

# Mirroring and matching of the SIFT-feature descriptors for detection of symmetries and repeated structures in images

Susanne Wenzel

`susanne.wenzel@uni-bonn.de`

TR-IGG-P-2007-04

August 01, 2007



Technical Report Nr. 04, 2007

Department of Photogrammetry  
Institute of Geodesy and Geoinformation  
University of Bonn

Available at  
<http://www.ipb.uni-bonn.de/technicalreports/>

# Spiegelung und Zuordnung der SIFT-Feature Deskriptoren für die Detektion von Symmetrien und wiederholten Strukturen in Bildern

Susanne Wenzel

susanne.wenzel@uni-bonn.de

August 01, 2007

## Abstract

Dieser Bericht geht auf die Details zur Spiegelung von SIFT-Feature Deskriptoren ein. Es wird gezeigt, wie durch einfaches Umsortieren der Elemente des Feature Deskriptors gespiegelte Versionen der Deskriptoren erlangt werden können.

Des Weiteren wird erläutert, wie Features innerhalb eines Bildes zugeordnet werden können. Die Besonderheit dieser Aufgabenstellung liegt in der gesuchten Zuordnung nicht eines - des besten - Matches, sondern in der Zuordnung aller Matches in einem Bild.

Die vorgestellten Methoden basieren auf [Wenzel, 2006].

*This report describes the details for mirroring the descriptors of the SIFT-features. We show how the mirrored versions are derived by simply resorting the descriptor elements.*

*Furthermore, we describe the matching of features within an image. The peculiarity of this task is the search for more than one - the best - match within an single image. The presented methods are based on the work of [Wenzel, 2006]. After the introduction the functionality of the SIFT-feature detector is drafted and the development of the descriptors is described in detail. The following sections describe the details of mirroring and matching the features.*

## 1 Einleitung

Der SIFT Operator ist ein vielverwendetes Werkzeug zur Detektion rotations- und skaleninvarianter Features in Bildern. Ergebnis der Featuredetektion sind für jeden Punkt die Bildkoordinaten, Orientierung und Skala des Features,

sowie der Deskriptor, der die lokale Umgebung des Features in einem 128 dimensionalem Vektor kodiert.

In [Wenzel et al., 2007a] und [Wenzel et al., 2007b] wurden Verfahren zur Detektion multipler Symmetrien sowie multipler wiederholter Strukturen in Bildern vorgestellt.

Für die Detektion von Symmetrien werden die SIFT Features eines Bildes mit ihren gespiegelten Kopien gematcht. Durch Clustering der Symmetrieachsen der zugeordneten Features werden im Anschluss die dominanten Symmetrien des Bildes abgeleitet.

Das Verfahren zur Detektion multipler wiederholter Strukturen basiert auf der gleichen Idee. Die Spiegelung der Deskriptoren entfällt hier. Beibehalten wird die Idee der Zuordnung der Features innerhalb des Bildes. Auch hier führt ein Clusterverfahren über die Translationen zwischen den gematchten Features zu den dominanten Verschiebungen im Bild.

Im folgenden soll kurz die Funktionsweise des SIFT Operators skizziert, sowie die Entstehung des Deskriptors erläutert werden. Es folgen die Beschreibung der Spiegelung der Deskriptoren sowie die Erläuterungen zum multiplen Matching der Features.

## 2 Der SIFT-Operator

Lowe hat mit der “Scale Invariant Feature Transform” (SIFT) eine Methode entwickelt, mit der aus Bilddaten skalen-invariante Interessenspunkte relativ zu lokalen Features erzeugt werden. Die Features sind stabil über verschiedenen Bildmaßstäbe (Skalen), invariant bezüglich Rotation und robust bezüglich Beleuchtung und Kameraposition. Details zur Bestimmung der SIFT-Features sind in [Lowe, 2004] nachzulesen.

Über alle Skalen und Bildpositionen werden die Extrema des Skalen-Raumes, eine kontinuierliche Funktion der Skala bestimmt. Als Kern dieser Funktion dient die Gauß-Funktion in Abhängigkeit von  $\sigma$ . Eine Veränderung des Maßstabs wird also durch eine Glättung des Bildes simuliert.

