UNIVERSITY OF APPLIED SCIENCES

Department of Computer Science

IT-Sicherheit Seminar

Approximation und Detektion von sichtbaren, digitalen Wasserzeichen

Eingereicht am:

1. Januar 2018

Eingereicht von:
Alina Claussen
E-mail: its103194@fh-wedel.de

Referent:
Prof. Dr. Gerd Beuster
Fachhochschule Wedel
Feldstraße 143
22880 Wedel
Phone: (041 03) 80 48-38
E-mail: gb@fh-wedel.de

„However, the fact that watermarks are added in a consistent manner to many images has thus far been overlooked.“ - On the Effectiveness of Visible Watermarks [Pap]

Inhaltsverzeichnis

1 Einleitung

Diese Ausarbeitung befasst sich mit dem Paper „On the Effectiveness of Visible Watermarks“[Pap] verfasst von Tali Dekel, Michael Rubinstein, Ce Liu und William T. Freeman. Hier wird vor allem auf die Approximation des digitalen Wasserzeichens und die Detektion des Wasserzeichens in digitalen Bildern eingegangen.

Sichtbare Wasserzeichen werden vielfach genutzt, um Bilder als urheberrechtlich geschützt zu kennzeichnen. Das Vervielfältigen und der damit verbundene Missbrauch von digitalen Bildern im Internet wird hierdurch unattraktiv gemacht. Diese Ausarbeitung konzentriert sich darauf zu zeigen, dass es mit einer hohen Treffergenauigkeit möglich ist, das originale Bild zurückzugewinnen und das Wasserzeichen zu extrahieren. Um das Wasserzeichen erkennen zu können, ist eine umfangreiche quantitative Analyse von Bilddaten nötig. Daher sind Wasserzeichen ein adäquates Mittel, um einzelne aber auch eine Kollektion von Bildern mit einem Copyright zu versehen.

Sichtbare Wasserzeichen beinhalten halbtransparente komplexe Strukturen, z.B. dünne Linien und Schatten, die es schwieriger machen sollen es von dem Originalbild zu eliminieren. Ein Wasserzeichen eines einzelnen Bildes ohne Benutzerinformationen entfernen zu können, stellt eine große Herausforderung dar. Wird ein Wasserzeichen in einer Kollektion von Bildern in der gleichen Art und Weise genutzt, z.B. ein Firmenlogo, ist es möglich den Prozess zu invertieren. Die Redundanz der Wasserzeichenbilddaten hat einen hohen Einfluss auf den Algorithmus des automatisierten Eliminierens des Wasserzeichens. Zunächst werden konsistente Bildstrukturen in der Bilderkollektion extrahiert, um ein mattiertes potentielles Wasserzeichen zu ermitteln. Anschließend wird die Region des Wasserzeichens in allen Bildern festgelegt. Es hat sich gezeigt, dass einige hundert Bilder ausreichen, um die Erkennung des Wasserzeichens zu gewährleisten. Es werden die Bereiche des Wasserzeichens nicht rekonstruiert, sondern der Prozess des Versehens des Bildes mit Wasserzeichen wird invertiert.

Es stellt sich die Frage wie man derartige Angriffe von konsistenten Wasserzeichen durchbrechen und abwehren kann. Die Veränderung der Position des Wasserzeichens, der Farbe oder der Opazität reicht nicht aus um Angriffe abzuwehren, hingegen kaum wahrnehmbare räumliche Änderungen an dem Wasserzeichen selbst kann die Qualität des gewonnenen Originalbildes erheblich beeinträchtigen.

Diese Ausarbeitung soll das Bewusstsein schärfen, dass sichtbare Wasserzeichen nicht nur einzelne Bilder, sondern auch Kollektion von Bildern, urheberrechtlich schützen sollen. Der Prozess des Versehens der Bilder mit Wasserzeichen muss sicherstellen, dass der Aufwand Algorithmen zu entwickeln, die das Originalbild wiederherstellen können, maximiert werden.

2 Grundlagen

In diesem Kapitel werden allgemeine Grundlagen in den Bereichen Mathematik und Bildverarbeitung erläutert. Zunächst wird der Nabla-Operator eingeführt. Dann werden digitale Bilder beschrieben und die in diesem Kontext benötigten Operationen und Filter dargestellt. Zum Schluss wird ein Überblick über die Poisson-Rekonstruktion gegeben.

