Visualisierung digitaler Wasserzeichen

Praxisarbeit Sommersemester 2005

Ausbildungsstätten	Fraunhofer IPSI Dolivostraße 15 64293 Darmstadt Berufsakademie Mosbach Lohrtalweg 10 74812 Mosbach
Studiengang	Digitale Medien
Kurs	DM04
Verfasser	Katja Deutschmann Urbanusstraße 13 63933 Mönchberg
Betreuer	Patrick Wolf
Einreichungsdatum	2006-04-20

Diese Praxisarbeit ist formal und inhaltlich geprüft und zur Einreichung an die Berufsakademie freigegeben.

Unterschrift Ausbildungsbeauftragter

Stempel Ausbildungsbetrieb

Inhalt

Inhalt i
Kurzfassung 1
1 Einleitung 1
2 Problembeschreibung 2
2.1 Wasserzeichen sind nicht-wahrnehmbar 2
2.2 Darstellung von Wasserzeichen 3
3 Problemanalyse 3
3.1 Definition von Wasserzeichen 3
3.2 Besonderheiten (Unterschiede) der Medien 4
3.2.1 Bilder 4
3.2.2 Videos 5
3.2.3 Audios 6
3.3 Arten von Bildwasserzeichen 7
3.3.1 Bildbereichsverfahren 7
3.3.2 Spektral-/Frequenzbereichsverfahren 8
4 Lösungsvorschläge 10
4.1 Einbettung in den Blaukanal (Bildbereich) 10
4.2 Wavelet-Blockbasierte Einbettung in den Frequenzbereich 11
4.3 Wavelet-basierte Einbettung in den Frequenzbereich 13
4.4 DCT-basierte Einbettung in den Frequenzbereich 14
5 Umsetzung mit Java im „ImageComparator“ 15
5.1 Graphische Benutzeroberfläche 15
5.2 Struktur 16
5.3 Funktionalitäten 17
5.3.1 Einlesen eines Bildes 17
5.3.2 Erzeugen eines Differenzbildes 18
5.3.3 Schreiben eines Bildes 18
5.4 Ausblick 19
6 Zusammenfassung 19
Quellen ii
Abbildungsverzeichnis iii

Kurzfassung

Wasserzeichen, also nicht-wahrnehmbare, informationstragende, systematische Veränderungen an Medien, sind oft eingesetzte Werkzeuge in der Mediensicherheit. Dabei ist es, zum Beispiel zu Schulungs-, Entwicklungs- und Evaluierungszwecken, häufig wichtig, festzustellen und zu visualisieren, wie genau sich diese auf ein markiertes Medium auswirken. Die Herausforderung liegt dabei darin, Nicht-wahrnehmbares so zu visualisieren, dass es wahrnehmbar wird. Daneben müssen bei der Darstellung zudem die Besonderheiten der verschiedenen Medien beachtet werden und auch sicherheitsrelevante Aspekte berücksichtigt werden. Eine Möglichkeit für visuelle Medien ist das Bilden von Differenzbildern und individuelle Verstärken der Unterschiede. Analog gilt dies für Audios, indem man Differenzen der Frequenzen bildet und diese entweder graphisch darstellt oder als Audio abspielt. Aufgabe dieser Arbeit ist es daher, geeignete Visualisierungen für einige Wasserzeichen zu entwickeln und diese mit Hilfe eines selbstgeschriebenen Java-Programmes - dem „ImageComparator" - zu erzeugen. Später sollen die hier vorgestellten Lösungen im „Fraunhofer Watermarking-Portal" [Watermarking-Portal] implementiert werden, um den Nutzern eben zu Schulungszwecken verständlich zu machen, was mit ihren Medien geschieht.

1 Einleitung

Gerade wegen der steigenden Anzahl an Online-Angeboten, die das illegale Kopieren von Multimedia-Daten erleichtern, gewinnen digitale Wasserzeichen, die Urheberschaften nachweisen oder Kunden identifizieren können, immer mehr an Bedeutung. Und auch Wasserzeichen, die Manipulationen an Datenmaterial nachweisen können, werden aufgrund leicht-bedienbarer und technisch-hochentwickelter Bild- und Videobearbeitungsprogramme immer wichtiger. Sie können aber auch einfach eine Alternative zur Kryptographie sein, weil versteckte Botschaften in die Medien eingebettet werden.

Da man bei der Gegenüberstellung von Ausgangsmedium und markierten Medium keine Unterschiede wahrnehmen kann und auch die Dateigröße sich gar nicht oder nur unwesentlich ändert, stellt sich die Frage, wo und wie genau ein Wasserzeichen das Medium verändert. Deshalb ist es Ziel dieser Arbeit, Lösungsvorschläge für die Visualisierung digitaler Wasserzeichen zu erarbeiten und vorzustellen.

Dazu wird zunächst die allgemeine Problematik, nicht-wahrnehmbare Änderungen zu visualisieren, beschrieben. In der Problemanalyse werden dann digitale Wasserzeichen vorgestellt und ihre verschiedenen Ausprägungen hinsichtlich der Darstellbarkeit analysiert. Danach werden die bei der Visualisierung zu berücksichtigenden Besonderheiten der einzelnen Medien aufgezeigt. Schließlich werden Lösungsvorschläge für die Visualisierung einiger Bildwasserzeichen erläutert und in Java umgesetzt.

2 Problembeschreibung

In diesem Kapitel wird die Problematik beschrieben, dass Wasserzeichen nicht-wahrnehmbar sind, man aber dennoch zu unterschiedlichen Zwecken eine wahrnehmbare Darstellung der Veränderungen, die sie erzeugen, benötigt.

2.1 Wasserzeichen sind nicht-wahrnehmbar

Digitale Wasserzeichen sind nicht-wahrnehmbare Veränderungen, die systematisch Informationen in ein digitales Medium einbetten [Dittmann – Digitale Wasserzeichen]. Nicht-wahrnehmbar bedeutet, dass die Modifizierungen natürlich schon vorhanden und theoretisch auch seh- und hörbar sind, allerdings sind diese so geschickt eingebettet, dass sie nicht auffallen. Konkret bedeutet das, dass die Aussage des Mediums nicht verändert wird, obwohl, vereinfacht gesagt, in Bildern und Videos Pixel auf verschiedenste Weise geringfügig verändert werden und in Audios gleichermaßen Frequenzen.

Dabei macht man sich zu Nutzen, dass das visuelle und auditive System des Menschen auf verschiedene Reize unterschiedlich empfindlich reagieren. So können Muster, die gezielt Blauzapfen anregen, vom Auge nicht so leicht aufgelöst werden wie rote oder grüne Muster [Farbwahrnehmung]. Denn die Empfindlichkeit des Auges gegenüber den kurzwelligen, blauen Lichtreizen, ist relativ gering, da der Anteil der blauempfindlichen Zapfen in der Retina nur bei insgesamt etwa neun Prozent liegt. Hinzu kommt außerdem, dass sie sich nur im peripheren Teil der Retina befinden. Da jedoch die Sehschärfe von der Dichte der Zapfen abhängt, können die roten und grünen Signale stärker wahrgenommen werden, denn sie befinden sich nicht nur peripher, sondern auch in der Fovea, also dem gelben Fleck, der für das zentrale, scharfe Sehen verantwortlich ist. Dabei geht man davon aus, dass es ca. doppelt so viele Rotzapfen wie Grünzapfen gibt. Veränderungen im Blaukanal des RGB-Modells fallen demnach weniger ins Gewicht, als in einem der anderen Kanäle. Deswegen betten manche Algorithmen die Wasserzeichen in den Blaukanal ein.

Man kann aber z. B. auch die Luminanz des Bildes ein wenig verändern und dort die Zusatzinformationen einbetten, denn ein Bild kann als Gestalt betrachtet werden und nach der Gestaltpsychologie haben Gestalten die Eigenschaft, transponierbar zu sein [Verhaltenswissenschaftliche Grundlagen]. Das bedeutet, dass die Gestalt bei einer gleichmäßigen Veränderung erhalten bleibt. Geringe Modifikationen, also zum Beispiel das minimale Aufhellen des Bildes, fallen daher nicht auf.

Nicht-wahrnehmbar sind ebenfalls Audiowasserzeichen, denn sie betten die Informationen so ein, dass kein Unterschied zwischen Ausgangsmedium und markiertem Medium hörbar ist. Konkret heißt das, das Wasserzeichen die Frequenzen ober- und unterhalb der Hörschwelle für die Einbettung ihrer Informationen nutzen können, da diese Bereiche generell nicht hörbar sind. Besser ist es jedoch, sogenannte Maskierungseffekte, auf die in Kapitel 3.2.3 noch näher eingegangen werden soll, auszunutzen, die die Änderungen ebenfalls nicht-wahrnehmbar machen.

2.2 Darstellung von Wasserzeichen

Nun ist es jedoch eben wegen der Nicht-Wahrnehmbarkeit interessant zu wissen, wo genau ein Wasserzeichen Modifikationen im Medium vornimmt. Gerade bei der Entwicklung neuer Algorithmen ist es sehr hilfreich eine Darstellung der Veränderungen analysieren zu können, um nachzuvollziehen, wo der Algorithmus das Medium modifiziert. Somit kann man schließlich auch sichergehen, dass das Verfahren an den richtigen Stellen ansetzt. Und auch bei Schulungen hilft es, den Vorgang des „Digital Watermarking“ verständlich zu machen, indem man visualisiert bzw. hörbar macht, was mit dem Medium geschieht.

