



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Emmanuel Lange

Verfolgung markanter Raumpunkte in
Videobildsequenzen anhand skalierungs- und
rotationsinvarianter Merkmale

Emmanuel Lange

Verfolgung markanter Raumpunkte in
Videobildsequenzen anhand skalierungs- und
rotationsinvarianter Merkmale

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.Ing. Andreas Meisel
Zweitgutachter : Prof. Dr. rer. nat. Reinhard Baran

Abgegeben am 10. Juli 2008

Emmanuel Lange

Thema der Bachelorarbeit

Verfolgung markanter Raumpunkte in Videobildsequenzen anhand skalierungs- und rotationsinvarianter Merkmale

Stichworte

Digitale Bildverarbeitung, SIFT, Skalenraum, markante Raumpunkte, Bildanalyse, Objekterkennung

Kurzzusammenfassung

In dieser Arbeit wird der Aufbau und Ablauf des SIFT-Verfahrens erläutert. Darüber hinaus werden die mit Hilfe dieses Verfahrens gefundenen markanten Raumpunkte in verschiedenen Szenarien auf unterschiedliche Bildeigenschaften hin analysiert.

Emmanuel Lange

Title of the paper

Tracking distinctive image features in video sequences on the basis of scale and rotation invariant keypoints

Keywords

Digital image processing, SIFT, Scale-space, distinctive image features, image analysis, Object recognition

Abstract

In this paper the structure and sequence of the SIFT algorithm will be described. In addition, the distinctive image features found by this approach will be analyzed on the basis of different picture properties.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
1. Einführung	7
1.1. Motivation	7
1.2. Kapitelüberblick	9
2. Digitale Bildverarbeitung	10
2.1. Darstellungsformen für Bilddaten	10
2.1.1. Grauwertbilder	10
2.2. Verfahren zur Bildverbesserung	11
2.2.1. Histogrammausgleich	11
2.2.2. Median-Filter	12
2.2.3. Tiefpassfilter	12
2.3. Verfahren zur Extraktion von Bildinformationen	13
2.3.1. Sobel Operator	13
3. Scale Invariant Feature Transform	14
3.1. Extrema Erkennung im Skalenraum	14
3.1.1. Der Skalenraum	15
3.1.2. Gauß- und Laplace-Pyramiden	16
3.1.3. Erkennung der Extrema	18
3.2. Merkmalspunkt Lokalisierung	20
3.3. Bestimmung der Orientierung	23
3.4. Erzeugen des Merkmalspunkt-Deskriptors	24
4. Analyse	26
4.1. Bildsequenzen	27
4.1.1. Kamerafahrt im Innenraum	27
4.1.2. Kamerafahrt durch einen Wald	29
4.1.3. Verkehrsszene mit ruhender Kamera	31
4.2. Bildpaare	32
4.2.1. Kunstwerk im Foto wiedererkennen	33
4.2.2. Stereobildpaare	35

4.2.3. Gebäude aus verschiedenen Aufnahmestandpunkten	36
4.3. Einfluss durch geänderte Aufnahmeeigenschaften	38
4.3.1. Abweichende Skalierungen	39
4.3.2. Unterschiedliche Betrachtungswinkel	40
4.3.3. Verschiedene Bildhelligkeiten	41
4.4. Fazit	42
5. Schluss	44
Quellenverzeichnis	46
A.	48
B.	49

Abbildungsverzeichnis

2.1. Verfahren zur Bildverbesserung - Histogrammausgleich	12
3.1. Skalenraum - Gauß-Unschärfe	15
3.2. Gauß- und Laplace-Pyramiden - Gaußpyramide	17
3.3. Gauß- und Laplace-Pyramiden - Difference of Gaussian	17
3.4. Gauß- und Laplace-Pyramiden - Gaußpyramide & Difference of Gaussian	19
3.5. Erkennung der Extrema - 26 Nachbarpunkte	19
3.6. Merkmalspunkt Lokalisierung - Klassifikation über den Eigenwert	22
3.7. Erzeugen des Merkmalspunkt-Deskriptors - Gradienten & Deskriptor	24
4.1. Innenraum - Flur markante Raumpunkte	28
4.2. Innenraum - Flur übereinstimmende Raumpunkte	28
4.3. Innenraum - Flur Bewegungspfeile	29
4.4. Wald - Waldboden bewuchs	30
4.5. Wald - Waldboden bewuchs Bewegungspfeile	30
4.6. Verkehrsszene - übereinstimmende Raumpunkte	31
4.7. Verkehrsszene - Bewegungspfeile	32
4.8. Kunstwerk - Gemälde „Les Usines“ markanten Raumpunkte	33
4.9. Kunstwerk - Gemälde „Les Usines“ der Vergleich	34
4.10. Kunstwerk - Gemälde „Junge mit Pfeife“ der Vergleich	34
4.11. Stereobildpaare - Teilbilder mit markanten Raumpunkten	35
4.12. Stereobildpaare - Teilbilder Vergleich	36
4.13. Gebäude - Rathaus markante Punkte	37
4.14. Gebäude - Rathaus Vergleich	37
4.15. Gebäude - Rathaus Vergleich großer Standpunktunterschied	38
4.16. Skalierung - verschiedene Entfernungen	39
4.17. Betrachtungswinkel - verschiedene Winkel	40
4.18. Bildhelligkeit - verschiedene Bildhelligkeiten	41
4.19. Fazit - Rathaus Kontrastunterschied	43

1. Einführung

Der Informatik-Teilbereich maschinelles Sehen (Computer Vision) beschäftigt sich damit die komplexen Vorgänge des Sehens so umzusetzen, dass Maschinen in die Lage versetzt werden zu sehen. Häufig werden Anwendungen im Bereich des maschinellen Sehens für Aufgaben wie: Lokalisierung von Objekten im Bild, Messtechnik zur Ermittlung der Maße von Objekten oder auch zur Identifikation von Objekten in einem Bild benötigt. Mit Hilfe des Verfahrens der Mustererkennung wird versucht logisch zusammengehörige Bildinhalte zu entdecken, diese zu gruppieren und so letztlich abgebildete Objekte (z.B. Buchstaben, Bauteile, Fahrzeuge) zu erkennen.

Um Anwendungen entwickeln zu können, die in der Lage sind Objekte in Bildern und Bildsequenzen wieder zuerkennen bedarf es komplexer Verfahren, welche wichtige Eigenschaften für eine erfolgreiche Wiedererkennung berücksichtigen. Durch diese Verfahren werden markante Raumpunkte aus Bildern extrahiert, welche für die Wiedererkennung von einzelnen Objekten genutzt werden können. Damit Objekte zuverlässig wiedererkannt werden können ist es wichtig, dass veränderliche Parameter beziehungsweise Bildeigenschaften bei der Erzeugung einer Beschreibung eines markanten Raumpunktes besonders beachtet werden. So dass die Erkennung möglichst unabhängig gegenüber solchen Änderungen erfolgen kann.

1.1. Motivation

Ein Verfahren, welches in der Lage sein soll zuverlässig Objekte wieder zuerkennen, sollte möglichst invariant gegenüber Änderung der Skalierung und Rotation sein. Außerdem sollte es möglichst unempfindlich auf Änderungen der Helligkeit und auf Abweichungen beim Aufnahmestandpunkt reagieren. Die Invarianz gegenüber der Skalierung ist wichtig, wenn die Aufnahmen des wiederzuerkennenden Objektes in verschiedenen Entfernungen entstehen. Hierbei ist es wichtig, dass gesuchte Objekte auch erkannt werden können, wenn sie im Vergleich zur „Original“-Aufnahme, von der die markanten Raumpunkte ermittelt und gespeichert wurden, größer oder kleiner skaliert im aktuell untersuchendem Bild vorkommen. Die Invarianz gegenüber Rotation kommt immer dann zum Tragen, wenn das wiederzuerkennende Objekt sich in einer veränderten Lage befindet. Dies kann zum Beispiel der Fall

sein, wenn sich das gesuchte Objekt in einem Bild senkrecht stehend befindet und im anderem Bild waagrecht liegend zu finden ist. Darüber hinaus ist es wünschenswert, dass das Verfahren markante Raumpunkte möglichst unempfindlich gegenüber Änderung der Bildhelligkeit wiedererkennen kann. Diese Eigenschaft spielt eine besonders wichtige Rolle, wenn Aufnahmen von Objekten verglichen werden sollen, welche bei unterschiedlichen Beleuchtungsverhältnissen aufgenommen wurden. Ein weiterer Aspekt, den es zu beachten gibt, ist der Aufnahmestandpunkt. Es soll möglich sein Objekte in Aufnahmen auch dann wiedererkennen zu können, wenn diese mit Objekten aus Aufnahmen mit abweichendem Aufnahmestandpunkt verglichen werden.

Ein Verfahren, welches verspricht diesen Anforderungen gerecht zu werden, ist das Scale Invariant Feature Transform (SIFT) Verfahren. Das SIFT-Verfahren, mit welchem Objekte in Bildern identifiziert und wiedererkannt werden können, wurde von David Lowe entwickelt und in dessen wissenschaftlicher Abhandlung „Distinctive Image Features from Scale-Invariant Keypoints“ (Lowe, 2004) beschrieben. Dieses Verfahren verspricht in nahezu Echtzeit-Verarbeitung sehr respektable Ergebnisse erzielen zu können.

Im Rahmen dieser Bachelorarbeit ist eine Anwendung entstanden, welche auf Basis des von David Lowe entwickelten SIFT-Verfahrens Kameraaufnahmen auswertet. Es werden hierbei markante Bildpunkte aus den einzelnen Bildern extrahiert, so dass diese in Folgebildern von Bildsequenzen oder in anderen Aufnahmen wiedererkannt werden können. Bei der Implementierung handelt es sich um ein Makro, welches für Impresario entwickelt wurde. Impresario ist ein Bildverarbeitungssystem, welches die Entwicklung und die Bewertung der entwickelten Algorithmen wesentlich vereinfacht. Es stellt eine Schnittstelle zur Anwendungsprogrammierung zur Verfügung, bei der man unter anderem auf eine grafische Benutzeroberfläche zurückgreifen und auf eine einfache Art und Weise einen selbst entwickelten Programmcode, die so genannten „Makros“, einbinden kann. Die eigentliche Implementierung des SIFT-Algorithmuses basiert hierbei auf der SIFT++ Implementierung von Andrea Vedaldi vom UCLA VisionLab (Vedaldi, 2006).

Ziel dieser Arbeit ist es, den Aufbau und Ablauf des SIFT-Verfahrens zu erläutern sowie die Analyse der durch das Verfahren gelieferten Ergebnisse. Hierbei wird das SIFT-Verfahren anhand mehrerer verschiedener Szenarien auf die Auswirkungen der Bildinhalte auf das zuverlässige Liefern von korrekten Ergebnissen untersucht. Darüber hinaus wird untersucht inwieweit sich die veränderlichen Eigenschaften auf die Wiedererkennung auswirken.

1.2. Kapitelüberblick

Diese Arbeit gliedert sich in drei Hauptkapitel, welche sich thematisch mit den allgemeinen Grundlagen der digitalen Bildverarbeitung, dem SIFT-Verfahren und dessen Analyse beschäftigen.

Das Kapitel über die Grundlagen der digitalen Bildverarbeitung befasst sich mit dem grundlegenden Aufbau von digitalen Bildern sowie mit dem Aufbau von Grauwertbildern. Darüber hinaus gibt es eine kurze Einführung in ausgewählte Verfahren zur Bildverbesserung und zur Extraktion von Bildinformationen.

Der folgende Abschnitt beschäftigt sich intensiv mit dem SIFT-Verfahren. Es wird Schritt für Schritt erläutert, wie die markanten Raumpunkte in diesem Verfahren gesucht, näher bestimmt und in Deskriptoren beschrieben werden. Dabei wird ein besonderes Augenmerk auf die Erklärung der einzelnen in diesem Verfahren verwendeten Algorithmen gelegt.

Das letzte Kapitel des Hauptteils beschäftigt sich mit der Analyse, der mittels des SIFT-Verfahrens gewonnenen markanten Raumpunkte. Hierbei werden besonders die Auswirkungen von unterschiedlichen Bildinhalten und dessen möglichen Transformation auf die Wiedererkennbarkeit der markanten Punkte untersucht. Es werden sowohl Bilder als auch Bildsequenzen analysiert.

Abschließend wird noch einmal erläutert in welchen Bereichen das SIFT-Verfahren seine Stärken beziehungsweise Schwächen besitzt und an welchen Stellen vielleicht noch einmal weitere Tests durchgeführt werden sollten. Darüber hinaus wird auch ein Ausblick auf die verschiedenen Einsatzmöglichkeiten dieses Verfahrens gegeben.

2. Digitale Bildverarbeitung

Bevor wir uns verschiedenen Operationen widmen, die auf digitale Bilder angewandt werden können, wollen wir uns erst einmal einen grundlegenden Überblick über den Aufbau von digitalen Bildern verschaffen. Bilder lassen sich in Form einer Matrix beschreiben. Diese Bildmatrix ist so aufgebaut, dass die Zählung der Spalten und Zeilen oben in der linken Ecke der Matrix mit 0 beginnt. Spalten, meistens die mit x gekennzeichnete Achse, werden dann von links nach rechts gezählt und Zeilen, die mit y beschriftete Achse, von oben nach unten. Die einzelnen Elemente der Bildmatrix werden Bildpunkte oder auch Pixel genannt. Jeder Bildpunkt wird durch eine bestimmte Position, (x, y) , in der Matrix und einem zu diesem Bildpunkt gehörenden Wert gekennzeichnet, der sogenannte Pixelwert. Die in diesem Wert gespeicherten Informationen hängen vom verwendeten Grafikformat ab, welches unterschiedliche Farbräume nutzt. Diese Farbräume werden in Farbmodellen beschrieben.

2.1. Darstellungsformen für Bilddaten

Je nach Anforderungen und Einsatzzweck werden verschiedene Grafikformate zur Darstellung von Bildern eingesetzt. So kommen zum Beispiel Grauwertbilder häufig zum Einsatz, wenn es um das Auffinden von Kanten in Bildern geht. Bei Grauwertbildern würde zum Beispiel jeder einzelne Pixelwert einer Bildmatrix aus einem Dezimalwert bestehen, welcher den Grauwert an der Position des Bildpunktes angibt. Im Folgendem wird der Aufbau von Grauwertbildern näher erläutert.

