



Identifikation von historischen Gebäuden und Bauteilen durch Bildklassifikation

Christof Wittmann

Bachelorarbeit

im Studiengang Angewandte Informatik
Fakultät Wirtschaftsinformatik und Angewandte Informatik
Otto-Friedrich-Universität Bamberg

4.4.2019

Wissenschaftliche Betreuung: Prof. Dr. Christoph Schlieder
Softwaretechnischer Ansprechpartner: Thomas Heinz
Lehrstuhl für Angewandte Informatik
in den Kultur-, Geschichts- und Geowissenschaften

Inhaltsverzeichnis

1. Einleitung.....	1
2. Problemstellung.....	2
3. Forschungsstand: Methoden der Bilderkennung.....	3
3.1. Grundprinzipien der Merkmalerkennung (Feature Detection)	3
3.2. Methoden der Merkmalerkennung	4
3.3. Scale-Invariant Feature Transform (SIFT).....	4
3.3.1. Scale-Space Extrema Detection	5
3.3.2. Keypoint Localization	6
3.3.3. Orientation Assignment	7
3.3.4. Keypoint Description.....	8
3.3.5. Matching.....	9
3.4. Weitere Algorithmen zur Merkmalerkennung	10
4. Forschungsstand: Bildauswahl	14
5. Lösungsansatz.....	16
6. Umsetzung.....	17
7. Evaluierung.....	18
8. Diskussion	19
9. Anhang	20
10. Literaturverzeichnis	21
A. Eidesstattliche Erklärung	22

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

2. Problemstellung

Mobile kartographische Anwendungen erlauben zwar durch GPS etc. die Ermittlung der momentanen Position, gerade durch neuen GPS-Standard. Kompass-Sensor erlaubt auch Identifizierung der gegenwärtigen Richtung, so dass Gebäude grob zugeordnet werden können, indem sich Person bzw. mobiles Gerät so lange drehen, bis die Kompassnadel auf Gebäude zeigt. Diese ist ggfs. wenig komfortabel und hat folgende Nachteile:

- Gebäudekomplexe oft nur in Gesamtheit erfasst [Screenshot]
- Bedeutsame Bauteile oft nicht als eigenes Element auf Karte erfasst
- ...

Alternativ dazu: Benutzer machen Bild von Gebäude, erhalten nicht nur GPD-Position und Richtung auf Karte angezeigt, sondern auch sichere Identifikation von Gebäude mitsamt Titel und weiterführenden Informationen. Bei größeren Gebäudekomplexen oder besonderen Gebäudebestandteilen (Portale, Statuen, Brunnen, Fenster...) auch genaue Angaben.

3. Forschungsstand: Methoden der Bilderkennung

3.1. Grundprinzipien der Merkmalerkennung (Feature Detection)

Um Bilder informatisch miteinander vergleichen und ihre Ähnlichkeit ermitteln zu können, ist es erforderlich, sich auf bestimmte Attribute dieser Bilder zu konzentrieren. Bei Verfahren der Merkmalerkennung werden deshalb interessante Punkte ermittelt, die besonders für den Bildvergleich geeignet sind. Als interessant kann dabei ein Punkt gelten, der in Bezug auf seine Nachbarschaft eine signifikante Veränderung aufweist, etwa hinsichtlich seiner Farbe, seines Helligkeitswertes oder seiner Richtung. Die solchen Verfahren zugrundeliegende Annahme ist, dass derart interessante Punkte mit hoher Wahrscheinlichkeit auf allen Bildern zu finden sind, die ein identisches Objekt abbilden. [Andersson & Marquez, S. 8]

Die fotografische Aufnahme eines Objekts kann auf sehr unterschiedliche Weise erfolgen, wobei die fotografierende Person eine Vielzahl von Faktoren variieren kann, um zum gewünschten Ergebnis zu kommen. Eigenschaften wie Perspektive, Entfernung und Richtung können direkt durch Positionsänderung von Kamera und Person beeinflusst werden, wobei die Möglichkeiten ggfs. durch die Umgebungssituation des Objekts eingeschränkt werden. Mittels der Kameraeinstellungen ist etwa die Helligkeit oder Farbbalance der Aufnahme wählbar, ebenso das Format des erzeugten Bildes. Weniger Einfluss hat die fotografierende Person auf die Lichtverhältnisse, insbesondere im Freien. Selbst die Wahl einer geeigneten Tageszeit und der Einsatz künstlicher Beleuchtung können nicht verhindern, dass örtliche Lichtverhältnisse stark durch örtliche Wetterverhältnisse beeinflusst werden. Beim Bildvergleich ist es deshalb von herausragender Bedeutung, dass bezüglich der genannten Faktoren eine größtmögliche Invarianz gegeben ist. Dies bedeutet, dass bei identischen Objekten idealerweise auch die gleichen Features identifiziert werden, auch wenn die Aufnahmen in vielerlei Hinsicht erheblich voneinander abweichen. [Andersson & Marquez, S. 7-10]

