University of Mannheim/Computer Science

Technical Report TR-2013-004

Analyse von Bildmerkmalen zur Identifikation wichtiger Bildregionen

Jakob Huber, Stephan Kopf, Philipp Schaber
Department of Computer Science IV
University of Mannheim, Mannheim, Germany
{kopf, schaber}@informatik.uni-mannheim.de
huber@pi4.informatik.uni-mannheim.de

ABSTRACT

Eine zuverlässige Erkennung wichtiger Bildregionen ist die Grundlage für viele Verfahren im Bereich der Bildverarbeitung wie beispielsweise bei der Bildkompression, bei Verfahren zur Anpassung der Bildauflösung oder beim Einfügen digitaler Wasserzeichen in Bilder. Es wurde ein System entwickelt, das Merkmalspunkte in Bildern identifiziert und diese nutzt, um wichtige Bildbereiche zu identifizieren. Zur Berechnung der Merkmalspunkte wird das SURF-Verfahren (Speeded Up Robust Features) [4] verwendet. Die gefundenen Merkmale werden in einem zweiten Schritt einzelnen Bildregionen zugeordnet. Die Qualität der ermittelten Regionen sowie das Laufzeitverhalten der verschiedenen Verfahren werden anhand der von [24] veröffentlichten Bilddatenbank analysiert.

Keywords

Image saliency, speeded-up robust features, image regions, watershed, saliency maps

1. EINLEITUNG

Das Erkennen wichtiger Regionen in Bildern ist Grundlage vieler Verfahren im Bereich der Bildverarbeitung. In diesem Bericht werden Verfahren vorgestellt, welche die Relevanz der einzelnen Bildpunkte bestimmen. Das Ergebnis ist eine Wichtigkeitskarte, die Saliency Map genannt wird.

Es existieren vielseitige Anwendungen, die Saliency Maps nutzen, da diese den Bildpunkten semantische Information verleihen. Die unmittelbarste Verwertbarkeit bezieht sich auf die Herausarbeitung von einzelnen Objekten [10, 8], die in weiteren Analyseschritten klassifiziert werden können [16, 18, 30, 17]. Hierbei geht man prinzipiell davon aus, dass eine im Bild markierte und als wichtig bewertete Region ein Objekt darstellt. Somit ist es zum Beispiel möglich bestimmte Objekte in mehreren Bildern zu suchen. Eine weitere Anwendungsmöglichkeit ist die Änderung der Größe eines Bildes oder eines Videos mit gleichzeitiger Veränderung des Seitenverhältnisses [2, 12, 14]. Insbesondere kleine mobile Geräte mit unterschiedlichen Displayauflösungen profitieren von automatischen Adaptionsverfahren [23, 21, 22, 15]. *Seam carving* ist ein bekanntes Adaptionsverfahren in diesem Bereich [3, 13, 22]. Ein weiteres Szenario ist die Erzeugung von Videozusammenfasssungen [26, 19, 20], die zum Teil auch auf Saliency Maps basiert. Desweiteren kann man Saliency Maps zur adaptiven Komprimierung von Bildern verwenden, da nur Regionen mit einer hohen Relevanz exakt gespeichert werden müssen.

Bei der Erzeugung von Saliency Maps geht es vor allem darum Regionen hervorzuheben, die auch der Mensch als wichtig wahrnimmt. Nicht zuletzt wird die Qualität der Verfahren daran gemessen, ob die Ergebnisse den Erwartungen des Menschen entsprechen. Daher existieren viele Verfahren zur Erkennung wichtiger Regionen, die an den Eigenschaften des menschlichen Sehprozesses angelehnt sind. Sie basieren auf der Annahme, dass Objekte vor allem dann auffällig sind, wenn sie sich auf Grund von Kontrastunterschieden von der Umgebung abheben [11].

Die *Saliency-Detection*-Verfahren werden daher im Wesentlichen in die Kategorien biologisch motiviert, das heißt die Verfahren implementieren Erkenntnisse über die menschliche Wahrnehmung, mathematisch basiert sowie eine Kombination beider Ansätze gegliedert [1]. Desweiteren unterscheidet man zwischen *top-down*-Verfahren, wo es darum geht spezifische Objekte zu erkennen und *bottom-up*-Verfahren, die ohne zusätzliche Information wichtige Regionen ermitteln. Letztere sind Bestandteil dieser Arbeit. Im Folgenden werden drei verschiedene Verfahren exemplarisch beschrieben.

1.1 Erkennung wichtiger Bildregionen mittels Kontrastanalyse

[27] entwarfen ein Verfahren, das die lokalen Kontrastunterschiede in einem Bild analysiert und Objekte dann mittels eines Algorithmus (*fuzzy growing*), der die menschliche Wahrnehmung in einem *bottom-up* Suchprozess simuliert, extrahiert. Desweiteren wurde ein Framework entwickelt, das darauf basierend wichtige Punkte und Regionen innerhalb eines Bildes herausarbeiten kann. Es wird die Annahme getroffen, dass der Mensch vor allem Bildbereiche mit hohen Kontrastunterschieden als wichtig wahrnimmt. Um die lokalen Kontrastunterschiede zu bestimmen, wird ein Bild, bezeichnet als Wahrnehmungsfeld, in Wahrnehmungseinheiten eingeteilt und dann für jede Einheit die Summe der Kontraste zu den angrenzenden Regionen als Gewicht berechnet. Die so erstellte Saliency Map beachtet also Farbkontraste, die Ausprägung lokaler Texturen wie auch die ungefähre Form der Regionen.

Zur Überführung der Graustufen-Saliency-Map in ein Binärbild, auf dem die wichtigen Regionen klar hervorgehoben sind, wird der *fuzzy growing*-Algorithmus verwendet. Dafür werden zunächst zwei Grenzwerte berechnet, die festlegen wann Bildpunkte zu einer wichtigen Region gehören und wann nicht. Danach werden, ausgehend von Bildpunkten, die als besonders wichtig bewertet wurden,

angrenzende Bildpunkte, die zwischen den beiden Grenzwerten liegen und mit hoher Wahrscheinlichkeit zur wichtigen Region gehören, ebenfalls zu dieser hinzugefügt.

Die beschriebenen Algorithmen werden in ein Framework eingebunden, das einen *top-down*-Prozess (z.B. zur Gesichtserkennung) sowie einen *bottom-up*-Prozess implementiert. Bei letzterem werden Maxima der Saliency Map als wichtige Bildpunkte bewertet sowie wichtige Regionen mittels des *fuzzy growing*-Algorithmus bestimmt. In einem weiteren Schritt wird zudem noch das visuelle Zentrum des Bildes berechnet und mit einem Rechteck gekennzeichnet.

1.2 Erkennung wichtiger Bildregionen mittels Frequenzanalyse

Das Verfahren von [1] ist inspiriert von dem biologischen Konzept, dass Objekte in einem Bild wichtig sind, wenn der Übergang zum Hintergrund sehr kontrastreich ist. Das Ziel des Verfahrens ist es Saliency Maps von Bildern zu erstellen, die die gleiche Auflösung haben, die Kanten des Objekts klar darstellen und zudem effizient und schnell zu berechnen sind. Dies sind die wesentlichen Ziele diese Verfahrens, da diese Kriterien bei anderen Verfahren als Schwachstellen identifiziert wurden. Um dies zu erreichen werden die sehr niedrigen und sehr hohen Kontrastunterschiede aus dem Bild entfernt. Die niedrigen Kontrastunterschiede werden entfernt, um große Regionen gleichermaßen als wichtig zu markieren. Da die maximalen Kontrastunterschiede an Stellen vermutet werden, an denen das Bild ungenau bzw. fehlerhaft ist, werden diese auch entfernt, da sie sonst das Ergebnis verfälschen würden. Die Relevanz eines Bildpunktes entspricht der euklidischen Distanz des Mittelwerts der Farben des Originalbildes im *L*a*b*-Farbraum zu dem Wert des entsprechenden Bildpunktes in dem Bild, auf dem hohe und niedrige Kontrastunterschiede eliminiert wurden. Somit werden Farb- und Helligkeitsinformationen verwertet um wichtige Regionen zu erkennen.

1.3 Erkennung wichtiger Bildregionen mittels globalem Kontrast

Ein weiteres biologisch motiviertes Verfahren wurde von [7] entwickelt. Es bewertet globale Kontrastunterschiede sowie den räumlichen Zusammenhang der Regionen, um Saliency Maps zu erstellen. In der ersten Version, der histogrammbasierten Kontrastermittlung, wird jedem Bildpunkt basierend auf der Farbdistanz zu allen anderen Bildpunkten ein Gewicht zugeordnet. Das heißt die Bildpunkte, die insgesamt die größte Farbdifferenz zu anderen Bildpunkten haben, werden unabhängig ihrer Position im Bild als besonders wichtig gewertet. Zur Berechnung der Farbdistanz wird die euklidische Distanz zweier Bildpunkte im *L*a*b*-Farbraum verwendet. Um die Laufzeit zu verbessern, werden ähnliche Farben im Bild zusammengefasst, da so deutlich weniger Vergleiche notwendig sind. Desweiteren wird die Formel der Gewichtung nun so modifiziert, dass Farben, die häufig im Bild vorkommen, stärker gewichtet werden. In der umfassenderen Version, der regionenbasierten Kontrastermittlung, werden bessere Ergebnisse erzielt, indem das Bild in Regionen geteilt wird und den einzelnen Regionen dann ein Gewicht basierend auf der räumlichen Lage und dem Kontrast zu anderen Regionen zugeordnet wird. Liegen hohe Kontrastunterschiede zwischen aneinander grenzenden Regionen vor, ist dies ein Zeichen dafür, dass eine der beide Regionen wichtig ist. Im Gegensatz dazu ist ein hoher Kontrastunterschied zwischen entfernt voneinander liegenden Regionen nicht so bedeutend. Daher wird nun für jede Region ein Gewicht bestimmt, das unter allen Regionen die Kontrastunterschiede unter Berücksichtigung der Lage und Größe berechnet.

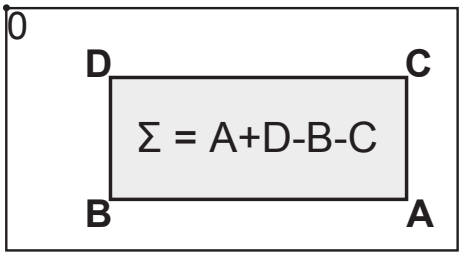


Figure 1: Integralbild. Um die Summe des grauen Rechtecks zu ermitteln sind unabhängig dessen Größe nur 4 Speicherzugriffe (A,B,C,D) und 3 Additionen nötig. (Quelle: [4])

2. SPEEDED-UP ROBUST FEATURES

Zur Beschreibung und Charakterisierung von Bildregionen ist es nötig Informationen zu verwerten, die Aufschluss über die Bedeutung einer bestimmten Region geben. Mittels der zusätzlichen Informationen soll eine Ordnung der Bildregionen gefunden werden, die es erlaubt wichtige Bildregionen zu identifizieren. In diesem Kapitel wird das Verfahren Speeded-Up Robust Features (SURF) genauer beschrieben und Möglichkeiten zur Beschreibung sowie Bewertung von Bildregionen aufgezeigt [4].

SURF ist ein Verfahren zur Erkennung und Beschreibung wichtiger Punkte eines Bildes, so genannter Keypoints. Die Besonderheit des Verfahrens ist, dass es unabhängig der Skalierung sowie der Rotation des Bildes identische Punkte identifizieren kann und dennoch eine sehr gute Laufzeit hat. Somit ist das Verfahren gut geeignet, um gleiche Bildpunkte aus Bildern zu finden, die zum Beispiel aus verschiedenen Blickwinkeln aufgenommen wurden. Zur Beschreibung von Bildregionen eines einzelnen Bildes eignet sich das Verfahren, da es in Bezug auf Genauigkeit, das heißt wichtige Bildpunkte werden als solche erkannt, und Laufzeit gute Ergebnisse erzielt. Das Verfahren besteht im Wesentlichen aus den beiden Abschnitten Entdeckung sowie Beschreibung der wichtigen Bildpunkte, welche im Folgenden erläutert werden. Eine Übersichtsgrafik, die die Zusammenhänge der Phasen darstellt, befindet sich im Anhang.

2.1 Entdeckung wichtiger Bildpunkte

Im ersten Teil des SURF-Verfahrens werden den Pixeln des Bildes Werte zugeordnet, die Aufschluss über die Wichtigkeit und Größe der relevanten Umgebung eines Pixels geben. Da die im SURF-Verfahren eingesetzten Methoden keine Farbinformationen benötigen, wird das Bild zunächst zu einem Graustufenbild konvertiert. Aus dem erstellten Graustufenbild wird dann ein Integralbild [32] des Bildes erstellt, was bedeutet, dass der Wert eines Pixels $\mathbf{x} = (x, y)^T$ die Summe der Werte aller Pixel im Rechteck zwischen dem Ursprung des Bildes und dem Pixel ist (Gleichung 1).

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{1}$$

So ist es möglich die Summe der Werte aller Pixel innerhalb eines beliebig gewählten Rechtecks innerhalb des Bildes mit nur vier Speicherzugriffen und drei Additionen zu bestimmen (Abb. 1). Von dieser Eigenschaft profitiert die Laufzeit des Verfahrens enorm, da diese Aktion in vielen Schritten des Verfahrens nötig ist.

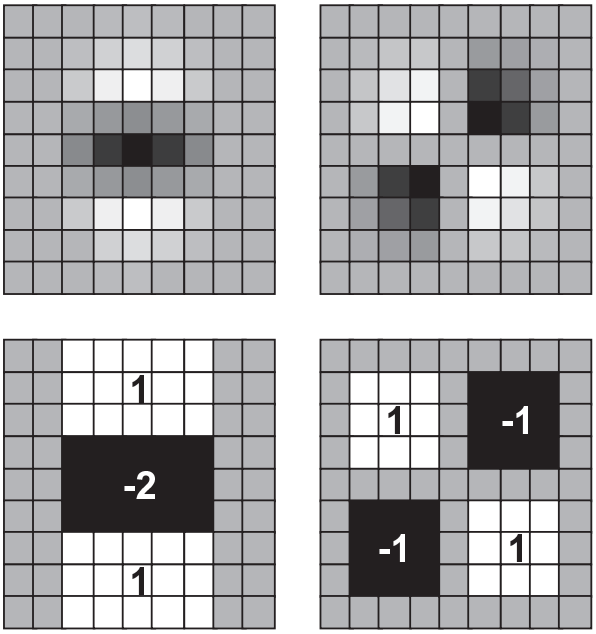


Figure 2: Filter basierend auf der zweiten Ableitung der Gauß-Funktion (abgeschnitten und diskretisiert) in vertikaler (L_{yy}) und diagonaler (L_{xy}) Richtung (oben), entsprechende approximierte Boxfilter D_{yy} und D_{xy} (unten) (Quelle: [4])

2.1.1 Fast-Hessian-Methode

Um die Wichtigkeit eines Pixels zu bestimmen, wird das Bild mit Filtern gefaltet, die auf der zweiten Ableitung der Gauß-Funktion basieren (Abb. 2). So erhält man einen Wert, der Aufschluss über die Kontrastverteilung im Umkreis des Bildpunktes gibt. Da dies in horizontaler (*xx*), vertikaler (*yy*) und diagonaler (*xy*) Richtung geschieht, gibt es drei verschiedene Filter mit denen das Bild gefaltet wird. Die Ergebnisse werden für jeden Punkt in eine Hesse-Matrix übertragen. Eine Hesse-Matrix ist eine Matrix, die alle möglichen Ableitungskombinationen einer Funktion darstellt. Da man sich ein Bild als zweidimensionale Funktion vorstellen kann, gibt es für die erste und zweite Ableitung die Kombinationsmöglichkeiten *xx*, *xy*, *yy* und *yx*. Diese Kombinationen entsprechen den drei Filtern, mit denen das Bild gefaltet wird. Da der Wert für *xy* und *yx* betragsweise gleich ist, wird nur ein diagonaler Filter angewendet. Die Funktion hat als zusätzlichen Parameter die Standardabweichung σ, da die Filter auf der Gauß-Funktion basieren. Mit einer Erhöhung von σ vergrößert sich die berücksichtigte Umgebung des Bildpunktes.

$$\mathcal{H}(\mathbf{x},\sigma) = \begin{pmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{yx}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{pmatrix} \tag{2}$$

Um alle Filter zu kombinieren und nur einen Wert für jeden Pixel zu erhalten, wird die Determinante der Hesse-Matrix bestimmt. Dieser Wert entspricht dem Gewicht des Bildpunktes. Je höher der Wert ist, desto kontrastreicher ist seine Umgebung und er kommt eher als Keypoint in Frage. Dies liegt daran, dass man Ecken innerhalb eines Bildes an kontrastreichen Stellen findet und sich wichtige Bildpunkte häufig in der Umgebung von Ecken befinden. Zudem wird so verhindert, dass Keypoints an horizontalen, vertikalen oder diagonalen Kanten gefunden werden, da die Determinante dann Null ist.

Um die Laufzeit zu verbessern werden die Filter durch stark vereinfachte und diskretisierte Box-Filter (Abb. 2) ersetzt, die sich ideal auf ein Integralbild anwenden lassen und es ermöglichen, dass die Kosten der Anwendung der Filter unabhängig von Filter- und Bildgröße ist. Dabei geht natürlich die Präzision verloren und das Verfahren ist zudem nicht mehr völlig rotationsunabhängig; dennoch rechtfertigt die enorme Verbesserung der Laufzeit diesen Schritt. Neben der Koordinate des Punktes (**x**) hat diese Funktion die Seitenlänge des Filters (*s*) als Parameter. Die Filter in Abbildung 2 basieren auf der Gauß-Funktion mit $\sigma = 1,2$ und werden zu Box-Filtern mit einer Seitenlänge von neun Bildpunkten übertragen.

$$\mathcal{H}_{approx.}(\mathbf{x},s) = \begin{pmatrix} D_{xx}(\mathbf{x},s) & D_{xy}(\mathbf{x},s) \\ D_{xy}(\mathbf{x},s) & D_{yy}(\mathbf{x},s) \end{pmatrix} \tag{3}$$

Um die Approximation etwas zu verbessern, wird bei der Berechnung der Determinante der zweite Summand mit einem Faktor $w \approx 0.912$ angepasst.

$$det(\mathcal{H}_{approx.}) = D_{xx}D_{yy} - (wD_{xy})^2 \qquad w \approx 0.912 \tag{4}$$

Dieser Teil des Verfahrens, das heißt die Anwendung der approximierten Gauß-Filter, die Erstellung der Hesse-Matrix sowie die Berechnung der Determinante, wird als "Fast-Hessian"-Methode bezeichnet.

Abbildung 3 illustriert, an welchen Stellen Keypoints gefunden werden und wie groß diese sind. Die Bilder haben eine Größe von 300x300 Pixel.

Im Folgenden wird beispielhaft gezeigt wie ein Bild mit den Filtern gefaltet wird und wie dann das Gewicht eines Bildpunktes bestimmt wird. Zunächst muss aus einem Graustufenbild ein Integralbild erstellt werden. Da der kleinste Filter eine Seitenlänge von neun Pixeln hat, muss das Beispielbild mindestens 9x9 Pixel groß sein, um die Filter zumindest auf den Bildmittelpunkt anwenden zu können. Das Beispielbild entspricht einem konstanten horizontalen Verlauf. Daher sind keine Keypoints innerhalb des Bildes zu erwarten.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & \mathbf{5} & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix} \xrightarrow{\text{Integralbild}}$$

$$\begin{pmatrix} 1 & 3 & 6 & 10 & 15 & 21 & 28 & 36 & 45 \\ 2 & 6 & 12 & 20 & 30 & 42 & 56 & 72 & 90 \\ 3 & 9 & 18 & 30 & 45 & 63 & 84 & 108 & 135 \\ 4 & 12 & 24 & 40 & 60 & 84 & 112 & 144 & 180 \\ 5 & 15 & 30 & 50 & 75 & 105 & 140 & 180 & 225 \\ 6 & 18 & 36 & 60 & 90 & 126 & 168 & 216 & 270 \\ 7 & 21 & 42 & 70 & 105 & 147 & 196 & 252 & 315 \\ 8 & 24 & 48 & 80 & 120 & 168 & 224 & 288 & 360 \\ 9 & 27 & 54 & 90 & 135 & 189 & 252 & 324 & 405 \end{pmatrix}$$

Basierend auf dem Integralbild ist es möglich die Werte der Hesse-Matrix des mittleren Punktes $\mathbf{x} = (5,5)^T$ des Bildes mit Filtern der Seitlänge 9 zu berechnen und dann die Determinante zu bestimmen:

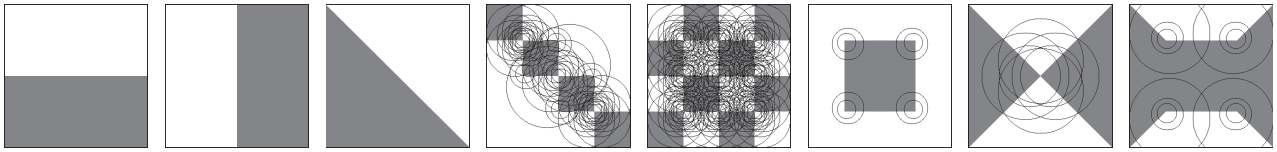


Figure 3: SURF: Beispielbilder. Auf Kanten werden keine Keypoints gefunden (a-c). Besonders viele Keypoints werden bei einem Mustern mit vielen Ecken gefunden (d-f). Sind die Ecken nicht so eindeutig, benötigt es größere Filter (g-h).

Horizontaler Boxfilter:

$$\begin{aligned} D_{xx}(\mathbf{x},9) &= 1*(42+0-0-12)- \\ &\quad 2*(147+12-42-42)+ \\ &\quad 1*(315+42-147-90) \\ &= 1*30-2*75+1*120=0 \end{aligned} \quad (5)$$

Vertikaler Boxfilter:

$$\begin{aligned} D_{yy}(\mathbf{x},9) &= 1*(84+0-9-0)- \\ &\quad 2*(168+9-18-84)+ \\ &\quad 1*(252+18-27-168) \\ &= 1*75-2*75+1*75=0 \end{aligned} \quad (6)$$

Diagonaler Boxfilter:

$$\begin{aligned} D_{xy}(\mathbf{x},9) &= 1*(40+1-4-10)- \\ &\quad 1*(144+15-60-36)- \\ &\quad 1*(80+5-8-50)- \\ &\quad 1*(288+75-120-180) \\ &= 1*27-1*63-1*27+1*63=0 \end{aligned} \quad (7)$$

Determinate von x:

$$det(\mathcal{H}_{approx.}) = D_{xx}D_{yy}-(0.912D_{xy})^2 = 0*0-(0.912*0)^2 = 0$$

Somit bekommt der gewählte Bildpunkt **x** das Gewicht Null zugeordnet und ist daher definitiv kein Keypoint. Dieses Ergebnis entspricht den Erwartungen, da sich der Punkt innerhalb eines konstanten Verlaufes befindet und es keine großen Intensitätsunterschiede gibt, die auf Ecken schließen lassen.

2.1.2 Oktaven und Ebenen

Um zu gewährleisten, dass wichtige Bildpunkte unabhängig des Maßstabes des Bildes erkannt werden, wird die Größe der Box-Filter, die zur Berechnung der Hesse-Matrix genutzt werden, variiert. Da ein Integralbild als Basis dient, entspricht die Vergrößerung der Box-Filter einer Verkleinerung des Bildes. Dies hat den Vorteil, dass man sich die Kosten für eine tatsächliche Verkleinerung des Bildes spart, die bei der Anwendung von Filtern mit konstanter Größe entstehen. Die unterschiedlichen Größen der angewendeten Filter werden Oktaven zugeordnet, wobei eine Oktave einem Skalierungsfaktor von etwa zwei entspricht. Die Filter innerhalb einer Oktave werden als Ebenen bezeichnet (Abb. 4). Mit drei Oktaven, die aus je vier Ebenen bestehen, werden nahezu alle wichtigen Punkte erkannt. Durch weitere Oktaven, die das Bild dann mindestens um den Skalierungsfaktor zehn vergrößern würden, werden kaum zusätzliche Bildpunkte gefunden.

