Praxisprojekt

Automatisches Schätzen der Haaranzahl in ausgefallenen Haarbüscheln

Janelle Pfeifer
Delpstraße 28
53359 Rheinbach
janelle.pfeifer@smail.inf.h-brs.de

Hochschule Bonn-Rhein-Sieg
Institute of Visual Computing
Fachbereich Informatik
Studiengang: Informatik (B.SC.)

Rheinbach, 24.4.2020

Inhaltsverzeichnis

Praxisprojekt . . . 1
1 Einleitung . . . 3
2 Grundlagen . . . 3
2.1 Computervision . . . 3
2.2 Template Matching . . . 3
2.3 Canny Edge Detection . . . 3
2.4 Dilatation . . . 4
2.5 Inversion eines Graustufenbildes . . . 4
2.6 Watershed Algorithmus . . . 4
2.7 Connected Components . . . 4
2.8 Skelettierung . . . 5
2.9 Outlier Removal . . . 5
2.10 fitted curve, lineare Regression, Spline . . . 5
3 Design und Implementation . . . 5
3.1 Verarbeitung der Haar-Bilder . . . 5
3.2 Kalibrieren . . . 9
3.3 Relation Daten zu Haar Menge . . . 10
3.4 Schätzen eines Haar-Bildes . . . 10
3.5 Tests . . . 12
3.6 Test: Knielange, dunkelrote Haare . . . 12
3.7 Test: Hüftlange, feine, blonde Haare . . . 13
3.8 Test: kurze Hellbraune Haare . . . 14
3.9 Test: Mama . . . 14
4 Ergebnisse . . . 14
5 Fazit . . . 15
A Anhang: 28 Datenpunkte pro Bild . . . 16
B Anhang x Wert berechungen für Schätzungs-Modelle . . . 18
C Anhang Schätzungs Modelle . . . 20
D Anhang: Blondes Haar . . . 22
E Anhang Schätzungs Modelle Blondes Haar . . . 24

Zusammenfassung. am ende hier ein abstract hin

1 Einleitung

Das Ausfallen von Haarsträhnen ist ein natürlicher Teil des Haarwachstum-Zyklus. Durchschnittlich verliert der Mensch 50 bis 150 Haarsträhnen am Tag. Vermehrter Haarverlust kann ein Anzeichen von Krankheiten oder Mangelerscheinungen sein.[1]

Haarausfall ist sehr variabel und wird durch viele Faktoren beeinflusst. Er kann sich abhängig von dem Haartyp, der Haarlänge und der Haarpflege-Methode jeden Tag ändern und Muster aufweisen. Jeder Mensch hat ein eigenes Haarausfall-Muster. Eine Langzeitüberwachung gibt die Möglichkeit diese Muster zu erkennen und Abweichungen herauszustellen. [3, 4, 2]

In dem Paper „The Hair shedding visual scale: A quick tool to assess hair loss in Women“ wird eine Methode beschrieben, in der Frauen anhand von Bildern den Umfang ihres täglichen Haarausfalls bestimmen können. Dabei werden den Frauen Bilder von abgezählten, ausgefallenen Haarbüscheln gezeigt, die ihrer eigenen Haarlänge entsprechen. Die Frauen wählen das Foto aus, welches ihrem persönlichen täglichen Haarausfall entspricht. Dabei wurde eine Korrelation festgestellt zwischen Frauen, die klinisch bestätigt Haarverlust erfahren und den Bildern, die sie auswählen. Der Haarausfall kann somit visuell, anhand von Bildern festgestellt werden.[5]

In dieser Arbeit wird ein Verfahren vorgestellt, mit dem der tägliche Haarausfall mithilfe von Methoden der Bildverarbeitung und statistischer Auswertung geschätzt werden kann. So wird eine Langzeitüberwachung möglichst effizient durchführbar.

2 Grundlagen

2.1 Computervision

In dem Bereich der Bildverarbeitung gibt es viele Algorithmen, die es ermöglichen Objekte in einem Bild erkennbar zu machen und zu verarbeiten.

2.2 Template Matching

Template Matching findet Bereiche eines Bildes, die einem gegebenen Bildteil, einem Template, ähnlich sind.