Bei der Merkmalerkennung handelt es sich um eine der beiden Hauptrichtungen der inhaltsbasierten Bildsuche und -klassifikation. Alternativ dazu existieren auch Methoden, die sich des Maschinellen Lernens bedienen, um den Inhalt des Bildes auf der höchstmöglichen Ebene zu beschreiben. Entsprechend trainierte neuronale Netzwerke können somit bestimmte Bildbestandteile erkennen und klassifizieren, wobei diese Verallgemeinerung

jedoch auch mit einem Informationsverlust verbunden ist, der dazu führt, dass die Gleichheit von Objekten einer gemeinsamen Kategorie auf diese Weise schwer festzustellen ist. Ein weiterer Nachteil des Maschinellen Lernens ist der Bedarf an umfangreichen Mengen von Trainingsdaten. Die Merkmalerkennung nimmt dagegen keine Generalisierung oder Klassifikation vor. Sie ist nicht nur für den direkten Ähnlichkeitsvergleich von Bildern einsetzbar, sondern etwa auch beim Videotracking oder bei der dreidimensionalen Rekonstruktion von Objekten auf Basis photographischer Aufnahmen. [Scherer, S. 2]

3.2. Methoden der Merkmalerkennung

Merkmalerkennungs-Algorithmen können generell einer der drei folgenden Kategorien zugeordnet werden:

- Kantendetektion (Edge Detection)
- Eckendetektion (Corner Detection)
- Blobdetektion (Blob Detection)

Die Kantendetektion identifiziert Bildpunkte, die entlang einer Linie liegen, die auffallende Unterschiede bzgl. der vorliegenden Helligkeits- bzw. Farbwerte aufweist. Für sich genommen ist die Kantendetektion jedoch ungeeignet für die Merkmalerkennung und ist für diese somit nur von historischer Bedeutung.

Die Eckendetektion, für die etwa die Harris Corner Detection als bekanntes Beispiel genannt werden kann, bedient sich der Kantendetektion und ermittelt auf deren Basis Schnittpunkte zwischen zwei oder mehreren Kanten. Die so identifizierten Ecken sind als Features deutlich besser geeignet als Kanten. Nichtsdestotrotz ist die Eckendetektion nicht in der Lage, eine Invarianz bezüglich der Skalierung zu gewährleisten. Deshalb wird die Eckendetektion in heutigen Merkmalerkennungs-Algorithmen entweder gar nicht oder nur in Verbindung mit der Blobdetektion verwendet.

Ein entscheidender Vorteil der Blobdetektion ist die Invarianz gegenüber Perspektive, Entfernung und Rotation, womit die entsprechenden Algorithmen für die meisten Anwendungszwecke als Regel als Mittel der Wahl gelten können. Ein bekanntes Beispiel hierfür ist der SIFT-Algorithmus, der im Folgenden vorgestellt werden soll. [Andersson & Marquez, S. 8-9]

3.3. Scale-Invariant Feature Transform (SIFT)

Der Scale-Invariant Feature Transform-Algorithmus (im Folgenden als SIFT abgekürzt) wurde 1999 von David Lowe entwickelt. Anhand des Namens ist

bereits erkennbar, dass die grundlegende Verbesserung gegenüber bisherigen Merkmalerkennungs-Verfahren in der Invarianz bezüglich der Skalierung besteht. Der SIFT-Algorithmus wird in die folgenden vier Schritte aufgeteilt, die in den folgenden Kapiteln detailliert vorgestellt werden sollen:

1. Scale-Space Extrema Detection
2. Keypoint Localization
3. Orientation Assignment
4. Keypoint Description

3.3.1. Scale-Space Extrema Detection

Das Ziel des ersten Verarbeitungsschritts besteht darin, eine Vielzahl interessanter Punkte innerhalb des gewählten Bildes zu identifizieren. Diese werden im Rahmen des SIFT-Algorithmus als Keypoints bezeichnet.

