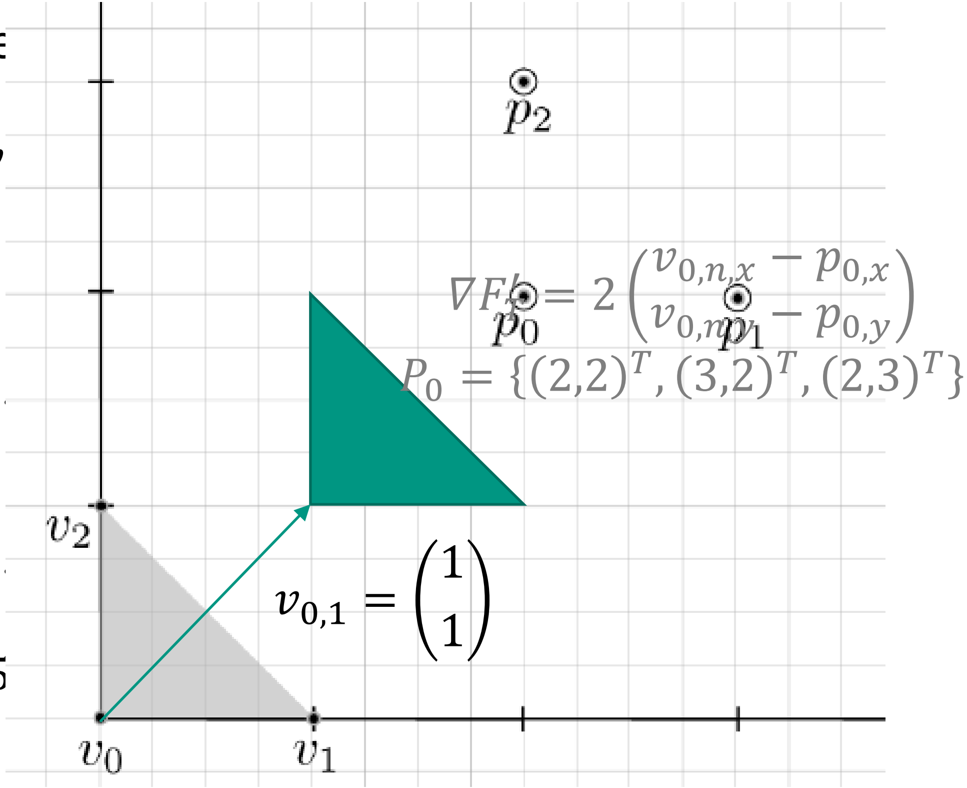
For the ICP algorithm from the second part of the task, specify the function that approximates V to P with a step size α. Draw the first two iterations for $\alpha = 0.25$ in Figure 3

1. + 2. Iteration

- Using the gradient $\nabla F_T'$ and step size can be derived:

$$v_{i,n} = v$$

- This allows to approximate P by:
  - $V_0 = \{(0,0)^T, (1,0)^T, (0,1)^T\}$
    - $v_{0,1} = v_{0,0} - \alpha \nabla F_T' = \binom{0}{0} - 0.25 \cdot 2$
  - $V_1 = \{(1,1)^T, (2,1)^T, (1,2)^T\}$
    - $v_{0,1} = v_{0,0} - \alpha \nabla F_T' = \binom{1}{1} - 0.25 \cdot 2$
  - $V_2 = \{(1.5,1.5)^T, (2.5, 1.5)^T, (1.5,2.5$



1. + 2. Iteration

Karlsruher Institut für Technologie

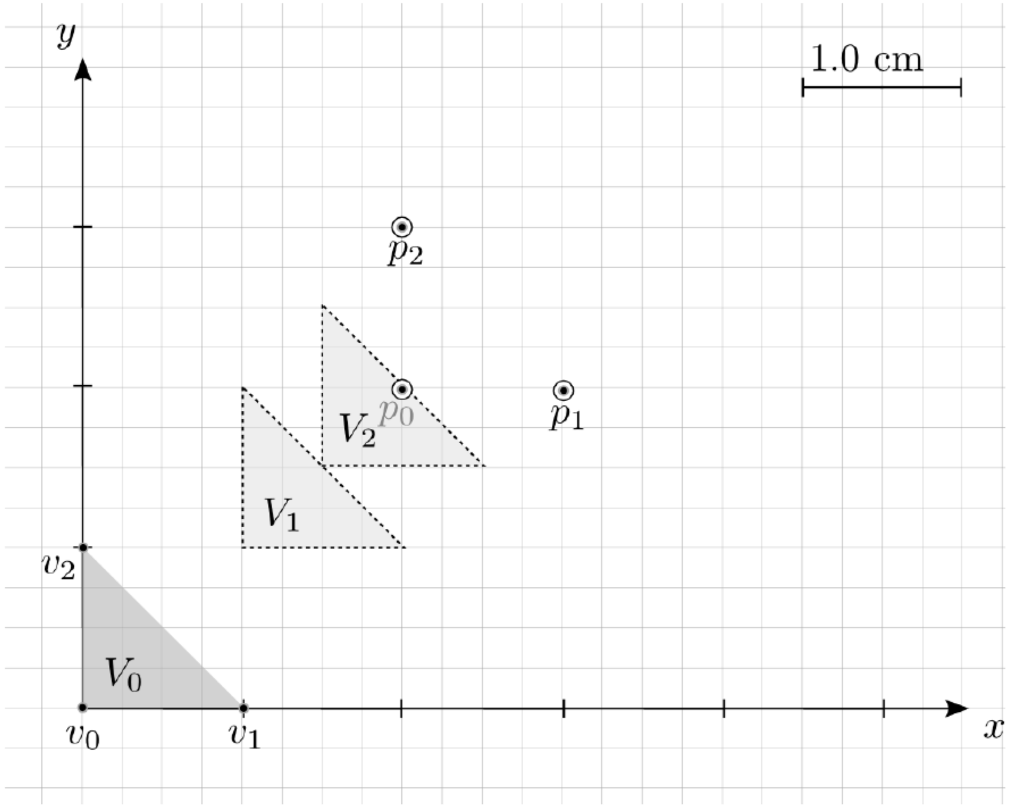
For the ICP algorithm from the second part of the task, specify the function that approximates V to P with a step size α. Draw the first two iterations for $\alpha = 0.25$ in Figure 3