Dazu wird ein Source Image und ein Template Image angegeben. Um die gesuchten Bereiche zu finden, wird das Template Image Pixelweise über das Source Image geschoben. An jeder Stelle wird das Template mit dem Source Image verglichen und die Ähnlichkeit zwischen den Beiden berechnet. Als Ergebnis erhält man eine Matrix die für jeden Pixel des Bildes eine Wahrscheinlichkeit dafür angibt, das sich dort das Template befindet.[6]

2.3 Canny Edge Detection

Canny Edge Detection ist ein Algorithmus der Kanten in einem Bild erkennt. Es handelt sich um einen Hochpassfilter. Diese suchen nach Stellen im Bild, wo sich die Farben bzw. Graustufen stark voneinander unterscheiden. Solche Stellen heißen

Hochfrequent. Linien und Kanten weise eine Hohe Frequenz auf. Niederfrequente Bereiche weisen eine einheitlichere Farbe oder Graustufe auf.[7]

Ein Hochpassfilter lässt hochfrequente Bildanteile stehen und schwächt Niederfrequente ab. Canny Edge Detection wird auf Graustufenbilder angewandt. An jeder Stelle des Bildes wird ein 3x3 Ausschnitt der Nachbarpixel betrachtet. Weisen die Nachbarpixel ein Bestimmtest Muster auf, indem der Graustufe-unterschied eine Linie beschreibt mit Helleren Pixel auf einer Seite und dunkleren Pixeln auf der anderen, wird die Stelle als Teil einer Kante betrachtet.[7]

2.4 Dilatation

Die Dilatation arbeitet auf binären Bildern. Es gibt nur schwarz und weiß. Die Dilatation fügt weiße Pixel an die Ränder von weißen Objekten hinzu. Bei der Dilatation handelt es sich um ein Morpholigschen Operator. Diese arbieten das Bild Pixelweise durch. Der Wert jedes Pixels basiert auf einem Vergleich zischen dem Pixel und dessen Nachbar-Pixeln.[8]

2.5 Inversion eines Graustufenbildes

Die Graustufenwerte eines Bildes werden invertiert. Schwarz wird Weiß und Weiß wird Schwarz, helle Graustufen werden dunkel und umgekehrt.

Die Graustufen eines Bildes werden in 8 bit dargestellt. Somit gibt es 255 Graustufen. 0 ist schwarz und 255 ist weiß. Somit gilt bei der Inversion für jedes Pixel mit dem Graustufenwert g: g neu = 255- g alt

2.6 Watershed Algorithmus

Der Watershed Algorithmus ist ein Region Growth Algorithmus. Er finden zusammenhängende Gruppen an Pixeln, die einander ähnlich sind. Diese Gruppen werden als Regionen bezeichnen.

Für den Watershed Algorithmus wird zunächst in jede Region, die der Algorithmus wachsen lassen soll, eine Markierung, ein seed, gesetzt. Für die seeds werden die Pixel, die dem seed ähnlich sind zu der Region zusammengefasst. So wachsen die Regionen, bis sie auf eine andere Region stößt.

Das Prinzip ist es die Graustufenwerte des Bildes als eine Topologische Karte anzusehen. Jeder seed liegt in einem Tal, in das Wasser hineingefühlt wird. Die Grenzen zwischen den Regionen liegen dort, wo das Wasser aus unterschiedlichen Tälern aufeinander trifft.[9, 10]

Region Growth ist ein Verfahren, bei dem ein Bild in Regionen unterteilt wird. So werden zum Beispiel Sektionen mit ähnlicher Farbe herausgestellt.

2.7 Connected Components

Connected Components sind benachbarte Pixel, die denselben Grauwert haben. [11]

2.8 Skelettierung

Bei der Skelettierung werden Objekte in einem Bild auf nur eine Pixelbreite reduziert. Dabei Gelten Objekte als Connected Components. So können Linien auf eine Pixelbreite geschrumpft werden.[12]

2.9 Outlier Removal

Ein Algorithmus der statistische Ausreißer entfernt. Basierend auf dem Tschebyscheff Theorem, wird der Mittelwert und die Varianz eines Satzes von Messdaten verwendet, um Ausreißer zu erkennen.[13]

2.10 fitted curve, lineare Regression, Spline

Lineare Regression ist eine Lineare Funktion durch eine menge an Messdaten legen so das sie die Messdaten am besten approximiert.

Spline. Berechnet eine Funktion, die durch angegebene Punkte verläuft.[**splineScipy**]

fitted curve. Eine Art von Funktion angeben, zum Beispiel linear, oder auch exponentiell oder logarithmisch. in der Art berechnet, das sie die Mesdames am besten approximiert, der Abstand zwischen den Daten und der Funktion also am geringsten ist.

Die Lineare Regression ist eine fitted curve mit einer linearen Funktion.[16, 17]

3 Design und Implementation

Das Ziel ist es die Einschätzung eines Haarbüschels, mithilfe von Computervision, zu automatisieren. Zunächst wird ein Kalibriervorgang durchgeführt, in dem der Nutzer abgezählte Haarbüschel-Bilder eingibt. Dann kann die Menge der Haare in nicht abgezählten Haarbüscheln automatisch geschätzt werden.

Die Schätzungen werden zugehörig zum Datum gespeichert. Um einen Zeitlichen verlauf festzuhalten. Zusätzlich zu den Schätzungen kann für jedes Datum eine Marker angegeben werden. So können regelmäßige Wiederholungen der Haarpflege, wie zum Beispiel eine Haarwäsche, notiert werden. Des weiteren können gesundheitliche Veränderungen oder Einnahme von Medikamenten vermerkt werden.

