

Bachelorarbeit
Untersuchung der Leistungsfähigkeit von
Software- und Hardware-AV-Codecs auf verschiedenen
Windows-Betriebssystemen

Thomas Glenz IN99 B(T)

14. Oktober 2002

1. Vorwort	3
2. Theoretische Grundlagen	5
2.1 Video- und Audiosignal	5
2.2 Kompressionsverfahren.....	5
2.2.1 M-JPEG	5
2.2.1.1 Die Indexverschiebung.....	7
2.2.1.2 Diskrete Kosinus Transformation (Forward DCT)	7
2.2.1.3 Quantisierung	9
2.2.1.4 Kodierung der Koeffizienten.....	11
2.2.2 MPEG.....	15
2.2.2.1 I-Bild (intra-codiertes Bild).....	16
2.2.2.2 P- und B-Bilder (prädiktiv-codierte Bilder, bidirektional ~)	17
2.2.3 H.263	19
2.2.4 Vergleich der Verfahren.....	19
2.3 Codecs	20
2.4 Datenübermittlung.....	21
2.4.1 Übertragungswege.....	21
2.4.2 Das RTP-Protokoll	22
2.4.2.1 Grundlagen	22
2.4.2.2 Aufbau des RTP-Paketes.....	24
2.4.2.3 Multiplexing	25
2.4.2.4 Mixer und Übersetzer.....	26
2.4.2.5 Zur Zeit definierte Daten-Typen	26
2.4.2.6 Das RTCP-Protokoll	27
2.4.2.7 Der RTCP-Header	30
2.4.2.8 Bandbreiten-Management	31
3. Untersuchung der verwendeten Codecs	33
3.1 Versuchsdurchführung	33
3.2 H263-Codec	34
3.3 M-JPEG-Codec	35
4. Schlußbetrachtung.....	37
Anhang A - Screenshots.....	38
Anhang B – Glossar	43
Anhang C- Literaturverzeichnis	45

1. Vorwort

Mit der Erfindung des Telefons im Jahre 1863 entstand eine neue Form der Kommunikation. Nunmehr war es möglich, über große Entfernungen hinweg ohne Verzögerungen zu kommunizieren. Mit der Zeit wurden Verbesserungen und Neuerungen eingeführt, die es zum Beispiel erlauben, dass mehrere Teilnehmer gleichzeitig miteinander reden können. Die einzige Einschränkung, die bis heute geblieben ist, stellt die Beschränkung auf gesprochene Informationen dar. So fehlt die Möglichkeit, die gesprochenen Informationen durch Visualisierung zu unterstützen. Dieses Manko sollte Ende der 90er Jahre durch Bildtelefone behoben werden. Allerdings waren diese Geräte teuer in der Anschaffung, lieferten gleichzeitig aber kaum mehr als Einzelbilder, so dass sie sich nie durchsetzen konnten.

Hier greift das Konzept der Videokommunikation. Gleichzeitige Übertragung von Ton- und Bild-Daten soll es ermöglichen, während des Gesprächs den Gesprächspartner zu sehen. Während dies für alltägliche Telefonate eher eine Spielerei ist, eröffnet sie in anderen Bereichen völlig neue Möglichkeiten. So ist zum Beispiel im Bereich der Bildung denkbar, dass eine Vorlesung in mehrere, von einander weit entfernte Hörsäle übertragen wird und die Studenten die Möglichkeit haben, Fragen an den Dozenten zu stellen. In der Medizin wäre es für einen Arzt möglich, im Falle bestimmter Erkrankungen mit einem Spezialisten Kontakt aufzunehmen, der dann per Videokonferenz an der Behandlung des Patienten teilnehmen kann.

Zwar wird Videokonferenz heutzutage bereits eingesetzt, jedoch vor allem nur in großen Unternehmen, in Medien oder im wissenschaftlichen Bereich. Im privaten Bereich ist sie dagegen noch nicht verbreitet. Die Hauptgründe liegen zum einem in dem noch hohen technischen Aufwand und den damit verbundenen Kosten, zum anderen an der Telekommunikations-Infrastruktur. Während einer Videokonferenz muss eine große Datenmenge übertragen werden und an den dafür nötigen Datenleitungen mangelt es noch.

Allerdings ermöglichen die steigende Anzahl an privaten DSL-Anschlüssen, die Verbesserungen bei AV-Kompressionsverfahren und nicht zuletzt die Entwicklungen im Hardwarebereich auch den Einzug von Videokommunikation im privaten Bereich. Aus diesem Grund soll im Rahmen dieser Arbeit die Leistungsfähigkeit heutiger Videokompressionsverfahren untersucht werden. Dabei wird ein System untersucht, welches plattformunabhängige Entwicklungen möglich macht. Besonders wird dabei geprüft, welche Bildqualität in Abhängigkeit verschiedenen Datenraten erreicht werden kann. Da die

verschiedenen Verfahren unterschiedliche Algorithmen für die Datenkompression verwenden und mit unterschiedlichen Zielen entwickelt wurden, soll getestet werden, für welche Einsatzgebiete diese sich eignen.

Zum Einsatz kamen zum einen ein H263- zum anderen ein M-JPEG-Codec, welche beide in Java implementiert wurden und somit plattformunabhängige Software ermöglichen. Beide Codecs zeigten deutliche Unterschiede, sowohl in der erreichten Qualität als auch in den auftretenden Datenraten. So wurde deutlich, dass eine gute Bildqualität hohe Bandbreite für die Übertragung voraussetzt. Sowohl gute Bilder als auch kleine Datenraten sind mit den untersuchten Technologien nicht möglich. Allerdings könnte der Einsatz anderer Verfahren wie MPEG hier Abhilfe schaffen. Leider war es nicht möglich die MPEG-Kompression unter gleichen Bedingungen zu untersuchen, da kein passender Codec zu Verfügung stand. Allerdings gibt es für die Zukunft hier positive Signale. So wird u.a. von IBM ein MPEG4-Codec für Java entwickelt.

2. Theoretische Grundlagen

2.1 Video- und Audiosignal

Für Videokommunikation ist es wichtig, dass die AV-Daten verzögerungsfrei übertragen werden, gleichzeitig aber von guter Qualität sind. Das heißt, die von der Kamera bzw. vom Mikrophon aufgenommenen Signale müssen im PC so aufgearbeitet werden, dass sie in Form eines Video-Streams über die bestehende Datenleitung übertragen werden können.

Zu einem Problem wird hier die entstehende Datenmenge, wobei dies insbesondere das Videosignal betrifft. Ein Bild im PAL-Format mit einer Auflösung von 710x576 Pixeln (=442.368 Pixel) und einer Farbtiefe von 24 Bit benötigt einen Speicherplatz von 1.327.104 Byte (442.368 Pixel x 3 Byte Farbtiefe). Bei einer üblichen Framerate von 25 fps bedeutet dies einen Speicherbedarf von 33 MB pro Sekunde. Halbiert man die Auflösung und verwendet eine Farbtiefe von 16 Bit, fallen immer noch 5 MB an Videodaten an. Eine Ethernet-Verbindung mit 100 MBit wäre nicht in der Lage, die Datenmenge des ersten Beispiels schnell genug zu übertragen, um das Video ohne zeitliche Verzögerung auf dem Zielrechner darstellen zu können.

Der einzige Ausweg besteht darin, die AV-Signale im PC zu komprimieren, um die zu übertragende Datenmenge zu reduzieren.

2.2 Kompressionsverfahren

2.2.1 M-JPEG

Beim M-JPEG--Verfahren werden die Einzelbilder der Videosequenz JPEG-komprimiert.

Bei der JPEG-Kompression stehen 4 verschiedene Modi zur Verfügung:

sequentiell mode:

Ein einzelner Durchgang durch das Bild von links oben nach rechts unten dekodiert das Bild. Dieser Modus ist für die meisten Anwendungen gut geeignet, liefert die besten Kompressionsraten und ist am leichtesten zu implementieren.

progressive mode:

Das Bild wird in mehreren Durchgängen kodiert bzw. dekodiert. Dabei wird das Bild von Durchgang zu Durchgang schärfer. Das Einsatzgebiet für dieses Verfahren ist

beispielsweise die Datenfernübertragung von Bildern. Sobald das Bild eine ausreichende Schärfe erreicht hat, kann die Übertragung abgebrochen werden.

hirarchical mode:

Das Bild wird erst in einer geringeren Auflösung gespeichert und dann in der vollen Auflösung. Das Bild mit der geringeren Auflösung kann bedeutend schneller dekodiert werden, ist somit gut als schnelles Preview des Bildes geeignet. Dies ist sinnvoll für den Einsatz in Bilddatenbanken, bei denen das Bild in der geringen Auflösung als Entscheidungshilfe gilt, ob das Bild den Wünschen entspricht.

lossless mode:

In diesem Modus wird, im Gegensatz zu den anderen Modi, verlustfrei kodiert und dekodiert. Jedes Bit wird genau so wiederhergestellt wie es kodiert wurde. Die verlustfreie Kodierung wird benötigt um Bilder zu komprimieren, die keinerlei Abweichungen zulassen oder nicht vom Menschen ausgewertet werden. Dieser Modus bietet verständlicherweise eine geringere Kompressionsrate, als die anderen Modi, da bei verlustfreier Kompression der Informationsgehalt nicht reduziert werden kann.

Da der progressive (progressiv mode) und der hierarchische Modus (hirarchical mode) aufgrund der Art, wie die Bilder dekodiert werden, für Videosequenzen ungeeignet sind und der verlustfreie Modus (lossless mode) eine zu geringe Kompressionsrate aufweist, bleibt als einziger anwendbarer Modus der sequenzielle (sequentiell mode) für das M-JPEG-Verfahren.

Die Komprimierung der Einzelbilder erfolgt in vier Schritten:

1. Indexverschiebung
2. Forward DCT
3. Quantisierung
4. Huffman Kompression

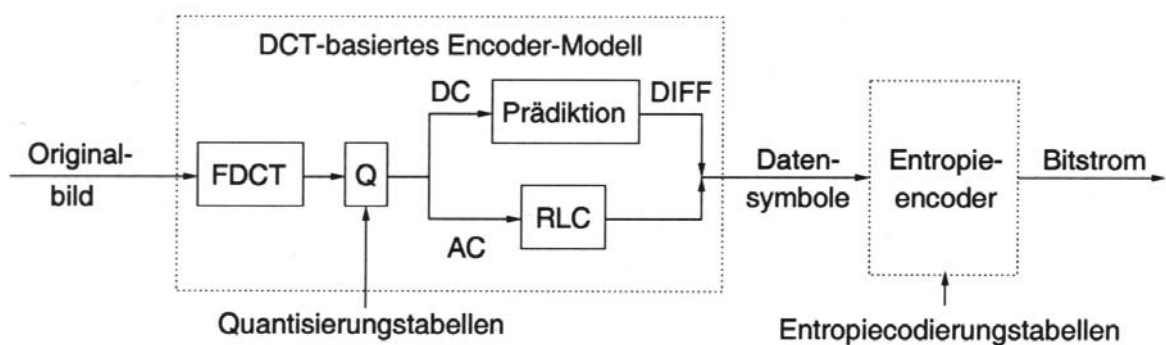


Abbildung 2.1: Blockschaltbild für die JPEG-Encodierung

Diese werden nun im Detail erläutert:

2.2.1.1 Die Indexverschiebung

Bei der Indexverschiebung werden die Werte des Blockes aus dem Bereich $0 \dots 2^P - 1$ in den Bereich $-2^{P-1} \dots 2^{P-1}$ verschoben. Im Falle der sequentiellen Modus mit $P=8$ bedeutet dies eine Verschiebung vom Bereich $0 \dots 255$ in den Bereich $-127 \dots +127$. Da die Kosinusfunktion symmetrisch ist, ergeben sich durch die Indexverschiebung nach der DCT kleinere Werte, die nach der Quantisierung besser zu komprimieren sind und damit eine höhere Kompressionsrate ergeben.

