Appunti sulla codifica MPEG-4

Appunti dalle lezioni dei corsi di
''Comunicazioni Elettriche'' e di *''Elaborazione Numerica dei Segnali''*,
tenute presso l'Università di Roma TRE nell'a.a. 1999-2000 dai Proff. A.Neri e G.Giunta.

Roma, marzo 2000

Introduzione

L'MPEG-4 supporta le seguenti funzionalità:

- **Interattività basata sul contenuto:** riguarda l'interazione tra l'utente e i dati
  - Strumenti per l'accesso basato sul contenuto ai dati multimediali
  - Manipolazione del bitstream basata sul contenuto
  - Codifica ibrida naturale e sintetica
  - Accesso casuale ai dati

  La struttura dell'MPEG-4 supporta la composizione di scene ibride, contenente oggetti sia naturali che sintetici. Ogni singolo oggetto può essere diviso in sotto-oggetti i quali a loro volta possono essere riordinati per formare un oggetto composto.
  Ogni oggetto può essere manipolato dall'utente in ogni sua caratteristica (colore, forma, dimensione ... etc) specificando le proprietà della trasformazione. Gli oggetti comprendono dati video e dati audio indipendenti tra di loro in modo da permettere la manipolazione da parte dell'utente. La composizione gerarchica di oggetti è supportata tramite multiplazione gerarchica del bitstream.

- **Compressione dati**: riguarda l'uso di metodi efficienti per l'immagazzinamento e la trasmissione di dati audiovisivi
  - Miglioramento dell'efficienza di codifica
  - Codifica di più flussi di dati concorrenti

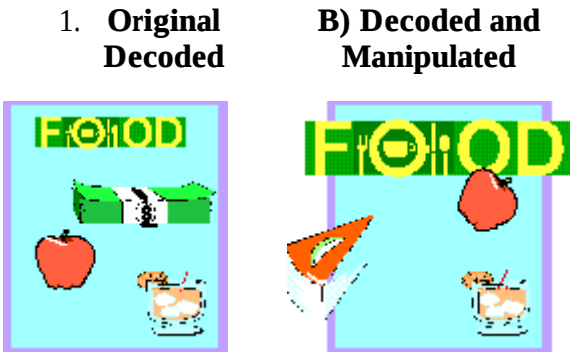
  Ogni singola immagine può essere suddivisa in frammenti che possono essere traslati e posizionati per ricreare l'immagine; questi frammenti sono funzioni particolari come la DCT oppure blocchi parzialmente o completamente ricostruiti. In questo modo un oggetto anche complesso può essere descritto efficacemente da linee di codice.

- **Accesso universale:** rende i dati codificati accessibili a decodificatori di diversa qualità
  - Robustezza agli errori
  - Scalabilità basata sul contenuto

  La qualità dell'immagine può essere suddivisa su diversi livelli, costruiti assegnando una priorità ad ogni oggetto ed effettuando un'interpretazione a partire dagli oggetti più importanti a quelli meno importanti. Questa suddivisione viene definita scalabilità della banda e della risoluzione. Questo metodo permette ai decodificatori di diversa qualità di visualizzare gli stessi dati.

Lo standard MPEG-4

Lo standard MPEG4 è stato sviluppato per permettere la manipolazione del video utilizzando la codifica ad oggetti con lo scopo di codificare sia dati immagine e sia dati audio in un modo altamente flessibile.
Lo standard video MPEG4 introduce il concetto di VOP (Video Objet Planes) , in cui ogni parte della sequenza video di ingresso viene divisa in un numero di regioni con immagini di forma arbitraria. Ogni regione può coprire una parte dell'immagine o il contenuto d'interesse ossia definire fisicamente oggetti o contenuti all'interno della scena, in cui l'informazione video non è più vincolata ad essere di forma rettangolare ma può rappresentare ad esempio una persona sola e non un'intera scena. Quindi il singolo oggetto audiovisivo può essere sia sintetico che naturale.