Das Programm wurde in Python implementiert und nutzt unter anderem das Plugin Open-cv. Der Nutzer steuert das Programm über die Kommandozeile.

Da die Haare jeder Person unterschiedlich sind, muss für jeden Nutzer eine eigene Kalibrierung durchgeführt werden. Um Vermischung von Daten unterschiedlicher Personen und häufige Neukalibrierungen zu vermeiden, kann das Programm beliebig viele Nutzer anlegen und verwalten.

3.1 Verarbeitung der Haar-Bilder

Sowohl für die Kalibrierung als auch für die Schätzung von Haar-Mengen müssen die eingegebenen Bilder zunächst verarbeitet werden und Daten, mit denen auf die Haar-Menge geschlossen werden kann herausgestellt werden. Dazu werden Methoden der Bildverarbeitung verwendet.

Die Methode detect implementiert die Verarbeitung eines Bildes. Als Eingabe wird von einem Haarbüschel auf einem einfarbige Hintergrund ausgegangen. Dabei sollte ein möglichst guter Kontrast zu der Haarfarbe gewählt worden sein. Beispielsweise ein weißer Hintergrund für dunkele haare oder ein schwarzer Hintergrund für blondes oder hellbraunes Haar. Die Ecken des Hintergrundes sind durch ein Symbol markiert. Vergleich Abbildung 1.

Abb. 1: Input

Im ersten Schritt der Bildverarbeitung wird das Bild auf die Markierungen in den Ecken zugeschnitten. Das ist in der Methode cropDots implementiert. Über einen Aufruf der Open cv methode matchTemplate wird nach vorkommen der Markierung gesucht. Das Resultat ist ein Karte an Wahrscheinlichkeiten, die für jedes Pixel angibt, zu welcher Wahrscheinlichkeit sich dort die Markierung befindet. Die 4 Stellen an denen die Wahrscheinlichkeit am höchsten ist, werden als Eckpunkte angesehen. Siehe Abbildung 2.[6]

Durch das Zuschneiden der Bilder wird sichergestellt, das die Haare immer etwa einen gleichgroßen Hintergrund haben. So werden die Daten, die aus den Bildern gezogen werden, nicht verfälscht dadurch, das die Haare eventuell von unterschiedlichen Abständen fotografiert wurden.

Nach dem Zuschneiden werden die Haare von dem Hintergrund getrennt. Das ist in der Methode edgeProcess implementiert. Mithilfe von Canny Edge detection werden die Haare grob erkannt. Weiße Pixel gelten als Haare, schwarze als Hintergrund. Siehe Abbildung 3a.

In der Methode hairPixelIntensity wird die Detektion verfeinert. Auf das Resultat der Kantendetektion wird eine Dilatation ausgeführt. Dadurch gelten mehr Pixel als Haare und eventuelle Löcher wachsen zusammen.

Da Haare halb-transparent sind, wird ihre Farbe intensiver, wenn sie sich überlagern. Sie weisen eine höhere Intensität auf. Um die Haar-Intensität des Bildes zu extrahieren, wird ein Bild gebaut, in dem Hintergrund Pixel weiß sind, und Haarpixel ihren Graustufenwert aus dem Original Bild erhalten.

(a) Die 4 erkannten Markierungen im Bild

(b) Resultat des Zuschneidens

Abb. 2: Der Prozess des Zuschneidens auf Markierungen in den Ecken

(a) Canny Kantendetektion

(b) Intensität-Abbildung der Haare

Abb. 3: Beispiel input von knielangen dunkelroten Haare

Wenn die Haare heller sind als der Hintergrund, es sich beispielsweise um blondes Haar auf schwarzem Hintergrund handelt haben die Haare eine Höhere Intensität, wenn sich heller sind. Bei dunklem Haar auf hellem Hintergrund ist es gegenteilig. Die Intensität ist höher, wenn das Haar dunkler ist. Damit beide Konstellationen zu einem ähnlichen Ergebnis kommen, werden die Graustufen invertiert, wenn die Haare hell sind. So ist auch bei ursprünglich hellem Haar, die Intensität höher, wenn die Graustufen dunkler sind.

Die Inversion dieses Bildes ergibt ein Bild mit schwarzem Hintergrund und Haaren, die Heller sind an stellen wo sie sich Überlagern. Siehe Abbildung 3b.

Es werden kleine vereinzelte Gebiete des Hintergrundes in der Kantendetektion erkannt und gelten somit als Haare. Um diese Gebiete zu erkennen und zu entfernen, wird der Watershed Algorithmus angewendet. Dabei werden Marker in alle verbundenen Gebiete gesetzt, die nicht schwarz sind. Der Watershed Algorithmus lässt die Regionen wachsen. Dann werden die Regionen entfernt, die im Verglich auf die Größe des Bildes sehr klein sind. Siehe Abbildung 4.

