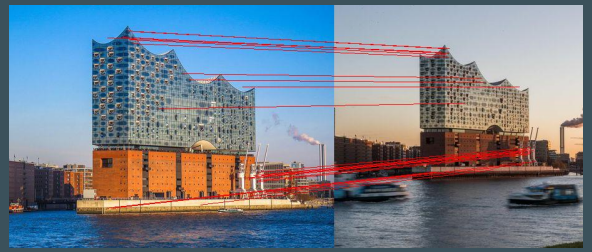
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



Luca Pankratz

SIFT – Scale-Invariant Feature Transform

Gliederung

- SIFT
- Praxisbeispiel
- PCA-SIFT
- CSIFT
- Quellen

SIFT

- Eine der bekanntesten Local-Feature-Descriptor-Methoden
- Wurde 1999 von David Lowe vorgestellt
- Motivation: Deskriptoren sollten invariant zur Skalierung, Position und Rotation sein
- Vektor mit 128 Dimensionen beschreibt einen Keypoint

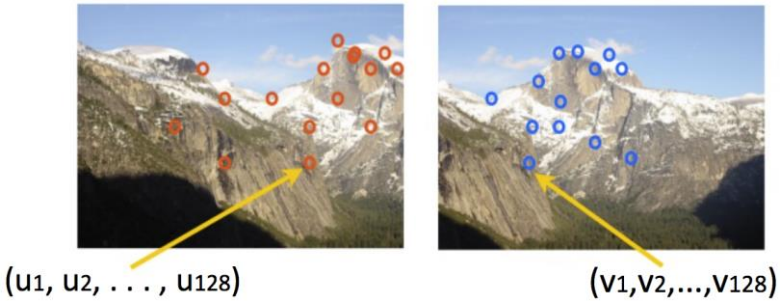


Abb. 1 – SIFT Keypoints

SIFT

Vorgehen: Keypoints finden

Prozedur wird über verschiedene Ebenen einer Bildpyramide durchgeführt

- Glätten mit Gauß-Filter
- Difference-of-Gaussians (DoG)
- Maxima-Unterdrückung an Kanten

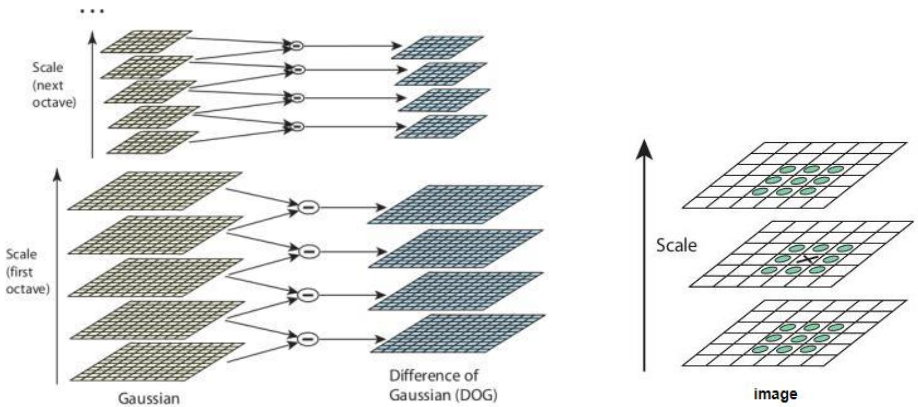


Abb. 2 – DoG

SIFT

Vorgehen: Keypoints finden

Prozedur wird über verschiedene Ebenen einer Bildpyramide durchgeführt

- Glätten mit Gauß-Filter
- Difference-of-Gaussians (DoG)
- Maxima-Unterdrückung an Kanten