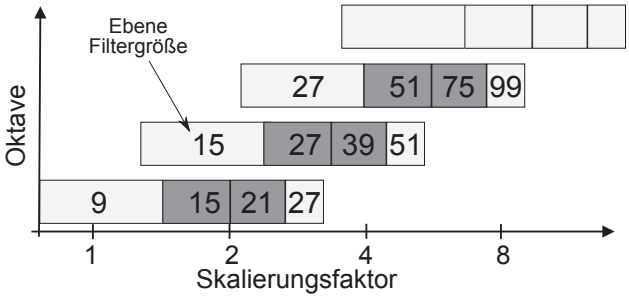


Figure 4: Oktaven sowie deren Ebenen und Filtergrößen (Quelle: [4])

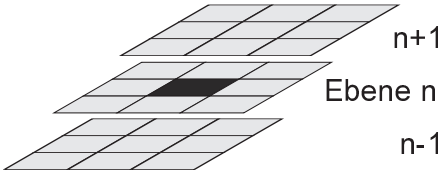


Figure 5: Umgebung zur Auswahl und Interpolation der Keypoints.

Die erste Oktave enthält Filter mit den Größen 9 x 9, 15 x 15, 21 x 21 sowie 27 x 27 (Abb. 4), wobei der kleinste (9 x 9) und größte (27 x 27) Filter nur zu Vergleichszwecken im Rahmen der Auswahl der relevanten Bildpunkte dient. Aus diesem Grund sind manche Filtergrößen in verschiedenen Oktaven zu finden, da man sonst Ebenen bzw. Filtergrößen überspringen würde, was dazu führen kann, dass nicht alle Keypoints gefunden werden. Die Größen ergeben sich dadurch, dass gefordert ist, dass alle Boxen innerhalb der Boxfilter (Abb. 2) einen diskreten Mittelpunkt haben und somit auf jeder Seite um 2 Pixel vergrößert werden müssen. Dies führt in der ersten Oktave dazu, dass die Filter um jeweils 6 Pixel vergrößert werden. Dieser Schritt wird bei nächsthöheren Oktaven immer verdoppelt, da eine Oktave bei der gleichen Anzahl von Ebenen den Skalierungsfaktor insgesamt immer verdoppeln soll.

2.1.3 Auswahl der Keypoints

Damit ein Punkt eines Bildes als wichtiger Punkt erkannt wird, muss sein mittels der "Fast-Hessian"-Methode berechnetes Gewicht in einer 3 x 3 x 3 Umgebung (Abb. 5) maximal sein. Diese Umgebung enthält alle angrenzen Pixel der gleichen Ebene, sowie die gleiche Region der nächsttieferen und nächsthöheren Ebene innerhalb der Oktave. Um eine solche Umgebung erstellen zu können, dient die niedrigste und höchste Ebene jeder Oktave nur für diese Vergleiche. In der ersten Oktave (Abb. 4) können daher mögli-

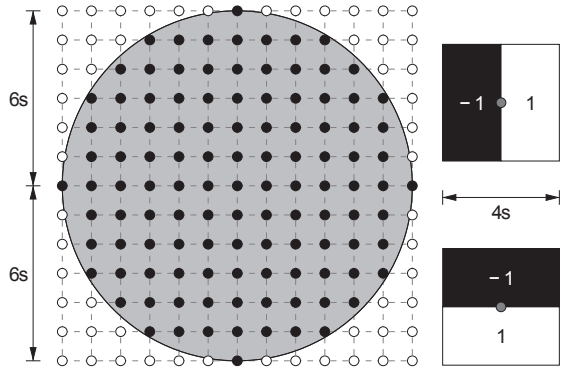


Figure 6: Mittels der Haar-Filter (rechts) werden im Umkreis des Keypoints an 113 Punkten die Haar-Wavelet-Resonanz in x- und y-Richtung bestimmt. (Quelle: [31])

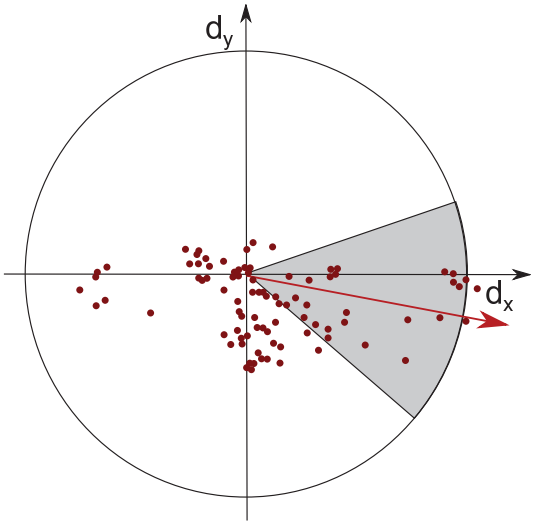


Figure 7: Es wird das Fenster mit der Größe $\frac{\pi}{3}$ so verschoben, dass die Summen der eingeschlossen Resonanzen maximal ist. (Quelle: [4])

che Keypoints nur in den Ebenen 15 x 15 sowie 21 x 21 gefunden werden, da die äußeren Ebenen (9 x 9 und 27 x 27) nur zur Bildung der 3 x 3 x 3 Umgebung dienen bzw. selbst nicht von zwei Ebenen eingeschlossen werden. Ist der Wert eines Bildpunktes innerhalb der Umgebung nicht maximal (*"non-maximum suppression"*), kommt er nicht als Keypoint in Frage. Um in diesem Schritt die Zahl der Keypoints zu minimieren, ist es zudem möglich einen konstanten Grenzwert (*hessian threshold*) festzulegen, den das Gewicht eines Keypoints überschreiten muss. Ist ein Bildpunkt in der dreidimensionalen Umgebung maximal und sein Wert größer als ein möglicher Grenzwert, wird seine Position sowie Größe mittels des Verfahrens von [6] basierend auf den Werten der 3 x 3 x 3 Umgebung interpoliert. Somit ist es möglich, dass letztlich Keypoints eine Größe haben, die keiner Filtergröße entspricht.

2.2 Beschreibung wichtiger Bildpunkte

Im zweiten Teil des SURF-Verfahrens werden weitere Informationen über die Umgebung, der im ersten Schritt ermittelten Keypoints, berechnet. Dies sind bei SURF die Orientierung sowie die auf dieser basierende Beschreibung der Intensitätsverteilung in der Umgebung jedes einzelnen Keypoints. Dieser Teil des Verfahrens ist theoretisch von der Bestimmung der Keypoints komplett unabhängig und benötigt neben der Position nur den Skalierungsfaktor s, der bei der Bestimmung verwendet wurde, als Eingabeinformation. Die folgenden Erläuterungen beziehen sich immer auf genau einen Keypoint.

2.2.1 Bestimmung der Orientierung

Es wird zunächst die Orientierung bestimmt, da diese Grundlage für die Bestimmung der relevanten Umgebung ist, deren Intensitätsverteilung beschrieben werden soll. Dies macht den Deskriptor rotationsunabhägig. Ist diese Eigenschaft nicht gefordert, ist es möglich diesen Schritt zu überspringen und eine Orientierung von 90° festzulegen. Diese Version des SURF-Verfahrens wird *upright version*, kurz *U-SURF*, genannt. Es werden zuerst die Haar-Wavelet Resonanzen in horizontaler (d_x) und vertikaler (d_y) Richtung in einem Umkreis mit dem Radius $6s$ um den Bildpunkt und einer Abtastrate von $1s$ bestimmt. Die hierfür verwendeten Haar-Filter (Abb. 6) haben eine Seitenlänge von $4s$. Somit sind alle relevanten Größen vom Maßstab abhängig, was gewährleistet, dass durch die Verwendung des Integralbildes mit nur sechs Operationen die Resonanz (d_x) oder (d_y) berechnet werden kann. Die ermittelten Resonanzen werden dann mit Gauß ($\sigma = 2s$) um den Bildpunkt gewichtet. So werden die Punkte, die näher am eigentlichen Keypoint liegen, bei der Bestimmung der Orientierung höher gewichtet. Die gewichteten Resonanzen werden dann für jeden Punkt zu einem Vektor (d_x, d_y) kombiniert und in ein Koordinatensystem übertragen. Um dessen Ursprung wird dann ein $\frac{\pi}{3}$ großes Fenster gedreht, um die Orientierung des Keypoints zu bestimmen. Dabei wird für jede Position des rotierenden Fensters ein Gesamtvektor aller Punkte innerhalb des Fensters erstellt, wobei letztendlich der insgesamt größte Vektor dann die Orientierung des Keypoints beschreibt (Abb. 7). Die eigentliche Beschreibung des Keypoints findet dann im nächsten Schritt statt.

2.2.2 Ermittlung des Feature Vektors

Um die Umgebung zu beschreiben, wird ein Quadrat mit der Seitlänge $20s$ und der zuvor ermittelten Orientierung bzw. 90° angelegt. Dieses wird in kleinere 4x4 (je $5s$ x $5s$) Regionen eingeteilt, in denen dann 5x5 Punkte gleichmäßig verteilt werden (Abb. 8). Die mehrstufige Zerteilung in kleinere Regionen verhindert, dass räumliche Informationen verloren gehen. Durch die spätere Kombination der Resonanzen innerhalb der Rechtecke, wird die Größe des Feature-Vektors minimiert, um die Geschwindigkeit eines Vergleiches zweier Vektoren zu verbessern. Für jeden Punkt (insgesamt $16 * 25 = 400$ Punkte) innerhalb der 16 Regionen wird die Haar-Wavelet Resonanz in x und y Richtung berechnet und mit Gauß ($\sigma = 2s$) um den Keypoint gewichtet. Die angewendeten Haar-Filter haben hierbei eine Seitenlänge von $2\ s$. Die Resonanzen werden dann innerhalb der 16 Subregionen summiert und zu einem 4-dimensionalen (4D) Feature-Vektor
$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ zusammengefasst. Die Kombination aller Feature-Vektoren bildet dann den kompletten 64D Feature-Vektor eines Keypoints. Somit enthält der Deskriptor zahlreiche Informationen über die räumliche Verteilung von Intensitätsverläufen sowie deren Polarität.

Es existiert eine umfassendere Version des SURF-Deskriptors, die insgesamt 128 Dimensionen hat (*SURF-128*), wobei nach negativen und positiven d_x bzw. d_y gruppiert wird. Das heißt es wird $\sum d_x, \sum |d_x|$ für $d_y < 0$ und $d_y \geq 0$ bestimmt, gleiches gilt für $\sum d_y$ und $\sum |d_y|$ bezüglich des Vorzeichens von d_x.

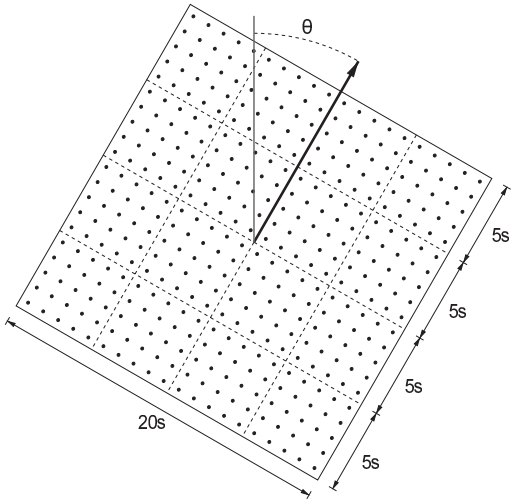


Figure 8: Regionen und Punkte zur Ermittlung des Feature-Vektors. (Quelle: [31])

2.3 Laufzeit

Dieser Abschnitt gibt zusammenfassend einen Überblick über die theoretische Laufzeit der einzelnen Phasen, die zur Bestimmung der SURF-Keypoints nötig sind. Da in dieser Arbeit der Deskriptor und die Orientierung nicht verwendet werden, wird auf deren Laufzeit nicht eingegangen. Es wird davon ausgegangen, dass das Eingabebild drei Farbkanäle hat und *n* Pixel groß ist.

1. Zunächst muss das Bild zu einem Graustufenbild konvertiert werden. Dafür sind für jedes Pixel drei Multiplikation sowie drei Additionen nötig, da jeder Kanal gewichtet und dann alle drei Kanäle zu einem Kanal zusammengefasst werden müssen. Dies ergibt also $\mathcal{O}(n)$.

2. Danach muss das Graustufenbild zu einem Integralbild überführt werden. Für jeden Bildpunkt, mit Ausnahme der Pixel an zwei Bildrändern, sind hierfür vier Addition nötig, was also $\mathcal{O}(n)$ ergibt.

3. Auf das Integralbild werden dann verschiedene Filter angewendet. Für einen Bildpunkt sind das drei Filter, die jeweils drei bzw. vier einzelne Boxen enthalten. Für die Berechnung der Summe aller Pixel innerhalb einer Box benötigt es vier Speicherzugriffe und drei Additionen. Insgesamt müssen 10 Boxen berechnet werden, was also insgesamt 30 Additionen und 40 Speicherzugriffen entspricht. Dazu kommen zusätzlich noch zwei Multiplikationen mit 2, da bei dem horizontalen und vertikalen Filter die mittlere Box mit diesem Faktor gewichtet wird. Die Berechnung der Hesse-Matrix erfordert eine weitere Addition und drei Multiplikationen. Diese Schritte müssen für jeden Bildpunkt und für jede Ebene durchgeführt werden. Dennoch bleibt es auch hier insgesamt bei $\mathcal{O}(n)$.

4. Im nächsten Schritt werden dann die endgültigen Keypoints identifiziert. Sie müssen in einer 3x3x3 Umgebung maximal sein. Das erfordert 27 Speicherzugriff und 26 Vergleiche. Dies wird für jeden Bildpunkt in allen nicht äußeren Ebenen der Oktaven gemacht, wodurch sich eine Laufzeit von $\mathcal{O}(n)$ ergibt.

5. Für alle Punkte, die in einer 3x3x3 Umgebung maximal sind, wird zur Interpolation ein Gleichungssystem aufgestellt, das sich in konstanter Zeit lösen lässt.

2.4 Implementierung in OpenCV

In OpenCV[1] [5] ist eine Implementierung[2] des SURF-Verfahrens enthalten, die alle Funktionen und Varianten abdeckt, die zuvor beschrieben wurden. Man kann die Anzahl der Oktaven und Ebenen je Oktave festlegen, einen Grenzwert für die Determinante der Hesse-Matrix setzen, zwischen den beiden Größen des Deskriptors wählen (64D oder 128D), sowie die Bestimmung des Winkels deaktivieren und somit *U-SURF* verwenden. Desweiteren ist es möglich die Bestimmung der Keypoints zu überspringen, indem man direkt Bildpunkte übergibt, für die der Deskriptor erstellt werden soll. Es ist zu beachten, dass sich die Anzahl der Ebenen je Oktave nur auf die mittleren Ebenen bezieht. Die beiden äußeren Ebenen, die nur für Vergleichszwecke dienen, werden intern hinzufügt.

Als Ergebnis erhält man eine Liste von Keypoints sowie die Deskriptoren aller Keypoints zusammengefasst zu einer weiteren Liste. Ein Keypoint enthält Informationen über dessen Position, den Durchmesser der relevanten Umgebung, dessen Gewicht, der Orientierung sowie die Oktave, in der er entdeckt wurde. Da der Deskriptor nur eine Liste ist, muss man wissen, dass er zeilenweise von links oben nach rechts unten erstellt wurde.

In der SURF Implementierung in OpenCV 2.3 ist die Bestimmung der Oktave bzw. die Zuordnung eines Keypoints zu einer Oktave fehlerhaft, so dass die Oktave, die beim Keypoint angegeben ist, nicht korrekt sein muss. Desweiteren stimmen die Ebenen in höheren Oktaven nicht exakt mit denen überein die [4] nennen.

2.5 Ansätze und Ideen zur Beschreibung von Regionen

Es existieren grundsätzlich zwei Ansätze, die man verfolgen kann, wenn man Keypoints zur Beschreibung von Regionen verwenden möchte. Zum einen ist es möglich die Eigenschaften der Keypoints auf bereits bestimmte Regionen zu übertragen und so deren Bedeutung zu bestimmen, zum anderen kann man sich theoretisch auch vorstellen, Keypoints zur Bestimmung von Regionen zu nutzen bzw. direkt zu markieren. Die folgenden Ansätze nutzen die Position, Größe sowie das Gewicht eines Keypoints.

Der Deskriptor der Umgebung sowie die Orientierung des Keypoints werden nicht verwendet. Dies liegt daran, dass der Deskriptor eher dafür geeignet ist Vektoren aus verschiedenen Bildern zu vergleichen. Die Orientierung beschreibt letztlich nur die Orientierung der Intensitätsverteilung und es ist nicht einfach festzustellen, welchen Bezug diese zur wichtigen Region hat.

2.5.1 Bewertung von Regionen

Dieser Ansatz geht davon aus, dass bereits Regionen im Bild identifiziert wurden, denen ein Gewicht zugeordnet werden muss, das Aufschluss über deren Bedeutung gibt. Hierbei gibt es folgenden Möglichkeiten:

- Keypointdichte
  Dieser Ansatz geht davon aus, dass eine hohe Keypointdichte darauf schließen lässt, dass die Region eine hohe Bedeutung hat. Das Gewicht einer Region entspricht daher der *Anzahl der Keypoints* geteilt durch die *Größe der Region*. Sind also viele Keypoints in einer recht kleinen Region, kann man das als Zeichen dafür werten, dass die Region wichtig ist. Der

[1] Open Computer Vision Library v2.3.1 www.opencv.org

[2] `http://opencv.itseez.com/modules/nonfree/doc/feature_detection.html\#surf`

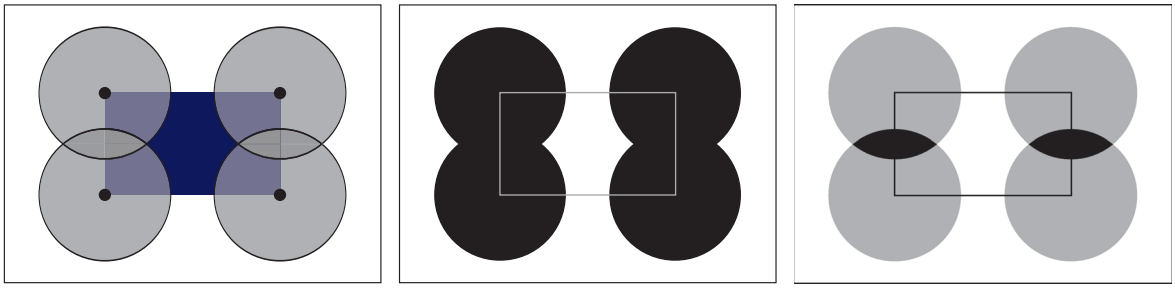


Figure 9: (a) Ein Beispielbild mit vier Keypoints, deren Umgebung markiert ist. Die Bilder (b) und (c) zeigen die markierten Regionen absolut (b) bzw. mittels Addition (c), wodurch die Schnittflächen ein höheres Gewicht erhalten. Jeder Pixel ist dann ein Keypoint, der der Region (dem Viereck) zugeordnet werden muss.

Nachteil dieser Variante ist, dass das Gewicht und die Größe der Keypoints nicht berücksichtigt werden. Zudem werden Keypoints, die sich knapp außerhalb der Region befinden, nicht berücksichtigt, obwohl bei deren Entdeckung auch die Kontrastverteilung innerhalb der Region beachtet wurde. Der Vorteil dieser Variante ist, dass sie ziemlich einfach und schnell ist.

- Summe des Gewichts aller Keypoints in einer Region
  Der vorhergehende Ansatz wird um die Beachtung des Gewichts der Keypoints ergänzt. Anstatt die Keypoints einfach zu zählen, werden nun deren Gewichte addiert. So können auch Regionen mit wenigen Keypoints, die allerdings ein hohes Gewicht haben, als wichtige Regionen beurteilt werden.

2.5.2 Bestimmung von Regionen

Da Keypoints neben ihrer Position und dem Gewicht auch eine Größe haben, ist es theoretisch auch möglich nur mittels der Keypoints wichtige Bildregionen zu kennzeichnen. Hierbei wird das gesamte Umfeld, dessen Durchmesser der Größe des Keypoints entspricht, als wichtig markiert. Um das Ergebnis zu optimieren kann es sinnvoll sein einen Grenzwert einzuführen, um Keypoints nicht zu berücksichtigen, die diesen unterschreiten. Das Ergebnis ist also ein Binärbild, auf dem die wichtigen Regionen hervorgehoben sind. Der Vorteil dieser Variante ist, dass sie äußerst einfach umzusetzen ist und es nicht erforderlich ist zuerst Regionen zu bestimmen, wodurch zudem eine sehr gute Laufzeit erzielt werden kann. Es könnte ein Nachteil sein, dass man das Gewicht der Keypoints nicht berücksichtigt, allerdings ist es so, dass Keypoints mit hohem Gewicht auch einen größeren Durchmesser haben. Somit ist das Gewicht zumindest indirekt durch die Größe berücksichtigt.

Eine kompliziertere Variante ist, dass der Umkreis des Keypoints mit dem tatsächlichen Gewicht bewertet wird. Sollten sich so mehrere Keypoints überlagern, wird das Gewicht der sich überlagernden Keypoints an den betreffenden Stellen summiert. Somit kann eine hohe Keypointdichte an bestimmten Stellen das niedrige Gewicht der einzelnen Keypoints ausgleichen. Der Vorteil dieser Variante ist, dass man keinen Grenzwert einführen muss und zudem ein differenzierteres Ergebnis erhält. Die Laufzeit ist allerdings nicht so gut, da eine große Anzahl von Additionen nötig sind, um Überlagerungen tatsächlich zu berücksichtigen.

2.5.3 Kombination beider Ansätze

Eine weitere Möglichkeit ist eine Kombination der zuvor beschriebenen Ansätze (Abb.9). Das heißt es wird zunächst ein Bild erstellt, auf dem die Regionen nur mittels Keypoints gekennzeichnet sind (Abb.9 (b) + (c)). Dabei können beide Varianten genutzt

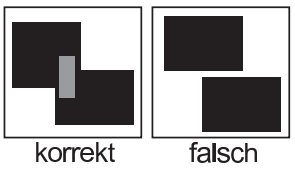


Figure 10: Die Farben auf den Bildern kennzeichnen die Regionen. Auf dem linken Bild ist ein korrektes Beispiel für die Verteilung von Regionen zu sehen. Auf dem rechten Bild gibt es zwei nicht verbundene Regionen, die gleich bezeichnet sind.

werden, die im letzen Abschnitt beschrieben wurden. Auf dem so erstellen Bild entspricht dann jeder Bildpunkt einem Keypoint, der nur eine Größe und ein Gewicht hat. Diese Informationen können dann mittels eines Verfahrens, das auf einer Variante der Keypointdichte basiert, auf Regionen übertragen werden. Dies hat den Vorteil, dass man bei der Verwendung von gekennzeichneten Regionen einen Keypoint mehreren Regionen zuordnet, wenn er im Grenzbereich der Regionen liegt. Dies macht besonders dann Sinn, wenn sich der Keypoint knapp außerhalb der tatsächlich relevanten Region befindet und sein Gewicht bei anderen Ansätzen einzig der unwichtigen Region angerechnet wird.

3. BILDREGIONEN

Dieses Kapitel soll Möglichkeiten aufzeigen, um Bildregionen zu erkennen. Unter einer Bildregion versteht man eine Gruppe von zusammenhängenden Bildpunkten, die auf Grund gemeinsamer Eigenschaften als solche wahrgenommen wird. Desweiteren soll dargestellt werden, welche Möglichkeiten bestehen diese zu bewerten, um zu entscheiden, welche Regionen eine hohe Relevanz haben.

3.1 Erkennung der Bildregionen

Die Erkennung der Regionen basiert darauf, dass alle Bildpunkte, die zusammenhängend sind und ähnliche Eigenschaften (z.B. Helligkeit) haben, als eine Region erkannt werden. Es ist möglich, dass eine Region andere Regionen umschließt (Abb. 10). Damit die Verfahren, die in Abschnitt 3.2 beschrieben werden, korrekt funktionieren, ist es wichtig, dass diese beiden Kriterien erfüllt sind. In diesem Kapitel werden zwei verschiedene Verfahren zum Bestimmen der Regionen erläutert. Das erste Verfahren (Abschnitt 3.1.1) gruppiert Bildpunkte, die eine ähnliche Helligkeit haben, das zweite Verfahren (Abschnitt 3.1.2) ist das *watershed*-Verfahren von [28].

