



# ORB

**(Oriented FAST and Rotated BRIEF)**

**von Timon Brüning**

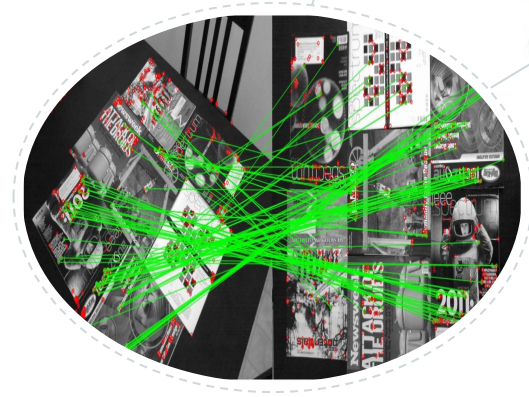


# Übersicht

1. Einführung
  - a. Motivation
  - b. Ähnlichkeitssuche in Multimedia Daten
2. Feature Detection / Description
  - a. Feature
  - b. Ermittlung interessanter Punkte
  - c. Beschreibung von Features
3. ORB
  - a. Oriented FAST and Rotated BRIEF
  - b. FAST (oFAST)
  - c. BRIEF (rBRIEF)
  - d. Demo
4. Zusammenfassung

# Einführung

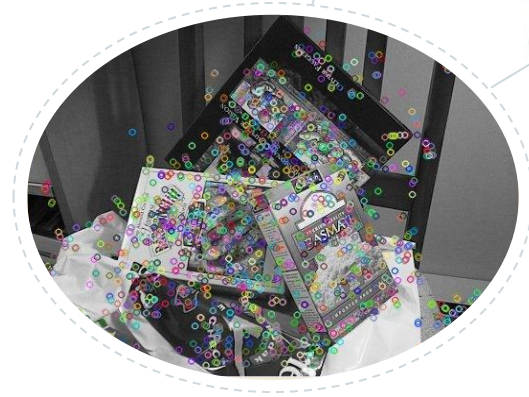
Ähnlichkeitssuche in  
Multimedia Daten



# Multimediale Ähnlichkeitssuche

- ◎ Wiedererkennen von Bildinhalten im Bereich des Maschinellen Sehens und Bildverarbeitung
  - Robotik
  - Multimediadaten
  - ◎ Nehmen massiv zu

# Feature Detection/ Description





“

*In computer vision and image processing, a feature is a **piece of information** which is **relevant** for solving the computational task related to a certain application.* [10]

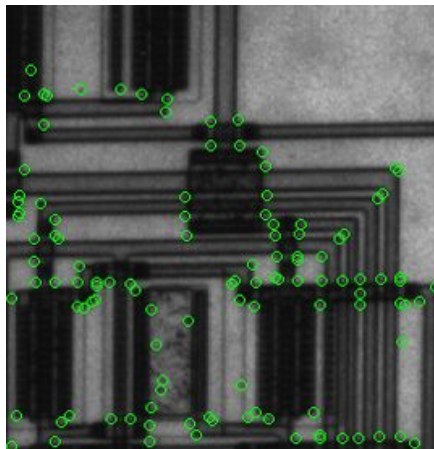
# Feature

- ⊙ Nicht einheitlich Definiert
- ⊙ “**interessante**” Teil eines Bildes
- ⊙ Bildet eine Art **Startpunkt**
- ⊙ Verschiedene **Typen** sind inhärent

# Feature - Typen



[1]



[2]



[3]

◎ Kante

◎ Ecke


◎ Blob

- ◎ Gradientenbasiert
- ◎ Templatebasiert
- ◎ Kontourbasiert



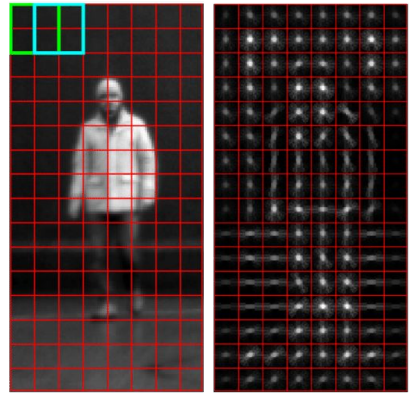
# Feature - Detector



- ◎ Erkennt “interessante” Bereiche
    - Position/Größe der Punkte/Gebiete
  - ◎ Leistungsindikatoren:
    - Wiederholbarkeit
    - Eindeutigkeit
    - Anzahl
    - Effizienz
- 

