

Bildunterschriften zu Videokompression

- **Folie 3:** Bewegtbilder, z.B. Filme, bestehen aus aufeinander folgenden Einzelbildern. Diese Bilder weisen typischerweise starke Ähnlichkeiten zu einander auf. Unbewegte Bildteile, z.B. der blaue Himmel, bleiben über viele Bilder hinweg unverändert (mit blauen Pfeilen markiert), wohingegen bewegte Bildteile oft nur von Bild zu Bild örtlich versetzt werden. Neu erscheinende Bildteile sind – abgesehen von wenigen Ausnahmen wie Szenenwechseln – eher selten.
- **Folie 5:** Um einen Block (rot umrahmt) in Bild N zu kodieren, wird im vorangegangenen Bild $N - 1$ rund um dieselbe Position (dünn rot strichliert angedeutet) ein Suchbereich definiert (dick rot strichliert), in dem alle möglichen Blöcke (Versätze) auf Ähnlichkeit zum Block in Bild N verglichen werden. Der (zweidimensionale) Versatz zwischen der Position des am besten übereinstimmenden Blockes (blau umrahmt) und der des Originalblockes in Bild N wird als Bewegungsvektor (als violetter Pfeil angedeutet) bezeichnet.
- **Folie 10:** Ein typischer H.264-Encoder unterteilt jedes zu kodierende Bild in $16 \cdot 16$ Pixel große Blöcke, die jeweils zuerst differenzkodiert und anschließend – analog zu JPEG – transformiert, quantisiert und entropiekodiert werden. Die Differenzkodierung (blaues Plus links oben) greift dabei wahlweise auf bereits kodierte Daten aus demselben Bild (Intraprediktion) oder aus vorangegangenen Bildern (Interprediktion über Motion Compensation) zurück. Um die bereits kodierte Daten zur Prediktion zu erhalten, werden die transformierten und quantisierten Koeffizienten invers quantisiert, rücktransformiert, invers differenzkodiert und mit einem Filter geglättet, um die gleichen Daten (mit Kompressionsartefakten) zu erhalten, die auch im Decoder verfügbar sind und dort bei der Dekodierung dargestellt werden. Jeder H.264-Encoder enthält daher typischerweise einen Decoder (grau eingefärbter Bereich), der die Basisdaten für die Differenzkodierung anhand der bisher komprimierten Daten liefert. Über den gesamten Kodierprozess wacht die Datenraten- und Qualitätssteuerung (*coder control*), die die Quantisierung und die Wahl der Differenzkodierung beeinflusst sowie notwendige Steuerdaten für den Decoder in den kodierten Datenstrom schreibt.
- **Folie 11:** Ein P-Bild (zweites Bild von links) verwendet Bildteile vorangegangener Bilder, z.B. die drei Kreise (strichliert umrahmt) aus dem benachbarten I-Bild. Ein B-Bild verwendet Bildteile von vorangegangenen und nachfolgenden Bildern, z.B. die drei Kreise aus dem vorangegangenen P-Bild und den einen zusätzlichen Kreis aus dem nachfolgenden I-Bild. Die Verwendung von Bildteilen als Ausgangspunkt für die Kodierung wird als Prediktion bezeichnet und mit (im Beispiel orangen) Pfeilen vom Bezugspunkt zum Verwendungsort illustriert.