2.2.1.2 Diskrete Kosinus Transformation (Forward DCT)

Bei der Diskreten Kosinus Transformation (Bild 2.1 - FDCT) wird das Bild in kleine Teilbereiche der Größe 8×8 zerlegt. Der 8×8 Block wird als ein diskretes Signal mit 64 Werten aufgefasst, die abhängig von den zwei räumlichen Dimensionen x und y sind. Diese Werte werden in ihr Spektrum transformiert, indem die FDCT einen Basiswechsel auf eine Basis von 64 orthogonalen, diskreten Signalen durchführt. Die Koeffizienten für diese Basissignale bilden die Ausgabewerte. Die DCT ist verwandt mit der weithin bekannten Fourier-Transformation.

Der Wert mit der horizontalen und der vertikalen räumlichen Frequenz von 0 wird als der Gleichstromkoeffizient (direct current, Bild 2.1 - DC) bezeichnet, die restlichen 63 Werte als die Wechselstromkoeffizienten (alternating current, Bild 2.1 - AC). Der DC-Wert ist der durchschnittliche Wert der 64 Punkte mal 8. In einem natürlichen Bild sind scharfe Linien und abrupte Farbwechsel eher selten. Daher ist der Unterschied zwischen 2 benachbarten Punkten meist sehr gering. Somit konzentriert sich der größte Teil des Ausgangssignals in den unteren Frequenzen. Die meisten AC-Werte sind deshalb sehr klein. Ein natürliches Bild hat gewöhnlich nach der FDCT folgende Eigenschaften.

- der DC-Koeffizient ist mit Abstand der größte Wert.
- die AC-Koeffizienten haben mit steigender Ortsfrequenz kleinere Werte (Bild 2.2 a).
- die Werte der meisten AC-Koeffizienten liegen nahe bei 0.

Die Rücktransformation IDCT kehrt den Transformationsprozess um. Sie bildet die 64 Werte des Frequenzbereiches in den Ortsbereich ab. Dies geschieht durch Aufsummierung der Basissignale für jeden Punkt des Ortsbereiches.

Die mathematischen Formeln für die Hin- und Rücktransformation lauten:

FDCT (forward discrete cosine transform)

$$F[u, v] = \frac{C_u C_v}{4} \sum_{x=0}^7 \sum_{y=0}^7 f[x, y] \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

IDCT (inverse discrete cosine transform)

$$f[x, y] = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C_u C_v}{4} F[u, v] \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

mit

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases} \quad C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } v = 0 \\ 1 & \text{für } v \neq 0 \end{cases}$$

Mathematisch gesehen sind beide Funktionen eineindeutig, das heißt, dass die Punkte des Ortsbereiches nach der Abbildung in den Frequenzbereich identisch wiederhergestellt werden können und umgekehrt. Es geht keine Information verloren, also ist dieser Schritt verlustfrei.

Praktisch können heutige Rechenanlagen, bedingt durch ihren physikalischen Aufbau, den Kosinus mit großer Genauigkeit aber nicht exakt ermitteln. Dadurch ergeben sich Abweichungen der praktisch ermittelten zu den mathematisch korrekten Werten. Dadurch ergibt sich ein Informationsverlust im Bild.

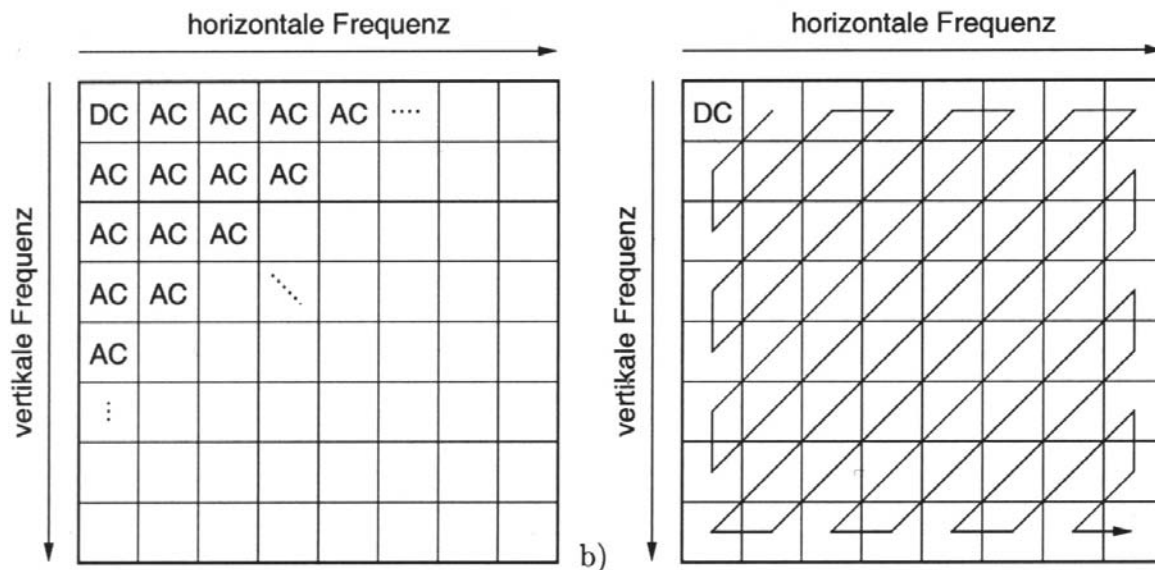


Abbildung 2.2: a) Position der DCT-Koeffizienten in Abhängigkeit von der Ortsfrequenz; b) Zick-Zack-Abtastung der Koeffizienten nach aufsteigenden Frequenzen

2.2.1.3 Quantisierung

Das Ergebnis der DCT sind ein DC und 63 AC Koeffizienten. Die 63 AC Werte variieren nur wenig. Mit steigender Ortsfrequenz erhält man kleinere Werte, die nur geringfügig größer bzw. kleiner als 0 sind. Diese Werte werden nun quantisiert, wobei mehrere Werte auf eine Zahl abgebildet werden können. Damit gehen Information verloren. Jeder der 64 Werte wird durch einen ihm zugeordneten Wert aus einer Quantisierungsmatrix dividiert. Dabei wird der Wert an der Position $F(u,v)$ durch den Wert $Q(u,v)$ der Quantisierungstabelle dividiert und das Ergebnis auf die nächste ganze Integerzahl gerundet.

$$\tilde{F} = \text{Integer} - \text{Round}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

Die Quantisierungstabelle benutzt eine feinere Quantisierung für die Koeffizienten der niedrigen Frequenzen und eine gröber Quantisierung für die Koeffizienten der höheren Frequenzen. Damit erreicht man, dass die höheren Frequenzen im Ergebnis einen Wert von 0 haben. Der JPEG-Standard macht bereits einen Vorschlag für die Quantisierungswerte, deren Wahl mit Berücksichtigung der menschlichen Wahrnehmung erfolgte. In einer Testreihe der JPEG-Gruppe wurde für jeden Koeffizienten ein mittlerer Quantisierungswert ermittelt, bei dem die Veränderung des Koeffizienten im rücktransformierten Bild noch nicht zu erkennen

ist. In Bild 2.3 sind diese aufgeführt. Die verwendeten Quantisierungstabellen werden in den Bitstrom eingefügt, damit der Decoder eine korrekte Rekonstruktion durchführen kann.

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	
a)	78	72	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
b)	99	99	99	99	99	99	99

Abbildung 2.3: Matrizen mit Quantisierungswerten für Luminanz (a) und Chrominanz (b)

Die Umkehrung der Quantisierung ist eine einfache Multiplikation des Wertes mit dem entsprechenden Eintrag in der Quantisierungstabelle.

$$F(u, v) = \tilde{F}(u, v) Q(u, v)$$

Hierbei wird deutlich, dass der Originalwert nicht wiederhergestellt werden kann, da, abgesehen von Rechenungenauigkeiten beim DCT, hier der eigentliche verlustbehaftete Teil des Verfahrens liegt. Jedoch bildet er durch die Reduktion der Daten die Grundlage zur effizienten Komprimierung der Daten. Zwar ist es möglich eine Quantisierungstabelle zu wählen, deren Einträge aus lauter Einsen bestehen, wodurch der Datenverlust bei der Quantisierung eliminiert, aber auch der Komprimierung eine wichtige Grundlage entzogen wird. Da aber beim DCT weiterhin ein Verlust entsteht, kann das Original also nicht verlustfrei kodiert werden.

Die Quantisierung beeinflusst die Genauigkeit der DCT-Koeffizienten. Bei einer Genauigkeit von 8 Bits pro Bildpunkt im Originalbild, führt die DCT zu einer 11-Bit-Genauigkeit, die Quantisierung mit dem Wert 16 schließlich nur noch zu einer Genauigkeit von 7 Bits.

Die blockweise Verarbeitung der Bildpunkte wirkt sich bei einer stärkeren Quantisierung nachteilig aus, da es zu Diskontinuitäten an den Blockgrenzen, Blockartefakte genannt,

kommt. Im Extremfall werden alle AC-Koeffizienten zu Null quantisiert und der gesamte Block nur noch durch den DC-Koeffizienten repräsentiert. Somit haben alle Bildpunkte des Blocks den selben Grauwert. Das Bild setzt sich in diesem Fall nur noch aus Blöcken mit konstanten Grauwerten zusammen.

2.2.1.4 Kodierung der Koeffizienten

Nach der Quantisierung ist ein Großteil der gespeicherten Information redundant. Somit wurde eine gute Ausgangsbasis zum komprimieren der Daten erreicht. Werte, die häufiger auftreten als andere, können mit kurzen Symbolen, seltenere Werte mit längeren Symbolen kodiert werden. Der Informationsgehalt eines Symboles mit der Wahrscheinlichkeit p beträgt in Bits $I = \log_2\left(\frac{1}{p}\right)$. Für Werte mit hoher Wahrscheinlichkeit sinkt der Logarithmus, es werden also weniger Bits benötigt. Je niedriger die Wahrscheinlichkeit für ein Zeichen ist umso mehr Bits werden benötigt, um dieses zu kodieren. Als Kodierungsverfahren wird die Huffman-Kodierung verwendet. Alternativ wäre auch eine arithmetische Kodierung möglich, deren Patent liegt aber bei den Firmen IBM, AT&T und Mitsubishi.