3.1.1 Regionen mit ähnlicher Helligkeit

Bei diesem Verfahren wird das Bild zunächst zu einem Graustufenbild konvertiert und dann geglättet, bevor Bildpunkte mit ähnlicher Helligkeit zusammengefasst werden.

Konvertierung zu einem Graustufenbild
Generell ist es so, dass ein Bild im *RGB*-Farbraum eingelesen wird. Daher muss es in Graustufenbild konvertiert werden, welches die Helligkeit der verschiedenen Farben darstellt. Dabei werden die drei Kanäle gewichtet und zu einem Kanal zusammengefasst. Die Gewichtung entspricht dem Einfluss auf die Helligkeit, das heißt Farbkanäle, die als heller wahrgenommen werden, werden höher gewichtet [9]:

$$Helligkeit = 0,299 * R + 0,587 * G + 0,114 * B. \quad (8)$$

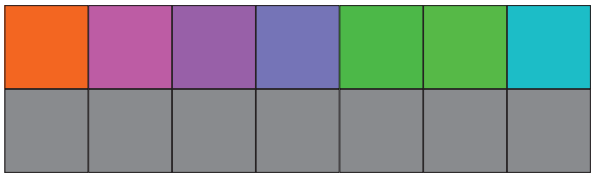


Figure 11: Konvertierung eines Bildes aus dem RGB-Farbraum (1. Zeile) in ein Graustufenbild (2. Zeile). Alle Farben erhalten den Graustufenwert 136.

Dies hat natürlich einen Informationsverlust zur Folge. Bewegen sich die Werte der einzelnen Kanäle im *RGB*-Bild zwischen 0 und 255, gibt es ca. 16,78 Millionen Kombinationsmöglichkeiten, die nun auf 256 Werte ($\approx 0,00015\%$) reduziert werden bzw. eine Graustufe lässt sich 65.536 Farben zuordnen. Daher haben Farben, die vor der Konvertierung klar voneinander zu unterscheiden waren, nun den gleichen Wert (Abb. 11). Das hat natürlich zur Folge, dass dieses Verfahren Bildpunkte, die aus diesen Farben bestehen zu einer Region zusammenfasst, obwohl man sie in einem Farbbild klar voneinander unterscheiden kann und somit verschiedenen Regionen zuordnen würde. Der Vorteil der Verwendung eines Graustufenbildes ist, dass das Verfahren Regionen schneller bestimmen kann, da nur ein Kanal betrachtet und verglichen werden muss. Desweiteren nimmt der Mensch Helligkeitsunterschiede besser als Farbunterschiede wahr, so dass die Nachteile nicht so gravierend sind, wie es auf den ersten Blick scheint [25].

Glättung des Bildes
Da in der Regel nur sehr wenige, aber große Regionen in einem Bild als wichtig wahrgenommen werden, gilt es zu vermeiden, dass viele kleine Regionen erkannt werden. Um dies zu verhindern kann man das Bild vor der Erkennung der Regionen mit einem *Gauß*-Filter [29] glätten. Das hat zur Folge hat, dass es etwas unschärfer wird, wodurch leichter zusammenhängende Regionen erkannt werden können und es somit weniger kleine Regionen gibt.

Regionen markieren
Nachdem das Bild zu einem Graustufenbild konvertiert und geglättet ist, können die Regionen markiert werden. Ausgehend von einem Punkt werden alle angrenzenden Punkte zur Region hinzugefügt, die eine ähnliche Helligkeit haben. Da das Graustufenbild einen maximalen Wertebereich von 0 bis 255 hat, muss man einen Wert festlegen, um den ein anderer Bildpunkt heller oder dunkler sein darf, damit er zur Region hinzugefügt wird. Um zu vermeiden, dass Helligkeitsverläufe als eine Region erkannt werden, ist der Wert des ersten Bildpunktes der Region der konstante Referenzwert, das heißt er wird für alle Vergleiche genutzt. Setzt man z.B. 20 als maximale Helligkeitsdifferenz und wählt einen Bildpunkt mit dem Wert 100 als Startpunkt, dürfen sich angrenzende Bildpunkte im Wertebereich von 80 bis 120 befinden. Um die Regionen zu bestimmen, werden folgende Schritte ausgeführt, bis alle Bildpunkte einer Region zugewiesen sind:

1. Wähle einen Bildpunkt, der noch keiner Region zugewiesen ist, als Startpunkt. Gebe ihm eine neue Regionennummer und speichere seinen Wert für die folgenden Vergleiche.

2. Betrachte die angrenzenden Bildpunkte (*oben, unten, rechts und links*) des Bildpunktes. Wenn ein Bildpunkt noch keiner Region zugewiesen wurde und seine Helligkeit, der des zuerst gewählten Bildpunktes ähnelt, wird er der Region hinzugefügt, das heißt er erhält die gleiche Regionennummer wie der Startpunkt der Region. Danach werden auch seine Nachbarn betrachtet.

3. Wenn alle angrenzenden Bildpunkte der Region nicht mehr zu dieser hinzugefügt werden können, geht es bei Punkt 1 weiter.

Im Anhang sind Beispielbilder zu diesem Verfahren zu sehen.

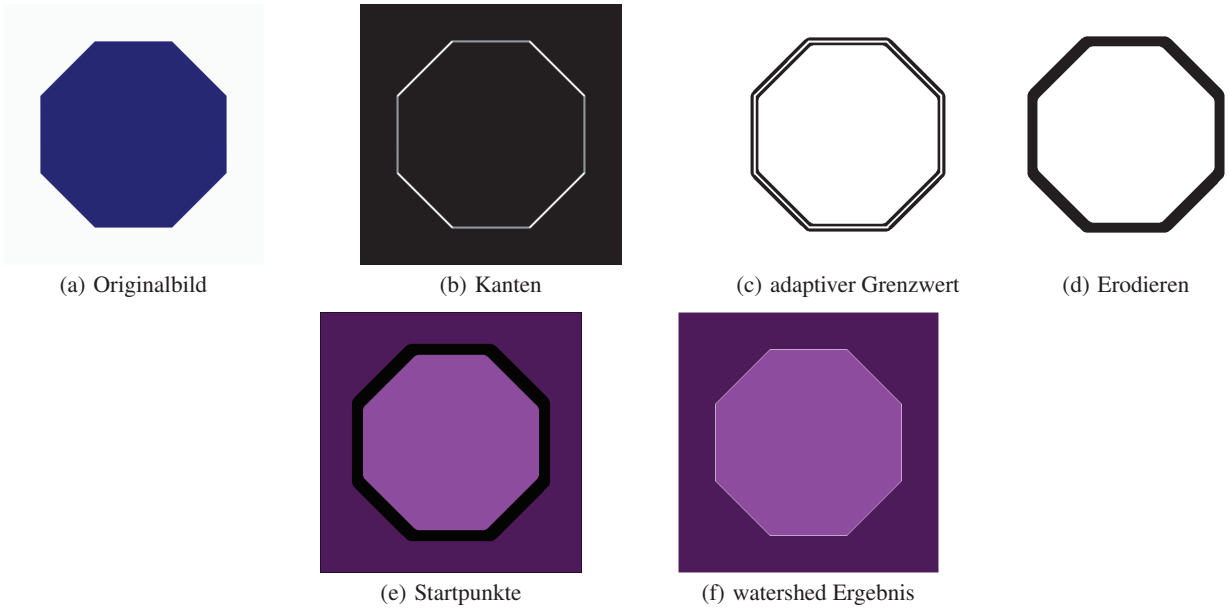
3.1.2 Das Watershed-Verfahren

Das *watershed*-Verfahren [28] betrachtet ein Bild als Topologie, wobei in einem Graustufenbild schwarz (0) dem tiefsten und weiß (255) dem höchsten Punkt innerhalb des Reliefs entspricht. Dies lässt sich auf Farbbilder übertragen, in dem man die Differenz zwischen Farben bestimmt. Ausgehend von Startpunkten (*Markern*) wird das Bild geflutet. Das heißt, einfach ausgedrückt, alle Startpunkte bzw. -regionen werden solange vergrößert, bis sie mit einer anderen Region kollidieren. Daher entspricht die Anzahl der Startregionen der Anzahl der ermittelten Regionen. Bevor der *watershed*-Algorithmus ausgeführt werden kann, müssen zunächst die *Marker* der Regionen bestimmt werden.

Ermittlung der Startpunkte
Um den *watershed*-Algorithmus auf ein Bild anwenden zu können, ist es nötig bestimmte Bildpunkte bzw. Bereiche festen Regionen zuzuordnen. Ausgehend von diesen Startpunkten können dann die exakten Bildregionen ermittelt werden. Um eine grobe Einteilung in Bildregionen zu erreichen, müssen zunächst die Kanten des Bildes herausgearbeitet werden. Bereiche, die dann von Kanten komplett umschlossen sind, können dann als Startregionen genutzt werden. Dabei wird wie folgt vorgegangen:

Schritt 1: Um die Kanten des Bildes zu finden, werden die einzelnen Kanäle eines Farbbildes im *YCrCb*-Farbraum mit einem *Gauß*-Filter geglättet und danach mit *Sobel*-Filtern [29] gefaltet. So entsteht für jeden Kanal ein Graustufenbild, auf dem die Bildpunkte, die auf Kanten liegen, mit verschiedenen Grautönen hervorgehoben sind. Je heller der Grauton ist, desto stärker ist die Kante im Originalbild an der entsprechenden Stelle ausgeprägt. Danach werden die drei Bilder wieder zu einem Bild zusammengesetzt, indem jeweils der maximale Wert für jeden Bildpunkt gewählt wird (Abb. 12 (b)). Theoretisch kann man auch ein Graustufenbild zur Bestimmung der Kanten nutzen, allerdings hat dies den Nachteil, dass man dann Kanten zwischen verschiedenen Farben, die eine ähnliche Helligkeit haben, nicht findet.

Schritt 2: Da man zum Bestimmen einer groben Einteilung des Bildes in Regionen ein Binärbild benötigt, muss das Graustufenbild, auf dem die Kanten gekennzeichnet sind, zu einem solchem



(a) Originalbild (b) Kanten (c) adaptiver Grenzwert (d) Erodieren

(e) Startpunkte (f) watershed Ergebnis

Figure 12: Berechnung der Marker für das watershed Verfahren (a-e), sowie das Ergebnis des watershed-Algorithmus (f).

konvertiert werden. Der Vorteil eines Binärbildes ist, dass eindeutig unterschieden werden kann, welche Bildpunkte in Kantennähe liegen und welche einer Region zugeordnet werden können. Die weißen Bildpunkte repräsentieren mögliche Startpunkte, während die schwarzen Bildpunkte erst mit dem *watershed*-Algorithmus einer Region zugeordnet werden. Das bedeutet also, dass Bildpunkte, die im Binärbild schwarz markiert werden, in Kantennähe liegen und eine Zuordnung zu einer Region nicht ohne Weiteres möglich ist. Bildpunkte, die weiß markiert werden, können hingegen direkt einer Region zugeordnet werden.

Zur Überführung des Kantenbildes in ein Binärbild (Abb. 12 (c)) muss ein Grenzwert gesetzt werden, den ein Bildpunkt überschreiten muss, um weiß markiert zu werden. An dieser Stelle wird ein adaptiver Grenzwert verwendet, was bedeutet, dass für jeden Bildpunkt ein individueller Grenzwert berechnet wird. Dieser entspricht dem Mittelwert der Umgebung des Bildpunktes abzüglich einer Konstanten. Der Mittelwert wird reduziert, damit auch Bildpunkte, deren Wert mit dem Mittelwert übereinstimmt oder knapp unter diesem liegt, weiß markiert werden. Dies ist sinnvoll, da solche Bildpunkte nicht in Kantennähe liegen.

Somit werden letztlich nur Bildpunkte schwarz markiert, die sich neben Kanten befinden (Abb. 12 (c)), da ihr Wert deutlich kleiner als der Mittelwert ihrer Umgebung ist. Dies liegt daran, dass Kanten den Mittelwert deutlich erhöhen und die angrenzenden Bildpunkte diesen nicht erreichen. Da die Bildpunkte, die direkt auf einer Kante liegen, zunächst weiß markiert werden, müssen sie in einem nächsten Schritt (Schritt 3) noch schwarz markiert werden.

Die Größe der zu betrachtenden Umgebung sowie die Konstante, um die der Mittelwert reduziert wird, kann man variieren. Wenn man eine größere Umgebung betrachtet, werden tendenziell mehr Bildpunkte schwarz markiert, da der Einflussbereich der Kanten so vergrößert wird. Dies kann man abfangen, indem man den Mittelwert um einen größeren Betrag reduziert. Eine Vergrößerung der betrachteten Umgebung sowie eine Reduktion der Konstanten führen also dazu, dass größere Flächen markiert werden. Desweiteren ist zu beachten, dass mit einer Vergrößerung der Umgebung auch die Breite der ursprünglichen Kante vergrößert wird, so dass man im nächsten Schritt eine größere Fläche überbrücken muss.

Schritt 3: Die ursprünglich gekennzeichneten Kanten sind an dieser Stelle noch weiß markiert (Abb. 12 (c)), da ihr Wert über dem Mittelwert ihrer Umgebung liegt. Dies ist darauf zurückzuführen, dass im Kantenbild einzig Bildpunkte, die auf Kanten liegen, einen hohen Wert haben. Um die ursprünglichen Kanten auch im Binärbild schwarz zu markieren, kann man die bereits als schwarz markierten Regionen, die die Kanten beidseitig umgeben, vergrößern, so dass sich die Lücke schließt (Abb. 12 (d)). Dieser Vorgang wird *Erodieren* genannt [29]. Dabei erhält jeder Bildpunkt den minimalsten Wert in seiner unmittelbaren Umgebung, wodurch die schwarzen Regionen wachsen. Je häufiger man den Vorgang ausführt, desto größer werden also die schwarzen Flächen. Somit sind nach dem *Erodieren* im Vergleich zum Kantenbild (siehe Schritt 1) die Kanten deutlich ungenauer, sowie invertiert (schwarz statt heller Grauton) dargestellt. Dies ist so gewünscht, da der *watershed*-Algorithmus die exakten Regionen bestimmen soll.

Schritt 4: Gegeben ist eine grobe Einteilung in Regionen. Es können Startpunkte gewählt werden, die der *watershed*-Algorithmus nutzen soll, um die Regionen zu bestimmen (Abb. 12 (e)). Zur Bestimmung der Startpunkte müssen zunächst die einzelnen Regionen im Binärbild bestimmt werden. Eine Region im Binärbild ist eine Grupe zusammenhängender weißer Bildpunkte, die von schwarzen Bildpunkten oder dem Bildrand umgeben ist. Es ist zudem möglich, dass sich innerhalb einer Region weitere Regionen befinden. Jede Region erhält eine eigene Nummer, die das *watershed*-Verfahren benötigt, um Bildpunkte, die zu der Region hinzugefügt werden, mit dieser zu kennzeichnen. Es ist möglich alle Bildpunkte einer Region als Startpunkte bzw. Startregion (Abb. 12 (e)) zu verwenden oder nur einzelne Bildpunkte auszuwählen. Beschränkt man sich auf wenige Bildpunkte, statt der gesamten Region, ist darauf zu achten, dass diese zusammenhängend sind, da es sonst möglich ist, dass nicht zusammenhängende Regionen entstehen. Dies stellt für den *watershed*-Algorithmus zwar kein Problem dar, allerdings besteht die Gefahr, dass die Bedingungen, welche Regionen in dieser Arbeit erfüllen müssen, verletzt werden. Wählt man weniger Startpunkte, kann das *watershed*-Verfahren die Regionen flexibler

bestimmen, da jeder Bildpunkt, der als Startpunkt markiert wird, endgültig einer Region zugeordnet ist. Um kleine Regionen zu vermeiden, kann man eine Mindestgröße festlegen, die eine Region im Binärbild überschreiten muss, damit sie Startpunkte setzen darf.

Der watershed-Algorithmus

Der *watershed*-Algorithmus nutzt die Kanäle eines *RGB*-Farbbildes, um die markierten Startpunkte zu erweitern und somit alle Bildpunkte einer Region zuzuordnen. Da das Verfahren ein Bild als Topologie betrachtet, ist es nötig zwei Farben so zu vergleichen, dass man daraus einen Höhenunterschied ableiten kann. Der Unterschied zweier Farben *x* und *y* entspricht hier der größten Differenz eines Farbkanals:

$$Farbdifferenz = max(|R_x - R_y|, |G_x - G_y|, |B_x - B_y|) \quad (9)$$

Das ist natürlich ein äußerst grober Farbvergleich. Dennoch ist er besser als der Vergleich von Graustufen, da eine geringe Farbdistanz nur bei ähnlichen Farbtönen vorliegt. Der Wertebereich der Farbdistanz bewegt sich zwischen 0 und 255. Für jeden dieser 256 Werte wird eine Warteschlange (*First In - First Out*) angelegt, in die dann Bildpunkte eingefügt werden, die noch keiner Region zugewiesen sind. Dabei sinkt die Priorität mit steigender Farbdifferenz. Dies führt dazu, dass zuerst Bildpunkte mit geringer Farbdistanz geflutet werden. Zudem wird so ein Plateau von allen Seiten gleichzeitig geflutet. Sobald eine Warteschlage leer ist, wird sie gelöscht. Sollten später noch Bildpunkte gefunden werden, für deren Farbdistanz keine Warteschlange mehr existiert, werden sie zu der Warteschlange hinzugefügt, die zu diesem Zeitpunkt die höchste Priorität hat.

1. **Initialisierung der Warteschlange.** Bevor die markierten Startregionen vergrößert werden können, müssen die Warteschlangen initialisiert werden (siehe Abb. 30 im Anhang). Dafür werden die angrenzenden Bildpunkte der Startregionen auf die Wartschlangen verteilt. Die Farbdistanz wird im Originalbild zwischen dem angrenzenden Bildpunkt und dem markierten Bildpunkt berechnet. Sobald ein Bildpunkt zu einer Warteschlange hinzugefügt wird, wird er markiert, so dass er nur einmal in eine Warteschlange eingefügt wird. Liegt ein Bildpunkte zwischen zwei unterschiedlich markierten Bildpunkten, ist es egal mit welchem man ihn vergleicht, da er später als Grenzpunkt markiert werden wird.

2. **Vergrößerung der Startregionen.** Zum Vergrößern der Startregionen (*region growing*) werden nacheinander die Bildpunkte aus den Warteschlangen, in der bereits erklärten Reihenfolge, betrachtet (Abb. 30). Hat ein Bildpunkt nur Nachbarn, die zu einer Region gehören oder noch keiner Region zugeordnet sind, wird er zu dieser Region hinzugefügt. Jeder Bildpunkt, der sich in der Warteschlange befindet, hat mindestens einen Nachbarn, der einer Region zugeordnet ist, da dies die Voraussetzung ist, um in die Wartschlange zu gelangen. Hat er Nachbarn, die zu verschiedenen Regionen gehören, wird er als Grenzpunkt markiert. Desweiteren werden alle angrenzenden Bildpunkte, die noch nicht in die Warteschlangen eingefügt wurden, nun hinzugefügt, in dem die Farbdifferenz zum aktuellen Bildpunkt bestimmt wird.

Nachdem die Warteschlangen geleert sind, ist jeder Bildpunkt einer Region zugewiesen oder als Grenzpunkt bzw. Scheidewand zwischen Regionen markiert. Da die weiteren Verfahren verlangen, dass jeder Bildpunkt einer Region zugewiesen ist, werden die entsprechenden Bildpunkte einer anliegenden Region zugewiesen. Dieser Schritt ist nicht Teil des *watershed*-Verfahrens.

3.2 Nachbearbeitung der erkannten Regionen

Nachdem eine grundlegende Einteilung in Regionen stattgefunden hat, das heißt jeder Bildpunkt ist einer Region zugeordnet, gilt es nun diese zu verbessern und Informationen zu sammeln, die es ermöglichen die Regionen ansatzweise zu bewerten. Die folgenden Ausführungen sind vollkommen unabhängig von dem Verfahren, das zur Bestimmung der Regionen genutzt wurde. Daher sind sie theoretisch auch mit anderen Verfahren zur Regionenbestimmung kompatibel.

3.2.1 Hintergrund erkennen

Zunächst gilt es festzustellen, ob bestimmte Regionen als Hintergrund erkannt werden können (Abb. 13). Das Ziel dieses Ansatzes ist es bestimmte Regionen von der Bewertung auszuschließen, da sie keine Bedeutung haben. Dies kann zum Beispiel dann der Fall sein, wenn eine Region mit mehreren Seitenrändern verbunden ist. In einem weiteren Schritt kann man Farbinformationen über die als Hintergrund identifizierten Regionen sammeln und auch Regionen innerhalb des gesamten Bildes dem Hintergrund zuordnen. Folgende Schritte sind zur Umsetzung notwendig:

1. Bestimme für jede der vier Seiten (*oben, rechts, unten, links*) des Bildes die angrenzenden Regionen. Die vier Eckpunkte des Bildes werden nicht berücksichtigt.

2. Bestimme darauf basierend die Regionen, die mit mindestens zwei Seiten verbunden sind, und daher als Hintergrund gewertet werden:
$Hintergrundregionen = (oben \cap rechts) \cup (oben \cap unten) \cup (oben \cap links) \cup (rechts \cap unten) \cup (rechts \cap links) \cup (unten \cap links)$

3. *(optional)* Sammle Farbinformationen über die Regionen, die bis zu diesem Schritt als *Hintergrundregionen* erkannt wurden. Bei einem 1-Kanal Bild kann das z.B. der Mittelwert aller Bildpunkte in der Region sein.

4. *(optional)* Suche Regionen im gesamten Bild, die ähnliche Farbeigenschaften wie die *Hintergrundregionen* haben und füge sie zu den *Hintergrundregionen* hinzu.

5. Markiere alle Regionen, die als *Hintergrundregionen* erkannt wurden, als Hintergrund, so dass diese bei der Bewertung der Regionen (Abschnitt 3.3) nicht berücksichtigt werden.

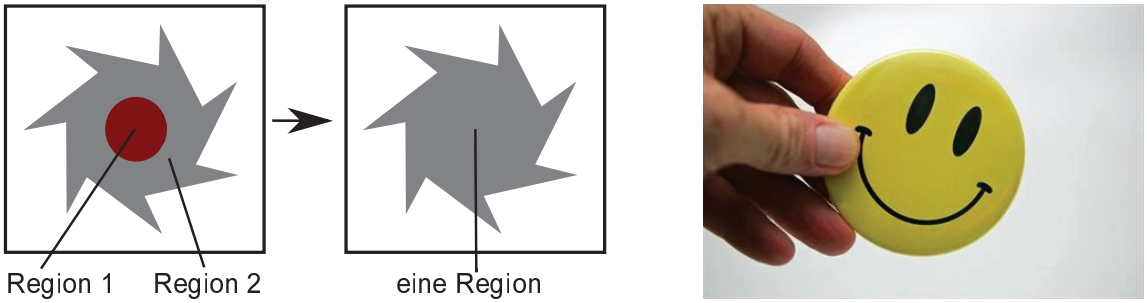
3.2.2 Regionen zusammenfügen

Desweiteren kann man berücksichtigen, dass eine Region, die von anderen Regionen eingeschlossen ist, zur sie umgebenden Region zugeordnet werden sollte (Abb. 14). Ist eine Region von mehr als einer Region umgeben, kann es sinnvoll sein sie zu der Region hinzuzufügen, die den größten Anteil am Rand hält, um so kleine Regionen zu vermeiden. Hierbei ist allerdings zu beachten, dass die umschlossene oder die umschließende Region nicht zum Hintergrund gehört. Würde man eine Region, die als Hintergrund gekennzeichnet ist, wieder zur äußeren Regionen hinzufügen, würde man das Verfahren zur Bestimmung des Hintergrundes (vorheriger Absatz) wieder umkehren. Anderseits macht es auch keinen Sinn eine Region, die vom Hintergrund umgeben ist zu diesem hinzufügen, da man so Objekte, die möglichicherweise wichtig sind, verlieren würde. Diese Überlegungen verdeutlichen die Relevanz eines Verfahrens zur Bestimmung von Hintergrundregionen. Das Ziel ist es die Zahl der Regionen innerhalb des Bildes zu minimieren und kleine Regionen aufzulösen.