2.1 Der Nabla-Operator

Der Nabla-Vektor ist eine Spezialform eines Vektors. Vektoren setzen sich aus mehreren Komponenten zusammen. Bei einem Nabla-Vektor, gekennzeichnet mit dem Nabla-Operator ∇, sind diese Komponenten Differential-Operatoren. [Fri73]

2.2 Aufbau eines digitalen Bildes

In dieser Ausarbeitung werden nur digitale Bilder betrachtet. Um digitale Bilder speichern zu können, müssen diese auf irgendeine Art und Weise als Zahlenmenge dargestellt werden können. Daher werden digitale Bilder als zweidimensionale Matrix von Zahlen gespeichert. Die Zahlen repräsentieren die Farbwerte. Man könnte auch sagen ein digitales Bild I ist eine zweidimensionale Funktion von den ganzzahligen Koordinaten $\mathbb{N}$ x $\mathbb{N}$ auf eine Menge von Bildwerten $\mathbb{P}$:

$$I(u, v) \in \mathbb{P} \quad und \quad u, v \in \mathbb{N} \tag{2.1}$$

Weiterhin wird angenommen, dass die digitalen Bilder rechteckig sind, das heißt die Bildgröße kann durch die Breite M und die Höhe N der zugehörigen Bildmatrix I bestimmt werden. Die Position der einzelnen Bildelemente oder auch Pixel genannt, wird über ein Koordinatensystem definiert. Dieses Koordinatensystem ist bezogen auf die üblichen Konventionen eines normalen Koordinatensystems in vertikaler Richtung umgedreht. Das bedeutet, dass die y-Richtung von oben nach unten verläuft und die x-Richtung wie üblich von links nach rechts verläuft. [Bur15]

2.3 Filter

Filter ist eine allgemeiner Begriff für Funktionen und Operationen, die ein digitales Bild nach einem bestimmten Algorithmus verändern. Dabei wird die Geometrie des Bildes nicht verändert. Für jeden Pixel des Ursprungsbildes wird ein neuer Wert aus einer umliegenden Menge von Pixeln aus dem Ursprungsbildes berechnet. Die Region, in der diese Menge von Pixeln liegen, wird Filterregion genannt. Die Filterregion legt das räumliche Ausmaß, also wie viele ursprüngliche Pixel

zur Berechnung verwendet werden, fest. Die Form der Filterregion ist zunächst nicht festgelegt. Eine Gewichtung der ursprünglichen Pixel für die Berechnung ist möglich.

Filter können durch eine Filtermatrix beschrieben werden. Die Größe der Filterregion ist äquivalent mit der Größe der Filtermatrix. Die Elemente der Filtermatrix beschreiben die Gewichtung der Pixelwerte. Jede Filtermatrix besitzt ein eigenes Koordinatensystem, indem der Ursprung, der sogenannte *Hot Spot*, frei wählbar ist. Eine einfache Glättungsmatrix sieht wie folgt aus [Bur15]:

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.2}$$

2.3.1 Gaußfilter

Der Gaußfilter ist ein Glättungsfilter, der wie folgt definiert ist:

$$H^{G,\sigma}(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3}$$

σ ist dabei die Standardabweichung. Die höchste Gewichtung hat das mittlere Bildelement und nach außen hin wird die Gewichtung geringer. Ein Beispiel für eine Filtermatrix für den Gaußfilter wäre [Bur15]:

$$H(i,j) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.4}$$

2.3.2 Prewitt- und Sobel-Operator

Der Prewitt und Sobel-Operator sind Kantenoperatoren, die sich einem 3×3 - Ableitungsfilter bedienen. Diese Operatoren können in x- und in y-Richtung angewandt werden und sind sich sehr ähnlich. Die Filter des Prewitt-Operators sind folgendermaßen definiert [Bur15]:

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad und \quad H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.5}$$

Die Filter des Sobel-Operators sehen wie folgt aus [Bur15]:

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad und \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.6}$$

2.4 Canny-Kantenoperator

Der Canny-Kantenoperator ist ein sehr verbreitetes Verfahren, welches zur Kantendetektion eingesetzt wird. Der Algorithmus teilt sich in drei Abschnitte auf. Zunächst wird in der Vorbereitungsphase das Bild geglättet und für jede Position der x/y-Gradient und dessen Betrag und Richtung berechnet. In der zweiten Phase werden die Kanten anhand eines lokalen Maximums entlang der Gradientenrichtung lokalisiert. Zuletzt werden zusammenhängende Kanten mithilfe eines Hysterese-Schwellwerts selektiert und verfolgt. [Bur15]

2.4.1 Erste Phase: Vorverarbeitung

Zunächst wird der Gaußfilter auf das Bild angewandt. Die Größe des Filters spezifiziert der Parameter σ. Dieses Vorgehen soll Bildrauschen reduzieren.