Bei der Huffman Kodierung geschieht die Ermittlung der Bitcode für die Symbole mittels eines Baumes der über die Wahrscheinlichkeiten der Symbole aufgebaut wird. Die beiden Symbole mit der geringsten Wahrscheinlichkeit des Auftretens werden zusammengefaßt zu einem neuen Symbol mit der Wahrscheinlichkeit addiert aus den Wahrscheinlichkeiten der beiden Symbole. Die Ausgangssymbole werden als linker und rechter, bzw. 0 und 1 Teilbaum angehängt und danach aus der Liste gestrichen. Dieses wird sooft angewendet, bis nur noch ein Symbol in der Liste ist, welches die Wurzel des Kodierbaumes bildet. Durch aneinander reihen der 0 und 1 der Teilbäume, welche auf dem Weg zwischen Wurzel und Symbol liegen, erhält man die Codes für die einzelnen Symbole.

Da der Baum auch zur Dekodierung benötigt wird, muss er oder eine entsprechende Übersetzungstabelle mit den Daten mitgeliefert werden. Um den Baum aufzubauen müssen die Wahrscheinlichkeiten der Symbole bekannt sein. Da im Falle von Bilddaten diese nicht bekannt sind, sind 2 Durchgänge durch die Daten erforderlich, einen ersten um die Häufigkeiten zu zählen und den Baum aufzubauen und einen zweiten um die eigentliche Kodierung mit diesem Baum vorzunehmen. Um den ersten Durchgang zum Zählen der

Häufigkeiten zu sparen, hat die JPEG-Gruppe mehrere Tabellen zum Kodieren der DC und AC Werte mitgeliefert. Diese Tabellen basieren auf empirischen Erfahrungen und stellen somit keine optimale Kodierung dar. Daher sind diese Tabellen kein Standard im JPEG-Verfahren, sondern nur als günstige Default-Tabellen angeboten, die sich in vielen Fällen bewährt haben.

Die Kodierung an sich läuft nun wie folgt ab. Der DC Wert, der meist mit Abstand der größte Wert ist, wird gesondert behandelt. Da sich die DC Werte von zwei benachbarten Blöcken meist nur geringfügig unterscheiden, wird die Differenz zum vorhergehenden Block kodiert, das heißt, der DC-Wert wird aus seinem Vorgänger vorausgesagt (Bild 2.1 – Prädiktion):

$$diff(k) = DC(k) - DC(k-1)$$

Dabei wird für den ersten Block der DC Wert des vorhergehenden mit 0 initialisiert. Die Dekodierung erfolgt durch Addition der Differenz (Bild 2.1 – DIFF) zum DC Wert des vorhergehenden Blocks:

$$DC(k) = diff(k) + DC(k-1)$$

Die erhaltene Differenz wird in eine Zwischendarstellung, bestehend aus einem Paar von 2 Symbolen (Bild 2.1) transformiert:

DC Wert:

Symbol1	Symbol2
(Anzahl Bits)	(Amplitude)

Symbol1 gibt an aus wie vielen Bits Symbol2 besteht, wobei es Werte von 0 bis 11 annehmen kann. Die Bits von Symbol2 werden als VLI (variable length integer) gespeichert und enthalten den Huffman-Code.

Dabei steckt folgende Idee hinter diesem Verfahren:

Mit n Bits können vorzeichenlos die Zahlen $0 \dots 2^n - 1$ dargestellt werden. Statt dessen wird nun die untere Hälfte der Zahlen $0 \dots 2^{n-1} - 1$ abgebildet auf $-2^{n-1} + 1 \dots -2^{n-1}$ und die restlichen Zahlen $2^{n-1} \dots 2^n - 1$ auf sich selber abgebildet.

Mit $n = 4$ wären die Zahlen $0 \dots 15$ darstellbar. In diesem Fall werden jedoch die Werte $0 \dots 7$ auf $-15 \dots -8$ abgebildet. So sind mit 4 Bit die Zahlen $-15 \dots -8, 8 \dots 15$ darstellbar.

Der Vorteil dieser Abbildung liegt in der Darstellung negativer Zahlen, die bei normalen Zahlen vom Typ Integer die ganzen Bits der Zahl benötigen, da bei betragsmäßig kleinen, negativen Zahlen die höheren Bits alle auf 1 gesetzt sind. Mit dieser Abbildung ist es aber möglich, die kleinen negativen Zahlen mit wenigen Bits darzustellen. Das erste Bit des Symbol2 ist als Vorzeichenbit nutzbar. Ist es 0, so muss der Wert abgezogen werden und die Zahl ist negativ. Ist das Bit 1, so ist die Zahl positiv. Von dieser 2-Symbol Zwischendarstellung wird das Symbol1 wie oben beschrieben Huffman-kodiert, genannt VLC (variable length code), und abgespeichert. Das Symbol2 wird im Gegensatz zu Symbol1 nicht Huffman-kodiert, da die 0 und 1 Bits ungefähr gleichverteilt sind, was eine Verbesserung durch Huffman Kodierung ausschließt. Die Kategorien, in denen die DC-Koeffizienten codiert werden, sind in Bild 2.4 aufgeführt.

Kategorie	DIFF-Wert	Code (Lum)	Code (Chrom)
0	0	00	00
1	-1, 1	010	01
2	-3, -2, 2, 3	011	10
3	-7, ..., -4, 4, ..., 7	100	110
4	-15, ..., -8, 8, ..., 15	101	1110
5	-31, ..., -16, 16, ..., 31	110	11110
6	-63, ..., -32, 32, ..., 63	1110	111110
7	-127, ..., -64, 64, ..., 127	11110	1111110
8	-255, ..., -128, 128, ..., 255	111110	11111110
9	-511, ..., -256, 256, ..., 511	1111110	111111110
10	-1023, ..., -512, 512, ..., 1023	11111110	1111111110
11	-2047, ..., -1024, 1024, ..., 2047	111111110	11111111110

Abbildung 2.4: Kategorien für die Codierung von DC-Koeffizienten und Huffman-Code-Beispiele für die Codierung von Luminanz und Chrominanz

Die AC Werte werden getrennt von den DC Werten behandelt. Nach der DCT sind die niedrigen Frequenzen hervorgetreten, während die hohen Frequenzen Werte zwischen -1 und

1 haben. Diese werden dann durch die Quantisation weitgehend auf 0 geändert. Werden nun beim Komprimieren die niedrigen Frequenzen vor den hohen Frequenzen bearbeitet, so kann eine bessere Kompression erzielt werden, da zum Schluss ausschließlich 0 Werte zu komprimieren sind. Um die niedrigen Frequenzen vor den hohen Frequenzen abzuarbeiten werden die AC-Werte in einer Zick-Zack Reihenfolge (Bild 2.2b) abgetastet. Auch die AC-Werte werden in eine 2-Symbol Zwischendarstellung transformiert. Das Symbol1 hat hier im Gegensatz zum Symbol1 des DC-Wertes eine erweiterte Bedeutung.

AC Werte:

Symbol1	Symbol2
(Lauflänge, Bits)	(Amplitude)

Das Symbol1 beinhaltet somit 2 Informationen. In den ersten 4 Bits wird die Lauflänge dargestellt und in den anderen 4 Bit die Anzahl der Bits für das folgende Symbol2. Die Lauflänge gibt an, wie viele AC-Werte in der Zick-Zack Reihenfolge zwischen dem letzten AC-Wert und dem aktuellen 0 waren. Bedingt durch die serielle Abarbeitung der 63 Werte mittels der Zick-Zack-Folge kann auf die explizite Kodierung von AC-Werten, die gleich 0 sind, verzichtet werden. Beim nächsten AC-Wert der nicht 0 ist wird einfach die Anzahl von 0 Werten zwischen den beiden AC-Werten angegeben. Man spart bei der Komprimierung die Kodierung der 0 ein. Das lohnt sich nur, wenn mehrere 0 Werte hintereinander vorkommen. Das Verfahren heißt Lauflängenkodierung. Es können nun aber mit 4 Bit nur die Zahlen 0...15 dargestellt werden. Trotzdem können mehr als 15 AC-Werte hintereinander 0 sein. Deshalb wurde der Lauflänge 15 eine besondere Bedeutung zugewiesen. Falls die Lauflänge 15 und gleichzeitig die Anzahl der Bits für Symbol 2 gleich 0 ist, dann wird eine Lauflänge von 16 angenommen. Da die Anzahl der Bits für Symbol2 gleich 0 ist, entfällt das Symbol2. Es können daher mehrere (15,0)- Symbol1-Werte hintereinander vorkommen. Die Kodierung der Anzahl der Bits und der Amplitude ist identisch mit der Kodierung beim DC-Wert, die oben beschrieben wurde. Falls alle AC-Werte, die noch in der Zick-Zack-Reihenfolge zu bearbeiten sind den Wert 0 haben, so kann die Bearbeitung des Blockes mit dem Symbol1 (0,0) beendet werden. Diese End-of-Block Markierung setzt alle restlichen Werte auf 0 und bricht den Block ab. Aufgrund der Zick-Zack-Abarbeitung wird die Häufung der 0 Werte ans Ende der Bearbeitungsreihenfolge geschoben. Somit sind oft für einen Block oft nur wenige Bytes zu kodieren. Die Zwischendarstellung muss nun in eine Bitreihenfolge überführt

werden. Dazu werden einfach die Symbole der Zwischendarstellung in Huffman Codes umgewandelt.

2.2.2 MPEG

Die Standards definieren, wie Audio und Videodaten komprimiert und die Datenströme verpackt werden. Die Systemschicht legt dabei die Syntax für die Zeitsteuerung sowie die Synchronisation und Verschachtelung der einzelnen Komponenten fest. Dadurch wird die Kombination von Audio- und Video-Elementarströmen in einem Bitstrom ermöglicht. Bei MPEG-2 ist es zudem möglich, mehrere Systemströme zu Transportströmen zu verbinden. Dabei werden die Daten paketiert und mit 4-Byte-Marken als Startcodes versehen, sodass eine parallele Übertragung von mehreren Videoströmen mit unterschiedlichen Zeitbasen durchführbar ist. Der Systemdecoder analysiert den Bitstrom, erkennt die Pakete anhand der Marken und verteilt sie entsprechend an die Audio- und Videomodule, welche die enthaltenen Daten decodieren.

