

## **Farbsymmetrien zur Detektion von Interessenregionen**

### **Studienarbeit im Studiengang Computervisualistik**

vorgelegt von

Andreas Kasten

Betreuer: Dipl.-Inform. Christian Schneider, Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich Informatik  
Erstgutachter: Dipl.-Inform. Christian Schneider, Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich Informatik  
Zweitgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik

Koblenz, im September 2007



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Arbeitsgruppe für Studien- und Diplomarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja  nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja  nein

Koblenz, den .....

Unterschrift



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	GraySym und ColorSym . . . . .	9
1.3	Gliederung der Arbeit . . . . .	10
<b>2</b>	<b>Stand der Wissenschaft</b>	<b>11</b>
<b>3</b>	<b>Beschreibung und Umsetzung des Verfahrens</b>	<b>15</b>
3.1	Überblick über ColorSym . . . . .	15
3.2	ColorSym . . . . .	17
3.2.1	Grundlagen . . . . .	17
3.2.2	Nachbarschaftsindexmenge $\Gamma$ . . . . .	18
3.2.3	Gradientenindexmenge $\Lambda$ . . . . .	18
3.2.4	Symmetriewert $c$ . . . . .	21
3.2.5	Symmetrieorientierung $\varphi$ . . . . .	22
3.2.6	Symmetriekarte $M$ . . . . .	25
3.2.7	Phasengewichtungsfunktion PWF . . . . .	27
3.2.8	Gradientengewichtungsfunktion GWF . . . . .	29

3.2.9	Distanzgewichtungsfunktion $DWF_{\sigma}$ . . . . .	29
3.2.10	Algorithmus . . . . .	30
3.3	Einbettung in PUMA 2 . . . . .	33
3.3.1	Voraussetzungen . . . . .	33
3.3.2	Implementierung . . . . .	34
<b>4</b>	<b>Experimente und Ergebnisse</b>	<b>39</b>
4.1	Unterscheidung von Symmetriegruppen . . . . .	39
4.2	Vorgehen bei den Experimenten . . . . .	40
4.3	Allgemeinene Funktion . . . . .	41
4.4	Lokalisierung ornamentaler Strukturen . . . . .	44
4.5	Dreipässe und Wirbelräder mit drei Speichen . . . . .	49
4.6	Flächensymmetrien und symmetrische Flächen . . . . .	54
4.7	Bedeutung des Parameters $d_{\max}$ . . . . .	59
4.8	Vergleich zwischen radialer und isotroper Symmetrie . . . . .	63
<b>5</b>	<b>Zusammenfassung</b>	<b>67</b>
<b>A</b>	<b>Demo-Applikation</b> <code>colorSymmetryDemo</code>	<b>71</b>
<b>B</b>	<b>Die Klasse</b> <code>FeatureImage</code>	<b>75</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Das Forschungsprojekt *Bildanalyse zur Ornamentklassifikation* hat es sich zur Aufgabe gemacht, ornamentale Strukturen in Bildern computergestützt zu lokalisieren, analysieren und klassifizieren. Grundlage des Projekts bildet eine umfangreiche Bilddatenbank, deren Abbildungen manuell vorsortiert sind. Durch Kombinationen mit Methoden der Bildverarbeitung und der Verwendung von Wissensdatenbanken (*Knowledge Databases*) soll diese Kategorisierung weiter verfeinert werden.

Sämtliche Bilder durchlaufen bis zum Prozess der Ornamentklassifikation mehrere Vorverarbeitungsschritte. Beginnend mit einem Normalisierungsprozess, bei dem das Bild u. a. entzerrt und entrauscht wird, werden im Anschluss Interessensregionen selektiert. Diese Regionen bilden die Grundlage für das spätere Lokalisieren der Ornamente. Aus ihnen werden mit unterschiedlichen Verfahren Merkmale extrahiert, die wiederum in der Datenbank gespeichert werden. Eine genaue Beschreibung des gesamten Prozesses findet sich unter [SSP06], eine Skizze des Ablaufs findet sich in Abbildung 1.1.

Den Schwerpunkt des Projekts bildet die Analyse und Klassifikation der Ornamente. Dennoch konnte dieser Teil bislang nicht vollständig automatisiert werden. Vor allem die Detektion der Interessensregionen bedarf noch einigen Verbesserungen. Bisherige Arbeiten

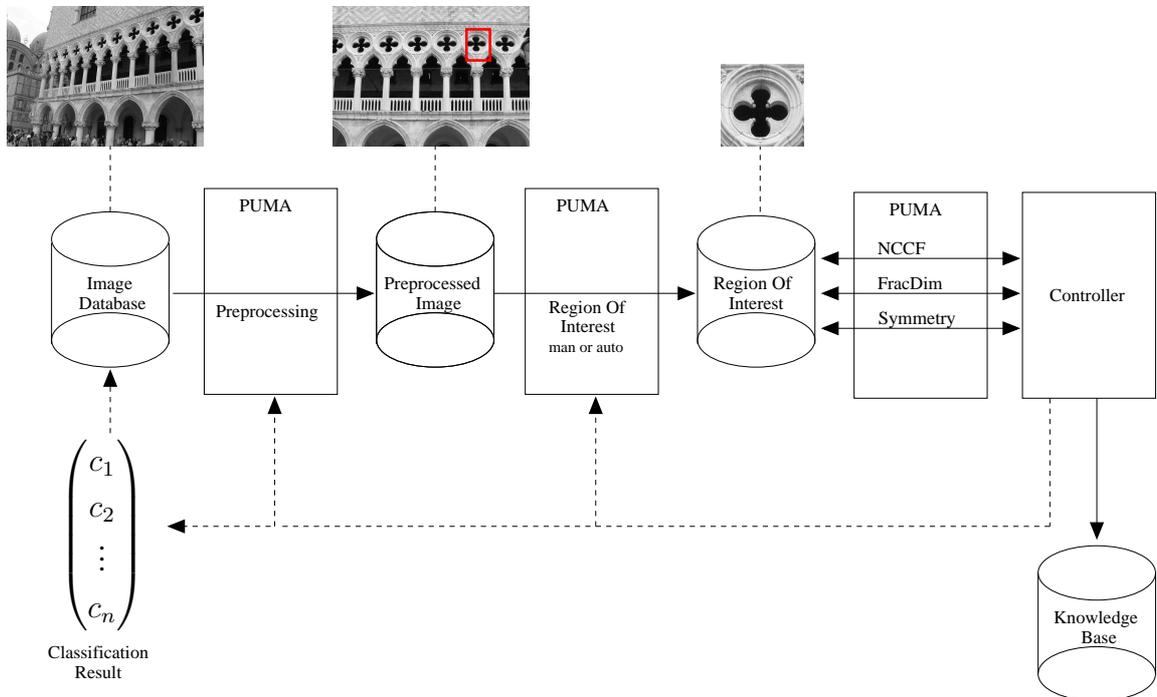


Abbildung 1.1: Schema des Projekts *Bildanalyse zur Ornamentklassifikation*. Vor der Analyse werden die aus der Datenbank gelesenen Bilder normalisiert. Nach dem Detektieren von Interessensregionen werden unter Zuhilfenahme einer Wissensdatenbank Merkmale extrahiert. Die Ergebnisse dessen werden wiederum in einer Datenbank abgelegt.

hierzu finden sich etwa unter [FÖ5, Pal04, Ste06]. Das Ziel der vorliegenden Arbeit ist nun, ein weiteres Verfahren hinsichtlich seiner Tauglichkeit zur Lokalisierung von ornamentalen Bildbereichen zu testen. Das Verfahren trägt den Titel *ColorSym* und wurde von Gunther Heidemann an der Universität Bielefeld entwickelt [Hei04]. Es bildet eine Erweiterung des von Daniel Reisfeld et al. vorgeschlagenen Verfahrens *GraySym*<sup>1</sup> [RWY95].

<sup>1</sup>Bei Reisfeld et al. trägt das Verfahren die Bezeichnung *Generalized Symmetry Transform*.

## 1.2 GraySym und ColorSym

GraySym ist ein Verfahren zur Berechnung von Interessensregionen, das lokale Symmetrie auf Graubildern errechnet. Vorgeschlagen wurde es von Reinfeld et al. in [RWY95]. Die Symmetrie wird dabei als entscheidendes Merkmal für die Definition einer Interessensregion verstanden. GraySym liefert eine Merkmalskarte zurück, die jedem Bildpunkt genau einen Symmetriewert zuweist. Die Bildpunkte werden dabei als potentielle Symmetriezentren aufgefasst. Die Maxima der Merkmalskarte ergeben die Interessenspunkte. GraySym benötigt keinerlei Vorwissen über die Objekte im Bild, sondern verwendet lediglich eine Gradientenkarte des Eingabebildes als Berechnungsgrundlage. Eine vorherige Segmentierung oder eine Unterscheidung zwischen Bildobjekt und Bildhintergrund muss nicht vorgenommen werden.

Symmetrie wird in [RWY95] als ein Merkmal verstanden, das ein Objekt in seiner Gesamtheit beschreibt. Im Gegensatz zu Kanten oder Ecken, die sich nur auf einen Teil des Bildobjektes beziehen, sei die Symmetrie direkt als Merkmal zur Objekterkennung einsetzbar. Weitere Zwischenverfahren, die lokale Merkmale zu größeren Bausteinen zusammensetzen, seien nicht nötig. Auch Vorwissen über die Orientierung und Lage der Objekte im Bild sei nicht erforderlich. Die Berechnung der Symmetrie erfolgt zwar auf den lokalen Merkmalen einer Gradientenkarte, verallgemeinert deren Informationsgehalt aber zu einem übergeordneten Merkmal [RWY95].

GraySym orientiert sich stark an der Kognitionspsychologie [Zab90]. Es verwendet keine strenge mathematische Definition der Symmetrie, sondern verfolgt einen Ansatz, der dem menschlichen Symmetrieverständnis nahe kommen soll [LN87]. In [PS00] konnte tatsächlich eine positive Korrelation zwischen den ermittelten Merkmalen des Verfahrens und den Fixationspunkten des menschlichen Auges nachgewiesen werden.

Heidemann erweitert in [Hei04] den Ansatz von Reinfeld et al. auf Farbbilder und modifiziert die Berechnung des Symmetriewertes (siehe dazu Abschnitt 3.2.7). Die Relevanz der Symmetrie zur Objekterkennung bleibt aber vorhanden. Beiden Verfahren ist gemein, dass sie kein Vorwissen über die Struktur der Eingabebilder benötigen und damit als universelle Operatoren bei vielen Anwendungsgebieten eingesetzt werden können. ColorSym wurde

im Rahmen dieser Arbeit in PUMA 2 (*Programmierungsumgebung für die Musteranalyse*) eingebunden und anhand einiger Beispiele evaluiert.

### **1.3 Gliederung der Arbeit**

Kapitel 2 liefert einen kurzen Überblick über weitere Verfahren zur Detektion von Interessensregionen. Überwiegend verwenden diese ebenfalls Symmetrie als entscheidendes Merkmal. Kapitel 3 beschreibt die theoretischen Grundlagen von ColorSym sowie dessen Einbettung in PUMA 2. Im Anschluss daran werden in Kapitel 4 die Ergebnisse von einigen Testdurchläufen präsentiert und kritisch hinterfragt. Den Abschluss bildet eine Zusammenfassung des Verfahrens und ein Ausblick über mögliche Modifikationen dessen in Kapitel 5.

# Kapitel 2

## Stand der Wissenschaft

Zum Erkennen von Objekten in Bildern werden auffällige Merkmale herangezogen, die die Objekte näher bestimmen. Ein markantes und zugleich komplexes Merkmal bildet die Symmetrie von Objekten. Vereinfacht formuliert, ist ein Objekt genau dann symmetrisch, wenn es sich durch bestimmte Transformationen auf sich selbst abbilden lässt [FFM90, RWY95]. Diese Transformationen entsprechen den Symmetrietransformationen. Die einfachste und zugleich häufigste Form der Symmetrie bildet dabei die Spiegelsymmetrie [OO]. Symmetrie bezieht sich stets auf das gesamte Objekt und ist damit in der Lage, es in seiner Gesamtheit zu beschreiben. Dies unterscheidet Symmetrie von lokalen Merkmalen wie Kanten oder Ecken (etwa [HS88, Mor77]), die nur einen Teil des Objektes betrachten. Symmetrie lässt sich in sehr vielen natürlichen und künstlichen Objekten nachweisen. Entsprechend relevant ist ihre Bedeutung für die menschliche Informationsverarbeitung. Symmetrische Objekte werden als harmonisch empfunden und lenken den Blick des Betrachters auf sich [LN87, Zab90]. Auf der Grundlage kognitionspsychologischer Untersuchungen fand die Symmetrie bereits in vielen Gebieten der Bildverarbeitung Anwendung (vgl. [RWY95]). Beispiele für die Verwendung in der Objekterkennung finden sich etwa unter [Mar89, WC94]. Obgleich die Symmetrie ein sehr allgemeines Merkmal ist, wird sie auch für spezifischere Aufgaben wie etwa der Detektion von Gesichtern eingesetzt [ML04, SHW98, ZRF<sup>+</sup>06].

Ideale Symmetrie kann vor allem in natürlichen Bildern aufgrund häufiger Verzerrungen und anderer Artefakte nur selten erreicht werden [RWY95]. Daher haben Zambrodsky et al. in [ZPA92, ZPA95] eine Symmetriedistanz  $SD$  entworfen, die den Symmetriehalt einer Fläche beschreibt. Die  $SD$  ist definiert als der minimale Aufwand, der benötigt wird, um eine beliebige Fläche in eine adäquate symmetrische Fläche zu transformieren. Unterschieden werden dabei drei Arten von Symmetrie: die Rotationssymmetrie, die Spiegelsymmetrie und die Kombination aus beiden. Für jede dieser drei Arten gibt es eine Symmetrietransformation  $ST$ , die jeden Punkt der Fläche so verschiebt, dass die resultierende Fläche symmetrisch ist. Hierbei kann es mehrere Lösungen geben. Für jede wird die gemittelte Summe über die Distanzen aller Punktverschiebungen berechnet. Die  $SD$  ist definiert als die kleinste so ermittelte Summe. Das Verfahren ist nur definiert für Polygone in Grauwertbildern. Flächen, die keine Polygone sind, müssen zunächst in solche umgewandelt werden. Komplexere Objekte mit mehreren Grauwerten werden für jeden Grauwert separat betrachtet oder in ein 3D-Höhenbild umgewandelt. Für letzteres wird dann eine 3D-Symmetrie ermittelt. Ferner lässt sich das Verfahren dazu einsetzen, das Symmetriezentrum teilweise verdeckter Objekte zu ermitteln und das Objekt zu rekonstruieren. Dabei wird iterativ ein Symmetriezentrum gesucht, das den Wert der  $SD$  minimiert. Eine weitere Anwendung kann das Verfahren beim Auffinden mehrerer symmetrischer Bereiche in einem Bild finden. Mit Hilfe von QuadTrees wird rekursiv für jeden quadratischen Bereich die  $SD$  ermittelt. Der Bereich mit der kleinsten  $SD$  wird weiter unterteilt, bis letztlich ein Interessenspunkt übrig bleibt. Durch Wiederholung dieser Prozedur lassen sich mehrere lokale Symmetriezentren auffinden. Bereits gefundene symmetrische Bereiche sind dabei herauszufiltern.

Inspiziert durch GraySym [RWY95] haben Loy und Zelinsky in [LZ02, LZ03] ein Verfahren entwickelt, das lokale radiale Symmetrie auf Graubildern errechnet. Für jeden Bildpunkt wird ermittelt, wie stark er die Symmetrie seiner Nachbarpunkte beeinflusst. Die Nachbarpunkte werden bestimmt durch mehrere Radien. Jeder Radius wird dabei separat betrachtet. Als Berechnungsgrundlage dient eine Gradientenkarte des Eingabebildes. Ferner wird für jeden Radius ein *Orientierungsprojektionsbild* und ein *Betragsprojektionsbild* definiert. Beide dieser Bilder sind initial leer und genauso groß wie das Eingabebild. Für jeden Bildpunkt werden auf Basis seines Gradienten zwei benachbarte Punkte definiert:

einer in Gradientenrichtung im Abstand des betrachteten Radius und ein zweiter entgegen der Gradientenrichtung im gleichen Abstand. Für beide ermittelte Punkte werden das Orientierungsprojektionsbild und das Betragsprojektionsbild modifiziert. Die Gradientenrichtung bestimmt dabei, ob die Projektionsbilder an den entsprechenden Stellen erhöht oder verringert werden. Auch negative Werte werden hierbei zugelassen. Für jeden Radius werden die resultierenden Projektionsbilder gewichtet, multiplikativ verrechnet und mit einem Gaußkern gefaltet. Das Resultat dessen wird für alle Radien aufsummiert und gemittelt. Das Ergebnis dieser Summe entspricht der Symmetriekarte. Die lokalen Minima und Maxima dieser Karte bestimmen dann die Interessensregionen. Das Verfahren wurde optimiert auf Echtzeitfähigkeit und mit ähnlichen Verfahren zur Detektion von Interessensregionen verglichen (darunter u. a. [RWY95, DGV95, SL97]). Vergleichbare Resultate seien vor allem mit der radialen Symmetrie von GraySym (vgl. Abschnitt 3.2.6) zu erzielen.

Ein weiteres Verfahren zum Ermitteln lokaler Symmetrie haben Loy und Eklundh in [LE06] vorgeschlagen. Das Verfahren basiert auf rotationsinvarianten Merkmalen, die vorab durch ein Merkmalsextraktionsverfahren ermittelt werden. Die gefundenen Merkmale werden dann auf zwei Arten kodiert: als Punktvektor und als Merkmalsvektor. Ein Punktvektor besitzt eine Positionsangabe im Originalbild sowie eine Orientierung und eine Skalierung und dient der eigentlichen Berechnung der Symmetrie. Merkmalsvektoren geben eine eindeutige Beschreibung des Merkmals an und werden zum Bestimmen der Ähnlichkeit zweier Merkmale eingesetzt. Das Verfahren bildet nun Paare von Merkmalen bzw. deren Punktvektoren. Für jedes Paar wird ermittelt, wie groß sein Symmetriehalt ist. Dabei wird unterschieden zwischen Spiegelsymmetrie und Rotationssymmetrie. Bei der Spiegelsymmetrie wird ein Paar erstellt, wenn die entsprechenden Merkmalsvektoren durch Spiegelung aufeinander abgebildet werden. Jedes Paar von Punktvektoren definiert dann eine Symmetrieachse. Alle solche Symmetrieachsen werden anschließend gewichtet und über eine Hough-Transformation akkumuliert. Die Maxima des Hough-Raumes entsprechen den Symmetrieachsen. Eine ähnliche Berechnung erfolgt bei der Rotationssymmetrie. Dort werden jedoch die Paare von Punktvektoren ohne vorherige Spiegelung gebildet. Jeder Punktvektor kann mit jedem anderen zu einem Paar zusammengefasst werden. Sind beide Punktvektoren nicht parallel, lassen sie sich durch eine Rotation aufeinander ab-

bilden. Ein entsprechendes Rotationszentrum wird dann ermittelt und gewichtet. Alle so errechneten Rotationszentren werden anschließend in einem Merkmalsbild akkumuliert. Die Maxima dieses Bildes geben die tatsächlichen Rotationszentren an. Die Berechnung beider Symmetriarten lässt sich auf komplexen Hintergründen anwenden und ist in der Lage, mehrere symmetrische Bereiche in einem Bild aufzufinden. Die Qualität der Berechnungen hängt jedoch stark von dem gewählten Merkmalsextraktionsverfahren ab. Loy und Eklundh haben in ihrer Implementierung SIFT [Low04] verwendet und damit brauchbare Ergebnisse erzielt.

# Kapitel 3

## Beschreibung und Umsetzung des Verfahrens

In den folgenden Abschnitten soll das Verfahren ColorSym von Gunther Heidemann vorgestellt werden. Da ColorSym eine Erweiterung zu GraySym von Reisfeld et al. darstellt, ist es an einigen Stellen erforderlich, ebenfalls auf dieses Verfahren kurz einzugehen. Dies gilt auch für solche Bereiche, in denen beide Verfahren voneinander abweichen. Zur besseren Unterscheidung beider Verfahren werden im Folgenden alle Definitionen für GraySym mit einem <sup>G</sup> markiert.

### 3.1 Überblick über ColorSym

ColorSym sucht in Farbbildern nach Symmetriezentren, die invariant gegenüber Rotationen, Translationen und Beleuchtungsänderungen sind [Hei04, Hei06]. Als Berechnungsgrundlage dient eine Gradientenkarte. Diese wird für jeden Farbkanal separat berechnet und ist entscheidend für die Qualität der späteren Ergebnisse. Ermittelt werden Gradientenorientierung und Gradientenintensität.

ColorSym betrachtet jeden Bildpunkt als potentiell Symmetriezentrum. Der Symmetriegehalt eines Punktes wird berechnet auf Basis seiner Nachbarschaft, d. h. die Nach-

barpunkte bestimmen den Symmetriehalt eines Zentrums (vgl. [LZ02, LZ03]). Aus dieser Nachbarschaft werden nun alle möglichen Paare von Punkten gebildet, die symmetrisch um das Zentrum liegen. Für jedes solche Paar wird anschließend ein entsprechendes Gradientenpaar ermittelt. Da ein Punkt je einen Gradienten pro Farbkanal besitzt, ergeben sich für ein Symmetriepaar auch mehrere mögliche Gradientenpaare. Kleinere Gradienten besitzen aber nur wenig Informationsgehalt und sind anfällig gegenüber Rauschen. Daher werden sie für die Erstellung der Gradientenpaare nicht berücksichtigt (vgl. [Hei04, LZ03]). Die zur weiteren Berechnung verwendeten Gradientenpaare werden aus allen verbleibenden Gradienten gebildet.

Für jedes Gradientenpaar wird im Anschluss der Symmetriewert bezüglich des aktuell betrachteten Zentrums ermittelt. Dies erfolgt auf Basis von drei Gewichtungsfunktionen: Die Phasengewichtungsfunktion bestimmt die Symmetrie anhand der Gradientenwinkel, die Gradientengewichtungsfunktion gewichtet die Intensitäten der Gradienten und die Distanzgewichtungsfunktion verrechnet die relative Distanz der beiden Gradienten zueinander (vgl. [LE06]). Der Symmetriehalt eines Bildpunktes ergibt sich durch die Summe aller so ermittelten Symmetriewerte seiner Nachbarpunkte. Das Ergebnis von ColorSym ist eine Symmetriekarte, die jedem Bildpunkt seinen berechtigten Symmetriehalt zuweist. Je höher dieser Wert für einen Punkt ist, desto größer ist die Wahrscheinlichkeit, dass dieser Punkt tatsächlich ein Symmetriezentrum darstellt. Die Maxima der Symmetriekarte definieren die Interessensregionen.

ColorSym ist implementiert für RGB-Bilder, lässt sich aber für beliebige andere mehrdimensionale Bilder verallgemeinern. Die Voraussetzung dafür ist, dass alle Kanäle gleichwertig sind und sich ähnlich verrechnen lassen. Bei einer Implementierung für HSV müsste das Verfahren etwa angepasst werden, da die drei Kanäle unterschiedlich betrachtet werden müssen. Im Folgenden wird das Verfahren lediglich für RGB-Bilder beschrieben werden.

## 3.2 ColorSym

Das Ziel dieses Abschnitts ist es, das theoretische Modell zur Berechnung der Farbsymmetrie zu beschreiben. Im Wesentlichen handelt es sich dabei um die Angabe mathematischer Definitionen. Die verwendeten Notationen weichen dabei minimal von denen in [Hei04, Hei06] ab.

### 3.2.1 Grundlagen

Dieser Abschnitt behandelt Grundlagen, die für das weitere Verständnis des Verfahrens wichtig sind. Da es sich hierbei um bereits Bekanntes handelt, wird es auch nur kurz umrissen werden. Sei  $I(\mathbf{p})$  ein RGB-Farbbild an der Position  $\mathbf{p} = (x, y)$ , dessen Symmetriekarte berechnet werden soll,  $I_c(\mathbf{p})$  der Farbkanal  $c$  mit  $c \in \{0, 1, 2\}$ . Die Kantenbilder  $I_{c,x}(\mathbf{p})$  und  $I_{c,y}(\mathbf{p})$  werden für die  $x$ - bzw.  $y$ -Richtung eines jeden Farbkanals berechnet wie folgt:

$$\begin{aligned} I_{c,x}(\mathbf{p}) &= \frac{\partial I_c(\mathbf{p})}{\partial x} \\ I_{c,y}(\mathbf{p}) &= \frac{\partial I_c(\mathbf{p})}{\partial y} \end{aligned} \quad (3.1)$$

Sei nun  $\mathbf{g}_c = (\mathbf{G}_c(\mathbf{p}), \Theta_c(\mathbf{p}))$  der Gradient an der Position  $\mathbf{p}$  im Farbkanal  $c$ .  $\mathbf{G}_c(\mathbf{p})$  sei dabei die Gradientenintensitätskarte und  $\Theta_c(\mathbf{p})$  die Gradientenwinkelkarte an der Position  $\mathbf{p}$  und wie folgt definiert:

$$\mathbf{G}_c(\mathbf{p}) = \sqrt{I_{c,x}(\mathbf{p})^2 + I_{c,y}(\mathbf{p})^2} \quad (3.2)$$

$$\Theta_c(\mathbf{p}) = \text{atan} \left( \frac{I_{c,y}(\mathbf{p})}{I_{c,x}(\mathbf{p})} \right) \quad (3.3)$$

Die Gradientenkarte für den Farbkanal  $c$  besteht damit aus dem Tupel  $(\mathbf{G}_c, \Theta_c)$ . Die gesamte Gradientenkarte sei definiert als  $(\mathbf{G}, \Theta)$ . Diese besitzt so viele Kanäle wie es Farbkanäle im Eingabebild  $I$  gibt.

