

---

# FSC - FloodFill Sky Classification

Kirill Bogomasov  
Heinrich-Heine-Universität Düsseldorf  
Institut für Informatik  
Universitätsstraße 1  
40225 Düsseldorf, Deutschland  
kyryl.bogomazov@uni-duesseldorf.de

## ABSTRACT

Diese Arbeit behandelt die Klassifizierung des Himmels in Bergbildern. Zu Beginn wird ein Verfahren zur Abschätzung des Himmels in einem Bild vorgestellt. Im Anschluss folgt die Vorstellung eines auf FloodFill-basierenden Klassifikators. Abschließend werden die Ergebnisse verschiedener Klassifikatoren vorgestellt. Eine Klassifizierung des Himmels dient der Vorbereitung für eine spätere Segmentierung der Bergsilhouette. Dabei soll jeder Himmelsklasse ein passendes Segmentierungsverfahren zugeordnet werden. Zu diesem Zweck wird ein neuer Himmelsklassifikator vorgestellt, welcher zwischen vier Himmelskategorien unterscheidet und viel versprechende Ergebnisse liefert.

## Categories and Subject Descriptors

I.4 [Image Processing And Computer Vision]: Feature Measurement, Scene Analysis

## Keywords

Classification, Sky Recognition, Silhouette Extraction

## 1. EINLEITUNG

In den letzten Jahrzehnten gewinnt der digitale Datenaustausch immer weiter an Bedeutung. Ein großer Anteil daraus, sind Bilddaten. Durch den kontinuierlichen Anstieg, der im Internet kursierenden Bilder, insbesondere in sozialen Netzwerken, ergibt sich die Notwendigkeit der Bildklassifizierung. Einen wichtigen Teilbereich stellt die Bergerkennung dar, welche die Segmentierung der Bergsilhouette voraussetzt. Dieses Forschungsschwerpunkt soll im Rahmen dieser Arbeit aufgegriffen werden.

Es existiert bereits eine Vielzahl etablierter Segmentierungsalgorithmen, darunter auch mehrere speziell auf Bergregionen angepasste Verfahren. Im Hinblick auf das Ziel, zu jeder Himmelsklasse ein passendes Segmentierungsverfahren zu finden, muss dieser zuerst klassifiziert werden. Diese Fragestellung ist darauf zurückzuführen,

dass der Himmel sehr vielfältig, und seine Eigenschaften sehr variationsreich sein können. Verschiedene Himmelsarten können verschiedene Artefakte bergen. Diese verursachen maßgebliche Schwierigkeiten in der Verarbeitung. Infolge dessen entstehen unterschiedliche Anforderungsprofile an einen Segmentierungsalgorithmus. Es ist plausibel, dass ein einzelnes Segmentierungsverfahren den Anforderungen kaum gerecht werden kann. Folglich entsteht die Frage, ob sich ein Verfahren finden lässt, welches für eine oder mehrere Himmelskategorien besonders gute Ergebnisse erzielt. Für die Wahl eines optimalen Segmentierungsverfahrens soll untersucht werden, ob eine Klassifizierung des Himmels möglich ist. Zu diesem Zweck wird ein eigenes Verfahren vorgestellt.

## 1.1 Verwandte Verfahren

Anna Heinle arbeitet in Paper [8] mit einem Set hauptsächlich statistischer Features, die sowohl die Farbe als auch die Struktur beschreiben. Zur Klassifizierung wird ein k-NN-Klassifikator benutzt. Dem Modell liegt eine Trainingsmenge von 200 Bildern je Trainingsklasse zugrunde. Die Grundannahme der Wolkenerkennung beruht auf einem geringeren Rotwert innerhalb der Bereiche des klaren Himmels. Yu Changhui [5] stellt in seiner Arbeit ein weiteres Verfahren vor, um die Trennung von bewölkten Bildern und solchen, die keine Wolken enthalten, sicherzustellen. Er verwendet eine Zusammensetzung aus Grauwert-, Frequenz- und Strukturfeatures. Einen weiteren interessanten Ansatz liefert Nurak Chairri in [4]. Dieser nutzt das "Multilayer Perceptron neural network (MLP-NN)". Gearbeitet wird mit 18 charakteristischen Parametern für Farb-, Struktur- und Power Spectrum Features. Anschließend kann der Klassifikator entscheiden, ob die Wolken zu Niederschlag neigen oder nicht. Auch [6] befasst sich mit Wolkenklassifikation. Die Idee liegt hier in der Benutzung von Textons. Das Texton-Lexikon wird aus k-Means Cluster Zentren erzeugt. Die Bilder der Deskriptoren werden über S-Filter erzeugt. Anschließend werden auf der Basis eines Vergleichs jeden Pixels mit der Texton-Bibliothek "discriminative" Histogramm-Modelle für jede Kategorie erzeugt. Ein weiterer Ansatz zur Wolkenklassifikation findet sich in [3]. Dieser basiert ebenfalls auf statistischen Messungen und wird durch Fourier Transformation ergänzt. Aus der Analyse der Daten nach der Transformation wurde eine Mustererkennung erhofft. Jedoch wurde der Ansatz wieder verworfen. Dies liegt hauptsächlich daran, dass aufgrund der hohen Variation im Bereich des Himmels die Transformation keine nennenswerte Erkenntnis aus den Daten liefert. Auch funktioniert die Kantenerkennung nicht optimal. Im [9] beschreibt Shuang