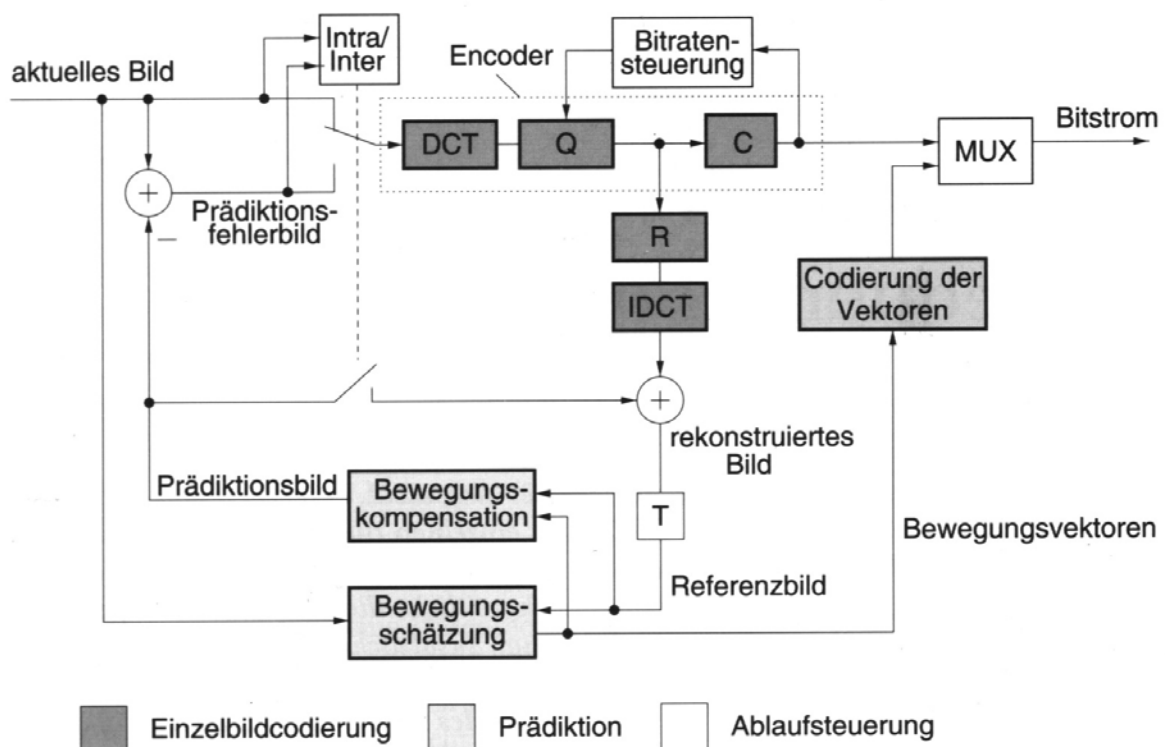


Abbildung 2.5: Blockschaltbild der MPEG-Codierung

Die Sequenzschicht des Video-Elementarstroms wird durch einen Sequenzkopf eingeleitet und mit einem Ende-Code beendet. Eine Sequenz gliedert sich dabei in ein oder mehrere Bildgruppen (GOP- Group of Pictures), die sich aus einer Abfolge von Bildern verschiedenen Typs zusammensetzen:

2.2.2.1 I-Bild (intra-codiertes Bild)

Das I-Bild wird wie ein Einzelbild ohne Informationen von anderen Bildern verarbeitet. Es handelt sich um ein vollständiges Einzelbild, welches, wie in Abbildung 2.5 zu sehen ist, einer Transformation (DCT) und einer Quantisierung (Q) unterzogen wird, bevor es kodiert wird (C). Der Vorgang gleicht damit der JPEG-Kodierung wobei als Quantisierungswert für die DC-Koeffizienten in MPEG-1 8 festgelegt ist, in MPEG-2 auch Werte von 4, 2 und 1 erlaubt sind, um eine höhere Bildqualität zu ermöglichen.

Die Koeffizienten werden mit der Vorschrift

$$q[u, v] = \frac{16F[u, v] + \text{sgn}(F[u, v])q_{scale}Q_{u, v}}{2q_{scale}Q_{u, v}}$$

quantisiert. Die Operationen werden im Festkomma-Format durchgeführt, sodass ein explizites Runden nicht erforderlich ist. Mit der Rekonstruktion

$$[F[u, v]]_Q = \frac{q[u, v]q_{scale}Q_{u, v}}{8}$$

führt dies auf eine gleichmäßige Quantisierung.

Die frei wählbaren Quantisierungswerte $Q_{u, v}$ sind in einer Matrix abgelegt. Der Parameter q_{scale} (1...31) kann die Quantisierungsstärke zusätzlich beeinflussen. Er wird hauptsächlich zur Variation der Kompressionsrate bei der Bitratensteuerung eingesetzt. Die Prädiktionsreihenfolge der DC-Koeffizienten ist durch die Makroblock-Struktur vorgegeben.

Die Codierung der AC-Koeffizienten erfolgt etwas anders als bei JPEG. Die kombinierten Datensymbole setzen sich direkt aus Lauflänge und Koeffizientenwert zusammen. Aufgrund

der sehr großen Zahl aller Kombinationsmöglichkeiten werden nur die häufigsten mit einem Code variabler Länge versehen. Der Rest wird mit Hilfe eines ESCAPE-Symbols übertragen.

2.2.2.2 P- und B-Bilder (prädiktiv-codierte Bilder, bidirektional ~)

Zwei aufeinanderfolgende Bilder sind nur in den seltensten Fällen identisch. Durch Licht- bzw. Farbveränderungen oder durch Bewegungen unterscheiden sich die Bilder. Das quantisierte Vorgängerbild wird rekonstruiert (Bild 2.5 – R) und rücktransformiert (Bild 2.5 – IDCT) und mit dem aktuellen Bild verglichen. Mit Hilfe der Bewegungskompensation kann auf Basis des Block-Matching-Verfahrens mit einer maximalen Genauigkeit von halben Bildpunkten ein Bewegungsvektor ermittelt werden, der die Änderungen zwischen zwei Bildern repräsentiert. Auf diese Weise muss nicht mehr ein vollständiges Bild sondern nur noch die Unterschiede zwischen den Bildern gespeichert und kodiert werden. Mit Hilfe des Prinzips der Bitmarkierung werden diejenigen 8x8 Blöcke (Luminanz und Chrominanz) eines Makroblocks gekennzeichnet, deren Quantisierungssymbole alle gleich Null sind. In einem solchen Fall ist der betreffende Block bei der Codierung auszulassen. Wenn alle Blöcke des Makroblocks diese Eigenschaft aufweisen, kann der gesamte Makroblock übersprungen werden. Der Decoder füllt den betreffenden Bereich des rekonstruierten Bildes einfach mit einer Kopie des Referenzblocks.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
a) 27	29	35	38	46	56	69	83

16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
b) 16	16	16	16	16	16	16	16	16

Abbildung 2.6 Matrizen mit Luminanz-Quantisierungswerten für Intra- und Intercodierung

Der Encoder entscheidet auf Makroblock-Ebene, ob der Block inter-codiert (bei erfolgreicher Prädiktion durch die Bewegungskompensation) oder intra-codiert wird (z.B. bei nicht voraussagbaren Bildinhalten). Die Intra-Codierung von Makroblöcken erfolgt analog zur Codierung von I-Bildern, bei der Inter-Codierung wird die Quantisierung der 8x8-Blöcke modifiziert. Die Berechnung der Quantisierungssymbole erfolgt durch

$$q[u, v] = \frac{F[u, v]}{q_{scale} Q_{u, v}}$$

mit implizitem Abrunden der Ergebnisse. Die Rekonstruktionsvorschrift lautet

$$[F[u, v]]_Q = \frac{(2q[u, v] + \text{sgn}(q[u, v]))q_{scale} Q_{u, v}}{16}$$

Dies entspricht einer gleichmäßigen Quantisierung mit Totzone. Da in Prädiktionsfehler-signalen alle Frequenzanteile gleich wichtig sind, werden alle DCT-Koeffizienten mit demselben Wert $Q_{u, v} = 16$ quantisiert.

Infolge der Bewegungsprädiktion bei inter-codierten Blöcken sind die DC-Koeffizienten nicht mehr miteinander korreliert, so dass sie mit den AC-Koeffizienten zusammen verarbeitet werden.

Bei signifikanten Bildunterschieden muss noch ein Fehlerbild codiert werden.

Bei B-Bildern kommt eine bidirektionale Prädiktion zum Einsatz, die sowohl vorangegangene als auch nachfolgende I- oder P-Bilder einbezieht. Dabei sind die Kompressionsraten bei B-Bildern allgemein am größten, allerdings ist der Aufwand zur Bewegungsschätzung etwa doppelt so hoch.

Die Anzahl der Bilder pro Gruppe und der Abstand von I- und P-Bildern können frei gewählt werden. Bei der Festlegung von B-Bildern ist zu beachten, dass sie alle zwischengespeichert werden müssen, bis das nachfolgende P-Bild übertragen wurde. Neben erhöhtem Speicheraufwand bedeutet dies auch eine Zeitverzögerung. Des weiteren verschlechtern sich

die Prädiktionmöglichkeiten für die P-Bilder, da der zeitliche Abstand und damit die Unterschiede zum Referenzbild größer werden.

2.2.3 H.263

H.263 ist eine Weiterentwicklung des H.261-Videokodierungsstandards, welcher 1990 durch ITU (International Telecom Union) veröffentlicht wurde.

H.261 wurde für Videoübertragungen per ISDN entwickelt und war deshalb für Datenraten von 64 Kbit/s und Mehrfachen davon ausgelegt. Das Grundprinzip der Videokomprimierung entspricht dem des MPEG-Verfahrens, welches ebenfalls eine Weiterentwicklung von H.261 ist.

H.263 beinhaltet einige Verbesserungen bzw. Änderungen gegenüber H.261, um einerseits die Performance zum anderen aber auch die Fehlerkorrektur zu verbessern. Hinzu kommt, dass die Limitierung der Datenrate aufgehoben wurde, so dass sowohl niedrigere als auch höhere Datenraten möglich sind. Des weiteren wurde die Genauigkeit der Bewegungskompensation von 1 Pixel auf 0.5 Pixel verbessert und die aus MPEG bekannten B-Bilder eingeführt.

H.263 unterstützt 5 Auflösungen: SQCIF (128x96), QCIF (176x144), CIF (352x288), 4CIF(704x576) und 16CIF (1408x1152).

2.2.4 Vergleich der Verfahren

Der Hauptunterschied der Verfahren besteht in der Verarbeitung der Einzelbilder. Während M-JPEG jedes Bild auf die gleiche Weise codiert, entstehen sowohl bei H.263 als auch bei MPEG Bildgruppen aus vollständigem Bild (I-Bild) und Bildern mit Informationen über Unterschiede zum Vorgängerbild (P- und B-Bild). Dadurch lassen sich wesentlich höhere Kompressionsraten erreichen. Andererseits ist mit der Bewegungskompensation ein höherer Aufwand für die Kodierung verbunden. Bei Videos mit schnellen bzw. vielen Bewegungen erhöht sich damit der Aufwand, was je nach gewünschter Qualität zu stärkeren Bildfehlern oder zu erhöhtem Rechenaufwand und damit verbunden zu zeitlichen Verzögerungen bei der

Übertragung führen kann. Hinzu kommt eine erhöhte Anfälligkeit bei Übertragungsfehlern. Bei M-JPEG wirkt sich ein Fehler in einem Bild nur auf dieses aus und ist im günstigsten Fall nicht zu sehen, bei H.263 und MPEG pflanzt sich ein Fehler im I-Bild auf alle folgenden P- und B-Bilder fort.