2.1.1. Grauwertbilder

Grauwertbilder sind Bilder, welche ihren Inhalt mittels verschiedener Grauwerte darstellen. Jedem Bildpunkt wird hierbei ein Grauwert zugeordnet, der entweder Schwarz, Weiß oder einem Grau in verschiedenen Abstufungen entspricht. Typischerweise besteht die Grauwertmenge aus 256 (8 Bit) verschiedenen Grauwerten, von 0 bis 255. Wobei in der Regel der Grauwert 255 als Weiß interpretiert wird, der Wert 0 als Schwarz und alle dazwischen liegenden Werte, die Abstufungen zwischen Weiß und Schwarz, als Graustufen. Wenn ein Bild

nur aus schwarzen und weißen Bildpunkten besteht, also die Menge der Grauwerte nur zwei Werte (1 Bit) umfasst, spricht man von einem Binärbild.

2.2. Verfahren zur Bildverbesserung

Digitale Bildverbesserungen werden benötigt, um digitale Bilder entweder für den menschlichen Betrachter aufzubereiten oder als Vorverarbeitungsschritt für eine später folgende Extraktion von Bildinformationen. Bildverbesserungen, die durchgeführt werden um Bilder für das menschliche Auge aufzubereiten, haben meistens das Ziel die Bilder so zu verändern, dass der Betrachter alle Details, die im Bild enthalten sind, möglichst optimal erkennen kann und dies möglichst ohne störende Elemente im Bild, wie zum Beispiel Bildrauschen. Bildrauschen ist eine Störung der Bildinformationen, bei der fehlerhafte Pixel vorhanden sind, die entweder eine falsche Farbe oder Helligkeit aufweisen. Wenn Bildverbesserungen als Vorverarbeitungsschritt für die Extraktion von Bildinformationen durchgeführt werden, haben diese meistens den Zweck das jeweilige Bild so vorzubereiten, dass folgende Operationen zur Extraktion von Bildinformationen, zum Beispiel zum Erkennen von Kanten in einem Bild, möglichst optimale Bedingungen vorfinden, die das Extrahieren von Informationen entweder überhaupt erst möglich machen oder es vereinfachen. Im Folgendem werden ein paar ausgewählte Verfahren zur Bildverbesserung vorgestellt.

2.2.1. Histogrammausgleich

Histogramme sind eine Möglichkeit die Häufigkeitsverteilung von Werten darzustellen. Beim Histogrammausgleich wird ein Histogramm verwendet, welches die Verteilung der Helligkeitswerte widerspiegelt. Liegt zum Beispiel ein Grauwertbild vor, welches sehr „dunkel“ ist, also nicht den gesamten Grauwertbereich ausnutzt, sondern nur die dunkleren Töne, so kann man mittels eines Histogrammausgleichs eine optimalere Ausnutzung des zur Verfügung stehenden Grauwertbereiches erreichen. Hierbei werden die Werte dann so neu verteilt, dass der gesamte Wertebereich ausgenutzt wird und die Werte, die am häufigstem vorkommen sich möglichst von den anderen Werten absetzen. Dadurch wird erreicht, dass nach der Bearbeitung des Bildes viel mehr Details in diesem zu erkennen sind als vorher.

Abbildung 2.1 zeigt deutlich, wie sich der Gesamteindruck des ursprünglich sehr dunklen Bildes auf der rechten Seite nach dem Histogrammausgleich deutlich verbessert hat. Das Ergebnis ist im Bild in der Mitte zu sehen. Man sieht anhand der beiden Histogramme sehr schön, wie die Grauwertverteilung vom „dunklen“ Bild, das Histogramm rechts oben, durch den Histogrammausgleich auf den ganzen Wertebereich, im Histogramm rechts unten gut zu erkennen, ausgedehnt wurde.



Abbildung 2.1.: Links das „dunkle“ Bild mit dem zugehörigen Histogramm rechts oben. Das mittlere Bild zeigt das Ergebnis nach dem Histogrammausgleich, das dazugehörige Histogramm ist rechts unten.

2.2.2. Median-Filter

Der Median-Filter ist eine Operation im Ortsbereich. Operationen, die sich nicht nur auf einen Bildpunkt beziehen, sondern auch Informationen von Bildpunkten nutzen, die einen ausgewählten Bildpunkt umgeben, werden Operationen im Ortsbereich genannt. Dies bedeutet, dass zum Beispiel bei einer Operation nicht nur der Grauwert des aktuell ausgewählten Bildpunktes, $f(x, y)$ in das Ergebnis mit einfließt, sondern auch die Grauwerte der benachbarten Bildpunkte, zum Beispiel des Bildpunktes rechts ($f(x + 1, y)$) vom aktuell untersuchten Punkt, mit zu dem Ergebnis beitragen.

Der Median-Filter wird zum Beispiel eingesetzt um „Salt and pepper noise“ aus Bildern zu entfernen. Hierbei handelt es sich um vereinzelt auftretende fehlerhafte weiße und schwarze Pixel, die sich im Bild befinden können. Bei diesem Filter werden die Grauwerte in der Umgebung eines Bildpunktes der Größe nach aufsteigend sortiert. In den meisten Fällen befinden sich die Werte in der Umgebung einer quadratischen Maske um den Bildpunkt. Die Maske hat eine ungerade Spalten- und Zeilenanzahl, zum Beispiel 3x3 oder 5x5. Der Zielgrauwert, also der Grauwert, der dem aktuell betrachteten Bildpunkt zugewiesen wird, wird aus den sortierten Grauwerten der Umgebung ermittelt. Der Wert, der in der Mitte der aufsteigend sortierten Werte steht, wird dem aktuellen Bildpunkt als Grauwert zugewiesen.

2.2.3. Tiefpassfilter

Bei dieser Filterung werden die hochfrequenten Anteile eines Bildes reduziert und/oder unterdrückt. Dadurch erscheint ein tiefpassgefiltertes Bild unschärfer als das Original. Im Bereich der Bildverarbeitung wird hierbei auch oft von einem „geglätteten“ Bild gesprochen. Hohe Ortsfrequenzen treten im Originalbild vor allem in den Grauwertkanten auf (vgl. Nieschwitz u. a., 2007, S. 535). Somit werden vor allem die Kanten in einem Bild unschärfer.

Würde man einen „idealen“ Tiefpassfilter auf ein Bild anwenden, so würden Frequenzanteile, die einen bestimmten Grenzwert überschreiten, genau ab diesem Grenzwert ausgefiltert. Dieser „harte“ Übergang würde aber für unschöne Artefakte im gefiltertem Bild sorgen. Bei Verwendung eines Gauß-Tiefpasses umgeht man diese. Beim Gauß-Tiefpass wird nicht „hart“ nach überschreiten eines Grenzwertes ausgefiltert, sondern es gibt einen eher „weichen“ Übergang. Dies bedeutet, dass die hohen Frequenzanteile umso mehr ausgefiltert werden, je mehr sie sich von der jeweiligen Filtergröße („Grenze“) entfernen. Sie werden also immer schwächer gewichtet. Tiefpassfilter werden zum Beispiel eingesetzt um Bildrauschen zu reduzieren.

2.3. Verfahren zur Extraktion von Bildinformationen

Verfahren zur Extraktion von Bildinformationen werden immer dann eingesetzt, wenn ein Interesse an der Gewinnung von bestimmten Informationen an einem Bild, vorliegt. Inhalte, die von Interesse sind, können zum Beispiel Kanten in einem Bild sein, einzelne Objekte, die separiert werden sollen für eine Zählung oder das Extrahieren von bestimmten Merkmalen eines Bildes, die zur Wiedererkennung von Objekten genutzt werden können. Komplexere Vorgänge, wie zum Beispiel das Bestimmen von Bildmerkmalen zur Wiedererkennung, erfordern oftmals das Anwenden vieler Verarbeitungsschritte hintereinander um adäquate Ergebnisse erzielen zu können. Im Folgendem wird ein Verfahren zur Erkennung von Kanten beschrieben.

2.3.1. Sobel Operator

Der Sobel-Operator wird zur Filterung von Kanten eingesetzt. Kanten werden bei diesem Filter über eine starke Änderung des Kontrastes, der Unterschied zwischen dunklen und hellen Bildbereichen, detektiert. Diese Kontraständerungen werden mit Hilfe von Gradienten bestimmt. Ein Gradient ist ein Vektor, der in die Richtung des größten Funktionswertes zeigt und dessen Länge ein Maß für den Anstieg des Funktionswertes ist. Für jeden Bildpunkt zeigt der Gradient also an, wie stark die stärkste Kontraständerung ist und in welcher Richtung diese liegt; vom aktuell betrachteten Bildpunkt aus gesehen.

3. Scale Invariant Feature Transform

Das „Scale Invariant Feature Transform (SIFT)“ Verfahren von David Lowe besitzt viele Eigenschaften, die für den Vergleich von verschiedenen Bildern eines Objektes oder einer Szene sehr wichtig sind. Die markanten Raumpunkte, die mit Hilfe des SIFT-Verfahrens gefunden werden, sind invariant gegen Änderungen der Skalierung sowie auch gegen Rotationen. Darüber hinaus sind die Punkte relativ robust gegen Änderungen der Helligkeit des Bildes sowie gegen Veränderungen des Aufnahmezeitpunktes. Das Verfahren setzt dabei auf effiziente Algorithmen und minimiert die benötigte Rechenzeit für das Extrahieren der Schlüsselpunkte durch eine geschickt gewählte Schachtelung der einzelnen Schritte der Filterung. Aufwendige Operationen werden nur auf Bildbereiche angewandt, welche vorher einen initialen Test bestanden haben. Die Berechnung der markanten Raumpunkte und der Deskriptoren erfolgt in vier Hauptschritten:

1. **Extrema Erkennung im Skalenraum:** Im ersten Schritt wird über alle Bildbereiche und Skalierungen hinweg nach potentiellen Schlüsselpunkt-Kandidaten gesucht.
2. **Merkmalspunkt Lokalisierung:** Von jedem Kandidaten wird die genaue Skalierung und Position ermittelt. Die einzelnen Schlüsselpunkte werden dabei anhand ihrer Stabilitätsmerkmale ausgewählt oder verworfen.
3. **Bestimmung der Orientierung:** Jeder Schlüsselpunkt-Position werden eine oder mehrere Orientierungen zugewiesen. Die einzelnen Orientierungen basieren dabei auf den lokalen Gradienten-Richtungen.
4. **Erzeugen des Merkmalspunkt-Deskriptors:** Abschließend werden die Deskriptoren erzeugt, welche die markanten Eigenschaften der Schlüsselpunkte beschreiben.

3.1. Extrema Erkennung im Skalenraum

Wie bereits in der Einleitung beschrieben, wird in einem ersten Schritt nach Schlüsselpunkt-Kandidaten im Skalenraum gesucht. Hierfür ist es notwendig, dass verschiedene Repräsentationen der Eingangsbilddaten erstellt werden. Aus den Bilddaten wird dabei zuerst eine Gaußpyramide errechnet und auf dessen Bildern aufbauend im Folgendem eine Laplacepyramide erstellt. Die einzelnen Bilder der Laplacepyramide bilden hierbei die Basis

für die Suche nach Schlüsselpunkt-Kandidaten. Die folgenden Abschnitte erläutern diesen Prozess von der Erstellung der Gauß- und Laplace-Pyramide bis hin zur Erkennung der Schlüsselpunkt-Kandidaten.

3.1.1. Der Skalenraum



Abbildung 3.1.: Beispiel für die Gauß-Unschärfe: Auf der linken Seite ist das Originalbild zu sehen und auf der rechten Seite das Bild nach Anwendung des Glättungskerns.

Es ist wünschenswert möglichst viele Informationen aus den Daten eines Bildes zu gewinnen. Der Skalenraum (Scale-space) ermöglicht es verschiedene Repräsentationen eines Bildes zu erzeugen. Der Scale-space repräsentiert hierbei die Eingangsbilddaten in verschiedenen Skalierungen. Durch die mehrfache Anwendung eines Glättungskerns auf das Eingangsbild entsteht eine Reihe von Bildern, die immer mehr geglättet werden. Die unterschiedliche Glättung der einzelnen Bilder erlauben es in den weniger geglätteten Bildern feine Strukturen zu erkennen, wo hingegen in den mehr geglätteten Bildern gröbere Strukturen zu erkennen sind. Die Suche nach markanten Raumpunkten im Scale-space ist ein wichtiger Teil des SIFT-Verfahrens, denn die so gefundenen Punkte sind invariant gegen Skalierungsunterschiede. Darüber hinaus werden durch die Glättung auch Bildfehler, wie zum Beispiel Bildrauschen unterdrückt, so dass diese bei einer späteren Weiterverarbeitung nicht mehr groß ins Gewicht fallen.

Das SIFT-Verfahren nutzt als Glättungskern zur Erstellung des Skalenraums die Gauß-Funktion, welche die Bilder tiefpass filtert. Dieser kann sehr effizient implementiert werden um den Rechenaufwand deutlich zu reduzieren. Des Weiteren besitzt er die wichtige Eigenschaft, dass die Filterwirkung in alle Richtungen gleich ist, so dass durch die Glättung keine neuen Strukturen entstehen (vgl. Lindeberg, 1994, S. 3). Somit werden nachfolgende Bildverarbeitungsschritte nicht durch den Filter beeinflusst. Abbildung 3.1 zeigt das Ergebnis der Anwendung des Gaußfilters auf ein Bild. Im rechten Bild ist deutlich sichtbar, dass nach der Anwendung des Filters die feinen Strukturen fast nicht mehr zu erkennen sind.

Der Skalenraum eines Bildes wird definiert als Funktion, $L(x, y, \sigma)$, diese ist das Produkt der Faltung eines variabel skalierten Gaußfilters, $G(x, y, \sigma)$, mit dem Eingangsbild, $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

wobei $*$ die Faltungsoperation in x und y darstellt, und

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.2)$$

(vgl. Lowe, 2004, S. 5).

3.1.2. Gauß- und Laplace-Pyramiden

Die Gauß- und Laplace-Pyramide sind Techniken in der Bildverarbeitung um Bilder mit geringem Rechenaufwand zu filtern. Es wird hierbei von Pyramiden gesprochen, weil die durch die Algorithmen erstellten Bilder im Laufe der Verarbeitung immer wieder in Zeilen- und Spalten-Anzahl halbiert werden. Betrachtet man diese immer wieder verkleinerten Bilder übereinander gelegt, beginnend mit dem Größten, so ergibt sich die Form einer Pyramide.