- **Folie 12:** Da eine Prädiktion mit Daten aus zukünftigen Bildern nicht praktikabel ist, werden die Bilder zur Kodierung umgeordnet. Beispielhaft ist eine IBBbP-Bildstruktur dargestellt, bei der zuerst das I-Bild (schwarz) und danach das P-Bild (weiß) kodiert wird. Die eigentlich dazwischen liegenden B-Bilder (grau) werden erst nach dem P-Bild kodiert, da letzteres zur Prädiktion der B-Bilder genutzt wird. Das dritte und letzte B-Bild der Struktur wird mit einem Kleinbuchstaben (b) bezeichnet, da es selbst nicht (wieder) zur Prädiktion verwendet wird. Die insgesamt aus fünf Bildern bestehende Prädiktionsstruktur wiederholt sich für die nachfolgenden vier Bilder (drei B-Bilder und ein P-Bild) analog, wobei das erste P-Bild bereits kodiert wurde und daher nicht erneut kodiert werden muss. Die Änderungen in der Bildabfolge sind nur für die Kodierung notwendig und haben keinen Einfluss auf eine spätere Wiedergabe – bei der Dekodierung muss die ursprüngliche Reihenfolge, auch Darstellungsreihenfolge genannt, wiederhergestellt werden, damit die Bewegtbilder wieder in der richtigen Abfolge angezeigt werden.
- **Folie 13:** $16 \cdot 16$ Pixel große Makroblöcke können in H.264 entweder direkt verarbeitet oder wahlweise in zwei $16 \cdot 8$, zwei $8 \cdot 16$ oder vier $8 \cdot 8$ Pixel große Partitionen unterteilt werden. Jede $8 \cdot 8$ Pixel große Makroblockpartition kann wiederum wahlweise in zwei $8 \cdot 4$, zwei $4 \cdot 8$ oder vier $4 \cdot 4$ Pixel große Subpartitionen unterteilt werden.
- **Folie 15:** Bei der Intraprädiktion nach H.264 wird ein $4 \cdot 4$ Pixel großer Block (grau dargestellt) nicht wie in JPEG unabhängig kodiert, sondern als Differenz zu geschätzten Pixelwerten, die aus den Nachbarpixeln extrapoliert werden. Wird beispielsweise ein Block mit horizontalem Helligkeitsverlauf kodiert, können die Helligkeitswerte des linken Nachbarblockes zeilenweise übernommen werden (Modus *horizontal*) – alle Pixel der ersten Blockzeile werden mit der gleichen Helligkeit angenommen wie der angrenzende Pixel in der ersten Zeile des linken Nachbarblockes (I), alle Pixel der zweiten Blockzeile mit der Helligkeit der zweiten Zeile des linken Nachbarblockes (J) u.s.w. Die Helligkeit des Blockes kann dann als Differenz zu den extrapolierten Daten kodiert werden. H.264 erlaubt insgesamt neun mögliche Extrapolationsvarianten (0-8), die je nach Charakteristik des zu kodierenden Blockes gewählt werden können. Die Wahl des Verfahrens muss im Datenstrom gespeichert werden, damit der Decoder die Basis für die Differenzberechnung gleich wie der Encoder berechnen kann.
- **Folie 16:** Der zu kodierende Block (links weiß eingerahmt) wird aus seinen Nachbarpixeln mit den verschiedenen Extrapolationsvarianten (0-8) prädiziert (rechts). Durch das Ausprobieren aller Varianten kann jene gewählt werden, die beispielsweise die kleinste (Helligkeits-)Differenz erzeugt (7, rot umrahmt).
- **Folie 17:** Zusätzlich zur Intraprädiktion von $4 \cdot 4$ Pixel großen Blöcken erlaubt H.264 eine reduzierte Anzahl von Extrapolationsvarianten für $16 \cdot 16$

Pixel große Blöcke (oben). Diese erlauben die vergleichsweise effizientere Kodierung großflächigerer Helligkeitsverläufe. Wie bei der größeren Anzahl von Prädiktionsvarianten bei $4 \cdot 4$ Pixel großen Blöcken können die vier Extrapolationsvarianten für $16 \cdot 16$ Pixel große Blöcke für jeden Block (links unten weiß abgegrenzt) ausprobiert werden (rechts unten), um jene Variante zu finden, die die kleinste Differenz (rot umrahmt) bzw. die effizienteste Kodierung erlaubt.

- **Folie 19:** Durch die Quantisierung entstandene Blockartefakte (links) können nach der Rekonstruktion, d.h. nach der inversen Quantisierung, Transformation und Differenzkodierung, durch ein adaptives Tiefpassfilter nahezu entfernt werden (rechts). Dieses bei H.264 als Deblocking-Filter bezeichnete Kodierwerkzeug führt durch die Glättung von Kanten zu Unschärfeartefakten, die subjektiv meist als qualitativ weniger qualitätsmindernd wahrgenommen werden als die ursprünglichen Blockartefakte.