



DIPLOMSTUDIENGANG VERMESSUNGSWESEN

DIPLOMARBEIT

VON

SUSANNE WENZEL

GEBOREN AM 26. APRIL 1978 IN BERLIN

DETEKTION WIEDERHOLTER UND SYMMETRISCHER  
STRUKTUREN VON OBJEKTEN IN BILDERN

BONN 2006

BETREUER:

PROF. DR.-ING. WOLFGANG FÖRSTNER  
DIPL. INFORM. MARTIN DRAUSCHKE

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN  
INSTITUT FÜR GEODÄSIE UND GEOINFORMATION  
FACHBEREICH PHOTOGRAMMETRIE



## **Diplomaufgabe**

für Frau cand. geod. Susanne Wenzel

### **Detektion wiederholter und symmetrischer Strukturen**

#### **von Objekten in Bildern**

Sich wiederholende bzw. symmetrische Strukturen sind Hinweise auf künstliche Objekte, führen aber auch zu Schwierigkeiten bei klassischen Bildzuordnungsverfahren. Die Suche und Gruppierung zusammengehöriger Features kann daher zur Identifikation künstlicher Objekte oder zur Verbesserung von Zuordnungsverfahren dienen. Darüber hinaus kann man aus einem Bild eines im Raum symmetrischen Objekts auf die 3D-Struktur dieses Objekts schließen.

Die Diplomarbeit soll das von Loy und Eklundh auf der ECCV 2006 vorgestellte Verfahren<sup>1</sup> zur Detektion symmetrischer und wiederholter Bildbereiche implementieren und hinsichtlich seiner Verwendbarkeit für photogrammetrische Gebäudeaufnahmen überprüfen. Insbesondere geht es um die Detektierbarkeit regelmäßiger Fassadenstrukturen in Abhängigkeit von ihrer Komplexität. Darüber hinaus ist zu klären, wie mehrfache Symmetrien identifiziert und ggf. für die 3D-Rekonstruktion des regelmäßigen Teils der Fassadenstruktur genutzt werden können.

Prof. Dr.-Ing. Wolfgang Förstner

Betreuer: W. Förstner, M. Drauschke

Ausgegeben am: 18. September 2006

Abgabetermin: 18. Dezember 2006

Abgegeben am:

---

<sup>1</sup>G. Loy, J.-O. Eklundh, Detecting Symmetry and Symmetric Constellations of Features, ECCV 2006, Part II, LNCS 3952, pp. 508-521, Springer-Verlag Berlin Heidelberg 2006

## ERKLÄRUNG

Hiermit erkläre ich eidesstattlich, dass ich die vorgelegte Arbeit selbständig und ohne unzulässige Hilfe angefertigt habe. Es wurden keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Euskirchen, den 14. Dezember 2006

Susanne Wenzel



# Inhaltsverzeichnis

<b>Einleitung – Motivation und Aufgabenstellung</b>	<b>6</b>
1.1 Einleitung . . . . .	6
1.2 Motivation . . . . .	8
1.3 Aufgabenstellung . . . . .	9
1.4 Verwandte Arbeiten . . . . .	9
1.5 Gliederung der Arbeit . . . . .	10
<b>Detektion von Symmetrien und wiederholten Strukturen – Grundlagen und Konzepte</b>	<b>11</b>
2.1 Modellbildung . . . . .	11
2.1.1 Objektmodell . . . . .	11
2.1.2 Sensor- und Bildmodell . . . . .	11
2.2 Begriffe . . . . .	12
2.2.1 Symmetrie . . . . .	12
2.2.2 Wiederholte Strukturen . . . . .	13
2.2.3 Features . . . . .	13
2.3 Detektion von Symmetrien . . . . .	14
2.3.1 Der SIFT-Operator . . . . .	14
2.3.2 Bestimmung von Symmetrien in Bildern . . . . .	16
2.4 Detektion von wiederholten Strukturen . . . . .	21
2.5 Ermittlung von Mustern in wiederholten Strukturen . . . . .	23
2.5.1 Modellannahmen . . . . .	23
2.5.2 Problembeschreibung . . . . .	24
2.5.3 Das Vorgehen . . . . .	25
2.6 Bestimmung der Relativen Orientierung aus einem Bild eines Objektes mit spiegelsymmetrischen Eigenschaften . . . . .	30
<b>Umsetzung – Implementierung und Ergebnisse</b>	<b>33</b>
3.1 Feature-Punkte . . . . .	33
3.2 Detektion von Symmetrien . . . . .	34
3.2.1 Implementation . . . . .	34
3.2.2 Spiegelung der Deskriptoren . . . . .	35
3.2.3 Matching der Feature-Deskriptoren . . . . .	35
3.2.4 Bewertung der Symmetrie einzelner Feature-Paare . . . . .	36
3.2.5 Ermittlung dominanter Symmetrien im Hough-Raum . . . . .	37
3.2.6 Ergebnisse . . . . .	39
3.3 Detektion von wiederholten Strukturen . . . . .	48
3.3.1 Implementation . . . . .	48
3.3.2 Ergebnisse . . . . .	48
3.4 Ermittlung von Mustern in wiederholten Strukturen . . . . .	53
3.4.1 Implementation . . . . .	53
3.4.2 Ergebnisse . . . . .	57

<b>Zusammenfassung und Ausblick – Was erreicht wurde und was noch fehlt</b>	<b>65</b>
4.1 Zusammenfassung . . . . .	65
4.2 Ausblick . . . . .	65
<b>Anhang</b>	<b>66</b>
<b>A Herleitungen</b>	<b>67</b>
A.1 Die mittlere Symmetrieachse . . . . .	67
A.2 Die mittlere Translation . . . . .	69
<b>Literaturverzeichnis</b>	<b>69</b>
<b>Stichwortverzeichnis</b>	<b>72</b>

# 1. Einleitung

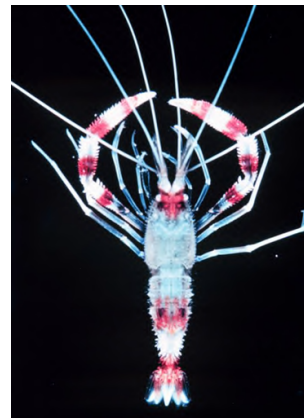
## Motivation und Aufgabenstellung

### 1.1 Einleitung

Die menschliche Wahrnehmung ist sehr gut abgestimmt auf die Detektion von Symmetrien und Regelmäßigkeiten. Zum einen empfindet der Mensch solche Objekte als sehr ästhetisch, zum anderen ermöglicht es die schnelle Erfassung der Struktur komplexer Objekte. Symmetrien und wiederholte Strukturen treten überall auf, sowohl natürlich als auch von Menschen verursacht.



(a)

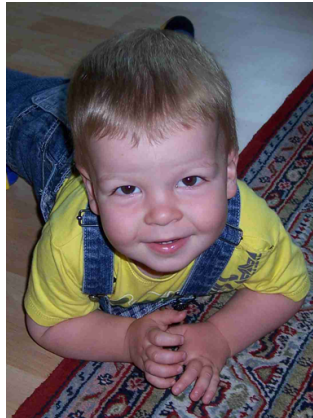


(b)

Abbildung 1.1: Beispiele aus der Biologie für symmetrisch aufgebaute Lebewesen. (a) Schirmquallen sind ihrem Grundbauplan nach radialsymmetrisch gebaut. ([http://upload.wikimedia.org/wikipedia/commons/8/84/Chrysaora\\_jelly.jpg](http://upload.wikimedia.org/wikipedia/commons/8/84/Chrysaora_jelly.jpg)) (b) Scherengarnelen sind bilateralsymmetrisch gebaute Gewebetiere, bei denen die linke spiegelbildlich zur rechten Körperhälfte aufgebaut ist. ([http://upload.wikimedia.org/wikipedia/commons/f/f9/Stenopus\\_hispidus.jpg](http://upload.wikimedia.org/wikipedia/commons/f/f9/Stenopus_hispidus.jpg))

In der Natur finden sich Lebewesen, die in ihrer Physiologie komplett symmetrisch aufgebaut sind, zwei Beispiele zeigt Abb. 1.1. Durch die Beweglichkeit der Extremitäten der Tiere sind diese Symmetrien hier nicht immer gleich sichtbar. Abb. 1.2 zeigt zwei weitere symmetrische Beispiele aus der Natur. Hier bleibt die Symmetrie in jedem Fall erhalten, auch wenn der Schmetterling beispielsweise die Flügel bewegt oder das Kind den Kopf neigt. Die Tatsache, dass beide Objekte durch genau eine Symmetrieachse gekennzeichnet sind, unterscheidet sie von symmetrischen Objekten, wie sie typischerweise von Menschen erstellt werden.

Diese Beispiele zeigen Symmetrien anhand natürlicher Objekte, wie wir sie auch für von Men-



(a) Gesicht



(b) Schmetterling

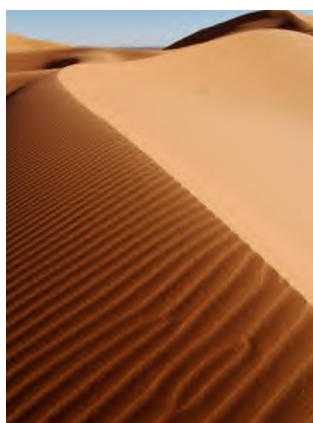
Abbildung 1.2: Beispiele für natürliche spiegelsymmetrische Objekte.

schen gemachte Objekte verstehen würden.

Abb. 1.3 zeigt zwei Beispiele aus der Natur für regelmäßige Strukturen. Ein Zebra ist gekennzeichnet durch seine Streifen. Diese sind aber nicht, wie etwa bei einem Zebrastrifen am Fußgängerüberweg, gleich lang, gleich dick und gleich abständig. Die Riffeln auf der windabgewandten Seite einer Düne zeigen ein regelmäßiges Muster. Aber auch hier sind die Streifen typischerweise unterschiedlich geformt.

Im Vergleich dazu zeigt die Abb. 1.4 drei typische Beispiele für von Menschen gemachte regelmäßige und symmetrische Objekte. Zunächst sind symmetrische Objekte hier typischerweise durch sehr viel mehr als nur eine Symmetrieachse gekennzeichnet. Unter Regelmäßigkeit wird dabei, im Gegensatz zu den eben gesehen Beispielen, die Wiederholung exakt gleicher Objektpatches in immer gleichen Abständen verstanden.

Diese Arbeit beschäftigt sich mit der letzten Gruppe der gezeigten Beispiele, insbesondere mit Symmetrien und Regelmäßigkeiten in Fassaden.



(a)



(b)

Abbildung 1.3: Beispiele für natürliche regelmäßige Objekte. (a) Durch den Wind geschaffene regelmäßige Muster auf einer Wüstendüne. ([http://de.wikipedia.org/wiki/Bild:Morocco\\_Africa\\_Flickr\\_Rosino\\_December\\_2005\\_84514010\\_edited\\_by\\_Buchling.jpg](http://de.wikipedia.org/wiki/Bild:Morocco_Africa_Flickr_Rosino_December_2005_84514010_edited_by_Buchling.jpg)) (b) Zebra mit seinem typischen Streifenmuster. (<http://www.shoarns.com/ZebraGallery.html>)

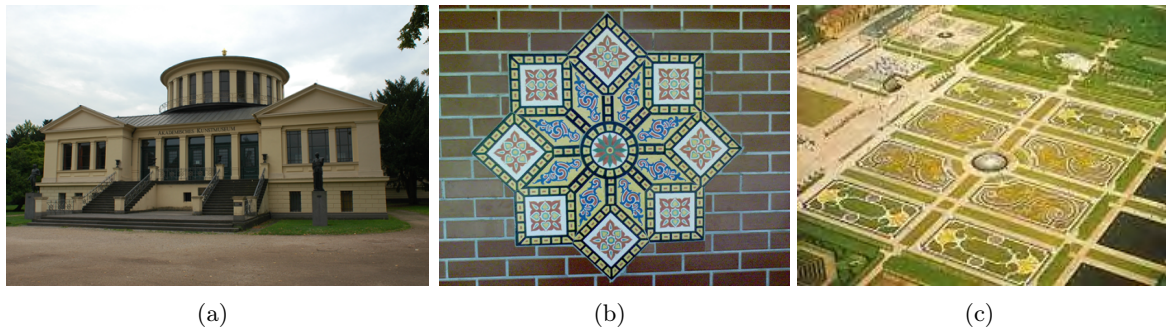


Abbildung 1.4: Beispiele für regelmäßige, symmetrische, von Menschen gemachte Objekte. (a) Akademisches Kunstmuseum Bonn (b) Drehsymmetrische Wandfliese in der Stuttgarter Wilhelma ([http://de.wikipedia.org/wiki/Bild:Fliese\\_Wilhelma.jpg](http://de.wikipedia.org/wiki/Bild:Fliese_Wilhelma.jpg)) (c) Barockgarten Herrenhausen (<http://media.wundermedia.de/events/41herrenhausenbarockgarte.jpg>)

## 1.2 Motivation

Computer Vision<sup>1</sup> beschäftigt sich mit der Erkennung von Objekten in (Digital-) Bildern und der Rekonstruktion ihrer räumlichen und physikalischen Merkmale. Ein zentrales Problem stellt dabei die Zuordnung dar. Dabei sollen korrespondierende Merkmale in verschiedenen Bildern einander zugeordnet werden, etwa bei der Stereobildanalyse zur Rekonstruktion der relativen Orientierung und der Objektkoordinaten.

In [Vosselman, 1991] werden drei Abstraktionsebenen der Zuordnung unterschieden. Die niedrigste Ebene repräsentiert das Bild durch seine Pixel bzw. ihre Grauwerte. Auf der nächst höheren Ebene erfolgt die Zuordnung aufgrund extrahierter Merkmale wie markante Punkte oder Linien und auf der höchsten Beschreibungsebene erfolgt die Zuordnung aufgrund der topologischen und geometrischen Relationen zwischen den Merkmalen. Man kann sich leicht vorstellen, dass die Zuordnung zweier Bilder eines Objektes, das durch wiederholte Strukturen gekennzeichnet ist, beispielsweise ein Schachbrett oder eine Fensterfront, auf der Pixelebene nicht trivial ist. Es finden sich immer die gleichen Bildmerkmale an vielen verschiedenen Stellen des Bildes wieder, was die Zuordnung korrespondierender Punktpaare unmöglich macht. Wenn man aber die Struktur des Objektes erfassen kann, z.B. durch eine kompakte Beschreibung der vorliegenden Symmetrien oder wiederholten Muster, kann man das Matching nicht nur ermöglichen, sondern auch noch effizienter gestalten.

Damit ergibt sich eine andere typische Anwendung der Detektion von Symmetrien und Gruppierungen in der effizienten Suche nach Bilddaten in Datenbanken mit Hilfe so genannter High-Level-Features.

In aktuellen Forschungsprojekten geht es z.B. darum, Häuser in Bildern automatisch zu erkennen. Wegen der Vielfalt und Komplexität von Gebäuden finden herkömmliche Objekterkennungsmethoden hier keine Anwendung. Urbane Gebiete und Häuser im Speziellen, sind nun aber durch wiederholte und symmetrische Strukturen gekennzeichnet, weshalb die Detektion solcher Muster in Bildern wichtige Hinweise auf von Menschen erzeugte Objekte gibt. Bei Fassaden findet man typischerweise horizontale und vertikale Symmetrien und mit den Fenstern wenigstens ein komplexes Merkmal, das sich mehrmals horizontal und vertikal innerhalb einer Fassade wiederholt. Damit bietet die Detektion komplexer regelmäßiger Strukturen zum einen den wichtigen Hinweis auf die Existenz eines künstlichen Objektes im Bild und zum anderen mit der kompakten Bildbeschreibung ein High-Level-Feature, das für den Prozess des modellbasierten Lernens von Häusern aus tausenden Beispielen verwendet werden kann.

<sup>1</sup>engl. für Rechnersehen, Maschinelles Sehen oder auch Bildverstehen

Weitere Beispiele für die Anwendung von Symmetrie-Detektions-Verfahren finden sich in der Gesichts-Bildanalyse (-erkennung), der Fahrzeug-Detektion und der 3D-Rekonstruktion [Loy and Eklundh, 2006]. In [Schaffalitzky and Zisserman, 2000] wird gezeigt, wie die gewonnenen Informationen über wiederholte Muster genutzt werden können, um aus der perspektiven Abbildung der Objekte, Informationen über ihre Form und Position abzuleiten.

### 1.3 Aufgabenstellung

Ziel dieser Arbeit ist es, das von Loy und Eklundh auf der ECCV 2006 [Loy and Eklundh, 2006] vorgestellte Verfahren zur Detektion symmetrischer und wiederholter Bildbereiche zu implementieren und hinsichtlich seiner Verwendbarkeit für photogrammetrische Gebäudeaufnahmen zu überprüfen. Insbesondere soll die Detektierbarkeit regelmäßiger Fassadenstrukturen in Abhängigkeit von ihrer Komplexität untersucht werden. Es soll geprüft werden, wie mehrfache Symmetrien identifiziert werden können und ggf. für die 3D-Rekonstruktion des regelmäßigen Teils der Fassadenstruktur genutzt werden können.

### 1.4 Verwandte Arbeiten

Bild-Analyse-Verfahren zur Bestimmung der Beziehungen von Bildelementen haben eine lange Tradition in der Computer Vision. Dabei wurden die verschiedensten Ziele verfolgt. Es wurden Parallelitäten, Symmetrien oder andere speziellere Beziehungen für gekrümmte Flächen, als Grundlage für die Gruppierung von Bildelementen, gesucht.

Schaffalitzky and Zisserman haben sich beispielsweise mit der geometrischen Gruppierung wiederholter Elemente in perspektivischen Bildern befasst ([Schaffalitzky and Zisserman, 1998], [Schaffalitzky and Zisserman, 1999]). Die diesen Arbeiten zugrunde liegende Idee ist, dass lokal wiederholte Bildelemente durch eine affine Transformation miteinander verbunden sind. Es werden also Bildelemente gruppiert, die durch gleiche affine Transformationen ineinander zu überführen sind. In [Schaffalitzky and Zisserman, 2000] wird dieser Ansatz dazu genutzt, aus den im Bild identifizierten regelmäßig angeordneten Elementen, Fluchtpunkte und -linien des aufgespannten Gitters abzuleiten.

Diese Verfahren basieren auf der Detektion markanter Punkte bzw. Kanten. Für glatt verputzte Fassaden sind aber kaum markante Punkte als Ergebnis eines Punkt-Operators zu erwarten, weshalb die Ergebnisse von [Schaffalitzky and Zisserman, 1999] auch hauptsächlich an Objekten wie Fußbodenfliesen oder Klinkerwänden gezeigt wurden. Auch die Suche nach markanten Linien in Bildern von Gebäuden kann sich als schwierig erweisen. Bei Überlagerung verschiedener Strukturen, bspw. der Abbildung eines Zaunes mit senkrechten Kanten vor der einer Treppe mit waagerechten Kanten, entstehen als Ergebnis der Kanten-Detektion nur kurze Kantenstücke, die für die Bestimmung der Fluchtpunkte ungeeignet sind.

Während sich diese Arbeiten mit planaren Geometrien befassten, werden in [Hays et al., 2006] regelmäßige Strukturen in beliebigen Texturen auf beliebigen Flächen gesucht. In einem iterativen Verfahren werden hier so genannte *Texel*<sup>2</sup> propagiert und Nachbarschaften dieser Texel zugeordnet. Die Iterationen maximieren die paarweisen Ähnlichkeiten der Texel und der aufgebauten geometrischen Zwänge des Gitters. Im Gegensatz zu [Schaffalitzky and Zisserman, 1998] oder [Schaffalitzky and Zisserman, 2000] ist die Form des durch die regelmäßige Struktur aufgespannten Gitters hier beliebig, es sind auch Splines als Gitterlinien denkbar.

---

<sup>2</sup>In Analogie zum Begriff Pixel definieren *Texel* die Grundeinheit einer Textur. Das Gitter der Texel, die sich berühren aber nicht überlappen, erzeugt die flächendeckende Textur.

Spezielle Arbeiten zum Thema Symmetrie-Detektion reichen bis in die 70er Jahre zurück. In globalen Ansätzen beispielsweise wurde das gesamte Bild als Signal betrachtet und Symmetrien über eine Frequenzanalyse abgeleitet. Oder aber die dominanten Symmetrieachsen wurden aus dem Histogramm der Gradientenorientierungen abgeleitet. In beiden Fällen können keine multiplen Symmetrien aufgedeckt werden.

In lokalen Feature-basierten Methoden werden typischerweise lokale Features (Punkte, Ecken, Kanten) genutzt und unter ihnen symmetrische Gruppierungen gesucht. Typischerweise werden dabei die Symmetrie-Hypothesen der zugeordneten Feature-Paare durch eine Hough- Transformation akkumuliert und die dominante Symmetrie abgeleitet. [Cornelius and Loy, 2006]

Alle diese Verfahren beschäftigen sich ganz allgemein mit Regularitäten oder Symmetrien in Bildern, in der Regel ohne besonderen Kontext. Der Hintergrund dieser Arbeit liegt speziell in der Analyse von Fassadenaufnahmen. Auch in [Brenner and Ripperda, 2006] werden speziell Fassaden untersucht. Der Ansatz hier unterscheidet sich aber stark von den anderen erwähnten Methoden der Computer Vision. Es wird eine formale Grammatik entwickelt, die die Struktur der Fassade in Form eines Baumes beschreibt. Rekursiv wird die Fassade vom Wurzelement aus in seine Bestandteile unterteilt - Sockel und Fassadenteil, symmetrische Fassadenteile, wiederholte Fassadenteile (z.B. Fensterreihen) bis hinunter zu den Blattknoten, den Türen und Fenstern. Der Aufbau des Baumes erfolgt mit Hilfe von „reversible jump Markov Chain Monte Carlo“ (rjMCMC) und unter Einführung von Randbedingungen bzw. Zwängen die von den Elementen des Baumes zu erfüllen sind<sup>3</sup>.

## 1.5 Gliederung der Arbeit

In Kap. 2 werden die grundlegenden Begrifflichkeiten und Voraussetzungen erläutert. Die Methoden zur Detektion von Symmetrien, wie sie von [Loy and Eklundh, 2006] eingeführt wurden, werden detailliert beschrieben. In den sich anschließenden Absätzen werden die darauf aufbauenden Methoden zur Detektion sich wiederholender Strukturen sowie der Ableitung von Mustern der Fassade aus den wiederholten Strukturen konzeptuell beschrieben. Eine Idee zur Bestimmung der Relativen Orientierung aus nur einem Bild eines symmetrischen Objektes wird vorgestellt, aber im weiteren Verlauf der Arbeit nicht weiter bearbeitet.

In Kap. 3 wird darauf eingegangen, wie die beschriebenen Konzepte umgesetzt wurden und welche Ergebnisse mit den erarbeiteten Algorithmen zu erzielen waren.

Im Kap. 4 schließt die Arbeit mit einer Zusammenfassung und einem Ausblick ab.

Die Implementation der vorgestellten Verfahren liegt der Arbeit als Matlab-Code auf CD bei.

---

<sup>3</sup>Beispielsweise ist das Blattknoten-Element auf den unteren Teil der Fassade beschränkt. Fenster liegen typischerweise in einer Reihe. usw.



# 2. Detektion von Symmetrien und wiederholten Strukturen

## Grundlagen und Konzepte

### 2.1 Modellbildung

#### 2.1.1 Objektmodell

Das von Loy und Eklundh vorgestellte Verfahren zur Detektion von Symmetrien in Bildern soll in dieser Arbeit ganz gezielt auf Fassadenaufnahmen angewendet werden. Diese zeichnen sich i.Allg. durch die Anordnung gleicher Fassadenelemente (Türen, Fenster, Verzierungen) durch sehr regelmäßige Strukturen und Symmetrien aus, die in den verschiedensten Skalen zu beobachten sind. Typischerweise ist eine Fassade oder Fensterfront durch eine oder zwei große Hauptsymmetrien gekennzeichnet, in Abb. 2.1 illustriert durch die roten Linien. Je mehr Details von einer solchen Fassade betrachtet werden, desto mehr Symmetrien und auch wiederholte Strukturen kann man beobachten. Für die Symmetrien sind das in der nächsten Skala die Fensterreihen (blau in Abb. 2.1), sowohl horizontal als auch vertikal und in der nächsten Stufe die Fenster selbst (grün). Das lässt sich fortsetzen bis zu den Fensterkreuzen oder der Struktur der kleineren Fassadenelemente. Eine typische wiederholte Struktur, neben den Fenstern, kann man sich leicht in Dachziegeln oder Klinkern vorstellen.

Die betrachteten Objekte müssen nicht planar sein. Gerade Wohngebäude sind durch Gestaltungselemente wie Balkone oder Erker gekennzeichnet, die die Struktur der Fassade maßgeblich mitbestimmen.

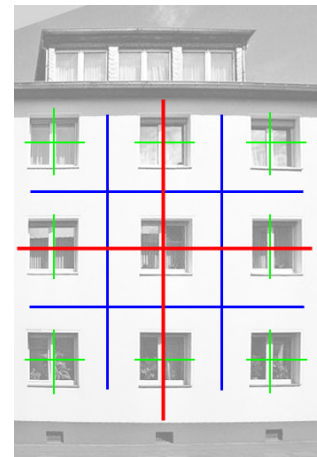


Abbildung 2.1: Symmetrieachsen einer typischen Fensterfront in den verschiedenen Skalen

#### 2.1.2 Sensor- und Bildmodell

Die verwendeten terrestrischen Einzelaufnahmen diverser Gebäude wurden mit verschiedenen Digitalkameras unter unterschiedlichen Bedingungen in Bonn, München, Berlin und Strasbourg aufgenommen und entzerrt. Es wird von geradentreuer Abbildung ausgegangen, so dass die Charakteristika der Fassaden in den Abbildungen auch als solche erhalten bleiben.

Außer einer Reduzierung auf Graustufenbilder<sup>1</sup> und in einzelnen Fällen einer automatischen

<sup>1</sup>Prinzipiell können die vorgestellten Konzepte natürlich auch auf RGB-Bilder angewendet werden. Dafür müssten die erforderlichen Berechnungen auf allen drei Kanälen ausgeführt werden und anschließend geeignet fusioniert werden.



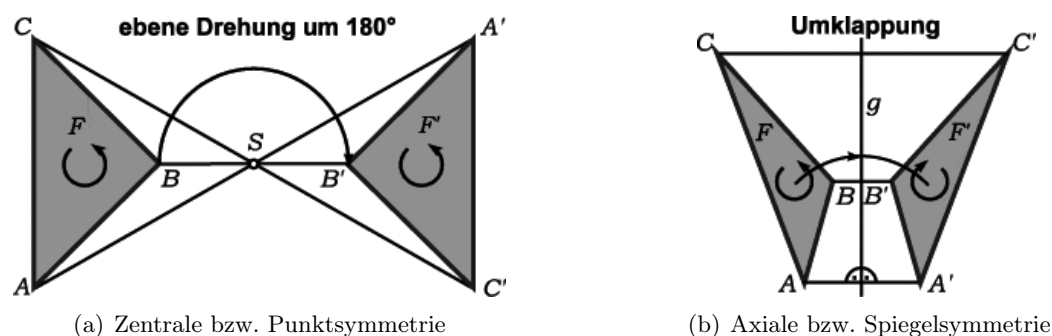


Abbildung 2.2: Symmetriearten[Bronstein et al., 2001]

Kontrastkorrektur wurden keine bildverbessernden Schritte unternommen.

### 2.1.2.1 Bemerkungen

Die Entzerrung der Testbilder erfolgte mit einem Matlab-Tool (`rectify_tool.m` bzw. `rectify_fassade.m`) des Fachbereiches Photogrammetrie, basierend auf [Förstner, 2004]. In einem semiautomatischen Verfahren werden vom User zwei im Objektraum zueinander rechtwinklige Paare paralleler Linien ausgewählt. Unter der Annahme, dass der Hauptpunkt sich im Zentrum des Bildes befindet, werden aus den angegebenen Linien die Fluchtpunkte bestimmt und aus diesen eine Rotationsmatrix aufgebaut. Diese kann als Homographie interpretiert werden, die die unendliche Ebene auf sich selbst abbildet, so dass die projektiv abgebildeten Linien ein Rechteck bilden. Angewendet auf die Bildpunkte, wird ein Bild erzeugt, dass scheinbar von einer (virtuellen) Kamera aufgenommen wurde, mit gleicher Position wie die reale Kamera, aber mit Aufnahmeorientierung senkrecht zur Fassade.

Die Details zu diesen Berechnungen sind in [Förstner, 2004] nachzulesen.

## 2.2 Begriffe

In diesem Abschnitt sollen einige Begriffe bzw. Konzepte näher erläutert werden, auf die im Laufe der Arbeit wiederholt Bezug genommen wird.

### 2.2.1 Symmetrie

Die Abbildung des gesamten Raumes in sich, bei der die Streckenlängen invariant sind und das abgebildete Objekt mit sich selbst zur Deckung kommt wird *Symmetrioperation* genannt. Die Menge aller Punkte im Raum, die durch diese Operation unverändert bleiben, heißen *Fixpunkte*. Man unterscheidet Symmetrioperationen ohne und solche mit mindestens einem Fixpunkt. Drehungen und Spiegelungen gehören zu den Operationen mit mindestens einem Fixpunkt [Bronstein et al., 2001].

Da sich die Untersuchungen dieser Arbeit auf Bilder beziehen, genügt an dieser Stelle die Betrachtung ebener Figuren. Es ist zwischen *Zentraler* bzw. *Punktsymmetrie* an einem Fixpunkt und *Axialer* bzw. *Spiegelsymmetrie* an einer Fixgeraden zu unterscheiden.

Nach [Bronstein et al., 2001] sind zwei Figuren *punktsymmetrisch*, wenn sie durch eine ebene Drehung um  $180^\circ$  um ein Symmetriezentrum S, den Fixpunkt, zur Deckung gebracht werden können (Abb. 2.2(a)). Größe, Form und Umlaufsinn der Eckpunkte sind Invariante dieser Transformation. Solche Figuren werden als gleichsinnig kongruent bezeichnet.

Von *Spiegelsymmetrie* spricht man, wenn zwei Figuren durch eine räumliche Drehung um  $180^\circ$  (Umklappung) um eine Fixgerade  $g$  in einander überführt werden können (Abb. 2.2(b)). Die senkrechten Abstände zur Spiegelachse der einander zugeordneten Punkte sind dabei gleich groß. Da

der Umlaufsinn der Figuren hier umgekehrt wird, wird von nichtgleichsinniger Kongruenz gesprochen.

Im folgenden soll mit dem Begriff Symmetrie immer Spiegelsymmetrie an einer Fixgeraden gemeint sein.

### 2.2.2 Wiederholte Strukturen

Wiederholte Strukturen im Kontext dieser Arbeit sind durch in regelmäßigen Abständen wiederkehrende Bildmerkmale gekennzeichnet. Die Grundelemente der Struktur sind dabei durch gleiche Translationen  $v$  verbunden, so dass man vom ersten Element  $x_0$  ausgehend alle  $n$  Elemente durch  $x_i = x_0 + i \cdot v$ , ( $i = 1 \dots n$ ,  $i \in \mathbb{N}$ ,  $v \in \mathbb{R}$ ) abbilden kann.

Translationen können als spezielle Symmetrieoperationen betrachtet werden. Sie gehören zu den Symmetrieoperationen ohne Fixpunkt und lassen sich ersetzen durch zwei Spiegelungen.

