GMD Report
34

GMD –
Forschungszentrum
Informationstechnik
GmbH

Mark Stabenau, Jana Dittmann

Digitale Wasserzeichen für MPEG Video

September 1998

GMD –
Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

Anschrift der Verfasser/Address of the authors:
Mark Stabeneau
Jana Dittmann
Institut für Integrierte Publikations- und Informationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Dolivostraße 15
D-64293 Darmstadt
E-mail: Mark.Stabenau@gmd.de
Jana.Dittmann@gmd.de

ISSN 1435-2702

Abstract

The development of new multimedia services and environments requires new concepts both to support the new working process on distributed computers and to protect the multimedia data during the production and the distribution in digital marketplaces. This study addresses copyright protection as a major security demand in digital marketplaces.
We propose and compare two watermarking techniques for MPEG video with the intention to show the advantages and the possible weakness in the schemes working in the frequency domain and in the spatial domain. To improve the view to the distortion of the watermarked frames we generate a 3D-difference view measuring the changes which were made during watermarking process and/or caused by several damaging attacks.

Keywords
Security and the media, digital watermarking for MPEG video, copyright protection

Kurzfassung

Mit der Entwicklung neuer multimedialer Dienste und Applikationen sind auch neue Arbeitsprozesse und Sicherheitsstandards zum Schutz digitaler Daten gefragt. Die Akzeptanz von Anbietern digitaler Dienste gegenüber dem Medium Internet hängt in hohem Maße davon ab, ob z.B. entsprechende Techniken für den Schutz von digitalen Videos existieren.
Es werden zwei Watermarking-Verfahren zum Schutz von Urheberrechten, speziell für MPEG-Videos, vorgestellt. Die Stärken und Schwächen der im Bild- bzw. Frequenzraum arbeitenden Verfahren, werden anhand von diversen Angriffen getesten und analysiert. Um die durch die Wasserzeichen und Attacken verursachten, kaum sichtbaren Bildstörungen besser zu veranschaulichen, werden 3D-Differenzbilder generiert.
Die vorgestellten Watermarking-Verfahren geben Autoren von digitalen Daten die Möglichkeit ihr Eigentum zu signieren und damit ihre rechtmäßigen Ansprüche sicher zu stellen.

Schlagwörter
Security und Multimedia, digitale Wasserzeichen für MPEG Video, Urheberrechtsschutz

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

BCH-Code	Bose Chaudhuri Hochquenghem-Code
CD	Compact Disc
DCT	Discrete Cosine Transformation
DVD	Digital Versatile Disc
FFT	Fast Fourier Transformation
IDCT	Inverse Discrete Cosine Transformation
ISBN	International Standard Book Number
ISDN	Integrated Services Digital Network
JFIF	JPEG File Interchange Format
JPEG	Joint Photographics Experts Group
MD	Mini Disc
MPEG	Motion Picture Expert Group

Einleitung

Die rasante Entwicklung der digitalen Technik, seien es Datenverarbeitungsgeräte wie Computer, CD-Player, Videogeräte mit DVD's oder Datenträger wie CD's, MD's, DVD's und Festplatten mit Kapazitäten von mehreren Gigabytes, führte in den letzten Jahren zu einer Veränderung in vielen Bereichen des gesellschaftlichen Lebens, aber auch zu veränderten Komsumgewohnheiten. Wir sind es gewohnt digitale Musik in bester Qualität zu hören, gewöhnen uns langsam an das digitale Telefon (ISDN) und warten auf die ersten Videorecorder mit DVD's. Wir benutzen digitale Kameras und leistungsfähige Computer, für die eine unglaubliche Menge von Programmen zur Verfügung stehen, um unterschiedlichste Aufgaben zu erfüllen. Haben wir Zugriff auf einen Computer mit Internetanschluß, können wir das weltweite Computernetz nach nützlichen Daten aller Art absuchen oder selbst eigene Daten über einen Server der Allgemeinheit anbieten. Gerade letzterer Bereich, das weltweite Internet, hat dem Markt der digitalen Medien neue Dimensionen eröffnet. Alles was in digitaler Form verfügbar ist kann in einfachster Weise angeboten und verkauft werden. Die folgende Liste nennt einige Beispiele von digitalen oder digitalisierten Daten, die von kommerziellem Wert sind:

Texte: z.B. Literatur, Nachrichten, aktuelle Aktienkurse

Zeichensätze: z.B. Vektor-Zeichensätze von Adobe

Ton: z.B. Musik oder Klänge

Computer-Programme

Bilder: z.B. Naturaufnahmen, Kataloge, Zeitschriften, Comics, Werbung

Filme/Videos: z.B. Werbefilme, Musikvideos, aktuelle Filme

Die Einfachheit mit der digitale Daten gehandhabt werden können birgt jedoch ein hohes Gefahrenpotential, wenn es um die Wahrung von Urheberrechten geht. Befinden sich die Daten erst einmal im Computer des Käufers, kann dieser ohne Qualitätseinbußen beliebig viele Kopien davon herstellen, unauthorisiert weitergeben oder sogar fälschlicherweise behaupten, daß er die Urheberrechte an diesen Daten besitzt. Um dies zu verhindern, versucht man die verkauften Daten mit Informationen zu versehen (z.B. eine Identifikationsnummer für den Käufer oder ein Copyrightvermerk). Diese Informationen müssen jedoch fest mit den Daten verwoben sein, um ein Entfernen unmöglich zu machen. Ein Header in einer Bilddatei könnte z.B. mit einem einfachen Hex-Editor entfernt werden. Verfahren, die Informationen in digitale Daten fest einbringen werden im folgenden Watermarking-Verfahren oder Watermarking Mechanismen genannt. Die eingebrachten Informationen werden als "digitale Wasserzeichen", "Wasserzeichen" oder "Watermark" bezeichnet. Watermarking-Verfahren gibt es hauptsächlich für Audio, Filme/Video und Einzelbilder. In dieser Studie werden ausschließlich Verfahren für visuelle Medien diskutiert.

Zielsetzung

Ziel dieser Studie ist es zwei bestehende Watermarking-Verfahren für Einzelbilder zu implementieren, zu verbessern, und in ein MPEG-En/Dekodier-Schema einzubetten. Eines dieser Verfahren beruht auf einer Veröffentlichung von Zhao und Koch [ZhK1995], auf dem auch ihr kommerzielles Verfahren SysCop aufbaut. In ihren Veröffentlichungen wird die Anwendung ihres Verfahrens auf Videodaten jedoch nicht dargestellt. Das andere Verfahren wurde von Jiri Fridrich entwickelt und wird in [Fri1998] beschrieben. An beiden Verfahren werden Verbesserungen vorgenommen, die z.B. die Sichtbarkeit des Wasserzeichens reduzieren oder die Fehleranfälligkeit durch die Einbringung eines fehlerkorrigierenden Codes verbessern.

Beide Verfahren werden daraufhin untersucht, wie stark sich die deutlich höheren Kompressionsraten der MPEG-Codierung gegenüber von Einzelbildkompressionsverfahren auf das Wasserzeichen auswirken. Die Robustheit gegenüber Formatkonvertierung, Mehrfachmarkierung und dem

Antiwatermark-Tool StirMark [Kuh1997] von Markus G. Kuhn werden überprüft.

In Kapitel 1 wird zunächst auf die historische Entwicklung von Wasserzeichen eingegangen. Weiterhin werden die verschiedenen Verwendungszwecke von digitalen Wasserzeichen dargestellt und die je nach Verwendungszweck unterschiedlichen Anforderungen diskutiert. Da zunächst Wasserzeichen recht allgemein und eher in Bezug auf Bilder diskutiert werden, wird am Schluß von Kapitel 1 auf Unterschiede bei der Markierung von MPEG-Videos eingegangen.

In Kapitel 2 werden nach einer kurzen technischen Einführung in die DCT-Transformation und in das JPEG-/MPEG-Kompressionverfahren, diverse Attacken und Gegenattacken auf Wasserzeichen vorgestellt. Anschließend wird ein Überblick über den aktuellen Stand der Technik bei Watermarking-Verfahren gegeben, um prinzipielle Vorgehensweisen zum Einfügen und Auslesen von Wasserzeichen darzustellen.

Kapitel 3 stellt die beiden untersuchten Ansätze im Bild- und Frequenzraum mit den vorgenommenen Verbesserungen ausführlich vor.

Details zur Implementierung werden in Kapitel 4 dargestellt.

In Kapitel 5 werden die durchgeführten Experimente mit den Ergebnistabellen für die gemessenen Fehlerraten vorgestellt.

In Kapitel 6 werden die Ergebnistabellen und die Beobachtungen aus den Experimenten ausführlich diskutiert und Schlußfolgerungen gezogen.

Nötige und wünschenswerte Verbesserungen am Programm und an den Verfahren werden schließlich in Kapitel 7 vorgestellt.

Kapitel 1

Watermarking

1.1 Historisches

Die Idee des digitalen Wasserzeichens findet seinen Ursprung in der Steganographie und was soviel wie "Versteckte Botschaft" heißt. Das Verstecken von geheimen Botschaften wird seit Jahrhunderten mit großem Ideenreichtum praktiziert. Eines der ältesten Dokumente über Steganographie stammt aus den Historien von Herodotus. Eine Geschichte daraus beschreibt, wie Demeratus Sparta benachrichtigen wollte, daß Xerxes vorhatte Griechenland zu invadieren. Dazu entfernte er das Wachs von einer Schreibtafel und schrieb die Nachricht auf das darunterliegende Holz. Anschließend überzog er die Tafel wieder mit Wachs und die Nachricht passierte ungehindert die Kontrollen.

Eine andere Methode war es einem Untergebenen den Kopf kahl zu rasieren und ihm die Nachricht auf die Kopfhaut zu tätowieren. War das Haar wieder nachgewachsen, passierte auch diese Nachricht die Kontrollen, ohne Verdacht hervorzurufen.

In den Anfängen des zweiten Weltkriegs wurden hauptsächlich unsichtbare Tinten verwendet (Milch, Essig, Fruchtsäfte und Urin), die sich beim Erwärmen dunkel färbten. Später wurden Flüssigkeiten benutzt, die nur auf bestimmte andere Chemikalien reagierten.

Mit dem Einstieg in das Computerzeitalter erfuhr die Steganographie einen enormen Aufschwung. Digitale Daten eignen sich hervorragend zum Verstecken von Botschaften. Ändert man z.B. die niederwertigsten Bits von

Bilddaten einfach in die Bitfolge der zu versteckenden Nachricht, ist das geänderte Bild mit bloßem Auge nicht vom Orginal zu unterscheiden.

Auch die Exportbeschränkungen die z.B. die US-Regierung auf kryptologische Verfahren auferlegt, hat nicht zuletzt zu einem Boom auf diesem Gebiet geführt.

Die Entwicklungen auf dem Gebiet der Steganographie machen sich heute kommerzielle Anbieter digitaler Daten zunutze, um ihre Produkte wie z.B. Bilder oder Texte im Internet anzubieten und gleichzeitig ihre Urheberrechte zu schützen. Steganographie bietet die Möglichkeit versteckte Informationen über Copyright, erlaubte Vervielfältigung, den Empfänger, etc. in das zu schützende Objekt einzubringen. Das digitale Wasserzeichen ist geboren.

1.2 Einteilung von Watermarking-Verfahren

Obwohl in den meisten Fällen digitale Wasserzeichen a priori als unsichtbar bzw. in Audio-Verfahren als unhörbar angenommen werden, tauchen doch gelegentlich Verfahren auf, die ganz bewußt sichtbare Markierungen an dem zu schützenden Objekt anbringen und ebenfalls als Wasserzeichen bezeichnet werden. Um die Verfahren, die in dieser Studie näher untersucht werden, gegen andere "Watermarking"-Verfahren abzugrenzen, wird folgende Kategorisierung vorgenommen:

- Sichtbare Wasserzeichen
- Unsichtbar-robuste Wasserzeichen
- Unsichtbar-"zerbrechliche" Wasserzeichen

In den folgenden Unterkapiteln wird nur kurz auf die sichtbaren und "zerbrechlichen" Wasserzeichen eingegangen, um die Abgrenzung gegenüber den unsichtbar-robusten Wasserzeichen zu erleichtern.

1.2.1 Sichtbare Wasserzeichen

Der Nutzen von sichtbaren Wasserzeichen ist offensichtlich. Sie werden benötigt, um ein nicht authorisiertes Kopieren von digitalen Bildern/Filmen

zu demotivieren, indem ein nicht oder nur schwer zu entfernendes Copyright-Symbol eingebracht wird. Dazu wird ein deutlich sichtbares Symbol oder eine deutlich sichtbare Veränderung an dem Bild vorgenommen. Beispiele für derartige Wasserzeichen finden sich in jeder Fernsehaufzeichnung in der zumeist ein Symbol des Fernsehsenders eine Bildecke ziert oder ein großes "C" auf jedem Bild der riesigen Bilddatenbank von Corbis. Es lassen sich einige **Anforderungen** an diese Art von Wasserzeichen aufstellen:

Visuelle Artefakte Ein sichtbares Wasserzeichen muß sowohl in jedem Farbbereich als auch in einem monochromen Bild deutlich sichtbar sein und einen großen Teil bzw. die wichtigen Teile des Bildes abdecken.

Qualitätsminderung Das Wasserzeichen sollte zwar verändernd sprich qualitätsmindernd auf das Bild wirken, jedoch gleichzeitig eine genaue Studie des Bildes und Kaufobjektes nicht verhindern.

Flexibilität Das Maß der Qualitätsminderung durch das Wasserzeichen sollte einstellbar sein, um verschiedenen Typen von Bildern gerecht zu werden. Hochauflösende Fotos, die an einen Kundenkreis gerichtet sind, die eben diese hohe Qualität benötigen (z.B. Werbebranche), werden durch ein geringes niederfrequentes Rauschen im Bild bereits empfindlich gestört, wobei das klassische Bild von Albert Einstein mit herausgestreckter Zunge noch mit leichten Störungen durchaus viele Homepages verschönern könnte.

Robustheit Das Wasserzeichen darf nicht leicht entfernbar sein. Schlimmstenfalls muß es ebenso teuer sein das Wasserzeichen zu entfernen, wie die Rechte am Bild selbst zu erstehen.

geringe Kosten Die Einbringung des Wasserzeichens sollte automatisch erfolgen können, bzw. sowenig wie möglich menschliche Interaktion erfordern.

1.2.2 Unsichtbar-robuste Wasserzeichen

Die unsichtbar-robusten Wasserzeichen sind Hauptthema dieser Studie. Der Nutzen dieser Wasserzeichen setzt ein, wo der Nutzen der sichtbaren Wasserzeichen endet: Wenn der Kunde sich entschlossen hat ein digitales Bild

zu kaufen. Dann wird das sichtbare Wasserzeichen entfernt und die Kontrolle des Verkäufers endet. Hier kann nun ein unsichtbar-robustes Wasserzeichen zu weiterem Schutz der Urheberrechte dienen. Verschiedene Beispiele von technischen Realisierungen solcher Wasserzeichen werden in Kapitel 2 genauer untersucht. Zunächst werden einige allgemeine Bedingungen an unsichtbar-robuste Wasserzeichen in Bezug auf Copyright-Informationen dargestellt. Diese müssen jedoch im Zusammenhang mit anderen Verwendungszwecken von Wasserzeichen relativiert werden. Deshalb werden nach den allgemeinen Bedingungen noch verschiedene Verwendungsmöglichkeiten von unsichtbar-robusten Wasserzeichen und deren speziellen Anforderungen aufgelistet.

Sichtbarkeit Das Wasserzeichen darf für einen Menschen mit einem normal ausgebildeten Sehvermögen nicht wahrnehmbar sein. Diese Bedingung führt jedoch zu einem inhärenten Problem gegenüber Robustheitsanforderungen. Wenn die Einbringung des Wasserzeichens zu keiner wahrnehmbaren Veränderung des Bildinhaltes führt, dann besteht das Wasserzeichen aus redundanter Information. Deswegen kann nicht garantiert werden, daß es irgendwann ein Kompressionsverfahren gibt, bei dem diese Information trotzdem verlorengeht. Gängige Watermarking-Verfahren berücksichtigen bisher hauptsächlich Kompressionsverfahren, die im Frequenzraum arbeiten.
Daher muß immer ein Kompromiß zwischen Sichtbarkeit und Robustheit des Wasserzeichens eingegangen werden.

Präsenz Das unsichtbare Wasserzeichen muß ebenso wie die sichtbaren Wasserzeichen einen großen bzw. den wichtigen Teil des Bildes abdecken. Das Vorhandensein des Wasserzeichens in einem breiten Frequenzraum wäre von Vorteil, um eine Detektion zu erschweren.

Richtige Detektion Eine falsches Erkennen eines Wasserzeichens, also eine Detektion (Retrieval) eines Wasserzeichens obwohl keines eingebracht worden ist, darf auf keinen Fall passieren. Verfahren, die auf guten Zufallsgeneratoren basieren, haben mit dieser Anforderung wenig Schwierigkeiten.

Robustheit Die Robustheit des Wasserzeichen gegenüber freundlichen und feindlichen Attacken ist sehr wichtig. Das Wasserzeichen darf weder

gelöscht noch das Auslesen unmöglich gemacht werden können. Eine ausführliche Darstellung und Klassifizierung möglicher Attacken findet sich in Kapitel 2.2.

Wie bereits angedeutet gibt es verschiedene Verwendungsmöglichkeiten für die Klasse der unsichtbar-robusten Wasserzeichen. Drei Hauptanwendungen werden nun auf Anforderungen in Bezug auf Datenrate, (Nicht-)Sichtbarkeit und Robustheit untersucht.

Einbringung von versteckten Botschaften Dieser Anwendungsbereich wurde bereits kurz im Kapitel 1.1 angeschnitten. Ziel ist es eine Nachricht so unauffällig wie möglich in Bildinformationen einzubetten. Deshalb gilt hier, daß die Nichtsichtbarkeit der eingebrachten Information oberste Priorität hat. Nichtsichtbarkeit bedeutet hier aber nicht nur für das menschliche Auge nicht sichtbar. Verfahren, die die niederwertigsten Bits eines Bildes mit denen der zu versteckenden Botschaft austauschen verändern zwar das sichtbare Bild praktisch nicht, ein Computer jedoch könnte anhand von statistischen Daten (z.B. Wechsel von niederwertigen Bits in glatten Flächen) ein markiertes Bild erkennen.
Der Anspruch an die Datenrate ist nicht allzu groß, da geheime Botschaften wohl eher kurz ausfallen. Da jedoch der Anspruch an die Robustheit sehr niedrig ist (um eine Attacke zu starten, müßte ja bekannt sein, daß das Bild eine versteckte Information enthält) können mit Hilfe von Steganographie-Verfahren auch große Datenmengen versteckt werden.
Diverse Steganographie-Verfahren werden in [Rin1997a] beschrieben und getestet.

Einbringung von Copyright-Informationen Es gibt hier eine interessante Unterteilung, die sich auf den Zeitpunkt der Einbringung von Copyright-Informationen bezieht. Diese können zum Zeitpunkt der Erstellung des digitalen Objektes oder während der Vervielfältigung eingebracht werden. Letzteres wird auch als *Fingerprinting* bezeichnet und hat den sehr nützlichen und wünschenswerten Vorteil, daß individuelle Kopien unterschiedlich markiert werden können. Wären z.B. bei einem Video-on-demand System die Geräte vor Ort mit einem Watermarking-System ausgerüstet, liessen sich unerlaubte Kopien

direkt auf den Urheber zurückverfolgen. Ein Fingerprinting-Verfahren wird z.B. in [ScU1998] dargestellt. Hier werden im Prinzip jedem Kunden Stützstellen einer geometrischen Figur zugeordnet. Dies sind einfach ganze Zahlen, die über einen mathematischen Zusammenhang z.B. eine Gerade beschreiben. Diese Zahlen, die die Stützstellen beschreiben, werden über ein Watermarking-Verfahren in Bilder eingebracht. Der Vorteil dieses Verfahrens ist, daß wenn ein eingeschränkter Kundenkreis versucht durch Mittelung der Bildinformation die Fingerprints zu entfernen, kann aus dem resultierenden Bild auf die Angreifer geschlossen werden. Anschaulich bedeutet ein Angriff, daß alle unterschiedlichen Punkte der geometrischen Figuren entfernt werden können, die Schnittpunkte jedoch im Bild verbleiben. Jeder Kunde, der einen verbliebenen Punkt in seiner eigenen geometrischen Figur besitzt war dann an dem Angriff beteiligt. Problematisch an diesem Verfahren ist, daß die Anzahl der Stützstellen mit der Anzahl der Kunden exponentiell wächst.
Technisch sind jedoch Geräte-basierte Fingerprinting-Verfahren schwer realisierbar, da alle Geräte (z.B. bei einem Video-on-Demand System) mit diesem Verfahren ausgerüstet werden müßten. Dies erfordert einen breit akzeptierten, weltweiten Standard, der schwer durchzusetzen ist. Ausserdem, um den hohen Robustheitsanforderungen zu genügen, müsste jede Kopie mit einem unterschiedlichen Schlüssel markiert werden, da die Entdeckung eines einzigen Schlüssels fatal wäre. Damit wäre aber eine automatische Verfolgung von Kopien im Internet ausgeschlossen, da das Ausprobieren jedes Schlüssels eines Benutzers nicht akzeptabel ist.
Es zeigt sich, daß *Fingerprinting* noch mit prinzipiellen und ungelösten Problemen belastet ist und deren Lösung in naher Zukunft nicht abzusehen ist.