In einem ersten Schritt wird die Gaußpyramide erstellt. Mittels der im vorhergehenden Abschnitts bereits kennengelernten Skalenraumrepräsentation von Bildern, wird eine Reihe von Bildern erzeugt, die immer unschärfer werden. Die erste Stufe (Oktave) der Pyramide wird durch das Eingangsbild gebildet, welches als Grauwertbild vorliegt. Alle darauf folgenden Stufen, werden durch ein iteratives Verfahren erzeugt: Hierbei wird auf die Bilder der vorherigen Stufe der Glättungsoperator angewandt und das so entstandene Bild in Zeilen- und Spalten-Anzahl halbiert. Die auf diese Weise neu entstehenden Bilder sind flächenmäßig immer ein viertel kleiner als das vorhergehende Bild. Der iterative Vorgang wird entweder solange fortgesetzt bis die Bildgröße 3×3 oder 2×2 Pixel erreicht ist oder aber eine vorher definierte anders große minimale Bildgröße erreicht wird. Das SIFT-Verfahren benutzt eine leicht abgewandelte Version der Gaußpyramide. Anstatt wie bei der vorher kennengelernten normalen Gaußpyramide, die in jeder Stufe nur ein Bild enthält, verwendet es mehrere Bilder pro Oktave. Die weiteren Bilder, die pro Oktave benötigt werden, entstehen durch Anwendung des Glättungsoperators auf das erste Bild der jeweiligen Oktave und auf die in der Folge daraus entstehenden Bilder. Diese Bilder innerhalb einer Oktave unterscheiden sich durch die zunehmende Unschärfe, wo hingegen die Größe der Bilder gleichbleibt. Abbildung 3.2 zeigt beispielhaft 3 Oktaven einer solchen Gaußpyramide. Jede Oktave enthält in diesem Beispiel vier Bilder, die mit einem Glättungsoperator bearbeitet wurden, der als Glättungskern die Gauß-Funktion nutzt. Die Bilder werden in diesem Beispiel in jeder Oktave von oben nach unten immer unschärfer. Die beiden Folgeoktaven auf der rechten Seite weisen jeweils eine Bildgröße auf, die einem Viertel der vorherigen Oktave entspricht. Um zu gewährleisten,



Abbildung 3.2.: Die im SIFT-Verfahren verwendete Gaußpyramide mit mehreren Bildern pro Stufe.

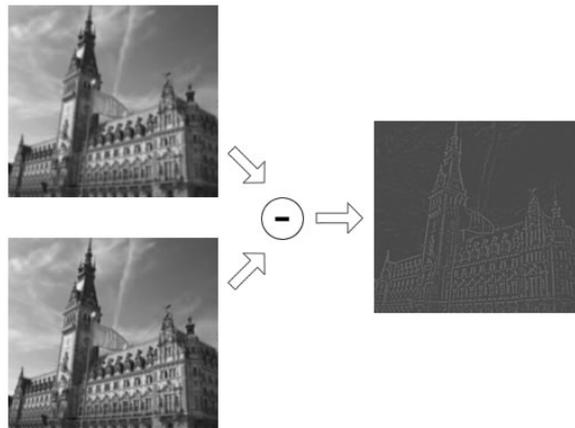


Abbildung 3.3.: Differenzenbildung zweier übereinander liegender gaußgefilterter Bilder.

dass die in einem der später folgenden Schritte gefundenen markanten Raumpunkte auch wirklich skalierungsinvariant sind, unterscheiden sich die einzelnen Bilder einer Oktave in ihrer Skalierung um einen konstanten Faktor (vgl. Lowe, 2004, S. 6). Die Laplacepyramide, die im folgendem beschrieben wird, setzt diesen konstanten Faktor voraus.

Nachdem die Gaußpyramide konstruiert wurde, wird auf dessen Bildern aufbauend die Laplacepyramide erstellt. Der LoG (Laplace of Gaussian) -Filter wird in der Bildverarbeitung zur Kantenerkennung eingesetzt. Das Bild wird dabei zuerst Tiefpass gefiltert um Bildrauschen zu unterdrücken. Anschließend wird es als „Grauwertgebirge“ betrachtet, hierbei wird das Bild als Funktion $f(x, y)$ aufgefasst, bei der die Funktionswerte in hellen Bildbereichen am Höchsten sind. Wird das Bild auf diese Weise betrachtet, so deuten Wendepunkte in diesem „Grauwertgebirge“ auf Kanten hin. Wendepunkte werden durch das Auffinden von Nulldurch-

gängen (Zero-crossings) in der zweiten Ableitung einer Funktion erkannt. Eine gute Näherung des LoG-Filters stellt der ‘DoG (Difference of Gaussian)’-Algorithmus dar, welcher sich wesentlich effizienter berechnen lässt, weil er auf die Bilder der bereits bestehenden Gaußpyramide zurückgreift.

Die einzelnen Stufen der Laplacepyramide werden durch Differenzbildung der übereinander liegenden Bilder der Gaußpyramide gebildet, der sogenannte DoG-Algorithmus. Die unterschiedlichen Stufen der Gaußpyramide unterscheiden sich in ihrer Größe, somit müssen vor der Differenzenbildung die Bilder auf dieselbe Größe gebracht werden. Hierbei wird immer das kleinere Bild auf die Größe des Bildes der nächst höheren Stufe gebracht. Abbildung 3.3 zeigt das Ergebnisbild der Differenzenbildung zweier übereinander liegender Bilder einer Oktave der Gaußpyramide.

Beim SIFT-Verfahren werden zur Differenzenbildung die Bilder verwendet, die sich innerhalb einer Oktave der Gaußpyramide befinden. Die einzelnen DoG-Bilder, $D(x, y, \sigma)$, entstehen durch das jeweilige Subtrahieren zweier dieser benachbarten Bilder einer Oktave, welche sich um einen konstanten Multiplikationsfaktor, k , welcher die Stärke der Unschärfe festlegt, unterscheiden:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (3.3)$$

Diese immer unschärfer werdenden Bilder einer Oktave haben bereits die benötigte Bildgröße. Hieraus ergeben sich auch bei der Laplacepyramide wieder mehrere Bilder pro Oktave, wie es auch schon bei der Gaußpyramide der Fall war. Es wird aus jeder Oktave der Gaußpyramide die Differenz aus den jeweils übereinander liegenden Bildern gebildet und die Ergebnisbilder in der entsprechenden Oktave der Laplacepyramide gespeichert. Abbildung 3.4 zeigt den Aufbau der Gauß- und Laplace-Pyramide beim SIFT-Verfahren. Die benachbarten Bilder jeder Oktave der Gaußpyramide werden voneinander subtrahiert, um die auf der rechten Seite gezeigten DoG-Bilder zu erzeugen.

3.1.3. Erkennung der Extrema

Die potentiellen markanten Merkmalspunkte eines Bildes werden im Skalenraum des Bildes gesucht. Hierbei werden alle Bilder, die die Laplacepyramide enthält, die einzelnen DoG-Bilder, auf lokale Extrema überprüft. Jeder Bildpunkt wird dabei mit seinen 26 Nachbarn verglichen. Acht dieser Nachbarn befinden sich im selben Bild, diese umgeben den aktuell untersuchten Bildpunkt. Neun Nachbarn befinden sich jeweils in den Bildern (Skalierungen) über beziehungsweise unter dem aktuellen Bild. Diese Nachbarn sind auch wieder die Punkte, die den aktuell untersuchten Punkt oberhalb beziehungsweise unterhalb der aktuellen

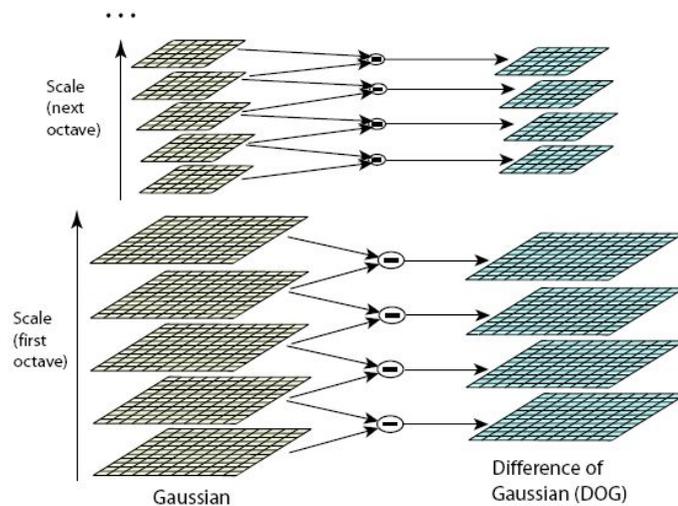


Abbildung 3.4.: Aufbau der Gauß- und Laplace-Pyramide. [Lowe 2004]

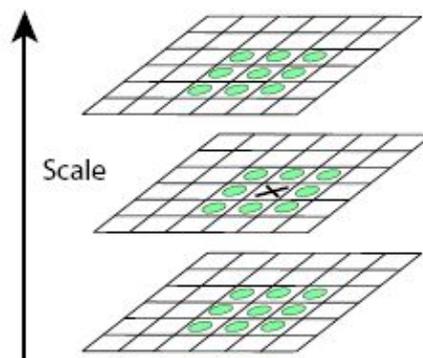


Abbildung 3.5.: Potentieller markanter Merkmalspunkt (mit X gekennzeichnet) und seine 26 Nachbarn. [Lowe 2004]

Skalierung umgeben. Abbildung 3.5 zeigt den aktuell untersuchten Bildpunkt, mit X markiert, und dessen acht Nachbarn im selben Bild, sowie die neun Nachbarn, die sich jeweils in der Skalierung über und unter der des untersuchten Punktes befinden. Ein untersuchter Bildpunkt kommt nur dann als markanter Merkmalspunkt-Kandidat in Frage, wenn er ein Extremum in seiner 26 Nachbarschaft darstellt. Der untersuchte Punkt muss also größer oder kleiner als alle seine 26 Nachbarn sein um ausgewählt zu werden. Das SIFT-Verfahren setzt dabei voraus, dass die einzelnen Pixel des Eingangsbildes einen Graustufenwert zwischen $[0, 1]$ aufweisen. Die benötigte Rechenzeit für diese Vergleiche ist relativ gering, da in den meisten Fällen ein untersuchter Punkt schon nach den ersten paar Vergleichen mit seinen

Nachbarn nicht mehr als markanter Merkmalspunkt in Frage kommt, weil er nicht größer oder kleiner als alle bis dahin untersuchten Nachbarn ist. Somit können die Vergleiche für die einzelnen Bildpunkte meistens schon frühzeitig abgebrochen werden (vgl. Lowe, 2004, S. 7).

3.2. Merkmalspunkt Lokalisierung

In einem ersten Schritt wurden die möglichen Kandidaten für markante Raumpunkte durch den Vergleich der einzelnen Pixel mit seinen Nachbarn gefunden. Im Folgendem werden die einzelnen Schlüsselpunkt-Kandidaten einer genaueren Untersuchung unterzogen. Hierbei wird zuerst die genaue Position des Schlüsselpunkt-Kandidaten bestimmt. Darauf folgt dann eine Überprüfung des Punktes auf einen möglichen zu schwachen Kontrast und ob sich der Kandidat auf einer schlecht wiederzuerkennenden Kante befindet.

Es wird für jeden Schlüsselpunkt-Kandidaten die genaue Position bestimmt. Dies ist notwendig, weil die Möglichkeit besteht, dass ein Merkmalspunkt zwischen Bildpunkten liegt. Die in der Laplacepyramide gefundenen Extrema, in ihrer jeweiligen 26 Nachbarschaft, können eine Abweichung von ± 1 in jeder Dimensionen haben (vgl. Brown und Lowe, 2002, S. 3). Um die genaue Position der einzelnen Merkmalspunkte bestimmen zu können, wird die quadratische Taylorreihe, diese wird als Näherung verwendet um die Funktion in der Umgebung des jeweiligen Schlüsselpunkt-Kandidaten anzunähern, mit den jeweiligen Kandidaten als Entwicklungspunkt, angewandt:

$$D(\vec{x}) = D + \frac{\partial D^T}{\partial \vec{x}} \vec{x} + \frac{1}{2} \vec{x}^T \frac{\partial^2 D}{\partial \vec{x}^2} \vec{x} \quad (3.4)$$

wobei $x = (x, y, \sigma)^T$ die Skalenraum-Koordinate und $D(x, y, \sigma)$ die Skalenraum-Funktion, die durch die „Difference of Gaussian“-Funktion approximiert Laplace-Funktion darstellen. Diese Gleichung ist im Anhang A auch noch einmal in ausführlicher Schreibweise hinterlegt um zu verdeutlichen wie komplex diese Berechnung im nicht diskreten Fall ist. Durch die Ableitung der Funktion, $D(x, y, \sigma)$, nach \vec{x} wird die Position des Extremums, \hat{x} , bestimmt:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial \vec{x}^2} \frac{\partial D}{\partial \vec{x}}. \quad (3.5)$$

Wenn der Abstand \hat{x} größer als 0,5 in irgendeiner Dimension ist, bedeutet dies, dass das Extremum näher an einem anderen Bildpunkt liegt als an dem aktuell betrachteten. In diesem Fall wird der zu untersuchende Bildpunkt geändert und die Interpolation wird für den neuen Punkt durchgeführt. Der nach dieser Berechnung erhaltene Abstand \hat{x} wird zur Position des Bildpunktes addiert, um so die interpolierte Position des Schlüsselpunkt-Kandidatens zu erhalten.

Der Funktionswert an der Position des Extremums, $D(\hat{x})$, kann darüber hinaus genutzt werden um unbeständig Extrema mit geringem Kontrast zurück zu weisen. Den Funktionswert erhält man durch das Einsetzen von Gleichung 2.5 in 2.4:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \bar{x}} \hat{x}. \quad (3.6)$$

Wenn ein Schlüsselpunkt-Kandidat einen Kontrast aufweist, der geringer ist als ein festgelegter Schwellenwert, so wird dieser Kandidat abgelehnt. In der wissenschaftlichen Abhandlung (vgl. Lowe, 2004, S. 11) von D. Lowe werden alle Extrema verworfen, die einen Graustufenwert kleiner als 0,03 aufweisen, $|D(\hat{x})| < 0,03$.

Um stabile Schlüsselpunkte zu erhalten, reicht es nicht aus nur die Kandidaten auszusortieren, welche einen geringen Kontrast haben. Die „Difference of Gaussian“-Funktion zeigt starke Reaktionen auf Kanten, selbst wenn diese nur sehr schwach sind. Punkte, die an schlecht wiederzuerkennenden Kanten liegen, reagieren besonders anfällig auf Bildstörungen und sind somit schlecht geeignet für die Wiedererkennung. Um solche Schlüsselpunkt-Kandidaten aussortieren zu können, wird der Ecken- und Kantendetektor von Harris und Stephens (1988) verwendet.