### 2.2.3 Features

Mit dem Begriff *Feature*<sup>2</sup> sind im Kontext der Bildverarbeitung charakteristische Bildmerkmale gemeint. Das können im einfachsten Fall charakteristische Punkte, Ecken oder Kanten sein, die sich durch ihre Grauwertgradienten auszeichnen oder aber komplexe Merkmale, die über eine einfache Punktkoordinate hinausgehen. Mit Hilfe solcher Features ist es möglich verschiedene Bilder (gleicher Objekte) zu matchen, um z.B. Bilder in einer Datenbank wiederzufinden oder aber korrespondierende Punkte für verschiedene photogrammetrische Anwendungen zu identifizieren (Herstellung der Kameraorientierung oder Berechnung von Objektpunktkoordinaten).

Moderne Methoden zur Detektion von Features bestimmen für jedes Bildmerkmal, zusätzlich zu seiner Position im Bildkoordinatensystem, Orientierung und Maßstab des detektierten Features und eine Beschreibung seiner lokalen Umgebung. Durch eine geeignete Normalisierung wird dabei Invarianz bzgl. dieser Parameter erreicht. Typischerweise wird jedes Feature durch einen Vektor beschrieben mit seiner Position als  $(x, y)$ -Bildkoordinaten, seiner Orientierung  $\phi$  und seinem Maßstab  $s$  sowie einem Vektor der die Beschreibung der lokalen Umgebung geeignet codiert, dem Deskriptor. Einige solcher Feature-Detektoren werden in [Mikolajczyk and Schmid, 2005] vorgestellt. Speziell die *Rotationsinvarianz* macht diese Methoden geeignet zur Detektion von symmetrischen Features. Abweichend vom ursprünglichen Zweck werden dabei Features innerhalb eines Bildes mit gespiegelten Kopien der Feature-Deskriptoren gematcht.

---

<sup>2</sup>engl., Merkmal, Kennzeichen, Charakteristik. Der Begriff hat sich auch im Deutschen als gebräuchlich erwiesen und soll auch hier so verwendet werden.

## 2.3 Detektion von Symmetrien

Symmetrien, die vom menschlichen Betrachter so einfach zu identifizieren sind, stellen die Computer Vision vor viele Probleme. Es wird eine Bildregion von unbekannter Größe gesucht, die, gespiegelt um eine beliebige Achse (bei Achssymmetrie) oder gedreht um einen beliebigen Winkel (bei Punktsymmetrie), ähnlich ist zu einem anderen beliebigen Bildausschnitt des gleichen Bildes.

Das im Folgenden vorgestellte Verfahren basiert auf einer Arbeit von Loy und Eklundh<sup>3</sup> [Loy and Eklundh, 2006].

Die Grundidee besteht darin, Paare von kleinen Bildausschnitten zu finden, die symmetrisch sind. Aus allen gefundenen Paaren soll eine für das Bild dominante Symmetrie, repräsentiert durch ihre Symmetrieachse, extrahiert werden.

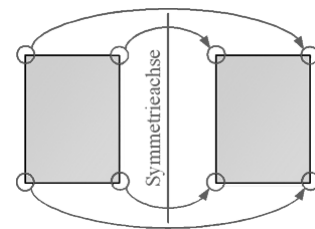


Abbildung 2.3: Beispiel für die Zuordnung symmetrischer Features und der dazugehörigen Symmetrieachse, exemplarisch für eine der möglichen Symmetrieachsen.

### 2.3.1 Der SIFT-Operator

Lowe hat mit der „Scale Invariant Feature Transform“ (SIFT) eine Methode entwickelt, mit der aus Bilddaten skalen-invariante Punktkoordinaten relativ zu lokalen *Features* erzeugt werden. Die Features sind stabil über verschiedenen Bildmaßstäbe (Skalen) und invariant bezüglich Rotation und nahezu invariant bezüglich Beleuchtung des Objektes und Kameraposition [Lowe, 2004]. Im folgenden sollen solche Punkte auch als *Keypoints* oder wegen des zugrunde liegenden Verfahrens auch als *Lowe-Punkte* bezeichnet werden.

Details zur Bestimmung der Lowe-Punkte sind in [Lowe, 2004] nachzulesen. An dieser Stelle seien zum besseren Verständnis nachfolgender Erläuterungen nur die Grundzüge des Verfahrens vorgestellt.

Die Invarianz bezüglich des Bildmaßstabs ist deshalb von besonderer Bedeutung, da sich Bildmerkmale mit sich änderndem Abstand der Kamera vom Objekt natürlich ändern. Bei der Feature-Detektion werden solche Bildmerkmale gesucht, die über alle Skalen zu identifizieren sind. Für eine solche Suche wird der so genannte Skalen-Raum, eine kontinuierliche Funktion der Skala, verwendet. Als Kern dieser Funktion dient die Gauß-Funktion in Abhängigkeit von  $\sigma$ . Eine Veränderung des Maßstabs wird also durch eine Glättung des Bildes simuliert. Diese Eigenschaft des Detektors ist für das Matching von Features innerhalb eines Bildes nicht von Bedeutung aber ein grundsätzlicher Bestandteil des Verfahrens und sei deshalb hier erwähnt. Über alle Skalen und Bildpositionen werden die Extrema des Skalen-Raumes bestimmt. 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.

<sup>3</sup>Computational Vision and Active Perception Laboratory (CVAP) des Royal Institut of Technology (KTH), Stockholm. Die Arbeit wurde im Mai 2006 in Graz auf der 9<sup>th</sup> European Conference on Computer Vision (ECCV) vorgestellt.

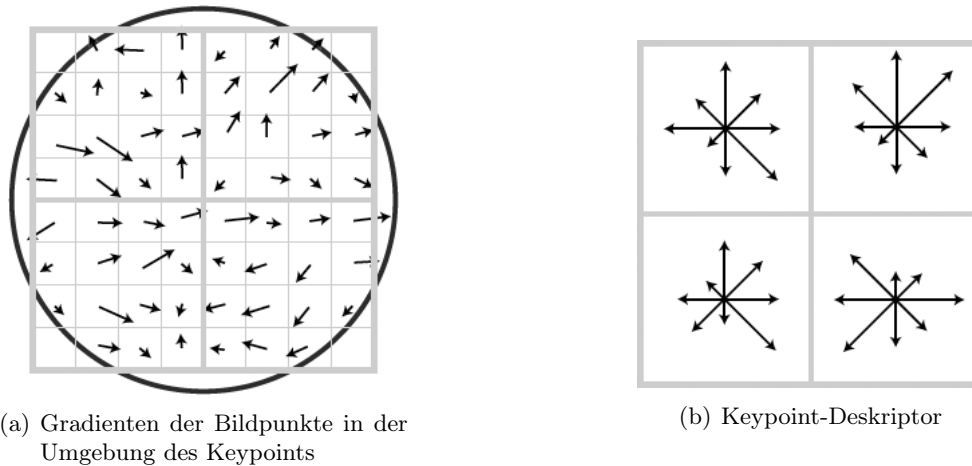


Abbildung 2.4: Entstehung des Keypoint-Deskriptors aus den Gradienten der Umgebung des Keypoints. Hier für einen 2x2 Deskriptor, entstanden aus den Gradienten einer 8x8 Pixel Umgebung des Keypoints. Der für diese Arbeit verwendete Algorithmus erzeugt einen 4x4 Deskriptor aus einer 16x16 Pixel Umgebung. [Lowe, 2004]

### 2.3.1.1 Die Deskriptoren

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

Abb. 2.4 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. 2.4(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. 2.4(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. 2.4(b), als Vektorelemente. Für den in Abb. 2.4 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 Invarianz bezüglich Beleuchtung bzw. Kontrast zu erreichen.

Der Algorithmus wird von Lowe als C-Code und auch als Matlab-Implementierung<sup>4</sup> zur Verfügung gestellt und wurde als solcher in dieser Arbeit direkt verwendet.

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

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

$$\mathbf{l} = [\text{row}, \text{col}, s, \phi] \quad (2.1)$$

<sup>4</sup>[www.cs.ubc.ca/~lowe/keypoints/](http://www.cs.ubc.ca/~lowe/keypoints/); Stand: 04.11.2006

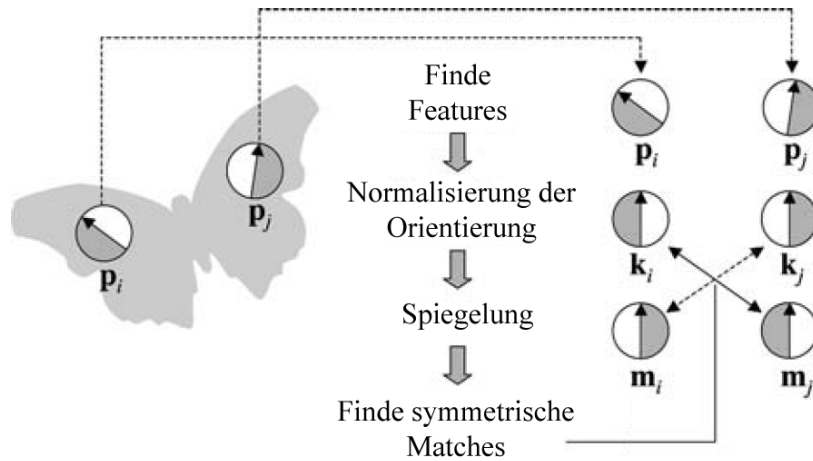


Abbildung 2.5: Schematische Darstellung der Detektion von Symmetrien [Loy and Eklundh, 2006] (bearb.)

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

### 2.3.2 Bestimmung von Symmetrien in Bildern

Abb. 2.5 zeigt schematisch das Vorgehen beim Finden einzelner symmetrischer Feature-Paare. Zunächst werden mit einem geeigneten Detektor charakteristische Bildmerkmale gesucht. Durch Spiegelung der Deskriptoren entsteht ein zweiter Satz von Features. Dabei sei  $\mathbf{m}_i$  die gespiegelte Version des „originalen“ Feature-Deskriptors  $\mathbf{k}_i$ . Diese beiden Sätze werden gematcht, um Kandidatenpaare  $(\mathbf{p}_i, \mathbf{p}_j)$  symmetrischer Features zu finden. Für jedes Kandidatenpaar  $(\mathbf{p}_i, \mathbf{p}_j)$  wird die Symmetrieachse durch ihre  $(\rho, \theta)$ -Polarkoordinaten ausgedrückt und in einem Hough-Array, gewichtet nach der Qualität der Symmetrie des Matches, akkumuliert. Dieser Hough-Akkumulator wird geglättet und das Maximum definiert die für das betrachtete Bild dominante Symmetrieachse.

In den Experimenten von Loy und Eklundh geht es an dieser Stelle zunächst nur um die Detektion *einer* dominanten Symmetrie. In dieser Arbeit soll untersucht werden, inwiefern dieses Verfahren geeignet ist, mehrere Symmetrien in einem Bild zu detektieren. Daher wird der Hough-Raum hier nach weiteren signifikanten Peaks durchsucht.

In den folgenden Abschnitten sollen nun die Details zum vorgestellten Verfahren näher beleuchtet werden.

#### 2.3.2.1 Der Feature Detektor

Das Verfahren ist prinzipiell unabhängig von der Wahl des Feature-Detektors. Er soll robust sein und Features mit Angabe der Orientierung und einer rotationsinvarianten Beschreibung der lokalen Umgebung liefern. Loy und Eklundh haben sich für den *SIFT-Operator* von Lowe entschieden und auch für die Untersuchungen im Rahmen dieser Arbeit wurde dieser Algorithmus verwendet. Dieses Verfahren hat sich in der Praxis bereits vielfach bewährt, es liefert charakteristische Bildmerkmale mit einer guten Wiederholbarkeit ([Loy and Eklundh, 2006]). Für jedes Feature wird hier der benötigte Deskriptor  $\mathbf{k}_i$  als Vektor mit 128 Elementen erzeugt, der die Erscheinung der lokalen Umgebung des Feature-Punktes rotationsinvariant kodiert.

#### 2.3.2.2 Spiegelung der Deskriptoren

Nun stellt sich die Frage, wie man zu den gespiegelten Features kommt. Zunächst lässt sich feststellen, dass durch die Wahl des Lowe-Operators 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.

Die einfache Variante ist sicherlich die, 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 benötigt man aber einen zweiten Durchlauf des Feature-Detektors 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.

Mit genauer Kenntnis des verwendeten Feature-Detektors sollte es möglich sein, die durch den Deskriptor beschriebenen Bildpatches zu spiegeln, d.h. die Deskriptoren selbst zu verändern.

Die Spiegelung der Features erfolgt also durch eine geeignete Umsortierung der Deskriptoren. Für das genaue Vorgehen dabei sei auf Abs. 2.3.2.2 (S.16) verwiesen.

### 2.3.2.3 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].

$$dist = \sqrt{\langle \mathbf{d} - \mathbf{d}_i, \mathbf{d} - \mathbf{d}_i \rangle} \rightarrow \min_i, \quad i = 1 \dots n \quad (2.2)$$

Die resultierende Distanz-Matrix, in der die Abstände zwischen allen Features registriert sind, ist symmetrisch. Daher ist nur die obere Dreiecksmatrix von Interesse.

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>5</sup>.

In der hier vorliegenden Anwendung muss aber beachtet werden, dass nicht nur *ein* Match (der *beste*) pro Keypoint gesucht wird, sondern möglichst viele Matches. Deshalb wird hier nicht das Verhältnis der Distanzen zwischen dem besten und dem zweitbesten Match betrachtet, sondern alle (in der Reihenfolge der Distanzen) benachbarten Matches und dann alle ähnlich „guten“ Matches akzeptiert, bis zu einem benachbarten Paar, deren Verhältnis die geforderte Bedingung überschreitet.

### 2.3.2.4 Repräsentation der Symmetrieachsen

Abb. 2.6 illustriert die Wahl des Koordinatensystems und die Definition der verwendeten Winkel. Zu beachten ist, dass das (*row*, *col*)-Bildkoordinatensystem seinen Ursprung oben links hat, das klassische mathematisch kartesische (*x*, *y*)-Bildkoordinatensystem dagegen unten links. Bei beiden Angaben handelt es sich um Rechtssysteme. Im ersten Fall folgt die Angabe der Spalten (*column*) der der Zeilen (*row*). Die (*x*, *y*)-Koordinierung bezieht sich auf die Spalten, gefolgt von der Zeilenangabe.

<sup>5</sup>In [Lowe, 2004] wurde bspw. für *distRatio* ein Wert von 0.8 gewählt.

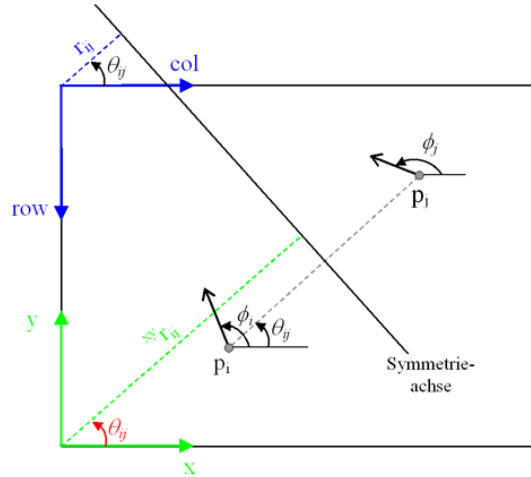


Abbildung 2.6: Definition der Koordinatensysteme und der Winkel

Zu beachten ist, dass sich die Komponenten des Location-Vektors aus dem SIFT-Algorithmus auf eine Mischform der beiden Systeme beziehen. Die ersten beiden Komponenten definieren die Position des Features in  $(row, col)$ -Koordinaten, während die dritte Komponente die Orientierung  $\phi$  bezogen auf die  $x$ - bzw.  $col$ -Achse im Uhrzeigersinn beschreibt. Abweichend vom Vorgehen in [Loy and Eklundh, 2006] wird für diese Arbeit das  $(row, col)$ -Koordinatensystem verwendet.

Für die Repräsentation der Symmetrieachsen in  $(\rho, \theta)$ -Polarkoordinaten ergibt sich damit:

$$\begin{aligned}\theta_{ij} &= \arctan \frac{col_j - col_i}{row_j - row_i} \\ \rho_{ij} &= col_m \sin \theta_{ij} + row_m \cos \theta_{ij}\end{aligned}\tag{2.3}$$

Mit den Koordinaten  $(row_m, col_m)$  des Mittelpunktes der Verbindung zwischen  $\mathbf{p}_i$  und  $\mathbf{p}_j$

$$\begin{aligned}row_m &= (row_i + row_j) / 2 \\ col_m &= (col_i + col_j) / 2\end{aligned}$$

Eine Transformation der Lowe-Punkte von  $(row, col)$  nach  $(x, y)$  wäre sicherlich eleganter gewesen und hätte einige Verwirrungen bei der Entwicklung weiterer Formeln vermieden.

### 2.3.2.5 Bewertung der Symmetrie einzelner Feature-Paare

Die Qualität der Symmetrie der einzelnen Feature-Paare wird anhand ihrer Orientierung, Skala und Distanz bewertet.

Mit dem Winkel-Symmetrie-Gewicht  $\Phi_{ij} \in [0, 2]$

$$\Phi_{ij} = 1 - \cos({}^{xy}\phi_i + {}^{xy}\phi_j - 2 {}^{xy}\theta_{ij})\tag{2.4}$$

werden diejenigen Feature-Paare hoch bewertet, deren Orientierung möglichst symmetrisch bzgl. der ermittelten Symmetrieachse dieses Paares ist<sup>6</sup>.

<sup>6</sup>Achtung! Hier beziehen sich die Richtungsangaben auf das  $(x, y)$ -Koordinatensystem, mit den Orientierungen  ${}^{xy}\phi_i$  und  ${}^{xy}\phi_j$  als Ergebnis des Feature-Detektors und  ${}^{xy}\theta_{ij} = \theta_{ij} - \pi/2$ .

Mit dem Skalen-Gewicht  $S_{ij} \in [0, 1]$

$$S_{ij} = \left[ \exp \left( \frac{-|s_i - s_j|}{\sigma_s (s_i + s_j)} \right) \right]^2 \quad (2.5)$$

kann über  $\sigma_s$  gesteuert werden, welche Skalen-Differenzen zwischen den Feature-Paaren zugelassen werden sollen.  $S_{ij}$  wird umso kleiner, je größer die Differenz zwischen den beiden Skalen ist. Je größer aber  $\sigma_s$  ist, desto größere Differenzen werden toleriert und desto langsamer fällt  $S_{ij}$  ab.

Und schließlich kann mit dem Distanz-Gewicht  $D_{ij}$

$$D_{ij} = \exp \left( \frac{-a^2}{2\sigma_a^2} \right) \quad (2.6)$$

mit dem Abstand  $a$  der Features zur Symmetrieachse, über  $\sigma_a$  gesteuert werden, welche Distanzen zur potentiellen Symmetrieachse zugelassen werden sollen oder anders ausgedrückt, welche Ausdehnung der symmetrischen Objekte zu erwarten ist.

Die Bewertung der Qualität der Symmetrie zweier Features  $\mathbf{p}_i$  und  $\mathbf{p}_j$  ergibt sich schließlich aus allen Gewichten als Symmetriegrad

$$M_{ij} = \Phi_{ij} S_{ij} D_{ij}. \quad (2.7)$$

Durch die Wahl der Multiplikation als Verknüpfungsoperation wird gewährleistet, dass alle Bestandteile des Symmetriegrades in hohem Maße erfüllt sein müssen. Falls eine der Komponenten nahe an Null reicht, so wird auch der gesamte Symmetriegrad des entsprechenden Feature-Paares sehr klein. Bei einer Addition der Komponenten wäre dem nicht so, die zwei anderen Komponenten könnten den Verlust ausgleichen, was für das hier behandelte Problem nicht wünschenswert wäre.

### 2.3.2.6 Ermittlung dominanter Symmetrien im Hough-Raum

Das Prinzip der Hough-Transformation besteht in der Dualität zwischen der Geraden- Repräsentation in kartesischen  $(x, y)$ -bzw.  $(row, col)$ -Koordinaten und in  $(\rho, \theta)$ -Polarkoordinaten. Jeder Punkt im  $(row, col)$ -Raum entspricht einer sinus-förmigen Kurve im  $(\rho, \theta)$ -Raum. Dual dazu entspricht jeder Punkt im  $(\rho, \theta)$ -Raum einer Geraden im  $(row, col)$ -Raum, illustriert in Abb. 2.7.

$$\rho = row \cdot \cos(\theta) + col \cdot \sin(\theta) \quad (2.8)$$

bzw.

$$col = \frac{\rho}{\sin(\theta)} - row \cdot \cot(\theta), \quad row = \frac{\rho}{\cos(\theta)} - col \cdot \tan(\theta) \quad (2.9)$$

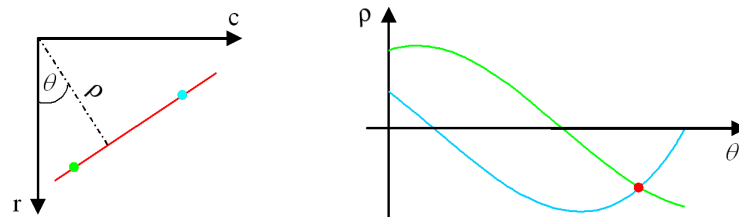


Abbildung 2.7: Prinzip der Hough-Transformation.



Eine typische Anwendung dieser Transformation besteht z.B. darin, aus einer Menge charakteristischer Bildpunkte im  $(row, col)$ -Raum, aus den Schnittpunkten der dualen Kurven im  $(\rho, \theta)$ -Raum, Geraden im Bild zu detektieren.

Für die Zwecke der Symmetrie-Detektion wird jede potentielle Symmetrieachse mit ihren Polarkoordinaten, gewichtet mit dem Symmetriegrad der beteiligten Features aus Gl. (2.7), in einem  $(\rho, \theta)$ -Hough-Array akkumuliert, um daraus dominante Geraden (Symmetrieachsen) im Bild abzuleiten.

Um zu vermeiden, dass einzelne, hoch gewichtete Symmetrieachsen große Gruppen, im Array eng beieinander liegender Symmetrieachsen dominieren, wird der Akkumulator mit einem Gauß-Filter geglättet. Das hat zur Folge, dass einzelne schmale Peaks abgeschwächt werden, signifikante (breite) Peaks dagegen erhalten bleiben.

Der größte Peak im resultierenden Array zeigt die dominante Symmetrie im untersuchten Bild an. Aus den Feature-Paaren, die zu den  $(\rho, \theta)$ -Koordinaten in einer kleinen Umgebung  $[d\rho, d\theta]$  um das Maximum gehören, wird schließlich die für diese Paare repräsentative mittlere Symmetrieachse berechnet. Dafür werden für jedes beteiligte Feature-Paar über Varianz-Fortpflanzung die Kovarianz-Matrizen der  $(\rho, \theta)$ -Repräsentation der jeweiligen Symmetrieachse berechnet und schließlich das mit dem Symmetriegrad  $M_i$  aus Gl. (2.7) gewichtete Mittel aller beteiligten Symmetrieachsen berechnet. Die Herleitung der entsprechenden Berechnungsschritte kann in Anhang A.1 nachgelesen werden.

Nachdem die für diesen Schritt berücksichtigten  $(\rho, \theta)$ -Paare aus dem Akkumulator entfernt wurden, wird das nächste Maximum gesucht und entsprechend behandelt. Auf diese Art und Weise werden alle Peaks im Akkumulator abgearbeitet, bis keine Symmetrieachse mehr gefunden wird, die nicht von einer Mindestanzahl an Feature-Paaren unterstützt wird.

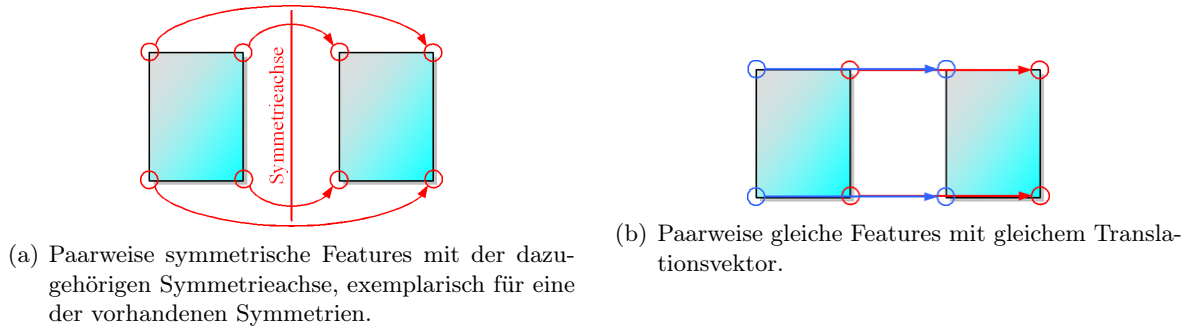


Abbildung 2.8: Vergleich des Prinzips der Detektion von Symmetrien und wiederholten Strukturen.

## 2.4 Detektion von wiederholten Strukturen

Für die Detektion wiederholter Strukturen im Bild wurde das in Abs. 2.3 vorgestellte Verfahren nur geringfügig verändert. Die detektierten Features werden hier nicht gespiegelt sondern direkt mit sich selbst gematcht. Das Ziel besteht darin, Gruppen von Feature-Paaren zu identifizieren, die durch die gleiche Translation miteinander verbunden sind.

Der prinzipielle Unterschied zwischen beiden Konzepten ist in Abb. 2.8 illustriert. Für die Symmetrie-Detektion werden die Features gespiegelt und dann der originale und der gespiegelte Featuresatz gematcht. Es werden Gruppen von Feature-Paaren gesucht, die eine gemeinsame Symmetrieachse haben. Für die Detektion wiederholter Strukturen wird der originale Featuresatz mit sich selbst gematcht und aus allen Paaren Gruppen gesucht, die durch den selben Translationsvektor gekennzeichnet sind.

Zur Repräsentation der Translation wurde auch hier eine Strecken-Richtungs-Form gewählt, da so die Implementation der Symmetrie-Detektion kaum geändert werden musste. Die Translation zwischen zwei Features wird ausgedrückt durch die Richtung  $\theta$  und die Länge  $r$  der Verschiebung.

Das weitere Vorgehen unterscheidet sich nur in der Art der Gewichtung der gefundenen Paare. Die Winkel-Symmetrie-Gewichtung aus Gl. (2.4) (S.18) wird zu einer Winkel-Gewichtung  $\Phi_{ij}^* \in [0, 2]$  mit

$$\Phi_{ij}^* = 1 + \cos(\phi_i - \phi_j), \quad (2.10)$$

mit 2 für exakt gleiche und 0 für genau entgegengesetzte Orientierungen  $\phi_i$  und  $\phi_j$ .

Die Anteile  $S_{ij}$  und  $D_{ij}$  an der Gewichtung  $M_{ij}^*$  bleiben wie in Gl. (2.5) und (2.6) (S.19). Die Qualität der Übereinstimmung zweier Features wird dann analog zu Gl. (2.7) bestimmt durch den Repetitionsgrad  $M_{ij}^*$

$$M_{ij}^* = \Phi_{ij}^* S_{ij} D_{ij}. \quad (2.11)$$

Wie auch bei der Symmetrie-Detektion werden alle Translationen zwischen den Feature-Paaren  $(\mathbf{p}_i, \mathbf{p}_j)$ , gewichtet mit ihrem Repetitionsgrad  $M_{ij}^*$ , in einem Hough-Array akkumuliert und anhand der Peaks Gruppen von Feature-Paaren identifiziert, die durch die gleiche Translation verbunden sind.

Die endgültige Repräsentation der gefundenen Translationen erfolgt dann, zur besseren Interpretation der Ergebnisse, in Form eines  $[drow, dcol]$ -Vektors für jede gefundene wiederholte Gruppe. Dafür wird aus allen  $n$  Paaren, die dem Maximum im Hough-Akkumulator in einer kleinen Umgebung  $(dr, d\theta)$  zugeordnet werden können, das mit  $M^*$  gewichtete Mittel für  $drow$  und  $dcol$  gebildet, siehe Anhang A.2. Da die beiden Größen unabhängig voneinander sind, ist hier keine Varianz-Fortpflanzung, wie in Abs. 2.3.2.6, für jede einzelne Translation nötig.

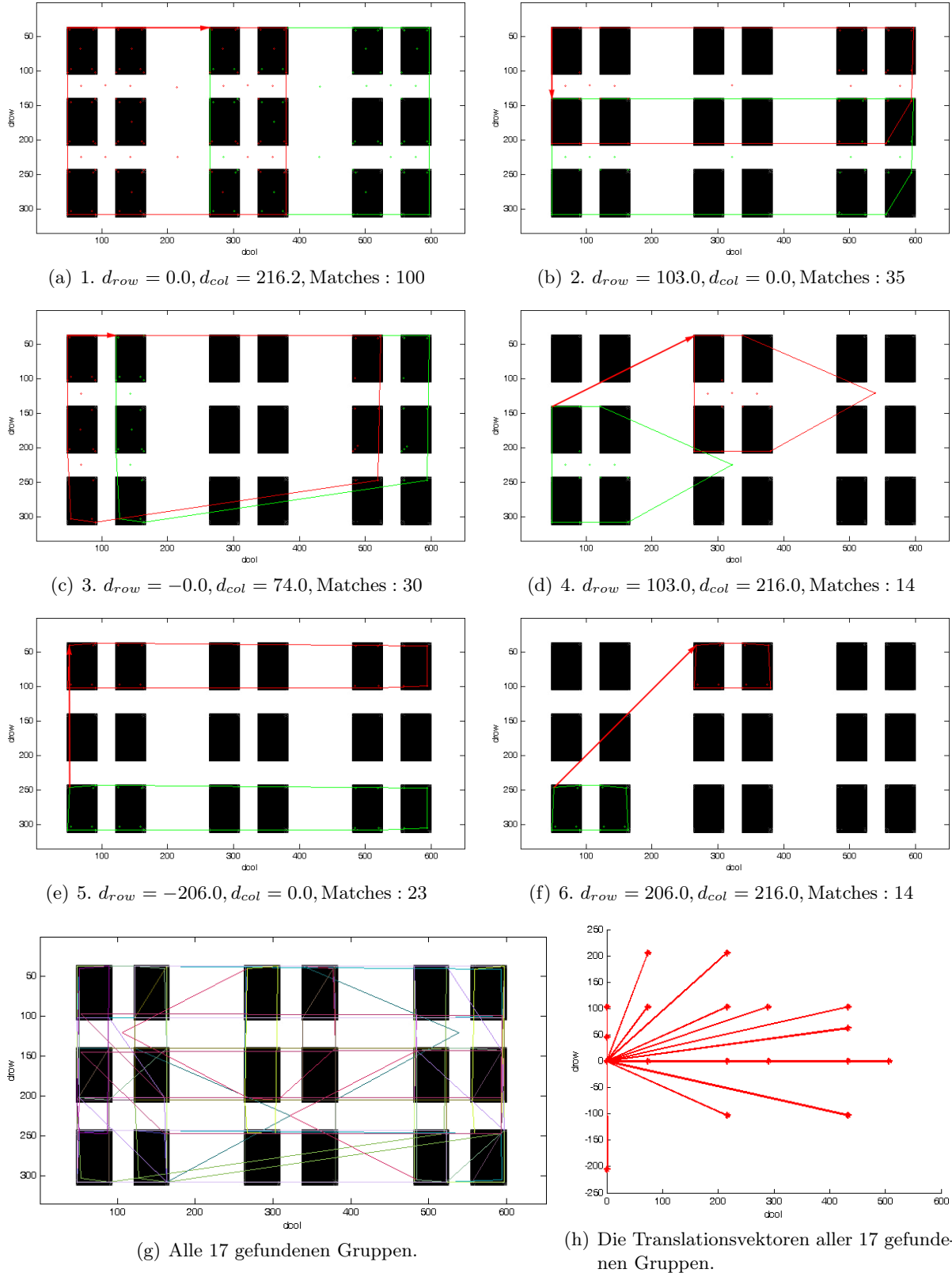


Abbildung 2.9: Ergebnisse der Detektion wiederholter Gruppen anhand des synthetischen Beispiels einer typischen Fensterfront. Mit  $distRatio = 0.8$  wurden 368 Keypoints detektiert und darunter 979 Matches gefunden. Die Abb. (a) bis (f) zeigen der Reihenfolge nach die ersten 6 gefundenen Gruppierungen. Abb. (g) zeigt alle 17 gefundenen Gruppen-Paare im Überblick. Für Abb. (h) schließlich wurden die Translationen der einzelnen Gruppen vom Ursprung aus abgetragen.