Gibt es ein robustes System zum Einbetten von Daten in digitale Medien, stellt sich immer noch die Frage welche Art von Informationen eingefügt werden soll. Diverse internationale Organisationen bemühen sich Standards und Umgebungen zu entwickeln, die zu besserem Schutz von Urheberrechten digitaler Medien führen soll: **AIDAA** (International Association for Audio-visual Authors), **AGICOA** (Association for International Management of Audio-visual Works), **BIEM** (Internatio-

nal Office of Mechanical Recording), **CISAC** (International Confederation of Authors' and Composers' Societies), **CESAC** (European Group of Authors an Composers), **FERA** (European Federation of Audiovisual Producers), **TALISMAN** (Tracing Authors' rights by Labeling Images Services And Monitoring Access Network). Obwohl man gerne sehr viele Informationen wie Objekttyp, Ursprungsland, Halter der Urheberrechte, bestehende Lizensverträge usw., in das Zielobjekt einfügen möchte, kristallisiert sich aus der Arbeit obiger Organisationen heraus, daß eine Identifikationsnummer (eine Art ISBN für digitale Medien) bereits ausreichen muß, da man aufgrund von hohen Robustheitsanforderungen an das Watermarking-Verfahren, keine hohe Datenrate erwarten kann. Eine Identifikationsnummer kann dann als Link in eine zentral verwaltete Datenbank dienen, in der sämtliche benötigten Informationen vorliegen und auch jederzeit aktualisiert werden können. Diese Identifikationsnummer sollte vorzugsweise maximal 64 Bit Länge besitzen, da bei der Entwicklung des MPEG-Standards bereits ein Feld mit dieser Größe für einen Copyrightvermerk vorgesehen ist. Trotzdem sollte natürlich die Identifikationsnummer als Wasserzeichen in das Video selbst eingebracht werden.
Eine andere Idee, die bereits von einigen kommerziellen Verfahren (Digimarc [Dig1998], Highwater Signum [Sig1998]) genutzt wird, bringt einen Hyperlink in das Bild ein, mit dem man direkten Zugriff auf einen Datenbankeintrag bekommt.
Es stellt sich nun die Frage wer eigentlich Zugriff auf die eingebrachten Daten hat. Vorstellbar wäre eine Unterscheidung in öffentliche und private Informationen (diese Unterscheidung kann auch bei Metainformationen gemacht werden), die nur mit einem entsprechendem Schlüssel lesbar sind. Die öffentlichen Informationen sind als allgemeiner Copyright-Vermerk gedacht, damit jederzeit Klarheit über eventuelle Lizensbestimmungen von veröffentlichten digitalen Objekten besteht. Die privaten Informationen dienen dem Vertreiber, möglicherweise noch mit Hilfe einer weiteren vertraulichen Organisation, vor einem Gericht seine Urheberrechte zu beweisen.
Öffentliche Copyrightinformationen haben jedoch ein gravierendes Robustheitsproblem: Wenn jeder in der Lage ist das Wasserzeichen ohne besonderen Schlüssel auszulesen, dann beruht die Robustheit des Wasserzeichens nur auf dem geheimzuhaltenden Verfahren. Würde dies einmal bekannt werden, wären die öffentlichen Copyrightinformationen

hinfällig. Dieses Problem ist aus der Kryptographie bereits bekannt und das Prinzip, worauf die Sicherheit eines Verfahrens nicht auf der Geheimhaltung desselben beruhen darf wird auch Kerckhoffsches Prinzip genannt. Dies muß noch nicht heißen, daß öffentliche Copyrightinformationen als Wasserzeichen sinnlos sind, schließlich werden sie bereits von diversen kommerziellen Produkten angeboten. Trotzdem muß ihr Nutzen und Gebrauch immer vor diesem Hintergrund relativiert werden. Die beiden in dieser Studie implementierten Verfahren bieten die Möglichkeit eines öffentlich auslesbaren Wasserzeichens noch nicht an. Die Hauptanforderung an Wasserzeichen die Copyright-Informationen darstellen ist die Robustheit. Obwohl die Anzahl an freundlichen und feindlichen Attacken auf derartige Wasserzeichen enorm ist (siehe Kapitel 2.2) und diese Anforderung deshalb unerfüllbar erscheint, kann man doch hoffen, auch aufgrund der wenigen Daten (64 Bit), die nur eingebracht werden müssen, daß neue Watermarking-Verfahren und neue Attacken sich ebenso gegenseitig anspornen werden, wie dies in der Kryptographie geschehen ist.

Einbringung von Metainformationen Diverse Metainformationen sind denkbar. Ein digitales Bild von einem Gemälde könnte mit der Biographie des Künstlers, mit einer Entstehungsgeschichte oder einfach mit einer kleinen Anekdote versehen werden. Solche Informationen wären zwar recht praktisch, könnten jedoch auch in Zusatzdateien oder einer Headerdatei untergebracht werden. Viel nützlicher sind jedoch Daten, die über einen Retrievalprozeß erreichbar wären. Würden z.B. alle Bilder oder Videos mit Bild- oder Szenenbeschreibungen versehen werden, können Computer das Internet Tag und Nacht nach Bildern absuchen, die man für bestimmte Zwecke benötigt.
Anbieter von Photo-CDs könnten Metainformationen in ihren Bildern nutzen, um potentiellen Kunden weitere Bilder anzubieten (sofern diese im Internet benutzt werden und ein Computer sie automatisch aufspürt).
Es ist klar, daß die Menge an Informationen über ein digitales Objekt praktisch beliebig groß sein kann. Damit ist die Hauptanforderung an ein Verfahren zur Einbringung von Metainformation eine hohe Datenrate. Das Verfahren sollte weiterhin gegenüber freundlichen Attacken (siehe 2.2.1) robust sein. Dabei gilt zwar je robuster, desto besser, aber da hiermit keine Sicherheitsaspekte verknüpft sind, ist diese

Anforderung nicht sehr hoch. Für die (Nicht-)Sichtbarkeit der Metainformationen gilt ähnliches, wie bei den Copyright-Informationen. Ein Kompromiß zwischen Robustheit und Qualität des markierten Objekts muß gefunden werden. Allerdings liegt der Kompromiß hier eher auf der Seite der Qualität, da Metainformationen nicht den Wert von Copyright-Informationen besitzen.

1.2.3 Unsichtbar-"zerbrechliche" Wasserzeichen

Zu guter Letzt gibt es noch die Klasse der unsichtbar-"zerbrechlichen" Wasserzeichen. "Zerbrechlich" werden sie deshalb genannt, weil es bei ihnen gerade wünschenswert ist, daß Bildveränderungen auch zu Änderungen im Wasserzeichen führen. Damit liesse sich dann die Orginalität eines Bildes nachweisen, z.B. um die Authenzität von Bildmaterial in einem Gerichtsverfahren zu überprüfen. Zusätzlich zu den ersten drei Anforderungen der "Unsichtbar-robusten Wasserzeichen" lassen sich zusätzliche Anforderungen aufstellen:

"Zerbrechlichkeit" Jede Änderung im Bildmaterial muß zu Änderungen im Wasserzeichen führen. Dies ist das Hauptmerkmal von "zerbrechlichen" Wasserzeichen. Eine Kompression des Bildes sollte das Wasserzeichen jedoch kaum beeinflussen, da diese Bildoperation nicht zu sichtbaren Veränderungen im Bild führt. Wünschenswert wäre es, wenn in dem geänderten Wasserzeichen die verfälschten Pixelpositionen ablesbar wären.

Sicherheit Das Wasserzeichen muß in dem Sinner sicher sein, daß es nicht möglich ist nach Änderung des Bildmaterials das ursprüngliche Wasserzeichen wieder herzustellen.

Geringe visuelle Artefakte Das Wasserzeichen sollte durch so wenig wie möglich Pixelabänderungen in das Bild eingebracht werden können. Obwohl dies auch für unsichtbar-robuste Wasserzeichen gilt, müssen beim unsichtbar-"zerbrechlichen" Wasserzeichen keine feindlichen Attacken sondern lediglich der Sicherheitsaspekt berücksichtigt werden. Dadurch sollte das "zerbrechliche" Wasserzeichen mit deutlich weniger Pixeländerungen auskommen.

1.3 Unterschiede zwischen Wasserzeichen in Bildern und Videos

Die bisherigen Überlegungen zu und Kategorisierung von Watermarking Mechanismen gelten, auch wenn im Text hauptsächlich Bilder als Beispiele genannt wurden, grundsätzlich sowohl für die Markierung von Bildern als auch für die Markierung von Videos. Bei Watermarking-Verfahren für Videos müssen jedoch einige zusätzliche Aspekte berücksichtigt werden:

- Das Wasserzeichen muß von vornherein eine hohe Robustheit gegen verlustbehaftete Kompression (MPEG) besitzen. Geht man von 24-Bit Einzelbildern aus, muß das Wasserzeichen Kompressionsfaktoren bis zu 1 : 200 oder sogar noch höher für B-Frames überstehen.

- Obwohl praktisch jedes Watermarking-Verfahren für Bilder auch für Videos benutzt werden kann indem das Video zuerst decodiert wird, anschließend die einzelnen Frames markiert und wieder zu einem Video encodiert werden, geht bei diesem De/Encoding-Prozeß Bildqualität verloren. Außerdem ist dieser Schritt natürlich sehr zeitintensiv. Deshalb ist es für ein Watermarking-Verfahren für Videos sehr vorteilhaft, wenn die Information direkt in den MPEG-Bitstrom eingebracht werden kann. Ein Ansatz für die beiden in dieser Studie vorgestellten Verfahren wird in Kapitel 7.4 vorgestellt.

- Im Vergleich zu hochqualitative Bildern kann die Gesamtheit aller in hochqualitativen Videos enthaltenen Frames mehr als hundert oder tausend mal größere Datenmengen zum Einbetten von Wasserzeichen bieten. Daher stellt sich hier besonders die Frage, welche Informationen eingebracht werden sollen und können.

Kapitel 2

Watermarking: Verfahren und Attacken

2.1 Technische Grundlagen

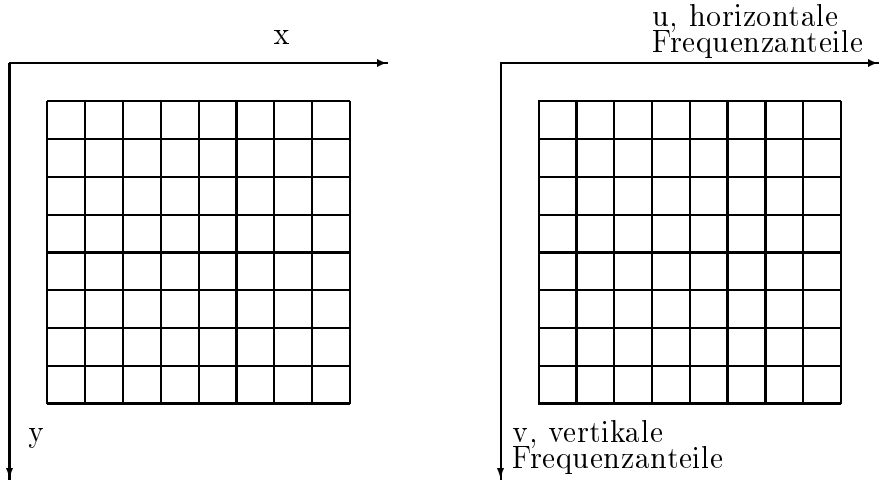
Bevor verschiedene Robustheitsanforderungen an und die Methodik von Watermarking-Verfahren sowie diverse Verfahren selbst ausführlicher diskutiert werden können, werden im folgenden einige technische Grundlagen dargestellt.

2.1.1 Die Kosinus Transformation

Die Kosinus Transformation (DCT für Discrete Cosine Transformation) ist die zentrale Technik, die in dem Bildkompressionsverfahren JPEG und in dem Videoformat MPEG benutzt wird. Sie dient der Umwandlung eines Bildes vom Bildraum in den Frequenzraum. Dabei wird das Bild in 8x8 Blöcke unterteilt, die anschließend transformiert werden.

Der oberste linke Koeffizient der DCT-Koeffizienten wird DC-Koeffizient genannt und entspricht dem durchschnittlichen Wert der Pixeldaten. Die anderen DCT-Koeffizienten werden AC-Koeffizienten genannt und geben die horizontalen und vertikalen Frequenzanteile an. Rechts neben dem

Abbildung 2.1: DCT-Transformation vom Bildraum in den Frequenzraum



DC-Koeffizient stehen die AC-Koeffizienten die ausschließlich horizontale Frequenzanteile enthalten in aufsteigender Reihenfolge. Unter dem DC-Koeffizient stehen die AC-Koeffizienten die ausschließlich vertikale Frequenzanteile enthalten in aufsteigender Reihenfolge. Ein Bild welches z.B. nur aus vertikalen Linien besteht würde nur horizontale Frequenzanteile besitzen und nur AC-Koeffizienten enthalten die rechts neben dem DC-Koeffizient stehen. Die zwei-dimensionale Matrix der Frequenzanteile enthält die gesamte Information über die Bildraummatrix, bis auf den Teil, der durch endliche arithmetische Präzision verlorengeht.

Die zwei-dimensionale DCT ist folgendermaßen definiert:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos(\pi(2x+1)u/16)\cos(\pi(2y+1)v/16)$$

Die inverse Funktion IDCT sieht entsprechend so aus:

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} F(u,v)C(u)C(v)\cos(\pi(2x+1)u/16)\cos(\pi(2y+1)v/16)$$

mit:	u, v, x, y = 0, 1, 2, ... 7
und	x,y = Bildraumkoordinaten
	u,v = Koordinaten der Frequenzanteile
C(u)	$= 1/\sqrt{2}$ für u=0 und 1 für $u > 0$
C(v)	$= 1/\sqrt{2}$ für v=0 und 1 für $v > 0$
F(u,v)	Frequenzdaten
f(x,y)	Bilddaten

2.1.2 JPEG

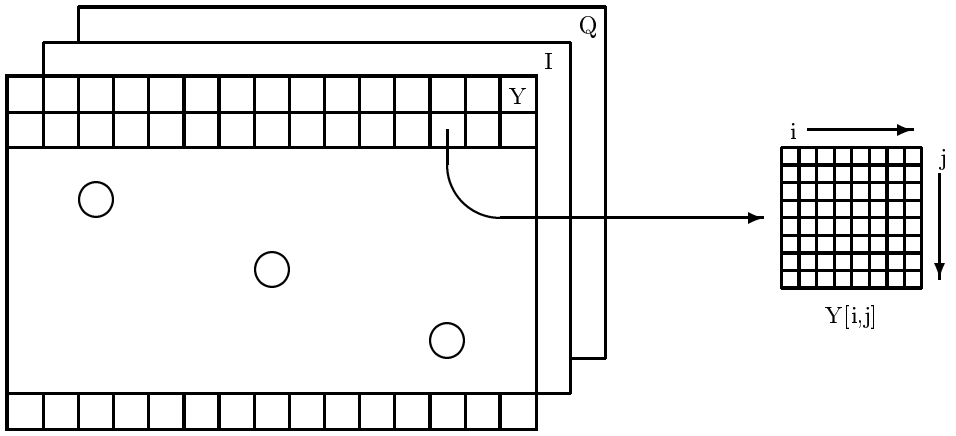
Auf der diskreten Kosinus-Transformation basiert das Kompressionsverfahren der Joint Photographic Expert Group (JPEG) für ihr Bildformat JFIF, welches fälschlicherweise oft als JPEG-Format bezeichnet wird.

Im folgenden soll der prinzipielle Ablauf des Kompressionsverfahrens anhand eines 24-Bit Bildes beschrieben werden, welches aus der Helligkeitskomponente Y und den zwei Farbkomponenten I und Q besteht. Es wird jedoch nur die Kompression der Helligkeitskomponente Y beschrieben, da die Kompression für die anderen beiden Komponenten fast identisch abläuft.

Zunächst wird die Helligkeitskomponente Y in 8x8 Blöcke aufgeteilt, die Matrizen aus 64 Einträgen darstellen und Werte zwischen 0 und 255 enthalten. Diese Werte werden zunächst auf den Bereich zwischen -128 und 127 normiert, (bei den Farbkomponenten ist dies nicht notwendig, da sie bereits in dieser Form vorliegen). Eine solche Matrix nennen wir Y[i,j] mit $i, j \in 0...7$. Abbildung 2.2 zeigt die Beziehung zwischen Y[i,j] und dem gesamten Bild.

Im zweiten Schritt wird die Matrix Y[i,j] mittels der DCT (s. 2.1.1) in eine Matrix Y[u,v] umgewandelt ($u, v \in 0...7$), die nun statt Helligkeitswerten Frequenzanteile enthält. Nun kommt der "verlustbehaftete" Schritt des Kompressionsverfahrens. Jedes Element der Matrix Y[u,v] wird durch einen Wert aus einer Quantisierungstabelle QT[u,v] dividiert und gerundet. Es entsteht die Matrix $Y_Q[u, v]$. In diesem sog. "Quantisierungsschritt" geht Information verloren, weil der inverse Schritt, das Multiplizieren von $Y_Q[u, v]$ mit QT[u,v] nicht mehr exakt die Werte aus Y[u,v] ergeben muß. Die Qualität des komprimierten Bildes und der Kompressionsfaktor hängen hauptsächlich von den Quantisierungstabellen ab. Für die Helligkeitskomponente und für die beiden

Abbildung 2.2: Definition von Y[i,j]

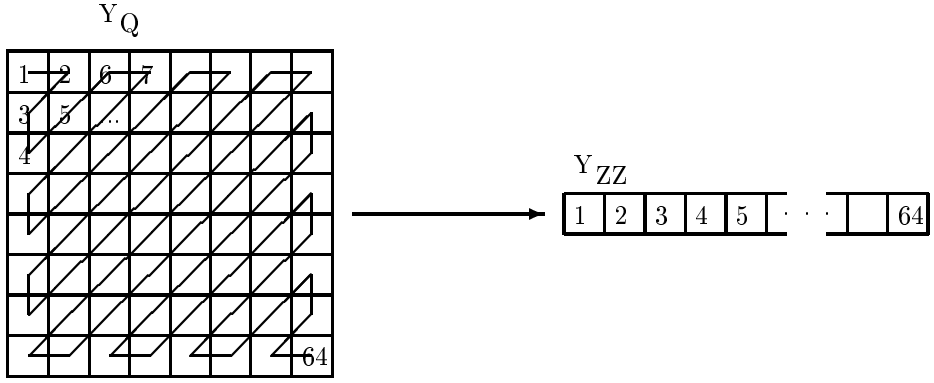


Farbkomponenten werden häufig unterschiedliche Quantisierungstabellen benutzt.

Die Matrix $Y_Q[u, v]$ enthält nun viele Nullen im rechten unteren Teil, da die höheren Frequenzen stärker quantisiert werden können ohne das die Bildqualität darunter leidet. Deshalb findet nun die "Zick-Zack Umsortierung" statt. Dazu wird die 8x8 Matrix $Y_Q[u, v]$ in einer Zick-Zack Reihenfolge in einen 64-elementigen Vektor $Y_{ZZ}[x]$ mit $x \in 0...63$ einsortiert.

Abbildung 2.3 illustriert diesen Vorgang.

Abbildung 2.3: Zick-Zack Umsortierung



Der Vektor $Y_{ZZ}[x]$ enthält jetzt vorzugsweise Nullen für große x. Dies macht

man sich mit der Lauflängenkodierung zunutze. Ein Wert im Vektor $Y_{ZZ}[x]$ wird jeweils durch ein Wertepaar (L,W) dargestellt, wobei L für die Anzahl der führenden Nullen vor dem Wert und W für den Wert selbst steht (z.B. 30 1 0 0 4 0 0 0 -3 wird zu (0,30), (0,1), (2,4), (3,-3)).

Im letzten Schritt werden die Wertepaare aus der Lauflängenkodierung noch entropiekodiert. Dies kann entweder mit dem bekannten Huffman Verfahren oder mit dem Arithmetischen Verfahren gemacht werden. Auf letzteres besteht allerdings ein Patent, sodaß das Huffman Verfahren das häufiger gewählte ist.

2.1.3 MPEG

Die folgenden Erläuterungen beziehen sich auf das MPEG-1 Verfahren. Das Kompressionsverfahren für MPEG-2 ist im wesentlichen identisch.

Die grundlegende Idee im MPEG (Moving Picture Coding Experts Group) Videokompressionsverfahren ist die Redundanz innerhalb eines Bildes bzw. Frames und zwischen den Bildern zu eliminieren. Ersterer Teil wird in ähnlicher Weise wie im JPEG-Kompressionsverfahren durchgeführt. Letzterer wird mittels der sog. "motion-compensation" erreicht. Eine weitere Kompression wird durch Resampling der Farbkomponenten I und Q (die Bilder müssen im YIQ-Format vorliegen) um den Faktor 2 bzw. 4 (X- und wahlweise Y-Richtung) erreicht.

Zunächst werden drei Frame-Typen definiert:

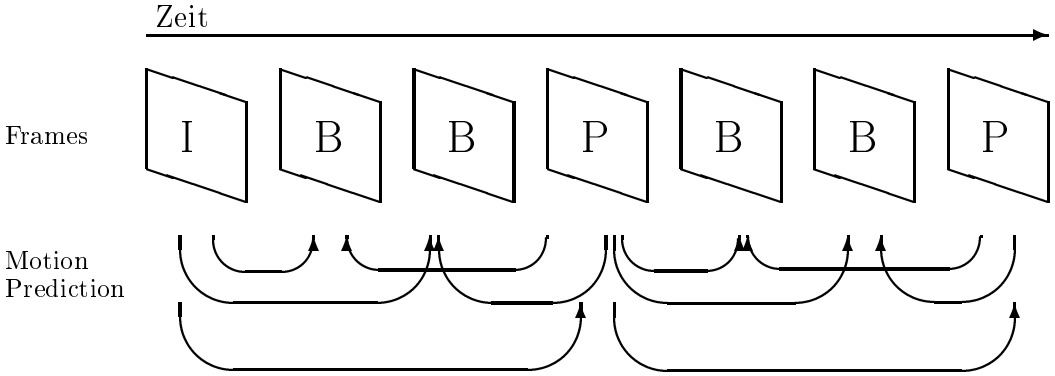
I-Frames Intra-kodierte Frames. Sie treten oft im Abstand von 12-15 Frames (GOP) auf und sind relativ schwach komprimiert (grober Richtwert: ca. 1 Bit/Pixel), weil sie nur die Redundanz innerhalb eines Frames ausnutzen. Sie dienen zum einen als Referenzframes für B- und P-Frames und zum anderen für den wahlfreien (Random Access) Zugriff in das gesamte Video, da zum Dekodieren von I-Frames keine Referenzdaten, wie bei B- und P-Frames, benötigt werden.

P-Frames Predicted Frames. Sie benötigen als Referenz ein vorrausgehendes I- oder P-Frame aus dem mit Bewegungsvektoren und zugehörigen Makroblöcken das P-Frame "vorrausgesagt" wird. Der Kompressionswert liegt in der Größenordnung bei ca. 0.1 Bit/Pixel.

B-Frames Bi-directional predicted Frames. Sie benötigen zwei Referenzframes (I- oder P-Frame), von denen eins zeitlich vor und eins zeitlich hinter dem B-Frame liegt, aus denen ebenfalls mit Bewegungsvektoren, zugehörigen Makroblöcken und einer Interpolationsregel das B-Frame berechnet wird. B-Frames selbst werden nicht als Referenzframes benutzt. Der Kompressionwert liegt in der Größenordnung bei ca. 0.015 Bit/Pixel.

Die folgende Abbildung beschreibt den Zusammenhang zwischen I-, P-, und B-Frame ([Ste1995]).

Abbildung 2.4: Anordnung von I-, P- und B-Frames



Wie arbeitet nun die *motion-compensation*?
Jedes Bild wird in Makroblöcke mit einer Ausdehnung von 16 mal 16 Pixeln aufgeteilt. Jeder dieser Makroblöcke wird nun versucht durch einen Makroblock aus einem verhergehenden bzw. nachfolgenden Bild (je nachdem, ob es sich um ein B- oder P-Frame handelt) vorherzusagen, da anzunehmen ist, daß die Inhalte von dicht aufeinanderfolgenden Frames sehr ähnlich sind. Um eine mögliche Bewegung (z.B. ein Kameraschwenk oder ein fahrendes Auto) ausgleichen zu können, wird ein ganzer Bereich in den Referenzframes nach einem optimal passenden Makroblock abgesucht. Das Verfahren hierfür ist im MPEG-Standard nicht spezifiziert.

Mit dem so gefundenen Makroblock wird die Differenz gebildet. Diese wird dann DCT-transformiert, quantisiert und entropiekodiert (nach dem gleichen

Schema, wie beim JPEG-Verfahren). Für die Entropiekodierung werden sowohl Codes mit variabler Länge als auch mit fester Länge benutzt, die jedoch im MPEG-Standard schon festgelegt sind.

Die Bewegungsvektoren werden untereinander differenzenkodiert. Damit macht man sich zunutze, daß die Bewegungsvektoren in einem Frame häufig sehr ähnlich sind, z.B. bei einem Kameraschwenk bei dem der Bildausschnitt einfach um ein Pixel nach links wandert.