Durch Differenzenbildung von Gauß-Funktionen werden potentiell interessante Punkte identifiziert, die invariant bezüglich Maßstab und Orientierung sind. Für jeden dieser Punkte werden Position  $(x, y)$  und Maßstab  $s$  bestimmt und anhand ihrer Stabilität über alle Skalen geeignete Keypoints selektiert.

Basierend auf den Gradientenrichtungen der Bildpunkte einer kleinen Umgebung um den betrachteten Punkt wird die Hauptorientierung  $\phi$  dieses Features bestimmt.

Bei Identifizierung mehrerer signifikanter Orientierungen, bspw. an Ecken, werden an dieser Stelle mehrere Keypoints mit gleicher Position aber unterschiedlicher Orientierung  $\phi$  erzeugt.

Die Beschreibung der Umgebung des Keypoints durch die Deskriptoren erfolgt bezüglich dieser ermittelten Orientierung, dadurch wird die erwähnte Rotationsinvarianz erreicht. Bei mehreren Keypoints an einer Position, aufgrund verschiedener signifikanter Gradientenrichtungen, unterscheiden sich also auch die Deskriptoren, obwohl der gleiche Bildausschnitt beschrieben wird.

### 3 Die Deskriptoren

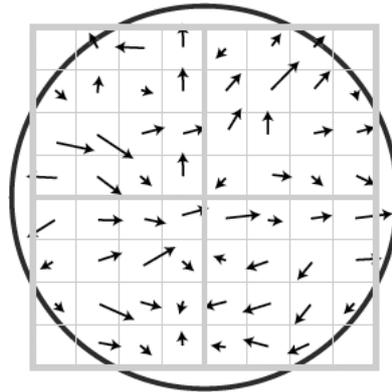
Die Entstehung der Deskriptoren bedarf einer genaueren Erläuterung, da diese für das Spiegeln der Features genau verstanden werden muss.

Abb. 1 illustriert die Entstehung des 2x2 Deskriptors für eine 8x8 Pixel Umgebung eines Keypoints. Für die Pixel der Umgebung des Keypoints werden die Richtungen und Beträge der Gradienten bestimmt. Der zuvor bestimmte Maßstab  $s$  des Keypoints definiert dabei die Stärke der gaußschen Glättung des Bildausschnittes. Für die erwähnte Rotationsinvarianz des Deskriptors werden alle Gradientenrichtungen bezüglich der Orientierung  $\phi$  des Keypoints bestimmt. Alle Gradienten werden mit einer weiteren Gauß-Funktion gewichtet, mit  $\sigma$  abhängig von der Größe der betrachteten Umgebung, illustriert durch den Kreis in Abb. 1(a). Dieses Vorgehen sorgt dafür, dass Gradienten nahe am Keypoint ein größeres Gewicht im Deskriptor bekommen als solche die ganz am Rand des Deskriptor-Fensters liegen. Dieses Fenster wird nun in Subregionen von 4x4 Pixeln unterteilt und die Gradienten dieser Regionen in Histogrammen mit je 8 Werten akkumuliert. Abb. 1(b) zeigt diesen Schritt für einen 2x2 Deskriptor. Die Pfeile zeigen die repräsentierte Richtung des Histogramms mit der Summe des jeweiligen Histogrammeintrages als Länge des Pfeiles.

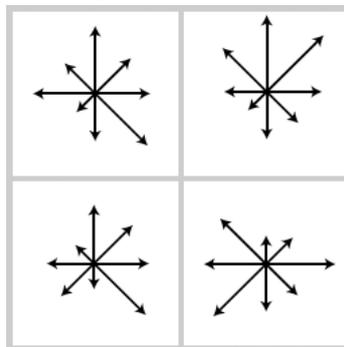
Der Deskriptor entsteht nun als Vektor mit den Werten der Histogrammeinträge, den Längen der Pfeile in Abb. 1(b), als Vektorelemente. Für den in Abb. 1 dargestellten Deskriptor ist das ein Vektor der Länge 32. Die Experimente in [Lowe, 2004] haben ergeben, dass die besten Ergebnisse mit Deskriptoren einer 16x16 Umgebung, akkumuliert in 4x4 Subregionen-Histogrammen zu erzielen sind. Diese Deskriptoren sind dann Vektoren mit  $4 \cdot 4 \cdot 8=128$  Elementen.

