



MPEG-4 objektbasierte Videocodierung

MPEG-4 Object-Based Video Coding

Thomas Sikora, Technische Universität Berlin

Zusammenfassung Neben einer hohen Kompressionseffizienz und einer guten Fehlerrobustheit stellt MPEG-4 Video als einziger internationaler Standard objektbasierte Funktionalitäten für Anwendungen zur Verfügung. In diesem Beitrag werden die Ziele und Techniken der MPEG-4 objektbasierten Kodierung

beschrieben. ▶▶▶ **Summary** MPEG-4 is the first international standard that supports applications with object-based functionalities. This article outlines the goals and technical details of the MPEG-4 object-based coding algorithm.

KEYWORDS I.4 [Image Processing] MPEG4, Video-Compression, Object-based Coding, Multimedia

1 Einführung

Nach der Entwicklung der erfolgreichen MPEG-1 und MPEG-2 Standards hat die MPEG Gruppe 1999, die zunehmende Verschmelzung der traditionell getrennten Industriezweige Telekommunikation, Computer und TV/Film berücksichtigend, offiziell eine erste Version des MPEG-4 Standards verabschiedet [1]. Die MPEG Gruppe hatte sich zum Ziel gesetzt neue Algorithmen für die Codierung audiovisueller Informationen zu entwickeln, die insbesondere eine hohe Interaktivität mit Ton- und Bildinhalten gewährleisten und eine hohe Datenkompression erzielen. Weiterhin, und dies ist eine wesentlich neue Funktionalität, ermöglicht MPEG-4 einen universellen Zugriff nicht nur auf einzelne Bilder (z.B. für schnellen Suchlauf, Editieren und Manipulieren wie bei bisherigen Standards), sondern darüber hinaus für Multimediaanwendungen auch den Zugriff auf einzelne Bildobjekte auf Bitstromebene. Das we-

sentliche Ziel ist die Bereitstellung von komprimierten Daten, die eine hohe bildinhaltsbezogene Funktionalität und Flexibilität für den Endbenutzer ermöglichen – z.B. um flexibel am Empfänger einzelne Bildinhalte in Videosequenzen separat zu decodieren, zu manipulieren und ggf. in andere Anwendungsumgebungen (in Textverarbeitungsprogramme, andere Videosequenzen, etc.) zu kopieren. Dies alles geschieht auf Bitstromebene, ohne dass eine zusätzliche Nachverarbeitung und Transcodierung notwendig wird. Bieten also die MPEG-1 und MPEG-2 Standards bisher eine so genannte *bildbezogene* Funktionalität, so könnte man für die MPEG-4 Video-Codierung von *bildinhaltsbezogenen* Funktionalitäten sprechen [4].

Diese inhaltsbezogenen Funktionalitäten werden mit MPEG-4 nicht nur für herkömmliches, pixelbasiertes Video, sondern auch für synthetische, computergenerierte 2D- und 3D-Bildsequenzen

ermöglicht, um so insbesondere die hybride Codierung von natürlichen und synthetischen Bildsequenzen zu unterstützen. Es ist anzunehmen, dass insbesondere die Entwicklung neuerer Computerspiele oder Telepräsenzsysteme hierdurch maßgeblich beeinflusst wird. Aber auch dem zunehmenden Einfluss der Computergrafik im Bereich der Studiotechnik (z.B. beim virtuellen Studio) wird so Rechnung getragen.

Weitere, nicht weniger wichtige Funktionalitäten des MPEG-4 Video Standards definieren sich aus den Randbedingungen aus Übertragungsaspekten. Hier ist insbesondere die mobile Bildkommunikation berücksichtigt. Eine hohe Toleranz bzw. Anpassung der Audio- und Videoqualität bei störungsbehafteter Übertragung und variabler Bandbreite ist eine zentrale Stärke des MPEG-4 Standards – aber auch die Möglichkeit einen MPEG-4 Bitstrom flexibel hinsichtlich variabler Decoderressourcen, zum Beispiel in Bezug auf Energieverbrauch

oder Prozessorleistung, decodieren zu können.

MPEG-4 ist der erste internationale Video-Kompressionsstandard, der objektorientierte Funktionalitäten und Anwendungen durch spezielle Codiertechniken unterstützt. Seit 1999 wird intensiv an weiteren MPEG-4 Codiertechnologien gearbeitet, insbesondere hinsichtlich einer skalierbaren Übertragung und einer weiter erhöhten Kompressionseffizienz (MPEG-4 *Advanced Video Codec* – AVC).

Wegen des innovativen Charakters der objektbasierten Codierung wird in diesem Beitrag im Wesentlichen die objektbasierte Kompressionsphilosophie von MPEG-4 und deren technische Implementierung dargestellt. Für eine detailliertere Beschreibung des MPEG-4 Visual Standards kann das MPEG-4 Buch von Pereira & Ebrahimi [8] empfohlen werden.