2.2 Attacken auf Wasserzeichen

Bei der Entwicklung von Watermarking-Verfahren muß eine Liste von Eingriffen in das Bildmaterial berücksichtigt werden, die auch zu Änderungen im eingebrachten Wasserzeichen führen. Je unempfindlicher das Wasserzeichen gegenüber diesen Eingriffen ist, desto robuster ist es. Diese Bildveränderungen, man kann sie aus der Sicht des Wasserzeichens auch als Attacken bezeichnen, können in freundliche und feindliche Attacken unterteilt werden. Freundliche Attacken bezeichnen Bildveränderungen, die ein Kunde am Bildmaterial vornimmt, ohne bewußt das Wasserzeichen verändern oder löschen zu wollen. Darunter würden z.B. Skalierungen und Kompressionsverfahren fallen. Natürlich können freundliche Attacken auch von einem Piraten benutzt werden um ein Wasserzeichen zu löschen. Feindliche Attacken sind Bildänderungen, die das Bildmaterial zwar kaum verändern, aber größtmöglichen Schaden am Wasserzeichen, z.B. mittels Aufaddieren eines schwachen Rauschens, verursachen.
In den folgenden zwei Unterkapiteln werden die verschiedenen Attacken mit ihrer jeweiligen Wirkungsweise auf ein eingebrachtes Wasserzeichen beschrieben. Jede Attacke auf Bilder läßt sich dabei prinzipiell auch auf Videos anwenden, da hierfür die einzelnen Bildveränderungen lediglich an jedem Frame des Videos durchgeführt werden müßten. Trotzdem sind manche Attacken eher bei Videos und andere Attacken eher bei Bildern zu erwarten.

2.2.1 Freundliche Attacken

Die Robustheit eines Wasserzeichens gegenüber freundlichen Attacken ist sehr wichtig, da praktisch jedes Orginal noch leicht verändert wird, bevor

es z.B. in einer Produkt-Präsentation im Internet eingesetzt wird. Geht das Wasserzeichen bereits bei diesen Bildänderungen verloren, kann die Robustheit gegenüber feindlichen Attacken, also Bildveränderungen, die zum Ziel haben das Wasserzeichen zu entfernen, erst recht nicht gewährleistet werden. Die folgenden Bildveränderungen gehören zu den typischen Bildoperationen, die ein Benutzer einsetzt, um sein Orginalbild an seine entsprechenden Anforderungen anzupassen.

Geometrische Translationen Dazu gehören *Skalierung, Rotation, Spiegelung* und *Verzerrung in X/Y-Richtung*. Diese bekannten und häufigen Bildoperationen erhalten sowohl die Bild- als auch die Wasserzeicheninformation bis auf numerische Artefakte. Obwohl viele bestehende Watermarking-Verfahren Probleme mit der Lokalisierung der eingebrachten Daten haben, sind diese trotzdem vorhanden, sodaß ein ausgereiftes Ausleseverfahren keine prinzipiellen Schwierigkeiten mit diesen Attacken haben sollte. In Kapitel 7.2 wird hierfür eine Technik vorgestellt.

Ausschnittbildung Neben der Tatsache, daß auch nach einer Ausschnittbildung die Lokalisation des eingebrachten Wasserzeichens schwierig ist, kommt das grundlegende Problem hinzu, daß bei dieser Operation Bild- und Wasserzeicheninformation verlorengeht. Soll das Wasserzeichen auch noch in kleinen Teilen des Bildes auszulesen sein, müssen die Informationsbits entsprechend redundant eingebracht werden. Dies geht auf Kosten der Datenrate. Eine Technik um den benannten Problemen zu begegnen wird ebenfalls in Kapitel 7.2 vorgestellt.

Verlustbehaftete Datenkompression Da dies eine der häufigsten Operationen auf Bildern ist, werden die meisten Watermarking-Verfahren von vornherein auf Robustheit gegenüber dieser Art von Attacke getrimmt. Solange hier zumeist das klassische JPEG-Verfahren angewendet wird ist es auch leicht das Wasserzeichen entsprechend robust einzubringen. Problematisch wird es, falls andere Datenkompressionsverfahren, wie z.B. fraktale Kompression oder erst in Zukunft entwickelte Verfahren, angewendet werden.

Formatkonvertierung Bilder und Videos werden in unterschiedlichsten Formaten verwendet. Das Wasserzeichen darf durch entsprechen-

de Konvertierungen nicht verloren gehen, d.h. das Watermarking-Verfahren darf nicht von dem Format abhängen. Eine Rückkonvertierung in das Ursprungsformat wird jedoch bei den meisten Ausleseverfahren nötig sein.

Digital/Analog Wandlung Obwohl sogar die meisten kommerziellen Watermarking-Verfahren hier versagen, ist es wünschenswert, wenn das Wasserzeichen nach Ausdruck und anschließendem Scan mit hoher Qualität, bzw. nach VHS-Konvertierung bei Videos, wieder erkennbar wäre. Zur Kompensation dieser Attacke muß die Wasserzeicheninformation entsprechend redundant eingebracht werden.

2.2.2 Feindliche Attacken

Es gibt wahrscheinlich ebensoviele verschiedene feindlichen Attacken, wie es verschieden Watermarking-Verfahren gibt. Im gegenseitigen Wettlauf um robustere Verfahren und ausgefeilteren Attacken, werden Watermarking-Verfahren hoffentlich einmal ähnlich robust wie die heutige kryptographische Verfahren. Die folgende Liste gibt einen Überblick über die bekanntesten Angriffsmöglichkeiten gegen Watermarking-Verfahren.

Tiefpaßfilter Aufgrund der menschlichen visuellen Eigenschaft hohe Frequenzen wesentlich schwächer wahrzunehmen als niedrige, ändert ein Tiefpaßfilter das wahrgenommene Bild kaum. Die Aussicht ein Wasserzeichen durch einen Tiefpaßfilter entfernen zu können ist jedoch gering, da die meisten Watermarking-Verfahren nur wenig Information (bzw. Energie) in den hohen Frequenzen einbringen. Dies liegt an der bereits erwähnten Tatsache, daß die Wasserzeichen gegen das JPEG-Kompressionsverfahren optimiert werden, welches ebenfalls hohe Frequenzen löscht bzw. am stärksten komprimiert.

Addition von Rauschanteilen Solange nur hochfrequente Rauschanteile addiert werden, gilt ähnliches wie für Tiefpaßfilter. Niederfrequentes Rauschen wirkt sich jedoch auch stärker auf die Bildqualität aus, sodaß mit Löschung von Wasserzeicheninformation auch die Bildqualität deutlich abnimmt.

Formatänderung durch Weglassen von Zeilen/Spalten/Frames
Das Weglassen von Zeilen/Spalten in Bildern hat kaum Einfluß auf die Bildqualität. Für einige Watermarking-Mechanismen, z.B. die beiden in dieser Studie näher untersuchten eingeschlossen, stellt dieser Angriff ein Problem dar, weil die Synchronität beim Auslesen verlorengeht.
Das Auslassen von Frames bei Videos wirft kein Synchronitätsproblem auf, da an jedem Frameanfang neu synchronisiert werden kann. Hier muß nur für ausreichende Redundanz, welche die verlorene Information ersetzen kann, gesorgt werden.

Mittelung gleicher Bilder Diese Attacke macht nur Sinn, wenn gleiche Bilder mit unterschiedlichen Wasserzeichen existieren. Dies ist z.B. bei Fingerprinting Szenarios der Fall. Durch die Mittelung wird die Stärke des Wasserzeichens reduziert. In [BoS1994] konstruieren Boneh und Show einen Code, dessen Länge von der Anzahl unterschiedlicher Wasserzeichen/Fingerprints abhängt, mit dem die Initiatoren dieser Attacke aufgrund des gemittelten Bildes ermittelt werden können.

Ermittelung der Watermark-Funktion In [FBS1998] beschreiben Fridrich et al., wie aufgrund eines kleinen Teils der Wasserzeicheninformation (Erkennung in glatten Flächen) Rückschlüsse auf die gesamte Watermark-Funktion gezogen werden können. Dieser Angriff bezieht sich jedoch nur auf Verfahren, die auf DCT-Koeffizienten arbeiten und das gesamte Bild mit einer Wasserzeichen-Funktion markieren. Dies gilt für das Verfahren von Cox et. at [CKL1996]. Allerdings funktioniert der Angriff nur mit Einschränkungen an die Anzahl der benutzten Koeffizienten, sowie mit hohem Rechenaufwand.
Für die in dieser Arbeit verwendeten Verfahren ist diese Attacke nicht schwerwiegend. Für das von Fridrich adaptierte Bildraumverfahren aus Ansatz II (s. 3.3 zeigt Fridrich selbst die Robustheit gegen diese Attacke. Das Frequenzraumverfahren aus Ansatz I (s. 3.2) ist zwar prinzipiell angreifbar, da die Information über Änderungen in den DCT-Koeffizienten eingebracht wird. Allerdings ist immer höchstens die Information aus einem markierten 8x8 Block erkennbar, da die Markierungen der Blöcke untereinander unabhängig, bzw. nur über einen Zufallsgenerator voneinander abhängig sind.

Mosaik-Attacke Bei der Mosaik-Attacke wird ein Bild in kleinere Teilbilder zerlegt, die einzeln aber aneinandergefügt in digitaler Form publi-

ziert werden. Sind die Teilbilder klein genug, daß die Wasserzeicheninformation nicht mehr ausgelesen werden kann, ist die automatische Verfolgung durch Computer verhindert. Hier können nur intelligente Algorithmen helfen, die zusammengefügte Teilbilder erkennen. Auf Videos ist diese Attacke kaum übertragbar, da nicht garantiert werden kann, daß die Teilvideos über ein Netzwerk oder das Internet synchron abgespielt werden können.

IBM-Attacke Diese Attacke wird in [CMY1997] ausführlich beschrieben. Angenommen zwei Personen A und B behaupten die Urheberrechte an einem Bild O zu besitzen. A ist der wirkliche Eigentümer des Bildes O und hat mit Wasserzeichen W_A versehene Kopien O_A weitergegeben. B besitzt eine Kopie O_A und fügt selbst ein Wasserzeichen W_B ein. Das entstehende Bild $O_{A;B}$ enthält sowohl das Wasserzeichen W_A von A, als auch das Wasserzeichen W_B von B. Im Streitfall kann A behaupten, daß das "Orginalbild" von B O_A das Wasserzeichen von A enthält. Umgekehrt enthält das wirkliche Orginalbild O von A kein Wasserzeichen von B und somit beansprucht A zurecht die Rechte an dem Bild O. Was passiert aber, wenn B, statt ein Wasserzeichen einzufügen, eines entfernt und dieses Bild $O_{-B;A}$ als sein "Orginalbild" bezeichnet? Dann enthält zwar $O_{-B;A}$ das Wasserzeichen W_A, aber das wirkliche Orginalbild O und das mit W_A markierte Bild O_A enthalten beide im Vergleich zu dem gefälschten Orginalbild $O_{-B;A}$ auch das Wasserzeichen W_B. Formal ausgedrückt:

$$\begin{aligned} O_A &= O + W_A \\ O_{-B;A} &= O_A - W_B \\ \Rightarrow O_A &= O_{-B;A} + W_B \end{aligned}$$

Eine Lösung für diese Problematik ist es das Wasserzeichen abhängig vom Bildinhalt zu machen. A kann W_A mittels O generieren. B hat jedoch für die Erstellung von W_B noch keinen Zugriff zu seinem "Orginal" $O_{-B;A}$, da dieses noch nicht existiert.

Mehrfach-Markierung Dazu wird mit demselben Watermarking-Verfahren, aber einem unterschiedlichen Schlüssel und/oder Text über das alte Wasserzeichen ein neues aufgebracht. Das alte Wasserzeichen

sollte dann noch lesbar sein. Diese Eigenschaft ist sehr vorteilhaft, da hiermit auch die Möglichkeit zu hierarchischem Watermarking gegeben ist. Damit ließen sich auch über mehrere Vertriebsstufen Wasserzeichen einbringen.

StirMark StirMark [Kuh1997] ist ein speziell zur Wasserzeichenentfernung entwickeltes Tool das kleinste geometrische Störungen in das Bild einbringt und das Bild resampled. Weiterhin wird ein schwaches niederfrequentes Rauschen aufaddiert. Die eingebrachten Störungen sind nicht sichtbar und mit denen durch Ausdrucken und anschließendem Scanning vergleichbar. Viele Tests die mit StirMark und kommerziellen Watermarking-Verfahren durchgeführt wurden zeigen, daß StirMark die meisten Wasserzeichen entfernt ohne die Bildqualität zu beeinträchtigen. Auch die in dieser Studie untersuchten Wasserzeichen werden von dieser Attacke schwer gestört. Das Bildraumverfahren aus Ansatz II zeigt jedoch eine gewisse Basisrobustheit gegen diese Attacke, wie an der Ergebnistabelle zum Video 1 (s. Seite 74) sehr gut zu sehen ist. Die Wasserzeicheninformation, die über Das Frequenzraumverfahren in Ansatz I eingebracht wird, überlebt die StirMark-Attacke praktisch nicht.

2.3 Veröffentlichte Verfahren

Um einen Überblick über verschiedene Ansätze für Watermarking-Mechanismen zu bekommen, werden nun diverse Verfahren vorgestellt. Viele der vorgestellten Verfahren, vor allem die Bildraumverfahren, sind vom Ansatz zunächst für Bilder gedacht. Trotzdem lassen sich natürlich mit jedem Verfahren einzelne decodierte Frames von Videos markieren und anschließend wieder encodieren.
Ein Großteil aller bekannten Verfahren läßt sich in verschiedene Kategorien einteilen. Zum einen gibt es Verfahren, die zum Auslesen des Wasserzeichens das Orginalbild benötigen und andere nicht. Letztere haben den Vorteil, daß ein Computer das Internet ständig nach illegalen Kopien durchsuchen kann. Außerdem können Browser, sofern es sich um ein öffentliches Wasserzeichen handelt, die Copyright-Informationen anzeigen. Im Gegensatz dazu sind Verfahren, die das Orginalbild benötigen meistens robuster. Die Ausleseverfahren

können dann eher Attacken auf das Wasserzeichen erkennen und entsprechende Gegenmaßnahmen durchführen.
Eine andere Einteilung bezieht sich darauf, ob die Wasserzeicheninformation im Bildraum, durch direkte Änderung von Farb- oder Helligkeitswerten, oder im Frequenzraum durch Änderung entsprechender Koeffizienten, eingebracht wird. Die im folgenden vorgestellten Verfahren werden zunächst hiernach unterteilt.
Da die hauptsächlich verwendeten Kompressionstechniken JPEG und MPEG von Bildern und Videos die Bilddaten mittels DCT in den Frequenzraum transformieren, arbeiten auch die meisten Watermarking-Verfahren mit der DCT. Prinzipiell kann jedoch ein Wasserzeichen auch in anders transformierten Bilddaten, z.B. über DFT (Discrete Fourier Transform), eingebracht werden. Der Vorteil der Frequenzraumverfahren ist, daß die Wasserzeicheninformation direkt in die komprimierten Bild-/Videodaten eingebracht werden können. Damit läßt sich die Markierung in Echtzeit vornehmen. Außerdem können diese Verfahren die Eigenschaften des menschlichen visuellen Wahrnehmungsvermögens besser ausnutzen, da die Wahrnehmung von Bildstörungen frequenzabhängig ist.
Der Vorteil der Bildraumverfahren liegt in der höheren Anschaulichkeit. Attacken wie geometrische Translationen, die im Bildraum arbeiten, können leichter begegnet werden, da ersichtlicher ist, was mit dem Wasserzeichen passiert.

2.3.1 Verfahren im Frequenzraum

Die folgende Auswahl an Frequenzraumverfahren soll einen Eindruck geben, wie prinzipiell unterschiedlich verschiedene Watermarking-Verfahren sein können. Das Verfahren von Zhao und Koch wurde für diese Arbeit ausgewählt, weil die Robustheit gegenüber Kompression in dem Verfahren praktisch frei gewählt werden kann. Dies geschieht über eine Vorab-Quantisierung der zu verändernden Frequenzkomponenten, so daß die verlustbehaftete Quantisierung während des MPEG-Encodings die Wasserzeicheninformation nicht löscht.

Jian Zhao, Eckhard Koch, [ZhK1995] Ihr Verfahren wird in Kapitel 3.2 ausführlich dargestellt, da Ansatz I auf ihrem Algorithmus basiert.

Kurz zusammengefaßt bringt ihr Verfahren ein Bit Wasserzeicheninformation pro 8x8 DCT-Block ein. Dazu werden drei DC-Koeffizienten aus dem mittleren Frequenzbereich so verändert, daß ihr Verhältnis zueinander ein Bit darstellt.

Frank Hartung, Bernd Girod, [HaG1996] Um eine Bitsequenz $a_j \in \{-1, 1\}$ in ein Bild einzubringen, werden die Informationsbits zunächst um einen Faktor cr (chip-rate) vervielfacht.

$$b_i = a_j, \qquad j \cdot cr \leq i \leq (j+i) \cdot cr$$

Die gespreizte Informationssequenz b_i wird nun mit einem Amplitudenfaktor α sowie einer binären Zufallssequenz $p_i \in \{-1, 1\}$ moduliert. Das so erhaltene Wasserzeichensignal $w_i = \alpha \cdot b_i \cdot p_i$ wird auf das Bildsignal v_i aufaddiert und man erhält das markierte Bildsignal $\hat{v}$.

$$\hat{v}_i = v_i + \alpha \cdot b_i \cdot w_i$$

Zum Auslesen der Wasserzeicheninformation wird das Bildsignal wieder mit der gleichen Zufallssequenz p_i multipliziert. Es ergibt sich:

$$s_j = \sum_{(j+1)\cdot cr-1}^{i=j\cdot cr} p_i \cdot \hat{v}_i = \sum_{(j+1)\cdot cr-1}^{i=j\cdot cr} p_i \cdot v_i + \sum_{(j+1)\cdot cr-1}^{i=j\cdot cr} p_i^2 \cdot \alpha \cdot b_i$$

Der erste Term auf der rechten Seite sollte ungefähr Null ergeben, da der Mittelwert der Zufallssequenz p_i auch Null ist. Die rechte Seite ergibt sich dann zu:

$$s_j = \sum_{(j+1)\cdot cr-1}^{i=j\cdot cr} p_i \cdot \hat{v}_i \approx cr \cdot \alpha \cdot a_j$$

Das ausgelesene Bit ergibt sich aus dem Vorzeichen von s_j.
Das oben beschriebene Verfahren arbeitet zunächst noch im Bildraum. Um im Frequenzraum das Wasserzeichensignal w_i einzubringen, wird jeweils ein 8x8 Block des Wasserzeichensignals transformiert und auf den 8x8 DCT-Block aufaddiert. Damit läßt sich das Wasserzeichen direkt in komprimierte JPEG-Bilder und MPEG-Videos einbringen. Um bei MPEG-Videos die Datenrate nicht zu erhöhen, dies ist nötig damit Hardwarepuffern nicht der Speicher ausgeht oder die Synchronisation zwischen Audio und Video nicht verlorengeht, wird für jeden zu

ändernden DC-Koeffizienten geprüft, ob das entsprechende Codewort (VLC's in MPEG) gleich lang oder kürzer ist als für den orginalen DC-Koeffizienten. Erst dann wird das entsprechende Wasserzeichensignal aufaddiert. Um Fehler in nachfolgenden Frames, die durch die motion-compensation entstehen können, zu vermeiden, wird zusätzlich zu dem Wasserzeichensignal eine *drift-compensation* berechnet und aufaddiert [HaG1997b].
Das Verfahren bringt die Wasserzeicheninformation im hochfrequenten Bereich ein, da binäre Zufallssequenzen benutzt werden. Die Robustheit gegenüber Kompressionsverfahren, die bekanntlich hochfrequente Informationen am stärksten komprimieren, wird nur über den hohen Redundanzfaktor *cr* erreicht. Daraus folgt, daß die meisten Bildänderungen überflüssig sind, weil sie während des MPEG-Encoding Prozesses verloren gehen. Deshalb wird das Verfahren als nicht gut geeignet für die Markierung von MPEG-Videos mit großen Datenmengen eingestuft.

G.C. Langelaar, J.C.A. van der Lubbe, R.L. Lagendijk [LLB1997]
Die Autoren stellen zwei verschieden Algorithmen vor, um Wasserzeichen in MPEG-Videos einzubringen. Das erste Verfahren ändert die niederwertigsten Bits von VLC's sofern ein gleich langer VLC mit gleicher *run-length* und einem Levelunterschied von 1 existiert. Obwohl die Datenrate des Wasserzeichens recht hoch ist (1.6 Kbit/s bei 2.0 Mbit/s Videodatenrate) und das Wasserzeichen sich in Echtzeit einbringen läßt, ist das Verfahren ähnlich wie Bildraumverfahren, die die niederwertigsten Bits der Pixeldaten benutzen, sehr empfindlich gegenüber jeder Art von Angriffen.
Das zweite Verfahren bringt ein Bit durch Einfügen einer Energiedifferenz in den DCT-Koeffizienten ein. Dazu wird ein Block mit n 8x8 DCT-Blöcken in zwei Gruppen mit n/2 8x8 Blöcken unterteilt und die Summe der Quadrate aller DCT-Koeffizienten in beiden Gruppen berechnet. Um das Bit einzufügen wird eine der beiden Summen verkleinert indem DCT-Koeffizienten hoher Frequenzen einfach gelöscht werden. Dadurch wird auch garantiert, daß die Datenrate des Videos nicht erhöht, sondern sogar verringert wird. Über n kann die Robustheit des eingebrachten Bits bestimmt werden. Zum Auslesen eines Bits werden die beiden Summen berechnet und verglichen. Das Orginalvideo wird zum Auslesen nicht benötigt. Dieses Verfahren

scheint ebenfalls für die Markierung von MPEG-Videos geeignet zu sein. Dem Verfahren von Zhao-Koch wurde jedoch der Vorzug gegeben, weil dort in jedem 8x8 Block ein Bit eingebracht werden kann. Das oben beschrieben Verfahren benötigt dazu $n(n >= 2)$ Blöcke.

I. J. Cox, J. Kilian, T. Leighton, T. Shamoon [CKL1996] Dieses Verfahren bringt n reelle, normalverteilte Zahlen $x_1...x_n$ in den n niedrigsten Frequenzen (DCT) $v_1...v_n$ ein. Dies kann entweder bildunabhängig $v_i' = v_i + \alpha x_i$ oder bildabhängig $v_i' = v_i(1 + \alpha x_i)$ bzw. $v_i' = v_i(e^{\alpha x_i})$ geschehen. Zum Auslesen wird das Orginal benötigt, welches von dem zu untersuchendem Bild subtrahiert wird. Die erhaltene Differenz X^* wird über einen Ähnlichkeitstest

$$sim(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X}\sqrt{X \cdot X}}$$

mit dem eingebrachten Wasserzeichen verglichen. Dieses Verfahren wurde in mehreren Quellen als sehr robust eingestuft. Es ist jedoch anfällig gegenüber einer bereits im Kapitel 2.2.2 beschriebenen Attacke (Ermittelung der Watermark-Funktion). Außerdem benötigt das Verfahren das Orginalbild zum Auslesen der Wasserzeicheninformation, was ebenfalls ein Nachteil für die automatischen Bildverfolgung im Internet ist.

2.3.2 Verfahren im Bildraum

Die Bildraumverfahren sind zumeist etwas schlechter als die Frequenzraumverfahren in bezug auf die Datenrate und die Sichtbarkeit des Wasserzeichens. Trotzdem sind einige Ideen sehr interessant, wie das in dieser Arbeit angewendete Verfahren von Fridrich. Im Hinblick auf die StirMark-Attacke ist das Fridrich-Verfahren sogar robuster als das erweiterte Frequenzraumverfahren von Zhao und Koch, wie im Experimentalteil (Kapitel 5) noch deutlich gezeigt wird.