Diese Kandidaten können durch Untersuchung der Krümmungen aussortiert werden. Punkte, die sich an schlecht wiederzuerkennenden Kanten befinden, besitzen eine große Hauptkrümmung entlang der Kante, aber eine kleine in senkrechter Richtung. Die Hauptkrümmungen können mit Hilfe einer 2×2 Hesse-Matrix, H , an der jeweiligen Position und Skalierung des untersuchten Schlüsselpunkt-Kandidaten bestimmt werden:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.7)$$

Die Eigenwerte von H stehen dabei im selben Verhältnis zueinander wie die Hauptkrümmungen von D . Wie Harris und Stephens (1988) festgestellt haben, kann man das explizite Berechnen der Eigenwerte in diesem Fall umgehen, indem man das Verhältnis der Eigenwerte zueinander betrachtet. Eigenwerte und Eigenvektoren einer $n \times n$ -Matrix \underline{A} :

Eigenvektoren, \vec{x}_e , einer Abbildung in der linearen Algebra sind Vektoren, welche durch eine Abbildung ihre Richtung nicht verändern. Diese Eigenvektoren müssen vom Nullvektor verschieden sein.

Eigenwert, α , wird der Faktor genannt um welchen der Eigenvektor gestreckt wird: $\underline{A} \vec{x}_e = \alpha \vec{x}_e$ (vgl. Papula, 2001, S. 122).

Es wird festgelegt, dass α den Größten und β den kleineren Eigenwert repräsentieren. Die Summe der Eigenwerte lässt sich aus der Spur der Matrix H berechnen. Die Spur einer Matrix, im Englischen Trace (Tr) genannt, ist die Summe der Diagonalelemente einer quadrati-

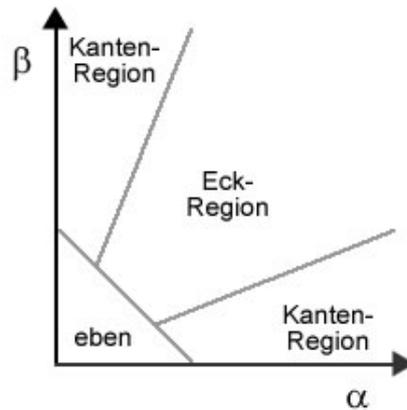


Abbildung 3.6.: Klassifikation der Ecken- und Kantenpunkte über das Verhältnis der Eigenwerte.

schen Matrix. Das Produkt der Eigenwerte erhält man durch Berechnung der Determinante der Matrix H :

$$\begin{aligned} \text{Tr}(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ \text{Det}(H) &= D_{xx} D_{yy} - (D_{xy})^2 = \alpha \beta. \end{aligned} \quad (3.8)$$

Für den unwahrscheinlichen Fall, dass die Determinante negativ sein sollte, was bedeuten würde, dass die Krümmungen verschiedene Vorzeichen hätten, würde der Schlüsselpunkt-Kandidat als kein Extremum verworfen. Dieser Fall ist im Anhang A, unter dem Punkt Eigenwerte auch noch einmal grafisch dargestellt. Abbildung 3.6 zeigt wie das Verhältnis der Eigenwerte zueinander genutzt wird um die Region um den Merkmalspunkt herum zu klassifizieren. Befindet sich der Schlüsselpunkt-Kandidat in einer eher homogenen Bildregion, so werden die beiden Eigenwerte relativ klein sein. Bei einer Kante hingegen werden sich die Eigenwerte deutlich voneinander unterscheiden, einer der beiden Eigenwerte wird in diesem Fall relativ groß sein und der andere klein.

Sei γ die Verhältniszahl zwischen dem Größten und einem kleineren Eigenwert, so dass: $\alpha = \gamma \beta$. Dann erhält man folgende Gleichung:

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha \beta} = \frac{(\gamma \beta + \beta)^2}{\gamma \beta^2} = \frac{(\gamma + 1)^2}{\gamma}, \quad (3.9)$$

welche nur noch von dem Verhältnis der Eigenwerte abhängig ist, anstatt von den einzelnen Werten. Die Quantität für $\frac{(\gamma+1)^2}{\gamma}$ ist am kleinsten, wenn die beiden Eigenwerte gleich groß sind und sie steigen mit zunehmenden γ an. Um zu überprüfen, ob sich das Verhältnis zwischen den Hauptkrümmungen unter einem bestimmten Grenzwert, γ , befindet, muss

folgende Bedingung geprüft werden:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(\gamma + 1)^2}{\beta}. \quad (3.10)$$

Lowe verwendet in seinen Experimenten einen Grenzwert von $\gamma = 10$, somit werden alle Schlüsselpunkt-Kandidaten zurückgewiesen, welche eine größere Verhältniszahl aufweisen (vgl. Lowe, 2004, S. 12). Die konkrete Implementierung, der in diesem Abschnitt beschriebenen Algorithmen zur genaueren Untersuchung der einzelnen Schlüsselpunkt-Kandidaten sowie der Suche nach dem im vorigem Kapitel beschriebenen Extrema, ist im Anhang B zu finden.

3.3. Bestimmung der Orientierung

Durch das konsequente Zuweisen von Orientierungen für jeden markanten Raumpunkt kann der Merkmalspunkt-Deskriptor in einem späteren Schritt relativ zur Orientierung des Schlüsselpunktes dargestellt werden, wodurch die Invarianz gegen die Bildrotation erreicht wird.

Ausgehend von der Skalierung des Schlüsselpunktes wird das Gauß geglättete Bild ausgewählt, L , welches der Skalierung, σ , des Schlüsselpunktes am Nächsten liegt, sodass alle folgenden Berechnungen in einem skalierungsinvarianten Umfeld durchgeführt werden können. Für das ausgewählte Bild, $L(x, y)$, mit der Skalierung σ werden die Gradienten-Längen, $m(x, y)$ und die Richtungen (Orientierung) der Gradienten, $\theta(x, y)$, durch Pixeldifferenzen berechnet:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.11)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{(L(x, y + 1) - L(x, y - 1))}{(L(x + 1, y) - L(x - 1, y))} \right) \quad (3.12)$$

Ein Gradient ist ein Vektor, der in die Richtung des größten Funktionswertes zeigt und dessen Länge ein Maß für die Steigung (Anstieg des Funktionswertes) ist. Es wird ein Orientierung-Histogramm aus den Orientierungs-Gradienten der Punkte, die den Schlüsselpunkt umgeben erstellt. Das Histogramm enthält 36 Einträge, die den 360° Bereich der Orientierungen abdecken. Jeder Eintrag, der dem Histogramm hinzugefügt wird, wird durch die Länge des Gradienten und mit Hilfe eines Gauß gewichteten kreisförmigen Fenster, mit einem σ welches 1,5 mal dem der Skalierung des Schlüsselpunktes entspricht, gewichtet. Höchstwerte im Orientierung-Histogramm entsprechen den dominierenden Richtungen der lokalen Gradienten. Im Folgenden wird der Höchstwert des Histogramms festgestellt und ein Schlüsselpunkt mit dessen Orientierung erstellt. Wenn weitere Höchstwerte im Histogramm, die sich innerhalb einer Spanne von bis zu minimal 80% des Höchstwertes befinden, enthalten

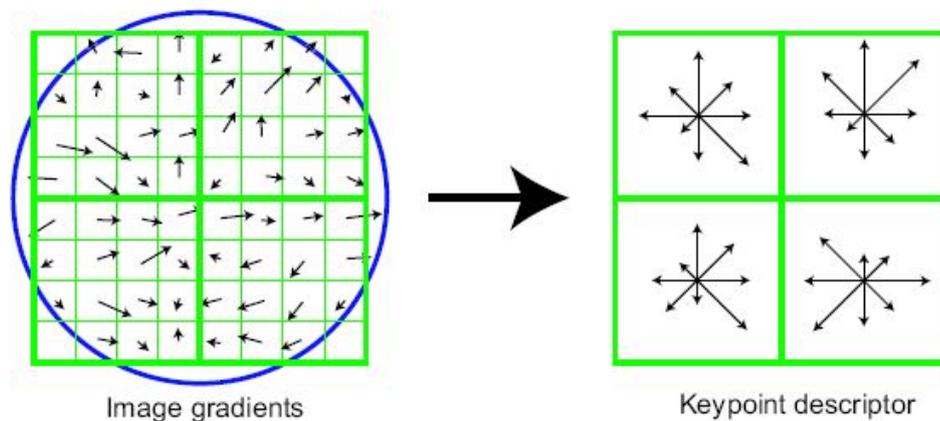


Abbildung 3.7.: Auf der linken Seite sind die Gradienten in der lokalen Umgebung eines Schlüsselpunktes dargestellt. Die rechte Abbildung zeigt beispielhaft den auf des Deskriptors, welcher in jedem Feld ein Orientierungs-Histogramm mit acht Einträgen enthält. [Lowe 2004]

sind, werden für diese weitere Schlüsselpunkte mit derselben Position und Skalierung, aber mit anderer Orientierung erstellt. Die Erstellung von mehreren Punkten, die sich nur in ihrer Orientierung unterscheiden, kommt aber nur in circa 15% aller Fälle vor. Diese erhöhen aber signifikant die Stabilität bei der Wiedererkennung der markanten Merkmalspunkte (vgl. Lowe, 2004, S. 13).

3.4. Erzeugen des Merkmalspunkt-Deskriptors

In den vorhergegangenen Schritten wurde jedem Schlüsselpunkt eine Position im Bild, Skalierung und Orientierung zugewiesen. Zum Abschluss wird nun für diese lokalen Bildregionen ein Deskriptor erzeugt. Der Deskriptor ist ein Vektor dessen Elemente den Schlüsselpunkt beschreiben. Durch die vorher angewandten Verfahren sind die Merkmalspunkte invariant gegenüber Änderung bezüglich der Position, Skalierung und Orientierung. Der Deskriptor soll die Schlüsselpunkte nicht nur in einer möglichst markanten Art und Weise beschreiben, sondern darüber hinaus dazu beitragen, dass verbliebene Einflüsse auf das Bild, wie eine veränderte Beleuchtung oder auch ein geänderter Aufnahmezustandpunkt, möglichst wenig Einfluss auf die Beschreibung der Schlüsselpunkte haben.

Aus Effizienzgründen werden die Gradienten aller Bilder der Gaußpyramide, wie im vorigem Abschnitt beschrieben, vorberechnet. Anhand der Skalierung des Schlüsselpunktes

wird dann wieder das zu der Skalierung korrespondierende Bild aus der Gaußpyramide ausgewählt. Der Deskriptor wird aus mehreren einzelnen Orientierungs-Histogrammen berechnet. Dabei geben die einzelnen Histogramme jeweils die Orientierungen in einer 4x4 großen Pixel-Nachbarschaft um den Schlüsselpunkt herum an. Der Deskriptor besteht aus einem 4x4 großem Array, wobei jeder Eintrag in diesem jeweils ein Histogramm mit 8 Orientierungen enthält. Somit ergibt sich ein Deskriptor mit 4x4x8 Elementen, der Deskriptor stellt also einen 128-dimensionalen Vektor dar.

Abbildung 3.7 zeigt auf der linken Seite die Gradienten, die in der Region rund um den Schlüsselpunkt berechnet werden. In diesem Beispiel wird zur besseren Übersicht nur ein 8x8 großes Fenster mit den jeweiligen Gradienten dargestellt. Wo hingegen die einzelnen Histogramme des Deskriptors aus einem 16x16 großem Fenster um den Schlüsselpunkt herum berechnet werden. Das betrachtete Fenster um den jeweiligen Schlüsselpunkt herum, wird jeweils in einzelne Histogramme unterteilt, die die Orientierung der Punkte in einer 4x4 großen Nachbarschaft enthalten. Die Stärke (Länge) jedes Gradienten wird mittels einer Gauß gewichteten Funktion gewichtet, bevor der Gradient dem Histogramm hinzugefügt wird. Dieses „Gaußsche-Fenster“ ist im linken Bild als Kreis angedeutet. Der Sinn der Gewichtung besteht darin, dass plötzliche Änderungen im Deskriptor durch kleine Veränderung der Position des betrachteten Ausschnittes vermieden werden sollen. Außerdem sollen so Gradienten, die sich weiter weg vom Zentrum befinden, einen kleineren Einfluss auf den Deskriptor haben. Auf der rechten Seite des Bildes sind jeweils die einzelnen Orientierungs-Histogramme zu erkennen, die jeweils acht Orientierungen enthalten. In diesem Beispiel ist auch bei dieser Abbildung zum Zweck der besseren Übersichtlichkeit eine kleinere 2x2 Variante gewählt worden. Der Deskriptor selbst ergibt sich aus einem 4x4 großes Array mit den jeweiligen Histogrammeinträgen in jedem Feld. Aus diesen Einträgen ergibt sich dann der 128-elementige Deskriptor. Dieser Vektor wird abschließend noch normalisiert um die Invarianz gegenüber Änderungen in der Beleuchtung zu verstärken (vgl. Lowe, 2004, S. 16).

4. Analyse

Das erfolgreiche Wiedererkennen von bekannten markanten Raumpunkten setzt voraus, dass die gefundenen Punkte möglichst unabhängig von Skalierung, Rotation, Lichtverhältnissen und dem Blickwinkel sind. Der SIFT-Deskriptor ist so konzipiert, dass er diese veränderlichen Eigenschaften normalisiert. Die gefundenen Merkmalspunkte eines Objekts können somit in anderen Szenarien, die dieses Objekt enthalten, wiedergefunden werden, wenn sie nicht über ein gewisses Maß hinaus transformiert worden sind. Dieses Kapitel soll aufzeigen, inwieweit die Bildinhalte und deren Transformation Auswirkungen auf die Wiedererkennung der markanten Raumpunkte des SIFT-Verfahrens haben. Hierbei werden die zu untersuchenden Bilder in zwei verschiedene Szenariotypen eingeteilt:

Zum Einen wird anhand von Bildsequenzen untersucht, inwiefern Merkmalspunkte in zwei aufeinander folgenden Bildern einer Sequenz, zuverlässig wiedererkannt werden können. Der zweite Szenariotyp widmet sich dem Auffinden von gleichen Merkmalspunkten bei Bildpaaren. Dies beinhaltet unter anderem das Wiederfinden von Objekten, die transformiert und in einer anderen Umgebung dargestellt werden.