2 Der MPEG-4 Video Standard

Der MPEG-4 Video Standard ist eine konsequente Erweiterung bereits bekannter DCT-basierter Verfahren mit Bewegungskompensation wie MPEG-1 und MPEG-2. Ein Überblick über diese Verfahren findet sich in [2;3]. Die folgenden Details beschreiben die grundlegenden Basiselemente und Unterschiede zu MPEG-1/2:

- Standard Y:U:V Luminanz und Chrominanz Beschreibung von regelmäßig abgetasteten Pixeln im 4:2:0 Format. Die Intensitätswerte jedes Pixels sind mit 8 Bit quantisiert. Die Größe und die Form der Eingangsbilder kann, je nach Anwendung, variabel sein.
- Codierung von Bildsequenzen mit Hilfe mehrerer willkürlich geformter *Video Object Planes* (VOP's) für die Unterstützung inhaltsbezogener Funktionalitäten. Grundsätzlich sind die für die MPEG-4 Video Codierung zu verarbeitenden Bilder nicht mehr notwendigerweise rechteckig wie bei MPEG-1 [5] und

MPEG-2 [6] – sondern können beliebig geformt sein, wobei die Form sich ggf. von Bild zu Bild unterscheidet.

- Codierung von Kontur- und Transparenzinformation für jeden willkürlich geformten VOP durch die Codierung von binären oder grauwertigen *Alpha Plane*-Bilder.
- Codierung von Intra (I) codierten VOP's wie auch zeitlich prädierten (P) und bi-direktional prädierten (B) VOP's. Die Codierung von herkömmlichen MPEG-1/2 und H.263 [7] I, P und B-frames ist als Sonderfall möglich.
- Unterstützung von fester und variabler zeitlicher Bildwiederholfrequenz (*frame rate*) der VOP Eingangsbilder willkürlicher Form. Die Bildfrequenz kann wiederum abhängig von den Anwendungsanforderungen gewählt werden.
- Block-basierte (8 × 8 Pixel) sowie Makroblock basierte (16 × 16 Pixel) Bewegungskompensation und Bewegungskompensation in VOP's.
- Texturcodierung in I, P und B-VOP's mit Hilfe der Diskreten Cosinus Transformation (DCT), die ggf. an Regionen willkürlicher Form an den Rändern von VOP's angepasst ist. Die

Quantisierung der DCT-Koeffizienten und die anschließende Lauflängen-Entropiecodierung wird wie bei den MPEG-1/2 und H.261/3 Standards durchgeführt.

- Eine Rückwärtskompatibilität mit den H.261/3 und MPEG-1/2 Codieralgorithmen ist grundsätzlich möglich, da auch normale Bildsequenzen rechteckiger Form in einem VOP codiert werden können.

2.1 Unterstützung inhaltsbezogener Funktionalitäten – Codierung von Bildern in „Video Object Planes“

Um die oben angesprochenen bildinhaltsbezogenen Funktionalitäten bei der MPEG-4 Videocodierung berücksichtigen zu können, hat MPEG-4 das Konzept der so genannten Video Object Planes (VOP's) eingeführt. Es wird hierbei angenommen, dass jedes Bild einer Videosequenz in segmentierter Form – als eine Anzahl willkürlich geformter Bildbereiche – vorliegt (*object planes*). Jedes dieser Segmente kann beispielsweise ein physikalisches Objekt in einer Videoszene beschreiben. Im Gegensatz zu den Bildsequenzen, die von den herkömmlichen Standards wie MPEG-1 und MPEG-2 co-