Es lässt sich daher nicht von vornherein eines der Verfahren als günstig für die Videokommunikation festlegen. Vielmehr müssen sie anhand vorhandener Systeme auf ihre Tauglichkeit verglichen werden.

2.3 Codecs

Ein Codec ist ein Algorithmus zur Kompression (COmpressor) und Dekompression (DECompressor) von Audio- und/oder Videodaten. Codecs können dabei rein softwarebasiert aufgebaut sein oder auch eine spezifische Video-Hardware verwenden. Durch Verwendung einer Hardware wird der Prozessor weniger belastet, allerdings setzt die Verwendung eines Hardware-Codecs voraus, dass auf der Empfängerseite ebenfalls die selbe Hardware vorhanden ist. Hier sind Software-Codecs flexibler, da sie nur auf den Prozessor zurückgreifen, damit aber auch das System stärker belasten.

Der Codec-Algorithmus beeinflusst ganz wesentlich die visuelle Qualität einer Video-Datei und die Geschwindigkeit, mit der sie am Monitor des Computers oder an einem TV-Bildschirm wiedergegeben wird.

Man unterscheidet zwischen drei verschiedenen Codec-Typen:

- reine Software-CODECs, wie Cinepack, VIDEO 1, INDEO INTERACTIV,
- CODECs, die eine Hardware-Unterstützung benötigen, wie Motion JPEG
- CODECs, die sowohl software-mäßig als auch mit Hardware-Unterstützung lauten, wie INDEO 3.1, INDEO 3.2, MPEG.

2.4 Datenübermittlung

2.4.1 Übertragungswege

Der Übertragungsweg hängt in erster Linie von den Teilnehmern ab, zwischen denen die Daten ausgetauscht werden soll. Teilnehmer, deren Rechner an ein LAN angeschlossen ist, werden über das lokale Netzwerk kommunizieren. Dabei sind, je nach Netzwerk, Geschwindigkeiten zwischen 10 und 1000 MBit/s möglich. Anders sieht es aus, wenn die Teilnehmer nicht im gleichen Netzwerk sind und ihre Verbindung direkt oder über ein WAN (z.B. Internet) aufbauen müssen. Hier hängt die zu erreichende Geschwindigkeit nicht nur von den Teilnehmern selbst sondern auch von eventuell zwischengeschalteten Knoten ab. Bild 2.7 enthält ein paar Beispiele für maximal erreichbare Geschwindigkeiten.

Medium	Upstream	Downstream
Modem V.90	33 kbit/s	56 kbit/s
Modem V.92	48 kbit/s	56 kbit/s
ISDN	64 Kbit/s	64 Kbit/s
ISDN-Kanalbündelung	128 Kbit/s	128 Kbit/s
T-DSL	768 Kbit/s	128 Kbit/s
ADSL	1,5 – 8 Mbit/s	1,5 – 8 Mbit/s

Bild 2.7: Erreichbare Geschwindigkeiten für verschiedene Medien

Allerdings sind die angegebenen Geschwindigkeiten theoretische Maximalwerte. In der Praxis liegen die tatsächlich erreichten darunter. Dies liegt zum einen an Störungen während der Übertragung zum anderen an Begrenzungen der Übertragungsgeschwindigkeit zwischen Knoten. Insbesondere bei Verbindungen nach Übersee kann es so leicht zu Engpässen kommen. Die Daten werden in Pakete aufgeteilt abgeschickt. Bei einer hohen Auslastung eines Knotens kann es passieren, dass aufgrund der vergebenen Zeitscheiben ein Paket nur mit Verzögerung weitergereicht wird. Kommt das Paket fehlerhaft an, muss es noch einmal geschickt werden, was zu einer weiteren Verzögerung führt.

Gerade aber bei AV-Stream-Daten ist es wichtig, dass diese verzögerungsfrei übertragen werden. Deshalb ist es wichtig, die AV-Daten auf die erreichbare Übertragungsgeschwindigkeit anzupassen.

2.4.2 Das RTP-Protokoll

2.4.2.1 Grundlagen

An die Übertragung von Daten mit „Echzeit-Charakteristik“, wie z.B. AV-Streams für Videokommunikation, sind strenge Bedingungen geknüpft. Die Übertragung muss nahezu verzögerungsfrei erfolgen, um Pausen oder Verzögerungen zu verhindern. Für die Übertragung solcher Daten wurde das RTP-Protokoll (Real-Time Transfer Protokoll) entwickelt.

Dieses stellt folgende Funktionen zur Verfügung:

- Daten-Typ-Identifikation: Informiert Applikationen, die das Protokoll verwenden über den Typ der übermittelten Informationen (Audio, Video, etc)
- Sequenz: Die zu übermittelnden Pakete werden mit einer Paketnummer versehen, um sie beim Empfänger wieder in der richtigen Reihenfolge zusammensetzen, da die Pakete in falscher Reihenfolge aufgrund unterschiedlicher Routen vom Sender zum Empfänger ankommen können.
- Zeitstempel: In Applikationen wie AV-Konferenzen ist es wichtig, die beteiligten Geräte zu synchronisieren, damit keine Informationen verloren gehen. Der Zeitstempel wird ebenfalls für überwachende Applikationen benötigt, damit diese die überwachten Vorgänge ordnungsgemäß aufzeichnen.
- Zustell-Überwachung: Ermöglicht den beteiligten Applikationen Informationen über die Performance der Datenübermittlung zu sammeln und Statistiken auszutauschen.
- Informationen über die Verbindung: Andere wichtige Informationen, z.B. Namen der Beteiligten oder Informationen über Ein- und Ausstieg Beteiligter, lassen sich austauschen.

Die letzten beiden Funktionen werden durch das RTCP-Protokoll bereitgestellt, welches das Kontroll-Protokoll für RTP ist.

Nicht bereit gestellt werden:

- Qualität des Services: RTP gehört zu den unteren Schichten der Kommunikationsprotokolle und ist nur zuständig für die in-time Zustellung der Pakete und für die Reservierung der entsprechenden Bandweite.
- Zuverlässigkeit der Paket-Zustellung: Es wird nicht geprüft, ob Pakete nicht oder falsch geliefert wurden. Diese Funktion muss von einem Protokoll einer höheren Ebene übernommen werden.
- Sicherheit: Verschlüsselungsmechanismen müssen in anderen Schichten des Kommunikations-Stacks implementiert werden. Jedoch lassen sich RTP-Pakete leicht verschlüsseln.

RTP ist in der Lage Multicasting-Mechanismen des verwendeten Netzwerkes zu nutzen um Eins-zu-viele-Verbindungen zu ermöglichen. Üblicherweise greift RTP auf UDP zurück und nutzt dessen Multiplexing- und Prüfsummen-Service. Es lassen sich jedoch auch adäquate Protokolle unterer Ebenen nutzen.

Für eine Übertragung mittels RTP werden mindestens zwei Ports benötigt. Einer für die Übermittlung der RTCP-Pakete der andere für die RTP-Pakete. Werden verschiedene Datenströme (z.B. Audio- und Video-Daten) übermittelt, ist pro Datenstrom ein Port nötig. Dies ermöglicht es dem Empfänger zu selektieren, welche Daten er empfangen möchte. Beteiligte mit geringer Bandweite haben so die Möglichkeit, nur die Audiodaten zu empfangen und damit Bandweite zu sparen.

Periodisch wird durch jeden Beteiligten ein Kontroll-Paket mit Informationen über Verbindung und Qualität des Empfangs verteilt. Bei Kommunikationsproblemen ist es so möglich, die Parameter neu auszuhandeln. Diese Möglichkeit kann auch genutzt werden, wenn sich ein neuer Teilnehmer mit begrenzter Bandbreite einschaltet.

So bald ein Teilnehmer die Verbindung abbricht, schickt er eine spezielle „Abmelde-Meldung“ durch den Kontroll-Kanal.

2.4.2.2 Aufbau des RTP-Paketes

Der Header des RTP-Paketes hat folgendes Format [Sch96, Seite 10]:

V	P	X	CC	M	PT	Sequence Number
Timestamp						
Synchronisation Source (SSRC) ID						
Contributing Source (CSRC) ID ...						

Bild 2.7: RTP-Header

- V: Version des Protokolls. Zur Zeit wird die Version 2 des Protokolls verwendet. (RFC 1889)
- P: Padding-Bit (Füll-Bit). Wenn dieses Bit gesetzt ist, enthält das Paket Füll-Bytes am Ende der Nutzdaten. Padding ist optional und kann benutzt werden, um das Paket auf eine, von einem tiefer liegenden Protokoll (z.B. Verschlüsselung) geforderte Größe zu bringen. Das letzte Byte des Paketes gibt die Anzahl der Padding-Bytes an.
- X: Extension-Bit (Erweiterungs-Bit). Ist dieses Bit gesetzt, folgt auf den Header eine 32-bit Erweiterung. Diese kann benutzt werden, um die Funktionalität der Applikationen zu erweitern.
- CC: CSRC (Contributing source)-Count. Standard Null. Werden mehrere RTP-Streams durch einen Mixer zusammengefasst, fügt der Mixer in den Header Informationen über die Originalquellen ein. Im CC-Feld steht dann die Anzahl der zusammengefassten Quellen.
- M: Marker-Bit (Markierungs-Bit). Die Bedeutung und Verwendung des Bits wird durch die Applikation festgelegt. Zum Beispiel kann damit das Startpaket für einen Rahmen markiert werden.
- PT: Payload-Type (Nutzdaten-Typ). Spezifiziert die Art der Informationen in den Nutzdaten.
- Sequence number: Sequenz-Nummer. Wird mit jedem gesendeten Paket erhöht. Der Empfänger kann damit die Originalreihenfolge der Pakete wieder herstellen bzw. feststellen, ob ein Paket verloren gegangen ist. Die Startnummer wird zufällig vergeben,

um eine Entschlüsselung schwieriger zu gestalten, sollte der Datenstrom verschlüsselt werden.

- **Timestamp:** Zeitstempel. Dieses Feld speichert den Samplingzeitpunkt des ersten Bytes der Paketdaten. Dieses Feld wird genutzt, um verschiedene Datenströme zu synchronisieren, Jitter zu behandeln und die Statistik über die Qualität des Services zu ermitteln. Die Timestamp-Frequenz ist abhängig vom Datentyp (z.B. Samplingrate bei Audiocodierern)
- **SSRC (Synchronization source):** Diese Nummer identifiziert die Quelle des RTP-Paketes. Sie wird zufällig gewählt und muss einmalig sein, damit alle Datenströme beim Empfänger ordnungsgemäß zugeordnet und dekodiert werden können. Sobald ein Knoten eine SSRC festgelegt hat, überprüft er ihre Einmaligkeit über RTCP.
- **CSRC list:** Wenn der CSRC-Zähler größer als Null ist, werden hier die Identifikationen der Verschiedenen Quellen aufgelistet. Es können bis zu 15 Identifikationen aufgelistet werden. Gibt es mehr als 15 Quellen, werden nur die ersten 15 aufgelistet.

