

Stochastische Mustererkennung zur Bildsegmentierung

Stochastic pattern recognition for image segmentation

Thomas Müller¹ und Bastian Erdnütz²

¹ Fraunhofer IOSB - Institut für Optronik, Systemtechnik u. Bildauswertung,
Fraunhoferstr. 1, 76131 Karlsruhe

² Karlsruher Institut für Technologie,
Institut für Photogrammetrie und Fernerkundung,
Englerstr. 7, 76131 Karlsruhe

Zusammenfassung Luftbilder zeigen häufig weite homogene Bereiche mit inhomogener Textur. Das sind Bereiche, die zwar lokal eine hohe Schwankung in den Farbwerten aufweisen, die jedoch stochastisch ein augenscheinlich so regelmäßiges Muster zeigen, dass man sie als Mensch leicht als zusammengehörige Fläche wie einen Wald, ein Feld, einen Weg oder ein Gewässer erkennt. Mit der vorliegenden Arbeit sollen solche Bereiche erkannt werden. Dazu werden zunächst Texturmodelle zu verschiedenen Bildbereichen generiert, die strukturelle Ähnlichkeit ermittelt und hinreichend unterschiedliche Texturmodelle gespeichert. Anschließend wird jedes Pixel des Bildes via Strukturmetrik einem der gefundenen Texturmodelle zugeordnet und entsprechend charakteristisch eingefärbt. Dabei werden drei unterschiedliche Ansätze zur Texturmodellierung untersucht und miteinander verglichen im Hinblick auf das Ziel, möglichst viele Bildbereiche als gleichartig zu segmentieren, die von einem Menschen als ähnlich angesehen werden (z.B. alle Baumkronen im Bild, Wege oder Äcker) und Segmentgrenzen dort zu erzielen, wo augenscheinliche Übergänge sind (z.B. Waldgrenzen, Auto auf einer Wiese, Fahrzeugsuren im Gelände, Vegetationsveränderungen).

Schlagwörter Bildtextur, Bildsegmentierung, Objekterkennung, Landnutzung.

Abstract Aerial image often show wide homogeneous regions with inhomogeneous texture. Such regions may contain high local color dynamics but nevertheless show apparently regular stochastic patterns which make it easy for a human observer to recognize them as associated structures like forests, acres, pathways, or waterbodies. This paper intends to recognize such regions. To do this, first, texture models are generated for different image regions, structural similarities are determined and sufficiently different texture models are stored. Afterwards, each image pixel is assigned to one of those texture models with a structural metric and is then colored characteristically for the assigned texture model. In this procedure, three different methods of texture modelling are examined and compared with each other in view of the goal to reach a segmentation result in which as many regions as possible are found to be similar that are seen as similar also by a human observer (e.g., all treetops in the image, pathways, or acres) and in which segment margins correspond to apparent transitions (e.g., boundary of a forest, a car on a grassland, vehicle tracks in the field, vegetation changes).

Keywords Image texture, image segmentation, object recognition, land use.

1 Einleitung

In dieser Arbeit werden drei unterschiedliche Textur-Verfahren untersucht und verglichen, welche die lokale stochastische Struktur von Bildbereichen modellieren. Ziel ist es, strukturell ähnliche und unterschiedliche Regionen in einem Bild möglichst gut zu erkennen, um mit diesen später weiterarbeiten zu können. Dabei wird auf kein vorab trainiertes Modellwissen zurückgegriffen. Die erfolgende Nutzung zur texturbasierten Bildsegmentierung bedeutet zum Einen eine Anwendung an sich, aber zum Anderen auch unmittelbar eine günstige Visualisierung der Ergebnisse zur Einschätzung der Leistung der Verfahren bzw. der qualitativen Eigenschaften und Unterschiede.

Das Basisvorgehen ist bei allen drei Ansätzen das gleiche. Zunächst werden gleichmäßig über das Bild verteilt Struktur-Beschreibungen berechnet und eine Teilmenge hinreichend unähnlicher Beschreibungen daraus ermittelt. Anschließend wird für jedes Bildpixel über ei-

ne Struktur-Metrik berechnet, ob und zu welcher dieser Struktur-Beschreibungen es passt. Dadurch ergibt sich die Segmentierung des Bildes, die farblich visualisiert wird.