Die Summe aller Graustufenwerte werden als Intensität gespeichert. Dabei wird der Hintergrund außer Acht gelassen, weil es schwarz ist, und somit den Graustufenwert 0 hat.

(a) Intensität-Abbildung der Haare

(b) Kleine Regionen Entfernt

Abb. 4: Enternen von kleinen Regionen, vorher nachher

In der Methode hairPixelPercentage werden die Pixel gezählt die als Haare gelten. Zusätzlich wird der Prozentsatz an Haar Pixeln und die Intensität des Bildes im Vergleich zur Bildgröße gespeichert.

Dann wird die Dichte der Haare Untersucht. In der Methode Dafür werden die Hintergrund Regionen bestimmt, die durch Haare getrennt werden. Dafür wird der Watershed Algorithmus angewendet. Die Marker werden in alle zusammenhängenden Hintergrundgebiete gesetzt. Dabei wird die Größe der Region, die die Haare umschließt gespeichert. Abbildung 5b. Zusätzlich wird die Anzahl der Regionen die von Haaren umschlossen wird und deren Durchschnittliche Größe gespeichert.[9, 10]

(a) watershed input

(b) Äußerte Region Isoliert und in weiß dargestellt.

Abb. 5: Anwendung von Watershed

Als nächstes werden die Gebiete des Bildes untersucht, die eine erhöhte Haar-Dichte aufweisen. Dafür wird das Intensität-Bild skelettiert. Siehe Abbildung 6a. Die Hintergrund Regionen werden wie zuvor untersucht.

In der Methode denseAndLoosePerc wird die Haar-Intensität in den dichteren Regionen und in den weniger dichten Regionen berechnet und abgespeichert.

Für das Bild werden 28 Datenpunkte extrahiert und abgespeichert. Siehe Anhang A.

3.2 Kalibrieren

Die Kalibrierung ist in der Methode calibration implementiert. Für die Kalibrierung werden mehrere Bilder als Input angegeben. Für jedes der Bilder wird die Menge der Haare in dem Titel des Bildes angegeben. Die Bilder werden alle mit detect Untersucht. Die Daten aus den Untersuchungen, sowie die Haar-Mengen werden abgespeichert. ganzen Ordner kalibrieren. einzelne Bilder hinzufügen.

(a) Skelett des Intensität-Bildes

(b) Äußerte Region des skelettierten Bildes

Abb. 6: Dichtere Regionen untersuchen

3.3 Relation Daten zu Haar Menge

3.4 Schätzen eines Haar-Bildes

Nachdem eine Kalibrierung durchgeführt wurde. Kann die Haar-Menge auf beliebigen Bildern geschätzt werden. Das Schätzen ist in der Methode readFilesAndGuess implementiert.

Zuerst wird das zu schätzende Bild mithilfe von detect Untersucht. Dann werden die Daten aus der Kalibrierung geladen. Die Daten werden teilweise miteinander verrechnet und dann auf die dazugehörigen Haar-Mengen abgebildet. So entstehen eine Menge an Graphen, bei denen die Haar-Menge jeweils auf der y Achse liegt. Siehe Abbildung 7.

Die Punkte in den Graphen werden durch Funktionen Approximiert. Die Funktionen können dann genutzt werden um Haar-Mengen zu schätzen. Für jeden Graph werden 3 Funktionen berechnet. Eine Lineare Regression, eine Spline-Interpolation, und eine benutzerdefinierte Funktion. Der Benutzer kann aus linear, logarithmisch und exponentiell aussuchen. Siehe Abbildung 8.

Von den 28 Datenpunkten die aus den Bildern ausgewertet werden, werden 17 Genutzt um 12 Modelle zu erstellt. Diese Modelle wurden als aussagekräftig gefunden, um die Haarmenge schätzen zu können. Untersucht werden die Menge der Pixel, die als Haare zählen, Intensität, Größe und Anzahl an Hintergrund-Regionen. Siehe Anhang B Für die Berechnungen der genutzten x Werte für die Modelle.

Siehe den Anhang C für die graphische Darstellung der Modelle.

Abb. 7: Haar-Menge in Relation zur Intensität in dichten Regionen normalisiert durch gesamte Intensität und Region-Größe

Abb. 8: Funktionen, die Datenpunkte Approximieren. Spline-Interpolation, lineare Regression und exponentielle Funktion.

Für jedes der Modelle werden 3 Funktionen approximiert. Mit den Funktionen werden jeweils 3 Schätzungen für das zu schätzdene Bild berechnet.