Es muss demnach also eine Möglichkeit geschaffen werden, die minimalen Unterschiede zwischen Ausgangsmedium und mit Wasserzeichen markiertem Medium wahrnehmbar zu machen. Eine Möglichkeit dazu ist, die Differenz der beiden Medien zu bilden. Bei Bildern und Videos entstehen so Differenzbilder, auf denen allerdings kaum etwas zu erkennen ist (siehe Abbildung 1). Deshalb müssen die Unterschiede speziell herausgearbeitet und hervorgehoben werden. Ähnlich ist es auch beim Audio, denn bildet man die Differenz zwischen markierten und nicht-markiertem Audio, ist fast nichts zu hören. Auch dort müssen die Unterschiede deutlich verstärkt werden, um z. B. ein Rauschen wahrzunehmen.

Abbildung 1: von links nach rechts: Ausgangsbild, markiertes Bild (Wavelet-Algorithmus), unverstärktes Differenzbild

3 Problemanalyse

Um die vorliegende Aufgabe präzise zu analysieren, wird zunächst der Begriff „Wasserzeichen“ genauer definiert und erläutert, auf welche Art von Medien sie sich anwenden lassen. Die sich daraus ergebenden Unterschiede, die bei einer Visualisierung des Wasserzeichens maßgeblich sind, werden daraufhin analysiert und schließlich auch die verschiedenen Arten von Wasserzeichenalgorithmen beschrieben.

3.1 Definition von Wasserzeichen

„The watermark is a digital code unremovably, robustly, and imperceptibly embedded in the host data and typically contains information about origin, status, and/or destination of the data.“ [Multimedia Watermarking Techniques, S. 1079]. Diese Zusatzinformationen werden mittels verschiedener Algorithmen untrennbar in das Medium eingebettet. Mit Hilfe eines geheimen Schlüssels können diese Informationen, die im Medium „versteckt“ und damit für den Menschen nicht wahrnehmbar sind, später wieder ausgelesen werden. Dabei können die Wasserzeichen sogar robust gegenüber bestimmten Veränderungen am Trägermedium (z. B. Formatwandelung, Skalierung, Abfotografieren etc.) sein.

Das Besondere an Wasserzeichen ist, dass die Zusatzinformationen nicht-wahrnehmbar und damit auch nicht störend sind. Man kann das Medium dadurch wie herkömmlich verwenden und muss sich nicht auf irgendeine Weise einschränken. Gerade deshalb werden sie z. B. als Copyright-Wasserzeichen eingesetzt, um Urheberrechtsverletzungen aufzuspüren oder als Integritäts-Wasserzeichen (so genannte inhalts-fragile Wasserzeichen), die bei Modifikationen des Trägermediums zerbrechen und mit denen Veränderungen am Inhalt nachvollzogen werden können. Darüberhinaus gibt es noch viele andere Anwendungsmöglichkeiten, wie z. B. die Authentizitätswasserzeichen, mit denen die Vertrauenswürdigkeit von Medien sichergestellt werden kann.

Derzeit lassen sich Wasserzeichen auf digitale Multimediadaten wie Bilder, Audios, Videos, Texte (hierbei handelt es sich meist um Text als Bild, wobei es auch Wasserzeichen gibt, die in die Schrift einbetten) und 3-D-Graphiken anwenden. Die vorliegende Arbeit beschränkt sich aufgrund der enormen Menge unterschiedlicher Verfahren bei der Ausarbeitung der Visualisierungsmöglichkeiten auf den Bildbereich.

3.2 Besonderheiten (Unterschiede) der Medien

Alle digitalen Medien haben für sie typische Eigenschaften, die beim Einbetten des Wasserzeichens und, daraus resultierend, bei der Visualisierung berücksichtigt werden müssen.

Da gibt es zum einen die visuellen Medien, also Bilder, Texte, 3-D-Graphiken und Videos. Zum anderen können auch Audios und die Kombination beider Medien – Audio-Videos – markiert werden.

3.2.1 Bilder

Bilder kann man zunächst mit Hilfe der klassischen Merkmale „Farbe, Textur und Kanten“ [Multimedia-Technologie, S. 152] in die Kategorien natürliche und synthetische Bilder unterteilen. Natürliche Bilder sind typischerweise Fotos. Sie enthalten chaotische Strukturen, also viele verschiedene Texturen und sehr viele Farben. Digitale Fotos werden normalerweise in Truecolor-Formaten abgespeichert, da diese die vom Menschen ungefähr erkennbaren 16,7 Millionen Farben darstellen können [Wikipedia, „True Color“]. Synthetische Bilder hingegen sind Graphiken, also mit dem PC erstellt. Sie zeichnen sich dadurch aus, dass sie relativ wenige Farben haben (oft auch indiziert) und eher klare Formen (wenige Texturen) mit starken Kanten als Farbverläufe oder Ähnliches.

Es ist natürlich viel einfacher, ein Bild zu markieren in dem sehr viele unterschiedliche Farbwerte sind, da Wasserzeichenalgorithmen z. B. Farbwerte geringfügig verändern. Bei einem synthetischen Bild, in dem z. B. ein großer roter Kreis vor einem weißen Hintergrund ist, fällt jede kleine Farbveränderung auf. Markiert man jedoch eine fotografierte Blumenwiese, kann man diese Änderung nicht wahrnehmen, weil das Bild einfach zu komplex ist und aus tausenden verschiedenen Farben besteht. Auch die Komplexität der Formen und Strukturen spielt dabei eine große Rolle. Bei dem roten Kreis fällt es auch viel eher als bei ungeordneten Strukturen auf, wenn man an seinem Rand ein paar Pixel verändert, da er dann nicht mehr richtig rund wirkt.

Genau dieses Problem stellt sich daher ebenfalls bei Text, denn ein Buchstabe besteht nur aus ein paar Pixeln der gleichen Farbe vor einem einfarbigen Hintergrund. Meistens sind diese Texte Binärbilder, also lediglich schwarz-weiß. Bei hochqualitativen Textbildern, also z. B. PDF, fallen geringe Änderungen daher sofort auf, da die Buchstaben dort, wo ein Pixelwerte verändert wurden, Löcher zu haben scheinen.

3-D-Graphiken sind ebenfalls synthetische Bilder, jedoch sind in ihnen meistens Farbverläufe, die die Anzahl der Farben erhöhen und damit zumindest einfacher als Texte zu markieren sind.

Ob nun synthetisch oder natürlich – alle diese Bilder lassen sich am besten in Form von Differenzbildern darstellen, da bei der direkten Gegenüberstellung keine Unterschiede wahrnehmbar sind. Differenzbilder sind Bilder, die erzeugt werden, indem der Wert jedes Pixels des markierten Bildes vom entsprechenden Pixelwert des Originalbildes abgezogen wird. Alle Pixel, die in beiden Medien identisch sind, werden so im klassischen Differenzbild schwarz. Alle veränderten Pixel erhalten einen anderen Wert. Da sich die Werte jedoch nur sehr geringfügig unterscheiden, muss das Differenzbild für eine gute Visualisierung je nach Art des Algorithmus verschiedene Werte verstärken und besonders hervorheben. Hinzu kommt, dass bei einem farbigen Differenzbild, die neuen Werte invertiert sind. Da man nicht gewohnt ist, invertierte Bilder zu sehen und sich das Gehirn nicht so einfach zu jeder Farbe die Gegenfarbe vorstellen kann, sind invertierte Bilder für eine Visualisierung von Wasserzeichen unbrauchbar.

In manchen markierten Bildern kann man bei sehr starkem Zoom Artefakte des Wasserzeichenalgorithmus erkennen. Eine weitere Visualisierung kann deshalb sein, einen vergrößerten Bildausschnitt des Originalbilds und den entsprechenden Ausschnitt im markierten Bild gegenüberzustellen. Genauere Ausarbeitungen und spezifische Lösungen für unterschiedliche Algorithmen finden sich in Kapitel 4 – „Lösungsvorschläge“.

3.2.2 Videos

Auch wenn sich Videos wieder auf Einzelbilder zurückführen lassen, gibt es dennoch einige erwähnenswerte Besonderheiten. Schließlich sieht man die Bilder, die im Videobereich „Frames“ genannt werden, nicht einzeln, sondern nimmt sie als Bewegung wahr. Diese so genannte Bewegungskontinuität erfasst das menschliche Auge ab einer Bildwiederholungsrate von 15 Frames pro Sekunde. Der europäische Standard für Bewegtbilder (PAL – Phase Alternating Line) hat daher eine Bildwiederholungsrate von 25 Hz, was 25 Frames pro Sekunde entspricht [Multimedia-Technologie, S. 89/90]. Im Normalfall ist es also recht schwierig, bei dieser Geschwindigkeit geringe Veränderungen wahrzunehmen. Das gilt natürlich besonders für die Teile der Einzelbilder, die sich in jedem darauffolgenden Frame verändern, also Bewegungen im Film an sich. In statischen Teilen, also zum Beispiel einem Hintergrund, nimmt man wechselnde Bildpunkte daher eher wahr.