## 2.5 Ermittlung von Mustern in wiederholten Strukturen

In diesem Abschnitt soll es darum gehen, aus den Ergebnissen der Detektion wiederholter Gruppen die Struktur der Bildelemente (im Kontext der Fassadenanalyse der Fenster) abzuleiten. Um dieses Ziel genauer zu erläutern, soll den Ergebnissen der Detektion wiederholter Gruppen an dieser Stelle etwas vorweggenommen werden.

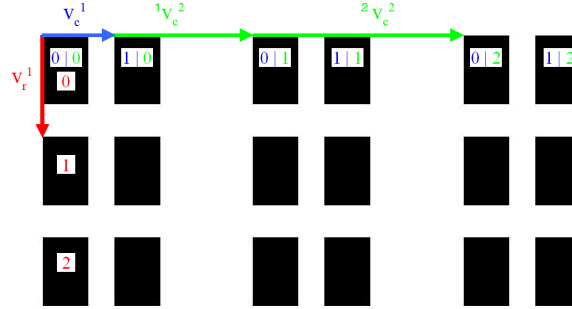


Abbildung 2.10: Basisvektoren zur Beschreibung der Struktur der Fassade

Abb. 2.9 zeigt die Ergebnisse der Detektion wiederholter Gruppen anhand des synthetischen Bildes einer typischen Fensterfront. Die Abb. 2.9(a) bis (f) zeigen der Reihenfolge nach die ersten sechs gefundenen Gruppierungen mit dem jeweiligen Translationsvektor (normiert, so dass  $d_{col} > 0$ ) und den beteiligten Lowe-Punkten. Abb. (g) zeigt alle 17 gefundenen Gruppen-Paare im Überblick. Für Abb. (h) schließlich wurden die Translationen der einzelnen Gruppen vom Ursprung aus abgetragen. In dieser Darstellung ist gut zu erkennen, dass sich die Beträge in  $d_{row}$ - und  $d_{col}$ -Richtung wiederholen. Man kann sich vorstellen ein rechtwinkliges Gitter über die Endpunkte der Translationsvektoren zu legen, so dass jeder Gitterpunkt mit Hilfe einer minimalen Anzahl von Basisvektoren beschrieben werden kann. Diese Basisvektoren beschreiben dann genau die untersuchte Fassade.

Für das hier gewählte Beispiel ist das gewünschte Ergebnis in Abb. 2.10 visualisiert. Diese Fensterfront ist in Zeilenrichtung gekennzeichnet durch drei Fensterreihen im Abstand  $v_r^1$ . In Spaltenrichtung sind zwei Gruppierungen zu beobachten. Eine Gruppe aus zwei Elementen (den Fenstern) im Abstand  $v_c^1$  und eine Gruppe aus drei Elementen (den Doppelfenstern) im Abstand  $v_c^2$ .

### 2.5.1 Modellannahmen

Da mit entzerrten Bildern gearbeitet wird, von denen angenommen werden kann, dass die Hauptrichtungen der Translationen achsparallel sind, wird die Suche nach einer geeigneten Basis zur Bildbeschreibung auf zwei eindimensionale Suchen beschränkt. D.h. die detektierten Gruppenverschiebungen  $d_{row_i}$  und  $d_{col_i}$  werden getrennt nach Zeilen und Spalten betrachtet<sup>7</sup>. Des Weiteren ist für eine eindimensionale Betrachtung die Richtung der Translation egal. Für die Suche nach Basistranslationen ist eine Verschiebung von  $a$  nach  $b$  gleichbedeutend einer Verschiebung von  $b$  nach  $a$ , weshalb die beobachteten Translationen auf ihre Beträge reduziert werden können.

Abb. 2.11 soll diesen Schritt anhand des Beispiels aus Abb. 2.9 verdeutlichen. Die detektierten Translationen sind noch einmal in Abb. 2.11(a) dargestellt. Die Abb. (b) und (c) zeigen nur noch die Beträge der Translationen in einer Richtung als Punkte auf der *row*- bzw. *col*-Achse. Zu beachten ist, dass die, in diesem künstlichen Beispiel so homogen erscheinenden Datenpunkte,

<sup>7</sup>Auch wenn besagte Hauptrichtungen nicht achsparallel wären, würden Basisvektoren in Zeilen- und Spaltenrichtung eine korrekte Beschreibung des Bildes liefern. Das Ergebnis wäre nur nicht so intuitiv zu interpretieren wie im achsparallelen Fall.

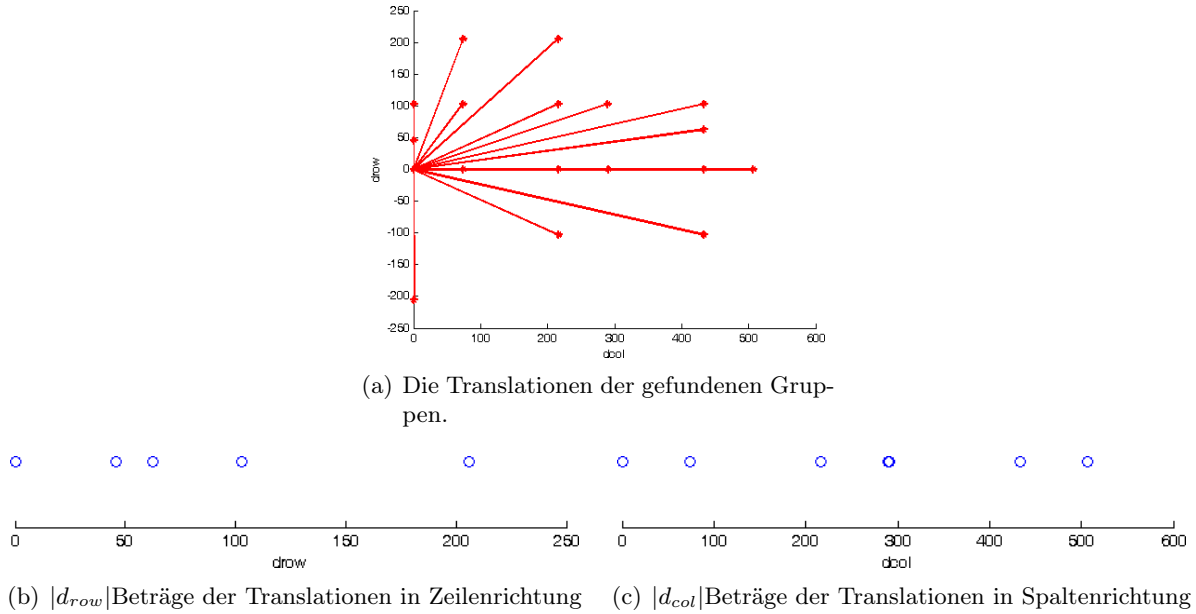


Abbildung 2.11: Trennung der aufgetretenen Translationen in Zeilen- und Spalten- Richtung

durch unterschiedlich viele Punkte in der 2D-Darstellung repräsentiert werden.

### 2.5.2 Problembeschreibung

Mathematisch lässt sich das resultierende Problem wie folgt beschreiben:

$$\begin{aligned}
 d_i &= \sum_{k=1}^K (\alpha_k \cdot v_k) + \varepsilon_i ; \quad \omega_i \\
 K &\rightarrow \min \\
 \omega_i \varepsilon_i^2 &\rightarrow \min \\
 d_i \in \mathbb{R} &- \text{beobachtete 1D-Translation} \\
 v_k \in \mathbb{R} &- \text{Translation, ein Basiselement} \\
 \alpha_k \in \mathbb{Z} &- \text{Anzahl der Wiederholungen des Basiselements} \\
 K \in \mathbb{N} &- \text{Anzahl der Basiselemente } v \\
 \varepsilon_i \in \mathbb{R} &- \text{Fehler} \\
 \omega_i \in \{0, 1\} &- \text{Flag, schaltet Gleichung } i \text{ „an“ oder „aus“}
 \end{aligned} \tag{2.12}$$

Jede beobachtete Translation  $d_i$  lässt sich mit Hilfe einer minimalen Anzahl von Basiselementen  $v_k$  ausdrücken. Die Koeffizienten  $\alpha_k$  sollen dabei ganze Zahlen sein und die Abweichungen  $\varepsilon$  möglichst klein. Der Parameter  $\omega_i$  soll Gleichungen von Beobachtungen ggf. ausschalten, die nicht durch die gesuchten Basiselemente auszudrücken sind. Solche Beobachtungen können aus Fehlern im Detektionsverfahren resultieren oder aber Strukturen im Bild repräsentieren, die eigentlich nicht gesucht sind, z.B. die Struktur der Fassadenverkleidung, statt der der Fensteraufteilung. Diese Beobachtungsgleichung ähnelt den so genannten *Linearen Diophantischen Gleichungen*<sup>8</sup>.

<sup>8</sup>Eine lineare Diophantische Gleichung in  $n$  Unbekannten ist eine Gleichung der Form  $a_1 + a_2x_2 + \dots + a_nx_n = b$  ( $a_i \in \mathbb{Z}, b \in \mathbb{Z}$ ) für die nur die ganzzahligen Lösungen gesucht werden. Unter der Bedingung, dass nicht alle  $a_i$  gleich 0 sind, ist eine solche Gleichung genau dann lösbar, wenn der größte gemeinsame Teiler (ggT) ( $a_1, a_2, \dots, a_n$ ) ein Teiler von  $b$  ist. Es existieren allgemeine Lösungsverfahren. Die Bezeichnung „Diophantisch“ erinnert an den griechischen Mathematiker Diophant. [Bronstein et al., 2001]

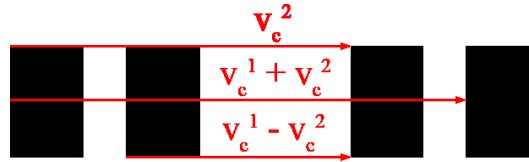


Abbildung 2.12: Überlagerung der Basiselemente

Hier handelt es sich aber um ein lineares Gleichungssystem mit reellwertigen Parametern und ganzzahligen Koeffizienten. Weder die Parameter  $v$  noch die Koeffizienten  $\alpha$  sind bekannt. Hinzu kommt, dass mit kleinen Fehlern in den Beobachtungen zu rechnen ist und mit solchen Beobachtungen, die nicht in das Gleichungssystem gehören.

Die Schwierigkeit besteht hier sicherlich in der Kürze der Datenreihen. Es werden Periodizitäten von nur einigen wenigen Perioden aus wenigen und dazu noch stark gestörten Datenpunkten gesucht, weshalb auch eine Harmonische Analyse über eine Fourier Transformation nicht in Frage kommt.

In [Sethares and Staley, 1999] wird mit der „Periodicity Transform“ (PT) ein Verfahren vorgestellt zur Detektion von Periodizitäten in Datenreihen. Leider bezieht sich das auf Sampling-Reihen, bei denen sich die Werte der Samples periodisch verhalten. In dem hier vorliegenden Fall handelt es sich aber um eine Reihe unsortierter Zahlenwerte, die untereinander ein regelmäßiges System aufbauen.

Da leider keine elegante mathematische Lösung für Gl. (2.12) gefunden wurde, wurde also versucht die Lösung algorithmisch zu bestimmen.

### 2.5.3 Das Vorgehen

Da sich die Features der detektierten Gruppen über das gesamte Bild erstrecken, die ermittelten Translationsvektoren also so genannte Freie Vektoren sind, können sie keinem bestimmten Bildelement zugeordnet werden.

Hinzu kommt, dass sich die verschiedenen Basiselemente in den detektierten Translationen überlagern. Abb. 2.12 zeigt diesen *Schatteneffekt* anhand des zweiten Basiselements  $v_c^2$  in Spaltenrichtung, des Beispiels aus Abb. 2.10. Es ergeben sich drei Möglichkeiten, um von der linken zur rechten Fenstergruppe zu gelangen. Nur eine davon, die oberste, entspricht dem gesuchten Basiselement  $v_c^2$  in dieser Richtung. Die anderen beiden entstehen durch Überlagerung mit dem ersten Basiselement  $v_c^1$ , dem Abstand zwischen den beiden Fenstern.

Die Idee ist nun eine relative Betrachtung der detektierten Translationen. Es werden Abstände/Differenzen zwischen den Translationen gesucht die oft vorkommen.

Anhand der Datenreihe aus Abb. 2.13(a), zum Beispiel aus Abb. 2.9, soll das weitere Vorgehen Schritt für Schritt erläutert werden.

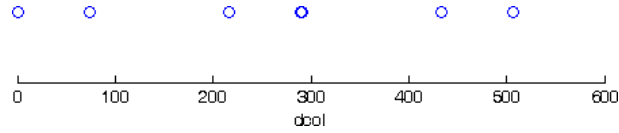
Alle in Spaltenrichtung detektierten Translationen seien im Vektor  $\mathbf{d}$  gespeichert.

#### 2.5.3.1 Basiselemente suchen

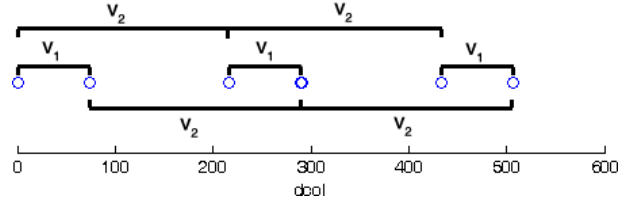
Gesucht sind gleich abständige Gruppierungen innerhalb dieser Reihe und die Beziehungen zwischen diesen Gruppen. Neben allen anderen möglichen Gruppierungen sind das die in Abb. 2.13(b) eingezeichneten Gruppen mit den Abständen  $v_1$  und  $v_2$ .

Zunächst werden also für alle Kombinationen von Punktpaaren deren Abstände bestimmt, d.h. es werden die Differenzen zwischen allen aufgetretenen Translationen dieser einen Richtung bestimmt. Die Annahme dahinter ist die, dass die Abstände, die die Struktur der Reihe bestimmen am häufigsten vorkommen.

Abb. 2.14(a) zeigt die Häufigkeitsverteilung für das betrachtete Beispiel in Form eines Histo-

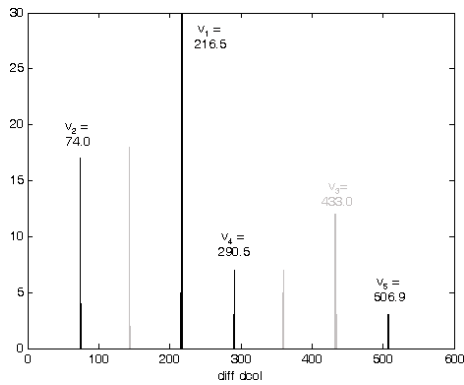


(a) Die detektierten Translationen in Spaltenrichtung aus dem Bsp. aus Abb. 2.9.

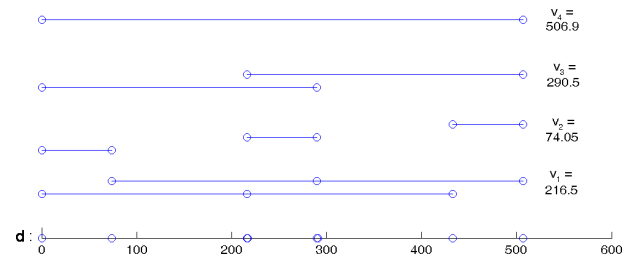


(b) Die gesuchten Gruppierungen in dieser Reihe.

Abbildung 2.13: Ziel der Gruppierung zu gleich abständigen Punkten



(a) Häufigkeitsverteilung der Differenzen zwischen den detektierten Translationen aus Abb. 2.13(a).



(b) Aus dem Histogramm abgeleitete potentielle Kandidaten für Basiselemente, sortiert nach ihrem Auftreten in  $\mathbf{d}$ , und die dazu gefundenen Gruppierungen in  $\mathbf{d}$ .

Abbildung 2.14: Ableitung der Gruppierungen aus dem Histogramm der Differenzen zwischen den detektierten Translationen. Beträge, die nicht im Eingangsvektor  $\mathbf{d}$  auftauchen, werden nicht berücksichtigt. Die entsprechenden Peaks sind hier grau markiert. Des Weiteren bleiben Werte unberücksichtigt, die Vielfache eines anderen sind.

gramms. Die Peaks dieses Histogramms werden im Folgenden als potentielle Kandidaten für die gesuchten Basiselemente betrachtet. Beträge, die nicht im Eingangsvektor  $\mathbf{d}$  auftauchen, werden dabei nicht berücksichtigt. Die entsprechenden Peaks sind hier grau markiert. Der Grund für diese Einschränkung liegt in der Annahme, dass die gesuchten Basiselemente so charakteristisch für die Bildstruktur sind, dass sie vom Detektor als Translation zwischen zwei wiederholten Gruppen detektiert werden. Diese Annahme ist gleichzeitig wichtige Voraussetzung für die nachfolgenden Schritte zur Selektion der minimalen Anzahl von Basiselementen aus den ermittelten potentiellen Kandidaten.

Als weitere Bedingung wird vorausgesetzt, dass die Basiselemente nicht Vielfache voneinander sind, andernfalls wäre das Problem (auch mathematisch) nicht zu lösen. Dementsprechend können Kandidaten entfernt werden, die Vielfache eines anderen Kandidaten sind. In diesem Fall wird der kleinere Wert beibehalten und der größere aus der Kandidatenmenge entfernt. Im gezeigten Beispiel ist das  $v_3$  mit 433.0 als Vielfaches von  $v_1$  mit 216.5.

Um die Kandidatenmenge weiter einzuschränken, werden sie nach ihrem Auftreten in  $\mathbf{d}$  sortiert. Damit wird die Qualität der detektierten Verschiebung berücksichtigt, die durch den akkumulierten Repetitionsgrad der beteiligten Matches der Gruppen und damit der Reihenfolge der Detektion bestimmt ist.

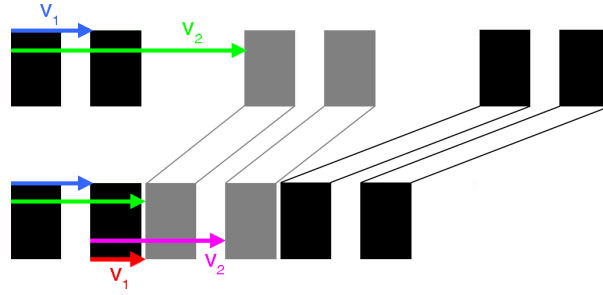


Abbildung 2.15: Was passiert, wenn  $(a_i + 1) \cdot v_i \geq v_j$  nicht erfüllt ist? Oben: Der Zustand wie in Abb. 2.10. Unten:  $(a_i + 1) \cdot v_i \leq v_j$  bei der gleichen Gruppierung der Elemente wie oben (grün und blau). Bleibt man aber bei der „Doppelfenster-Interpretation“, so werden andere „Einzelfenster“ zu „Doppelfenster“ gruppiert (rot und magenta). Für diese Gruppierung ist die geforderte Bedingung wieder erfüllt.

Für jede der verbleibenden Translationen  $v_i$  werden nun die Gruppen von Punkten in der Reihe gesucht, die genau diesen Abstand haben, siehe Abb. 2.14(b).

Anschließend werden Überlappungen zwischen den Gruppen der verschiedenen Kandidaten gesucht. Sei  $a_i$  die Anzahl der Wiederholungen eines Basiselements  $v_i$  und  $v_j$  ein weiteres potentielles Basiselement. Wenn  $(a_i + 1) \cdot v_i \geq v_j$ , dann Überlappen sich diese beiden Basiselemente. In diesem Fall wird dem Kandidaten der Vorzug gegeben, der in der Sortierung weiter vorn zu finden ist, der nach dem akkumulierten Repetitionsgrad aus der Detektion der Gruppen also mit der größeren Sicherheit gefunden wurde. Der Begriff Überlappung ist hier eigentlich nicht ganz präzise gewählt. Für eine Überlappung der Basiselemente genügt schon die Bedingung  $a_i \cdot v_i \geq v_j$ . Die Erweiterung um eine Wiederholung sorgt dafür, dass die einzelnen Gruppen eines Basiselements so weit voneinander entfernt sind, dass der Abstand zwischen den Gruppen größer ist, als der zwischen den einzelnen Elementen.

Zur Veranschaulichung soll das Beispiel aus Abb. 2.15 dienen. Zum Basiselement  $v_1$  zählen zwei Elemente, die „Einzelfenster“. Anders ausgedrückt: Das Element Fenster wiederholt sich einmal im Abstand  $v_1$ . Zum Basiselement  $v_2$  zählen drei Elemente, die „Doppelfenster“, die sich aus den „Einzelfenster“ im Abstand  $v_1$  zusammensetzen. Das Element „Doppelfenster“ wiederholt sich zweimal im Abstand  $v_2$ .

Wenn die Bedingung  $2 \cdot v_1 \geq v_2$  nicht erfüllt wäre, dann würden die „Doppelfenster“ so sehr zusammenrücken, dass der Abstand zwischen ihnen kleiner wäre, als der zwischen den „Einzelfenster“. Damit würde sich aber die Interpretation der Struktur völlig verändern. Es würden andere Gruppierungen zu „Doppelfenster“ entstehen. Für das Beispiel in Abb. 2.15 (unten) erfasst die (in rot und magenta eingezeichnete) Basis nur die beiden mittleren „Doppelfenster“. In diesem Fall wäre die „falsche“ Basis, mit der Gruppierung der Elemente wie oben sogar die bessere Wahl. Für die Lösung des Problems - Wahl der „richtigen“ Basiselemente - mussten aber geeignete Kriterien gefunden werden, um aus der Menge der potentiellen Kandidaten eine minimale Anzahl von Basiselementen zu finden, die die Struktur der Basis optimal beschreiben. Für dieses Ziel war die Einführung solcher Randbedingungen zwingend notwendig.

Abb. 2.16 zeigt schließlich die nach diesen Selektionsschritten verbliebenen Basiselemente und Gruppen für das betrachtete Beispiel.

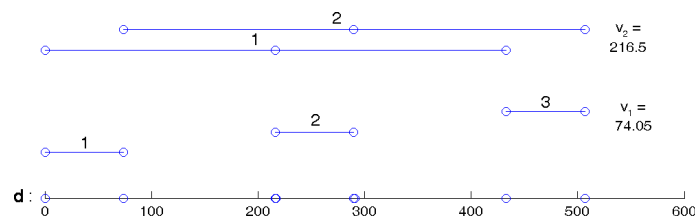


Abbildung 2.16: Die nach allen Selektionsschritten verbliebenen Basiselemente und Gruppen, der Größe nach sortiert.



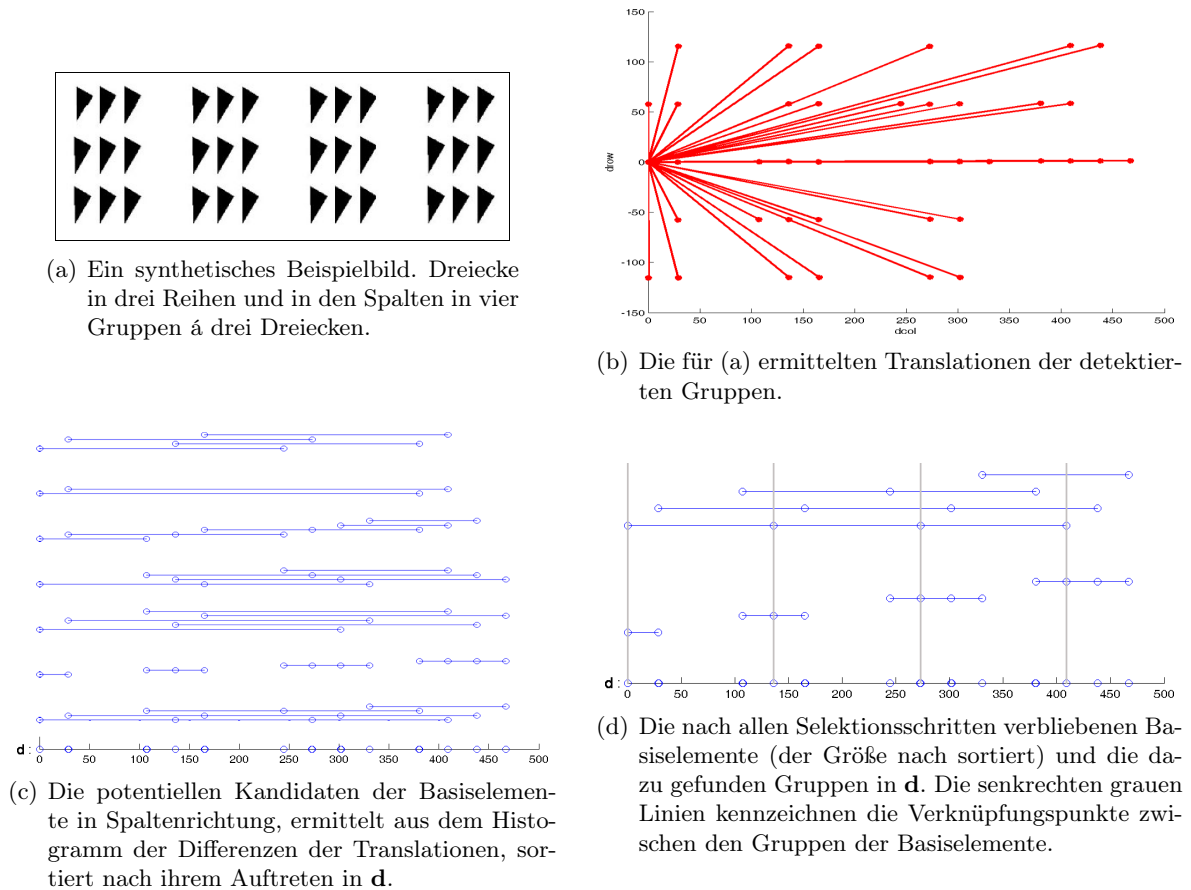


Abbildung 2.17: Die Ergebnisse der Detektion wiederholter Gruppen für ein anderes synthetisches Beispiel, in Form der detektierten Translationsvektoren und die Ergebnisse der Analyse dieser Translationen.

### 2.5.3.2 Verknüpfungen suchen

Für dieses Beispiel wäre das Problem an dieser Stelle schon gelöst. Es wurden zwei Basiselemente identifiziert und alle Gruppen dieser Basiselemente haben die gleiche Anzahl von Wiederholungen. Das Ergebnis wäre hier  $v_1 = 74.0$  mit zwei Elementen pro Gruppe und  $v_2 = 216.5$  mit drei Elementen pro Gruppe.

Das die Aufgabe i.Allg. an dieser Stelle noch nicht umfassend gelöst ist, soll das Bsp. aus Abb. 2.17 verdeutlichen. Dargestellt ist wieder ein synthetisches Beispiel, hier Dreiecke in drei Reihen, in vier Gruppen á drei Dreiecken. Die Abb. 2.17(b) bis (d) visualisiert die Schritte bis zur endgültigen Auswahl der Basiselemente, wie sie in den letzten Abschnitten beschrieben wurden.

In der Darstellung aller zu den Basiselementen in **d** gefundenen Gruppen in Abb. 2.17(d) ist zu erkennen, dass hier nicht alle zu einem Basiselement gehörigen Gruppen die gleiche Anzahl von Elementen besitzen. Das resultiert zum einen aus Fehlern in der Detektion, d.h. nicht alle Translationen zwischen den Elementen der Bildstruktur, hier den Dreiecken, wurden detektiert. Deshalb fehlt z.B. in der ersten Gruppe des kleineren Basiselements das dritte Gruppenelement. Des Weiteren macht sich hier der bereits in Abs. 2.5.2 (S.24) erwähnte „Schatteneffekt“ bemerkbar, weshalb andere Gruppen dieses Basiselements durch vier statt drei Elemente repräsentiert werden.

Um also zu entscheiden, wie viele Elemente einem Basiselement zuzuordnen sind, sind diejenigen Gruppen zu identifizieren die untereinander, also zwischen den Basiselementen, verbunden sind.

Zur Verdeutlichung dieses Gedankens sei noch einmal die Abb. 2.10 (S.23) betrachtet. Das gesamte Bild, bzw. der durch die Basis beschriebene regelmäßige Anteil eines Bildes, soll sich durch die Kombination der gefundenen Basiselemente beschreiben lassen. D.h. jeder der kleineren Pfeile lässt sich an den Knoten der größeren Elemente anknüpfen, um zu einem weiteren Element der Fassade zu gelangen. Diese Verknüpfungspunkte müssen also durch die Elemente der Basis gewährleistet sein.

Nachdem die Basiselemente der Größe nach sortiert wurden, werden sie von oben nach unten durchsucht. Für die oberste Gruppe kann nur die erste Gruppe, mit dem ersten Element bei 0, die gesuchte Gruppe sein. Hier kann auch nur die Anzahl der Elemente dieser Gruppe für dieses Basiselement übernommen werden. Für alle folgenden Elemente werden nun die Gruppen gesucht, die gemeinsame Punkte mit den ausgewählten Gruppen des Vorgänger-Basiselements haben. Diese Verknüpfungspunkte, in Abb. 2.17(d) durch die senkrechten grauen Linien symbolisiert, definieren für jede ausgewählte Gruppe den „Startpunkt“. Die maximale Anzahl der Elemente einer Gruppe nach ihrem Startpunkt definiert dann die Anzahl der Elemente dieses Basiselements im Bild.

Damit sind alle Basiselemente bekannt, mit Betrag der Translation und Anzahl der Elemente bzw. Wiederholungen mit dieser Verschiebung.

## 2.6 Bestimmung der Relativen Orientierung aus einem Bild eines Objektes mit spiegelsymmetrischen Eigenschaften

Falls ein Bild ein spiegelsymmetrisches Objekt wiedergibt, so kann man sich eine virtuelle Kamera denken, die genau spiegelsymmetrisch zur Originalaufnahmesituation positioniert ist. Die im Bild identifizierten symmetrischen Features ermöglichen die Zuordnung homologer Punkte zwischen dem Original- und dem gespiegelten Bild. Diese Situation ist in Abb. 2.18 dargestellt.

Die Relative Orientierung für kalibrierte Kameras ist durch fünf Parameter bestimmt, drei für die räumliche Rotation und zwei für die Richtung der Basis, die Länge der Basis bleibt unbestimmt ([McGlone, 2004], Abs. 3.2.2.2).

In dieser speziellen Situation ergeben sich besondere Beziehungen der Drehwinkel der Kamerakoordinatensysteme. Zur Bezeichnung der Achsen und Winkel siehe Abb. 2.18.