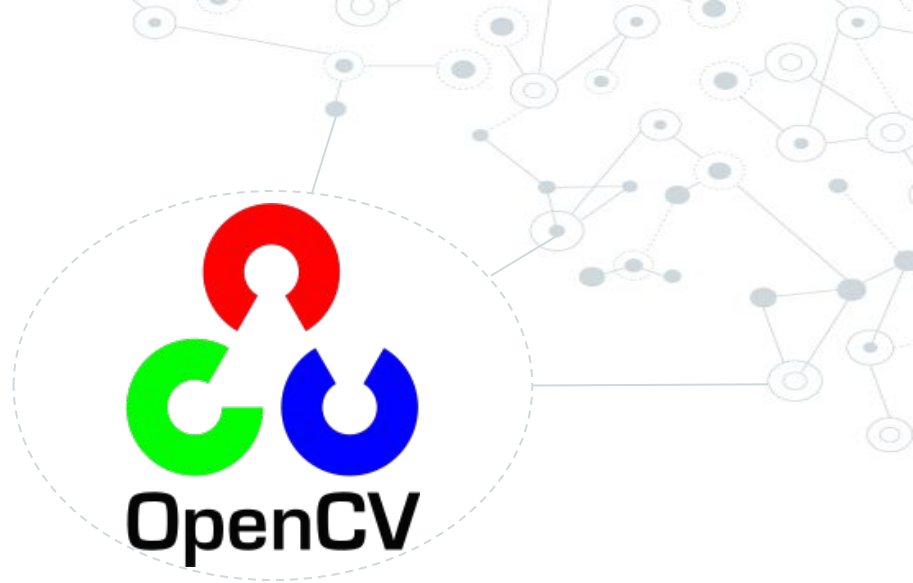
# Feature - Descriptor

- ◎ Berechnet numerischen Fingerabdruck
  - Umgebung eines Features wird betrachtet (z.B. Rohwerte oder Histogramm of oriented Gradients)
- ◎ **Leistungsindikatoren:**
  - Rotations-/ Skaleninvariant
  - Eindeutig
  - Effizient



[4]

# Feature Description: ORB



# Oriented Fast and Rotated Brief

- ◎ 2011 veröffentlicht von “OpenCV Labs”
  - → **patentlose** Alternative zu SIFT/SURF
  - Entwickelt von E. Rublee et al.
- ◎ **Effiziente Alternative**
  - Feature detection:  $\approx$  SIFT,  $>$  SURF
  - 1 - 2 Größenordnungen **schneller** als SURF / SIFT
  - Geringere **Platzkomplexität**
- ◎ **Basiert auf:**
  - FAST (detector)
  - BRIEF (descriptor)

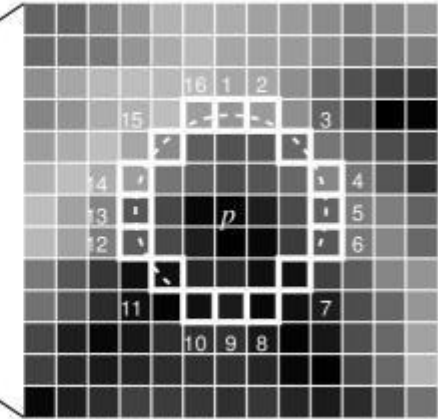
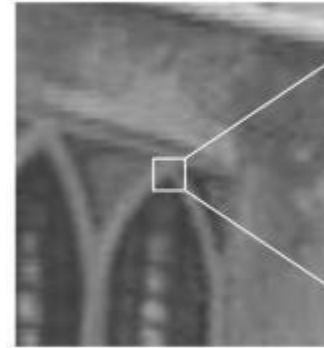
# (o)FAST

- ◎ Features from Accelerated Segment Test
- ◎ Aufgabe: Auffinden von Ecken
- ◎ Methode: Template
- ◎ Sehr Effizient durch einfache Vergleichsoperationen
- ◎ oFast erweitert um Skalen-, Rotationsinvarianz

# FAST

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t & \text{(darker)} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t & \text{(similar)} \\ b, & I_p + t \leq I_{p \rightarrow x} & \text{(brighter)} \end{cases}$$

$$C(p) = \max(S_{\text{bright}}, S_{\text{dark}})$$



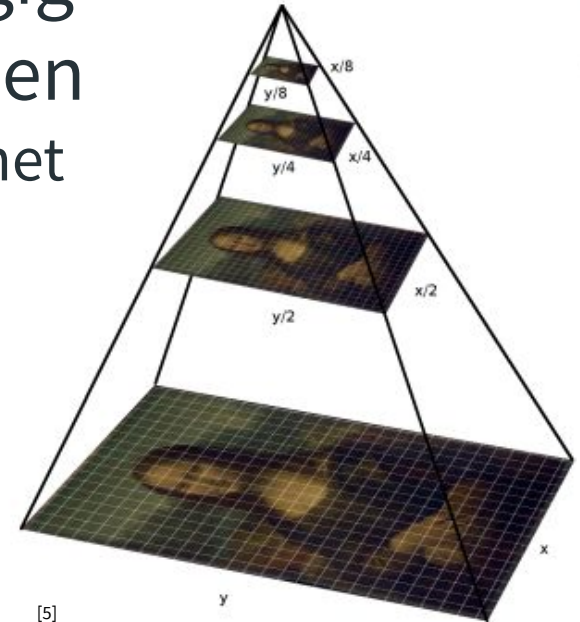
Wenn nun  $C(p) \geq N \rightarrow p$  ist Ecke

- ⊙ N=9 liefert sehr gute Ergebnisse (FAST-9)
- ⊙ t ist Kontextabhängig zu wählen
- ⊙ Radius 3 um p bleibt unverändert

[6]

