

Real-Time Image Processing

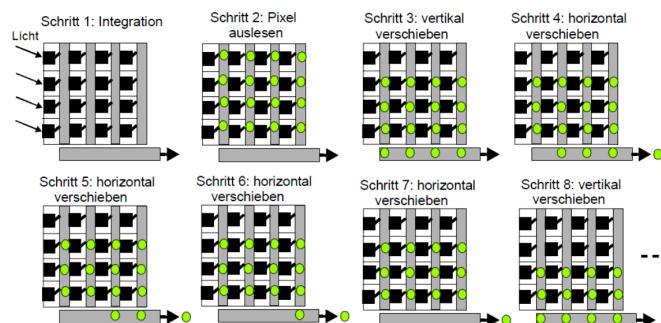
Zusammenfassung

Joel von Rotz / [Quelldateien](#)

Einführung

CCD Sensoren wird der photoelektrische Effekt genutzt. Dabei wird bei Wellenlänge 1100nm (IR) Elektronen-Lochpaare erzeugt. Die Anzahl erzeugten Elektronen entspricht der Intensität des einfallenden Lichtes und werden beim Lesen aus dem Sensor herausgeschoben → *Eimerkettenprinzip*

CMOS Sensoren Identisch zu CCD, ausser das jedes Pixel eine eigene Verstärker-Stufe besitzt. Dies erlaubt einen grösseren Dynamikbereich durch die Anpassung der individuellen Verstärker.



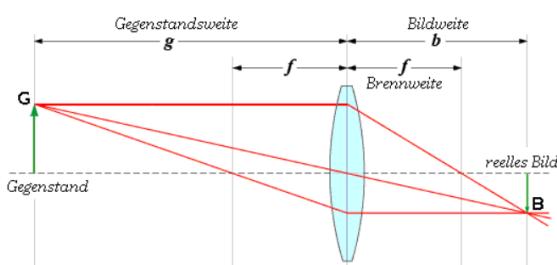
Quantisierung

Bild-Auflösung abhängig von Pixelanzahl des Sensors. Dynamikbereich/Grautiefe-Auflösung abhängig von Grenzen der A/D Wandler.

Reduzieren der Bild-Auflösung reduziert den Dynamik-Bereich nicht, sondern schneidet die obere und untere Tiefengrenze ab.

Rasterung

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f}$$



Globale Charakterisierung von Bildern

Mittelwert Varianz

$$\mu_I = \frac{1}{M \cdot N} \sum_{m,n} I_{m,n} \quad \sigma_I^2 = \frac{1}{M \cdot N} \sum_{m,n} (I_{m,n} - \mu_I)^2$$

Histogramm Das Histogramm gibt die absolute oder relative Häufigkeit $p_I(g)$ aller Grauwerte $g \in [0, 255]$ eines Bildes an:

$$0 \leq p_I(g) \leq 1 \quad \forall g \quad \rightarrow \sum_g p_I(g) = 1$$

Die kumulierte oder Summenhäufigkeit $h_I(g)$:

$$h_I(g) = \sum_{g' \leq g} p_I(g')$$

Punkt Operationen & Bildverknüpfungen

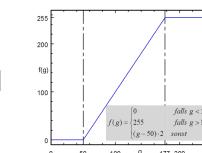
Look Up Tables LUTs werden verwendet, um Transformationen von diskreten Werten (Ganzzahlen) zu machen.

Lineare Grauwerttransformationen

Lassen sich anhand Notation schreiben $f : [0, 255] \rightarrow [0, 255] \subset \text{IR}$

Spreizung Um ein Grauwert-Intervall (z.B. [50,166]) auf das gesamte Intervall [0,255] zu verteilen

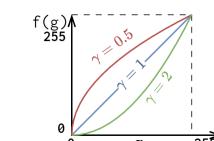
$$f(g) = \begin{cases} 0 & \text{falls } g < g_1 \\ 255 \cdot \frac{g - g_1}{g_2 - g_1} & \text{falls } g \in [g_1, g_2] \\ 255 & \text{falls } g > g_2 \end{cases}$$



Nichtlineare Operationen

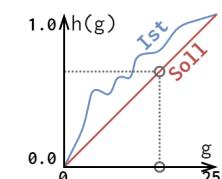
Gammakorrektur um menschliche Helligkeitswahrnehmung zu korrigieren.

$$f(g) = 255 \cdot \left[\frac{g}{255} \right]^\gamma$$



Histogrammausgleich Teilt die Wahrscheinlichkeit der Grauwerte gleichmässig auf

$$f(g) = g_{\max} \cdot \sum_{g'=0}^g p_I(g')$$



Arithmetische & Logische Bildverknüpfungen

Differenzbildung

Für Bewegungsdetektion!

$$\Delta I_{k+n} = \frac{1}{2} \cdot (255 + \overbrace{I_{k+n}}^{\text{neuer}} - \overbrace{I_k}^{\text{älter}}) \\ = \frac{1}{2} \cdot (255 + I((k+n) \cdot \Delta t) - I(k \cdot \Delta t))$$