Jiri Fridrich [FBS1998] Dieses Verfahrens ist Grundlage für den zweiten Ansatz in dieser Studie und wird in Kapitel 3.3 ausführlich beschrieben. Die Idee ist aus einem Schlüssel mit Hilfe eines zellularen Automatens ein niederfrequentes Muster zu erzeugen, welches auf das Bild aufaddiert wird.

Martin Kutter, Frederic Jordan, Frank Bossen [KJB1997] Kutter et al. benutzen den Blaukanal eines Bildes um Wasserzeicheninformation einzubringen. Ein Bit s des Wasserzeichens wird an einer Position (i,j), die über einen Zufallszahlengenerator erzeugt wird, helligkeitsabhängig eingebracht:

$$B_{ij} = B_{ij} + (2s - 1)L_{ij}q$$

Dabei beschreibt q die Wasserzeichenstärke, L_{ij} die Helligkeit des Bildes an der Position (i,j) und B_{ij} den Wert des Blaukanals an der Position (i,j).
Zum Auslesen der Wasserzeicheninformation s an der Position (i,j) werden die Durchschnittswerte des Blaukanals in der Kreuznachbarschaft der Größe c mit dem Wert an der Position (i,j) selbst verglichen:

$$\hat{B}_{ij} = \frac{1}{4c}(\sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{ij})$$

Ist B_{ij} größer als $\hat{B}_{ij}$, dann wird 1 als eingebrachtes Bit angenommen, ansonsten 0. Die Idee des Verfahrens beruht auf einer statistischen Überlegung. Es wird angenommen, daß die Blauwerte der umgebenden Kreuznachbarschaft ähnliche Werte haben, wie der Punkt selbst. Wird der Blauwert des Punktes nun deutlich größer oder kleiner als der Durchschnittswert der umgebenden Blauwerte gemacht, kann damit ein Bit codiert werden. Um die Robustheit des Verfahrens zu erhöhen wird jedes Bit mehrfach eingebracht.
Um das Wasserzeichen aus geometrisch transformierten Bildern wiederzugewinnen, werden der Wasserzeicheninformation zwei Bit "01" vorangestellt. Diese formen im Bild ein festes Muster. Nun werden mit dem veränderten Bild inverse Transformationen durchgeführt und jeweils versucht das fest eingebrachte Muster auszulesen. Gelingt dies, ist die richtige Transformation gefunden worden und das restliche Wasserzeichen kann ausgelesen werden. Die Suche nach der richtigen Transformation kann erheblich eingeschränkt werden, wenn die Art der Transformation (z.B. eine reine Rotation oder Skalierung) bekannt ist. Die Idee um aus geometrisch transformierten Bildern die Wasserzeicheninformation wieder auszulesen, kann problemlos auf die in dieser Arbeit verwendeten Verfahren übertragen werden.
Das oben beschriebene Watermarking-Verfahren wird als nicht robust

eingestuft, da die eingebettete Information von der exakten Position der zum Auslesen verwendeten Pixel abhängt. Ein Angriff, der z.B. einzelne benachbarte Blauwerte gegeneinander austauscht, würde die Wasserzeicheninformation zerstören. Aufgrund der Einbettung der Information in exakt einem Pixel kann diese Bildänderung ebenfalls als hochfrequent betrachtet werden. Deshalb scheint dieses Verfahren nicht für die Markierung von MPEG-Videos geeignet zu sein.

Germano Caronni [Car1995] Caronni stellt ein einfaches Verfahren vor, welches das Bild in einzelne Blöcke unterteilt und abhängig vom einzubringenden Bit die Helligkeit der Blöcke erhöht bzw. erniedrigt. Zum Auslesen benutzt Caronni das Orginal.
Dieses Verfahren ähnelt vom Prinzip her sehr stark dem Verfahren von Fridrich. Der Nachteil des Caronni-Verfahrens ist jedoch, daß die markierten Blöcke aufgrund der Blockkanten in manchen Bildbereichen stark auffallen.

Nikos Nikolaidis, Ioannis Pitas [NiP1996] Das erste Verfahren dieser Autoren teilt das Bild mittels Zufallszahlengenerator in zwei Hälften ein. Um die Wasserzeicheninformation einzubringen, wird auf eine Hälfte ein konstanter Wert k aufaddiert. Um das Vorhandensein des Wasserzeichens zu testen, werden die Durchschnittshelligkeiten der beiden Hälften voneinander abgezogen und mit einem Schwellenwert verglichen. Ein annähernd identisches Verfahren wird auch in [BGM1996]beschrieben.
Da das Wasserzeichensignal sehr hochfrequenter Natur ist, ist das Verfahren sehr anfällig gegen JPEG und andere Tiefpaßfilter. Deshalb geben die Autoren ein verbessertes Verfahren an, in dem die Teilbilder aus 2x2 oder 3x3 Blöcken bestehen. Dadurch wird das einzubettende Signal niederfrequenter, aber auch sichtbarer.
Eine andere Verbesserungsmöglichkeit beruht auf der Idee nicht alle Pixel um den Wert k zu modulieren, sondern einige Pixel stärker und andere Pixel entsprechend schwächer zu ändern. Um ein niederfrequenteres Wasserzeichensignal zu erhalten, werden jeweils 8x8 Blöcke des Bildes mittels DCT transformiert und die einzelnen Pixel so moduliert, daß der Anteil hoher Frequenzen minimal ist. Die Gesamtänderung der Pixel bleibt jedoch gleich, d.h. bei r zu ändernden Pixeln in einem 8x8

Block muß die Summe der Änderungen $r \cdot k$ sein.
Auch dieses Verfahren ähnelt dem Fridrich-Verfahren. Die Grundidee ist immer dieselbe. Das Helligkeitsbild wird über irgendein Verfahren in zwei Teile zerlegt. Caronni benutzt einfache Blöcke, Nikolaidis et al. benutzen zufällig ausgewählte Pixel und Fridrich benutzt ein aus zufällig ausgewählten Pixel berechnetes Muster. Ein Teil wird dann, je nach einzufügendem Bit heller oder dunkler gemacht. Das von Fridrich vorgestellte Verfahren scheint dabei das ausgereifteste zu sein und wurde deshalb für die Markierung der MPEG-Videos ausgewählt.

2.4 Kommerzielle Produkte

Einige kommerzielle Watermarking-Programme befinden sich inzwischen auf dem Markt und wurden in [Rin1997b] ausführlich getestet. Das marktführende Programm PictureMarc von DigiMarc [Dig1998] gibt es sogar als Plug-in für Adobe Photoshop 4.0, CorelDraw und Micrografx Graphics Suite und wird demnächst sogar Bilder der Corbis-Bildgalerie schützen. Trotzdem übersteht selbst dieses Produkt eine Attacke des Antiwatermark-Tools StirMark nicht.

Kapitel 3

Zwei Watermarking-Verfahren

Aus den im letzten Kapitel beschriebenen Verfahren wurden zwei herausgesucht, verbessert und auf Verwendbarkeit für MPEG-Videos überprüft. Dabei fiel die Wahl auf das Frequenzraumverfahren von Zhao und Koch [ZhK1995] und das Bildraumverfahren von Fridrich [FBS1998]. Beide Verfahren haben Vor- und Nachteile. Das Verfahren von Zhao und Koch hat den bereits erwähnten Vorteil, daß die Information direkt in das komprimierte Bild/Video eingebracht werden kann. Darüberhinaus ist die Datenrate recht hoch und das Verfahren benötigt zum Auslesen das Orginal nicht. Das Verfahren bringt die Wasserzeicheninformation in mittleren bis niedrigen Frequenzen ein und scheint damit eher als die anderen Frequenzraumverfahren für die Markierung von MPEG-Videos geeignet zu sein. Da die Information jedoch unabhängig von den Bildeigenschaften eingebracht wird, gibt es bereits bei geringer Wasserzeichenstärke sichtbare Bildstörungen. Dies tritt vor allem im Bereich von Kanten und glatten Flächen auf. Um diesen Nachteil auszugleichen wird das Verfahren dahingehend verbessert, daß die Wasserzeichenstärke sich automatisch an die Bildeigenschaften anpasst. Diese Anpassung wird in der "smooth-block/edge detection"-Phase vorgenommen. Um die Fehlerraten des Verfahrens zu minimieren, wird das Wasserzeichen mittels eines "intelligenten" Redundanzcodes, der die Zuverlässigkeit ausgelesener Bits berücksichtigt, und eines (31, 6, 15)-BCH-Codes codiert. Der Redundanzcode besteht einfach aus einer Vervielfachung der eingebrachten Information. In den durchgeführten Experimenten wurde hierfür vierfache Redundanz in beiden Verfahren verwendet. Beim Auslesen wird die Wasserzeichenstärke, die über die *smooth-block/edge detection*-Phase ermittelt

wird, berücksichtigt. Sind in vier ausgelesenen Bits, die ein eingebrachtes Wasserzeichenbit darstellen, gleich viele Nullen und Einsen, entscheidet das Informationsbit, welches mit der höchsten Wasserzeichenstärke eingebracht wurde. Der benutzte BCH-Code codiert 6 Wasserzeichenbits in eine Codewortlänge von 31 und hat eine Hammingdistanz von 15 ([TsH1993]). Damit ist er in der Lage bis zu 7 beliebig plazierte Fehler im Codewort zu korrigieren.
Hier noch einmal die Vor- und Nachteile sowie die Verbesserungen für das Frequenzraumverfahren zusammengefaßt:

Vorteile:

- Möglichkeit zur direkten Einbettung des Wasserzeichens in den MPEG-Bitstrom.
- Hohe Datenrate des Wasserzeichens.
- Auslesen des Wasserzeichens ist ohne Orginal möglich.
- Einbettung des Wasserzeichens in mittleren bis niedrigen Frequenzen und damit höhere Robustheit gegen Kompressionsverfahren.

Nachteile:

- Uniforme d.h. nicht an Bildeigenschaften angepaßte Wasserzeichenstärke.
- Fehlender fehlerkorrigierender Code.

Verbesserungen:

- *smooth-block/edge detection*-Phase in der die Wasserzeichenstärke an die Bildeigenschaften und damit an das menschliche visuelle Wahrnehmungsvermögen angepaßt wird.
- Einfügen von fehlerkorrigierenden Codes.

Das Fridrich-Verfahren wurde als Bildraumverfahren ausgewählt, weil es Nachteile anderer Bildraumverfahren, wie Anfälligkeit gegen Tiefpaßfilter (Kutter et al. [KJB1997]) und leichte Detektion des Wasserzeichens (Caronni [Car1995]) nicht hat. Das Fridrich-Verfahren hat trotzdem einige Nachteile, die es auszugleichen gilt. Das Verfahren benötigt noch das Orginal zum Auslesen des Wasserzeichens und die Datenrate ist extrem klein. Ersterem wird mit einem statistischen Ausleseverfahren begegnet. Letzterem wird durch Verkleinerung der markierten Blöcke begegnet, wodurch jedoch die Fehlerrate ansteigt. Deshalb wird auch hier ein Redundanzcode sowie ein (31, 6, 15)-BCH-Code benutzt, um die Wasserzeicheninformation zu codieren. Die "smooth-block/edge detection"-Phase zur Reduktion visueller Artefakte wird beim Fridrich-Verfahren genauso angewendet, wie beim Zhao-Koch Verfahren.
Die Vor- und Nachteile sowie die eingefügten Verbesserungen für das Bildraumverfahren noch einmal im Überblick:

Vorteile:

- Die generierten Muster sind schwerer zu erkennen als z.B. feste Blockmuster.
- Starke Abhängigkeit des Musters vom Benutzerschlüssel und damit erhöhte Sicherheit.
- Die Musterbeschaffenheit hat niedrigen Frequenzcharakter und eignet sich damit am ehesten für die Markierung von MPEG-Videos.

Nachteile:

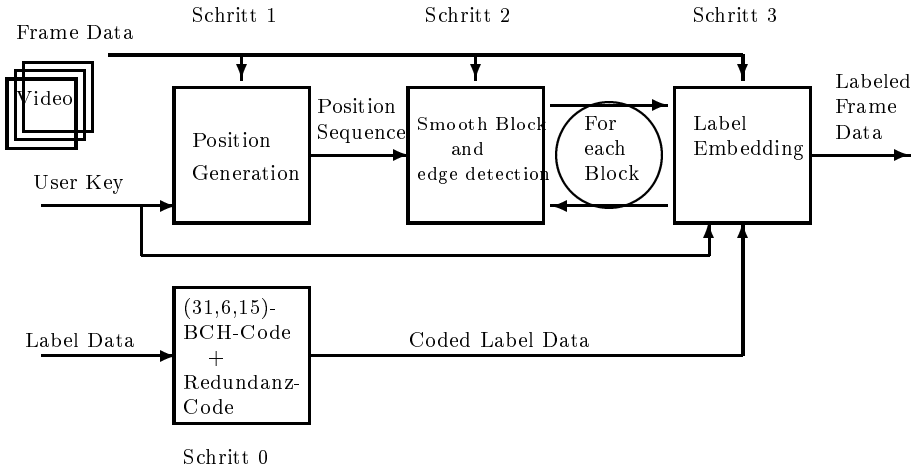
- Uniforme d.h. nicht an Bildeigenschaften angepaßte Wasserzeichenstärke.
- Fehlender fehlerkorrigierender Code.
- Niedrige Datenrate (1 Bit).
- Abhängigkeit vom Orginal.

Verbesserungen:

- *smooth-block/edge detection*-Phase in der die Wasserzeichenstärke an die Bildeigenschaften und damit an das menschliche visuelle Wahrnehmungsvermögen angepaßt wird.
- Einfügen von fehlerkorrigierenden Codes.
- Erhöhung der Datenrate durch Verwendung kleinerer Muster.
- Unabhängigkeit vom Orginal durch statistisches Ausleseverfahren.

Beide Verfahren betten die Information in die Helligkeitskomponente der Bildinformationen ein, da diese am schwächsten komprimiert wird. Die Farbkomponenten werden noch nicht zur Markierung benutzt. Sie bieten deutlich weniger Raum zum Markieren, da sie bis zu Faktor 4 stärker komprimiert werden. In späteren Erweiterungen der Verfahren können sie jedoch mitberücksichtigt werden.
Das Einbetten der Wasserzeicheninformation (Label Data) läßt sich für beide Verfahren in einem allgemeinen Schema darstellen:

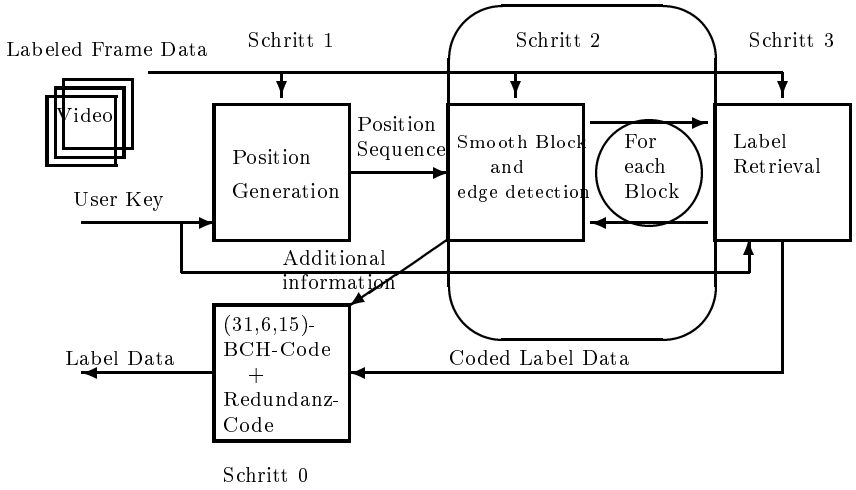
Abbildung 3.1: Schema zum Einbetten der Wasserzeicheninformation



In Schritt 1 (Position Generation) wird die Reihenfolge der Blöcke festgelegt in der einzelne Blöcke des Bildes markiert werden. Diese Reihenfolge ist von einem Schlüssel (User Key) abhängig, der von dem Benutzer vorgegeben wird und ohne den das Auslesen des Wasserzeichens nicht möglich ist. Die Blockgröße ist im Frequenzraumverfahren (Ansatz 1) fest vorgegeben (8x8

Pixel). Das Bildraumverfahren (Ansatz 2) erlaubt eine variable Einstellung der Blockgröße in festen Schritten von jeweils 8 Pixeln (8x8, 16x16, 24x24 ...).
In Schritt 3 wird in Abhängigkeit von Schritt 2 in dem die Wasserzeichenstärke festgelegt wird, die codierte Wasserzeicheninformation (Coded Label Data) eingebracht. Dieser Schritt wird in den nächsten beiden Unterkapiteln ausführlich dargestellt.
Das Auslesen des Wasserzeichens kann in einem ähnlichen Schema dargestellt werden. Schritt 2 (Smooth block and edge detection) ist hierfür nicht mehr notwendig und kann bei zeitkritischen Anwendungen weggelassen werden. Da die Stärke des Wasserzeichens jedoch ein Maß für die Zuverlässigkeit der ausgelesenen Bits ist und damit beim Decodieren als Entscheidungskriterium bei gestörten Blöcken (gleiche Anzahl von Nullen und Einsen im Redundanzcode) dienen kann, bleibt dieser Schritt zunächst erhalten.

Abbildung 3.2: Schema zum Auslesen der Wasserzeicheninformation



3.1 "Smooth-block/edge detection"

Die *smooth-block/edge detection*-Phase ist die entscheidende Verbesserung der beiden Verfahren. Sie liefert für einen 8x8 Block mit Helligkeitswerten einen Indikator der die visuelle Kapazität des Blocks beschreibt. Dieser

Tabelle 3.1: Quantisierungsmatrix Q_m

niedrige Frequenzen							
16	**11**	**10**	**16**	**24**	40	51	61
12	**12**	**14**	19	26	58	60	55
14	**13**	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99
							hohe Frequenzen

Indikator zeigt an wie groß Störungen in diesem Block sein dürfen ohne sichtbar zu werden und ist damit das Maß wie stark ein Wasserzeichen in diesem Block eingebracht werden darf. Das Verfahren zur Berechnung dieses Indikators ist bewußt einfach gehalten, da sie in späteren Versionen im MPEG-Strom und in Echtzeit durchgeführt werden soll.
Zunächst wird der 8x8 Block mittels DCT in den Frequenzraum transformiert und mit der Matrix Q_m quantisiert.

Die Transformation benötigt in Ansatz I keinen zusätzlichen Rechenaufwand, da alle Blöcke bereits transformiert vorliegen. Zwei Parameter werden nun berechnet: *smooth* gibt im Prinzip die Anzahl der DCT-Koeffizienten an, die nach Quantisierung nicht Null sind. Für Absolutwerte größer Eins wird jedoch 2 auf *smooth* aufaddiert und für Absolutwerte kleiner oder gleich Eins nur 1. Dies resultiert aus der Erfahrung, daß häufig fast glatte Regionen viele Einsen in ihren quantisierten Frequenzanteilen enthalten. Ansonsten deuten hohe Werte von *smooth* auf viele Frequenzanteile und damit auf eine große Toleranz gegenüber zusätzlichen Störungen durch ein Wasserzeichen hin. Dies resultiert aus der Tatsache, daß das menschliche visuelle Wahrnehmungsvermögen Störungen in hohen Frequenzbereichen wesentlich schwächer wahrnimmt als Störungen in niedrigen Frequenzbereichen. Allerdings ist dieses Verfahren alleine nicht zuverlässig, weil Blöcke mit scharfen Kantenverläufen, nach DC-Transformation, auch viele Frequenzanteile enthalten.

Solche Blöcke sollen aber nicht stärker markiert werden. Deshalb wird ein zweiter Parameter eingeführt. Dieser Parameter *edge* ist die Summe aus den Absolutwerten der DCT-Koeffizienten 1, 2, 3, 4, 8, 9, 10, 16, 17, 24 und 32, die die niedrigen Frequenzanteile des Blockes darstellen. Die entsprechenden Positionen sind in der Quantisierungsmatrix Q_m fett markiert. Hohe Werte in diesen niedrigen Frequenzanteilen deuten auf Kanten und damit auf eine geringe Toleranz gegenüber zusätzlichen Störungen durch ein Wasserzeichen hin. Werden beide Parameter entsprechend kombiniert kann erwartet werden, daß Blöcke mit hohen Frequenzanteilen und damit einer hohen visuellen Toleranz gegen eingebrachte Störungen stark markiert, gleichzeitig aber Blöcke mit Kantenverläufen immer noch schwach markiert werden. Eine Linearkombination aus beiden Parametern ergibt den Indikator *Level*. $Level = smoothscale \cdot smooth + edgescale \cdot edge + offset$. Über *offset* wird eine Basiswasserzeichenstärke eingestellt. Schließlich wird *Level* noch mit der von außen einstellbaren Wasserzeichenstärke *WatermarkStrength* gewichtet: $Level = Level/WatermarkStrength$. Da *Level* negativ werden kann, wird *Level* auf den Bereich zwischen 0 und 50 begrenzt: Wenn Level $\geq 50 \Rightarrow$ Level $= 50$, wenn Level $\leq 0 \Rightarrow$ Level $= 0$. Schließlich wird dieser Wert zur besseren Handhabung auf den Bereich zwischen 1 und 11 skaliert, also: Level = (Level / 5 + 1). Je kleiner *Level* ist, desto stärker kann das Wasserzeichen sein ohne das sichtbare Störungen im Bild zu erkennen sind. Erfahrungswerte aus Experimenten haben ergeben, daß für *smoothscale* Werte zwischen -8 (schwache Markierung von hochfrequenten Bereichen) und -12 (starke Markierung von hochfrequenten Bereichen), für *edgescale* Werte zwischen 0.3 (Kanten werden schwach markiert) und 0.6 (Kanten werden stark markiert), für *offset* Werte zwischen 30 (schwache Basismarkierung) und 60 (starke Basismarkierung) und für *WatermarkStrength* Werte zwischen 1.0 (schwaches Wasserzeichen) und 3.0 (starkes Wasserzeichen) sinnvoll sind. Alle Parameter lassen sich über die Parameterdatei frei einstellen. Beim Einstellen von *smoothscale* und *edgescale* ist zu beachten, daß beide Parameter sich auch gegenseitig beeinflussen. Wird *smoothscale* auf ein starke Markierung von hochfrequenten Bereichen gestellt, werden auch Kanten stärker markiert und wird *edgescale* auf eine schwache Markierung von Kanten eingestellt werden auch hochfrequente Bereiche entsprechend schwächer markiert. Eine optimale Einstellung erfordert viele Erfahrungswerte.