Esempio di manipolazione di oggetti in MPEG4 di una sequenza immagine.

- **Video**: Lo scopo dell'uso dell'MPEG4 nelle applicazioni immagini/video (vedi figura) è quello di codificare la sequenza in modo tale da permettere all'utente di separare la decodifica dalla ricostruzione degli oggetti, al fine di rendere la scena iniziale rappresentabile e decodificabile in un modo più flessibile. Lo standard video di codifica MPEG4 produce una stringa di bit per ogni oggetto, tutte organizzate secondo uno schema a strati. Ogni oggetto è codificato secondo una stringa di bit diversi. La forma e la trasparenza dell'oggetto, così come le coordinate spaziali ed i parametri aggiuntivi che descrivono le dimensioni e la posizione, quali lo zoom, la rotazione, la traslazioni o simili, sono incluse nella stringa di bit. L'utente può ricostruire la sequenza originale decodificando tutti gli "strati dell'oggetto" e visualizzando gli oggetti con dimensioni e posizioni iniziali , oppure può modificare la sequenza immagine con delle semplici operazioni. Per esempio nella figura alcuni oggetti non sono stati decodificati ed usati per la ricostruzione, mentre altri sono stati decodificati e visualizzati usando lo "scaling", la traslazione e la rotazione. Queste ultime proprietà possono essere modificate direttamente nella stringa di bit con semplici operazioni di manipolazione, senza il bisogno di un'ulteriore codifica. In più possono essere aggiunti nuovi oggetti che non appartengono alla scena iniziale oppure trascurare alcuni oggetti; ciò è possibile con l'aggiunta o l'eliminazione di stringhe di bit nella

forma a strati con la quale è codificata la sequenza senza ulteriore codifica.

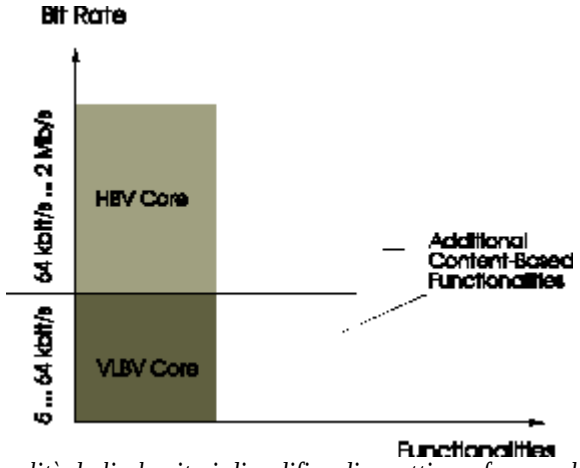
- **Audio**: Si utilizza lo stesso schema per l'immagine ma in questo caso le stringhe di bit rappresentano dati audio.

- **SNHC**: (Syntetic Natural Hybrid Coding) Ogni singolo oggetto può essere rappresentato da oggetti elementari; con l'SNHC si vogliono trasferire le proprietà di ogni oggetto ai suoi singoli componenti. Ciò permette di ridurre il numero di bit necessari a memorizzare e trasmettere un oggetto.

- **Sistemi**: L'architettura dell'MPEG4 permette la codifica separata di oggetti audio e video, reali o artificiali, e una consona multiplazione di ogni singolo oggetto elementare produce un'unica stringa di bit. Come per l'MPEG1 e l'MPEG2, i sistemi standard dell'MPEG4 sono stati sviluppati per la multiplazione dei flussi elementari, per la sincronizzazione e la compressione. Nei sistemi dell'MPEG4 la multiplazione inserisce all'inizio di ogni bitstream rappresentante di un oggetto i parametri base di rappresentazione e manipolazione.

Standard video dell'MPEG-4

Gli algoritmi di codifica dell'MPEG4 comprendono le funzionalità dell'MPEG1 e dell'MPEG2, una classificazione del bit rates e delle funzionalità attualmente fornite dall'MPEG per modelli di codifica video sono rappresentati nella seguente figura



Funzionalità degli algoritmi di codifica di oggetti con forme arbitrarie.