Danach wird ein Differenzfilter wie zum Beispiel die Prewitt- und Sobel-Operatoren in x- und y-Richtung auf das Bild angewandt. Die beiden Ergebnisse werden wieder vereint, indem pro Pixelposition die jeweiligen Kanten-Gradienten wie folgt verrechnet werden:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2.7}$$

Die Richtung der Kante wird ebenfalls pro Pixelposition berechnet und wird in einem Winkel angegeben, der wie folgt berechnet wird [Bur15][Can]:

$$\theta = \tan^{-1}(\frac{G_y}{G_x}) \tag{2.8}$$

2.4.2 Zweite Phase: Kantenlokalisierung

Mit der Technik der „Non-Maximum Suppression“ werden die Kanten isoliert. Das Prinzip besteht darin, lokale Maxima entlang der Kanten zu detektieren. Der Betrag der berechneten Richtungen der Gradienten aus der ersten Phase wird auf vier diskrete Richtungen gerundet. Abbildung 2.1 zeigt die vier Richtungen. Ein Bildpunkt wird nur dann als Kantenelement betrachtet, wenn der Betrag des Gradienten größer ist als seine Nachbarelemente. Wenn ein Element als Kantenelement identifiziert wurde, bleibt der Wert bestehen. Wenn dies nicht der Fall ist, wird der Wert auf 0 gesetzt. [Bur15]

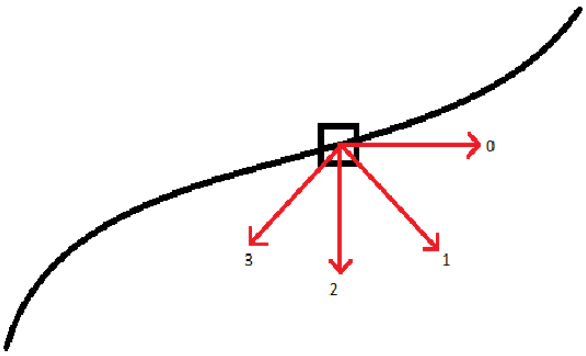

Abbildung 2.1: Vier mögliche Richtungen der Gradienten

2.4.3 Dritte Phase: Kantenverfolgung

In dieser Phase werden die Kantenelemente aus der vorherigen Phase unter Verwendung des Hysterese-Schwellwerts ergänzt. Zunächst werden zwei Schwellwerte t_{hi} und t_{lo} festgelegt. Dabei gilt $t_{hi} > t_{lo}$. Alle Werte in dem Bild, die größer gleich t_{hi} sind, sind definitiv Kanten. Analog dazu sind alle Werte, die kleiner als t_{lo} sind, keine Kanten. Die Werte, die kleiner als t_{hi} und größer gleich t_{lo} sind, sind potentielle Kanten. Bildpunkte, die Werte in diesem Bereich haben, werden als Kanten identifiziert, wenn sie eine Verbindung zu einer eindeutig identifizierten Kante haben. [Bur15][Can]

2.5 Poisson-Rekonstruktion

Die Poisson-Rekonstruktion stellt aus einer gegebenen Menge von Punkten mit Normalen die ursprüngliche Oberfläche wieder her. Bedingung ist, dass die gegebenen Punkte die Oberfläche grob definieren. Es wird eine Indikatorfunktion definiert, die besagt, ob sich ein Punkt auf der Oberfläche befindet. So wird die Punktemenge mit der resultierenden Oberfläche in Relation gesetzt. Für die Berechnung der Indikatorfunktion wird sich den Gradienten bedient. Der Gradient wird um die Divergenz erweitert und wird so zum allgemeinen Problem der Poisson Gleichung. Die Poisson Gleichung ist eine partielle Differentialgleichung zweiter Ordnung und findet meistens ihre Anwendung in dem Bereich der Physik oder bei Randwertproblemen. [Poi]

3

Gewinnung des Wasserzeichens

In diesem Kapitel wird die Vorgehensweise erläutert, wie das Wasserzeichen mit Hilfe einer Kollektion von Bildern approximiert wird. Ziel ist es, das Wasserzeichen zu isolieren.