28<sup>th</sup> GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 24.05.2016 - 27.05.2016, Göttingen, Germany.  
Copyright is held by the author/owner(s).

Liu ein zweistufiges Verfahren, welches die wenig funktionale Fourier Transformation durch LBP Histogramme ersetzt. Diese sind rotationsbeständig und werden für jedes Bild aus jeder Klasse erzeugt, die Wolken enthält. Für die Auswertung wird eine Grauwert-Konvertierung vorgenommen.

Eine wichtige Beobachtung aus den betrachteten Verfahren ist, dass die präziseren Ergebnisse auf Farb- und Strukturfeatures zurückgehen. Bei den Farbfeatures wird grundsätzlich über den Vergleich der verschiedenen Farbkanaäle gearbeitet. Die Strukturfeatures lassen sich auf die Arbeit [7] zurückführen. Die Einschränkung bei allen diesen Verfahren liegt in der Benutzung von sogenannten "whole sky images". Dabei handelt es sich um Bilder, die ausschließlich Himmel enthalten. Dadurch wird die Abgrenzung des untersuchten Bereiches erspart und die Herausforderung deutlich reduziert. Weiterhin findet in den vorgestellten Werken hauptsächlich eine Trennung in bewölkt und unbewölkt statt, jedoch keine echte Klassifizierung des Himmels. Dies ist daran zu erkennen, dass beispielsweise der Fall "Sonne" bei keinem Verfahren vorgestellt ist.

## 1.2 Bildersammlung und Datengrundlage

Die genutzten Bilder sind Grundlage des von Georges Baatz, Olivier Saurer, Kevin Köser und Marc Pollefeys unter dem Titel "Large Scale Visual Geo-Localization of Images in Mountainous Terrain" veröffentlichten Papers [1].

Zu Beginn werden die unsortierten Bilder in Kategorien eingeteilt. Dazu wurden drei Kategorien gewählt. Diese sind "klarer Himmel", "sonnig" und "bewölkt", sie sind essentiell. Die Kategorie "sonnig" definiert Bilder, auf denen entweder die komplette Sonne oder ein Teil davon abgebildet ist. Diese kann von Bergen verdeckt oder am Rande des Bildes vorzufinden sein. Dadurch ist sie oftmals nicht vollständig. Auch Fotos, auf denen besonders helle Sonnenstrahlen festgehalten sind und die Sonne hinter dem Bildrand oder dem Berg vermuten lassen, besitzen ähnliche Eigenschaften bezüglich der Bildhelligkeit und der Farben des Himmels. Somit gehören sie auch der Klasse "sonnig" an. Die Klasse "bewölkt" enthält alle Bilder, in denen eine Bewölkung vorliegt. Die Intensität und die Dichte spielen im Gegensatz zur Anfangsannahme keine Rolle mehr. Die Klasse "klarer Himmel" enthält die restlichen Bilder der Datenbank. Sie umfasst alle Bilder, in denen weder Sonne noch Wolken abgebildet sind. Durch das strikte Kategorisieren der Bilder entsteht das Problem der mehrfachen Zuordnung in die einzelnen Klassen. So können auf einem Bild sowohl Sonne als auch Wolken zu sehen sein. Da dieser Fall bei einer nicht vernachlässigbaren Anzahl von Bildern vorzufinden ist, wird eine weitere Klasse definiert - "bewölkt und sonnig".

Für weitere Bilder wurden mehrere Online Bilderdatenbanken nach lizenzfreien Bildern durchsucht. Die meisten der gefundenen Bilder stammen von [www.flickr.com](http://www.flickr.com). Die Auflösung der zusätzlichen Aufnahmen wurde nach Möglichkeit ähnlich der Auflösung der bereits vorhandenen Objekte gewählt. In anderen Fällen wurden die Bilder einem Schnitt unterzogen. Zu allen neuen Bildern wurde vom Lehrstuhl mithilfe eines bereits fertigen interaktiven Verfahrens eine Binärmaske erzeugt.

Insgesamt umfasst die Datenbank 43 Bilder der Klasse "bewölkt", 45 Bilder in der Klasse "bewölkt und sonnig", 40 Bilder in der Klasse "sonnig" und 124 Bilder in der Klasse

"klarer Himmel". Die Anzahl der Bilder in der Klasse "klarer Himmel" wurde nicht reduziert, weil auf verfügbare Informationen nicht verzichtet werden sollte.