Um die Region bzw. die Regionen zu bestimmen, die eine Region umgeben, ist es notwendig, einen Algorithmus zu entwerfen,

(a) Hintergrund erkennen (b) Beispiel aus [24]

Figure 13: Erkennung von Hintergrundregionen:
***(a)* Es werden die Farben Grün und Weiß als Hintergrundfarben erkannt, da diese mit mindestens zwei Seitenrändern verbunden sind. Daher kann optional die Fläche innerhalb des "Sterns" auch als Hintergrund erkannt werden.**
***(b)* reales Beispiel: Der Bereich, der vom Henkel der Kanne umschlossen wird, gehört zum Hintergrund.**



(a) Regionen zusammenfügen (b) Beispiel aus [24]

Figure 14: Regionen zusammenfügen
***(a)* Bei diesem Beispiel kann die dunkelrote Fläche innerhalb des "Sterns" zu diesem hinzugefügt werden, da sie komplett von ihm umschlossen wird.**
***(b)* reales Beispiel: Die Augen kann man dem gelben Pin/Button zuordnen. Für den Mund gilt das nicht unbedingt, da er auch mit dem Daumen der Hand verbunden ist.**

der es ermöglicht den Rand einer Region abzulaufen und dabei Informationen über die angrenzenden Regionen sammelt. Wenn man davon ausgeht, dass man sich bereits am äußeren Rand der Region befindet, gibt es nur drei Möglichkeiten für den nächsten Schritt, da ein digitales Bild diskrete Wert hat, die über ein Raster verteilt sind. Der beschriebene Algorithmus läuft den Rand einer Region im Uhrzeigersinn ab, weshalb sich dieser immer links neben der aktuellen Position (ausgehend von der Bewegungsrichtung) befinden muss. Wenn man die möglichen Schritte auf die wesentlichen zusammenfasst, bleiben neben dem aktuellen Feld (Bildpunkt) und dem Rand nur zwei weitere Felder, die man beachten muss, um den nächsten Schritt zu bestimmen (Abb. 15 a), da man beim Ablaufen des Randes nicht direkt zurück bzw. nach rechts geht. Somit existieren nur drei Möglichkeit für den nächsten Schritt: *90° nach rechts drehen*, *nach oben* oder *diagonal nach links oben* (Abb. 15 (b-d)). Damit es bei diesen Möglichkeiten bleibt, ist bei den Schritten *90° nach rechts drehen* sowie *diagonal nach links oben* die Sicht zur Entscheidung des nächsten Schrittes um 90° nach rechts bzw. links gedreht (Tab. 1). Der Algorithmus bricht ab, wenn der Startpunkt wieder erreicht ist oder man dreimal in Folge den Schritt *90° nach rechts drehen* gewählt hat. Ist letzteres der Fall umfasst die Region nur einen Bildpunkt.

Wählt man beim Start fälschlicherweise nicht den äußeren Rand, was nur möglich ist, wenn die Region selbst andere Region(en) umschließt, läuft man den inneren Rand ab und würde gegebenenfalls die äußere Region zur inneren Region hinzufügen. Da dies nicht gewünscht ist, ist es wichtig, dass der Startpunkt tatsächlich auf dem äußeren Rand liegt. Um dies sicherzustellen, wird immer zuerst die äußerste Region aufgerufen. Dies kann man erreichen, indem man das Bild zeilenweise von links oben nach rechts unten durchgeht und jede Region nur einmal betrachtet.

Nachdem der Rand einer Region abgelaufen ist, hat man Informationen über die Länge des Randes sowie die Häufigkeit der angrenzenden Regionen (Abb. 16). Auf Basis dieser Informationen kann dann entschieden werden, ob die Region zu einer anderen sie umgebenden Region hinzugefügt wird. Hierfür müssen Entscheidungsregeln und zwei Konstanten festgelegt werden:

- **prozentualer Anteil am Rand:** Dieser Wert legt fest um wie viel Prozent eine andere Region, mit Ausnahme des Hintergrunds, die aktuelle Region umranden muss, damit die aktuelle Region zur sie umrandenden Region hinzugefügt wird. Dieser Wert sollte mindestens bei 50% liegen, da sonst keine eindeutige Umrandung vorliegt. Setzt man den Wert auf 100% muss die äußere Region die innere Region komplett umschließen, damit die Regionen vereinigt werden.
- **minimale Länge des Randes:** Da ganz kleine Regionen nicht gewünscht sind, wird ein Wert festgelegt, den die Länge des Randes unterschreiten muss, damit Regionen auch dann zu-

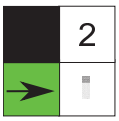
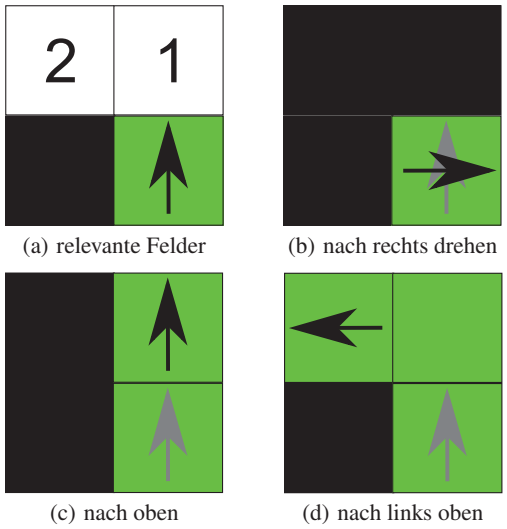
Feld 1:		außerhalb	zugehörig	
Feld 2:		-	außerhalb	zugehörig
Oriertierung *o*		**nächster Schritt**		
0		Orientierung 1	Orientierung 0	Orientierung 3
1		Orientierung 2	Orientierung1	Orientierung 0
2		Orientierung 3	Orientierung 2	Orientierung 1
3		Orientierung 0	Orientierung 3	Orientierung 2
Aktion:		nach rechts drehen	zu Feld 1	zu Feld 2
neue Orientierung *o*:		$o_{neu} = (o_{alt} + 1) \mod 4$	$o_{neu} = o_{alt}$	$o_{neu} = (o_{alt} + 3) \mod 4$

Table 1: Rand einer Region ablaufen. Anhand der aktuellen Orientierung (linke Spalte) werden die beiden Felder bestimmt, die geprüft werden müssen, um den nächsten Schritt zu bestimmen. Gehört Feld 1 nicht zur Region, das heißt es liegt außerhalb der Region, ist der nächste Schritte eine Drehung nach rechts und man muss Feld 2 nicht prüfen. Gehört Feld 2 auch nicht zur Region behält man die Orientierung und geht einen Schritt weiter. Die aktuelle Position/Orientierung ist durch das grüne Feld mit Pfeil bestimmt. Wichtig ist, dass das Feld links (aus Sicht der Orientierung) nicht zur Region gehört.



(a) relevante Felder (b) nach rechts drehen

(c) nach oben (d) nach links oben

Figure 15: Identifikation angrenzender Regionen
***(a)* Diese Grafik stellt die Ausgangslage dar. Die angrenzende Region (schwarzes Feld) liegt links neben der aktuellen Position (grünes Feld). Um den nächsten Schritt zu bestimmen, müssen die mit 1 und 2 markierten Felder ausgelesen werden.**
***(b - d)* Diese Grafiken zeigen die möglichen Schritte. Die eigene Region ist immer grün markiert, die angrenzende Region schwarz. Der graue Pfeil stellt die Sicht vor dem Schritt dar. Der schwarze Pfeil veranschaulicht, zu welchem Feld man sich bewegt und wie sich die Orientierung geändert hat.**

sammengefügt werden, wenn die erste Bedingung nicht erfüllt ist. Hierbei wird die aktuelle Region zu der Region hinzugefügt, die den größten Teil der Umrandung ausmacht. Es ist zudem möglich, dass eine Region zum Hintergrund hinzugefügt wird, wenn dieser den größten Anteil am Rand hat. Um zu vermeiden, dass Bildregionen bestehen bleiben, die deutlich kleiner als 100 Bildpunkte sind, wird ein Wert von 40 festgelegt. Das bedeutet, dass eine Region mit dieser Randlänge maximal 100 (10 x 10) Bildpunkte groß sein kann. Je nach Anordnung der Bildpunkte können aber dennoch kleinere Regionen bestehen bleiben.

3.3 Bewertung der Bildregionen

Da nun die Regionen bereinigt sind, ist es an der Zeit den einzelnen Regionen ein Gewicht zuzuordnen, das Aufschluss über die Bedeutung der Region gibt. Bis jetzt wurden nur Regionen, die als Hintergrund erkannt wurden, ausgeschlossen. Somit ist es prinzipiell möglich alle Regionen, die nicht als Hintergrund erkannt wurden, als gleichermaßen wichtig anzusehen. Da hierbei aber nicht unterschiedlich wichtige Regionen unterschieden werden, gilt es zusätzliche Informationen zu berücksichtigen. In dieser Arbeit soll das SURF-Verfahren als ergänzende Information genutzt werden.

Eine Übersichtsgrafik, die die Zusammenhänge der Phasen darstellt, befindet sich im Anhang.

4. EVALUATION UND LAUFZEITMESSUNGEN

In diesem Kapitel werden die beschriebenen Verfahren und Ansätze evaluiert und bewertet. Die Grundlage aller Tests ist hierbei die Bilddatenbank von [24], die zu jedem Bild ein Binärbild

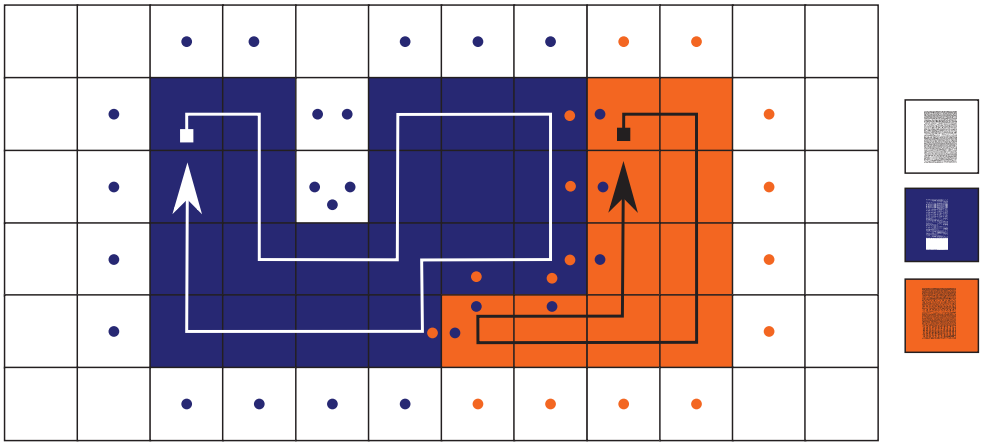


Figure 16: Die Grafik zeigt zwei Regionen (1+2) sowie Hintergrund (0). Die Punkte markieren die angrenzenden Bildpunkte der Regionen. Die Pfeile innerhalb der Regionen verdeutlichen wie der Rand abgelaufen wurde. Region 1 hat eine Kantenlänge von 24, davon fallen 18 (18/24 = 75%) auf den Hintergrund sowie 6 (6/24 = 25%) auf Region 2. Region 2 hat eine Kantenlänge von 16, davon fallen 10 (62,5%) auf den Hintergrund sowie 6 (37,5%) auf Region 1.

enthält, auf dem die wichtigen Regionen gekennzeichnet sind. Es werden zunächst die grundlegenden Eigenschaften der Keypoints des SURF-Verfahrens ermittelt und in einem nächsten Schritt untersucht, wie die gewonnen Informationen auf Regionen übertragen werden können. Bei den Tests steht im Vordergrund zu prüfen, wie genau wichtige Bildregionen durch die beschriebenen Verfahren identifiziert werden.

4.1 Grundlegendes zur Evaluation

Die Bilddatenbank von [24], die zur Evaluation verwendet wird, enthält 1.000 Bilder sowie deren zugehörige Referenzbilder (Abb. 17). Das Referenzbild ist ein Binärbild, auf dem die wichtigen Bildregionen gekennzeichnet sind. Um die Qualität eines Verfahrens zu bewerten, muss man zunächst das Ergebnis der verschiedenen Verfahren in ein Binärbild überführen, um es mit dem Referenzbild vergleichen zu können. Die beschriebenen Verfahren generieren Bilder, auf denen die Relevanz eines Bildpunktes mit Werten zwischen 0 (schwarz, keine Relevanz) und 1 (weiß, höchste Relevanz) bewertet wird. Um ein solches Bild zu einen Binärbild zu konvertieren, muss ein Grenzwert festgelegt werden, der entscheidet, welche Werte auf 0 bzw. 1 gesetzt werden. Setzt man den Grenzwert zu niedrig (hoch) an, ist es möglich, dass zu viele (wenige) Bildpunkte als wichtig bewertet werden. Daher hat der Grenzwert einen wesentlichen Einfluss auf das Ergebnis der Qualitätsbewertung.

Um die Qualität der Verfahren objektiv zu beurteilen, wird geprüft wie gut bzw. genau wichtige Regionen markiert werden. Das heißt es wird untersucht, ob die wichtigen Regionen komplett beschrieben werden (*recall*) und wie genau das Verfahren wichtige Bildregionen beschreibt (*precision*). Zudem wird das harmonische Mittel beider Werte bestimmt (*F-measure*), um einen Wert zu erhalten, der beide Aspekte berücksichtigt. Diese Werte lassen sich wie folgt berechnen:

$$\begin{aligned} recall &= \frac{z}{x} \\ precision &= \frac{z}{y} \\ fmeasure &= \frac{2 * precision * recall}{precision + recall} \end{aligned} \tag{10}$$

- x = Anzahl der wichtigen Bildpunkte im Refenzbild
- y = Anzahl der mit dem zu evaluierenden Verfahren als wichtig markierte Bildpunkte
- $z = x \cap y$ = Anz. der markierten Bildpunkte, die in tatsächlich wichtigen Regionen liegen

Einen hohen *recall* erhält man also, wenn eine große Fläche der wichtigen Regionen markiert wird. Für eine gute *precision* ist es nötig, dass nahezu alle als wichtig markierten Bildpunkte in wichtigen Regionen liegen. Um einen guten *F-measure* Wert zu erreichen, müssen sowohl *recall* als auch *precision* gute Werte haben. Daher ist dieser Wert am ehesten dazu geeignet die Qualität der Verfahren zu beurteilen.

Vergleicht man die Referenzbilder mit weißen Vergleichsbildern, was bedeutet, dass das gesamte Bild als wichtig markiert ist, erhält man natürlich einen *recall* von 100%, da die wichtigen Regionen komplett markiert wurden. Die so ermittelten Werte für *precision* und *F-measure* dienen als Referenzwerte, die ein Verfahren übertreffen sollte. Auf dem Testdatensatz wurde so eine durchschnittliche *precision* von 19,85% erreicht, was bedeutet, dass im Schnitt fast ein Fünftel der Bildfläche zu wichtigen Regionen gehört (Tab. 2).

precision	recall	F-measure
$\approx 0,1985$	1,0	$\approx 0,3211$

Table 2: Informationen über den Testdatensatz: Precision, Recall und F-measure

Anwendung auf einen Datensatz: Wenn man für einen Datensatz *precision*, *recall* und *F-measure* berechnen will, gibt es verschiedene Ansätze, wie man die einzelnen Werte ermitteln kann. Eine Möglichkeit ist *precision*, *recall* und *F-measure* für jedes Bild zu bestimmen und dann die Mittelwerte zu berechnen (*macro*-Variante). Dies hat den Vorteil, dass jeder Eintrag den gleichen Einfluss auf das Gesamtergebnis hat. Die andere Möglichkeit ist, die Werte für x, y und z (siehe oben) über den Datensatz zu summieren und dann die Werte zu berechnen (*micro*-Variante). In den folgenden Abschnitten wird immer die *macro* Variante genutzt, wenn es nicht

(a) Originalbild (b) Referenzbild

Figure 17: Bildpaar aus der Datenbank von [24].

anders angegeben ist.

Testsystem: Alle Tests wurden auf einem Computer mit folgender Konfiguration ausgeführt: Betriebssystem: Windows 7 (64-Bit), CPU: Intel Core2Duo E8400 (3 GHz), Arbeitsspeicher: 4 GB. Die Implementierungen basieren auf der C++ Version von OpenCV 2.3 [5].

4.2 Evaluation des SURF-Verfahrens

Da das SURF-Verfahren in dieser Arbeit Grundlage zur Beschreibung von Regionen ist, werden zunächst die unterschiedlichen Merkmale der SURF-Keypoints untersucht und deren Eignung zur Beschreibung von Regionen bewertet. Um die Tests in einem überschaubaren Rahmen zu halten, werden als konstante Parameter die Anzahl der Oktaven auf 3 und die Anzahl der Ebenen je Oktave auf 4 gesetzt. Wird in diesem Abschnitt von Grenzwert gesprochen, bezieht sich das immer auf das Gewicht eines Keypoints.

4.2.1 Eigenschaften der Keypoints

Um einen Eindruck über die Eigenschaften der Keypoints, die auf dem Testdatensatz gefunden werden, zu gewinnen, werden diese zunächst analysiert. Die Testbilder sind mit einer durchschnittlichen Größe von 373 x 326 Pixel relativ klein, so dass das SURF-Verfahren die Keypoints eines Bildes in nur 58 Millisekunden bestimmen kann (Tab. 3). Da kein Grenzwert angewendet wurde, haben ein Großteil der Keypoints ein niedriges Gewicht und sind vermutlich außerhalb der wichtigen Regionen positioniert (siehe Abschnitt 4.2.2). Da die Größe der Keypoints maßgeblich von der verwendeten Filtergröße abhängt, sind Keypoints, die in höheren Oktaven gefunden werden, in der Regel größer. Desweiteren ist festzustellen, dass mit zunehmender Größe das Gewicht eines Keypoints steigt (Tab. 4).

4.2.2 Eignung zur Beschreibung von Regionen

Nachdem im letzten Absatz die Eigenschaften der Keypoints auf dem Testdatensatz untersucht wurden, gilt es nun deren Eignung zur Beschreibung von Regionen zu bewerten. Im Wesentlichen geht es darum zu klären, ob sich die Keypoints in wichtigen Regionen befinden, welcher Einfluss das Gewicht auf die Wahrscheinlichkeit, dass sich der Keypoint in einer wichtigen Region befindet, hat, und inwieweit die Größe des Keypoints nutzbar ist.

Größe	alle Oktaven		Oktave
	Anzahl	∅ Gewicht	
$0-10$	1.210	$\approx 159,72$	1
$11-20$	694.040	$\approx 650,91$	1; 2
$21-30$	303.037	$\approx 965,48$	1; 2
$31-40$	81.086	$\approx 1.444,78$	2; 3
$41-50$	47.098	$\approx 1.263,36$	2; 3
$51-60$	10.841	$\approx 1.503,86$	2; 3
$61-70$	9.763	$\approx 3.154,57$	3
$71-80$	3.430	$\approx 1.745,86$	3
$81-90$	5.995	$\approx 2.785,99$	3
$91-100$	1.251	$\approx 3.393,11$	3
$101-110$	22	$\approx 638,85$	3
Σ	1.157.733	$\approx 859,61$	-

Table 4: Verteilung der Keypoints auf dem Testdatensatz. Die Zuordnung zur Oktave ist nur approximiert. Auf Grund eines Fehlers in der OpenCV Implementierung ist eine genaue Zuordnung nicht möglich.

Position der Keypoints

Das wichtigste Kriterium, das einen Eindruck über die Eignung der SURF-Keypoints zur Beschreibung von Regionen gibt, ist deren Position. Sollten sich die Keypoints nicht in wichtigen Regionen befinden, ist es vermutlich aussichtslos ein Verfahren zu entwickeln, das mittels der SURF-Keypoints wichtige Regionen zuverlässig identifizieren oder bewerten kann.

Um die Zahl der Keypoints zu minimieren bzw. Keypoints auszuschließen, die wahrscheinlich nicht in wichtigen Regionen liegen, wird ein Grenzwert eingeführt, der Keypoints nicht berücksichtigt, deren Gewicht niedriger als der Grenzwert ist. Zudem wird so geprüft, ob ein Zusammenhang zwischen dem Gewicht des Keypoints und der tatsächlichen Relevanz besteht. In den Tests wurden folgende Grenzwerte verwendet: von 0 bis 100 in 10er Schritten, sowie 100 bis 1.000 (50er Schritte), 1.000 bis 2.500 (100er Schritte) und 2.500 bis 10.000 (500er Schritte). Die Aufteilung orientiert sich an der Verteilung der Keypoints, da die Zahl der Keypoints pro Bild auf diesem Testdatensatz mit der Erhöhung des Gewichts abnimmt (Tab. 4).

Die Tests ergeben, dass das Gewicht eines Keypoints und die Wahrscheinlichkeit, dass er sich in einer wichtigen Region befindet, po-

Anzahl der Bilder	∅ Breite	∅ Höhe	Σ Keypoints	∅ Keypoints	∅ Zeit [ms]
1.000	$\approx 372,515$	$\approx 325,5$	1.157.773	$\approx 1.157,77$	$\approx$ 57,977

Table 3: Informationen über den Testdatensatz: Größe und Keypoints

sitiv korreliert sind (Abb. 18). So liegt die *precision* bei einem Grenzwert von über 4.000 zwischen 50% und 60%. Im Gegensatz zu etwa 25% ohne Grenzwert, ist das eine deutliche Steigerung. Mit steigendem Grenzwert nimmt die Zahl der Keypoints allerdings exponentiell ab, so dass man ihn nicht zu hoch ansetzen sollte, da sonst die Zahl der Keypoints zu gering sein könnte, um Regionen zu beschreiben. Bei einem Grenzwert von 10.000 ist die *precision* zwar bei 60%, allerdings werden pro Bild durchschnittlich auch nur 18 Bildpunkte gefunden. Bei einem Grenzwert von 4.000 hat man immerhin noch 60 Keypoints pro Bild, aber nur etwa 10% weniger Genauigkeit im Vergleich zum Grenzwert 10.000. Je nachdem wie man die SURF-Keypoints nutzen möchte, kann es durchaus sinnvoll sein, die *precision* etwas zu vernachlässigen, um mehr Keypoints als Eingabeinformation zu behalten. Nun ist es nicht so, dass Keypoints mit niedrigem Gewicht generell nicht in wichtigen Regionen liegen. Um die *precision* von 25% auf 50% zu steigern, wird die Zahl der Keypoints pro Bild um 95% (von 1200 auf 60) gesenkt, was verdeutlicht, dass viele Keypoints mit relativ niedrigem Gewicht auch in wichtigen Regionen liegen.
Diese Ergebnisse zeigen zudem, dass es möglich ist, mittels Keypoints Regionen zu beschreiben. Geht man davon aus, dass die Regionen bereits korrekt identifiziert wurden und man nur noch entscheiden muss, welche wichtig sind, wäre es auf diesem Testdatensatz so, dass mittels eines hoch angesetzten Grenzwerts über 50% der Keypoints 20% der Bildfläche (Tab. 2) zuzuordnen sind. Mittels des einfachen Maßes der Keypointdichte, würde die wichtige Region dann ein 4-fach höheres Gewicht erhalten, als die nicht wichtige Region. Diese Interpretation gilt natürlich nur für diesen Datensatz und profitiert davon, dass die wichtige Region verhältnismäßig klein ist. Allerdings ist es auch so, dass bei größeren wichtigen Regionen die Wahrscheinlichkeit, dass Keypoints in dieser liegen, ebenso höher ist.

Keypoints in Randbereichen der wichtigen Region: Auf Grund der Art und Weise wie die SURF-Keypoints bestimmt werden, ist davon auszugehen, dass sie sich in Kantennähe befinden, da in diesen Bereichen Kontrastunterschiede zu erwarten sind. Daher gilt es zu untersuchen wie viele Keypoints knapp innerhalb sowie knapp außerhalb der wichtigen Region liegen. Die Ergebnisse dieser Tests geben Aufschluss darüber, ob es möglich ist Objekte auf Grund der Umrandung durch Keypoints zu segmentieren, in dem man diese verbindet.
Um zu bestimmen, wie groß der Anteil der Keypoints in Randbereichen der wichtigen Regionen ist, ist es nötig Referenzbilder zu erstellen, auf denen nur der äußere (innere) Rand gekennzeichnet ist. Dafür werden zuerst die wichtigen Regionen in dem Referenzbild vergrößert (verkleinert), indem die wichtige Region auf allen Seiten pro Iteration um einen Bildpunkt vergrößert (verkleinert) wird. Basierend auf dem so erstellten Bild und dem ursprünglichen Referenzbild wird das neue Referenzbild erstellt, auf dem Bildpunkte als wichtig markiert sind, die nur in einem der beiden Bilder gekennzeichnet sind. So ist die wichtige Region auf dem neuen Referenzbild dann nur noch der äußere (innere) Rand, dessen Breite abhängig von der Anzahl der im ersten Schritt getätigten Iterationen ist.