Ausreißer werden entfernt. Dazu werden zunächst alle Schätzungen von weniger als 0 Haaren entfernt. Zusätzlich werden statistische Ausreißer mithilfe eines Algorithmus basierend auf dem Tschebyscheff Theorem entfernt.[13]

Der Mittelwert, der übrig gebliebenen Schätzungen, ist das Endergebnis. Siehe Abbildung **??** für den Verlauf einer Schätzung auf der Kommandozeile.

```
D:\Eigene Dateien\Dokumente\GitHub\HairEstimation\HairEstimationPython>python Estimation.py guess Users/Mummel/estimationInput/IMG_20200406_153354.jpg -d -r
active user: Mummel

guessing the amount of hair in the picture
image path Users/Mummel/estimationInput/IMG_20200406_153354.jpg
cropping image to TemplateDot using PatternMatching
Users/Mummel
detecting Hair via edge detection
backgroundcolor 77.54571671611444
removing smaller Regions...
done
processing background regions...
innerSectionNum 203
done
processing background regions...
innerSectionNum 69
done
Guess by hairpercent [ spl 17.10918697133671 ;func 16.068781620734256 ;linguess 8.817563600952198 ]
Guess by density * hairsection size [ spl 17.109186971336708 ;func 16.068779751624007 ;linguess 8.817563600952205 ]
Guess by density per sectionsize per hairperc [ spl 17.024852282387712 ;func 16.08739465599372 ;linguess 12.660421865313491 ]
Guess by density of dense section in realtion to section size [ spl -5.032616520694617 ;func 12.65011382719596 ;linguess -0.7966692059873983 ]
Guess by dense Density to looseDensity ratio [ spl 21.878453935854328 ;func 25.152241123554553 ;linguess -6.112537002512814 ]
Guess by density by background sections peeking through [ spl 17.841441686572296 ;func 12.290643639856775 ;linguess 12.290643505525784 ]
Guess by dense Density by background sections peeking through [ spl 14.153844827316952 ;func 10.218164223287738 ;linguess 10.21816431054834 ]
Guess by intensity in relation to hairPixels and hairsection size [ spl 16.646633135931893 ;func 18.824654298332096 ]
Guess by average backgorund section sizes in dense section [ spl -19.15505299724856 ;func -1.5712803441519831 ]
Guess by dense average background section sizes compared to looseaveragebackgorund sections size [ spl 23.017721516155298 ;func 26.624021893042254 ]
Guess by denseIntensity/intensitySum * denseSectionSize [ spl 30.189213701279854 ;func 15.04653160017627 ;linguess -1.9621895800223128 ]
Guess by denseIntensity/denseHairSum* denseSectionSize [ spl 24.005185075441997 ;func 6.7281532831400686 ;linguess 6.728152779303436 ]
[8.82 17.11 16.07 8.82 17.11 16.07 12.66 17.02 16.09 12.65 21.88 25.15
 12.29 17.84 12.29 10.22 14.15 10.22 16.65 18.82 23.02 26.62 30.19 15.05
 6.73 24.01 6.73]
outliers Removed [17.11 16.07 17.11 16.07 12.66 17.02 16.09 12.65 21.88 12.29 17.84 12.29
 10.22 14.15 10.22 16.65 18.82 15.05]
mean 15.232660728295835 res 15.0 median 16.06878068617913
found duplicate date. Using handeling mode: r
replacing old datapoint with 2020-04-06 15.0 ImagePath: Users/Mummel/estimationInput/IMG_20200406_153354.jpg
```

Abb. 9: Kommandozeilenaufruf und Verlauf einer Schätzung

3.5 Tests

3.6 Test: Knielange, dunkelrote Haare

Knielange dunkelroten Haare wurden auf einem weißem Hintergrund aufgenommen. Die Ecken wurden mit dunkeln Punkten markiert. Siehe Abbildung 11.

es wurde eine Untersuchung in einem Zeitraum von 7 Wochen von dem tägliche Haarausfall angestellt.

Um die Fehlerrate zu testen zu können wurde für einige der eingage Bilder die Anzahl der Haare per Hand gezählt. Mit einer großen Menge an Kalibrations-Bildern konnten die folgenden Fehler in den Schätzungen vermerkt werden:

['15' '15' '18' '22' '22' '25' '30' '30' '37' '40' '60' '95'] calibration images [27 22 23 22 67 17 23 16 28 19 17 14 13 20 27 26 89 25 13 23 19 14 13 34 estimated 21 16 56 27 20] (estimated, actual) [(27, 22), (22, 22), (23, 25), (22, 18), (67, 65), (17, 22), (23, 22), (16, 18), (28, 26), (19, 21), (17, 18), (14, 9), (13, 17), (20, 27), (27, 30), (26, 27), (89, 95), (25, 40), (13, 15), (23, 31), (19, 31), (14, 11), (13, 16), (34, 50), (21, 30), (16, 12), (56, 45), (27, 34), (20, 16)] error per estimation [5 0 -2 4 2 -5 1 -2 2 -2 -1 5 -4 -7 -3 -1 -6 -15 -2 -8 -12 3 -3 -16 -9 4 11 -7 4]