Schließlich wird dieser Deskriptor noch auf den Betrag 1 normiert, um Robustheit bezüglich Beleuchtung bzw. Kontrast zu erreichen.

Der Algorithmus wird von Lowe als C-Code und auch als Matlab - Implemen-



(a) Gradienten der Bildpunkte in der Umgebung des Keypoints



(b) Keypoint-Deskriptor

Abbildung 1: Entstehung des Keypoint-Deskriptors aus den Gradienten der Umgebung des Keypoints. Hier für einen 2x2 Deskriptor, entstanden aus den Gradienten einer 8x8 Pixel Umgebung des Keypoints.

tierung zur Verfügung gestellt <sup>1</sup> und wurde als solcher in [Wenzel et al., 2007a] und [Wenzel et al., 2007b] direkt verwendet. <sup>2</sup>

Mit diesen Funktionen ergibt sich also für jeden Keypoint  $\mathbf{p}$ :

- **Position  $\mathbf{l}$**  (Location) mit Zeile, Spalte, Maßstab und Orientierung

$$\mathbf{l} = [\text{row}, \text{col}, s, \phi]$$

- **Deskriptor  $\mathbf{d}$**   $[1 \times 128]$ ,  $|\mathbf{d}| = 1$

## 4 Spiegelung der Deskriptoren

Im folgenden soll das Vorgehen zur Bestimmung der gespiegelten Feature Deskriptoren näher erläutert werden.

Zunächst lässt sich feststellen, dass die Features nicht mehr bezüglich ihrer Orientierung normalisiert werden müssen. Die Beschreibung der Umgebung der Features durch die Deskriptoren bezieht sich schon auf die Orientierung  $\phi$  des Lowe-Punktes. Durch diese Normalisierung ist die Wahl der Spiegelachse egal.

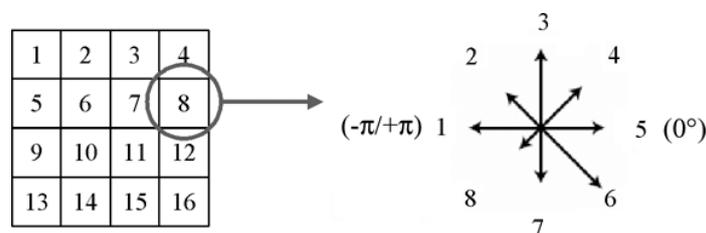
Ein sehr intuitives Vorgehen zum Spiegeln der Features wäre sicherlich, das Bild zu spiegeln und Features in dieser gespiegelten Version zu detektieren. Nach Zuordnung der korrespondierenden Features beider Versionen ergibt sich zu jedem originalen Feature-Deskriptor  $\mathbf{k}_i$  eine gespiegelte Version  $\mathbf{m}_i$ . Hier wird aber einen zweiten Durchlauf des Feature-Detektors benötigt und es wäre zu untersuchen, welchen Effekt die Bildspiegelung auf die Ergebnisse des Detektors hat. Auch die Suche nach den korrespondierenden Punktpaaren macht diese Methode im Vergleich zu folgendem Vorschlag aufwendig.

Die Deskriptoren beschreiben die lokalen Umgebung des Features in kodierter Form. Bei genauer Kenntnis des verwendeten Feature-Detektors insbesondere der Entstehung der Deskriptoren kann demzufolge eine gespiegelte Version des Features durch Manipulation, d.h. Umsortierung der Elemente des Deskriptorvektors erlangt werden.