- In basso " VLBV Core" (Very Low Bit Rates Video) rappresenta algoritmi e proprietà per applicazioni che lavorano con un bit rates tra i 5..64 Kbits/s, supportando sequenze di immagini con una bassa risoluzione spaziale ed un basso frame rate (0..15 Hz). Le funzionalità specifiche delle applicazioni base supportate dal VLBL Core comprendono:
  - Una codifica VLBV di sequenze di immagini di forma rettangolare con un'alta efficienza di codifica ed un'alta robustezza nei confronti dell'errore, con bassa complessità per le applicazioni di comunicazioni in tempo reale.
  - Predisposizione per accessi casuali e per operazioni fast/forward e fast/reverse per la memorizzazione di basi dati multimediali.

- In alto “HBV Core” (Higher Bit Rates Video) Supporta le stesse funzioni base del VLBV con un più alto intervallo di parametri di ingresso spaziali e temporali, utilizzando gli stessi algoritmi e le stesse proprietà del VLBV Core.

Video Verification Model

Il gruppo video dell’MPEG stabilisce i modelli di prova o modelli di verifica per uniformare tecniche di codifica di immagini e video.
Il modello di verifica dell’MPEG4 (VM) usa quattro differenti classi per definire la sintassi di una sequenza video:

- Video Session (VS) compone le sequenze video incorporando oggetti delle altre tre classi.
- Video Object (VO) è un oggetto all’interno di una scena
- Video Object Layer(VOL) utilizzata per esaltare la risoluzione spaziale e temporale di ciascun VO
- Video Object Plane (VOP) è un’occorrenza di un VO ad un certo istante. Due diversi VOP possono appartenere allo stesso oggetto ma in istanti diversi, senza dover essere due oggetti separati

(Una VS contiene uno o più VO, ciascuno dei quali possiede uno o più VOL, ed ogni VOL e costituito da una sequenza di VOP)

Il modello di verifica descrive un insieme di algoritmi e di proprietà per la codifica e per la decodifica, le sue caratteristiche principali sono:

- Rappresentazione basata sul contenuto dei dati video: ogni sequenza video è scomposta in VO (video object), ognuno dei quali ha sue particolari proprietà quali la forma il movimento e la "tessitura"(ciò che è all'interno dei confini della forma). In questo modo si permette all'utente la manipolazione degli oggetti interagendo su quattro livelli:
  - Codifica : l'utente può decidere la distribuzione del bit rate disponibile tra i vari VO.
  - Multiplazione : l'utente può alterare la descrizione della scena.
  - Demultiplazione : l'utente può richiedere la ricezione di solo una parte del *bitstream.*
  - Decodifica : l'utente può agire sulla composizione degli oggetti.
- Ogni VO è codificato e trasmesso al decodificatore; dopo la decodifica, l’oggetto viene rappresentato con un insieme di componenti YUV, che ne rappresentano la luminanza (Y) e la crominanza (UV) di ogni pixel, e con delle informazioni sulla forma. Lo standard YUV prevede il campionamento secondo il formato 4:2:0 ed ogni campione utilizza 8 bit per la codifica.
- Codifica della forma di ciascun VOP utilizzando una codifica binaria e una basata sulla scala dei grigi.
- Supporto nella codifica VOP delle immagini di tipo Intra(I), Predette (P) ed Interpolate (B).
- Possibilità di *frame rates* fissi e variabili a seconda dell'applicazione.
- Compensazione e stima del movimento mediante la suddivisione dei VOP in blocchi 8x8 pixel o in macroblocchi 16x16 pixel (riduzione della ridondanza temporale).
- Codifica della "tessitura" secondo gli standard I,B,P usando una DCT su blocchi 8x8 . (riduzione della ridondanza spaziale).
- Predizione efficiente dei coefficienti DC ed AC della DCT.
- Scalabilità temporale e spaziale dei VOP.
- Compatibilità minore con algoritmi di codifica che usano una struttura rettangolare per i VOP.