Abb. 10: Schätzungen aus einem Zeitraum von 7 Wochen

Mit nur 3 Kalibrationsbilern der Menge: 22, 60, 120 konnte diese Fehlerrate gefunden werden:

['15' '60' '95'] [30 26 25 23 81 14 22 9 33 17 12 9 8 19 24 28 94 26 7 23 16 6 6 39 18 10 62 31 19] (estimated, actual) [(30, 22), (26, 22), (25, 25), (23, 18), (81, 65), (14, 22), (22, 22), (9, 18), (33, 26), (17, 21), (12, 18), (9, 9), (8, 17), (19, 27), (24, 30), (28, 27), (94, 95), (26, 40), (7, 15), (23, 31), (16, 31), (6, 11), (6, 16), (39, 50), (18, 30), (10, 12), (62, 45), (31, 34), (19, 16)] error per estimation [8 4 0 5 16 -8 0 -9 7 -4 -6 0 -9 -8 -6 1 -1 -14 -8 -8 -15 -5 -10 -11 -12 -2 17 -3 3] mean error 6.896551724137931 small calibration room

3.7 Test: Hüftlange, feine, blonde Haare

Blonde Haare wurden vor einem Schwarzen Hintergrund aufgenommen.

(a) Geschätzt auf 23 Haare. Tatsächlich 25 Haare

(b) Geschätzt auf 89 Haare. Tatsächlich 95.

Abb. 11: Beispiel input von blonden langen Haaren

Für den schwarzen Hintergrund wurde besonders matte schwarze Pappe verwendet. Wenn die Struktur der Pappe zu grob ist, oder durch einen Glanz der pappe sichbar wird, wir sie von der Canny Edge detection erkannt.

Auf schwarzem Hintergrund ist Staub besonders gut sichtbar. Um die Wahrscheinlichkeit zu verringern, das Verschmutzungen auf der Pappe für die Markierungen in der Ecke gehalten werden, wurden die Markierungen von einem einfachen Punkt zu einem kreuz geändert.

Mit ? Kalibrations-Bilder kann folgende Fehlerrate gefunden werden. ['10' '10' '10' '120' '124' '18' '20' '20' '25' '3' '30' '30' '30' '30' '40' '5' '50' '51' '55' '70'] [4 10 30 44 50 24 24 71] (estimated, actual) [(4, 3), (10, 10), (30, 30), (44, 50), (50, 55), (24, 20), (24, 20), (71, 70)] error per estimation [1 0 0 -6 -5 4 4 1] mean error 2.625

Siehe Anhang B für Schätzungs modelle und die Verarbeitung von Blonden Haaren.

3.8 Test: kurze Hellbraune Haare

Für hellbraune haare hat es sich gezeigt, das ein schwarzen Hintergrund einen besseren Kontrast bietet.

(a) Geschätzt auf ?? Haare. Tatsächlich 10 Haare

(b) Geschätzt auf ?? Haare. Tatsächlich 40.

Abb. 12: Beispiel input von blonden langen Haaren

Die Haare sind recht kurz und haben somit weniger Masse, im vergleich zu den vorherigen Testszenarien. Auch hier liegt die Fehlerrate bei :

['10' '20' '40'] [8 10 14 20 16 31 36 40 37 10] (estimated, actual) [(8, 5), (10, 10), (14, 14), (20, 20), (16, 20), (31, 30), (36, 30), (40, 40), (37, 40), (10, 7)] error per estimation [3 0 0 0 -4 1 6 0 -3 3] mean error 2.0

3.9 Test: Mama

TODO

4 Ergebnisse

Durchschnittliche Fehler von 5 manchmal fehler von 10 Haaren. Da Haare manchmal mehr kürzere und manchmal mehr längere ausfallen kann man das nicht wirklich akkurat zählen. aber die generelle menge scheit gut geschätzt zu werden.

ergebnisse more detailed als in dem paper. in dem paper wurde in inkrementen von 10, 50, 100, 200 etc gewertet.

5 Fazit

Das Schätzen von nicht zählbaren Mengen an Haaren zeigt sich als funktionsfähig. In der Zukunft können die unterschiedlichen Schäztungs Modelle miteinander verglichen werden, um die Kombination der akkuratesten Modelle zu finden. Des weiteren kann mit noch mehr Testsituationen mit unterschiedlichen Haar Typen aufgebaut werden.

Die Bildverarbeitung dauert recht lange, das könnte verbessert werden.

Literatur

[1] Rodney Sinclair. „Hair Shedding In Women: How much is too much?“ In: *The British journal of dermatology* 173 (Apr. 2015). DOI: 10.1111/bjd.13873.

