



Universität Hamburg

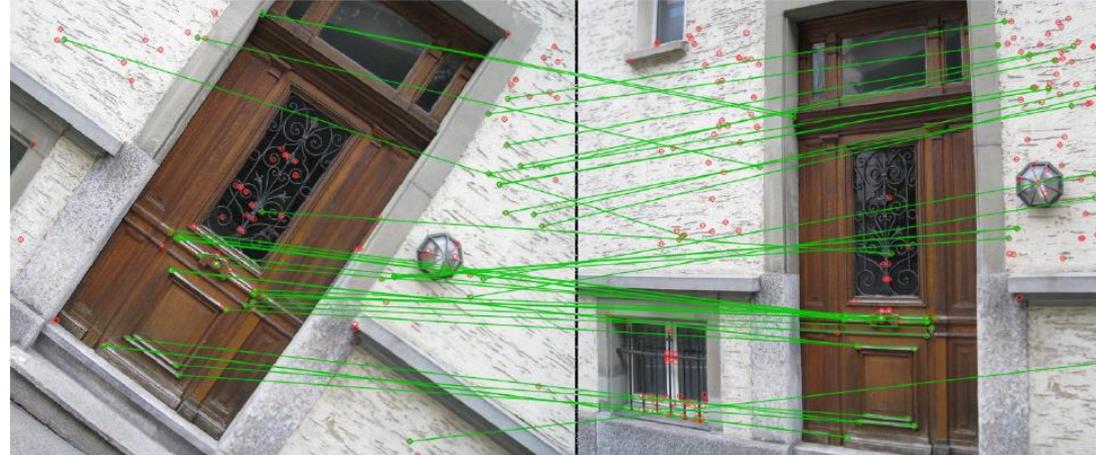
DER FORSCHUNG | DER LEHRE | DER BILDUNG

BRISK: Binary Robust Invariant Scalable Keypoints

Christian Brzeski

- SIFT und SURF sind führende Algorithmen zur schnellen und robusten Erkennung von Bildmerkmalen
- SURF Algorithmus weist bislang beste Performance auf (Stand 2011)
- BRISK verspricht qualitativ hochwertige Performance verbunden mit geringerem Rechenaufwand

- Bild wird in Regionen zerlegt
- Ein **Keypoint Detector** ist robust und findet auch nach Transformation immer wieder dieselben Regionen
- Ein **Keypoint Descriptor** extrahiert dabei die wichtigsten Informationen aus den gegebenen Regionen

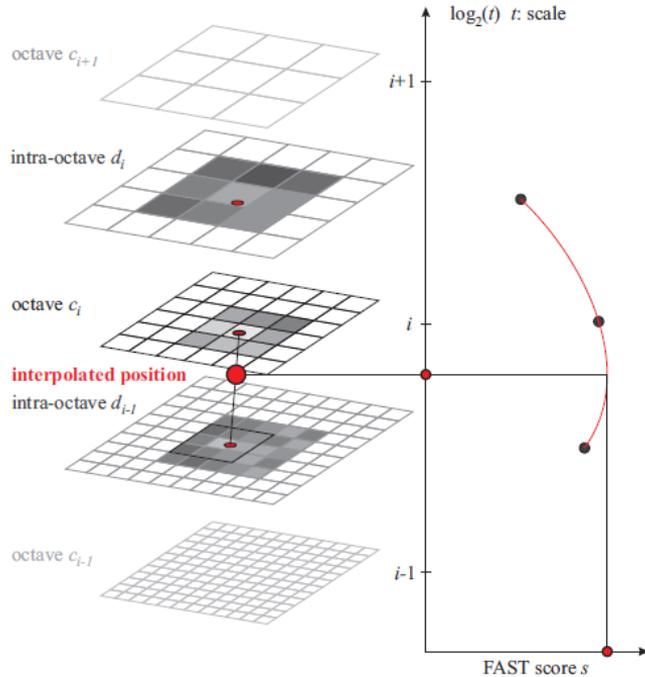


Quelle: <http://www.asl.ethz.ch/people/lestefan/personal/BRISK>

Scale Space Keypoint Detection

- Keypoints sollen invariant gegenüber verschiedenen Skalierungen sein
- FAST: Methode um markante Stellen (z.B. Kanten, Ecken) in Bildern zu identifizieren
- Um diese Invarianz zu garantieren wird das Bild in sogenannte Oktaven zerlegt und untersucht

Methode: BRISK



- FAST: Vergleich von Subpixeln mit zentralem Pixel auf den verschiedenen Oktaven
- Threshold T zur Identifikation von herausragenden Regionen
- In den Regionen werden die FAST Scores jedes einzelnen Pixels mit den Scores der benachbarten Pixel verglichen
- Wird Maximum Bedingung erfüllt, werden die FAST Scores der darunter und darüber liegenden Oktave ermittelt und mit Score des betrachteten Pixels verglichen

Quelle: S. Leutenegger et al.: BRISK: Binary Robust invariant scalable keypoints.