Für GraySym diene das Graubild  $I^G(\mathbf{p})$  als Eingabe. Die zugehörige Gradientenkarte sei analog zu den Gleichungen 3.1, 3.2 und 3.3 definiert als  $(\mathbf{G}^G, \Theta^G)$ , der Gradient an der Position  $\mathbf{p}$  sei definiert als  $\mathbf{g}^G = (\mathbf{G}^G(\mathbf{p}), \Theta^G(\mathbf{p}))$ .

### 3.2.2 Nachbarschaftsindexmenge $\Gamma$

Die Gradientenkarte bildet die Grundlage zur Berechnung der Symmetriekarte. Um den Symmetriehalt an einem Bildpunkt  $\mathbf{p}$  zu ermitteln, muss angegeben werden, welche Bildbereiche dafür betrachtet werden sollen. Dies wird beschrieben durch die zwei Indexmengen  $\Gamma(\mathbf{p})$  und  $\Lambda(\mathbf{p}, l, r)$ .  $\Gamma(\mathbf{p})$  definiert eine Nachbarschaft um den Zentrumspunkt  $\mathbf{p}$ . Die Menge enthält paarweise Indizes  $(l, r)$ , die entsprechende Bildpunkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  referenzieren. Die Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  liegen symmetrisch um den Zentrumspunkt  $\mathbf{p}$ . Alle drei Punkte liegen auf einer Geraden. Die Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  haben jeweils den Abstand  $d$  zu  $\mathbf{p}$  und zueinander den Abstand  $2d$ .  $d$  selbst wird nach oben beschränkt durch den maximalen Abstand  $d_{\max}$ , der damit die Größe der betrachteten Nachbarschaft angibt. Skizziert wird dieser Zusammenhang in Abbildung 3.1, mathematisch formuliert lautet er wie folgt:

$$\Gamma(\mathbf{p}) = \left\{ (l, r) \mid l, r \in \mathbb{N}_0 \wedge \frac{\mathbf{p}_l + \mathbf{p}_r}{2} = \mathbf{p} \wedge \|\mathbf{p}_l - \mathbf{p}_r\| \leq 2d_{\max} \right\} \quad (3.4)$$

Die Indexmenge  $\Gamma(\mathbf{p})$  beschreibt also eine kreisförmige Nachbarschaft um den Punkt  $\mathbf{p}$  mit dem Radius  $d_{\max}$ . Für den Symmetriehalt an dieser Position werden alle Paare  $(l, r)$  der Menge  $\Gamma(\mathbf{p})$  bzw. ihre entsprechenden Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  berücksichtigt. Da  $\Gamma(\mathbf{p})$  unabhängig ist von den zu betrachtenden Farbkanälen, ist die obige Definition der Menge sowohl für ColorSym als auch für GraySym gültig.

### 3.2.3 Gradientenindexmenge $\Lambda$

Für die Bildpositionen  $\mathbf{p}_l$  und  $\mathbf{p}_r$  muss ferner festgelegt werden, wie die Gradientenkanäle  $\mathbf{G}_c$  zur Berechnung verwendet werden sollen. Eine einfache Variante wäre hierbei, für jeden Farbkanal separat einen Symmetriewert zu ermitteln. Damit ließen sich aber nur solche symmetrischen Bereiche auffinden, die vollständig in mindestens einem Farbka-

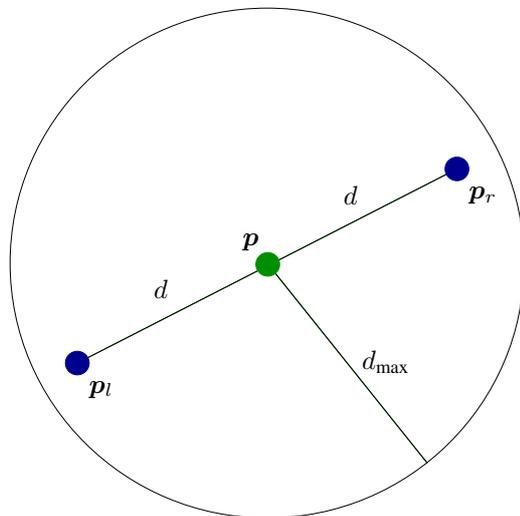


Abbildung 3.1: Beispiel für ein Paar  $(l, r)$  der Menge  $\Gamma(\mathbf{p})$ . Der Maximalradius  $d_{\max}$  spannt eine kreisförmige Nachbarschaft um den Zentrumspunkt  $\mathbf{p}$  auf. Innerhalb dieser Nachbarschaft werden Paare von Punkten  $(\mathbf{p}_l, \mathbf{p}_r)$  gebildet. Die Punkte  $\mathbf{p}$ ,  $\mathbf{p}_l$  und  $\mathbf{p}_r$  liegen dabei auf einer Geraden.  $\mathbf{p}_l$  und  $\mathbf{p}_r$  haben den Abstand  $d$  von  $\mathbf{p}$  und zueinander den Abstand  $2d$ . Es gilt  $d \leq d_{\max}$ .

nal vorhanden sind. Symmetrien, die sich hingegen auf mehrere Farbkanäle verteilen, könnten damit nicht ermittelt werden. Ein Beispiel hierfür zeigt etwa Abbildung 3.2. Heidemann hat in [Hei04] auf diese Schwäche hingewiesen und daher die Gradientenindexmenge  $\Lambda(\mathbf{p}, l, r)$  eingeführt. Diese legt in Abhängigkeit des Symmetriezentrums  $\mathbf{p}$  und des Symmetriepaars  $(\mathbf{p}_l, \mathbf{p}_r)$  fest, wie die einzelnen Kanäle der Gradientenkarte zur weiteren Berechnung verwendet werden sollen. Für jeden der beiden Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  stehen jeweils drei Kanäle zur Verfügung, wodurch sich insgesamt neun mögliche Kombinationen  $(i, j)$  mit  $i, j \in \{0, 1, 2\}$  ergeben. Einige von diesen Kombinationen besitzen jedoch nur einen geringen Informationsgehalt. Der Informationsgehalt wird definiert durch die Intensität der Gradienten an den Positionen  $\mathbf{p}_l$  und  $\mathbf{p}_r$ . Durch Setzen entsprechender Schwellwerte  $\vartheta_c$  wird für jedes der drei Gradientenintensitätsbilder  $G_c$  eine minimale Intensität festgelegt, die der minimal zu erreichenden Information entspricht. Ein Indexpaar  $(i, j)$  wird in die Menge  $\Lambda(\mathbf{p}, l, r)$  aufgenommen, wenn die Gradientenintensität  $G_i(\mathbf{p}_l)$  mindestens so groß ist wie der Schwellwert  $\vartheta_i$  und zugleich die Gradienten-

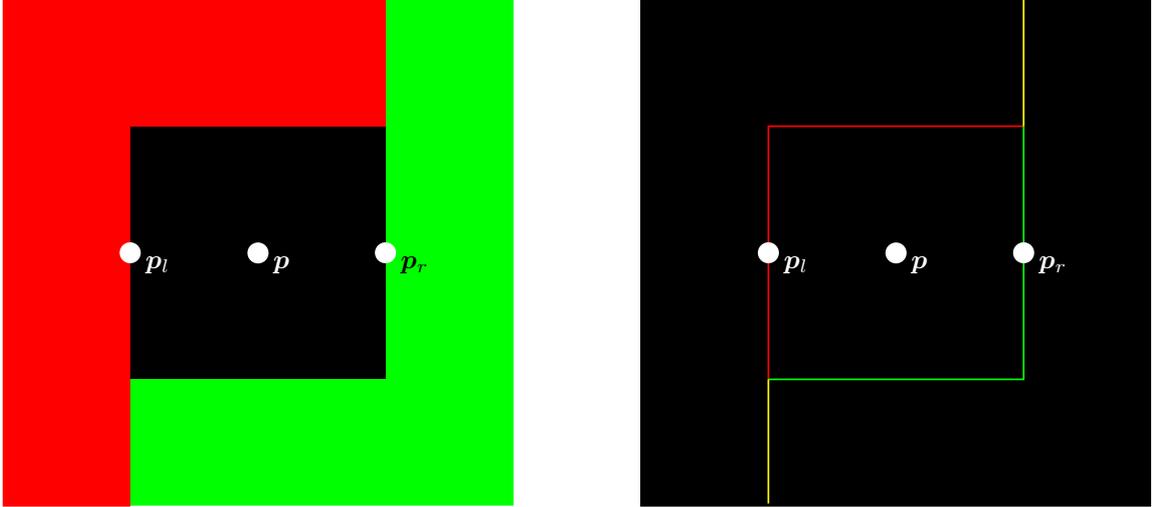


Abbildung 3.2: Beispiel für  $\Lambda(\mathbf{p}, l, r)$ . Links zu sehen ist das Originalbild, rechts die drei überlagerten Gradientenkarten. Keine der Gradientenkarten beinhaltet die vollständige Kontur des schwarzen Quadrats. Die Symmetrie des Objekts ist nur dann zu ermitteln, wenn die Gradienten  $\mathbf{g}_{l,0}$  (rot) und  $\mathbf{g}_{r,1}$  (grün) gemeinsam betrachtet werden. Für das Beispiel gilt  $\Lambda(\mathbf{p}, l, r) = \{(0, 1)\}$ .

tenintensität  $\mathbf{G}_j(\mathbf{p}_r)$  mindestens so groß ist wie der Schwellwert  $\vartheta_j$ . Ein Beispiel für die Indexmenge  $\Lambda(\mathbf{p}, l, r)$  findet sich in [Abbildung 3.2](#). Ihre Definition lautet wie folgt:

$$\Lambda(\mathbf{p}, l, r) = \{(i, j) \mid i, j \in \{0, 1, 2\} \wedge \mathbf{G}_i(\mathbf{p}_l) \geq \vartheta_i \wedge \mathbf{G}_j(\mathbf{p}_r) \geq \vartheta_r\} \quad (3.5)$$

Die Indexmenge  $\Lambda(\mathbf{p}, l, r)$  reduziert lediglich den Berechnungsaufwand für die spätere Implementierung des Algorithmus [\[Hei06\]](#). Sie ist für die eigentliche Berechnung des Symmetriewertes nicht zwingend erforderlich, da auch Gradienten mit geringer Intensität berücksichtigt werden könnten. Durch eine spätere Gewichtung auf Basis der Gradientenintensität könnte dann ein ähnliches Resultat erreicht werden.

Da GraySym lediglich auf Graubildern arbeitet, ist für dieses Verfahren keine mit  $\Lambda(\mathbf{p}, l, r)$  vergleichbare Indexmenge definiert. Stattdessen wird in [\[RWY95\]](#) der einzige Gradientenkanal  $(\mathbf{G}^G, \Theta^G)$  zur Berechnung verwendet.

### 3.2.4 Symmetriewert $c$

Die Indexmengen  $\Gamma(\mathbf{p})$  und  $\Lambda(\mathbf{p}, l, r)$  bestimmen, welche Gradienten zur Berechnung der Symmetrie im Zentrum  $\mathbf{p}$  verwendet werden sollen. Die Gradienten werden dabei stets paarweise betrachtet. Sei  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  ein solches Gradientenpaar. Entsprechend den Indexmengen sei  $\mathbf{g}_{i,l}$  hierbei der Gradient für den Punkt  $\mathbf{p}_l$  im Farbkanal  $i$  und  $\mathbf{g}_{r,j}$  der Gradient für den Punkt  $\mathbf{p}_r$  im Farbkanal  $j$ . Für jedes Paar  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  lässt sich eine eigene Symmetrieachse berechnen. Diese durchläuft das Zentrum  $\mathbf{p}$  und sei definiert als ein Vektor  $\mathbf{s}_{l,r,i,j} = (c_{l,r,i,j}, \tilde{\varphi}_{l,r,i,j})$  (siehe Abbildung 3.3).  $\tilde{\varphi}_{l,r,i,j}$  entspricht dabei der Orientierung des Vektors [RWY95]. Diese gibt die Richtung der Symmetrie des Paares  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  an.  $c_{l,r,i,j}$  bildet die Intensität des Vektors und entspricht dem eigentlichen Symmetriewert. Je größer dieser Wert ist, desto symmetrischer ist das Gradientenpaar  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$ .  $c_{l,r,i,j}$  wird ermittelt aus drei Gewichtungsfunktionen, die zur Berechnung die Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  verwenden: PWF( $\mathbf{p}, l, r, i, j$ ) bildet die *Phasengewichtungsfunktion*, die die Winkel der Gradienten verrechnet. GWF( $\mathbf{p}, l, r, i, j$ ) entspricht der *Gradientengewichtungsfunktion* und verarbeitet die Gradientenintensitäten. DWF $_{\sigma}$ ( $\mathbf{p}, l, r, i, j$ ) ist die *Distanzgewichtungsfunktion*, die die relative Distanz der beiden Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  gewichtet. Der Symmetriewert  $c_{l,r,i,j}$  ist wie folgt definiert:

$$c_{l,r,i,j} = \text{PWF}(\mathbf{p}, l, r, i, j) \cdot \text{GWF}(\mathbf{p}, l, r, i, j) \cdot \text{DWF}_{\sigma}(\mathbf{p}, l, r, i, j) \quad (3.6)$$

$c_{l,r,i,j}$  bestimmt also den Symmetriewert zweier Punkte für genau ein Gradientenpaar. Die Indexmengen legen dabei fest, wo etwas berechnet wird, die Gewichtungsfunktionen definieren, was berechnet wird. Eine weitere Gewichtungsfunktion, die hier nicht aufgeführt ist, bildet die Farbgewichtungsfunktion CWF( $\mathbf{p}, l, r, i, j$ ). Diese wird von Heidemann in [Hei04, Hei06] nicht erläutert, jedoch in seiner Implementierung als zusätzlicher Parameter bereit gestellt.<sup>1</sup>

Der zu  $\mathbf{s}_{l,r,i,j}$  analoge Symmetrievektor für GraySym sei als  $\mathbf{s}_{l,r}^G = (c_{l,r}^G, \varphi_{l,r}^G)$  definiert.  $c_{l,r}^G$  sei dabei der Symmetriewert,  $\varphi_{l,r}^G$  bilde die Symmetrieorientierung. Für  $c_{l,r}^G$  entfällt

---

<sup>1</sup>Wie dem Aufbau der Funktion sowie deren Anwendung zu entnehmen ist, soll sie offenbar die Farbwerte des Eingabebildes an den Positionen  $\mathbf{p}_l$  und  $\mathbf{p}_r$  verrechnen.

die Betrachtung der Gradientenindexmenge  $\Lambda(\mathbf{p}, l, r)$ , da nur ein einziger Farbkanal zur Berechnung verwendet wird.  $c_{l,r}^G$  ist damit wie folgt definiert:

$$c_{l,r}^G = \text{PWF}^G(\mathbf{p}, l, r) \cdot \text{GWF}^G(\mathbf{p}, l, r) \cdot \text{DWF}_\sigma^G(\mathbf{p}, l, r) \quad (3.7)$$

$c_{l,r,i,j}$  und  $c_{l,r}^G$  unterscheiden sich demnach nur durch ihre Indizes, die als Parameter an die Gewichtungsfunktionen weitergegeben werden. Der weitere Aufbau der Gewichtungsfunktionen ist für GraySym und ColorSym weitgehend identisch. Daher können grundsätzlich für beide Verfahren dieselben Gewichtungsfunktionen eingesetzt werden.

### 3.2.5 Symmetrieorientierung $\varphi$

Heidemann betrachtet in [Hei04] zur Berechnung der Symmetrie nicht die gesamte Symmetrieachse  $\mathbf{s}_{l,r,i,j}$ , sondern lediglich deren Intensität, also den Symmetriewert  $c_{l,r,i,j}$ . Die Orientierung  $\tilde{\varphi}_{l,r,i,j}$  bleibt hingegen unberücksichtigt, da sie nur für die Berechnung der radialen Symmetrie verwendet wird. Heidemann bezieht sich allerdings nur auf die isotrope Symmetrie, die diese Orientierung nicht beachtet. Im Folgenden soll daher zunächst die Orientierung  $\varphi_{l,r}^G$  definiert werden, wie sie von Reisfeld et al. für GraySym in [RWY95] verwendet wird. Darauf aufbauend wird im Anschluss eine entsprechende Modifikation für ColorSym vorgeschlagen.

Die Symmetrieorientierung  $\varphi_{l,r}^G$  gibt an, in welchem Winkel die Symmetrieachse  $\mathbf{s}_{l,r}^G$  die Gerade durch die beiden Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  schneidet (vgl. Abbildung 3.3). Liegt eine ideale Symmetrie vor, so steht  $\mathbf{s}_{l,r}^G$  genau senkrecht auf dieser Geraden.  $\varphi_{l,r}^G$  wird dann berechnet aus den Gradientenwinkeln  $\Theta^G(\mathbf{p}_l)$  und  $\Theta^G(\mathbf{p}_r)$  wie folgt:

$$\varphi_{l,r}^G = \frac{\Theta^G(\mathbf{p}_l) + \Theta^G(\mathbf{p}_r)}{2} \quad (3.8)$$

Diese ideale Symmetrie ist nur dann gegeben, wenn die Symmetrieachse  $\mathbf{s}_{l,r}^G$  eine Spiegelachse bezüglich der beiden Gradienten  $\mathbf{g}_l^G$  und  $\mathbf{g}_r^G$  bildet. Reisfeld et al. weisen in [RWY95] aber darauf hin, dass natürliche Bilder – wenn auch nur in geringem Maße – perspektivischen Verzerrungen unterliegen und  $\mathbf{s}_{l,r}^G$  damit generell nicht senkrecht zur

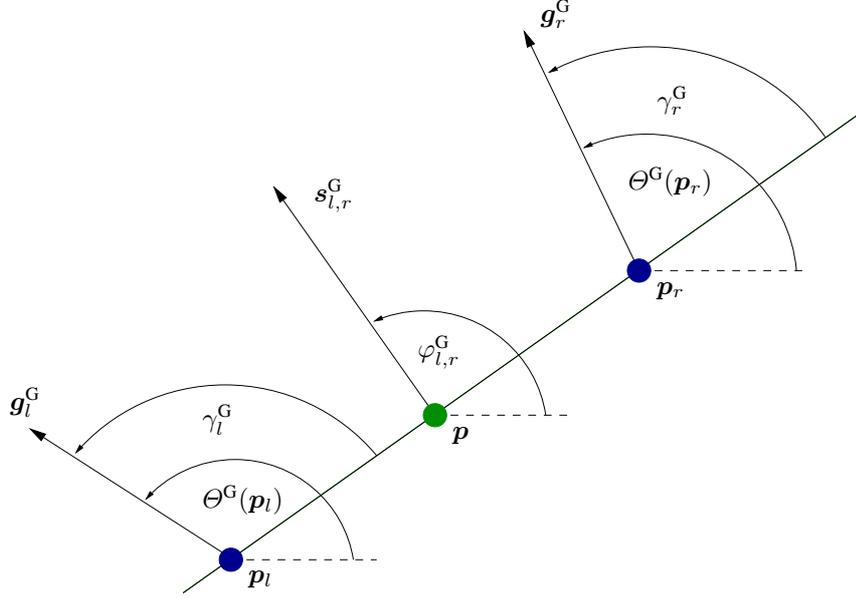


Abbildung 3.3: Jedes Gradientenpaar  $(g_l^G, g_r^G)$  definiert eine Symmetrieachse  $s_{l,r}^G$ . Diese beschreibt die Intensität und die Orientierung der Symmetrie.  $s_{l,r}^G$  steht genau dann senkrecht auf der Geraden durch  $p_l$  und  $p_r$ , wenn  $\Theta^G(p_l) + \Theta^G(p_r) = 180^\circ$ . Die Skizze ist beispielhaft für GraySym definiert (nach [RWY95]).

Geraden durch  $p_l$  und  $p_r$  steht. Daher sollte an Stelle der in Gleichung 3.8 definierten Orientierung eine solche verwendet werden, die diese Verzerrungen berücksichtigt (vgl. hierzu auch [KK83]). Diese Orientierung definieren Reisfeld et al. wie folgt:

$$\tilde{\varphi}_{l,r}^G = \arctan \frac{2 \cdot \cos(\gamma_l^G) \cdot \cos(\gamma_r^G)}{\sin(\gamma_l^G + \gamma_r^G)} \quad (3.9)$$

$\gamma_l^G$  entspricht dabei dem Winkel zwischen dem Gradienten  $g_l^G$  und der Geraden durch die Punkte  $p_l$  und  $p_r$ .  $\gamma_r^G$  bildet den Winkel zwischen dieser Geraden und dem Gradienten  $g_r^G$  (siehe Abbildung 3.3). Die genaue Herleitung sowie weitere Erläuterungen zu  $\tilde{\varphi}_{l,r}^G$  findet sich unter [Rei94].

Der Winkel  $\tilde{\varphi}_{l,r}^G$  beschreibt lediglich die Orientierung der Symmetrieachse  $s_{l,r}^G$ , also die Symmetrierichtung bezüglich des Paares  $(p_l, p_r)$ . Für den Zentrumspunkt  $p$  werden aber entsprechend der Nachbarschaftsindexmenge  $\Gamma(p)$  noch weitere solche Paare betrachtet,

die allesamt zum Symmetriegehalt von  $\mathbf{p}$  beitragen. Jedes dieser Paare definiert eine eigene Symmetrieachse, die wiederum eine entsprechende Orientierung  $\tilde{\varphi}_{l,r}^G$  besitzt. Um die allgemeine Symmetrierichtung für den Punkt  $\mathbf{p}$  zu bestimmen, haben Reisfeld et al. in [RWY95] die Hauptorientierung  $\phi^G(\mathbf{p})$  wie folgt definiert:

$$\phi^G(\mathbf{p}) = \tilde{\varphi}_{a,b}^G, \quad (a, b) = \operatorname{argmax}_{(n,m)} (c_{n,m}^G), \quad (n, m) \in \Gamma(\mathbf{p}) \quad (3.10)$$

Die Symmetriewerte  $c_{l,r}^G$  bestimmen, wie symmetrisch ihre korrespondierenden Gradientenpaare  $(\mathbf{g}_l^G, \mathbf{g}_r^G)$  zum Zentrum  $\mathbf{p}$  sind. Je größer der Wert von  $c_{l,r}^G$ , desto größer ist auch der Einfluss auf den gesamten Symmetriegehalt von  $\mathbf{p}$ . Daher wird die Hauptsymmetrierichtung  $\phi^G(\mathbf{p})$  in Gleichung 3.10 definiert als die Orientierung derjenigen Symmetrieachse, die die größte Intensität  $c_{l,r}^G$  besitzt.