Sei das Objektkoordinatensystem mit seiner  $X$ -Achse genau in Richtung der Basis  $\mathbf{b}$  angeordnet.

$$\Rightarrow \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.13)$$

Die Orientierungen der Kameras sind nur bezüglich ihrer Drehungen um die  $^c x$ - und  $^c z$ -Achsen veränderlich, so dass gilt

$$\omega'' = \omega', \quad \phi'' = -\phi', \quad \kappa'' = -\kappa'. \quad (2.14)$$

Die Relative Orientierung ist hier also durch nur zwei Parameter bestimmt und lässt sich aus nur zwei Punktpaaren im Bild ableiten.

Rotationen lassen sich bequem mit Hilfe von Quaternionen ausdrücken. Ein Quaternion  $\mathbf{q} = [q_0, q_1, q_2, q_3]$  normiert, so dass  $\mathbf{q} = [1, \tan \alpha/2 \mathbf{r}]$  stellt die Beziehung zu einer Darstellung der Rotation in Drechachse, repräsentiert durch den Einheitsvektor  $\mathbf{r}$ , und Drehwinkel  $\alpha$  um diese Achse her ([McGlone, 2004], Abs. 2.1.2.5). Da die Kameras hier nur um die  $X$ -Achse drehbar sind, ist die erste Komponente des Vektors  $\mathbf{r}$  0. Damit ergibt sich die Reduktion der Quaternionen-Rotationen auf nur zwei Parameter.

Die Rotations-Matrizen der beiden Kameras ergeben sich damit in Quaternionen-Schreibweise zu

$$\mathbf{R}' = \mathbf{R}_Q([1, 0, q_2, q_3]) = \frac{1}{1 + q_2^2 + q_3^2} \begin{bmatrix} 1 - q_2^2 - q_3^2 & -2q_3 & 2q_2 \\ 2q_3 & 1 + q_2^2 - q_3^2 & 2q_2q_3 \\ -2q_2 & 2q_2q_3 & 1 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.15)$$

und

$$\mathbf{R}'' = \mathbf{R}_Q([1, 0, -q_2, -q_3]) = \frac{1}{1 + q_2^2 + q_3^2} \begin{bmatrix} 1 - q_2^2 - q_3^2 & 2q_3 & -2q_2 \\ -2q_3 & 1 + q_2^2 - q_3^2 & 2q_2q_3 \\ 2q_2 & 2q_2q_3 & 1 - q_2^2 + q_3^2 \end{bmatrix}. \quad (2.16)$$

Sowohl die Basis  $\mathbf{b}$  als auch die Verbindungslinien der symmetrischen Features bzw. homologen Punkte sind per Definition rechtwinklig zur Symmetrieebene. Damit schneiden sie sich in einem gemeinsamen Fluchtpunkt, dem unendlichen Punkt der Basis, dem Epipol (des ersten oder zweiten Bildes).

Aus den Rotationsmatrizen und der schiefsymmetrischen Matrix

$$\mathbf{S}([x, y, z]) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \Rightarrow \mathbf{S}(\mathbf{b}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.17)$$

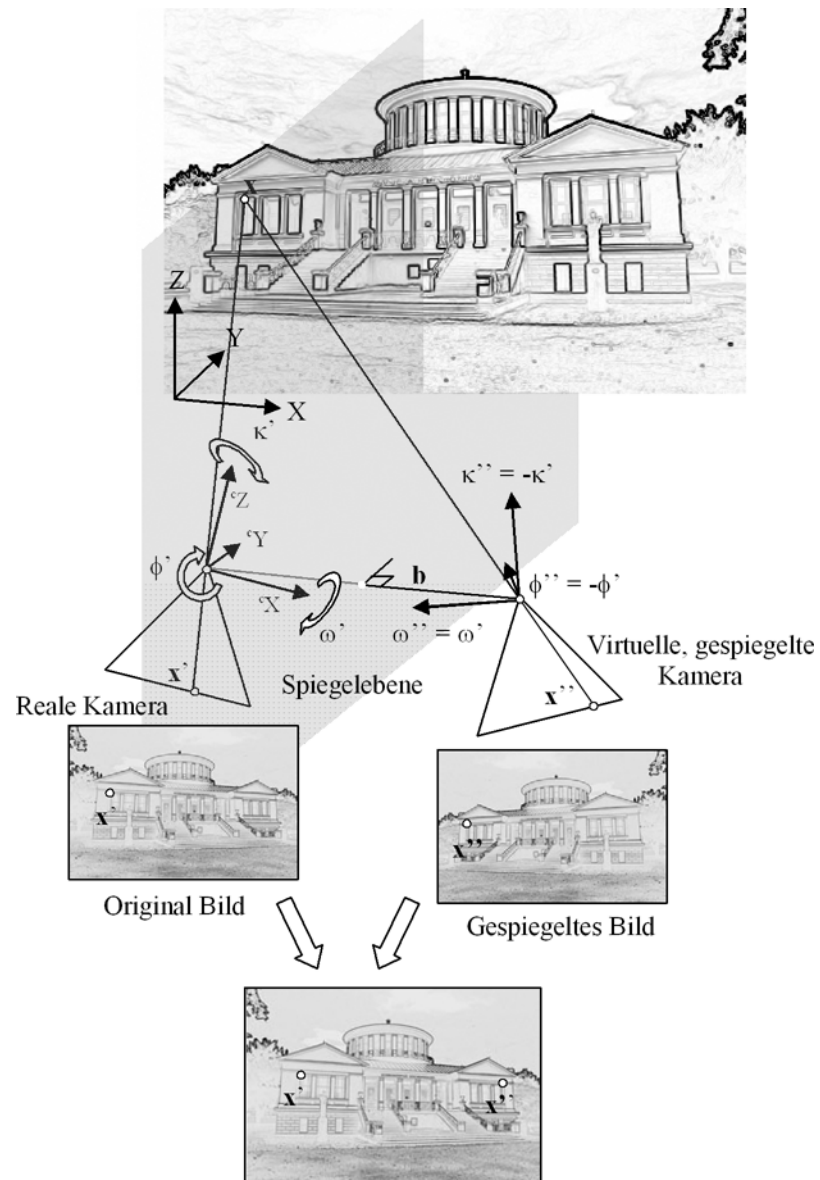


Abbildung 2.18: Prinzip der Zuordnung homologer Punkte bei spiegelsymmetrischen Objekten zur Herstellung der Relativen Orientierung aus nur einem Bild.

ergibt sich die Essentielle Matrix zu

$$\mathbf{E} = \mathbf{R}'\mathbf{S}(\mathbf{b})\mathbf{R}''^T = \frac{1}{1 + q_2^2 + q_3^2} \begin{bmatrix} 0 & 2q_2 & 2q_3 \\ 2q_2 & 0 & -1 + q_2^2 + q_3^2 \\ 2q_3 & 1 - q_2^2 - q_3^2 & 0 \end{bmatrix}. \quad (2.18)$$

Die Koplanaritätsbedingung für die Epipole lautet

$$\mathbf{E}\mathbf{e}'' = 0 \text{ bzw. } \mathbf{e}'^T\mathbf{E} = 0. \quad (2.19)$$

Der Epipol im ersten Bild lässt sich damit ausdrücken als

$$\mathbf{e}' \sim \begin{bmatrix} 1 - q_2^2 - q_3^2 \\ 2q_3 \\ -2q_2 \end{bmatrix}. \quad (2.20)$$

Die Bildkoordinaten in homogener Schreibweise sind gegeben durch:

$$\mathbf{x}_i^* = \begin{bmatrix} x_i^* \\ y_i^* \\ 1 \end{bmatrix}. \quad (2.21)$$

Die Verbindungslinien der homologen Punkte ergeben sich über das Kreuzprodukt ihrer Bildkoordinaten  $\mathbf{l}_1 = (\mathbf{x}'_1 \times \mathbf{x}''_1)$ ,  $\mathbf{l}_2 = (\mathbf{x}'_2 \times \mathbf{x}''_2)$ . Der Schnittpunkt dieser Linien wiederum ergibt sich aus dem Kreuzprodukt ihrer homogenen Koordinaten

$$\mathbf{e}' = (\mathbf{l}_1 \times \mathbf{l}_2) = (\mathbf{x}'_1 \times \mathbf{x}''_1) \times (\mathbf{x}'_2 \times \mathbf{x}''_2). \quad (2.22)$$

Damit lassen sich aus zwei Punktpaaren symmetrischer Features die Epipole und aus diesem dann über Gl.(2.20) die Parameter zur Berechnung der Rotationen in Gl. (2.15) bzw. Gl. (2.16) bestimmen.

Mit der Herstellung der Relativen Orientierung gelangt man zum so genannten photogrammetrischen Modell. Durch räumliche Vorwärtsschnitte der Strahlenbündel beider Kameras kann so eine maßstäbliche 3D-Rekonstruktion des Objektes erfolgen.

Da sich die Überlegungen dieser Arbeit auf die Detektion der Symmetrien und wiederholten Strukturen und insbesondere auf die Erkennung der regelmäßigen Struktur von Fassaden konzentrierten, seien diese Überlegungen hier nur der Vollständigkeit halber präsentiert. Dieses Konzept wurde nur theoretisch durchdacht, konnte aber nicht mehr praktisch umgesetzt werden.

# 3. Umsetzung

## Implementierung und Ergebnisse

In diesem Kapitel soll darauf eingegangen werden, wie die beschriebenen Konzepte umgesetzt wurden und welche Ergebnisse mit den erarbeiteten Algorithmen zu erzielen waren.

Da das Ziel dieser Arbeit nicht in der Erarbeitung eines kompletten leistungsfähigen Softwarepaketes bestand, sondern vielmehr in der Erprobung einzelner Konzepte, wurde auf die komfortable Entwicklungsumgebung Matlab<sup>1</sup> zurückgegriffen. Hier stehen zahlreiche Methoden beispielsweise der Linearen Algebra oder Bildverarbeitung bereits vorgefertigt zur Verfügung und müssen nicht selbst implementiert oder über externe Bibliotheken importiert werden.

Die in Abs. 2.3 und Abs. 2.4 vorgestellten Verfahren basieren auf der Detektion von charakteristischen Bildmerkmalen. Der dafür verwendete Feature-Detektor SIFT, vorgestellt in Abs. 2.3.1 (S. 14), wurde in der von Lowe zur Verfügung gestellten Matlab-Implementierung<sup>2</sup> verwendet.

### 3.1 Feature-Punkte

Der erste Schritt besteht also in der Detektion der Features im Bild.

```
[image, desc, locs] = sift(pict);
```

In der hier verwendeten (auf der CD beiliegenden) Version des SIFT-Codes ist `pict` ein Graustufenbild in beliebigem Format<sup>2</sup>. Für die Untersuchungen dieser Arbeit wurden JPEG-Bilder verwendet.

Falls keine anderen Angaben gemacht werden, handelt es sich bei den Testbildern um Graustufenbilder im Format 512x340 Pixel.

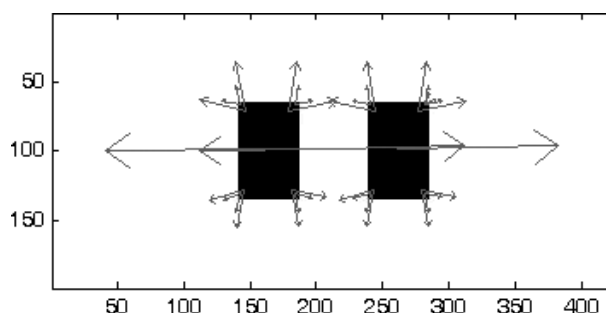
Zurückgegeben werden neben der Bild-Matrix `image` die in Abs. 2.3.1 (S.14) erläuterten Vektoren zur Beschreibung der  $n$  detektierten Lowe-Punkte.

- `locs` ist eine  $n \times 4$  Matrix, mit den Angaben  $row_i$ ,  $column_i$ ,  $scale_i$  und  $\phi_i$  ( $i = 1 \dots n$ ) für jedes detektierte Feature in den Zeilen.
- `desc` ist eine  $n \times 128$  Matrix, mit den Deskriptoren  $k_i$  ( $i = 1 \dots n$ ) jedes Features in den Zeilen.

---

<sup>1</sup>MATLAB *The Language of Technical Computing*, Version 7.0.4.365 (R14) Service Pack 2, The MathWorks, Inc.

<sup>2</sup>Falls die verwendete Matlab-Lizenz die Image Processing Toolbox umfasst, so kann durch Auskommentierung der Zeilen 26-28 in `sift.m` erreicht werden, dass auch RGB-Bilder als Eingabe akzeptiert werden. Diese werden dann aber auch in Graustufenbilder umgewandelt und im Folgenden als ein Kanal betrachtet.

Abbildung 3.1: Von `sift.m` detektierte Keypoints für ein synthetisches Bild

Die detektierten Lowe-Punkte können mit

```
showkeys(image, locs);
```

visualisiert werden.

Abb. 3.1 zeigt das Ergebnis der Feature-Detektion anhand eines kleinen synthetischen Beispiels eines „Doppelfensters“. Gut zu erkennen ist die doppelte Belegung mit Feature-Punkten an den Ecken der Fenster. Die Kanten erzeugen hier verschiedene Hauptgradientenrichtungen, so dass jeweils zwei Feature-Punkte erzeugt werden. Die unterschiedliche Länge der Pfeile deutet auf die unterschiedlichen Skalen der Features. Die großen Pfeile deuten auf eine große Skala und beziehen sich auf das gesamte „Doppelfenster“ als ein Feature und sind deshalb auch mittig positioniert. Die ganz kleinen Pfeile dagegen sitzen sehr eng an den Fensterecken, sie beziehen sich, mit einer kleinen Skala, auf kleinere Bildmerkmale, hier die Ecken.

In den Experimenten von [Loy and Eklundh, 2006] wurden keine Gebäude gezeigt. Die Resultate wurden hier anhand von Schmetterlingen, Gesichtern oder Autos präsentiert. Daher stellte sich bei der Konzeption der Aufgabenstellung die Frage, ob das Verfahren auf Fassaden überhaupt anwendbar ist. Eine wichtige Fragestellung dabei war, ob die Ergebnisse der Feature-Detektion auf Fassadenbildern gut genug sind, um aus ihnen symmetrische oder wiederholte Strukturen abzuleiten. Die in Abs. 3.2.6 und 3.3.2 vorgestellten Ergebnisse werden zeigen, dass dem sehr wohl so ist. Es wurden bei 512x340Pixel großen Aufnahmen, je nach Struktur der Fassade und Kontrast der Aufnahme, ca. 500-1500 Features detektiert und darunter teilweise über 1000 symmetrische oder wiederholte Matches gefunden.

## 3.2 Detektion von Symmetrien

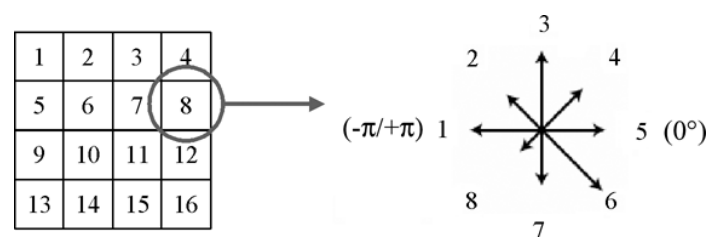
### 3.2.1 Implementation

Für die Umsetzung des in Abs. 2.3.2 (S.16) beschriebenen Verfahrens wurde die Matlab-Funktion `detectSym` geschrieben. Der Funktionsaufruf lautet:

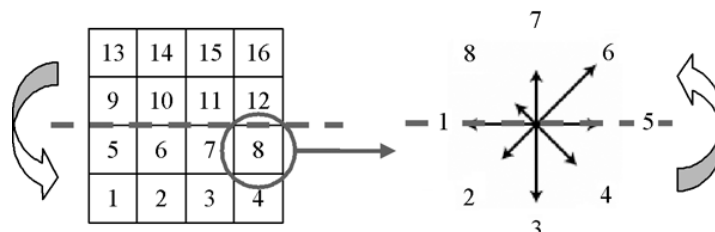
```
Sym = detectSym(image, desc, locs);
```

Als Eingabe-Parameter werden die Ergebnisse des Feature-Detektors übergeben. Der Rückgabe-Parameter `Sym` ist ein Array von Structs mit den Parametern jeder gefundenen Symmetrieachse:

- `.r` und `.theta` : Die Polarkoordinaten der gefundenen Symmetrieachse.
- `.s_r` und `.s_theta` : Die entsprechenden Standardabweichungen.
- `.points_a` und `.points_b` : Die Bildkoordinaten der zur Symmetrieachse gehörigen Keypoints und ihre symmetrischen Matches, jeweils als `[1x2]`-Vektor.



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



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

Abbildung 3.2: Spiegelung des Deskriptors

Die Details zur Umsetzung der vorgestellten Konzepte sollen in den folgenden Abschnitten näher erläutert werden.

### 3.2.2 Spiegelung der Deskriptoren

Für die Suche symmetrischer Feature-Paare stellte sich in Abs. 2.3.2.2 (S.16) die Frage, wie man zu den gespiegelten Features gelangt. Es wurde entschieden, die Deskriptoren derart zu verändern, dass sie eine gespiegelte Kopie der lokalen Umgebung des Keypoints darstellen.

In Abs. 2.3.1.1 (S.15) wurde erläutert, wie die Deskriptoren der Features entstehen. In der Kommentierung zum Lowe-Code ist leider nicht verzeichnet, in welcher Reihenfolge die Elemente der Histogramme abgespeichert werden. Für 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. Die Autoren Loy und Eklundh haben auf dieses Detail leider auch verzichtet und nur darauf hingewiesen, dass für die Spiegelung der Deskriptoren ihre Elemente geeignet umzusortieren sind.

Durch einige kleine Versuche mit den Ergebnissen des SIFT-Operators hat sich das in Abb. 3.2 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. 3.2(a). Es müssen also zunächst die 8er-Blöcke der Orientierungshistogramme derart umsortiert werden, dass die Blöcke zeilenweise vertikal gespiegelt werden, wie in Abb. 3.2(b) links dargestellt. Und auch die Histogramme selbst werden vertikal gespiegelt, d.h. die Einträge werden umsortiert, wie in Abb. 3.2(b) rechts dargestellt.

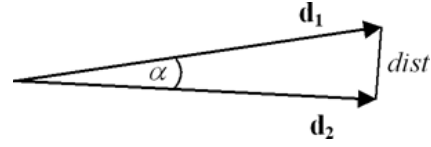
### 3.2.3 Matching der Feature-Deskriptoren

Beim Matching sollen die Feature-Paare mit der besten Übereinstimmung ihrer Deskriptoren gefunden werden. Die besten Kandidaten dafür ergeben sich aus den nächsten Nachbarn im



128-dimensionalen Raum der Deskriptoren. Dafür werden die euklidischen Distanzen aller Kombinationen von Feature-Paaren bestimmt.

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.



$$\alpha \cong \arccos \langle \mathbf{d}_1, \mathbf{d}_2 \rangle \quad \text{für kleine } \alpha \text{ und } |\mathbf{d}_1| = |\mathbf{d}_2| = 1 \quad (3.1)$$

Für die Bestimmung der euklidischen Abstände zwischen allen  $n$  Deskriptoren nach Gl. (2.2) sind  $n!$  Differenzen zu berechnen. Für die Näherung aus Gl. (3.1) dagegen kann die Berechnung aller Skalarprodukte in einer Matrixoperation erfolgen. Nach Bildung des  $\arccos$  ist von der resultierenden  $[n \times n]$  Matrix dann nur noch die obere Dreiecksmatrix von Interesse, da jedes Feature-Paar doppelt registriert ist.

Da sich diese Arbeit mit wiederholten Strukturen beschäftigt, kann natürlich nicht davon ausgegangen werden, genau einen Match pro Feature zu finden. Im Unterschied zu dem in [Lowe, 2004] beschriebenen Vorgehen ist der Zweck des Matchens hier nicht der, gleiche Feature in einem anderen Bild wiederzufinden (Gleich im Objektraum, in den Abbildungen ergeben sich natürlich Unterschiede in Beleuchtung, Blickwinkel, Maßstab . . .). Hier besteht die Idee darin, sich wiederholende Features innerhalb *eines* Bildes zu finden. Daher werden korrekte Matches hier wie folgt identifiziert:

Zu einem Feature werden die euklidischen Distanzen, bzw.  $\alpha$  nach Gl.(3.1), zu allen gespiegelten Features bestimmt<sup>3</sup> und der Größe nach sortiert. Vom kleinsten Wert beginnend, wird nun ein „Bruch“ in der Reihe gesucht, an dem das Verhältnis zwischen zwei aufeinander folgenden Werten größer  $distRatio$  ist. Alle Features die zu den Distanzen vor dem „Bruch“ gehören, werden als Match zum betrachteten Feature registriert.

Für die Experimente dieser Arbeit hat sich  $distRatio = 0.8$  bewährt.

### 3.2.4 Bewertung der Symmetrie einzelner Feature-Paare

Die Bewertung der Qualität der Symmetrie der gematchten Feature-Paare erfolgt über den in Gl. (2.7) (S.19) vorgestellten Symmetriegrad  $M$ .

Da a priori keine Aussagen über die zu erwartende Ausdehnung der symmetrischen Objekte getroffen werden kann, wurde für das Distanz-Gewicht aus Gl. (2.6)  $D_{ij} = 1$  gewählt.

Damit berechnet sich der Symmetriegrad aus:

$$M_{ij} = \Phi_{ij} S_{ij} \quad (3.2)$$

Mit dem Winkel-Symmetrie-Gewicht  $\Phi_{ij} \in [0, 2]$  aus Gl. (2.4)

$$\Phi_{ij} = 1 - \cos({}^{xy}\phi_i + {}^{xy}\phi_j - 2 {}^{xy}\theta_{ij})$$

<sup>3</sup>D.h. es wird eine Zeile der Distanz-Matrix ab dem Diagonalelement betrachtet.



Abbildung 3.3: Bester und schlechtester Match für symmetrische Features. In *grün* der beste Match mit einem Symmetriegrad von  $M = 3.99$  und in *rot* der schlechteste (noch akzeptierte) Match mit  $M \approx 10^{-8}$ .

und dem Skalen-Gewicht  $S_{ij} \in [0, 1]$  aus Gl. (2.5) mit  $\sigma_s = 1$

$$S_{ij} = \left[ \exp \left( \frac{-|s_i - s_j|}{s_i + s_j} \right) \right]^2. \quad (3.3)$$

Abb. 3.3 zeigt für ein Beispielbild exemplarisch den besten (*grün*) und schlechtesten (*rot*) Match. Für den besten Match wurde ein Symmetriegrad von  $M = 3.99$  ermittelt. Die Pfeile sind augenscheinlich symmetrisch zur gedachten Symmetrieachse und haben die gleiche Länge. Die Positionen der Features beziehen sich bis auf einen Pixel Unterschied auf die entgegengesetzten Ecken zweier Fliesen.

Für den schlechtesten Match mit  $M \approx 10^{-8}$  sind die Pfeile offensichtlich nicht exakt symmetrisch zur gedachten Symmetrieachse und auch nicht exakt gleich lang. Trotz dieser Differenzen deutet die Position der Features tatsächlich auf symmetrische Objekte im Bild hin, weshalb solche Matches durchaus ihren Beitrag zur Gesamtanalyse leisten können und deshalb, mit geringerer Gewichtung, auch berücksichtigt werden sollten.

### 3.2.5 Ermittlung dominanter Symmetrien im Hough-Raum

Der Hough-Raum wird als 2D-Histogramm über die ermittelten  $\rho$  und  $\theta$  aufgebaut. Jeder Match wird einer Akkumulator-Zelle zugeordnet und diese Zelle dann entsprechend dem Symmetriegrad  $M$  des Matches inkrementiert. Für die hier gezeigten Experimente wurde, für die gaußsche Glättung des Akkumulators, einheitlich ein kleiner Filter der Größe  $3 \times 3$  mit  $\sigma = 1$  verwendet.

Abb. 3.4(b) zeigt das geglättete Hough-Array für das schon in Abb. 2.9 (S.22) vorgestellte Beispiel. In  $x$ -Richtung sind die Richtungen  $\theta$  und in  $y$ -Richtung die Abstände  $\rho$  zum Ursprung abgetragen. Je heller der Punkt, desto größer der Wert im Akkumulator. Deutlich zu erkennen sind die Peaks bei  $90^\circ$  und  $0^\circ$  bzw.  $180^\circ$ , wie sie für Fassaden typischerweise zu erwarten sind. In dieser Abbildung ist gut zu erkennen, dass der Wertebereich für  $\theta$  etwas unglücklich gewählt wurde. Die Polarkoordinaten der Symmetrieachsen der Matches wurden so normiert, dass  $\theta$  im Bereich  $[0, \pi[$  zum liegen kommt. So ist immer mit Peaks an den Akkumulator-Rändern zu rechnen, was die programmiertechnische Umsetzung etwas kompliziert. Sinnvoller ist sicherlich, einen um  $45^\circ$  oder  $22^\circ$  verschobenen Wertebereich für  $\theta$  zu wählen. So würden, unter der Annahme, dass bei Fassadenbildern typischerweise horizontale oder vertikale Symmetrien zu finden sind,

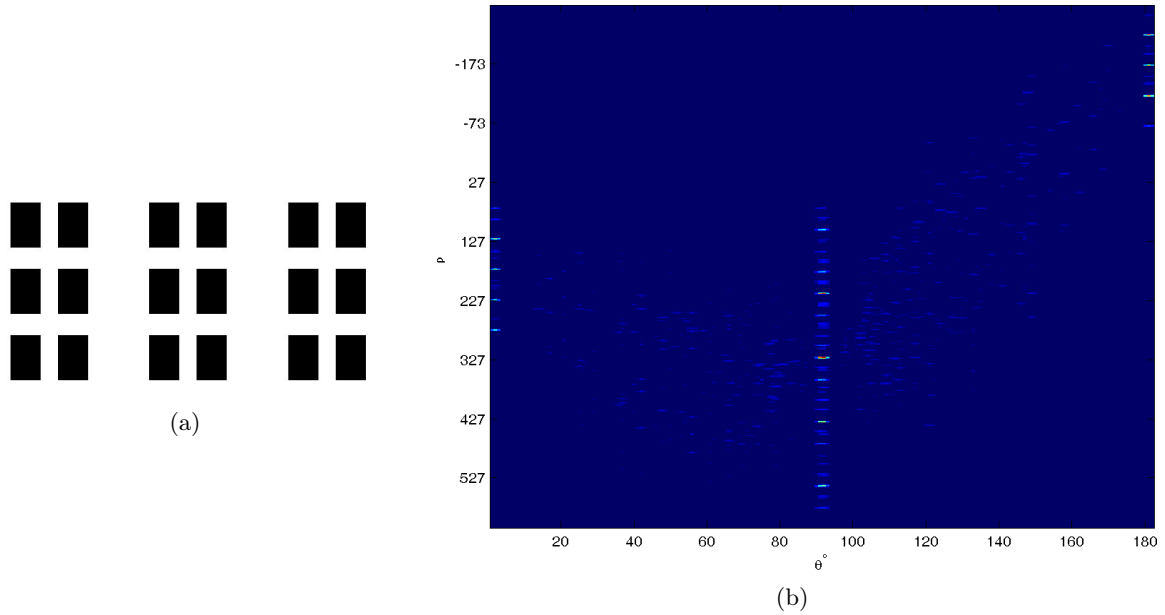


Abbildung 3.4: (a) Das schon bekannte Beispielbild einer typischen Fensterfront aus Abb. 2.9. (b) Der geglättete Hough-Akkumulator für die Symmetrie-Detektion, für dieses Beispiel. Je heller der Punkt, desto größer ist der Wert dieser Zelle. Zu beachten ist, dass die  $x$ - Achse in  $180^\circ$  Schritte und die  $y$ - Achse in ca. 700 1-Pixel Schritte unterteilt ist. Wegen dieser ungleichen Achseinteilung erscheinen die Peaks so langgezogen.

alle Peaks innerhalb des Akkumulators zum liegen kommen und die Ränder könnten außer acht gelassen werden.

Eine Normierung der Polarkoordinaten der Symmetrieachsen der Matches auf einen  $180^\circ$  breiten Wertebereich ist deshalb nötig, weil gleiche Achsen sonst durch zwei, um  $180^\circ$  versetzte Peaks im Akkumulator repräsentiert werden würden.

Aus der Polar-Darstellung der Symmetrieachsen, wie in Gl. (2.3) (S.18) ist leicht ersichtlich, dass sich das Vorzeichen von  $\rho$  umkehrt, falls  $\theta$  bzw. die Symmetrieachse um den Winkel  $\pi$  gedreht wird.

Im Hough-Akkumulator macht sich das dadurch bemerkbar, dass die „halben“ Peaks an den Rändern sich auf der anderen Seite des Akkumulators um  $2|\rho|$  verschoben fortsetzen.

Der Wertebereich für  $\rho$  wird dynamisch an die Ausdehnung des jeweiligen Datensatzes angepasst.

In der Matrix  $\mathbf{H}$   $[n \times 7]$  sind pro Zeile alle gefundenen Matches gespeichert mit

- den Koordinaten der gematchten Features  $row_i, col_i, row_j, col_j$
- der Polar-Darstellung der entsprechenden Symmetrieachse mit  $\rho$  und  $\theta$
- dem Symmetriegrad des Matches  $M$ .

Alle diese  $\rho$  und  $\theta$  werden im Hough-Array  $\bar{\mathbf{H}}$  gewichtet mit  $M$  akkumuliert.

Zur Bestimmung der dominanten Symmetrien aus dem Hough-Akkumulator wird dann wie folgt vorgegangen:

1. Suche Maximum im Hough-Array,  $\max(\bar{\mathbf{H}}) \rightarrow [x_{max}, y_{max}]$
2. Umrechnung der Akkumulator-Zelle  $[x_{max}, y_{max}]$  in  $[\rho_{max}, \theta_{max}]$

3. Suche alle Matches in **H**, deren Polar-Koordinaten der Symmetrieachse in einer kleinen Umgebung  $[d\rho, d\theta]^*$  um dieses Maximum liegen. (Beachte die Ränder des Akkumulators!\*\* )
4. (Anzahl der gefundenen Matches in **H**) < *Unterstuetzer*? \*\*\*
  - ja :  $\longrightarrow$  fertig, alle relevanten Peaks abgearbeitet.
  - nein : relevante Symmetrieachse gefunden
    - (a) Mittelbildung \*\*\*\*  $\longrightarrow \bar{\rho}, \bar{\theta}, \sigma_{\bar{\rho}}, \sigma_{\bar{\theta}}$ .
    - (b) Entferne alle Matches mit  $\rho = \rho_{max} \pm d\rho$  und  $\theta = \theta_{max} \pm d\theta$  aus dem Akkumulator. (Beachte die Ränder!\*\*) )
    - (c) Neues Maximum suchen  $\hookrightarrow 1$ .

\* Hier wurde standardmäßig mit  $d\rho = 6pix$  und  $d\theta = 1.5^\circ$  gearbeitet.

\*\* D.h.: falls  $\theta > (180^\circ - d\theta)$ , dann müssen Matches mit  $\dot{\theta} = \theta - 180^\circ$  und  $\dot{\rho} = -\rho$  mit betrachtet werden. Analog für  $\theta < 0^\circ + d\theta$ .

\*\*\* Für die Anzahl der beteiligten Matches wurde für die hier gezeigten Beispiele *Unterstuetzer* = 3 gewählt. D.h. jede Symmetrieachse muss durch mindestens drei Feature-Paare unterstützt werden.

\*\*\*\* Wie in Abs. 2.3.2.6 (S. 19) beschrieben.