- With the gradient $\nabla F_T'$ below step width $\alpha = 0.25$ the following update function can be derived:

$$\boldsymbol{v_{i,n}} = \boldsymbol{v_{i,n-1}} - \alpha \nabla F_T'$$

- This allows to approximate P as follows:
  - $V_0 = \{(0,0)^T, (1,0)^T, (0,1)^T\}$
  - $V_1 = \{(1,1)^T, (2,1)^T, (1,2)^T\}$
  - $V_2 = \{(1.5,1.5)^T, (2.5, 1.5)^T, (1.5,2.5)^T\}$

The first 2 Interations



Extra Question

- What would happen in this task if we used $\alpha = 1$?

a) Exact solution in a single step
b) Faster convergence but still asymptotically
c) Oscillation around of two solutions
d) Result diverging from solution

Extra Question

- What would happen in this task if we used $\alpha = 1$?

a) Exact solution in a single step

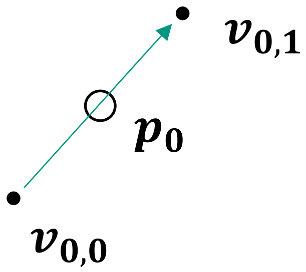
b) Faster convergence but still asymptotically

c) Oscillation around of two solutions

d) Result diverging from solution

$$\boldsymbol{v}_{\boldsymbol{i},\boldsymbol{n}} = \boldsymbol{v}_{\boldsymbol{i},\boldsymbol{n}-\mathbf{1}} - \alpha \nabla F_T'$$

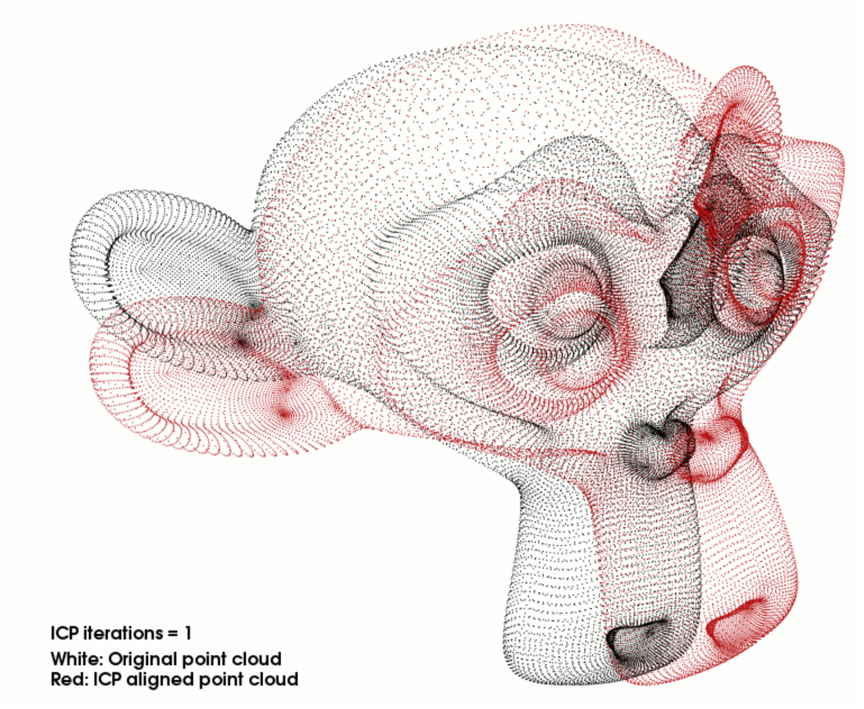
$$\nabla F_T' = 2(\boldsymbol{v}_{\mathbf{0}} - \boldsymbol{p}_{\mathbf{0}})$$



Interaktives Iterative Closest Point

- In jeder Iteration versucht ICP eine möglichst optimale Annährung von 2 Punktwolken zu finden

- Sehr abhängig von der Qualität und der Ähnlichkeit der Punktwolken, sowie den Startbedingungen



https://pcl.readthedocs.io/projects/tutorials/en/latest/interactive_icp.html

Exam

- Credit points: 6 ECTS

- Exam in winter term 2024/2025
  - **Written exam in English (schriftlich)**
  - Date: **February 24, 2025, 08:00 – 10:00**
    - **Place will be announced in the lecture and in ILIAS**
  - Registration: **Campus-System,** https://campus.studium.kit.edu
  - Last registration date: **February 19, 2025**

- All information regarding lectures and exams will also be published on our homepage: https://www.humanoids.kit.edu

Exam – 120 minutes

- Compared to previous semesters:
  More time for the exam, namely **120min** instead of 60min
- Expect, for example:
  - More **in-depth** tasks, requiring a **deeper understanding and to think more**
  - Longer tasks/calculations
  - More topics covered

- No additional material
  - Only writing utensils