Verwendung findet die Orientierung  $\tilde{\varphi}_{l,r}^G$  zur Berechnung der radialen Symmetrie (siehe Gleichung 3.16). Da Heidemann in [Hei04] aber nur die Berechnung der isotropen Symmetrie behandelt, gibt er für ColorSym auch keine mit  $\tilde{\varphi}_{l,r}^G$  vergleichbare Orientierung an. Testweise wird daher eine Variation von  $\tilde{\varphi}_{l,r}^G$  für Farbbilder wie folgt definiert:

$$\tilde{\varphi}_{l,r,i,j} = \arctan \frac{2 \cdot \cos(\gamma_{l,i}) \cdot \cos(\gamma_{r,j})}{\sin(\gamma_{l,i} + \gamma_{r,j})} \quad (3.11)$$

Im Gegensatz zu GraySym gibt es bei ColorSym für jedes Paar  $(\mathbf{p}_l, \mathbf{p}_r)$  mehrere mögliche Gradientenpaare  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$ , die eine eigene Orientierung  $\tilde{\varphi}_{l,r,i,j}$  definieren. Zur Berechnung der Hauptsymmetrie im Punkt  $\mathbf{p}$  müssen daher alle Paare  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  von  $(\mathbf{p}_l, \mathbf{p}_r)$  berücksichtigt werden. Bestimmt werden diese durch die Indexmenge  $\Lambda(\mathbf{p}, l, r)$ . Ferner müssen alle Symmetriepaare  $(\mathbf{p}_l, \mathbf{p}_r)$  von  $\mathbf{p}$  betrachtet werden, die wiederum durch die Indexmenge  $\Gamma(\mathbf{p})$  beschrieben werden. Analog zu  $\phi^G(\mathbf{p})$  aus Gleichung 3.10 entspricht  $\phi(\mathbf{p})$  der Orientierung derjenigen Symmetrieachse  $\mathbf{s}_{l,r,i,j}$ , die von allen Symmetrieachsen von  $\mathbf{p}$  die größte Intensität  $c_{l,r,i,j}$  besitzt.  $\phi(\mathbf{p})$  ist wie folgt definiert:

$$\begin{aligned} \phi(\mathbf{p}) &= \tilde{\varphi}_{a,b,c,d} \\ (a, b, c, d) &= \operatorname{argmax}_{(n,m,o,p)} (c_{n,m,o,p}), \\ &\quad (n, m) \in \Gamma(\mathbf{p}), (o, p) \in \Lambda(\mathbf{p}, n, m) \end{aligned} \quad (3.12)$$

### 3.2.6 Symmetriekarte $M$

Zur Berechnung der Symmetriekarte werden nun alle Symmetrieachsen  $s_{l,r,i,j}$  verwendet, die sich für das Zentrum  $\mathbf{p}$  ergeben. Reisfeld et al. unterscheiden dabei zwei Arten von Symmetrien [RWY95]: die *isotrope Symmetrie*  $M_{\text{iso}}(\mathbf{p})$  und die *radiale Symmetrie*  $M_{\text{rad}}(\mathbf{p})$ . Diese Unterscheidung basiert auf der Verrechnung der Symmetrieorientierung  $\phi(\mathbf{p})$ . Die isotrope Symmetrie  $M_{\text{iso}}(\mathbf{p})$  ist für ColorSym wie folgt definiert [Hei04]:

$$M_{\text{iso}}(\mathbf{p}) = \sum_{(l,r) \in \Gamma(\mathbf{p})} \sum_{(i,j) \in \Lambda(\mathbf{p},l,r)} c_{l,r,i,j} \quad (3.13)$$

$M_{\text{iso}}(\mathbf{p})$  betrachtet die Orientierungen  $\tilde{\varphi}_{l,r,i,j}$  nicht, sondern summiert lediglich alle Symmetriewerte  $c_{l,r,i,j}$  auf. Da keine weiteren Skalierungen vorgenommen werden, besitzt der größte Wert  $c_{l,r,i,j}$  den stärksten Einfluss auf die Doppelsumme und damit auf die Symmetrie im Punkt  $\mathbf{p}$ . Nach Abschnitt 3.2.5 bestimmt aber dieser Wert die Hauptorientierung  $\phi(\mathbf{p})$ . Dementsprechend überwiegt auch der Beitrag dieser Orientierung bei der Berechnung der isotropen Symmetrie.  $M_{\text{iso}}(\mathbf{p})$  gibt also an, wie groß der Symmetriegehalt in Richtung  $\phi(\mathbf{p})$  im Punkt  $\mathbf{p}$  ist [RWY95].

Für GraySym, das keine vergleichbare Gradientenindexmenge  $\Lambda(\mathbf{p}, l, r)$  besitzt, ist die isotrope Symmetrie  $M_{\text{iso}}^{\text{G}}(\mathbf{p})$  wie folgt definiert [RWY95]:

$$M_{\text{iso}}^{\text{G}}(\mathbf{p}) = \sum_{(l,r) \in \Gamma(\mathbf{p})} c_{l,r}^{\text{G}} \quad (3.14)$$

Da Heidemann für ColorSym keine Definition für die radiale Symmetrie angibt, wird zunächst die radiale Symmetrie für Graubilder  $M_{\text{rad}}^{\text{G}}(\mathbf{p})$  beschrieben werden, wie sie von Reisfeld et al. in [RWY95] für GraySym definiert wird. Anschließend wird eine Modifikation für ColorSym vorgeschlagen werden.

$M_{\text{rad}}^{\text{G}}(\mathbf{p})$  verrechnet die Symmetrierichtungen  $\tilde{\varphi}_{l,r}^{\text{G}}$  in Form einer Gewichtung für die Symmetriewerte  $c_{l,r}^{\text{G}}$ . Diese Gewichtung eliminiert den Symmetriewert  $c_{l,r}^{\text{G}}$  für die Hauptrichtung  $\phi^{\text{G}}(\mathbf{p})$ . Je näher eine Orientierung  $\tilde{\varphi}_{l,r}^{\text{G}}$  an der Hauptrichtung liegt, desto schwächer wird der entsprechende Symmetriewert  $c_{l,r}^{\text{G}}$  gewichtet. Die Symmetrieachsen, die senkrecht zu  $\phi^{\text{G}}(\mathbf{p})$  stehen, werden am stärksten gewichtet. Dadurch wird erreicht, dass die

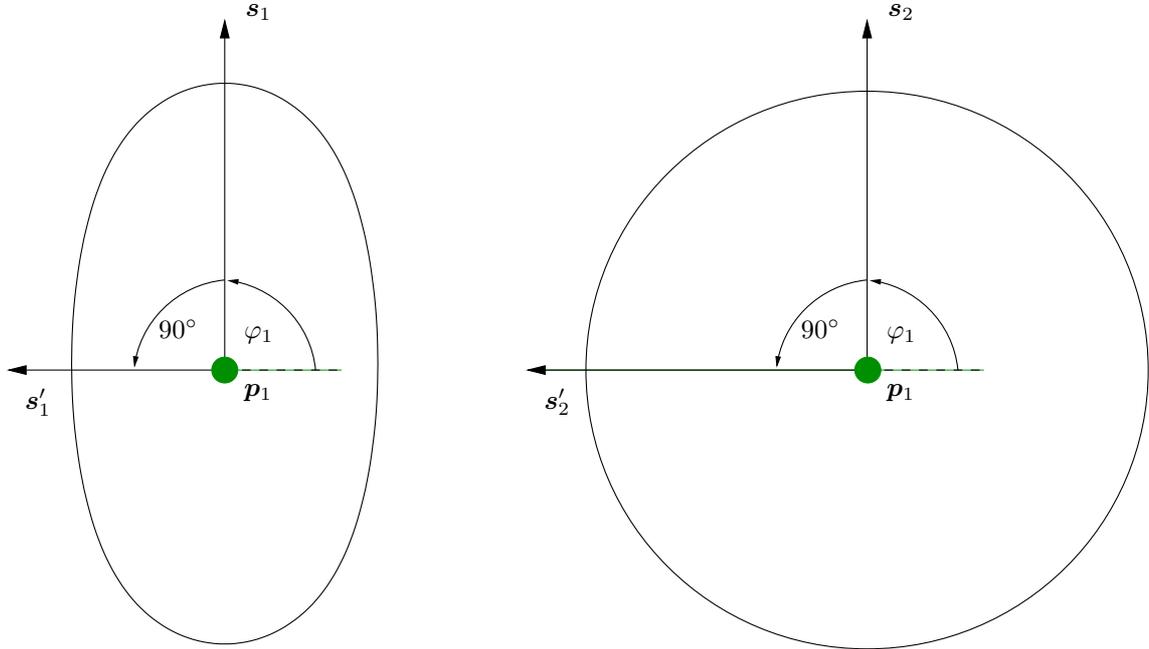


Abbildung 3.4: Skizze zur radialen Symmetrie. Die Achsen  $s_1$  und  $s_2$  geben die Hauptrichtung der Symmetrie von den abgebildeten Objekten an. Je näher eine Symmetrieachse der jeweiligen Hauptachse ist, desto geringer ist ihr Beitrag zum Symmetriehalt des Zentrums. Die Achsen  $s'_1$  und  $s'_2$  besitzen die stärkste Gewichtung, da sie orthogonal zu den Hauptachsen sind. Die radiale Symmetrie bevorzugt keine Hauptrichtung, sondern gewichtet alle Orientierungen so, dass kreisförmige Objekte eine stärkere Symmetrie aufweisen.

Symmetrie des Objektes nicht primär für eine Hauptrichtung ermittelt wird. Stattdessen werden alle Richtungen so gewichtet, dass sie einen ähnlichen Beitrag leisten. Die Symmetrie von Objekte, die symmetrisch um einen Zentrumspunkt sind, wird damit stärker hervorgehoben als die von Objekten mit einer anderen Struktur (vgl. Abschnitt 4.8). Definiert ist  $M_{\text{rad}}^G(\mathbf{p})$  wie folgt:

$$M_{\text{rad}}^G(\mathbf{p}) = \sum_{(l,r) \in \Gamma(\mathbf{p})} c_{l,r}^G \cdot \sin^2(\tilde{\varphi}_{l,r}^G - \phi^G(\mathbf{p})) \quad (3.15)$$

Eine Skizze für diesen Zusammenhang findet sich in Abbildung 3.4, ein praktisches Beispiel in Abschnitt 4.8. Ähnlich zu Gleichung 3.15 sei die radiale Symmetrie für ColorSym wie folgt definiert:

$$\mathbf{M}_{\text{rad}}(\mathbf{p}) = \sum_{(l,r) \in \Gamma(\mathbf{p})} \sum_{(i,j) \in \Lambda(\mathbf{p},l,r)} c_{l,r,i,j} \cdot \sin^2(\tilde{\varphi}_{l,r,i,j} - \phi(\mathbf{p})) \quad (3.16)$$

Die Eigenschaften von  $\mathbf{M}_{\text{rad}}(\mathbf{p})$  entsprechen dabei weitgehend denen von  $\mathbf{M}_{\text{rad}}^{\text{G}}(\mathbf{p})$  und werden daher nicht näher erläutert.

### 3.2.7 Phasengewichtungsfunktion PWF

Die Gewichtungsfunktionen bestimmen, wie sich ein konkreter Symmetriewert  $c_{l,r,i,j}$  ergibt. Als Parameter erhalten sie entsprechende Indizes, die letztlich genau ein Gradientenpaar  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  ermitteln. Bei ColorSym werden hierfür vier Parameter benötigt, bei GraySym nur zwei. Der restliche Aufbau der Gewichtungsfunktionen ist aber identisch. Im Folgenden werden daher die Gewichtungsfunktionen nur für ColorSym behandelt werden.

Die Phasengewichtungsfunktion  $\text{PWF}(\mathbf{p}, l, r, i, j)$  legt fest, ob und wie die Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  bezüglich dem Zentrum  $\mathbf{p}$  eine Symmetrie aufweisen. Dabei werden ausschließlich die Gradientenwinkel berücksichtigt. Die  $\text{PWF}(\mathbf{p}, l, r, i, j)$  nach [Hei04] ist wie folgt definiert:

$$\text{PWF}(\mathbf{p}, l, r, i, j) = \cos^2(\gamma_{l,i} + \gamma_{r,j}) \cdot \cos^2(\gamma_{l,i}) \cdot \cos^2(\gamma_{r,j}) \quad (3.17)$$

Die Winkel  $\gamma_{l,i}$  und  $\gamma_{r,j}$  sind dabei definiert wie in Abschnitt 3.2.5. Da die  $\text{PWF}(\mathbf{p}, l, r, i, j)$  eine wichtige Grundlage für das Symmetrieverständnis des gesamten Operators bildet, ist es nötig, ihre Eigenschaften näher zu untersuchen. Zum besseren Verständnis werden hierzu die Teilfunktionen  $\text{PWF}_{\text{M}}(\mathbf{p}, l, r, i, j)$  und  $\text{PWF}_{\text{P}}(\mathbf{p}, l, r, i, j)$  eingeführt. Es gilt:

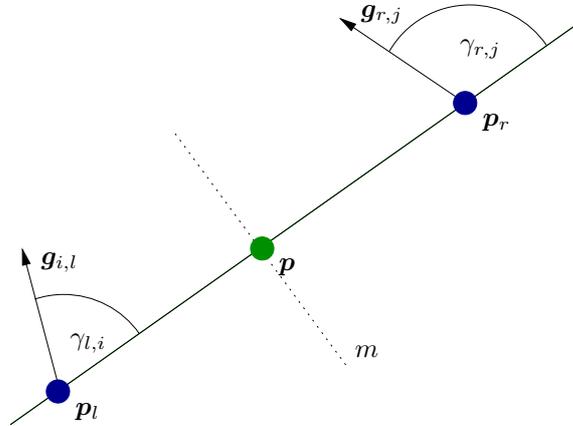


Abbildung 3.5:  $\text{PWF}_M(\mathbf{p}, l, r, i, j)$  erreicht genau dann ein Maximum, wenn  $m$  eine Spiegelachse bezüglich der beiden Winkel  $\gamma_{l,i}$  und  $\gamma_{r,j}$  darstellt. Dann gilt  $\gamma_{l,i} = 180^\circ - \gamma_{r,j}$ .

$$\text{PWF}_M(\mathbf{p}, l, r, i, j) = \cos^2(\gamma_{l,i} + \gamma_{r,j}) \quad (3.18)$$

$$\text{PWF}_P(\mathbf{p}, l, r, i, j) = \cos^2(\gamma_{l,i}) \cdot \cos^2(\gamma_{r,j}) \quad (3.19)$$

$\text{PWF}_M(\mathbf{p}, l, r, i, j)$  gibt an, wie symmetrisch die beiden Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  bezüglich einer Spiegelachse  $m$  liegen.  $m$  durchläuft dabei den Punkt  $\mathbf{p}$  und steht senkrecht auf der Geraden durch die Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$ . Perfekte Spiegelsymmetrie wird genau dann erreicht, wenn die Summe der beiden Winkel  $\gamma_{l,i}$  und  $\gamma_{r,j}$   $0^\circ$  oder  $180^\circ$  ergibt (siehe Abbildung 3.5). Je weiter die Summe von diesem idealen Wert entfernt ist, desto geringer ist die Gewichtung, die durch den Wert  $\text{PWF}_M(\mathbf{p}, l, r, i, j)$  zu erzielen ist. Das Minimum ist genau dann erreicht, wenn die beiden Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  jeweils einen Winkel von  $45^\circ$  oder von  $270^\circ$  besitzen.

$\text{PWF}_P(\mathbf{p}, l, r, i, j)$  gibt an, wie parallel die beiden Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  zur Geraden durch die Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  liegen. Ein Maximum liegt vor, wenn beide Gradientenwinkel jeweils  $0^\circ$  oder  $180^\circ$  erreichen. Je weiter die Gradientenwinkel von diesem Optimum entfernt liegen, desto geringer wird die Gewichtung durch  $\text{PWF}_P(\mathbf{p}, l, r, i, j)$ . Steht mindestens einer der Gradienten senkrecht auf der Geraden, so wird die minimale Gewichtung 0 erreicht.

Die gesamte Phasengewichtungsfunktion  $\text{PWF}(\mathbf{p}, l, r, i, j)$ , die sich aus  $\text{PWF}_M(\mathbf{p}, l, r, i, j)$  und  $\text{PWF}_P(\mathbf{p}, l, r, i, j)$  ergibt, erreicht genau dann ein Maximum, wenn die beiden Gradienten  $\mathbf{g}_{i,l}$  und  $\mathbf{g}_{r,j}$  sowohl eine optimale Spiegelsymmetrie bezüglich der Achse  $m$  und zugleich eine optimale Parallelität bezüglich der Geraden durch die Punkte  $\mathbf{p}_l$  und  $\mathbf{p}_r$  aufweisen. Dies ist genau dann der Fall, wenn die Summe der beiden Winkel  $\gamma_{l,i}$  und  $\gamma_{r,j}$   $0^\circ$  oder  $180^\circ$  ergibt. Diese Bedingung trifft zu, wenn beide Gradienten entweder in dieselbe Richtung zeigen, exakt auf den Zentrumspunkt  $\mathbf{p}$  verweisen oder von diesem weg weisen.

Die Quadrierung der cos-Funktionen sorgt bei den Gradienten für eine Rotationsinvarianz um  $180^\circ$ . Damit ist es nicht relevant, ob ein Gradientenwinkel den Wert  $\beta$  oder  $\beta + 180^\circ$  hat. Heidemann weicht damit in [Hei04] von der Phasengewichtungsfunktion von Reisfeld et al. ab, wo diese Quadrierung nicht vorhanden ist [RWY95]. Diese Änderung bewirkt, dass auch Symmetrien von Objekten gefunden werden können, die sowohl an hellere als auch an dunklere Bereiche angrenzen.

### 3.2.8 Gradientengewichtungsfunktion GWF

Die Gradientengewichtungsfunktion  $\text{GWF}(\mathbf{p}, l, r, i, j)$  entspricht einer Gewichtung auf Basis der Gradientenintensitäten  $\mathbf{G}_i(\mathbf{p}_l)$  und  $\mathbf{G}_j(\mathbf{p}_r)$ . Sie ist wie folgt definiert:

$$\text{GWF}(\mathbf{p}, l, r, i, j) = \log(1 + \mathbf{G}_i(\mathbf{p}_l)) \cdot \log(1 + \mathbf{G}_j(\mathbf{p}_r)) \quad (3.20)$$

Durch die logarithmische Skalierung wird verhindert, dass große Intensitätswerte den Symmetriewert  $c_{l,r,i,j}$  zu stark beeinflussen.

### 3.2.9 Distanzgewichtungsfunktion $\text{DWF}_\sigma$

Die Distanzgewichtungsfunktion  $\text{DWF}_\sigma(\mathbf{p}, l, r, i, j)$  gewichtet die relative Distanz zwischen den beiden Punkten  $\mathbf{p}_l$  und  $\mathbf{p}_r$ . Je weiter diese voneinander entfernt sind, desto geringer ist ihr Beitrag zum Symmetriewert  $c_{l,r,i,j}$ . Die  $\text{DWF}_\sigma(\mathbf{p}, l, r, i, j)$  entspricht nach [Hei04, RWY95] der Gaußfunktion und ist wie folgt definiert:

$$\text{DWF}_\sigma(\mathbf{p}, l, r, i, j) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{\|\mathbf{p}_l - \mathbf{p}_r\|}{2\sigma^2}\right) \quad (3.21)$$

Die Standardabweichung  $\sigma$  bildet einen Parameter der  $\text{DWF}_\sigma(\mathbf{p}, l, r, i, j)$ . Je größer  $\sigma$  ist, desto stärker ist die Gewichtung eines Symmetriepaares  $(\mathbf{p}_l, \mathbf{p}_r)$ , deren Punkte weit voneinander entfernt sind. Die Punkte, deren Distanz größer ist als  $3\sigma$ , erhalten hingegen nur eine sehr geringe Gewichtung.  $\sigma$  bestimmt daher neben  $d_{\max}$  die Größe der Nachbarschaft um den Punkt  $\mathbf{p}$  (vgl. Abschnitt 3.2.2).  $d_{\max}$  gewichtet ein Paar  $(\mathbf{p}_l, \mathbf{p}_r)$  nur binär, d. h.  $(\mathbf{p}_l, \mathbf{p}_r)$  liegt entweder in der Nachbarschaft oder nicht.  $\sigma$  lässt sich als kontinuierliche Nachbarschaft verstehen: Je weiter ein Paar  $(\mathbf{p}_l, \mathbf{p}_r)$  vom Zentrum  $\mathbf{p}$  entfernt ist, desto weniger gehört es zur Nachbarschaft von  $\mathbf{p}$ . Das Verhältnis zwischen  $\sigma$  und  $d_{\max}$  bestimmt, welcher dieser beiden Werte einen stärkeren Einfluss auf die Größe der Nachbarschaft hat.

Aus Effizienzgründen schlägt Heidemann in [Hei04] für die isotrope Symmetrie vor, die Distanzgewichtungsfunktion aus der Berechnung zu entfernen und stattdessen die resultierende Symmetriekarte  $M_{\text{iso}}$  mit einem Gaußkern zu falten. Das Resultat sei laut Heidemann in beiden Fällen eine leichte Weichzeichnung der Symmetriekarte. Dies konnte zwar in Testdurchläufen bestätigt werden, jedoch unterschieden sich die beiden Resultate leicht voneinander. Dies gilt vor allem bei der radialen Symmetrie, wo  $\text{DWF}_\sigma(\mathbf{p}, l, r, i, j)$  die Orientierung  $\phi(\mathbf{p})$  und damit die Berechnung von  $M_{\text{rad}}(\mathbf{p})$  beeinflusst. Da sich in Tests allerdings gezeigt hat, dass die Konvolution mit einem Gaußkern bessere Ergebnisse liefert als die Distanzgewichtungsfunktion, wird im Folgenden auch die Konvolution verwendet werden.  $\sigma$  bleibt dabei als Parameter vorhanden, da auch der Gaußkern eine solche Standardabweichung besitzt.

### 3.2.10 Algorithmus

Der finale Algorithmus von ColorSym ist stark an die Gleichungen zu den Symmetriekarten aus Abschnitt 3.2.6 angelehnt. Auch hier gilt es wieder, zwischen isotroper und radialer Symmetrie zu unterscheiden. Die Grundlage des Algorithmus bildet in beiden Fällen die Gradientenkarte  $(G, \Theta)$ . Nach deren Berechnung werden für jeden Bildpunkt  $\mathbf{p}$  durch die Indexmengen  $\Gamma(\mathbf{p})$  und  $\Lambda(\mathbf{p}, l, r)$  alle Gradientenpaare  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  bestimmt, die zur Be-

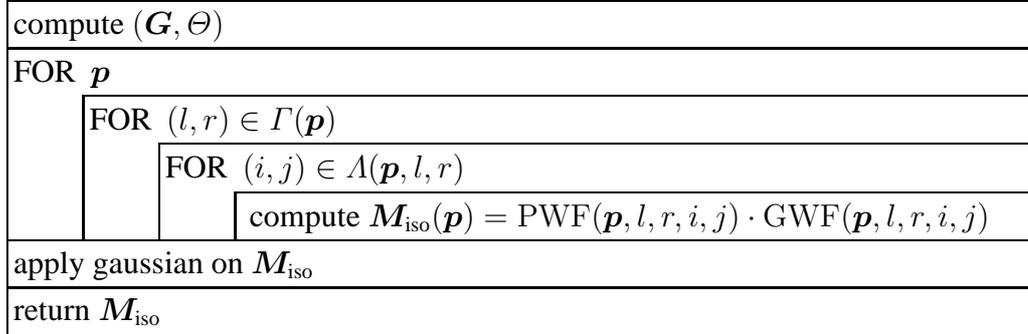


Abbildung 3.6: Vereinfachtes Struktogramm zur Berechnung der isotropen Symmetrie. Abgebildet ist der Algorithmus von ColorSym.

rechnung verwendet werden. Für jedes solche Paar wird dann ein Symmetriewert  $c_{l,r,i,j}$  errechnet. Die Summe aller dieser Werte ergibt den vorläufigen Symmetriegehalt an der Position  $\mathbf{p}$ . Nachdem ein solcher Wert für alle Bildpunkte  $\mathbf{p}$  berechnet worden ist, wird die vorläufige Symmetriekarte mit einem Gaußkern gefaltet. Das Resultat dieser Konvolution entspricht der endgültigen Symmetriekarte, deren Maxima die Interessensregionen bestimmen.

Das Struktogramm für die isotrope Symmetrie findet sich in Abbildung 3.6, das Struktogramm für die radiale Symmetrie ist in Abbildung 3.7 abgebildet. Dargestellt sind jeweils die Algorithmen für ColorSym. Für GraySym ergibt sich ein ähnliches Struktogramm, jedoch ohne die Gradientenindexmenge  $\Lambda(\mathbf{p}, l, r)$ . Da die Implementierung von GraySym aber nicht das primäre Ziel dieser Arbeit ist, soll es an dieser Stelle auch nicht durch ein Struktogramm dargestellt werden.

Die Unterschiede zwischen der isotropen und der radialen Symmetrie liegen in der Verrechnung der Symmetrieorientierungen  $\tilde{\varphi}_{l,r}^G$  (siehe Abschnitt 3.2.5). Diese und die Hauptorientierung  $\phi(\mathbf{p})$  werden bei der radialen Symmetrie zur Gewichtung der Symmetriewerte  $c_{l,r,i,j}$  verwendet. Der Algorithmus verwendet dazu einige temporäre Variablen als Zwischenspeicher, die sich auch im Struktogramm wiederfinden.

Der wichtigste Parameter für ColorSym ist der Maximalradius  $d_{\text{max}}$  (vgl. Abschnitt 3.2.2). Dieser bestimmt den Inhalt der Indexmenge  $\Gamma(\mathbf{p})$  und gibt damit an, aus welchem Bereich die Symmetriepaare  $(\mathbf{p}_l, \mathbf{p}_r)$  verwendet werden sollen. Zur kompletten Erfassung

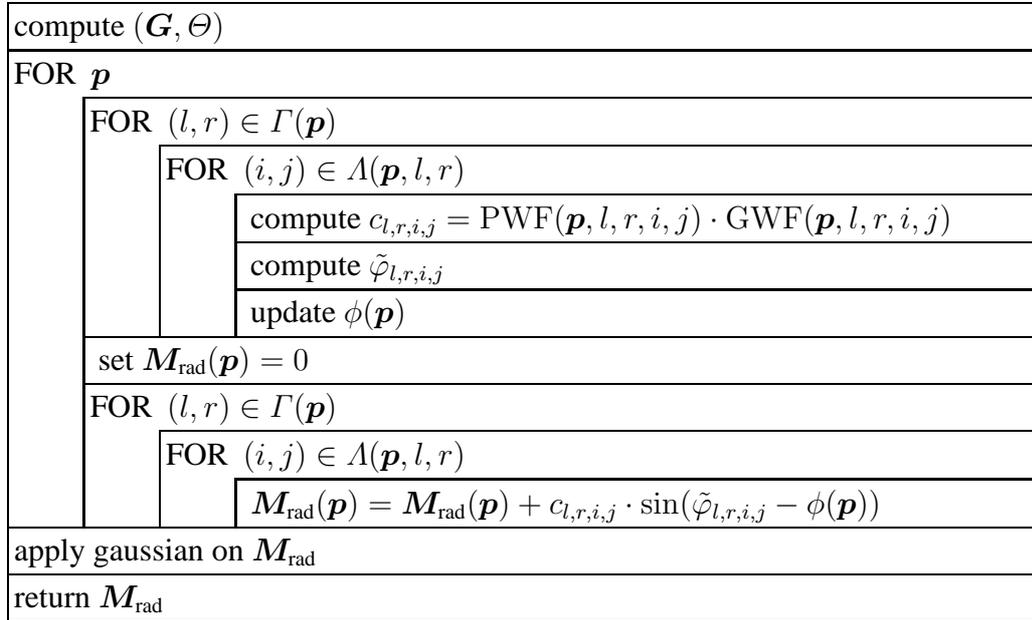


Abbildung 3.7: Vereinfachtes Struktogramm zur Berechnung der radialen Symmetrie. Abgebildet ist der Algorithmus von ColorSym.

eines symmetrischen Objektes muss  $d_{\text{max}}$  mindestens so groß gewählt werden, dass ein gedachter Kreis mit dem Radius  $d_{\text{max}}$  um das reale Symmetriezentrum des Objekts dieses vollständig abdeckt (siehe Abbildung 3.8). Ein solcher Kreis markiert all diejenigen Punkte, die zur Berechnung der Symmetrie verwendet werden. Wird das Objekt nicht vollständig von dem Kreis abgedeckt, so kann nicht die vollständige Symmetrie berechnet werden. Die resultierende Symmetriekarte ist dann ggf. unbrauchbar.

Weitere Parameter bilden die minimalen Gradientenintensitäten  $\vartheta_0$ ,  $\vartheta_1$  und  $\vartheta_2$ , die für die Definition der Gradientenindexmenge  $\Lambda(\mathbf{p}, l, r)$  benötigt werden (vgl. 3.2.3). Diese dienen dazu, solche Gradienten herauszufiltern, die anfällig gegen Rauschen sind. I. d. R. können diese Schwellwerte unabhängig vom jeweiligen Eingabebild fest gesetzt werden. Der letzte Parameter ist die Standardabweichung  $\sigma$ . Diese wird zum Erstellen des Gaußkerns verwendet und bestimmt, wie stark die Symmetriekarte weichgezeichnet werden soll.

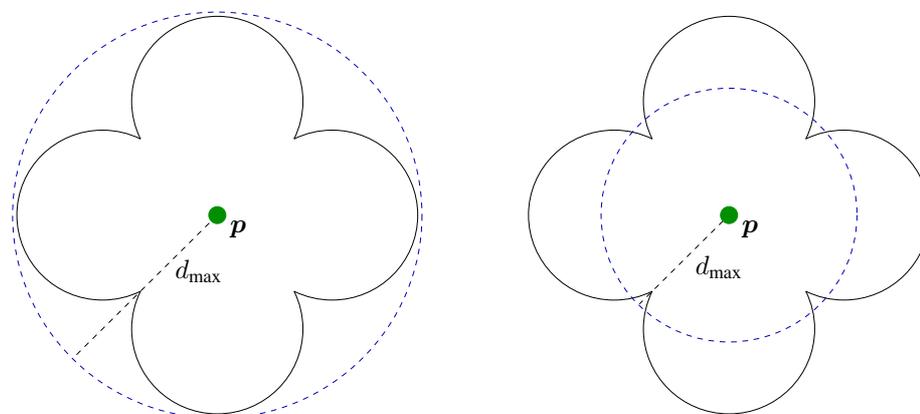


Abbildung 3.8: Es gilt, die Symmetrie des Vierpasses zu ermitteln. Der Parameter  $d_{\max}$  bestimmt, welche Punkte zur Berechnung der Symmetrie verwendet werden sollen. Diese Punkte befinden sich in einem gedachten Kreis mit dem Radius  $d_{\max}$  um das Zentrum  $p$ . Ist  $d_{\max}$  zu klein, so kann ggf. nicht die gesamte Symmetrie des Objekts erfasst werden. Im rechten Bild wird nur ein begrenzter Teil des Vierpasses betrachtet, der Rest des Objektes bleibt für das Zentrum  $p$  unberücksichtigt.  $d_{\max}$  sollte daher stets an die Größe der zu suchenden Objekte angepasst werden, wie im linken Bild geschehen.

## 3.3 Einbettung in PUMA 2

### 3.3.1 Voraussetzungen

Das Ziel dieser Arbeit ist es, das unter Abschnitt 3.2 beschriebene Verfahren als Operator in PUMA 2 einzubinden. PUMA 2 (*Programmierungsumgebung für die Musteranalyse*) ist ein in der Programmiersprache C++ geschriebenes Framework für Aufgaben aus dem Bereich der Bildverarbeitung. Es wird entwickelt von der Arbeitsgruppe Aktives Sehen an der Universität Koblenz-Landau, Campus Koblenz. PUMA 2 ist der Nachfolger von PUMA und verfolgt verstärkt ein objektorientiertes Konzept mit einer umfassenden Klassenhierarchie. Diese deckt sowohl Klassen zum Laden, Speichern und Repräsentieren von Bildern als auch Klassen für Bildoperationen ab. Zur Zeit, als diese Arbeit geschrieben wurde, befand sich PUMA 2 noch im Entwicklungsstadium und war dementsprechend nicht vollständig ausgereift.

Die Grundlage für die Einbettung von ColorSym in PUMA 2 bildete eine von Gunther Heidemann bereit gestellte Implementierung in der Programmiersprache C. Diese liegt im ANSI-Standard X3.159-1989 vor<sup>2</sup> und umfasst die C-Dateien `color_sym.c` und `color_sym_routines.c` sowie das Shell-Script `do_color_sym`. Diese drei Dateien bilden eine Demo-Applikation, deren Bedienung Heidemann in einer ergänzenden README erläutert.<sup>3</sup> Die Datei `color_sym_routines.c` bildet ein Fragment eines größeren Softwarepakets. Sie stellt sämtliche Funktionen zur Verfügung, die für die Implementierung von ColorSym sowie das Laden und Speichern von Bilddateien notwendig sind. Die Datei besitzt keine ausgereifte Struktur, sondern entspricht einer ungeordneten Sammlung von Funktionen, wie dies auch in der README beschrieben wird. Diese Funktionen werden aufgerufen von `color_sym.c`, das die eigentliche Demo-Applikation enthält. Das Programm lädt zunächst ein Bild und erzeugt für dieses mit ColorSym eine isotrope Symmetriekarte. Das Script `do_color_sym` ruft diese Demo-Applikation mit vordefinierten Standardwerten auf. Durch die verständliche Kommentierung der Quellcode-dateien bildete Heidemanns Implementierung unter Zuhilfenahme seiner Publikation eine solide Grundlage zur weiteren Arbeit.

### 3.3.2 Implementierung

Zur Einbettung von ColorSym wurde Heidemanns Quellcode weitgehend unverändert übernommen. Der Großteil der Modifikationen war lediglich strukturell bedingt und befasste sich mit der Änderung der Interfaces. Die implementierten Klassen dienen überwiegend als Wrapperklassen, die den eigentlichen Programmkern in die Klassenhierarchie von PUMA 2 integrieren. Hierbei zeigten sich jedoch einige Schwierigkeiten, da die Struktur von PUMA 2 an manchen Stellen noch nicht ganz ausgereift war. Um das Einbetten eines Operators für ColorSym möglich zu machen, mussten daher noch einige zusätzliche Klassen implementiert werden. Diese werden in Anhang B kurz beschrieben.

---

<sup>2</sup>Eine Kompilierung der Quellcodedateien ist unter Verwendung dieses Standards fehlerfrei möglich.

<sup>3</sup>Die vier Dateien `color_sym.c`, `color_sym_routines.c`, `do_color_sym` und README finden sich auch auf der beiliegenden CD-ROM zur Druckversion dieser Arbeit.

Der Algorithmus von ColorSym wurde durch drei Klassen in PUMA 2 eingebettet. Die Klasse `ColorSymmetryOperatorRGB8` implementiert einen Bild-zu-Bild-Operator, der ein Farbbild als Eingabe erwartet und eine Symmetriekarte als Ausgabe liefert. Die Klasse `ColorSymmetryOperatorProRGB8` bildet ebenfalls einen Operator, berechnet aber zu einem Eingabebild mehrere Symmetriekarten. Diese werden durch ihren Maximalradius  $d_{\max}$  voneinander unterschieden (vgl. Gleichung 3.4). Gemeinsamkeiten beider Klassen werden in der Superklasse `ColorSymmetryOperatorBaseRGB8` zusammengefasst. Ein vereinfachtes Klassendiagramm, das diesen Zusammenhang darstellt, findet sich in Abbildung 3.9. Alle erstellten Symmetriekarten werden durch Instanzen der Klasse `ScalarFeatureImage<float>` dargestellt (siehe Anhang B). Sämtliche Parameter für ColorSym, die in Abschnitt 3.2.10 erläutert werden, finden sich – ggf. unter einer anderen Bezeichnung – auch in den Klassen wieder. Eine genaue Beschreibung aller Parameter findet sich im Quelltext der jeweiligen Dateien. Im Folgenden soll hingegen nur kurz auf den Parameter `pwfType` eingegangen werden.

Der Parameter `pwfType` gibt an, wie die Symmetrieorientierungen  $\tilde{\varphi}_{l,r,i,j}$  für die Berechnung der Symmetriekarte verwendet werden sollen. `pwfType` bestimmt dabei zum einen, welche Phasengewichtungsfunktion verwendet werden soll. Implementiert sind die Originalfunktion von Reisfeld et al. [RWY95] und Heidemanns Modifikation, die in Gleichung 3.17 beschrieben wird. Ferner legt `pwfType` fest, ob die isotrope oder die radiale Symmetrie berechnet werden soll (vgl. Abschnitt 3.2.6). Zur Zeit finden sich in der Implementierung fünf verschiedene Modi von `pwfType` wie folgt:

`pwfType = 0` verwendet die Phasengewichtungsfunktion von Reisfeld et al. [RWY95].

Diese wird auf jeden Farbkanal separat angewendet, d. h. die Indexmenge  $\Lambda(\mathbf{p}, l, r)$  wird nicht verwendet. Berechnet wird die isotrope Symmetrie. Wird ein Bild mit drei identischen Kanälen als Eingabe verwendet, entspricht dieser Modus dem Algorithmus von GraySym nach [RWY95].

`pwfType = 1` verwendet ebenfalls die Phasengewichtungsfunktion von Reisfeld et al.

Im Unterschied zu Modus 0 wird jedoch die Indexmenge  $\Lambda(\mathbf{p}, l, r)$  verwendet (vgl. Gleichung 3.5). Berechnet wird die isotrope Symmetrie  $M_{\text{iso}}$ , wie sie im Struktogramm in Abbildung 3.6 dargestellt wird.

`pwfType = 2` verwendet die von Heidemann vorgeschlagene Phasengewichtungsfunktion aus Gleichung 3.17 unter Verwendung der Indexmenge  $\Lambda(\mathbf{p}, l, r)$ . Berechnet wird die isotrope Symmetrie  $M_{\text{iso}}$ . Dieser Modus entspricht dem Algorithmus von ColorSym nach [Hei04].

`pwfType = 3` verwendet die von Heidemann vorgeschlagene Phasengewichtungsfunktion unter der Verwendung der Indexmenge  $\Lambda(\mathbf{p}, l, r)$ . Berechnet wird die radiale Symmetrie  $M_{\text{rad}}$ , wie sie im Struktogramm in Abbildung 3.7 dargestellt wird.

`pwfType = 4` bildet eine Kombination der Modi 2 und 3. Die beiden resultierenden Symmetriekarten  $M_{\text{iso}}$  und  $M_{\text{rad}}$  werden in Anlehnung an eine Funktion des freien Grafikprogramms GIMP<sup>4</sup> zum Ebenenüberblenden<sup>5</sup> miteinander verrechnet. Dazu werden zunächst beide Symmetriekarten so auf einen Wertebereich  $[0, \dots, v_{\text{max}}]$  skaliert, dass der maximale Wert einer jeden Karte dem Wert  $v_{\text{max}}$  entspricht. Die Symmetriekarte  $M_{\text{pwf4}}$  wird dann wie folgt berechnet:

$$M_{\text{pwf4}}(\mathbf{p}) = \begin{cases} M_{\text{iso}}(\mathbf{p}) + M_{\text{rad}}(\mathbf{p}), & \text{wenn } M_{\text{iso}}(\mathbf{p}) + M_{\text{rad}}(\mathbf{p}) \geq \frac{v_{\text{max}}}{2} \\ 0, & \text{sonst} \end{cases}$$

Diese Verrechnung bildet lediglich einen Test und bedarf daher noch einigen Verbesserungen.

Die ersten drei Modi wurden bereits von Heidemann in seinem Quellcode zur Verfügung gestellt. Neu hinzugefügt wurden lediglich die letzten beiden Modi.

Zum Testen beider Bildoperatoren wurde die Demoapplikation `colorSymmetryDemo` implementiert. Diese ermöglicht das Berechnen von Interessenspunkten mit ColorSym auf beliebigen Bildern. Nach dem Laden eines Bildes unter Verwendung weiterer Klassen von PUMA 2 wird der Algorithmus von ColorSym angewendet. Die resultierenden Symmetriekarte wird als Bild gespeichert. Ferner werden die Interessensregionen als lokale Maxima der Symmetriekarte ermittelt und in einer Textdatei abgelegt. Diese speichert Position

---

<sup>4</sup> <http://www.gimp.org>

<sup>5</sup>Die Funktion trägt die Bezeichnung *Faser mischen* (*grain merge*) und wird unter den Ebenenmodi (Layer Modes) kurz beschrieben.

und Symmetriewert eines jeden Interessenspunktes. Weitere Informationen zur Bedienung sowie zur Ausgabe des Programmes finden sich in Anhang [A](#).

Die Implementierung von ColorSym stellt lediglich einen Prototypen dar. Obgleich Heide-  
mann in seinem Quellcode durch die Verwendung von Look-Up-Tabellen eine möglichst  
effiziente Implementierung anstrebt, kann diese durch geringe Modifikationen bereits re-  
duziert worden sein. Die Implementierungen für die radiale Symmetrie wurden gar nicht  
optimiert und besaßen in den Tests daher eine äußerst lange Laufzeit, die je nach Eingabe-  
paramatern und Größe des Eingabebildes bis zu einigen Stunden entsprach. Aufgrund der  
fehlenden Optimierungen wird in dieser Arbeit auch von einer Untersuchung des Zeitauf-  
wands abgesehen.

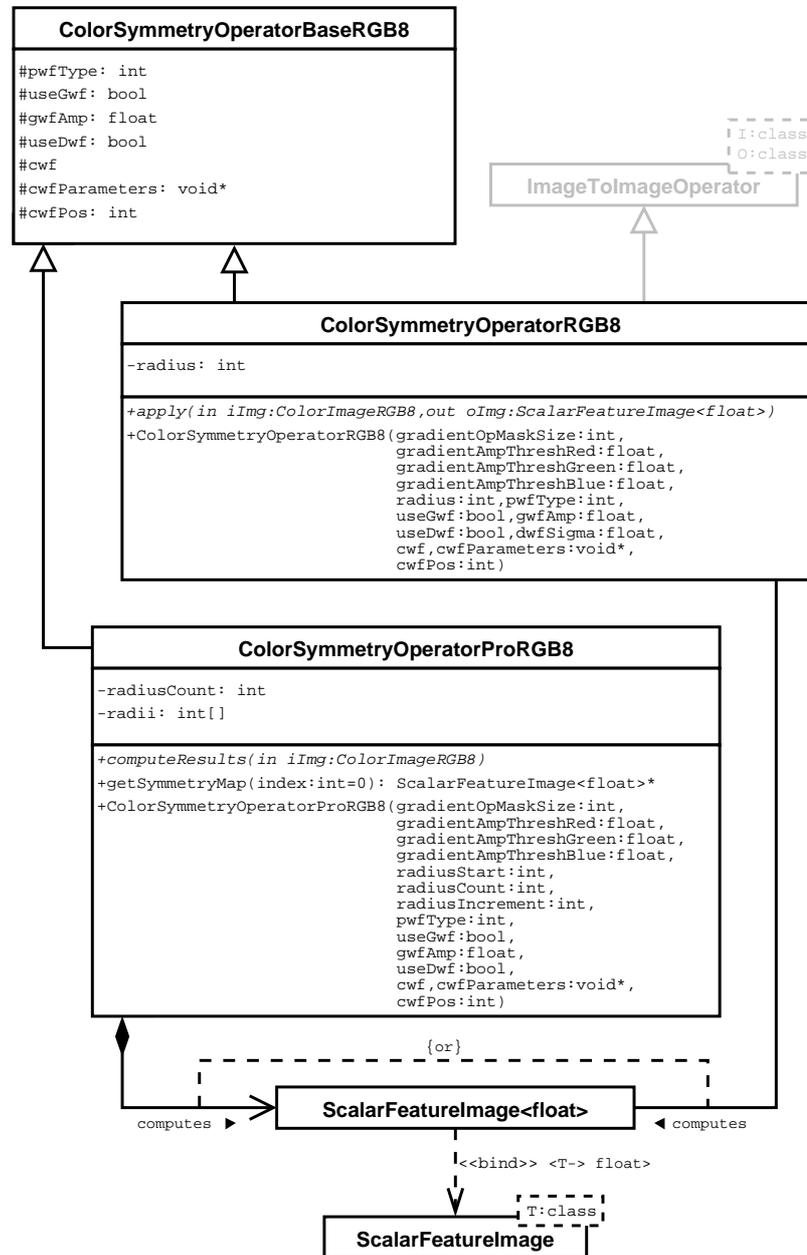


Abbildung 3.9: Klassendiagramm um die Klasse `ColorSymmetryOperatorBase`. Grau unterlegte Klassen waren bereits in PUMA 2 vorhanden und wurden nicht modifiziert. Sie dienen lediglich zur Orientierung. Eine genaue Beschreibung der abgebildeten Attribute findet sich im Quelltext der jeweiligen Klassen. Weitere Erläuterungen zur Klasse `ScalarFeatureImage` finden sich in Anhang B.

# Kapitel 4

## Experimente und Ergebnisse

Ziel dieses Kapitels ist es, das vorgestellte Verfahren zur Berechnung der Farbsymmetrie sowohl auf Basis theoretischer Untersuchungen als auch mit praktischen Beispielen näher zu untersuchen und zu reflektieren.

### 4.1 Unterscheidung von Symmetriegruppen

ColorSym ist ein universelles Verfahren, das ohne spezifisches Vorwissen über die zu verarbeitenden Bilder in vielen Bereichen Anwendung finden kann. Nach Heidemann sind die von ColorSym gefundenen Interessensregionen invariant gegenüber Translationen, Rotationen und Beleuchtungsänderungen [Hei04]. Ferner konnte Heidemann für die gefundenen Interessensregionen eine hohe Unterscheidbarkeit nachweisen. Die genannten Eigenschaften machen ColorSym damit potentiell brauchbar für die Objekterkennung. Offen bleibt aber, wie weit es sich im Speziellen für die Lokalisierung von Ornamenten einsetzen lässt.

Um dies besser beurteilen zu können, werden vier Symmetriegruppen unterschieden. Die Grundlage dieser Klassifikation bilden die drei Symmetrieeoperationen Translation, Rotation und Spiegelung. Eine Kombination mehrerer Symmetrieeoperationen ist ebenfalls möglich. Die vier Gruppen sind wie folgt (vgl. hierzu auch [FFM90, Den05]):

**Wirbelräder** sind rotationssymmetrische, runde Strukturen. Ein  $n$ -speichiges Wirbelrad wird durch eine Rotation um ein Vielfaches von  $\frac{360^\circ}{n}$  auf sich selbst abgebildet. Ein Beispiel eines Wirbelrades findet sich in Abbildung 4.12.

**Vielpässe** sind sowohl spiegelsymmetrisch als auch rotationssymmetrisch. Ein Vielpass besteht aus  $n$  Blättern und besitzt  $n$  Spiegelachsen. Er wird durch eine Rotation um ein Vielfaches von  $\frac{360^\circ}{n}$  auf sich selbst abgebildet. Ein Beispiel findet sich in Abbildung 4.9. Die häufigsten Vielpässe bilden Vierpässe mit  $n = 4$ .

**Friesensymmetrien** sind Strukturen, die durch Translation in eine Dimension auf sich selbst abgebildet werden. Zusätzlich können sie noch weitere Symmetrien besitzen. Die wiederholte Struktur kann dabei wiederum ein Vielpass oder ein Wirbelrad sein.

**Flächensymmetrien** sind translationsinvariant bezüglich zwei linear unabhängiger Dimensionen. Ähnlich zu Friesensymmetrien können sie zusätzlich noch andere Symmetrieeigenschaften besitzen und aus kleineren symmetrischen Strukturen bestehen. Ein Beispiel für eine Flächensymmetrie zeigt Abbildung 4.14.

ColorSym unterscheidet als universelles Verfahren diese Symmetriegruppen nicht und kann sich daher auch nicht automatisch an die Eigenschaften der jeweiligen Gruppe anpassen. Entsprechend können die berechneten Symmetriekarten für einige der Gruppen unbrauchbar sein.

## 4.2 Vorgehen bei den Experimenten

In den folgenden Abschnitten soll die praktische Verwendbarkeit von ColorSym untersucht werden. Dabei wird unterschieden zwischen der Objekterkennung im Allgemeinen und der speziellen Anwendung im Projekt Bildanalyse zur Ornamentklassifikation [SSP06]. Die Symmetriekarten der Testbilder wurden mit Hilfe der Demo-Applikation `colorSymmetryDemo` erstellt (vgl. Anhang A). Als Testbilder wurden sowohl natürliche als auch künstliche Bilder verwendet. Dadurch soll die nahezu ideale Symmetrie, wie sie bei den künstlichen Bildern vorhanden ist, mit der in der Praxis vorhandenen Sym-

metrie natürlicher Bilder verglichen werden. Ferner wurde für jede der in Abschnitt 4.1 beschriebenen Symmetriegruppe mindestens ein Testbild verwendet.

Für einige Bilder wurde sowohl eine radiale Symmetriekarte als auch eine isotrope Symmetriekarte berechnet (vgl. hierzu die Abschnitte 3.2.7 und 3.3.2). Umgesetzt wurde dies mit dem Parameter `pwfType`. Neben diesem wurde zur Berechnung lediglich der Parameter  $d_{\max}$  variiert, der die maximale Größe des zu suchenden Symmetriebereiches angibt. Die Parameter  $\vartheta_0$ ,  $\vartheta_1$  und  $\vartheta_2$ , die die minimalen Gradientenintensitäten bestimmen, wurden für alle Berechnungen mit einem festen Wert von 10 versehen. Die Standardabweichung  $\sigma$ , die für die anschließende Konvolution mit dem Gaußkern benötigt wird, wurde mit dem ebenfalls festen Wert 1.5 belegt (vgl. Abschnitt 3.2.10). Eine Distanzgewichtungsfunktion sowie eine Farbgewichtungsfunktion wurden nicht verwendet (vgl. die Abschnitte 3.2.4 und 3.2.9).

Zur Bestimmung der Interessensregionen wurden für jede Symmetriekarte die lokalen Maxima ermittelt. Dabei wurde die Methode `getLocalMaxima(T thresh)` der Klasse `FeatureImage` verwendet, deren Arbeitsweise in Anhang B beschrieben wird.

## 4.3 Allgemeine Funktion

`ColorSym` ist ein universelles Verfahren zur Merkmalsextraktion. Dementsprechend ist es nötig, zunächst die Tauglichkeit des Verfahrens für allgemeine Bilder zu testen. Das Symmetrieverständnis von `ColorSym` basiert auf der Nachbarschaftsindexmenge  $\Gamma(\mathbf{p})$  aus Abschnitt 3.2.2 und der Phasengewichtungsfunktion  $\text{PWF}(\mathbf{p}, l, r, i, j)$  aus Abschnitt 3.2.7.  $\Gamma(\mathbf{p})$  legt fest, dass zur Berechnung der Symmetrie im Punkt  $\mathbf{p}$  mehrere Paare von Punkten verwendet werden, die jeweils denselben Abstand zu  $\mathbf{p}$  haben.  $\text{PWF}(\mathbf{p}, l, r, i, j)$  gewichtet die Gradientenwinkel dieser Punkte. Ein Maximum in dieser Gewichtung wird etwa dann erreicht, wenn beide Gradienten von  $\mathbf{p}$  weg weisen. Eine geometrische Figur, bei der diese Voraussetzungen gegeben sind, ist der Kreis, bei dem alle Randpunkte vom Zentrum weg zeigen. Dies schlägt sich entsprechend in der Phasengewichtungsfunktion nieder. Abbildung 4.1 skizziert diesen Zusammenhang.

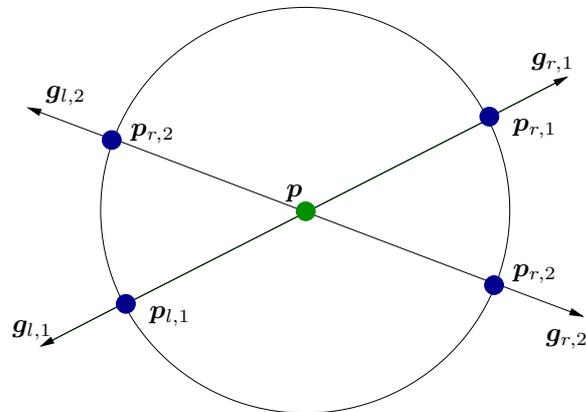


Abbildung 4.1: Ein Kreis bildet aufgrund seiner Struktur die idealen Voraussetzungen, um von ColorSym gefunden zu werden. Durch die paarweise Betrachtung der Nachbarpunkte vom Zentrumspunkt  $p$  werden auch paarweise die Randpunkte des Kreises betrachtet. Deren Gradienten weisen jeweils vom Zentrum weg und erreichen damit ein Maximum in der Phasengewichtungsfunktion.

Abbildung 4.2 zeigt eine künstliche Szene von vier rosa<sup>1</sup> Kreisen auf einem bunten Hintergrund. Die berechnete Symmetriekarte bestätigt, dass kreisförmige Strukturen von ColorSym gefunden werden können. Je heller ein Punkt in der Symmetriekarte ist, desto symmetrischer ist die entsprechende Position im Originalbild. Die vier größten Werte der Symmetriekarte bestimmen hierbei die Interessensregionen und sind identisch mit den vier Kreiszentren.

Der schwarze Rand einer jeden Symmetriekarte ergibt sich durch den Parameter  $d_{\max}$ . Eine gesonderte Betrachtung der Randpunkte gibt es bei ColorSym nicht, da viele bekannten Verfahren zur Randbehandlung das Ergebnis verfälschen können. Ein mögliches und häufig praktiziertes Vorgehen bei einer solchen Randbehandlung ist das Erweitern des Bildes durch Kopier- und Spiegeloperationen. Das Bild wird dabei an jedem Rand gespiegelt, so dass möglichst fließende Übergänge zwischen den jeweiligen Bereichen entstehen. Für die ursprünglichen Randpunkte stehen damit genügend Nachbarpunkte zur Verfügung, die zur weiteren Berechnung verwendet werden können. Eine solche Randbehandlung fügt aber

<sup>1</sup>Die genaue Farbgebung entspricht der RGB-Farbe (255, 0, 192) und ist damit eigentlich eine Mischung aus rosa und pink.

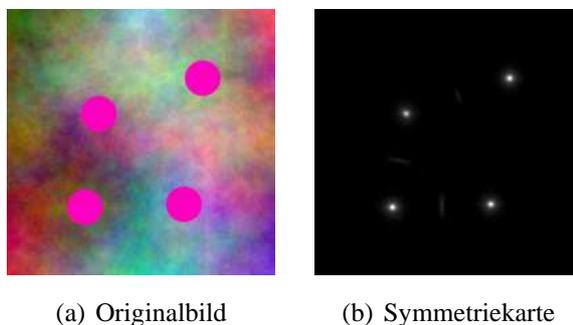


Abbildung 4.2: Rosa Kreise auf buntem Grund. Die Kreise besitzen einen Durchmesser von etwa 26 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 28$  und  $\text{pwfType} = 2$  gesetzt.

eine künstliche Symmetrie hinzu, die das Ergebnis von ColorSym verfälscht. Die Randpunkte weisen dann einen hohen Symmetriegehalt auf, der im Ursprungsbild aber nicht vorhanden ist. Jede weitere Veränderung der Bildinformation hätte einen ähnlichen Effekt. Das Hinzufügen eines monochromen Randes zum Bild würde zwar die Berechnung nicht beeinflussen, bringt aber aufgrund der fehlenden Gradienten in Bereichen keine weiteren Vorteile zur Berechnung (vgl. Abschnitt 3.2.8) und ist daher ebenfalls nicht zu empfehlen.

Eine wesentlich komplexere Szene zeigt ein Foto eines Schilderwalds der Stadt Passau<sup>2</sup> (Abbildung 4.3). Jedes abgebildete Verkehrsschild besitzt ein hohes Maß an Symmetrie. Durch die Wahl eines entsprechenden Maximalradius  $d_{\max}$  konnten drei der Verkehrsschilder gefunden werden. Werden weitere Symmetriekarten mit anderen Maximalradien berechnet, so lassen sich auch die restlichen Verkehrszeichen bzw. deren Symmetriezentren finden.

Die Resultate beider Testbilder zeigen, dass sich ColorSym im Allgemeinen durchaus zur Objekterkennung einsetzen lässt. Das Verfahren ist in der Lage, die Symmetriezentren symmetrischer Objekte zu lokalisieren und damit markante Merkmale zu ermitteln, die sich für weitere Verarbeitungsschritte verwenden lassen. Bei komplexeren Objekten, die keine (einfache) Symmetrie aufweisen, kann ColorSym aber ggf. unbrauchbare Ergebnisse liefern. Dies zeigten etwa Tests mit Labyrinth-ähnlichen Mustern oder Muster von

<sup>2</sup>Quelle: <http://de.wikipedia.org/wiki/Verkehrszeichen>, Stand vom 12.09.2007.



(a) Originalbild



(b) Symmetriekarte

Abbildung 4.3: Schilderwald. Die beiden Schilder zum Halteverbot im Vordergrund haben eine Größe von etwa  $35 \times 45$  Pixeln. Das Schild zur Geschwindigkeitsbegrenzung im Vordergrund hat einen Innenradius von 45 Pixeln und einen Außenradius von 64 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 45$  und  $\text{pwfType} = 2$  gesetzt.

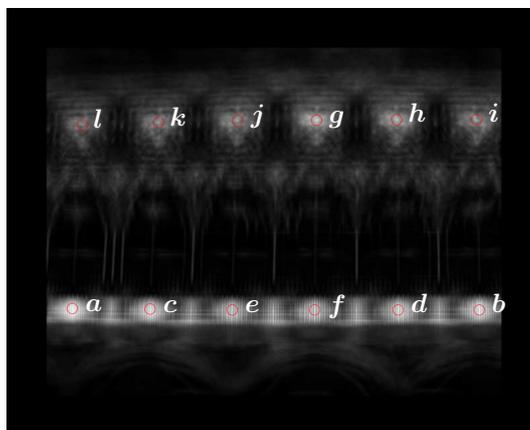
Sinusschwingungen. Für den praktischen Gebrauch sind solche Fälle aber eher zu vernachlässigen, da sie ohnehin nichts beinhalten, was sich als Objekt erfassen ließe.

## 4.4 Lokalisierung ornamentaler Strukturen

Eines der häufigsten Ornamente bildet der Vierpass. Abbildung 4.4 zeigt einen Ausschnitt des Dogenpalastes in Venedig, dessen Außenfassade viele solcher Vierpässe umfasst. Die ermittelte Symmetriekarte zeigt, dass ColorSym durchaus in der Lage ist, die Zentren dieser Vierpässe zu finden. Sechs der zwölf größten lokalen Maxima der Karte verweisen auf die Vierpässe. Neben den Vierpässen umfasst Abbildung 4.4 aber noch eine längere Balustrade, deren zahlreiche Balusters einen Bildbereich mit einem hohen Symmetriegehalt erzeugen. Dies führt dazu, dass die sechs Säulen im Bild im Bereich dieser Balustrade



(a) Originalbild



(b) Symmetriekarte

Abbildung 4.4: Dogenpalast. Die Vierpassformen haben eine Größe von etwa  $80 \times 75$  Pixeln, die vollständige Rosette besitzt hingegen eine Größe von etwa  $120 \times 110$  Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 60$  und `pwfType = 2` gesetzt.

ebenfalls eine hohe Symmetrie besitzen. Ferner entsprechen die sechs größten lokalen Maxima der Symmetriekarte Bildpunkten, die auf diesen Säulen liegen. Die Suche nach den sechs größten lokalen Maxima ergäbe damit sechs Positionen an der Balustrade und keinesfalls die gewünschten Symmetriezentren der sechs Vierpässe. Dies muss bei der Bestimmung der Interessensregionen berücksichtigt werden.

Ähnlich verhält es sich mit der Fassade der Ca' d'Oro, die in Abbildung 4.5 zu sehen ist. Neben den zu suchenden Vierpässen finden sich noch weitere architektonische Verzie-

#	$(x, y)$	Wert	#	$(x, y)$	Wert
<b>a</b>	(99, 454)	3433.790	<b>g</b>	(464, 168)	2823.180
<b>b</b>	(707, 454)	3353.860	<b>h</b>	(583, 168)	2638.830
<b>c</b>	(215, 455)	3312.300	<b>i</b>	(701, 168)	2295.980
<b>d</b>	(586, 455)	3183.180	<b>j</b>	(345, 169)	2228.800
<b>e</b>	(337, 455)	3105.060	<b>k</b>	(226, 172)	2069.160
<b>f</b>	(461, 456)	3048.820	<b>l</b>	(112, 175)	1821.510

Tabelle 4.1: Lokale Maxima zur Symmetriekarte von Abbildung 4.4.

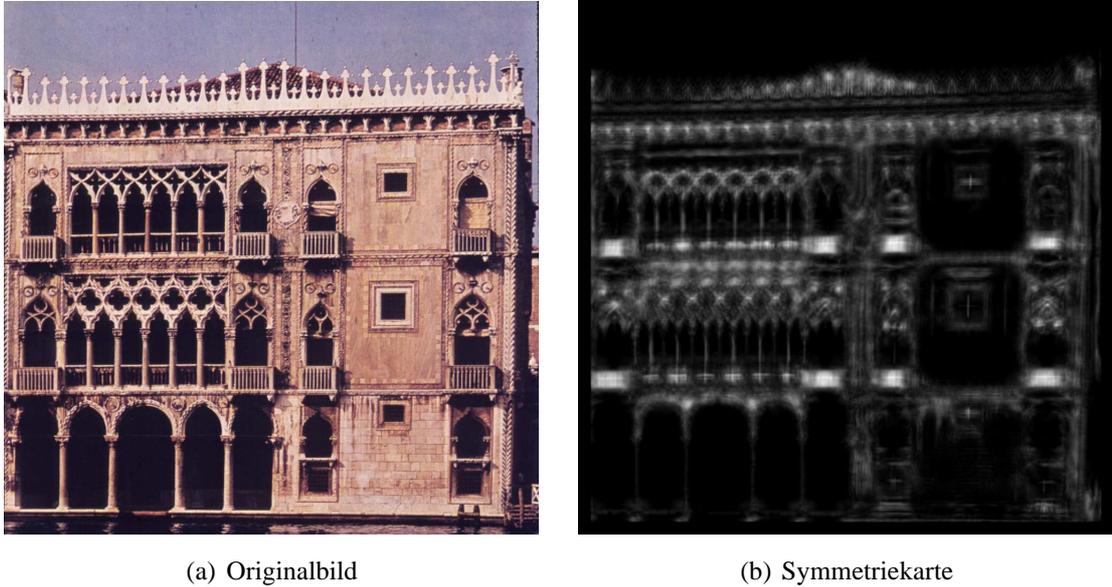


Abbildung 4.5: Ca' d'Oro. Die Vierpassformen haben einen Durchmesser von etwa 48 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 24$  und `pwfType = 2` gesetzt.

rungen, die ebenfalls einen hohen Symmetriegehalt aufweisen. Besonders auffallend sind hierbei wieder die Balustraden, die einige der lokalen Maxima der Symmetriekarte und damit die Interessensregionen bestimmen. Insgesamt wurden für die Symmetriekarte aus Abbildung 4.5 257 lokale Maxima berechnet. Bei einer Bildgröße von  $1000 \times 995$  Pixeln und einem Maximalradius  $d_{\max}$  von 24 liegt damit etwa die Hälfte aller Bildpunkte in einer Interessensregion.

Die Berechnung einer Symmetriekarte wird immer dann erschwert, wenn sich neben den zu suchenden Strukturen noch weitere symmetrische Bereiche im Bild finden. `ColorSym` betrachtet jeden Punkt als Symmetriezentrum und ermittelt in Abhängigkeit davon seinen tatsächlichen Symmetriegehalt. Weitere Strukturmerkmale werden nicht untersucht. Wie in Abschnitt 4.3 beschrieben, werden die Symmetriezentren von runden Objekten durch die Phasengewichtungsfunktion dabei stärker hervorgehoben. Andere Bildbereiche mit vielen Kanten innerhalb des zur Berechnung verwendeten Bildausschnitts liefern allerdings mehr Gradientenpaare, die in die Berechnung mit einfließen (vgl. hierzu Abschnitt 3.2.4). Eine große Zahl solcher Paare kann dabei trotz einer geringeren Gewich-

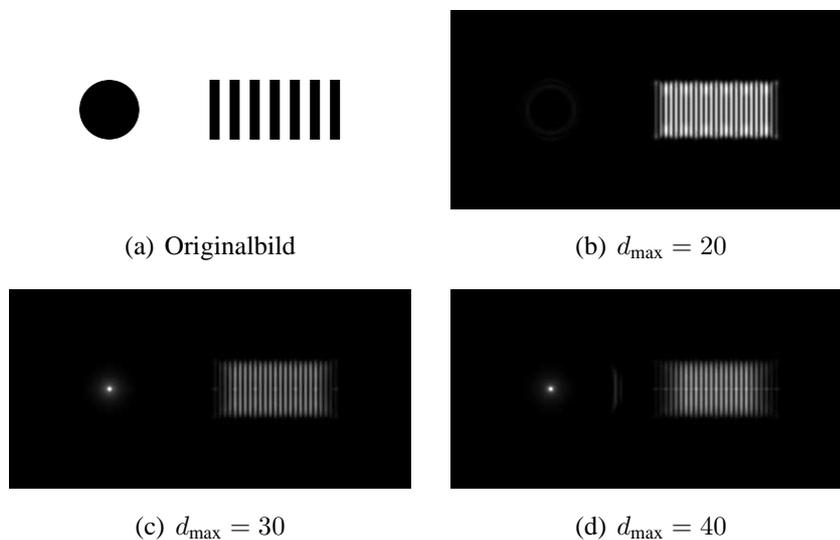
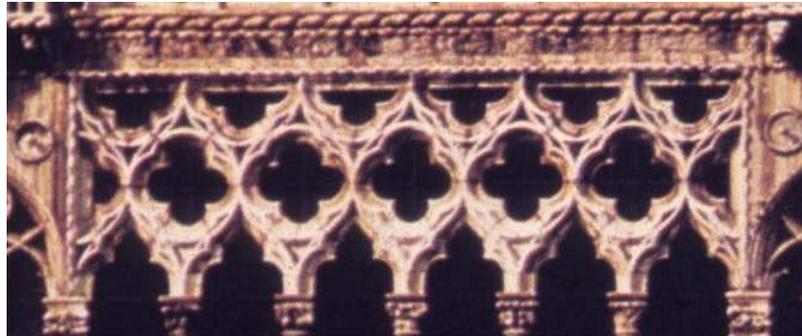


Abbildung 4.6: Kreis mit Balken. Der Kreis besitzt einen Radius von 30 Pixeln, die Balken besitzen jeweils eine Größe von  $10 \times 60$  Pixeln. Je nach Wahl von  $d_{\max}$  überragt der berechnete Symmetriegehalt der Balken den des Kreises. Zur Berechnung der Symmetriekarte wurde `pwfType = 2` gesetzt.

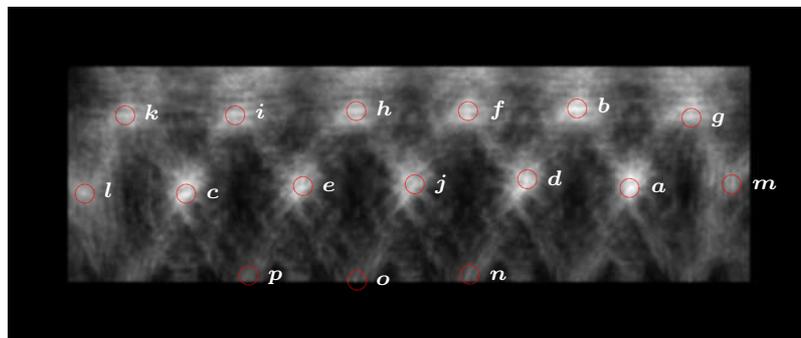
tion durch die Phasengewichtungsfunktion einen hohen Symmetriegehalt erreichen. Eine Veranschaulichung hierzu zeigt eine künstliche Szene in [Abbildung 4.6](#).

Um Beeinträchtigungen der Symmetriekarte durch ohnehin uninteressante Bildregionen zu unterbinden, lassen sich für die Berechnungen vorab mögliche Einschränkungen vornehmen. Diese Einschränkungen können etwa in Form von Bildausschnitten umgesetzt werden. [Abbildung 4.7](#) entspricht einen Ausschnitt aus [Abbildung 4.5](#). Es zeigt fünf vollständige Vierpässe sowie acht Vierpasshälften und sechs Nonnenköpfe. Bei der resultierenden Symmetriekarte wurden insgesamt 16 lokale Maxima ermittelt, die die entsprechenden Interessensregionen definieren. 13 dieser Regionen verweisen auf vollständige oder fragmentarische Vierpässe. Die übrigen drei Interessensregionen markieren drei der Nonnenköpfe.

Einen anderen Ausschnitt aus [Abbildung 4.5](#) zeigt [Abbildung 4.7](#). Zu sehen ist eine ähnliche Struktur, wiederum bestehend aus fünf vollständigen Vierpassen, acht Vierpasshälften und sechs Nonnenköpfen. Die Symmetriekarte weist ebenfalls 16 lokale Maxima auf. Fünf



(a) Originalbild



(b) Symmetriekarte

#	$(x, y)$	Wert	#	$(x, y)$	Wert
<b>a</b>	(207, 79)	1340.380	<b>i</b>	(254, 103)	1196.910
<b>b</b>	(296, 78)	1264.210	<b>j</b>	(474, 74)	1172.850
<b>c</b>	(343, 102)	1245.260	<b>k</b>	(385, 152)	1151.420
<b>d</b>	(297, 153)	1239.980	<b>l</b>	(420, 112)	1140.970
<b>e</b>	(564, 72)	1231.170	<b>m</b>	(471, 153)	1037.430
<b>f</b>	(88, 91)	1230.020	<b>n</b>	(116, 160)	961.677
<b>g</b>	(158, 119)	1216.420	<b>o</b>	(557, 155)	864.718
<b>h</b>	(520, 98)	1211.010	<b>p</b>	(56, 182)	692.558

Abbildung 4.7: Ca' d'Oro, Ausschnitt 1. Der Durchmesser der Vierpässe beträgt etwa 92 Pixel. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 50$  gesetzt.

davon verweisen auf die Zentren der Vierpässe, vier davon entsprechen den Zentren von Vierpasshälften. Von den Nonnenköpfen werden fünf gefunden. Zwei der lokalen Maxima markieren keine relevanten Bildbereiche. Obgleich die Vierpässe in Abbildung 4.7 deutlich von einer idealen Kreisstruktur abweichen, können ihre Symmetriezentren dennoch gefunden werden.

Die Abbildungen 4.7 und 4.8 zeigen, dass durch eine vorab vorgenommene Beschränkung des Eingabebildes brauchbarere Symmetriekarten errechnet werden können. Die Einschränkung sollte dabei möglichst viele irrelevante Bildbereiche herausfiltern. Für die hier aufgeführten Beispiele wurden die Begrenzungen noch manuell vorgenommen. Dieser Prozess sollte daher noch automatisiert werden.

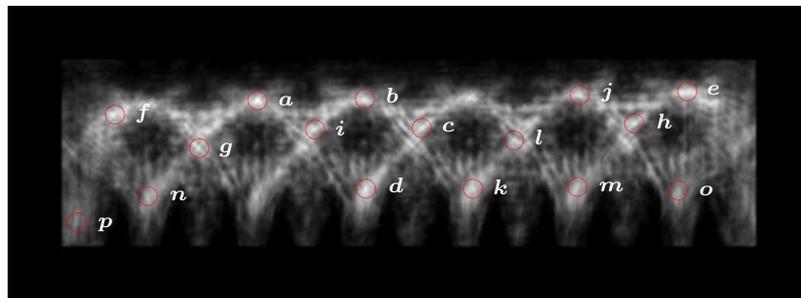
## 4.5 Dreipässe und Wirbelräder mit drei Speichen

Wirbelräder und Vielpässe besitzen eine zentralsymmetrische Konstruktion [Den05], d. h. sie besitzen genau ein Symmetriezentrum. Beide ornamentale Strukturen lassen sich einteilen in  $n$  Elemente. Bei Wirbelrädern bestimmt  $n$  die Zahl der Speichen, bei Vielpässen die Zahl der Blätter. Für ColorSym ist zu unterscheiden, ob  $n$  gerade oder ungerade ist. Im Folgenden werden vereinfacht nur solche Ornamente betrachtet, die aus einer einzigen Konturlinie bestehen. Zwei solcher Beispiele für Vielpässe finden sich in Abbildung 4.9. Eine Verallgemeinerung zu komplexeren Strukturen ist hierbei leicht möglich.

Als Berechnungsgrundlage für die Symmetriekarte dient gemäß Abschnitt 3.2.10 eine Gradientenkarte des Eingabebildes. Jeder Punkt  $\mathbf{p}$  dieses Bildes wird als potentiellles Symmetriezentrum angenommen. Durch die Indexmenge  $\Gamma(\mathbf{p})$  werden dann für  $\mathbf{p}$  Paare von Nachbarpunkten  $\mathbf{p}_l$  und  $\mathbf{p}_r$  definiert, die denselben Abstand zu  $\mathbf{p}$  haben und mit diesem auf einer Geraden liegen (vgl. Abbildung 4.9). Für jedes solche Paar werden wiederum mehrere Gradientenpaare  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  entsprechend der Indexmenge  $\Lambda(\mathbf{p}, l, r)$  ermittelt. Die Gradientenpaare dienen dann zur eigentlichen Berechnung und liefern jeweils einen eigenen Symmetriewert. Dabei ist zu beachten, dass jedes Paar  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  separat ausgewertet wird.



(a) Originalbild



(b) Symmetriekarte

#	$(x, y)$	Wert	#	$(x, y)$	Wert
<b>a</b>	(207, 79)	1340.380	<b>i</b>	(254, 103)	1196.910
<b>b</b>	(296, 78)	1264.210	<b>j</b>	(474, 74)	1172.850
<b>c</b>	(343, 102)	1245.260	<b>k</b>	(385, 152)	1151.420
<b>d</b>	(297, 153)	1239.980	<b>l</b>	(420, 112)	1140.970
<b>e</b>	(564, 72)	1231.170	<b>m</b>	(471, 153)	1037.430
<b>f</b>	(88, 91)	1230.020	<b>n</b>	(116, 160)	961.677
<b>g</b>	(158, 119)	1216.420	<b>o</b>	(557, 155)	864.718
<b>h</b>	(520, 98)	1211.010	<b>p</b>	(56, 182)	692.558

Abbildung 4.8: Ca' d'Oro, Ausschnitt 2. Der Durchmesser der Vierpässe beträgt etwa 88 Pixel. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 45$  und  $\text{pwfType} = 2$  gesetzt.

Sofern sich die Gradientenpaare  $(\mathbf{g}_{i,l}, \mathbf{g}_{r,j})$  tatsächlich zur Berechnung der real vorhandenen Symmetrie verwenden lassen, erscheint diese Herangehensweise durchaus sinnvoll. Werden für eine symmetrische Struktur aber zu wenige solcher Gradientenpaare definiert, so lässt sich die Symmetrie dieser Struktur nur bedingt ermitteln. Dies ist etwa dann der Fall, wenn  $n$ , die Zahl der Elemente des betrachteten Ornaments, ungerade ist. Wie in Abbildung 4.9 für einen Dreipass exemplarisch skizziert, kann die eingeschränkte Betrachtung von Paaren  $(\mathbf{p}_l, \mathbf{p}_r)$  die tatsächliche Symmetrie des Objektes nicht adäquat erfassen. Die Symmetrie eines Dreipasses lässt sich nicht durch solche Paare beschreiben, sofern diese alle separat und nicht gemeinsam betrachtet werden. Für Wirbelräder mit einer ungeraden Anzahl von Speichen gilt ähnliches. Die Symmetriekarten, die ColorSym für diese ornamentalen Strukturen berechnet, können daher ggf. unbrauchbar sein. Die Ursache hierfür ist die Nachbarschaftsindexmenge  $\Gamma(\mathbf{p})$ , die unabhängig von der realen Struktur eines Bildobjekts jeweils nur zwei Punkte mit demselben Abstand zum Zentrum betrachtet und dabei fordert, dass alle drei Punkte auf einer gemeinsamen Geraden liegen. Solche Voraussetzungen sind aber nicht bei allen ornamentalen Strukturen gegeben.

Abbildung 4.10 zeigt ein Bild von drei Dreipässen und dessen berechnete Symmetriekarten. Für jeden dieser Dreipässe lässt sich ein Kreis so konstruieren, dass er den Dreipass vollständig umschließt. Der Mittelpunkt eines jeden Kreises entspricht dabei dem realen Symmetriezentrum der Dreipässe. Der Parameter  $d_{\max}$  wurde angepasst an den Radius der kleinsten so konstruierbaren Kreise. Zur Berechnung des Symmetriegehalts eines jeden Kreiscentrums wird für diese jeweils eine solche Nachbarschaft konstruiert, die alle Punkte des Dreipasses (und ggf. noch weitere) umfasst. D. h. zur Berechnung werden alle Punkte des Dreipasses berücksichtigt. Durch die paarweise Betrachtung der Nachbarschaftspunkte können allerdings nicht die realen Symmetriezentren der Dreipässe gefunden werden. Stattdessen liefern die Symmetriekarten aus Abbildung 4.10 zu jedem Dreipass die Zentren von drei Zweipässen, aus denen sich der Dreipass zusammensetzt. Für die radiale und die isotrope Symmetrie sind die Resultate ähnlich, da bei beiden dieselbe Indexmenge  $\Gamma(\mathbf{p})$  verwendet wird.

Für Wirbelräder mit einer ungeraden Zahl von Speichen ergeben sich ähnliche Schwächen. Abbildung 4.11 zeigt drei Wirbelräder mit je drei Speichen. Die berechnete Symmetriekarte ist kaum brauchbar, da sie an den realen Symmetriezentren der Wirbelräder einen

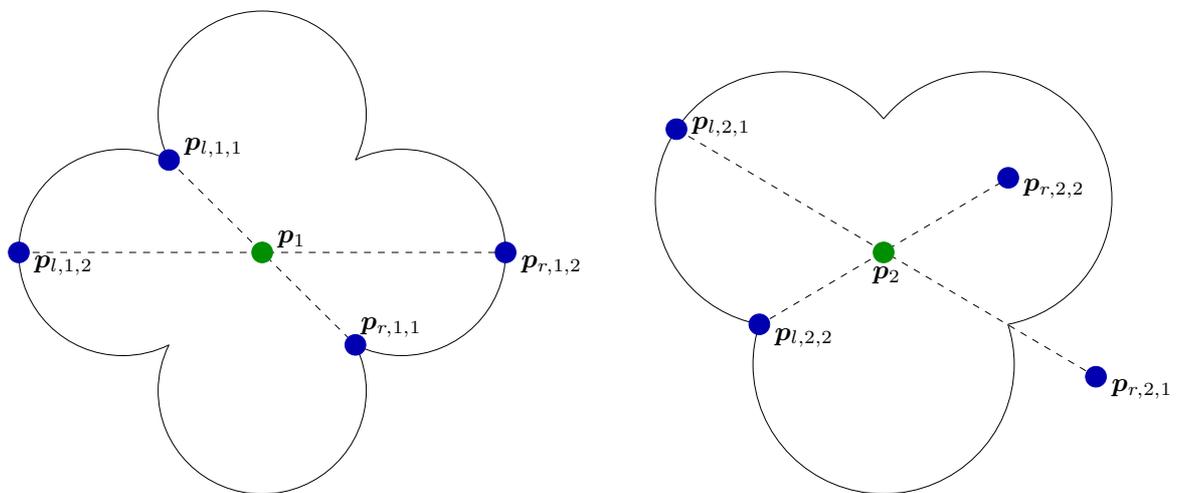


Abbildung 4.9: ColorSym betrachtet für jede Position  $p$  mehrere Paare von Punkten  $(p_l, p_r)$ .  $p_l$  und  $p_r$  haben denselben Abstand zu  $p$ . Die Symmetrie von  $p$  wird auf Basis dieser Punkte berechnet. Lässt sich aus diesen aber nicht die tatsächliche Symmetrie der Struktur ableiten, können die Ergebnisse von ColorSym ggf. unbrauchbar sein. Dies gilt etwa für Vielpässe mit einer ungeraden Zahl von Spiegelachsen. Strukturen mit einer geraden Anzahl von Elementen davon können hingegen erfasst werden.

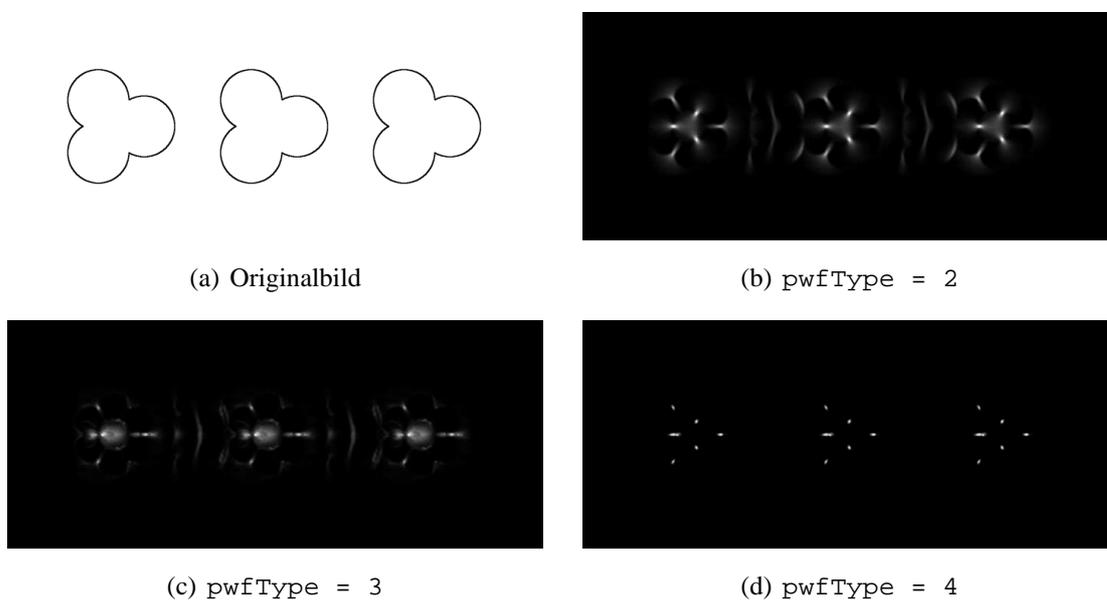


Abbildung 4.10: Dreipässe. Jedes Blatt eines Dreipass hat einen Radius von etwa 40 Pixeln. Der kleinste Kreis, der einen Dreipass vollständig umschließt, hat einen Radius von etwa 80 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 85$  gesetzt.



(a) Originalbild



(b) Symmetriekarte

Abbildung 4.11: Wirbelräder (1). Die Wirbelräder besitzen einen Radius von etwa 80 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 85$  und `pwfType = 2` gesetzt.

deutlich zu kleinen Symmetriehalt aufweist. Die lokalen Maxima der Symmetriekarte, die die Interessensregionen definieren, finden sich innerhalb der Speichen.

Bei den Wirbelrädern aus Abbildung 4.11 handelt es sich jedoch um künstliche Konstrukte, die in dieser oder einer ähnlichen Form nur selten in der Ornamentik verwendet werden. Wirbelräder werden im Überwiegenden in kreisartige Strukturen eingebettet, wie dies etwa in Abbildung 4.12 zu sehen ist. Für solche Ornamente ergeben sich zum Teil gravierende Unterschiede in der berechneten Symmetriekarte. Wie in Abschnitt 4.3 gezeigt wurde, besitzt bereits ein einzelner Kreis genügend Symmetrie, um für sein Zentrum einen hohen Wert in der berechneten Symmetriekarte zu liefern. Entsprechend ist es nur von geringer Bedeutung, ob eine ornamentale Struktur wie aus Abbildung 4.12 neben dem kreisförmigen Umriss noch weitere symmetrische Eigenschaften besitzt – selbst wenn diese nicht durch ColorSym erfasst werden können. Dies kann sich einerseits als Vorteil erweisen, wenn dadurch die Lokalisierung von Ornamenten verbessert wird. Befinden sich in dem zu betrachtenden Bild allerdings zusätzliche kreisartige Objekte (wie etwa Verkehrszeichen), so könnte dies wiederum die Berechnung der Symmetriekarte negativ beeinflussen.

## 4.6 Flächensymmetrien und symmetrische Flächen

Flächensymmetrien bilden aufgrund ihrer weit gefassten Definition eine große Symmetriegruppe und können sich in ihrer Konstruktion zum Teil deutlich voneinander unterscheiden. Allen gemeinsam ist aber, dass sie durch die Wiederholung eines Musters eine

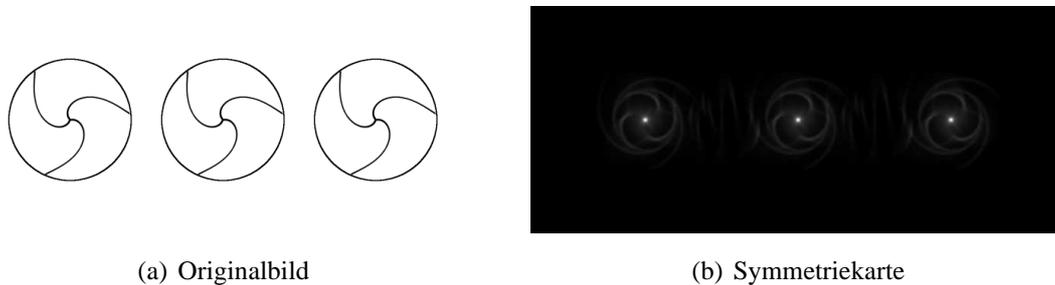


Abbildung 4.12: Wirbelräder (2). Die Wirbelräder besitzen einen Radius von etwa 80 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 85$  und `pwfType = 2` gesetzt.

starke symmetrische Grundstruktur besitzen. Dieses Grundmuster selbst kann dabei wieder ein symmetrisches Objekt mit einem eigenen Symmetriezentrum bilden. Die gesamte Flächensymmetrie umfasst dann zahlreiche solcher Symmetriezentren. [Abbildung 4.13](#) zeigt eine Flächensymmetrie mit einer einfachen Struktur, deren Grundmuster ein Vierpass bildet. In der zugehörigen Symmetriekarte entsprechen die Symmetriezentren der Vierpässe den lokalen Maxima. Bei einfachen Mustern ist `ColorSym` also in der Lage, ein brauchbares Ergebnis zu liefern. Für komplexere Strukturen können sich hingegen einige Schwächen bei den Symmetriekarten ergeben.

Jedes ermittelte Symmetriezentrum definiert nach [Abschnitt 3.2.10](#) eine Interessensregion. Das Ziel einer Interessensregion ist es, einen markanten Bildbereich zu definieren, der dann für weitere Verarbeitungsprozesse verwendet wird. Interessensregionen sind also nur dann sinnvoll, wenn sie den zu verarbeitenden Bildinhalt einschränken. Werden für ein Bild zu viele Interessensregionen definiert, besitzen sie nur noch einen geringen Informationsgehalt. Für komplexere Flächensymmetrien, die sehr viele Symmetriezentren und damit auch viele potentielle Interessensregionen besitzen, kann die berechnete Symmetriekarte daher ggf. unbrauchbar sein. Ein Beispiel hierfür zeigt [Abbildung 4.14](#). Durch die Überschneidung des Grundmusters dieser Flächensymmetrie werden mehr Symmetriezentren erzeugt als für die weiteren Verarbeitungsprozesse benötigt werden.

[Abbildung 4.15](#) zeigt eine weitere Flächensymmetrie, die eine komplexe Struktur besitzt. Diese setzt sich zusammen aus zwei Grundmustern, die jeweils einem Vierpass entsprechen. Beide dieser Grundmuster weisen eine deutlich unterscheidbare Größe auf. Zur Be-

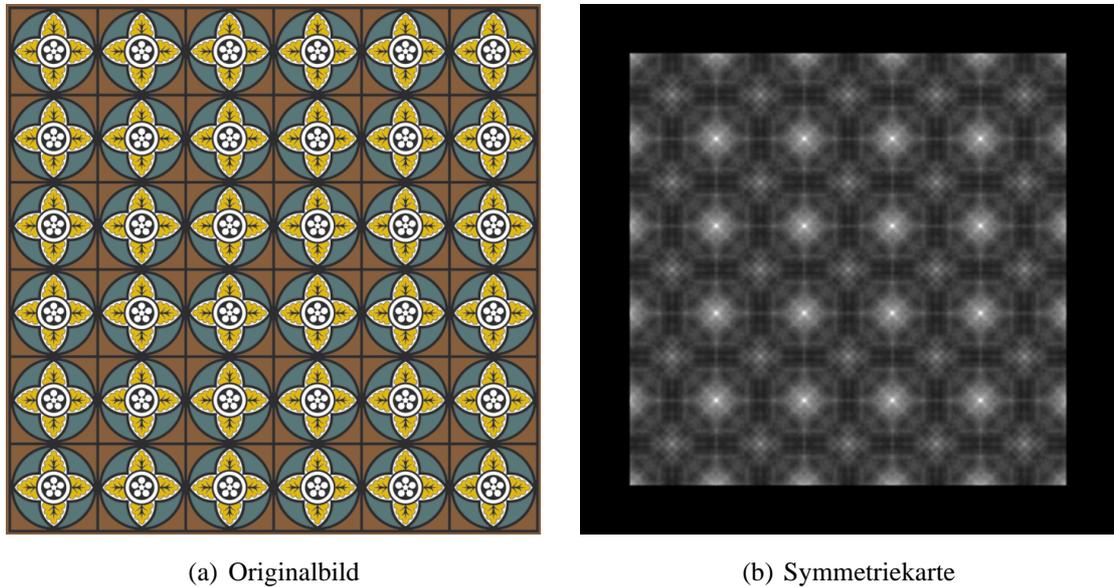


Abbildung 4.13: Flächensymmetrie mit Vierpässen (1). Jede quadratische Zelle hat eine Größe von etwa  $117 \times 117$  Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 65$  und  $\text{pwfType} = 2$  gesetzt.

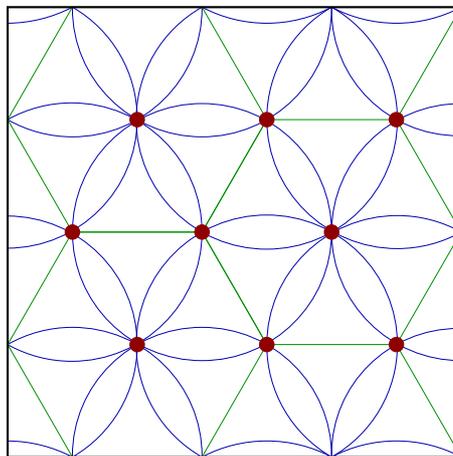


Abbildung 4.14: Je mehr Symmetriezentren eine Struktur aufweist, desto mehr Symmetriezentren können potentiell von ColorSym gefunden werden. In der obigen Abbildung sind alle möglichen Symmetriezentren rot markiert. Durch die strukturelle Überlagerung des Grundmusters liegen mehr Symmetriezentren vor als es symmetrische Bereiche gibt.

rechnung der Symmetriekarte wurde nun der Parameter  $d_{\max}$  so gewählt, dass die größere Vierpassestruktur gefunden werden sollte. Wie in der resultierenden Symmetriekarte zu sehen, befinden sich die lokalen Maxima jedoch ausschließlich an den Bildpunkten, die den Symmetriezentren der kleineren Grundmuster entsprechen. Die große Struktur besteht zwar aus drei ineinandergeschachtelten Vierpässen und verstärkt damit gemäß dem menschlichen Symmetrieverständnis den Symmetriegehalt ihres Zentrums. Wie in Abschnitt 4.3 beschrieben, bevorzugt ColorSym allerdings kreisartige Muster gegenüber anderen Strukturen. Abbildung 4.16 zeigt, dass die weißen Blätter der großen Vierpässe ein solches kreisförmiges Muster erzeugen, deren Zentrum genau dem Mittelpunkt einer kleinen Vierpassestruktur entspricht. Für die großen Vierpässe selbst bilden die Blätter hingegen keine Kreisstruktur. Wie der Symmetriekarte zu entnehmen ist, ist die Symmetrie der Kreismuster so groß, dass sie die markantesten Symmetriezentren der Karte definieren. Wann immer sich also in einem Bild viele kreisförmige Strukturen finden, deren Mittelpunkte nicht identisch zu den Zentren der zu suchenden Ornamente sind, so ist damit zu rechnen, dass die resultierenden Symmetriekarten einige Schwächen aufweisen können.

Viele der zu untersuchenden Ornamente im Projekt Bildanalyse zur Ornamentklassifikation finden sich an Mauerwerken oder vergleichbaren Strukturen. Die Grundsteine eines Mauerwerks weisen oft eine eigene Symmetrie auf und machen damit aus der gesamten Mauer eine Flächensymmetrie. Obgleich solche Symmetrien für die Lokalisierung der eigentlich gesuchten Ornamente wenig relevant sind, können sie diese Suche dennoch beeinflussen. Abbildung 4.17 zeigt einen Ausschnitt der Stadtmauer von Worms<sup>3</sup>, die ein Beispiel für eine solche Flächensymmetrie bildet. ColorSym erkennt eine horizontal strukturierte Symmetrie, die sich durch die verschiedenen Steinreihen ergibt. Werden nun dem Originalbild sechs künstliche Kreise hinzugefügt, so werden deren Zentren ebenfalls von ColorSym gefunden. Die Zentren entsprechen dabei lokalen Maxima in der Symmetriekarte. Wie in Tabelle 4.2 aufgeführt, entsprechen nur vier der sechs größten lokalen Maxima den gewünschten Kreiszentren. Die übrigen Zentren tauchen erst an den Positionen sieben und acht auf. Um alle Mittelpunkte der Kreise erfolgreich finden zu können, muss die Suche nach den lokalen Maxima modifiziert oder bereits bei der Berechnung der Symmetriekarte entsprechende Einschränkungen vorgenommen werden (vgl. Abschnitt 4.4).

---

<sup>3</sup>Quelle: [http://en.wikipedia.org/wiki/Stone\\_wall](http://en.wikipedia.org/wiki/Stone_wall), Stand vom 12.09.2007.

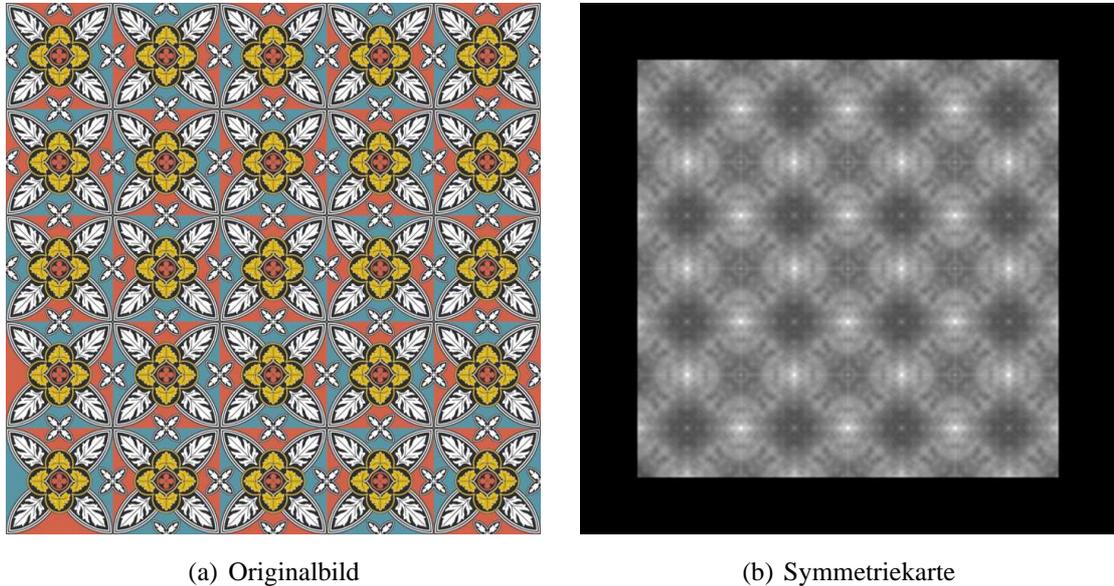


Abbildung 4.15: Flächensymmetrie mit Vierpässen (2). Jede quadratische Zelle hat eine Größe von etwa  $140 \times 140$  Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 75$  und  $\text{pwfType} = 2$  gesetzt.



Abbildung 4.16: Das Grundmuster der Flächensymmetrie aus Abbildung 4.15 weist kreisförmige Muster auf, deren Mittelpunkte den Symmetriezentren der großen Vierpässe widersprechen. Dies resultiert in einer unbrauchbaren Symmetriekarte.

#	$(x, y)$	Wert
<b>a</b>	(625, 351)	929.939
<b>b</b>	(227, 178)	890.985
<b>c</b>	(439, 76)	889.872
<b>d</b>	(204, 415)	858.808
<b>e</b>	(339, 286)	760.487
<b>f</b>	(675, 287)	739.085
<b>g</b>	(430, 420)	730.092
<b>h</b>	(500, 181)	716.470
<b>i</b>	(680, 179)	710.366

Tabelle 4.2: Lokale Maxima zur Symmetriekarte von Abbildung 4.17(d).

Hierbei gilt es wieder, irrelevante Bildbereiche als Interessensregionen möglichst auszuschließen.

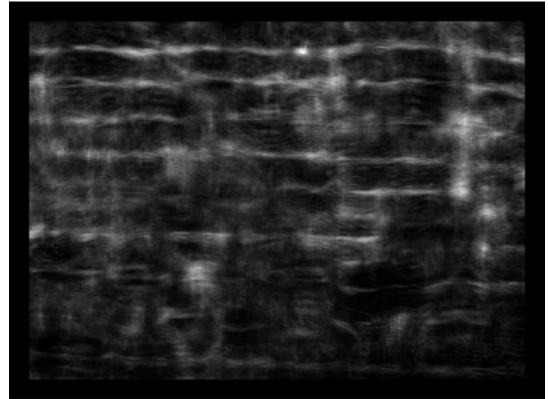
## 4.7 Bedeutung des Parameters $d_{\text{max}}$

Wie in Abschnitt 3.2.10 beschrieben, ist der Maximalradius  $d_{\text{max}}$  der wichtigste Parameter für die Berechnung einer Symmetriekarte.  $d_{\text{max}}$  definiert eine kreisförmige Nachbarschaft um einen Bildpunkt  $p$ . Alle Punkte innerhalb dieser Nachbarschaft werden zur Berechnung des Symmetriehalts von  $p$  verwendet.  $d_{\text{max}}$  muss stets an die Größe der zu suchenden Symmetrie angepasst werden. Besonders wichtig ist die richtige Wahl von  $d_{\text{max}}$  bei der Suche nach Bildobjekten, die selbst wiederum aus symmetrischen Strukturen bestehen. Viele Ornamente sind genau so konstruiert und benötigen daher einen entsprechend exakten Wert von  $d_{\text{max}}$ . Ein Beispiel hierfür bilden etwa Vielpässe, wie sie u. a. in Abbildung 4.9 skizziert werden. Die Blätter der dort abgebildeten Vielpässe besitzen eine kreisförmige Struktur und sind damit zentralsymmetrisch – wie auch der gesamte Vielpass selbst.

Abbildung 4.18 zeigt einen Fensterbogen mit zwei Dreipässen und einem Vierpass. Alle Vielpässe besitzen kreisrunde und damit zentralsymmetrische Blätter von unterschiedli-



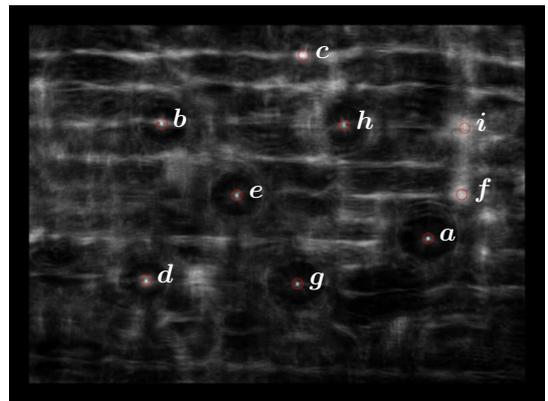
(a) Originalbild



(b) Symmetriekarte zu 4.17(a)



(c) Originalbild mit Kreisen



(d) Symmetriekarte zu 4.17(c)

Abbildung 4.17: Stadtmauer von Worms. Die Höhe der Steinreihen schwankt zwischen 50 und 60 Pixeln. Je zwei der hinzugefügten Kreise besitzen einen Radius von 45, von 50 und von 55 Pixeln. Zur Berechnung der Symmetriekarte wurde  $d_{\max} = 30$  und  $\text{pwfType} = 2$  gesetzt.