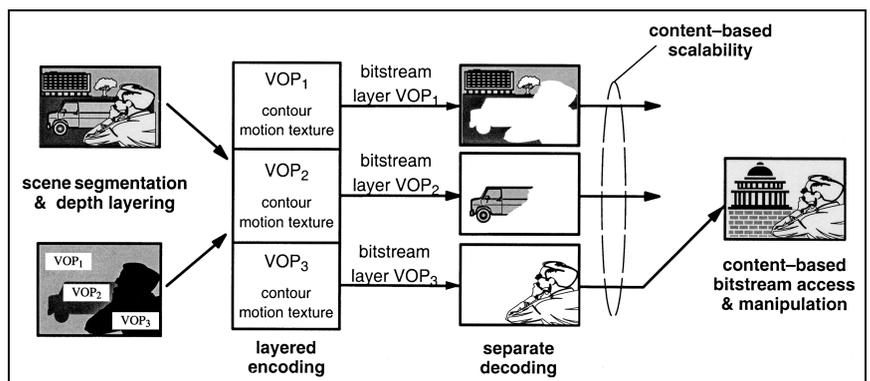


Bild 1 Die Codierung von Bildsequenzen mit Hilfe von MPEG-4 Video Object Planes (VOP's) ermöglicht die Unterstützung grundlegender inhaltsbezogener Funktionalitäten am Decoder. Jedes VOP beschreibt spezielle Bildinhalte und wird in einem separaten VOP-Layer codiert. Die Decodierung aller VOP-Layer ermöglicht die Rekonstruktion der Original-Bildsequenz in ihrer Gesamtheit. Einzelne VOP-Bildinhalte können separat decodiert und rekonstruiert werden (inhaltsbezogene Skalierbarkeit). Diese grundlegende Eigenschaft der VOP Struktur erlaubt die inhaltsbezogene Manipulation von Bilddaten am Decoder ohne eine erneute Transcodierung.

diert werden, wird für die MPEG-4 Codierung eine Bildsequenz nicht mehr notwendigerweise als rechteckig geformt angesehen.

Dieses Konzept der VOP Codierung ist in Bild 1 anhand eines Beispiels dargestellt. Eine Eingangsbildsequenz ist hier in drei VOP's zerteilt: Hintergrund VOP₁ sowie zwei Vordergrund VOP's, VOP₂ und VOP₃. Diese VOP's werden nun getrennt codiert, wobei sich die Form und Position der zu codierenden VOP Bildfolgen von Bild zu Bild ändern kann, je nach Bewegung der Bildinhalte (in unserem Beispiel des Autos und der Person). Die Zerlegung der Bildsequenzen kann entweder mit Hilfe so genannter on-line oder semi-on-line Segmentierungsalgorithmen durchgeführt werden, oder die einzelnen Bildinhalte sind zum Beispiel durch Chromakey Verfahren bereits bei der Videogenerierung segmentiert vorhanden.

Grundsätzlich werden für jeden VOP entsprechende Kontur-, Textur- und Bewegungsinformatio-

nen in einem separaten Layer codiert. Jeder Bitstrom-Layer beinhaltet auch Informationen über den Bildinhalt des Layers und auch darüber, wie die unterschiedlichen Layer am Decoder wieder zusammengesetzt werden – um so die Originalbildsequenz mit allen VOP's an den richtigen Positionen wieder zu rekonstruieren. Aufgrund dieser Informationen kann der Decoder nun auch einzelne Layer (z.B. Bildinhalte sowie deren Form, Bewegung und Textur) identifizieren und Objekte in Bildern völlig separat von den anderen Inhalten der Originalbildsequenz rekonstruieren. Innerhalb von MPEG-4 wird diese Funktionalität – also die Möglichkeit einzelne Bildinhalte zu identifizieren und völlig separat zu decodieren und zu rekonstruieren als „inhaltsbezogene Skalierbarkeit“ bezeichnet. Die inhaltsbezogene Skalierbarkeit von Video bildet die elementare Technik, um innerhalb des MPEG-4 Video Szenarios auch eine Reihe anderer bildinhaltsbezogener MPEG-4 Funktionalitäten auf Bitstromebene

ohne eine zusätzliche Segmentierung und Transcodierung zu ermöglichen. So zum Beispiel eine erhöhte Interaktivität mit Bildinhalten, die Portabilität von Bildinhalten in Multimediaanwendungen sowie die zeitliche und räumliche Skalierbarkeit von Bildinhalten. Die Möglichkeit der inhaltsbezogenen Bildmanipulation auf Bitstromebene ist in Bild 1 ebenfalls skizziert, indem der Bildinhalt von VOP₃ vor einem Hintergrund gezeigt wird, der in der Originalszene nicht enthalten war.

Um diese VOP-Konzept noch zu verdeutlichen, ist in Bild 2a ein Bild der MPEG-4 Testsequenz AKIYO gezeigt. Die Testsequenz besteht im Wesentlichen aus der sich bewegenden Vordergrundperson Akiyo und einem stationären Hintergrund, hier segmentiert in einen Vordergrund VOP₂ und einen Hintergrund VOP₁. Ein binäres Alpha-Plane Bild, wie in Bild 2b gezeigt, wird für jedes Bild der AKIYO Sequenz codiert, um so dem Empfänger die Kontur und Position des Vordergrundobjektes VOP₁ mit



Bild 2a–f Beispiel für die Beschreibung von VOP's mit Hilfe einer binären Alpha-Plane Maske. (a) Ein Bild der Originalsequenz AKIYO. (b) Binäre Alpha-Plane Maske, die die Form und Position des Vordergrundes in VOP₂ beschreibt. (c) Bildinhalt des Hintergrundes in VOP₁ für nicht-überlappende VOP's. (d) Bildinhalt des Vordergrundes VOP₂. (e) und (f) Bildinhalt des Hintergrundes in VOP₁ und des Vordergrundes in VOP₂ für überlappende VOP's.