Architettura MPEG-4

Per trasmettere una sequenza di immagini devo inviare dal codificatore al decodificare i VO. Dal lato del codificatore i VO vengono compressi, protetti dagli errori e multiplati; dal lato del codificatore i VO sono demultiplati, corretti dagli errori di trasmissione , decompressi , composti e presentati all'utente che ha sempre la possibilità di manipolare la sequenza agendo sulle proprietà dei VO. Prima di trasmettere i VO il codificatore ed il decodificatore devono scambiarsi i dati sulle configurazioni ossia il codificatore deve inviare informazioni sulle proprietà e sugli algoritmi al decodificatore per permettergli di elaborare i VO. Tali proprietà possono essere a priori memorizzate nel decodificatore (classi standard), oppure aggiunte dall'utente o scaricate dalla rete.

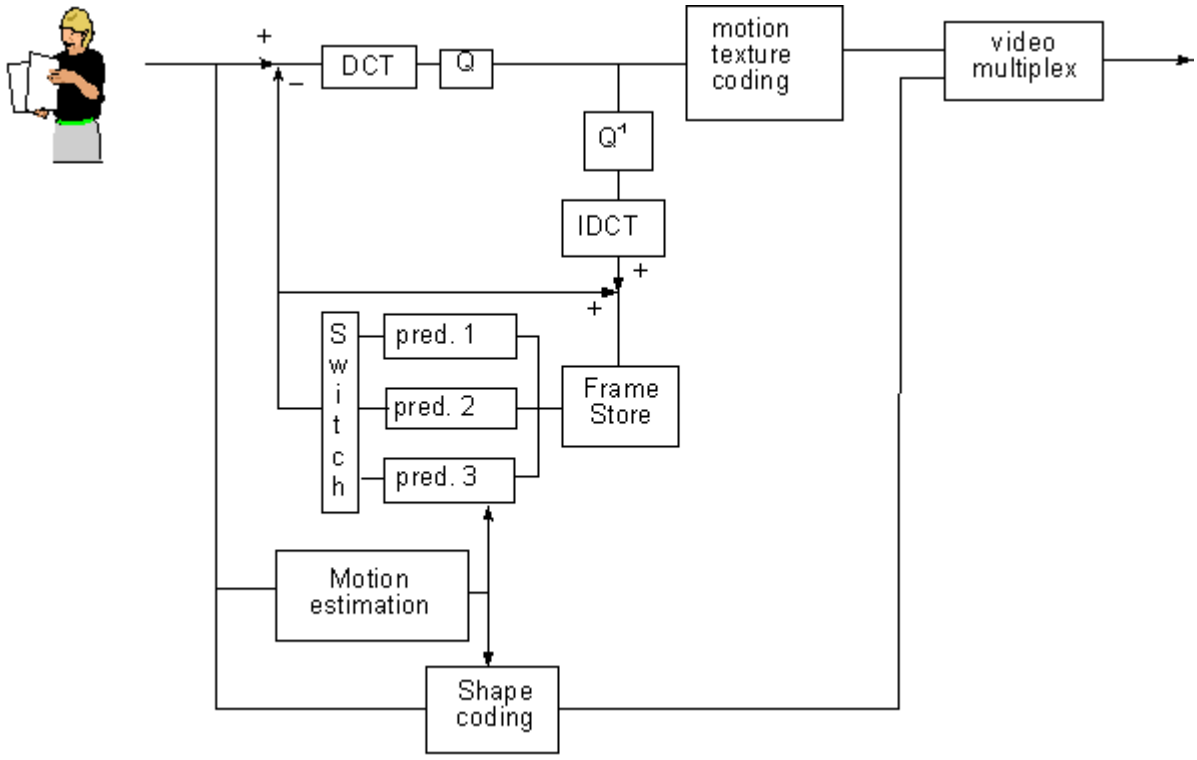
Ogni VO trasmesso è composto da due parti: un'intestazione, che definisce la classe a cui appartiene l'oggetto, e un corpo che contiene i dati dell'oggetto. Se il VO da trasmettere appartiene ad una classe già nota al decodificatore l'intestazione non è necessaria; in questo modo non è necessario trasmettere più volte l'intestazione di oggetti che appartengono ad una stessa classe. Il corpo è necessario solo per oggetti naturali, mentre per quelli sintetici si trasmettono gli elementi base che ricompongono l'oggetto (ad esempio per un cerchio è sufficiente inviare i dati del centro e del raggio per ricostruirlo).