Die in diesem Kapitel durchgeführten Analysen basieren auf Tests und Vergleichen, die mit Hilfe des für Impresario entwickeltem SIFT++ Makro durchgeführt wurden. Für die grundlegenden SIFT-Algorithmen wurde hierbei auf die SIFT++ Implementierung von Andrea Vedaldi vom UCLA VisionLab zurück gegriffen. Alle weiteren Bildverarbeitungsschritte so wie Vergleiche wurden in C++ auf Basis der LTI-Lib implementiert. Diese Bildverarbeitungsbibliothek bietet viele Algorithmen und Datenstrukturen auf die man bei der Programmierung im Bereich der Bildverarbeitung zurückgreifen kann. So wird zum Beispiel der Vergleich und die Speicherung der SIFT-Deskriptoren anhand eines in der LTI-Lib vorimplementierten Kd-Trees realisiert. In diesem werden die Deskriptoren der Größe nach sortiert abgespeichert. Jeder Knoten im Baum enthält dabei als Schlüssel den 128-elementigen Deskriptor und als Daten die Koordinaten des Bildpunktes im untersuchten Bild, damit später Vergleiche vorgenommen werden können, die sich auf die Position der Punkte im „Original-Bild“ beziehen. Verglichen werden die Deskriptoren hierbei mit Hilfe des Nearest-Neighbour-Verfahren (Arya und Mount, 1993, vgl.).

4.1. Bildsequenzen

In diesem Abschnitt soll untersucht werden, wie zuverlässig unter verschiedenen Bedingungen gefundene Merkmalspunkte in aufeinander folgenden Bildern wiedergefunden werden können. Anhand beispielhaft ausgewählten Szenarien wird veranschaulicht, welchen Einfluss die Bildinhalte auf die Ergebnisse des SIFT-Algorithmus haben. Die einzelnen Szenariotypen unterscheiden sich deutlich in ihren Bildinhalten, um konkrete Aussagen über die Eignung des Algorithmuses für bestimmte Einsatzgebiete treffen zu können.

Ein spezielles Anwendungsgebiet des SIFT-Algorithmus auf Bildsequenzen könnte zum Beispiel die Navigation autonomer Fahrzeuge sein. Es würde sich die Möglichkeit bieten, alle bei einer Kamerafahrt gefundenen Schlüsselpunkte abzuspeichern. Diese so gewonnenen Umgebungsinformationen könnten für eine spätere autonome Streckenfindung eines Fahrzeuges genutzt werden.

4.1.1. Kamerafahrt im Innenraum

Kamerafahrten in Innenräumen haben besondere Bildeigenschaften. In Räumen herrschen oft Lichtverhältnisse, die entweder komplett von künstlichen Lichtquellen geschaffen werden oder eine Mischung aus Kunst- und Tageslicht darstellen. Die Kontrastverhältnisse in geschlossenen Räumen können einen großen Einfluss auf die Suche nach Schlüsselpunkten in Bildern haben. Ein zu schwacher Kontrast kann beim SIFT-Verfahren zur Zurückweisung eines möglichen Schlüsselpunkt-Kandidatens führen. Somit hat der Kontrastumfang eines Bildes eine direkte Auswirkung auf das Erkennen von markanten Raumpunkten in Aufnahmen.

Die Aufnahmen von Abbildung 4.1 sind in einem geschlossenen Flur entstanden, der ausschließlich durch Kunstlicht beleuchtet wird. In diesem Bild ist auf der rechten Seite das aktuelle Bild einer Videosequenz zu sehen und auf der linken Seite das vorherige Bild dieser Sequenz. In der linken Aufnahme sind 152 markante Raumpunkte gefunden worden und in der Rechten sind es 132 erkannte Punkte. Die markanten Raumpunkte der beiden aufeinander folgenden Bilder wurden verglichen. Punkte, die als übereinstimmend in beiden Ausschnitten erkannt wurden, sind jeweils mit einer Linie verbunden worden (Abbildung 4.2). Insgesamt konnten in den beiden Bildern 119 Punkte als übereinstimmend wiedergefunden werden. In Abbildung 4.3 ist die Verschiebung der gefundenen Punkte eingezeichnet. In dieser Videosequenz wird die Kamera stetig nach vorne verschoben, somit bewegen sich die wiedererkannten Punkte auf die Kamera zu.



Abbildung 4.1.: Diese Bilder sind in einem nur von Kunstlicht beleuchteten Flur aufgenommen worden. (Links das n-1 Bild und rechts das aktuelle Bild mit den jeweils markanten Raumpunkten)



Abbildung 4.2.: Übereinstimmend gefundene markante Raumpunkte in zwei aufeinander folgenden Bildern.



Abbildung 4.3.: Bewegungspfeile die die Verschiebungsrichtung der einzelnen Punkte angeben.

4.1.2. Kamerafahrt durch einen Wald

Aufnahmen, die in Wäldern entstehen, können häufig recht gleichmäßige Bildinhalte enthalten. In einem typischen Waldszenario werden die einzelnen Bilder oftmals mehrfach dieselben Pflanzen enthalten. Der Boden eines Waldes kann zum Beispiel sehr gleichmäßig mit Pflanzen derselben Art bewachsen sein. Es gilt zu untersuchen, wie sich diese Gleichmäßigkeit auf die Suche und das Wiedererkennen von Schlüsselpunkten auswirkt.

Ein dicht bewachsener Grund eines Waldes, wie er in Abbildung 4.4 auf der linken Seite zu sehen ist, stellt einen eher homogenen Bildinhalt dar. Es gibt keine Objekte, die sich deutlich von anderen absetzen und dennoch findet der SIFT-Algorithmus beachtlich viele markante Punkte im Bild, wie in Abbildung 4.4 auf der rechten Seite zu erkennen ist. Darüber hinaus sind diese in diesem Szenario gefundenen Punkte auch sehr stabile Merkmale. In Bild 4.5 wird die Verschiebung wieder erkannter Punkte gezeigt. Das Bild zeigt die Folgeaufnahme bei einem Kameraschwenk nach oben, des zuvor gezeigten Bildes des Waldbodens mit eingezeichneten Bewegungspfeilen. Eine Vielzahl der zuvor erkannten markanten Raumpunkte



Abbildung 4.4.: Auf der linken Seite ist die Aufnahme eines dicht bewachsenen Waldbodens zu sehen. Rechts werden die in dem Bild gefundenen markanten Raumpunkte dargestellt.

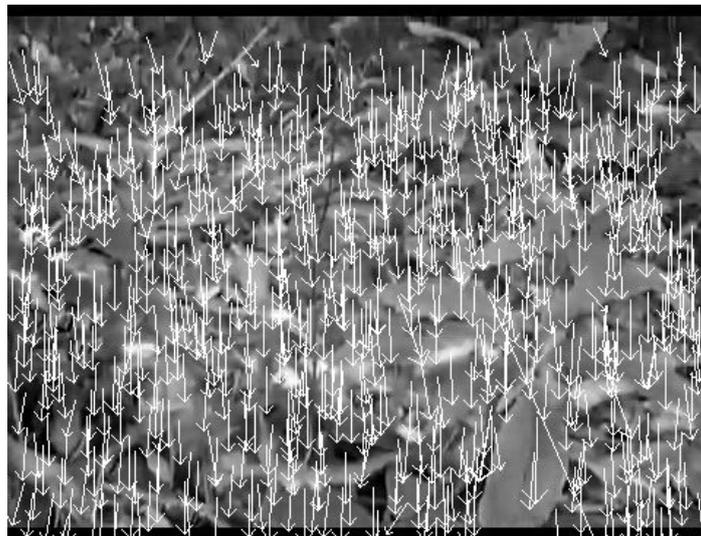


Abbildung 4.5.: Kameraschwenk nach oben, die Bewegungspfeile zeigen die Verschiebung der erkannten markanten Raumpunkte an.

ist wiedererkannt worden und mit den entsprechenden Verschiebungsrichtungen in dieser Abbildung dargestellt.



Abbildung 4.6.: Vergleich zweier aufeinander folgender Bilder einer Sequenz die mittels einer ruhenden Kamera entstanden sind.

4.1.3. Verkehrsszene mit ruhender Kamera

In diesem Szenario werden Aufnahmen mit einer ruhenden Kamera gemacht. Dies bedeutet, dass die Kamera fest an einem Punkt installiert ist und während der Aufnahmezeit nicht bewegt wird. Daraus ergibt sich eine Szene, in der sowohl sich bewegende Objekte, wie auch sich nicht bewegende Objekte vorkommen.

In dieser sehr komplexen Szene sind viele einzelne Objekte zu erkennen, die sich teilweise auch deutlich voneinander abheben. Wie in Abbildung 4.6 zu erkennen ist, werden in einer solchen Umgebung sehr viele markante Raumpunkte gefunden und auch im Folgebild der Sequenz sehr stabil wiedererkannt. Wie in diesem Szenario zu erkennen ist, gibt es Bildinhalte, die immer an derselben Position im Bild bleiben, wie zum Beispiel die Fassaden der Gebäude oder die Verkehrsschilder. Andere Inhalte hingegen bewegen sich, wie zum Beispiel die Personenkraftwagen oder Motorroller. In Abbildung 4.7 wird durch die Bewegungspfeile angezeigt, in welche Richtung sich wiedererkannte Punkte bewegt haben. Wie deutlich zu erkennen ist, fahren die Fahrzeuge auf der unteren Spur nach rechts und diejenigen, die sich auf der oberen Spur bewegen nach links. Markante Raumpunkte, die sich nicht bewegt haben, werden als Punkt dargestellt.



Abbildung 4.7.: Bewegung wieder erkannter markanter Raumpunkte.

4.2. Bildpaare

Dieser Abschnitt befasst sich mit dem Vergleich von gefundenen Merkmalspunkten zweier Bilder. Die zu untersuchenden Bildpaare lassen sich grob in zwei Kategorien unterteilen: Zum Einen in das Wiedererkennen von Objekten in Bildern, zum Anderen in das Vergleichen von Stereobildpaaren. Die Suche nach einzelnen Objekten in Szenarien kann für viele Anwendungsgebiete genutzt werden, wie zum Beispiel für „portable Guides“. Es würde sich zum Beispiel anbieten eine Datenbank aufzubauen, die sowohl die Schlüsselpunkte sowie auch allgemeine Informationen zu Kunstwerken enthält. Mittels eines portablen Endgerätes, welches eine Kamera enthält, könnten so zum Beispiel Besucher eines Museums näher Informationen zu einem gewünschten Kunstobjekt erhalten.

Um feststellen zu können wie gut sich der SIFT-Algorithmus zur Wiedererkennung von Objekten eignet und ob es Objekte gibt, die sich besser wiedererkennen lassen als andere, wird das Verfahren an verschiedenen Szenarien/Objekten getestet.

4.2.1. Kunstwerk im Foto wiedererkennen

Ein interessantes Einsatzgebiet des SIFT-Algorithmuses ist das Wiedererkennen von Objekten. Hierbei wird in einer Szene, die mehrere Objekte enthält, nach einem bestimmten Objekt gesucht. Dabei gilt es besonders zu beobachten, in wieweit das gesuchte Objekt nach Abweichungen in der Skalierung und der Lage im Bild wiederzuerkennen ist.

Die Abbildung 4.8 zeigt auf der linken Seite das Gemälde „Les Usines“ von Fernand Léger. Auf dem rechten Bild ist das Gemälde in der Mitte als Teil einer Ausstellung wiederzufinden. In beiden Bildern sind die gefundenen markanten Raumpunkte eingezeichnet. Es gilt zu beachten, dass sich das Gemälde in beiden Bildern sowohl in der Skalierung und Lage als auch im Kontrast voneinander unterscheidet. Der Vergleich aller in beiden Bildern gefundenen markanten Raumpunkte ist in Abbildung 4.9 zu sehen. Wie deutlich zu erkennen ist, sind mehrere Punkte im Gemälde als übereinstimmend wiedererkannt worden. Abbildung 4.10 zeigt als weiteres Beispiel das Gemälde „Junge mit Pfeife“ von Pablo Picasso bei einer Sotheby's Auktion. Die Anzahl der übereinstimmend gefundenen Merkmale ist groß genug um davon ausgehen zu können, dass eine automatisierte Suche nach Objekten in komplexeren Szenen mit Hilfe des SIFT-Verfahrens eine gute Aussicht auf Erfolg hat.



Abbildung 4.8.: Das Gemälde „Les Usines“ von Fernand Léger und eine Szene die dieses Bild enthält, jeweils mit eingezeichneten markanten Raumpunkten. [Pitzke 2007]



Abbildung 4.9.: Vergleich der wiedererkannten markanten Raumpunkte. [Pitzke 2007]

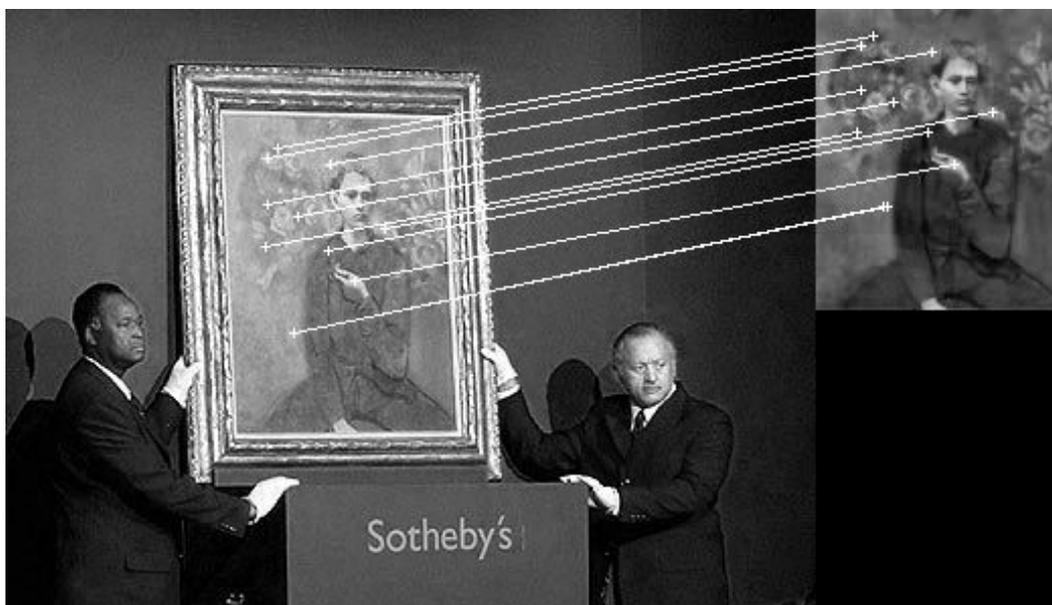


Abbildung 4.10.: Vergleich wiedererkannte markante Raumpunkte im Gemälde „Junge mit Pfeife“ von Pablo Picasso bei einer Sotheby's Auktion. [Pitzke 2007]

4.2.2. Stereobildpaare

Erläuterung des Begriffs „Stereobildpaar“: „Für ein 3D-Foto benötigt man zwei Teilbilder, für jedes Auge eines. Die Teilbilder müssen seitlich versetzt sein, dürfen aber weder gegeneinander verdreht noch verkantet sein. Auch ist der Abstand der beiden Teilbilder voneinander wichtig für den Eindruck von Raumtiefe“ (Schnehagen, 2004). Diese beiden Teilbilder zusammen werden als Stereobildpaar bezeichnet. Mittels des SIFT-Verfahrens können die gefundenen Merkmalspunkte dieser beiden Teilbilder verglichen werden.