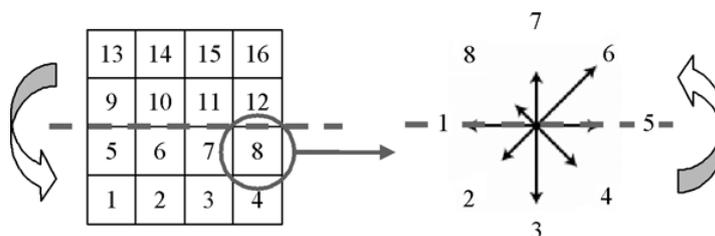
In der Kommentierung zum Lowe-Code ist leider nicht verzeichnet, in welcher Reihenfolge die Elemente der Histogramme abgespeichert werden. Für

<sup>1</sup>[www.cs.ubc.ca/~lowe/keypoints/](http://www.cs.ubc.ca/~lowe/keypoints/); Stand: 04.11.2006

<sup>2</sup>Die auf der angegebenen Webseite veröffentlichte Software ist ausdrücklich nur für Forschungszwecke zur Verfügung gestellt. Die Software ist durch ein US Patent geschützt. Für eine kommerzielle Nutzung muss eine Lizenz von der University of British Columbia erworben werden.



(a) Original. Links: Die 4x4 Subregionen des Deskriptor-Fensters; Rechts: Die Histogrammeinträge einer Subregion, von  $-\pi$  bis  $+\pi$  im Uhrzeigersinn.



(b) Spiegelung an der horizontalen Achse. Links: Die Reihen der 4x4 Subregionen von oben nach unten getauscht/gespiegelt; Rechts: Auch die Histogrammeinträge werden von oben nach unten getauscht/gespiegelt.

Abbildung 2: Spiegelung des Deskriptors

ein „einfaches“ Matching von Keypoints ist diese Information auch nicht von Interesse, es reicht zu wissen, dass der Deskriptor für alle Keypoints gleich aufgebaut ist.

Durch einige kleine Versuche mit den Ergebnissen des SIFT-Operators hat sich das in Abb. 2 dargestellte Vorgehen für die Spiegelung des Deskriptors ergeben. Die Subregionen des Deskriptor-Fensters sind offensichtlich zeilenweise und die Elemente der Histogrammeinträge pro Subregion von  $-\pi$  bis  $+\pi$  im Uhrzeigersinn abgespeichert, siehe Abb. 2(a). Es müssen also zunächst die 8er-Blöcke der Orientierungshistogramme derart umsortiert werden, dass die Blöcke zeilenweise vertikal gespiegelt werden, wie in Abb. 2(b) links dargestellt. Und auch die Histogramme selbst werden vertikal gespiegelt, d.h. die Einträge werden umsortiert, wie in Abb. 2(b) rechts dargestellt.

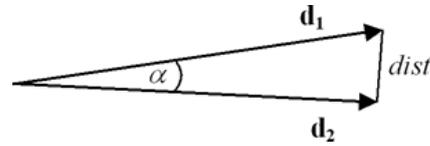
## 5 Matching der Feature-Deskriptoren

Die Feature-Deskriptoren beschreiben die Umgebung eines kleinen Bildpatches um den detektierten Bildpunkt. Beim Matching soll nun entschieden

werden, wie gut zwei solcher Patches übereinstimmen, bzw. der Bildausschnitt mit der besten Übereinstimmung zum betrachteten Feature-Deskriptor gefunden werden. Der beste Kandidat dafür ergibt sich aus dem nächsten Nachbarn in dem 128-dimensionalen Raum der Deskriptoren. Dieser lässt sich leicht identifizieren als der Deskriptor  $\mathbf{d}_i$  mit der geringsten euklidischen Distanz zum betrachteten Feature-Deskriptor  $\mathbf{d}$  [Lowe, 2004].

$$dist = \sqrt{\langle \mathbf{d} - \mathbf{d}_i, \mathbf{d} - \mathbf{d}_i \rangle} \rightarrow \min_i, \quad i = 1 \dots n \quad (1)$$

In Anlehnung an das Vorgehen in der Demo-Software zum SIFT-Operator, wird die Berechnung der euklidischen Distanzen aus Performance Gründen ersetzt durch eine Approximation der Winkel zwischen den betrachteten Deskriptor-Vektoren. Da die Deskriptoren auf 1 normiert sind und gerade die kleinen Abstände gesucht werden, ist das eine gute Näherung.



$$\alpha \cong \arccos \langle \mathbf{d}_1, \mathbf{d}_2 \rangle \quad \text{für kleine } \alpha \text{ und } |\mathbf{d}_1| = |\mathbf{d}_2| = 1 \quad (2)$$

Für die Bestimmung der euklidischen Abstände zwischen allen  $n$  Deskriptoren nach Gl. (1) sind  $n!$  Differenzen zu berechnen. Für die Näherung aus Gl. (2) dagegen kann die Berechnung aller Skalarprodukte in einer Matrixoperation erfolgen. Nach Bildung des  $\arccos$  ist von der resultierenden  $[n \times n]$  Distanz-Matrix dann nur noch die obere Dreiecksmatrix von Interesse, da jedes Feature-Paar doppelt registriert ist.