Dal lato del decodificatore i VO vengono proiettati sulla scena su uno o più piani di proiezione.

I VO sono gerarchici, possono essere costituiti da altri VO , detti "primitivi", ed allora l'oggetto si chiama "composito". Una scena dell'MPEG-4 è dunque strutturata secondo una configurazione ad albero in cui la radice è lo sfondo, mentre le foglie sono i VO. Tale struttura può non essere statica e variare a seconda dello sfondo.

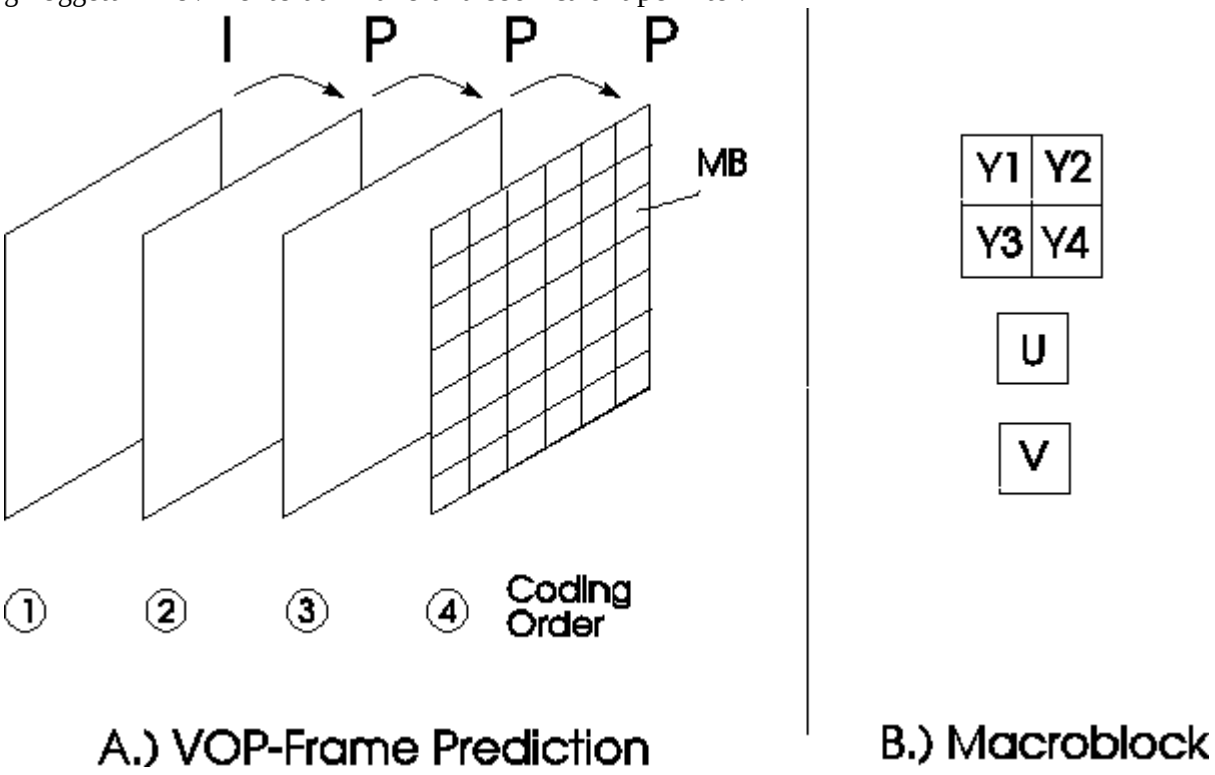
Di conseguenza anche il *bitstream* e la composizione della scena sono strutturate in maniera gerarchica.