Bezug auf den Hintergrund mitzuteilen. Erst danach werden für den Vordergrund, wie auch für den Hintergrund, die Textur und Bewegungsinformationen übertragen – getrennt für beide VOP's, um so eine separate Decodierung zu ermöglichen. Grundsätzlich ist im Algorithmus des MPEG-4 Video Standards auch die Möglichkeit der Grauwert-Alpha-Plane Codierung vorgesehen, um ggf. verschiedene Stufen von Transparenz zwischen VOP's bei der Rekonstruktion zu berücksichtigen.

Bild 2c und 2d zeigen am Beispiel des Originalbildes und der Segmentierungsmaske in Bild 2a und 2b die Bildinhalte der beiden zu codierenden VOP's. Die VOP's sind in diesem Beispiel *nicht-überlappend*, d.h. die Summe der Pixel in beiden VOP's entspricht der des Originalbildes in Bild 2a. Beide VOP's sind willkürlich geformt (VOP₁ hat ein Loch willkürlicher Form in der Mitte) und die Kontur und Position der VOP's hängt, wie bereits angedeutet, von der Bewegung der Vordergrundperson ab. MPEG-4 erlaubt zusätzlich die Codierung von sich *überlappenden* VOP's wie in den Bildern 2e und 2f dargestellt. Der Vordergrund in Bild 2f ist identisch mit dem in Bild 2d – allerdings ist in diesem Beispiel der Hintergrund in VOP₁ rechteckig geformt und beschreibt den gesamten, also auch den von der Vordergrundperson verdeckten, Hintergrund. Auch in dieser Anwendung werden beide VOP separat codiert, um so die separate Decodierung und Rekonstruktion von nur Hintergrund oder Vordergrund zu ermöglichen. Wenn der Bildinhalt des Hintergrundes in VOP₁ stationär ist, sich also nicht verändert, genügt bei der MPEG-4 Codierung die Übertragung eines einzelnen Hintergrundbildes, um zusammen mit der Information des VOP₂ Layers die Rekonstruktion der Originalsequenz zu ermöglichen.

Auf alle Fälle unterstützt MPEG-4 auch die Codierung von nur einem VOP mit normalen rechteckigen Bildsequenzen, um so eine Codierung wie bei MPEG-1/2 oder

H.261/3 zu ermöglichen. In diesem Sinne kann das VOP Konzept als konsequente Erweiterung des herkömmlichen *Single Layer* Konzeptes mit rechteckigen Bildern angesehen werden, wobei die Möglichkeit der Codierung wie bei den MPEG Standards als Sonderfall in dem MPEG-4 Szenario enthalten ist.

2.2 Codierung der Kontur-, Bewegungs- und Texturinformation innerhalb eines VOP's

Wie in Bild 1 angedeutet, werden die Informationen hinsichtlich Kontur, Bewegung und Textur für jeden VOP in einem separaten VOP-Layer codiert um so grundsätzlich ein separates Decodieren der VOP's zu ermöglichen. Der MPEG-4 Video Standard nutzt einen identischen Codieralgorithmus für jeden VOP-Layer. Dieser Codieralgorithmus basiert auf viele der Techniken und Tools, die schon bei den MPEG-1/2 und H.261/3 Algorithmen erfolgreich zur Anwendung kommen [1]. Wie in Bild 3a anhand einer Bildsequenz dargestellt (hier als Sonderfall eines VOP's mit rechteckigen Bildern), wird das erste VOP-Bild in Intra Mode codiert (I-VOP). Jedes danach folgende Bild wird mit Hilfe einer Inter-Bild Prädiktion (P-VOP) codiert – also nur Pixel des letzten schon codierten VOP-Bildes werden für eine bewegungskompensierte Prädiktion genutzt. MPEG-4 unterstützt ebenfalls die Codierung von bi-direktional prädizierten VOP-Bildern (B-VOP's). Für rechteckige Bildsequenzen ist die P-VOP und B-VOP Prädiktion identisch

mit der Standard MPEG P-frame und B-frame Prädiktion.