Abbildung 4.11 zeigt ein solches Stereobildpaar, in das die jeweiligen markanten Raumpunkte eingezeichnet wurden. Auf der linken Seite ist das linke Teilbild und auf der rechten Seite das rechte Teilbild abgebildet. Bei genauerer Betrachtung der Bilder fällt sofort das unterschiedliche Blickfeld auf. Bei der linken Aufnahme ist zum Beispiel rechts von der Tür, die Fensterfront fast komplett zu sehen, während diese im rechten Bild nur im Ansatz zu erkennen ist. Abbildung 4.12 zeigt den Vergleich beider Bilder mit den als übereinstimmend wiedererkannten markanten Merkmalspunkten.



Abbildung 4.11.: Auf der linken Seite ist das linke Teilbild und auf der rechten Seite das rechte Teilbild mit markanten Raumpunkten dargestellt. [Klein 2007]



Abbildung 4.12.: Vergleich der übereinstimmend gefundenen markanten Raumpunkte in beiden Teilbildern. [Klein 2007]

4.2.3. Gebäude aus verschiedenen Aufnahmestandpunkten

In diesem Abschnitt soll untersucht werden, wie sich verschiedene Aufnahmestandpunkte auf das Wiedererkennen von Gebäuden auswirken. Dabei werden Aufnahmen vom Gebäude aus unterschiedlichen Perspektiven verglichen.

In Abbildung 4.13 ist deutlich zu erkennen, dass am Gebäude eine Vielzahl von markanten Raumpunkten gefunden wurden. Die beiden Aufnahmen sind von einem leicht unterschiedlichen Aufnahmestandpunkt ausgemacht worden. In Abbildung 4.14 sind die als übereinstimmend wiedererkannten Punkte eingezeichnet. Bei dieser eher geringen Änderung des Aufnahmestandpunktes wurden, wie zu erwarten war, viele der Punkte wiedergefunden. Beim Vergleich zweier Aufnahmen, die aus einem deutlich voneinander abweichenden Standpunkt gemacht wurden, zeigt sich jedoch, dass es annähernd keine korrekten Übereinstimmungen mehr gibt. Abbildung 4.15 zeigt das Ergebnis eines direkten Vergleichs der markanten Raumpunkte zweier Bilder, die jeweils einzeln mehr als 600 markante Punkte aufweisen. Wie im Bild gut zu erkennen ist, ist die Übereinstimmung des Punktes in der Flagge und im Turm ein Fehler, der bei dem Vergleich der beiden Deskriptoren auftrat. Die zweite Übereinstimmung zeigt einen markanten Punkt, der an einer „Skulptur“ gefunden wurde, die mehrfach an der Fassade des Gebäudes zu finden ist.



Abbildung 4.13.: Das Hamburger Rathaus aus zwei verschiedenen Aufnahmestandpunkten.
[Landvogt 2007]



Abbildung 4.14.: Vergleich der als übereinstimmend gefundenen markanten Raumpunkte.
[Landvogt 2007]



Abbildung 4.15.: Vergleich zweier Bilder die mit großem Unterschied des Aufnahmezustandes entstanden sind.

4.3. Einfluss durch geänderte Aufnahmeeigenschaften

Der folgende Abschnitt beschäftigt sich mit den Auswirkungen, die unterschiedlich Objektkalibrierungen und abweichende Aufnahmewinkel auf die Wiedererkennung markanter Raumpunkte haben können. Des Weiteren wird auch der Einfluss von verschiedenen Bildhelligkeiten auf die Erkennung und Wiedererkennung der markanten Punkte untersucht. Diese Untersuchungen sind notwendig, weil auf diese Weise ermittelt werden kann unter welchen Bedingungen das SIFT-Verfahren „optimale“ Ergebnisse liefert. Darüber hinaus wird so auch in Erfahrung gebracht, wo die Grenzen des Verfahrens liegen. Um die Auswirkungen der zuvor genannten Parameter zu untersuchen, wurde ein übersichtliches Szenario gewählt, so dass relativ schnell erkannt werden kann, welche Auswirkung die Änderung eines Parameters auf die Wiedererkennung markanter Raumpunkte hat. In den folgenden Szenarien werden immer die markanten Punkte eines „ursprünglichen“ Bildes im direkten Vergleich mit denen einer im Aufnahmezustandpunkt oder der Bildhelligkeit abweichendem Aufnahme gezeigt.

4.3.1. Abweichende Skalierungen

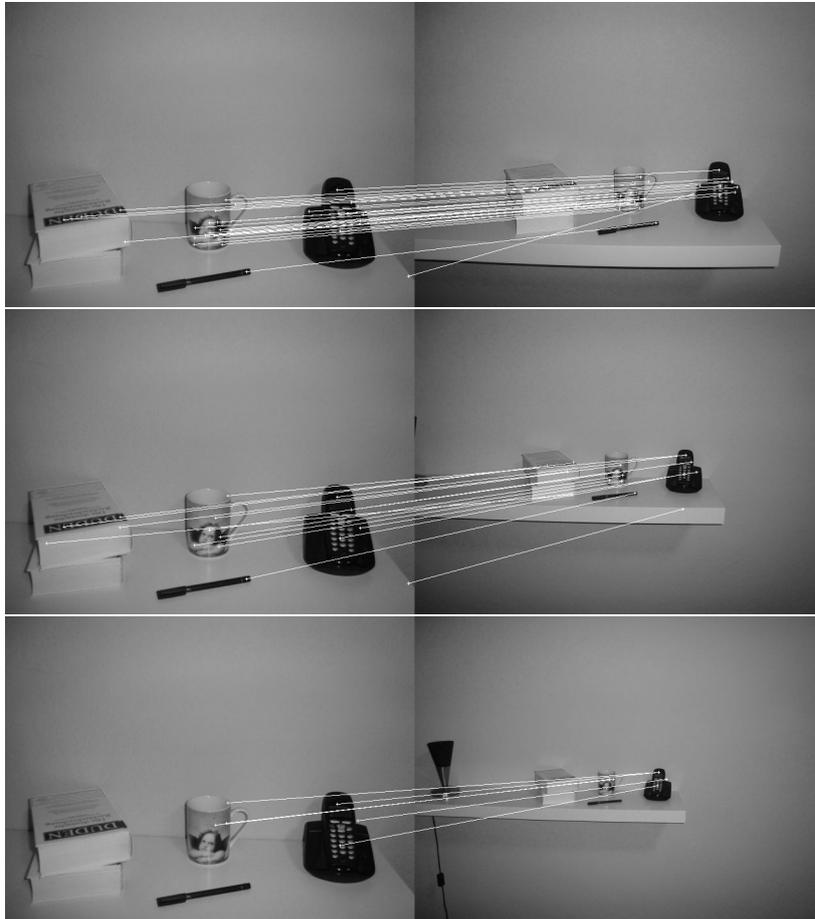


Abbildung 4.16.: Vergleich von Objekten bei unterschiedlichen Skalierungen. Auf der linken Seite ist jeweils das ursprüngliche Bild zu erkennen und rechts die Szene mit zunehmenden Abstand aufgenommen.

Um zu untersuchen wie sich abweichende Skalierungen von Objekten auf die Wiedererkennbarkeit dieser auswirkt, wurden Aufnahmen angefertigt, welche Objekte mit zunehmenden Abstand darstellen. Bei der Auswertung dieser Aufnahmen, welche in Abbildung 4.16 zu sehen sind, wurden die markanten Raumpunkte des „ursprünglichen“ Bildes, die immer auf der linken Seite zu sehen sind, mit denen einer Aufnahme aus weiterer Entfernung verglichen. Punkte, die in beiden Aufnahmen als übereinstimmend erkannt wurden, sind mit einer Linie verbunden eingezeichnet worden. Die Auswertung hat gezeigt, dass die mit dem SIFT-Verfahren gefundenen markanten Raumpunkte bis ungefähr zur halben „Ursprungsgröße“ vielversprechend wiedererkannt werden können. Nach dieser „Marke“ nimmt die Rate der Wiedererkennbarkeit dann deutlich und stetig ab.

4.3.2. Unterschiedliche Betrachtungswinkel

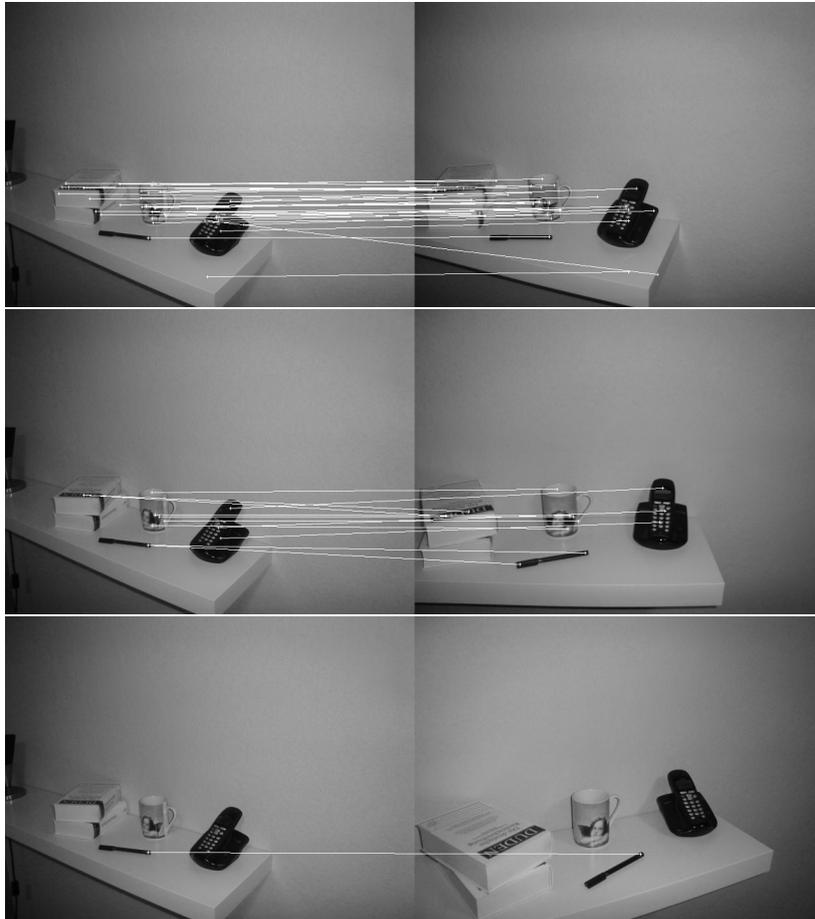


Abbildung 4.17.: Vergleich von Objekten bei unterschiedlichen Betrachtungswinkeln. Links ist immer die ursprüngliche Aufnahme gezeigt und rechts sind die Aufnahmen mit zunehmender Änderung des Betrachtungswinkel zu sehen.

Um zu zeigen, welche Ergebnisse die Wiedererkennung bei abweichenden Aufnahmestandpunkten liefert wurden die Objekte aus unterschiedlichen Winkeln aufgenommen. Die markanten Raumpunkte der „ursprünglichen“ Aufnahme, in Abbildung 4.17 auf der linken Seite zu sehen, wurden mit denen verglichen, welche aus den Aufnahmen mit abweichenden Winkeln extrahiert wurden. Die Folgeaufnahmen mit abweichenden Aufnahmestandpunkten weichen dabei im zunehmenden Maße vom ursprünglichen Standpunkt ab. Wie auch schon im Falle der abweichenden Skalierungen zeigt sich auch hier eine stetig sinkende Rate von erfolgreich wiedererkannten Punkten. Bei Winkeländerungen im Bereich von ca. 20° pro Aufnahme hat sich hierbei die Anzahl der wiedererkannten markanten Raumpunkte ungefähr halbiert.

4.3.3. Verschiedene Bildhelligkeiten

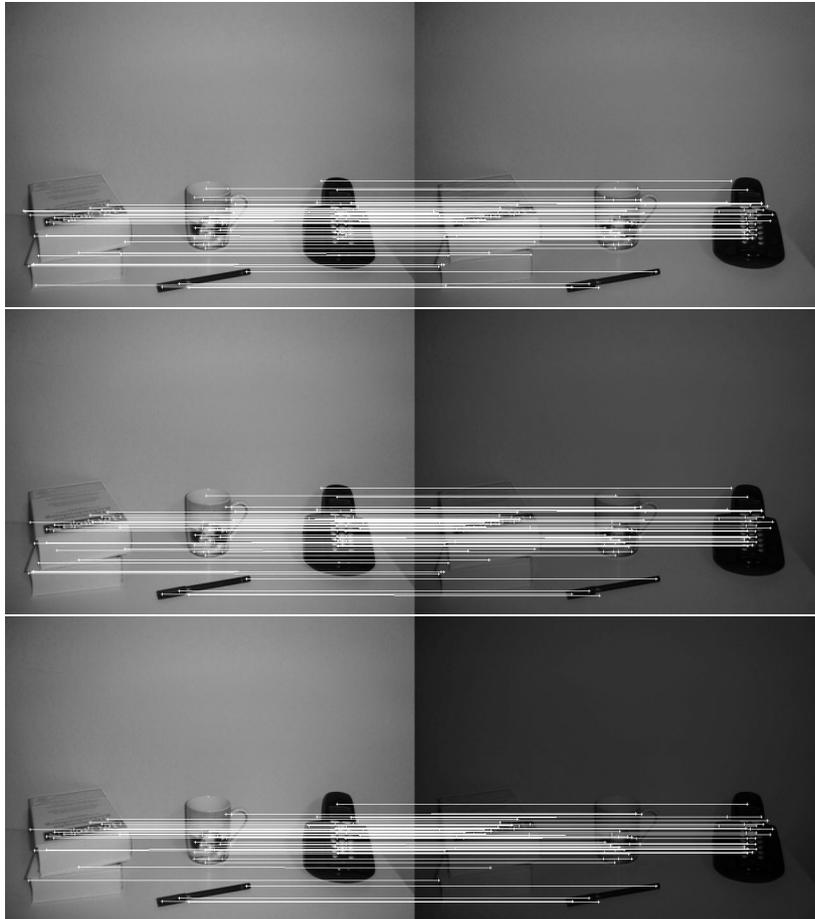


Abbildung 4.18.: Vergleich von Objekten bei verschiedenen Bildhelligkeiten. Auf der linken Seite ist jeweils das ursprüngliche Bild zu sehen und auf der rechten Seite die Bilder mit abnehmender Helligkeit.