2.4.2.3 Multiplexing

Wenn gleichzeitig verschiedene Arten von Informationen in einer RTP-Sitzung übertragen werden, werden die Datenströme auf mehrere Verbindungen mit unterschiedlichen Ports aufgeteilt.

Theoretisch wäre es auch möglich, alle Datenströme über den selben RTP-Strom zu übertragen, da in jedem Paket die Art der Informationen gekennzeichnet ist. Dies führt jedoch zu folgenden Problemen:

- Wenn die Aufnahme-Raten der verschiedenen Medien nicht gleich sind, führt dies zu Problemen in der Handhabung von Sequenznummern und Zeitstempeln.
- Die errechnete Performance-Statistik würde inkorrekt ermittelt werden.
- Die Informationen aller Pakete müssten vom Empfänger gelesen werden. Wenn einer der Empfänger aufgrund geringer Bandbreite entschieden hat, nur Audio-Daten zu empfangen, würde er trotzdem auch die Videodaten geliefert bekommen. Damit würden die empfangenen Videodaten beim Empfänger gelöscht werden und so der Übertragungskanal uneffektiv genutzt werden.

- Die Arbeit eines Mixers würde erschwert werden.

2.4.2.4 Mixer und Übersetzer

Beide sind in der RTP-Protokoll-Definition spezifiziert.

Mixer dienen dazu, mehrere Datenströme gleichen Typs zusammenzufassen. Dabei müssen neue Zeitstempel und Synchronisationsinformationen generiert werden. Bei Bedarf kann das Format der Informationen geändert werden. Mixer werden z.B. eingesetzt, wenn ein Knoten über eine wesentlich geringere Bandbreite kommuniziert als alle anderen beteiligten Knoten. In diesem Fall kann der Mixer die Datenströme der schnelleren Knoten zusammenfassen und als einen Strom an den langsameren Knoten senden.

Übersetzer ändern nur das Format der Daten. Zeitstempel und Synchronisationsdaten bleiben unbeeinflusst. Übersetzer können z. B. eingesetzt werden, wenn Netzwerke und Transport-Schichten der beteiligten nicht zusammenpassen (z.B. UDP/IP und SPX/IPX).

Neben den Daten müssen beide auch Kontrollinformationen ändern, damit Empfänger die Informationen richtig interpretieren und Statistiken korrekt erstellt werden können. Deshalb müssen auch Änderungen am RTCP-Strom durchgeführt werden.

2.4.2.5 Zur Zeit definierte Daten-Typen

Audio:

- 1016 CODED AUDIO
- DVI 4
- G.721
- G.722
- G.728
- GSM

- L8
- L16
- LPC
- MPA
- PCMA
- PCMU
- VDVI

Video:

- CELB
- JPEG
- H.261
- H.263
- MPV
- MP2T
- BT.656

2.4.2.6 Das RTCP-Protokoll

Das RTCP (RTP Control Protocol) dient zum Austausch von Informationen über die Verbindung wie Statistiken, Benutzerinformationen und Kontrollinformationen zwischen den beteiligten Knoten. Der Austausch der Informationen erfolgt dabei über einen eigenen Kanal.

Die Funktionen von RTCP:

- Versorgung der Teilnehmer mit Informationen über die Qualität des Dienstes. Im Falle von Stauungen oder Verbindungsfehlern können die Teilnehmer so andere Datenkodierverfahren vereinbaren, die die Bandbreite effizienter nutzen oder problematische Knoten ausschließen. Bei einer Multicast-Verbindung kann der Netzwerkadministrator so Statistiken über den Service sammeln.
- Transport einer festen Identifikation für die Datenströme, auch canonical name genannt. Da die SSRC von Sitzung zu Sitzung unterschiedlich ist bzw. sich bei

Konflikten ändern kann, wird der canonical name verwendet, damit die Nutzer übereinander auf dem laufenden bleiben und die Datenströme korrekt zugeordnet und synchronisiert werden können. Das Standardformat des canonical name ist „user@host“ (bzw nur „host“ bei einem Einbenutzer-System).

- Ermöglicht den Teilnehmern zu wissen, wie viele Nutzer in der Kommunikationsgruppe sind, damit Anpassungen an der Kontrolldaten-Rate vorgenommen werden können.
- Bietet ein Minimum an Kontrollinformationen, um zum Beispiel mitzuteilen, wenn ein Nutzer der Gruppe beitrifft oder diese verlässt.

Es gibt 5 verschiedene Typen von RTCP-Paketen:

1. Der Senderbericht (Sender-Report)

Dieses Paket wird von allen Stationen, die mindestens ein Daten-Paket seit dem letzten Report verschickt haben, versandt.

Das Paket enthält Informationen über den Sender und alle ihm bekannten Stationen (CSRCs). Die Station informiert auf diesem Weg über die eigene Leistung und die Statistiken, die sie von den Teilnehmern gesammelt hat.

Es sind zwei Zeitstempel und die Anzahl der übertragenen Pakete und Bytes enthalten. Der eine Zeitstempel ist ein absoluter, vom Netzwerk-Zeit-Protokoll vergebener, der andere ein zu den, in den Datenpaketen verwendeten Zeitstempeln relativer.

Außerdem wird für jede bekannte Station Informationen über die Verlustrate an Paketen seit dem letzten Report, die Gesamtzahl der verlorenen Pakete, die höchste erhaltene Paket-Nummer, eine Schätzung über die Varianz der Paketankunftszeiten, der Zeitstempel des letzten Reports und die Zeit seit dem letzten Report übermittelt.

2. Empfängerbericht (Receiver-Report)

Dieses wird von allen Stationen, die seit Empfang des letzten Reports keine Daten gesendet haben, übermittelt. Es hat das gleiche Format wie das Sender-Report-Paket, jedoch ohne die Sender-Informationen.

3. Quellenbeschreibende Pakete (Source Description)

Dieses Paket wird von allen Teilnehmern versandt, um eine Bindung zwischen den SSRC-Identifikationsnummern und den canonical name herzustellen.

Das Format hängt davon ab, welcher Typ Teilnehmer es sendet. Bei einem Endsystem enthält es nur Informationen über das System, bei einem Mixer Informationen über den Mixer selbst sowie über alle beitragenden Quellen.

Zusätzlich kann das Paket noch folgende Informationen enthalten:

- Den realen Namen des Teilnehmers
- Seine/ihre eMail-Adresse
- Die Telefonnummer
- Den geographischen Standort
- Den Namen der verwendeten Applikation
- Außergewöhnliche Informationen über den Status des Benutzers
- Private Erweiterungen

Diese Informationen gehören zum Protokoll-Standard.

4. Ende-der-Teilnahme-Paket (BYE)

Mit diesem Paket wird mitgeteilt, dass eine oder mehrere Quellen nicht mehr aktiv sein werden.

Bei einer einzelnen Quelle wird nur ein Paket mit der Identifikation, bei einem Mixer zusätzlich die Identifikationen aller beteiligten Quellen geschickt.

Zusätzlich existiert ein Feld, in dem der Grund für die Abmeldung mitgesandt werden kann.

5. Anwendungsspezifische Paket

Dieser Typ Paket ist für experimentelle Zwecke. Er kann bei der Entwicklung einer neuen Anwendung verwendet werden, ohne dass ein neuer Paket-Typ beantragt werden muss.

Die einzelnen Pakete wurden so entworfen, dass sie zusammengefasst und als ein Paket übertragen werden können. Es gibt kein Limit für die Anzahl der kombinierten Pakete. Allerdings müssen folgende Bedingungen erfüllt sein:

- Jedes kombinierte Paket muss ein Report-Paket enthalten, da diese so oft wie möglich versandt werden sollten, um die Granularität der Statistiken zu verbessern.
- Jedes Paket muss ein SDES-Paket mit dem kanonischen Namen des Nutzers enthalten. Dies ermöglicht neuen Teilnehmern schnellstmöglich Informationen zu erhalten, wer verbunden ist und die verschiedenen Datenströme zu synchronisieren.

Die Pakete müssen in folgender Reihenfolge angeordnet werden: Bei Verschlüsselung des Paketes wird ein bestimmtes Präfix vorangestellt. Danach folgt der obligatorische Report sowie weitere Empfängerberichte. Als nächstes folgt ein SDES-Paket, welches mindestens den canonical name enthält. Als letztes können BYE- und anwendungsspezifische Pakete folgen. Die Anzahl der Paket-Typen im kombinierten Paket sollten allerdings begrenzt werden, um Übertragungsfehler in den Paketen zu vermeiden.

2.4.2.7 Der RTCP-Header

Der Header des RTP-Paketes hat folgendes Format [Sch96, Seite 31]:

V	P	RC, SC oder Subtype	Packet-Type	Length
Synchronisation Source (SSRC) ID				

Bild 2.8 : RTCP-Header

- V: Version des Protokolls. Zur Zeit wird Version 2 verwendet.

- P: Padding-Bit (Füll-Bit): Zeigt an, ob am Ende Füll-Bytes angehängen wurden. Das letzte Byte gibt die Anzahl der Füll-Bytes an.
- Reception Report Count (RC), Source Count (SC) oder Subtype: Gibt die im Paket enthaltenen Sender- oder Empfängerberichte (RC) bzw. SSRC-Bezeichner (SC; bei SDES- oder BYE-Paketen) an. Bei Anwendungsspezifischen Paketen kann es zur Unterscheidung unterschiedlicher Pakete verwendet werden.
- Packet-Type: Zeigt den Pakettyp an.
- Length: Zeigt die Länge des Paketes an. Es lassen sich einzelne Pakete mit Hilfe dieser Information miteinander verbinden.
- SSRC-ID: Beinhaltet den Bezeichner für den Sender des Paketes.

2.4.2.8 Bandbreiten-Management

Das Problem beim Bandbreiten-Management liegt in der Natur von Multimedia-Informationen.

Bei Applikationen, die nur Audiodaten übertragen (z.B. Audio-Konferenz) ist die Bandbreite nahezu konstant, unabhängig von der Anzahl der Teilnehmer. Bei Übermittlung von Videoinformationen werden weit höhere Bandbreiten benötigt. Bei einer Videokonferenz müsste jeder Teilnehmer seine Video-Informationen verbreiten. In diesem Fall empfiehlt es sich einen Mixer einzusetzen, um die Videoströme aller Nutzer zu einem zusammenzufassen.

Hinzu kommt, dass auch die Kontrollströme Bandbreite in Anspruch nehmen. Da jede beteiligte Station RTCP-Pakete verbreiten muss, kann es bei vielen Teilnehmern dazu führen, dass ein Großteil der Bandbreite für die Übermittlung der Kontrollpakete nötig ist.