Ähnlich wie bei den H.261/3 und MPEG-1/2 Standards werden VOP-Bilder mit dem MPEG-4 Verfahren blockbasiert verarbeitet und codiert. In einem ersten Schritt wird jedoch mit Hilfe der Konturcodierung für ein VOP-Bild zunächst die Konturinformation codiert, um so mit dem Decoder gemeinsam ein Blockraster zu definieren – dazu wird jedes VOP-Farbbild entsprechend seiner Form in sich nicht überlappende Makroblöcke zerteilt. Dies ist anhand der rechteckigen VOP-Bildsequenz in Bild 3b dargestellt und in Bild 5 für ein willkürlich geformtes VOP-Bild. Jeder Makroblock besteht dabei aus 6 Blöcken, die jeweils Pixel aus den Luminanz und Chrominanzbändern enthalten – vier Luminanz Blöcke (Y₁, Y₂, Y₃, Y₄) und zwei Chrominanz Blöcke (U, V), jeder Block mit 8 × 8 Pixeln. Das grundlegende MPEG-4 Codierschema für die Codierung der Y- und U-Blöcke ist, wie bereits bei den MPEG und ITU-T Standards, ein hybrides blockbasiertes MC/DCT Verfahren – bestehend aus Verfahren zur bewegungskompensierten Prädiktion (*motion compensated*: MC) von Pixeln zwischen VOP-Bildern und der Codierung der Prädiktionsfehler mit Hilfe der Diskreten Cosinus Transformation (DCT). Die Struktur dieses Codiersystems ist in Bild 4 skizziert. Ein bereits codiertes VOP-Bild N – 1 ist in einem VOP-Bildpeicher (frame store: FS) am Coder und Decoder gespeichert. Eine bewegungskompensierte Prädiktion (MC) wird

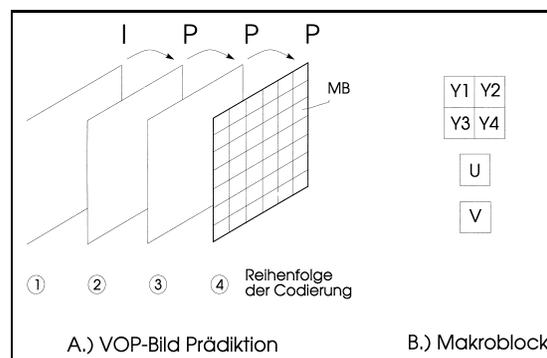


Bild 3 (a) I-Bild VOP (I-VOP) und P-Bild VOP's (P-VOP's) in einer Videosequenz. (b) Für jeden Makroblock (MB, 16 × 16 Pixel) werden Pixelinformationen für vier Luminanz Blöcke (Y₁, Y₂, Y₃, Y₄, jeder 8 × 8 Pixel) codiert.

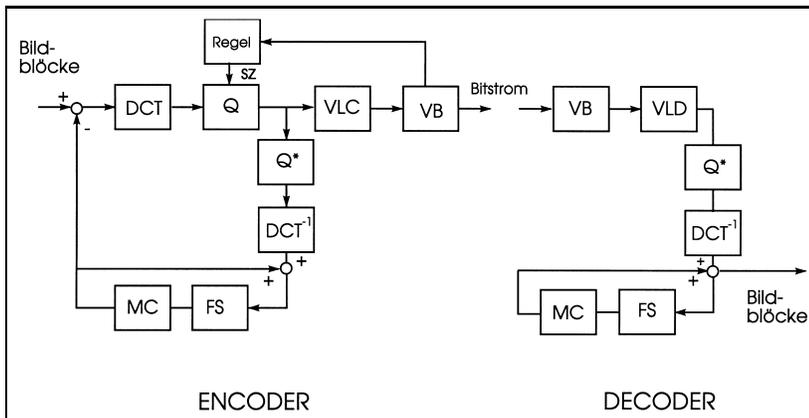


Bild 4 Blockdiagramm eines MPEG-4 Coders und Decoders. Das grundlegende Codierschema ist wie bei den MPEG-1 und MPEG-2 Standards blockorientiert und basiert auf dem erfolgreichen Konzept der hybriden MC/DCT Codierphilosophie.