### 3.2.6 Ergebnisse

Abb. 3.5<sup>4</sup> zeigt die Ergebnisse der Matlab-Funktion **detectSym** für das bereits vorgestellte synthetische Beispiel einer typischen Fensterfront. Die Abb. (a) bis (k) zeigen der Reihenfolge nach die ersten 11 detektierten Symmetrieachsen mit den Bounding Boxes der beteiligten Features. Und Abb. (l) zeigt alle 30 gefundenen Achsen im Überblick. In den Bildunterschriften finden sich die Angaben zur jeweiligen Symmetrie, die Koordinaten der Symmetrieachse  $[\bar{\rho}, \bar{\theta}]$  mit den jeweiligen Standardabweichungen  $[\sigma_{\bar{\rho}}, \sigma_{\bar{\theta}}]$ , sowie die Anzahl der beteiligten Matches, d.h. die Anzahl der Feature-Paare, die diese Achse unterstützen.

Abb. 3.6 zeigt in gleicher Art und Weise die Ergebnisse der Symmetrie-Detektion für die „echte“ Aufnahme einer großen Fensterfront.

Zunächst ist zu beobachten, dass die Anzahl der unterstützenden Features nicht direkt auf die Qualität der detektierten Symmetrie schließen lässt. Darin spiegelt sich die unterschiedliche Gewichtung der Feature-Paare, je nach ihrem Symmetriegrad wieder.

In den beiden Abbildungen zeigt sich, exemplarisch für alle anderen Experimente, dass die Hauptsymmetrien, so wie sie auch der menschliche Beobachter intuitiv wahrnimmt, zuerst, also mit der höchsten Qualität, erkannt werden.

Dass natürlich auch unerwünschte Nebeneffekte auftreten können zeigen die Abb. 3.5(j) und (k). Hier vermischen sich die Eigenschaften der Fassade. Die Struktur ist zum einen durch zahlreiche Symmetrien gekennzeichnet und zum anderen durch wiederholte Elemente.

Damit ergibt sich als weiterführende Aufgabe die Struktur der detektierten Symmetrien zu erfassen. Eine minimale Beschreibung, wie sie in Abs. 2.5 für wiederholte Gruppen abgeleitet wurde, müsste auch hier noch gefunden werden. Hinzu kommt die gemeinsame Betrachtung der beiden Beschreibungen *Symmetrie* und *Wiederholung*.

<sup>4</sup>Um die Ergebnisse besser erkennen zu können, wurden die Testbilder in nahezu allen Abbildungen um 20% heller dargestellt.

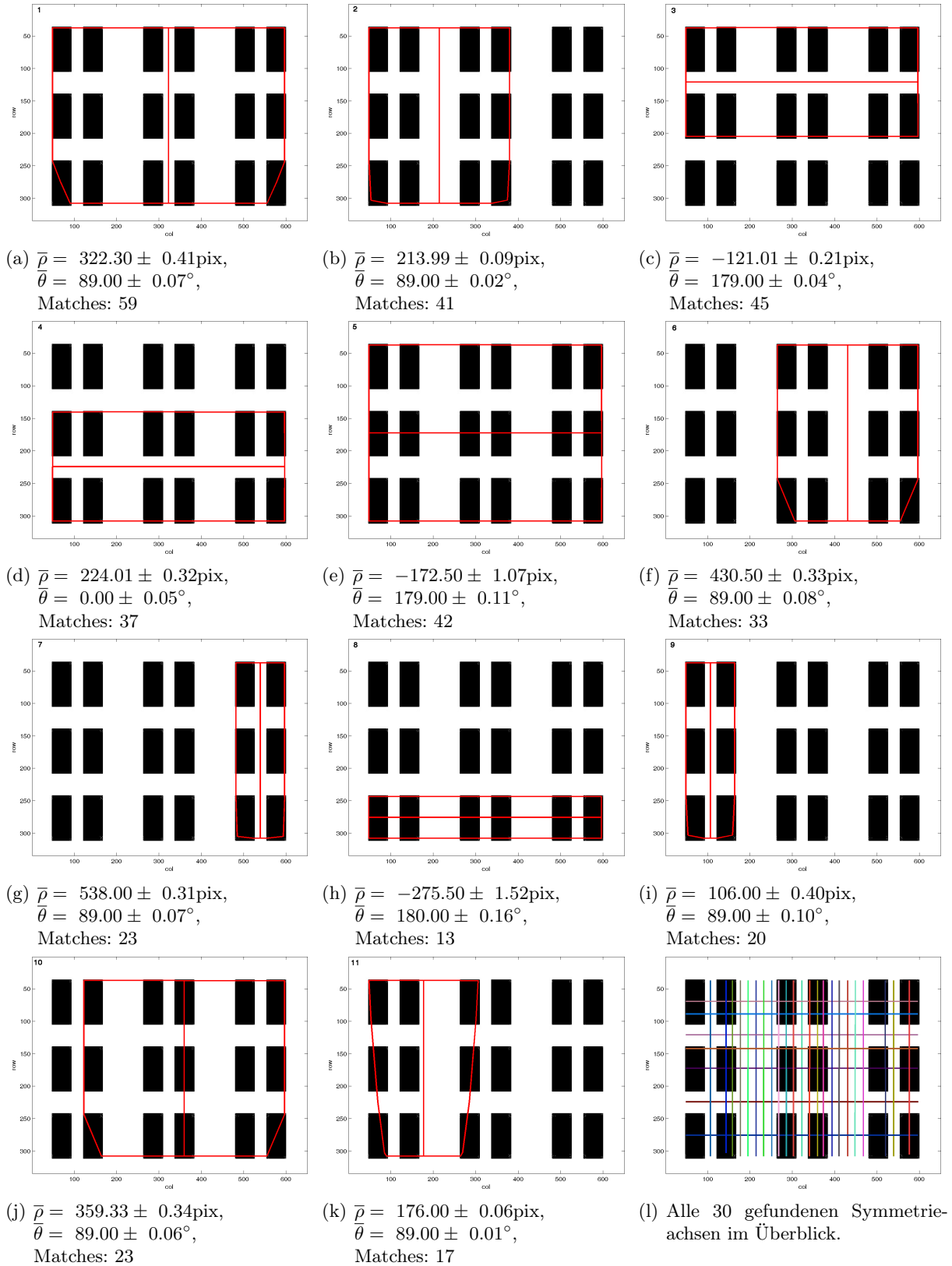


Abbildung 3.5: Ergebnisse der Symmetrie-Detektion für ein synthetisches Beispiel (Format: 650x334). Es wurden 368 Keypoints detektiert und darunter 2692 symmetrische Matches gefunden. Aus diesen wurden schließlich 30 symmetrische Konstellationen separiert. Die Abb. (a) bis (k) zeigen der Reihenfolge nach die ersten detektierten Symmetrieachsen mit den Bounding Boxes der beteiligten Features. Abb. (l) zeigt alle 30 gefundenen Achsen im Überblick.

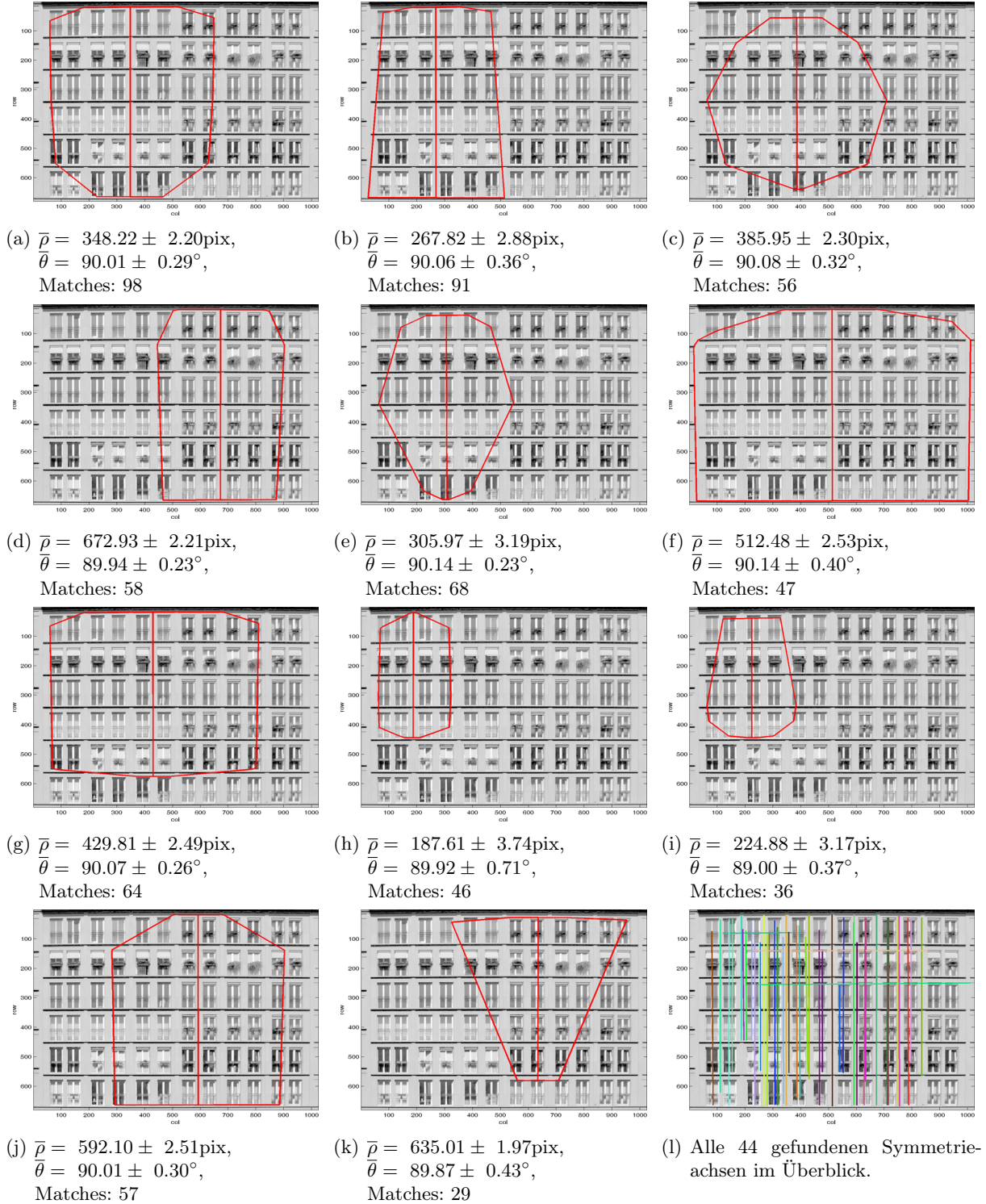
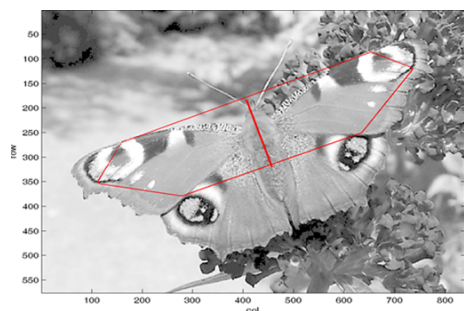
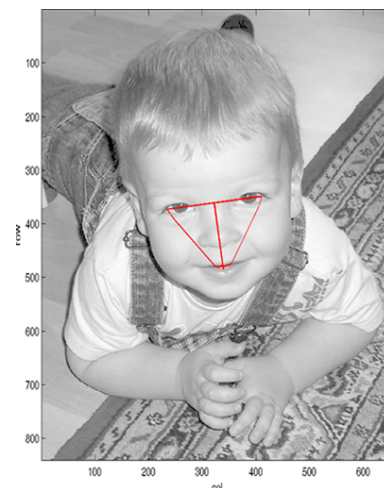


Abbildung 3.6: Ergebnisse der Symmetrie-Detektion für ein „echtes“ Beispiel (Format: 1024x680). Es wurden 6149 Keypoints detektiert und darunter 2587 symmetrische Matches gefunden. Aus diesen wurden schließlich 44 symmetrische Konstellationen separiert. Die Abb. (a) bis (k) zeigen der Reihenfolge nach die ersten detektierten Symmetrieachsen mit den Bounding Boxes der beteiligten Features. Abb. (l) zeigt alle gefundenen Achsen im Überblick.



(a) Schmetterling, Format: 840x576, 4667 Keypoints, 246 symmetrische Matches, 2 Symmetrieachsen detektiert mit 14 und 8 Unterstützern. Hier ist aus Anschaulichkeitsgründen nur die erste Symmetrieachse abgebildet. Die zweite detektierte Symmetrie bezieht sich auf ein kleines unscheinbares Element des rechten Fühlers und ist hier nicht von Interesse.



(b) Ein Kindergesicht, Format: 658x840, 4528 Keypoints, 195 symmetrische Matches, 1 Symmetrieachse mit 11 Unterstützern.

Abbildung 3.7: Zwei typische Beispiele für natürlich auftretenden Symmetrien.

Da sich die Untersuchungen dieser Arbeit aber verstärkt auf die wiederholten Gruppen in Fassaden konzentrierten, wurden an dieser Stelle keine weiterführenden Überlegungen angestellt.

Es ist festzuhalten, dass in der Regel viel mehr Symmetrien detektiert werden, als zur Beschreibung der Fassade nötig sind, die aber (für weiterführende Arbeiten) dazu dienen können, die Struktur der Fassade genau zu beschreiben, ggf. in Verbindung mit den Ergebnissen der Detektion wiederholter Gruppen.

Die beiden Darstellungen der Abb. 3.7 sollen zeigen, dass die Implementation auch auf anderen Objekten als Gebäuden funktioniert. Sowohl für den Schmetterling (a) als auch für das Gesicht (b) wurde die Hauptsymmetrie korrekt detektiert. Störungen im Hintergrund wie Pflanzen oder das Teppichmuster blieben unberücksichtigt. Beim Schmetterling wurde noch eine kleine Nebensymmetrie im rechten Fühler erkannt, die in der Abbildung nicht mit dargestellt ist.

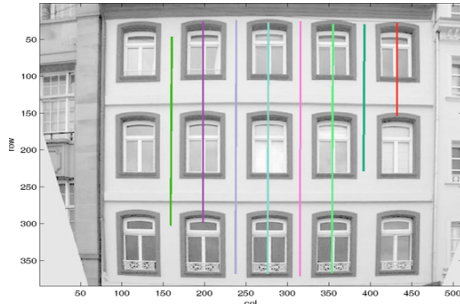
In den Abb. 3.8(a) bis (g) sind schließlich einige der sehr guten Ergebnisse der zahlreichen Experimente auf Fassadenbildern dargestellt. Hier wird jeweils nur der Überblick über alle detektierten Symmetrien in Form der Symmetrieachsen visualisiert.

In Bsp. (a) wurden bis auf eine alle wichtigen Symmetrien, die durch die Fenster und die zwischen den Fenstern, erkannt und keine „falschen“ Achsen erzeugt. Das gleiche gilt für Bsp. (b), mit dem Unterschied, dass das Bild hier nicht so vollständig erfasst wurde. Für eine minimale Beschreibung der Fassade reichen aber zwei der vier erkannten Symmetrien aus.

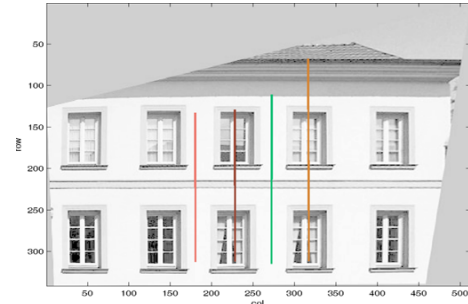
Es sind keine qualitativen Unterschiede in der Symmetrie-Detektion zwischen Aufnahmen alter und neuer, industriell gefertigter Gebäude zu beobachten. Ebenso erfreulich sind die gleich bleibend guten Ergebnisse bei kleiner werdendem Abbildungsmaßstab. Mit der damit verbundenen zunehmenden Komplexität der Fassadenstruktur nimmt aber auch die Zahl der Detektionen und der möglichen Fehldetektionen zu. Hier ist wieder ein Verfahren gefragt, aus der Vielzahl der erkannten Symmetrien, die relevanten zu selektieren.

Für die Beispiele der Abb. 3.9 bis 3.13 sollen die Ergebnisse noch einmal in den einzelnen Stufen gezeigt werden. Es sind jeweils die ersten 5 Einzelergebnisse und schließlich ein Überblick über alle in diesem Bild detektierten Symmetrien visualisiert.

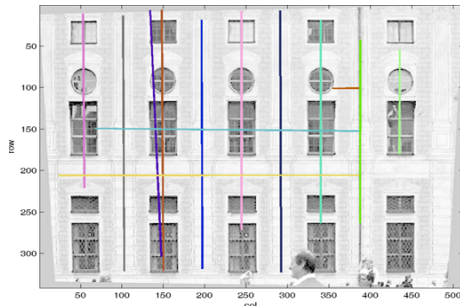




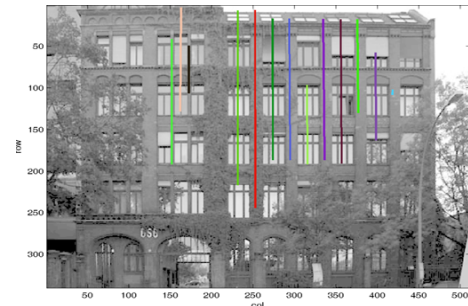
(a) 928 Keypoints, 568 symmetrische Matches, 8 symmetrische Konstellationen.



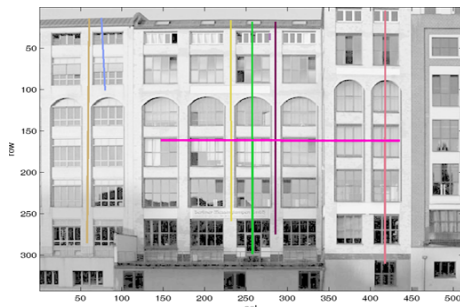
(b) 695 Keypoints, 218 symmetrische Matches, 4 symmetrische Konstellationen.



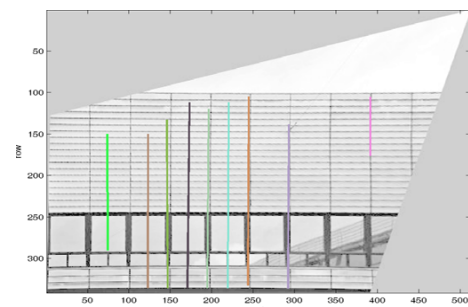
(c) 1744 Keypoints, 689 symmetrische Matches, 15 symmetrische Konstellationen.



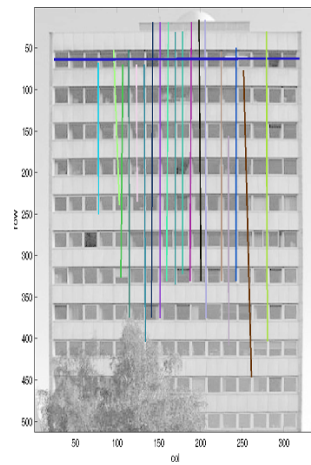
(d) 1715 Keypoints, 520 symmetrische Matches, 13 symmetrische Konstellationen.



(e) 1590 Keypoints, 368 symmetrische Matches, 8 symmetrische Konstellationen.



(f) 683 Keypoints, 455 symmetrische Matches, 9 symmetrische Konstellationen.



(g) 1653 Keypoints, 1323 symmetrische Matches, 20 symmetrische Konstellationen.

Abbildung 3.8: Beispiele für nahezu optimale Ergebnisse.



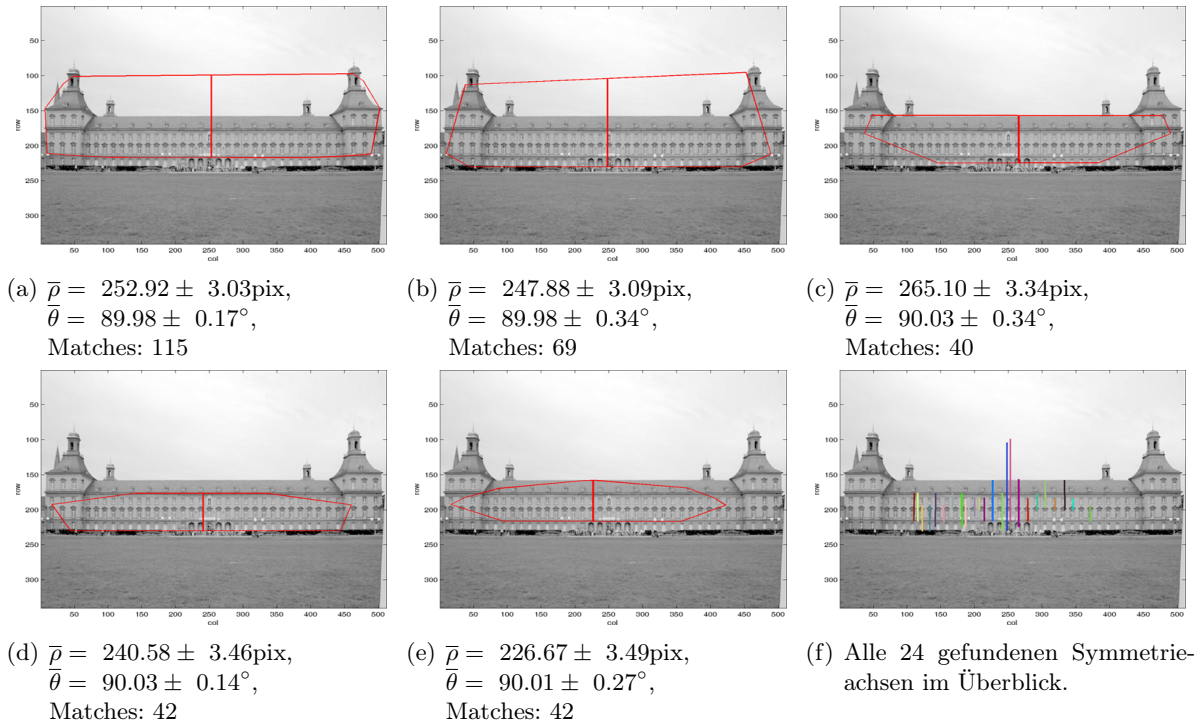


Abbildung 3.9: 733 Keypoints, 675 symmetrische Matches, 24 symmetrische Konstellationen.

Abb. 3.9 soll zeigen, dass auch bei einem so kleinen Abbildungsmaßstab plausible Ergebnisse zu erwarten sind.

Die Beispiele der Abb. 3.10 bis 3.13 sollen zum einen weitere gute Ergebnisse des Verfahrens zeigen, aber zum anderen auch verdeutlichen, dass eine Beschränkung der Suchregion von Vorteil wäre. Die Hauptsymmetrien werden in allen Fällen gut erkannt, es kommt aber verstärkt zu Fehldetektionen durch Störungen der Fassadenstruktur. Features in der Dachfläche, dem Sockelbereich oder sogar der Straße führen zu solchen Fehldetektionen. Oder aber es werden Symmetrien in solchen Regionen, z.B. dem Dach, ganz korrekt und getrennt vom Rest der Fassade detektiert. Dann kommt es aber zu einer Mischung der Beschreibung zweier verschiedener Bildregionen. Für das Beispiel aus Abb. 3.14 wurde die regelmäßig symmetrische Region aus Abb. 3.13 ausgeschnitten und erneut untersucht. Das Resultat zeigt ein nahezu optimales Ergebnis. Dass im linken Bereich der Fensterreihen keine Symmetrien detektiert wurden, liegt vermutlich an der Beschaffenheit der Fenster, der Position der Rollläden und dem Schatten auf den Gardinen.

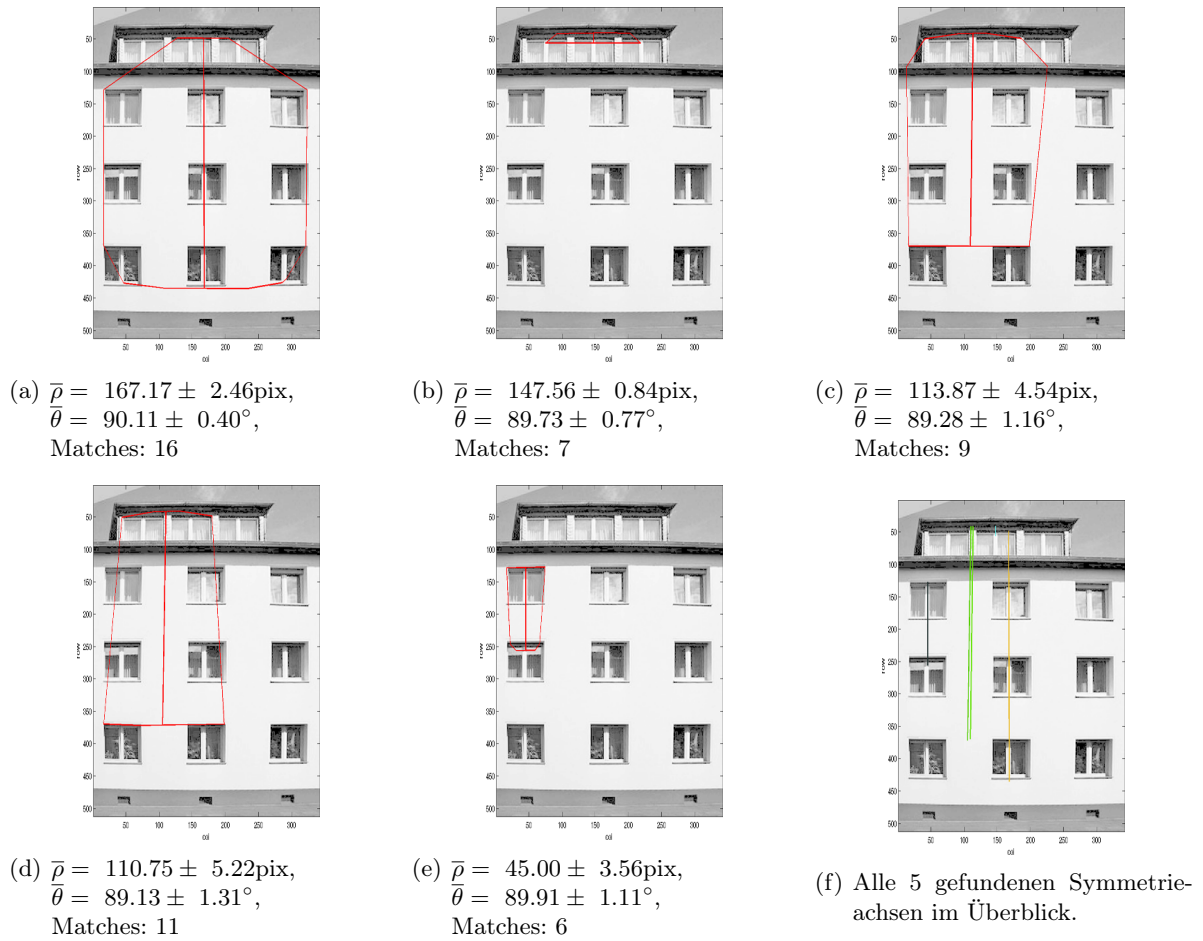


Abbildung 3.10: 655 Keypoints, 229 symmetrische Matches, 5 symmetrische Konstellationen.

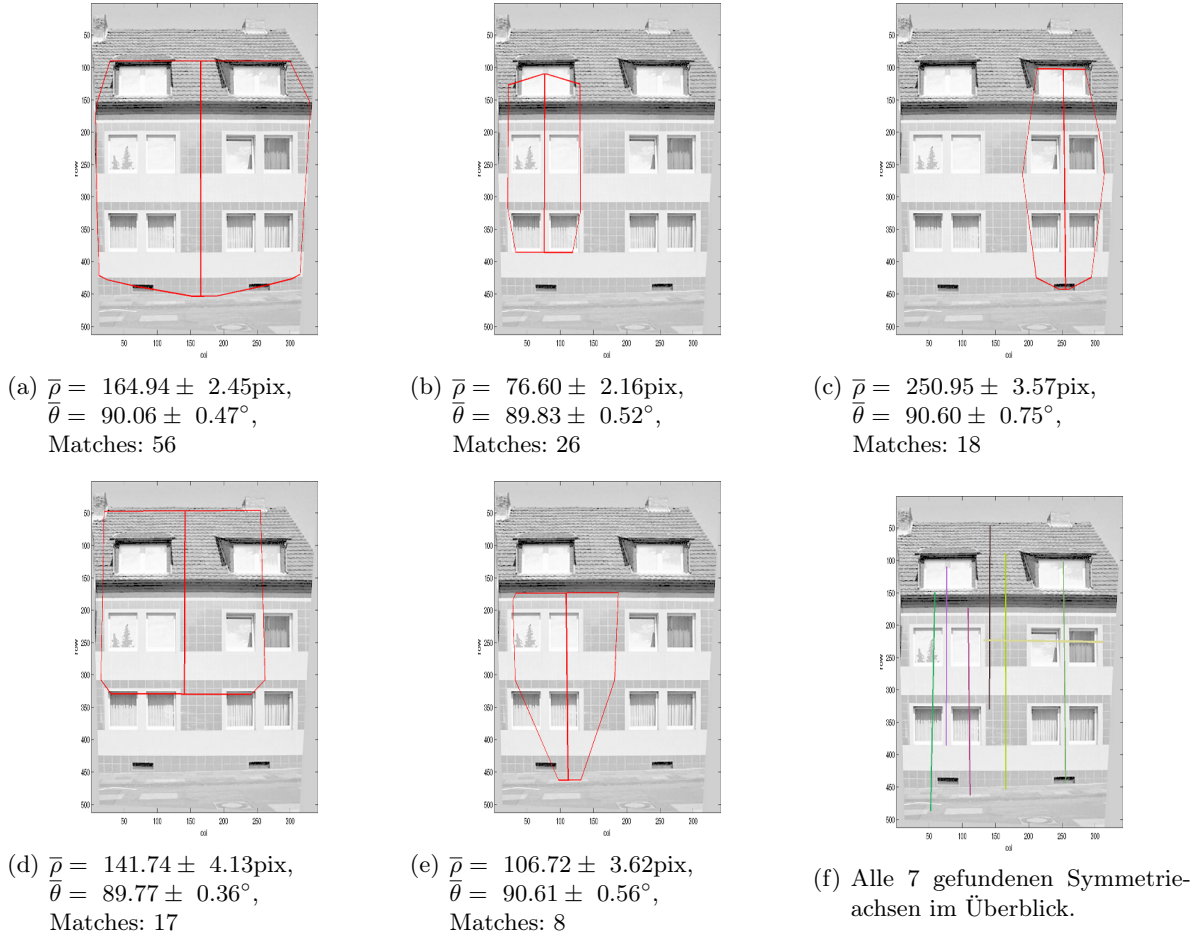


Abbildung 3.11: 1505 Keypoints, 434 symmetrische Matches, 7 symmetrische Konstellationen.