[2] E.Y. Hsiang u. a. „Seasonality of hair loss: a time series analysis of Google Trends data 2004 to 2016“. In: *British Journal of Dermatology* 178 (Okt. 2017). DOI: 10.1111/bjd.16075.

[3] Howard Maibach und Ian Yamaguchi. „Chemically Induced Hair Loss/Alopecia“. In: Jan. 2012, S. 1935–1942. ISBN: 978-3-642-02034-6. DOI: 10.1007/978-3-642-02035-3_205.

[4] Leonid Trost, Wilma Bergfeld und Ellen Calogeras. „The diagnosis and treatment of iron deficiency and its potential relationship to hair loss“. In: *Journal of the American Academy of Dermatology* 54 (Juni 2006), S. 824–44. DOI: 10.1016/j.jaad.2005.11.1104.

[5] María Martínez-Velasco u. a. „The Hair Shedding Visual Scale: A Quick Tool to Assess Hair Loss in Women“. In: *Dermatology and Therapy* 7 (Feb. 2017). DOI: 10.1007/s13555-017-0171-8.

[6] *Template Matching*. URL: https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html (besucht am 17. 04. 2020).

[7] *Canny Edge Detector*. URL: https://docs.opencv.org/master/da/d5c/tutorial_canny_detector.html (besucht am 17. 04. 2020).

[8] *Eroding and Dilating*. URL: https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html (besucht am 17. 04. 2020).

[9] Anton Kornilov und Ilia Safonov. „An Overview of Watershed Algorithm Implementations in Open Source Libraries“. In: *Journal of Imaging* 4 (Okt. 2018), S. 123. DOI: 10.3390/jimaging4100123.

[10] *Image Segmentation with Watershed Algorithm*. URL: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html (besucht am 17. 04. 2020).

[11] *Connected Components in OpenCV*. URL: https://davidlavy.wordpress.com/opencv/connected-components-in-opencv/ (besucht am 17. 04. 2020).

[12] *Skeletonize*. URL: https://scikit-image.org/docs/dev/auto_examples/edges/plot_skeleton.html (besucht am 17. 04. 2020).

[13] B.G. Amidan, Thomas Ferryman und Scott Cooley. „Data outlier detection using the Chebyshev theorem“. In: Apr. 2005, S. 3814–3819. DOI: 10.1109/AERO.2005.1559688.

[14] *Matplotlib*. URL: https://matplotlib.org/ (besucht am 17. 04. 2020).

[15] *Numpy*. URL: https://numpy.org/ (besucht am 17. 04. 2020).

[16] *Introduciton to Curve Fitting*. URL: https://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Introduction_to_Curve_Fitting.pdf (besucht am 17. 04. 2020).

[17] *scipy curve$_f$it*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html (besucht am 17. 04. 2020).

A Anhang: 28 Datenpunkte pro Bild

– intensitySum - Summe der Graustufenwerte des Intensität-Bildes

- intensityShare - Intensität pro HaarPixel
- hairpixels - Anzahl der Pixel, die Haare darstellen
- imagepixels - Gesammtanzahl der Pixel im Bild
- hairpixel percentage - Prozentsatz der Haarpixel im Bild
- number of sections - Anzahl an Hintergrund Regionen, die Watershed findet.
- number of background sections - Anzahl an Regionen, die den Hintergrund darstelle. (2 weniger als number of section. Watershed zählt Haare und übergänge zwischen Regionen mit)
- outerSectionSum - Summe an Pixeln, die sich in der Region befinden, die die Haare umschließt
- outerSectionPercentage - outerSectionSum/imagepixels
- innerSectionSum - Summa an Pixeln, die sich in Regionen befinden, die von Haaren umschlossen werden
- innserSectionAvgSize - Druchschnittliche Pixelanzahl, die sich in den umschlossenen Regionen befinden
- innerSectionAvgSize Percentage - innersectionAvgSize/imagepixels
- innerSectionSizeVariance - statistische Varianz der Pixelmengen, die sich in umschlossenen Regionen befinden
- innerSection standard deviation - Standardabweichung der Pixelmengen, die sich in umschlossenen Regionen befinden
- number of sections only concidering dense hair - Anzahl der Regionen, die Watershed findet nachdem der input Skelettiert wurde. Regionen werden von dichteren Haaren unterteilt.
- number of background sections inclosed by dense hair - Anzahl an Regionen, die den Hintergrund zwischen dichten Haaren darstellen
- dense hair outerSectionSum - Summe an Pixeln, die die Regionen umschließen, die von dichten Haaren umschlossen werden
- dense outerSectionPercentage - dense hair outerSectionSum / imagepixels
- dense innerSectionSum - Summe an Pixeln, die sich in den Regionen befinden, die von dichten Haaren umschlossen werden
- dense hair innserSectionAvgSize - Durschnittliche anzahl der Pixel, die sich in Regionen befinden, die von dichten Haaren umschlossen werden
- dense hair innerSectionAvgSize Percentage - dense hair innerSectionAvgSize / imagepixels
- dense innerSectionSizeVariance - statistische Varianz der Pixelmengen
- dense innerSection Size standard deviation - Standardabweichung der Pixelmengen