Ein Bild, welches mit einem Wasserzeichen versehen wird, lässt sich aus dem Wasserzeichen W und dem natürlichem Bild I wie folgt berechnen:

$$J(p) = \alpha(p)W(p) + (1 - \alpha(p))I(p) \tag{3.1}$$

Dabei entspricht $p = (x, y)$ der Pixelposition und $\alpha(p)$ beschreibt den Deckungsfaktor des Wasserzeichens. α lässt sich auch so beschreiben: $\alpha = c \cdot \alpha_n$. c ist dabei ein konstanter Blendungsfaktor und für die normalisierte Alphamatte α_n gilt $\alpha_n \in [0, 1]$.

Um das natürliche Bild I zu extrahieren, kann die oben genannte Formel umgestellt werden:

$$I(p) = \frac{J(p) - \alpha(p)W(p)}{1 - \alpha(p)} \tag{3.2}$$

Wenn W und α bekannt wären, könnte aus jedem beliebigen Bild mit diesem Wasserzeichen, dieses entfernt werden. Daher ist das Ziel, das Wasserzeichen W und die dazugehörigen Deckungsfaktoren α zu ermitteln.

An dieser Stelle wird die Eigenschaft, dass das Wasserzeichen mit gleicher Deckungskraft auf viele unterschiedliche Bilder angewandt wird, ausgenutzt. Ausgangssituation ist somit, dass K Bilder vorhanden sind, von denen bekannt ist, dass alle das gleiche Wasserzeichen W enthalten. Mathematisch ausgedrückt sieht das so aus:

$$J_k = \alpha W + (1 - \alpha)I_k, \quad k = 1, ..., K \tag{3.3}$$

Damit ermittelt werden kann, welche Strukturen zum Wasserzeichen gehören, wird die Kenntnis benötigt, welcher Bereich mit einem Wasserzeichen versehen ist. Hier wird vorausgesetzt, dass diese Region angegeben wird. Das folgende Verfahren gleicht dem Canny-Kantenoperator mit ein paar entscheidenden Unterschieden.

Zunächst werden für jedes Bild der Kollektion die Gradienten in x- und y-Richtung berechnet. Hierfür werden zum Beispiel die Prewitt- und Sobel-Operatoren genutzt. Aus diesen Bildern wird nun ein neues Bild erstellt. An jedem Bildpunkt wird unabhängig voneinander der Median in x- und y-Richtung ermittelt:

$$\nabla\widehat{W}_m(p) = median_k(\nabla J_k(p)) \tag{3.4}$$

Je mehr Bilder verarbeitet werden, desto genauer wird die Approximation des Wasserzeichens $W_m = \alpha W$. Das Weichzeichnen am Anfang der ersten Phase des Canny-Kantenoperators ist durch die Anwendung des Medians nicht notwendig. Im Gegensatz zum Canny-Kantenoperator wird nun die Größe des approximierten Wasserzeichens berechnet und entsprechend ausgeschnitten. Im Anschluss wird mit den Prozessschritten des Canny-Kantenoperator fortgefahren und wie beschrieben bis zum Ende durchgeführt. Zum Schluss wird das Wasserzeichen mit der Poisson-Rekonstruktion gewonnen. [Pap]

4

Detektion des Wasserzeichens

In diesem Kapitel wird das gegebene Wasserzeichen $\nabla\widehat{W}_m$ in den Bildern der Kollektion detektiert. Hierzu wird eine Distanztransformation und das Chamfer Matching eingesetzt. Der Chamfer-Algorithmus ist ein effizientes Verfahren eine Distanztransformation zu berechnen. Das Chamfer Matching dient der Mustererkennung.

4.1 Distanztransformation

Eine Distanztransformation ist eine morphologische Operation auf Binärbildern. Ziel ist es, für jede Bildposition die Distanz zum nächstgelegenen Vordergrundpixel zu bestimmen. Binärbilder sind Bilder, wo nur zwei Pixelwerte genutzt werden. Diese Pixelwerte sind 0 und 1, was so viel heißt wie schwarz und weiß.