Ein weiterer Unterschied zu den Bildern aus Kapitel 3.2.1 ergibt sich aus den Geräten, mit denen man typischerweise die Medien betrachtet. Im Unterschied zu heutzutage üblichen PC-Bildschirmen, die eine Mindestauflösung von 1024 * 768 Pixeln haben, können Fernseher für PAL gerade einmal 575 Zeilen [Multimedia-Technologie, S. 151] darstellen, was beim gleichen Seitenverhältnis von 4:3 eine Auflösung von 702 * 575 Pixeln bedeutet. Das heißt,

dass zum Einbetten von Wasserzeichen vergleichsweise viel weniger Pixel zur Verfügung stehen, wenn man von vollformatigen Bildern ausgeht.

Da eine Markierung jedes einzelnen Frames zu aufwändig wäre, beschränkt sich die Anzahl der Frames, in die ein Wasserzeichen eingebettet werden kann, auf die I-Frames (Intra Coded Pictures), da im Normalfall Videos im MPEG-Format markiert werden. I-Frames sind beim MPEG-Standard die Standbilder, also die Frames, die „ohne zusätzliche Informationen bzgl. anderer Einzelbilder kodiert“ [Multimedia-Technologie, S. 152] werden. Denn ein Teil der Komprimierung bei MPEG ist, dass bestimmte Frames voneinander abhängig sind und redundante Bildinformationen eingespart werden.

Da die anderen Frames zum Teil sehr wenige Bildinformationen enthalten, beschränkt man sich bei der Markierung eben auf diese I-Frames, da sie alle Bildinformationen vollständig selbst enthalten und so genügend Bildpunkte für die Einbettung eines Wasserzeichens zur Verfügung stehen. Dieses Bild kann dann theoretisch mit einem üblichen Bildwasserzeichenalgorithmus markiert werden.

Daher sind die Visualisierungsmöglichkeiten für Videos grundsätzlich die gleichen, wie die in Kapitel 3.2.1 genannten für Bilder, denn auch hier machen Differenzbilder und Gegenüberstellungen von vergrößerten Bildabschnitten Sinn. Dabei kann man entweder einzelne Frames gegenüberstellen bzw. einzelne Differenzbilder zeigen oder ganze Sequenzen aus Differenzbildern abspielen.

Da Videos meistens aber nicht nur aus reiner Bildinformation bestehen, sondern auch eine Audiospur haben, kann man auch dort Wasserzeichen einbetten.

3.2.3 Audios

Der größte Unterschied zwischen Audio und einem visuellen Medium liegt nahe: Audios lassen sich nicht direkt visualisieren. Außerdem reagiert das menschliche psychoakustische Modell (auch HAS – „human audible system“ [Multimedia Watermarking Techniques, S. 1097]) viel sensibler auf Signale als das psychovisuelle. Das bedeutet, dass Modifikationen am Ton besser wahrnehmbar sind als an einem Bild.

Generell betten digitale Audiowasserzeichen – wie einige Bildwasserzeichen auch – in den Frequenzbereich ein. Dafür werden die sogenannten „Samples“, die die digitalisierten Schwingungswerte einer Audiodatei sind, in bestimmte Abschnitte – die „Frames“ – eingeteilt. Diese Frames umfassen beispielsweise 6000 Samples.

Mit Hilfe der Fouriertransformation, auf die hier nicht näher eingegangen werden soll, werden die Samples eines Frames wiederum in Frequenzen umgerechnet. Eine Frequenz ist dabei durch Tonhöhe und Lautstärke bestimmt. Pseudozufällig werden nun einige der 2048 Frequenzen eines Frames ausgewählt und in zwei Gruppen aufgeteilt. Soll eine 0 in das Frame eingebettet werden, werden die Frequenzen der ersten Gruppe erhöht und die der zweiten erniedrigt; umgekehrt erfolgt der Vorgang bei Einbettung einer 1.

Auch wenn es sich eigentlich anböte, in die Bereiche einzubetten, die für das menschliche Ohr nicht wahrnehmbar sind, also in die Frequenzen ober- und unterhalb der Hörschwelle, die zwischen 16 Hz und 19 kHz liegt, vermeidet man dies, da Audio-Kompressionsverfahren

die Frequenzen dieser Bereiche eben auf diesem Grunde entfernen und damit auch das Wasserzeichen zerstören könnten.

Stattdessen nutzt man sogenannte Maskierungseffekte. Einer dieser Effekte ist der, dass eine laute Frequenz ihre benachbarten Frequenzen überdeckt. Daher können die benachbarten Frequenzen bis zu einem gewissen Grad nicht-wahrnehmbar modifiziert werden.

Ein weiterer Effekt ist ein zeitlicher, nämlich, dass eine leise Frequenz nach und kurz vor einer lauten Frequenz vom Menschen nicht wahrgenommen wird. Demnach kann man auch sie modifizieren und somit zur Einbettung nutzen [Digitale Wasserzeichen für Audiodaten].

Die Visualisierung eines Audiowasserzeichens könnte daher die Gegenüberstellung der Ausgangsfrequenzen und der markierten Frequenzen sein. Die Frequenzen sollten dabei in einem Koordinatensystem mit Tonhöhe (x) und Lautstärke (y) abgetragen sein. Man könnte dann beide Frequenzbilder getrennt übereinander angeordnet zeigen, was die Unterschiede aber nicht besonders gut wahrnehmbar machen wird oder sie in einem Bild halbtransparent übereinander legen, so dass die Unterschiede besser erkennbar sind. Die Unterschiede könnten dann noch mit einer extra Farbe hervorgehoben werden.

3.3 Arten von Bildwasserzeichen

Da es sehr viele verschiedene Arten von Wasserzeichenalgorithmen für die in Kapitel 3.2 genannten Medien gibt, sollen hier nur einige für Bilder vorgestellt werden. Diese lassen sich grundlegend in Bild- und Spektralbereichsverfahren (auch Frequenzbereichsverfahren genannt) [Wefers – Digitale Wasserzeichen, B-6] einteilen. Alle Wasserzeichen können theoretisch mit Hilfe des psychovisuellen Modells (auch HVS – „human visual model“ [Watermarking Digital Image and Video Data, S. 22]) Bildbereiche herausfinden, in denen Modifikationen besonders wenig auffallen. Das psychovisuelle Modell achtet so z. B. darauf, dass in nahezu einfarbigen Flächen wenn möglich keine Pixel verändert werden. Besonders notwendig ist es natürlich, wenn sich Personen im Bild befinden – es gibt sogar spezielle Algorithmen, die darauf achten, menschliche Gesichter oder andere „Regions of Interest“ (ROI) weitgehend unangetastet zu lassen, weil wir aufgrund unserer Natur Veränderungen dort sehr stark wahrnehmen.

3.3.1 Bildbereichsverfahren

Die Bildbereichsverfahren manipulieren das Bild direkt im Bildbereich (auch „spatial domain“). Das Bild wird demnach so genommen, wie es ist und nicht erst transformiert oder anderweitig besonders für den Algorithmus vorbereitet, um das Wasserzeichen einzubetten. Eine Möglichkeit ist, das Wasserzeichen im Blaukanal des Bildes einzubetten. Dabei werden mit einem „Pseudozufallsgenerator“ [Wefers – Digitale Wasserzeichen, B-5] (pseudozufällig, da Berechnungen durch ein Computer immer deterministisch sind) Pixelpositionen definiert, die verändert werden sollen [Wefers – Digitale Wasserzeichen, B-4/B-5]. Bei diesen Pixeln wird nun jeweils eine geringe Veränderung im Blaukanal des RGB-Wertes vorgenommen, der Rot- und Grünkanal bleiben unangetastet. Ein Pixel, das so z. B. vorher den RGB-Wert (100, 150, 60) hatte, bekommt nun den Wert (100, 150, 62).

Ein weiteres Bildbereichsverfahren ist das Hinzufügen von einem so genannten „Rauschen“ (auch „noise pattern“). Dieses Rauschen ist eine Art Störungsfilter, wie man ihn auch von Bildbearbeitungsprogrammen wie „Adobe Photoshop“ kennt. Dazu werden wie auch beim oben genannten Algorithmus, der in den Blaukanal einbettet, pseudozufällig Positionen im Bild ermittelt, deren Pixel verändert werden. Allerdings wird hierbei nicht speziell der Blaukanal verändert sondern generell die Luminanz – also der Helligkeitswert – des Pixels [Watermarking Digital Image and Video Data, S. 23]. Die Luminanz kann dabei durch jede Änderung in einem, beiden oder allen der drei RGB-Farbkanäle verändert werden.

Zu den Bildbereichsverfahren zählen auch noch die sichtbaren Wasserzeichen, die meist in Form von Logos oder Texten in Bilder eingebracht werden. Bei ihnen werden ausgewählte Pixel gezielt auf bestimmte Farbwerte gesetzt, um eben die gewünschten Formen im Bild zu erzeugen. Auf diese Art von Wasserzeichen soll hier aber nicht näher eingegangen werden, da sie auch ohne besondere Visualisierungen wahrnehmbar sind.

3.3.2 Spektral-/Frequenzbereichsverfahren

Bei den Spektral- oder Frequenzbereichsverfahren wird das Bild, im Gegensatz zu den Bildbereichsverfahren, zunächst in seine spektralen Komponenten zerlegt [Wefers – Digitale Wasserzeichen, B-6]. Diese spektralen Komponenten kann man mit räumlichen Frequenzen vergleichen, die sich im Bild befinden. Eine hohe Frequenz ist komplexer und kontrastreicher als eine niedrige (siehe Abbildung 2) [Verhaltenswissenschaftliche Grundlagen].

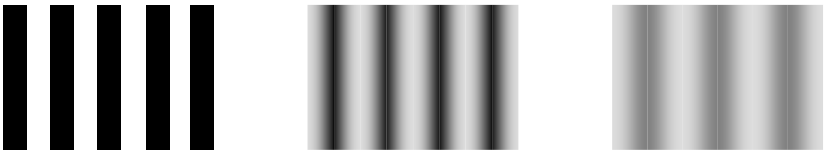

Abbildung 2: von links nach rechts: hohe, mittlere, niedrige räumliche Frequenz

Analysiert man Abbildung 3, kann man sehen, dass der blaue Himmel im Hintergrund sehr einfach, fast einfarbig und recht kontrastarm ist. Er hat demnach eine sehr geringe Frequenz. Betrachtet man nun den Baum, der im Hintergrund den Himmel überlagert, kann man einen Helligkeits- und Kontrastwechsel zwischen Baum und Himmel feststellen, wobei die Konturen sehr weich sind. Man spricht hier von einer mittleren Frequenz. Im Gegensatz dazu hat die Wiese eine relativ hohe Frequenz. Im markierten Bildausschnitt sind die Grashalme ziemlich scharf abgebildet und dunkle und helle Halme wechseln sich ständig ab. Diese häufigen Helligkeits- und Kontrastwechsel mit den scharfen Konturen sprechen für eine hohe Frequenz.

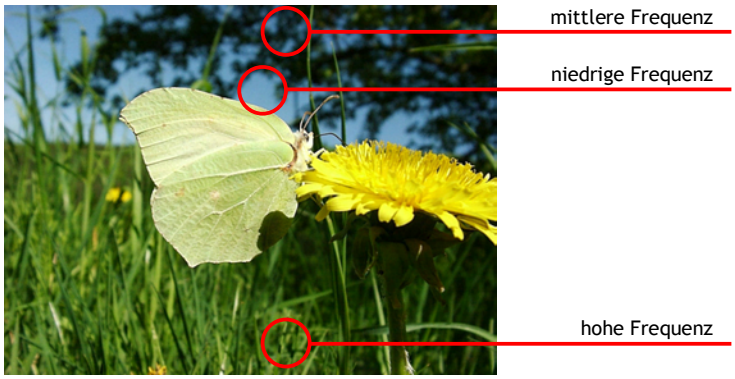

Abbildung 3: Frequenzen in einem Bild

Die Übertragung von Bildern (und auch anderen Multimediadaten) in einen solchen Frequenzbereich erfolgt mit Hilfe von verschiedenen Transformationen. Beispiele hierfür sind die diskrete Cosinus-Transformation und die Wavelet-Transformation.

Die diskrete Cosinus-Transformation (DCT), die eine Variante der Fourier-Transformation ist, bildet die Basis für viele Algorithmen. Aus ihr lassen sich Basisbilder für – typischerweise – 8x8-DCTs ableiten (siehe Abbildung 4).

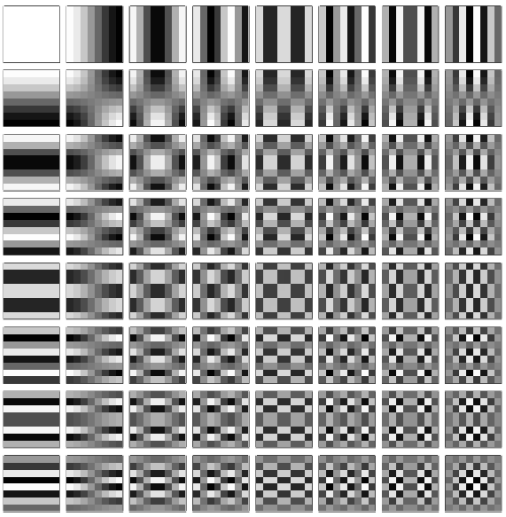

Abbildung 4: Basisbilder einer 8x8-DCT]

Durch Kombination der einzelnen Basisbilder kann man jede 8x8-DCT darstellen. Diese sogenannte 8x8-DCT ist eine Matrix bestehend aus 64 Basisbildern, die bei der Einbettung von Wasserzeichen typischerweise genutzt wird. Es gibt auch andere DCT-Matrizen – die Größe bestimmt dabei die spektrale Auflösung des Bildes.

Um nun ein Bild mit der diskreten Cosinus-Transformation im Frequenzbereich darzustellen, wird für dieses eine DCT-Koeffizientenmatrix errechtet. Diese ergibt sich aus der Linearkombination der Basisbilder zu einem konkreten Bild. Dabei ist jedem Basisbild ein DCT-Koeffizient zugeordnet. Die DCT-Koeffizienten bestimmen daher die spektrale Zusammensetzung des Bildes und lassen sich nicht bestimmten Pixeln zuordnen, da sich die Transformation auf das Gesamtbild beziehen.

Das eigentliche Wasserzeichen wird dann durch Modifikationen der Koeffizienten eingebettet, die die spektralen Anteile des Bildes gezielt verändern. Dabei sucht sich der Algorithmus die mittel- bis hochfrequenten Bereiche aus, da in den niederfrequenten Regionen, wie schon in Kapitel 3.2.1 beschrieben, Artefakte leichter wahrnehmbar sind [Wefers – Digitale Wasserzeichen, B-6/B-7].

4 Lösungsvorschläge

Da nun die Besonderheiten der einzelnen Medien und die grundsätzlichen Verfahren des „Digital Watermarking“ bekannt sind, folgen nun einige Beispiellösungen, wie Wasserzeichen visualisiert werden können. Dazu werden zunächst die ausgewählten Algorithmen kurz beschrieben und zu jedem dieser Algorithmen der zugehörige Lösungsvorschlag für eine Visualisierung dargestellt und erklärt.

4.1 Einbettung in den Blaukanal (Bildbereich)

Der hier verwendete Algorithmus ist ein Bildbereichsverfahren. Er verändert ausschließlich Werte im blauen Farbkanal, da Veränderungen in diesem Kanal – wie in Kapitel 2.1 erklärt – weniger gut wahrnehmbar sind. In Abbildung 5, die Originalbild, markiertes Bild und ein absolutes Differenzbild, auf dem alle Pixel, die nicht verändert wurden, schwarz sind, zeigt, sieht man, dass diese Veränderungen fast das ganze Bild betreffen, also wie eine Art Störungsfilter auf ihm liegen. Ausnahmen finden sich v. a. bei der gelben Blüte der Blume, die nur teilweise modifiziert wurde. Außerdem beweist die Gegenüberstellung von Ausgangsbild und markiertem Bild, dass die Unterschiede tatsächlich nicht-wahrnehmbar sind.

Abbildung 5: von links nach rechts: Ausgangsbild, markiertes Bild, absolutes Differenzbild

Da das absolute Differenzbild nur zeigt, wo etwas verändert wurde, braucht man auch ein Differenzbild, das darstellt, was genau am Bild verändert wurde. Da der Algorithmus die Farbe zur Einbettung nutzt, ist es sinnvoll sich den Farbbereich des Bildes genauer anzusehen. Beim Differenzbild in Abbildung 6 wird für jeden Bildpunkt geprüft, in welchen der drei RGB-Farbkanäle der Unterschied zwischen den beiden Bildern maximal ist. Der entsprechende Pixel im Differenzbild erhält dann die Farbe des Kanals mit dem maximalen Unterschied, wird also Rot, Grün oder Blau. Sind zwei Kanäle gleich stark verändert worden und ist diese Differenz größer als die des übrig bleibenden Kanals, bekommt das Pixel im Differenzbild die Farbe der Summe der beiden Kanäle. Sind Rot und Grün also gleich und maximal, wird das Pixel Gelb. Analog ergibt Rot und Blau Magenta und Blau und Grün Cyan. Sind alle drei Kanäle um denselben Wert verändert worden, wird das Pixel grau. Pixel, die in Originalbild und markiertem Bild gleich sind, bekommen eine Hintergrundfarbe, die standardmäßig Schwarz ist. Nach dieser Methodik ergäbe ein Pixel aus

einem Originalbild mit dem Wert (100, 150, 60) und im markierten mit dem Wert (101, 149, 63) so beispielsweise einen blauen Pixel im Differenzbild, da die Differenz in diesem Kanal maximal ist.

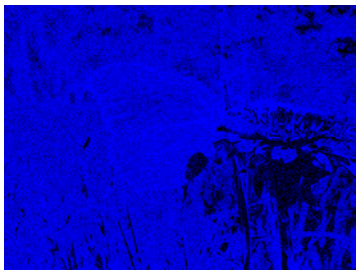

Abbildung 6: Farbintensiviertes Differenzbild für maximale Farbkanaländerungen

Wie man schon in Abbildung 6 erkennt, wurde ausschließlich in den blauen Farbkanal eingebettet. Das beweist auch Abbildung 7, in der intensivierte Differenzbilder für Änderungen im roten, grünen und blauen RGB-Kanal zu sehen sind. In diesen Abbildungen sieht man auch, wie und wie stark der Wert verändert wurde. Dazu wird die maximale Differenz aller Werte zwischen Originalbild und markiertem Bild im jeweiligen Farbkanal berechnet. Eine Art Tonwertspreizung sorgt dafür, dass die Pixel, die die maximale Differenz erzeugen, auch die maximalen Farbwerte haben (weiß bzw. schwarz). Alle anderen Differenzwerte liegen gleichmäßig verteilt zwischen den Extremen, so dass alle Zwischenfarben Abstufungen der Farbe des Farbkanals sind. So reicht die Spannweite des Blau-Kanal-Bildes beispielsweise von Weiß über Hellblau, Blau, Dunkelblau bis hin zu Schwarz, wobei positive Differenzen ein helleres und negative ein dunkleres Blau erzeugen. Unveränderte Pixel erhalten die Hintergrundfarbe Grau.

Abbildung 7: von links nach rechts: intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal

Zu beachten ist hierbei, dass der Blau-Kanal des RGB-Models alleine lediglich von einem voll gesättigten Blau bis hin zu Schwarz reicht. Bei dieser Differenzbildart werden die übrigen Farbwerte daher aus den anderen beiden Farbkanälen gemischt, um einen Hell-Dunkel-Effekt erzeugen zu können, der anzeigt, ob ein Pixelwert ab- oder zugenommen hat. Notwendig ist diese Tonwertspreizung außerdem, um die eigentlich minimalen Unterschiede in der Farbintensität überhaupt wahrnehmen zu können.

4.2 Wavelet-Blockbasierte Einbettung in den Frequenzbereich

Der folgende Algorithmus bettet das Wasserzeichen mit Hilfe der Wavelet-Transformation in den Frequenzbereich ein. Die Veränderungen erstrecken sich über das gesamte Bild, wobei hauptsächlich die Helligkeitswerte des Bildes verändert werden. Das Besondere an

ihm ist außerdem, dass die Einbettung blockbasiert ist. Schon im absoluten Differenzbild in Abbildung 8 kann man diese 32 mal 32 Pixel großen Blöcke erkennen.

Abbildung 8: von links nach rechts: Ausgangsbild, markiertes Bild, absolutes Differenzbild

Um die Blöcke stärker hervorzuheben, ist es sinnvoll den Kontrast des Bildes zu maximieren. Das kontrastmaximierte Differenzbild in Abbildung 9 beruht auf dem Prinzip, alle Pixel, deren Differenz aus Ausgangsbild und markiertem Bild positiv ist, die also heller geworden sind, weiß werden zu lassen und umgekehrt alle mit negativer Differenz, also alle dunkler gewordenen Pixeln, schwarz zu machen. Die Pixel, die in beiden Bildern identisch sind, werden grau.

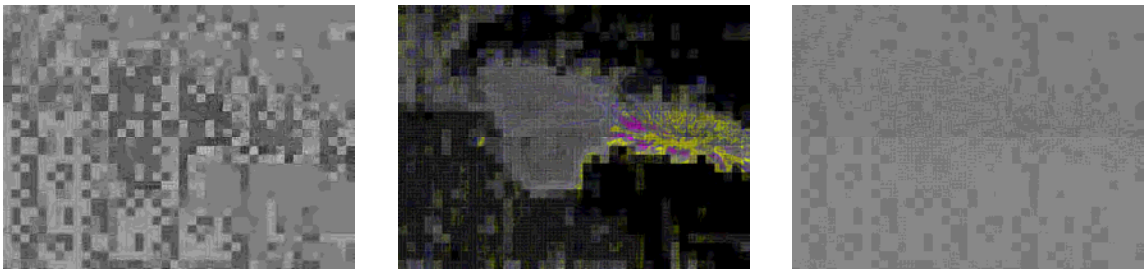

Abbildung 9: von links nach rechts: kontrastmaximiertes Differenzbild, farbintensiviertes Differenzbild für maximale Farbkanaländerungen, intensiviertes Differenzbild für Luminanzänderungen

Dass die Luminanz verändert wurde, erkennt man im Differenzbild für maximale Farbkanaländerungen in Abbildung 9, bei dem – wie schon in Kapitel 4.1 beschrieben – die Pixel die Farbe des Farbkanals bzw. der beiden Farbkanäle mit der größten Differenz erhalten, identische Pixel die Hintergrundfarbe Schwarz tragen und alle in jedem Kanal gleichmäßig veränderten Pixel grau dargestellt werden. Man sieht deutlich, dass hier nicht nur ein bestimmter, sondern mehrere Farbkanäle, z. T. auch gleichmäßig um denselben Wert, modifiziert wurden. Zu erkennen ist das daran, dass zum Beispiel der Flügel des Schmetterlings im Differenzbild grau ist. Da dieser im Ausgangsbild annähernd weiß ist, macht es Sinn alle Kanäle gleichmäßig zu verändern, da das nur den Grauwert des Flügels verändert und ihm keinen Farbstich gibt. Ähnlich ist es mit der Blüte der Blume, die auch im Differenzbild gelb ist, weil der rote und grüne Kanal gleichmäßig geändert wurden, um ein helleres oder dunkleres Gelb zu erzeugen. Auffällig ist auch, dass die Bereiche, die im Originalbild sehr dunkel sind, nicht verändert wurden, da sie im Differenzbild schwarz sind. Man kann außerdem erkennen, dass wenn nur in einen Farbkanal eingebettet wurde, dies überwiegend der blaue war, für den wir Menschen – wie bereits in Kapitel 2.1 erwähnt – relativ unempfindlich sind.

Das intensivierte Differenzbild für Luminanzänderungen aus Abbildung 9 zeigt nun die Intensität und Art der Veränderung der Helligkeitswerte, also der Luminanz. Es ist ein reines Graustufendifferenzbild, das einer Tonwertspreizung ähnlich der aus Kapitel 4.1

unterzogen wurde, damit die größte Wertveränderung auch in der maximalen Intensität dargestellt werden kann. Die Abstufungen der Grautöne richten sich somit wieder nach der maximalen Differenz, die im gesamten Bild gefunden wird.

4.3 Wavelet-basierte Einbettung in den Frequenzbereich

Das folgende Wasserzeichen beruht ebenfalls auf der Wavelet-Transformation und wird somit wiederum im Frequenzbereich eingebettet und verändert die Luminanz des Bildes. Die beiden Algorithmen unterscheiden sich jedoch dadurch, dass der nun folgende nicht blockbasiert ist und daher keine Blockstrukturen aufweist. Außerdem ist er besonders robust gegenüber einscannen und abfotografieren, weswegen der Rand des Bildes in einigen Zentimetern Dicke nicht markiert. Das verhindert, dass Teile des Wasserzeichens nach ungenauem Digitalisieren (z. B. leicht schiefem Einscannen), bei der Bildbearbeitung weggeschnitten werden. Diesen unmarkierten Rand sieht man im absoluten Differenzbild in Abbildung 10 sehr deutlich, bei dem alle unveränderten Pixel schwarz sind. Dort ist außerdem zu erkennen, dass sich die Veränderungen innerhalb des markierten Bereiches sehr gleichmäßig, jedoch unstrukturiert verteilen.

Abbildung 10: von links nach rechts: Ausgangsbild, markiertes Bild, absolutes Differenzbild

Mit Hilfe des kontrastmaximierten Differenzbildes in Abbildung 11 lässt sich dennoch eine gewisse Regelmäßigkeit der Formen erkennen, wie man v. a. im vergrößerten Ausschnitt des Differenzbildes erkennt.

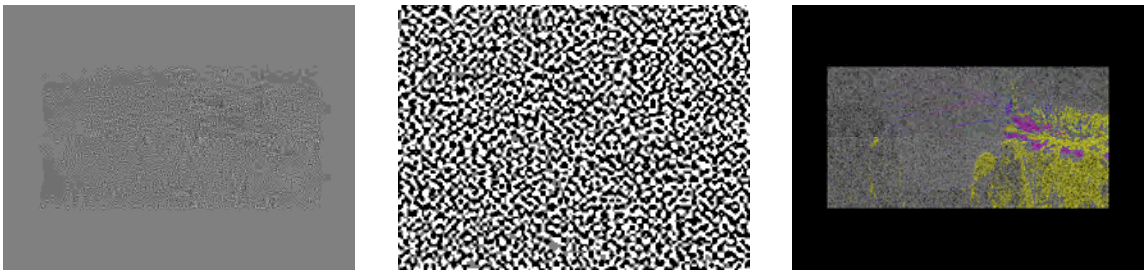

Abbildung 11: von links nach rechts: kontrastmaximiertes Differenzbild, vergrößerter Ausschnitt des linken Differenzbildes, Differenzbild für maximale Farbkanaländerungen

Interessant ist auch das Differenzbild für maximale Farbkanaländerungen für dieses Bild, denn man erkennt, dass im Gegensatz zum blockbasierten Wavelet-Algorithmus des vorigen Kapitels, auch die dunklen Bereiche des Bildes verändert wurden. Dennoch sind sich beide Bilder im Groben sehr ähnlich, was die Farbanteile zeigen.

Auffällig ist, dass überwiegend alle drei Farbkanäle gleichmäßig verändert wurden (im Differenzbild für Farbkanaländerungen Grau). Bei den 2-Kanal-Veränderungen wurde vor

allem der rote in Kombination mit dem grünen Farbkanal (im Differenzbild gelb) modifiziert.

Bestätigt wird das auch durch die Differenzbilder in Abbildung 12, wo die Blume im intensivierten Differenzbild für Änderungen im Blau-Kanal nahezu ausgespart ist, in denen des roten und grünen Kanals jedoch nicht.

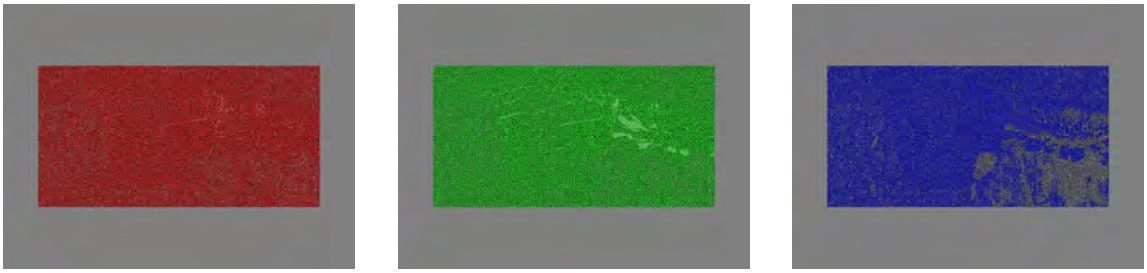

Abbildung 12: von links nach rechts: intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal

4.4 DCT-basierte Einbettung in den Frequenzbereich

Der nun folgende Algorithmus basiert auf der diskreten Cosinus-Transformation und bettet die Zusatzinformationen somit wie der Wavelet-basierte Algorithmus ebenfalls in den Frequenzbereich ein. Die Veränderungen erstrecken sich dabei wieder über das komplette Bild, wie man dem absoluten Differenzbild in Abbildung 13 entnehmen kann und sind scheinbar nicht blockartig strukturiert.

Abbildung 13: von links nach rechts: Ausgangsbild, markiertes Bild, absolutes Differenzbild

Bei genauerer Betrachtung kann man dennoch einige 8x8-Blöcke der DCT-Transformation erkennen, die man besonders im vergrößerten Ausschnitt des intensivierten Differenzbildes für Luminanzänderungen in Abbildung 14 erkennen kann. Dieses Differenzbild ist hierfür besonders gut eignet, um diese Artefakte, die in ähnlicher Weise auch bei der JPEG-Kompression, die ebenfalls auf die DCT-Transformation aufbaut, entstehen, sichtbar zu machen.

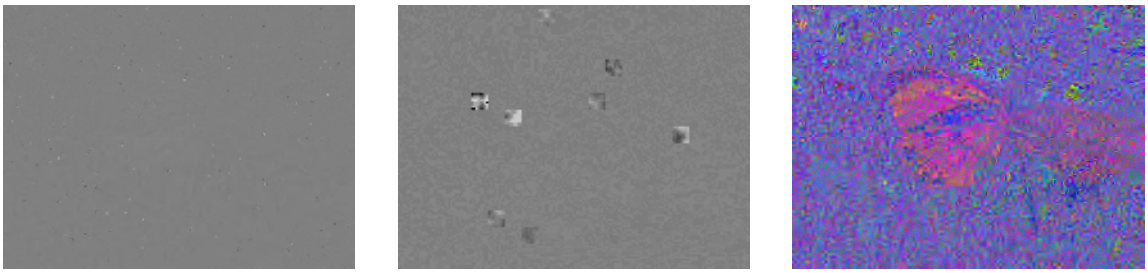

Abbildung 14: von links nach rechts: intensiviertes Differenzbild für Luminanzveränderungen, vergrößerter Ausschnitt des linken Differenzbildes, Differenzbild für maximale Farbkanaländerungen

Da es sich um eine Einbettung in den Frequenzbereich handelt, sind die RGB-Werte relativ wahllos verändert worden. Wo die Veränderungen der einzelnen Farbkanäle beim wavelet-basierten Algorithmus aus Kapitel 4.3 noch recht gleichmäßig wirken, scheinen sie bei diesem Algorithmus sehr willkürlich, wie man deutlich am Differenzbild für maximale Farbkanaländerungen in Abbildung 14 sehen kann.

Die Auszüge aus den einzelnen Farbkanälen in Abbildung 15 bestätigen dies. Gleichzeitig scheinen der rote und blaue RGB-Kanal eher negative Differenzen erzeugt zu haben, was aber nicht zu interpretieren ist, da sich der Algorithmus bei der Berechnung der Pixelwerte ja immer auf das ganze Bild bezieht und nicht auf einzelne Pixel. Es jedoch interessant zu sehen, dass man bei diesen Differenzbildern immer noch recht deutlich Umrisse und bestimmte Formen aus dem Originalbild wieder erkennt.

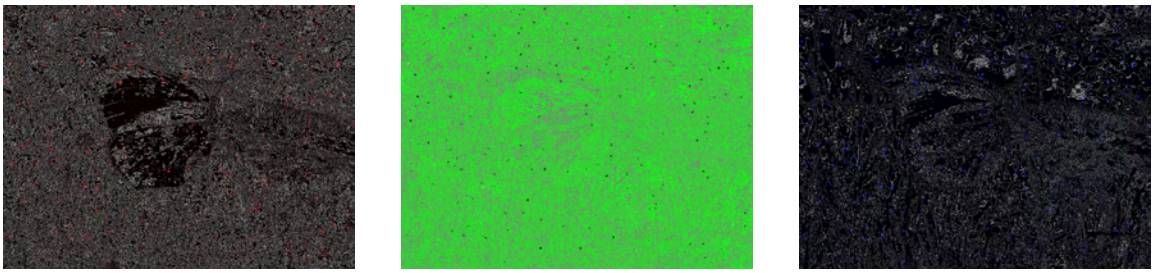

Abbildung 15: von links nach rechts: intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal

5 Umsetzung mit Java im „ImageComparator“

Nachdem die wesentlichen Arten von Differenzbildern anhand der Algorithmen präsentiert wurden, wird nun das Programm, mit dem diese Differenzbilder erstellt wurden, vorgestellt. Hierbei soll auf die graphische Benutzeroberfläche, die Struktur und die Funktionalitäten eingegangen werden. Im Zusammenhang der Funktionalitäten wird kurz auf das Einlesen eines Bildes eingegangen und dann das Kreieren eines Differenzbildes bis hin zum Schreiben auf den Datenträger erläutert.

5.1 Graphische Benutzeroberfläche

Das Java-Programm, das die Differenzbilder dieser Arbeit erzeugt hat, ist der „ImageComparator“. Es hat eine graphische Benutzeroberfläche, die mit dem Java-Paket „Swing“ erstellt wurde. Wie man Abbildung 16 entnehmen kann, stellt diese Benutzeroberfläche

eine Menü- und Symbolleiste zum Interagieren bereit. Die Arbeitsfläche ist dreigeteilt und zeigt die Pfade der aktuell geladen Bilder und Vergleichsbilder sowie der damit erzeugten Differenzbilder an. Diese Pfade sind dabei interaktiv und öffnen beim Anklicken jeweils ein internes Fenster, in dem das ausgewählte Bild zu sehen ist. Außerdem kann der Nutzer dem Feld am unteren Bildschirmrand Informationen über den Programmstatus und eventuelle Warnungen entnehmen. Hilfestellung zur Programmbenutzung und Informationen über die unterschiedlichen Arten von Differenzbildern befinden sich am rechten Bildschirmrand. Die Hilfe bietet dabei einen einfachen Einstieg in das Programm („Getting started“) und einen vertieften Überblick („Tutorial“) über alle Funktionen inklusive aller Einstellungsmöglichkeiten, die der Nutzer individuell vornehmen kann und die dann auch nach Programmneustart gültig sind.

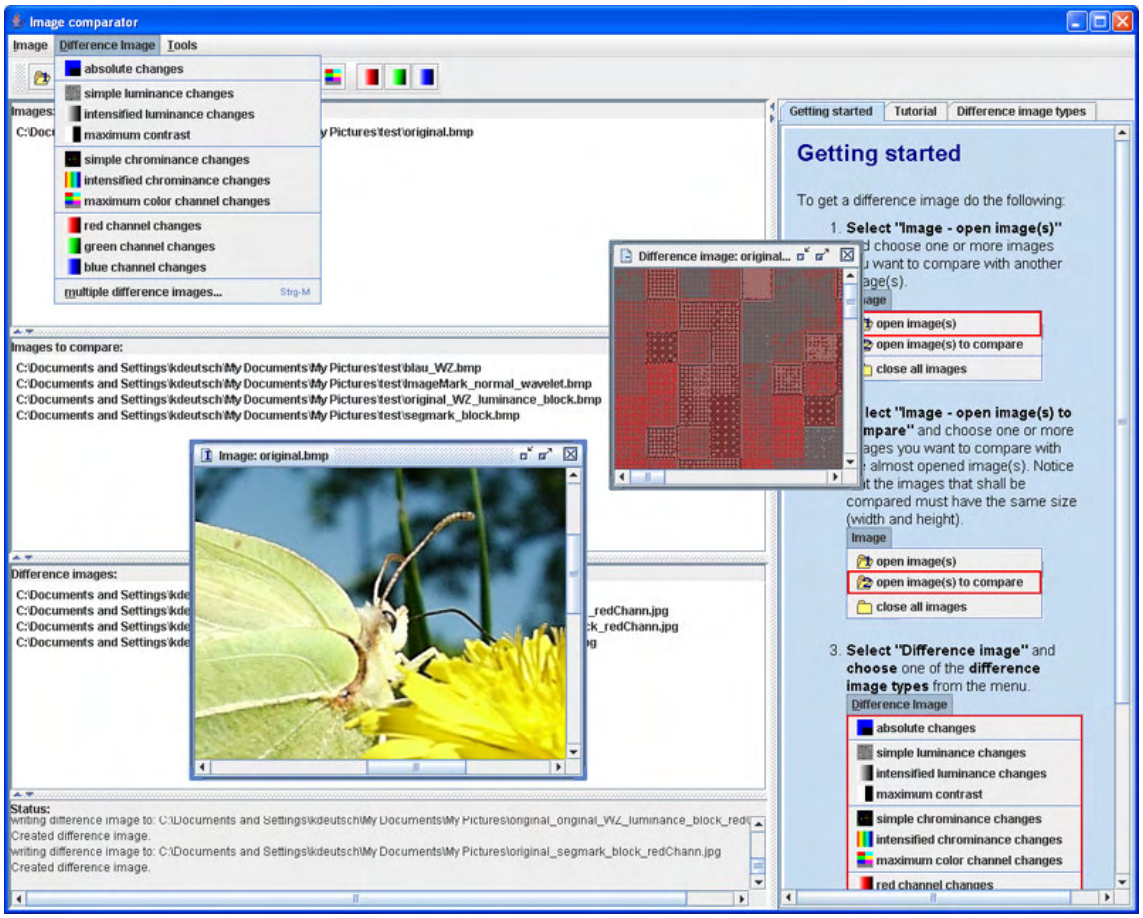

Abbildung 16: graphische Benutzeroberfläche des „ImageComparator“

5.2 Struktur

Der grundlegende, strukturelle Aufbau des „ImageComparator“ ist dabei der folgende (siehe Abbildung 17):

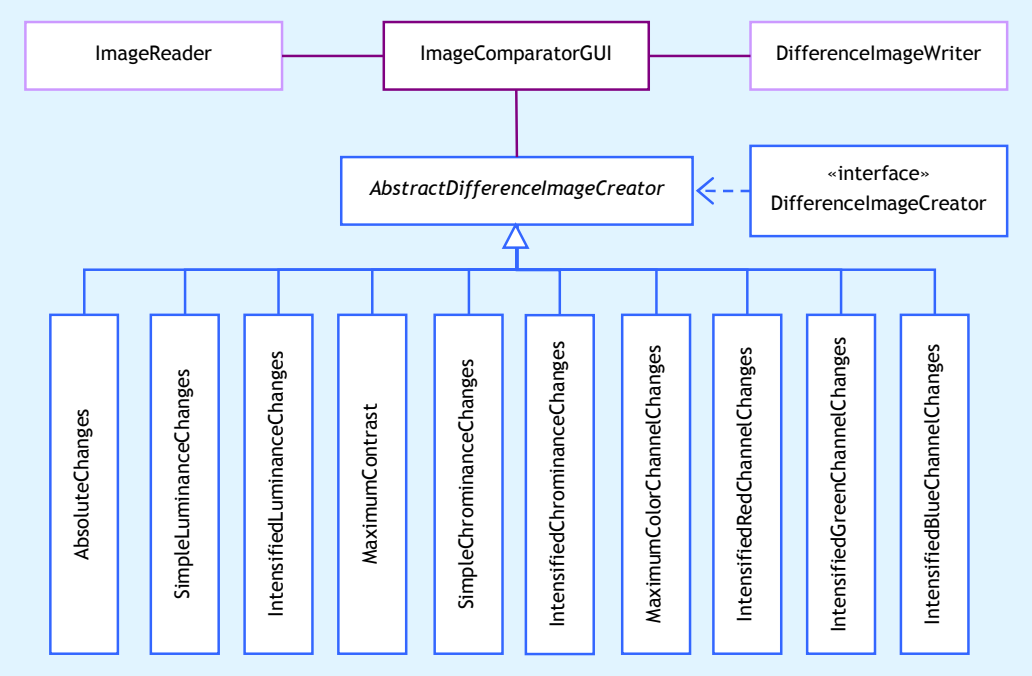

Abbildung 17: vereinfachtes Klassendiagramm des ImageComparator

Da die unterschiedlichen Differenzbildarten vieles gemeinsam haben, ist es sinnvoll, für jede Differenzbildart zwar eine selbständige Klasse anzulegen, diese Klassen aber von einer gemeinsamen Oberklasse erben zu lassen. Diese Oberklasse implementiert ein Interface, das einige Methoden für einen Differenzbildkreator vorschreibt. Da man mit der Oberklasse kein konkretes Differenzbild erzeugen kann, ist sie abstrakt.

Diese Vererbungsbeziehung erleichtert es außerdem, das Programm jederzeit auf einfache Weise um neue Differenzbildarten zu erweitern, da beim Erzeugen der Differenzbilder die entsprechenden Methoden auf einem Objekt der Oberklasse, hinter dem sich ein spezieller Differenzbild-Kreator verbirgt, aufgerufen werden können.

Durch die Trennung von Funktionalität und Darstellung, ist es weiterhin jederzeit möglich die Darstellung auszutauschen.

5.3 Funktionalitäten

Die eigentlichen Funktionalitäten des „ImageComparator“ sind dabei das Einlesen von Bildern und Vergleichsbilder, das Erstellen von Differenzbildern und das Schreiben dieser auf einen Datenträger. Das Zusammenspiel der dafür zuständigen Klassen wird derzeit von der bereits im Kapitel 5.1 beschriebenen graphischen Benutzeroberfläche arrangiert, die in Abbildung 17 nur grob als „ImageComparatorGUI“ aufgeführt ist. Sie entscheidet, wann etwas geladen, gespeichert oder erzeugt wird.

5.3.1 Einlesen eines Bildes

Das Lesen der Bilder übernimmt hierfür die Klasse „ImageReader“, die es neben dem Einlesen von einfachen Bilddateien beispielsweise auch ermöglicht, Bilder aus den in einer

Textdatei gespeicherten Pfaden zu erzeugen oder alle Bilder aus einem angegebenen Ordner zu erzeugen.

Selbstverständlich gibt es dabei viele verschiedene Arten, ein Bild mit Java zu erzeugen. Generell bietet Java neben dem Paket „java.io“, das Klassen für den allgemeinen Umgang mit Dateien, wie Einlesen und Schreiben, ermöglicht, auch ein spezielles Paket für den Umgang mit Bildern, nämlich „javax.imageio“.

Der „ImageReader“ nutzt diese beiden Pakete und erhält zum Einlesen von der Benutzeroberfläche ein oder mehrere File-Objekt(e) der Klasse „java.io.File“. Sieht man vom Ausnahmefall, das eine Textdatei oder ein Ordner ausgelesen werden, ab, dann wird dieses File-Object nun der statischen Methode read(File input) der Klasse „javax.imageio.ImageIO“ übergeben, die als Rückgabewert ein „java.awt.image.BufferedImage“ liefert. „BufferedImages“ haben den Vorteil, dass sie einen pixelgenauen Zugriff auf das Bild erlauben. Mit zwei dieser „BufferedImages“ kann nun ein Differenzbild erzeugt werden.

5.3.2 Erzeugen eines Differenzbildes

Die generelle Vorgehensweise beim Erstellen eines Differenzbildes ist dabei, Pixel für Pixel über zwei Bilder gleichzeitig zu iterieren und von beiden Bildpunkten jeweils den RGB-Farbwert auszulesen. Diese Werte werden dann einem neuen „BufferedImage“ zugewiesen. Für das pixelgenaue Auslesen des RGB-Wertes stellt „BufferedImage“ die Methode getRGB(int x, int y) bereit, der man die Koordinaten des Pixels mitteilt. Sie gibt eine gemeinsame int-Interpretation des RGB-Wertes zurück. Über diesen ist es dann auch möglich die Werte der einzelnen Farbkanäle zu erhalten.

Im weiteren Verlauf unterscheiden sich die Differenzbildarten: graustufige Differenzbilder extrahieren die Grauwerte der Pixel, andere Differenzbilder lesen die einzelnen Farbkanäle aus und manche wiederum benutzen den RGB-Wert so wie er ist. Grundsätzlich wird dann die Differenz der beiden Werte gebildet. Anhand dieser Differenz entscheidet die erzeugende Klasse, wie der Pixel an dieser Koordinate des Differenzbildes auszusehen hat. Differenzbilder mit einem Hell-Dunkel-Effekt erhalten z. B. bei einer positiven Differenz ein Pixel, das heller als die festgelegte Grundfarbe ist, analog bei einer negativen Differenz ein dunkleres.

Wird eine Tonwertspreizung benötigt, werden schon bei Beginn der Erzeugung die maximalen Wert-Differenzen beider Bilder errechnet. Anhand dieses Maximalwertes wird dann ein Faktor bemessen, der den Farb-Wert des neuen Bildpunktes und vor allem dessen Intensität bestimmt.

Nachdem der Farbwert feststeht, wird dieser dem Differenzbild an der besagten Stelle zugewiesen. Sind alle Pixel verarbeitet, kann das Bild nun auf einen Datenträger geschrieben werden.

5.3.3 Schreiben eines Bildes

Auf den Datenträger werden die erzeugten Differenzbilder mit der Klasse „DifferenceImageWriter“ geschrieben. Sie hat zum komfortablen Schreiben von Bildern auch die Fähig-

keit mit unterschiedlichen Pfadangaben umzugehen. Bei Angabe eines Ordners als Ausgabeort werden beispielsweise alle Dateinamen automatisch generiert.

Generell erzeugt der „DifferenceImageWriter“ zum Schreiben des Differenzbildes einen „javax.imageio.stream.FileImageOutputStream“, dem der Dateipfad des Zielortes übergeben wird. Dieser wird benötigt, um ihn einem „javax.imageio.ImageWriter“ als Ort für die Ausgabe zu setzen. Dem „javax.imageio.ImageWriter“ kann man außerdem mitteilen, welches Bildformat er erstellen soll, was über einen Einstellungsdialog der Benutzeroberfläche des „ImageComparator“ auch festgelegt ist. Schließlich wird das Bild mit der Methode write(BufferedImage image) des „javax.imageio.ImageWriter“ auf den Datenträger geschrieben und der Ausgabestrom geschlossen.

Das Differenzbild ist damit fertiggestellt und kann betrachtet werden und hilft damit, die Änderungen, die Wasserzeichen an digitalen Bildern vornehmen, zu veranschaulichen.

5.4 Ausblick

Aufgrund der Vielzahl an unterschiedlichen Wasserzeichen-Algorithmen und deren verschiedenartigen Artefakte, müssen neue Differenzbildarten auf einfachem Wege zu entwickeln und zu implementieren sein. Grundlage dafür bietet der „ImageComparator“ mit seiner in Kapitel 5.2 beschriebenen Struktur.

Als zukünftigen Einsatz des Programms könnte man sich weiterhin vorstellen, neben dem reinen Erstellen von Differenzbildern, herauszufinden, ob ein Bild überhaupt ein Wasserzeichen enthält oder nicht. Nebenbei entstehen beim Vergleich zweier willkürlich gewählter Bilder recht eindrucksvolle und manchmal schon fast künstlerisch anmutende Differenzbilder, die auch dem nicht Wasserzeichen-affinen Menschen Freude bereiten können.

Im Themenfeld der Wasserzeichen ist jedoch auch die Sicherheitsfrage nicht zu vernachlässigen, denn Differenzbilder legen die Algorithmen nahezu offen. Dadurch ist die Sicherheit dieser nach dem heutigen Stand der Entwicklung noch gefährdet, denn die Sicherheitsstandards sind noch nicht soweit entwickelt, wie das beispielsweise in der Verschlüsselungstechnik der Fall ist.

6 Zusammenfassung

Auch wenn es ganz verschiedene Arten von digitalen Wasserzeichen gibt – ob für Bild, Text, Video oder Audio – eines haben sie im Allgemeinen gemeinsam: sie sind zwar im Medium vorhanden, aber für den Menschen nicht wahrnehmbar. Diese Nicht-Wahrnehmbarkeit macht ihre Visualisierung zu einer Herausforderung, der sich diese Arbeit für den Bereich der digitalen Bildwasserzeichen angenommen hat.

Die Algorithmen arbeiten dabei entweder im Bild- oder im Frequenzbereich des Bildes. Während im Bildbereich die RGB-Werte einzelner Pixel modifiziert werden, beziehen sich Veränderungen im Frequenzbereich immer auf das Bild als Gesamtheit. Innerhalb beider Bereiche gibt es verschiedene Verfahren, zum Beispiel das Einbetten in den Blaukanal

(Bildbereich) oder die Einbettung per Modifikation der spektralen Anteile des Bildes auf Grundlage der Wavelet- oder DCT-Transformation (Frequenzbereich).

Da Differenzbilder sehr gut geeignet sind, um Wasserzeichen zu visualisieren, die geringen Unterschiede zwischen Ausgangsmedium und markiertem Bild allerdings unverstärkt nicht verdeutlicht werden können, wurden verschiedene intensivierte Differenzbildarten erarbeitet und anhand einiger dieser Wasserzeichenalgorithmen erklärt und erläutert. Die Typen von Differenzbildern arbeiten dabei beispielsweise mit Tonwertspreizungen, Auszügen aus einzelnen RGB-Kanälen und Kontrastmaximierungen.

Erzeugt wurden diese Differenzbilder mit dem „ImageComparator“, einem selbstentwickelten Java-Programm, das eine graphische Benutzeroberfläche, verschiedene Differenzbildarten und die Funktionalitäten zum Erstellen dieser, bietet. Außerdem ermöglicht seine Struktur, neue Differenzbildarten auf einfache Weise zu kreieren und zu implementieren. Damit sind auch in Zukunft weitere Einsatzgebiete denkbar.

Quellen

[Digitale Wasserzeichen für Audiodaten]
Martin Steinebach, „Digitale Wasserzeichen für Audiodaten“, Dissertationsschrift, Shaker Verlag Aachen, 2004

[Dittmann – Digitale Wasserzeichen]
Jana Dittmann, „Digitale Wasserzeichen“, Springer Verlag, 2000

[Farbwahrnehmung]
Karl R. Gegenfurtner, „Farbwahrnehmung“, http://www.allpsych.uni-giessen.de/karl/teach/farbe.html, 2005-12-12

[Multimedia-Technologie]
Ralf Steinmetz, „Multimedia-Technologie“ (3. überarbeitete Auflage), Springer, 2000

[Multimedia Watermarking Techniques]
Frank Hartung, Martin Kutter, „Multimedia Watermarking Techniques”, Proceeding of the IEEE, Vol. 87, No. 7, July 1999

[Verhaltenswissenschaftliche Grundlagen]
Dr. Prof. Thomas Wirth, „Verhaltswissenschaftliche Grundlagen Teil I“, BA-Mosbach, Vorlesung Sommersemester 2005

[Watermarking Digital Image and Video Data]
Gerhard C. Langelaar, Iwan Setyawan, Reinald L. Lagendijk, “Watermarking Digital Image and Video Data, A State-of-the-Art Overview”, IEEE SIGNAL PROCESSING MAGAZINE, September 2000

[Watermarking-Portal]
Katja Deutschmann, „Generische Ablaufsteuerung einer Web-Applikation am Beispiel ‚Wasserzeichen-Portal’“, Praxisbericht WS04, 2004; http://watermarkingportal.ipsi.fraunhofer.de

[Wefers – Digitale Wasserzeichen]
Frank Wefers, „Digitale Wasserzeichen“, erschienen im Rahmen des Konferenzseminars „Verlässliche Verteilte Systeme“, RWTH-Aachen, Sommersemester 2004

[Wikipedia, „Hörschwelle“]
„Hörschwelle“, http://de.wikipedia.org/wiki/H%C3%B6rschwelle, 2005-12-12

[Wikipedia, „Licht“]
„Licht“, http://de.wikipedia.org/wiki/Licht, 2005-12-12

[Wikipedia, „True Color“]
„True Color”, http://en.wikipedia.org/wiki/True_color, 2005-12-12

Abbildungsverzeichnis

Anmerkung: Alle Abbildungen außer Abbildung 4 wurden im Rahmen dieser Arbeit selbst erstellt und tragen daher keinen gesonderten Quellenvermerk.

[1] Ausgangsbild, markiertes Bild und unverstärktes Differenzbild einer Wavelet-basierten Einbettung in den Frequenzbereich

[2] Veranschaulichung hoher, mittlerer und niedriger räumlicher Frequenzen

[3] Frequenzen in einem Beispielbild

[4] Basisbilder einer 8x8-DCT,
Quelle: http://oku.edu.mie-u.ac.jp/~okumura/compression/dct.html

[5] Ausgangsbild, markiertes Bild und absolutes Differenzbild einer Wasserzeichen-Einbettung in den Blaukanal (Bildbereich)

[6] Farbintensiviertes Differenzbild für maximale Farbkanaländerungen einer Wasserzeichen-Einbettung in den Blaukanal (Bildbereich)

[7] intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal einer Wasserzeichen-Einbettung in den Blaukanal (Bildbereich)

[8] Ausgangsbild, markiertes Bild und absolutes Differenzbild einer Wavelet-Block-basierten Einbettung in den Frequenzbereich

[9] kontrastmaximiertes Differenzbild, farbintensiviertes Differenzbild für maximale Farbkanaländerungen und intensiviertes Differenzbild für Luminanzänderungen einer Wavelet-Blockbasierten Einbettung in den Frequenzbereich

[10] Ausgangsbild, markiertes Bild, absolutes Differenzbild einer Wavelet-basierten Einbettung in den Frequenzbereich

[11] kontrastmaximiertes Differenzbild, vergrößerter Ausschnitt des kontrastmaximierten Differenzbildes und Differenzbild für maximale Farbkanaländerungen einer Wavelet-basierten Einbettung in den Frequenzbereich

[12] intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal einer Wavelet-Blockbasierten Einbettung in den Frequenzbereich

[13] Ausgangsbild, markiertes Bild, absolutes Differenzbild einer DCT-basierten Einbettung in den Frequenzbereich

[14] intensiviertes Differenzbild für Luminanzveränderungen, vergrößerter Ausschnitt des intensivierten Differenzbildes für Luminanzveränderungen und Differenzbild für maximale Farbkanaländerungen einer DCT-basierten Einbettung in den Frequenzbereich

[15] intensivierte Differenzbilder für Änderungen im Rot-, Grün- und Blau-Kanal einer DCT-basierten Einbettung in den Frequenzbereich

[16] graphische Benutzeroberfläche des „ImageComparator“

[17] vereinfachtes Klassendiagramm des ImageComparator