Zu Beginn werden aus dem Ursprungsbild weitere Bilder erzeugt, die sich bezüglich Skalierung und Weichzeichnungsgrad voneinander unterscheiden. Dabei wird das Bild in der Ausgangsgröße zuerst stufenweise immer stärker weichgezeichnet, wobei der Gaussian Scale-Space Kernel zur Anwendung kommt. Alle Bilder der gleichen Größe werden als Oktave bezeichnet.

Anschließend wird das Bild mit dem größten Weichzeichnungsgrad auf die Hälfte seiner Größe verkleinert und erneut stufenweise weichgezeichnet. Dieser Prozess wiederholt sich für weitere Oktaven, bis ein Schwellenwert erreicht ist. Abbildung x zeigt exemplarisch die dabei erzeugten Bilder.

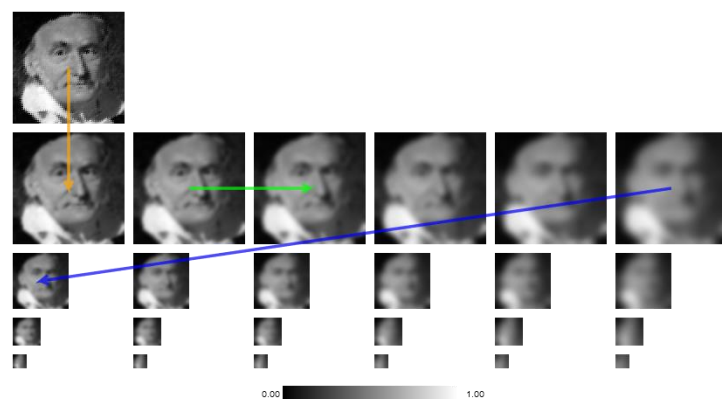


Abbildung x: Bilderzeugung im Rahmen der Scale-Space Extrema Detection (Quelle)

Nun werden aus diesen Bildern mittels der Difference of Gaussian (DoG)-Methode Differenzbilder generiert. Hierfür werden jeweils zwei innerhalb einer Oktave nebeneinanderliegende Bilder verwendet. In Abbildung x ist das Ergebnis zu sehen, wobei die Zahl der Bilder pro Oktave nun um eins verringert ist.

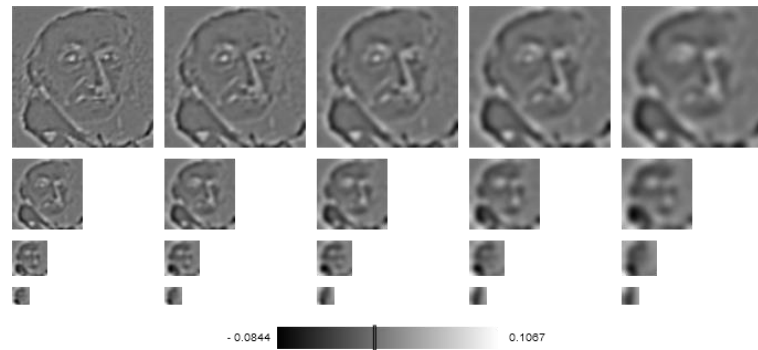


Abbildung x: Differenzbilder durch Difference of Gaussian-Berechnung (Quelle)

Schließlich werden die Pixel in diesen Differenzbildern anhand von Nachbarschaftsvergleichen auf ihre Eignung als interessante Punkte geprüft. Dabei werden nicht nur die umliegenden acht Pixel als Vergleichspunkte gewählt, sondern auch die angrenzenden neun Pixel in den Differenzbildern der nächstoberen und nächstunteren Oktaven.

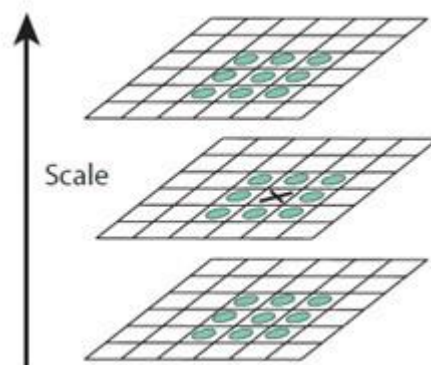


Abbildung x: Pixel-Nachbarschaftsvergleiche bei der Scale-Space Extrema Detection (Quelle)

Um als potentieller Keypoint in Frage zu kommen, muss ein Pixel einen höheren bzw. niedrigeren Wert aufweisen als alle 26 Nachbarn. Hierdurch wird die Skalierungsinvarianz gewährleistet (Warum? Sinn erklären.) [Lowe, S. 94-97, Andersson & Marquez, S. 10-11]

3.3.2. Keypoint Localization

Die im letzten Schritt ermittelten Keypoints müssen nun weiter eingegrenzt werden, da nicht alle von ihnen als Merkmale für die Bildidentifikation geeignet sind. Gründe für die fehlende Eignung sind entweder ein zu niedriger Kontrast oder die Lage entlang einer Kante.