Deshalb ist es nötig, den Verkehr jeder RTP-Sitzung durch eine konstante Sitzungs-Bandbreite, die sowohl die Bandbreite des Daten- als auch die des Kontroll-Kanals enthält, zu begrenzen. Die Bandbreite des Datenkanals kann dabei durch die Kodierung des Datenstroms geregelt werden. Für den Kontroll-Kanal lässt sich eine Regelung über die Versand-Rate der Pakete erreichen. Da jeder Knoten über die Anzahl der Teilnehmer informiert ist, kann der RTP-Algorithmus die Rate für den Versand einstellen. Es ist dabei notwendig, die Bandbreite des Kontrollkanals auf 5% der Sitzungsbandbreite zu beschränken.

Neben den Statistiken, verbrauchen vor allem die Quell-Beschreibungs-Pakete viel Bandbreite. Deshalb enthalten diese Pakete meist nur den canonical name des Nutzers.

3. Untersuchung der verwendeten Codecs

3.1 Versuchsdurchführung

Für die Durchführung der Versuche wurde ein Pentium 4 mit 1.600 MHz, 256 MB RAM und einer Asus V7700 GeForce2 Grafikkarte mit 32 MB RAM verwendet. Als Betriebssysteme wurden Windows 98, Windows 2000 mit Servicepack 3 und Windows XP installiert.

Für die Videoübertragung kam das Java Media-Framework-Paket 2.1.1 [Sun02] von Sun zum Einsatz. Dieses stellt alle notwendigen Grundlagen für die Entwicklung plattformunabhängiger Videokommunikations-Software bereit. Die Kodierung der Videodaten kann wahlweise per H263- oder MJPEG-Codec erfolgen. Zwar besteht die Möglichkeit, MPEG-kodierte Video-Daten per RTP-Protokoll zu übermitteln, da JMF jedoch keinen MPEG-Codec beinhaltet und die Videodaten deshalb vorkodiert sein müssen, wurde auf eine Untersuchung verzichtet, da somit keine gleichen Bedingungen gewährt wären.

Als Sender kam ein Beispiel-Programm von Sun zum Einsatz. Zwar stellt JMF eine Funktion zur Verfügung, mit der für den Ausgabestrom eine Datenrate festgelegt werden kann, allerdings scheint diese nicht zu funktionieren, da Vorgaben keinerlei Einfluss auf Bildqualität oder Datenrate zeigten. Deshalb wurde alternativ der Ausgabestrom des Codecs beeinflusst. Im Fall des H263-Codecs konnten Datenraten zwischen 5.000 und 1.000.000 Bit/s eingestellt werden. Für den MJPEG-Codec ließen sich keine Datenraten festlegen. Deshalb musste hier auf die Festlegung der Qualität zurückgegriffen werden.

Um einen Vergleich zu ermöglichen, diente als Grundlage der Versuche ein unkomprimiertes 30 sekündiges Video mit einer Auflösung von 352x288 und einer Farbtiefe von 16 bit. Dieses musste vom Sender einmal in einer Auflösung von 352x288 (CIF) und einmal in einer Auflösung von 176x144(QCIF) übermittelt werden. Am Ende jeder Übertragung gab der Sender die Anzahl der übertragenen Bytes aus. Auf diese Weise war ein Vergleich beider Codecs möglich. Sowohl Sender als auch Empfänger (hier kam das Programm JMStudio, welches mit dem JMF-Paket mitgeliefert wird, zum Einsatz) hatten einen Kontroll-Monitor zur Anzeige des Videos. Somit konnten eventuell auftretende zeitliche Verzögerungen erkannt werden. Zusätzlich hatte der Empfänger eine Anzeige für die Framerate. Sender und Empfänger befanden sich auf dem selben PC. Die Übertragung erfolgte über das Loop-Back-

Device des Rechners. Unter Windows 2000 und Windows XP wurde auf die Möglichkeit zurück gegriffen, die CPU-Lastung vom System anzeigen zu lassen.

3.2 H263-Codec

Für diesen Codec wurden für jede Auflösung folgende Ausgaberraten festgelegt: 64.000 Bit/s, 128.000 Bit/s, 768.000 Bit/s und 1.000.000 Bit/s.

	352x288				176x144			
	64.000	128.000	768.000	1.000.000	64.000	128.000	768.000	1.000.000
1	568.209	583.980	1.986.870	2.012.812	245.845	487.433	1.241.299	1.241.299
2	565.399	581.241	1.992.047	2.010.447	245.085	487.808	1.243.976	1.243.976
3	565.598	581.075	1.981.379	2.010.050	245.085	487.808	1.243.976	1.243.976
4	565.635	581.342	1.981.379	2.010.050	245.085	487.808	1.243.976	1.243.976
5	565.635	581.342	1.978.679	2.010.050	245.085	487.808	1.243.976	1.243.976
Ø	566.095	581.796	1.984.071	2.010.682	245.237	487.733	1.243.441	1.243.441

Bild 3.1: Übertragene Bytes beim H263-Codec in Abhängigkeit von Auflösung und Ausgabe-Datenrate

Der H263-Codec liefert keine konstanten Datenraten (Bild 3.1). Bei den jeweils 5 Testläufen variierten die übertragenen Bytes geringfügig. Es waren jedoch keine sichtbaren Unterschiede festzustellen. Für die weiteren Betrachtungen wird deshalb der Durchschnittswert betrachtet.

Bit/s	Übertragungsrate		Windows 98		Windows 2000		Windows XP	
	Bytes	Bit/s	fps	fps	CPU	fps	CPU	
64.000	245.237	65.397	21.0 - 25.0	25.0	39% - 41%	25.0	38% - 40%	
128.000	487.733	130.062	21.0 - 25.0	25.0	38% - 42%	25.0	37% - 40%	
768.000	1.243.441	331.584	21.0 - 25.0	25.0	40% - 61%	25.0	39% - 42%	
1.000.000	1.243.441	331.584	21.0 - 25.0	25.0	40% - 72%	25.0	41% - 44%	

Bild 3.2: Ergebnisse des H263-Codex bei einer Auflösung von 176x144

Unabhängig von der gewählten Ausgaberrate gibt es keinerlei Probleme, das Video mit 25 Bildern pro Sekunde zu übertragen. Betrachtet man die erreichten Datenraten (Bild 3.2), so zeigt sich, dass das Video sogar problemlos mit einfacher ISDN-Geschwindigkeit übertragen werden kann. Allerdings nur mit sehr schlechter Qualität, wie auf Bild A.2 zu sehen ist. Deutlich sind die Artefakte auszumachen. Aufgrund der Bildgröße führt dies dazu, dass kaum noch etwas zu erkennen ist. Bei doppelter Datenrate (Bild A.3) dagegen wären die Ergebnisse durchaus annehmbar. Qualitativ nehmen sich höhere Geschwindigkeiten (Bild A.4 und A.5) kaum, allerdings benötigt ein Ausgabestrom von 128kBit/s die komplette Bandbreite bei

ISDN mit Kanalbündelung. Erlaubt man größere Datenraten, so zeigt sich, dass es eine Obergrenze gibt, welche bei 331.584 liegt. Allerdings führen Datenraten größer als 128 kBit/s zu keinen erkennbaren Qualitätsverbesserungen.

Bit/s	Übertragungsrate		Windows 98	Windows 2000		Windows XP	
	Bytes	Bit/s	fps	fps	CPU	Fps	CPU
64.000	566.095	150.959	12.1 - 15.8	13.5 - 17.5	80% - 91%	14.5 - 17.5	70% - 93%
128.000	581.796	155.146	12.3 - 15.2	16.5 - 17.0	80% - 90%	14.5 - 17.5	70% - 93%
768.000	1.984.071	529.086	8.3 - 12.8	10.5 - 14.5	75% - 81%	10.0 - 15.0	70% - 93%
1.000.000	2.010.682	536.182	8.8 - 12.3	10.5 - 14.5	79% - 84%	10.0 - 15.0	70% - 93%

Bild 3.3: Ergebnisse des H263-Codecs bei einer Auflösung von 352x288

Betrachtet man zum Vergleich die erzielten Ergebnisse bei doppelter Auflösung, so stellt man fest, dass sich die Qualität zwischen 64.000 Bit/s und 128.000Bit/s kaum unterscheidet, beide haben deutlich sichtbare Artefakte. Auch bei höheren Datenraten sind diese noch schwach zu erkennen. Allerdings ist das Ergebnis hier akzeptabel. Zudem haben sich die Datenraten deutlich erhöht, wie in Bild 3.3 zu erkennen ist. Eine Übertragung mit ISDN-Geschwindigkeit in Echtzeit wäre nicht mehr möglich. Hier ist mindestens DSL-Geschwindigkeit notwendig, um die Daten in Echtzeit zu empfangen oder zu senden. Gleichzeitiges Senden und Empfangen wäre nur mit Übertragungsgeschwindigkeiten ab 1MBit/s möglich.

Die niedrigere Framerate ist zu erkennen, das Video läuft sichtbar langsamer ab. Zudem ist eine geringe Verzögerung zu erkennen, wenn es im Video zu Szenenwechseln kommt. Alles in allem sind die Ergebnisse bei Datenraten von 768 KBit/s und höher jedoch akzeptabel.

3.3 M-JPEG-Codec

Da der M-JPEG-Codec keine Vorgabe für Datenraten unterstützt, wurden die Ergebnisse bei 5 unterschiedlichen Qualitätsstufen (1%, 25%, 50%, 75% und 100%) ermittelt. Zwar lässt sich auch eine Qualität von 0% vorgeben, allerdings kommt es dann zu extremen Farbfehlern, so dass das Bild nicht mehr erkennbar ist.

Im Gegensatz zum H263-Codec liefert der M-JPEG-Codec konstante Datenraten.

Bit/s	Übertragungsrate		Windows 98	Windows 2000		Windows XP	
	Bytes	Bit/s	fps	fps	CPU	fps	CPU
1%	1.082.031	288.542	24.5 - 25.0	25.0	13% - 22%	25.0	19% - 27%
25%	1.853.513	494.270	24.5 - 25.0	25.0	37% - 39%	25.0	29% - 31%
50%	2.896.429	772.381	24.5 - 25.0	25.0	37% - 39%	25.0	19% - 34%
75%	7.194.700	1.918.587	23.1 - 24.6	25.0	37% - 39%	25.0	38% - 41%
100%	28.185.847	7.516.226	10.1 - 11.5	16.5 - 17.0	55% - 64%	15.0 - 17.0	54% - 64%

Bild 3.4: Ergebnisse des M-JPEG-Codecs bei einer Auflösung von 176x144

Bild 3.4 zeigt deutlich, dass die Übertragungsraten weit über denen des H263-Codecs liegen. Vergleicht man die Bilder A.6 bis A.10, so stellt man fest, dass ab einer Qualität von 50% das Ergebnis zufriedenstellend ist. Die daraus resultierende Datenrate macht allerdings deutlich, dass mindestens DSL-Geschwindigkeit für die Übertragung notwendig ist. Höhere Qualitätsstufen bringen bei dieser Bildgröße keine Verbesserungen, so dass man auf sie, auch in Anbetracht der Übertragungsraten, verzichten kann und sollte.