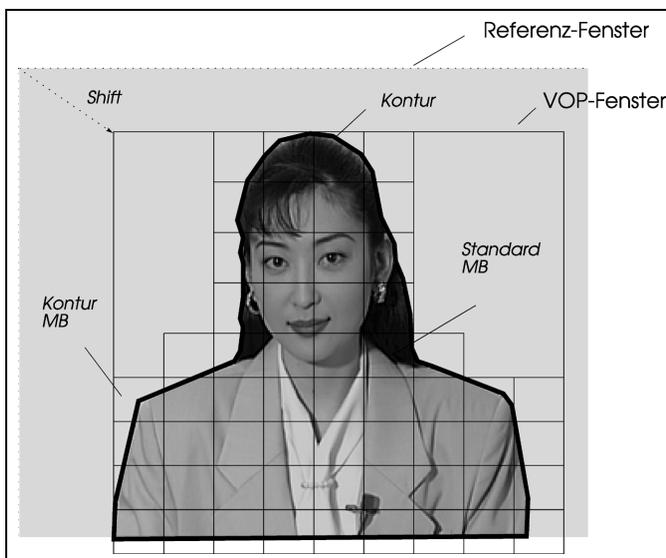


Bild 5 Beispiel eines konturadaptiven MPEG-4 Makroblock-Gitters für das AKIYO Vordergrund VOP₂ Bild. Ein VOP-Fenster, mit einer Größe, die einem Vielfachen von 16×16 Pixeln entspricht, umrandet das Vordergrundobjekt und spezifiziert so konturadaptiv ein Makroblock-Gitter. Das VOP-Fenster berührt am oberen Ende und an der linken Seite die Kontur des Vordergrundobjektes. Ein Shift-Parameter beschreibt die Position des VOP-Fensters innerhalb eines „Referenzfensters“.

wahlweise auf Block oder Makroblöcken durchgeführt – nur ein Bewegungsvektor wird für je einen Block oder Makroblock codiert [1]. Der Prädiktionsfehler wird berechnet, indem jeder Pixelwert in einem Block oder Makroblock in VOP-Bild N von seinem bewegungsverschobenen Äquivalent im vorherigen VOP-Bild $N - 1$ subtrahiert wird. Zur Codierung wird dann eine 8×8 DCT auf jeden Block angewandt. Die Transformations-Koeffizienten werden daraufhin quantisiert (Q) und mit einer Lauffängen- und Entro-

piecodierung (VLC) codiert. Ein VOP-Buffer (VB) wird genutzt, um ggf. eine konstante Bitrate am Coderausgang zu generieren. Für die Regelung der Bildqualität in Abhängigkeit der zur Verfügung stehenden Datenrate kann die Quantisierungsschrittweite (sz) der DCT-Koeffizienten genutzt werden. Der VOP-Decoder nutzt im Wesentlichen den umgekehrten Prozess zur Rekonstruktion der VOP-Bilder.

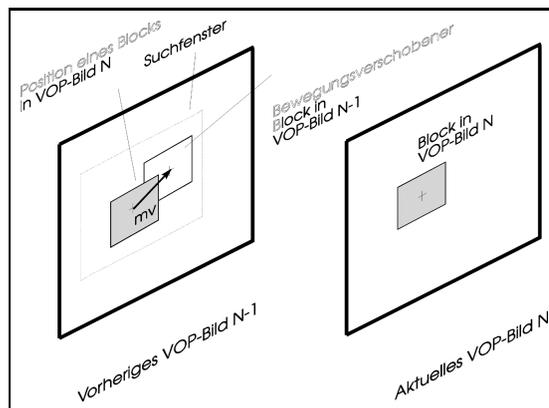
Im Allgemeinen können die zu codierenden VOP-Bilder willkürlich geformt sein und die Form und Po-

sition der VOP-Bilder kann sich, in Bezug auf ein über eine VOP-Sequenz konstantes Referenzfenster, von Bild zu Bild ändern. MPEG-4 nutzt hierfür ein Konzept für die blockbasierte Codierung, das auf der Definition eines VOP-Bildfensters variabler Form und Position basiert. Dieses VOP-Bildfenster ist in Bild 5 skizziert und bewegt sich innerhalb des oben angesprochenen Referenzfensters konstanter Größe und Position. Alle zu codierenden VOP's einer Bildsequenz sind in Bezug auf dieses Referenzfenster definiert. Für jedes VOP-Bild wird dann mit Hilfe des entsprechenden VOP-Fensters in einem ersten Schritt die Konturinformation des VOP's codiert. Sowohl Coder wie auch Decoder verfügen dann über die VOP-Konturinformation, die in einem nächsten Schritt für die Definition eines konturangepassten Makroblock-Gitters genutzt wird. In weiteren Schritten wird nur Bewegungs- und Texturinformation für die Blöcke und Makroblöcke codiert, die entweder ganz innerhalb eines VOP's liegen, oder aber die Kontur eines VOP's schneiden.

Konturcodierung. Grundsätzlich wird die Konturcodierung in MPEG-4 als Alpha-Plane Codierung bezeichnet. MPEG-4 hat zwei Arten der Alpha-Plane Codierung vorgesehen: die binäre Alpha-Plane Codierung (siehe auch Bild 2b) und die Codierung von grauwertigen Alpha-Planes. Für die binäre Alpha-Plane Codierung ist ein hierarchischer blockbasierter Quadtree Algorithmus beschrieben. Für die Alpha-Plane Grauwertcodierung, mit der neben der Konturinformation auch noch Transparenz zwischen VOP's zum Decoder signalisiert wird, ist ein Quadtree Algorithmus mit Grauwert-Vektorquantisierung spezifiziert.