- denseHairSum - Summe an Pixeln, die als Dichte Haare angesehen werden
- looseHairSum - Summe an Pixeln, die als Haare angesehen werden, jedoch nicht als Dicht gelten
- IntensitySum in Loose Section - Summe der Graustufenwerte in nicht-dichten Regionen
- IntensitySum in Dense Section - Summe der Graustufenwerte in dichten Regionen

B Anhang x Wert berechungen für Schätzungs-Modelle

- hairpercent
  - Prozentsatz der Haarpixel im Bild.
- (hairpixels / hairSectionSize) * (1 - outerSectionPerc)
  - HaarPixel Dichte in der Region, in der sich Haare Befinden. Gibt dasselbe Ergebnis wie hairpercent.
- (hairpixels/hairSectionSize)*(1-outerSectionPerc)*hairPerc
- (((denseHairSum / denseInnerSectionSize) * (1 - denseSectionperc))) = denseDensity
  - Dichte in den Dichten Regionen.
- denseDensity / looseDensity
  - Verhältnis dichte Regionen zu loosen Regionen.
- backgroundSectionNum*(1-outerSectionPerc)
  - Anzahl der HintergrundRegionen in Region, in der sich Haare Befinden.
- denseSectionNum*(1-outerSectionPerc)
  - Anzahl der HintergrundRegionen, die von dichten Haaren getrennt sind, in Haar-Region.
- intensitySum*hairpixels*(1-outerSectionPerc)
  - Intensitäts Summe pro HaarPixel in Haar-Region.
- denseSectionAVGSize/(1-denseSectionperc)
  - Durchschnittliche Größe der HintergrundRegionen, die durch dichte Haare getrennt sind.
- (denseSectionAVGSize/looseSectionAVGSize)/(1-outerSectionPerc)
  - Durchschnittliche Größe von Hintergrundregionen in dichten Regionen verglichen mit Durchschnittlicher Größe von Hintergrundregionen in loosen Regionen.
- (denseIntensity/intensitySum)*(1-denseSectionperc)
  - Intensität in Dichten Regionen verglichen mit gesammter Intensität.

– (denseIntensity/denseHairSum)*(1-denseSectionperc)

- Intensität in Dichten Regionen verglichen mit Summe an Haarpixeln, die als dicht gelten.

C Anhang Schätzungs Modelle

(a) hairpercent

(b) density*hairsection size

(c) density per sectionsize per hairperc

(d) density of dense section in relation to section size

(e) dense Density to loose Density ratio

(f) density by background section peeking through

Abb. 13: Schätzungsmodelle angewandt auf lange rotbraune Haare (1)

(a) dense Density by background section peeking through

(b) intensity in relation to hairPixels and hairsectionsize

(c) average background section sizes in dense section

(d) dense average background section size compared to looseaveragebackground section size

(e) densityIntensity/intensitySum * denseSectionSize

(f) denseIntensity/denseHairSum * denseSectionSize

Abb. 14: Schätzngsmodelle angewandt auf lenge rotbraune Haare (2)

D Anhang: Blondes Haar

(a) Input

(b) Wahrscheinlichkeiten von matchTemplate

(c) 4 gefundene Markierungen

(d) Auf Markierungen zugeschnitten

(e) Kantendetektion

(f) Intensität

Abb. 15: My complicated figure

(a) Kleine Regionen entfernt

(b) Intensität auf Input(Grayscale) ausgegeben. Zeigt wie viele Haare nicht erkannt wurden (hier: nicht viele)

(c) Intensität-Bild dient als Input für Hintergrund-Region-Untersuchung

(d) Hintergrund Region, die die Haare umschließt, in weiß

(e) Intensität-Bild nach Skelettierung. Input für 2. Hintergrund Untersuchung

(f) Hintergrund Region, die dichte Haare umschließt

Abb. 16: My complicated figure

E Anhang Schätzungs Modelle Blondes Haar

(a) hairpercent

(b) density*hairsection size

(c) density per sectionsize per hairperc

(d) density of dense section in relation to section size

(e) dense Density to loose Density ratio

(f) density by background section peeking through

Abb. 17: Schätzungsmodelle angewandt auf feine blonde Haare (1)

(a) hairpercent

(b) density*hairsection size

(c) density per sectionsize per hairperc

(d) density of dense section in relation to section size

(e) dense Density to loose Density ratio

(f) density by background section peeking through

Abb. 18: Schätzungsmodelle angewandt auf feine blonde Haare (2)