Dopo aver ricomposto la sequenza video i *frame* vengono inviati ad un sistema di presentazione ed immagazzinati in un Buffer e quindi inviati all'utente tramite dispositivi di uscita. Il sistema di presentazione deve provvedere ad adattare la risoluzione ed il formato cromatico dei *frame* ai dispositivi di uscita (scalabilità temporale e spaziale). In questo modo la composizione dei VO è indipendente dai sistemi di uscita e ciò permette un accesso universale ai *bitstream* da parte dell'utente.



Codifica dei VOP

Negli standard precedenti all'MPEG-4 le sequenze video sono codificate come immagini fisse di forma rettangolare; l'MPEG-4 pur mantenendo la piena compatibilità con gli standard precedenti, utilizza una codifica di oggetti separati di forma arbitraria (VO) aggiungendo i dati relativi alla forma rispetto agli standard precedenti.
L'algoritmo di codifica si basa sulla DCT secondo blocchi 8x8 (per ridurre la ridondanza spaziale) e sulla compensazione del movimento su macroblocchi (blocchi 16x16, per ridurre la ridondanza temporale). Le immagini statiche, in particolare lo sfondo, utilizzano una codifica di tipo Intra mentre gli oggetti i movimento utilizzano una codifica di tipo Inter.

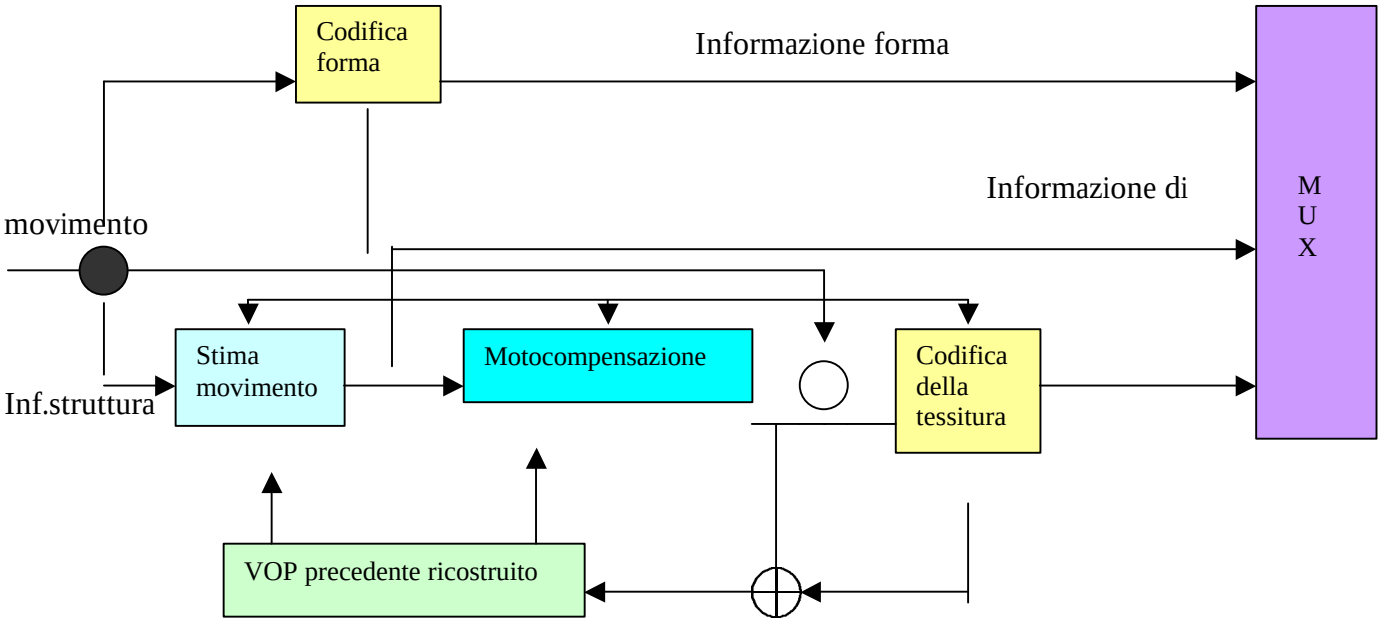


Gli oggetti da codificare sono i VO; l'insieme di questi in un determinato istante di tempo è chiamato VOP. Il codificatore trasmette il VOP e le informazioni per indicare dove e quando questo deve essere visualizzato. In generale un VOP è costituito da informazioni sulla forma ed da informazioni YUV.

L'operazione di codifica dei Vop è composta da due parti principali:

- Codifica della forma
- Codifica di movimento e tessitura

Le informazioni sul movimento, la forma e la "tessitura" possono essere combinate producendo un unico *bitstream* in cui i bit sono organizzati VOP per VOP, oppure lasciate separate in cui il *bitstream* contiene, una dopo l'altra, sequenze di bit che portano informazioni diverse sul movimento forma e "tessitura".



Struttura del codificatore dei VOP

Codifica della forma

Tale codifica contiene due tipi di informazioni inerenti la forma , una binaria e l'altra a scala di grigi .
Con la codifica dell'informazione di forma binaria si intende l'informazione che definisce quali pixel del supporto dell'oggetto appartengono all'oggetto in un certo istante.
La rappresentazione è quella di una matrice della stessa dimensione del blocco che include l'oggetto, e che ne costituisce il supporto : ogni elemento della matrice può assumere due possibili valori, 0 o 255, a seconda che ogni pixel sia interno o esterno all'oggetto.
La rappresentazione dell'informazione di forma a scala di grigi è simile alla precedente, ma ogni elemento della matrice indica il grado di trasparenza, variabile tra 0 e 255.
Nel caso di forma binaria si codifica mediante una tecnica *block-based* con *motion compensation*, che consente compressioni con perdite o senza perdite; invece nell'altro caso si utilizzano le stesse tecniche ma sfruttando anche la DCT, consentendo solo compressioni con perdite.
Il supporto è scelto in modo da minimizzare il numero di macroblocchi (16x16 pixel) che contengono pixel esterni all'oggetto.

La codifica binaria sull'informazione della forma per prima cosa suddivide il VOP contenuto nel supporto rettangolare in macroblocchi 16x16 pixel che verranno codificati e trasmessi.
La codifica di un macroblocco sfrutta il metodo *run-lenght* , rileva i cambiamenti di valore tra un pixel e l'altro e calcola le distanza fra i cambiamenti successivi. Se tutti i pixel di un macroblocco sono uguali viene trasmessa l'informazione che il macroblocco contiene solo pixel esterni o interni al VOP.