Eine Vergrößerung des Grenzbereichs hat zur Folge, dass sich immer mehr Keypoints in diesem befinden (Abb. 19 (a)). Betrachtet man nur einen fünf Bildpunkte breiten Bereich um die wichtige Region zusätzlich, erhöht sich die *precision* um 10%. Auffallend ist, dass der Anteil der Keypoints, die sich knapp außerhalb befinden, im Verhältnis zu allen Keypoints unabhängig des Grenzwertes nahezu konstant ist (Abb. 19 (b)). Zudem ist festzustellen, dass der Anteil der Keypoints in den Randbereichen über die Hälfte aller korrekt zugeordneten Keypoints beträgt. Da die Keypointdichte innerhalb der wichtigen Region deutlich höher ist als im Rest des Bildes, liegen mehr Keypoints knapp innerhalb als knapp außerhalb dieser (Abb. 18). Das wird auch dadurch nicht ausgeglichen, dass die Fläche, die als knapp außerhalb bezeichnet wird, größer ist als die Fläche, die als knapp innerhalb bezeichnet wird. Nur bis zu der Betrachtung eines beidseitig drei Bildpunkte breiten Grenzbereiches, ist die Zahl der Keypoints fast gleich (Abb. 19 (a)).
Festzuhalten bleibt also, dass sich ein nicht unbeträchtlicher Anteil der Keypoints knapp außerhalb der wichtigen Region befindet. Es scheint nicht möglich zu sein nur mittels Keypoints Objekte genau zu segmentieren, da hierfür erforderlich ist, dass in einem sehr schmalen Grenzbereich der wichtigen Region verhältnismäßig viele Keypoints gefunden werden.

Größe der Keypoints
Neben der Position und dem Gewicht ist die Größe der Keypoints eine weitere Eigenschaft, die man zur Beschreibung von Regionen nutzen kann. Konkret soll untersucht werden, ob der Bereich, der um den Keypoint liegt, in einer wichtigen Region liegt und ob es möglich ist, mittels der Keypoints große Flächen der wichtigen Regionen abzudecken. Somit ist neben der *precision* nun auch die Entwicklung des *recalls* ein Kriterium, das es zu untersuchen gilt. Die hier ermittelten Werte sind ein Benchmark für Verfahren, die die Informationen der Keypoints nutzen, um wichtige Regionen zu erkennen. Wie im letzten Abschnitt wird auch bei diesen Tests der Grenzwert schrittweise erhöht.
Der Hintergrund dieser Tests ist, dass die Keypoints sich nicht direkt auf einen einzelnen Punkt beziehen, sondern auf Grund ihrer Umgebung identifiziert wurden. Somit ist es naheliegend auch den Bildpunkten im Umkreis des Keypoints das gleiche Gewicht zu geben wie dem Keypoint selbst. Der Durchmesser der Umgebung hängt von der Filtergröße, die bei der Ermittlung des Keypoints verwendet wurde, ab. Zunächst wird nicht berücksichtigt, dass Bereiche, an denen sich Keypoints überlagern, ein höheres Gewicht haben könnten als das eines einzelnen Keypoints.

Unveränderte Größe
Bei dem ersten Test wird die Größe des Keypoints unverändert übernommen. Es ist festzustellen, dass diese Variante zur Bestimmung von wichtigen Regionen ohne Grenzwert etwas besser ist, als wenn man das gesamte Bild als wichtig markieren würde (Abschnitt 2). Es werden nur ca. 84,52% der Bildfläche als wichtig markiert und dennoch 98,77% der wichtigen Regionen gekennzeichnet. Das ist ein recht gutes Ergebnis, da man, im Gegensatz zum einfachsten Verfahren, die als wichtig markierte Region um 15% verkleinert und dabei der *recall* nur um etwas mehr als 1% zurück geht. Da die *precision* nur bei 22,93% liegt, ist das natürlich kein gutes Ergebnis, aber es zeigt doch, dass es möglich ist nur mit SURF-Keypoints wichtige Regionen zu markieren (Abb. 20).

Durch die Einführung eines Grenzwerts von 10.000 ist es mög-

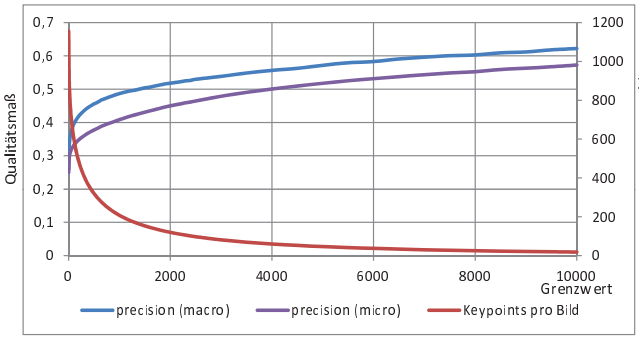
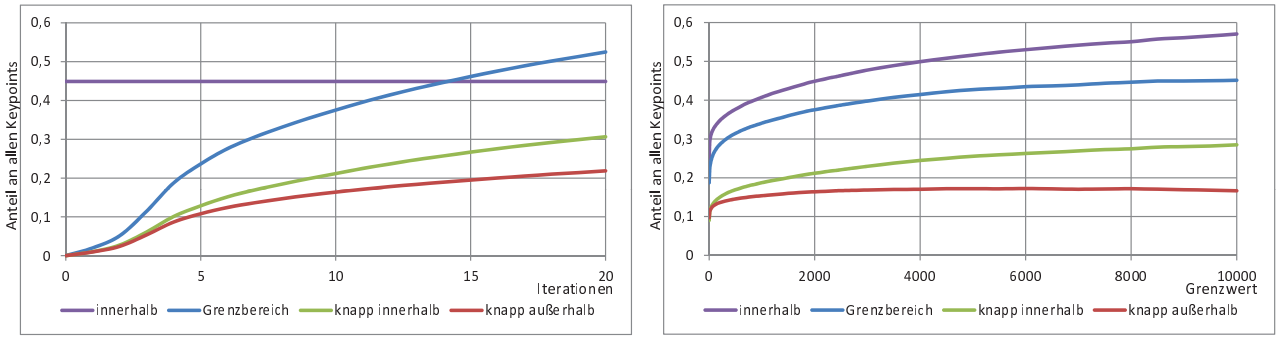


Figure 18: Einfluss eines Grenzwertes auf die Qualität der SURF-Keypoints (Position).



(a) Grenzwert der Keypoints ist 2.000

(b) Anzahl der Iterationen ist 10

Figure 19: SURF-Keypoints in Randbereichen. Diagramm (a) zeigt den Einfluss der Vergrößerung des Grenzbereichs bei gleichbleibenden Keypoints. Diagramm (b) verdeutlicht wie Keypoints mit höherem Gewicht positioniert sind. Der Rand in der Datenreihe Grenzbereich ist doppelt so breit wie die Anzahl der Iterationen, da der dargestellt Werte der Summe der knapp außerhalb sowie knapp innerhalb liegenden Keypoints im Verhältnis zu allen Keypoints entspricht.

lich die *precision* auf 53,56% zu steigern, was in etwa den Ergebnissen zu den Tests bezüglich der Position der Keypoints entspricht, wobei noch ein *recall* von 26,12% (im Gegensatz zu 0,0005%) erreicht wird. Am besten ist das Ergebnis insgesamt bei einem Grenzwert von 1.600, da bei diesem Wert der *F-measure* mit 49,05% maximal ist. Es werden 72,46% der wichtigen Regionen mit einer *precision* von 42,23% markiert.
Die Laufzeit dieser Variante ist ziemlich gut, da zu den ca. 58,50 Millisekunden, die zur Bestimmung der Keypoints nötig sind, nur durchschnittlich bis zu 3 Millisekunden hinzukommen, wenn die Zahl der Keypoints nicht durch einen Grenzwert minimiert wird. Setzt man den Grenzwert auf 1.600 dauert das Markieren sogar nur 0,74 Millisekunden.

Verkleinerung der Keypoints

Die vorhergehenden Tests haben gezeigt, dass der *F-measure* bei einem Grenzwert von über 1.500 am besten ist. In diesem Bereich sind nahezu alle Keypoints aus der ersten Oktave (Tab. 4) ausgeschlossen worden. Daher ist es interessant zu untersuchen, ob sich eine Verbesserung erzielen lässt, wenn man die Größe der Keypoints ab der zweiten Oktave halbiert. Wenn sich die Keypoints knapp innerhalb der wichtigen Region befinden, ist es vorstellbar, dass sich die *precision* erhöhen lässt, wenn man den Radius der Umgebung minimiert. Diese Vermutung ist darin begründet, dass Keypoints ab der zweiten Oktave einen Durchmesser von über 30 Pixeln haben, was bei diesem Testdatensatz etwa 10% der Höhe und Breite entspricht. Somit bedeckt ein Keypoint ab der zweiten

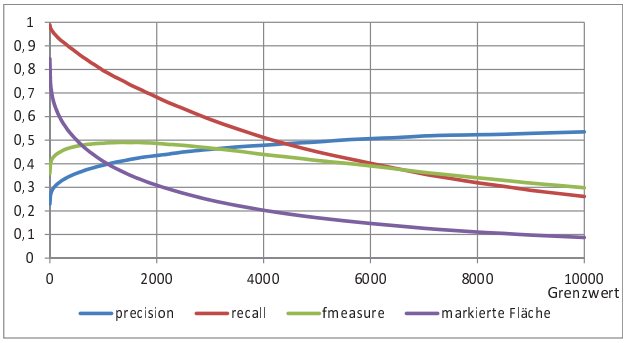


Figure 20: Einfluss eines Grenzwertes auf die Qualität der SURF-Keypoints (Position, Größe).

Oktave mindestens 1% der Bildfläche, was deutet, dass etwas mehr als 20 Keypoints ausreichen würden, um wichtige Regionen vollständig zu markieren, da auf diesem Datensatz im Schnitt 20% der Bildfläche wichtig ist (Tab. 2). Da sich die Keypoints auch gegenseitig überlagern können, wenn in einem kleinen Bildbereich hohe Kontrastunterschiede vorliegen, ist das natürlich nur eine theoretische Annahme.
Die Tests ergeben, dass sich die *precision* so um durchschnittlich 5% erhöht. Allerdings verliert man bis zu 20% (durchnittlich 13,8%) *recall*, was sicher nicht akzeptabel ist, wenn man nur mittels der

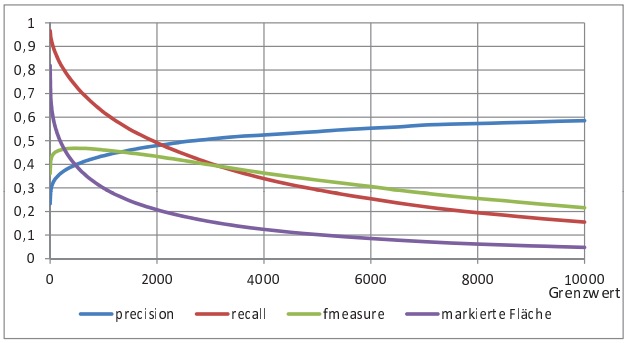


Figure 21: Einfluss eines Grenzwertes auf die Qualität der SURF-Keypoints (Position, verkleinerter Durchmesser).

SURF-Keypoints Regionen identifizieren und beschreiben möchte (Abb. 21). Somit erzielt der Ansatz die Keypoints zu verkleinern nicht den erhofften Effekt. Zudem liegt das Maximum des *F-measures* bei einem niedrigeren Grenzwert, was nicht wünschenswert ist, da bei einem niedrigeren Grenzwert die Zahl der Keypoints, die man auf Regionen übertragen muss, höher ist, wodurch die Laufzeit anderer Verfahren, die SURF-Keypoints nutzen, negativ beeinflusst wird.

Eine Erhöhung der Größe der Keypoints macht keinen Sinn, da die kleinen Keypoints ein geringes Gewicht haben und durch einen Grenzwert sowieso ausgeschlossen werden. Die Keypoints mit hohem Gewicht, die also nicht durch einen Grenzwert ausgeschlossen werden, haben einen großen Durchmesser und werden daher durch eine Vergrößerung auch mehr Bereiche außerhalb der wichtigen Region markieren. Dies liegt daran, dass sich diese Keypoints zum Teil auch an den Rändern der Regionen befinden, wodurch unwichtige und wichtige Regionen gleichermaßen markiert werden. Dies bestätigt die *precision* von knapp über 50% bei hohem Grenzwert (Abb. 18, 20), was im Prinzip bedeutet, dass ein Hälfte der Keypoints innerhalb und die andere Hälfte außerhalb der wichtigen Region liegt.

Überlagernde Keypoints

Bei den bisherigen Tests wurde komplett vernachlässigt, dass Bereiche, in denen sich die Umfelder verschiedener Keypoints überlagern, ein höheres Gewicht haben könnten als das des einzelnen Keypoints. Wird ein bestimmter Bereich von mehreren Keypoints markiert, könnte man davon ausgehen, dass er zu einer wichtigen Region gehört, da eine hohe Keypointdichte vorliegen muss. Sollte das Gewicht der einzelnen Keypoints, die den Bereich markieren, unter dem Grenzwert liegen, wird der Bereich bisher nicht berücksichtigt. Um diese Situation abzufangen, werden nun die Gewichte der sich überschneidenden Keypoints in den betreffenden Bereichen addiert. Liegt die Summe dann über dem Grenzwert, wird der Bereich auch berücksichtigt, wenn kein einzelner Keypoint, der den Bereich markiert hat, über dem Grenzwert liegt. Somit ist auf jeden Fall ein höherer *recall* zu erreichen, da bei der Erhöhung des Grenzwertes weniger Fläche ausgeschlossen wird. Man könnte vermuten, dass die *precision* zumindest unverändert bleibt, da die Überlagerungen nur durch eine hohe Keypointdichte entstehen können, und somit als Zeichen für einen wichtigen Bereich bewertetet werden könnten.

Die Tests belegen die Vermutung nur teilweise. Eine Erhöhung des *recall* ist ohne Zweifel festzustellen, allerdings ist diese unmittelbar mit einem Verlust von *precision* verbunden, so dass der *F-measure* insgesamt unwesentlich erhöht wird. Das Maximum des *F-measures* steigt um weniger als 1% an, und verschiebt sich nach rechts, was bedeutet, dass fehlerhaft als wichtig markierte Regionen erst bei einem höheren Grenzwert ausgeschlossen werden (Abb.22). Insgesamt ist also keine wesentliche Verbesserung festzustellen, so dass dieser Ansatz angesichts der hohen Kosten nicht besonders sinnvoll ist. Ein Keypoint mit einem Durchmesser von 50 Pixeln, markiert fast 2.000 Pixel, wodurch im Vergleich zur Methode ohne Berücksichtigung von Überlagerungen, allein für einen einzelnen Keypoint mittlerer Größe etwa 2.000 zusätzliche Additionen nötig sind.

Dies wirkt sich äußerst negativ auf die Laufzeit aus. Das Markieren der wichtigen Regionen dauert nun durchschnittlich über 560 Millisekunden pro Bild. Im Vergleich zur Variante ohne Berücksichtigung von Überlagerungen, dauert es also über 160-mal (3 ms zu 500 ms) so lange die wichtigen Bereiche zu markieren. Insgesamt ist das Verfahren fast um den Faktor 10 langsamer (60 ms zu 560 ms). Zudem lässt sich diese Variante auch nicht wesentlich beschleunigen, indem man zum Beispiel einen zweiten Grenzwert einführen würde, der die Zahl der Keypoints, die berücksichtigt werden, reduziert. Dies würde der Idee des Ansatzes entgegenlaufen, Bereiche zu markieren, die von mehreren Keypoints mit niedrigem Gewicht überlagert werden.

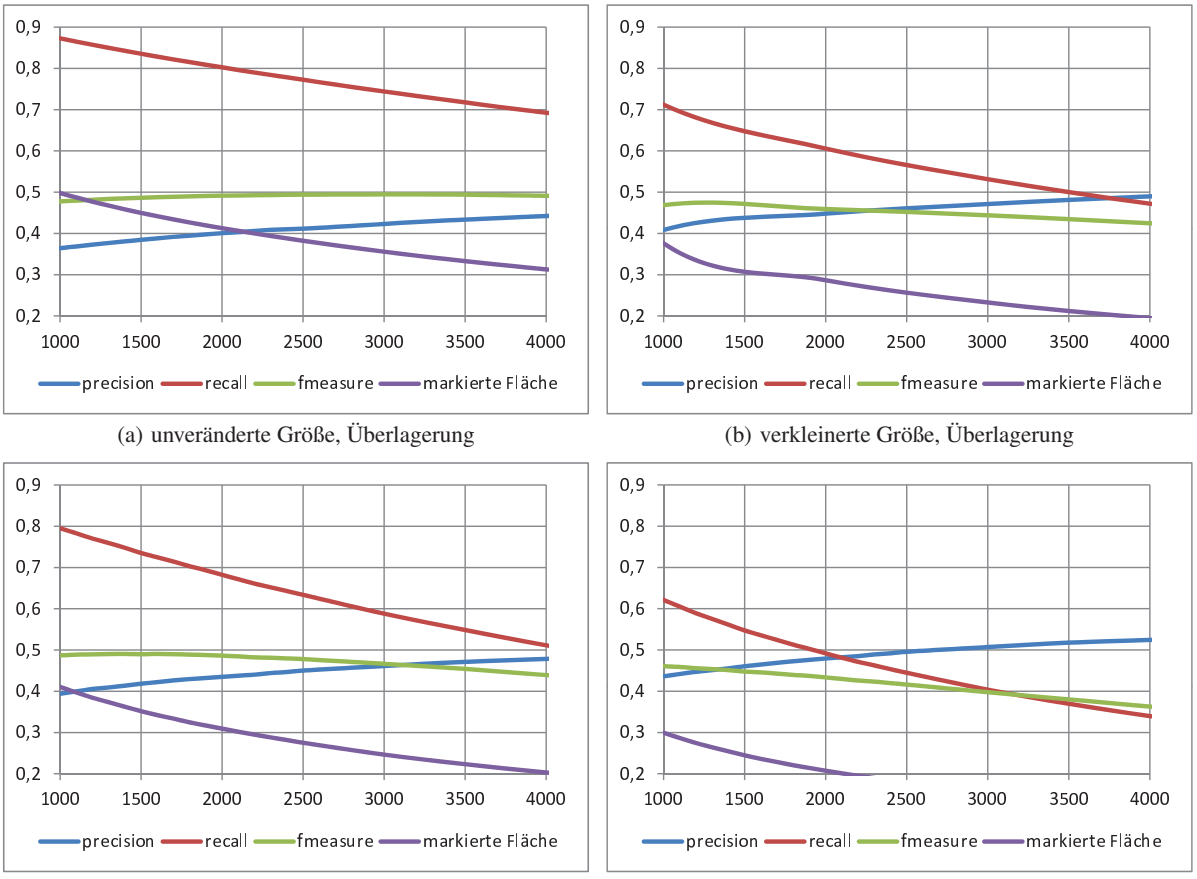
Fazit

Insgesamt ist festzustellen, dass es möglich ist mit Keypoints wichtige Regionen zu erkennen und zu beschreiben. Die Ergebnisse sind in der Regel dann gut, wenn die wichtigen Regionen im Gegensatz zum Hintergrund bzw. der unwichtigen Region sehr kontrastreich sind. Ist dies nicht der Fall oder ist die Kontrastverteilung im gesamten Bild ziemlich gleichmäßig, dann ist es nicht möglich mittels SURF-Keypoints gute Ergebnisse zu erzielen (Abb. 23), da sich die Keypoints dann ziemlich gleichmäßig über das Bild verteilen und auch ein ähnliches Gewicht haben. Da der relevante Bereich in der Regel kleiner ist als der unwichtige Bereich, ist es dann nicht möglich mittels der SURF-Keypoints dieser Region ein höheres Gewicht zuzuordnen - sowohl in der Summe wie auch in Bezug auf die Größe der Region.

Es wurde mit keinem der getesteten Ansätze eine *precision* von deutlich über 50% erreicht. Dies ist darauf zurückzuführen, da sich die Keypoints auf beiden Seiten der Ränder wichtiger Regionen befinden. Angesicht der Tatsache, dass die wichtigen Regionen nur etwa 20% der Bildfläche einnehmen sollte es aber dennoch möglich sein wichtige Regionen als solche zu erkennen. Die Tests haben zudem gezeigt, dass es nicht sinnvoll ist die Größe der Keypoints zu verkleinern oder überlagernde Bereiche zu berücksichtigen. Somit bleiben als akzeptable Ansätze nur die Position der Keypoints sowie die Berücksichtigung der unveränderten Größe der Keypoints. Die besten Werte wurden hier bei letztgenannter Variante mit einem Grenzwert von 1.600 erzielt. Dabei wurden durchschnittlich folgende Werte erreicht: 42,23% *precision*, 72,48% *recall* sowie 49,05% *F-measure* (Tab. 5). Für die Bestimmung des Gewichts einer Region ist es somit naheliegend Verfahren zu verwenden, die auf der Keypointdichte innerhalb einer Region basieren.

4.3 Evaluation der Regionen

In diesem Abschnitt wird untersucht wie gut die beschriebenen Verfahren Regionen bestimmen. Da den gefundenen Regionen später ein Gewicht zugeordnet werden soll, ist ein wichtiges Kriterium wie genau die Regionen erkannt werden. Eine Region ist auf jeden Fall dann schlecht bestimmt, wenn sie Bereiche in einer wichtigen wie auch unwichtigen Region umfasst. Dies ist naheliegend, da man dem Bereich innerhalb der Region ein hohes Gewicht und



(a) unveränderte Größe, Überlagerung

(b) verkleinerte Größe, Überlagerung

(c) unveränderte Größe, keine Überlagerung

(d) verkleinerte Größe, keine Überlagerung

Figure 22: Übersicht der getesteten Varianten. Ergebnisse der Methode mit überlagernden Keypoints (oben) und ohne Berücksichtigung von Überlagerung (unten) mit unveränderter Größe (links) sowie verkleinerten Keypoints (rechts).

dem Bereich außerhalb der Region ein niedriges Gewicht zuordnen sollte. Daher wird für jede gefundene Bildregion bestimmt, wie viele Bildpunkte in der wichtigen Region liegen und wie viele Bildpunkte im Hintergrund zu finden sind. Um nicht zu viele Regionen als falsch bestimmt zu kennzeichnen, ist es ausreichend, wenn 95% der Bildpunkte einer Region in einer der beiden Regionen im Referenzbild zu finden sind. So werden zudem große Bildregionen gegenüber kleinen Regionen nicht zu sehr benachteiligt, wenn sie nur einen minimalen Bereich in einer anderen Region abdecken.

Es werden zunächst die beiden Verfahren optimiert, ohne die Nachbearbeitungsalgorithmen auszuführen. Dies liegt daran, dass sie das hier verwendete Qualitätsmaß der Regionenermittlung nicht verbessern können, da sie Regionen nur zusammenfassen. Erst in einem zweiten Schritt wird der Einfluss der Nachbearbeitungsalgorithmen auf die Qualität der Regionen untersucht.

4.3.1 *Regionen mit ähnlicher Helligkeit*

Die Zusammenfassung von Regionen mit ähnlicher Helligkeit hat als einzigen Parameter die erlaubte Helligkeitsdifferenz zwischen dem Startpunkt der Region und jedem weiteren Bildpunkt der zur Region hinzugefügt werden kann. Der Wertebereich dieses Parameters bewegt sich im Bereich zwischen Null und 255. Mit einer Erhöhung des Parameters werden mehr Bildpunkte zusammengefasst und die Zahl der Regionen im Bild reduziert. Setzt man den Wert auf Null, entspricht die Zahl der Regionen fast der Anzahl der Bildpunkte und es werden sehr viele Regionen als sehr gut zugeordnet bewertet. Setzt man den Wert auf 255, wird das gesamte Bild zu einer Region zusammengefasst (Abb. 24).