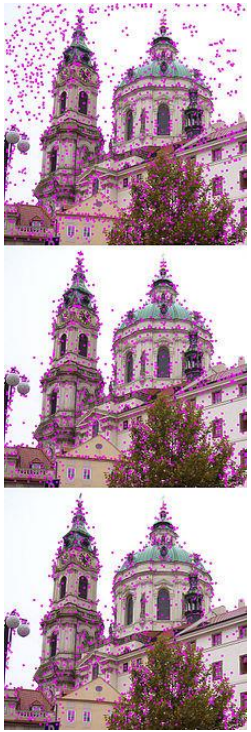


Abb. 3 – Keypoints auswählen

SIFT

Vorgehen: Keypoint Deskriptor

Vektor = <p,s,r,f>

p = Position

s = Skalierung

r = Orientierung

f = Feature

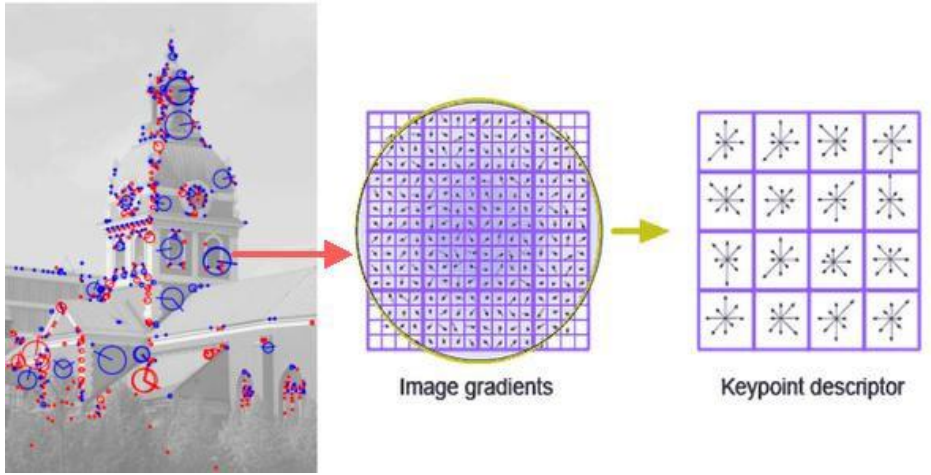


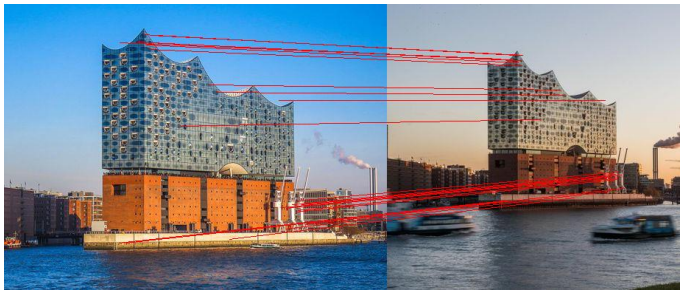
Abb. 4 – Keypoint Deskriptor

SIFT

Vorgehen: SIFT Matching

Falsche Verknüpfungen?