Bewegungsschätzung und Bewegungskompensation. Für die block- oder makroblockbasierte Bewegungsschätzung und Bewegungskompensation zwischen aufeinander

Bild 6 Block-Matching Verfahren für die Bewegungsschätzung und Bewegungskompensation.



folgenden VOP-Bildern unterschiedlicher Form und Position wird das konturadaptive Makroblock-Gitter genutzt. Ein Standard *Block-Matching* Verfahren [1], wie in Bild 6 dargestellt, wird benutzt, um für einen Block oder Makroblock in VOP-Bild N einen Bewegungsvektor (mv) aus einem VOP-Bild $N-1$ zu schätzen. MPEG-4 unterstützt die Codierung von einem Bewegungsvektor pro 8×8 Block oder 16×16 Makroblock sowie eine blocküberlappende Bewegungskompensation, wie sie etwa für den ITU H.263 Standard spezifiziert wurde [7].

Für die Bewegungsschätzung und -kompensation von Pixeln in Blöcken an Konturgrenzen muss das oben beschriebene Verfahren jedoch angepasst werden. Hierfür wird bei MPEG-4 Video ein Pixel-Extrapolationsverfahren für das VOP-Referenzbild $N-1$ genutzt. Ziel der Pixel-Extrapolation ist es,

im VOP Referenzbild $N-1$ für die Bewegungsschätzung und die Bewegungskompensation Pixelwerte außerhalb der Kontur des ursprünglichen VOP-Bildes zu erzeugen, um so eine umfassende Vektorsuche zu ermöglichen. In Bild 7 ist das Ergebnis einer solchen Pixel-Extrapolation dargestellt. Für die Vektorsuche werden dann allerdings nur die Pixel genutzt, die innerhalb des VOP-Bildes liegen.

Texturcodierung. Wie angedeutet werden sowohl in I-VOP's wie auch die Prädiktionsfehler in P- oder B-VOP's mit Hilfe der 8×8 DCT codiert. Wiederum ist für die Blöcke an den VOP-Grenzen eine Anpassung notwendig, da innerhalb eines Blockes die Werte der Pixel, die nicht zum VOP-Bild gehören, nicht definiert sind. Auch hier werden die Werte dieser Pixel mit Hilfe



Bild 7 Pixel-Extrapolation außerhalb der VOP-Grenzen für die Bewegungsschätzung, Bewegungskompensation und die DCT-Codierung der Textur.

der oben beschriebenen Pixel-Extrapolation berechnet, um so die Codiereffizienz zu erhöhen (siehe Bild 7).

Multiplexen der Kontur-, Bewegungs- und Texturinformation. Nahezu alle Tools (z.B. die DCT, blockbasierte Bewegungskompensation, etc.), wie sie in den H.263 und MPEG-1 bzw. MPEG-2/Main Profile Standards definiert sind, werden von MPEG-4 unterstützt. Die komprimierten Alpha-Plane-Informationen sowie die Bewegungsvektoren und DCT-Bits werden für jeden VOP in einen eigenen VOP-Layer Bitstrom multiplext, indem zunächst die Konturinformation übertragen/gespeichert wird, gefolgt von Bewegungsvektor-Bits und DCT-Bits ausgehend von MPEG und H.263 Makroblockdefinitionen.

3 Zusammenfassung und Ausblick

In diesem Aufsatz wurden die Ziele und die grundlegende Struktur des objektbasierten Teils des MPEG-4 Video Codierstandards aufgezeigt. Es ist allerdings hervorzuheben, dass MPEG-4 alle bereits bekannten Funktionalitäten existierender Videokompressionsverfahren (MPEG-1, MPEG-2, H.263, etc.) mit großer Effizienz ermöglicht. Dies betrifft hauptsächlich die Kompression von herkömmlichen Videosequenzen mit hoher Qualität und die hohe Robustheit von MPEG-4 bei Übertragungsfehlern in mobilen Netzen.

Die neue, inhaltsbezogene Funktionalität unterscheidet MPEG-4 von allen bisher bekannten Videokompressionsverfahren. Diese Funktionalität ermöglicht eine Vielzahl von innovativen Produkten in einem breiten Spektrum von Anwendungsfeldern. Aus der Vielzahl möglicher Anwendungen ist in Bild 8 eine immersive Video-Telekommunikationsanwendung mit MPEG-4 Codierung skizziert. Ziel dieser MPEG-4 Anwendung ist eine hohe Telepräsenz bei Videokonferenzsystemen. Teilnehmer an un-