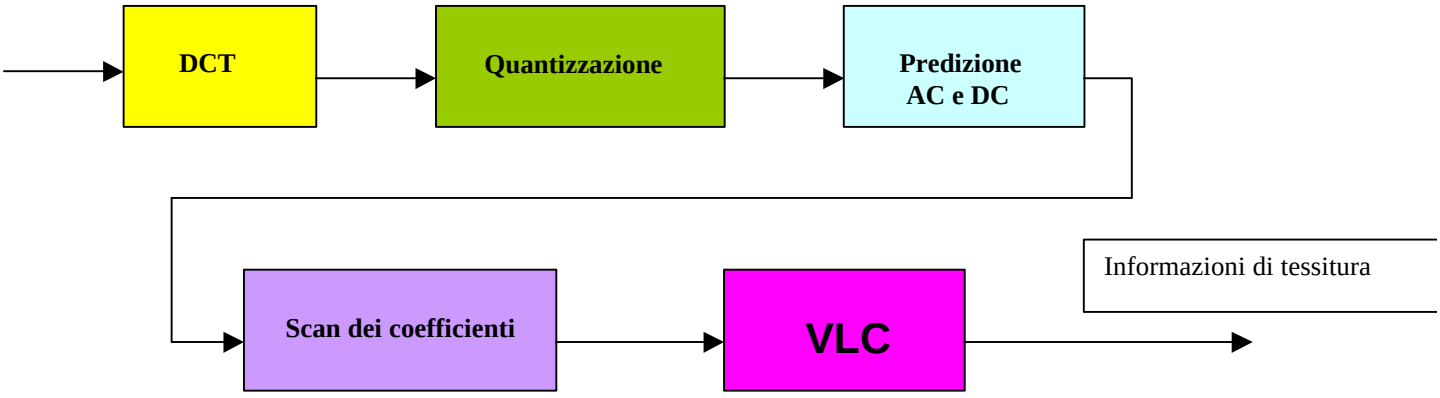
La codifica a scala di grigi avviene separando il processo di codifica del supporto dell'immagine da quello dell'informazione della trasparenza.

Compensazione e stima del movimento

Si usano algoritmi basati sulla suddivisione dei VOP in macroblocchi e su due nuove proprietà chiamate "*padding*" e "*modified block matching motion estimation*". I macroblocchi che si trovano all'esterno del VOP sono ignorati, quelli che si trovano all'interno vengono sottoposti alla compensazione del movimento utilizzata nei precedenti standard MPEG, mentre quelli che si trovano parzialmente dentro il VOP sono sottoposti al "*modified block matching motion estimation*". L'errore di *block matching* si calcola come somma dei valori assoluti delle differenze fra i pixel che sono all'interno del VOP attuale ed i pixel del VOP di riferimento. Poichè la forma di ogni VOP può cambiare da un quadro al successivo si utilizza una tecnica di riempimento del VOP di riferimento chiamata "*padding*" : consiste nell'estendere le proprietà dei blocchi al bordo del VOP di riferimento all'esterno dello stesso. Ogni VOP può essere codificato usando una codifica I (codifica intra), P (codifica di predizione), B (codifica bidirezionale),

Codifica della "tessitura"

La tecnica di codifica è simile a quella usata per gli standard precedenti con l'aggiunta della suddivisione del VOP in macroblocchi di dimensione 16x16. Ogni macroblocco è codificato con la DCT, separando le componenti di luminanza, formate da quattro blocchi 8x8 (Y), e quelle di crominanza , formate da due blocchi 8x8 (U V). I macroblocchi interni al VOP vengono codificati tramite DCT, quelli esterni non vengono codificati, per quelli parzialmente all'interno si usa la tecnica di "*padding*". Successivamente i coefficienti della DCT vengono sottoposti a quantizzazione, *zig zag* , e codifica di Huffman. Per la quantizzazione si hanno due alternative: la prima utilizza lo stesso passo di quantizzazione per coefficienti di frequenza non nulla, la seconda utilizza una matrice di pesi che assegna maggiore importanza ai coefficiente di frequenza più bassa.



Codifica scalabile

Un *bitstrream* si definisce scalabile se un suo sottoinsieme è sufficiente a generare una rappresentazione accettabile in base ai criteri dell'utente. In pratica la scalabilità permette la decodifica di alcune parti del flusso dati tali da generare immagini complete di qualità proporzionale alla grandezza della parte decodificata. Per questo il codificatore deve generare il flusso di dati che faciliti l'estrazione di video a differenti qualità dal *bitstream.*