## 1.3 Abschätzung des Himmels

Eine Klassifizierung des Wetters erfordert eine möglichst treffende Abschätzung des Himmels. Dabei ist es wichtig mit der Abschätzung möglichst nah an die Grenze zwischen Berg und Himmel zugelangen. Es ist nicht ungewöhnlich, dass sich Wolken in einer geringen Höhe oberhalb der Berge befinden. Auch Sonnenauf- und Sonnenuntergänge sind ein häufiges Bildmotiv. Eine zu geringe Fläche führt zu einem potentiellen Klassifikationsfehler. Eine zu großgeschätzte Fläche führt zu neuen Problemen. In diesem Fall werden Bergspitzen mit ausgewertet. Dies kann dazu führen, dass Berge wolkenähnliche Eigenschaften besitzen und als solche gedeutet werden. In Folge dessen würde eine Missklassifizierung begünstigt werden. An diesen Fällen wird deutlich, dass eine ungenaue Abschätzung der Ränder maßgebliche Folgen für die Klassifizierung haben kann.

Es gibt unterschiedliche Möglichkeiten, eine Abschätzung vorzunehmen. Eine naive Abschätzung stellt zum Beispiel den einfachsten Weg dar. Hierbei wird eine Boundary Box in Abhängigkeit von der Bildgröße aufgestellt. Die Problematik die dadurch entsteht liegt darin, dass die Höhe des Horizonts sehr stark variieren kann und deshalb eine Abschätzung mittels eines festen Wertes sehr schwierig ist. Für eine solche Abschätzung kommt nur ein Segmentierungsverfahren in Frage. Am besten eignen sich hier Methoden, die iterativ das Bild "fluten".

Aus diesem Grund fällt die Entscheidung auf FloodFill. Der Algorithmus findet Flächen, die dieselbe Farbe besitzen, und ersetzt diese durch eine andere. Zu Beginn des Programmablaufs muss ein Pixel als Startpunkt definiert werden. Ein weiterer Pflichtparameter ist die Zielfarbe. Ist die Farbe ersetzt worden, wird die Nachbarschaft des jeweiligen Pixels betrachtet. Der Aufruf erfolgt rekursiv. Es kann die Vierer-(engl. 4-Neighbour) oder die Achter-Nachbarschaft (engl. 8-Neighbour) betrachtet werden. Für die Laufzeitoptimierung des Algorithmus sowie zum Zwecke der Ausreißerminimierung werden nur Vierer-Nachbarschaften (engl. 4-Neighbour) betrachtet.

Um den Algorithmus effektiver zu gestalten, kann die akzeptierte Farbe mit Hilfe einer tolerierten Mindestbeziehungsweise Höchstabweichung erweitert werden. In Bergbildern hat sich eine Abweichung von 5 Helligkeitsstufen als sinnvoll erwiesen.

Bei der betrachteten Problemstellung besteht die Hoffnung, dass mit FloodFill homogene Bildelemente im Bereich des Himmels zu Regionen verschmolzen werden. Als Ausgangspunkt dient der Punkt mit der höchsten Wahrscheinlichkeit, ein Himmelspunkt zu sein. Die gesamte oberste Pixelreihe des Bildes besitzt die höchste Wahrscheinlichkeit dem Himmel anzugehören. Es ist anzunehmen, dass auf keinem Bild in der obersten Reihe ein Berg vorzufinden ist. Insbesondere eignet sich FloodFill, um umrandete oder angrenzende Flächen auszufüllen. Im Idealfall stellt die Grenze zwischen dem Berg und dem Himmel den Rand dar. Als Resultat entsteht eine Segmentierungsmaske. [2]

Das Resultat einer Abschätzung sollte ein Polygon sein, welches im Idealfall die Silhouette des Berges beschreibt. Zu diesem Zweck wurde ein eigenes Verfahren zur Himmelsabschätzung entwickelt.