Im Folgenden werden drei Texturmodelle mit ihren Struktur-Metriken untersucht: Eine basierend auf Auto-Regressiven Filtern (AR-Filtern), eine basierend auf einer Erweiterung der Schablonenanpassung (engl. Template Matching) und eine basierend auf Histogrammen von Grauwerten/Farben und Farbübergängen (vgl. HoG-Features, HoG = Histogram of oriented Gradients).

2 Auto-Regressiver Filter (AR-Filter)

2.1 Texturmodell

Um ein Test-Pixel x herum wird eine Umgebung A_x betrachtet. Für jedes Pixel $y \in A_x$ in der Umgebung werden die Intensitätswerte der wiederum y umgebenden Pixel $z \in B_y \setminus \{y\}$ zu einem Vektor v_y aufgelistet. Diese Vektoren werden als Zeilen einer Matrix $M_x := (v_y^T)_{y \in A_x}$ zusammengefasst. Die dabei ausgesparten Intensitätswerte der Zentralpixel y bilden die entsprechenden Komponenten eines Vektors b .

Für jedes Pixel $y \in A_x$ entsteht so eine Zeile in M_x bzw. b . M_x und b haben $m := |A_x|$ Zeilen, und M_x hat $n := |B_y \setminus \{y\}|$ Spalten, wobei $|\cdot|$ die Kardinalität einer Menge ist. Jede Zeile im linearen Gleichungssystem

$$M_x p = b \tag{1}$$

bedeutet anschaulich, dass ein Zentralpixel y (in b) als Linearkombination p der Umgebung $B_y \setminus \{y\}$ (Spalten der Zeile in M_x) beschrieben wird. Wenn also das Testpixel x Teil einer stochastisch einheitlichen Textur ist, so ist die Umgebung B_x um x herum ein „Sample“ dieser Textur, ebenso wie auch alle Umgebungen B_y um die benachbarten Pixel $y \in A_x$. Grundidee des Ansatzes ist, dass dieses Sample durch Lösen des Gleichungssystems 1 nach der kleinste Quadrate-Methode gefunden wird.

Grundsätzlich können A_x und B_y von beliebiger (aber fixer) Form sein. Die Reihenfolge bei der Abarbeitung der Pixel in den beiden Umgebungen entscheidet über die Reihenfolge der Zeilen und Spalten in M_x bzw. b . Sie ist grundsätzlich beliebig, muss aber bei allen Berechnungen fix bleiben. In dieser Arbeit werden die Umgebungen A_x und B_y kreisförmig gewählt mit den Radien r_A bzw. r_B Pixel. Die Reihenfol-

Tabelle 1: Größen m und n in Abhängigkeit der Radien r_A bzw. r_B .

r_A bzw. r_B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m bzw. $n + 1$	5	13	29	49	81	113	149	197	253	317	377	441	529	613	709
r_A bzw. r_B	16	17	18	19	20	21	22	23	24	25	26				
m bzw. $n + 1$	797	901	1009	1129	1257	1373	1517	1653	1793	1961	2121				

ge bei der Abarbeitung erfolgt ähnlich wie bei Pixelbildern zuerst zeilenweise und dann spaltenweise. Tabelle 1 illustriert die Anzahl Pixel m bzw. n in den Umgebungen A_x bzw. B_y abhängig von r_A bzw. r_B .

Die Umgebung A_x muss mindestens so groß wie B_y sein, d.h. $r_A \geq r_B$, sodass $m \geq n$ und das lineare Gleichungssystem 1 bei geeigneten Eingabedaten gelöst werden kann. Wird keine Lösung gefunden, so existiert keine hinreichende Widerspruchsfreiheit in der Umgebung um x und somit befindet sich dort auch keine via AR-Filter beschreibbare Textur. Kann eine Lösung berechnet werden, so ist der ermittelte n -dimensionale Vektor p eine Beschreibung der Textur um den Punkt x , d.h. *Texturmodell* zu x .