Bild 8 Immersive Mixed-Reality Videokonferenz mit MPEG-4 (Quelle: Heinrich Hertz Institut, FhG, Berlin). Die Teilnehmer der Konferenz an unterschiedlichen Standorten sind segmentiert, mit MPEG-4 VOP's übertragen und in einer virtuellen Konferenzumgebung zusammengemischt. Es entsteht der Eindruck einer normalen Tisch-Konferenzsituation mit gegenüber herkömmlicher Videokonferenz erhöhter Telepräsenz.

terschiedlichen Standorten werden separat aufgenommen. Die jeweilige Person im Videobild wird mittels Bild-Segmentierungsverfahren vom Hintergrund getrennt und separat als MPEG-4 Objekt (VOP) zu den anderen Teilnehmern übertragen. Bei den Empfängern werden alle so als MPEG-4 Objekte übertragenen Personen in ein gemeinsames Bild mit neuem Hintergrund zusammengemischt. Hierdurch lässt sich – bei entsprechender Übertragungsqualität – eine normale Tisch-Konferenzsituation mit hoher Telepräsenz erzeugen.

Interessant ist die Frage, ob sich eine solche inhaltsbezogene Funktionalität mit MPEG-4 bit-effizient unterstützen lässt. Tatsächlich zeigen die Feldversuche in MPEG, dass Bilder entweder objektbasiert oder normal wie bei MPEG-2 mit etwa gleicher Bitrate und Qualität übertragen werden können. Durch die Unterstützung der objektbasierten Kodierung ist also kein Verlust an Codiereffizienz zu beobachten. Allerdings steigt die Komplexität von Coder- und Decoderhardware bzw. -Software.

Eine solche MPEG-4 Telepräsenz-anwendung ist im Rahmen verschiedener europäischer und natio-

ner Projekte bereits implementiert worden, in Deutschland am Heinrich-Hertz-Institut der Fraunhofer Gesellschaft in Berlin. Andere Projekte konzentrieren sich auf Internetanwendungen wie z.B. im EU Projekt SoNG (*portals of Next Generation*, www.song-opi.org). Hier wurde ein virtueller Telefonshop entwickelt, bei dem ein Verkäufer per segmentiertem Video den Kunden beim Kauf beraten kann.

Zum heutigen Zeitpunkt haben sich aber noch keine nennenswerten Produkte am Markt etabliert, die die neuen, inhaltsbasierten MPEG-4 Funktionalitäten nutzen. Die nächsten 5 Jahre werden zeigen, ob sich diese Funktionalitäten am Markt durchsetzen können. Dagegen haben sich die herkömmlichen Funktionalitäten von MPEG-4 (Kompression und Fehlerrobustheit) bereits in vielen Anwendungen bewährt, wie beispielsweise in Camcordern und in vielen Mobiltelefonen. Insbesondere im Internet ist MPEG-4 ein beliebtes Format für die Übertragung und Speicherung von digitalem Video, getrieben von Produkten wie z.B. DIVX und XviD und das open Source Projekt MPEG4IP (<http://mpeg4ip.sourceforge.net>).

Literatur

- [1] T. Sikora, L Chiariglione: „The MPEG-4 Video Standard and its Potential for Future Multimedia Applications“, IEEE ISCAS Conference, Honkong, June 1997.
- [2] T. Sikora: „MPEG Digital Video Coding Standards“, IEEE Signal Processing Magazine, Vol. 14, No. 5, September 1997.
- [3] R. Schäfer, T. Sikora: „Digital video coding standards and their role in video communications“, Proc. of the IEEE, Vol. 83, No. 6, June 1995.
- [4] T. Sikora: „The MPEG-4 Video Standard Verification Model“, IEEE Trans. Circuits and System for Video Technology, Vol. 7, No. 1, February 1997.
- [5] D. J. Le Gall: „The MPEG Video Compression Algorithm“, Signal Processing: Image Communication, Vol. 4, No. 4, 1992.
- [6] ISO/IEC 13818-2 MPEG-2 Video Coding Standard: „Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video“, March 1995.
- [7] ITU-T Group for Line Transmission of Non-Telephone Signals: „Draft Recommendation H.263 – Video Coding for Low Bitrate Communication“, December 1995.
- [8] F. Pereira, T. Ebrahimi: „The MPEG-4 Book“, Prentice Hall, 2002.



Prof. Dr.-Ing. Thomas Sikora lehrt an der Technischen Universität Berlin im Fachgebiet Nachrichtenübertragung. Er beschäftigt sich seit etwa 15 Jahren mit Aspekten der Videokodierung. Als Chairman der ISO-MPEG-Gruppe war er verantwortlich für die Entwicklung des MPEG-4 Standards. Adresse: Technische Universität Berlin, Institut für Telekommunikationssysteme, Fachgebiet Nachrichtenübertragung, Sekr. EN 1, Einsteinufer 17, D-10587 Berlin. E-Mail: sikora@nue.tu-berlin.de