## 2. FSE - FLOODFILL SKY ESTIMATION

Zu Beginn des Algorithmus wird FloodFill ausgeführt. Dabei wird das Verfahren in zwei der drei möglichen Varianten (Color, Grey-Mask) aufgerufen. Der Grund für einen mehrfachen Aufruf ist, dass jede Methode in verschiedenen Helligkeitsverhältnissen ihre Stärken aufweist. So hat das Konzept, welches Farbkanäle vorsieht, seine Vorzüge bei klarem blauem Himmel, sowie bei so gut wie allen Wolken-typen. Einzig bei tiefen und konturlosen Wolken schneidet es schlecht ab. Mit der Grey-Mask-Variante hingegen funktioniert das Verfahren recht gut bei nebeligen Bildern. Auch bei sonnigen Bildern schneidet es deutlich besser ab. Somit scheint es sinnvoll zu sein, beide Verfahren zu kombinieren. Nachdem als Resultat zwei verschiedene Bilder erzeugt wurden müssen die Ergebnisse analysiert werden. Die Analyse dient dazu, das Bild mit den wenigsten Fehlern zu finden. Zu Beginn wird das Resultat des Farbbildes analysiert. Der Analyse liegen drei Aspekte zugrunde. Es wird untersucht, ob die entstandene Fläche brüchig ist, so wie dies beim FloodFill-Resultat mit Farbe der Fall ist. Außerdem wird geprüft, ob Bildpunkte im Bereich des Bodens überschrieben wurden. Wenn dies der Fall ist, so ist FloodFill bis zum Boden durchgelaufen. In der Regel sind auf den Bildern jedoch Berge zu sehen, die die gesamte Breite des Bildes einnehmen. Läuft FloodFill von oben, wo der Startpunkt liegt, bis unten durch, so handelt es sich höchstwahrscheinlich um einen Fehler.

Des Weiteren wird der Anteil der gefundenen Himmelsfläche am Gesamtbild geprüft. Ist dieser zu groß, kann das Resultat verworfen werden. Als Grenze dient ein Schwellenwert von 80%, welcher im Rahmen der Untersuchung entstand. Dieser Schwellenwert ist darauf zurückzuführen, dass eine Abbildung von weniger als 20 % des Berges auf der Bildfläche unüblich ist.

Werden die drei Bedingungen erfüllt, kann mit dem Bild weiter gearbeitet werden. Andernfalls muss es verworfen und mit der nächsten FloodFill-Variante, der "Grey Mask", gearbeitet werden. In beiden Fällen wird im nächsten Schritt FloodFill von unten nach oben durchlaufen.

Anschließend wird das Resultat aufgearbeitet. Die leeren inneren Flächen, die vom Berg umschlossen sind werden, aufgefüllt. Dies ist deutlich in den Abbildungen 1 und 2 zu erkennen.

Gehäuft treten Fehler im Bereich der Ränder auf. Um das Problem zu beseitigen, werden vor der Polygon-Extraktion aus dem Resultat die Ränder bereinigt. Es hat sich herausgestellt, dass die besten Ergebnisse zustande kommen, wenn im oberen Bereich des Bildes die oberste Pixelreihe nachgebessert wird. An den seitlichen Rändern werden jeweils die äußersten Spalten überschrieben. Im Laufe der Bereinigung werden alle Randpixel, die nach der Segmentierung zum Berg gehören, dem Himmel zugeordnet. Die Höhe der neuen seitlichen Randspalten des Berges richtet sich nach der Höhe der zweiten, beziehungsweise der vorletzten Spalte des Bildrasters. Anschließend findet ein erneuter FloodFill-Aufruf statt. Bei diesem werden die fehlerhaften Flächen innerhalb des Himmels nicht mehr mit berücksichtigt. Diese Abschätzung wird im folgenden Klassifizierungsverfahren genutzt.

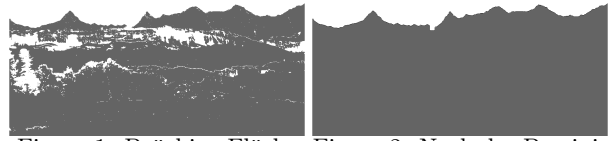


Figure 1: Brüchige Fläche Figure 2: Nach der Bereinigung

## 3. FSC - FLOODFILL SKY CLASSIFICATION

Bei der FSC-Klassifizierung handelt es sich um einen Klassifikator, welcher nicht erlernt werden muss und sofort mit der Klassifikation beginnen kann. Zu Beginn erfolgt für jedes Bild der Testmenge eine Abschätzung des Himmels mit dem Verfahren nach Kapitel 2. Sobald der Bereich des Himmels mit dem Silhouette-Polygon definiert ist, kann mit der Analyse des Himmels begonnen werden. Dem Algorithmus liegt die "FloodFill"-Technik zugrunde. Es wird mit der "Graumask" gearbeitet. Zu Beginn des Ablaufs erfolgt eine statistische Analyse der Grauwerte. An dieser Stelle muss geprüft werden, wie hoch der Anteil der schwarzen Pixel am gesamten Bild ist. Liegt ihr Anteil bei über 95 %, so ist davon auszugehen, dass das resultierende Bild sehr verrauscht ist. Meistens liegt die Ursache in einer Unschärfe, die durch Nebel oder Sonne erzeugt wird. Wird das Bild als verrauscht erkannt, muss eine Sonderbehandlung erfolgen. Es folgt eine Optimierung der FloodFill-Parameter, ähnlich wie bereits während der Abschätzung.