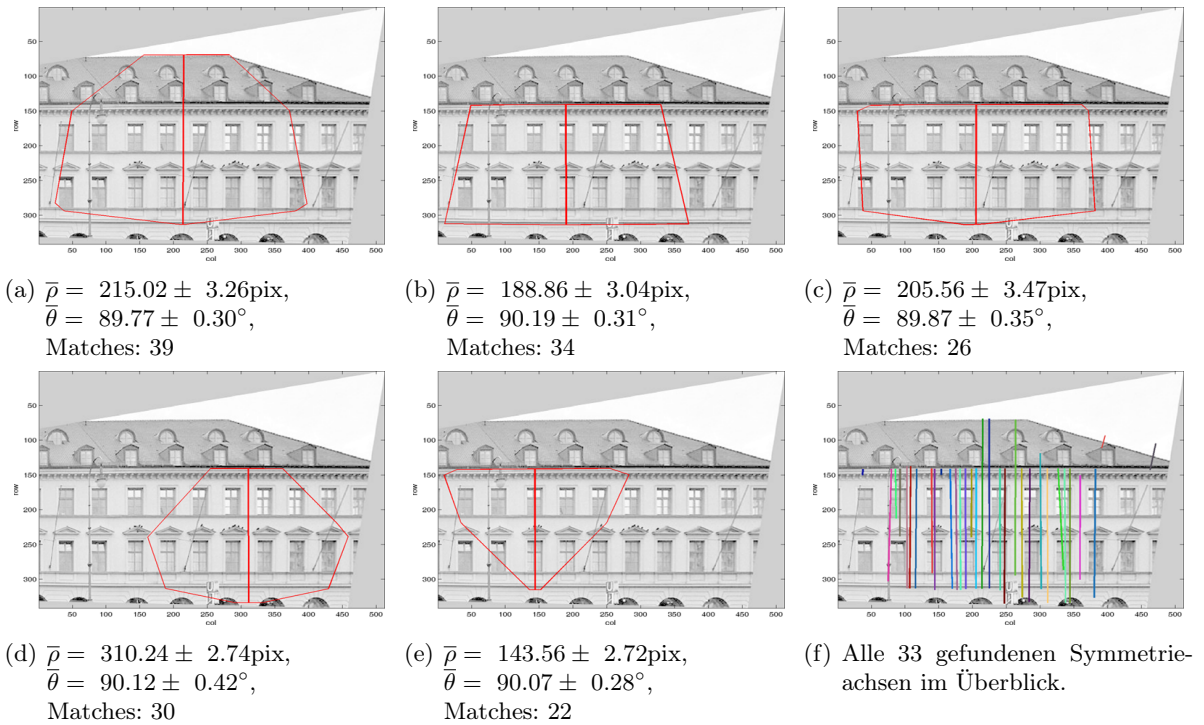


Abbildung 3.12: 1743 Keypoints, 710 symmetrische Matches, 33 symmetrische Konstellationen.

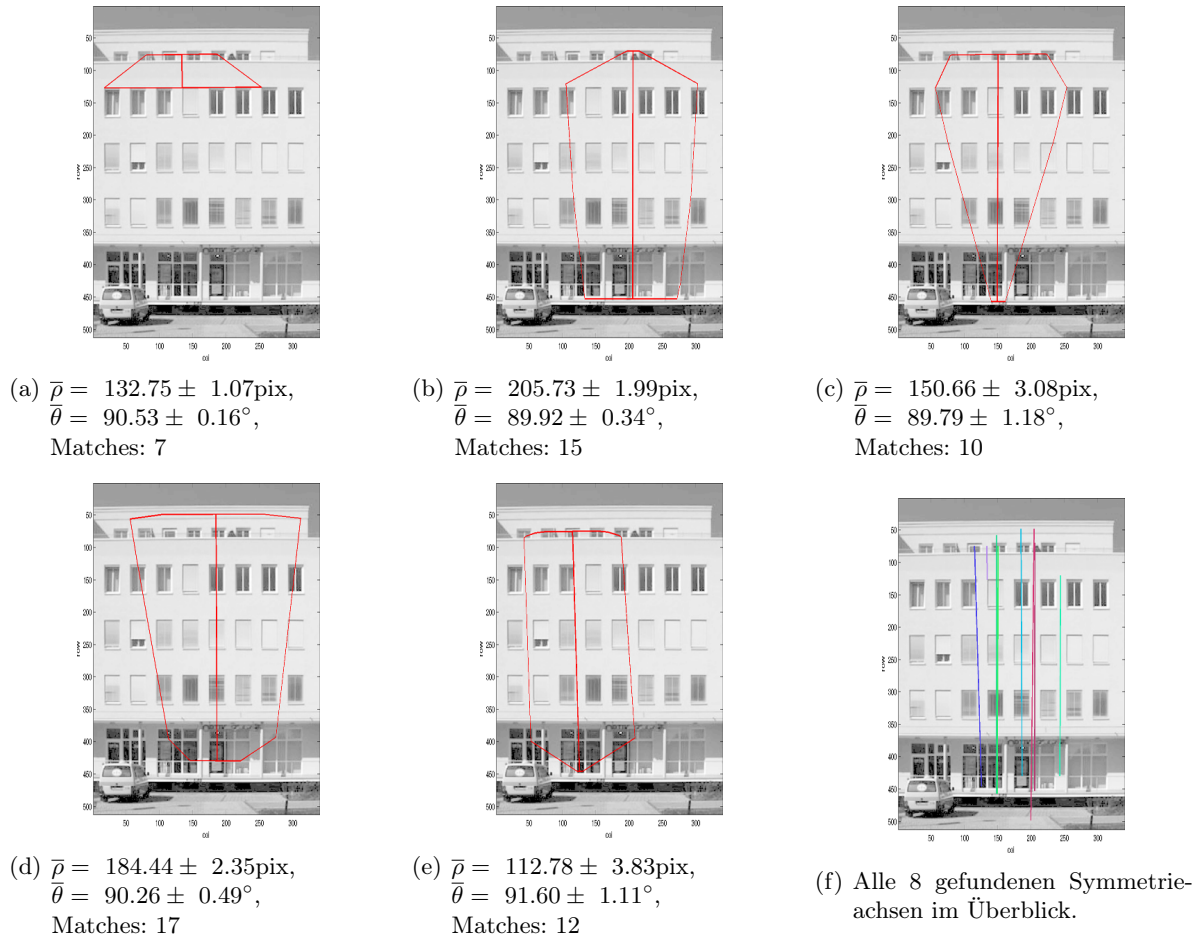


Abbildung 3.13: 1352 Keypoints, 462 symmetrische Matches, 8 symmetrische Konstellationen.

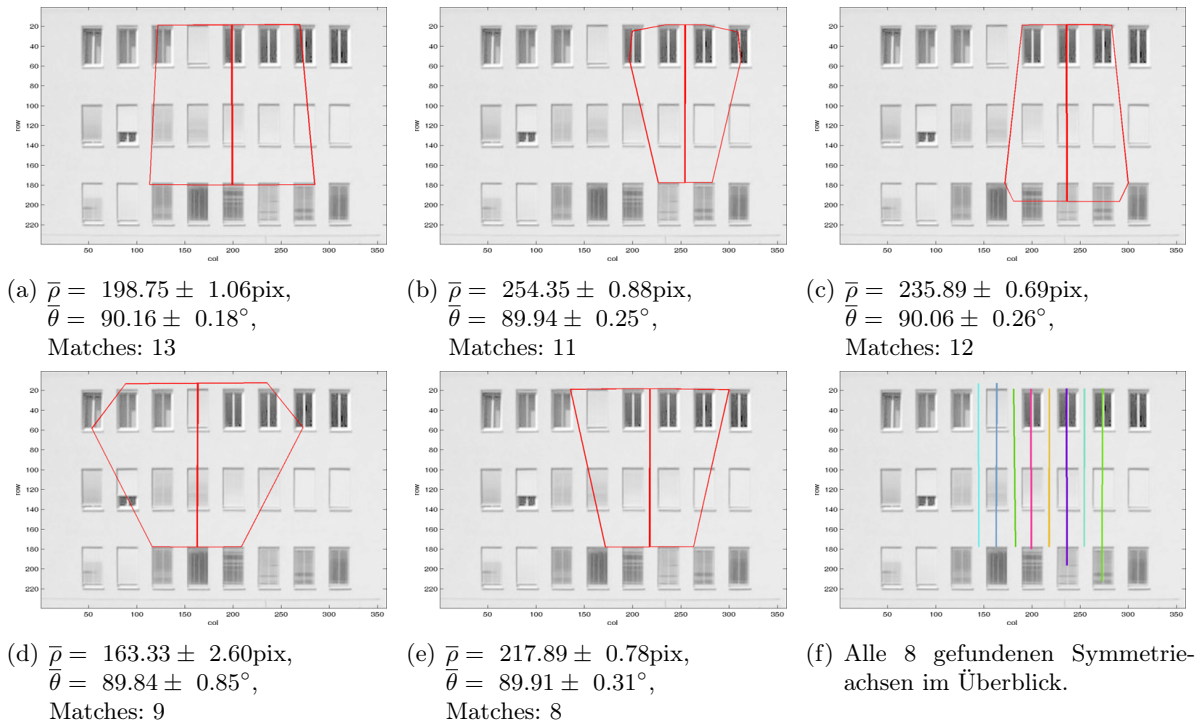


Abbildung 3.14: Format: 360x240, 355 Keypoints, 245 symmetrische Matches, 8 symmetrische Konstellationen.

### 3.3 Detektion von wiederholten Strukturen

Wie auch bei der Symmetrie-Detektion besteht der erste Schritt auf dem Weg zur Detektion wiederholter Gruppen in der Detektion der Features im Bild.

```
[image, desc, locs] = sift(pict);
```

#### 3.3.1 Implementation

Für die Umsetzung des in Abs. 2.4 beschriebenen Verfahrens wurde die Matlab-Funktion `detectGroups` geschrieben. Der Funktionsaufruf lautet:

```
Groups = detectGroups(image, desc, locs);
```

Als Eingabe-Parameter werden die Ergebnisse des Feature-Detektors übergeben. Der Rückgabe-Parameter `Groups` ist ein Array von Structs mit den Parametern jeder gefundenen wiederholten Gruppe:

- `.drow` und `.dcol` : Die Translation zwischen den gefundenen Gruppen.
- `.s_drow` und `.s_dcol` : Die entsprechenden Standardabweichungen.
- `.points_a` und `.points_b` : Die Bildkoordinaten der zu den beiden Gruppen gehörenden Features, jeweils als  $[1 \times 2]$ -Vektor.

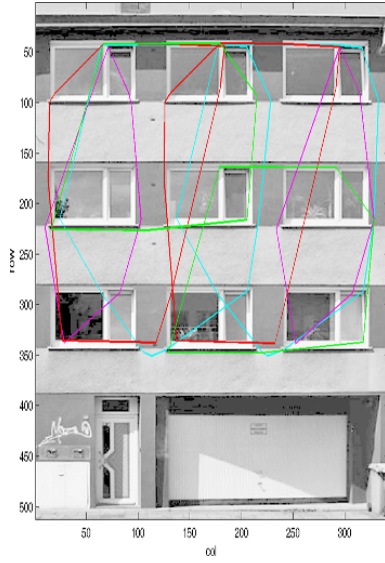
Die Implementation unterscheidet sich kaum von der des Verfahrens zur Detektion symmetrischer Strukturen. Der wichtigste Unterschied besteht in der Behandlung der detektierten Features. Während für die Symmetrie-Detektion der Feature-Satz gespiegelt und dann mit sich selbst gematcht wurde, werden die Features hier unverändert mit sich selbst gematcht, um Bildmerkmale zu identifizieren, die wiederholt im Bild auftauchen. Das Vorgehen ist also äquivalent zu dem in Abs. 3.2 bzw. Abs. 3.2.3, mit dem Unterschied, dass die Spiegelung der Deskriptoren entfällt.

Die Berechnung des Repetitionsgrades erfolgt analog zu der des Symmetriegrades, entsprechend Gl. (2.11) mit der erläuterten Abwandlung des Winkel-Gewichtes zu seiner Form in Gl. (2.10).

Um den Quellcode aus `detectSym.m` so wenig wie möglich abwandeln zu müssen, wurden die Translationen zwischen den gematchten Features auch hier in Form von  $[\rho, \theta]$ -Polarkoordinaten repräsentiert. Dabei steht  $\rho$  für den Betrag der Translation und  $\theta$  für ihre Richtung. So konnte die Ermittlung dominanter Symmetrien bzw. hier Translationen unverändert übernommen werden. Nach Ermittlung aller zu einer Translation gehörenden Features wurde aus diesen schließlich eine mittlere Translation in Form eines kartesischen freien Vektors  $[\bar{d}_{row}, \bar{d}_{col}]$  bestimmt.

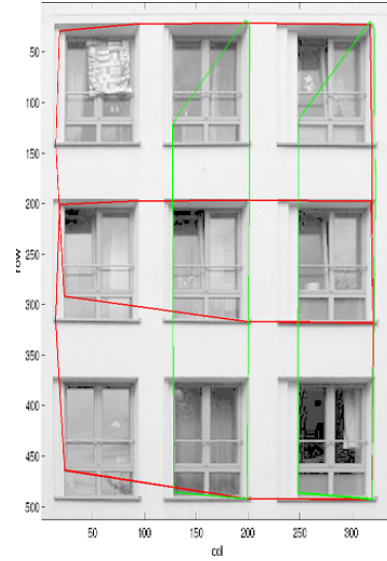
#### 3.3.2 Ergebnisse

Für das bekannte synthetische Beispiel einer typischen Fensterfront wurden die Ergebnisse der Detektion wiederholter Gruppen bereits in Abb. 2.9 auf Seite 22 gezeigt. Abb. 3.15 zeigt die Ergebnisse der Matlab-Funktion `detectGroups` anhand einiger ausgewählter „echter“ Testbilder. Wie schon bei der Symmetrie-Detektion variiert die Anzahl der beteiligten Matches stark und ist nicht direkt mit der Qualität der Repetition zu verbinden. Für Abb. 3.15(c) bspw. wird die beste Gruppierung von nur 10 und die zweitbeste von 19 Matches unterstützt. Hier spiegelt sich erneut die unterschiedliche Gewichtung der Feature-Paare wieder.



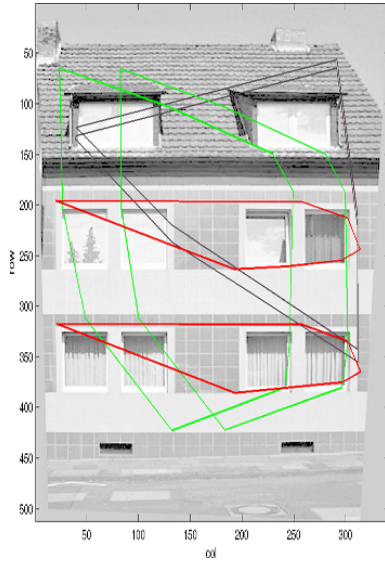
(a) 724 Keypoints, 504 Matches

1.  $\bar{d}_{row} = -0.96 \pm 0.95\text{pix}$ ,  
 $\bar{d}_{col} = 112.25 \pm 1.82\text{pix}$ , Matches: 54
2.  $\bar{d}_{row} = -0.38 \pm 0.72\text{pix}$ ,  
 $\bar{d}_{col} = 224.75 \pm 1.53\text{pix}$ , Matches: 17
3.  $\bar{d}_{row} = -0.92 \pm 1.13\text{pix}$ ,  
 $\bar{d}_{col} = 112.25 \pm 1.92\text{pix}$ , Matches: 18
4.  $\bar{d}_{row} = -121.20 \pm 1.22\text{pix}$ ,  
 $\bar{d}_{col} = 112.81 \pm 0.60\text{pix}$ , Matches: 24



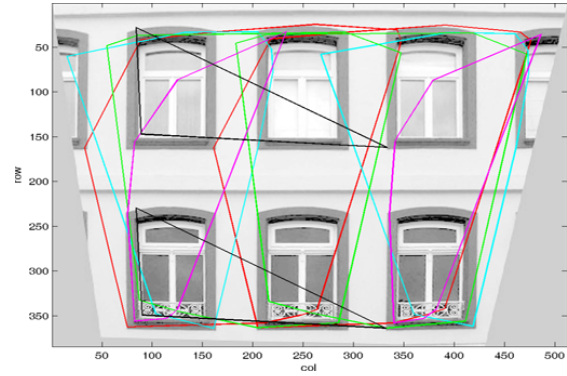
(b) 1045 Keypoints, 565 Matches

1.  $\bar{d}_{row} = -172.73 \pm 2.12\text{pix}$ ,  
 $\bar{d}_{col} = 0.35 \pm 0.82\text{pix}$ , Matches: 52
2.  $\bar{d}_{row} = -0.37 \pm 1.05\text{pix}$ ,  
 $\bar{d}_{col} = 119.99 \pm 1.21\text{pix}$ , Matches: 45



(c) 1505 Keypoints, 499 Matches

1.  $\bar{d}_{row} = 11.44 \pm 2.98\text{pix}$ ,  
 $\bar{d}_{col} = 0.05 \pm 0.14\text{pix}$ , Matches: 10
2.  $\bar{d}_{row} = -0.29 \pm 0.41\text{pix}$ ,  
 $\bar{d}_{col} = 56.37 \pm 4.11\text{pix}$ , Matches: 19
3.  $\bar{d}_{row} = -121.88 \pm 1.27\text{pix}$ ,  
 $\bar{d}_{col} = 0.19 \pm 0.89\text{pix}$ , Matches: 11



(d) 613 Keypoints, 437 Matches

1.  $\bar{d}_{row} = -0.29 \pm 0.69\text{pix}$ ,  
 $\bar{d}_{col} = 128.41 \pm 2.06\text{pix}$ , Matches: 53
2.  $\bar{d}_{row} = 0.46 \pm 1.17\text{pix}$ ,  
 $\bar{d}_{col} = 257.67 \pm 2.39\text{pix}$ , Matches: 21
3.  $\bar{d}_{row} = 0.49 \pm 1.11\text{pix}$ ,  
 $\bar{d}_{col} = 126.60 \pm 1.68\text{pix}$ , Matches: 27
4.  $\bar{d}_{row} = -0.55 \pm 0.93\text{pix}$ ,  
 $\bar{d}_{col} = 256.07 \pm 2.08\text{pix}$ , Matches: 27
5.  $\bar{d}_{row} = 202.0 \pm 0.429\text{pix}$ ,  
 $\bar{d}_{col} = 0.11 \pm 0.94\text{pix}$ , Matches: 4

Abbildung 3.15: Einige Ergebnisse der Detektion wiederholter Strukturen durch `detectGroups`. Dargestellt sind die Bounding Boxes der beteiligten Features. Zusammengehörige Gruppen sind durch gleiche Farben gekennzeichnet. In den Bildunterschriften sind die wichtigsten Ergebnisse des Detektions-Prozesses zusammengefasst und die Translationsparameter der detektierten Gruppen aufgelistet.



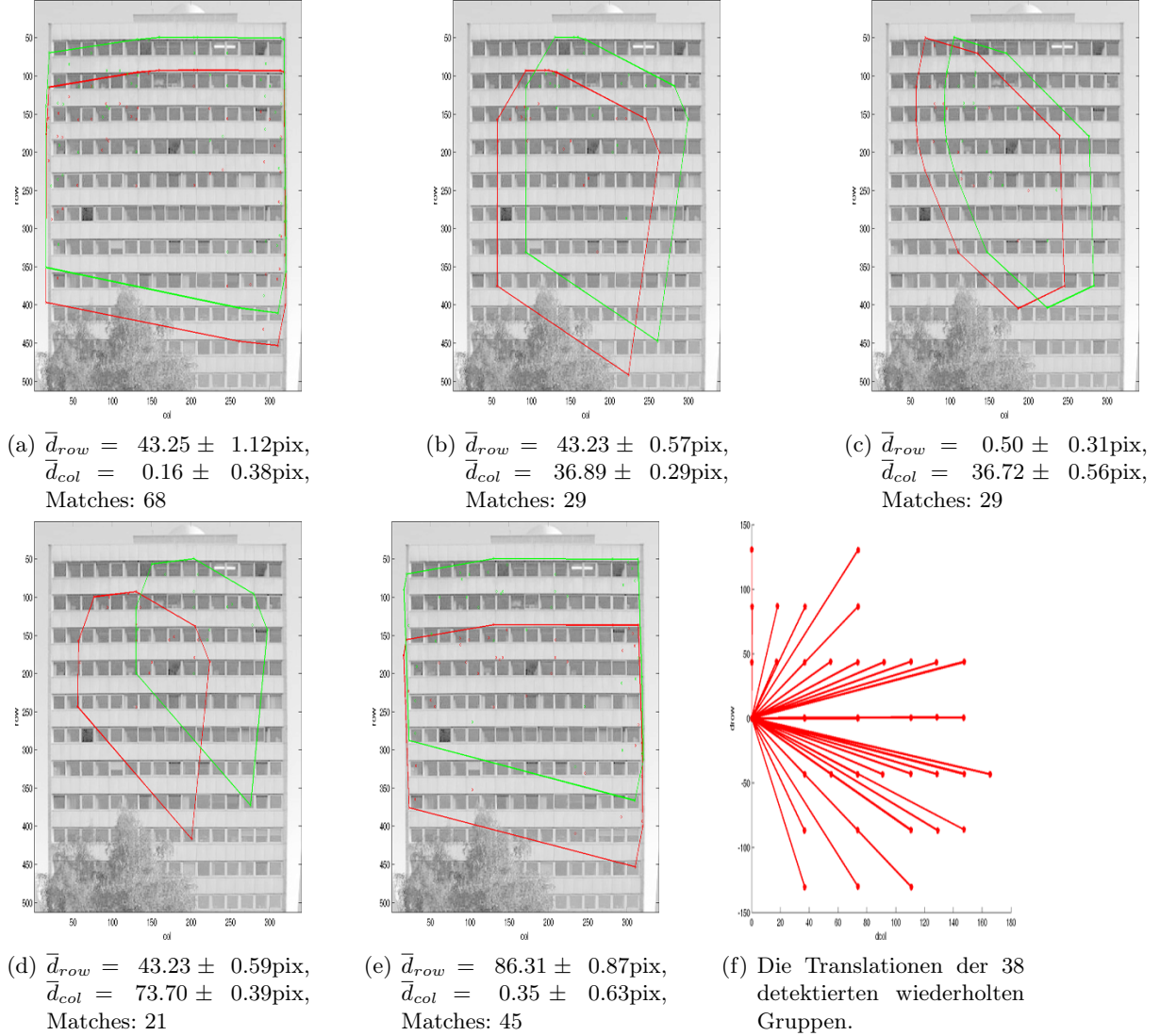


Abbildung 3.16: Die Einzelergebnisse eines ausgewählten Beispiels. Dargestellt sind der Reihenfolge nach die ersten fünf detektierten wiederholten Strukturen in Form der Bounding Boxes der Features der beiden Gruppen. In den Bildunterschriften sind jeweils die Parameter der Translation angegeben sowie die Anzahl der gematchten Punktpaare. Es wurden 1653 Keypoints detektiert, darunter 1474 Feature-Paare gematcht und 38 wiederholte Strukturen gefunden. Auf die Darstellung aller 38 gefundenen wiederholten Gruppen in einer Abbildung wird hier verzichtet. Statt dessen zeigt Abb. (f) alle Translationsvektoren in einer Grafik. In dieser Darstellung ist sehr gut zu erkennen, dass die Endpunkte der Translationsvektoren ein Gitter aufspannen. (In dieser Abbildung ist zu beachten, dass die Achseinteilung nicht mit der der Abb. (a) bis (e) übereinstimmen!)

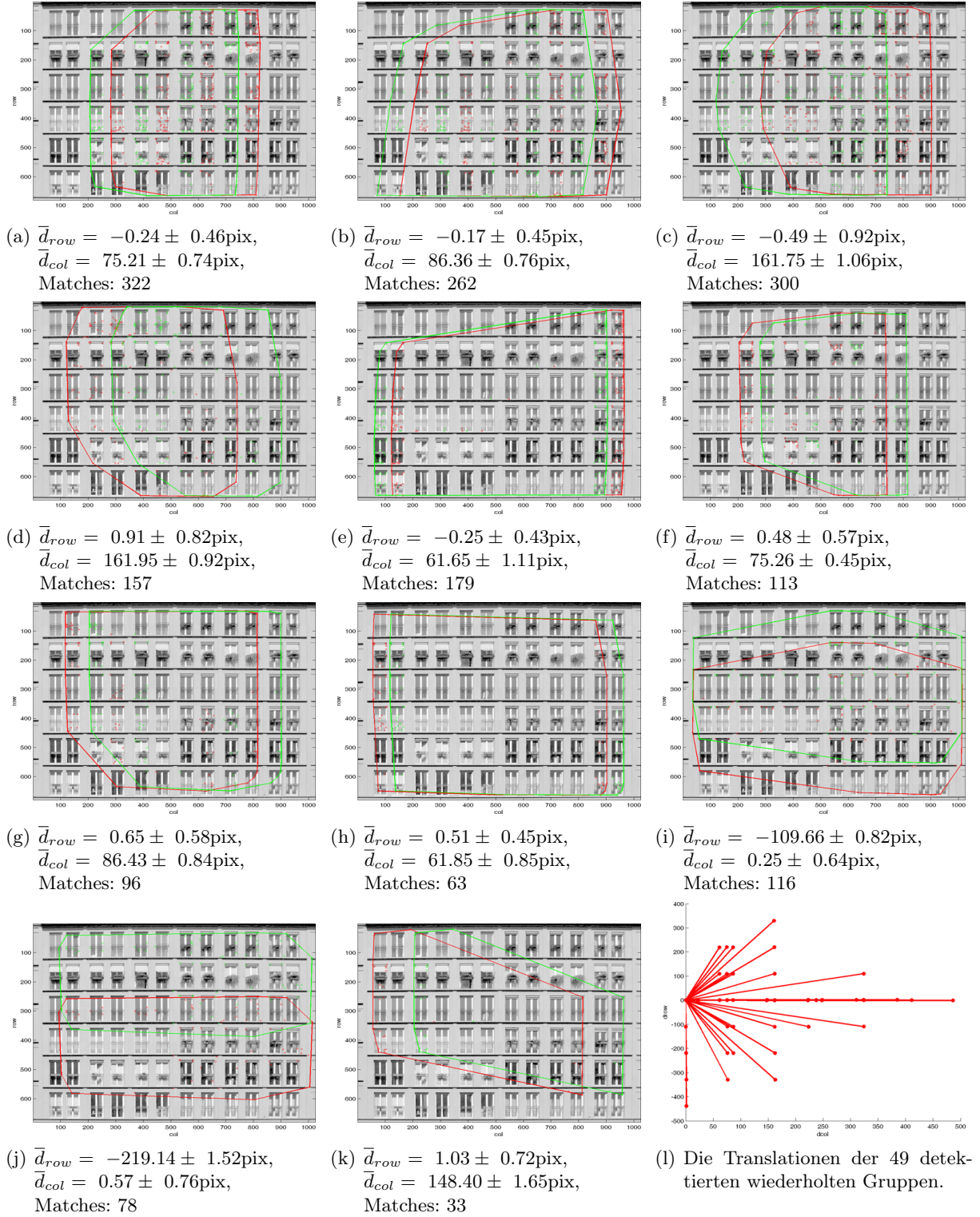


Abbildung 3.17: Einzelergebnisse eines weiteren komplexeren Beispiel (Format: 1024x680). Es wurden 6149 Keypoints detektiert und darunter 4194 Matches gefunden. Aus diesen wurden schließlich 49 wiederholte Gruppen separiert. Es werden der Reihenfolge nach die ersten 11 detektierten Gruppen mit den Bounding Boxes der beteiligten Features visualisiert. Abb. (l) zeigt alle 49 Translationsvektoren in einer Grafik. Auch hier ist das von den Translationsvektoren aufgespannte Gitter gut zu erkennen.



Es ist ersichtlich, dass sich die Features der Gruppen über das gesamte Bild erstrecken. Es kann nicht von der Ausdehnung der Gruppen auf die Größe oder Verteilung einzelner Bildelemente, insbesondere der Fenster, geschlossen werden. Und trotzdem kann „durch genaues Hinsehen“ anhand einzelner Eckpunkte der Bounding Boxes der Gruppen auf die Beziehung zwischen den Gruppen geschlossen werden. Es ist in der Regel zu erkennen, dass die Translation zwischen zwei Fenstern oder Fensterreihen gefunden wurde.

Anhand dieser „echten“ Testbilder wird schnell deutlich, dass die Ergebnis-Ausbeute hier sehr viel geringer ausfällt als bei den synthetischen Testdaten. Aufgrund der Beschaffenheit solcher Fassadenaufnahmen (Beleuchtung, Kontrast, Störungen) werden nicht unbedingt weniger Features detektiert, aufgrund der Komplexität der Fassade sogar eher mehr, aber der gesamte Feature-Satz ist natürlich weniger homogen als bei synthetischen Daten.

Die Abb. 3.16 und 3.17 zeigen die Ergebnisse der Detektion wiederholter Gruppen für zwei große Fensterfronten. Hier wurden 38 bzw. 49 wiederholte Gruppen detektiert. Die jeweils letzte Teilabbildung stellt die Translationen aller gefundenen Gruppen in einem Plot dar. In dieser Darstellungsform ist gut zu erkennen, dass die Translationsvektoren ein Gitter aufspannen, dass der Struktur der Fassade entspricht.

Betrachtet man die Visualisierungen der Einzelergebnisse, Abb. 3.16(a)-(e) bzw. 3.17(a)-(k), so fällt auf, dass die kleinsten Strukturen, die benachbarten Fenster und Fensterreihen, als erstes erkannt werden. Das erscheint insofern sinnvoll, als dass sich Gruppen mit der kleinsten gegenseitigen Verschiebung am weitesten über das Bild erstrecken können und somit die meisten Unterstützer finden werden.

## 3.4 Ermittlung von Mustern in wiederholten Strukturen

### 3.4.1 Implementation

Für die Umsetzung des in Abs. 2.5 beschriebenen Verfahrens wurden die Matlab-Funktionen `descGroups`<sup>5</sup> und `findPattern` geschrieben.

Der Funktionsaufruf zur Ermittlung der Basis aus den Translationen der detektierten wiederholten Gruppen lautet:

```
basis = descGroups(Groups);
```

Als Eingabe-Parameter werden die Ergebnisse der Detektion wiederholter Gruppen übergeben. Der Rückgabe-Parameter `basis` ist ein Struct zur Beschreibung der ermittelten Basiselemente in:

- `.row`: Zeilenrichtung

- `.col`: Spaltenrichtung

jeweils als Array von Structs für jedes Basiselement mit

- `.v`: Basiselement/Translation

- `.anz`: Anzahl der Elemente mit diesem Abstand

- `.row_con` und `.col_con`<sup>6</sup>: Nummern/Indizes der in Zusammenhang stehenden Basisvektoren  
0, falls mehrere Basiselemente gefunden wurden, die aber nicht im Zusammenhang stehen

Diese Funktion ist dafür zuständig die Daten für `findPattern` aufzubereiten. Die Translationen werden in Zeilen- und Spaltenrichtung getrennt und die Beträge der jeweiligen Verschiebungen werden als ein Vektor `d` an `findPattern` übergeben.

Der Funktionsaufruf für die Bestimmung einer 1D-Basis aus einer Reihe reeller Zahlen `d` lautet:

```
[basis, connected] = findPattern(d);
```

Die Rückgabeparameter sind

- `basis`: Array von Structs zur Beschreibung der 1D-Basis mit

- `.v`: Basiselement/Translation

- `.anz`: Anzahl der Elemente mit diesem Abstand

- `connected`: Nummern/Indizes der in Zusammenhang stehenden Basisvektoren  
0, falls mehrere Basiselemente gefunden wurden, die aber nicht im Zusammenhang stehen

Die Details zur Umsetzung der vorgestellten Konzepte sollen in den folgenden Abschnitten näher erläutert werden.