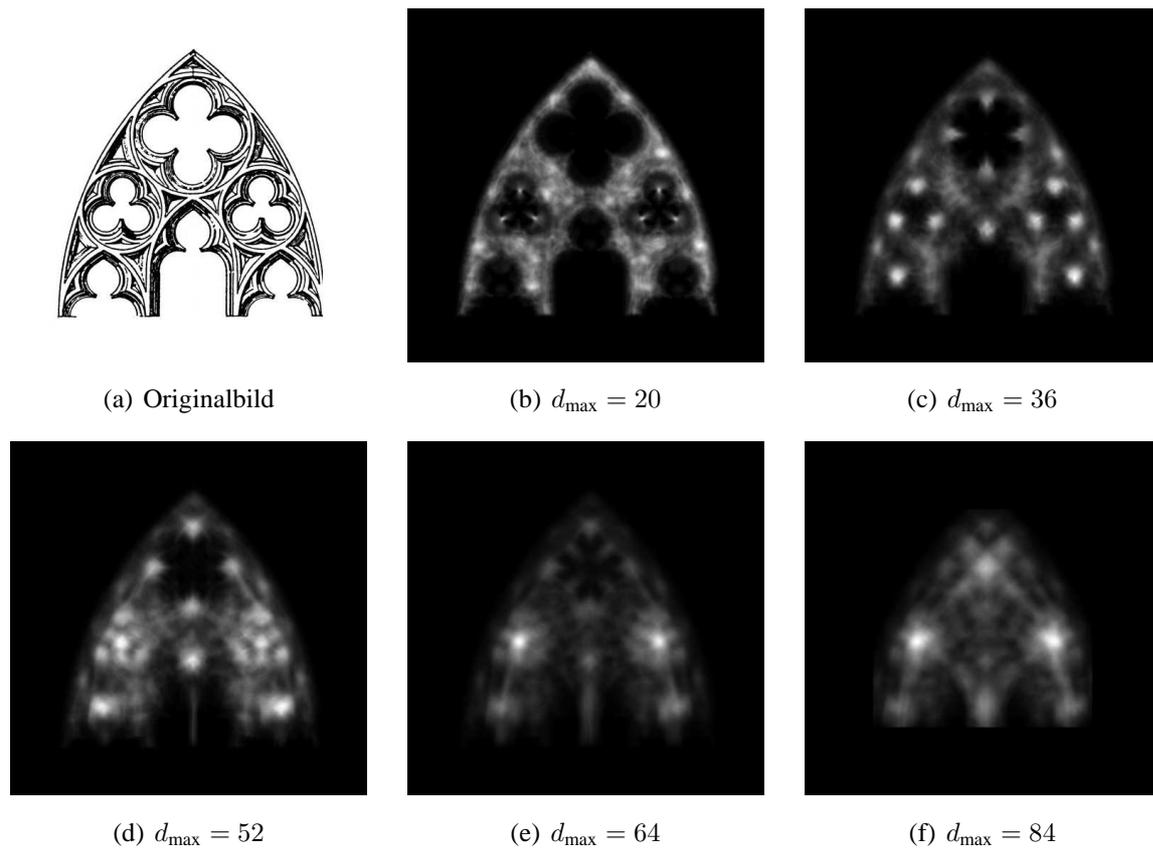


Abbildung 4.18: Dreipässe und Vierpass. Die Blätter der Dreipässe haben etwa einen Radius von 18 Pixeln. Der kleinste Kreis, der einen Dreipass vollständig umschließt, hat einen Radius von etwa 44 Pixeln. Für den Vierpass sind diese Radien jeweils etwa 27 und 70 Pixel groß. Zur Berechnung der Symmetriekarten wurde `pwfType = 2` gesetzt.

cher Größe.  $d_{\max}$  bestimmt nun, welche symmetrischen Bereiche des Bildes im Konkreten gefunden werden sollen. Wird  $d_{\max}$  mit einem kleinen Wert belegt, können nur kleine Bereiche wie etwa die Blätter der Vielpässe gefunden werden. Durch eine sukzessive Vergrößerung von  $d_{\max}$  wird es dann möglich, auch größere symmetrische Bildregionen zu finden. Je größer ein symmetrischer Bildbereich ist, desto mehr potentielle Bildpunkte gibt es, die auf diese Symmetrie verweisen. Die Blätter eines Vielpasses sind kleiner als der komplette Vielpass selbst und besitzen damit auch deutlich weniger solcher Bildpunkte. D. h. die effektiv zur Symmetrieberechnung beitragende Bildregion ist bei den Blättern deutlich kleiner als bei dem kompletten Vielpass. Wird  $d_{\max}$  nun so gewählt, dass die Symmetrie des vollständigen Vielpasses berechnet werden kann, wird zugleich auch die Symmetrie jedes Blattes lückenlos erfasst. Die unterschiedliche Zahl der Bildpunkte, die hierbei effektiv zur Berechnung der Symmetrie beitragen, führt aber dazu, dass der Symmetriegehalt eines jeden Blattes deutlich kleiner ist als der des Vielpasses. In Abbildung 4.18(c) stechen die Zentren der Blätter von den Dreipässen noch deutlich hervor. Durch eine Vergrößerung von  $d_{\max}$  in Abbildung 4.18(d) relativiert sich die Symmetrie dieser Zentren jedoch und die Mittelpunkte der kompletten Dreipässe bilden die neuen lokalen Maxima. Die Dreipässe können in diesem Beispiel gefunden werden, da sie in kreisförmige Strukturen eingebettet sind. Wie in Abschnitt 4.3 erläutert wurde, kann dies die Berechnung der Symmetriekarte beeinflussen.

Generell gilt, dass  $d_{\max}$  stets an die Größe der zu suchenden Strukturen angepasst werden muss. Je größer  $d_{\max}$  im Vergleich zu seinem idealen Wert ist, desto mehr irrelevante Bildregionen können in die Berechnung der Symmetrie mit einfließen. Dies gilt besonders bei der Betrachtung von Vielpässen, die selbst wiederum aus symmetrischen Teilbereichen bestehen. Ferner kann ein zu großer Wert von  $d_{\max}$  dazu führen, dass auch solche Bildbereiche als symmetrisch aufgefasst werden, deren Symmetrie völlig irrelevant ist. Hierzu zählt etwa Bildrauschen oder die Symmetrie eines kompletten Objektes, das mit Ornamenten geschmückt ist. Ein Beispiel dazu bildet etwa ein Achtpass, dessen Blätter aus Vierpässen bestehen. Um einen möglichst genauen Wert für  $d_{\max}$  zu erhalten, wurde für alle Testdurchläufe der vorliegenden Arbeit der Parameter stets manuell angepasst. Sofern mehrere Parameterwerte verwendet wurden, wurde stets die beste Symmetriekarte ausgewählt und abgebildet.

## 4.8 Vergleich zwischen radialer und isotroper Symmetrie

Ziel dieses Abschnitts ist es, einen Vergleich zwischen radialer und isotroper Symmetrie herzustellen. Für zwei Eingabebilder wurden dazu jeweils drei Symmetriekarten berechnet, die sich durch den Wert des Parameters `pwfType` voneinander unterscheiden (vgl. Abschnitt 3.3.2). Verwendet wurden hierbei die Werten 2, 3 und 4.

Abbildung 4.19 zeigt ein einfaches Schachbrettmuster. Jedes Feld besitzt eine eigene Symmetrie, das gesamte Schachbrettmuster bildet eine Flächensymmetrie.  $d_{\max}$  wurde so gewählt, dass `ColorSym` die Symmetriezentren eines jeden Feldes erkennen sollte. Dies wurde auch für alle drei berechneten Symmetriekarten erreicht, d. h. die lokalen Maxima der Karten entsprechen den Symmetriezentren der Schachbrettfelder. Die isotrope Symmetrie wurde mit `pwfType` = 2 berechnet. Neben den Zentren der Schachbrettfelder erfasst sie ferner die lineare Strukturierung des Schachbretts, die sich durch die einzelnen Zeilen und Spalten ergibt. Bei komplexeren Eingabebildern können sich ähnliche Artefakte ergeben, wodurch die berechnete Symmetriekarte ungenau werden kann. Die einzelnen Schachbrettfelder sind zentralsymmetrisch und besitzen damit die idealen Voraussetzungen für die Berechnung der radialen Symmetrie. Daher ist bei der radialen Symmetriekarte die lineare Struktur auch weniger stark ausgeprägt. Die Mittelpunkte der Schachbrettfelder können mit einer größeren Präzision bestimmt werden als bei der isotrophen Symmetrie. Berechnet wird die radiale Symmetrie mit `pwfType` = 3. Durch die Kombination der radialen und der isotrophen Symmetrie mit `pwfType` = 4 ergibt sich eine deutlich kontrastreiche Symmetriekarte. Zur Bestimmung der Interessensregionen werden aber stets die lokalen Maxima einer Symmetriekarte verwendet und nicht die gesamte Karte. Der Kontrast einer solchen Karte hat dementsprechend auch nur eine äußerst geringe Bedeutung und lässt sich lediglich zu Anschauungszwecken verwenden. Der Kontrast selbst entsteht durch das Setzen eines Schwellwertes, der der Hälfte des größten Wertes der Karte entspricht (vgl. Abschnitt 3.3.2).

Abbildung 4.20 zeigt einen Ausschnitt der Kathedrale St. Nazaire. Das Eingabebild zeigt eine Art Friesensymmetrie, bestehend aus sechs Vierpässen in quadratischen Rahmen. Zusätzlich findet sich im Bild eine Reihe von sieben Nonnenköpfen, deren Breite ähnlich zur Breite der Vierpässe ist. Obgleich sich die drei berechneten Symmetriekarten in ihrer

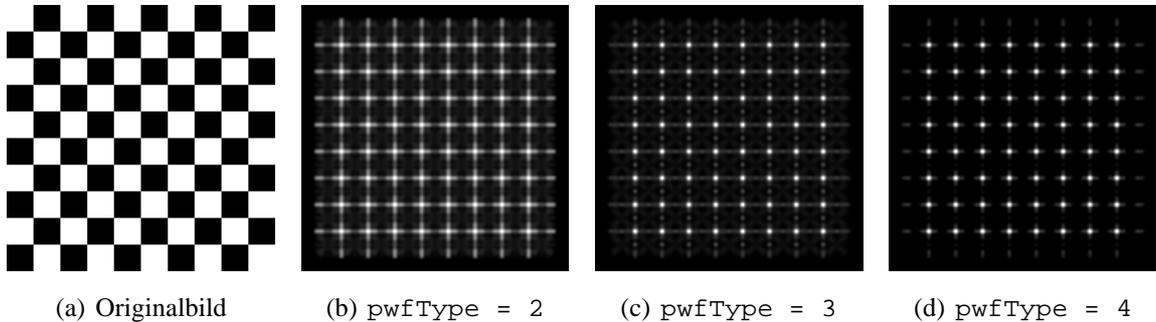


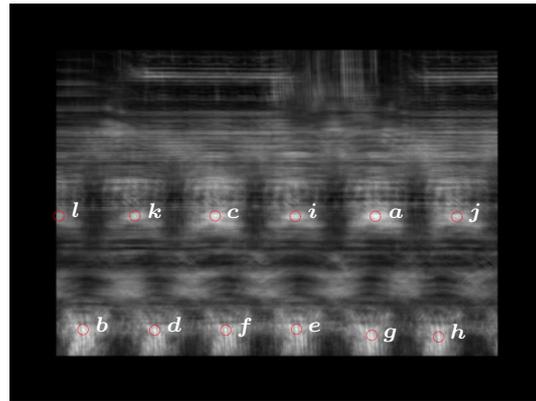
Abbildung 4.19: Schachbrettmuster. Ein quadratisches Feld hat eine Größe von  $20 \times 20$  Pixeln. Zur Berechnung der Symmetriekarten wurde  $d_{\max} = 10$  gesetzt.

Gesamtheit zum Teil deutlich voneinander unterscheiden, sind ihre lokalen Maxima sehr ähnlich. Da diese letztlich die Interessensregionen bestimmen, ist der praktische Unterschied zwischen den drei Symmetriekarten auch nur minimal. Allen drei Symmetriekarten ist gemein, dass sowohl die Zentren der Vierpässe als auch die Zentren der Nonnenköpfe gefunden werden. Unterschiede finden sich hierbei lediglich in der Reihenfolge der größten lokalen Maxima. Eine Gegenüberstellung der lokalen Maxima für die einzelnen Symmetriekarten zeigt Tabelle 4.3.

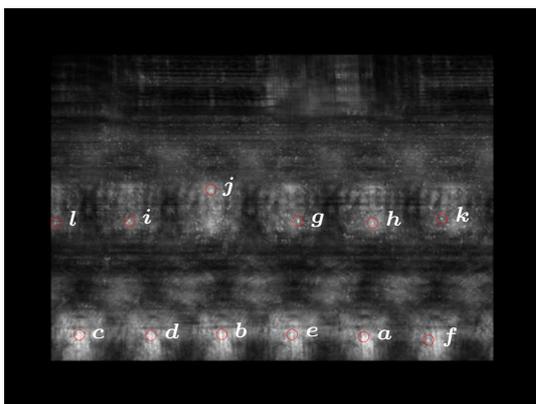
Beide Testbilder zeigen, dass sowohl die radiale als auch die isotrope Symmetriekarte brauchbare Ergebnisse liefern. Je nach Eingabebild können sich ihre lokalen Maxima dabei minimal voneinander unterscheiden. Bei Abbildung 4.19 liefert die radiale Symmetriekarte bessere Ergebnisse, da die lokalen Maxima eine deutliche Unterscheidbarkeit zu ihren Nachbarpunkten besitzen. Für das Eingabebild aus Abbildung 4.20 weisen die Interessensregionen der isotropen Symmetriekarte eine höhere Stabilität auf und haben damit auch eine bessere Qualität als die der radialen Symmetriekarte. Eine Kombination beider Karten mit  $\text{pwfType} = 4$  ist in beiden Fällen nur bedingt brauchbar und zeigt keine nennenswerten Vorteile.



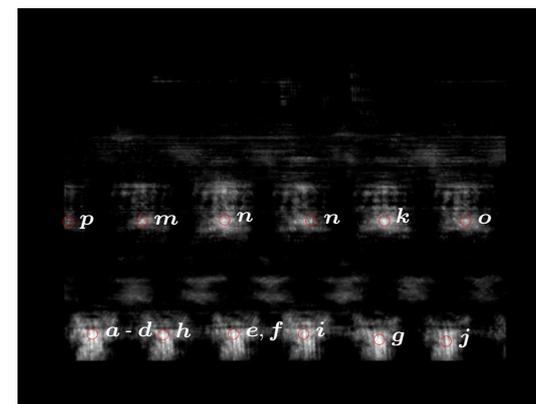
(a) Originalbild



(b) pwfType = 2



(c) pwfType = 3



(d) pwfType = 4

Abbildung 4.20: Cathedrale St. Nazaire. Die Vierpassformen bilden eine Friesensymmetrie. Jeder Vierpass hat eine Größe von etwa  $90 \times 85$  Pixeln, das vollständige quadratische Muster eines jeden Vierpass besitzt hingegen eine Größe von etwa  $125 \times 115$  Pixeln. Zur Berechnung der Symmetriekarten wurde  $d_{\max} = 70$  gesetzt.

	pwfType = 2		pwfType = 3		pwfType = 4	
#	$(x, y)$	Wert	$(x, y)$	Wert	$(x, y)$	Wert
<b>a</b>	(547, 319)	3019.44	(536, 495)	1416.10	(110, 490)	122.072
<b>b</b>	(110, 490)	2962.38	(323, 492)	1387.45	(110, 491)	122.072
<b>c</b>	(308, 318)	2875.48	(111, 490)	1381.88	(110, 492)	122.072
<b>d</b>	(217, 491)	2854.00	(218, 494)	1347.72	(110, 493)	122.072
<b>e</b>	(429, 489)	2816.70	(428, 490)	1288.02	(323, 491)	122.072
<b>f</b>	(323, 491)	2763.17	(632, 499)	1259.26	(323, 492)	122.072
<b>g</b>	(541, 501)	2695.68	(439, 320)	1183.26	(541, 500)	121.959
<b>h</b>	(642, 501)	2641.16	(549, 323)	1149.96	(217, 492)	121.868
<b>i</b>	(428, 319)	2554.96	(187, 319)	1106.86	(428, 490)	118.910
<b>j</b>	(669, 320)	2488.45	(307, 275)	1092.50	(642, 501)	118.468
<b>k</b>	(187, 318)	2431.59	(652, 316)	971.29	(547, 319)	115.907
<b>l</b>	(76, 317)	1845.75	(75, 323)	807.31	(308, 318)	109.441
<b>m</b>					(187, 319)	98.042
<b>n</b>					(438, 319)	94.783
<b>o</b>					(670, 319)	91.099
<b>p</b>					(75, 323)	54.732

Tabelle 4.3: Lokale Maxima zu den Symmetriekarten von Abbildung 4.20. Aufgeführt sind jeweils nur die ersten  $n$  lokalen Maxima einer Symmetriekarte, die alle gesuchten Symmetriezentren umfassen.

# Kapitel 5

## Zusammenfassung

Im Rahmen des Projekts Bildanalyse zur Ornamentklassifikation werden ornamentale Strukturen in Bildern lokalisiert und klassifiziert. Für die Lokalisierung werden Merkmalsextraktionsverfahren verwendet. In dieser Arbeit wurde ein weiteres solches Verfahren implementiert und auf seine mögliche Verwendung in dem Projekt untersucht. Das Verfahren trägt den Titel ColorSym und wurde von Gunther Heidemann in [Hei04, Hei06] vorgeschlagen. Auf Basis einer kreisförmigen Nachbarschaftsrelation sucht ColorSym nach lokalen Symmetrien in Fabbildern. Jeder Bildpunkt wird dabei als potentiell Symmetriezentrum betrachtet. Das Resultat der Berechnung entspricht einer Symmetriekarte, deren lokale Maxima die Punkte mit der größten Symmetrie angeben. Diese Punkte entsprechen den gesuchten Merkmalen. Die Grundlage der Berechnungen bildet eine Gradientenkarte des Eingabebildes.

ColorSym ist eine Erweiterung von GraySym, das von Reisfeld et al. in [RWY95] vorgeschlagen wurde. Sowohl ColorSym als auch GraySym sind als universelle Verfahren nicht auf ein konkretes Aufgabengebiet beschränkt. GraySym arbeitet im Gegensatz zu ColorSym ausschließlich auf Graubildern und besitzt einige Unterschiede in der Berechnung der Symmetriekarte. Reisfeld et al. unterscheiden bei GraySym eine radiale und eine isotrope Symmetrie, die sich jeweils durch die Verrechnung der Symmetrierichtung voneinander unterscheiden. Heidemann definiert für ColorSym lediglich die isotrope Symmetrie.

Im Rahmen dieser Arbeit wurde daher testweise eine radiale Symmetrie für ColorSym definiert, die sich deutlich an der Definition von Reisfeld et al. für GraySym orientiert.

ColorSym wurde als Operator in PUMA 2 eingebettet. PUMA 2 ist eine Programmierumgebung für Aufgaben aus dem Bereich der Bildverarbeitung und wird entwickelt an der Arbeitsgruppe Aktives Sehen an der Universität Koblenz-Landau, Campus Koblenz. Bei der Implementierung von ColorSym wurde der von Gunther Heidemann bereit gestellte Quellcode als Grundlage verwendet. Die vorgenommenen Änderungen waren überwiegend strukturell und umfassen das Hinzufügen für die Berechnung der radialen Symmetrie. Zusätzlich wurde zu Testzwecken noch eine Kombination von radialer und isotroper Symmetrie implementiert. In Tests konnten jedoch keine relevanten Unterschiede zwischen den drei resultierenden Symmetriekarten nachgewiesen werden.

Evaluiert wurde ColorSym sowohl hinsichtlich seiner allgemeinen Funktion zur Objekterkennung als auch für die spezielle Verwendbarkeit im Projekt Bildanalyse zur Ornamentklassifikation. Das Verfahren lieferte durchaus brauchbare Ergebnisse, zeigte aber in einigen Fällen auch Schwächen. ColorSym verwendet eine sehr eingeschränkte Verwendung des Symmetriebegriffs und ist daher nicht in der Lage, alle möglichen Formen zu erkennen. Hierunter fallen etwa Dreipässe oder vergleichbare Strukturen. Das von ColorSym implementierte Symmetrieverständnis ist optimiert für kreisartige Strukturen. Kreisförmige Objekte lassen sich also gut in Bildern finden. Sind die Mittelpunkte der gefundenen Kreise identisch zu den gesuchten Symmetriezentren, so kann sich dies als Vorteil erweisen. In einigen Tests hat sich aber gezeigt, dass diese Kreise aber auch den gewünschten Zentren widersprechen können und daher die resultierende Symmetriekarte verfälschen. Um diesen Effekt zu reduzieren, bietet es sich an, das Symmetrieverständnis von ColorSym entsprechend zu modifizieren.

Ein ähnliches Ergebnis zeigte sich bei Bildern, die zahlreiche symmetrische Bereiche aufweisen. Darunter fallen etwa einfache Mauerwerke, die eine symmetrische Grundstruktur aufweisen. Diese wird von ColorSym auch gefunden, ist aber i. d. R. gar nicht gefragt. Da Ornamente aber oft auf Mauerwerken oder ähnlichen Strukturen auftauchen, wird ihre genaue Lokalisierung damit erschwert. Auch Balustraden zeigten sich als ein solches Hindernis, da sie ebenfalls eine große, wenn auch irrelevante, Symmetrie aufweisen. Um die Ergebnisse von ColorSym zu verbessern, bietet es sich an, bereits vorab potentiell re-

levante Bildbereiche auszuwählen und nur für diese eine Symmetriekarte zu berechnen. Im Rahmen dieser Arbeit wurde dies noch manuell vorgenommen. Diesen Schritt gilt es daher noch zu automatisieren. Für sich wiederholende Ornamente könnte sich hierzu etwa [Ste06] eignen.

Die Größe der Nachbarschaft, die ColorSym zur Berechnung der lokalen Symmetrie verwendet, wird vorab durch einen Eingabeparameter festgelegt. Dieser Parameter bestimmt dann, wie groß die symmetrischen Bereiche im Bild maximal sein dürfen, um vollständig gefunden werden zu können. In Tests zeigte sich, dass der Parameter möglichst exakt an die Größe der zu suchenden Strukturen angepasst werden sollte. Dies wurde bisher noch manuell vorgenommen. Auch dieser Schritt sollte daher automatisiert werden. Ein anderes Verfahren zur Berechnung lokaler Symmetrie, das ohne eine lokale Nachbarschaft auskommt, haben Loy und Eklundh in [LE06] vorgeschlagen. Das Verfahren besitzt allerdings einen deutlich anderen Ansatz als ColorSym.

ColorSym lässt sich für das Projekts Bildanalyse zur Ornamentklassifikation nur dann sinnvoll verwenden, wenn möglichst genau festgelegt wird, für welche Bildpunkte die lokale Symmetrie mit welchen Parametern berechnet werden soll. Pauschal mit Standardwerten auf das gesamte Eingabebild angewendet, kann das Ergebnis nicht immer für weitere Verarbeitungsprozesse brauchbar sein. Daher sollte ColorSym möglichst mit anderen Verfahren kombiniert eingesetzt werden, mit denen sich entsprechende Einschränkungen vornehmen lassen.



# Anhang A

## Demo-Applikation

### `colorSymmetryDemo`

Wie in Abschnitt 3.3.2 beschrieben, wurde `ColorSym` als Operator in PUMA 2 eingebettet. Die Grundlage bildeten dabei die beiden Klassen `ColorSymmetryOperatorRGB8` und `ColorSymmetryOperatorProRGB8` sowie eine gemeinsame Superklasse. Zur einfacheren Bedienung dieser Klassen – und damit auch zur einfacheren Berechnung der Symmetriekarten – wurde zudem die Demo-Applikation `colorSymmetryDemo` implementiert. Dabei handelt es sich um eine Konsolenapplikation, deren Optionen die genaue Berechnung der Symmetriekarte festlegen. Einige Parameter der oben genannten Klassen sind über diese Optionen konfigurierbar, besitzen aber jeweils einen eingeschränkten Wertebereich und einen vordefinierten Standardwert. Die übrigen Parameter sind fest gesetzt und durch die Demo-Applikation nicht weiter veränderbar. Hierunter fallen lediglich solche Parameter, die i. d. R. ohnehin nicht modifiziert werden müssen. Bei der Erstellung der Demo-Applikation `colorSymmetryDemo` wurde auf eine einfache und schnelle Bedienbarkeit Wert gelegt. Mögliche Einschränkungen bei der Konfigurierbarkeit wurden dabei bewusst in Kauf genommen.