Die bisherige Beschreibung mit Intensitätswerten geht von Grauwertbildern aus. Bei Farbbildern werden drei Gleichungssysteme 1 (für jeden Farbkanal eines) aufgestellt und gelöst. Alternativ wurde auch untersucht, nur ein Gleichungssystem mit $3m$ Zeilen und $3n$ Unbekannten zu lösen. Dies führte jedoch zu etwas schlechteren Endergebnissen.

Die erhaltenen Ergebnisse lassen sich verbessern, wenn statt des Gleichungssystems 1 das entsprechende mit mittelwertbereinigten Spalten gelöst wird. D.h. vor Lösung des Gleichungssystems wird zu jeder Spalte in M_x und b der Mittelwert berechnet und von jedem Spaltenelement subtrahiert.

Beim zu lösenden Gleichungssystem kommt es nur darauf an, ob sich Pixel unterscheiden oder nicht. Große Unterschiede sind nicht gravierender als kleine und müssen daher auch nicht stärker gewichtet werden. Eine weitere Verbesserung der Endergebnisse stellt man daher fest, wenn das Gleichungssystem 1 so gelöst wird, dass die L1-Norm des Restfehlers minimiert wird anstatt der Euklidnorm (Huber-Schätzer, Reweighting-Methode) [1,2].

Werden r_A und r_B so gewählt, dass sich m und n zu wenig unterscheiden, stellt man schlechte Endergebnisse fest. Ebenso wirkt es sich auch ungünstig aus, wenn m viel größer als n ist. Oft (aber nicht immer)

wurden die besten Endergebnisse bei $r_A = r_B + 1$ beobachtet. Letztendlich hat am besten funktioniert, r_B vorzugeben und für r_A den kleinsten Wert zu wählen, bei dem M_x mindestens 50 Prozent mehr Zeilen als Spalten hat, d.h. $m \geq 1,5 \cdot n$ (vgl. Tabelle 1).

2.2 Evaluation

Zur Quantifizierung, inwieweit ein Pixel x einem zuvor (i.A. zu einem völlig anderen Pixel t) berechneten Texturmodell p folgt, wird zunächst M_x und b um x sowie die Differenz $d := M_{xp} - b$ berechnet und die *Struktur-Metrik* definiert als

$$\delta := \sum_{i=1}^m \left(\begin{cases} 1, & \text{falls } |d_i| > \epsilon \\ 0, & \text{sonst} \end{cases} \right) + 10^{-6} \cdot |d_i| \quad , \quad (2)$$

wobei d_i die i -te Komponente von d bezeichnet. Der erste Summand nach dem Summenzeichen ist der wesentliche, er zählt die Pixel in A_x , die dem Texturmodell widersprechen. Falls bei zwei Texturmodellen das gleiche Zählergebnis entsteht, sorgt der zweite Summand dafür, dass das Modell mit der kleineren Abweichung in d zu einem kleineren Wert δ führt. ϵ ist eine einstellbare Konstante, die angibt, bis zu welcher Differenz die Intensitätswerte zweier Pixel nicht unterschieden werden sollen, z.B. $\epsilon = 15$. Bei Berechnung von d wird analog zu Abschnitt 2.1 mit mittelwertbereinigten Spalten gerechnet.

x wird als *kompatibel zum Modell p* angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist. In dieser Arbeit wird der relativ kleine Wert $c = 30\%$ verwendet. Dies ermöglicht die Zuordnung von Texturen zueinander auch bei starken örtlich konzentrierten oder verteilten Unterschieden.

2.3 Bildsegmentierung

Über das Bild wird ein regelmäßiges Raster gelegt und an den Rasterpunkten die Textur untersucht. In einer Liste von Texturmodellen werden die gefundenen, unterschiedlichen Texturmodelle gespeichert. Dazu wird für jeden Rasterpunkt zunächst geprüft, ob er kompatibel ist zu einem bereits gespeicherten Texturmodell der Liste. Falls das nicht der Fall ist, wird p für den Rasterpunkt berechnet und in die Liste aufgenommen.