- RANSAC-Algorithmus
- PCA-SIFT, C-SIFT



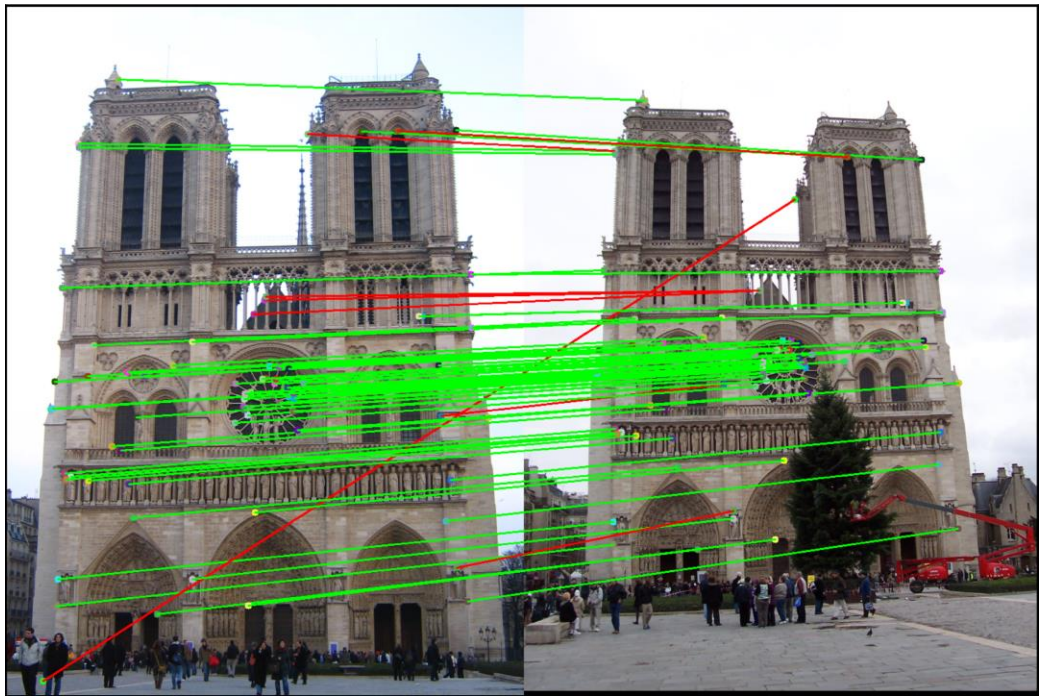


Abb. 5 – SIFT Matching

Praxisbeispiel SIFT & RANSAC mit OpenImaj

```
DoGSIFTEngine engine = new DoGSIFTEngine();
LocalFeatureList<Keypoint> queryKeypoints = engine.findFeatures(query.flatten());
LocalFeatureList<Keypoint> targetKeypoints = engine.findFeatures(target.flatten());

LocalFeatureMatcher<Keypoint> matcher = new BasicTwoWayMatcher<Keypoint>();
matcher.setModelFeatures(queryKeypoints);
matcher.findMatches(targetKeypoints);

MBFImage basicMatches = MatchingUtilities.drawMatches(query, target, matcher.getMatches(), RGBColour.RED);
DisplayUtilities.display(basicMatches);

RobustAffineTransformEstimator modelFitter = new RobustAffineTransformEstimator(5.0, 1500,
          new RANSAC.PercentageInliersStoppingCondition(0.5));
        matcher = new ConsistentLocalFeatureMatcher2d<Keypoint>(new FastBasicKeypointMatcher<Keypoint>(8), modelFitter);

        matcher.setModelFeatures(queryKeypoints);
        matcher.findMatches(targetKeypoints);

        MBFImage consistentMatches = MatchingUtilities.drawMatches(query, target, matcher.getMatches(),
          RGBColour.RED);
        int j = 0;
        List<Pair<Keypoint>> list1 = matcher.getMatches();
        for(int i = 0; i < list1.size(); i++) {
              j++;
        }
        System.out.println("Anzahl der gefundenen robusten Matches: " + j);
        DisplayUtilities.display(consistentMatches);
```

PCA-SIFT

- Verändert den Deskriptor der Keypoints
- Statt Histogramm wird eine Hauptkomponentenanalyse durchgeführt
- Kompakter Feature-Vektor
- Weniger Speicher notwendig
- Schnelleres Matchen