Keypoint Description

- Um Invarianz gegenüber Rotation sicherzustellen, wird im folgenden Schritt die Ausrichtung der gegebenen Keypoints ermittelt
- Für $N \cdot \frac{N-1}{2}$ Punktpaare (p_i, p_j) werden die Gradienten $g(p_i, p_j)$ berechnet
- $$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2}$$

- Gradient für Keypoint k : $g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in L} g(p_i, p_j)$
- Das betrachtete Bild wird um $\alpha = \arctan2(g_y, g_x)$ rotiert
- BRISK Descriptor d_k vergleicht Intensität über $(p_i^\alpha, p_j^\alpha) \in S$

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & \textit{otherwise} \end{cases}$$

- Ähnlichkeitsmaß zwischen zwei Bildern berechnet sich aus Hamming Distanzen zwischen den d_k

Testdatensatz:



(a) Graffiti



(b) Wall



(c) Boat



(d) Ubc



(e) Leuven

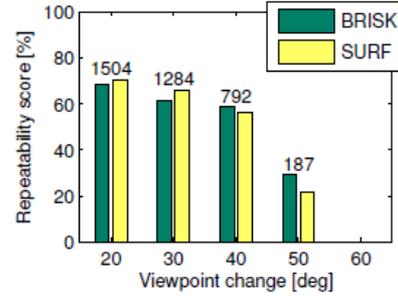


(f) Bikes

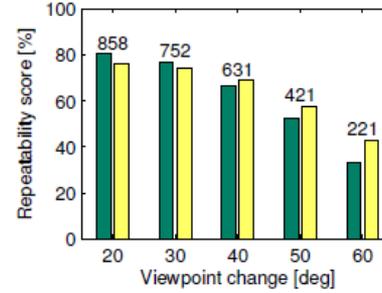


(g) Trees

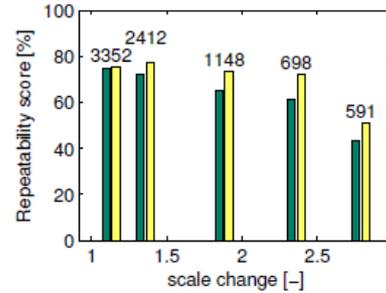
Quelle: S. Leutenegger et al.: BRISK: Binary Robust invariant scalable keypoints



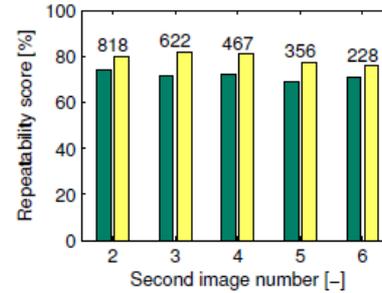
(a) Graffiti



(b) Wall



(c) Boat



(d) Leuven

	SIFT	SURF	BRISK
Detection threshold	4.4	45700	67
Number of points	1851	1557	1051
Detection time [ms]	1611	107.9	17.20
Description time [ms]	9784	559.1	22.08
Total time [ms]	11395	667.0	39.28
Time per point (ms)	6.156	0.4284	0.03737

Table 1. Detection and extraction timings for the first image in the Graffiti sequence (size: 800×640 pixels).

	SIFT	SURF	BRISK
Points in first image	1851	1557	1051
Points in second image	2347	1888	1385
Total time [ms]	291.6	194.6	29.92
Time per comparison [ns]	67.12	66.20	20.55

Table 2. Matching timings for the Graffiti image 1 and 3 setup.

- Aus FAST Keypoint Detector sowie Bit String Descriptor ergeben sich Performancevorteile gegenüber anderen Algorithmen (z.B. SIFT, SURF)
- Praktische Relevanz von BRISK bleibt jedoch offen
 - BRISK Veröffentlichung: 2011
 - Neue Methoden: local textron XOR patterns, local tree directional patterns
 - SIFT und SURF weiterhin weit verbreitet aufgrund von guten Recall- und Precision- Werten

Quellverzeichnis

- H. Bay, A. Ess, T. Tuytelaars, L. J. Van Gool: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110(3), 346-359 (2008)
- A. E. Abdel-Hakim, A. A. Farag: CSIFT: A SIFT Descriptor with Color Invariant Characteristics. CVPR (2), 1978-1983 (2006)
- S. Leutenegger, M. Chli, R. Siegwart: BRISK: Binary Robust invariant scalable keypoints. ICCV, 2548-2555 (2011)
- <https://de.wikipedia.org/wiki/SURF>
- https://en.wikipedia.org/wiki/Features_from_accelerated_segment_test
- https://de.wikipedia.org/wiki/Canny-Algorithmus#Non-maximum_suppression