Um Punkte mit niedrigem Kontrast zu identifizieren, wird zuerst mittels Taylorentwicklung die genaue Position lokaler Extrema bestimmt. Aus den so ermittelten Extrempunkten werden solche herausgefiltert, deren Wert einen gegebenen Schwellenwert von 0,03 unterschreiten.

Zur Entfernung von Kantenpunkten bedient man sich einem Verfahren, das der Harris Corner Detection verwandt ist. Um die beiden Hauptkrümmungen für alle Keypoints zu berechnen, wird die Hesse-Matrix verwendet. Anschließend wird das Verhältnis dieser Hauptkrümmungen ermittelt. Liegt dieses oberhalb des Schwellenwerts 10, so wird davon ausgegangen, dass der Punkt sich auf einer Kante befindet, weshalb er verworfen wird. [Lowe, S. 97-99, Andersson & Marquez, S. 11]

3.3.3. Orientation Assignment

Um die Invarianz gegenüber der Rotation sicherzustellen, wird nun jedem Keypoint eine Orientierung zugewiesen. Zuerst betrachtet man hierfür die Nachbarschaft des Punktes. Da alle Keypoints Pixel in einem weichgezeichneten Bild sind, besteht ihre Umgebung aus Helligkeitsverläufen (Gradients). Für diese Verläufe, welche in Abbildung x zu sehen sind, können sowohl Intensität als auch Richtung ermittelt werden.

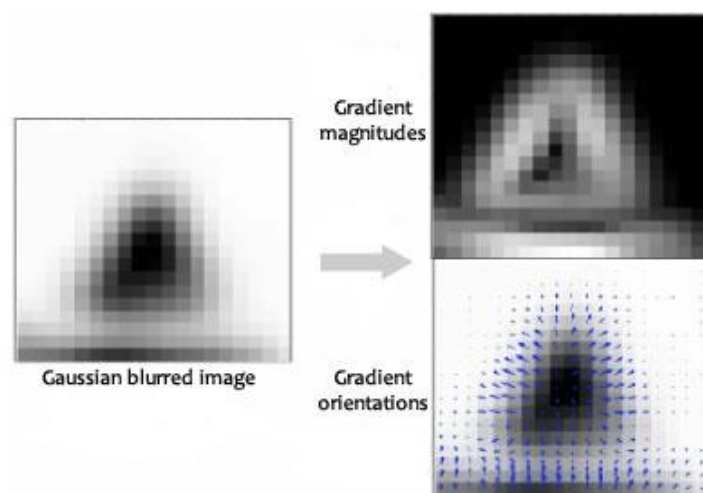


Abbildung x: Helligkeitsverläufe in Pixel-Nachbarschaft eines Keypoints (Quelle)

Es wird nun für jeden Keypoint ein Histogramm angelegt, in dem die Intensität des Verlaufs für die jede Orientierung hinterlegt wird. Aus Performancegründen teilt man die 360°-Umgebung jedoch in 36 Behälter auf, die jeweils einem 10°-Abschnitt entsprechen. Ein Beispielhistogramm ist in Abbildung x zu sehen.

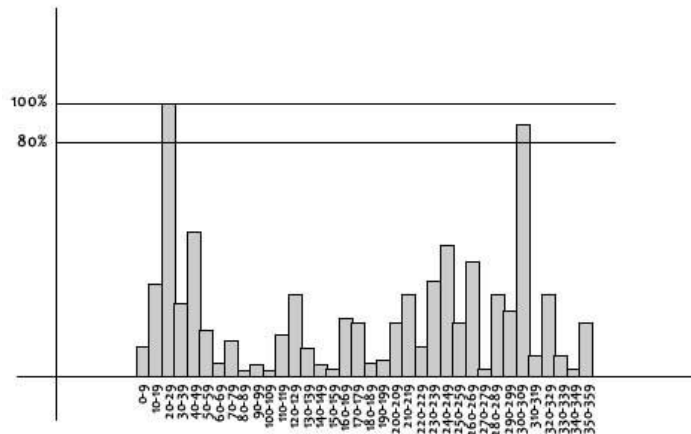


Abbildung x: Orientierungshistogramm eines Keypoints (<https://medium.com/analytics-vidhya/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>)