Schwellwert

$$f(g) = \begin{cases} 255 & \text{falls } g < g_1 \vee g_2 < g \\ 0 & \text{falls } g \in [g_1, g_2] \end{cases}$$

mit $g_1 = 128 - \text{thres}$
 $g_2 = 128 + \text{thres}$



Hintergrundschätzung

$$B_k = \alpha \cdot B_{k-1} + (1 - \alpha) \cdot I_k \quad I_k \xrightarrow{+} B_k \quad \alpha \in (0, 1)$$

$$\rightarrow \Delta I_k = \frac{1}{2} \cdot (255 + B_k - I_k)$$

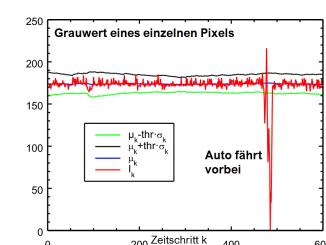
Statistische Analyse

Mittelwert μ_k & Varianz σ_k des Bildes

$$\mu_k = \alpha \cdot \mu_{k-1} + (1 - \alpha) \cdot I_k$$

$$\sigma_k = \alpha \cdot \sigma_{k-1} + (1 - \alpha) \cdots$$

$$(\mu_k - I_k)^T \cdot (\mu_k - I_k)$$



Filteroperatoren im Ortsraum

Lineare Filter

Faltung (convolution)

$$I \otimes h : I_{m,n} \rightarrow \sum_{p=-u}^u \sum_{q=-v}^v I_{m-p, n-q} \cdot h_{p,q}$$

Kommutativität: $I \otimes J = J \otimes I$

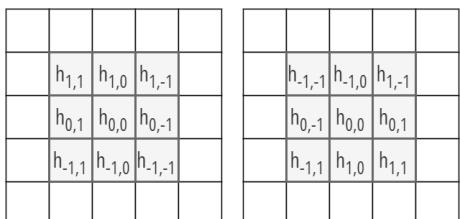
Assoziativität: $(I \otimes J) \otimes K = I \otimes (J \otimes K)$

Distributivität: $I \otimes (J + K) = I \otimes J + I \otimes K$

$$\text{Asso. mit Faktor: } a \cdot (I \otimes J) = \begin{cases} (a \cdot I) \otimes J \\ I \otimes (a \cdot J) \end{cases}$$

Damit die Indizierung verständlicher ist, wird anstatt Faltung die Korrelation (correlation) verwendet:

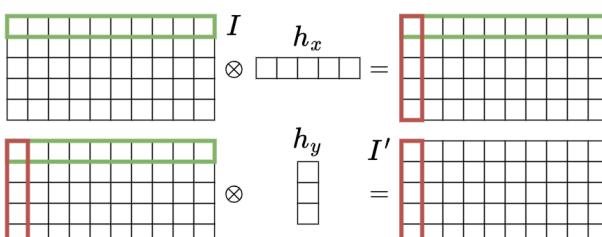
$$I \tilde{\otimes} h : I_{m,n} \rightarrow \sum_{p=-u}^u \sum_{q=-v}^v I_{m+p, n+q} \cdot h_{p,q}$$



Separierbare Masken können mit der Faltung zu einer gänzlichen Maske zusammengeführt werden.

$$\begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array} = \begin{array}{|c|c|} \hline & \\ \hline & \\ \hline \end{array} \otimes \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline \end{array}$$

Anhand der Assoziativität können zwei sukzessive Faltungen mit eindimensionalen Masken durchgeführt werden.



Tiefpass / Glätten

Mittelung eines Pixels mit den Nachbarpixeln ist die einfachste Rauschunterdrückung.

$$TP = \frac{1}{\sum \text{Mask}} \cdot \text{Mask}$$

$$\text{Rechteck} = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Gauss} = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Hochpass / Kantenhervorhebung

Via Ableitungen $\partial I / \partial x$ & $\partial I / \partial y$ können Kanten hervorgehoben werden. Folgend ist eine rauschanfällige Version...

$$h_x = \frac{1}{2} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad h_y = \frac{1}{2} \cdot \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

... welche folgend mit Prewitt und Sobel Filter verbessert wurde.

Prewitt-Filter gleichmässige Pixelgewichtung

$$h_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes [-1 \ 0 \ 1] = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \otimes [1 \ 1 \ 1] = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Sobel-Filter gewichtet die Pixel, die sich näher an der Mitte der Maske befinden, mehr.

$$h_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes [-1 \ 0 \ 1] = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \otimes [1 \ 2 \ 1] = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Bildschärfung

β Schärfungsparameter

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$I = \beta \cdot L$$

Nichtlineare Filter

Rangordnungsoperatoren

Pixel im Frame werden der Grösse nach geordnet und dann wird der Pixelwert mit einer Strategie bestimmt.

Median Filter

$$\begin{array}{|c|c|c|} \hline 5 & 3 & 4 \\ \hline 3 & 10 & 5 \\ \hline 3 & 4 & 5 \\ \hline \end{array} \xrightarrow{\text{Median Filter}} \begin{array}{|c|c|c|} \hline 3, 3, 3, 4, 4, \\ \hline 5, 5, 5, 10 \\ \hline 3 & 4 & 5 \\ \hline \end{array}$$

Min/Max Filter Mit dem gleichen Prinzip, einfach werden entsprechend der maximale oder minimale Wert genommen. Zusätzlich kann das **lokale** Maxima und Minima ermittelt werden.

$$\text{LocMin} = (I == \min_{R \times R}(I))$$

$$\text{LocMax} = (I == \max_{R \times R}(I))$$

Morphologische Operationen

Dilation \oplus Pixel setzen, wenn mindestens ein Pixel gesetzt ist. Ist Kommutativ $I \oplus J = J \oplus I$ und Assoziativ $(I \oplus J) \oplus K = I \oplus (J \oplus K)$.

$$I \oplus h = \{(m, n) \mid (\hat{h})_{m,n} \cap I \neq \emptyset\}$$

(Un)symmetrische Strukturelemente

\hat{h} bedeutet (Punkt-)Spiegelung. Bei symmetrischen Strukturelementen ist dies belanglos. Bei Unsymmetrischen muss für die Dilation (& Schließung) immer mit dem gespiegelten Strukturelement gearbeitet werden.

Erosion — Pixel setzen, wenn alle Pixel im Frame gesetzt sind. Ist weder kommutativ, noch assoziativ.

$$I - h = \{(m, n) \mid (h)_{m,n} \subset I\}$$

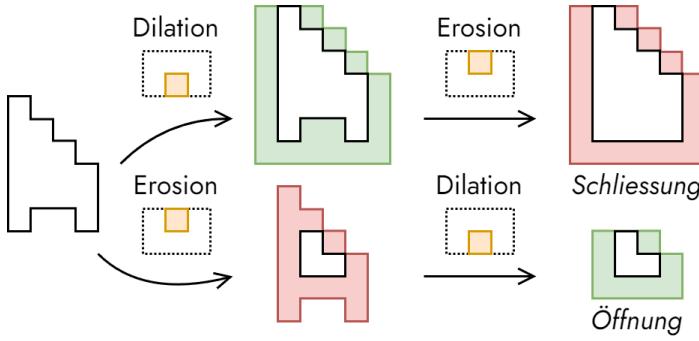
Kettenregel: $I - (h \oplus g) = (I - h) - g$

Schließung • Mehr vom Vordergrund zeigen. Ist idempotent ($I \bullet g$) $\bullet g = I \bullet g$

$$I \bullet h = (I \oplus h) - h$$

Öffnung o Mehr vom Hintergrund zeigen. Ist idempotent ($I \circ g$) $\circ g = I \circ g$

$$I \circ h = (I - h) \oplus h$$



$$I - g \subseteq I \circ g \subseteq I \subseteq I \bullet g \subseteq I \oplus g$$

Hit- & Miss Operation

Mit Hit- & Miss Operationen wird geprüft, wo die Nachbarschaftsumgebung eines Pixels genau der Struktur der Einsen und Nullen im Struk-Element entspricht. Das Strukturelement besteht gleichzeitig aus der Hit-Maske (mit Einsen ; h_1) und Miss-Maske (mit Nullen ; h_0).

$$I \pm h = (I - h_1^i) \cap (I^C - h_0)$$

I^C =Komplement aller Pixel ungleich Null im Bild I. Strukturelemente werden solange angewendet, bis keine Operation gemacht werden kann und dies für alle i Strukturelemente!

$$h^1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h^3 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad h^4 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \dots$$

Fouriertransformation

Die 2D Fourier Transformation ist eine Faltung von zwei 1D Fourier Transformationen

! Crash Kurs

$\text{rect}\left(\frac{t}{T}\right)$	$ T \cdot \text{si}(\pi T f)$	$ T \cdot \text{si}\left(\frac{T}{2} \omega\right)$
$\text{si}\left(\pi \frac{t}{T}\right)$	$ T \cdot \text{rect}(T f)$	$ T \cdot \text{rect}\left(\frac{T}{2\pi} \omega\right)$
$A\left(\frac{t}{T}\right)$	$ T \cdot \text{si}^2(\pi T f)$	$ T \cdot \text{si}^2\left(\frac{T}{2} \omega\right)$
$\text{si}^2\left(\pi \frac{t}{T}\right)$	$ T \cdot A(T f)$	$ T \cdot A\left(\frac{T}{2\pi} \omega\right)$

2D DFT $t \rightarrow f \Leftrightarrow f \rightarrow t$

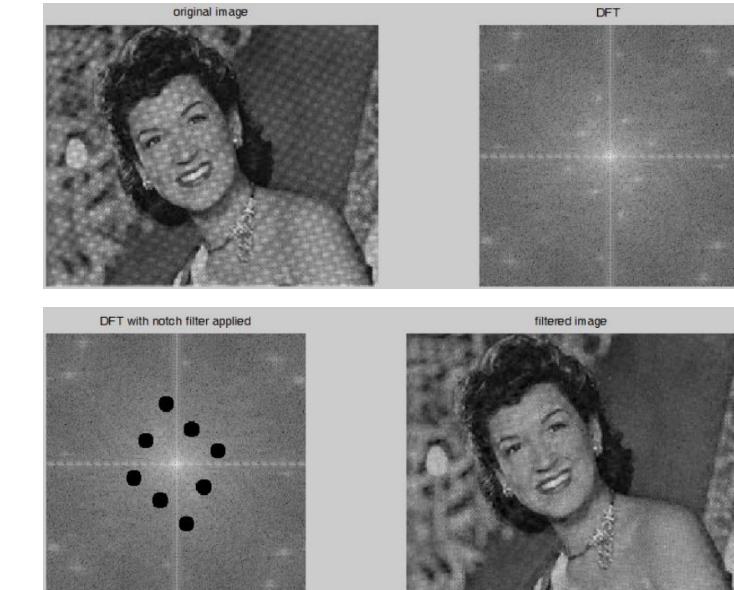
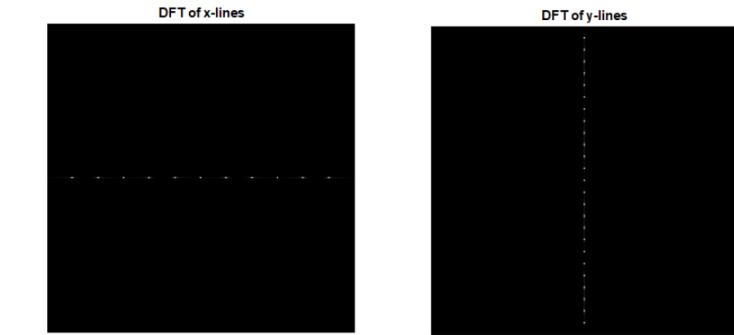
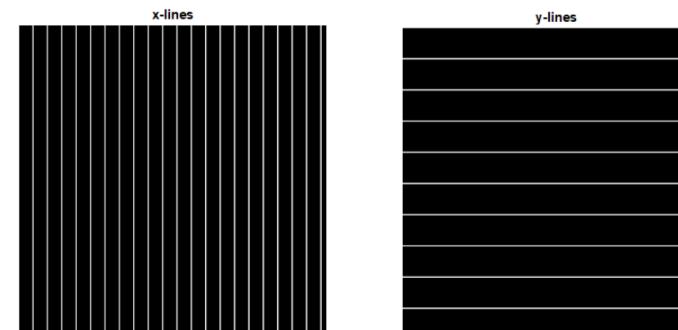
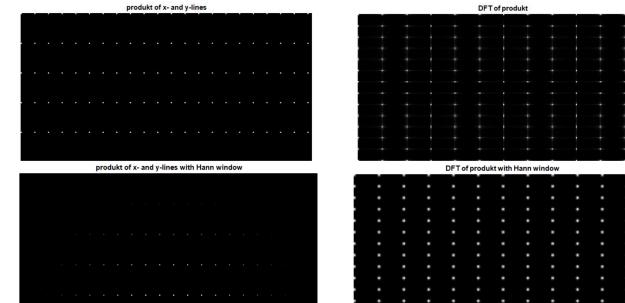
$$\hat{h}_{lk} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h_{mn} \cdot e^{-i \cdot 2\pi \cdot [\frac{l \cdot m}{M} + \frac{k \cdot n}{N}]} \\ \Leftrightarrow \\ h_{mn} = \frac{1}{M \cdot N} \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} \hat{h}_{lk} \cdot e^{i \cdot 2\pi \cdot [\frac{l \cdot m}{M} + \frac{k \cdot n}{N}]}$$