# oFast

- FAST ist von Auflösung abhängig
- oFAST verwendet Bildpyramiden
  - Ecken werden auf Ebenen berechnet
  - partielle Skaleninvarianz

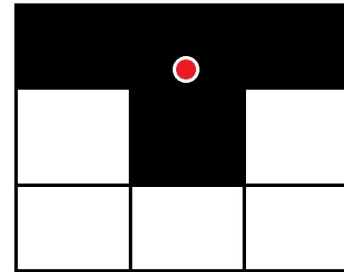


[5]

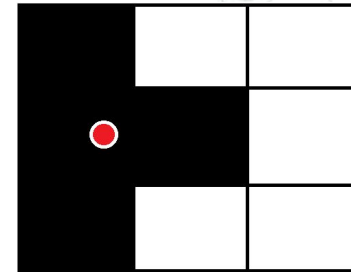
# oFast

- FAST ist nicht rotationsrobust
- oFAST verwendet Intensitätsschwerpunkte (intensity centroids)

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$



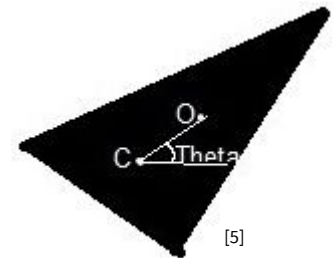
m00=4 | m10=8 | m01=11 | C=(2,2.75)



m00=4 | m10=5 | m01=8 | C=(1.25,2)

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

$$\theta = \text{atan2}(m_{01}, m_{10})$$



[5]



# (r)BRIEF

- ◎ Binary Robust Independent Elementary Features
- ◎ Aufgabe: Umgebungsbeschreibung von Feature
- ◎ Binäre Featurebeschreibung (128-512 bits)
  - → Ressourcenschonend

# BRIEF

1. Definiere die Nachbarschaft  $N$  um das Feature  $F$
2. Ausgehend von Gauß-Verteilung  $G$  mit Abweichung  $\sigma$  und  $F$ , wähle ein Pixel  $p_1$  in  $N$
3. Ausgehend von  $p_1$  und  $G$  mit  $\sigma/2$ , wähle  $p_2$
4. Wenn  $I(p_1) > I(p_2)$  wird 1 gewählt, sonst 0



[5]

# rBRIEF

- ◎ Standard BRIEF ist anfällig gegenüber Rotation
- ◎ rBRIEF (Rotation-aware BRIEF) wirkt entgegen
- ◎ Adaption an verschiedenen Stellen
  - Steuerung durch Orientierung der Features
  - Generierung einer Lookup-Tabelle durch ein Machine Learning Verfahren

**Live-Demo**





# Fazit



# Zusammenfassung

- ◎ Feature, Feature Detector / Descriptor
- ◎ FAST / oFAST
- ◎ BRIEF / rBRIEF
- ◎ ORB eine Alternative?

Detector	ORB	SURF	SIFT
Time per frame (ms)	<b>15.3</b>	217.3	5228.7

[7]

# Quellen

- [1] <https://www.mathworks.com/discovery/edge-detection.html>
- [2] <https://www.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html>
- [3] <https://sites.google.com/a/umich.edu/eecs442-winter2015/homework/blobs>
- [4] [https://www.researchgate.net/figure/Histogram-of-oriented-gradients-descriptor-a-The-histogram-of-oriented-gradients-HoG\\_fig3\\_256451629](https://www.researchgate.net/figure/Histogram-of-oriented-gradients-descriptor-a-The-histogram-of-oriented-gradients-HoG_fig3_256451629)
- [5] <https://medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf>
- [6] [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_fast/py\\_fast.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html)
- [7] <https://ieeexplore.ieee.org/document/6126544> - [http://www.willowgarage.com/sites/default/files/orb\\_final.pdf](http://www.willowgarage.com/sites/default/files/orb_final.pdf)
- [8] [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_orb/py\\_orb.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html)
- [9] [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_brief/py\\_brief.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_brief/py_brief.html)
- [10] [https://en.wikipedia.org/wiki/Feature\\_\(computer\\_vision\)](https://en.wikipedia.org/wiki/Feature_(computer_vision))
- [11] [https://www.researchgate.net/publication/221304115\\_BRIEF\\_Binary\\_Robust\\_Independent\\_Elementary\\_Features](https://www.researchgate.net/publication/221304115_BRIEF_Binary_Robust_Independent_Elementary_Features)
- [12] [https://link.springer.com/chapter/10.1007%2F11744023\\_34](https://link.springer.com/chapter/10.1007%2F11744023_34) & [https://www.researchgate.net/publication/38114446\\_FASTER\\_and\\_better\\_A\\_Machine\\_Learning\\_Approach\\_to\\_Corner\\_Detection](https://www.researchgate.net/publication/38114446_FASTER_and_better_A_Machine_Learning_Approach_to_Corner_Detection)



**Vielen Dank!**



A network diagram consisting of various nodes (circles) connected by lines. A large, empty circle with a dashed border is positioned in the upper right quadrant of the image. The nodes are scattered around it, with some connected to the dashed circle's boundary.

# Backup