Da es nicht zu jedem Feature einen passenden Match geben muss, ein globaler Schwellwert für die Distanz zum nächsten Feature aber nicht sinnvoll ist, wurde in [Lowe, 2004] das Verhältnis zwischen nächstem und zweitnächstem Nachbarn als Bewertungsmaßstab vorgeschlagen. Unter der Voraussetzung, dass das Objekt nur unterschiedliche Objektbestandteile enthält, d.h. immer nur *ein* passender Match zu erwarten ist, wird die euklidische Distanz zum nächsten Nachbarn signifikant kleiner sein, als die zum nächsten inkorrekten Match. Falls kein korrekter Match vorhanden ist, werden alle Distanzen ähnlich kleine Abstände voneinander haben. Ein Match wird nur dann als solcher akzeptiert, wenn die Distanz zum nächsten Nachbarn  $distRatio$  mal größer ist als die zum zweitnächsten Nachbarn<sup>3</sup>.

<sup>3</sup>In [Lowe, 2004] wurde bspw. für  $distRatio$  ein Wert von 0.8 gewählt.

In der hier vorliegenden Anwendung muss aber beachtet werden, dass nicht nur *ein* Match (der *beste*) pro Keypoint gesucht wird, sondern möglichst viele Matches.

Man stelle sich vor, es wird zu einem Keypoint nur der eine beste Match im Bild gesucht. Dieser ist, wie in [Lowe, 2004] beschrieben, gefunden, wenn das Verhältnis der euklidischen Distanzen des gefundenen Punktes zum Punkt mit der nächstgrößten Distanz *distRatio* überschreitet. Man stelle sich desweiteren vor, dieser gefundene Punkt wird beim weiteren Vorgehen ignoriert, dass entsprechende Bildpatch beispielsweise geschwärzt, und ein neuer Match gesucht. Für diesen Match gilt das gleiche Kriterium wie zuvor. Dieses Vorgehen wird wiederholt, bis für den aktuell betrachteten Keypoint kein Match im Bild mehr gefunden wird. Diese Überlegungen führen zu folgendem Vorgehen bei der Suche aller Matches eines Keypoints.

**Die euklidischen Distanzen aller Features zu dem aktuell betrachteten Keypoint werden aufsteigend sortiert. Es werden alle benachbarten Matches bis zum letzten auftretenden Paar, dessen Verhältnis *distRatio* überschreitet als Matches akzeptiert.**

*The feature descriptors describe the local image patch of the features. They are coded as a 128 dimensional vector. The distance between two features is given by the euclidean distance according equations 1 and 2, respectively. We sort the distances to all features to find the matches of one keypoint. All values are accepted as a match, where the ratio to the next value is as greater as *distRatio*.*

## Literatur

- [Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110. [3, 4, 8, 9]
- [Wenzel, 2006] Wenzel, S. (2006). Diplomarbeit. Detektion wiederholter und symmetrischer Strukturen von Objekten in Bildern. Master's thesis, Institut für Geodäsie und Geoinformation, Fachbereich Photogrammetrie, Rheinische Friedrich-Wilhelms-Universität Bonn. [2]
- [Wenzel et al., 2007a] Wenzel, S., Drauschke, M., and Förstner, W. (2007a). Detection of repeated structures in facade images. *eingereicht für die Zeitschrift der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung DGPF Photogrammetrie - Fernerkundung - Geoinformation*. [3, 6]

- [Wenzel et al., 2007b] Wenzel, S., Drauschke, M., and Förstner, W. (2007b). Detection of repeated structures in facade images (accepted/in print). In *Proceedings of the OGRW-7-2007, 7th Open German/Russian Workshop on Pattern Recognition and Image Understanding. August 2003, 2007. Ettlingen, Germany.* [3, 6]