Parseval Theorem Wie in der 1D gilt auch das Parseval Theorem in 2D

$$M \cdot N \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h_{mn}^2 = \sum_{j=0}^{M-1} \sum_{l=0}^{N-1} |\hat{h}_{lk}|^2$$

Da die DFT für periodische Signale ausgelegt ist, können Artefakte entstehen. Mit einem **Hann**-Filter alle Pixel um die Bildmitte schwächen (im Bild Pixelwert $\rightarrow 0$)

$$w_n = \frac{1}{2} \cdot \left[1 - \cos\left(\frac{2\pi \cdot n}{N-1}\right) \right]$$



Segmentierung und Merkmalsextraktion —

Otsu / Automatische Schwellwertbestimmung

Wahrscheinlichkeit ω_n , Mittelwert μ_n , Varianz σ_n^2

Klasse C_0

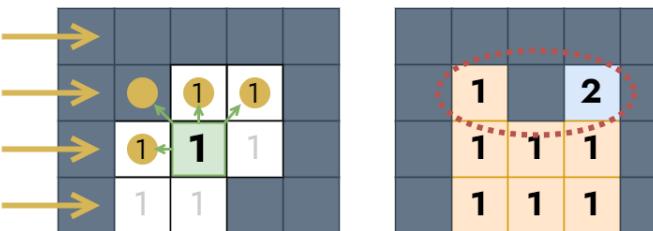
$$\omega_0 = \sum_{g=0}^K p_1(g) \quad \mu_0 = \frac{1}{\omega_0} \sum_{g=0}^K p_1(g) \cdot g$$

$$\sigma_0^2 = \frac{1}{\omega_0} \sum_{g=0}^K p_1(g) \cdot (g - \mu_0)^2$$

Klasse C1

$$\omega_1 = \sum_{g=K+1}^{255} p_1(g) \quad \mu_1 = \frac{1}{\omega_1} \sum_{g=K+1}^{255} p_1(g) \cdot g$$

$$\sigma_1^2 = \frac{1}{\omega_1} \sum_{g=K+1}^{255} p_1(g) \cdot (g - \mu_1)^2$$



ROIs müssen dem Pixel-Wertebereich des Bildes entsprechen → 1-Byte Grauwertbild erlaubt 255 ROIs / 2-Byte erlaubt $2^{16}-1$ ROIs.

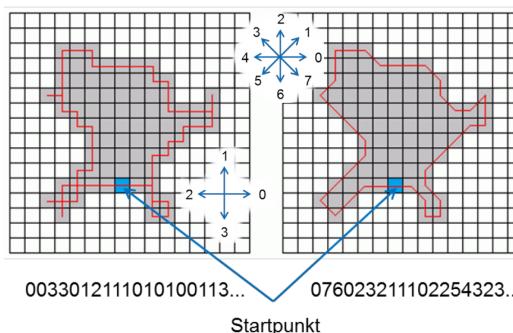
Region Labeling hat grossen Speicherbedarf → Kettencodes

Kontour - Kettencodes

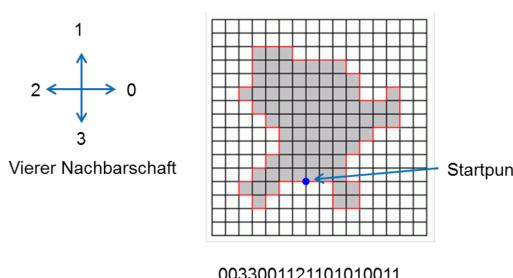
Es wird der **Startpunkt** gemerkt und die Richtungen (N -Bits → 2^N Richtungen)

Freeman Code

Mittelpunkt vom Pixel



Eine "korrekte" Beschreibung des Rands mit Crack Codes als Trennlinie zwischen Vorder- und Hintergrund.