Bit/s	Übertragungsrate		Windows 98	Windows 2000		Windows XP	
	Bytes	Bit/s	fps	fps	CPU	fps	CPU
1%	3.908.767	1.042.338	15.6 - 19.7	25.0	59% - 64%	24.0 - 25.0	60% - 75%
25%	5.151.112	1.373.630	16.1 - 21.2	24.5 - 25.0	61% - 75%	23.0 - 24.5	60% - 75%
50%	8.647.177	2.305.914	14.2 - 18.2	22.0 - 23.5	63% - 79%	19.5 - 22.0	60% - 75%
75%	28.184.359	7.515.829	4.4 - 10.1	13.0 - 14.0	63% - 68%	13.0 - 14.5	60% - 75%
100%	107.163.000	28.576.800	0.0 - 3.5	4.0 - 5.0	53% - 62%	4.5 - 5.0	60% - 75%

Bild 3.5: Ergebnisse des M-JPEG-Codecs bei einer Auflösung von 352x288

Vergleicht man Bild 3.3 und Bild 3.5 so zeigt sich, dass der M-JPEG-Codec auch betriebssystem-abhängig ist. Bei der Auflösung von 352x288 zeigen sich deutliche Unterschiede zwischen Windows 98 und Windows 2000 bzw. Windows XP. Auch bei der kleineren Auflösung werden keine konstanten 25 fps erreicht. Die Ressourcen-Verwaltung unter den auf Windows NT basierenden Betriebssystemen scheint besser zu sein. Während beide auch bei einer Qualität von 75% noch akzeptable Framerates erzielen, kommt es bei Windows 98 zu starken Verzögerungen. Stellenweise erhält man nur noch ein Standbild. Besonders extrem zeigte es sich bei der höchsten Qualitätsstufe, bei der das Bild sekundenlang stehen blieb. Auch wenn die Bildqualität ab 75% an die des Originalbildes heranreicht, sollte man angesichts der Datenraten auch hier maximal eine Qualität von 50% verwenden, da auch diese noch bessere Ergebnisse als der H263-Codec liefert.

4. Schlußbetrachtung

Vergleicht man beide Codecs, so zeigt sich, dass das M-JPEG-Verfahren qualitativ bessere Ergebnisse liefert. Allerdings sind die Datenraten im Vergleich bis zu 50 mal größer bei annähernd gleicher Qualität. Damit ist dieses Verfahren für den Einsatz im privaten Bereich eher ungeeignet, da hier vor allem Verbindungen über Modem, ISDN bzw. DSL vorherrschen. Auch für den Einsatz bei Videokonferenzen sollte eher auf den H263-Codec zurück gegriffen werden, da hier abhängig von der Anzahl der Teilnehmer große Datenmengen auftreten können. Eine ausreichend große Bandbreite vorausgesetzt, eignet sich die M-JPEG-Komprimierung allerdings für einseitige Übertragungen zum Beispiel von Live-Veranstaltungen (Konzerte, Vorlesungen) bzw. Filmen. Mit ihr können die Videodaten in einer, für diese Zwecke, guten Qualität übertragen werden.

Da kein MPEG-Codec zur Verfügung stand, lässt sich hier kein Vergleich ziehen. Geht man allerdings davon aus, dass das MPEG-Verfahren kleinere Datenmengen als M-JPEG bei gleichzeitig besserer Qualität als H263 liefert, so könnte MPEG eine gute Alternative darstellen. Hier sind jedoch die Hardware-Voraussetzungen zu prüfen, die höher als die der untersuchten Verfahren liegen dürften. Von IBM wird ein auf Java basierender MPEG-4-Codec angeboten, der aus lizenzrechtlichen Gründen jedoch nur MPEG4-Daten dekomprimieren kann. Dieser kann damit zumindest auf Empfängerseite eingesetzt werden, da er das RTP-Protokoll unterstützt.

Die durchgeführten Versuche haben gezeigt, dass das von Sun angebotene JMF-Paket durchaus für die Entwicklung professioneller Software im Videokommunikationsbereich verwendbar ist. Das relativ einfach gehaltene Sender-Programm belegte zwischen 24 und 27 MB an Speicher, ein gleichzeitiges Arbeiten von Sender und Empfänger war auf dem eingesetzten System problemlos möglich.

Anhang A - Screenshots



Bild A.1: unkomprimiertes Original, Auflösung 176x144

H263



Bild A.2: 64.000 Bit/s



Bild A.3: 128.000 Bit/s



Bild A.4: 768.000 Bit/s



Bild A.5: 1.000.000 Bit/s

M-JPEG



Bild A.6: Qualität 1%



Bild A.7: Qualität 25%



Bild A.8: Qualität 50%



Bild A.9: Qualität 75%



Bild A.10: Qualität 100%



Bild A.11: unkomprimiertes Original, 352x288



Bild A.12: H263, 64.000 Bit/s



Bild A.13: H263, 128.000 Bit/s



Bild A.14: H263, 768.000 Bit/s



Bild A.15: H263, 1.000.000 Bit/s



Bild A.16: M-JPEG, Qualität 1%



Bild A.17: M-JPEG, Qualität 25%



Bild A.18: M-JPEG, Qualität 50%



Bild A.19: M-JPEG, Qualität 75%



Bild A.20: M-JPEG, Qualität 100%

Anhang B – Glossar

Artefakte: Bildfehler, die bei verlustbehafteter Kompression entstehen. Bei JPEG- und MPEG-Kompression treten sie in Form von blockartigen Bildfehlern auf.

CIF: Common Intermediate Format. Teil des H.261-Standards mit einer Bildgröße von 352x288 Pixeln. Weiterhin gibt es QCIF (Quarter CIF) mit 176x144, SQCIF (Sub-Quarter CIF) mit 128x96, 4CIF mit 704x576 und 16CIF mit 1408x1152.

Frame-Rate: Abspielgeschwindigkeit einer Animation bzw. eines Videos. Die Angabe erfolgt in Frames pro Sekunde (fps). Beim PAL-System werden Filme mit 25 fps abgespielt.

Jitter: Verzerrung eines Signals aufgrund schlechter Synchronisation. Bei Übertragung von RTP-Paketen z.B. durch unterschiedliche Laufzeiten der einzelnen Pakete.

JPEG: Abkürzung für "joint photographic expert group". Das Expertengremium erarbeitet seit 1988 international gültige Standards auf ISO und ITU- Basis für JPEG und JBIG. Der offizielle Titel dieses Gremiums lautet ISO/IEC JTC1 SC29 Working Group 1.

Das Bildformat bietet ähnliche Merkmale wie GIF- Bilder, kann aber bis zu 16,7 Mio. Farben darstellen und unterliegt keinen Copyright-Merkmalen. Die JPEG-Kompression beschränkt sich nicht auf das Packen von Daten nach den üblichen Algorithmen, sondern beinhaltet Verfahren, die selektiv einzelne Bildinformationen löschen, was bei sehr hohen Kompressionen zu Artefakten führen kann. Die Kompressionsrate ist in der Regel einstellbar.

LAN: Abkürzung für "Local Area Network" (lokal angelegtes Netzwerk). Im Gegensatz zu WAN, das überregional das Arbeitsstationen und Netzwerke verbindet. "Lokal" bezieht sich in diesem Sinne auf einen gemeinsamen Standort, wie beispielsweise ein Firmengelände oder einen Raum.

Makroblock: Makroblöcke setzen sich aus je vier 8x8-Luminanzblöcken und zwei 8x8-Chrominanzblöcken (MPEG-1) bzw. 2, 4 oder 8 Chrominanzblöcken (MPEG-2) zusammen.

M-JPEG: Abkürzung für "Motion-Joint Picture Expert Group". Dateiformat und hardwareabhängiges Kompressionsverfahren, in dem jedes Einzelbild einer Videosequenz vollständig - und zwar für sich selbst JPEG-komprimiert - abgespeichert wird. Dieser Standard ist nicht hundertprozentig genormt, so dass in der Regel M-JPEG-komprimierte Dateien meist nicht auf der Hardware anderer Hersteller abgespielt werden können.

MPEG: Abkürzung für "motion pictures expert group" (www.mpeg.org). Von dieser Gruppe wurden bzw. werden Dateiformate und Verfahren zum platzsparenden Komprimieren und Speichern von Video- bzw. Multimediadaten (Video, Bild- und Tondaten) in hoher Qualität festgelegt. Der MPEG-Standard unterteilt sich inzwischen in MPEG-1, MPEG-2, MPEG-3 und MPEG-4, wobei der MPEG-3-Standard mittlerweile in MPEG-2 integriert wurde.

PAL: Abkürzung für "Phase Alternation Line" (zeilenweise Phasenänderung). Norm für Farbfernsehern; deutsche Fernsehnorm mit einer Auflösung von 768 x 576 und einer Framerate von 25 fps bzw. 50 Halbbilder pro Sekunde

Quantisierung mit Totzone: Bei der gleichmäßigen Quantisierung wird der Wertebereich des Signals in Intervalle gleicher Breite unterteilt. Sind kleine Signalamplituden irrelevant, so wird das mittlere Intervall aufgeweitet. Es entsteht eine Totzone (engl.: deadzone) in der alle Amplituden zu 0 quantisiert werden.

UDP: „Users Datagram Protocol“. Kommunikationsprotokoll, welches über das Internet Protocol kommuniziert. Beim Versand der Pakete wird nicht geprüft, ob diese das Ziel erreichen. Damit ist das Protokoll einerseits fehleranfällig andererseits schneller als Protokolle mit Fehlerprüfung. Damit ist eine flüssigere Datenübermittlung möglich, wie sie zum Beispiel bei Live-Streams nötig ist.

WAN: Abkürzung für "Wide Area Network". Netzwerk, das weltumspannend angelegt sein kann, im Gegensatz zum LAN-Netzwerk.. Beim Internet handelt es sich im Prinzip um ein WAN.

Anhang C- Literaturverzeichnis

- [Che96] Dr. Peter Cherriman, H.261 Video Coding,
<http://www-mobile.ecs.soton.ac.uk/peter/h261/h261.html>, 1996
- [Che99] Dr. Peter Cherriman, H.263 Video Coding,
<http://www-mobile.ecs.soton.ac.uk/peter/h263/h263.html>, 1999
- [Lam95] Frank Lamers, JPEG-Decoder als virtuelles Videogerät,
<http://www.ti5.tu-harburg.de/Staff/Lamers/stab/stab.html>, 1995
- [Mad00] Juan M. Madrid, EECS 845 Final Project - Multimedia Information Transport over the Internet – The RTP-Protocol,
<http://www.ittc.ku.edu/~jmadrid/845proj/index.html>, 2000
- [Sch96] H. Schulzrinne, RFC 1889 - RTP: A Transport Protocol for Real-Time Applications, <http://rfc.sunsite.dk/rfc/rfc1889.html>, 1996
- [Str00] Tilo Strutz, Bilddatenkompression, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 2000
- [Sun02] Sun Microsystems, Java™ Media Framework 2.1.1,
<http://java.sun.com/products/java-media/jmf/2.1.1/index.html>, 2002