Die Tests zeigen, dass die Qualität der ermittelten Regionen mit der Erhöhung der Helligkeitsdifferenz wie auch die Laufzeit stetig abnimmt, während sich die durchschnittliche Größe einer Region erhöht (Tab. 6). Wählt man einen niedrigen Wert, wird ein Bild in sehr viele kleine Regionen eingeteilt, die eine spätere Bewertung der Regionen verkomplizieren, da wichtige Regionen aus vielen kleinen Bereichen bestehen, die alle ein ähnliches Gewicht erhalten sollten. Verwendet man einen hohen Wert, befinden sich große Bereiche in wichtigen und auch unwichtigen Regionen, so dass eine gute Bewertung nicht möglich ist. Somit liegt ein sinnvoller Wert im Bereich zwischen zehn und 20, da die Qualität der Regionen, mit 90% korrekt zugeordneter Bildfläche, wie auch die durchschnittliche Größe der Regionen noch akzeptabel sind. Einzelne Regionen können in weiteren Schritten durch Nachbearbeitungsalgortihmen zusammengefasst werden, so dass sich die durchschnittliche Größe der Regionen erhöhen lässt. Da dies auf Kosten der Genauigkeit geschieht, ist es besser die Regionenermittlung möglichst genau zu konfigurieren.

4.3.2 *Watershed-Regionen*

Der *watershed*-Algorithmus hat als variable Parameter einzig die Startpunkte, die zur Bestimmung der Regionen genutzt werden.

(a) precision = 0,90; recall = 0,94; F-measure = 0,92

(b) precision = 0,05; recall = 0,01; F-measure = 0,02

Figure 23: Die Abbildung zeigt beispielhaft bei welchen Bildern man mittels der Größe der SURF-Keypoints und einem Grenzwert von 1.600 Regionen gut bzw. schlecht kennzeichnen kann. Das linke Bild ist das Originalbild, das mittlere Bild ist das Referenzbild und das rechte Bild zeigt die mittels der SURF-Keypoints markierten Regionen.

Verfahren	**Grenzwert**	***precision***	***recall***	***F-measure***	**Laufzeit**
Keypoint-Position	10.000	62,2%	0,0005%	0,001%	58 ms
Größe					
unveränderte Größe	1.600	42,23%	72,48%	49,05%	59 ms
verkleinerte Größe	450	39,47%	74,14%	46,8%	61 ms
Überlagerung					
unveränderte Größe	3.100	42,55%	73,86%	49,46%	560 ms
verkleinerte Größe	1.000	40,88%	71,2%	46,85%	560 ms

Table 5: Überblick der Ergebnisse der direkten Tests des SURF-Verfahrens.

HD	**∅ falscher Anteil**	**∅ Größe einer Region**	**∅ Laufzeit**
5	1,43%	8,68 px	3.276 ms
10	4,00%	19,21 px	1.494 ms
15	7,90%	34,12 px	907 ms
20	12,39%	54,2 px	607 ms
30	21,62%	112,9 px	374 ms
40	30,33%	202,41 px	254 ms
50	37,94%	321,94 px	187 ms

Table 6: Die Tabelle verdeutlicht den Einfluss der erlaubten Helligkeitsdifferenz (HD) auf die Qualität der Regionen. Der falsche Anteil bezeichnet den Anteil an der Bildfläche, der durch Regionen abgedeckt wird, die in wichtigen und unwichtigen Regionen liegen.

Daher wird hier untersucht mit welcher Einstellung die Startmarker bestimmt werden sollten, damit der *watershed*-Algorithmus die Regionen präzise erkennen kann. Die Parameter, die hier untersucht werden, setzen an dem Punkt an, an dem bereits ein Kantenbild des Originalbildes vorliegt. Die Startpunkte befinden sich in Bildbereichen, die von Kanten umschlossen werden. Dafür werden die gefundenen Kanten vergrößert, um die Startbereiche einzuschränken, was im Wesentlichen von drei Parametern abhängt. Es wird zunächst ein adaptiver Grenzwert angewendet, bei dem man die *Größe der Umgebung* sowie die *Reduktion des Mittelwerts* ändern kann. Bei dem so erstellten Bild müssen die Kantenregionen vergrößert werden, wobei sich die Ausprägung über die *Anzahl der Iterationen des Erodierens* steuern lässt.

Mit einem vierten Parameter kann man zudem noch die minimale Größe einer Startregion festlegen. Um die Kombinationsmöglichkeiten einzuschränken und auf Grund der Tatsache, dass das Qualitätsmaß kleine Regionen bevorzugt, wird dieser Wert auf fünf Bildpunkte festgelegt. So liegt die minimale Größe vor der Vergrößerung nur knapp unter dem Niveau des Verfahrens, das Regionen mit ähnlicher Helligkeit zusammenfasst. Mit einer Erhöhung des Wertes wird die Qualität der Regionen unter Anwendung des verwendeten Qualitätsmaßes auf jeden Fall negativ beeinflusst.

Tendenziell werden die Ergebnisse besser, wenn man die Parameter so konfiguriert, dass die Zahl der Regionen steigt. Zudem scheint es sinnvoll zu sein, wenn man die Kantenbereiche kaum vergrößert und bei dem adaptiven Grenzwert nur einen kleinen Bereich um den jeweiligen Bildpunkt betrachtet. Dies hat zur Fol-

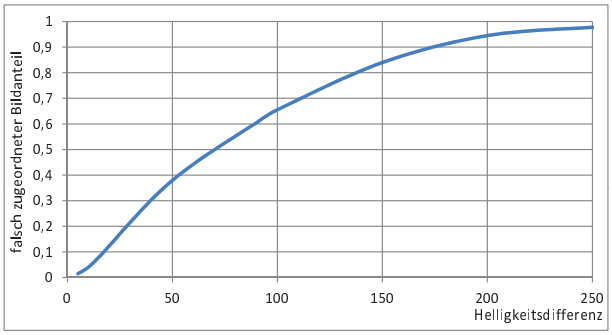


Figure 24: Das Diagramm zeigt, dass die Qualität der Regionen mit einer Erhöhung der akzeptablen Helligkeitsdifferenz abnimmt, was daran liegt, dass größere Bereiche zu einer Region zusammengefasst werden.

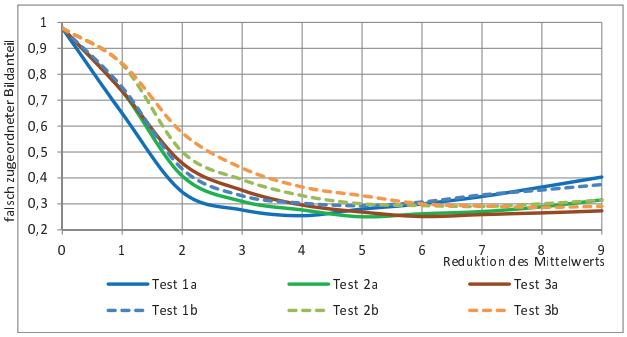


Figure 25: Das Diagramm zeigt den Einfluss des Parameters "Reduktion des Mittelwerts" während die restlichen Parameter in den einzelnen Tests unverändert bleiben (Tab. 7). Bis zu einem Wert von 255 würden alle Testreihen wieder das komplette Bild falsch kennzeichnen.

ge, dass das Verfahren, das zur Bestimmung der Kanten verwendet wird, einen großen Einfluss auf das Ergebnis hat, da die so gesetzten Konturen kaum verändert werden.

Die Tests (Abb. 25, Tab. 8) zeigen, dass die Qualität mit der Vergrößerung der Kantenbereiche relativ deutlich sinkt, da viele kleine Regionen geschlossen werden, die dann nicht mehr als Startregionen genutzt werden können. Desweiteren ist auffällig, dass man mit einer Vergrößerung der zu betrachtenden Umgebung bei der Berechnung des adaptiven Grenzwertes ebenso den Wert, um den der Mittelwert reduziert wird, erhöhen sollte. Dies liegt daran, dass man dem steigenden Einfluss der Kanten, die den Mittelwert in die Höhe treiben, entgegenwirken muss. Mit einer Vergrößerung der Umgebung reduziert man zudem die Anzahl der Startregionen, was sich negativ auf die Qualität auswirkt. Auf dem Testdatensatz werden die Regionen am präzisesten ermittelt, wenn das Verfahren so wie in Test 2a (Tab. 8) konfiguriert wird.

4.3.3 *Einfluss der Nachbearbeitungsverfahren*

Nach der Untersuchung der Auswirkungen von Parameteränderungen bei der Bestimmung der Regionen wird nun geprüft wie sich die Ergebnisse verändern, wenn man die Nachbearbeitungsverfahren auf die identifizierten Regionen anwendet. Dabei ist davon auszugehen, dass die durchschnittliche Größe einer Region steigt, da Regionen verbunden werden. Desweiteren kann die Qualität der Regionen sinken, da Zusammenfassungen von Regionen möglich sind, die in verschiedenen Bereichen im Referenzbild liegen. Für die Tests werden die zuvor bestimmten Konfigurationen verwendet, also eine erlaubte Helligkeitsdifferenz von 15, sowie die Einstellungen von Test 2a (Tab. 8) beim *watershed*-Verfahren.

Im ersten Schritt werden Hintergrundregionen erkannt, wobei die Bestimmung von Hintergrundregionen innerhalb des Bildes optional ist. Für die Entscheidung, ob eine Region innerhalb des Bildes (optionaler Teil) zum Hintergrund gehört, wird für jede Hintergrundregion der Mittelwert im Graustufenbild berechnet. Diese Werte werden mit den Mittelwerten der anderen Regionen verglichen und bei einer Helligkeitsdifferenz von höchstens fünf werden andere Regionen ebenso zum Hintergrund hinzugefügt. Bei beiden Verfahren hat die Hintergrunderkennung keinen großen Einfluss, das heißt das Qualitätsmaß verschlechtert sich nicht wesentlich, wenn man Regionen innerhalb des Bildes nicht als Hintergrund markiert. Je kleiner die Regionen davor sind, desto geringer ist der Einfluss der Hintergrunderkennung, da dann nur sehr wenige Regionen mit zwei Bildrändern verbunden sein können. Sobald man auch Regionen innerhalb des Bildes als Hintegrund markiert, geht die Qualität sowie Zahl der Regionen deutlich zurück, so dass die Anwendung dieser Zusatzfunktion nicht empfehlenswert ist (Tab. 9).

Nachdem die Hintergrundregionen erkannt sind, können im zweiten Schritt weitere Regionen zusammengefügt werden. Der erforderliche *prozentuale Anteil am Rand*, der festlegt wann Regionen zusammengefasst werden, wird variiert (50%, 75% und 100%), während die *minimale Länge des Randes* auf 40 festgesetzt wird. Dieser Schritt hat einen großen Einfluss auf die helligkeitsbasierte Regionenerkennung. Es werden viele Regionen zusammengefasst, so dass die durchschnittliche Größe einer Region um mehr als 10-fach steigt. Dabei sinkt die Genauigkeit der Regionen nur um ca. 3%. Dies kann vorteilhaft sein, da die wichtigen Regionen nun nicht mehr aus vielen kleinen Regionen bestehen, was die Bewertung verbessern sollte. Da größere Regionen gegenüber einer minimal geringeren Präzision der Regionen zu bevorzugen sind, sollte man eine Region zur äußeren Region hinzufügen, wenn diese an 75% des Randes angrenzt. Der einzige Nachteil ist, dass sich die Gesamtlaufzeit fast vervierfacht. Beim *watershed*-Verfahren ist der Einfluss nicht besonders groß, da wenige Regionen von anderen Regionen eingschlossen werden. Bei 75% ist das Ergebnis am besten, obwohl mehr Regionen als bei 100% verbunden werden. Das liegt daran, dass das Qualitätsmaß eine Region bereits als korrekt wertet, wenn mehr als 95% in der wichtigen oder unwichtigen Region liegen (Tab. 10).

4.3.4 *Vergleich der Qualität der Verfahren*

Ein sinnvoller Vergleich der beiden Verfahren ist schwer möglich, da ihre Ergebnisse völlig unterschiedlich sind. Zudem ist die Aussagekraft des Qualitätsmaßes zur Bewertung der Genauigkeit der Regionen ziemlich fragwürdig, da kleine Regionen klar bevorteilt werden, so dass beide Verfahren dann den besten Wert erzielen, wenn die Zahl der Regionen pro Bild maximiert wird. Die Ermittlung von Regionen basierend auf ähnlicher Helligkeit ist sehr präzise, wenn man einen niedrigen Grenzwert setzt. Dies hat allerdings zur Folge, dass ein Bild in sehr viele kleine Regionen eingeteilt wird, wodurch das direkte Segmentieren eines Objektes unmöglich ist. Das *watershed*-Verfahren findet deutlich weniger und somit größere Regionen, wodurch es als deutlich unpräziser bewertet wird. Dafür ist es mit einer Laufzeit von knapp unter 40 Millisekunden im Gegensatz zu etwa 900 Millisekunden deutlich schneller. Die einzige Gemeinsamkeit haben die beiden Verfahren bei der optimalen Konfiguration der Nachbearbeitungsalgorithmen. Beide nutzen die einfache Version der Hintergrunderkennung und fügen

	Größe der Umgebung	Vergrößerung	Mindestgröße
Test 1a	3	2	5
Test 1b	3	3	5
Test 2a	5	2	5
Test 2b	5	3	5
Test 3a	7	2	5
Test 3b	7	3	5

Table 7: Die Tabelle zeigt die verwendeten Konfigurationen der verschiedenen Tests, die in Abb. 25 dargestellt sind. Mit der Testnummer wird die Größe der Umgebung erhöht. Zudem wird die Vergrößerung der Kantenbereiche variiert (a=2,b=3).

	RM	∅ falscher Anteil	∅ Größe einer Region	∅ Laufzeit
Test 1a	4	25,44%	3.851,94 px	34,58 ms
Test 1b	5	29,27%	5.016,51 px	28,21 ms
Test 2a	5	25,00%	4.016,21 px	37,32 ms
Test 2b	7	29,00%	5.195,18 px	27,56 ms
Test 3a	6	25,12%	4.246,79 px	43,33 ms
Test 3b	8	28,64%	5.413,93 px	38,14 ms

Table 8: Die Tabelle fasst die besten Ergebnisse der verschieden Tests zusammen. RM beschreibt die Reduktion des Mittelwerts. Die restlichen Paramater sind in Tabelle 7 gelistet.

eine Region zu einer anderen Region hinzu, wenn sie um mehr als 75% von dieser eingeschlossen wird.

4.4 Evaluation der Saliency Maps

Ausgehend von den in Abschnitt 4.3 ermittelten Konfigurationen der Verfahren, können nun die einzelnen Regionen bewertet werden. Hierbei stehen die SURF-Keypoints wieder im Mittelpunkt, da sie dafür verwendet werden, um den Regionen ein Gewicht zuzuweisen. Zudem ist es hier auch interessant den Einfluss der Nachbearbeitungsverfahren zur Verbesserung der Regionen zu prüfen. Für die Regionenbestimmung wird also jedes der beiden Verfahren ohne und mit den verschiedenen Stufen der Nachbearbeitung der Regionen getestet, so dass es ingesamt sechs relevante Testreihen gibt:

- **Test 4:** Regionen mit ähnlicher Helligkeit
  - a: erlaubte Helligkeitsdifferenz: 15
  - b: a + einfache Version der Hintegrunderkennung
  - c: b + Zusammenfügen der Region (erforderlicher Anteil: 75%, Mindestlänge: 40 px)
- **Test 5:** *watershed*-Verfahren
  - a: siehe Test 2a (Tab. 8))
  - b: a + einfache Version der Hintegrunderkennung
  - c: b + Zusammenfügen der Region (erforderlicher Anteil: 75%, Mindestlänge: 40 px)

Für die Bewertung der Regionen wird die Variante gewählt bei der die Bildpunkte im Umkreis der Keypoints markiert werden. Das Gewicht einer Region entspricht dann dem Verhältnis der markierten Bildpunkte zu der Größe der Region. Diese Variante wird gewählt, da das Gewicht der Keypoints indirekt über die Größe der markierten Umgebung berücksichtigt wird und zudem Keypoints, die sich knapp außerhalb der wichtigen Region befinden, diese noch anteilig kennzeichnen können.

4.4.1 Ermittlung des SURF-Grenzwertes

Zunächst muss für jede der sechs Testvarianten der optimale Grenzwert der SURF-Keypoints ermittelt werden. Keypoints, die den SURF-Grenzwert unterschreiten, dürfen keine Bildpunkte markieren. Um die erstellten Saliency Maps, auf denen jeder Bildpunkt mit einem Wert zwischen Null und Eins markiert ist, in ein Binärbild zu überführen, werden mehrere Grenzwerte getestet und dann für jedes Bild der Optimale gewählt. Diese Vorgehensweise dient nur dazu die theoretisch beste Konfiguration zu ermitteln und kann etwas bessere Werte liefern als ein konstanter Grenzwert.

Es ist bei allen sechs Testreihen festzustellen, dass die *precision* bis zu einem SURF-Grenzwert von knapp über 1.000 deutlich steigt und danach das hohe Niveau weiter hält (Abb. 26 (a)). Da der *recall* demgegenüber mit der Erhöhung des SURF-Grenzwertes zurückgeht, sind die besten Ergebnisse mit einem Grenzwert von etwa 1.500 optimal (Abb. 26 (b)). Diese Ergebnisse entsprechen somit tendenziell dem direkten Markieren wichtiger Regionen mittels SURF-Keypoints (Abb. 20), da hier der optimale Grenzwert ebenfalls bei etwa 1.500 liegt. Interessant ist, dass die Tests, die auf dem *watershed*-Verfahren basieren, eine höhere *precision* haben als die Tests, die die helligkeitsbasierten Regionenerkennung verwenden, da dies den Ergebnissen der Evaluation der Regionen in Abschnitt 4.3 widerspricht. Demgegenüber liefern die helligkeitsbasierten Regionenermittlungsverfahren einen höheren *recall*, was daran liegt, dass die Regionen so klein sind, dass die von den Keypoints markierten Bereiche nur minimal verändert übernommen werden. Dass der Grenzwert, der für Saliency Maps verwendet wird (Abb. 26 (d)), mit der Erhöhung des SURF-Grenzwertes zurückgeht, liegt daran, dass die SURF-Keypoints immer weniger Fläche markieren und so die Keypointdichte in den Regionen sinkt. Die Testreihen *Test 4a* und *Test 4b* liefern nahezu identische Ergebnisse, da die Regionen so klein sind, dass kaum Hintergrundregionen gefunden werden. Ähnliches gilt für *Test 5b* und *Test 5c*. Bei diesen Tests schließen die ermittelten Regionen kaum andere Regionen ein; daher werden kaum Regionen zusammengefügt. Dies hat zur Folge, dass die markierten Regionen bei beiden Tests fast identisch sind. Aus diesen Gründen werden die Testreihen *Test 4b* und *Test 5c* nicht weiter untersucht.

4.4.2 Ermittlung der Ergebnisse

Nachdem nun die endgültigen Konfigurationen feststehen (Tab. 26), wird der Einfluss des Saliency-Map-Grenzwertes der vier verbleibenden Testreihen untersucht. Es ist festzustellen, dass bei einem konstanten Grenzwert für alle Bilder die Ergebnisse deutlich

	∅ falscher Anteil	∅ Größe einer Region	∅ Laufzeit
Regionen mit ähnlicher Helligkeit			
original	7,89%	34,12 px	907 ms
standard	8,81%	34,17 px	913 ms
erweitert	26,47%	46,35 px	1.900 ms
***watershed*-Verfahren**			
original	25,00%	4.016,21 px	33,1 ms
standard	26,77%	4.621,13 px	38,7 ms
erweitert	38,01%	6.973,97 px	44,0 ms

Table 9: Ergebnisse der Hintergrunderkennung. Die Originalwerte dienen als Referenzwerte. Bei der Standardversion der Hintergrunderkennung werden im Gegensatz zur erweiterten Version keine Regionen innerhalb des Bildes als Hintergrund gekennzeichnet.

	∅ falscher Anteil	∅ Größe einer Region	∅ Laufzeit
Regionen mit ähnlicher Helligkeit			
Hge.	8,81%	34,17 px	913 ms
50%	12,10%	637,10px	3.593 ms
75%	11,29%	496,21 px	3.162 ms
100%	11,10%	456,38 px	3.578 ms
***watershed*-Verfahren**			
Hge.	26,77%	4.621,13 px	38,7 ms
50%	28,50%	5.764,13 px	42,3 ms
75%	26,85%	4.960,00 px	41,2 ms
100%	26,79%	4.864,55 px	43,3 ms

Table 10: Ergebnisse des Regionenzusammenfügens. Ausgehend von den besten Ergebnissen der Hintergrunderkennung (Hge.) werden Regionen zusammengefügt, wobei der nötige Randanteil variiert wird.

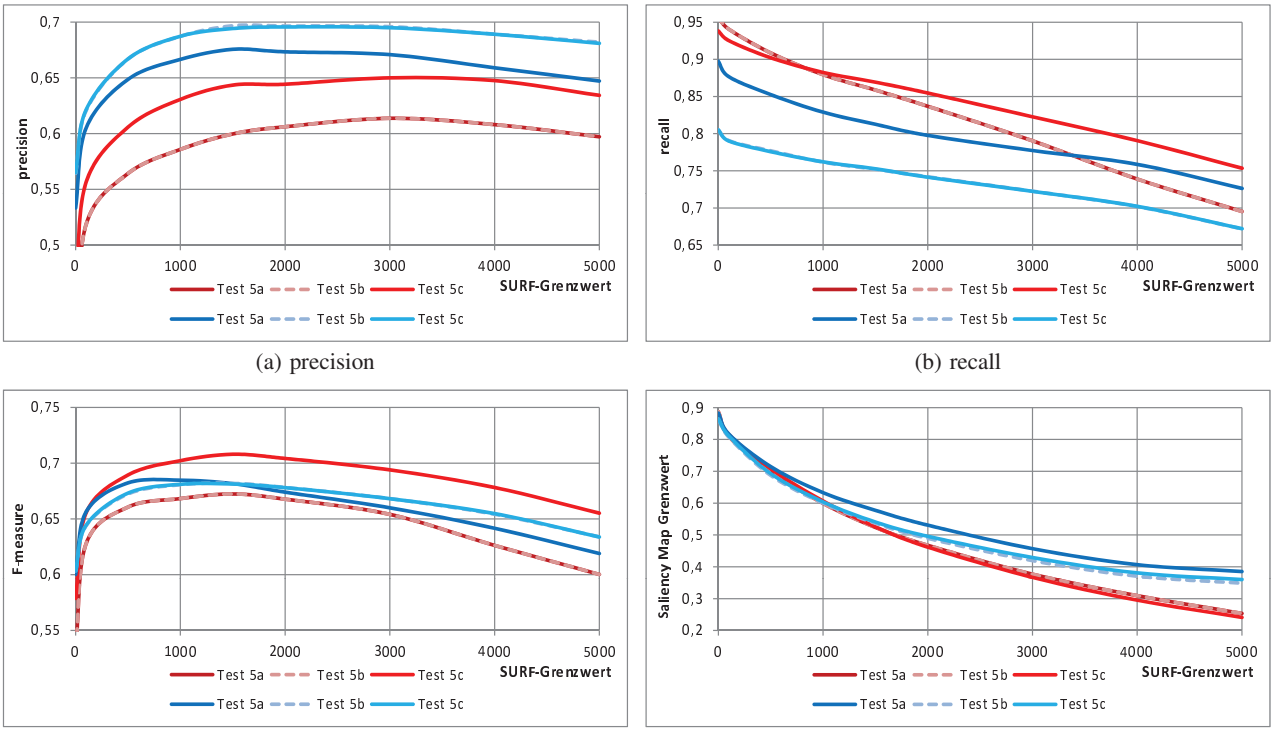
schlechter sind (Abb. 27, Tab. 12). Die konstanten Grenzwerte sind alle niedriger als der Durchschnitt der optimalen Grenzwerte. Daher ist es auf den ersten Blick überraschend, dass der *recall* so deutlich zurückgegangen ist. Dies ist darauf zurückzuführen, dass der dynamische Grenzwert für jedes Bild optimiert wurde und auch häufig deutlich unter dem neuen Grenzwert lag, so dass der *recall* bei allen Bilder auf einem hohen Niveau blieb. Durch den konstanten Grenzwert gibt es einige Bilder, bei denen der *recall* ziemlich schlecht ist, da der Saliency-Map-Grenzwert für diese Bilder zu hoch angesetzt ist. Der Rückgang der *precision* ist im Prinzip auf den gleichen Grund zurückzuführen, wobei hier der Grenzwert bei einigen Bildern höher sein könnte als der einheitliche Wert. Die Erkenntnis, dass das *watershed*-Verfahren genauer ist, während die helligkeitsbasierte Regionenerkennung größere Anteile der wichtigen Regionen abdeckt, gilt weiterhin. Die Nachbearbeitungsalgorithmen verbessern beide Verfahren, indem sie vor allem den *recall* erhöhen und dabei kaum *precision* verloren geht. Insgesamt bewegen sich die Ergebnisse aller getesteten Verfahren aber fast auf einem Niveau, was unter anderem daran liegen, dass alle Testsreihen fast die identischen Informationen zur Regionenbewertung verwenden (Abb. 27 (d)).
Wenn *precision* und *recall* bei vielen Bildern sehr unterschiedlich sind, das heißt *precision* ist häufig deutlich höher als *recall* und umgekehrt, fällt der *F-measure* im Mittelwert deutlich niedriger aus als wenn man ihn mit den Mittelwerten von *precision* und *recall* berechnen würde. Daher sind die Werte in Abbildung 27 (d) korrekt, obwohl es auf den ersten Blick falsch aussieht.