Um zu untersuchen wie sich Helligkeitsänderungen auf die Wiedererkennung auswirken, wurden Bilder verglichen, welche in der Helligkeit jeweils deutlich abnehmen. Die Abbildung 4.18 zeigt die ursprüngliche Aufnahme jeweils auf der linken Seite des Bildes und rechts sind jeweils die Bilder mit geringer werdender Helligkeit zu sehen. Wie diese Vergleiche deutlich zeigen, hat die Veränderung der Bildhelligkeit nur geringe Auswirkungen. Eine Vielzahl der im ursprünglichen Bild gefundenen markanten Raumpunkte konnte auch in den dunkleren Bildern gefunden werden. Dementsprechend sind auch beim Vergleich der markanten Punkte, der jeweiligen Bilder, die meisten Punkte übereinstimmend wiedererkannt worden.

4.4. Fazit

Die in diesem Kapitel gezeigten Beispiele für die einzelnen Szenarien sind ein repräsentativ ausgewählter Querschnitt aus den untersuchten Bildsequenzen und Bildpaaren. Grundsätzlich lässt sich sagen, dass der SIFT-Algorithmus bei allen getesteten Szenarien sehr viel versprechende Ergebnisse geliefert hat. Der Bildinhalt hat hierbei keine große Rolle gespielt. Eine Vielzahl der gefundenen markanten Raumpunkte konnte sowohl bei homogenen, wie auch heterogenen Bildinhalten in Folgebildern einer Sequenz oder als gesuchtes Objekt in einer Szene wiedererkannt werden. Hierbei war die Fehlerquote von dem Vergleich der einzelnen Deskriptoren auf Übereinstimmung sehr gering, wie es auch deutlich die vorhergegangenen Abbildungen der Vergleiche gezeigt haben.

Das Verfahren hat sich als robust gegen Änderungen bei Skalierung und Rotation erwiesen. Dies wurde insbesondere bei der Suche nach Objekten sichtbar. Im Abschnitt „Kunstwerk im Foto wiedererkennen“ hat sich klar gezeigt, dass genug markante Raumpunkte wiedererkannt werden können um ein Objekt, wie in diesem Fall ein Gemälde, eindeutig identifizieren zu können. Des Weiteren haben die Untersuchungen mit variierenden Skalierungen und Aufnahmestandpunkten gezeigt, dass sich größere Änderungen deutlich in der Rate der wiedererkannten Punkte bemerkbar gemacht haben. Die Reduzierung der Bildhelligkeit hingegen hat nur eine geringe Auswirkung auf die Anzahl der erkannten und wiedererkannten markanten Raumpunkte gehabt.

In den untersuchten Bildern ist darüber hinaus auch gut zu erkennen, dass es hin und wieder „falsch“ als übereinstimmend zugeordnete markante Raumpunkte im Vergleich zweier Aufnahmen gibt. In diesen Fällen wurden beim Vergleich der Deskriptoren mittels des Nearest Neighbour - Verfahrens ein Abstand zwischen den verglichenen Deskriptoren festgestellt, welcher ausreichend gering war um als übereinstimmend gewertet zu werden. Bei diesen Fällen gibt es also markante Raumpunkte in den verglichenen Bildern, welche einen nahe zu identisches Deskriptor besitzen, aber nicht dem gesuchten Merkmalspunkt „entsprechen“. Um zum Beispiel bei der Identifizierung von Objekten sicherstellen zu können, dass es sich mit sehr hoher Wahrscheinlichkeit um ein bekanntes Objekt in einer Aufnahme handelt, wäre es sinnvoll ein Cluster von markanten Merkmalspunkten nachzuweisen. Ein Objekt würde also durch eine vorausgesetzte minimale Anzahl von übereinstimmend erkannten Merkmalspunkten als erkannt gewertet werden.

Des Weiteren haben Untersuchungen gezeigt, dass durch eine geeignet gewählte Vorverarbeitung der Eingangsbilder, Einfluss auf die später durch das SIFT-Verfahren gewonnenen markanten Raumpunkte genommen werden kann. Selbstverständlich lässt sich auch mittels Änderung ausgewählter Parameter des SIFT-Algorithmuses Einfluss auf die Gewinnung der markanten Raumpunkte nehmen. Wenn aber der Kontrast des Eingangsbildes stark verringert wird, werden so in eher homogenen Bereichen des Bildes so gut wie keine markanten



Abbildung 4.19.: Links das Rathaus ohne Nachbearbeitung und rechts mit geringem Kontrast. Es ist deutlich zu erkennen wie im Kontrast ärmeren Bild in relativ homogenen Bildbereichen weniger markante Raumpunkte gefunden werden.

Punkte mehr gefunden. Diesen Effekt kann man zum Beispiel bei Aufnahmen von Gebäuden nutzen, um nicht unnötig viele markante Raumpunkte in relativ homogenen Bildbereichen, wie zum Beispiel den Himmel, zu erhalten (s. Abbildung 4.19). Denn Punkte, die nicht von Interesse sind, sind im Zweifelsfall nur verschwendete Rechenzeit. Diese potenziell markanten Raumpunkte müssten dann weiteren genaueren Untersuchungen unterzogen werden. Darüber hinaus würden diese Punkte dann zum Beispiel auch unnötig gespeichert und bei der Suche nach Übereinstimmung mit anderen Deskriptoren überflüssigerweise verglichen werden.

5. Schluss

Die Analyse hat gezeigt, dass das SIFT-Verfahren sich auf einem breiten Spektrum von Bildinhalten anwenden lässt, ohne dabei markante Unterschiede in der Qualität der Ergebnisse aufzuweisen. Sowohl in Aufnahmen mit eher homogenen Bildinhalten als auch in Bildern mit sich deutlich voneinander absetzenden Inhalten wurden bei der Wiedererkennung von markanten Raumpunkten gute Ergebnisse erzielt. Auch wurden in homogenen Bildern eine Vielzahl von Schlüsselpunkten gefunden, die auch zum größten Teil erfolgreich wiedererkannt werden konnten. Des Weiteren hat auch der Ort der Aufnahme und die damit verbundenen unterschiedlichen Lichtverhältnisse keine große Auswirkung auf die Suche und Wiedererkennung markanter Raumpunkte gezeigt. Sowohl in Innenräumen, welche nur mit Kunstlicht beleuchtet waren, als auch bei Aufnahmen im Freien und bei Mischung der Lichtquellen, sind gleich gute Ergebnisse erzielt worden. Die gefundenen Merkmale haben sich darüber hinaus auch als sehr robust gegenüber Änderungen der Skalierung und Rotation gezeigt. Merkmalspunkte verschiedener Objekte konnten erfolgreich wiedererkannt werden bei einer abweichenden Größe und Lage der verglichenen Objekte. Dies haben zum Beispiel die Untersuchungen anhand der Kunstwerke deutlich gezeigt. Auch haben Veränderungen der Helligkeit keinen großen Einfluss auf das Wiedererkennen der markanten Raumpunkte gezeigt. Allerdings sollte bei Aufnahmen darauf geachtet werden, dass der Aufnahmestandpunkt nicht allzu viel von dem, bei der Speicherung der markanten Raumpunkte abweicht. Um sicherstellen, dass genügend markante Raumpunkte eines Objektes wiedererkannt werden können um Objekte eindeutig identifizieren zu können.

Es hat sich gezeigt, dass das SIFT-Verfahren eine solide Basis für Anwendungen bieten kann, welche sich mit Aufgaben der Erkennung und/oder Wiedererkennung von Objekten oder Raumpunkten in Grauwertbildern beschäftigen. Aufbauend auf diesem Verfahren könnte man sich eine Vielzahl von Anwendungen vorstellen, welche auf die Erkennung von markanten Raumpunkten angewiesen sind. Eine solche Anwendung, die auf das SIFT-Verfahren zurückgreift, könnte zum Beispiel ein „Portable Guide“ darstellen. Hierbei könnten zum Beispiel Handhelds oder leistungsstarke Mobiltelefone mit eingebauter Kamera als Plattform-Basis eingesetzt werden. Für diese Plattformen könnte dann eine Software entwickelt werden, welche mittels der integrierten Kamera Bilder auswertet. Je nach Einsatzgebiet könnte man Datenbanken aufbauen, welche zum Beispiel Daten für Kunstwerke zur Verfügung stellen. So wäre es denkbar, dass diese Datenbank die markanten Punkte von Gemälden enthält. Darüber hinaus könnten weiterführende Informationen zu jedem Kunstwerk gespeichert

werden, wie zum Beispiel Informationen zur Entstehung des Kunstwerkes. Sodass zum Beispiel der Besucher einer Kunsthalle eine Aufnahme von einem Gemälde macht, von welchem er gerne weiterführende Informationen hätte. Die Anwendung gleicht dann die markanten Punkte des Gemäldes mit ihrer Datenbank ab und stellt bei Übereinstimmung die dazugehörigen weiterführenden Informationen dar. Ein weiteres Gebiet auf dem man sich den Einsatz des SIFT-Verfahrens vorstellen kann, wäre bei der Routenfindung von autonomen Fahrzeugen. Dieses Fahrzeug könnte zum Beispiel vom Menschen navigiert eine bestimmte Strecke abfahren und sich markante Punkte entlang dieser Route merken. Bei einer späteren autonomen Fahrt könnte sich das Fahrzeug dann mit Hilfe dieser zuvor gefundenen markanten Raumpunkte orientieren um zum Beispiel eine festgelegte Route abzufahren.

Bevor allerdings daran gedacht werden sollte, eine größere Anwendung zu realisieren, welche auf das SIFT-Verfahren zurückgreift, sollten noch weitere Untersuchungen durchgeführt werden. Ein Punkt, den es noch besonders zu überprüfen gilt, ist der Aufbau und das Sammeln von Erfahrungen im Umgang mit einer großen Datenbank von SIFT-Deskriptoren. Es muss gezeigt werden, dass auch für eine größere Anzahl von Deskriptoren die Vergleiche sicher zu richtigen Ergebnissen führen und diese auch noch in einer angemessenen Rechenzeit erzielt werden können. Hierbei könnten dann unter anderem durch Versuche gezeigt werden, welches Verfahren sich am Besten für die Suche nach einem übereinstimmenden Deskriptor in einem Datenbestand eignet. Das Augenmerk könnte dabei erst einmal auf den von Lowe favorisierten Best-Bin-First-Algorithmus gelegt werden. Dessen Ergebnisse könnten aber mit den von anderen Suchalgorithmen, wie zum Beispiel dem Nearest-Neighbour, verglichen werden.

Darüber hinaus ist das SIFT-Verfahren nicht das einzige vielversprechende Verfahren im Bereich der Objekterkennung. Das SURF (Speeded Up Robust Features) -Verfahren (Bay u. a., 2006) ist zum Beispiel auch ein sehr viel versprechendes Verfahren auf diesem Gebiet. Welches verspricht besonders performant markante Raumpunkte, welche invariant gegenüber Skalierungs- und Rotationsänderungen sein sollen, aus Aufnahmen zu extrahieren und in einem Deskriptor zu beschreiben. Es wäre wohl ratsam erst einmal mehrere Verfahren miteinander zu vergleichen um herauszufinden, in welchen Bereichen die einzelnen Verfahren ihre Stärken und Schwächen aufweisen. Vielleicht bieten sich einige Verfahren mehr für bestimmte Einsatzgebiete an als Andere. Sodass das zu verwendende Verfahren eventuell je nach Zuverlässigkeit und Performance anhand bestimmten einzuhaltenden Kriterien ausgewählt werden kann.

Abschließend lässt sich sagen, dass es im Bereich der Objekterkennung vielversprechende Verfahren gibt. Es gibt eine Fülle von möglichen Anwendungen, bei welchen diese Verfahren zum Einsatz kommen könnten. Und mit der stetig steigenden Rechenleistung werden solche Verfahren mit ziemlicher Sicherheit in Zukunft, in nahezu Echtzeit operierend, in den verschiedensten Anwendungen zum Einsatz kommen.

Quellenverzeichnis

- [Analysis 2005] Analysis für Informatiker, Johann Wolfgang Goethe-Universität, Fachbereich Informatik und Mathematik, 2005. – URL <http://ismi.math.uni-frankfurt.de/analysisfuerinformatiker/Merkblatt6a.pdf>. – Zugriffsdatum: 08.07.2008
- [Arya und Mount 1993] ARYA, Sunil ; MOUNT, David: Approximate Nearest Neighbor Queries in Fixed Dimensions. In: *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*. Philadelphia : Society for Industrial and Applied Mathematics, 1993, S. 271–280. – ISBN 0-89871-313-7
- [Bay u. a. 2006] BAY, Herbert ; TUYTELAARS, Tinne ; VAN GOOL, Luc: SURF: Speeded Up Robust Features. In: *9th European Conference on Computer Vision*. Heidelberg : Springer-Verlag GmbH, 2006, S. 404–417. – URL <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>. – Zugriffsdatum: 08.07.2008. – ISBN 3-540-33832-2
- [Brown und Lowe 2002] BROWN, Matthew ; LOWE, David: *Invariant features from interest point groups*. 2002. – URL citeseer.ist.psu.edu/brown02invariant.html. – Zugriffsdatum: 08.07.2008
- [Davies 2005] DAVIES, Roy: *Machine Vision*. 3. Edition. Amsterdam : Elsevier, 2005. – ISBN 0-12-206093-8
- [Harris und Stephens 1988] HARRIS, Chris ; STEPHENS, mike: A Combined Corner and Edge Detection. In: *Proceedings of The Fourth Alvey Vision Conference*, URL http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A_Combined_Corner_and_Edge_Detector.pdf. – Zugriffsdatum: 08.07.2008, 1988, S. 147–151
- [Klein 2007] KLEIN, Alexander: Stereoscopy.com - The Gallery, URL <http://www.stereoscopy.com/gallery/index.html>. – Zugriffsdatum: 08.07.2008, 2007
- [Landvogt 2007] LANDVOGT, Axel: Hamburger-Ansichten: Rathaus, URL <http://www.hamburger-ansichten.de/rathaus/index.html>. – Zugriffsdatum: 08.07.2008, 2007

- [Lindeberg 1994] LINDBERG, Tony: Scale-space theory: A basic tool for analysing structures at different scales. In: *J. of Applied Statistics* 21(2) (1994), S. 224–270. – URL citeseer.ist.psu.edu/lindeberg94scalespace.html. – Zugriffsdatum: 08.07.2008
- [Lowe 2004] LOWE, David: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* (2004), Nr. 2, S. 91–110. – URL <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>. – Zugriffsdatum: 08.07.2008
- [Nieschwitz u. a. 2007] NIESCHWITZ, Alfred ; FISCHER, Max ; HABERÄCKER, Peter: *Computergrafik und Bildverarbeitung*. 2. Auflage. Wiesbaden : Vieweg, 2007. – ISBN 978-3-8348-0186-9
- [Papula 2001] PAPULA, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler Band 2. Ein Lehr- und Arbeitsbuch für das Grundstudium*. 10. Auflage. Wiesbaden : Vieweg Verlag, 2001. – ISBN 3528942371
- [Pitzke 2007] PITZKE, Marc: Auktionshaus Sotheby's: Gemälde so teuer wie Luxus-Villen, URL <http://www.spiegel.de/wirtschaft/0,1518,481287,00.html>. – Zugriffsdatum: 08.07.2008, 2007
- [Schnehagen 2004] SCHNEHAGEN, Peter: Aufnahme von Stereobildern, Deutsche Gesellschaft für Stereoskopie e.V., 2004. – URL <http://www.stereoskopie.org/Theorie/Aufd1.htm>. – Zugriffsdatum: 08.07.2008
- [Vedaldi 2006] VEDALDI, Andrea: A lightweight C++ implementation of SIFT, University of California, 2006. – URL <http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html>. – Zugriffsdatum: 08.07.2008

Anhang A.