Um nun eine Distanztransformation durchführen zu können, werden Vordergrund- und Hintergrundpixel definiert. Bei einer Bildgröße von $M \times N$ beschreiben $u = 0, ..., M-1$ und $v = 0, ..., N-1$ die Bildposition. In einem Binärbild $I(u, v) = I(p)$ werden die Vordergrund- bzw. die Hintergrundpixel wie folgt definiert:

$$FG(I) = p|I(p) = 1 \tag{4.1}$$
$$BG(I) = p|I(p) = 0 \tag{4.2}$$

Die Distanztransformation von I, $D(p) \in \mathbb{R}$, wird folgendermaßen definiert:

$$D(p) := \min_{p' \in FG(I)} dist(p, p') \tag{4.3}$$

Dabei muss $p' \in FG(I)$ gelten. Die Funktion $dist(p, p')$ soll dabei den geometrischen Abstand der beiden Bildpositionen $p = (u, v)$ und $p' = (u', v')$ angeben. Für diese Funktion können verschiedene Distanzfunktionen verwendet werden. Gängig sind die euklidische Distanz und die Manhattan-Distanz. In dieser Ausarbeitung wird die euklidische Distanz benutzt. Die euklidische Distanz wird wie folgt definiert [Bur15]:

$$dist(p, p') = \|p - p'\| = \sqrt{(u - u')^2 + (v - v')^2} \tag{4.4}$$

4.1.1 Chamfer-Algorithmus

Der Chamfer-Algorithmus ist eine Variante, um die Distanztransformation effizient zu berechnen. Für die Berechnung der Abstandswerte werden zwei Bilddurchläufe benötigt. Der erste Bilddurchlauf fängt in der linken oberen Ecke an und läuft bis in die rechte untere Ecke. Der zweite Bilddurchlauf verläuft in entgegengesetzter Richtung von der Ecke rechts unten bis in die Ecke links oben. Für die zwei Durchläufe werden zwei verschiedene Distanzmasken verwendet:

$$M^L = \begin{bmatrix} m_2 & m_1 & m_2 \\ m_1 & \times & . \\ . & . & . \end{bmatrix} \quad und \quad M^R = \begin{bmatrix} . & . & . \\ . & \times & m_1 \\ m_2 & m_1 & m_2 \end{bmatrix} \tag{4.5}$$

Der aktuelle Bildpunkt ist mit dem Zeichen $\times$ gekennzeichnet. Die Werte für m_1 und m_2 hängen von der gewählten Distanzfunktion $dist(p, p')$ ab. Für den aktuellen Bildpunkt werden die Abstände zu den relevanten Nachbarn berechnet und addiert. Die kleinste Wert von diesen Distanzen und dem aktuellen Wert wird als Distanz gespeichert.

Die Masken für den Chamfer-Algorithmus unter Verwendung der euklidischen Distanz sehen wie folgt aus [Bur15]:

$$M_E^L = \begin{bmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & \times & . \\ . & . & . \end{bmatrix} \quad und \quad M_E^R = \begin{bmatrix} . & . & . \\ . & \times & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{bmatrix} \tag{4.6}$$

4.2 Chamfer Matching

Das Chamfer Matching dient dazu ein Template R in einem Binärbild I zu erkennen. In diesem Anwendungsfall ist das Template das Wasserzeichen. Dazu wird die Distanztransformation genutzt. Um aus dem mit einem Wasserzeichen versehendes Bild ein Binärbild zu erhalten, wird in diesem Anwendungsfall der Canny-Kantenoperator auf das Bild angewandt. Das Chamfer Matching ist keine ideale Lösung, da bei unterschiedlicher Form, Lage und Größe des Wasserzeichens mit Problemen gerechnet werden muss. Der Algorithmus betrachtet keine Skalierung, Rotation oder Verformungen.

Das Template R wird über das Binärbild I bewegt. An jeder Position wird ein Maß der Übereinstimmung berechnet. Je kleiner der Wert, desto größer ist die Übereinstimmung. Das Maß Q der Übereinstimmung wird wie folgt bestimmt:

$$Q(r, s) = \frac{1}{|FG(R)|} \sum_{(i,j) \in FG(R)} D(r + i, s + j) \tag{4.7}$$

Die Distanzwerte von dem Binärbild, die in dem Template ein Vordergrundspixel sind, werden aufsummiert und am Ende durch die Anzahl der Vordergrundpixel im Template geteilt. (r, s) beschreibt die Ausgangsposition in dem Binärbild. [Bur15]

5 Ausblick

In diesem Kapitel wird kurz auf das weitere Verfahren eingegangen und Möglichkeiten aufgezeigt, diesem Angriff zu entgehen. Zum Schluss wird die Qualität des Algorithmus evaluiert.