Ausgehend von den Konfigurationen von Test 4c und Test 5b (Tab. 12) ist noch der Einfluss der Bewertung mittels der SURF-Keypoints interessant, da beide Verfahren die Regionen sehr unterschiedlich bestimmen (Tab. 10) und denoch vergleichbare Ergebnisse erzielen (Tab. 12). Hierfür werden zum einem die Differenz der *F-measure* Werte unter den beiden Verfahren sowie zu den markierten SURF-Keypoints bestimmt (Abb. 28). Vor allem das helligkeitsbasierte Verfahren hat bei einem Großteil der Bilder nur sehr geringe Abweichung zu den Ergebnissen, die nur mit den SURF-Keypoints erzielt werden, was an den kleinen Regionen liegt.

Laufzeitmessungen
Das *watershed*-Verfahren ist deutlich schneller als die helligkeitsbasierte Regionenermittlung (Abb. 29), was vor allem an der deutlich schnelleren Bestimmung und geringeren Anzahl der Regionen liegt. Unterschiede bei der Bestimmung der SURF-Keypoints und der Kennzeichnung ihrer Umgebung gibt es natürlich nicht, da diese Schritte nur von dem Originalbild abhängen. Am längsten dauert das Zusammenfügen von Regionen, was nur bei *Test 4c* ausgeführt wird (Abb. 29). Der letzte Schritt, in dem Regionen auf Basis der Informationen der Keypoints verknüpft werden, hat auch noch einen großen Anteil an der Gesamtlaufzeit. An dieser Stelle besteht mit Sicherheit noch Optimierungspotenzial, so dass sich die Laufzeit verbessern lassen sollte.

4.4.3 Fazit

Zum Abschluss der Evaluation stehen die optimalen Konfigurationen der verschiedenen Verfahren fest. In einem ersten Schritt wurde für jedes Verfahren geprüft mit welcher Konfiguration es Regionen am präzisesten ermittelt. Ausgehend von ermittelten verfahrensspezifischen Parametern wurde geprüft wie und ob die Nachbearbeitungsalgorithmen (Hintergrunderkennung, Regionen zusammenfügen) sinnvoll eingesetzt werden können. Auf die Regionen wurden im nächsten Schritt die SURF-Keypoints übertragen. Dabei musste zunächst der Grenzwert ermittelt werden, den Keypoints überschreiten müssen, damit sie Bildpunkte als wichtige markieren dürfen. Mit diesen Informationen ist es möglich die Saliency Maps zu erstellen. Da der Vergleich mit den Referenzbildern und damit die objektive Bewertung der Verfahren die Konvertierung zu einem Binärbild erfordert, wurden in einem letzten Schritt noch konstante



(a) precision

(b) recall

(c) F-measure

(d) Saliency Map Grenzwert

Figure 26: Die Abbildung zeigen die Ergebnisse der Tests in Abhängigkeit des SURF-Grenzwertes.

	SURF-Grenzwert	∅ SM-Grenzwert	*precision*	*recall*	*F-measure*
Test 4					
a	1.500	0,52	59,97%	85,86%	67,23%
b	1.500	0,51	61,05%	85,74%	68,15%
c	1.500	0,52	64,36%	86,93%	70,79%
Test 5					
a	1.000	0,63	66,67%	82,90%	68,48%
b	1.500	0,54	69,68%	75,20%	68,18%
c	1.500	0,54	69,42%	75,27%	68,12%

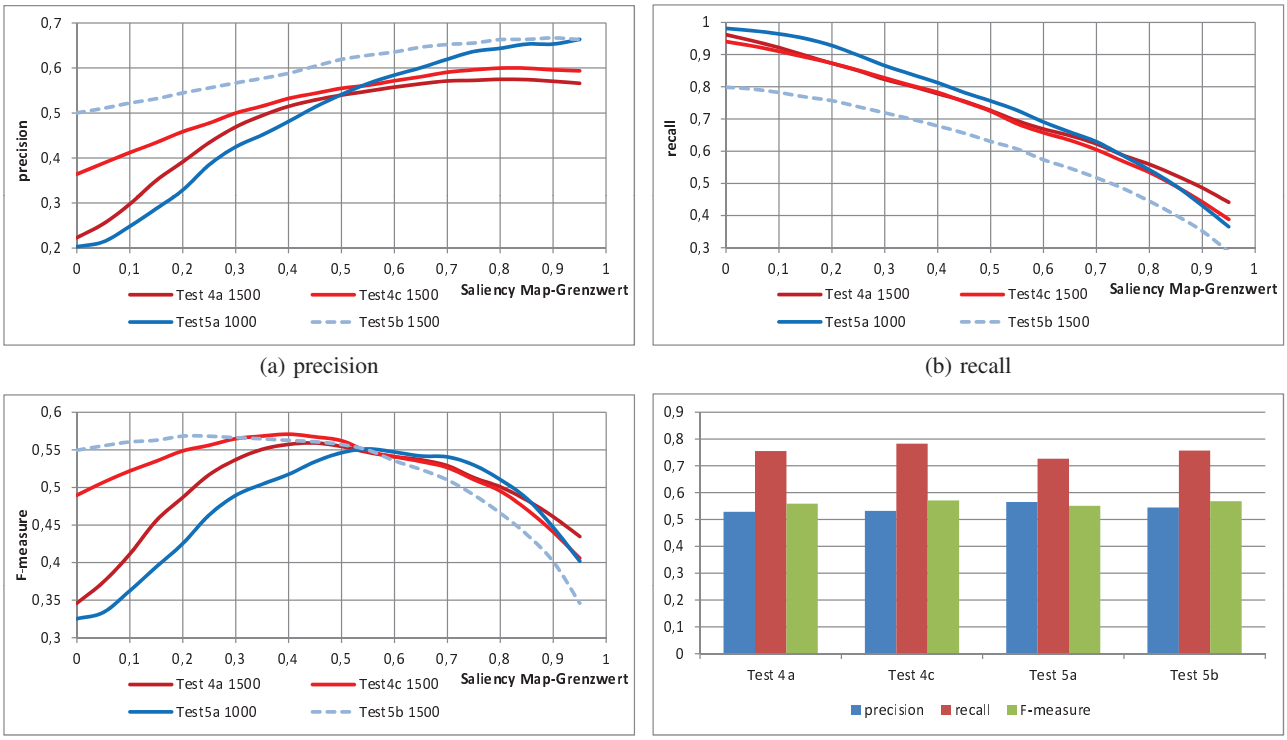
Table 11: Die Tabelle zeigt unter anderem bei welchem SURF-Grenzwert die Ergebnisse der verschiedenen Tests optimal sind. Der SM-Grenzwert ist der Wert, ab dem Bildpunkte der Saliency Maps für den Vergleich mit dem Referenzbild übernommen werden.

Grenzwerte für diese Konvertierung bestimmt. Somit ergeben sich folgende Konfigurationen:

- **Regionen mit ähnlicher Helligkeit**
  - erlaubte Helligkeitsdifferenz: 15
  - einfache Version der Hintegrunderkennung
  - Zusammenfügen der Region (erforderlicher Anteil: 75%, Mindestlänge: 40 px)
  - SURF-Grenzwert: 1.500
  - Saliency-Map-Grenzwert: 0,4 (bzw. $0,4 * 255$)
- ***watershed*-Verfahren**
  - Größe der Umgebung: 5; Vergrößerung: 2; Reduktion des Mittelwerts: 5; Mindestgröße: 5
  - erlaubte Helligkeitsdifferenz: 15
  - einfache Version der Hintegrunderkennung
  - Zusammenfügen der Region (erforderlicher Anteil: 75%, Mindestlänge: 40 px)
  - SURF-Grenzwert: 1.500
  - Saliency-Map-Grenzwert: 0,2 (bzw. $0,4 * 255$)

Beide Verfahren erzielen qualitativ gleichwertige Ergebnisse. Daher ist das *watershed*-Verfahren nur auf Grund der deutlich besseren Laufzeit zu bevorzugen. Je nach Anwendungsbereich kann es sogar sinnvoll nur die Keypoints ohne Regionen zu verwenden, um Saliency Maps zu erstellen. Hierfür spricht die äußerst gute Laufzeit von nur 60 Millisekunden (Tab. 5) und Abweichungen bei *precision*, *recall* und *F-measure* von weniger als zehn Prozent.

Die Ergebnisse sind dann gut, wenn sich die wichtigen Regionen klar vom Hintergrund abheben und dieser zudem nicht besonders kontrastreich ist. So können die Regionen genau bestimmt werden, das heißt es gibt wenige Regionen, die wichtige und unwichtige Bereiche abdecken. Desweiteren ist so gewährleistet, dass sich



(a) precision (b) recall

(c) F-measure (d) Ergebnisse

Figure 27: Die Abbildungen zeigen die Ergebnisse der Tests in Abhängigkeit des Grenzwertes der Saliency Maps.

	SM-Grenzwert	*precision*	*recall*	*F-measure*	∅ Laufzeit
Test 4					
a	0,45	52,88%	75,54%	55,90%	1.314,87 ms
c	0,4	53,22%	78,26%	57,08%	2.525,66 ms
Test 5					
a	0,55	56,54%	72,69%	55,12%	234,11 ms
b	0,20	54,47%	75,70%	56,84%	245,98 ms

Table 12: Ergebnisse der Saliency Maps (SM). Die Tests entsprechen denen in Tabelle 11 mit dem einzigen Unterschied, dass ein konstanter Grenzwert zur Überführung der Saliency Maps in Binärbilder gewählt wurde.

SURF-Keypoints innerhalb bzw. zumindest am Rand der wichtigen Region befinden, da hier hohe Kontrastunterschiede vorliegen. Die wichtigen Objekte können durchaus kontrastreich sein, da sich dann definitiv auch Keypoints innerhalb der wichtigen Region befinden, so dass die Keypoints das komplette Objekt abdecken können. Dies hat allerdings den Nachteil, dass im wichtigen Objekt viele kleinere Regionen ermittelt werden. Das ist dann problematisch, wenn es zum Hintergrund keinen klaren Kontrast gibt oder der Hintergrund ebenfalls kontrastreich ist.

5. ZUSAMMENFASSUNG UND AUSBLICK

In dieser Arbeit wurde das SURF-Verfahren beschrieben und dessen Eignung zur Beschreibung und Bewertung von Bildregionen untersucht. Dabei hat sich herausgestellt, dass das Verfahren Regionen am besten bewertet, wenn man Ansätze verwendet, die auf der Keypointdichte innerhalb einer Region basieren. Da viele Keypoints knapp außerhalb der wichtigen Region liegen, ist es sinnvoll den Bereich um den Keypoint ebenfalls als wichtig zu markieren, da er so noch Bereiche innerhalb der wichtigen Region kennzeichnen kann. Aus dieser Erkenntnis leitet sich zudem ab, dass es nicht möglich ist Objekte mittels Keypoints zu segmentieren, indem man die Keypoints gruppiert und so das Objekt mit ihnen einschließt.

Zur Bestimmung der Regionen wurden zwei Verfahren verwendet, die zudem um Nachbearbeitungsalgorithmen erweitert wurden. Es wurden Bildpunkte mit ähnlicher Helligkeit zu Regionen zusammengefasst sowie das *watershed*-Verfahren verwendet. Ausgehend von den bestimmten Regionen wurden Regionen als Hintergrund markiert, die mit mehreren Seitenrändern verbunden sind. In einem weiteren Schritt wurde eine Region, die von einer anderen Region komplett oder anteilig umschlossen wurde, zu dieser hinzugefügt. Beides hat zur Folge, dass die durchschnittliche Größe der Regionen gestiegen ist, wodurch die Genauigkeit etwas zurück ging. Da dafür aber mehr wichtige Bildpunkte erkannt wurden, hat sich das dennoch positiv auf das Ergebnis ausgewirkt.

Insgesamt wurden mit beiden Regionenermittlungsverfahren sehr ähnliche Ergebnisse erzielt, was darauf zurückzuführen ist, dass sie zur Bewertung der Regionen die gleichen Informationen verwendet

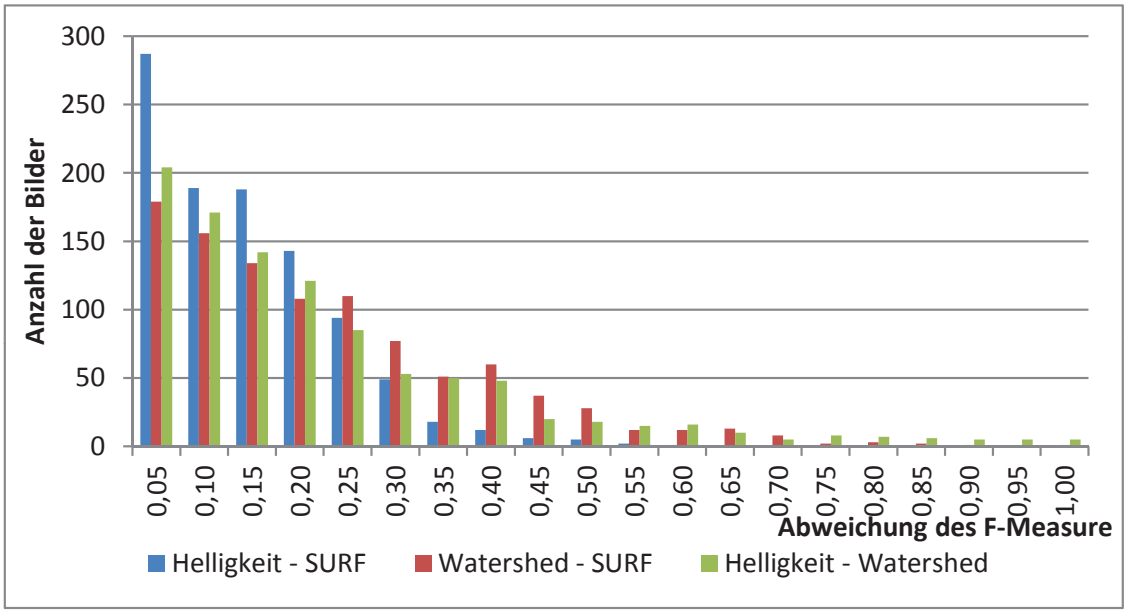


Figure 28: Die Grafik verdeutlicht wie sich die F-Measure-Werte der verschiedenen Verfahren bei den einzelnen Bildern unterscheiden.

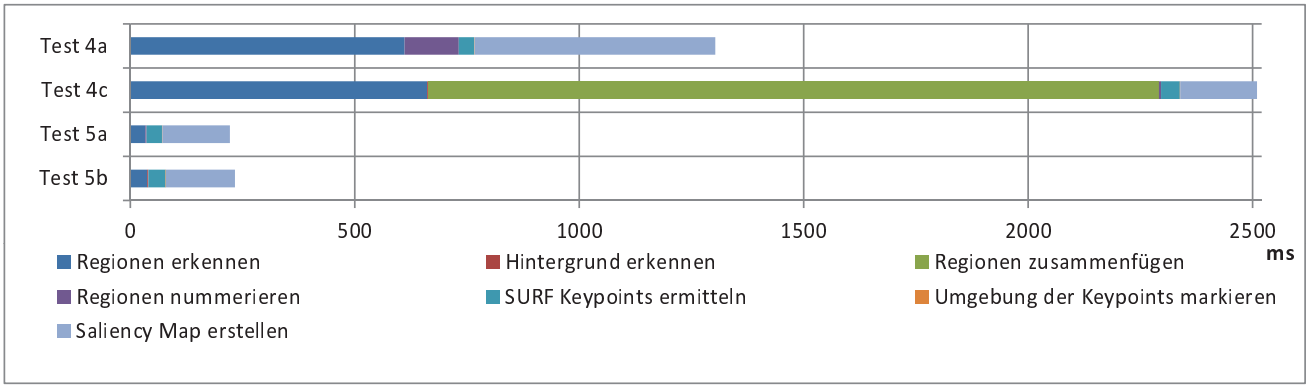


Figure 29: Übersicht der ∅ Laufzeiten der einzelnen Phasen innerhalb der Testreihen (Tab. 12) Manche Legendeneinträge sind im Diagramm nicht zu erkennen, da ihre Werte entweder zu niedrig sind oder sie in dem betreffenden Test nicht ausgeführt wurden.

haben. Es wurden etwa 70% bis 80% der wichtigen Region abgedeckt und der Anteil aller als wichtig markierten Bildpunkte, die nicht innerhalb der wichtigen Region lagen, betrug etwas weniger als 50%. Während also fast die komplette wichtige Region markiert wurde, ist vor allem die Genauigkeit mit der dies geschehen ist nicht sehr hoch.

Wenngleich die erreichten Ergebnisse nicht optimal sind, sollte es durchaus möglich sein wichtige Bildpunkte zu nutzen, um wichtige Regionen zu erkennen. Das Problem ist hierbei in erster Linie eine zuverlässige Erkennung von Bildregionen. An dieser Stelle sollte angesetzt werden und neue Verfahren entwickelt werden, die hierbei gute Ergebnisse erzielen. Sollte dies gelingen, dann dürften die Ergebnisse insgesamt deutlich besser ausfallen, da die Keypointdichte zumindest auf dem hier verwendeten Testdatensatz in der wichtigen Region deutlich höher ist als im Rest des Bildes. Dabei ist es vor allem wichtig, dass wenige Regionen gefunden werden, die die wichtigen Bildbereiche sehr genau abdecken. Sind die Regionen zu klein, entspricht das Ergebnis fast dem des SURF-Verfahrens ohne die Verwendung von Regionen. Da das SURF-Verfahren sehr effizient und effektiv ist, sollte dies auch für die Erkennung von Regionen gelten, damit das gesamte Verfahren eine sehr gute Laufzeit erreichen kann. Desweiteren sollte sich die Qualität des Ergebnisses deutlich verbessern lassen, wenn man einen adaptiven Grenzwert zur Überführung des Graustufenbildes, auf dem die Relevanz eines Bildpunktes seiner Helligkeit entspricht, in ein Binärbild, auf dem eindeutig zwischen wichtiger und unwichtiger Region unterschieden wird, verwenden würde. Bei einem konstanten Grenzwert für einen kompletten Datensatz ist das Ergebnis nicht optimal. Desweiteren hat sich gezeigt, dass die hier verwendeten Ansätze zur Nachbearbeitung der Regionen die Ergebnisse positiv beeinflussen. Da diese Algorithmen komplett unabhängig von dem Regionenermittlungsverfahren sind, ist es interessant zu untersuchen, ob sie sich auch in andere Verfahren sinnvoll integrieren lassen.

6. ANHANG

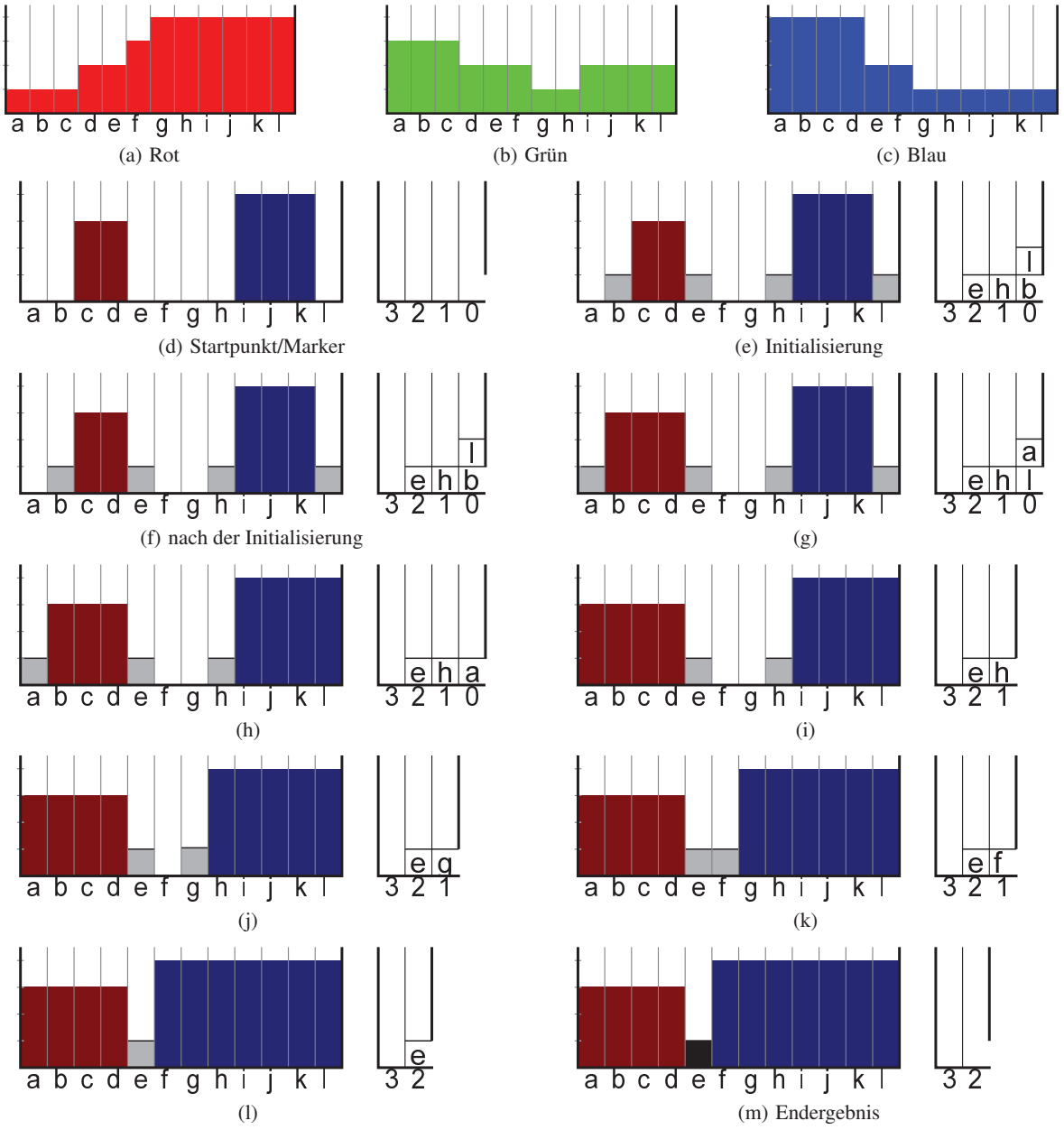


Figure 30: Beispiel zu Watershed: Beispiel zu Watershed: Die einzelnen Farbkanäle eines RGB-Bildes mit einem Wertebereich von 1-4 (a - c). Initialisierung der Warteschlangen (e) anhand der Startpunkte (d) und den Farbkanälen (a - c). In beiden Abbildung (d + e) befindet sich links das Regionenbild und rechts die Warteschlangen. Es gibt zwei Startregionen (dunkelrot (3) und dunkelblau (4)). Bildpunkte, die sich in einer Warteschlange befinden, sind grau markiert. Vergrößerung der Regionen: Die Warteschlangen werden abgearbeitet und die Regionen schrittweise vergrößert. In Grafik (m) wird der Bildpunkt 'e' schwarz markiert, da er zwischen zwei Regionen liegt, d.h. er ist Grenzpunkt (Quelle: [28]).