In den meisten Fällen wird nun der Behälter mit dem höchsten Wert gewählt und dessen Orientierung als Orientierung des Keypoints festgelegt. Wie in Abbildung x zu sehen ist, können jedoch auch mehrere Orientierungen vorliegen, die eine ähnliche Intensität aufweisen. Deshalb vergleicht man die Intensität aller Behälter mit der des Behälters mit dem Maximalwert. Für Behälter, die mindestens 80% von dessen Intensität erreichen, wird jeweils ein weiterer Keypoint mit der jeweiligen Orientierung des Behälters erstellt. Somit kann die endgültige Menge an Keypoints auch solche enthalten, deren Lage und Skalierung identische ist, und die sich lediglich hinsichtlich der Orientierung unterscheiden. [Lowe, S. 99-100, Andersson & Marquez, S. 11]

3.3.4. Keypoint Description

Nachdem jeder Keypoint bereits über eine Position, eine Skalierung sowie eine Orientierung verfügt, wird nun abschließend eine Beschreibung der Keypoint-Umgebung hinzugefügt. Diese dient dazu, den Keypoint eindeutig zu identifizieren und somit den Ähnlichkeitsvergleich von Bildern zu ermöglichen.

Zu diesem Zweck werden die Pixel in der Umgebung des Keypoints betrachtet. In Abbildung x ist auf der linken Seite die Nachbarschaft als Quadrat mit Seitenlänge 16 Pixeln zu sehen. Diese Umgebung wird nun in 16 Teilquadrate mit je 4 x 4 Pixeln aufgeteilt. Für jedes dieser Teilquadrate wird nun ähnliche wie im Schritt Orientation Assignment die Intensität der Helligkeitsverläufe und der Orientierung berechnet.

Die Ergebnisse dieser Berechnung werden nun für jedes Teilquadrat in einem Histogramm mit 8 Behältern gespeichert. Diese Behälter teilen die 360°-Umgebung in Bereich von jeweils 45°. Hierbei ist noch zu bemerken, dass Pixel,

die vom Keypoint weiter entfernt sind, schwächer gewichtet werden als solche, die diesem näher sind. Das Histogramm kann auch als Vektor verstanden werden, bei dem jedem der 16 Teilbereiche 8 Vektoren zugeordnet sind, deren Länge jeweils die Intensität des Helligkeitsverlauf in diese Richtung angeben. Abbildung x zeigt auf der rechten Seite den so entstandenen 128-dimensionalen Merkmalsvektor.

Um die Invarianz bzgl. der Rotation herzustellen, wird jeweils die Orientierung des Keypoints von den ermittelten Orientierungen subtrahiert. Die Helligkeitsinvarianz wird dagegen durch eine Normalisierung gewährt, bei der man einen oberen Schwellenwert für die auftretenden Vektoren festlegt. [Lowe, S. 100-103, Andersson & Marquez, S. 11-12]

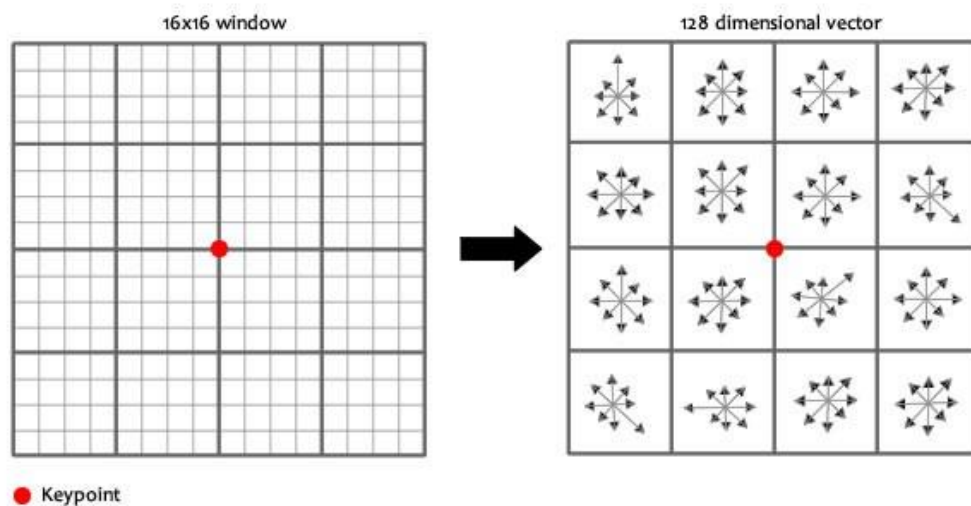


Abbildung x: Generierung eines 128-dimensionalen Vektors für einen Keypoint (<https://medium.com/analytics-vidhya/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>)