Merkmalextraktor

Fläche A

$$\begin{aligned} A &= \sum_{I_m n \in ROI} 1 = - \underbrace{\oint_{\text{Rand}} y(x) dx}_{\text{mit Crack Codes}} \\ &= \sum_{I_m n \in \{2-\text{Seg.}\}} m - \underbrace{\sum_{I_m n \in \{0-\text{Seg.}\}} m}_{\text{Vertikale Richtungen haben kein Einfluss}} \end{aligned}$$

Massenmittelpunkt (x_s, y_s)

$$x_s = \frac{1}{A} \sum_{I_m n \in ROI} n \quad y_s = \frac{1}{A} \sum_{I_m n \in ROI} m$$

Orientierung M

$$M = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}$$

$$M_{yy} = \frac{1}{A} \sum_{I_m n \in ROI} (m^2 - y_s^2) \quad M_{xx} = \frac{1}{A} \sum_{I_m n \in ROI} (n^2 - x_s^2)$$

$$M_{xy} = M_{yx} = \frac{1}{A} \sum_{I_m n \in ROI} (m \cdot n - x_s \cdot y_s)$$

Bounding Box

ROI wird mit einem Rechteck umfasst.

Linien Segmentierung und Merkmalsextraktion

Kantendetektion

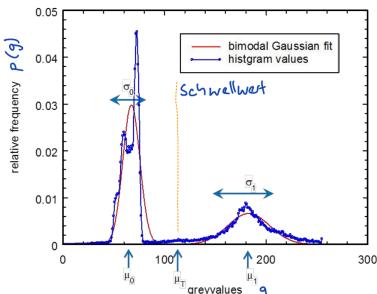
Problem: Kanten deutlich grösser als ein Pixel, globaler Schwellwert ergibt nicht in allen Bereichen des Bildes zufriedenstellende Kanten, zusammengehörige Kanten unterbrochen bei zu hohem Schwellwert.

Canny Kantendetektionsalgorithmus

① **Glättung** mittels Gaussfilter ② **Kantenfilter** mittels Sobel oder Prewitt ③ **Bestimmung der lokalen Maxima** entlang der Richtung der Gradienten (Pixel selektiert, wenn $|\text{Gradient}|$ grösser als Nachbarpixel) ④ **Kantenextraktion** anhand "oberen" Schwellwert (lok. Maxima über diesem wird es als Kante betrachtet) und sobald Pixel gefunden ist entlang der Kante mittels "unterem" Schwellwert zur Kante zugeordnet, durch Hysterese kann dynamisch auf Kontrast-schwankungen im Bild reagiert werden.

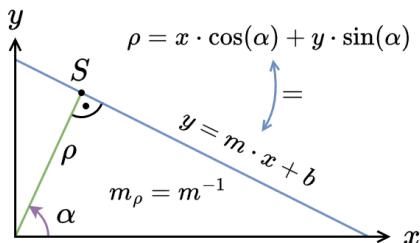
Linienextraktion mittels Hough-Transformation

Die Gradienten-Informationen (Betrag und Richtung) in neuen Raum transferieren und dort kann aufgrund von Ansammlungen auf die Formen im Originalbild geschlossen werden.

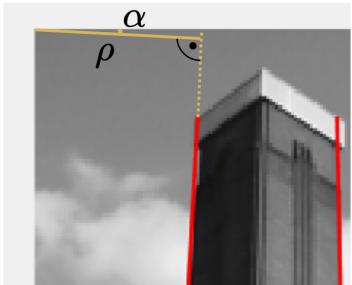


Region Labeling

Bitmap Bild wird von einer Seite durchgelabelt → Wird ein Bit erkannt, werden die Nachbarspixel geprüft. Ist bereits einer der Nachbarn in einer Regions of Interest (ROI), wird dieses Label übernommen, ansonsten wird eine neue Region kreiert.

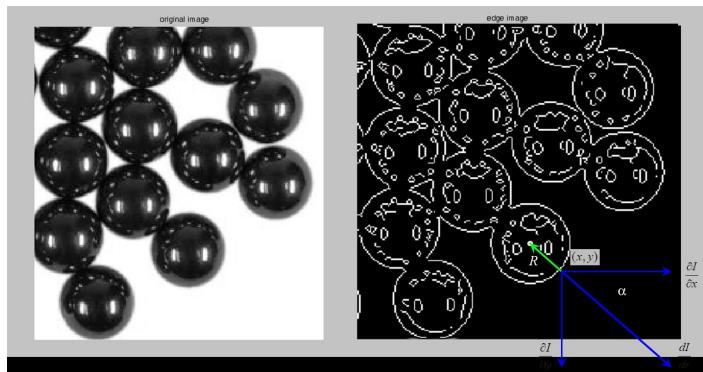


$$\begin{aligned}\alpha &\in [-90^\circ, 90^\circ] \\ \rho &\in [-D, D] \\ D &= \sqrt{\Delta x^2 + \Delta y^2}\end{aligned}$$



Detection of Circles via Hough

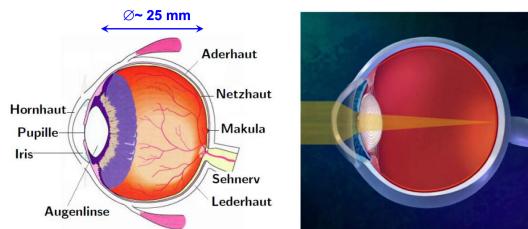
$$x_c = x - R \cdot \frac{\partial I}{\partial x} \cdot \left[\frac{dI}{dr} \right]^{-1} \quad y_c = y - R \cdot \frac{\partial I}{\partial y} \cdot \left[\frac{dI}{dr} \right]^{-1}$$