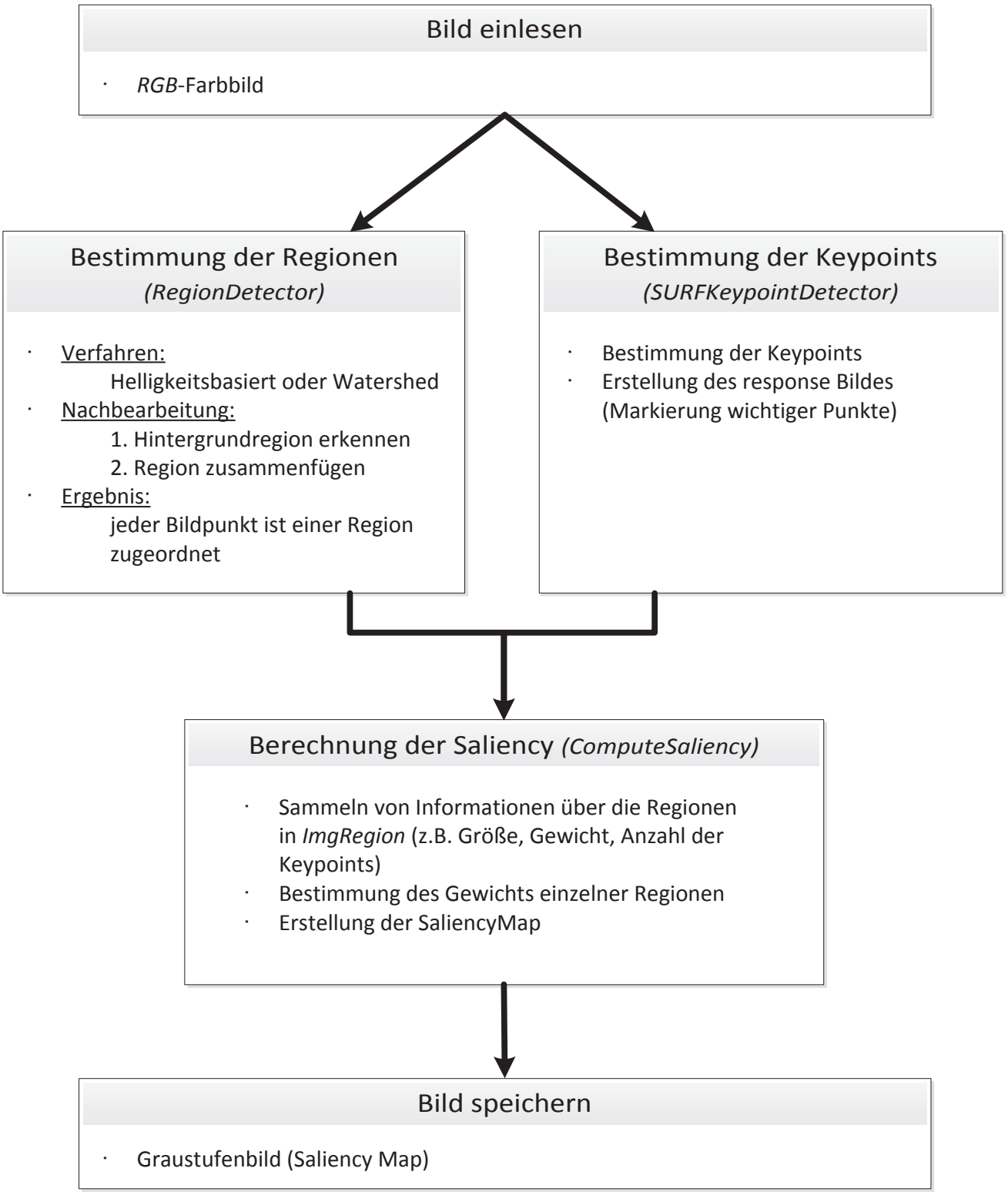


Figure 31: Die Grafik verdeutlicht den Zusammenhang der verschiedenen Phasen.

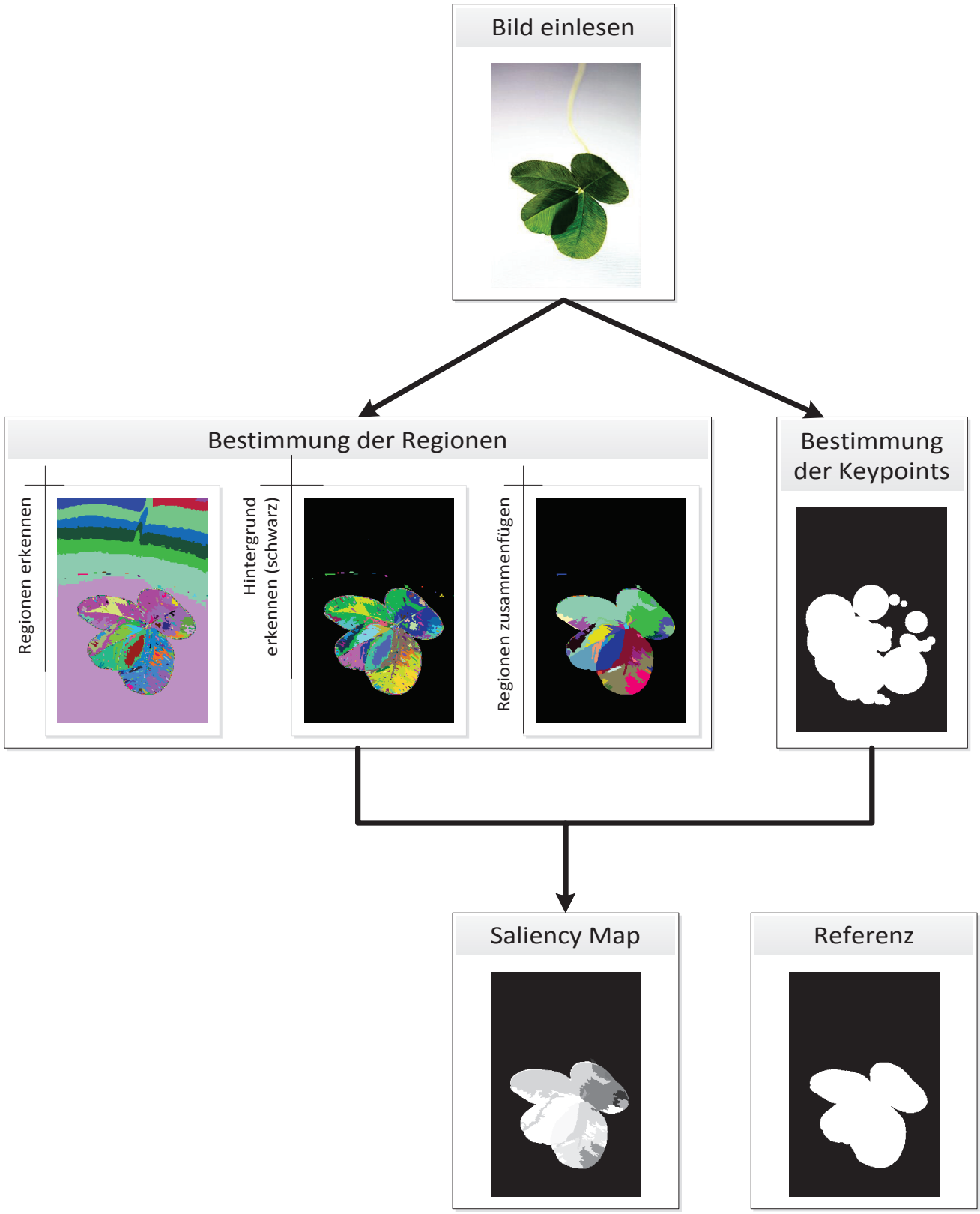


Figure 32: Diese Grafik stellt das Ergebnis der verschiedenen Phasen visuell dar.

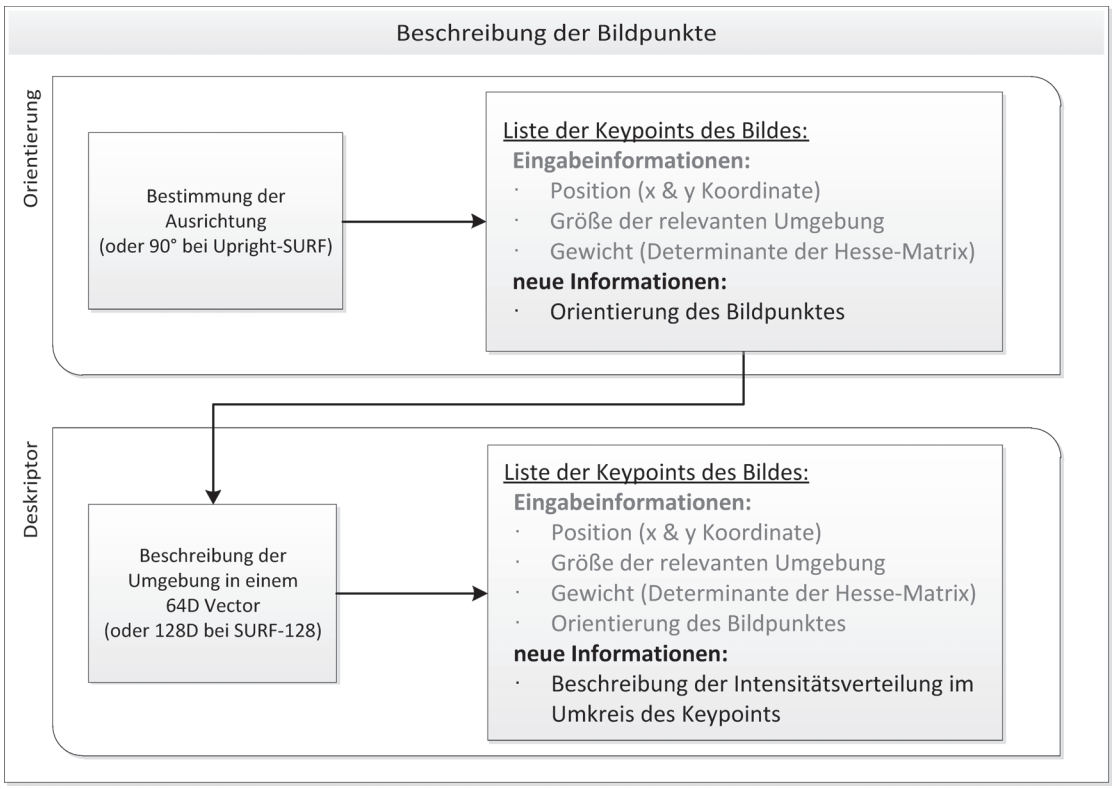


Figure 33: Die Diagramme verdeutlichen die Zusammenhänge der verschiedenen Schritte innerhalb des SURF-Verfahrens. Desweiteren wird dargestellt, welche Informationen beim jeweiligen Schritt gewonnen werden und welche als Eingabeinformation nötig sind.

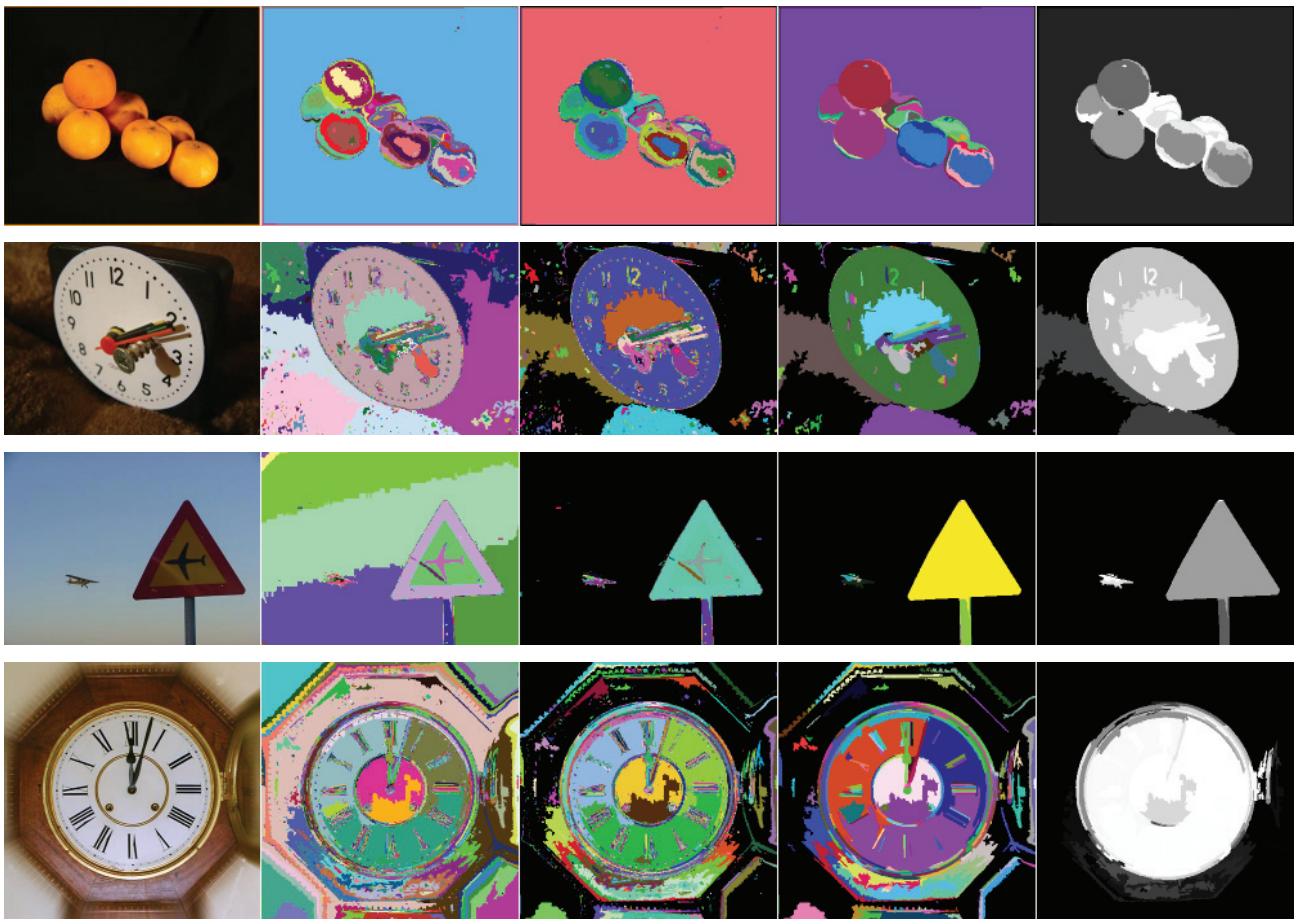


Figure 34: Die Bilder verdeutlichen die helligkeitsbasierte Regionenermittlung und den Einfluss der Nachbearbeitungsverfahren. Die Bilder zeigen jeweils von links nach rechts: das Originalbild [24], die bestimmten Regionen, die Erkennung der Hintergrundregionen, das Zusammenfügen von Regionen und die Saliency Map.

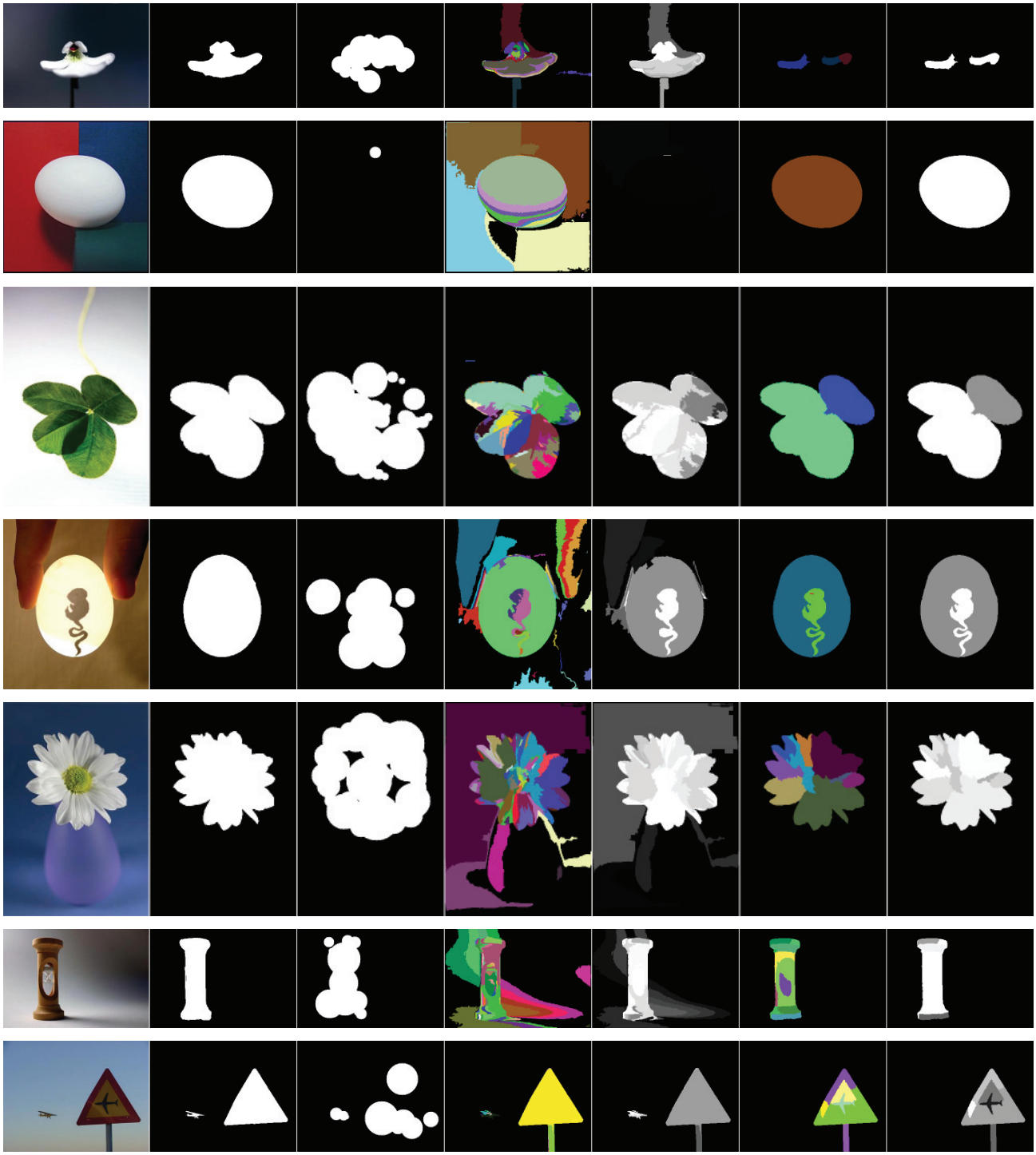


Figure 35: Die Bilder zeigen die Ergebnisse der verschiedenen Verfahren. Die Bilder zeigen jeweils von links nach rechts: das Originalbild [24], das Referenzbild, die markierten SURF-Keypoints, die helligkeitsbasierten Regionen, die helligkeitsbasierte Salieny Map, die watershed-Regionen sowie die watershed-Saliency Map.

7. REFERENCES

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 1597–1604. IEEE, 2009.

[2] R. Achanta and S. Süsstrunk. Saliency detection for content-aware image resizing. *16th IEEE International Conference on Image Processing (ICIP), 2009*, pages 1005 – 1008, 2009.

[3] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics (TOG)*, 2007.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110(3):246–359, 2008.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] M. Brown and D. Lowe. Invariant features from interest point groups. *British Machine Vision*, 2002.

[7] M. Cheng and G. Zhang. Global contrast based salient region detection. *Conference on Computer Vision and Pattern Recognition (CVPR), 2011 IEEE*, 2011.

[8] D. Farin, T. Haenselmann, S. Kopf, G. Kühne, and W. Effelsberg. Segmentation and classification of moving video objects. In B. Furht and O. Marques, editors, *Handbook of Video Databases: Design and Applications*, volume 8 of *Internet and Communications Series*, pages 561–591. CRC Press, Boca Raton, FL, USA, 2003.

[9] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.

[10] J. Han and K. Ngan. Unsupervised extraction of visual attention objects in color images. *IEEE Transactions on Circuits and Systems for Video Technology*, 2006.

[11] S. He, J. Han, X. Hu, and M. Xu. A biologically inspired computational model for image saliency detection. *Proceedings of the 19th ACM international conference on Multimedia*, 2011.

[12] J. Kiess, B. Guthier, S. Kopf, and W. Effelsberg. SeamCrop: Changing the size and aspect ratio of videos. In *Proceedings of the 4th Workshop on Mobile Video*, MoVid '12, pages 13–18, New York, NY, USA, 2012. ACM.

[13] J. Kiess, S. Kopf, B. Guthier, and W. Effelsberg. Seam carving with improved edge preservation. In *Proceedings of IS&T/SPIE Electronic Imaging (EI) on Multimedia on Mobile Devices*, volume 7542(1), pages 75420G:01 – 75420G:11, January 2010.

[14] S. Kopf and W. Effelsberg. Mobile cinema: Canonical processes for video adaptation. In *Multimedia Systems*, volume 14(6), pages 369–375. Springer, December 2008.

[15] S. Kopf, B. Guthier, H. Lemelson, and W. Effelsberg. Adaptation of web pages and images for mobile applications. In *Proceedings of IS&T/SPIE conference on Multimedia on Mobile Devices*, volume 7256, pages 72560C–12, 2009.

[16] S. Kopf, T. Haenselmann, and W. Effelsberg. Enhancing curvature scale space features for robust shape classification. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, July 2005.

[17] S. Kopf, T. Haenselmann, and W. Effelsberg. Robust character recognition in low-resolution images and videos. Technical Report TR-05-002, Department for Mathematics and Computer Science, University of Mannheim, Germany, 2005.

[18] S. Kopf, T. Haenselmann, and W. Effelsberg. Shape-based posture and gesture recognition in videos. In *Proceedings of IS&T/SPIE Electronic Imaging (EI) on Storage and Retrieval Methods and Applications for Multimedia*, volume 5682, pages 114–124, January 2005.

[19] S. Kopf, T. Haenselmann, D. Farin, and W. Effelsberg. Automatic generation of summaries for the Web. In *Proceedings of IS&T/SPIE Electronic Imaging (EI) on Storage and Retrieval Methods and Applications for Multimedia*, volume 5307, January 2004.

[20] S. Kopf, T. Haenselmann, D. Farin, and W. Effelsberg. Automatic generation of video summaries for historical films. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 27–30. IEEE Computer Society Press, June 2004.

[21] S. Kopf, T. Haenselmann, J. Kiess, B. Guthier, and W. Effelsberg. Algorithms for video retargeting. *Multimedia Tools and Applications (MTAP), Special Issue: Hot Research Topics in Multimedia*, 51:819–861, January 2011.

[22] S. Kopf, J. Kiess, H. Lemelson, and W. Effelsberg. FSCAV: Fast seam carving for size adaptation of videos. In *Proceedings of the 17th ACM international conference on Multimedia (MM)*, pages 321–330, October 2009.

[23] S. Kopf, F. Lampi, T. King, and W. Effelsberg. Automatic scaling and cropping of videos for devices with limited screen resolution. In *Proceedings of the 14th ACM international conference on Multimedia (MM)*, pages 957–958, October 2006.

[24] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. In *Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR*, pages 1–8. CVPR, 2007.

[25] M. Livingstone and D. Hubel. Segregation of Form, Color, Movement, and Depth: Anatomy, Physiology, and Perception. *Science*, 240:740–749, 1988.

[26] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the 10th ACM international conference on Multimedia*, pages 533–542. ACM Press, 2002.

[27] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. *Proceedings of the eleventh ACM international conference on Multimedia*, 2003.

[28] F. Meyer. Color image segmentation. *Image Processing and its Applications, 1992.*, 1992.

[29] M. Nixon and A. Aguado. *Feature Extraction & Image Processing (second edition)*. Elsevier Ltd., 2008.

[30] S. Richter, G. Kühne, and O. Schuster. Contour-based classification of video objects. In *Proceedings of IS&T/SPIE conference on Storage and Retrieval for Media Databases*, volume 4315, pages 608–618, January 2001.

[31] T. B. Terriberry, L. M. French, and J. Helmsen. GPU Accelerating Speeded-Up Robust Features. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 355–362. Citeseer, 2008.

[32] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2001.