---

<sup>5</sup>`descGroups` steht für *describe Groups*

<sup>6</sup>`con` steht für *connected Elements*

### 3.4.1.1 Bemerkungen

Das in Abs.2.5.3 (S.25) beschriebene Verfahren ist durch viele Versuche, mit vielen verschiedenen künstlichen wie echten Testbildern, durch „scharfes Hinsehen“ und Ausprobieren entstanden.

Die erste Idee bestand tatsächlich in der Bildung der Differenzen der Translationen in einer Richtung, mit dem Gedanken, dass die kleinsten Basiselemente am häufigsten vorkommen. Da, wie sich in den zahlreichen Versuchen gezeigt hat, dem aber leider nicht so ist, kam der Gedanke hinzu, die Ergebnisse dieser ersten Selektion nach ihrem Auftreten in der Detektion wiederholter Gruppen zu sortieren. Dann stellt sich aber die Frage, ob der Umweg über die Häufigkeitsverteilung der Differenzen zwischen den detektierten Translationen überhaupt nötig ist, oder ob nicht direkt mit den ersten  $x$  detektierten Translationen gearbeitet werden kann.

Die gesuchten Basiselemente erscheinen in den Ergebnissen der Detektion in der Regel unter den ersten Elementen, aber eben auch mit „Ausreißern“. Da bei künstlichen Bildern nicht immer mit vollständig akkuraten Detektionen zu rechnen ist, soll das Verfahren also in dieser Kombination bestehen bleiben. Wie Abs. 3.4.2 zeigen wird, hat es sich so auch bewährt.

### 3.4.1.2 Basiselemente suchen

Es werden alle möglichen Differenzen zwischen den Elementen des Vektors  $\mathbf{d}$  berechnet und in `diff` zusammen mit den Indizes der beiden beteiligten Elemente aus  $\mathbf{d}$ , im Sinne einer positiven Verschiebung, gespeichert. Aus dem Histogramm dieser Differenzen werden, wie in Abs. 2.5.3.1 (S. 25) beschrieben, die potentiellen Kandidaten für Basiselemente ermittelt. Dabei wird jeweils das Mittel aller Werte in einer Umgebung von 1.5 Pixel um den gerade betrachteten Peak  $i$  als Translation bzw. Basiselement  $v_i$  übernommen. Alle  $v_i$  der potentiellen Kandidaten sind auch in  $\mathbf{d}$  vertreten und sind nicht Vielfache voneinander.

Für alle getesteten Beispiele war dabei die Auswahl der besten 10 Peaks im Histogramm für die Zusammenstellung der potentiellen Kandidaten ausreichend.

Für diese Translationen  $v_i$  werden nun die Gruppen von Punkten in der Reihe  $\mathbf{d}$  gesucht, die genau den Abstand  $v_i$  haben. Diese Gruppensuche wurde wie folgt umgesetzt:

```
. für alle v
```

```
{
```

```
. Initialisierung der Gruppen-Matrix G
```

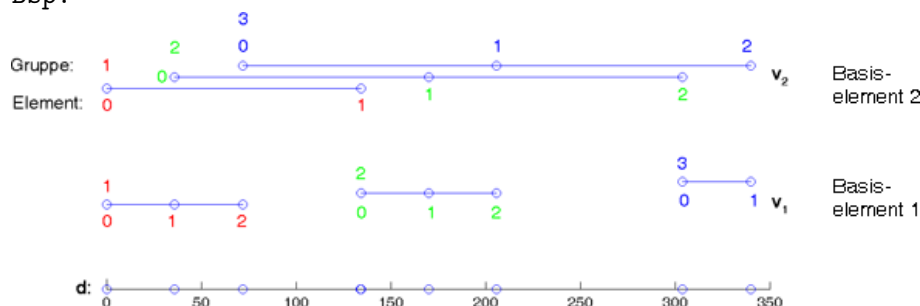
```
G = -100 * ones(n,2), n - Anzahl der Punkte in d
```

```
Jede Zeile verweist auf einen Eintrag in d
```

```
G(:,1) - Element in der Gruppe
```

```
G(:,2) - Gruppe
```

```
Bsp:
```



```
. Initialisierung des Gruppenzählers: gruppe_max = 1
```

```
. Suche alle Paare in diff, die den Abstand v haben
```

```
→ paare[Punkt1, Punkt2]
```

```
. für alle gefundenen Paare i
```

```
{
```

```

. Initialisierung der Hilfsgrößen
  . a: Element Punkt1  $\rightarrow a = G(\text{paare}(i,1),1)$ 
  . gr_a: Gruppe Punkt1  $\rightarrow \text{gr\_a} = G(\text{paare}(i,1),2)$ 
  . b: Element Punkt2  $\rightarrow b = G(\text{paare}(i,2),1)$ 
  . gr_b: Gruppe Punkt2  $\rightarrow \text{gr\_b} = G(\text{paare}(i,2),2)$ 
  . Idee: da die beiden Punkte immer den Abstand  $v$  haben, müssen a
    und b um genau 1 auseinander liegen und zur gleichen Gruppe
    gehören
    z.B.  $a=1 \Rightarrow b=2$  oder  $b=3 \Rightarrow a=1$ 
? falls der Abstand zwischen a und b schon 1 ist
   $\text{abs}(b - a) == 1 \ \& \ \text{gr\_a} == \text{gr\_b}$ 
 $\Rightarrow$  fertig mit diesem Paar
? a und b haben noch den Startwert aus der Initialisierung  $\rightarrow$  beide
  Punkte unbekannt, bisher nicht zugeordnet
   $a == -100 \ \& \ b == -100$ 
    . da nichts weiter bekannt,  $a=0$ ,  $b=1$ 
    . neue Gruppe eröffnen
      Gruppenzähler inkrementieren:  $\text{gruppe\_max} = \text{gruppe\_max}+1$ 
      Elemente der neuen Gruppe zuordnen:  $\text{gruppe} = \text{gruppe\_max}$ ;
? b schon bekannt, a noch unbekannt
   $b > a \ \& \ a == -100$ 
    .  $a = b - 1$ 
    . Gruppe für a wird von b übernommen:  $\text{gruppe} = \text{gr\_b}$ 
? a schon bekannt, b noch unbekannt
   $b < a \ \& \ b == -100$ 
    .  $b = a + 1$ 
    . Gruppe für b wird von a übernommen:  $\text{gruppe} = \text{gr\_a}$ 
? beide schon bekannt, haben aber nicht die gleiche Gruppe
   $\text{gr\_a} \neq -100 \ \&\& \ \text{gr\_b} \neq -100 \ \&\& \ \text{gr\_a} \neq \text{gr\_b} \Rightarrow$  Kollision zweier als
  getrennt eingestufte Gruppen, deren Elemente aber tatsächlich zu
  einer Gruppe gehören
    ? Gruppe a < Gruppe b
       $\Rightarrow$  alle Elemente mit Zuordnung zu Gruppe b werden auf Gruppe a
      umgeschrieben und um den Betrag korrigiert, um den b verschoben
      werden muss, so dass  $b-a = 1$ 
       $\text{gruppe} = \text{gr\_a}$ 
    ? Gruppe b < Gruppe a
      analog, nur andersherum
       $\text{gruppe} = \text{gr\_b}$ 
    . Da eine Gruppe geändert wurde, müssen alle nachfolgenden Gruppen
      aufgerückt werden.
      Gruppenzähler dekrementieren:  $\text{gruppe\_max} = \text{gruppe\_max}-1$ ;
  . Zuordnungen für a,b und gr_a, gr_b in Gruppen-Matrix schreiben
     $G(\text{paare}(i,1),:) = [a, \text{gruppe}]$ 
     $G(\text{paare}(i,2),:) = [b, \text{gruppe}]$ 
} ( für alle gefundenen Paare  $i$  )

. Verschiebung der Elementen-Indizes, sodass alle Gruppen mit dem Element
  0 beginnen

```

```

. Die Anzahl der Elemente der ersten Gruppe wird (vorläufig) als Anzahl
  der Elemente dieses Basiselements übernommen. (in der Regel richtig
  aber eben nicht immer)
  → basis(i).v und basis(i).anz

} ( für alle v )

```

Auf diese Art und Weise wird die erste Auswahl an Basiselementen und ihren möglichen Gruppierungen in **d** zusammengestellt. Im Folgenden werden sie nach ihrem Auftreten in **d** sortiert. Des Weiteren werden, wie in Abs. 2.5.3.1 (S.25) beschrieben, Basiselemente entfernt, deren Gruppen sich mit den Gruppen anderer Basiselemente überlappen. Dabei wird dem Basiselement der Vorzug gegeben, das nach der Sortierung nach dem Auftreten in **d**, den höheren Rang hat.

### 3.4.1.3 Verknüpfungen suchen

Für die genaue Bestimmung der Anzahl der Elemente eines Basiselements, werden die Verknüpfungen zwischen den Gruppen der Basiselemente gesucht. Dafür werden sie erneut umsortiert, diesmal nach dem Betrag ihrer Translation  $v$ . Wie in Abs. 2.5.3.2 (S.28) begründet, müssen die Gruppen der Basiselemente untereinander verknüpft sein. Mit Verknüpfung ist hier gemeint, dass sie gleiche Werte aus **d** enthalten.

Es wird also zunächst für jedes Basiselement ein Vektor **x** angelegt mit allen Werten aus **d**, die seine Gruppenelemente vertreten.

Die benachbarten Basiselemente werden von oben nach unten auf ihre Verknüpfungspunkte untersucht. Diese müssen für ihre jeweilige Gruppe das 0-Element sein, alle Elemente einer Gruppe vor dem 0-Element werden durch den in Abs. 2.5.3 (S.25) beschriebenen Schatteneffekt verursacht. Die endgültige Anzahl der Elemente eines Basiselements ergibt sich dann aus dem größten auftretenden Index der Elemente in den Gruppen des Basiselements.

```

. Alle Gruppen aller Basiselemente werden mit flag=false markiert.
  Zeigt an, dass diese Gruppe noch nicht mit einer anderen (eines anderen
  Basiselements) verknüpft werden konnte.

. Für oberstes Basiselement wird erste Gruppe übernommen

  . → basis(n).v=v und basis(n).anz = Anzahl der Elemente der ersten
    Gruppe (größter Index)
  . mit n - Anzahl der verbliebenen Basiselemente
  . basis(n).gruppe(1).flag = true

. für alle weiteren Basiselemente i, von oben nach unten
{

. Initialisierung der Anzahl der Elemente dieses Basiselements mit Anzahl der
  Elemente der ersten Gruppe
  → max_anz

. für alle Gruppen des oberen Nachbar-Basiselements i + 1
{

  ? falls für betrachtete Gruppe flag=true
  {
    . für alle Gruppen des aktuellen Basiselements i

```

```

{
  ? haben beide Gruppen gemeinsames Element?
  ja → Verknüpfung gefunden
    . flag = true für aktuellen Gruppe des aktuellen Basiselements
    . Vergleiche Anzahl der Elemente der aktuellen Gruppe des
      aktuellen Basiselements mit dem bisher registrierten Wert für
      max_anz
      Registriere den größere Wert
    . dem Vektor connected können Indizes der beteiligten
      Basiselemente hinzugefügt werden
  } ( für alle Gruppen des aktuellen Basiselements i )
  } ( falls für betrachtete Gruppe flag=true )

} ( für alle Gruppen des oberen Nachbar-Basiselements i + 1 )

. Anzahl der Elemente für das aktuelle Basiselement übernehmen
basis(i).anz = max_anz

} ( für alle weiteren Basiselemente i, von oben nach unten )

. den Vektor connected um alle doppelten Einträge bereinigen

```

Damit sind alle Angaben zur Beschreibung der Basis gefunden:

```

basis(i).v:      Translation des Basiselements i
basis(i).anz:    Anzahl der Elemente mit dieser Translation
                  i = 1...n, mit n - Anzahl der gefundenen Basiselemente
basis.connected: Indizes der verbundenen Basiselemente

```

Der Index-Vektor **connected** wurde eingeführt, weil trotz der vielen Schritte, die Menge der Kandidaten einzuschränken, mitunter einzelne Basiselemente in der Auswahl verbleiben, ohne Bezug zu anderen Basiselementen.

### 3.4.2 Ergebnisse

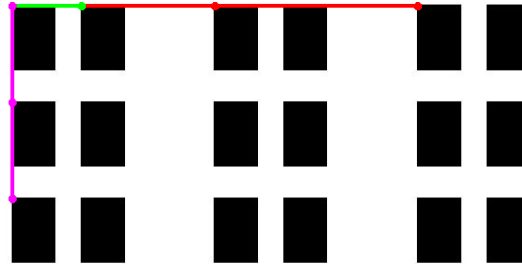
Für die Visualisierung der gefundenen Basiselemente wurde die Matlab-Funktion **visualize\_basis** erzeugt. Der Funktionsaufruf lautet

```
visualize_basis(basis, image);.
```

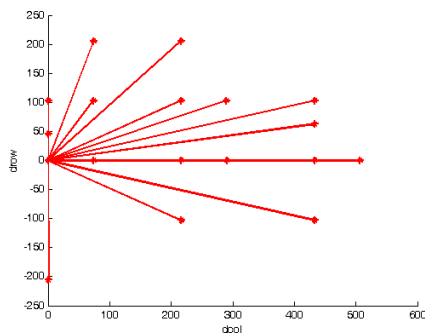
Es werden das Struct **basis** als Ergebnis von **descGroups**, mit den in Zeilen- und Spaltenrichtung gefundenen Basiselementen, und die Bildmatrix **image** übergeben.

Es existiert kein Bezugspunkt für die Basiselemente, die gefundenen Translationen sind freie Vektoren. Um das Ergebnis aber dennoch möglichst anschaulich darstellen zu können, wurde hier eine semiautomatische Visualisierung gewählt. Der User wird aufgefordert den Nullpunkt der Basis per Mausklick selbst zu wählen. So kann die Basis bspw. auf einer Fensterecke platziert werden. Alle zusammenhängenden Basiselemente werden, farblich differenziert, übereinander dargestellt, die kleineren über den größeren. So können die gefundenen Basiselemente direkt mit der Struktur der Fassade verglichen werden. Falls die gefundene Basis aus mehreren, nicht verbundenen Elementen besteht, so werden diese mit einem kleinen Versatz dargestellt, um zu verdeutlichen, dass die eingezeichneten Translationen nicht zu verknüpfen sind.

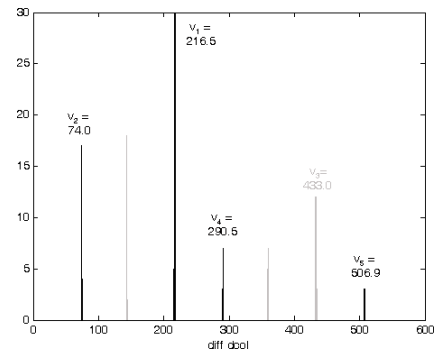
Für die hier gezeigten Ergebnisse wurden die Basiselemente auf diese Art und Weise in den Ergebnisbildern platziert.



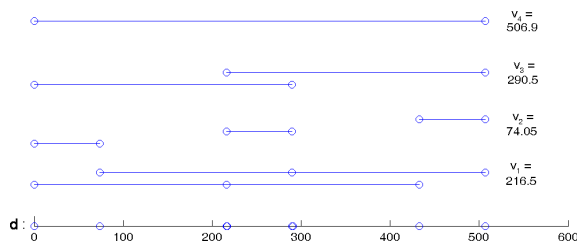
(a) gefundene Basis



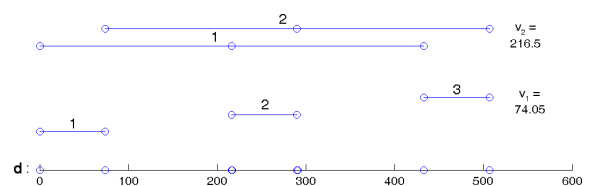
(b) Die Translationen der für das Beispiel aus (a) detektierten wiederholten Gruppen. Es wurden 368 Keypoints detektiert und darunter 2692 Feature-Paare gematcht. Schließlich wurden 17 wiederholte Gruppen gefunden.



(c) Das Histogramm der Differenzen der Translationen in Spaltenrichtung.



(d) Die erste Auswahl der potentiellen Kandidaten für die Basiselemente, abgeleitet aus dem Histogramm der Differenzen, sowie die mit den jeweiligen Abständen  $v$  gefundenen Gruppierungen in d.

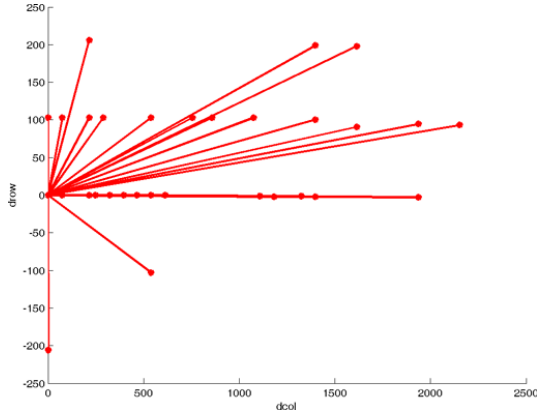


(e) Die endgültige Auswahl der Basiselemente.

Abbildung 3.18: Die Zwischenergebnisse für die Spaltenrichtung auf dem Weg zur Beschreibung der Struktur des synthetischen Bildes der typischen Fensterfront. Die gefundenen Basiselemente: in Zeilenrichtung:  $3 \times 102.984$ . In Spaltenrichtung: 1.  $2 \times 74.0$ , 2.  $3 \times 216.5$ , Connected: 1 2.



(a) Ermittelte Basiselemente für das komplexeste getestete synthetische Beispielbild.



(b) Die Ergebnisse der Detektion wiederholter Gruppen für das Beispiel aus (a) in Form der Translationsvektoren. Es wurden 944 Keypoints im Bild detektiert und darunter 16752 Feature-Paare gemacht. Daraus wurden 30 wiederholte Gruppen selektiert.

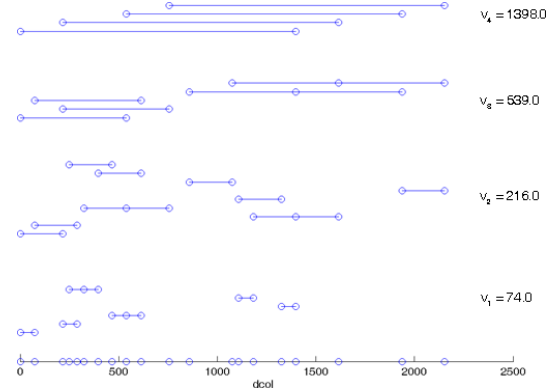
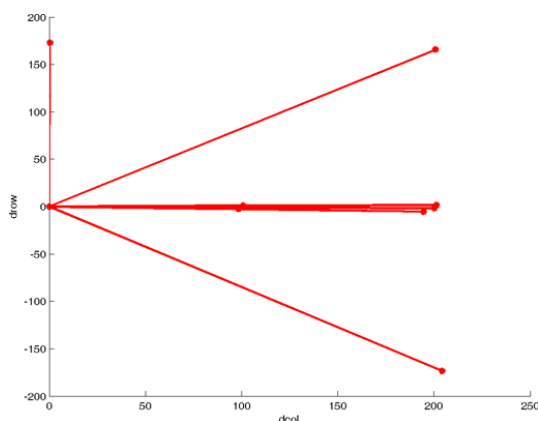
(c) Die endgültig ausgewählten Basiselemente in Spaltenrichtung und die jeweils gefundenen Gruppen in **d**.

Abbildung 3.19: Ergebnis für das komplexeste der getesteten synthetischen Beispiele (Format: 2387x334). Die ermittelten Basiselemente: in Zeilenrichtung 3 x 102.779, in Spaltenrichtung: 1. 2 x 74.0, 2. 2 x 216.0, 3. 2 x 539.0, 4. 2 x 1398.0, Connected: 1 2 3 4.

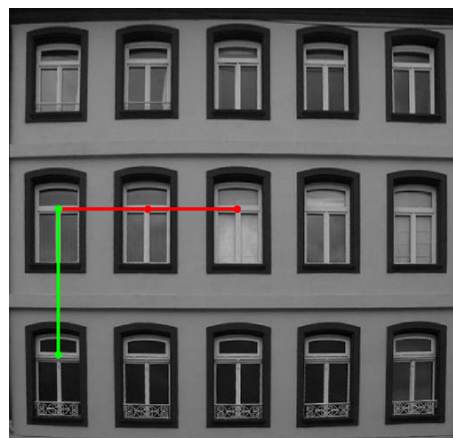
Abb. 3.18 zeigt die Zwischenergebnisse der in den letzten Abschnitten beschriebenen Schritte auf dem Weg zur Beschreibung der Struktur des in (a) gezeigten Beispielbildes. In (b) sind noch einmal die Translationen der detektierten wiederholten Gruppen für dieses Beispiel dargestellt. Hier ist die Struktur der Bildelemente schon gut in dem von den Vektoren aufgespannten Gitter zu erkennen. Die Abb. (c) bis (e) schließlich veranschaulichen die Suche der Basiselemente und ihrer Gruppen in den detektierten Translationen in Spaltenrichtung. Schon in Abb. (e) ist das Endergebnis leicht abzulesen. Für die kleinste Translation  $v_1^c = 74.0$  wurden drei gleichgroße Gruppen á zwei Elementen identifiziert. Für das größere Basiselement mit  $v_2^c = 216.51$  wurden ebenfalls zwei Gruppen gefunden, mit jeweils drei Elementen. In Abb.(a) sind in Spaltenrichtung also zwei Basiselemente eingezeichnet worden. Das größere Basiselement (rot) mit drei Elementen im Abstand  $v_2^c$  und das kleinere (grün) mit zwei Elementen im Abstand  $v_1$ . An jeden Knoten des roten Basiselements kann das grüne Basiselement angeknüpft werden, um ein weiteres Bildelement zu erreichen. So kann durch Linearkombinationen der beiden Basiselemente die gesamte Bildstruktur (in Spaltenrichtung) beschrieben werden.

Abb. 3.19 zeigt die ermittelten Basiselemente für das komplexeste der verwendeten synthetischen Testbilder. Die Struktur dieser modellhaften Fensterfront ist in Spaltenrichtung durch Wiederholungen in vier Skalen gekennzeichnet. Abb. (c) zeigt die endgültig ausgewählten Basiselemente in Spaltenrichtung und die jeweils gefundenen Gruppierungen in **d**. Hier wird deutlich wie viele verschiedene Möglichkeiten entstehen, die potentiellen Basiselemente in **d** zu gruppieren. Aber der entwickelte Algorithmus hat seine Aufgabe hier vollständig und korrekt erfüllt, trotz der Komplexität der vorgelegten Struktur.



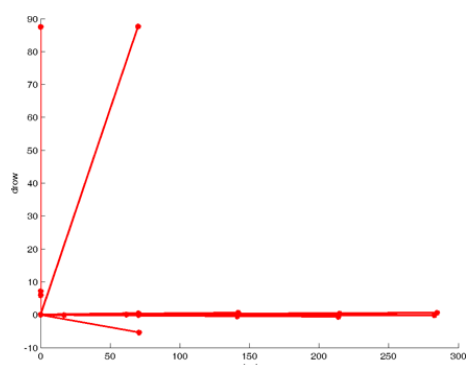


(a) Ergebnisse der Detektion wiederholter Gruppen: 951 Keypoints, 778 Matches, 9 wiederholte Gruppen.

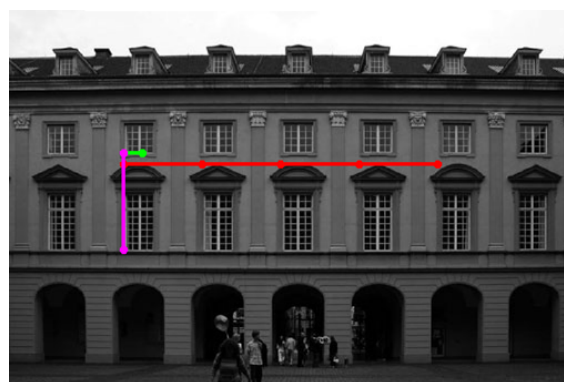


(b) gefundene Basis

Abbildung 3.20: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $2 \times 164.2$ , in Spaltenrichtung  $3 \times 100.6$ . (Für dieses Experiment wurde das Testbild aus Abb. 3.8(a) (S.43) weiter ausgeschnitten, um den regelmäßigen Teil der Fassade zu selektieren. Die Verzerrungen am unteren Rand der Abbildung sind nicht durch Objektiv-Verzeichnungen entstanden. Das abgebildete Gebäude ist sehr alt und tatsächlich derart verbogen. (Format: 512x492))



(a) Ergebnisse der Detektion wiederholter Gruppen: 1368 Keypoints, 1346 Matches, 15 wiederholte Gruppen.



(b) gefundene Basis

Abbildung 3.21: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $2 \times 87.2$ , in Spaltenrichtung  $1.2 \times 16.9$ ,  $2.5 \times 70.7$ , Not connected

In Spaltenrichtung wurden zwei unabhängige Basiselemente ermittelt. Das kleinere bezieht sich auf die Fensterkreuze und das größere auf die Fenster selbst.

Es folgen einige Ergebnisse „echter“ Testdaten.

Die Abb. 3.20 und 3.21 zeigen die Ergebnisse zweier Experimente mit Aufnahmen sehr alter Gebäude. Die Grundstrukturen wurden korrekt erkannt. In beiden Fällen wurden aber nicht alle dargestellten Bildelemente (Fenster) erfasst.

Die Ursachen dafür liegen aber vor der eigentlichen Suche der Struktur der Fassade. Die ermittelte Basis ist nur so vollständig wie die detektierten wiederholten Gruppen. Aufgrund der Beschaffenheit von Fensterfronten, mit Spiegelungen, Schatten, verschiedene Gardinen etc. werden eben nicht alle Fenster oder Fensterreihen als wiederholte Gruppe detektiert und damit fehlen Elemente im Gitter der Translationsvektoren.

Die für das Beispiel aus Abb. 3.21 detektierten wiederholten Gruppen erstrecken sich über das gesamte Bild, beziehen also auch Features im Dach oder im Bereich des Erdgeschosses mit ein, die nicht zur Fensterfront gehören. Und trotzdem werden die korrekten Basiselemente aus allen

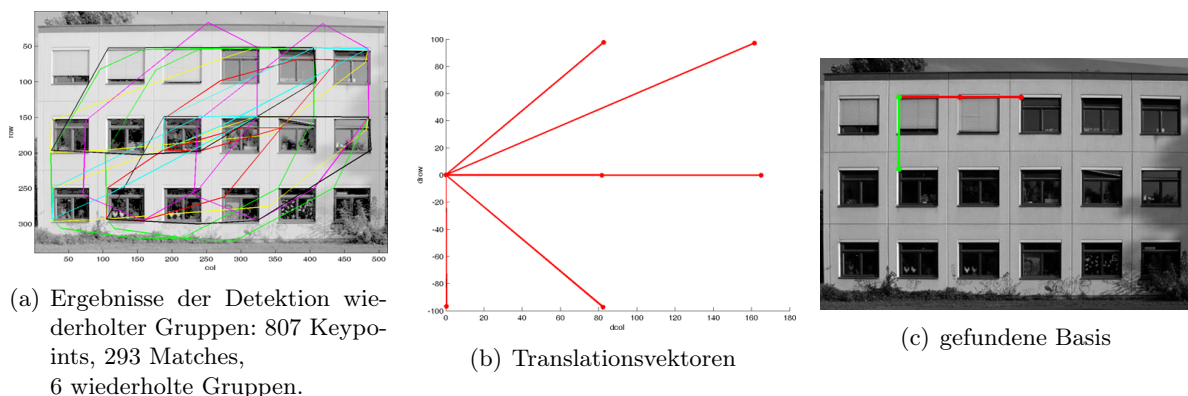


Abbildung 3.22: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $2 \times 97.1$ , in Spaltenrichtung  $3 \times 82.5$ .

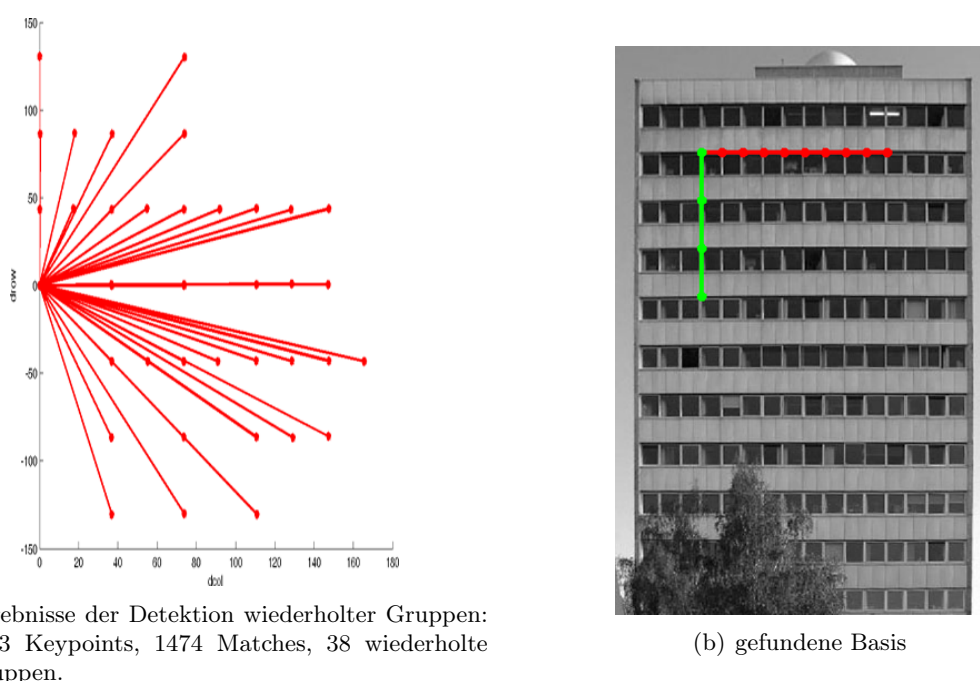
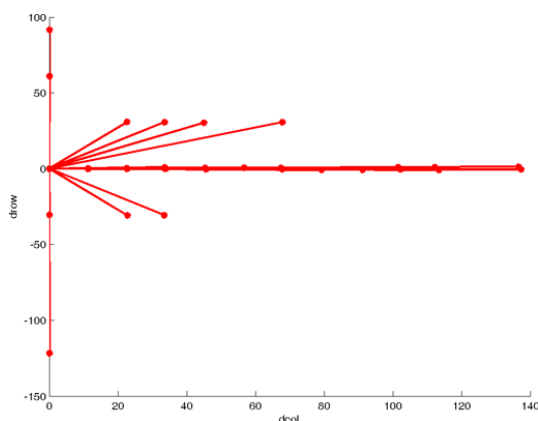


Abbildung 3.23: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $4 \times 43.2$ , in Spaltenrichtung  $10 \times 18.5$ .

Translationsvektoren herausgefiltert. Das gleiche gilt für Abb. 3.24.

Abb. 3.22 zeigt ein weiteres einfaches Beispiel, hier für ein modernes, schlichtes Gebäude. Auch hier wurde die Struktur, die Abstände zwischen den Fenstern, korrekt erkannt aber nicht die gesamte Bildstruktur erfasst. In diesem Punkt sind keine qualitativen Unterschiede zwischen Aufnahmen von Altbauten und von industriell gefertigten modernen Gebäuden zu erkennen. Die beschriebenen Effekte durch Spiegelungen, Schatten, Gardinen etc. haben einen größeren Einfluss als eventuelle bauliche Ungenauigkeiten in der Ausführung der einzelnen Fassadenelemente. Die Abb. 3.23 und 3.24 zeigen die Ergebnisse zweier Experimente mit sehr großen Fensterfronten. Für diese Beispiele wurden 38 zw. 31 wiederholte Gruppen detektiert. Die Abstände der Fenster werden in Zeilen- und Spaltenrichtung korrekt erkannt. Es werden viele, bis zu 11, wiederholte Bildelemente erfasst, aber wieder nicht über die gesamte Fassade.

Die ermittelte Basis kann die Struktur der Fassade nur dann in ihrer ganzen Breite wiedergeben, wenn durch die Detektion der wiederholten Gruppen, die Wiederholung zwischen erstem und letztem Fenster der Reihe und aller auftretenden Translationen dazwischen erfasst werden.

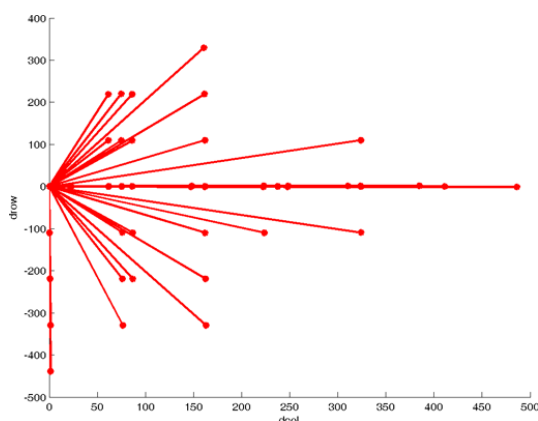


(a) Ergebnisse der Detektion wiederholter Gruppen: 3360 Keypoints, 2329 Matches, 31 wiederholte Gruppen.

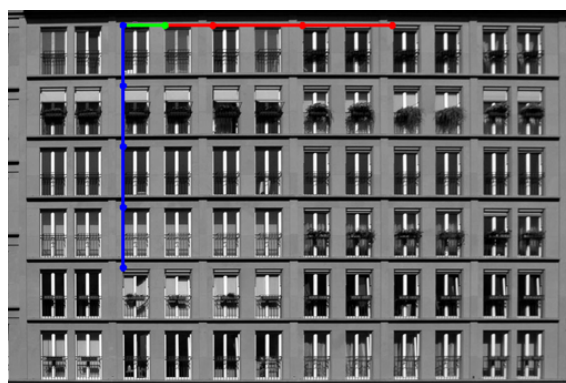


(b) gefundene Basis

Abbildung 3.24: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $5 \times 30.2$ , in Spaltenrichtung  $11 \times 11.3$ .



(a) Ergebnisse der Detektion wiederholter Gruppen: 6149 Keypoints, 4194 Matches, 49 wiederholte Gruppen.



(b) gefundene Basis (Für die Spaltenrichtung wurden hier nur zwei der drei gefundenen Basiselemente visualisiert.)

Abbildung 3.25: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $5 \times 109.5$ , in Spaltenrichtung  $1. \times 21.8$ ,  $2. \times 75.5$ ,  $3. \times 161.8$ , Connected: 2 3.

Das kleinste der drei Basiselemente in Spaltenrichtung steht nicht im Zusammenhang mit dem 2. und 3. Basiselement. Es bezieht sich auf die Fensterkreuze und wurde hier nicht mit dargestellt.

Gerade bei den großen Fensterfronten ist zu beobachten, dass das nicht erfüllt wird.

Das Testbild aus Abb. 3.17 (S.51) wird in Abb. 3.25 erneut aufgegriffen. In Spaltenrichtung werden hier die unterschiedlichen Abstände zwischen den Fenstern registriert und für diese Richtung eine Basis bestehend aus zwei Basiselementen aufgebaut, die, bis auf eine Doppelfensterreihe, das gesamte Bild erfassen. In Zeilenrichtung werden bis auf eine alle Fensterreihen erkannt. Ein drittes kleines Basiselement in Spaltenrichtung, ohne Bezug zu den anderen, wurde in der Abbildung nicht mit dargestellt.

Ein weiteres gutes Ergebnis zeigt Abb. 3.26. Augenscheinlich sind die Fenster gleich abständig. Es wurde in Spaltenrichtung aber eine Basis bestehend aus zwei Basiselementen aufgebaut, die die gesamte Breite der regelmäßigen Struktur abdeckt. Und das ist auch richtig so, der mittlere Fensterabstand ist tatsächlich 10Pixel bzw. 30% größer als die anderen beiden. In Zeilenrichtung wurde der Abstand der Fensterreihen korrekt registriert aber nur drei der sechs Reihen erkannt. Hierfür ist sicherlich der große Schatten links unten verantwortlich zu machen. Abb. (a) zeigt die für dieses Beispiel detektierten wiederholten Gruppen. Es ist zu erkennen, dass sich alle Gruppen

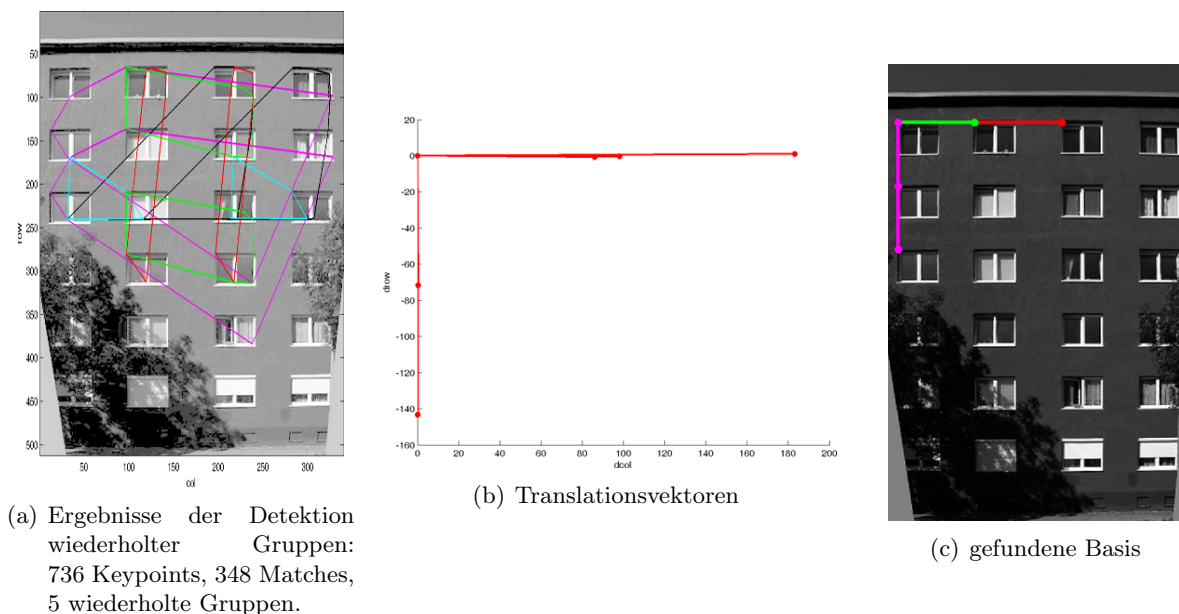


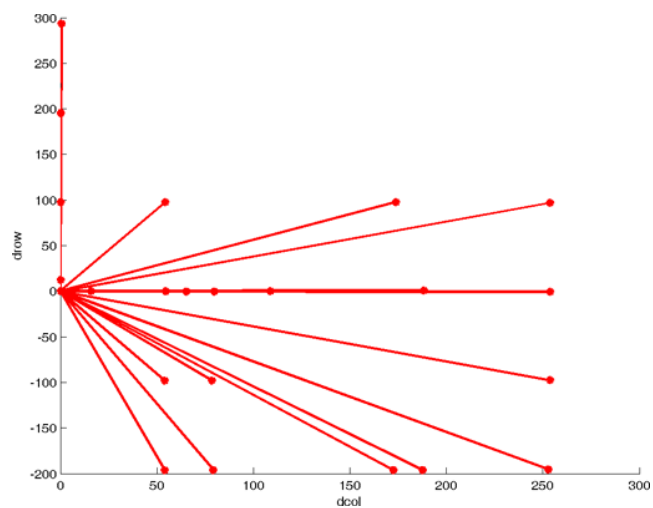
Abbildung 3.26: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung  $3 \times 71.1617$ , in Spaltenrichtung  $1. \ 2 \times 85.5$ ,  $2. \ 2 \times 183.0$ , Connected: 1 2.

auf den Bildteil ohne Schatten beschränken.

Ein letztes Experiment soll mit Abb.3.27 vorgestellt werden. Es wurden insgesamt fünf nicht zusammenhängende Basiselemente identifiziert, zwei für die Zeilen- und drei für die Spaltenrichtung. Die jeweils kleinsten Elemente beziehen sich auf die Fensterkreuze, die anderen auf die Anordnung der Fenster in der Fassade. Für die Zeilenrichtung wurde mit dem größeren der beiden Basiselemente der Abstand der Fensterreihen korrekt identifiziert und auch alle vollständig sichtbaren Fenster erfasst. In Spaltenrichtung wurden zwei weitere nicht zusammenhängende Basiselemente erzeugt. Die für Abb. 3.27(b) manipulierte Darstellung der Basiselemente macht deutlich, dass diese Trennung der Struktur der Fassade genau entspricht. Die Fenster sind so angeordnet, dass sich die größeren Fenster an den Rändern wiederholen und die kleineren Fenster in einer Dreier-Gruppe in der Mitte der Fensterfront angeordnet sind.

Hier zeigt sich, dass verschiedene nicht zusammenhängende Basiselemente durchaus sinnvoll sein können. In diesem Fall kann nicht entschieden werden, welche der (beiden größeren) Basiselemente das „richtige“ ist.

So erfreulich der Erfolg dieses Experimentes ist, macht es doch auch ein Manko des bisherigen Algorithmus deutlich: Den noch fehlenden Bezug zu den Elementen der Fassadenstruktur bzw. zwischen den nicht im Zusammenhang stehenden Basiselementen. Bei Weiterführung dieses Projektes wäre es sicherlich wünschenswert diesen Bezug herzustellen.



(a) 1255 Keypoints, 918 Matches, 22 wiederholte Gruppen



(b) gefundene Basis. Die Abbildung der Basiselemente wurde hier nachträglich aus drei verschiedenen Basisvisualisierungen photomontiert, um den Bezug der verschiedenen Elemente zur Fassade besser zu verdeutlichen.

Abbildung 3.27: Für dieses Beispiel wurden folgende Basiselemente ermittelt: in Zeilenrichtung 1.  $2 \times 11.9$ , 2.  $4 \times 97.8$ , Not connected, in Spaltenrichtung 1.  $2 \times 15.3$ , 2.  $3 \times 54.2$ , 3.  $2 \times 253.3$ , Not connected.

## 4. Zusammenfassung und Ausblick

### Was erreicht wurde und was noch fehlt

#### 4.1 Zusammenfassung

Das von Loy und Eklundh in [Loy and Eklundh, 2006] vorgestellte Verfahren konnte vollständig umgesetzt werden. Eine Methode zur Detektion multipler Symmetrien wurde implementiert und es hat sich gezeigt, dass die Anwendung auf Fassadenaufnahmen sehr gute Ergebnisse liefert. Des Weiteren konnten die erarbeiteten Konzepte auf die Detektion wiederholter Strukturen erweitert werden. Für die Ergebnisse dieser Detektion wurde ein Verfahren entwickelt, das die Struktur der Fassade in Form von Basiselementen wiedergibt.

Die in Matlab implementierten Funktionen arbeiten auf entzerrten Graustufenbildern. Für alle aufgetretenen Schwellwerte und Parameter konnten allgemeine Werte gefunden werden, die für alle Testdaten gleichermaßen gute Ergebnisse lieferten.

#### 4.2 Ausblick

Die Hauptelemente der Charakteristik der Fassade wurden durchweg korrekt erkannt. Es wurde aber auch deutlich, dass der Bezug zwischen gefundenen Basiselementen und Fassadenstruktur noch fehlt. Bei Weiterführung dieses Projektes wäre es sicherlich wünschenswert diesen Bezug herzustellen. Dazu sollten die Fasadenelemente, also die Fenster und Türen selbst bzw. die Bildelemente aus denen sich die gesamte Fassade zusammensetzen lässt, identifiziert werden, um aus einer Kombination von Bildelementen und Basiselementen die gesamte Fassade rekonstruieren zu können. Dann wird es auch möglich sein, auf Lücken in der erkannten Fassadenstruktur zu schließen.

In [Cornelius and Loy, 2006] wird eine Weiterführung der Arbeit aus [Loy and Eklundh, 2006] vorgestellt. Das auch in dieser Arbeit verwendete Konzept wird erweitert auf die Detektion affiner Symmetrien, also in perspektivisch abgebildeten Bildern. Dominante Symmetrien werden hier aus Paaren symmetrischer Features abgeleitet. Dafür werden die jeweiligen Symmetriachsen aus den Fluchtpunkten der Verbindungslinien der beteiligten Features bestimmt. Eine derartige Erweiterung der hier implementierten Verfahren ist für die Zukunft denkbar.

Ein ebenfalls in [Cornelius and Loy, 2006] vorgeschlagenes Detail ist die Reduzierung der Vielfalt der detektierten Symmetrien auf diejenigen, die mindestens den halben Support der besten Symmetrieachse haben. In vielen der hier getesteten Beispiele hat sich gezeigt, dass gerade die Symmetrieachsen mit geringem Support oftmals Symmetrien im Bild unterstützen, die nicht mit

der eigentlichen regelmäßigen Struktur der Fassade im Zusammenhang stehen. Die Einführung solcher Beschränkungen sollte in Erwägung gezogen werden.

Ein weiteres Detail, das für weiterführende Arbeiten in Erwägung gezogen werden könnte, ist die Einführung eines Distanzgewichtes entsprechend Gl. (2.6) (S.19). Es könnte ein zweiter Durchlauf durch den Algorithmus eingeführt werden, mit einem Distanzgewicht in Abhängigkeit von der zuvor ermittelten Fensterbreite. Hier wäre dann zu untersuchen, wie sich die Ergebnisse der Detektionen in Bezug auf die Struktur der Fenster verbessern.

Das im Fachbereich Photogrammetrie vertretene EU-Projekt eTrims<sup>1</sup> beschäftigt sich u.a. mit der automatischen Erkennung von Gebäuden in Bildern. Wie schon in Abs. 1.2 erwähnt, finden hier, wegen der Vielfalt und Komplexität von Gebäuden, herkömmliche Objekterkennungsmethoden keine Anwendung. Die Ergebnisse dieser Arbeit sollen für die Forschungen im Rahmen dieses Projektes verwendet werden, als Grundlage für eine kompakte Bildbeschreibung als High-Level-Feature für das modellbasierte Lernen von Häusern.

---

<sup>1</sup>**eTRIMS** - E-Training for Interpreting Images of Man-Made Scenes  
<http://www.ipb.uni-bonn.de/projects/etrims/>

# Anhang A

## Herleitungen

### A.1 Die mittlere Symmetrieachse

In Abs. 2.3.2.6 (S.19) wurde beschrieben, wie dominante Symmetrien aus den detektierten Features selektiert werden.

An dieser Stelle soll gezeigt werden, wie über Varianz-Fortpflanzung die für eine Gruppe zusammengehöriger Feature-Paare gültige mittlere Symmetrieachse abgeleitet wird.  $\bar{\rho}$  und  $\bar{\theta}$  entstehen dabei als gewichtetes Mittel aus den Polarkoordinaten aller beteiligten Symmetrieachsen.

Seien  $\mathbf{p}$  und  $\mathbf{q}$  die Positionen zweier symmetrischer Features, mit den Koordinaten

$$\mathbf{p} = \begin{bmatrix} r_p \\ c_p \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} r_q \\ c_q \end{bmatrix},$$

dann ergeben sich die Polar-Koordinaten der zugehörigen Symmetrieachse aus

$$\begin{aligned} \theta &= \arctan \frac{c_q - c_p}{r_q - r_p} \\ \rho &= \frac{r_p + r_q}{2} \cdot \cos(\theta) + \frac{c_p + c_q}{2} \cdot \sin(\theta). \end{aligned} \tag{A.1}$$

Abb. A.1(b) zeigt schematisch den funktionalen Zusammenhang für die Varianz-Fortpflanzung, bei der Berechnung der  $(\rho, \theta)$ -Repräsentation der Symmetrieachse *eines* Feature-Paares.

Für die Jakobi-Matrizen ergibt sich, mit den Identitätsbedingungen für die beteiligten Koordinaten:

$$\begin{aligned} \mathbf{F}_1 &= \begin{bmatrix} \partial_{r_p} r_p & \partial_{c_p} r_p & \partial_{r_q} r_p & \partial_{c_q} r_p \\ \partial_{r_p} c_p & \partial_{c_p} c_p & \partial_{r_q} c_p & \partial_{c_q} c_p \\ \partial_{r_p} r_q & \partial_{c_p} r_q & \partial_{r_q} r_q & \partial_{c_q} r_q \\ \partial_{r_p} c_q & \partial_{c_p} c_q & \partial_{r_q} c_q & \partial_{c_q} c_q \\ \partial_{r_p} \theta & \partial_{c_p} \theta & \partial_{r_q} \theta & \partial_{c_q} \theta \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-c_p + c_q}{(c_p - c_q)^2 + (r_p - r_q)^2} & \frac{r_p - r_q}{(c_p - c_q)^2 + (r_p - r_q)^2} & \frac{c_p - c_q}{(c_p - c_q)^2 + (r_p - r_q)^2} & \frac{-r_p + r_q}{(c_p - c_q)^2 + (r_p - r_q)^2} \end{bmatrix} \end{aligned} \tag{A.2}$$

sowie



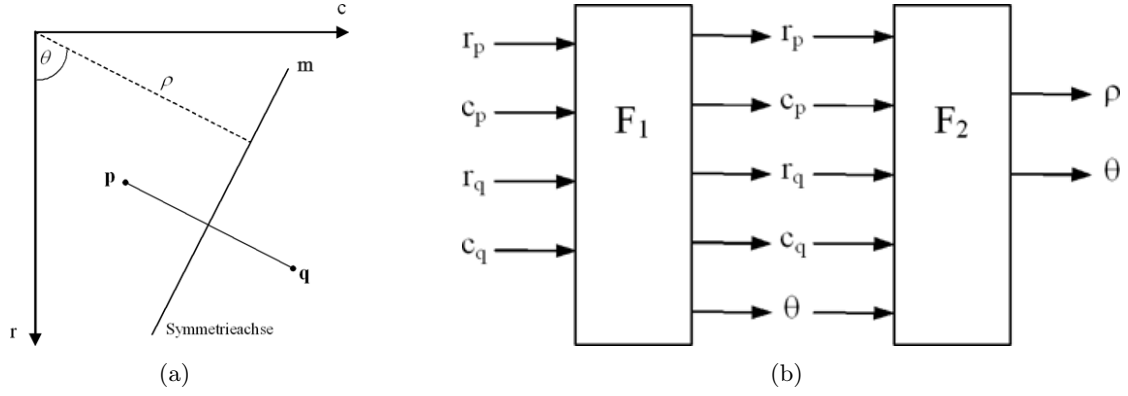


Abbildung A.1: Funktionaler Zusammenhang für die Varianz-Fortpflanzung bei der Berechnung der Symmetrieachse eines Feature-Paares.

$$\begin{aligned}
 \mathbf{F}_2 &= \begin{bmatrix} \frac{\partial r_p \rho}{\partial r_p \theta} & \frac{\partial c_p \rho}{\partial c_p \theta} & \frac{\partial r_q \rho}{\partial r_q \theta} & \frac{\partial c_q \rho}{\partial c_q \theta} & \frac{\partial \theta \rho}{\partial \theta \theta} \\ \frac{\partial r_p \rho}{\partial r_p \theta} & \frac{\partial c_p \rho}{\partial c_p \theta} & \frac{\partial r_q \rho}{\partial r_q \theta} & \frac{\partial c_q \rho}{\partial c_q \theta} & \frac{\partial \theta \rho}{\partial \theta \theta} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\cos \theta}{2} & \frac{\sin \theta}{2} & \frac{\cos \theta}{2} & \frac{\sin \theta}{2} & \frac{(c_p + c_q) \cdot \cos \theta}{2} + \frac{(r_p + r_q) \cdot \sin \theta}{2} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{A.3}$$

Die Gewichts-Matrix wird in Abhängigkeit vom Symmetriegrad  $M$  aus Gl. (2.7) (S.19) des Feature-Paares und der Summe der Symmetriegrade aller beteiligten Feature-Paare gewählt, sodass für die Summe der Gewichte  $\sum p_i = 1$  gilt.

$$\mathbf{P} = \mathbf{E}_4 \cdot M / \sum (M_i) \tag{A.4}$$

Mit  $\sigma_0 = 1$  ergibt sich die Kovarianz-Matrix der Beobachtungen damit zu

$$\Sigma_{\mathbf{pq}} = \mathbf{P}^{-1} \tag{A.5}$$

und die Kovarianz-Matrix der abgeleiteten Größen  $\rho$  und  $\theta$  zu

$$\Sigma_{\rho\theta} = \mathbf{F}_2 \cdot \mathbf{F}_1 \cdot \Sigma_{\mathbf{pq}} \cdot \mathbf{F}_1^T \cdot \mathbf{F}_2^T. \tag{A.6}$$

Seien nun alle  $n$  beteiligten Symmetrieachsen repräsentiert durch

$$\mathbf{m}_i = \begin{bmatrix} \rho_i \\ \theta_i \end{bmatrix}, \quad \Sigma_{\rho_i \theta_i}, \quad i = 1 \dots n.$$

Dann ergibt sich die mittlere Symmetrieachse als gewichtetes Mittel aus allen zuvor berechneten Symmetrieachsen aus

$$\bar{\mathbf{m}} = \sum_i \left( \Sigma_{\rho_i \theta_i}^{-1} \right)^{-1} \cdot \sum_i \left( \Sigma_{\rho_i \theta_i}^{-1} \cdot \mathbf{m}_i \right), \tag{A.7}$$

mit den Varianzen aus

$$\begin{aligned}
 \Sigma_{\bar{\mathbf{m}}} &= \frac{\mathbf{v}^T \mathbf{v}}{n-1} \\
 \text{mit} \\
 \mathbf{v}_i &= \bar{\mathbf{m}} - \mathbf{m}_i \\
 \sigma_\rho^2 &= \Sigma_{\bar{\mathbf{m}}} 1,1 \\
 \sigma_\theta^2 &= \Sigma_{\bar{\mathbf{m}}} 2,2.
 \end{aligned} \tag{A.8}$$

## A.2 Die mittlere Translation

In Abs. 2.4 (S.21) wurde gezeigt, wie mit nur wenigen Abwandlungen des Verfahrens zur Symmetrie-Detektion, wiederholte Gruppen in Bildern gefunden werden können.

Die endgültige Repräsentation der gefundenen Translationen erfolgt dabei, zur besseren Interpretation der Ergebnisse, in Form eines  $[drow, dcol]$ -Vektors für jede gefundene wiederholte Gruppe. Dafür wird aus allen  $n$  Paaren, die dem Maximum im Hough-Akkumulator in einer kleinen Umgebung  $(dr, d\theta)$  zugeordnet werden können, das mit  $M^*$  gewichtete Mittel für  $drow$  und  $dcol$  gebildet. Da die beiden Größen unabhängig voneinander sind, ist hier keine Varianz-Fortpflanzung, wie in Anhang A.1, für jede einzelne Translation nötig. Beide Größen können getrennt voneinander betrachtet werden.

Für jedes Feature-Paar ergeben sich die Translationen pro Achsrichtung

$$\begin{aligned} drow_{ij} &= row_j - row_i \\ dcol_{ij} &= col_j - col_i. \end{aligned} \quad i, j = 1 \dots n \quad (\text{A.9})$$

Die Elemente des Translationsvektors ergeben sich dann als gewichtets Mittel aus

$$\begin{aligned} \overline{drow} &= \frac{\sum_k (M_k \cdot drow_k)}{\sum_k M_k} \\ \overline{dcol} &= \frac{\sum_k (M_k \cdot dcol_k)}{\sum_k M_k} \end{aligned} \quad k = 1 \dots n \quad (\text{A.10})$$

und die entsprechenden Varianzen aus den Verbesserungen zum zuvor berechneten Mittel mit

$$\begin{aligned} v_{drow_k} &= \overline{drow} - drow_k \\ v_{dcol_k} &= \overline{dcol} - dcol_k \\ \sigma_{drow}^2 &= \frac{\mathbf{v}_{drow}^T \cdot \mathbf{v}_{drow}}{n - 1} \\ \sigma_{dcol}^2 &= \frac{\mathbf{v}_{dcol}^T \cdot \mathbf{v}_{dcol}}{n - 1}. \end{aligned} \quad (\text{A.11})$$

# Literaturverzeichnis

- [Brenner and Ripperda, 2006] Brenner, C. and Ripperda, N. (2006). Extraction of Façades using RJMCMC and Constraint Equations. In Förstner, W. and Steffen, R., editors, *Proceedings of the ISPRS Comm. III Symposium Photogrammetric Computer Vision PCV'06, IAPRS*, volume XXXVI, Bonn. [10]
- [Bronstein et al., 2001] Bronstein, I. N., Semendjajew, K., Musiol, G., and Mühlig, H. (2001). *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt am Main. 5., überarbeitete und erweiterte Auflage, isbn = "3-8171-2015-X". [12, 24]
- [Cornelius and Loy, 2006] Cornelius, H. and Loy, G. (2006). Detecting Bilateral Symmetry in Perspective. *Conference on Computer Vision and Pattern Recognition Workshop (CV-PRW'06)*, 0. [10, 65]
- [Förstner, 2004] Förstner, W. (2004). Shortcuts and pitfalls in basic photogrammetric education or Can a general rotation matrix represent a Homography? Commemorative volume for the 60th birthday of Prof. Dr. Armin Grün, herausgegeben von: Photogrammetry and Remote Sensing Group, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology; ETH-Zürich. [12]
- [Hays et al., 2006] Hays, J., Leordeanu, M., Efros, A. A., and Liu, Y. (2006). Discovering Texture Regularity as a Higher-Order Correspondence Problem. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Proceedings of ECCV 2006*, volume II, pages 522–535. Springer-Verlag Berlin Heidelberg. LNCS 3952. [9]
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110. [14, 15, 17, 36]
- [Loy and Eklundh, 2006] Loy, G. and Eklundh, J.-O. (2006). Detecting Symmetry and Symmetric Constellations of Features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Proceedings of ECCV 2006*, volume II, pages 508–521. Springer-Verlag Berlin Heidelberg. LNCS 3952. [9, 10, 14, 16, 18, 34, 65]
- [McGlone, 2004] McGlone, J. C. (2004). *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing, 5th edition edition. [30]
- [Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630. [13]
- [Schaffalitzky and Zisserman, 1998] Schaffalitzky, F. and Zisserman, A. (1998). Geometric Grouping of Repeated Elements within Images. In *Proceedings of 9th British Machine Vision Conference (BMVC), Southampton*, pages 13–22. [9]
- [Schaffalitzky and Zisserman, 1999] Schaffalitzky, F. and Zisserman, A. (1999). Geometric Grouping of Repeated Elements within Images. In *Shape, Contour and Grouping in Computer Vision*, pages 165–181, London, UK. Springer-Verlag. [9]

- [Schaffalitzky and Zisserman, 2000] Schaffalitzky, F. and Zisserman, A. (2000). Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing (IVC)*, 18(9):647–658. [9]
- [Sethares and Staley, 1999] Sethares, W. A. and Staley, T. W. (1999). Periodicity Transforms. *IEEE Transactions on Signal Processing*, 47(11):2953–2964. [25]
- [Vosselman, 1991] Vosselman, G. (1991). Kurzauszug der Doktorarbeit „Relational Matching“. Landwirtschaftliche Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn. [8]

# Stichwortverzeichnis

- Abstraktionsebenen, 8
- Computer Vision, 8
- Deskriptor, 14–16
  - Fenster, 15
  - Spiegelung, 16, 35
- Deskriptoren-Matching, 17, 35, 48
- Diophantischen Gleichungen, Lineare, 24
- Distanz, euklidisch, 17, 36
- Distanz-Matrix, 17, 36
- Dualität, 19
- Epipol, 30
- Essentielle Matrix, 32
- eTrims, 66
- Feature, 13, 14
  - Deskriptor, 13
  - Detektion, 33, 48
  - Detektor, 13, 16
  - Orientierung, 13, 18
  - Skala, 18
- Fixgerade, 12
- Fixpunkt, 12
- Fluchtlinien, 9
- Fluchtpunkte, 9, 12, 30, 65
- Gauß
  - Filter, 20
  - Funktion, 14, 15
- Gewicht
  - Distanz-, 19
  - Skalen-, 19, 37
  - Winkel-, 21
  - Winkel-Symmetrie-, 18, 21, 36
- ggT
  - größter gemeinsamer Teiler, 24
- Glättung, gaußsche, 15, 37
- Gradienten, 15
- Gradientenrichtung, 10, 14, 15, 34
- High-Level-Feature, 8, 66
- Histogramm, 10, 15, 26, 35, 37, 54
- Homographie, 12
- Hough
  - Akkumulator / Array, 16, 20, 21, 37, 38
  - Raum, 16, 37
  - Transformation, 10, 19
- Keypoint, 14, 15
  - Deskriptor, 15
- Kongruenz
  - gleichsinnige, 12
  - nichtgleichsinnige, 13
- Koplanaritätsbedingung, 32
- Lowe
  - Operator, 16
  - Punkte, 14, 17, 18, 23, 33, 34
- Maßstab, 13
- Markov Chain Monte Carlo, 10
- Matching, 17, 35
- Matlab, 33
- Matlab-Funktionen
  - `descGroups.m`, 53
  - `detectGroups.m`, 48
  - `detectSym.m`, 34, 39
  - `findPattern`, 53
  - `rectify_fassade.m`, 12
  - `rectify_tool.m`, 12
  - `showkeys.m`, 34
  - `sift.m`, 33, 48
  - `visualize_basis`, 57
- Mittel, gewichtetes, 20, 67
- Modell
  - bildung, 11
  - Bild-, 11
  - Objekt-, 11
  - Sensor-, 11
- Parameter
  - Unterstuetzer*, 39
  - $[d\rho, d\theta]$ , 39
  - $\sigma_s$ , 37
  - distRatio*, 17, 22, 36
- Periodicity Transform, 25
- photogrammetrisches Modell, 32
- Quaternion, 30

- Rechnersehen, 8
- Repetitionsgrad, 21, 26, 27, 48
- rjMCMC, 10
- Rotationsinvarianz, 13, 14
- Rotationsmatrix, 12, 30
  
- Schatteneffekt, 25, 28, 56
- schiefsymmetrische Matrix, 30
- SIFT, 14, 33
  - Operator, 14, 16
- Skalen-Raum, 14
- Symmetrie, 12
  - operation, 12
  - zentrum, 12
  - Axiale bzw. Spiegelsymmetrie, 12
  - Detektion, 14, 21, 34, 37
  - Zentrale bzw. Punktsymmetrie, 12
- Symmetrieachse, 14, 16, 18–21
  - mittlere, 20
  - Repräsentation, 17, 20
- Symmetriegrad, 19, 20, 36–39, 48, 68
  
- Texel, 9
- Translation, 21
  - Repräsentation, 21
- Translationsvektor, 21, 23, 25
  
- unendliche Ebene, 12
  
- Varianz-Fortpflanzung, 20, 67
  
- Wiederholte Strukturen, 13
  - Detektion, 21, 48
  
- Zuordnung, 8