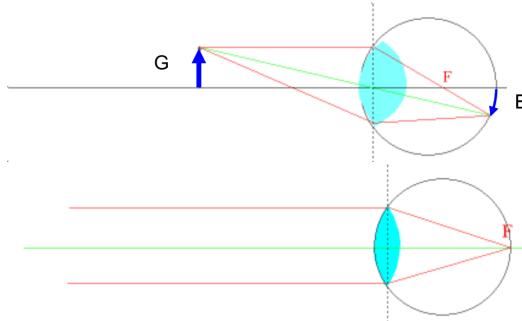
Farbe

Das Auge

Das Auge nimmt Helligkeit via Makula's Stäbchen wahr und über die Netzhaut sind **S**mall-, **M**- und **L**-Zapfen (445, 540, 565nm) verteilt im Verhältnis 1:5:10, welche als Farbfilter funktionieren.

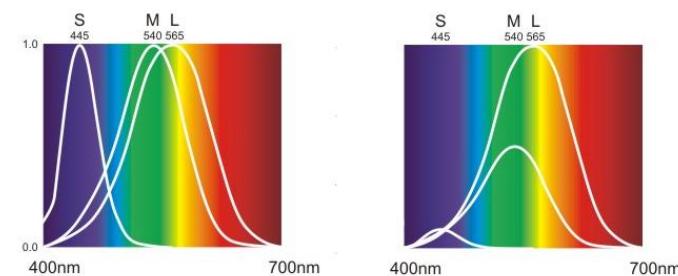


Die Linse des Auges besteht aus Hornhaut und Augenlinse. Ein paar Infos:

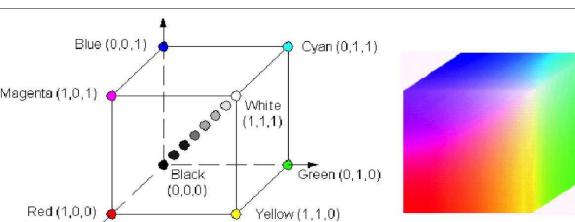


Hornhaut & Augenlinse \leftrightarrow Objektiv, Iris \leftrightarrow Blende, Netzhaut \leftrightarrow Diafilm/CCD, Akkommodation durch Änderung der Linsenkrümmung \leftrightarrow Akkommodation durch Verschieben des Objektives.

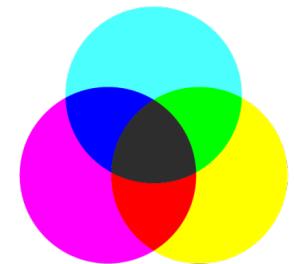
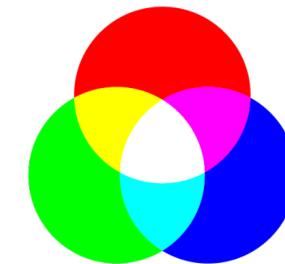
i Strahlengang (Bild von außen nach innen) ist beim Auge gleich wie bei der Kamera: auf dem Kopf.



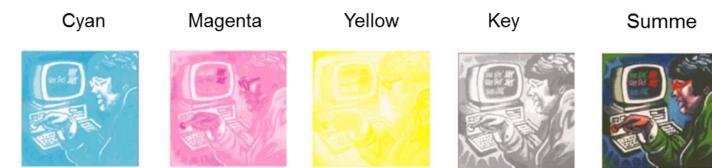
RGB (additiv) & CMY (subtraktiv)



Additive colour Subtractive colour

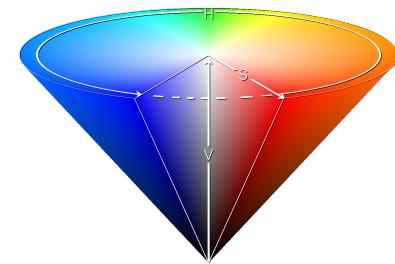


$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



HSV

Hue, Saturation, Visibility



YCbCr

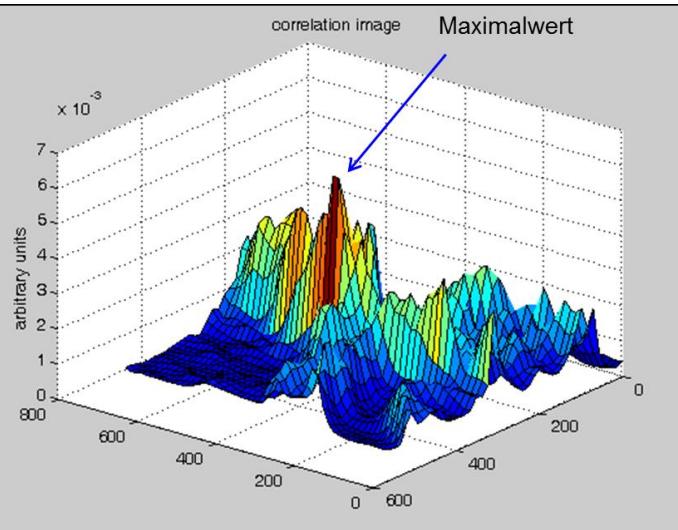
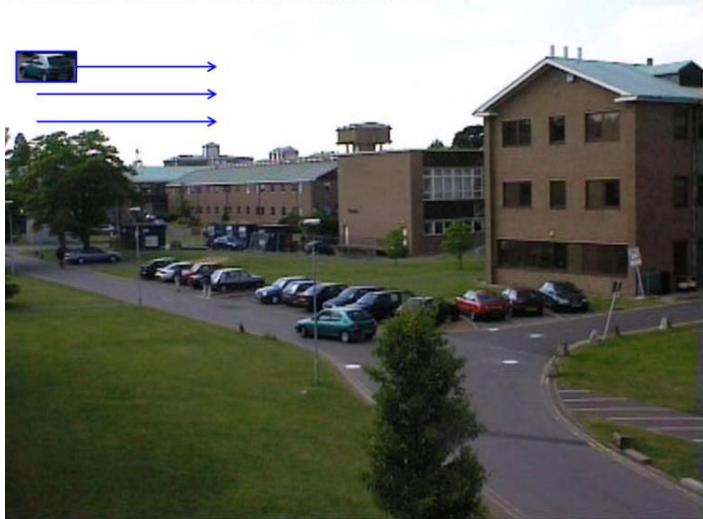
YCbCr separates the colors into Luminance- (Y ; Helligkeits- oder Grau-) and Chrominance- (Cb, Cr ; Farb-) values.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Mit $R, G, B \in [0, 1]$ und $16 \leq Y \leq 235$, $16 \leq Cb, Cr \leq 240$