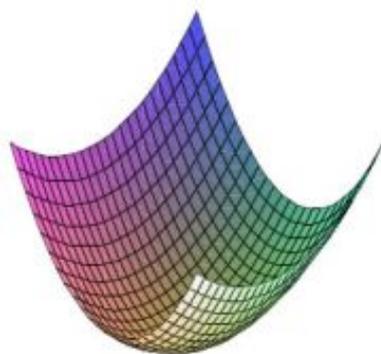
Taylorreihenentwicklung

Nachfolgend ist die Taylorreihenentwicklung für die Näherung an der jeweiligen Position des Schlüsselpunkt-Kandidaten bis zum quadratischen Term dargestellt. Diese Gleichung stellt die ausführliche Schreibweise der Gleichung 3.4 aus dem Kapitel 3.2 Merkmalspunkt Lokalisierung da:

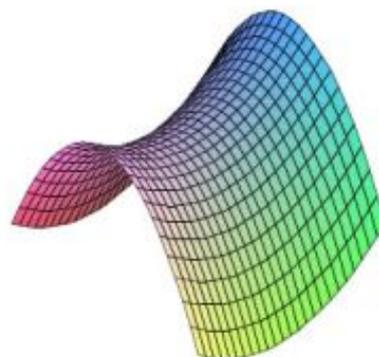
$$D(\vec{x}) = D(0, 0, 0) + \left(\frac{\partial D}{\partial x}, \frac{\partial D}{\partial y}, \frac{\partial D}{\partial \sigma} \right)_{\substack{x=0 \\ y=0 \\ \sigma=0}} \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix} + \frac{1}{2} \begin{pmatrix} (x, y, \sigma) \end{pmatrix} \begin{pmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial x \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial \sigma} & \frac{\partial^2 D}{\partial y \partial \sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{pmatrix}_{\substack{x=0 \\ y=0 \\ \sigma=0}} \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Eigenwerte

Auf der linken Seite der Abbildung [Analysis (2005)] ist eine elliptische Mulde mit einem lokalem Minimum zu erkennen. Auf der rechten Seite ist eine Sattelfläche zu sehen, die beiden Eigenwerte weisen hierbei unterschiedliche Vorzeichen auf was zu einem zurückweisen des Schlüsselpunkt-Kandidat, als kein Extremum, führen würde.



$\alpha > 0 \quad \beta > 0$



$\alpha > 0 \quad \beta < 0$

Anhang B.

Der nachfolgende Quelltext stammt aus der SIFT++ Implementierung von Andrea Vedaldi vom UCLA VisionLab (Vedaldi, 2006). Der Quelltext zeigt die Funktion, welche für das Auffinden und Evaluieren von markanten Raumpunkten zuständig ist. Zuerst werden hierbei die einzelnen DoG-Bilder durch Subtraktion von jeweils zwei benachbarten Bildern einer Oktave der Gaußpyramide erzeugt. Nachfolgend wird dann in den DoG-Bildern nach Extrema, die potenziellen Schlüsselpunkte, gesucht. Die so gefundenen Schlüsselpunkt-Kandidaten werden daraufhin einer genaueren Untersuchung unterzogen. Hierbei wird dann die genaue Position der Schlüsselpunkt-Kandidaten berechnet. Darüber hinaus werden Kandidaten mit zu geringem Kontrast oder welche, die zu nahe an schlecht wiederzuerkennenden Kanten liegen, zurückgewiesen.

```
/** @brief Sift detector
**
** The function runs the SIFT detector on the stored Gaussian scale
** space (see process()). The detector consists in three steps
**
** – local maxima detection;
** – subpixel interpolation;
** – rejection of weak keypoints (@a threhsold);
** – rejection of keypoints on edge-like structures (@a edgeThreshold).
**
** As they are found, keypoints are added to an internal list. This
** list can be accessed by means of the member functions
** getKeypointsBegin() and getKeypointsEnd(). The list is ordered by
** octave, which is useful to speed-up computeKeypointOrientations()
** and computeKeypointDescriptor().
**/
void
Sift::detectKeypoints(VL::float_t threshold, VL::float_t edgeThreshold)
{
    keypoints.clear();

    int nValidatedKeypoints = 0;
```

```

// Process one octave per time
for(int o = omin ; o < omin + O ; ++o) {

    int const xo = 1 ;
    int const yo = getOctaveWidth(o) ;
    int const so = getOctaveWidth(o) * getOctaveHeight(o) ;
    int const ow = getOctaveWidth(o) ;
    int const oh = getOctaveHeight(o) ;

    VL::float_t xperiod = getOctaveSamplingPeriod(o) ;

    // -----
    //                                     Difference of Gaussians
    // -----
    pixel_t* dog = temp ;
    templsGrad = false ;
    {
        pixel_t* pt = dog ;
        for(int s = smin ; s <= smax-1 ; ++s) {
            pixel_t* srca = getLevel(o, s ) ;
            pixel_t* srcb = getLevel(o, s+1) ;
            pixel_t* enda = srcb ;
            while( srca != enda ) {
                *pt++ = *srcb++ - *srca++ ;
            }
        }
    }

    // -----
    //                                     Find points of extremum
    // -----
    {
        pixel_t* pt = dog + xo + yo + so ;
        for(int s = smin+1 ; s <= smax-2 ; ++s) {
            for(int y = 1 ; y < oh - 1 ; ++y) {
                for(int x = 1 ; x < ow - 1 ; ++x) {
                    pixel_t v = *pt ;

#define CHECK_NEIGHBORS(CMP,SGN) \
        ( v CMP ## = SGN 0.8 * threshold && \
          v CMP *(pt + xo) && \
          v CMP *(pt - xo) && \

```

```

v CMP *(pt + so) && \
v CMP *(pt - so) && \
v CMP *(pt + yo) && \
v CMP *(pt - yo) && \
\
v CMP *(pt + yo + xo) && \
v CMP *(pt + yo - xo) && \
v CMP *(pt - yo + xo) && \
v CMP *(pt - yo - xo) && \
\
v CMP *(pt + xo + so) && \
v CMP *(pt - xo + so) && \
v CMP *(pt + yo + so) && \
v CMP *(pt - yo + so) && \
v CMP *(pt + yo + xo + so) && \
v CMP *(pt + yo - xo + so) && \
v CMP *(pt - yo + xo + so) && \
v CMP *(pt - yo - xo + so) && \
\
v CMP *(pt + xo - so) && \
v CMP *(pt - xo - so) && \
v CMP *(pt + yo - so) && \
v CMP *(pt - yo - so) && \
v CMP *(pt + yo + xo - so) && \
v CMP *(pt + yo - xo - so) && \
v CMP *(pt - yo + xo - so) && \
v CMP *(pt - yo - xo - so) )

if ( CHECK_NEIGHBORS(>,+) || CHECK_NEIGHBORS(<,-) ) {

    Keypoint k ;
    k.ix = x ;
    k.iy = y ;
    k.is = s ;
    keypoints.push_back(k) ;
}
pt += 1 ;
}
pt += 2 ;
}
pt += 2*yo ;
}
}

```

```

// -----
//                                     Refine local maxima
// -----
{ // refine
  KeypointsIter siter ;
  KeypointsIter diter ;

  for(diter = siter = keypointsBegin() + nValidatedKeypoints ;
      siter != keypointsEnd() ;
      ++siter) {

    int x = int( siter->ix ) ;
    int y = int( siter->iy ) ;
    int s = int( siter->is ) ;

    VL::float_t Dx=0,Dy=0,Ds=0,Dxx=0,Dyy=0,Dss=0,Dxy=0,Dxs=0,Dys=0 ;
    VL::float_t b [3] ;
    pixel_t* pt ;
    int dx = 0 ;
    int dy = 0 ;

    // must be exec. at least once
    for(int iter = 0 ; iter < 5 ; ++iter) {

      VL::float_t A[3*3] ;

      x += dx ;
      y += dy ;

      pt = dog
        + xo * x
        + yo * y
        + so * (s - smin) ;

#define at(dx,dy,ds) (*( pt + (dx)*xo + (dy)*yo + (ds)*so))
#define Aat(i,j)      (A[(i)+(j)*3])

      /* Compute the gradient. */
      Dx = 0.5 * (at(+1,0,0) - at(-1,0,0)) ;
      Dy = 0.5 * (at(0,+1,0) - at(0,-1,0)) ;
      Ds = 0.5 * (at(0,0,+1) - at(0,0,-1)) ;

      /* Compute the Hessian. */
      Dxx = (at(+1,0,0) + at(-1,0,0) - 2.0 * at(0,0,0)) ;

```

```

Dyy = ( at(0,+1,0) + at(0,-1,0) - 2.0 * at(0,0,0) ) ;
Dss = ( at(0,0,+1) + at(0,0,-1) - 2.0 * at(0,0,0) ) ;

Dxy = 0.25 * ( at(+1,+1,0) + at(-1,-1,0) - at(-1,+1,0) - at
  (+1,-1,0) ) ;
Dxs = 0.25 * ( at(+1,0,+1) + at(-1,0,-1) - at(-1,0,+1) - at
  (+1,0,-1) ) ;
Dys = 0.25 * ( at(0,+1,+1) + at(0,-1,-1) - at(0,-1,+1) - at
  (0,+1,-1) ) ;

/* Solve linear system. */
Aat(0,0) = Dxx ;
Aat(1,1) = Dyy ;
Aat(2,2) = Dss ;
Aat(0,1) = Aat(1,0) = Dxy ;
Aat(0,2) = Aat(2,0) = Dxs ;
Aat(1,2) = Aat(2,1) = Dys ;

b[0] = - Dx ;
b[1] = - Dy ;
b[2] = - Ds ;

// Gauss elimination
for(int j = 0 ; j < 3 ; ++j) {

    // look for leading pivot
    VL::float_t maxa = 0 ;
    VL::float_t maxabsa = 0 ;
    int maxi = -1 ;
    int i ;
    for(i = j ; i < 3 ; ++i) {
        VL::float_t a = Aat(i,j) ;
        VL::float_t absa = fabsf( a ) ;
        if ( absa > maxabsa ) {
            maxa = a ;
            maxabsa = absa ;
            maxi = i ;
        }
    }

    // singular?
    if( maxabsa < 1e-10f ) {
        b[0] = 0 ;
        b[1] = 0 ;
    }
}

```

```

        b[2] = 0 ;
        break ;
    }

    i = maxi ;

    // swap j-th row with i-th row and
    // normalize j-th row
    for(int jj = j ; jj < 3 ; ++jj) {
        std::swap( Aat(j, jj) , Aat(i, jj) ) ;
        Aat(j, jj) /= maxa ;
    }
    std::swap( b[j], b[i] ) ;
    b[j] /= maxa ;

    // elimination
    for(int ii = j+1 ; ii < 3 ; ++ii) {
        VL::float_t x = Aat(ii, j) ;
        for(int jj = j ; jj < 3 ; ++jj) {
            Aat(ii, jj) -= x * Aat(j, jj) ;
        }
        b[ii] -= x * b[j] ;
    }
}

// backward substitution
for(int i = 2 ; i > 0 ; --i) {
    VL::float_t x = b[i] ;
    for(int ii = i-1 ; ii >= 0 ; --ii) {
        b[ii] -= x * Aat(ii, i) ;
    }
}

/* If the translation of the keypoint is big, move the keypoint
 * and re-iterate the computation. Otherwise we are all set.
 */
dx= ((b[0] > 0.6 && x < ow-2) ? 1 : 0 )
+ ((b[0] < -0.6 && x > 1 ) ? -1 : 0 ) ;

dy= ((b[1] > 0.6 && y < oh-2) ? 1 : 0 )
+ ((b[1] < -0.6 && y > 1 ) ? -1 : 0 ) ;

if( dx == 0 && dy == 0 ) break ;

```

```

    }

    // Accept-reject keypoint
    {
        VL::float_t val = at(0,0,0) + 0.5 * (Dx * b[0] + Dy * b[1] + Ds
            * b[2]) ;
        VL::float_t score = (Dxx+Dyy)*(Dxx+Dyy) / (Dxx*Dyy - Dxy*Dxy) ;
        VL::float_t xn = x + b[0] ;
        VL::float_t yn = y + b[1] ;
        VL::float_t sn = s + b[2] ;

        if (fast_abs(val) > threshold &&
            score < (edgeThreshold+1)*(edgeThreshold+1)/edgeThreshold &&
            score >= 0 &&
            fast_abs(b[0]) < 1.5 &&
            fast_abs(b[1]) < 1.5 &&
            fast_abs(b[2]) < 1.5 &&
            xn >= 0 &&
            xn <= ow-1 &&
            yn >= 0 &&
            yn <= oh-1 &&
            sn >= smin &&
            sn <= smax ) {

            diter->o = o ;

            diter->ix = x ;
            diter->iy = y ;
            diter->is = s ;

            diter->x = xn * xperiod ;
            diter->y = yn * xperiod ;
            diter->s = sn ;

            diter->sigma = getScaleFromIndex(o,sn) ;

            ++diter ;
        }
    }
} // next candidate keypoint

// prepare for next octave
keypoints.resize( diter - keypoints.begin() ) ;

```

```
        nValidatedKeypoints = keypoints.size() ;  
    } // refine block  
  
} // next octave  
}
```

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 10. Juli 2008

Ort, Datum

Unterschrift