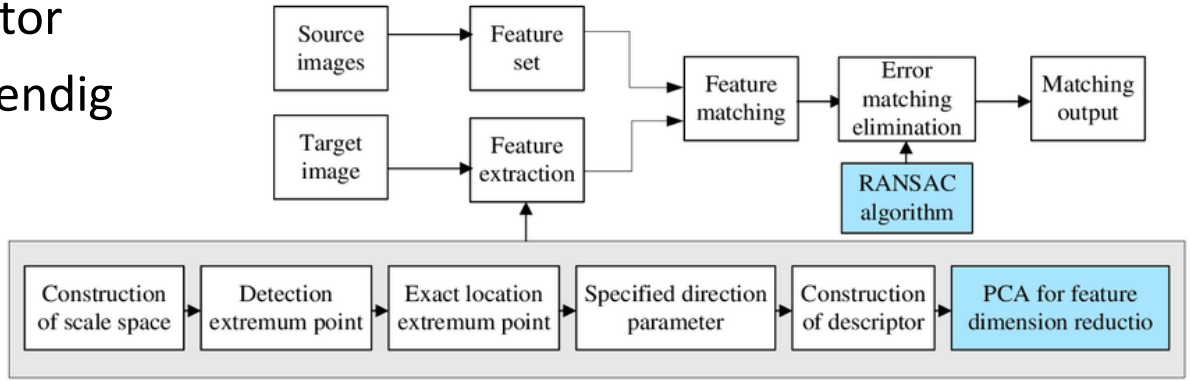


Abb. 6 – Einsatz von PCA und RANSAC

PCA-SIFT

Abb. 7 – SIFT & PCA-SIFT Vergleich

C-SIFT

- Viele Objekte werden falsch klassifiziert, wenn die Farbe missachtet wird
- Deskriptor baut auf einem „color invariance model“ auf
- Durch CSIFT wird der Algorithmus robuster gegen Belichtungsunterschiede

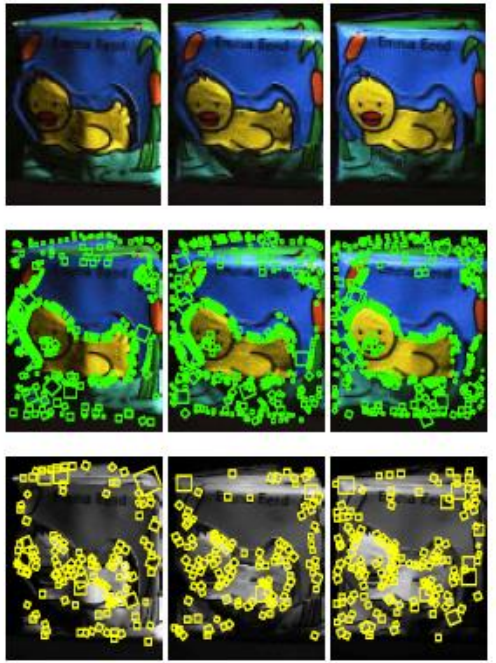


Abb. 8 – SIFT & CSIFT Vergleich

Quellen:

Y. Ke und R. Sukthankar - PCA-SIFT: A More Distinctive Representation for Local Image Descriptors

A. E. Abdel-Hakim und A. A. Farag - CSIFT: A SIFT Descriptor with Color Invariant Characteristics

http://ai.stanford.edu/~syyeung/cvweb/tutorial2.html

https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html

http://openimaj.org/tutorial/sift-and-feature-matching.html

https://www.youtube.com/watch?v=oT9c_LlFBqs&t=3440s

Bildquellen:

Abb. 1: http://ai.stanford.edu/~syyeung/cvweb/tutorial2.html
Abb. 2: http://ai.stanford.edu/~syyeung/cvweb/tutorial2.html
Abb. 3: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform#/media/File:Sift_keypoints_filtering.jpg
Abb. 4: https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774
Abb. 5: https://www.cc.gatech.edu/~hays/compvision/proj2/
Abb. 6: https://www.researchgate.net/figure/The-flow-chart-of-the-algorithm-for-PCA-SIFT_fig1_327325372
Abb. 7: A. E. Abdel-Hakim und A. A. Farag - CSIFT: A SIFT Descriptor with Color Invariant Characteristics
Abb. 8: Y. Ke und R. Sukthankar - PCA-SIFT: A More Distinctive Representation for Local Image Descriptors