Farbbasierte Objekterkennung

Mit Korrelation können Objekte in einem Bild gefunden werden.

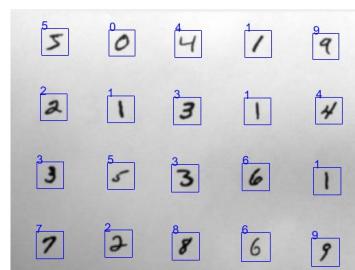
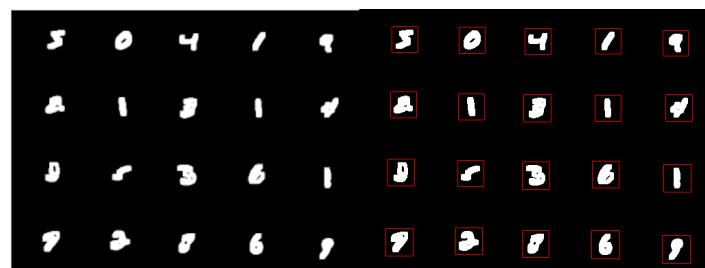
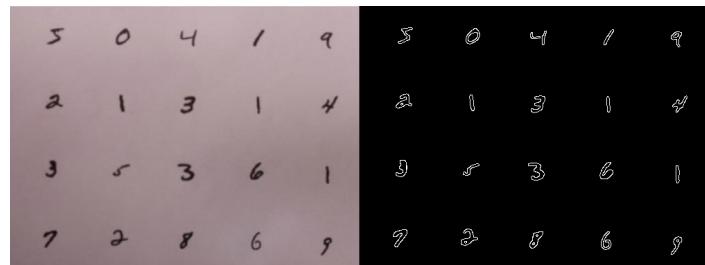
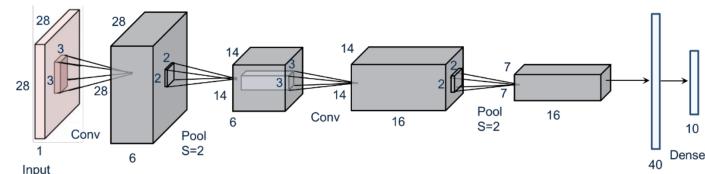


Klassifikation mittels HOG

Interest Points, Feature Descriptor, Tracking

Klassifikation/Kategorisierung mittels CNN

Die 'beste' Art Objekte in Echtzeit zu erkennen. Das Bild wird mit Filtermasken verkleinert (z.B. Canny, Sobel-Filter) und daraus wichtige Informationen entnommen, wie z.B. Form der erkannten Objekten.



Canny Kantendetektion → Dilation → Mit Bounding Boxes die Nummern ausschneiden → in CNN stecken und mit einem Wahrsch.-Schwellwert die Nummer auswählen.

Matlab

Uebung01: Bild einlesen, umwandeln Grauwert, Darstellung Histogramm

Uebung02: LUT, Gamma, Gleitendes Mittel, MotionDetektion, Spreiz.

Uebung03: Bildschärfung, Faltung/Korrelation, Glättung, Gradient

Uebung04: Skeleton, KugelnSeparieren, Morphologie

Uebung05: Aliasing, NotchFilter, DFT, Frequenzen aus bild Filtern

Uebung06: Otsu, Knossos, Kugeln Label/Zählen, MerkmalsExtraktion

Uebung07: GradientCanny/Simple, HoughTransform

Uebung08: BlueScreen, Background, FarbHistogramm

Uebung13: Training Mnist CNN,

Git Crashkurs

Setup & init

```
git config --global user.name "<name>"  
git config --global user.email "<email>"  
git init # initialize folder as git repo  
git clone <url> # retrieve a repo from given URL
```

Stage & Snapshot

```
git status # show staged, modified & untracked files  
git add <file> # adds file to the staged  
git reset <file> # unstage file  
git diff # shows unstaged changes  
git diff --staged # shows staged changes  
git commit -m "<msg>" # commit your staged content as a  
→ new commit snapshot
```

Branch & Merge

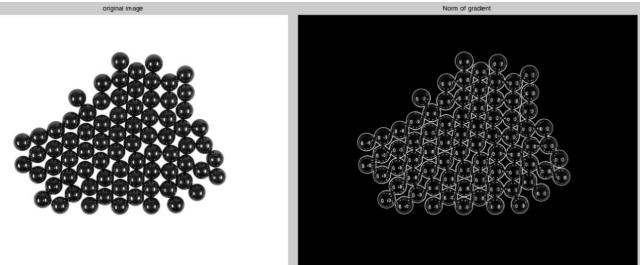
```
git branch # list branches  
git branch <branch-name> # create branch  
git checkout <branch-name> # switch to another branch  
git merge <branch> # merge given branch into current  
→ branch
```

Inspect & Compare

```
git log # show the commit history for the currently  
→ active branch  
git log branchB..branchA # show the commits on branchA  
→ that are not on branchB  
git log --follow <file> # show the commits that changed  
→ file, even across renames  
git diff branchB...branchA # show the diff of what is in  
→ branchA that is not in branchB  
git show <SHA> # show any object in Git in human-readable  
→ format
```

Anwendung

Erkennung, Zählung und Markierung von kreisförmigen Objekten



Tracking Path Changes

```
git rm <file> # delete the file from project and stage  
→ the removal for commit  
git mv <existing-path> <new-path> # change an existing  
→ file path and stage the move  
git log --stat -M # show all commit logs with indication  
→ of any paths that moved
```

Share & Update

Retrieving updates from another repository and updating local repos

```
git remote add <alias> <url> # add a git URL as an alias  
git fetch <alias> # fetch down all the branches from that  
→ Git remote  
git merge <alias>/<branch> # merge a remote branch into  
→ your current branch to bring it up to date  
git push <alias> <branch> # Transmit local branch commits  
→ to the remote repository branch  
git pull # fetch and merge any commits from the tracking  
→ remote branch  
git pull -t <url> # additionally includes all tags
```

Temporary Commits

Temporarily store modified, tracked files in order to change branches

```
git stash # Save modified and staged changes  
git stash list # list stack-order of stashed file changes  
git stash pop # write working from top of stash stack  
git stash drop # discard the changes from top of stash  
→ stack
```

make & Makefiles

make wird zum Automatisieren von Kompilierung und Linking gemacht.

Targets

```
make <target> # build target <target>, or on empty build  
→ first target in the makefile
```

Im Makefile wird folgend aufgeführt.

```
<target>: <depend1> <depend2> ...  
→ <command1>  
→ <command2>
```

Pattern Regeln

```
%:o: %.c '%' ist Wildcard ; erzeugte .o Dateien aller  
→ .c Dateien  
→ _gcc -Wall -g -c $<  
  
#'$<' erste Abhängigkeit  
#'$@' Name des targets  
#'$+' Liste aller Abhängigkeiten  
#'$^' Liste aller Abhängigkeiten (ohne Doppelträge)
```

Variablen

```
VERSION = 3.02  
CC = /usr/bin/gcc  
CFLAGS = -Wall -g -D_REENTRANT  
LDFLAGS = -lm -lpthread  
  
OBJ = datei1.o datei2.o datei3.o datei4.o datei5.o  
  
prog: $(OBJ)  
→ _$(CC) $(CFLAGS) -o prog $(OBJ) $(LDFLAGS)  
  
.o: %.c  
→ _$(CC) $(CFLAGS) -c $<
```

Beispiel

```
COMPILER=g++  
OPTIONS=-g -std=c++17 -pedantic -Wall -Wextra -Werror  
→ -Wshadow -Wconversion -Wunreachable-code  
COMPILE=$(COMPILER) $(OPTIONS)  
BUILD=build  
  
# Compile main by default  
all: program  
  
# $(BUILD)/*.o expands to all .o files in the $(BUILD)  
→ directory  
# In this case, we'll get $(BUILD)/file1.o  
→ $(BUILD)/file2.o  
program: main.cpp $(BUILD)/file1.o $(BUILD)/file2.o  
→ _$(COMPILE) $< $(BUILD)/*.o -o $@  
  
$(BUILD)/file1.o: file1.cpp file1.h build  
→ _$(COMPILE) -c $< -o $@  
  
$(BUILD)/file2.o: file2.cpp build  
→ _$(COMPILE) -c $< -o $@  
  
# Make the build directory if it doesn't exist  
build:  
→ _mkdir -p $(BUILD)  
  
# Delete the build directory and program
```

clean:

```
→ _rm -rf $(BUILD) program
```

```
# These rules do not correspond to a specific file  
.PHONY: build clean
```

Pi Stuff

Filehandling: - ls -l: Zeigt den detaillierten Inhalt des aktuellen Verzeichnisses an - pwd: Zeigt den aktuellen Pfad an - cp: Kopiert ein File oder ein Verzeichnis - rm: löscht ein File oder ein Verzeichnis - chown -R pi:pi: Ändert die Besitzer- und Gruppenzugehörigkeit der angegebenen Datei(en) rekursiv auf pi.

Prozesse: - chmod a+x app: erteilt allen Benutzern Ausführrechte der Applikation app13 - ./app: Started die ausführbare Applikation app14 - ./run.sh: Startet Shell Skript run.sh - top: Zeigt alle aktuell laufenden Prozess an (mit Ctrl-C beenden)

Netzwerk: - ping 192.168.1.10: Sendet einen Ping Request an den Raspberry Pi - ifconfig: Zeigt bzw. ändert die aktuelle Netzwerkkonfiguration - sudo service networking restart: Neustart der Netzwerkinterfaces15 - netstat -r: Zeigt die Routing Tabelle an. - route del: Löscht eine Route - route add: Fügt eine Route hinzu

Makefiles: - make: erstellt eine Applikation mittels Makefile - make clean: löscht alle Binär- und Objektfiles einer Applikation

Remote: - ssh pi@192.168.1.10 oder ssh 192.168.1.10: Stellt eine SSH-Verbindung zum oder Raspberry Pi her16.