Das einzige Argument, das der Demo-Applikation `colorSymmetryDemo` stets übergeben werden muss, ist der Dateiname des Eingabebildes, für das eine Symmetriekarte berechnet werden soll. Werden keine zusätzlichen Optionen gesetzt, wird eine einzeln-

de Symmetriekarte mit vordefinierten Standardwerten berechnet. Die resultierende Karte bildet eine Instanz der Klasse `ScalarFeatureImage<float>` (vgl. Anhang B). Für diese werden dann mit der Methode `getLocalMaxima(T thresh)` die lokalen Maxima ermittelt. Die Maxima werden schließlich in einer Textdatei gespeichert, deren Format in Abbildung A.1 beschrieben wird. Zusätzlich wird die Symmetriekarte in Form einer Graubild Darstellung gespeichert. Eine Markierung der berechneten lokalen Maxima wird dabei nicht vorgenommen. Die Dateinamen der beiden Ausgabedateien werden jeweils so konstruiert, dass sie den Namen des Eingabebildes als Wortstamm besitzen und die zur Berechnung der Symmetriekarte verwendeten Optionswerte als Postfix enthalten. Der Aufruf der Demo-Applikation erfolgt durch den Befehl

```
./colorSymmetryDemo [<options>] <filename>
```

`<filename>` gibt dabei den Dateinamen des Eingabebildes an. `<options>` kann eine Liste von Optionen umfassen. Jede Option darf dabei maximal nur ein mal auftauchen. Alle zur Verfügung stehenden Optionen von `colorSymmetryDemo` lauten wie folgt:

- `--multi` legt fest, dass für das Eingabebild mehrere Symmetriekarten berechnet werden sollen. Die Symmetriekarten unterscheiden sich dann jeweils durch ihren Maximalradius  $d_{\max}$ . Bestimmt werden die Maximalradien durch die zusätzlichen Optionen `--firstradius` und `--num`.
- `--firstradius <int>` gibt den ersten Maximalradius an, der zur Berechnung verwendet werden soll. Diese Option wird nur dann berücksichtigt, wenn zugleich die Option `--multi` gesetzt ist. Ist dies nicht der Fall, wird der Wert ignoriert. Der kleinste einstellbare Wert für den ersten Maximalradius ist 3, der größte Wert ist 200 und der Standardwert ist 10.
- `--increment <int>` bestimmt die Abstände der zur Berechnung verwendeten Maximalradien. Diese Option wird nur dann berücksichtigt, wenn mehrere Symmetriekarten berechnet werden sollen. Besitzt der erste zu verwendende Maximalradius einen Wert von  $d_0$ , so ergibt sich der zweite Radius durch die Summe  $d_0 + \text{<increment>}$ . Die übrigen Radien werden analog berechnet. Der kleinste mögliche Wert für die Inkrementierung ist 1, der größte Wert ist 50 und der Standardwert ist 5.

- `--num <int>` gibt an, wie viele verschiedene Symmetriekarten insgesamt berechnet werden sollen. Diese Option wird nur dann berücksichtigt, wenn zugleich die Option `--multi` gesetzt ist. Der größte Maximalradius, der zur Berechnung verwendet wird, entspricht dann dem Wert  $d_0 + \text{<increment>} * (\text{<num>} - 1)$ . Der kleinste mögliche Wert für `--num` beträgt 1, der größte Wert entspricht 100 und der Standardwert ist 5.
- `--radius <int>` bestimmt den Maximalradius, der zur Berechnung der Symmetriekarte verwendet werden soll. Im Gegensatz zur Option `--firstradius` wird `--radius` nur dann ausgewertet, wenn lediglich eine einzige Symmetriekarte ermittelt werden soll. Dies ist genau dann der Fall, wenn die Option `--multi` nicht gesetzt ist. Der kleinste mögliche Wert für den Radius ist 3, der größte Wert ist 200 und der Standardwert ist 10.
- `--pwf <pwfType>` gibt an, welche Phasengewichtungsfunktion zur Berechnung verwendet werden soll. `--pwf` kann die Werte 0, 1, 2, 3 und 4 annehmen. Der Standardwert ist 2 und entspricht der Definition von `ColorSym` nach Heidemann in [Hei04, Hei06]. Weitere Informationen zu den einzelnen Phasengewichtungsfunktionen findet sich in Abschnitt 3.3.2.
- `--sigma <float>` bestimmt die Standardabweichung  $\sigma$  des Gaußkerns, der zur Weichzeichnung der resultierenden Symmetriekarte verwendet wird. Eine Distanzgewichtungsfunktion wird nicht verwendet. Der kleinste mögliche Wert für die Standardabweichung beträgt 0.5, der größte Wert ist 10.5 und der Standardwert ist 1.5.
- `--quiet` legt fest, dass sämtliche Ausgaben der Applikation unterdrückt werden sollen. Diese Option unterbindet auch eventuell auftretende Fehlermeldungen.
- `--help` gibt einen kleinen Hilfetext zur Bedienung des Programms auf der Konsole aus. Der Hilfetext liefert eine kurze Beschreibung aller zur Verfügung stehenden Optionen der Demo-Applikation. Der Hilfetext wird auch ausgegeben, wenn mindestens eine Optionen mit einem ungültigen Wert belegt wurde – selbst dann, wenn diese Option selbst gar nicht für die Berechnung verwendet wird.

```
# radius                = <radius>
# pwf type              = <pwfType>
# sigma                = <sigma>
# number of features = <number_of_features>
<x_0> <y_0> <value_0>
<x_1> <y_1> <value_1>
<x_2> <y_2> <value_2>
...
<x_n> <y_n> <value_n>
```

Abbildung A.1: Ausgabeformat der ermittelten lokalen Maxima. Die Maxima werden als Liste in eine Textdatei geschrieben. Für jedes Maximum wird die Bildposition  $\langle x_i \rangle$   $\langle y_i \rangle$  sowie der Symmetriehalt  $\langle value_i \rangle$  gespeichert. Die Anzahl aller lokalen Maxima wird mit  $\langle number\_of\_features \rangle$  angegeben. Ferner beinhaltet die Textdatei die zur Berechnung der Symmetriekarte verwendeten Optionswerte.

# Anhang B

## Die Klasse `FeatureImage`

Merkmalsextraktionsverfahren werden dazu eingesetzt, markante Bereiche in Bildern zu finden, die dann in weiteren Prozessen verarbeitet werden. Diese Bereiche werden als Interessensregionen bezeichnet. Die Gesamtheit der Interessensregionen eines Bildes lässt sich darstellen als eine Sammlung von Merkmalspunkten. Jeder Punkt verweist dabei auf genau eine Interessensregion. Implementieren lässt sich eine Sammlung von Merkmalspunkten etwa durch eine `vector`-Klasse. Eine alternative Darstellung bilden Merkmalskarten oder Interessenskarten, die das Ergebnis eines Merkmalsextraktionsverfahren wieder als Bild darstellen. Dieses Bild ist genauso groß wie das Eingabebild. Jedes gefundene Merkmal entspricht einem Extremum im Merkmalsbild. Beide Darstellungen haben Vor- und Nachteile. Merkmalskarten werden meist von den Merkmalsextraktionsverfahren während der Berechnung als Zwischenablage verwendet und sind damit ohnehin schon vorhanden. Liegt eine solche Bilddarstellung erst einmal vor, lässt sie sich auch weiteren Vorverarbeitungen unterziehen, bis schließlich die eigentlichen Merkmale extrahiert werden. Zudem bieten Merkmalskarten die Möglichkeit, die gewünschten Merkmale auf unterschiedliche Arten zu entnehmen, ohne dabei die gesamte Karte erneut berechnen zu müssen. Sammlungen von Merkmalspunkten hingegen haben den Vorteil, dass sie einfacher zu bedienen sind, da sie näher am benötigten Ausgabeformat liegen. I. d. R. werden ohnehin konkrete Punkte zur weiteren Verarbeitung verwendet und keine ganzen Merkmalskarten.

Das in Kapitel 3 beschriebene Verfahren ColorSym wurde so implementiert, dass es Merkmalskarten berechnet und diese auch als Ergebnis liefert. Dadurch soll eine größere Flexibilität bei der Bestimmung der Merkmalspunkte ermöglicht werden. Ferner macht sich diese Bilddarstellung zu Nutze, weitere Bildoperationen anwenden zu können. Dabei wird im Konkreten eine Konvolution der Merkmalskarte mit einem Gaußkern vorgenommen, wie dies von Gunther Heidemann in [Hei04, Hei06] vorgeschlagen wird.

Da PUMA 2 zur Zeit der Erstellung dieser Arbeit mit Ausnahme der Klasse `EdgeImage` noch keine Klassen zum Darstellen von Merkmalskarten besaß, wurde eine neue Klasse `FeatureImage` eingeführt. Diese bildet eine abstrakte Superklasse für alle konkreten Merkmalskarten. `FeatureImage` ist als Templateklasse implementiert und besitzt mehrere Methoden zum Suchen nach lokalen und globalen Extrema – und damit auch nach Merkmalspunkten. Für jede konkrete Merkmalskarte muss eine separate Klasse implementiert und von `FeatureImage` abgeleitet werden. Um eine einheitliche Struktur für alle Merkmalsrepräsentationen zu schaffen, wurde die bereits vorhandene Klasse `EdgeImage` zu einer Subklasse von `FeatureImage` umgeschrieben. Zusätzlich wurde die Klasse `RGBEdgeImage` implementiert, die als Kantenbild für RGB-Farbbilder dient. Beide Klassen lösen die Templatestruktur ihrer gemeinsamen Superklasse auf, wie dies im UML-Klassendiagramm in Abbildung B.2 dargestellt ist. Damit für alle skalaren Merkmale nicht jeweils eine eigene Klasse implementiert werden muss, wurde ferner die Templateklasse `ScalarFeatureImage` als Subklasse von `FeatureImage` eingeführt. `ScalarFeatureImage` ist im Gegensatz zu ihrer Superklasse nicht abstrakt und kann daher direkt instanziiert werden.

`FeatureImage` bietet acht unterschiedliche Methoden zur Suche nach Extrema an. Dabei wird unterschieden zwischen Maxima und Minima sowie zwischen lokalen und globalen Extrema. All diese Methoden liefern Instanzen der Klasse `Feature` zurück. Ein `Feature` besitzt eine Position  $(x, y)$  sowie einen Wert  $v$ .  $v$  entspricht dem Wert der Merkmalskarte an der Position  $(x, y)$ . Ein `Feature` beschreibt genau dann ein globales Maximum, wenn es keinen anderen Wert  $v'$  in der Merkmalskarte gibt, der größer ist als  $v$ . Globale Minima sind analog definiert. Zur Berechnung von lokalen Maxima wird für jede Position  $(x, y)$  eine kreisförmige Nachbarschaft mit dem Radius  $d$  definiert.  $(x, y)$  bildet dabei das Zentrum des Kreises. Der Wert an dieser Position sei  $v$ .  $(x, y)$  beschreibt genau dann

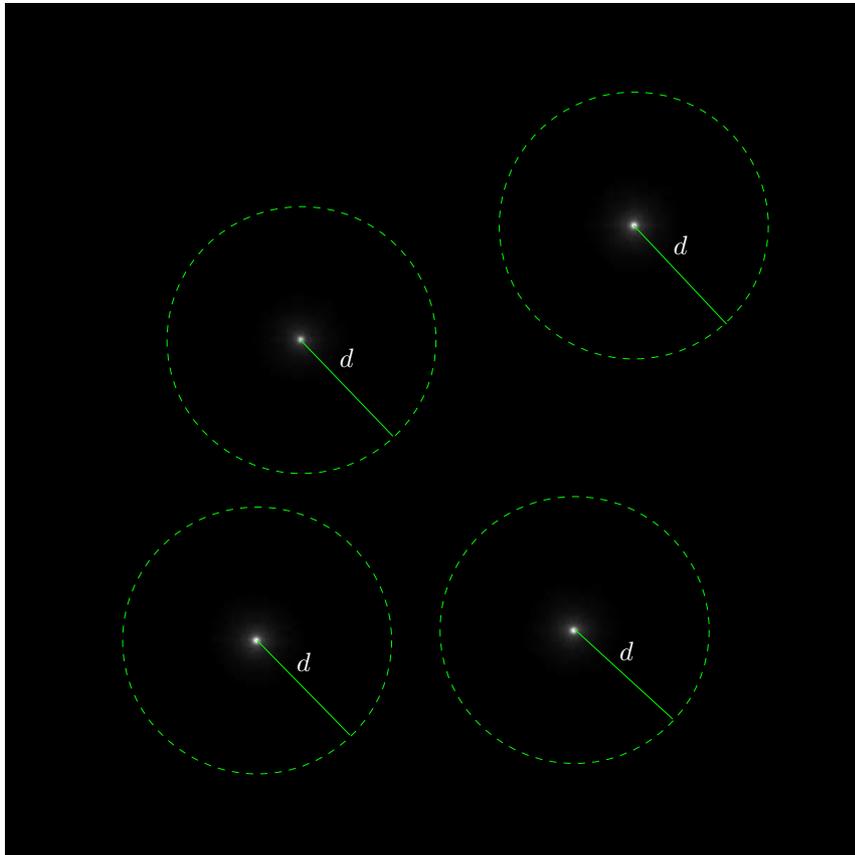


Abbildung B.1: Ein Punkt  $(x, y)$  mit dem Wert  $v$  definiert genau dann ein lokales Extremum, wenn es im Umkreis von  $d$  keinen Punkt  $(x', y')$  gibt mit einem größeren Wert als  $v$ .

ein lokales Maximum, wenn es innerhalb der Nachbarschaft keine andere Position  $(x', y')$  gibt, deren Wert größer ist als  $v$ . Skizziert wird dieser Zusammenhang in [Abbildung B.1](#). Um zu verhindern, dass Werte, die offensichtlich keine lokalen Maxima sind, auch wenn sie gemäß dieses Verfahrens als solche verstanden werden, wurde zusätzlich ein Schwellwert  $t$  eingeführt. Lokale Maxima werden nur dann als solche gewertet, wenn ihr Wert  $v$  größer ist als  $t$ . Hierbei reicht es bereits aus,  $t$  auf den minimal möglichen Wert zu setzen. Bei Merkmalskarten, die nur positive Werte liefern, ist dies etwa der Wert 0. Die Suche nach lokalen Minima erfolgt analog.

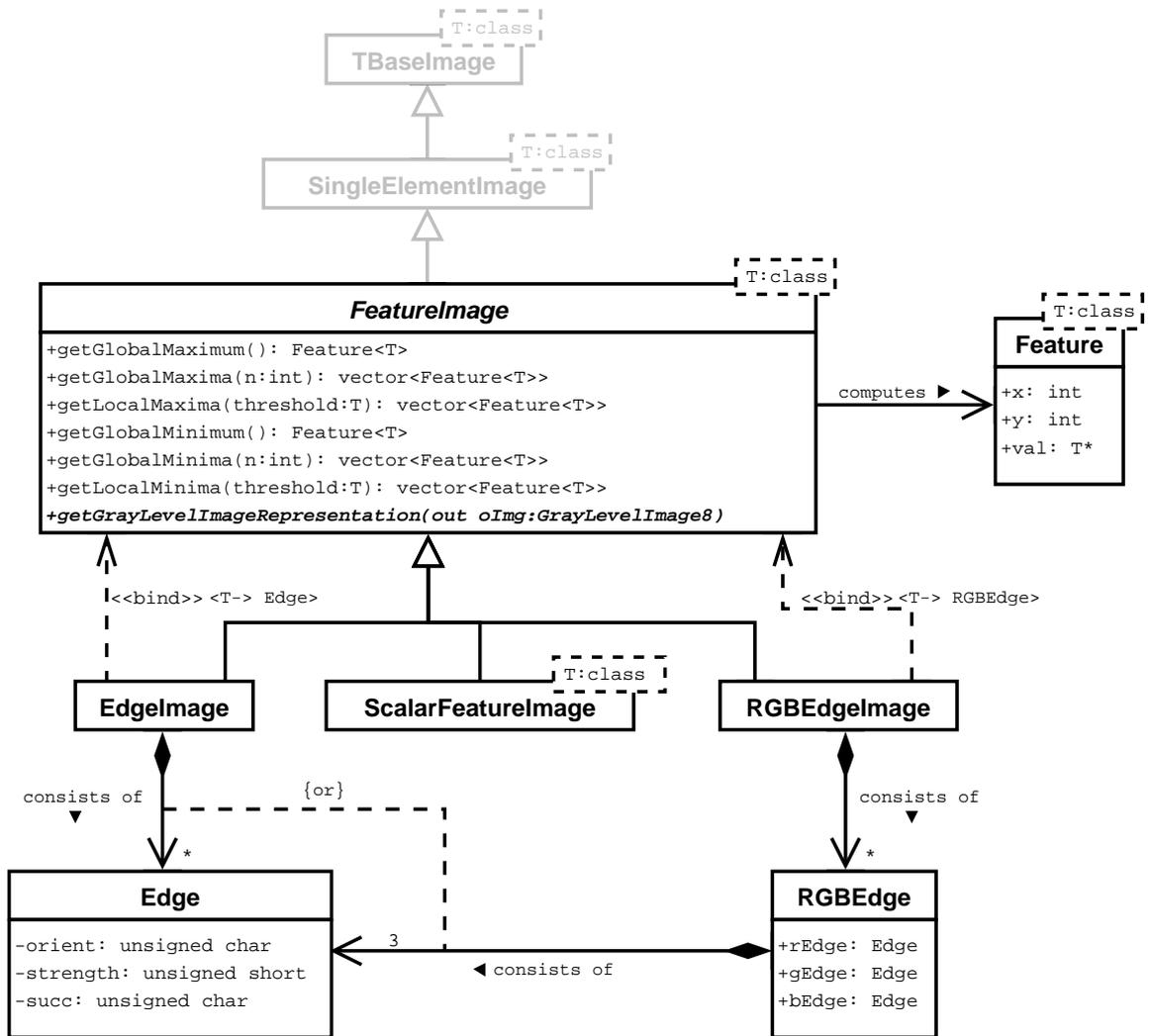


Abbildung B.2: Klassendiagramm um die Klasse `FeatureImage`. Grau unterlegte Klassen waren bereits in PUMA 2 vorhanden und wurden nicht weiter modifiziert. Diese Klassen dienen lediglich zum Herstellen eines Kontextes und damit zur besseren Orientierung. Bei dem Klassendiagramm handelt es sich um eine Vereinfachung, die nicht alle implementierten Methoden darstellt.

# Abbildungsverzeichnis

1.1	Schema des Projekts <i>Bildanalyse zur Ornamentklassifikation</i> . . . . .	8
3.1	Beispiel für ein Paar der Menge $\Gamma(\mathbf{p})$ . . . . .	19
3.2	Beispiel für $\Lambda(\mathbf{p}, l, r)$ . . . . .	20
3.3	Skizze zur Symmetrieachse $\mathbf{s}_{l,r,i,j}$ . . . . .	23
3.4	Skizze zur radialen Symmetrie . . . . .	26
3.5	Beispiel für $\text{PWF}_M(\mathbf{p}, l, r, i, j)$ . . . . .	28
3.6	Struktogramm zur Berechnung der isotropen Symmetrie . . . . .	31
3.7	Struktogramm zur Berechnung der radialen Symmetrie . . . . .	32
3.8	Skizze zum Maximalradius $d_{\max}$ . . . . .	33
3.9	Klassendiagramm um die Klasse <code>ColorSymmetryOperatorBase</code> . . . . .	38
4.1	Kreis . . . . .	42
4.2	Rosa Kreise . . . . .	43
4.3	Schilderwald . . . . .	44
4.4	Dogenpalast . . . . .	45
4.5	Ca' d'Oro . . . . .	46
4.6	Kreis mit Balken . . . . .	47

4.7	Ca' d'Oro, Ausschnitt 1 . . . . .	48
4.8	Ca' d'Oro, Ausschnitt 2 . . . . .	50
4.9	Vergleich zwischen Dreipass und Vierpass . . . . .	52
4.10	Dreipässe . . . . .	53
4.11	Wirbelräder (1) . . . . .	54
4.12	Wirbelräder (2) . . . . .	55
4.13	Flächensymmetrie mit Vierpassen (1) . . . . .	56
4.14	Ornament mit Flächensymmetrie . . . . .	56
4.15	Flächensymmetrie mit Vierpassen (2) . . . . .	58
4.16	Ausschnitt aus Abbildung 4.15 . . . . .	58
4.17	Standtmauer von Worms . . . . .	60
4.18	Dreipässe und Vierpass . . . . .	61
4.19	Schachbrettmuster . . . . .	64
4.20	Cathedrale St. Nazaire . . . . .	65
A.1	Schema des Ausgabeformats von <code>colorSymmetryDemo</code> . . . . .	74
B.1	Skizze zur Berechnung lokaler Extrema . . . . .	77
B.2	Klassendiagramm um die Klasse <code>FeatureImage</code> . . . . .	78

# Literaturverzeichnis

- [Den05] Matthias Dennhardt. Symmetriesuche in ornamentbildern. Studienarbeit, Universität Koblenz-Landau, 2005.
- [DGV95] Vito Di Gesu and Cesare Valenti. The discrete symmetry transform in computer vision. Technical Report Technical Report DMA 011/95, Università di Palermo, Palermo, 1995.
- [Fö5] Andrea Fürsich. Texturanalyse von bildern des augenhintergrunds. Studienarbeit, Universität Koblenz-Landau, 5 2005.
- [FFM90] Jürgen Flachsmeier, Uwe Feiste, and Karl Manteuffel. *Mathematik und ornamentale Kunstformen*. Verlag Harri Deutsch, 1990.
- [Hei04] Gunther Heidemann. Focus on attention from local color symmetries. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(7), 6 2004.
- [Hei06] Gunther Heidemann. Color symmetry for interest point detection. *Workshop Farbbildverarbeitung*, 12, 10 2006.
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.
- [KK83] John R. Kender and Takeo Kanade. Mapping image properties into shape constraints: Skewed symmetry and affine-transformable patterns, and the shape-from-texture paradigm. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 237–257. Academic Press, 1983.

- [LE06] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II*, volume 3952 of *Lecture Notes in Computer Science*, pages 508–521, 2006.
- [LN87] Paul J. Locher and Calvin Nodine. Symmetry catches the eye. In J. K. O'Regan and A. Levy-Schoen, editors, *Eye Movements: from Physiology to Cognition*, 353–361. Elsevier Science, Amsterdam, London, New York, Oxford, Paris, Shannon, Tokyo, 1987.
- [Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [LZ02] Gareth Loy and Alexander Zelinsky. A fast radial symmetry transform for detecting points of interest. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 358–368. Springer Verlag Berlin Heidelberg, 2002.
- [LZ03] Gareth Loy and Alexander Zelinsky. A fast radial symmetry transform for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):959–973, 8 2003.
- [Mar89] Giovanni Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1 1989.
- [ML04] Sinjini Mitra and Yanxi Liu. Local facial asymmetry for expression classification. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, pages 889–894. IEEE, 2004.
- [Mor77] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th International Joint Conference on Artificial Intelligence*, 1977.

- [OO] Robyn Owens and David O'Mara. Measuring bilateral symmetry in three dimensional magnetic resonance images.
- [Pal04] Marta Palmowska. Bildanalyse von ornamentik. Studienarbeit, Universität Koblenz-Landau, 7 2004.
- [PS00] Claudio M. Privitera and Lawrence W. Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):970–982, 9 2000.
- [Rei94] Daniel Reinfeld. *Generalized Symmetry Transforms: Attentional Mechanisms and Face Recognition*. PhD thesis, Tel Aviv University, 1 1994.
- [RWY95] Daniel Reinfeld, Haim Wolfson, and Yehezkel Yeshurun. Context-free attentional operators: The generalized symmetry transform. In Katsushi Ikeuchi and Jean Ponce, editors, *International Journal of Computer Vision*, volume 14, pages 119–130. Springer Netherlands, 3 1995.
- [SHW98] Q. B. Sun, W. M. Huang, and J. K. Wu. Face detection based on color and local symmetry information. In *3rd. International Conference on Face & Gesture Recognition*, pages 130 – 135, 1998.
- [SL97] Gal Sela and Martin D. Levine. Real-time attention for robotic vision. *Real-Time Imaging*, 3(3):173–194, 6 1997.
- [SSP06] Clemens Schmidt, Christian Schneider, and Dietrich Paulus. Knowledge-based image analysis applied to ornaments in arts. In *Visual Media Production (CVMP 2006)*, pages 97–105, 11 2006.
- [Ste06] Stephan Steneberg. Detektion ornamentaler bereiche in architektonischen bildern. Studienarbeit, Universität Koblenz-Landau, Campus Koblenz, Fachbereich 4 Informatik, Institut für Computervisualistik, 6 2006.
- [WC94] Nicholas P. Walmsley and K. M. Curtis. Robust object recognition using symmetry. In Aggelos K. Katsaggelos, editor, *Visual Communications and Image*

*Processing '94*, volume 2308, pages 581–590. The International Society for Optical Engineering, 9 1994.

- [Zab90] Hagit Zabrodsky. Symmetry - a review. Technical Report 90-16, Department of Computer Science, Hebrew University of Jerusalem, 91904 Jerusalem, Israel, 5 1990.
- [ZPA92] Hagit Zabrodsky, S. Peleg, and D. Avnir. A measure of symmetry based on shape similarity. *IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, 1992.
- [ZPA95] Hagit Zabrodsky, Shmuel Peleg, and David Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1154–1166, 1995.
- [ZRF<sup>+</sup>06] Liyan Zhang, Anshuman Razdan, Gerald Farin, John Femiani, Myungsoo Bae, and Charles Lockwood. 3d face authentication and recognition based on bilateral symmetry analysis. *The Visual Computer*, 22(1):43–55, 1 2006.