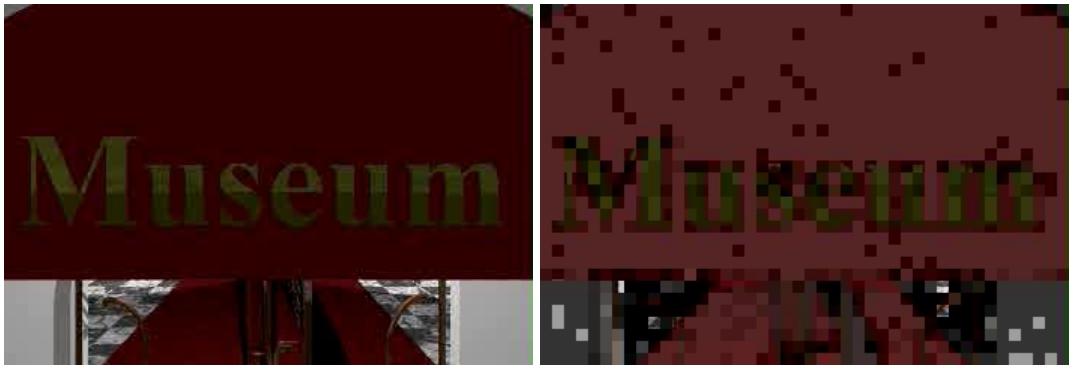


Abbildung 3.3: Blockklassifizierung durch die *smooth-block/edge detection*-Phase am Beispiel eines computergenerierten Bildes

Abbildung 3.3 und 3.4 zeigen anhand der jeweils ersten Frames aus den ersten beiden untersuchten Videos in Kapitel 5 die Leistungsfähigkeit der *smooth-block/edge detection*-Phase. Blöcke, die stärker markiert werden sollen sind dunkler dargestellt. Besonders in Abbildung 3.3 im unteren Teil sieht man deutlich anhand der schwarzen Blöcke, daß die hochfrequenten Teile des Bildes erkannt und als stark markierbar klassifiziert werden.

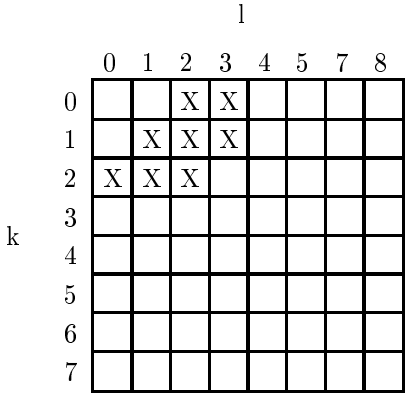
Abbildung 3.4: Blockklassifizierung durch die *smooth-block/edge detection*-Phase am Beispiel einer Szene aus dem Film "North by North-West" von Alfred Hitchcock

3.2 Ansatz I: Modulierung im Frequenzraum

Ansatz I beruht auf [ZhK1995]. Der Algorithmus zum Einbetten und Auslesen der Wasserzeichenbits entspricht dem der Veröffentlichung. Hinzu kommt die *smooth-block/edge detection*-Phase und der fehlerkorrigierende Code.

Abbildung 3.5: Geeignete Frequenzkombinationen

Set Nr.	(k_1, l_1)	(k_2, l_2)	(k_3, l_3)
1	2(0,2)	9(1,1)	10(1,2)
2	9(1,1)	2(0,2)	10(1,2)
3	3(0,3)	10(1,2)	11(1,3)
4	10(1,2)	3(0,3)	11(1,3)
5	9(1,1)	2(0,2)	10(1,2)
6	2(0,2)	9(1,1)	10(1,2)
7	9(1,1)	16(2,0)	2(0,2)
8	16(2,0)	9(1,1)	2(0,2)
9	2(0,2)	9(1,1)	16(2,0)
10	9(1,1)	2(0,2)	16(2,0)
11	10(1,2)	17(2,1)	3(0,3)
12	17(2,1)	10(1,2)	3(0,3)
13	10(1,2)	3(0,3)	17(2,1)
14	3(0,3)	10(1,2)	17(2,1)
15	9(1,1)	16(2,0)	17(2,1)
16	16(2,0)	9(1,1)	17(2,1)
17	10(1,2)	17(2,1)	18(2,2)
18	17(2,1)	10(1,2)	18(2,2)



3.2.1 Einbetten der Wasserzeicheninformation

Gegeben sind n Blöcke $B_{0..n-1}(0..7, 0..7)$, die die frequenztransformierten Blöcke mit Helligkeitsinformationen darstellen. Weiterhin sind 6/31 * n/BITREDUNDANZ Bits Orginal-Wasserzeicheninformation gegeben

Tabelle 3.2: Frequenzmuster zur Bitcodierung

Bit	1	1	1	1	0	0	0	0	invalid		
(k_1, l_1)	H	H	M	M	L	L	M	M	H	L	M
(k_2, l_2)	H	M	H	M	L	M	L	M	L	H	M
(k_3, l_3)	L	L	L	L	H	H	H	H	M	M	M

aus denen über einen (31, 6, 15)-BCH Code und einem anschließenden Redundanzcode (BITREDUNDANZfache Wiederholung des Orginalbits) n Bits einzubettende Information $c_{0..n-1}$ entstehen. Ein Bit c wird durch die Einstellung eines Verhältnisses zwischen drei Frequenzkomponenten im mittleren Bereich eines Blocks B dargestellt (siehe Tabelle 3.2). Zhao und Koch haben 18 verschiedene Kombinationen dreier Frequenzkomponenten als geeignet qualifiziert (siehe Abbildung 3.5).

Vorgehensweise:

1. $i = 0$
2. Wenn $i \leq n \Rightarrow$ fertig.
3. Untersuche Block B_i mittels *smooth-block/edge detection*. Mit dem berechneten Level (siehe Kapitel 3.1, lege einen zusätzlichen Quantisierungsfaktor Q_z folgendermaßen fest: Q_z = Level-1. Wenn Q_z=0 ist, dann Q_z=1. In diesem Quantisierungsfaktor steckt die Wasserzeichenstärke. Je stärker ein Block quantisiert ist, desto stärker wirken sich die Änderungen durch das Wasserzeichen aus.
4. Quantisiere B_i mit $Q_m(0..63) \cdot Q_z$.
5. Rufe **check_write**(B_i, c_i) auf. Liefert diese Funktion **true** zurück, dann sind die ausgewählten Frequenzkomponenten geeignet ein Bit zu codieren und die Funktion **write**(B_i, c_i) wird aufgerufen. Andernfalls sind die Änderungen durch das einzubettende Wasserzeichenbit zu gravierend und die Frequenzkomponenten werden in ein *invalid*-Muster (siehe Tabelle 3.2) abgeändert (wird bereits in check_write durchgeführt).

6. De-quantisiere B_i, transformiere den erhaltene Block in den Bildraum zurück (**IDCT** **I**nverse **D**iscrete **C**osine **T**ransform) und überschreibe damit den ursprünglichen Helligkeitsblock.

7. Inkrementiere i und gehe zurück nach 2).

Funktion: check_write()
In **check_write** wird getestet, ob die Frequenzkomponenten geeignet sind ein Bit zu codieren. Liegen die einzufügenden Änderungen durch das Wasserzeichen über einen Schwellwert MD, werden die Frequenzkomponenten in ein *invalid*-Muster (siehe Tabelle 3.2) abgeändert.

1. Wähle pseudo-zufällig (über den *User Key*) eine der 18 Kombinationen aus Abbildung 3.5 aus. Die Absolutwerte an den Positionen (k_1, l_1), (k_2, l_2) und (k_3, l_3) werden mit Y1, Y2 und Y3 bezeichnet.

2. Wenn $c_i = 1$ und min(Y1,Y2)+MD $\leq$ Y3 ist, MD ist ein Schwellwert, der die maximal änderbare Distanz zum Referenzwert Y3 angibt, dann ändere Y1, Y2 und Y3 so ab, daß ihre Werte ein *invalid*-Muster aus Tabelle 3.2 darstellen, return **false**.

3. Wenn $c_i = 0$ und max(Y1,Y2)+MD $\geq$ Y3 ist, dann ändere Y1, Y2 und Y3 wiederum so ab, daß ihre Werte ein *invalid*-Muster aus Tabelle 3.2 darstellen, return **false**.

4. Andernfalls return **true**.

Funktion: write()
In der Funktion **write** werden die Frequenzwerte so abgeändert, daß sie ein Muster aus Tabelle 3.2 und damit ein Wasserzeichenbit darstellen.

1. Y1, Y2 und Y3 bezeichnen die Frequenzwerte, die bei **check_write** ausgesucht wurden.

2. Wenn $c_i = 1$, ändere Y1, Y2 und Y3 so ab, daß Y1 $\geq$ Y3+D und Y2 $\geq$ Y3+D ist. Je höher D ist, desto zuverlässiger ist das eingebettete Bit.

3. Wenn $c_i = 0$, ändere Y1, Y2 und Y3 so ab, daß Y1+D $\leq$ Y3 und Y2+D $\leq$ Y3 ist.

3.2.2 Auslesen der Wasserzeicheninformation

Zum Auslesen der Wasserzeicheninformation müssen in jedem Block die pseudo-zufällig ausgewählten Frequenzkomponenten auf die Muster in Tabelle 3.2 geprüft werden.

Vorgehensweise:

1. $i = 0$
2. Wenn $i \leq n \Rightarrow$ fertig.
3. Untersuche Block B_i mittels *smooth-block/edge detection*. Dies ist zum Auslesen nicht unbedingt notwendig. Die Information kann aber für die Decodierung des Redundanzcodes wichtig sein. Dies wird im Anschluß an die Algorithmusbeschreibung näher erläutert.
4. Rufe **check_read**(B_i, c_i) auf. Liefert diese Funktion **true** zurück, rufe **read**(B_i, c_i) auf. Ansonsten setze c_i auf 'i'.
5. Inkrementiere i und gehe zurück nach 2).

Funktion: check_read()
In **check_read** wird getestet, ob die pseudo-zufällig ausgewählten Frequenzkomponenten ein *invalid*-Muster aus Tabelle 3.2 darstellen. Ist dies der Fall gibt die Funktion false zurück, andernfalls true.

1. Wähle pseudo-zufällig (über den *User Key*) eine der 18 Kombinationen aus Abbildung 3.5 aus. Die Absolutwerte an den Positionen (k_1, l_1), (k_2, l_2) und (k_3, l_3) werden mit Y1, Y2 und Y3 bezeichnet.
2. Formen Y1, Y2 und Y3 eine der *invalid*-Muster aus Tabelle 3.2 return **false**, ansonsten return **true**.

Funktion: read()
Die Funktion **read** liest die tatsächliche Wasserzeicheninformation aus.

1. Y1, Y2 und Y3 bezeichnen die Frequenzwerte, die bei **check_read** ausgesucht wurden.

2. Wenn Y1 $\geq$ Y3 und Y2 $\geq$ Y3 ist, return 1.
3. Wenn Y1 $\leq$ Y3 und Y2 $\leq$ Y3 ist, return 0.
4. Andernfalls return 2, um zu markieren, daß die Information aus dem Block zerstört worden ist.

Zum Decodieren der ausgelesenen Bits werden für den Redundanzcode jeweils BITREDUNDANZ Bits betrachtet. Sie können die Werte '0', '1', '2' und 'i' enthalten. '2' bedeutet zerstörte Information und 'i' bedeutet, daß keine Information in den entsprechenden Block eingebettet wurde. Ob die BITREDUNDANZ Bits als '0' oder '1' interpretiert werden, wird durch die Anzahl der '0'- und '1'-Werte entschieden. Sind gleich viele '0'- und '1'-Werte vorhanden entscheidet der Wert dessen Block auf die höchste Wasserzeichenstärke, die in der *smooth-block/edge detection*-Phase festgestellt wurde, schließen läßt. Nach weiterer Decodierung mittels des (31, 6, 15)-BCH-Codes ist die Orginalinformation wieder hergestellt.

3.3 Ansatz II: Modulierung im Bildraum

Ansatz II beruht auf [Fri1998]. Dieses im Bildraum arbeitende Verfahren bringt ein Bit Information in ein Frame ein, indem ein niederfrequentes binäres Muster auf die Helligkeitsinformation addiert wird. Beim Auslesen des Wasserzeichens kann lediglich entschieden werden, ob das untersuchte Frame markiert worden ist oder nicht. Dazu wird das Orginalframe abgezogen und das Differenzbild auf Korrelation mit dem eingebrachten Muster überprüft. Fridrich zeigt, daß dieses Verfahren robust gegen Tiefpaßfilter, Kompressionsverfahren (JPEG), Ausschnittbildung und Rauschaddition ist. In **Ansatz II** wird dieses Verfahren um folgende Eigenschaften verbessert:

Erhöhung der Datenrate Durch Verkleinerung der aufaddierten Muster kann die Datenrate erheblich gesteigert werden.

Fehlerkorrigierender Code Mit Erhöhung der Datenrate steigt auch die Fehleranfälligkeit des Verfahrens. Diesem Problem wird mit den gleichen fehlerkorrigierenden Codes wie in Ansatz I (BCH-, und Redundanzcode) entgegengewirkt.

Anpassung der Wasserzeichenstärke Auch in diesem Ansatz kann, genau wie in Ansatz I, die Wasserzeichenstärke an die Bildeigenschaften angepasst werden. Dazu wird die in Ansatz I bereits beschriebene "smooth-block/edge detection" benutzt.

Unabhängigkeit vom Orginal Um zum Auslesen nicht immer das Orginal benutzen zu müssen, wird ein Korrelationsverfahren, wie in [HaG1996] benutzt.

3.3.1 Mustergenerierung

Die prinzipielle Art dieses Verfahrens, Einbettung eines Musters in die Helligkeitsinformation eines Bildes, ist nicht neu und wird bereits in [Car1995] beschrieben. Neu ist jedoch die Art des eingebrachten Musters. Aus einem Schlüssel (*User Key*) wird zunächst ein hochfrequentes Bitmuster generiert. Dieses Muster wäre jedoch aufgrund des hochfrequenten Charakters anfällig gegen diverse Angriffe wie Tiefpaßfilter, Kompressionsverfahren (insbesondere JPEG) und Rauschaddition. Deshalb wird mittels eines zellularen Automaten mit einfachen Entscheidungsregeln aus diesem Muster ein Niederfrequenteres berechnet. Der Vorteil dieses Musters, neben der Eliminierung eben erwähnter Nachteile, ist zum einen die eindeutige und einfache Generierung aus einem Schlüssel und zum anderen die schwer ausmachbare Lokalisation, die z.B. bei einfachen Rechteckmustern nicht gegeben ist. Die Generierung des Musters wird an einem 8x8 Block erklärt, arbeitet jedoch synonym für größere Blöcke.
Der zellulare Automat arbeitet wie folgt:
Zuerst wird aus dem *User Key* ein 8x8 Pixel großes Zufallsmuster erzeugt (Schritt 1). Um hohe Frequenzen zu eliminieren wird nun ein zellularer Automat mit zwei einfachen Regeln benutzt: Für jede Position im Muster wird die Anzahl der benachbarten Einsen berechnet (max. 8). Ist die Anzahl größer als 5 wird die Position selbst auf '1' gesetzt, ist sie kleiner als 3 wird die Position auf '0' gesetzt. Wird dieser Vorgang mehrmals wiederholt, entsteht ein niederfrequenteres Muster M (Schritt 2-4). Die Addition von '1' auf das zu markierende Bild ist jedoch zu schwach. Deshalb wird jede Position im Muster mit einer '1' durch '3' und jede Position mit '0' durch '-3' ersetzt (Schritt 5). Um die Sichtbarkeit des Musters weiter zu reduzieren wird vor der Einbettung jeder Wert im Muster M noch mit dem Durchschnittswert

aller angrenzenden Positionen ersetzt (Schritt 6). Abbildung 3.6 zeigt den Vorgang der Mustergenerierung an einem Beispiel und Abbildung 3.3.1 zeigt ein Frame des ersten Testvideos in dem die Helligkeitsinformation durch das Muster (20fach verstärkt) ersetzt worden ist.

Schritt 1

1	0	0	0	1	0	1	1
1	0	0	0	0	1	1	1
0	1	0	0	0	0	1	1
1	0	1	0	0	0	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1

Schritt 2

1	0	0	0	0	0	1	1
1	0	0	0	0	1	1	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1

Schritt 3

1	0	0	0	0	0	1	1
1	0	0	0	0	1	1	1
1	1	0	0	0	0	1	1
1	1	1	0	0	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1

Schritt 4

1	0	0	0	0	0	1	1
1	0	0	0	0	1	1	1
1	1	0	0	0	0	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	1	0	1
1	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1

Schritt 5

3	-3	-3	-3	-3	-3	3	3
3	-3	-3	-3	-3	3	3	3
3	3	-3	-3	-3	-3	3	3
3	3	3	-3	-3	3	3	3
3	3	3	3	3	3	-3	3
3	-3	3	3	3	-3	-3	-3
-3	-3	-3	-3	-3	-3	-3	3
-3	-3	-3	-3	-3	-3	3	3

Schritt 6

0	-1	-3	-3	-2	0	1	3
1	0	-2	-3	-2	0	2	3
2	1	-1	-2	-2	0	2	3
3	2	1	0	0	0	1	2
2	2	2	2	1	0	0	0
0	0	0	1	0	-1	-1	-1
-2	-2	-2	-1	-2	-2	-1	0
-3	-3	-3	-3	-3	-2	0	1

Abbildung 3.6: **Schritt 1** Generierung eines Zufallsmusters über den "User Key"; **Schritt 2-4** Generierung des niederfrequenten Musters M mit den Regeln: #1 $> 5 \Rightarrow 1$ und #1$< 3 \Rightarrow 0$; **Schritt 5-6** Glättung des Musters.

3.3.2 Einbetten der Wasserzeicheninformation

Zum Einbetten eines Bits c in einen Block B durchläuft Block B die bereits in Ansatz I beschriebene "smooth-block/edge detection"-Phase. Aus dem berechneten *Level* ergibt sich die Gewichtung des Musters M und damit die Wasserzeichenstärke w: $w = 6/$ Level. Die Werte aus M werden dann einfach mit w multipliziert.Ist c=1 wird das gewichtete Muster auf den Block B addiert, ist c=0 wird das gewichtete Muster subtrahiert.

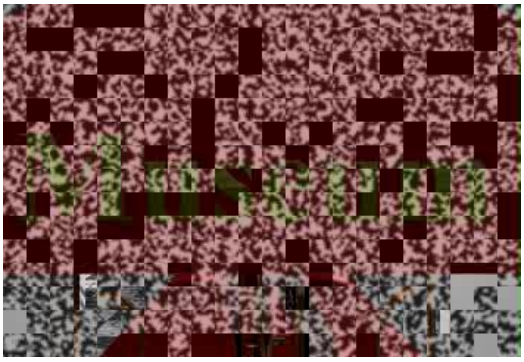
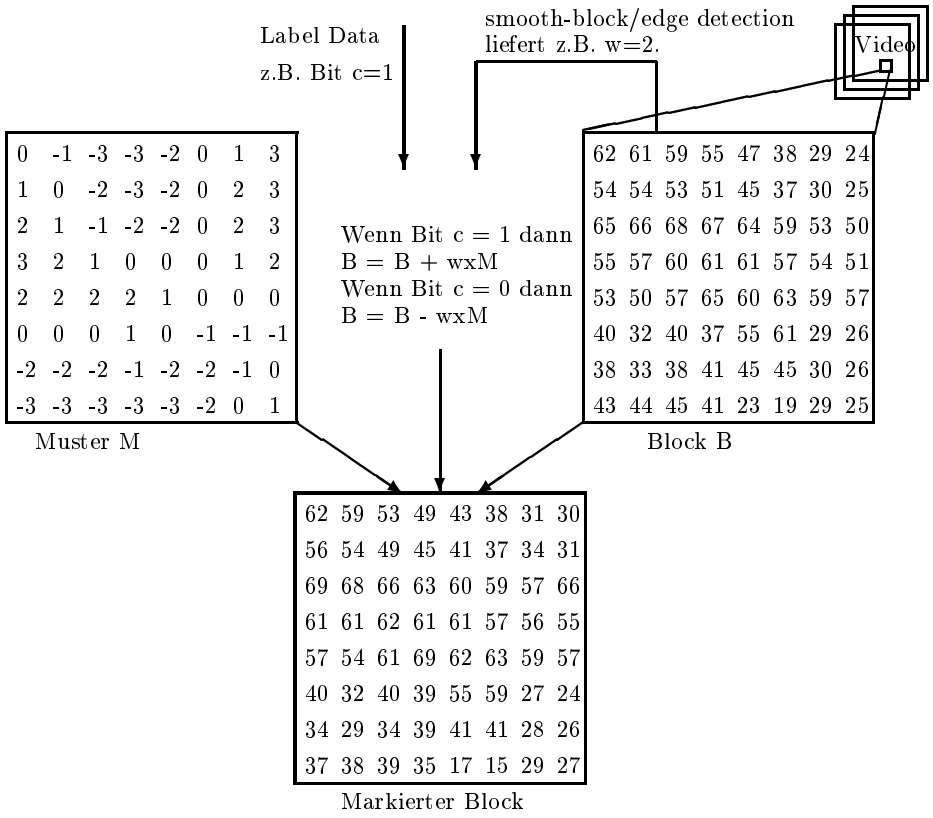


Abbildung 3.7: Beispielbild generierter Muster der Größe 16x16

3.3.3 Auslesen der Wasserzeicheninformation

Zum Auslesen eines Bits aus einem markierten Block B wird die im Einbettungsprozeß eingebrachte Korrelation zwischen dem Bild und dem Muster ausgenutzt. Zuerst wird wieder das Muster M das zum Einbetten der Information benutzt wurde generiert. Nun werden zwei Durchschnittswerte gebildet. Alle Positionen in Block B, die an korrespondierenden Positionen in Muster M einen positiven Wert haben werden aufaddiert. Die so gebildete Summe wird durch die Anzahl der aufaddierten Positionen geteilt. Dasselbe geschieht mit den Helligkeitswerten in Block B, die an korrespondierenden Positionen in Muster M einen negativen Wert haben. Sind die Helligkeitswerte von Block B und die Werte in Muster M voneinander unabhängig, ist der Erwartungswert der Differenz der beiden Durchschnittswerte 0. Aufgrund des Einbettungsprozesses wurde jedoch ein Korrelation zwischen Muster M und Block B eingefügt. Je nach eingefügtem Bit wurde eine Summe um ca. $2 \cdot w$ vergrößert und die andere Summe um ca. $2 \cdot w$ verkleinert. w bezeichnet dabei die Wasserzeichenstärke, die sich aus der "smooth-block/edge detection-"Phase beim Einbetten der Information ergab. Der Erwartungswert der Differenz beider Durchschnittswerte liegt damit bei ungefähr $\pm 4 \cdot w$ und dient als Entscheidungskriterium für das eingebettete Bit.
Bei einer Blockgröße von 8x8 Pixeln reicht bei ca. 10% aller Blöcke die eingefügte Korrelation nicht aus, um das Bit einzubetten. Diese Fehlerrate muß durch den fehlerkorrigierenden Code abgefangen werden. Um die Bildqualität der markierten Frames zu erhöhen kann man bereits beim Einbringen eines Bits testen, ob die eingebrachte Korrelation ausreicht das Bit zu codieren.

Abbildung 3.8: Einbetten eines Informationsbits



Reicht sie nicht aus kann man den entsprechenden Block auslassen, da die Information in diesem Block sowieso fehlerhaft ausgelesen wird. Der entsprechende Test, Auslesen des Bits und Vergleich mit dem eingebrachten Bit, dauert natürlich genauso lange, wie der Ausleseprozeß. Ob dies durchgeführt werden kann ist eine Frage an die Rechenzeitanforderungen im Einbettungsprozeß.