3.3.5. Matching

Nach Abschluss der vier Schritte liegt nun für das Ausgangsbild eine Menge von Keypoints vor. Möchte man nun zwei Bilder auf ihre Ähnlichkeit hin überprüfen, erfolgt dies durch den Vergleich ihrer Keypoints. Hierbei wird jeder Keypoint mit jedem Keypoint des anderen Bildes verglichen und jeweils der euklidische Abstand zwischen diesen Punkten ermittelt. Als Match kann derjenige Keypoint des anderen Bildes gelten, zu dem der euklidische Abstand am geringsten ist.

Hierbei ergibt sich jedoch das Problem, dass auch in Bildpaaren ohne gemeinsame Inhalte derartige falsch positive Übereinstimmungen auftreten. Zwar könnte man versuchen, dies durch die Festlegung eines globalen Schwellenwerts für den euklidischen Abstand zu verhindern, doch wird dieser

Ansatz der heterogenen Natur der Deskriptoren kaum gerecht. Alternativ dazu betrachtet man das Verhältnis zwischen dem kleinsten und zweitkleinsten Abstand und entfernt Matches, bei denen dieses Verhältnis zu groß ist. Hierdurch wird die Menge falsch positiver Matches im globalen Mittel um etwa 90% verringert. [Dawson-Howe, S. 127ff. Lowe & Marquez, S. 103 ff.]

Doch selbst durch diese Maßnahme kann keine endgültige Gewissheit bestehen, dass beim Vorliegen eines Matches tatsächlich ein identisches Objekt bzw. ein Bestandteil desselben auf beiden Bildern zu erkennen ist. Es ist deshalb unbedingt erforderlich, für den jeweiligen Anwendungsfall zu untersuchen, welche Abstände zwischen den Keypoints bei identischen Bildern zu erwarten sind und wie diese durch die Umstände der Bildkomposition beeinflusst werden. Das genaue Vorgehen für das in dieser Arbeit besprochene Anwendungsbeispiel wird in Kapitel x detailliert beschrieben.

3.4. Weitere Algorithmen zur Merkmalerkennung

Auch über 20 Jahre nach seiner ersten Veröffentlichung wird der SIFT-Algorithmus noch häufig zu Zwecken der Merkmalerkennung eingesetzt. In der Zwischenzeit haben sich jedoch zahlreiche weitere Algorithmen zu diesem hinzugesellt, deren Schöpfer den Anspruch haben, SIFT hinsichtlich Erkennungsgenauigkeit oder Geschwindigkeit zu übertreffen. Tabelle x liefert eine Übersicht über die populärsten Vertreter, die in den folgenden Kapiteln – besonders in Hinblick auf deren Unterschiede zu SIFT – genauer beschrieben werden sollen.

TABELLE X. ALGORITHMEN ZUR MERKMALSERKENNUNG

Name	Jahr der Veröffentlichung	Autor(en)
SIFT	1999	David Lowe
SURF	2006	Bay, Tuytelaars, Van Gool
BRISK	2011	Leutenegger, Chli, Siegwart
ORB	2011	Ethan Rublee
KAZE	2012	Alcantarilla, Bartoli, Davison
A-KAZE	2013	Alcantarilla, Nuevo, Bartoli

Tab. 1. Übersicht über populäre Algorithmen zur Merkmalerkennung (Quelle)

3.4.1. Speeded Up Robust Features (SURF)

Der Speeded Up Robust Features-Algorithmus kann als eine Weiterentwicklung von SIFT verstanden werden. Das Hauptziel bei der Entwicklung von SURF war dabei die Erhöhung der Berechnungsgeschwindigkeit gegenüber SIFT bei gleichzeitiger Beibehaltung von dessen hoher Erkennungsrate.

Eine der beiden Hauptneuerungen stellt der Fast-Hessian Detector dar, der die genaue Berechnung der zweiten Ableitung durch eine Approximation ersetzt. Hierbei bedient man sich Boxfiltern und Integralbildern, um zum gewünschten Ergebnis zu kommen. Das Ergebnis der Approximation ist in Abbildung x zu betrachten.