5.1 Entfernung des Wasserzeichens

Mit den gewonnenen Erkenntnissen wäre der nächste Schritt das Kernproblem, welches die Gleichung 3.3 beschreibt, zu lösen. Ziel ist die Rekonstruierung des Originalbildes, indem das Bild mit Wasserzeichen in die drei Komponenten Wasserzeichen, Alphamatte und Originalbild zerlegt wird. Dabei wird versucht das Verfahren des Hinzufügens des Wasserzeichens rückgängig zu machen und nicht eine Approximation zu finden. Zu diesem Zeitpunkt ist die Alphamatte noch nicht bekannt, jedoch ist bekannt welche Strukturen zum Wasserzeichen gehören und welche nicht. Der Deckungsfaktor muss noch ermittelt werden. Um die drei Komponenten zu ermitteln, wird ein Optimierungsproblem formuliert. Auf dieses Optimierungsproblem wird in dieser Ausarbeitung nicht weiter eingegangen. Nachdem alle Komponenten bekannt sind, ist die Entfernung des Wasserzeichens trivial. [Pap]

5.2 Mögliche Schutzmaßnahmen

Wie bei jedem Angriff stellt sich die Frage, wie dem Angriff entgegengewirkt werden kann. Eine Möglichkeit wäre unterschiedliche Wasserzeichen zu verwenden. Dies ist allerdings sehr aufwendig. Wasserzeichen werden oft mit großem Aufwand gestaltet und bewusst platziert, daher stellt sich diese Herangehensweise als sehr aufwendig und als große Herausforderung dar. Besser wäre ein Algorithmus zu entwickeln, der kaum merkliche Veränderungen am Wasserzeichen vornimmt.

Grundsätzlich sind mehrere Variationen möglich. Zum einen sind Änderungen in der Deckfähigkeit des Wasserzeichens denkbar, zum anderen kann das Wasserzeichen kaum merklich räumlich abgeändert werden. Eine Änderung in der Deckfähigkeit oder der Position des Wasserzeichens ist kein ausreichendes Verfahren, um die Rekonstruktion des Originalbildes entgegenzuwirken, da Google auch hier einen mathematischen Algorithmus zur Umgehung vorweisen kann. Geometrische Änderungen können allerdings nicht ganz so einfach ermittelt werden und scheinen die effizienteste Variante zu sein, um sich vor Angriffen zu schützen. Der Algorithmus funktioniert in diesem Fall dennoch, das Ergebnis weist aber am Ende sichtbare Fehler auf. [Pap]

5.3 Resultat

Mit dieser beschriebenen Methodik können Wasserzeichen, die in einheitlicher Art und Weise einem Bild hinzugefügt wurden, akkurat entfernt werden. Laut eigenen Angaben hat Google die Methode

intensiv bei Bildern von mehreren Stockfotografie-Anbietern getestet. Der Algorithmus wird nicht von dem Deckungsfaktor des Wasserzeichens oder der Position beeinträchtigt. Auch die Struktur des Wasserzeichens ist nicht relevant und kann exakt bestimmt werden. Das Ergebnis wird allerdings bei Qualitätsunterschieden der natürlichen Bilder beeinträchtigt. Der effektivste Weg sich vor diesem Angriff zu schützen, sind kleine geometrische Änderungen an dem Wasserzeichen vorzunehmen, denn so kann das Ergebnis starke Qualitätseinbußen aufweisen. [Pap]

Literaturverzeichnis

[Bur15] Mark James Burger, Wilhelm und Burge. *Digitale Bildverarbeitung - Eine algorithmische Einführung mit Java*. Springer Vieweg, Berlin Heidelberg, Deutschland, 2015.

[Can] Canny-Kantendetektor. https://docs.opencv.org/3.3.1/da/d22/tutorial_py_canny.html. [Online; accessed 29-Dezember-2017].

[Fri73] Joachim Frisius. *Vektorrechnung*. Vogel-Verlag, Würzburg, Deutschland, 1973.

[Pap] On the Effectiveness of Visible Watermarks. http://openaccess.thecvf.com/content_cvpr_2017/papers/Dekel_On_the_Effectiveness_CVPR_2017_paper.pdf. [Online; accessed 28-Dezember-2017].

[Poi] Poisson Surface Reconstruction. http://www.cs.jhu.edu/~misha/MyPapers/SGP06.pdf. [Online; accessed 29-Dezember-2017].