Zur Bildsegmentierung wird für jedes Bildpixel das Texturmodell der Liste mit dem kleinsten δ bestimmt. Ist das Pixel kompatibel zum gefundenen Texturmodell, wird das Pixel mit einer Farbe coloriert, die für dieses Texturmodell steht.

3 Schablonenanpassung (Template Matching)

3.1 Texturmodell

Wie beim AR-Filter wird die kreisförmige Umgebung A_x um ein Test-Pixel x mit m Elementen und wohldefinierter Reihenfolge der Pixelpositionen betrachtet. Die $k := m$ Intensitätswerte bei Grauwertbildern bzw. $k := 3m$ Farbwerte bei 3-Kanal-Farbbildern in der Umgebung A_x um x seien bezeichnet mit s_i^x , $i = 1, \dots, k$. s_*^x heisst *Texturmodell* zu x und dient als Schablone (engl. Template) für Texturvergleiche.

3.2 Evaluation

Ein Texturmodell s_*^t korrespondiert nach Definition mit einem Pixel t , um welchen es erzeugt wurde. Zur Quantifizierung, inwieweit ein anderes Pixel x dem gleichen Texturmodell folgt, werden zunächst die absoluten Differenzen

$$d_i := |s_i^x - s_i^t| \quad (3)$$

zwischen beiden Texturmodellen berechnet, $i = 1, \dots, k$. Die *Struktur-Metrik* wird definiert als

$$\delta := \sum_{i=1}^k \begin{cases} 1, & \text{falls } d_i > \epsilon \\ 10^{-6} \cdot d_i, & \text{sonst} \end{cases}, \quad (4)$$

wobei ϵ wieder eine einstellbare Konstante ist, die angibt, bis zu welcher Differenz die Intensitäts- bzw. Farbkanäle zweier Pixel nicht unterschieden werden sollen. In dieser Arbeit wird $\epsilon = 15$ verwendet. In der Formel wird im Wesentlichen gezählt, wieviele Elemente in s_*^x und s_*^t unterschiedlich sind. Die Fallunterscheidung in der Formel dient dazu, ähnliche Texturmodelle mit gleicher Zählung unterscheiden zu können.

x wird als *kompatibel zum Modell* s_*^t angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist.

3.3 Bildsegmentierung

Bei der Bildsegmentierung wird in einer ersten untersuchten Verfahrensvariante genauso vorgegangen wie in Abschnitt 2.3 beschrieben, wobei zur Erstellung der Liste von Texturmodellen $c = 50\%$ verwendet wird und beim Einfärben der Pixel $c = 40\%$.

Im Gegensatz zum AR-Filter, bei dem kleine Verschiebungen von Bildtexturen nach Konstruktion zu keiner bedeutenden Veränderung des Texturmodells führen, hat dieses Vorgehen den potenziell entscheidenden Nachteil, dass sich ein Texturmodell s_*^{At} bei einer Verschiebung von Bildtexturen (oder bei Veränderung der Pixelposition t) stark ändern kann. Solche Veränderungen werden zwar über das Konstrukt mit dem Parameter c kompensiert, dieses sollte sich jedoch besser nur auf inhaltlich-strukturelle Unterschiede beziehen. Daher wird in einer zweiten untersuchten Variante des Verfahrens eine Verschiebungsinvarianz integriert: Einem Pixel x wird bei der Bildsegmentierung nicht das Texturmodell zugeordnet, das nur genau bei x zur kleinsten Strukturmetrik δ führt, sondern das Texturmodell der Liste, das in der Umgebung A_x von x zum kleinsten δ führt.