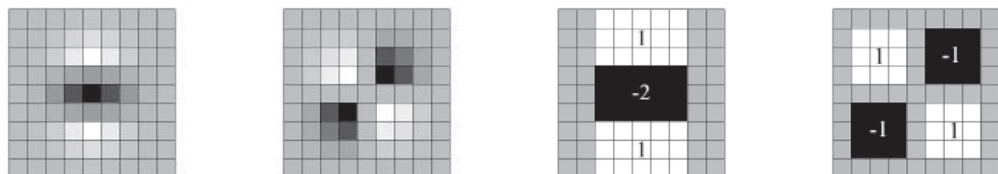


Abbildung x: Gaußsche Ableitungen (links) und deren Approximation durch Boxfilter (rechts) [Bay [et al.](#)]

Als weiterer wichtiger Unterschied zu SIFT kann die Verwendung des neuen SURF-Deskriptors gelten. Um die Komplexität der Berechnung, und damit deren Dauer, zu verringern, wurde die Dimensionalität des Deskriptors verringert. Als erster Schritt werden dabei in einem kreisförmigen Nachbarschaftsbereich um den Keypoint Filterantworten anhand von Haar-Wavelets berechnet. Nachdem man auf diese Weise eine Orientierung ermittelt hat, wird nun eine quadratische Region um den Keypoint festgelegt, die um den Wert der Orientierung rotiert ist (siehe Abbildung x). Diese Region wird nun in Unterregionen mit je 4 x 4 Pixeln geteilt. Für diese wird, ebenfalls unter Verwendung von Haar-Wavelets, ein vierdimensionaler Beschreibungsvektor berechnet. Jeder Keypoint verfügt somit lediglich über einen 64-dimensionalen Deskriptor, während die Dimensionalität von SIFT bei 128 liegt.



Abbildung x: Von SURF verwendete Haar-Wavelets (links) und quadratische Regionen um Keypoints [Bay [et al.](#)]

3.4.2. Binary Robust Invariant Scalable Keypoints (BRISK)

Leutenegger et al. bauen mit ihrem Binary Robust Invariant Scalable Keypoints-Algorithmus auf SIFT und SURF auf. Auch sie sind primär an einer Erhöhung der Berechnungsgeschwindigkeit interessiert, während bezüglich der Treffergenauigkeit lediglich eine Äquivalenz zu SIFT und SURF beabsichtigt wird. Zwar ist bezüglich des Ablaufs des Algorithmus eine signifikante Ähnlichkeit zu SIFT und SURF zu beobachten, es existieren jedoch auch nennenswerte Unterschiede, etwa die Verwendung von FAST zur Ermittlung von Keypoints sowie des binären Deskriptors BRIEF.

Der FAST-Algorithmus (Features from Accelerated Segment Test) von Rosten und Drummond ist im Bereich der Eckendetektion anzusiedeln. Wird ein potentieller Keypoint auf seine Eignung hin untersucht, erfolgt ein Vergleich mit 16 Pixeln, die alle auf einem Kreis um diesen Punkt liegen (siehe Abbildung x). Liegen auf diesem Kreis eine Mindestzahl von zusammenhängenden Pixeln, die allesamt heller oder niedriger als der Mittelpunkt sind, so kann der Punkt als geeignet gelten [Rosten et al.].

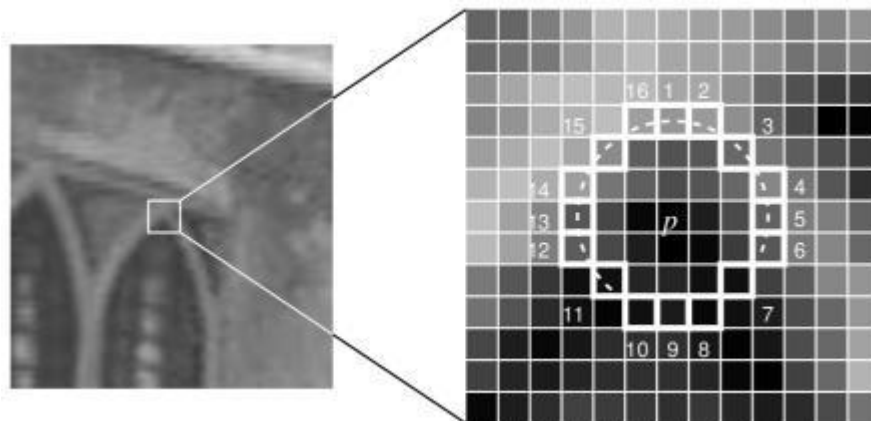


Abbildung x: Identifizierung von Keypoints durch Vergleich mit Kreispunkten [Rosten et al.]