Im Anschluss findet eine erneute Überprüfung der Pixelhelligkeit statt. Ist das Bild noch immer sehr verrauscht und überschreitet den festgelegten Grenzwert, wird es in eine künstliche Klasse "bright/low contrast" abgelegt. Diese Klasse stellt einen Sonderfall dar und kann bisher nicht anders behandelt werden. Jedoch ist der Anteil solcher Bilder relativ gering und liegt bei den vorhandenen Aufnahmen bei unter 1 %. Erfüllt das Bild alle Kriterien, so beginnt die Klassifizierung. Das Verfahren lässt sich in zwei Schritte untergliedern.

Im ersten findet eine Wolkensuche statt. Im zweiten wird nach der Sonne gesucht. Die Wolkensuche erfolgt nach dem Algorithmus 1

### 3.1 Suche von Wolken

Das Verfahren trifft für jeden Pixel die Entscheidung, ob dieser Bestandteil einer Wolke ist oder nicht.

Ein Pixel wird nur dann betrachtet, wenn dieser oberhalb der Silhouette liegt. Nur dann gehört er zum Himmel. Die Bestimmung, ob ein Pixel als "Wolke" klassifiziert werden kann oder nicht, beruht auf dem Vergleich der Intensitäten  $\alpha$  von Farbkanälen wie in [8].

$$\alpha = \frac{\text{red\_channel\_value}}{\text{blue\_channel\_value}}$$

Liegt der Wert  $\alpha$  bei 0.9 oder höher, ist der Pixel als Wolke zu klassifizieren. Der Wert, der bereits von A.Heinle in ihrem Paper vorgeschlagen wurde, wurde während einer empirischen Untersuchung bestätigt. Dabei wurden  $\alpha$ -Werte im Intervall  $[0, 1]$  betrachtet.

---

**Algorithm 1:** Die Suche nach Wolken in Bildern

---

```

1 findTheClouds (mountSil, picPath);
   Input : mountSil - polygon that describes the
           mountain silhouette, picPath - path to the
           picture to be analyzed
   Output: true if clouds are found on the picture, false
           otherwise
2 read in the picture from the given path;
3 counterSkyPixel = 0;
4 counterCloudyPixel = 0;
5 foreach Pixel do
6   | if pixel belongs to sky then
7     | counterSkyPixel++;
8     | if pixelIsCloudy then
9       | countCloudyPixel++;
10    | end
11  | end
12 end
13 if counterCloudyPixel/counterSkyPixel > 0.2 then
14 | return true;
15 else
16 | return false;
17 end

```

---

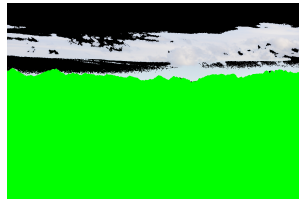


Figure 3: Originalbild      Figure 4: Wolkenuche

Das Bild 4 zeigt das Endergebnis. Die grüne Fläche ersetzt dabei den Berg nach der Segmentierungsmaske. Die schwarze Fläche im Bereich des Himmels beschreibt die Pixel, die als nicht bewölkt eingestuft wurden.

Im nächsten Schritt kann die Suche der Sonne erfolgen.

### 3.2 Suche der Sonne

Die Sonne ist ein wichtiges Attribut des Wetters. Aus diesem Grund wurde für die Bilder mit sonnigem Wetter eine eigene Klasse erzeugt. Dieser Klasse wurden Bilder zugeordnet, auf denen entweder die Sonne zu sehen ist oder besondere helle Stellen im Bereich des Himmels, die die Sonne vermuten lassen. Hierbei ist es nicht wichtig, zwischen den Klassen "sonnig" und "sonnig und bewölkt" zu unterscheiden, da die Sonne in jedem Bild zu Beginn der Analyse gesucht wird. Der gegebene Pseudocode 2 schafft Einblick in das entstandene Verfahren. In diesem Algorithmus werden Eigenschaften einer möglichen Sonne untersucht. Dabei wird zu Beginn die größte zusammenhängende Fläche der hellen Pixel in einem Bild gesucht. Um die Helligkeit einzustufen, wird die folgende Formel verwendet:

$$\text{luminance} = \frac{r * 0.2126 + g * 0.7152 + b * 0.0722}{255}$$

wobei  $r, g, b$  Farbkanäle des RGB-Modells sind.