4 Histogramme von Farben und Farbübergängen

4.1 Texturmodell

Wiederum wird die kreisförmige Umgebung A_x um ein Test-Pixel x mit m Elementen betrachtet (vgl. Abschnitte 2.1 und 3.1). Es sollen Histogramme berechnet werden. Dazu werden die Grauwerte bzw. Farben reduziert. r sei eine benutzerdefinierte Farbreduktionskonstante und $k := \lfloor 255/r \rfloor + 1$. Der Grauwert eines Grauwertbildes bzw. ein Farbkanal eines Farbbildes zu Pixel $y \in A_x$ sei bezeichnet mit $g_y \in \{0, \dots, 255\}$ und die Farbreduktion sei $\bar{g}_y := \lfloor g_y/r \rfloor \in K$, $K := \{0, \dots, k-1\}$. Als *Texturmodell* zu x wird das Histogramm h_* reduzierter Farben verwendet, welches wie folgt fünf Teilhistogramme konkateniert:

$$h_{i+0k} := |\{y \in A_x \mid \bar{g}_y = i\}|, \quad (5)$$

$$h_{i+1k} := |\{y \in A_x \mid \bar{g}_{y_r} - \bar{g}_y = i\}|, \quad (6)$$

$$h_{i+2k} := |\{y \in A_x \mid \bar{g}_y - \bar{g}_{y_r} = i, i \neq 0\}|, \quad (7)$$

$$h_{i+3k} := |\{y \in A_x \mid \bar{g}_{y_u} - \bar{g}_y = i\}|, \quad (8)$$

$$h_{i+4k} := |\{y \in A_x \mid \bar{g}_y - \bar{g}_{y_u} = i, i \neq 0\}|, \quad (9)$$

$i = 0, \dots, k - 1$. y_r bezeichne dabei das Nachbarpixel rechts von y und y_u das Nachbarpixel darunter.

Bei Farbbildern wird das für jeden Farbkanal gemacht und die drei sich ergebenden Teile ebenfalls konkateniert. Bei Grauwertbildern hat das Texturmodell h_* insgesamt $j := 5k$ und bei Farbbildern $j := 15k$ Einträge. In dieser Arbeit wird $r = 8$ verwendet, d.h. $k = 32$.

Anschaulich speichert h_{i+ak} , welche Intensitäten wie oft ($a = 0$) bzw. welche nach rechts/unten zunehmenden ($a = 1, 3$) und abnehmenden Gradienten ($a = 2, 4$) wie oft zwischen horizontalen ($a = 1, 2$) und vertikalen Nachbarpixeln ($a = 3, 4$) auftreten, in Form eines Vektors $h \in K^j$.

4.2 Evaluation

Ein Texturmodell $h = h_*^t$ korrespondiert nach Konstruktion mit einem Pixel t , zu dessen Umgebung A_t es berechnet wurde. Zur Quantifizierung, inwieweit ein anderes Pixel x dem gleichen Texturmodell folgt, wird zunächst h_*^x zu x berechnet. Es sei $U_i := \{i - 1, i + 1\} \cap K$ und

$$\beta := \sum_{\substack{i = 0, \dots, k-1 \\ a = 0, \dots, 4}} \begin{cases} \gamma_{i+ak}, & \text{falls } \max_{u \in U_i} h_{u+ak}^x < h_{i+ak}^t \wedge \gamma_{i+ak} \geq \rho \\ 0, & \text{sonst} \end{cases} \quad (10)$$

mit dem Histogrammunterschied $\gamma_{i+ak} := h_{i+ak}^t - h_{i+ak}^x$, der wertet, wie viel in h_*^t von h_*^x unerreicht bleibt. Er wird in der Formel jedoch nicht gewertet, wenn er sich rein durch Bildrauschen erklären lässt, d.h. unter einer Rauschschwelle $\rho := 3$ bleibt. Durch die Farbreduktion können zudem bei Rauscheinflüssen an sich gleiche Bildstrukturen leicht in unterschiedlichen (aber benachbarten) Bins von h_* landen. Daher wird der Histogrammunterschied γ_{i+ak} in der Formel nur gewertet, wenn auch die Nachbarn h_{u+ak}^x innerhalb der betreffenden Teilhistogramme den Mangel bei h_{i+ak}^x nicht kompensieren können. Bei Farbbildern summiert β auch über die drei Farbkanäle.

Die *Struktur-Metrik* wird definiert als $\delta := \beta \cdot m / \sum_{i=0}^{j-1} h_i^t$. x wird als *kompatibel zum Modell* h_*^t angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist. In dieser Arbeit wird $c = 30\%$ verwendet.

4.3 Bildsegmentierung

Die Bildsegmentierung erfolgt wie beim AR-Filter, siehe Abschnitt 2.3.

5 Ergebnisse

Abb 1 zeigt oben die verwendeten Originalbilder und darunter für $r_A = 10$ die erhaltenen Ergebnisse: erst zum AR-Filter ($r_B = 8$), darunter die Varianten 1 und 2 zur Schablonenanpassung und ganz unten zum Histogrammansatz. Wie man erkennen kann, eignen sich prinzipiell alle Verfahren/Varianten: insbesondere Wege, Bewuchs und Vegetationsunterschiede werden gut herauspräpariert.

Die beiden AR-Filter-Ergebnisse wirken unruhiger als die anderen, der AR-Filter wirkt also feingranularer. Die beiden Template-Varianten unterscheiden sich nicht substantiell, Variante 2 vermag vorteilhafterweise jedoch deutlich mehr und dichter zu segmentieren. Interessant ist, dass alle Verfahren/Varianten beim rechten Bild Übergänge im unteren linken Bereich feststellen, die man mit bloßem Auge nicht ohne Weiteres erkennen kann. Der Histogramm-Ansatz ist klar am besten: Er ist der einzige, der die Baumkronen geschlossen erkennt und sauber segmentiert. Auch der Pkw-Parkbereich im rechten Teil des linken Bildes wird am besten erfasst. Er segmentiert zudem die meisten Bildpixel.

Abschließend illustriert Abb. 2 das Verhalten des AR-Filter-Ansatzes bei unterschiedlichen Werten für r_B (von oben nach unten 2, 4, 6, 10 und 11) und Abb. 3 das des Histogramm-Ansatzes bei unterschiedlichen Werten für r_A (von oben nach unten 3, 5, 7, 9 und 12).

6 Zusammenfassung

Es wurden drei Ansätze zur Texturmodellierung bzw. Bildsegmentierung vorgestellt und verglichen. Alle zeigen sich prinzipiell als geeignet, jedoch liefert der Histogrammansatz klar die besten Ergebnisse. Er zeigt sich als vielversprechend für weitere geplante Optimierungen und Weiterverarbeitungsmöglichkeiten.

Literatur

1. A. Ruckstuhl, "Einführung in die robusten Schätzmethoden," Juni 2008.
2. F. Wicki, "Robuste Schätzverfahren für die Parameterschätzung in geodätischen Netzen," *Dissertation an der Eidgenössischen Hochschule (ETH) Zürich, Schweiz*, 1998.

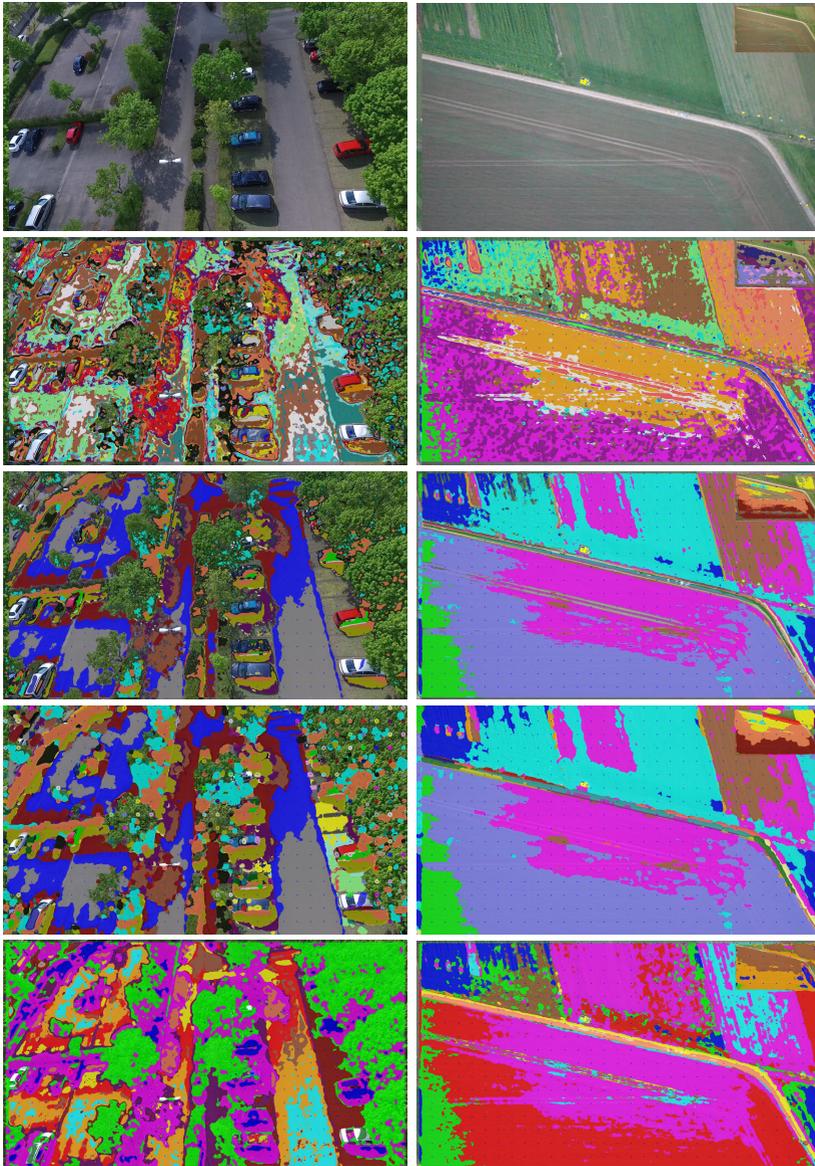


Abbildung 1: Testbilder (oben) und Ergebnisse für $r_A = 10$ (siehe Text).

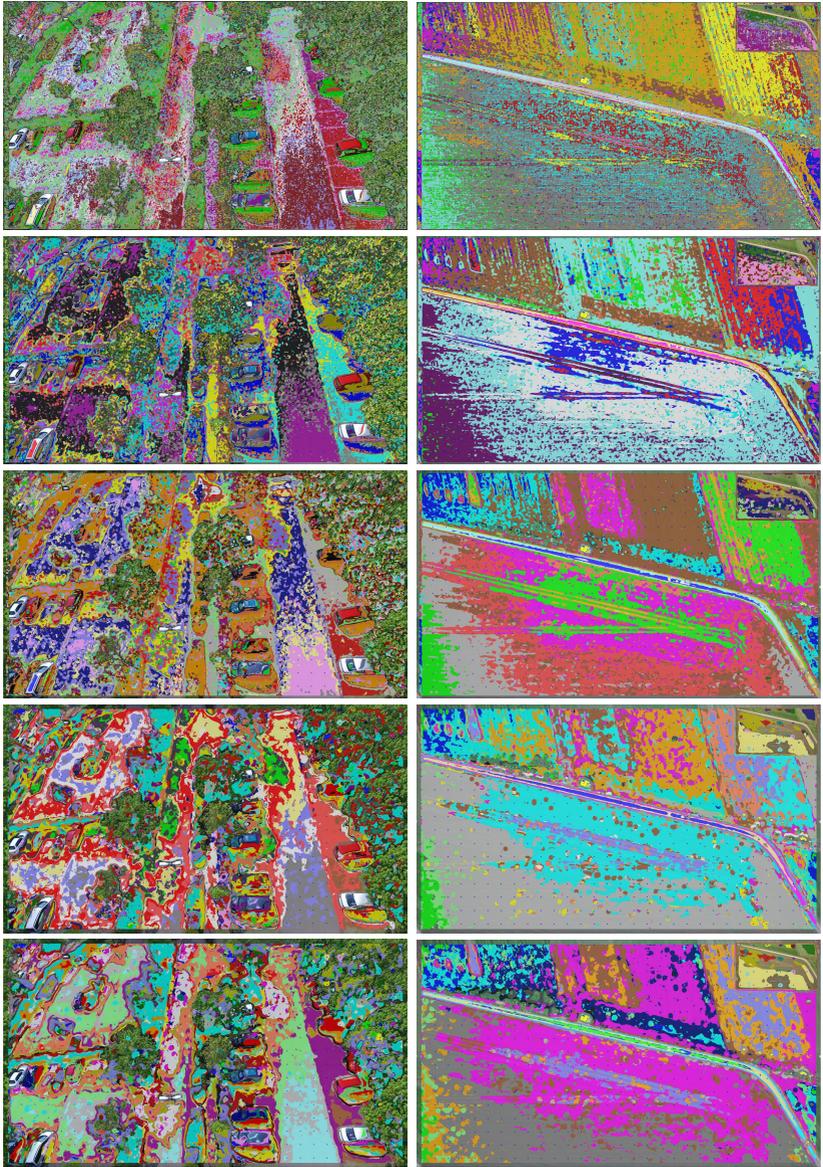


Abbildung 2: Ergebnisse zum AR-Filter für verschiedene r_B .

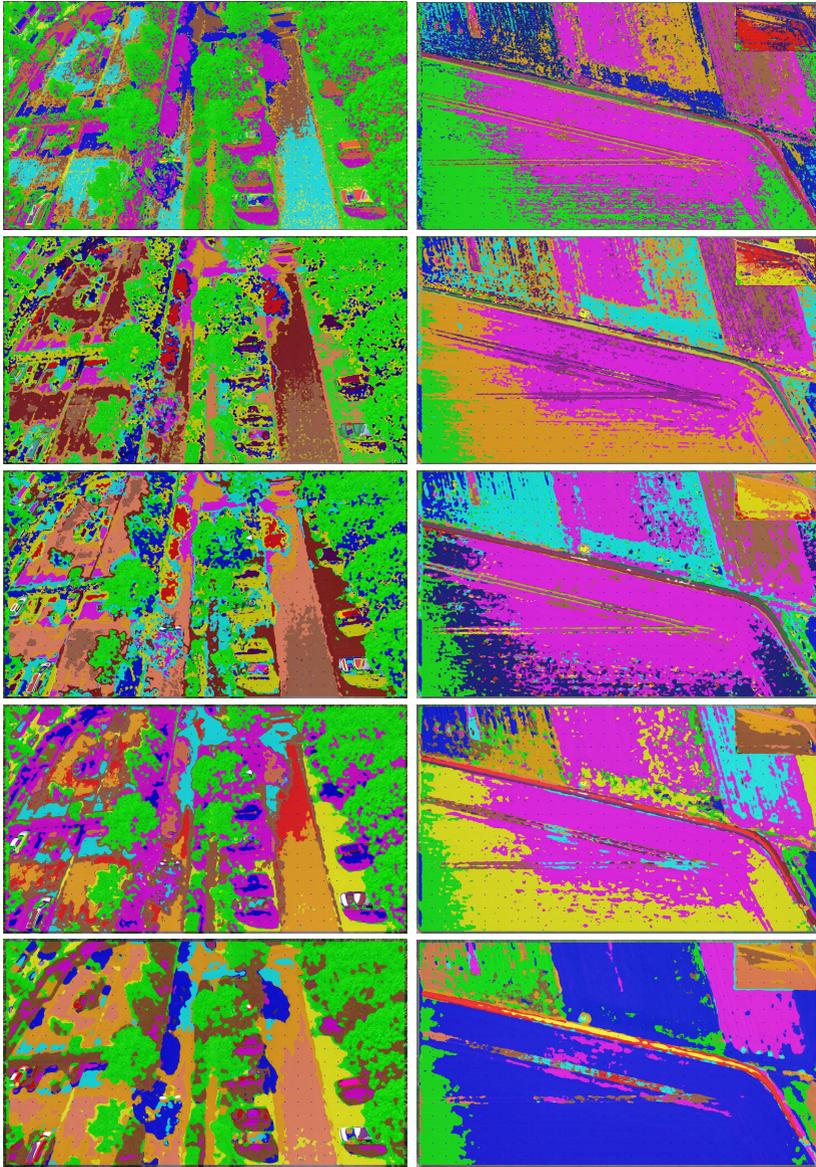


Abbildung 3: Ergebnisse zum Histogramm-Ansatz für verschiedene r_A .