Kapitel 4

Implementierungsteil

Die beiden untersuchten Ansätze wurden in einen frei erhältlichen MPEG-Decoder mit dem entsprechenden Source-Code ([MPE1994]) eingebettet. Beide Watermarking-Mechanismen sind in der Datei **watermark.c** untergebracht. Desweiteren existiert eine Klasse *Parameter* in dem File **parameter.C** und **parameter.h**, die für das Einlesen und die Verwaltung der Parameter beider Algorithmen aus der Datei **Params.txt** verantwortlich ist. Zum Abspeichern von statistischen Daten wie den eingebrachten Wasserzeichenbits **WtrmarkBits.txt** oder dem jeweiligen Blockstatus **Blockstatus.txt** im Zhao-Koch Verfahren ist die Klasse *Statistik* zuständig. In ihr wurden während der Entwicklungsphase auch diverse andere Daten, wie Korrelationswerte im Fridrich Verfahren oder Level-Werte aus der *smooth block/edge detection*-Phase gespeichert, die jedoch für den Ablauf des Programms nicht notwendig sind.
Der Decoder wird über die Kommandozeile gesteuert und zu den diversen Optionen wurde eine **w-Option** zum Einbetten und Auslesen eines Wasserzeichens und eine **d-Option** zum Festlegen eines Debug-Levels hinzugefügt. Letztere Option wird zum Einbetten oder Auslesen eines Wasserzeichens nicht benötigt. Sie dient nur dazu verschiedene zusätzliche Informationen während des Betriebs anzuzeigen. Eine Besonderheit ist der Debug Level der Stufe 2 im Fridrich Verfahren. Ist er eingestellt werden die Blöcke der Frames mit einem Grauwert markiert, der dem Level der *smooth block/edge detection*-Phase entspricht (siehe Abbildung 3.3 und 3.4).
Ein Aufruf, um z.B. in das Video *test.mpg* ein Wasserzeichen einzubringen sieht folgendermaßen aus:

mpeg2decode -b test.mpg -w1 -o3 frame%d

Dabei dient die **b-Option** zur Angabe des zu markierenden/decodierenden Videos und die **o3-Option** sagt, daß die erzeugten Frames im PPM-Format abgespeichert werden. Die %d-Angabe in "frame%d" gibt die Position für die Nummerierung der erzeugten Frames an. Mit der Datei **Params.txt** im aufrufenden Verzeichnis in der der Algorithmustyp, Parameter für die *smooth block/edge detection*-Phase, der User Key, der Wasserzeichentext und die Blockgröße bei Verwendung des Fridrich Verfahrens angegeben sind, entstehen die bereits erwähnten Statistikdateien **WtrBits.txt** und **Blockstatus.txt** sowie die decodierten und markierten Frames **frame0.ppm, frame1.ppm, frame2.ppm ...**. Aus diesen Frames kann mit jedem handelsüblichen MPEG-Encoder wieder ein Video erstellt werden. In den durchgeführten Experimenten (Kapitel 5 wurde der ebenfalls frei erhältliche Decoder der MPEG Software Simulation Group ([MPE1994]) verwendet. Genauere Details zur Verwendung dieses Encoders werden im Kapitel 5 erläutert. Nach dem Encodieren kann das Wasserzeichen aus dem markierten Video (z.B. markedtest.mpg) wieder mit Hilfe des Decoders und der **w2-Option** ausgelesen werden:

mpeg2decode -b markedtest.mpg -w2 -o3 markedframe%d

Die dabei erzeugten Bilder werden nicht weiter benötigt, können aber zur Überprüfung der Bildqualität nach dem Encodingprozeß dienen. Die folgende Liste beschreibt alle für das Projekt wichtigen Dateien:

bch3.c, bch3.h Diese beide Dateien enthalten den Code des BCH-Error Correcting Codes und entstammen aus einem frei erhältlichen BCH-En/Decoders, der so umgeschrieben wurde, daß nur der verwendete (31, 6, 15)BCH-Code benutzt wird. Mit *initbchencoder()* werden die notwendigen Initialisierungen durchgeführt. *encode_bch(vektor)* encodet einen 6-elementigen Vektor und liefert den 31-elementigen Code in diesem Vektor zurück. *decode_bch(vektor)* decodet entsprechend einen 31-elementigen Vektor. Das decodete 6-elementige Wort steht in den letzten sechs Elementen des übergebenen Vektors.

fdctref.c, idct.c Mit den Funktionen *fdct(Block)* bzw. *Fast_IDCT(Block)* aus diesen Dateien wird ein 8x8 Block mittels diskreter Cosinus Transformation in den Frequenzraum bzw. vom Frequenzraum in den Bildraum transformiert.

global.h In dieser Datei werden die zusätzlichen Optionen (**w-Option** und **d-Option**) des Decoders für die Verwendung der Wasserzeichen Algorithmen deklariert.

mpeg2dec.c ist die Datei mit der Einstiegsfunktion des Decoders. Hier werden die zusätzlichen Optionen definiert.

parameter.C, parameter.h enthalten die Klasse Parameter, die beim Start des Decoders das Parameterfile **Params.txt** ausliest. Die benutzten Parameter sind:

SmoothScale Gibt einen Faktor an, wie stark in der *smooth block/edge detection*-Phase glatte Flächen markiert werden dürfen. Sinnvolle Werte liegen zwischen -8.0 (stark) und -15.0 (schwach).

EdgeScale Gibt einen Faktor an, wie stark in der *smooth block/edge detection*-Phase Kanten markiert werden dürfen. Sinnvolle Werte liegen zwischen 0.1 (stark) und 0.5 (schwach).

Offset Gibt den Basiswert der *smooth block/edge detection*-Phase vor und damit die Basisstärke des Wasserzeichens. Sinnvolle Werte liegen zwischen 20 (stark) und 60 (schwach).

WatermarkStrength Ist eine richtige Abstimmung von *SmoothScale*, *EdgeScale* und *Offset* gefunden worden, kann über diesen Parameter eine lineare Einstellung der Wasserzeichenstärke vorgenommen werden. Sinnvolle Werte hängen zwar von obigen Parametern ab, liegen aber in den durchgeführten Experimenten in Kapitel 5 zwischen 1.0 (schwach) und 2.5 (stark).

UserKey Als Benutzerschlüssel werden Integerzahlen benutzt.

AlgorithmusTyp Eine '1' in diesem Eintrag bedeutet, daß das Verfahren von Zhao und Koch (Ansatz 1) benutzt wird. Entsprechend steht eine '2' für das Fridrich Verfahren (Ansatz 2).

Blocksize Wird das Fridrich Verfahren zum Markieren benutzt, muß hier die benutzte Blockgröße als Vielfache von 8 eingestellt werden.

Multi In diesem Eintrag steht die Anzahl (Hierarchien) der Wasserzeichen, die eingebracht werden.

WatermarkText Für jede Wasserzeichenhierarchie muß hier ein Wasserzeichentext angegeben werden.

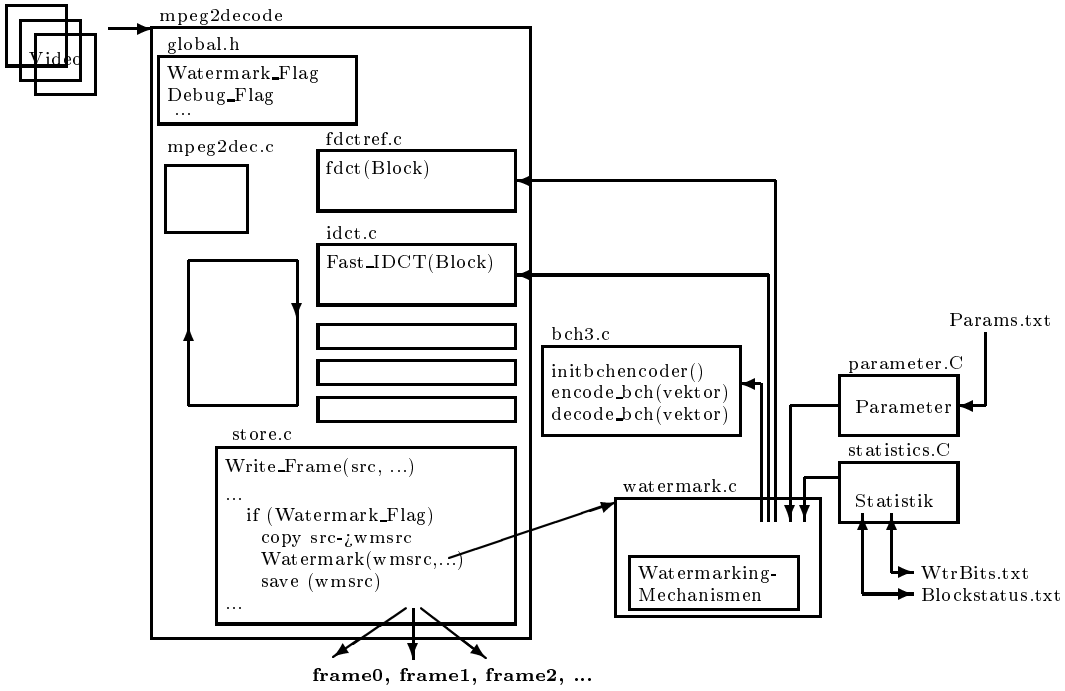
statistics.C, statistics.h enthalten die Containerklasse Statistik für die Ansammlung statistischer Daten, die während der Markierung anfallen. Für die Auswertung der Experimente speichert diese Klasse die eingebrachten Wasserzeichenbits in der Datei **WtrBits.txt** und die Blockstatusinformationen in der Datei **BlockStatus.txt**.

store.c In dieser Datei steht die Funktion *Write_Frame()*, die immer dann vom Decoder aufgerufen wird, wenn ein Frame in der Darstellungsreihenfolge vollständig decodiert ist. In *Write_Frame()* wird die w-Option abgefragt und dementsprechend die Hauptfunktion aus **watermark.c** *Watermark()* aufgerufen. Die zu markierenden Helligkeitsinformationen stehen in *SRC[0] und werden vor der Markierung nach *WMSRC[0] kopiert, da die Orginalinformationen aus *SRC[0] zur Decodierung weiterer Frames benötigt wird. Andernfalls würden sich die Wasserzeichenmarkierungen durch die *forward prediction* der MPEG-Codierung vervielfachen.

watermark.c, watermark.h sind die Hauptdateien und enthalten sämtliche Funktionen der beiden untersuchten Algorithmen. Eine ausführlich Beschreibung der Funktionen findet sich in der Headerdatei **watermark.h**.

Das Zusammenspiel der Dateien zeigt die Abbildung 4.1.

Abbildung 4.1: Zusammenspiel der beteiligten Dateien



Kapitel 5

Experimentalteil

In dem folgenden experimentellen Teil werden die Ansätze mit unterschiedlichen Parametereinstellungen an drei verschiedenen Videosequenzen getestet. Von jedem Video werden die ersten 30 Frames markiert. **Video 1** besteht aus einer Kamerafahrt durch ein virtuelles Museum. Die ersten 30 Frames zeigen den Eingang mit einem gelben Schriftzug "Museum" auf rotem Hintergrund und ein kurzes Wegzoomen. Dabei wird ein größerer Ausschnitt des Eingangs mit einer Drehtür sichtbar. Die Videosequenz enthält viele glatte, einfarbige Flächen (Wandbereich und Hintergrund des Schriftzugs), scharfe Kanten (Schrift- und Schattenbereiche) und wenig hochfrequente Bereiche. Dieses Video wurde ausgewählt, weil dieser Typ Video im allgemeinen schwer zu markieren ist, da gerade in hochfrequenten Bildbereichen die Wasserzeicheninformation am stärksten untergebracht werden kann, ohne zu sichtbaren Qualitätseinbußen im Video zu führen. Abbildung 5.1 zeigt das erste und letze Frame von **Video 1**.

Video 2 und **Video 3** zeigen jeweils eine kurze Sequenz aus dem Film "North by North-West" von Alfred Hitchcock. Sie enthalten eine Mischung aus glatten Flächen, kantigen und hochfrequenten Bereichen, wie sie in Filmen aus der realen Welt häufig vorkommt. An ihnen kann die Praxistauglichkeit der Verfahren getestet werden. Abbildung 5.2 und 5.3 zeigen wieder die ersten und letzten Frames der markierten Videosequenz von **Video 2** und **Video 3**.

Zum Encodieren der markierten Frames wurde der frei erhältliche Encoder der MPEG Software Simulation Group verwendet [MPE1994]. Es ist be-

Abbildung 5.1: Erstes und letztes Frame von Video 1

Abbildung 5.2: Erstes und letztes Frame von Video 2

kannt, daß verschiedene MPEG-Encoder durchaus gravierende Unterschiede in Bezug auf Bildqualität, Datenkomprimierung und Laufzeitverhalten haben können. Deshalb wird die Bildqualität des markierten Videos immer mit einem nicht markierten, decodierten und wieder encodierten Video verglichen.
Für das Bildraumverfahren (Ansatz II) wurden Testläufe mit den Blockgrößen 16 und 24 durchgeführt. Encodiert wurde mit 1 Mbit/s und 2 Mbit/s, um die Abhängigkeit der Robustheit vom Kompressionsfaktor zu untersuchen. Die Fehlerraten beim Auslesen des Wasserzeichens werden nach MPEG-Encodierung, Quicktimekonvertierung und dem StirMark-Angriff angegeben. Die Mehrfachmarkierung wurde nur mit der Blockgröße 24 und 2 Mbit/s Datenrate durchgeführt, da schwächere Einstellungen zu inakzeptablen Fehlerraten führen.

Abbildung 5.3: Erstes und letztes Frame von Video 3

Für das Frequenzraumverfahren (Ansatz I), das nur auf Blöcken der Größe 8 arbeitet, wurden unterschiedliche Wasserzeichenstärken bei Datenraten von 1 Mbit/s und 2 Mbit/s untersucht. Auch hier werden die Fehlerraten beim Auslesen des Wasserzeichens nach MPEG-Encodierung, Quicktimekonvertierung und dem StirMark-Angriff angegeben. Die Mehrfachmarkierung wurde ebenfalls bei einer Datenrate von 2 Mbit/s durchgeführt.
Für beide Verfahren wurden die Parameter der *smooth block/edge detection*-Phase (Smoothscale, Edgescale und Offset) optimal angepaßt, damit so wenig wie möglich visuelle Artefakte entstehen.
Für den StirMark-Angriff wurden 7 Frames "geStirMarkt" und anschließend mit einer Datenrate von 4 Mbit/s reencodiert. Die hohe Datenrate wurde benutzt, um die Schwere des StirMark-Angriffs für sich allein beurteilen zu können. Die Konvertierung des Quicktime-Videos zum MPEG-Video wurde ebenfalls bei einer Datenrate von 4 Mbit/s durchgeführt.
Die Fehlerraten in den Tabellen sind in Prozent angegeben und beziehen sich jeweils auf die Anzahl der eingebrachten Bits.

Auf eine Abbildung markierter Frames wurde bewußt verzichtet, da die Bildstörungen in einzelnen Frames nur bei sehr schlechter Videoqualität überhaupt sichtbar sind. Im Anhang sind einige automatisch generierte VRML-Szenen abgebildet, welche die Differenzen zwischen Orginalframes und markierten bzw. "geStirMarkten" Frames dreidimensional darstellen [DSN1998].

5.1 Video 1

Ansatz I – Frequenzraumverfahren

In den folgenden drei Versuchen wird jeweils der Einfluß der Wasserzeichenstärke auf die Bildqualität und der Einfluß der Datenrate des MPEG-Videos auf die Fehlerraten untersucht. In Versuch 4 wird der Einfluß von mehreren Wasserzeichen auf die Bildqualität und auf die Fehlerraten der einzelnen Wasserzeichen untersucht.

Versuch 1: Wasserzeichenstärke 1.5
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -10.0 Edgescale: 0.37 Offset: 40
Markierte Blöcke: 1240 Eingebrachte Bits: 60

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	3.7	6.0	20.3	6.3	6.3	29.1	40.0	36.7	39.2
2 Mbit/s	0.0	0.7	0.1	0.7	2.0	1.3	40.0	36.7	39.2

Versuch 2: Wasserzeichenstärke 2.0
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -10.0 Edgescale: 0.37 Offset: 40
Markierte Blöcke: 1240 Eingebrachte Bits: 60

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	0.0	1.7	8.6	1.3	2.7	13.9	42.5	30.0	25.8

Versuch 3: Wasserzeichenstärke 1.0
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -10.0 Edgescale: 0.37 Offset: 40
Markierte Blöcke: 1240 Eingebrachte Bits: 60

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
2 Mbit/s	1.7	2.7	9.0	6.0	7.3	17.5	38.3	45.0	47.9

Versuch 4: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -10.0 Edgescale: 0.37 Offset: 40

Wasserzeichenstärke: 1.5 / 1.0
Datenrate: 2 Mbit/s
Markierte Blöcke: 1240 Eingebrachte Bits: 2*60 / 3*60

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2 (1.5)	0.0	5.3	7.8	0.0	4.7	2.7

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3 (1.5)	0.7	6.3	11.2	0.0	0.7	9.2	0.0	0.0	7.2
3 (1.0)	12.0	16.0	22.1	6.3	10.3	22.7	6.3	5.3	23.5

Beschreibung der Bildqualität für Ansatz I

Die Bildqualität der markierten Videos muß bei Einbringung von verschiedenen Informationsbits in jedes einzelne Frame als schlecht bezeichnet werden. Obwohl in den einzelnen Frames kaum ein Unterschied zu den Orginalframes auszumachen ist, machen sich die unterschiedlichen Markierungen in aufeinanderfolgenden Frames deutlich bemerkbar. Besonders in den Kantenbereichen der Schrift und des Schattens sind Störungen auszumachen. Gelegentlich sind einzelne 8x8 Blöcke deutlich gestört, was auf eine fehlerhafte Erkennung in der *smooth block/edge detection*-Phase zurückzuführen ist. Selbst das am schwächsten markierte Video aus Versuch 3 ist noch durch das ungeübte Auge vom Orginalvideo zu unterscheiden. Ein weiteres Absenken der Wasserzeichenstärke führt jedoch zu hohen Fehlerraten, vor allem in B-Frames. Die Bildqualität der mehrfach markierten Videos ist erwartungsgemäß noch schlechter und inakzeptabel für kommerzielle Anwendungen.

Ansatz II – Bildraumverfahren

In den Tests des Bildraumverfahrens wird die Wasserzeichenstärke auf einem mittleren Level belassen und der Einfluß der Blockgrößen (16 bzw. 24) auf die Fehlerrate untersucht. In Versuch 2 wird wieder der Einfluß mehrerer Wasserzeichen auf die Fehlerraten und auf die Bildqualität untersucht.

Versuch 1: Wasserzeichenstärke 1.5
Parameter der *smooth block/edge detection*-Phase:

Smoothscale: -10.0 Edgescale: 0.27 Offset: 50
Markierte Blöcke: 248 * 16x16 / 124 * 24x24 Eingebrachte Bits: 12 / 6

Blockparameter	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
16 - 1 Mbit/s	0.0	11.7	16.2	13.3	20.0	27.2	4.2	16.7	8.2
16 - 2 Mbit/s	0.0	1.7	0.0	1.7	1.7	5.3	4.2	16.7	8.2
24 - 1 Mbit/s	0.0	0.0	1.8	6.7	6.7	10.5	8.3	0.0	8.3
24 - 2 Mbit/s	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	8.3

Versuch 2: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -10.0 Edgescale: 0.37 Offset: 50
Wasserzeichenstärke: 1.5
Markierte Blöcke: 248 Eingebrachte Bits: 2*12 / 3*12

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2	3.3	0.0	6.1	0.0	0.0	7.9

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3	0.0	0.0	14.9	0.0	6.7	15.8	6.7	3.3	8.8

Beschreibung der Bildqualität für Ansatz II

Die Bildqualität der markierten Videos in Ansatz II ist ähnlich schlecht, wie die in Ansatz I, wenn unterschiedliche Informationsbits in jedes Frame eingebracht werden. Es treten die bereits beschriebenen Artefakte in Kantenbereichen auf. Die Bildstörungen sind jedoch etwas niederfrequenterer Natur. Die Bildqualität des mit 24x24 Blöcken markierten Videos ist sichtbar besser, als die des mit 16x16 Blöcken markierten Videos.

5.2 Video 2

Die Tests für Video 2 und Video 3 sind ähnlich angelegt, wie für Video 1. Lediglich die Parameter für dies*mooth block/edge detection*-Phase wurden

etwas verändert, um den hochfrequenteren Bildcharakter besser zu berücksichtigen. Dementsprechend kann auch die Wasserzeichenstärke etwas angehoben werden.

Ansatz I – Frequenzraumverfahren

Versuch 1: Wasserzeichenstärke 2.0
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	0.0	2.1	9.4	0.0	4.0	11.8	36.5	38.5	35.9
2 Mbit/s	0.0	0.0	0.0	0.0	0.0	0.0	36.5	38.5	35.9

Versuch 2: Wasserzeichenstärke 2.5
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	0.0	0.0	2.4	0.0	0.0	2.5	37.8	33.3	35.9

Versuch 3: Wasserzeichenstärke 1.5
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
2 Mbit/s	0.0	0.0	0.0	0.0	0.0	0.0	38.5	41.0	35.3

Versuch 4: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30

Wasserzeichenstärke: 1.5 / 1.0
Datenrate: 2 Mbit/s
Markierte Blöcke: 1612 Eingebrachte Bits: 2*78 / 3*78

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2 (1.5)	0.0	0.5	1.8	0.0	0.0	1.6

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3 (1.0)	2.8	12.8	17.9	1.8	2.6	18.6	0.5	1.3	12.8

Beschreibung der Bildqualität für Ansatz I

Die Bildqualität des zweiten markierten Videos ist deutlich besser als die Bildqualität des ersten Videos. Trotzdem sind einige Störungen bei genauerer Betrachtung sichtbar. Auch hier fallen wieder besonders einige Blöcke in Kantenbereichen auf. Durchgängig ist in allen Videos ein leichtes Flimmern in den Bildbereichen feststellbar, die sich im Zeitablauf kaum verändern. Dies liegt wiederum an den unterschiedlichen Markierungen in aufeinanderfolgenden Frames. Auch die Bildqualität der mehrfach markierten Videos ist deutlich besser als in Video 1. Der Unterschied zum Orginalvideo ist jedoch immer noch sichtbar.
Es fällt auf, daß Störungen in dunklen Bildbereichen stärker sichtbar sind, als in hellen.

Ansatz II – Bildraumverfahren

Versuch 1: Wasserzeichenstärke 2.0
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 40
Markierte Blöcke: 372 * 16x16 / 124 * 24x24 Eingebrachte Bits: 18 / 6

Blockparameter	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
16 - 1 Mbit/s	0.0	1.1	6.4	1.1	2.2	8.8	33.3	33.3	33.3
16 - 2 Mbit/s	0.0	0.0	0.9	0.0	0.0	1.4	33.3	33.3	33.3
24 - 1 Mbit/s	0.0	0.0	6.1	0.0	0.0	7.0	16.7	16.7	16.7
24 - 2 Mbit/s	0.0	0.0	0.0	0.0	0.0	0.0	16.7	16.7	16.7

Versuch 2: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 40
Wasserzeichenstärke: 2.0 / 1.5
Markierte Blöcke: 372 Eingebrachte Bits: 2*18 / 3*18

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2 (2.0)	0.0	0.0	0.6	1.0	1.1	2.3

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3 (1.5)	8.9	8.9	9.6	18.9	18.9	16.4	8.9	0.0	6.1

Beschreibung der Bildqualität für Ansatz II

Auch in diesem Versuchsblock fällt sofort die deutlich bessere Bildqualität im Gegensatz zu Video 1 auf. Das Flimmern des Bildes, welches durch die unterschiedlichen Markierungsmuster verursacht wird, ist etwas stärker als in Ansatz I. Die Abhängigkeit der Bildqualität von der benutzten Blockgröße fällt in diesem Video stärker ins Gewicht, als bei Video 1. Die Bildqualität bei allen Versuchen ist durchgängig etwas schlechter als in Ansatz I.

5.3 Video 3

Ansatz I – Frequenzraumverfahren

Versuch 1: Wasserzeichenstärke 2.0
Parameter der *smooth block/edge detection*-Phase:

Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	0.0	1.3	20.4	0.0	2.3	20.4	46.8	42.3	48.7
2 Mbit/s	0.0	0.0	0.0	0.0	0.0	0.1	46.8	42.3	48.7

Versuch 2: Wasserzeichenstärke 2.5
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
1 Mbit/s	0.0	0.0	6.6	0.0	0.0	7.9	36.5	35.9	46.5

Versuch 3: Wasserzeichenstärke 1.5
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Markierte Blöcke: 1612 Eingebrachte Bits: 78

Datenrate	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
2 Mbit/s	0.0	0.0	0.2	0.0	0.0	1.1	41.7	38.5	44.9

Versuch 4: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 30
Wasserzeichenstärke: 1.5 / 1.0
Datenrate: 2 Mbit/s
Markierte Blöcke: 1612 Eingebrachte Bits: 2*78 / 3*78

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2 (1.5)	0.3	4.4	9.9	0.0	2.8	7.8

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3 (1.0)	2.6	6.7	14.8	1.8	5.1	13.8	0.0	1.5	8.8

Beschreibung der Bildqualität für Ansatz I

In diesem Video sind die wenigsten Artefakte zu sehen. Die Bildqualität ist durchgängig etwas besser, als im zweiten Video. Die Qualität des am schwächsten markierten Videos (Video 3) ist so gut, daß Testpersonen den Unterschied zum nicht markierten Video im direkten Vergleich nicht wahrnehmen konnten. Das Flimmern des Bildhintergrundes ist noch wahrnehmbar, fällt aber nicht mehr so stark ins Gewicht, wie in Video 2. Störender sind hier Artefakte in Kantenbereichen.
Die Bildqualität der mehrfach markierten Videos ist jedoch immer noch nicht gut.

Ansatz II – Bildraumverfahren

Versuch 1: Wasserzeichenstärke 2.0
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 40
Markierte Blöcke: 372 * 16x16 / 124 * 24x24 Eingebrachte Bits: 18 / 6

Blockparameter	MPEG			Quicktime			StirMark		
	I	P	B	I	P	B	I	P	B
16 - 1 Mbit/s	4.4	3.3	15.2	4.4	4.4	15.5	19.4	11.1	17.4
16 - 2 Mbit/s	1.1	3.3	3.8	1.1	3.3	4.7	19.4	11.1	17.4
24 - 1 Mbit/s	0.0	0.0	14.9	0.0	3.3	18.4	0.0	0.0	20.8
24 - 2 Mbit/s	0.0	0.0	0.9	0.0	0.0	1.8	0.0	0.0	20.8

Versuch 2: Multiple Watermarking
Parameter der *smooth block/edge detection*-Phase:
Smoothscale: -8.0 Edgescale: 0.47 Offset: 40
Wasserzeichenstärke: 2.0 / 1.5
Markierte Blöcke: 372 Eingebrachte Bits: 2*18 / 3*18

Anzahl WM	WM 1			WM 2		
	I	P	B	I	P	B
2 (2.0)	0.0	2.2	4.1	5.6	4.4	4.4

Anzahl WM	WM 1			WM 2			WM 3		
	I	P	B	I	P	B	I	P	B
3 (1.5)	1.1	3.3	2.0	3.3	1.1	9.1	2.2	0.0	7.0

Beschreibung der Bildqualität für Ansatz II

Auch in Ansatz II ist die Bildqualität in diesem Video am besten. Aber wiederum ist sie etwas schlechter, als in Ansatz I. Das Flimmern des Bildhintergrundes fällt in Ansatz II stärker auf.

Kapitel 6

Ergebnisse

Die Problematik bei der Markierung von Video 1 fällt sofort auf. Durch das Fehlen hochfrequenter Blöcke muß entweder die Wasserzeichenstärke soweit zurückgenommen werden, daß die Fehlerrate inakzeptabel hoch wird oder aber die Wasserzeichenstärke ist ausreichend, dann ist die Bildqualität inakzeptabel. Das Problem ist bereits bei Watermark-Verfahren für Bilder bekannt. Als Gegenmaßnahme muß hier eine deutliche Reduzierung der eingebrachten Bits in Betracht gezogen werden, was z.B. durch das wiederholte Einbringen gleicher Informationen in aufeinanderfolgenden Frames getan werden kann. Entweder muß der Benutzer interaktiv entscheiden, ob das Video nur für wenig Wasserzeicheninformation geeignet ist oder der Computer kann automatisch aufgrund einfacher Kriterien, wie z.B. große einfarbige Flächen oder einem Schwellenwert für die Anzahl scharfer Kanten, diese Entscheidung treffen und entsprechen weniger Bits einbringen.
An Video 1 kann auch besonders gut die noch gelegentlich fehlerhaft arbeitende *smooth block/edge detection*-Phase beurteilt werden. In Kantenbereichen sind häufig einzelne stark gestörte Blöcke zu sehen. Wird jedoch der Parameter *EdgeScale* zu empfindlich eingestellt werden hochfrequente Blöcke nicht mehr ausreichend markiert. Dies liegt daran, daß hochfrequente Blöcke ein ähnliches Frequenzspektrum wie Blöcke mit Kantencharakteristiken haben. Allerdings heißt die gelegentliche Fehleinstufung eines Blocks nicht, daß die *smooth block/edge detection*-Phase nicht benutzt werden kann. Möglicherweise liegt das Problem in einer falschen Parameterkombination der Parameter *EdgeScale, SmoothScale, Offset* und *WatermarkStrength*. Obwohl in langen Vorversuchen und während der Entwicklung der Algorithmen bereits gute

Einstellungen gefunden wurden, variiert die optimale Einstellung doch stark mit der Art des zu markierenden Videos. In Video 2 und Video 3 arbeitet die *smooth block/edge detection*-Phase wesentlich besser. Dies liegt natürlich auch daran, daß der Parameter *Edgescale* von 0.37 auf 0.47 geändert wurde und damit eine empfindlichere Einstellung statt gefunden hat. Dies wiederum wurde nur dadurch möglich, daß in den letzten beiden Videos mehr hochfrequente Bereiche markierbar sind, so daß sich die schwächeren Markierungen nicht so stark auf die Fehlerrate auswirken.
Interessant ist die Auswirkung des Kompressionsfaktors bzw. Datenrate beim Reencoding-Prozeß auf die Fehlerrate. Bei einer Datenrate von 1 Mbit/s liegen die Fehlerraten in I- und P-Frames meist zwischen 0.0% und 5.0%. Diese Werte sind vielversprechend, wenn man sich eine sinnvolle Wasserzeichendatenrate überlegt. Eine Wasserzeichendatenrate von nur 100 Bit/s wäre z.B. ausreichend, um jede Szene des Videos mit diversen Copyrightinformationen inklusive einer bereits früher erwähnten Identifikationsnummer (S. 21) zu versehen. Bei momentan 60-80 Bits/s (je nach Bildgröße) für das Frequenzraum-Verfahren aus Ansatz I und 25 Frames/s ergibt sich eine Wasserzeichendatenrate von 1500 bis 2000 Bits/s. Damit erhält man einen weiteren Redundanzspielraum um den Faktor 15 bis 20, um die Fehlerrate noch deutlich zu reduzieren. Für das Bildraum-Verfahren aus Ansatz II liegt dieser Redundanzspielraum bei derselben angestrebten Wasserzeichendatenrate nur bei 1.5 bis 5 (je nach benutzter Block- und Bildgröße), da 4 bis 10 mal weniger Information in jedes Frame eingebracht wird. Dieser Spielraum ist zu klein, um die Fehlerrate, vor allem in B-Frames, auf akzeptable Werte zu senken. Für dieses Verfahren ist daher nur eine Wasserzeichendatenrate von 30 bis 50 Bits/s realistisch.
Eine Möglichkeit diese Redundanz auszunutzen wäre z.B. die gleiche Information in aufeinanderfolgende Frames einzubringen. Die Vorteile, die sich daraus ergeben, werden in Kapitel 7.1 dargestellt.
Die Fehlerrate für B-Frames bei einer Datenrate des Encoders von 1 MBit/s liegt zwischen 5% und 20% ist damit zu groß. Bei einer Datenrate von 2 MBit/s fällt die Fehlerrate stark ab, liegt bei den letzten beiden Videos zwischen 0% und 5% und kann bei Ausnutzung des Redundanzspielraums unter besonderer Berücksichtigung des Verfahrens in Kapitel 7.1 deutlich gesenkt werden. Die Encoding-Datenrate von 1 MBit/s muß für beide Verfahren als kritisch angesehen werden. Senkt man sie weiter ab, müssen Abstriche an der Wasserzeichendatenrate in Kauf genommen werden.
Auffällig ist die unterschiedliche Empfindlichkeit beider Verfahren gegenüber

der Encoder-Datenrate. Besonders gut sieht man dies für Ansatz I in Versuch 1 im dritten Video. Dort fällt die Fehlerrate bei B-Frames von über 20% auf 0.0% ab. Ein derartiger Sprung ist in Ansatz II nicht zu beobachten. Dies läßt den Schluß zu, daß die Helligkeitsmuster aus Ansatz II niederfrequenter und damit robuster gegenüber Kompression sind. Das dies auf Kosten der Bildqualität geht, kann man an allen drei Videos beobachten. Die Bildqualität aus Ansatz I ist durchweg besser, als die aus Ansatz II. Dies macht sich vor allem am etwas stärkeren Flimmern des Bildes bemerkbar.
Trotz diverser Nachteile des Verfahrens aus Ansatz II (schlechtere Bildqualität, höherer Berechnungsaufwand für Helligkeitsmuster, niedrigere Wasserzeichendatenrate) zeigen die Experimente doch einen deutlichen Vorteil auf. Das Verfahren aus Ansatz II zeigt sogar ohne spezielle Gegenmaßnahmen, sozusagen eine natürliche Robustheit gegenüber des gefürchteten (diverse kommerzielle Wasserzeichen werden durch die StirMark-Attacke gelöscht) StirMark-Angriffs. Obwohl die Fehlerraten natürlich immer noch inakzeptabel hoch sind (teilweise bis 17% in I- und P-Frames), liegen sie deutlich unter den Fehlerraten des Verfahrens aus Ansatz I. Die Fehlerraten im dritten Video liegen sogar in derselben Größenordnung wie nach einfacher MPEG-Encodierung. Weiterhin muß berücksichtigt werden, daß der StirMark-Angriff sehr kostenintensiv ist. Der StirMark-Angriff auf eine Stunde Video würde auf einer SGI-O2 ca. 500 Stunden dauern. Trotzdem muß ein Wasserzeichen natürlich auch solche Attacken überleben können, weshalb weitere Forschungen in diesem Bereich notwendig sind.
Die eingebrachten Störungen durch die Quicktimekonvertierung sind nicht besonders groß und sollten durch zusätzliche Redundanzcodes abfangbar sein. Für Mehrfachmarkierungen scheinen sich beide Verfahren nicht besonders gut zu eignen. Die Bildqualität wird zu schlecht. Dies muß jedoch nicht unbedingt einen Nachteil bedeuten. Durch das Abnehmen der Bildqualität bei Aufbringung mehrerer Wasserzeichen, kann die Mehrfachmarkierung nicht als Attacke benutzt werden. Ist Mehrfachmarkierung für eine Anwendung zwingend notwendig, kann jedoch durch eine Aufteilung der angestrebten Wasserzeichendatenrate auf die Anzahl der Wasserzeichenebenen die Bildqualität deutlich erhöht werden.

6.1 Schlußfolgerung

Beide Verfahren haben sich als prinzipiell anwendbar für die Markierung von MPEG-Videosequenzen erwiesen. Dies läßt sich deutlich an den geringen Fehlerraten nach einfacher MPEG-Encodierung vor allem in I- und P-Frames ablesen. Mittels weiterer fehlerkorrigierender Codes (siehe Kapitel 7.1), welche die Wasserzeicheninformation über verschiedene Frames und Frametypen verteilen, sollte eine Fehlerrate erreicht werden können, die für kommerzielle Applikationen ausreicht. Für das Frequenzraumverfahren aus Ansatz I kann dabei eine Wasserzeichendatenrate von 100 Bits/s und für das Bildraumverfahren aus Ansatz II eine Wasserzeichendatenrate von 30 bis 50 Bits/s angestrebt werden.
Die Funktionalität der *smooth block/edge detection* wurde in Kapitel 3.1 anhand von Beispielbildern belegt und erfolgreich auf beide Verfahren appliziert. Durch die Parametrisierung der *smooth block/edge detection* kann das Wasserzeichen individuell an die Videocharakteristika angepaßt werden.
Die Bildqualität der Verfahren ist jedoch noch nicht ausreichend und **muß** weiter verbessert werden! Sie ist in den untersuchten Videos, besonders in Video 2 und Video 3 zwar recht gut, aber für das geübte Auge, insbesondere wenn der Betrachter weiß worauf er achten muß, sind Artefakte schnell erkennbar. In Kapitel 7 wird hierfür ein Verfahren vorgestellt.
Es hat sich gezeigt, daß die Auswahl der Verfahren sinnvoll getätigt wurde. Durch die Auswahl eines Bildraum- und eines Frequenzraumverfahrens konnte gezeigt werden, daß beide Verfahren Vor- und Nachteile besitzen, die sich gegenseitig kompensieren können. Die unterschiedliche Robustheit gegenüber der StirMark-Attacke legt den Vorschlag nahe Informationen auch mit beiden Verfahren einzubringen. Hier sind weitere Untersuchungen notwendig.
Für Mehrfachmarkierungen sind beide Verfahren nur beschränkt geeignet. Ist sie notwendig, kann sie durch eine deutliche Reduzierung der Datenrate der einzelnen Markierungen erreicht werden.

Kapitel 7

Verbesserungen und Ausblick

Aufgrund der Testergebnisse vor allem im Bereich Bildqualität ergibt sich eine Hauptforderung an die untersuchten Wasserzeichenverfahren. Die Bildqualität muß weiter verbessert werden. Mit dem im Ergebnisteil diskutierten Redundanzspielraum um den Faktor 5 bis 20 läßt sich ein Markierungsverfahren anwenden, welches die Bildqualität drastisch erhöht und die Fehlerrate in P- und B-Frames weiter absenkt. Dieses Verfahren wird in Kapitel 7.1 erläutert.
Zusätzlich zu den bisher untersuchten Attacken Kompression, Formatkonvertierung und StirMark-Attacke, ist es zwingend erforderlich gegenüber Skalierung robust zu sein. Die unbedingte Notwendigkeit dieser Forderung ergibt sich aus der einfachen Ausführung dieser definitionsgemäß freundlichen Attacke, da Skalierung bereits im MPEG-Standard vorgesehen ist. Die prinzipielle Vorgehensweise, um gegen Skalierung aber auch gegen Ausschnittbildung robust zu werden wird in Kapitel 7.2 beschrieben.
Mit der Entwicklung des MPEG4-Standards, in dem einzelne Objekte in Videosequenzen definiert und strukturiert gespeichert werden, stellt sich die Frage, ob die Möglichkeit einer Wasserzeichenmarkierung solcher Objekte besteht. Schließlich kann auch der kommerzielle Wert einzelner Objekte in Videos, z.B. computeranimierte Dinosaurier in dem Film "Jurassic Park", sehr groß sein. Ein Verfahren zur Markierung einzelner Objekte in Videosequenzen wird in Kapitel 7.3 vorgestellt.
Die in dieser Studie untersuchten Ansätze arbeiten noch sehr langsam. Die Markierung während des Decodiervorgangs eines einzelnen Frames dauert mehrere Sekunden. Ausserdem stört der De- und Encoding Prozeß bereits

die eingebrachte Wasserzeicheninformation. In Kapitel 7.4 wird versucht, ein Verfahren zu beschreiben, bei dem die Wasserzeicheninformation direkt in den MPEG-Bitstrom eingebracht wird, so daß der De- und Encoding Prozeß umgangen werden kann.
Zum Abschluß muß auch die jetzige Struktur des Programms überdacht werden. Die fehlerkorrigierenden Codes sind noch zu stark mit den verwendeten Wasserzeichenalgorithmen verwoben. Verbesserungsvorschläge hierfür werden in Kapitel 7.5 gegeben.

7.1 Verbesserung der Bildqualität

Es werden jeweils N aufeinanderfolgende Frames mit dem gleichen Wasserzeichen (Helligkeitsmuster) versehen. Dabei gibt N den zusätzlichen Redundanzfaktor an. Die Vorteile sind:

Drastische Verbesserung der Bildqualität Dadurch das in Bildteilen, die sich im zeitlichen Ablauf kaum verändern, in jedem Frame ein unterschiedliches Helligkeitsmuster aufaddiert wird, entsteht das in allen Videos beobachtete Flimmern des Bildes. Dieses Flimmern verschwindet völlig, wenn in aufeinanderfolgenden Frames immer dasselbe Helligkeitsmuster aufaddiert wird. Der "Sprung" nach N Frames ($N > 5$) zu einem neuen Muster ist kaum zu beobachten, wie erste Tests beweisen. Vorstellbar ist auch, jeweils ein ganze Szene mit den gleichen Informationsbits zu markieren, so daß der "Sprung" zu einem neuen Muster im Szenenwechsel untergeht. Auch die bisher zu beobachtenden störenden Artefakte von fehlerhaft eingestuften Blöcken in der *smooth block/edge detection*-Phase sind deutlich schwächer geworden und nur im direkten Vergleich mit dem Orginalvideo vom geübten Auge zu sehen.

Verbesserung der Fehlerrate Hiermit ist nicht nur die verbesserte Fehlerrate aufgrund der erhöhten Redundanz gemeint. Vielmehr verbessert sich auch die Fehlerrate in den einzelnen Frames. Dies hat seinen Ursprung in der MPEG-Codierung. P- und B-Frames werden aus den Referenz-Frames berechnet. Ein P-Frame besteht nur aus der Differenzinformation zum vorher gespeicherten Referenz-Frame (I-Frame). Diese

Differenzinformation ist in statischen Bildteilen sehr klein. Enthält ein solcher Bildausschnitt ein anderes Wasserzeichen als das Referenzframe, muß die durch das Wasserzeichen eingebrachte Differenz gespeichert werden. Diese Differenz ist aber aufgrund der Natur unsichtbarer Wasserzeichen sehr klein und kann im Quantisierungsprozeß der MPEG-Codierung leicht verlorengehen. Für B-Frames gilt dies noch viel stärker, da für sie sogar nur die Differenz zu zwei interpolierten Frames gespeichert wird. Wird jedoch in alle Frametypen dieselbe Wasserzeicheninformation eingebracht, muß in P- und B-Frames kaum zusätzliche Information für die Wasserzeichen gespeichert werden, da sie in den Referenzframes bereits vorhanden sind. Die Fehlerrate bei I-Frames sollte sich nach diesen Überlegungen nicht ändern.

"Intelligenter" Redundanzcode Beim Decodieren des Redundanzcodes kann berücksichtigt werden, aus welchem Frametyp die Wasserzeicheninformation stammt und entsprechend gewichtet werden. Die Information aus den I-Frames würden demnach am stärksten und die Information aus den B-Frames am schwächsten für die decodierte Information berücksichtigt werden.

7.2 Robustheit gegenüber Skalierung und Ausschnittbildung

Das Problem beim Auslesen eines markierten und skalierten Videos ist, daß die für das Auslesen benutzten Helligkeitsmuster in Ansatz II in der Größe nicht mehr mit den eingebrachten Helligkeitsmustern übereinstimmen. Dadurch wird die Korrelation zwischen der Videoinformation und dem Helligkeitsmuster verfälscht. Um dieser Attacke entgegenzuwirken kann beim Einbringen auf einen Teil der Wasserzeicheninformation verzichtet und statt dessen ein festes Bitmuster eingebracht werden. Das sich aus dem festen Bitmuster ergebende Helligkeitsmuster wird dann in einem Brute-Force Ansatz auf alle Größen skaliert. Mit den skalierten Helligkeitsmustern wird die Korrelation zum skalierten Video getestet. Scheint eine Skalierung aufgrund der Korrelation wahrscheinlich können weitere Frames mit der gleichen Skalierung getestet werden, sofern in allen Frames ein festes Bitmuster eingebracht wird. Aufgrund der zumeist hohen Anzahl von

Frames kann eine hohe Detektionswahrscheinlichkeit der richtigen Skalierung erwartet werden. Ein Gradientensuchverfahren zur Detektion der richtigen Skalierung ist ebenso vorstellbar. Dies setzt allerdings voraus, daß kleine Abweichungen von der richtigen Skalierung auch nur zu kleinen Abweichungen in der Korrelation führen. Dies läßt sich jedoch nicht vorraussetzen und müßte in entsprechenden Experimenten überprüft werden. Für das Frequenzraumverfahren aus Ansatz I läßt sich dieser Algorithmus nicht anwenden, da das aufgebrachte Helligkeitsmuster von der Beschaffenheit der Frequenzen in den einzelnen Blöcken abhängt. Hier wäre eine Kombination beider Verfahren denkbar. Ein zusätzlich mit Ansatz II aufgebrachtes Helligkeitsmuster könnte, auf oben beschriebenen Weg, zur Detektion der richtigen Skalierung dienen.

Das Problem der Ausschnittbildung ist zum einen die Synchronisation des eingebrachten Helligkeitsmusters mit dem Auslesemuster und zum anderen der offensichtliche Verlust an Bild- und Wasserzeicheninformation. Letzterem kann nur mit erhöhter Redundanz entgegengewirkt werden, da im Gegensatz zu den meisten anderen Attacken die Wasserzeicheninformation nicht verfälscht wird, sondern definitiv verloren geht. Da die Attacke "Ausschnittbildung" sich bei Videos jedoch hauptsächlich auf die Bildmitte konzentriert, kann man den Aufwand an erhöhter Redundanz dadurch minimieren, daß die Wasserzeicheninformation ebenfalls hauptsächlich in der Bildmitte konzentriert wird.

Der Verlust der Synchronisation wiegt schwerer. Durch Ausschnittbildung werden die Startpunkte der einzelnen markierten Blöcke mitunter verschoben. Diese sind jedoch für den Lesevorgang unabdingbar. Hier kann das bereits bei der Skalierung vorgeschlagene Verfahren Abhilfe schaffen. Ein festes Bitmuster und daraus resultierendes Helligkeitsmuster, welches natürlich auch in dem ausgeschnittenen Video zum Teil vorhanden sein muß, wird solange über den Ausschnitt geschoben bis die Korrelation maximal oder zumindest groß ist. Die so gefundene Synchronisation kann an weiteren Frames überprüft werden. Auch hier gilt, daß für Ansatz I beide Verfahren kombiniert werden können. Durch Einbringung eines Helligkeitsmusters über Ansatz II kann die richtige Synchronisation für Ansatz I gefunden werden.

7.3 Objekt-Watermarking

Beide Watermarking-Verfahren wurden daraufhin untersucht, ob mit ihnen Objekte so markiert werden können, daß sie ein Ausleseverfahren mit dem richtigen Schlüssel wiederfinden kann, ohne die Position der Objekte zu kennen. Die Größe der Objekte soll zunächst auf genau 64x64 Pixel beschränkt sein.

Die Markierung mit Hilfe des Bildraumverfahrens ist einfach. Es wird ein entsprechend großes 64x64 Pixel Muster über den User Key generiert und auf das Objekt aufaddiert. Zum Auffinden des Objekts muß die Korrelation von jedem 64x64 Block des Bildes getestet werden. Liegt sie über einem bestimmten Schwellwert kann das Objekt als markiert angenommen werden. Versuchsreihen haben jedoch ergeben, daß die Wasserzeichenstärke sehr groß sein muß, damit sich die Korrelation eines markierten Objekts von allen nichtmarkierten Objekten unterscheidet. Dadurch kommt es zu sichtbaren Artefakten in den Objekten. Um die visuellen Störungen abzuschwächen, wurde die Wasserzeichenstärke an die Helligkeit der Pixel angepaßt. Dies trägt der Tatsache Rechnung, daß Bildänderungen in hellen Bereichen weniger auffällig sind, als in dunklen Bildbereichen. Die dunkelsten Pixel wurden somit nur um 1 und die hellsten Pixel um 6 abgeändert. Trotzdem blieb das Wasserzeichen im markierten Objekt deutlich sichtbar. Dieser Ansatz wurde daraufhin verworfen und ein Ansatz über das Frequenzraumverfahren getestet..

Um ein Objekt mittels des Frequenzraumverfahrens zu markieren, werden abwechselnd Nullen und Einsen in aufeinanderfolgende 8x8 Blöcke eingebracht. Zum Auslesen wird für jeden 64x64 Block die Anzahl der übereinstimmenden Bits getestet. Auch hier muß ein passender Schwellwert gefunden werden. Dieses Verfahren liefert bessere Ergebnisse als das Bildraumverfahren. Obwohl bei schwacher Wasserzeichenstärke und damit geringen visuellen Störungen mehrere Dutzend 64x64 Blöcke über dem Schwellwert von 30 übereinstimmenden Bits lagen, gibt es bei den markierten Blöcken eine Besonderheit. In der Entfernung von einem Pixel um die markierten Objekte, wird ebenfalls eine erhöhte Übereinstimmung der Bits festgestellt. Dies ist in der Umgebung der nicht markierten Objekte nicht der Fall. Nutzt man diese Tatsache aus und wählt man die Wasserzeichenstärke hoch genug läßt sich das Objekt eindeutig wiederfinden. Trotzdem sind bei genauer Betrachtung die Markierungen noch sichtbar.

7.4 Einbettung in den MPEG-Bitstrom

Das Frequenzraumverfahren aus Ansatz I sollte sich problemlos in den MPEG-Bitstrom einfügen lassen. Hierbei würde sogar die Rückumwandlung der Helligkeitsinformationen in den Frequenzraum wegfallen, die bei der Markierung der Bildraumdaten anfällt. Für Ansatz II müßten die Helligkeitsmuster vorweg in den Frequenzraum transformiert werden. Eine Geschwindigkeitsoptimierung gäbe es hierbei nicht. Lediglich der qualitätsmindernde Schritt des De- und Encodings würde wegfallen. Um die *smooth block/edge detection*-Phase einbringen zu können, muß während der Markierung des MPEG-Bitstroms jeweils ein ganzer 8x8 Block Frequenzdaten gepuffert werden aus dem die blockspezifische Wasserzeichenstärke evaluiert wird. Nach Festlegung der Wasserzeichenstärke kann das Wasserzeichen eingebracht und der entsprechende Block zurückgeschrieben werden.
Ein gravierendes Problem entsteht durch die *motion vectors* der MPEG-Codierung. In P- und B-Frames werden in Bildblöcken Differenzen zu Bildblöcken der Referenzframes gespeichert, die nicht nur zeitlich sondern auch räumlich versetzt sind. Der räumliche Versatz wird in den *motion vectors* gespeichert. Dadurch wird die in den Referenzframes eingebrachte Wasserzeicheninformation zeitlich und räumlich weiterkopiert. Um diesen unerwünschten Nebeneffekt zu verhindern, müssen die eingebrachten Änderungen der Referenzframes gespeichert werden. Bei Einbringung eines Wasserzeichens in P- und B-Frames müssen die *motion vectors* überprüft und ein entsprechender Term, der das in den Referenzframes eingebrachte Wasserzeichen kompensiert, aufaddiert werden (*drift compensation*).

7.5 Struktur des Programms

Die Struktur des Programms hat aufgrund des experimentellen Charakters etwas gelitten. Besonders nachteilig und als sehr unflexibel hat sich die sehr starke Verbindung zwischen fehlerkorrigierendem Code und Art des Algorithmus' herausgestellt. Der Algorithmus für Ansatz I hat vier verschiedene Blockzustände beim Auslesen: "0" und "1" für die Wasserzeicheninformation, "2" für eine zerstörte Wasserzeicheninformation und "i" für einen ungültigen Block und damit nicht vorhandener Wasserzeicheninformation. Der Algorithmus für Ansatz I hat jedoch nur die beiden Zustände "0" und

"1" für die ausgelesene Wasserzeicheninformation. Daher unterscheidet sich die Funktion *analyse_bits*, für beide Algorithmen, die für die Umwandlung der ausgelesenen Blockzustände in decodierte Wasserzeicheninformation verantwortlich sind. Diese Abhängigkeit muß aufgelöst werden, vor allem wenn weitere fehlerkorrigierende Codes zwischen den Frames benutzt werden. Hier wäre eine Abhängigkeit vom verwendeten Algorithmus unerklärlich und demzufolge überflüssig.

Kapitel 8

Zusammenfassung

In dieser Studie wurde zunächst ein großer Überblick über das Thema Watermarking für visuelle Medien gegeben. Diverse bestehende Watermarking-Verfahren wurden dargestellt und bewertet. Weiterhin wurden verschiedene Typen von Angriffen auf Wasserzeichen untersucht und, wenn möglich, ein Ansatz für mögliche Gegenmaßnahmen aufgezeigt.
Aus den aufgeführten Watermarking-Verfahren, die zunächst nur für die Markierung von Einzelbildern gedacht sind, wurden zwei herausgewählt und um ein Verfahren zur Verbesserung der Bildqualität erweitert. Beide verbesserten Verfahren wurden implementiert und insbesondere daraufhin untersucht, ob die Wasserzeichen auch nach einer MPEG-Codierung mit verschiedenen Kompressionsraten (bzw. Datenraten) noch auslesbar sind. Die Robustheit der Watermarking-Verfahren gegenüber Formatkonvertierung, Mehrfachmarkierung und gegenüber dem Antiwatermark-Tool "StirMark" wurde getestet. Aus den Testergebnissen wurde die prinzipielle Eignung beider verbesserten Watermarking-Verfahren für die Markierung von MPEG-Videos abgeleitet.

Anhang A

Anhang

In den folgenden VRML-Szenen wird die Differenz zwischen je zwei Video-Frames als 3D-Hügellandschaft dargestellt. Dadurch werden die Bildstörungen durch die Wasserzeichen, durch die MPEG-Codierung und besonders durch den "StirMark"-Angriff anschaulich gemacht.

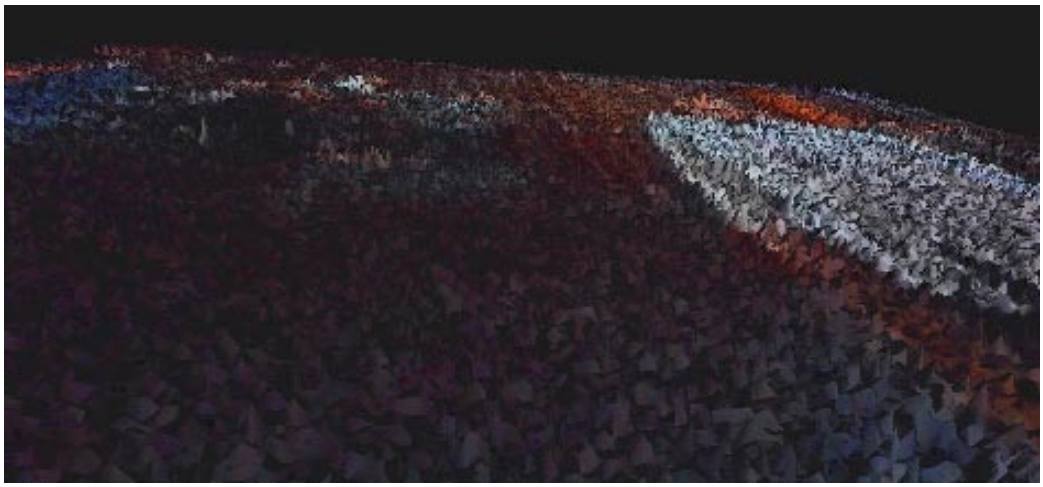


Abbildung A.1: Differenzbild zwischen Orginal-Frame und nach MPEG-Codierung (Video 2)

Abbildung A.2: Differenzbild zwischen Orginal-Frame und nach “StirMark”-Angriff (Video 2)

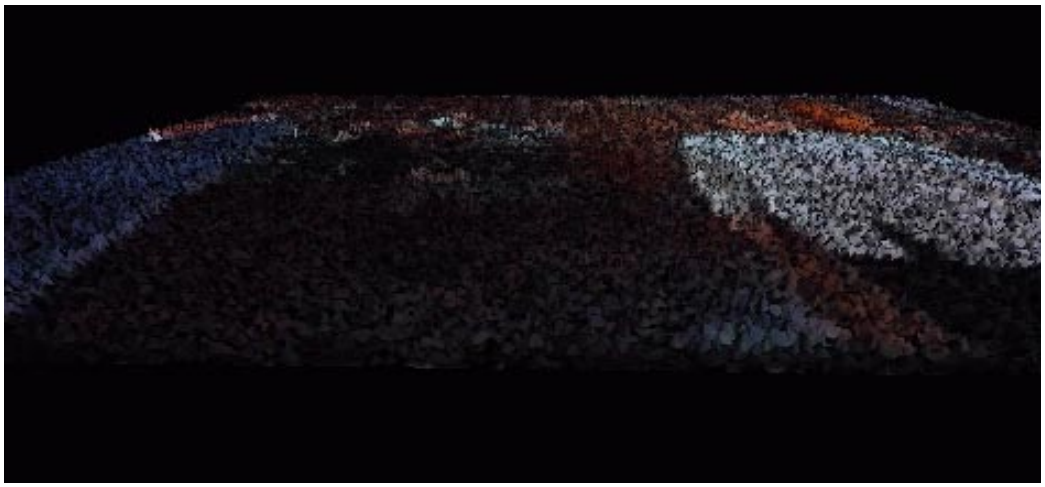


Abbildung A.3: Differenzbild zwischen Orginal-Frame und nach Markierung mit dem Bildraumverfahren (Video 2)

Abbildung A.4: Differenzbild zwischen Orginal-Frame und nach Markierung mit dem Frequenzraumverfahren (Video 2)

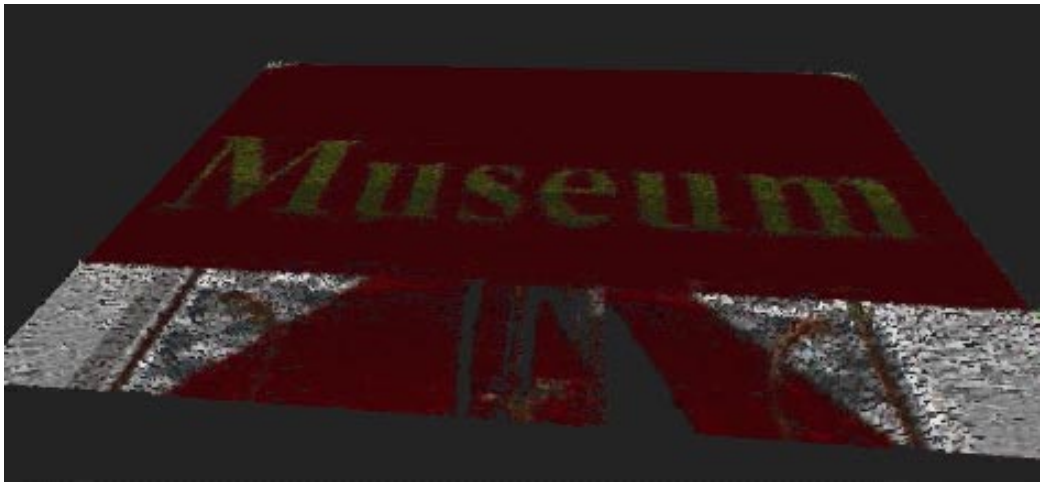


Abbildung A.5: Differenzbild zwischen Orginal-Frame und nach MPEG-Codierung (Video 1)

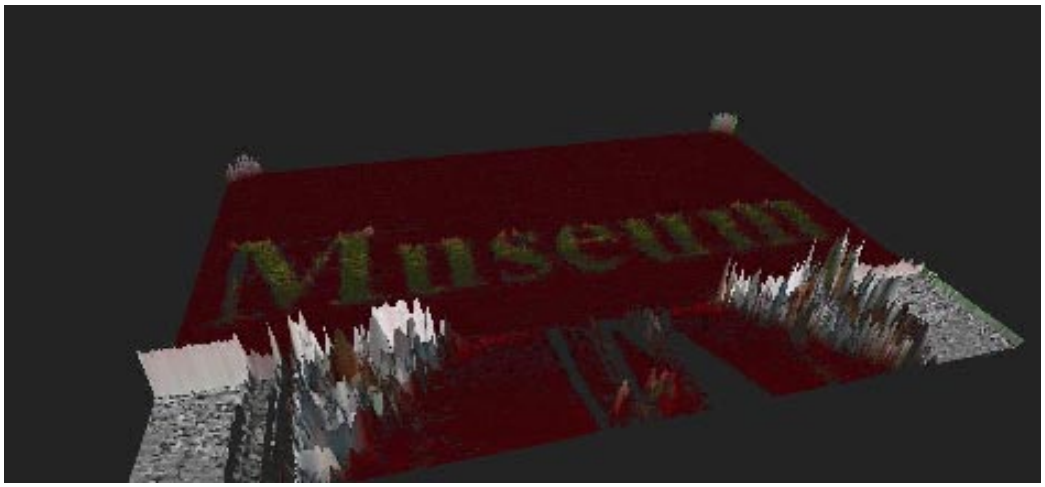


Abbildung A.6: Differenzbild zwischen Orginal-Frame und nach “StirMark”-Angriff (Video 1)

Abbildung A.7: Differenzbild zwischen Orginal-Frame und nach Markierung mit dem Bildraumverfahren (Video 1)

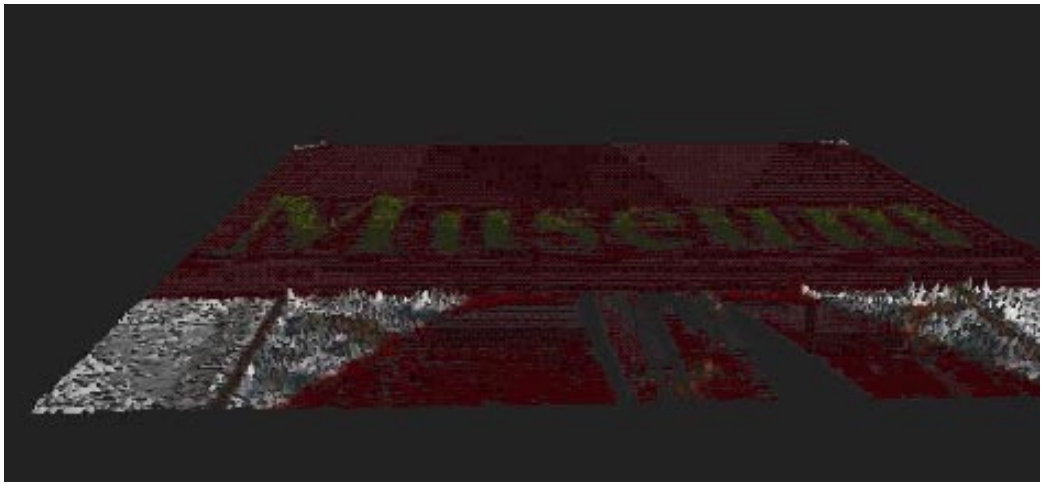


Abbildung A.8: Differenzbild zwischen Orginal-Frame und nach Markierung mit dem Frequenzraumverfahren (Video 1)

Literaturverzeichnis

[BGM1996] **Walter Bender, Daniel Gruhl, Norishige Morimoto, A. Lu** Techniques for data hiding, *IBM Systems Journal, Vol. 35, No. 3-4, pp. 313-316, 1996.*

[BMY1997] **Dave Benham, Nasir Memom, Boon-Lock Yeo, Minerva Yeung** Fast Watermarking of DCT-based Compressed Images, *Proc. International Conference on Imaging Science, Systems and Technology (CISST '97), pp. 243-252, Las Vegas, Nevada, USA, 30 June-3 July, 1997.*

[BRD1995] **F.M. Boland, J.J.K. O Ruanaidh, C. Dautzenberg** Watermarking Digital Images for Copyright Protection, *Proc. 5th International Conference on Image Processing and its Applications, No. 410, Edinburgh, July, 1995.*

[BoS1994] **Dan Boneh, James Shaw** Collusion-Secure Fingerprinting for Digital Data, *Advances in Cryptology (Crypto '95), Vol. 963 of Lecture Notes in Computer Science, pp. 452-465, Springer-Verlag, 27-31 August, 1995,*
ftp://ftp.cs.princeton.edu/reports/1994/468.ps.Z

[Car1995] **Germano Caronni** Assuring Ownership Rights for Digital Images, *H. H. Brüggeman and W. Gerhardt-Haeckl, editors, Reliable IT Systems VIS '95, pp. 251-263, Vieweg Publishing Company, Germany, 1995.*

[CKL1996] **I. Cox, J. Kilian, F. T. Leighton, T. Shamoon** A Secure, Robust Watermark for Multimedia, *First International Workshop on Information Hiding, R. Anderson, ed., vol. 1174 of Lecture Notes in Computer Science, pp. 185-206, Springer-Verlag, 1996.*

[CMY1997] **Scott Craver, Nasir Memon, Boon-Lock Yeo, Minerva Yeung** Can invisible watermarks resolve rightful ownerships?, *Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases V, Vol. 3022, pp. 310-321, San Jose, CA, USA, 13-14 Feb., 1997.*

[DBQ1996] **J.-F. Delaigle, J.-M.Boucqueau, J.-J. Quiquater, B. Macq** Digital Images protection techniques in a broadcast framework: An overview, *European Conference on Multimedia Applications, Services and Techniques (ECMAST '96), pp. 711-727, Louvain-la-Neuve, Belgium, 18-20 May, 1996.*

[Dig1998] **Digimarc Corporation**
www.digimarc.com

[DSN1998] **J. Dittmann, A. Steinmetz, F. Nack, R. Steinmetz** Interactive Watermarking Environments, *to appear in IEEE Multimedia '98, Austin, Texas, 1998.*

[FBS1998] **J. Fridrich, A. C. Baldoza, R. J. Simard** Robust digital watermarking based on key-dependent basis functions, *submitted to 2nd Information Hiding Workshop, Portland, OR, USA, 15-17 April, 1998.*

[Fri1998] **Jiri Fridrich** Methods for data hiding,
http://ssie.binghamton.edu/∼jirif/

[HaG1996] **Frank Hartung, Bernd Girod** Digital Watermarking of Raw and Compressed Video, *Proc. European EOS/SPIE Syposium on Advanced Imaging and Network Technologies, pp. 205-213, Berlin, Germany, October, 1996.*

[HaG1997a] **Frank Hartung, Bernd Girod** Einbettung digitaler Wasserzeichen in MEPG-2 codierte Videosequenzen, *Proc. 7. Dortmunder Fernsehseminar, Dortmund, Germany, October 1997.*

[HaG1997b] **Frank Hartung, Bernd Girod** Digital Watermarking of MPEG-2 Coded Video in the Bitstream Domain, *Proc. ICASSP 1997, Vol. 4, pp. 2621-2624, Munich, Germany, April, 1997.*

[KJB1997] **Martin Kutter, Frederic Jordan, Frank Bossen** Digital Signature of Color Images using Amplitude Modulation, *Proc. SPIE '97 Storage and retrieval for image an video databases, Vol. 3022, pp. 518-526, San Jose, USA, February, 1997.*

[Kuh1997] **Markus G. Kuhn** StirMark, *November, 1997.*
http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark/

[LLB1997] **Gerrit C. Langelaar, Reginald L. Lagendijk, Jan Biemond** Real-time Labeling Methods for MPEG Compressed Video, *Proc. 18th Syposium on Information theory in the Benelux, Veldhoven, The Netherlands, 15-16 May, 1997.*

[MPE1993] **MPEG International Standard ISO/IEC 11172:** Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, *Part 1: Systems, Part 2: Video, Part 3: Audio, 1993.*

[MPE1994] **MPEG Software Simulation Group (MPEG-L@netcom.com)** MPEG-2 Encoder/Decoder, Version 1.1, *June 1994,*
http://www.bok.net/~tristan/MPEG/MSSG/

[NiP1996] **N. Nikolaidis, I. Pitas** Copyright protection of images using robust digital signatures, *Proc. International Conference on Acoustic, Speech and Signal Processing (ICASSP '96), Vol. 4, pp. 2168-2171, Atlanta, Georgia, USA, May, 1996.*

[PAK1998] **Fabian Petitcolas, Ross J. Anderson, Markus G. Kuhn** Attacks on Copyright Marking Systems, *submitted to Second International Workshop on Information Hiding, 15-17 April 1998,*
http://www.cl.cam.ac.uk/~fapp2/

[RiE1998] **Jürgen Rink, A.E.** Digitale Wasserzeichen löschen, *c't 4/98, S. 47.*

[Rin1997a] **Jürgen Rink** Hinters Licht geführt, Bits in Bild- und Audiodateien verstecken, *c't 6/97, S. 330.*

[Rin1997b] **Jürgen Rink** Bildergeschichten, Digitale Wasserzeichen unterstützen den Urheberrechtsschutz im Internet, *c't 8/97, S. 162.*

[ScU1998] **J. Schwenk, J. Ueberberg** Patentantrag: Verfahren zum sicheren Einbringen Digitaler Fingerabdrücke in elektronische Dokumente, *1998.*

[SGV1995] **C. Simon, E. Goray, G. Vercken, B. Delivat, J.-F. Delaigle, J.-M. Boucqueau** Digital Images protection management in a broadcast framework: "Overview/TALISMAN solution", *ftp://ftp.tele.ucl.ac.be/pub/TALISMAN/*

[Sig1998] **Signum Technologies** *www.signumtech.com*

[SmR1993] **Brian C. Smith, Lawrence A. Rowe** A New Family of Algorithms for Manipulating Compressed Images, *Computer Graphics and Applications Vol. 13, No 5, pp. 34-42, September, 1993, http://bmrc.berkeley.edu/papers/1993/126/126.html*

[Ste1995] **R. Steinmetz** Multimedia Technologie, *Springer Verlag, 1995.*

[TsH1993] **H. Tzschach, G. Haßlinger** Codes für den störungssicheren Datentransfer, *Oldenbourg Verlag, 1993.*

[YMB1997] **Minerva M. Yeung, Frederick C. Mintzer, Gordon W. Braudaway, A. R. Rao** Digital Watermarking for High-quality Imaging, *Proc. IEEE First Workshop on Multimedia Signal Processing, Princeton, NJ, 23-25 June, 1997.*

[Zha1997] **Jian Zhao** Applying Digital Watermarking Techniques to Online Multimedia Commerce, *Proc. International Conference on Imaging Science, Systems and Technology (CISST '97), pp. 288-294, Las Vegas, Nevada, USA, 30 June-3 July, 1997.*

[ZhK1995] **Jian Zhao, Eckhard Koch** Embedding Robust Labels into Images for Copyright Protection, *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, Austria, 21-25 August, 1995.*