Der Grenzwert von  $\delta = 0.99$  für die Lichtintensität ist das Resultat der Auswertung der Ergebnisse der Sonnensuche in allen vorliegenden Bildern. Dabei wurden Bilder

---

**Algorithm 2:** Die Suche der Sonne in Bildern

---

```

1 findTheSun (mountSil, picPath);
   Input : mountSil - polygon that describes the
           mountain silhouette, picPath - path to the
           picture to be analyzed
   Output: true if the sun is found on the picture, false
           otherwise
2 read in the picture from the given path;
3 imageTempSunClass := copy of the original image held
   in memory;
4 coherentAreaList := empty list of pixel sets;
5 sunFound := false;
6 foreach pixel do
7   | if pixel belongs to sky then
8     | if pixel is bright enough then
9       | // eventually it belongs to the sun
10      | add pixel to coherentAreaList;
11      | set pixel color to bright green in
12      | imageTempSunClass;
13    | end
14  | end
15 potentialSun := biggest coherent area in
   coherentAreaList;
16 countSunPixels := number of pixels in potentialSun;
17 // consider area as sun if it has min. 30 pixels
18 if countSunPixels > 30 then
19 | if sun is round then
20 | | sunFound := true;
21 | else if sun is at the picture border then
22 | | sunFound := true;
23 | else if sun is not at picture border then
24 | | if sun is covered by a mountain then
25 | | | sunFound := true;
26 | | end
27 end
28 write imageTempSunClass to new file;
29 return sunFound;

```

---

berücksichtigt, die als "sonnig" klassifiziert wurden. Wird ein kleineres  $\delta$  aus dem Intervall  $[0,1]$  gewählt, so werden andere Objekte wie zum Beispiel weiße Wolken, als "Sonne" falsch klassifiziert. Wird ein größeres  $\delta$  gewählt, so entsteht das Problem, dass so gut wie auf keinem Bild die dargestellte Sonne die vorgegebene Helligkeit erreicht. Somit erweist sich der angegebene Wert als optimal. Die Abbildung 5 zeigt, wie die Spiegelung der Sonne im Wasser als "sonnig" erkannt wird. Wird der Grenzwert auf 0.99 erhöht, behebt dies das Problem. (Siehe Abbildung 6) Wird bei demselben Bild der Wert noch weiter erhöht, so erfüllt kein einziger Pixel die Voraussetzung und die Sonne wird nicht mehr erkannt. Erfüllt ein Pixel diese Bedingung, so wird er zu einer oder zwei Listen mit sonnigen Pixeln hinzugefügt. Jede Liste enthält dabei nur benachbarte Pixel. Werden zwei Nachbarn in verschiedenen Listen gefunden, müssen diese zu einer neuen Liste verschmolzen werden, bevor der neue Knoten hinzugefügt werden kann. Betrachtet wird die Vier-Nachbarschaft. Wie sich in den ersten experimentellen Versuchen gezeigt hat, stellt die Grenze von 30 Objekten eine gute Schranke dar, um das Rauschen zu minimieren. Enthält die größte Liste weniger als 30 Elemente, kann davon

ausgegangen werden, dass es sich bei den hellen Pixeln um Rauschen handelt. Dieses Rauschen kann viele Ursachen haben. Einige davon sind: eine Reflexion des Kamerablitzes, eine Stromleitung oder helle Bergpixel, die aufgrund einer Fehleinschätzung dem Himmel zugeordnet werden. Sind mehr als 30 Pixel gefunden worden, so muss die Menge eine der folgenden drei Eigenschaften erfüllen: runde Form, Position an der Grenze des Bildes oder vom Berg teilweise verdeckt.

Die erste Eigenschaft scheint selbsterklärend zu sein. Die Form einer unter idealen Wetterverhältnissen aufgenommenen Sonne ist rund. Leider entspricht diese im überwiegenden Teil der Aufnahmen keiner Kreisform. Auch abgesehen von den in der Datenbank vorliegenden Bildern, stellt die Aufnahme der Sonne eine schwierige Angelegenheit dar. Diese setzt eine besondere Ausstattung voraus. In den meisten Fällen werden Bilder jedoch mit einfachen Mitteln erzeugt, sodass die Qualität häufig nicht optimal ist.

Selbst bei Bildern, in denen eine vollständige kreisrunde Sonne abgebildet ist, ist die Helligkeit der Pixel in der Mitte stets höher als am Rand. Es kann also nicht davon ausgegangen werden, dass alle Randpixel gleichermaßen in die Menge der hellen Pixel aufgenommen werden. Da die Form trotzdem analysiert werden soll, empfiehlt es sich, eine gewisse Varianz einzuführen. So kann beispielsweise eine ovale Sonne als rund akzeptiert werden.

Der Untersuchungsansatz basiert auf der Annahme, dass der Abstand der äußersten Punkte der Sonne zum Mittelpunkt in etwa gleich groß sein muss. Um dies zu prüfen, muss zu Beginn der Mittelpunkt der größten zusammenhängenden Fläche der hellen Pixel berechnet werden. Der Berechnung des Mittelpunktes liegt die Durchschnittsberechnung zugrunde. Hierbei werden die Koordinaten aller Punkte je Dimension aufsummiert und durch die Anzahl der betrachteten Punkte dividiert. Um die erste Distanz zu berechnen, muss zuerst die Entscheidung getroffen werden, welche Punkte verglichen werden sollen. Theoretisch ist es möglich, jeden einzelnen Randpunkt der potenziellen Sonne zu betrachten. Dieser Ansatz ist zwar genau, jedoch sehr zeitintensiv. Eine Verbesserung der Idee liegt darin, Repräsentanten der Randpunkte zu bestimmen. Diese müssen alle im gleichen Abstand gewählt werden.

Anschließend wird der Abstand zwischen dem Mittelpunkt und einem Repräsentanten berechnet. Hierfür dient das Ziffernblatt einer Uhr als anschauliches Beispiel. Ähnlich wie bei einer Uhr werden zwölf Punkte mit der gleichen Winkelweite ausgewählt. Zu diesem Zweck wird die Menge aller hellen Pixelpunkte an ihrem Mittelpunkt gedreht. Der Rotationswinkel liegt stets bei  $30^\circ$ , so dass nach 12 Umdrehungen wieder die Ausgangsmenge erreicht wird. Auf diese Weise entstehen die Sonnenprofile. Von jedem einzelnen Sonnenprofil kann nun der Abstand zwischen dem Mittelpunkt und dem äußersten Punkt orthogonal zur x-Achse berechnet werden.

Die Rotationsmatrix hat dabei die folgende Form:

$$\text{newXValue} = c.X + (p.X - c.X) * \cos(\beta) - (p.Y - c.Y) * \sin(\beta)$$

$$\text{newYValue} = c.Y + (p.X - c.X) * \sin(\beta) + (p.Y - c.Y) * \cos(\beta)$$

dabei ist p der Punkt, c das Zentrum und  $\beta$  der Drehwinkel

Anschließend werden alle Distanzen mit einander verglichen. Dabei darf die kleinste Distanz nicht mehr als um 20% von der größten abweichen. Wird die zusammenhängende Fläche als rund erkannt, ist davon auszugehen, dass es sich bei der Fläche um die Sonne handelt und sich die restliche Laufzeit ersparen. Ist die Fläche jedoch nicht rund, so kann dies daran liegen, dass die Sonne am Bildrand liegt und stellenweise nicht komplett ins Bild mit aufgenommen wurde. Um dies zu prüfen, müssen die Koordinaten der äußersten Punkte der analysierten Fläche betrachtet werden. Auch hier wird eine gewisse Abweichung in der Helligkeit akzeptiert. Es ist ein bekanntes Problem der Fotografie, dass die Schärfe in den Bildern zum Rand hin etwas abschwächen kann. Aus diesem Grund scheint auch ein Abschwächen der Helligkeit plausibel zu sein. Bei der Untersuchung der Randpixel werden bis zu vier Pixel Abstand vom Rand toleriert. Liegt die Sonne nicht am Bildrand, so kann sich ihre Position am Bergrand befinden. In diesem Fall ist die Form der Sonne unberechenbar. Einerseits kann es sich um beliebig große Kreisteile handeln, andererseits auch um eine untergehende oder aufgehende Sonne handeln, die aufgrund der Sonnenstrahlen beliebige Formen annehmen kann. Da bereits davon auszugehen ist, dass die Fläche groß genug ist, reicht es auch hier, den Abstand zum Berg zu untersuchen. Diese Stelle ist besonders fehleranfällig. Sobald die Abschätzung des Berges von der tatsächlichen Bergsilhouette abweicht, liegt die Sonne entweder zu weit oder aber gar nicht im Bereich des Himmels. Um diese möglichen Abweichungen in die Berechnung mit einzubeziehen, wurde eine höhere Distanztoleranz angenommen. Um jedoch mögliche "false positives", die aufgrund der höheren Distanztoleranz entstehen, abzuschwächen, sucht der Algorithmus nach größeren Pixelflächen. Entgegen der Untersuchung der Randposition ist es hier also nicht ausreichend, einen einzelnen Pixel zu finden, der nah genug ist.

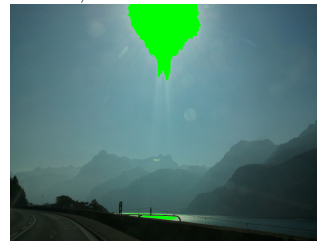


Figure 5:  $\delta = 0.7$

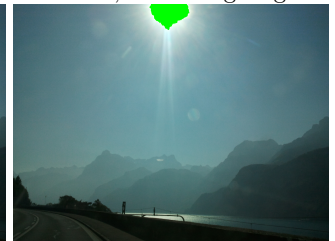


Figure 6:  $\delta = 0.99$

### 3.3 Zusammenfassung

Zum Schluss müssen die Ergebnisse ausgewertet werden. Sind sowohl Wolken als auch Sonne gefunden worden, ist das Bild als "sonnig und bewölkt" zu klassifizieren. Sind nur Wolken und keine Sonne gefunden worden, so kann auf die Klasse "bewölkt" geschlossen werden. Wurde ausschließlich die Sonne gefunden, ist von der Klasse "sonnig" auszugehen. Wurden im Himmel keine relevanten Objekte gefunden, so handelt es sich um "klaren Himmel".

Die Vielfalt der Möglichkeiten in den Bergbildern ist grenzenlos. Ein perfektes Klassifizierungs-Modell wird daher möglicherweise nie existieren.

## 4. AUSWERTUNG

Nachfolgend werden in einer abschließenden Betrachtung die wichtigsten Erkenntnisse zusammengefasst.

Bildklasse	Precision	Recall	F1-Measure
Sonnig	0.744	0.800	0.771
Bewölkt	0.853	0.674	0.753
Bewölkt und sonnig	0.875	0.457	0.600
Klarer Himmel	0.811	0.968	0.883

Table 1: FSC

Verfahren	SVM CSift	k-NN CSift	k-NN Spektr.	k-NN Text.u.Spekt.	FCS
F1-Measure	0.356	0.281	0.515	0.276	0.752

Table 2: Durschnitts F1 über alle Klassen

Im Laufe der Arbeit wurden verschiedene Optionen der Klassifizierung des Himmels erforscht. Dabei wurde eine Methode zur effektiven Abschätzung der Himmelsfläche entwickelt. Es wurden verschiedene Klassifikatoren implementiert und an unterschiedlichen Bildmerkmalen getestet. Eine Klassifizierung des Himmels mittels k-NN- und SVM-Klassifikatoren, anhand von CSIFT Deskriptoren, hat moderate Ergebnisse gezeigt. (Siehe Tab.2) Dabei wurde die Nachbarschaft im Bereich von [1-10] im Falle von k-NN betrachtet. Bei SVM wurde mit dem RBF-Kernel gearbeitet und den empfohlenen Standardparametern. Zwar wurden alle Deskriptoren mit CSIFT berechnet, jedoch sind nicht alle darunter invariant bezüglich Translation, Rotation und Skalierung, weil alle Merkmale im Bereich des Himmels nach einem Gitter-Prinzip extrahiert wurden. Eine Keypoint-Suche mittels des CSIFT bliebe für den Himmel beinahe erfolglos.

Deutlich bessere Ergebnisse zeigten sich während der Klassifizierung an Spektral-Merkmalen, insbesondere unter Einbeziehung von Sonnen- und Wolken-Features. Die Ergänzung der Features durch Textur-Merkmale hat keine Verbesserung der Evaluationsergebnisse erzeugt. Unter den Klassifikatoren hat, der in dieser Arbeit zum ersten Mal vorgestellte FSC-Algorithmus, die höchste Genauigkeit mit einem durchschnittlichen F1-Wert von 0.752 demonstriert. Einen tieferen Einblick zeigt die Tabelle 1.

## 5. AUSBLICK

Mit Hilfe des FSC-Klassifikators lässt sich der Himmel in Bildern kategorisieren. Im nächsten Schritt kann die Suche nach dem leistungsstärksten Segmentierungsalgorithmus für jede Himmelsklasse in Abhängigkeit von den Bildeigenschaften erfolgen. Wünschenswert bleibt eine größere Bildersammlung mit bereits vorliegenden Segmentierungsmasken. Bisher bleibt [1] die einzige bekannte freiverfügbare Bergbildersammlung. In der Zukunft gibt es die Option diese weiter zu erweitern. Außerdem besteht die Überlegung eine Vorverarbeitung mittels Kantenschärfung vorzunehmen und auf diesem Wege den Übergang zwischen der Bergebene und dem Himmel deutlicher hervorzuheben.

## 6. REFERENCES

[1] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *Computer Vision-ECCV 2012*, pages 517–530. Springer, 2012.

[2] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[3] J. Calbo and J. Sabburg. Feature extraction from whole-sky ground-based images for cloud-type recognition. *Journal of Atmospheric and Oceanic Technology*, 25(1):3–14, 2008.

[4] N. Chaisri, S. Vanichprapa, and R. Chaisricharoen. Cloud classification for weather application.

[5] Y. Changhui, Y. Yuan, M. Minjing, and Z. Menglu. Cloud detection method based on feature extraction in remote sensing images. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(1):173–177, 2013.

[6] S. Dev, Y. H. Lee, and S. Winkler. Categorization of cloud image patches using an improved texton-based approach. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 422–426. IEEE, 2015.

[7] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

[8] A. Heinle, A. Macke, and A. Srivastav. Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques*, 3(3):557–567, 2010.

[9] S. Liu and Z. Zhang. Learning discriminative salient lbp for cloud classification in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015, 2015.