BRISK verwendet nun eine Modifizierung des FAST-Algorithmus namens AGAST, die eine noch weiter erhöhte Geschwindigkeit verspricht. Um die Invarianz gegenüber der Skalierung sicherzustellen, erfolgen die Vergleiche mit den auf dem Kreis liegenden Punkten nicht nur innerhalb eines einzigen Bildes sondern, analog zu SIFT, zusätzlich mit Bildern anderer Oktaven, wobei hier außerdem sogenannten Interoktaven zur Anwendung kommen. [Leutenegger et al.]

Der Deskriptor BRIEF zeichnet sich dadurch aus, dass er Informationen über die zu beschreibenden Merkmale in Form binärer Zeichenketten abspeichert. Dadurch ergeben sich sowohl bei der Generierung als auch beim Matching deutlich Zeiteinsparungen. Statt die Deskriptoren zuerst in herkömmlicher Form

zu generieren und anschließend in Binärcode umzuwandeln, wird dieser bei BRIEF direkt erzeugt. Dabei wird der Keypoint mit einer festen Zahl von Punkten verglichen, die in festen Abständen voneinander auf konzentrischen Kreisen liegen, welche den Keypoint umgeben. Abbildung x zeigt exemplarisch die Anordnung der Punkte im Rahmen von BRISK. Beim Vergleich wird nun untersucht, ob entweder der Keypoint oder der Vergleichspunkt einen höheren Helligkeitswert haben. Ist der Wert des Vergleichspunkts höher, wird im Deskriptor 1 eingetragen, ansonsten 0. Insgesamt hat die binäre Zeichenkette des BRIEF64-Deskriptor eine Länge von nur 512 Bit, was einer achtfachen Verkleinerung gegenüber SIFT und einer vierfachen gegenüber SURF gleichkommt.

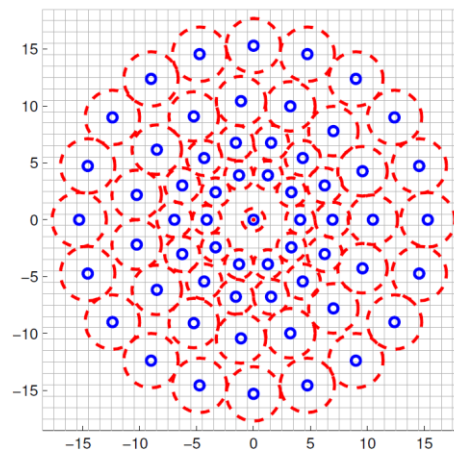


Abbildung x: Von SURF verwendete Haar-Wavelets (links) und quadratische Regionen um Keypoints [Leutenegger et al.]

4. Forschungsstand: Technisch

Alternativen? Selbstentwicklung Beispiele.

OpenCV

Python, Java, JavaScript

Copyright

Performance

5. Forschungsstand: Bildauswahl

Zurich Building Image Database. 1005 Bilder von 201 Gebäuden. Jeweils 5 Bilder pro Objekt. Jeweils leichte Variation von Zoomstufe und Winkel. Aufnahme zur gleichen Zeit mit gleicher Kamera. Keine systematische Variation der Bildeigenschaften.

<http://www.vision.ee.ethz.ch/showroom/zubud/>

„five images were acquired at random arbitrary view points“
“all the images were taken under different seasons, weather conditions and by two different cameras” aber jeweils nur ein Zeitpunkt / eine Kamera für ein Gebäude.

<https://webcache.googleusercontent.com/search?q=cache:oPXu5mapeJoJ:www.vision.ee.ethz.ch/showroom/zubud/report-db.ps+&cd=2&hl=en&ct=clnk&gl=de>

The Oxford Buildings Dataset:

Von Flickr über Queries bezogen, dann manuell mit Qualitätskategorie (Good, OK, Bad, Junk) annotiert. Insgesamt 5062 Bilder für 11 Sehenswürdigkeiten. Bildeigenschaften zufällig bzw. von Flickr-Nutzern bestimmt.

<https://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>

6. Lösungsansatz

7. Umsetzung

8. Evaluierung

9. Diskussion

Ausblick: Nutzer können selbst Bauteile durch Fotos hinzufügen statt diese selbst in Karte (z.B. OSM) hinzufügen zu müssen. Identifizierung durch Position, Richtung und Visibility-Algorithmus.

10. Anhang

11.Literaturverzeichnis

A. Eidesstattliche Erklärung

Ich erkläre hiermit gemäß §17 Abs. 2 APO, dass ich die vorstehende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift