Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

UNIVERSITÀ DEGI STUDI DI BOLOGNA

INGEGNERIA DELLE TELECOMUNICAZIONI

ANNO ACCADEMICO 1999/2000

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

EVOLUZIONE DEGLI STANDARD MPEG

LUCA MARCEGLIA

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

INTRODUZIONE

Fin da quando è comparso sulla Terra, l'uomo ha sempre avuto il desiderio e la necessità di arricchire la propria esperienza e conoscenza e per questo ha dato forma ai suoni, alle idee, ai sentimenti per trasmetterli agli altri. I mezzi impiegati a tale scopo sono stati molteplici, dai graffiti, alle incisioni, ai segnali di fumo - strumenti tuttora impiegati dalle popolazioni odierne considerate "sottosviluppate". Da allora moltissimi secoli sono passati, ed anche se è cambiato il mezzo, l'interesse verso la comunicazione non è scemato; anzi, è stato proprio questo impulso a dare vita ad esempio all'invenzione della stamperia di Gutemberg.

Ai nostri giorni, i mezzi che possiamo impiegare per comunicare con il prossimo sono innumerevoli, e spaziano dalle rotative, alla televisione, ad Internet; ma, se da un lato questo progresso ha avuto lo scopo sperato, dall'altro ha creato, paradossalmente, dei problemi nella comunicazione stessa: la televisione esiste in tutto il mondo, ma una videocassetta comprata in Giappone non si può vedere in Italia; un cellulare europeo non funziona negli Stati Uniti; un riproduttore di CD comprato in Inghilterra funziona con un voltaggio diverso da quello italiano.

Questo tipo di incompatibilità tra strumenti nati per soddisfare le stesse esigenze nacque, verso la fine degli anni '80, anche nel campo del "multimediale", ovvero di tutto ciò che concerne audio, video e la loro interazione con l'uomo. Il multimediale è stata un'innovazione molto importante del secolo XX, ed ha dato una forte risposta alla richiesta di informazione e comunicazione: si pensi alle enciclopedie su CD-ROM, ai musei virtuali, ma anche al cinema di qualità e persino ai giochi cosiddetti "multiplayer", che possono favorire nuove amicizie. Tutto ormai è multimediale, e fa parte di noi e della nostra cultura.

Nel mondo si parlano moltissime lingue, e questo a volte rende difficile, o comunque rallenta, la comunicazione dell'uomo con i propri simili; ed anche se sono stati fatti numerosi tentativi per risolvere l'annoso problema, come ad esempio l'introduzione dell'Esperanto, i limiti rimangono. Lo stesso problema ha colpito le moderne comunicazioni, nate in ambienti diversi, con mercati diversi e mentalità diverse ed in questo caso le varie "lingue" si chiamano standard. Sono proprio i diversi standard a non far funzionare il cellulare ovunque, e a non permetterci di comprare in Giappone un televisore per portarlo in Italia e spesso, a differenza di quanto avviene per le lingue straniere, non esistono "traduzioni" efficaci che ci consentano di ovviare al problema.

Fortunatamente, però, nel campo del multimediale esiste un linguaggio comune, un Esperanto informatico, una soluzione a tutti i mali. Questa soluzione di chiama MPEG, dal nome del gruppo che per primo lo inventò nel 1987 (Motion Picture Expert Group), e questa tesi vuole ripercorrere la

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

storia e gli sviluppi di questo standard riconosciuto ed accettato in tutto il mondo da ormai più di 12 anni.

Il successo di MPEG dipende da vari elementi. Forse, il più importante è che venne creato al momento giusto, quando gli algoritmi di codifica audio e video ormai erano estremamente efficaci, e si era ormai capaci di portare su silicio algoritmi così complicati. MPEG riuscì a catturare l'attenzione di tutte le industrie importanti del settore, che parteciparono con suggerimenti, consigli e test di verifica. Inoltre, per la prima volta non veniva standardizzato il codificatore, ma solo il decodificatore, lasciando quindi libertà alle industrie stesse nella realizzazione di encoder più o meno complessi a seconda delle esigenze, e si studiavano i segnali video ed audio non in modo separato, come invece era sempre avvenuto prima, bensì come un unicum comandato da vari segnali di temporizzazione; e tutto questo, al fine di soddisfare l'utente finale più che il rivenditore.

Tratterò dunque della nascita di MPEG, dei vari protocolli studiati ed accettati nel corso degli anni, ovvero di MPEG-1, MPEG-2 ed MPEG-4, e del futuro dei questo gruppo, con i protocolli MPEG-7 e MPEG-21, il primo disponibile nel 2001.

Infine, ho aggiunto nel CD-ROM allegato vari esempi e programmi software perché meglio si possa capire lo sviluppo del multimediale, se ancora ve ne fosse bisogno, e l'enorme salto in avanti compiuto da MPEG durante questi anni.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

1 NASCITA DI MPEG

Verso la fine degli anni ’80, nel mondo delle Telecomunicazioni esistevano già numerosi standard per le codifiche audio e video:

- l’ITU-T, con i suoi standard SG XV WP 1 (per il parlato) e SG XV WP 2 (per il video);
- l’ITU-R, con l’ SG 10 (audio) e SG 11 (video);
- l’IEC, che definiva lo standard per la registrazione dell’audio (SC 60 A) e del video (SC 60 B), nonché per gli apparati audio-visivi (TC 84);
- l’ISO, infine, gestiva gli standard per la fotografia (TC 42) e la cinematografia (TC 36).

Oltre a ciò, molte erano le iniziative in atto:

- Il CCITT stava concludendo le raccomandazioni H.261 per la codifica video;
- Era nato il DVI (Digital Video Interactive) presso il Centro di Ricerca Sarnoff, per la codifica di video interattivo, audio ed altro su CD-ROM;
- La Philips stava lavorando sulla stessa idea per creare i CD-I (CD Interactive);
- Il CMTT stava studiando un modo per codificare il segnale video a 34/45 Mb/s;
- Negli Stati Uniti, la FCA (Federal Communications Commission) stava abbandonando l’idea dell’analogico per lanciarsi nello studio della televisione digitale;
- Telettra e la RAI stavano studiando la distribuzione della HDTV (High Definition TeleVision, la televisione ad alta definizione) via satellite;
- Era stato appena creato il progetto EU 147 – DAB (Digital Audio Broadcasting) per la trasmissione via radio di musica con qualità pari a quella di un CD.

Tutta questa “confusione” di standard e brulicare di studi derivavano dal fatto che nel campo del multimediale, nuovo e quindi potenzialmente molto redditizio, moltissime industrie erano interessate a risolvere solo i problemi tecnici legati a ciò che volevano produrre, ed a portarlo a compimento nel minor tempo possibile per accaparrarsi una grande fetta di mercato. Non a caso, Sarnoff e la Philips stavano lavorando allo stesso progetto, ma in concorrenza tra loro.
Insomma, i prodotti ormai affermatisi nel mercato stabilivano uno standard “de facto” che poteva variare da nazione a nazione e di anno in anno; non era il mercato a seguire gli standard, ma al contrario era il mercato ad imporli. Questo rallentava anche lo sviluppo di nuove tecnologie proprio

perché nel campo del multimediale non c'era mai stato un vero studio del problema a tavolino. Leonardo Chiariglione affrontò il problema.

1.1 MPEG E LA FILOSOFIA DI CHIARIGLIONE

Chiariglione si interessò seriamente alla televisione solo dopo essere entrato in CSELT (Centro Studi e Laboratori Telecomunicazioni SpA) all'inizio degli anni '70. Dopo essere stato deluso, verso la metà degli anni '80, dal fallimento dell'adozione di un unico standard mondiale per la creazione dell'HDTV, egli cercò il momento propizio per lanciare l'idea di standard molto più uniformi.

Nel Marzo del 1987 Chiariglione rimase molto interessato dal meeting di JPEG (Joint Photographic Experts Group), perché il lavoro di questi esperti, non soggetti ad alcun legame con le industrie, era davvero notevole. A capo di JPEG c'era Hiroshi Yasuda, suo amico di studi all'Università del Giappone, e Chiariglione persuase abbastanza facilmente Yasuda ad appoggiarlo nella creazione di un analogo gruppo di esperti, da dedicare però alla codifica di immagini in movimento[Birkmaier].

Figura 1: Leonardo Chiariglione

Poco dopo avvenne la fusione del Comitato Tecnico 97 ISO, dal nome "Data Processing", con il TC 46 della IEC, "Microprocessors". Nacque così il JTC1 (Joint Technical Committee 1), dal nome "Information Technology". Questo venne poi suddiviso in vari sottocomitati, tra cui l' SC29 ("Coding of Picture, Audio, Multimedia and Hypermedia").

Al Working Group numero 11 del sottocomitato SC29, dal titolo "Coding of Moving Pictures and Audio", venne dato il soprannome MPEG (Motion Pictures Expert Group).

La prima riunione di MPEG si svolse ad Ottawa dal 10 al 12 Maggio 1988 e venne seguita da una quindicina di persone (perlopiù provenienti dal gruppo parallelo JPEG). Sono passati ormai più di 11 anni da quel giorno, ma la filosofia di questo gruppo è rimasta immutata. Vediamone i punti salienti[Chiariglione96]:

- *Stick to the deadline*: MPEG è un comitato che pensa e produce un prodotto, che in questo caso si chiama "standard"; ed esattamente come qualsiasi industria che voglia immettere un prodotto di qualità nel mercato, anche MPEG si dà delle scadenze, perché non rientra nella filosofia di

questo gruppo il concetto di "consegnare qualcosa prima o poi". Per questo, MPEG ha una tabella di marcia che stabilisce i tempi limite per il termine dei Working Draft, Committee Draft, Draft International Standards e International Standards.

- *A-priori standardisation*: Spesso, a causa del fatto che molti gruppi di studio non si danno dei tempi prefissati, il mercato si trova un passo più avanti degli standard e così si abbandona lo studio per standardizzare il prodotto già finito ed affermato; in questo modo, i gruppi di studio passano dal campo tecnico a quello meramente commerciale. L'approccio di MPEG invece segue dei passi molto precisi:
  1. Viene identificata la maturità della tecnologia da standardizzare prima che le industrie la usino;
  2. Viene generalmente redatto un "Call For Proposals" a tutte le ditte interessate, contando anche sulla loro partecipazione;
  3. In ogni caso, la standardizzazione finale spetta a MPEG.

  Si capisce come le industrie sono quindi ben accette nella collaborazione ed hanno anche la possibilità di continuare a creare degli "standard" propri in mancanza di specifiche MPEG.
- *Not systems but tools*: Dato che le industrie creano i loro prodotti (*systems*) attraverso dei componenti (*tools*), la cosa importante per MPEG è quella di standardizzare i componenti e lasciare alle industrie ampia libertà nella costruzione del prodotto finale. In questo modo, le industrie dovranno:
  1. Scegliere le applicazioni per le quali una certa tecnologia deve essere applicata;
  2. Creare una lista di tutte le funzionalità usate dalle applicazioni;
  3. Frammentarle in componenti di complessità ridotta;
  4. Identificare i componenti comuni;
  5. Specificare i tool che supportano i componenti identificati;
  6. Verificare che, una volta scelti, i tool possano effettivamente essere assemblati per la produzione di una determinata applicazione.

  Non sempre comunque è facile definire un tool. Per esempio, può essere complicato definire un tool per la codifica di un singolo canale audio e uno per il multicanale, oppure per la televisione normale e quella ad alta definizione (HDTV). In questi casi, la filosofia di MPEG divide uno stesso tool in diversi gradi, chiamati livelli (*levels*).
- *Specify the minimum*: Al fine di evitare che lo standard converga sempre più ad una specifica di prodotto (come nel caso di "qualità del servizio"), viene standardizzato solo il minimo

indispensabile per la compatibilità. Nel caso si voglia superare questo limite, devono essere coinvolte tutte le industrie del settore.

- *One functionality – one tool* : Uno dei problemi della compatibilità è che, una volta definito il tool, nascono sempre delle opzioni che ne compromettono lo standard: un esempio è dato dall'ISDN che, a causa delle moltissime opzioni nella gestione dei segnali introdotte nei diversi Paesi, fu incompatibile per 10 anni tra le diverse Compagnie di Telecomunicazioni europee.
  La filosofia di MPEG stabilisce dunque che queste opzioni vengano abolite o studiate di comune accordo.
- *Relocation of tools*: Generalmente, le industrie definivano anche dove il tool dovesse risiedere nel sistema. Così facendo, però, se un certo produttore avesse voluto usare quel tool altrove, non avrebbe usato lo standard, ed anzi lo avrebbe combattuto.
- *Verification of standard*: Tramite un "Verification Test", MPEG controlla che il lavoro svolto fino a quel momento soddisfi effettivamente le richieste iniziali. In questo modo, si può comprendere quanto funziona lo standard, e come viene accettato dal mercato.

1.2 MPEG OGGI E GLI SCOPI DI MPEG

Dopo 11 anni e 48 meeting, MPEG può contare su più di 300 membri, tra cui più di 200 rappresentanti di diverse industrie; si riunisce 4-5 volte l'anno e crea circa 40 nuovi gruppi ad hoc ad ogni incontro. Al momento è organizzato e suddiviso come mostrato in Tabella 1.
Fino ad oggi, MPEG ha studiato e creato diversi standard[Chiariglione99]:

- *MPEG-1*: E' stato il primo standard integrato audio-video (ISO/IEC 11172). Nato nel 1992, è stato anche il primo standard a definire il ricevitore e non il trasmettitore, il primo con codifica video indipendente dal formato NTSC/PAL/SECAM, il primo sviluppato da tutte le industrie del settore, il primo sviluppato completamente via software e che includeva una implementazione via software. Nato con lo scopo di riprodurre dei film su CD con qualità audio paragonabile a quella di un CD stereo, e con qualità video pari a quella di un videoregistratore, ha venduto decine di milioni di lettori nella sola Repubblica Popolare Cinese; è "il" formato di riproduzione audio e video per PC nei sistemi Windows; è stato utilizzato dai sistemi DAB (Digital Audio Broadcasting) in Europa e Canada; è implementato nelle Webcams per videoconferenza; il protocollo audio Layer 3 è diffusissimo per la musica sul web.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Requirements	Sviluppa gli standard sotto studio (al momento, MPEG-4 e MPEG-7)
1. DSM	Sviluppa gli standard per le interfacce tra Digital Storage Media (DSM), server e client allo scopo di gestire le risorse DSM e controllare il trasporto dei bitstream MPEG e dei dati associati
2. Delivery	Sviluppa gli standard per l'interfaccia tra le applicazioni MPEG-4 ed i mezzi di trasmissione allo scopo di gestire le risorse di gestione del trasporto
3. Systems	Sviluppa gli standard per la codifica della combinazione di audio codificato individualmente, immagini in movimento ed informazioni correlate, in modo da usare la combinazione in qualunque applicazione
4. Video	Sviluppa gli standard per la rappresentazione codificata di video di origine naturale
5. Audio	Sviluppa gli standard per la rappresentazione codificata di audio di origine naturale
6. SNHC	Synthetic – Natural Hybrid Coding: sviluppa gli standard per la codifica di audio di origine sintetica
7. Test	Verifica la qualità della codifica audio e video, presi singolarmente e non, e sviluppa nuovi metodi di test soggettivo
8. Implementation	Verifica le tecniche di codifica per aiutare gli altri gruppi sui vari parametri di implementazione della codifica
9. Liaison	Gestisce i rapporti con gli altri gruppi esterni all'MPEG
10. HoD	Heads of Delegations: Consiglieri su fatti di varia natura

Tabella 1: Organizzazione e suddivisione di MPEG

- *MPEG-2*: Nasce nel novembre 1994, con la sigla ISO/IEC 13818, per la codifica del segnale televisivo in migrazione dall'analogico al digitale. Lo standard precedente, infatti, non era adatto per la codifica di segnali televisivi e non prevedeva un controllo degli errori, perché pensato solo per l'archiviazione su CD. Molte compagnie di diffusione televisiva volevano invece migliorare la qualità e la quantità dei loro programmi trasmessi via satellite, e la banda di segnale occupata dall'analogico non lo permetteva. MPEG-2 fornisce ottima qualità audio e video, e permette funzioni particolari come l'acquisto di un determinato programma (funzione PPV, Pay per View) o la visione di un determinato evento all'ora voluta dallo spettatore e non

dal fornitore (N-VOD, Near Video On Demand). Sono stati venduti decine di milioni di ricevitori digitali per televisione via cavo e satellite, nonché più di due milioni di DVD, per un giro di circa 30 miliardi di dollari americani.

- *MPEG-3*: Venne creato per lo studio della televisione ad alta definizione (HDTV), ma poi si scoprì che lo stesso MPEG-2 poteva essere modificato per ottenere lo stesso scopo.
- *MPEG-4*: Nato nell'ottobre 1998 (come MPEG-4 fase 1, ISO/IEC 14496) e nel dicembre 1999 (fase 2), è caratterizzato da una architettura ad oggetti e dalla capacità di integrare video ed audio naturale con quello generato sinteticamente attraverso i computer. Fornisce una compressione maggiore rispetto agli standard precedenti, e permette la trasmissione di audio e video a bit rate molto basse; introduce la codifica della voce; un controllo molto robusto dell'errore in applicazioni quali Internet o telefonia mobile; codifica in maniera efficiente oggetti in 3D; genera musica sintetica migliorando di molto ciò che finora era possibile grazie al MIDI; permette l'interazione dell'utente attraverso la scelta di diverse inquadrature per una stessa scena, o di funzioni quali zoom, rotazione, traslazione degli oggetti; si presta molto bene a manipolazioni di immagini (aggiunta od eliminazione di un certo personaggio in una scena, cambiamento dello sfondo…). Al momento sono in studio le versioni 3 e 4, e si propone come standard per le comunicazioni multimediali.
- *MPEG-5*: Non definito.
- *MPEG-6*: Non definito. MPEG decise di non voler seguire un ordine sequenziale preciso e quindi saltò dal 4 al 7.
- *MPEG-7:* Lo studio è iniziato nel 1997 (ISO/IEC 15938) e sarà pronto nel 2001. Servirà ad effettuare ricerche su database di tipo video ed audio tramite la standardizzazione di descrittori e strutture di descrittori. I motori di ricerca basati su MPEG-7 forniranno in risposta immagini simili a quelle proposte dall'utente; sequenze musicali a partire da poche note; vestiti con forma, modello e colore come da richiesta ("Vorrei un maglione scollato a V taglia 42 Armani verde scuro"); modelli di automobili a partire da indicazioni sul comfort, velocità e consumo ("Cerco un'automobile con sedili in pelle, con un'autonomia di almeno 15 Km/l e velocità massima di 180 Km/h"); particolari sequenze in una trasmissione sportiva ("Mostrami tutti i testacoda nel GP di Imola 1996"); ed altro ancora. Lo standard non dice come creare i motori di ricerca, ma piuttosto come descrivere determinate caratteristiche di un oggetto quali colore, forma, emozione suscitata.
- *MPEG-21*: Lo studio di questo standard è iniziato nell'ottobre 1999, ed al momento attuale sono state sviluppate solo le prime idee introduttive. Si propone di identificare, nel campo del

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

commercio elettronico (*e-commerce*), l'eventuale necessità di nuovi standard appartenenti alle competenze del gruppo.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

2 MPEG-1

Il primo standard sviluppato dal gruppo riguardava la codifica di segnali combinati di tipo audio e video con una bit rate attorno a 1.5 Mb/sec. Ciò nasceva dal fatto che, da studi effettuati nel 1988, si evinceva la possibilità di registrare filmati con qualità confrontabile a quella di una cassetta VHS su un compact disc.
Il titolo ufficiale dell'MPEG-1 è *ISO/IEC 1172 "Information technology – Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbit/sec", © 1993* e consiste di cinque parti. Le prime tre vennero approvate nel 1993; la parte 4 nel 1994 e l'ultima nel 1995.

- *Systems*: Riguarda sia la sintassi per il trasporto dei pacchetti audio e video su canali digitali e DSM, sia la sintassi necessaria alla sincronizzazione video ed audio;
- *Video*: Descrive la sintassi e la semantica della codifica video;
- *Audio*: Descrive la sintassi e la semantica della codifica audio;
- *Conformance*: Definisce la conformità dello standard MPEG per le tre precedenti parti e fornisce due tipi di linee guida per determinare una conformità nei bitstream e nei decoder;
- *Software Simulation*: Contiene un esempio di encoder scritto in ANSI C ed un decoder conforme alle specifiche sia per il video che per l'audio. Viene anche fornito un codec di sistema in modo da multiplare e demultilre stream diversi di audio e video in file per PC.

Esaminiamo adesso nei dettagli le parti concernenti il video e l'audio.

2.1 LA CODIFICA VIDEO IN MPEG-1

Generalmente, le sequenze video contengono una significativa quantità di ridondanza statistica e soggettiva nei e tra i frame. Lo scopo della codifica video è la riduzione del bit rate sfruttando queste ridondanze per codificare le minime informazioni possibili da trasmettere. La qualità e la quantità del segnale compresso dipendono sia dalla ridondanza presente nell'immagine, sia dall'efficienza dell'algoritmo di compressione.
Possiamo distinguere due tipi di codifica del segnale video: uno cosiddetto *lossy* e uno di tipo *lossless*. Nel caso del lossless, si richiede che la qualità finale dell'immagine sia la stessa di

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

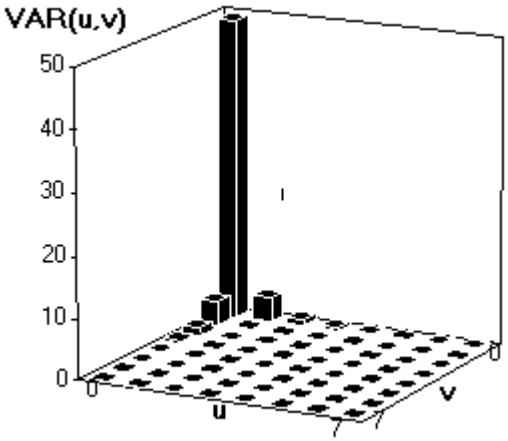


Figura 2: Varianza in un blocco 8x8 pixel

partenza dopo la decodifica, mentre lo scopo della codifica lossy – sfruttato dagli standard MPEG-1 e MPEG-2 – è quello di trasmettere i dati ad un certo bit rate. In quest'ultimo caso, soprattutto per applicazioni a banda stretta, la qualità oggettiva del segnale è peggiore dopo la decodifica.

L'approccio di tipo MPEG alla codifica di un generico segnale video è quello di suddividere una certa sequenza in movimento in tre diversi tipi di frame:

- Frame di tipo **I**
- Frame di tipo **P**
- Frame di tipo **B**

2.1.1 FRAME DI TIPO I

L'idea è quella di decorrelare il contenuto dell'immagine e di codificarne i coefficienti della trasformata piuttosto che i veri e propri pixel. A tal scopo, le immagini vengono divise in blocchi **b** di NxN pixel che sono poi trasformati da una matrice **A** secondo la formula[Sikora97]

$$c = A\, b\, A^T$$

In questo modo, il blocco **b** può essere ricostruito in fase di decodifica, dato che

$$b = A^T\, c\, A$$

Tra le varie alternative, la trasformata migliore si è rivelata essere la Discrete Cosine Transform, che è di tipo lossless.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Si è visto che un blocco di 8x8 pixel basta ad esplorare in maniera efficiente la correlazione fra i vari pixel, e dunque la tecnica di codifica consiste nel dividere un frame in macroblocchi di tali dimensioni, per poi applicarvi la DCT. La figura 2 mostra la varianza dei coefficienti DCT su un blocco di 8x8 pixel. I coefficienti con piccoli valori sono i meno importanti per la ricostruzione

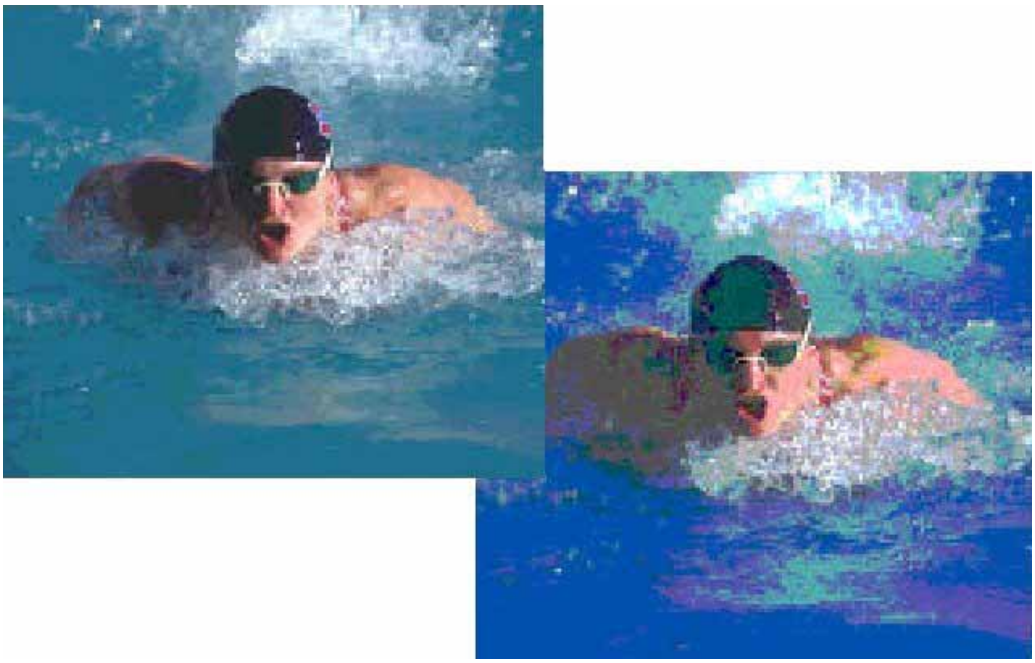


Figura 3: Immagine originale (a sinistra) e quantizzata (a destra)

dell'immagine, mentre quelli con i valori più alti sono i più rilevanti; come si vede dall'immagine, dunque, basta trasmettere solo pochi coefficienti per avere un'approssimazione molto buona dell'immagine iniziale. Dato che i valori più importanti si trovano vicino all'origine, aumentare la grandezza di ogni singolo macroblocco non solo non aumenta il grado di compressione, ma al contrario peggiora la velocità di codifica.

Finora non è stata ridotta la bit rate, ed un modo efficace per farlo consiste nel ridurre i coefficienti della trasformata dividendoli per un determinato fattore. La riduzione dei coefficienti della DCT è molto utile per ridurre l'ampiezza di banda, dato che molti di essi andranno a zero e non dovranno essere trasmessi, ma una compressione troppo elevata porta ad un'esagerata *quantizzazione* dell'immagine, come si può notare dalla figura 3. Un'ulteriore miglioria che si può apportare a questo tipo di codifica consiste nell'aumentare il numero dei coefficienti con zeri consecutivi, per poi codificarli con l'algoritmo "run-length" prima della loro trasmissione. Questo algoritmo infatti conta quante volte un certo valore è ripetuto in maniera consecutiva e ne codifica solo il valore ed il numero delle occorrenze[Fogg][hp]. Una tecnica efficiente è risultata essere il cosiddetto "Zig-zag

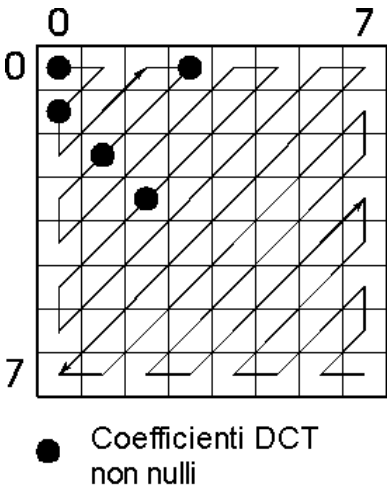


Figura 4: Codifica Zig-Zag

scanning", che cerca di "seguire" i coefficienti in ordine di importanza, dato che, come già sappiamo, questi perdono di valore man mano che ci si allontana dall'origine. La figura 4 chiarisce meglio il concetto.

2.1.2 FRAME DI TIPO P

Nel paragrafo precedente si è trattato di come comprimere e trasmettere un determinato frame studiandone solo la ridondanza spaziale. Dato che però MPEG si occupa di immagini in movimento, è importante anche studiare quale ridondanza e correlazione esistano tra frame adiacenti in modo da minimizzarne la bit rate.

Se per esempio ci troviamo di fronte alla scena di un'auto in movimento, moltissimi pixel saranno presenti nei vari frame approssimativamente traslati di una stessa distanza. In tal caso, basta controllare di quanto si sono spostati i singoli macroblocchi per poi codificarli calcolandone la posizione con un vettore di moto a partire da quelli precedenti (come si vede dalla figura 5) ed aggiungendo la sola codifica dei pixel che non rispettano questa regola. Se poi un determinato

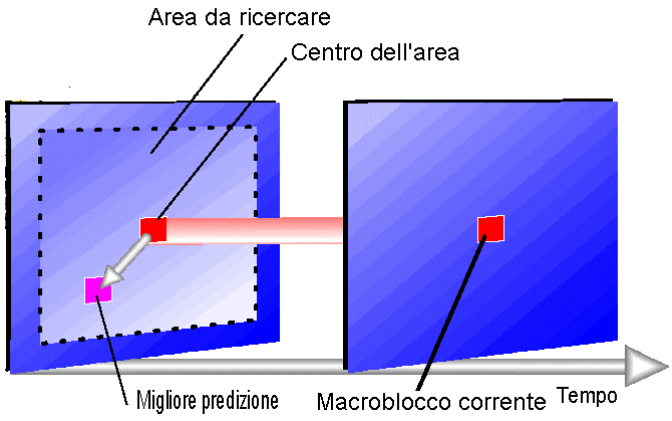


Figura 5: Calcolo del Vettore di Moto

macroblocco rimane nella stessa posizione (si pensi ad uno sfondo fisso), nessuna codifica è necessaria: basta inviare un codice di *Skipped Macroblock* ed il decoder, in fase di lettura, opererà con una semplice funzione di copia/incolla. Generalmente, si utilizzano dei macroblocchi della dimensione di 16x16 pixel, e la combinazione di questa tecnica differenziale (denominata DPCM) e

della tecnica DCT è la chiave dello standard di compressione MPEG. Nelle immagini che seguono possiamo apprezzare la quantità di immagine da codificare senza e con il vettore di moto.

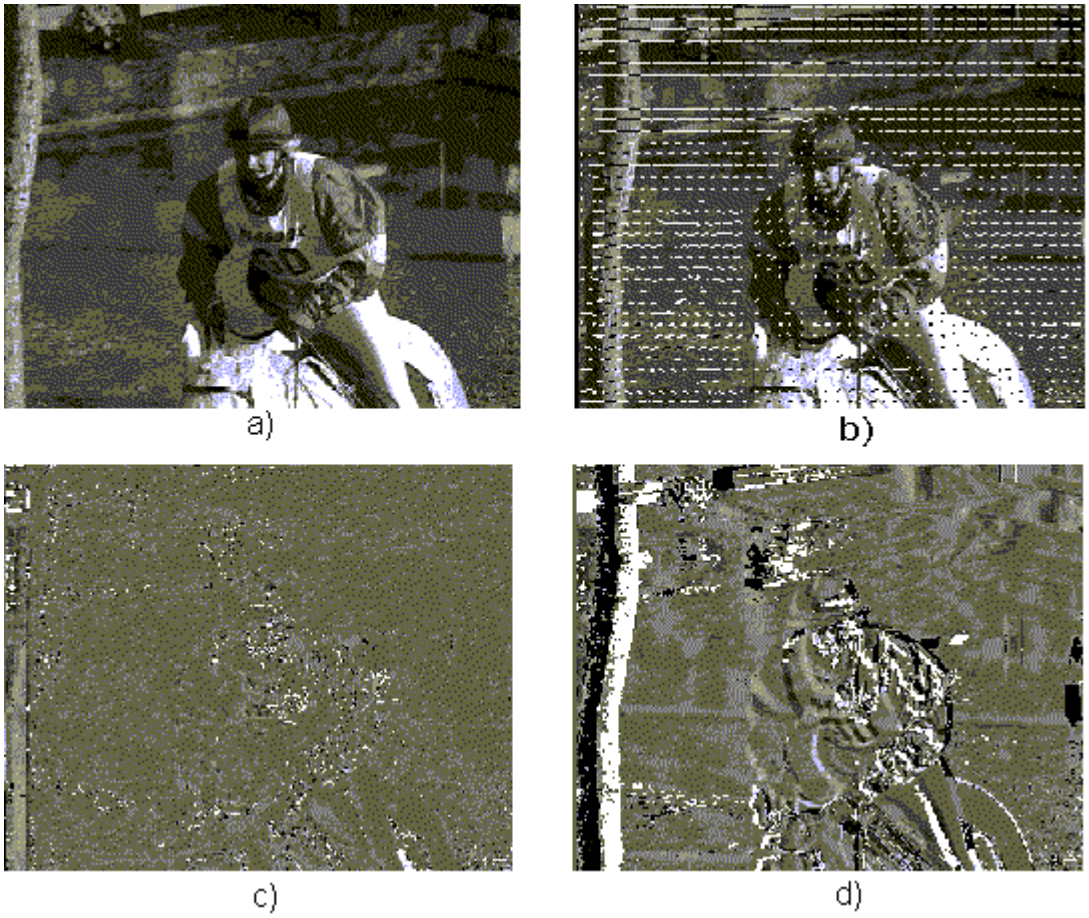


Figura 6: a) Frame iniziale; b) Frame successivo; c) Codifica tramite vettore di moto; d) Codifica senza vettore di moto

2.1.3 FRAME DI TIPO B

Un altro tipo di codifica, in grado di offrire la massima compressione ma anche maggiormente sensibile agli errori, consiste nel codificare un nuovo frame non solo a partire da quello precedente (I o P), come appena visto, ma anche usando i frame I o P successivi. La maggior codifica dipende dal fatto che una parte del frame potrebbe essere correlata solo con quello precedente, ed un'altra solo con quello successivo: la somma delle due correlazioni permette un ampio uso del vettore di moto e quindi una bassa quantità di dati da trasmettere. Inoltre, Una delle caratteristiche di MPEG-1 è quella di permettere un accesso casuale (RA) al filmato, così come l'avanzamento veloce (FF) ed

il ritorno veloce (FR). Anche per questo motivo sono stati introdotti i frame di tipo B **(***B*i-directional predicted frame). Infatti, mentre i frame di tipo P non permettono un accesso casuale perché dipendono solo da frame precedenti, quelli di tipo I lo consentono ma a prezzo di una scarsa compressione. Codificando invece un frame a partire sia da quello precedente, sia dal successivo, si ottiene l'accesso casuale e una buona codifica allo stesso tempo.

2.2 COMPRESSIONE IN MPEG-1

Entrando nei dettagli della specifica MPEG-1, la codifica viene eseguita a partire da un segnale video conosciuto come "D-1" o "CCIR 601", codificato a 270 Mbit/sec non interlacciato. L'immagine, prima di essere compressa, viene divisa nelle componenti YUV (una di luminanza e due di crominanza). Il segnale iniziale è dunque caratterizzato da[Fogg]:

- *Canale Y:* 858 sample/linea x 525 linee/frame x 30 frame/sec x 10 bit/sample (135 Mbit/sec)
- *Canale U*: 429 sample/linea x 525 linee/frame x 30 frame/sec x 10 bit/sample (68 Mbit/sec)
- *Canale V*: 429 sample/linea x 525 linee/frame x 30 frame/sec x 10 bit/sample (68 Mbit/sec)

Il rapporto tra le componenti Y, U e V viene chiamato *chroma ratio* e viene definito come Y:U:V. In genere, ad Y viene assegnato il valore 4; il secondo valore deriva dal rapporto $\frac{U}{Y}\cdot 4$ calcolato sui sample per linea (in questo caso, $\frac{429}{858}\cdot 4 = 2$); il terzo è uguale al secondo se il rapporto linee/frame non cambia e 0 se viene ridotto della metà (nel nostro caso, il rapporto è uguale e quindi il valore di V è 2). Dunque, il segnale CCIR-601 ha un chroma ratio di 4:2:2.
In un film, solo tra i 704 ed i 720 sample su 858 e dalle 480 alle 496 linee su 525 contengono informazioni per il canale Y, mentre per gli altri due canali sono "utili" solo 352 su 459 sample. MPEG ha stabilito dunque che il decoder debba avere una conformità di 704 sample x 480 linee per il canale Y, e di 352 sample x 480 linee per gli altri due canali, con un sample di 8 bit anziché 10.
Dato che l'occhio umano è più sensibile alle variazioni di luminosità che di crominanza, è possibile ridurre ulteriormente l'informazione trasportata sui canali U e V: queste ultime componenti vengono dunque sottocampionate per raggiungere un chroma ratio di 4:2:0 e l'immagine viene convertita in formato SIF (Source Input Format – 352 sample/linea x 240 linee/frame). Il formato finale dunque è il seguente:

- *Canale Y*: 352 sample/linea x 240 linee/frame x 30 frame/sec x 8 bit/sample (21 Mbit/sec)
- *Canali U,V*: 176 sample/linea x 120 linee/frame x 30 frame/sec x 8 bit/sample (10 Mbit/sec)

I valori riportati sopra si riferiscono agli USA ed al Giappone. In Europa si passa da 240 a 288 linee, da 120 a 144, da 30 frame a 25. Dato però che $288 \cdot 25 = 240 \cdot 30$ e che $120 \cdot 30 = 144 \cdot 25$, i due formati hanno le stesse caratteristiche e quindi il formato MPEG-1 è valido per i sistemi NTSC/PAL/SECAM[Fogg].
Un particolare da tenere in mente è che lo standard MPEG-1 viene applicato ai film, che vengono creati a 24 frame/sec; quindi i 6 frame che restano sono ridondanti e non devono essere codificati.
Pertanto, la bit rate finale è data da:

- *Canale Y*: 352 sample/linea x 240 linee/frame x 24 frame/sec x 8 bit/sample (16 Mbit/sec)
- *Canali U,V*: 176 sample/linea x 120 linee/frame x 24 frame/sec x 8 bit/sample (8 Mbit/sec)

Inoltre, il White Book per il compact disc stabilisce un bit rate standard di 1,15 Mbit/sec. La vera compressione che si riesce ad ottenere quindi con MPEG-1 è 24 Mbit/sec:1.15 Mbit/sec, ovvero 21:1.
Una volta ridotta la grandezza dell'immagine ed il suo chroma ratio, ha luogo la suddivisione in frame di tipo I, P o B. Sarà il codificatore a decidere quale sequenza meglio si adatta ad una efficiente compressione del filmato. Ad esempio, se due frame successivi contengono immagini completamente diverse (a causa di uno "stacco" sulla scena), la codifica P non risulterà possibile, e si opterà per un frame di tipo B o, nella peggiore delle ipotesi, di tipo I. Quest'ultima codifica, infatti, non tiene conto né del passato, né del futuro e bene si presta a soluzioni "estreme", anche se a discapito di una grande compressione.
In generale, si potrebbe organizzare una sequenza di frame diversa a seconda delle circostanze: per esempio, un filmato basato su una codifica di tipo I I I I I … permetterebbe un ottimo FF/FR/RA con compressione molto bassa; una codifica I P P P P P P …, invece, farebbe diminuire il RA ma aumenterebbe la compressione; infine, una codifica I B B P B B P B B P B B I B B P… raggiunge un ottimo grado di compressione con un ragionevole FF/FR/RA, ma per esempio non è adatto a servizi di videotelefonia in tempo reale a causa del ritardo introdotto dalla codifica. Si noti la presenza del frame I non solo all'inizio dello stream, ma anche ogni 12 frame. Infatti, basarsi solo sui frame B e P è rischioso perché un piccolo errore di codifica si propagherebbe lungo tutto il film,

essendo una codifica DPCM, e quindi, per motivi di sicurezza, i frame I spezzano questa pericolosa catena.

Una volta codificato il segnale, il decoder dovrà leggere i frame in maniera diversa per poterli decodificare correttamente: i frame P infatti verranno letti prima dei B in modo da permetterne la decodifica secondo questa sequenza:

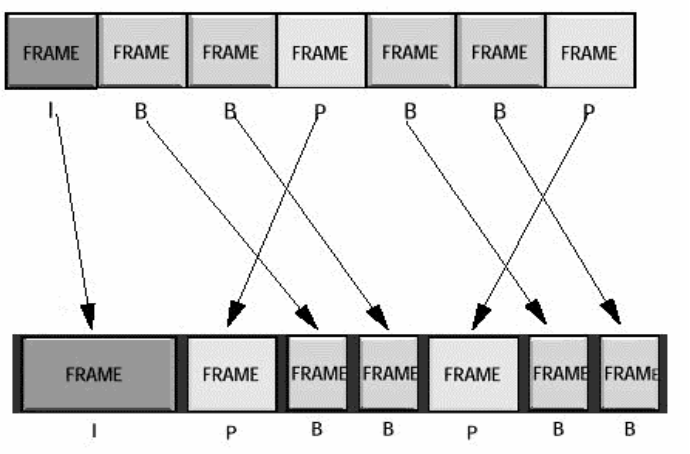


Figura 7: Sequenza dei frame I,P,B in MPEG

Un'importante caratteristica supportata da MPEG-1 è la capacità di ridurre il bit rate a seconda delle situazioni e delle applicazioni tramite variazione della codifica dei coefficienti DCT. E' anche possibile variare il tipo di quantizzazione da macroblocco a macroblocco, gestendo così in maniera ottimale la qualità dell'immagine. Inoltre, MPEG-1 fornisce bit rate costanti e variabili per la registrazione e riproduzione di video compressi. A tale scopo, esiste un buffer video (VB) con ingresso variabile ed uscita costante per massimizzare la qualità e la fluidità dell'immagine. La grandezza di questo buffer è programmabile anche se MPEG ha stabilito un valore minimo che deve essere supportato da qualunque decoder.

2.3 LA CODIFICA AUDIO IN MPEG-1: ASPETTI GENERALI

Lo standard MPEG-1 stabilisce un bit rate massimo di 256 Kbit/sec in totale per due canali audio con la stessa qualità di un CD musicale, il cui bit rate teorico è di 1,41 Mbit/sec (dati da 44100 Hz * 2 * 16 bit); a causa della sincronizzazione e del controllo degli errori, però, la rappresentazione di ogni campione a 16 bit occupa 49 bit e quindi il bit rate vero è di 4,32 Mbit/sec. Pertanto, la compressione minima da CD audio è dell'ordine di 17:1, la compressione da file per PC si aggira su 6:1.

Le tecnologie che permettono una tale compressione sono quattro: il *perceptual coding*, il *frequency-domain coding*, il *window switching* ed il *dynamic bit allocation*[Noll97][Otholit][Steinmetz].

2.3.1 IL "PERCEPTUAL CODING"

L'orecchio umano, tramite la membrana, analizza e filtra il suono ricevuto ed è capace di sentire frequenze comprese fra i 20Hz ed i 20Khz, con maggiore sensibilità tra i 2 ed i 4 kHz. Lo spettro di potenza non è rappresentato in una scala di frequenze lineari, ma in bande limitate in frequenze diverse chiamate *bande critiche.* L'orecchio umano può dunque essere definito in prima approssimazione come un banco di filtri passabanda caratterizzato da filtri sovrapposti tra loro con

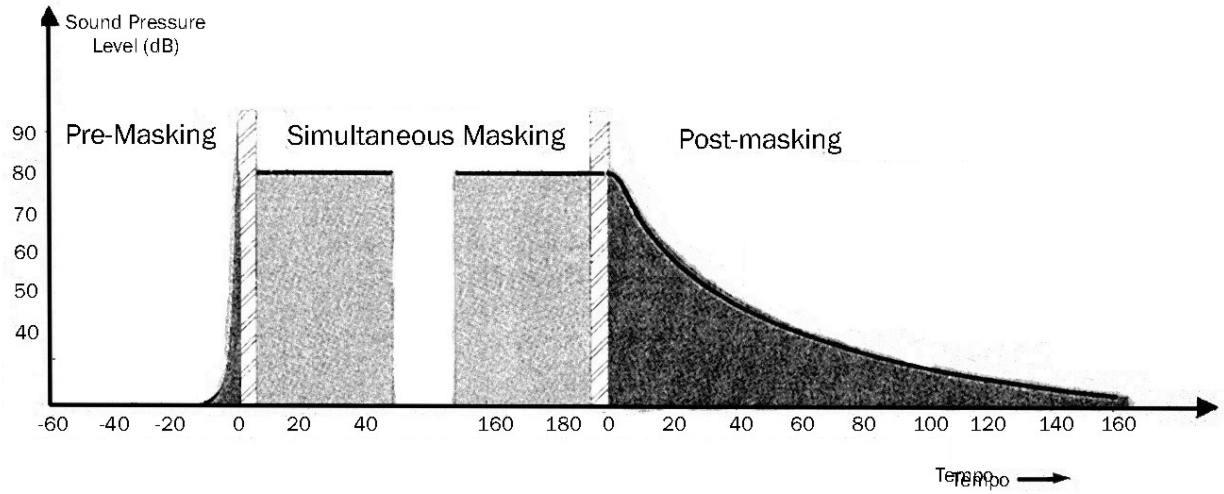


Figura 8: Simultaneous masking

larghezza di banda compresa fra 50 e 100 Hz per segnali sotto i 500 Hz, e con larghezza di banda fino a 5 kHz per segnali ad alta frequenza. Sono state prese in considerazione 26 bande critiche per frequenze fino a 24 kHz.

Un fenomeno importante dell'orecchio umano è il mascheramento in frequenza (*simultaneous masking*), che rende inudibile un segnale a basso livello (*mascherato*) se in contemporanea è presente, un segnale più forte (*mascherante*) ad una frequenza vicina. Quest'ultimo segnale genera una *soglia di maschera* che cancella tutti i segnali al di sotto di un certo valore (vedi figura 8). La soglia varia con il tempo, e dipende dal livello di pressione del suono, dalla frequenza del mascherante, e dalle caratteristiche del mascherato e del mascherante. La pendenza di questa soglia è più ripida verso le basse frequenze: questo significa che le frequenze più alte vengono mascherate

più facilmente. Se i mascheranti sono più di uno, si genera una *soglia globale di maschera* data dalla sovrapposizione delle varie soglie.

In mancanza di un mascherante, esiste comunque una soglia minima al di sotto della quale nessun suono è percepibile (figura 9).

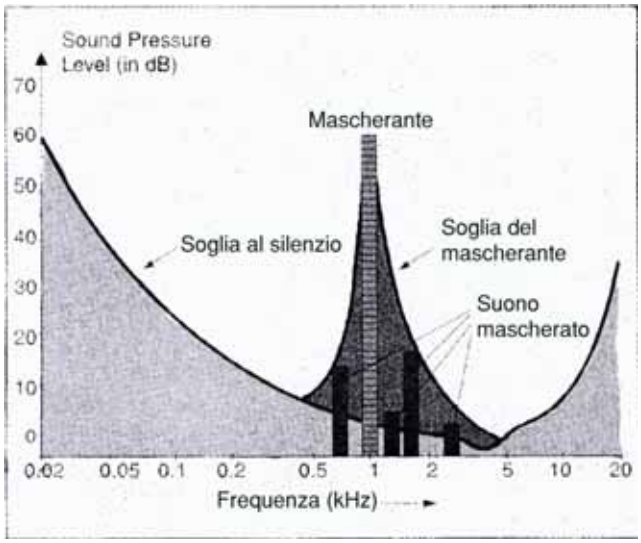


Figura 9: Soglia di maschera

Un'altra caratteristica interessante è il *mascheramento temporale*, che ha luogo quando, dopo un forte suono, ne segue un altro a breve distanza di tempo. In questo caso, anche se il mascherante precede il mascherato, è possibile che il suono più debole non venga percepito. Questo è importante per lo studio dei fenomeni di *pre-eco*, causati generalmente da "attacchi" nei brani musicali che introducono errori di quantizzazione. Grazie al fenomeno del mascheramento temporale, questi problemi possono essere risolti o quantomeno ridotti.

La distanza tra il livello del mascherante e la soglia di maschera viene chiamato *rapporto segnale maschera* (*signal-to-mask ratio, SMR*), mentre il rapporto segnale rumore (SNR) è dato dalla differenza tra il livello del mascherante ed il livello di rumore generato dalla quantizzazione ad *m* bit del segnale audio. La differenza tra l'SMR(m) e l'SNR viene definita come *rapporto rumore maschera* (*noise-to-mask ratio, NMR*) ed ogni volta che questo valore è negativo, il rumore di quantizzazione e le distorsioni del segnale non saranno udibili.

2.3.2 FREQUENCY-DOMAIN CODING

I codificatori di questo tipo, per ridurre il bit rate dei dati trasmessi, devono togliere la ridondanza (ovvero tutto ciò che non apporta nessuna informazione utile a livello oggettivo) e l'irrilevanza dei segnali audio originali (ovvero tutti i segnali sotto soglia e con frequenza soggettvamente inudibile all'uomo). Per fare ciò, lo spettro sorgente viene suddiviso in bande di frequenza per generare così componenti spettrali quasi incorrelate tra loro e quantizzarle separatamente. Esistono due metodi diversi di codifica: la *transform coding* (TC) e la *subband coding* (SBC).

Nella *SBC*, il segnale sorgente viene inviato ad un banco di filtri formati da M filtri passabanda contigui in frequenza in modo da poter poi ricombinare additivamente i segnali e riprodurre il suono

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

originale. All'uscita di ogni filtro avviene una decimazione di un fattore uguale a M, creando così un numero di campioni nelle sottobande uguale a quello del segnale sorgente. Nel ricevitore, il sampling rate di ogni sottobanda viene aumentato fino a raggiungere quello del segnale sorgente. Nel *TC*, un blocco di campioni in ingresso viene trasformato in un gruppo di coefficienti, quasi incorrelati fra di loro, che vengono poi quantizzati e trasmessi in digitale dal decoder. Tipiche funzioni di trasformazione sono le FFT, le DCT o le MDCT (Modified DCT).
La tecnica SBC può essere unita alla TC creando dei filtri di banchi ibridi (usati per esempio nella codifica del Layer III), mentre per i Layer I e II si usa solo la SBC.

2.3.3 WINDOW SWITCHING

Uno dei problemi precedentemente accennati è quello del *pre-eco* (figura 10), dovuto agli "attacchi" in un suono (come per esempio una percussione dopo un silenzio). Sia il modello di tipo SBC, sia quello TC distribuiscono degli errori nel blocco da codificare, per cui il suono ricostruito non avrà più una parte di silenzio iniziale, bensì del rumore di sottofondo paragonabile a quello dei nastri analogici copiati. Il fenomeno del temporal masking ci aiuta, ma se il blocco da codificare è grande, i risultati non sono molto buoni; più il blocco è piccolo, invece, e maggiore è la qualità del suono. Una soluzione consiste allora nell'avere blocchi di dimensioni diverse: piccoli per i fenomeni di pre-echo, e grandi negli altri casi. Questa tecnica prende il nome di *window switching.*

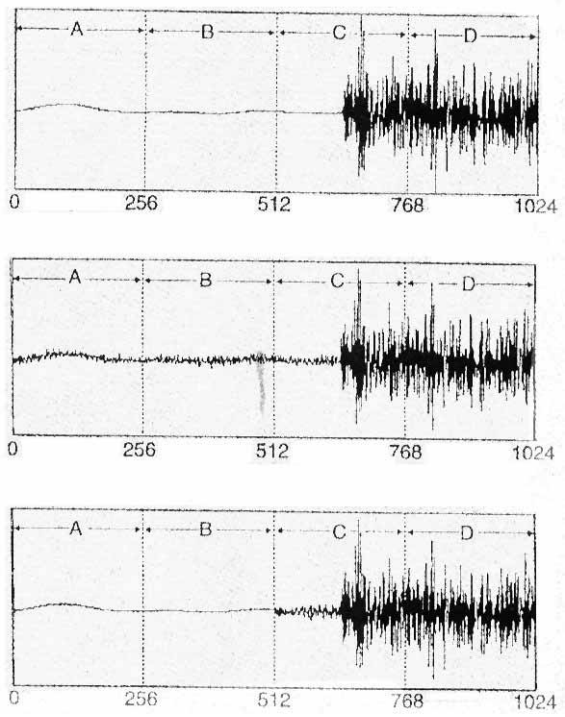


Figura 10: In alto: suono originale; al centro: fenomeno di pre-eco; in basso: migliorie al pre-eco tramite window switching

2.4 LA CODIFICA AUDIO MPEG-1

Lo standard MPEG-1 consiste di tre livelli (Layer I, II, III) di complessità crescente e compatibili verso il basso. Vengono supportati quattro modalità diverse di suono: mono, stereo, duale (utile per i programmi bilingue) e joint stereo. Quest'ultima opzione codifica come un unico segnale mono,

dato dalla media dei due canali, le basse frequenze, lasciando le alte in stereo. Un'altra opzione, presente solo nel Layer III, è la mono/stereo, dove vengono codificati sia la somma, sia la differenza dei canali destro e sinistro.

2.4.1 LAYER I

I Layer I e II mappano il segnale di input in 32 sottobande tramite filtri passabanda equamente spaziati; la mappatura in frequenza usa un filtro polifase con 512 coefficienti per evitare l'aliasing in fase di ricostruzione, la complessità di calcolo ed i ritardi. Il lato negativo è che a basse frequenze, a causa della spazialità costante, una singola sottobanda copre molte bande critiche adiacenti a causa della propria larghezza. Infatti, ad una frequenza di campionamento di 48kHz ogni banda ha una larghezza massima di $\frac{24000}{32} = 750Hz$, mentre ricordiamo che alle basse frequenze ogni banda critica ha una larghezza di banda di circa 100 Hz. Il risultato della FFT serve a studiare dettagliatamente il segnale di ingresso per ricavare i mascheranti tonali (sinusoidali) e non tonali (rumore). Dato che per ogni sottobanda ogni mascherante produce una soglia di maschera, si genera una soglia di maschera globale che, confrontata con il massimo segnale in ingresso, fornisce l'SMR. Ciò che ho appena descritto è il *modello psicoacustico*, e la sua complessità, non standardizzata da MPEG, rende i vari encoder più o meno lenti e performanti.
Successivamente, viene esaminata ogni sottobanda e per ognuna di esse viene calcolato il massimo valore dei campioni e quantizzato a 6 bit. Questo valore è detto *fattore di scala*. Dato che deve essere minimizzato l'NMR(dB), e, dato che questo è in funzione dei bit di campionamento m, per ogni sottobanda si calcola il valore di m necessario allocando uno o più bit (ogni bit in più introduce 6dB di rumore ma aumenta la compressione).
Ogni sottobanda infine contiene 12 campioni (che corrispondono a 8 ms di audio campionato a 48 kHz), e dato che abbiamo 32 sottobande, il segnale PCM in ingresso viene processato a blocchi di 384 campioni alla volta (ovvero 96 ms).

2.4.2 LAYER II

La mappatura avviene sempre in 32 sottobande, ma il filtro usa 1024 coefficienti. Questo implica una maggiore distinzione fra i mascheranti tonali e non tonali, proporzionalmente al tempo di calcolo.
La quantizzazione avviene sempre a 6 bit, ma anziché 12 campioni per sottobanda se ne usano 36 (3 blocchi consecutivi da 12) e la trasmissione dei fattori di scala viene ottimizzata: per ogni gruppo di 12 campioni viene generato il fattore di scala, e l'encoder ne usa uno, due o tutti e tre a seconda di quanto differiscono tra di loro. Questo in media fa sì che la bit rate di allocazione dei fattori di scala sia ridotta di circa la metà. Il numero di campioni nel Layer II è 1152, e quindi il segnale PCM viene processato a blocchi di 288 ms. A causa di questi motivi, il Layer II è lievemente migliore del suo predecessore.

2.4.3 LAYER III

Il Layer III introduce molte nuove caratteristiche, tra cui il banco di filtri ibrido e un controllo avanzato del pre-eco. Per avere migliori prestazioni, le 32 sottobande vengono ulteriormente divise in frequenza tramite una trasformata MDCT (modified DCT) a 18 punti con sovrapposizione al 50%. Questo fa sì che le sottobande totali siano $32 \cdot 18 = 576$ e quindi, per un segnale campionato a 48k Hz, la larghezza di banda è solo $\frac{24000}{576} = 41.67 Hz$, inferiore alla larghezza della più piccola banda critica. Inoltre, per aumentare il rapporto di compressione, la quantizzazione dei campioni non è uniforme, e gli stessi campioni vengono codificati tramite algoritmi di tipo run-length e Huffman.
Per ottimizzare l'NMR, viene impiegata una tecnica iterativa di *analisi per sintesi* che fa sì che il rumore di quantizzazione si trovi sempre sotto la soglia. Nella codifica di tipo Layer I e II, infatti, il rumore si trova solo *mediamente* sottosoglia, e non puntualmente come nel Layer III. Esiste infine una tecnica di buffer detta *bit reservoir* che assicura che il buffer in decodifica non sia mai saturo o vuoto, in modo da permettere una trasmissione dei dati a rate costante.

2.4.4 DECODIFICA DEL SEGNALE AUDIO

Nel caso del Layer I e II, la decodifica è immediata: le sottobande vengono ricostruite a partire dai 12 o 36 blocchi, dai fattori di scala e dai bit di blocco. Se il bit vale zero in una sottobanda, questa

non viene processata; alla fine, si mandano le 32 sottobande ad un banco di filtri di analisi che ricostruiscono il segnale in formato PCM a 16 bit.
In Layer III, la ricostruzione del segnale avviene in maniera un po' più complicata a causa del metodo iterativo di analisi per sintesi che deve essere ricostruito anche in fase di lettura.

2.4.5 QUALITÀ E CARATTERISTICHE DEI SEGNALI AUDIO

Tutti e tre i Layer supportano frequenze di campionamento di 32, 44.1 e 48 kHz. Le bit rate variano invece a seconda del layer:

- *Layer 1*: da 32kb/s (mono) a 448 kb/s (stereo) – Il target ISO è di 192 kb/s
- *Layer 2*: da 32 kb/s a 384 kb/s – il target ISO è di 128 kb/s ed è un compromesso tra qualità e complessità. E' attualmente in uso nei sistemi DAB (Digital Audio Broadcasting)
- *Layer 3*: da 32 kb/s a 320 kb/s – il target ISO è di 64 kb/s e viene usato in reti di telecomunicazioni e sistemi di annuncio vocale. La qualità audio è preservata a discapito della velocità di compressione.

La qualità oggettiva del segnale audio sopracitata venne testata da MPEG grazie all'aiuto di una sessantina di ascoltatori con esperienza nel settore; vennero svolti dieci test diversi in stereo sia con cuffie che con casse, e fu chiamato come organo supervisore la radio svedese. Ogni singolo test consisteva nell'ascoltare tre volte lo stesso pezzo di un brano, secondo la sequenza ABC, dove A era il segnale originale, B e C o C e B il segnale originale e codificato (*tecnica doppio cieco*). Né il gruppo di esperti né l'organizzatore dei test sapevano se il segnale codificato fosse C o B, ed ai soggetti veniva richiesto di decidere quale fosse il segnale originale e di assegnare un voto a quello codificato, variabile tra 5 e 1 secondo questo schema:

- 5.0 = suono trasparente (da assegnarsi al suono originale)
- 4.0 = differenza percettibile, ma non fastidiosa
- 3.0 = suono lievemente peggiore
- 2.0 = suono con poca qualità
- 1.0 = suono con pochissima qualità

Ad esempio, il Layer 2 ottenne un voto tra 2.1 e 2.6 a 64 kb/s per canale, mentre il Layer 3 ottenne tra 3.6 e 3.8. Dato che lo standard MPEG definisce il decoder e non l'encoder, esistono molti algoritmi diversi di compressione come per esempio *Fraunhofer*, *Blade*, o *Xing*, e molte migliorie sono state apportate a partire dal 1991.

2.5 APPLICAZIONE DELLO STANDARD MPEG-1 SECONDO IL WHITE BOOK

Lo scopo di MPEG-1 è codificare un film in un CD. Le definizioni specifiche del White Book sono le seguenti:

- *Codifica Audio:* Layer II
- *Frequenza di campionamento:* 44,1 kHz
- *Modo:* stereo, dual channel, intensity stereo
- *Formati permessi:*
  - 352 pixel/linea x 240 linee/frame x 29.97 frame/s (NTSC)
  - 352 pixel/linea x 240 linee/frame x 23.976 frame/s (NTSC film)
  - 352 pixel/linea x 288 linee/frame x 25 frame/s (PAL)
- *Massima bitrate:* 1.1519291 Mb/s
- Massima durata del film: 75'

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

3 MPEG-2

L'era del video digitale era nata con successo grazie a MPEG-1, e questo fece crescere l'interesse nelle industrie del settore verso immagini a più alta risoluzione ed a bit rate maggiore. Infatti, la codifica da CCIR-601 a SIF non era accettabile per la maggior parte dell'home entertainment, ed un convegno tenutosi in Giappone nel novembre 1991 considerava 4 Mbit/s un bit rate di buona qualità.

MPEG-1 forniva ottime prestazioni, ma la codifica avveniva ad un bit rate di soli 1,5 Mb/s! Nonostante comunque si fosse dimostrato un buon comportamento di MPEG-1 anche a maggiori risoluzioni e bit rate di quelle standard, il maggiore problema derivava dal fatto che questo protocollo non era adatto alla codifica di immagini televisive, e quindi interallacciate: semplicemente unendo il quadro pari e quello dispari, si osservavano problemi di fluidità, mentre la codifica separata dei due quadri era poco efficiente in quanto non teneva conto della ridondanza e quindi sprecava molta banda. Inoltre, MPEG-1 aveva come scopo la memorizzazione su supporto magnetico di immagini locali, e non prevedeva quindi un ambiente caratterizzato da errori in trasmissione.

Una delle maggiori richieste invece proveniva proprio dal mondo della televisione per una sempre maggior differenziazione dell'offerta al pubblico, con canali tematici e multilingua anche a pagamento. La banda occupata da un singolo canale video analogico, oltre alla minore qualità audio e video, costringeva i vari broadcaster a lanciare nuovi satelliti per avere nuovi transponder, con conseguenti costi molto elevati; per non parlare dell'impossibilità di funzioni quali PPV (Pay Per View) o N-VOD (Near-Video On Demand).

A partire da Giugno 1990, dunque, con la collaborazione dell'ITU-T SG 15 Experts Group for ATM Video Coding, si iniziò lo studio di MPEG-2 per risolvere le limitazioni di MPEG-1, e nel 1995 nacque il nuovo standard ISO/IEC 13818, dal titolo "*Information technology – Generic coding of moving pictures and associated audio"*, suddiviso in 10 parti:

- *Systems*: Riguarda sia la sintassi per il trasporto dei pacchetti audio e video su canali digitali e DSM, sia la sintassi necessaria alla sincronizzazione video ed audio;
- *Video*: Descrive la sintassi e la semantica della codifica video;
- *Audio*: Descrive la sintassi e la semantica della codifica audio;

- *Conformance*: Definisce le linee guida per determinare se i bitstream generati sono o meno conformi allo standard.
- *Software Simulation*: Contiene un esempio di encoder scritto in ANSI C ed un decoder conforme alle specifiche sia per il video che per l'audio.
- *Digital Storage Medium Command and Control (DSM-CC)*: Specifica un insieme di protocolli atti a fornire funzionalità di controllo e di operazioni utili alla gestione degli stream MPEG. Ad esempio, possono essere comandi come Fermo Immagine, Riavvolgimento veloce, GoTo, ma anche funzioni quali VOD, Fast Internet (connessione ad Internet via satellite) ecc. Il protocollo deve funzionare sia nei sistemi stand-alone sia in quelli client-server. Per maggiori informazioni, si consulti la Bibliografia [ISO-N1559].
- *Non Backwards Compatible Audio (NBC)*: Rappresenta la parte non compatibile verso il basso di MPEG-2 per una migliore qualità audio, come la gestione del dolby surround e del sistema 5.1.
- *10-bit Video Extension*: Questa parte, inizialmente studiata per convertire i segnali video da 8 a 10 bit (come nello standard CCIR-601), venne poi abbandonata per la mancanza di interesse da parte delle industrie.
- *Real Time Interface (RTI)*: Definisce la sintassi per segnali di controllo tra set-top box ed end-server al fine di ottenere informazioni in tempo reale (menu interattivo, commutazione di lingua, VOD, PPV etc.) e per creare le guide elettroniche (EPG, Electronic Program Guide).
- *Conformance Testing of DSM-CC:* Definisce suggerimenti e linee guida per controllare l'effettiva conformità dei protocollo DSM-CC.

3.1 CODIFICA VIDEO IN MPEG-2

Possiamo considerare la codifica video di MPEG-2 come un ampliamento di quella già esistente in MPEG-1, anche perché venne creata con lo scopo di essere compatibile verso il basso: ogni decoder di tipo MPEG-2, dunque, può essere in grado di leggere stream in formato MPEG-1.
In generale, viene utilizzato lo stesso schema DCPM/DCT, suddivisione dell'immagine in macroblocchi, vettori di moto e frame di tipo I, P, B.
Per permettere l'utilizzo del nuovo standard in varie applicazioni, MPEG-2 ha introdotto il concetto di *Profili* e *Livelli*, in grado di definire una riduzione della complessità in termini di risoluzione e bit rate in caso di ridotte capacità del decoder. Così, ad esempio, esiste un profilo di tipo *MAIN* con le

codifiche di frame I, P, B, ed uno di tipo *SIMPLE* che raggruppa solo i frame I e P. Tutte le possibili soluzioni sono illustrate nella seguente tabella[Sikora97][Sikora97a]

Tabella dei livelli	
HIGH	1920 sample/linea x 1152 linee/frame x 60 frame/s Codifica a 80 Mbit/s
HIGH 1440	1440 sample/linea x 1152 linee/frame x 60 frame/s Codifica a 60 Mbit/s
MAIN	720 sample/linea x 576 linee/frame x 30 frame/s Codifica a 15 Mbit/s
LOW	352 sample/linea x 288 linee/frame x 30 frame/s Codifica a 4 Mbit/s

Tabella 2 : Tabella dei livelli in MPEG-2

Tabella dei profili	
HIGH	Supporta le funzioni previste dal profilo Spatial Scalable con la previsione di supportare 3 livelli di codifica. Chroma ratio: 4:2:2
SPATIAL Scalable	Supporta le funzioni del profilo SNR Scalable ed aggiunge un algoritmo per la codifica spaziale (2 livelli sono permessi). Chroma ratio: 4:2:0
SNR Scalable	Supporta le funzioni del profilo MAIN con l'aggiunta dell'algoritmo di codifica SNR a 2 livelli. Chroma ratio 4:2:0
MAIN	Codifica non scalare per codificare segnali video tramite frame I, P, B. Chroma ratio 4:2:0
SIMPLE	Include le funzioni del livello MAIN ma non supporta frame di tipo B. Rappresentazione 4:2:0

Tabella 3: Tabella dei profili in MPEG-2

Come regola generale, ogni profilo definisce un nuovo insieme di algoritmi che si vanno ad aggiungere a quelli già presenti nel livello inferiore, mentre ogni livello specifica i margini dei parametri supportati dall'applicazione (come grandezza dell'immagine, frame rate e bit rate). Normalmente, è richiesto che le applicazioni televisive (come la TV digitale) siano conformi ad una codifica di tipo Main Profile e Main Level, ovvero codifica a 15 Mb/s con chroma ratio 4:2:0 e frame di tipo I, P e B. Questo tipo di specifica viene più brevemente chiamato MP@ML (*Main Profile at Main Level*).

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

3.1.1 CODIFICA DEI FRAME I

Un nuovo concetto introdotto da MPEG-2 è quello di campi e quadri (*field/frame pictures)* e di predizione su campi e quadri (*field/frame prediction*) in modo da poter codificare sia segnali video standard, sia interallacciati.
Nel primo caso, utile soprattutto per codificare i film, dove tutti i "pixel" vengono catturati nello stesso istante, i campi pari e dispari vengono codificati insieme come se si trattasse di un solo frame (*frame pictures*); nel caso di un segnale interallacciato, generato per esempio da videocamere digitali, i due campi vengono codificati separatamente perché distanziati nel tempo (*field pictures*): in successione, arriva prima quello dispari e poi quello pari. A questo punto, si applica la DCT sui macroblocchi tenendo conto non solo della correlazione all'interno del frame, ma anche di quella presente tra i due campi. E' interessante sottolineare il fatto che all'interno di una stessa sequenza video possono esserci contemporaneamente campi e quadri.

3.1.2 CODIFICA DEI FRAME P

La creazione di un field di tipo P può essere ottenuta tramite predizione di un field precedente, sia esso pari o dispari (*field prediction*); quella di un frame, invece, o da un frame precedente o da un field precedente (*frame prediction*), e la scelta può essere fatta macroblocco per macroblocco.
Dato che esistono varie possibilità di codifica, il decoder deve sapere, in fase di lettura, se usare il vettore di moto a partire da un frame, da un field pari o da uno dispari, e per questo viene codificato anche quale tipo di frame o field è stato usato per la codifica di tipo P.

3.1.3 MODALITÀ DI PREDIZIONE DEI FRAME/FIELD

Sono stati introdotti nuovi algoritmi di predizione, sia per i frame che per i field. Rinviando alla Bibliografia per ulteriori informazioni [mpeg1], le alternative sono:

- *Frame, Field, Dual Prime* (per i Frame);

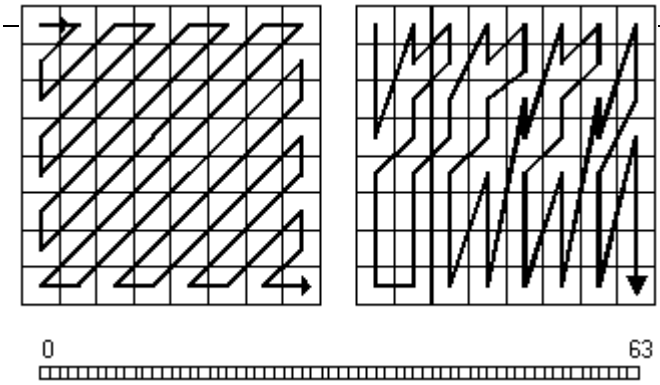


Figura 11: Alternate scan per una codifica ottimale delle immagini interallacciate

- *Field, 16x8, Dual Prime* (per i Field).

Una interessante novità è anche data dal fatto che i macroblocchi non sono necessariamente formati da un quadrato di 16x16 pixel, ma possono essere formati anche da due rettangoli di 16x8 pixel.
Inoltre, la codifica dei macroblocchi per i canali U,V ora si avvale non solo della tecnica zig-zag, utile per le immagini progressive (non interallacciate), ma anche della *alternate scan*, utile per codificare segnali interallacciati (vedi figura 11).

3.1.4 CHROMA RATIO

Anche il formato Y:U:V ha subito delle modifiche. La sintassi del profilo *MAIN* è limitata a 4:2:0, già implementato in MPEG-1. Sono stati introdotti il formato 4:2:2 (*Studio Profile*), generalmente adatto ad apparati di editing video professionale, ed il 4:4:4, utile per il Computer Graphics ed al momento non definito in alcun profilo. E' presente inoltre una sintassi per convertire immagini in formato 16:9 su uno schermo 4:3.

3.1.5 SCALABILITÀ

Un concetto molto importante introdotto da MPEG-2 è la scalabilità. L'idea è quella di fornire interoperabilità fra i diversi servizi e supportare anche i ricevitori con diverse capacità di risoluzione. In questo modo, in fase di decodifica verrà scelta la risoluzione più alta o più bassa a seconda delle capacità hardware, e questo rende compatibili sistemi precedentemente incompatibili, come per esempio una ricezione dati su HDTV (High Digital TeleVision) a partire da una trasmissione SDTV (Standard Definition TeleVision).
Affinché ciò sia possibile, in fase di codifica vengono forniti due livelli: uno a bassa risoluzione ottenuto da un downscaling del segnale originario, uno con la codifica differenziale dei due segnali. Se il decoder non è in grado di ricostruire tutto il segnale, decodificherà solo quello a bassa risoluzione, altrimenti potrà fare l'upscaling del segnale a bassa risoluzione e tramite decodifica DPCM dei due livelli ottenere il segnale originario. Il downscaling può essere sia spaziale (riferito

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

cioè allo stesso frame) che temporale (riferito a frame successivi), a seconda delle funzionalità richieste.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

3.2 DIFFERENZE FRA MPEG-1 E MPEG-2

Nella seguente tabella vengono riassunte le principali differenze di codifica tra gli standard MPEG-1 e MPEG-2:

	MPEG-1	MPEG-2	
Standardizzazione	1992	1994	
Applicazione primaria	Video digitale su CD-ROM	TV Digitale	HDTV
Risoluzione spaziale	Formato SIF (352x288)	Formato TV (576*720)	Formato 4*TV (1152*1440)
Risoluzione temporale	25-30 frame/s	50/60 campi/s	100-120 campi/s
Bitrate	1.5 Mbit/s	4 Mbit/s	~20 Mbit/s
Qualità	VHS	NTSC (352x480x24 Hz progressivo) PAL (544x480x30 Hz interallacciato)	
Rapporto di compressione su PCM	20-30 : 1	30-40 : 1	30-40 : 1

Tabella 4: Differenze tra MPEG-1 e MPEG-2

3.3 IL TRASPORTO DELLE INFORMAZIONI IN MPEG-2

Il sistema di trasporto dei dati del formato MPEG-2 (TS, Transport Stream) è formato da un frame di 188 byte, come si può vedere dalla figura 12. Questa grandezza non è casuale perché corrisponde ad 8 celle di tipo ATM con 8 byte di overhead (associati all'AAL ATM).
I campi più importanti sono[hp_2][erg]:

- *Sync Byte*: permette l'invio di un pacchetto TS ed abilita il sincronismo nella trasmissione;
- *PID (Packet Identifier)*: contiene un valore che punta ad una tabella (*PSI*) in grado di identificare il tipo di programma, i pacchetti audio, video e dati , la frequenza del transponder, il tipo di servizio (free o a pagamento) ecc.
- *PES (Packetised Elementary Stream)*: contiene tutti i frame (I, P, B) che costituiscono il filmato, nonché il tipo di quantizzazione, la grandezza dei macroblocchi, il tipo di vettori di moto, la grandezza dell'immagine e così via.

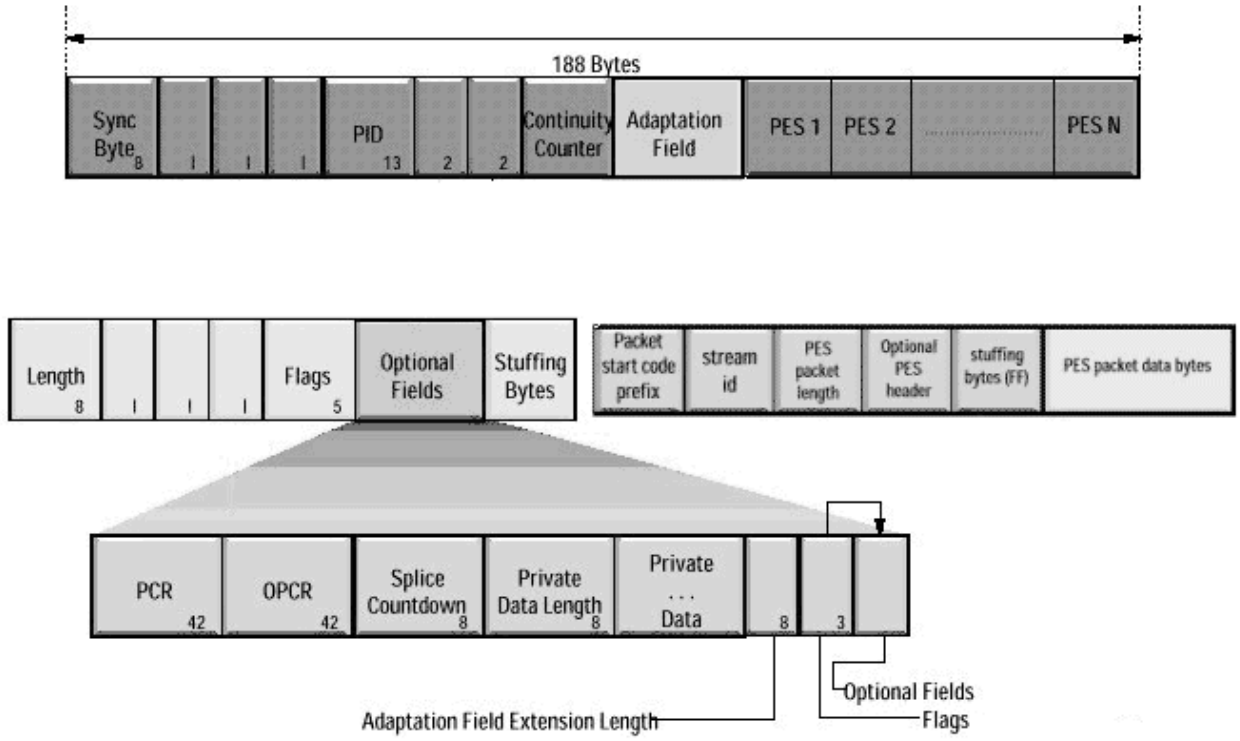


Figura 12: Transport Strean: Frame di MPEG-2

3.3.1 PROGRAM SERVICE INFORMATIONS (PSI)

Dopo aver multiplato i canali audio e video, il decoder deve anche sapere come ricostruire il solo segnale audio e video che ci interessa. Per questo è stato introdotto il campo PID, che punta ad una tabella comprendente tutte le informazioni necessarie[hp][hp_2][erg].

Saputo quale programma decodificare (diciamo “Program 1”), il decoder cerca, nella PAT (Program Association Table – Tabella di associazione dei programmi), il PID corrispondente a Program 1 (nel nostro esempio, il PID è 22). Questo punta ad un’altra tabella, detta PMT (Program Map Table – Tabella di mappatura del programma), che indica tutti i PID di tipo audio e video che servono per la riscostruzione dei PES, la temporizzazione dei frame ed eventuali restrizioni sul programma (se per esempio è un canale a pagamento).

Nella PAT sono presenti anche dei PID speciali che puntano a particolari tabelle contenenti informazioni molto utili alla creazione di guide elettroniche per l’utente. Queste tabelle non sono state standardizzate da MPEG, ma sono state impiegate da DVB (Digital Video Broadcasting) ed ATSC (Advanced Television Systems Commitee). Qui di seguito sono riportate per completezza le principali tabelle adottate da DVB:

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

- EIT (Event Information Table – Tabella di informazione eventi) : Segnala all'utente di quale tipo di programma si tratta (sport, musica, documentari) ed un'eventuale descrizione dello stesso
- SDT (Service Description Table – Tabella di descrizione del servizio): Fornisce il nome del canale ed altre informazioni
- BAT (Bouquet Association Table – Tabella di associazione del bouquet): Raggruppa canali tematici
- NIT (Network Information Table – Tabella di informazione del fornitore): Contiene dati quali il numero del trasnponder, la frequenza, la polarizzazione ecc.
- TDT (Time and Date Table – Tabella Data e Ora): fornisce la data e l'ora corrente

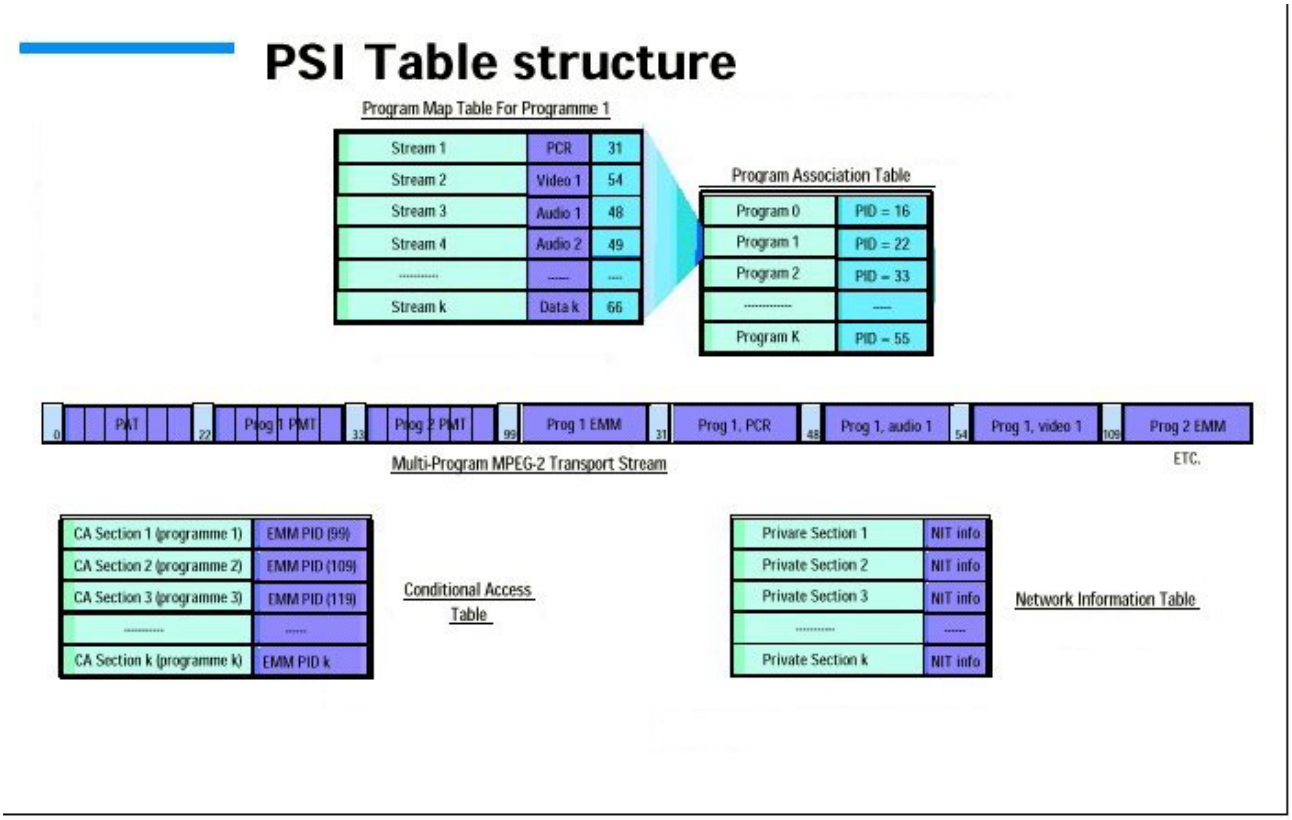


Figura 13: Struttura della tabella PSI

La figura 13 chiarisce e riassume i concetti finora descritti.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

4 GESTIONE DELL'ERRORE IN TRASMISSIONE/RICEZIONE

La tecnica di trasporto di MPEG-2 è efficiente, perché permette di multiplare di molti PES. Il lato negativo della medaglia è che un sistema di questo tipo è molto fragile a causa della voluta economicità del sistema di trasporto. Infatti,

- La corruzione di un header di tipo PES fa perdere i frame dell'immagine;
- Un errore sul TS fa perdere molti PES;
- Un errore sul PID può far perdere varie funzionalità quali la PPV e/o la guida elettronica;
- Un errore sul PCR può far perdere il sincronismo nell'immagine.

In caso di errore, dato che nel TS non esistono header associati a blocchi e macroblocchi, il decoder normalmente salta i macroblocchi fino a raggiungere l'header dello slice successivo (uno slice è formato da 16 righe). Il risultato è mostrato nelle figure 14 e 15. Per migliorare la situazione, sono stati proposti alcuni algoritmi interessanti (vedi ad esempio [LeBuhan96]) che però esulano dagli scopi di questa trattazione.

4.1 CODIFICA AUDIO IN MPEG-2

4.1.1 COMPATIBILITÀ CON MPEG-1

Figura 14: Codifica corretta dell'immagine…

Pensato per applicazioni a banda stretta, in MPEG-2 sono state definite frequenze di campionamento più basse (16, 22.05, 24 kHz) in modo da sfruttare bande a 64 kb/s o meno. La sintassi, semantica e tecnica di codifica sono identiche a quelle di MPEG-1, a parte alcune lievi modifiche, e dunque ci si può ancora basare sui Layer I, II e III. L'estensione a frequenze più basse porta a compressioni maggiori. L'utilizzo della

codifica Layer III fornisce i migliori risultati[Noll97].

Un passo in avanti nella codifica dell'audio digitale è la rappresentazione di audio in formato multicanale, per poter così ricreare suono realistico in sistemi audiovisivi incluso videoconferenza e soprattutto cinema. Inoltre, questo tipo di canali possono fornire un sistema multilingue da impiegarsi ad esempio nella proiezione di film in voli transcontinentali.

Figura 15: …e decodifica errata in MPEG-2

L'ITU-T ed altri hanno raccomandato una configurazione a cinque canali, chiamata anche *3/2 stereo*, data da un singolo canale destro (R), uno sinistro (L), uno centrale (C) e due di surround posti posteriormente di lato (LS e RS). Inoltre, per sfruttare appieno la capacità dell'HDTV, può essere aggiunto un canale per il subwoofer, creando una configurazione cosiddetta Sistema 5.1 (*5.1 System*).

Come è stato detto, lo standard MPEG-2 è compatibile verso il basso, e per far sì che un lettore MPEG-1 possa decodificare correttamente i segnali audio, i cinque canali vengono mixati (*matrixing*) per ottenerne solo due (L0 e R0). Questi ultimi vengono trasmessi nel frame audio usato da MPEG-1, mentre gli altri tre (C,LS,RS) vengono trasmessi in una parte non usata dal frame MPEG-1 (vedi figura 16).

In questo modo, un decoder di prima generazione decodificherà solo i canali L0 ed R0; gli altri

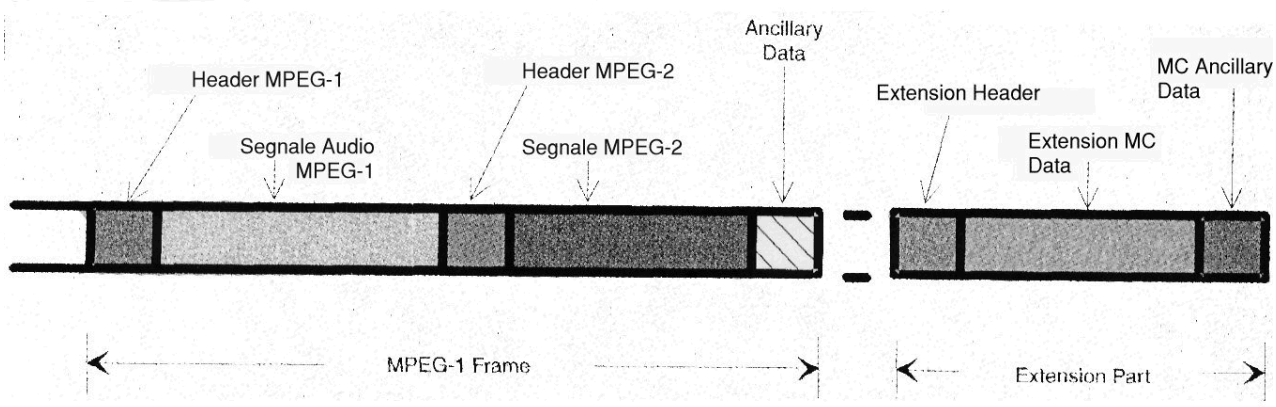


Figura 16: Inserimento della codifica C, LS, RS nella trama non usata di MPEG-1

opereranno un processo di *dematrixing* recuperando i 5 canali originali.
I lati negativi di questa tecnica sono fondamentalmente due: non si ottiene una grande compressione audio (infatti, i canali L0 ed R0 sono funzione dei 5 canali, quindi viene trasmessa due volte la stessa informazione); in fase di *dematrixing*, a causa del perceptual coding ci possono essere dei problemi sulla soglia di maschera con conseguente rumore di quantizzazione [Kate92].

4.1.2 NON BACKWARDS COMPATIBILE AUDIO (NBC) – CODIFICA AAC

Esiste un secondo standard audio non compatibile con MPEG-1, dal nome AAC (*Advanceed Audio Coding*). Non è più necessario dunque il dematrixing e tutti i potenziali problemi svaniscono. La codifica prevede un banco di filtri dato da una MDCT a 1024 punti, ed il modello psicoacustico è quello usato in MPEG-1. Viene anche impiegato un predittore del secondo ordine di tipo adattativo in retroazione che migliora la codifica. Infatti, il predittore riduce la bit rate per la codifica di campioni di sottobande successive in una data sottobanda, e basa la propria predizione sullo spettro quantizzato del blocco precedente. Una funzione iterativa controlla inoltre la quantizzazione in modo da mantenere il rumore di quantizzazione sempre sotto la soglia di maschera in tutte le bande critiche.
Dato che il procedimento è di notevole complessità, esistono tre profili diversi: il *main profile* offre la qualità più alta, il *low complexity profile* lavora senza predizione, con notevole riduzione di calcoli, ed il *sample-rate scalable profile* offre la complessità minore usando un banco di filtri ibrido.
L'MPEG-2 AAC supporta varie configurazioni, come per esempio mono, stereo, 5.1 system; il bit rate varia tra 320 e 384 kb/s totali, ovvero da 64 a 76.8 Kb/s per canale.
Come nel caso di MPEG-1, anche per l'MPEG-2 AAC sono stati fatti dei test soggettivi in maniera indipendente sia presso la Deutsche Telekom, sia presso la BBC con codecs operanti a 256, 320, 384 kb/s. Il voto finale è stato un netto 5.0 (suono indistinguibile dall'originale) nel caso di codifica a 64 Kb/s mono (128 Kb/s stereo), anche se il risultato dipendeva in parte dal tipo di musica codificata.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

5 MPEG-4

Dopo lo straordinario successo della televisione digitale, e sullo sfondo di una rapida convergenza delle industrie di telecomunicazioni, computer e cinema, MPEG iniziò la standardizzazione di MPEG-4 nel 1994. L'obiettivo era specificare i tool e gli algoritmi per ottenere compressione e rappresentazione flessibile di dati audiovisivi, in modo da soddisfare le esigenze di futuri servizi ed applicazioni multimediali. In particolare, MPEG-4 doveva assolvere i seguenti scopi[Chiariglione99][Schaefer98]:

- *Accessibilità universale e robustezza in ambienti fortemente soggetti ad errori*: Anche se lo standard MPEG-4 deve essere indipendente dal tipo di trasmissione, gli algoritmi ed i tool devono essere progettati anche per sistemi ad alto tasso di errore, quali ad esempio le comunicazioni mobili e le trasmissioni via Internet a bassa banda.
- *Alta funzionalità interattiva*: Le future applicazioni multimediali dovranno consentire la manipolazione di dati audiovisivi da parte dell'utente, e la ricerca del "contenuto" del materiale audiovisivo in una maniera molto flessibile.
- *Codifica di dati sintetici e naturali*: Le trasmissioni della prossima generazione vedranno la contemporanea presenza di immagini naturali e generate dal computer; per questo, MPEG-4 si propone di studiare sia la codifica di dati naturali sia quella di dati sintetici.
- *Efficienza di compressione*: La sempre maggiore compressione dei dati senza compromettere di molto la qualità è un obiettivo di primaria importanza per MPEG. Bande inferiori a 64 Kb/s devono essere supportate da questo nuovo standard: tipicamente, da 5 a 64 Kb/s per applicazioni su cellulare o telefono fisso; da 4 a 15 Mb/s per applicazioni nel campo televisivo/cinematografico.

Sono state studiate e create 4 versioni di MPEG-4. La prima è diventata standard nell'ottobre 1998; la seconda nel dicembre 1999. Al momento, sono allo studio le versioni 3 e 4. Il nome ufficiale è *ISO/IEC 14496 Coding of Audiovisual Objects* e consta di sei parti[ISO-N1909]:

- *Systems*: Descrive la sincronizzazione e la composizione di multipli oggetti audiovisivi (AV), e fornisce primitive per la composizione di scene 2D e 3D; gestisce il testo in modalità multilingue, l'indipendenza del canale di trasporto ed i meccanismi di controllo di errore.
- *Visual*: Gestisce la codifica video naturale e sintetico;

- *Audio*: Gestisce la codifica audio naturale e sintetica. E' composto dalle seguenti sottoparti:
  - *Parametric Coding*: Studia la codifica di tipo parametrico (tra 2 e 6 Kb/s);
  - *CELP Coding*: Studia la codifica di tipo CELP (Code Excited Linear Predictive) per una larghezza di banda compresa tra i 6 ed i 24 Kb/s;
  - *Time/Frequency Coding*: Studia la codifica per le bit rate tra 24 e 64 Kb/s;
  - *Structured Audio*: Implementa la codifica e la riproduzione di audio sintetico;
  - *Text To Speech*: Si occupa di convertire del testo in fonemi e parole, e della sincronizzazione delle stesse con il labiale;
- *Conformance Testing*: Questa parte specifica come verificare, tramite linee guida e procedure, che i bitstream ed i decoder seguano gli standard specificati nelle parti 1, 2 e 3 dello standard ISO/IEC 14496; specifica inoltre, per la parte 6 dello stesso standard, come verificare il trasporto dei vari bitstream attraverso le tecnologie già esistenti. Gli encoder vengono invece definiti appartenenti al suddetto standard se sono in grado di generare bitstream conformi a MPEG-4. Per ulteriori dettagli, si consulti la Bibliografia[ISO-N2804].
- *Reference Software*: E' l'implementazione software dello standard MPEG-4. Comprende vari bitstream in formato MPEG-4, riproduttori e decodificatori di bitstream, e tutti i sorgenti in ANSI C[ISO-N2805].
- *DMIF (Delivery Multimedia Integration Framework)*: E' un'interfaccia trasparente ai diversi protocolli di trasmissione punto-punto, ed ai diversi canali di trasmissione (IP, ATM, PSTN, mobile, ISDN); viene anche utilizzato per controllare i diritti di proprietà di un determinato oggetto audiovisivo. Migliora ed espande le funzionalità di MPEG-2 DSM-CC.

Per permettere al maggior numero possibile di applicazioni di usare MPEG-4, sono stati creati inoltre vari *Profili*, ovvero dei sottoinsiemi di tool ed algoritmi, riprendendo ed ampliando il concetto già esistente in MPEG-2.

5.1 LIVELLO SYSTEMS

Analizziamo in dettaglio come l'informazione audiovisiva viene composta, compressa e multiplexata in uno o più bitstream, e come questi stream vengono poi demultiplexati, decompressi ed assemblati per ricostruire l'immagine di partenza.
Lo scopo di MPEG è quello di poter influenzare interattivamente la presentazione in fase di ricezione. A tal scopo, sono stati introdotti dei meccanismi che permettono l'integrazione di oggetti

naturali con quelli sintetici, e che rendono possibile il multiplexing e la sincronizzazione di vari oggetti audiovisivi (*audiovisual object, AVO*).

Per ogni AVO, si genera un flusso elementare (*elementary stream, ES)*. Tutti gli ES con la stessa qualità di servizio in trasmissione (ad esempio, bit rate massima, tasso d'errore ecc.) vengono raggruppati nel layer FlexMux (*Flexible Multiplexing)*, formando così stream di tipo FlexMux.

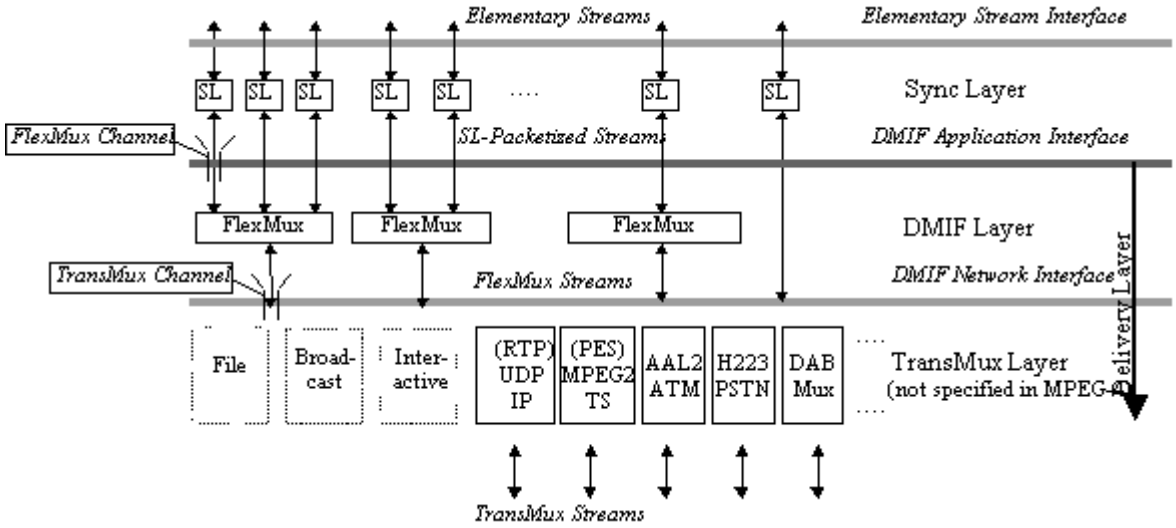


Figura 17: Interfacciamento di MPEG-4

Questo layer fornisce anche meccanismi di recupero dell'errore.

Il trasporto dei dati avviene tramite il TransMux (*Transport Multiplexing)*, non specificato in MPEG-4, anche se possono essere utilizzati protocolli di trasporto già esistenti, come IP (UDP), ATM (AAL5), o MPEG-2 (TS). L'interconnessione tra i diversi protocolli è a cura del DMIF (vedi figura 17).

Arrivati al decoder, gli stream FlexMux vengono demultiplexati, recuperando così gli ES, che

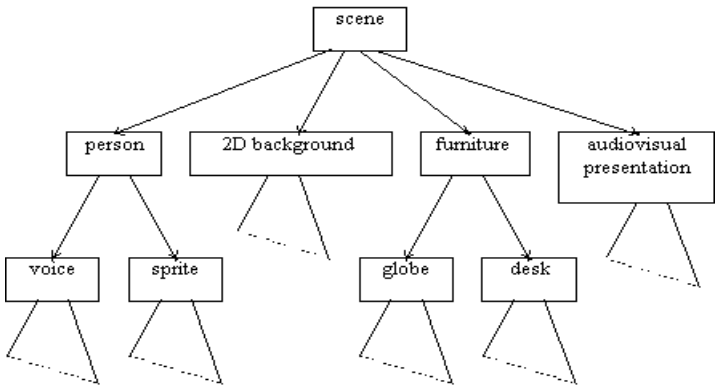


Figura 18: BIFS

ricostruiranno i diversi AVO. In aggiunta, vengono generate delle tabelle di descrizione dell'immagine (figura 18) utilizzando il formato BIFS (*Binary Format of Scene)*[ISO-N2995]. Tali tabelle possono essere manipolate a piacere dall'utente. Finalmente, la scena viene visualizzata nella sua interezza.

5.2 CODIFICA VIDEO

La codifica video in MPEG-4 comprende i seguenti argomenti:

- Video naturale:
  - Codifica di VOP (Video Object Planes) e del texture;
  - Trasmissione a banda stretta e larga;
  - Introduzione di Profili e Livelli;
  - Codifica scalabile;
- Video sintetico:
  - Animazione facciale;
  - Animazione del corpo.
- Codifica SNHC.

5.2.1 CODIFICA VIDEO NATURALE

Il nuovo approccio di MPEG-4 è quello di considerare un video non come un insieme di pixel, ma come un "contenuto" di diversi oggetti. Così, di un'immagine possiamo vedere lo sfondo, svariati oggetti e del testo. L'idea di base è quella di codificare ogni singolo oggetto, anziché una singola immagine, anche con tecniche diverse, per poi poterlo decodificare e ricostruire la scena iniziale. E' dunque necessario trasmettere anche le coordinate dei singoli oggetti per individuarne la posizione al momento della ricostruzione della scena; oltre a ciò, MPEG-4 aggiunge altri parametri, che permettono di variare la scala e la rotazione degli stessi oggetti, in modo da poter sì ricreare la sequenza originale, ma anche da variarne il contenuto ruotando degli oggetti, ingrandendone altri, togliendone od aggiungendone altri ancora; e tutto questo senza ulteriori codifiche e decodifiche, e dunque senza aumentare la complessità di calcolo.

A tal fine, MPEG-4 introduce il concetto di *video object planes* (VOP): ogni frame video è suddiviso in varie parti di dimensioni arbitrarie, ognuna delle quali rappresenta un oggetto fisico o un contenuto di particolare interesse (come ad esempio uno sfondo o un colore). In questo modo, a

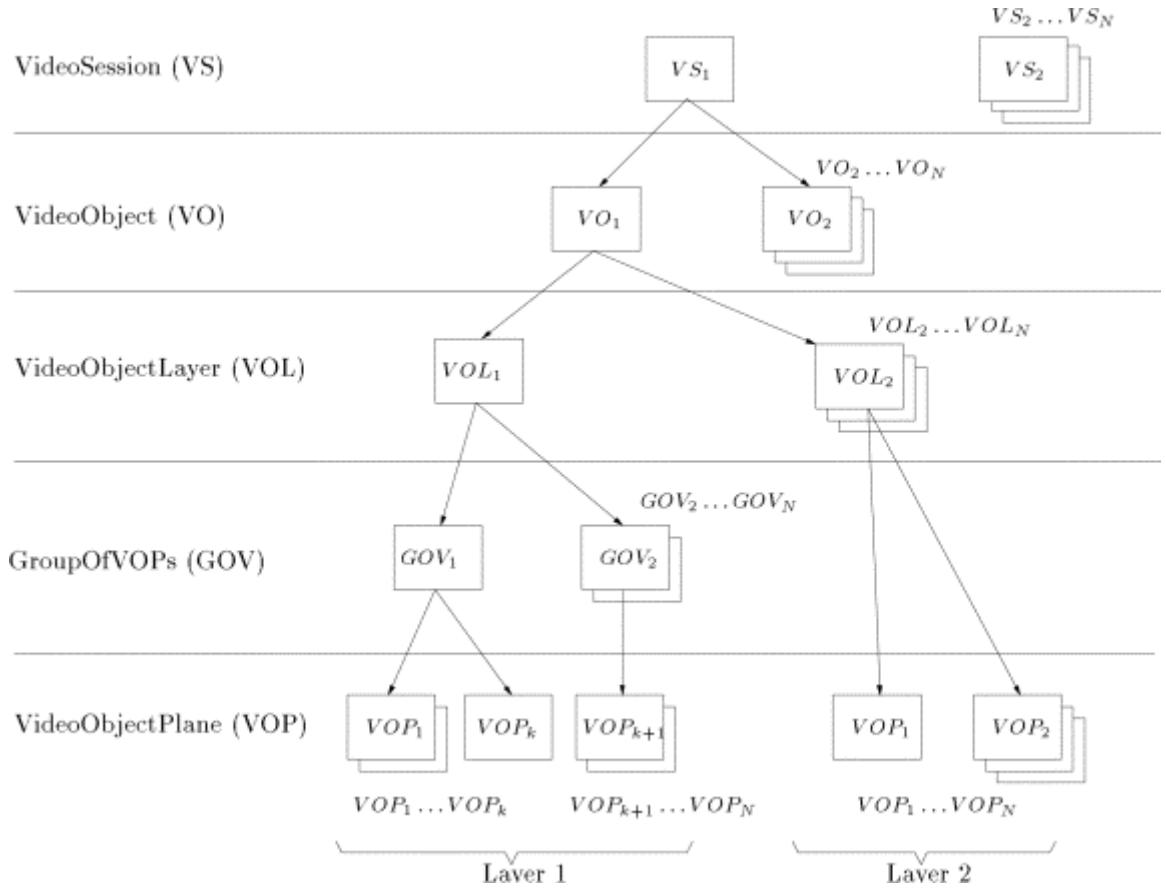


Figura 19: Formato degli AVO in MPEG-4

differenza dei precedenti standard MPEG-1 ed MPEG-2, la sorgente video non è più rettangolare ma può avere dimensioni qualunque; inoltre, da frame a frame le dimensioni e le coordinate dell'oggetto possono variare.

La rappresentazione di ogni singolo frame segue un preciso schema gerarchico:

- Il GOV (Groups of VOPs), opzionale, raggruppa tutti i VOP di uno stesso layer per poter poi inserire funzioni di Random Access in determinati punti del bitstream;Ogni VOP appartenente allo stesso layer viene raggruppato in un VOL (Video Object Layer), e codificato sia in modalità multilayer (MPEG-4) o single-layer (come nel caso MPEG-1/2);
- I vari VOL vengono raggruppati in un VO (Video Object), che rappresenta il singolo oggetto 2D da mostrare a video;
- Il VS (Video Session) rappresenta infine la scena finale, comprendente tutti gli oggetti, sia 2D che 3D, naturali o sintetici.

In tal modo, ogni VOP può essere decodificato separatamente e la sequenza video può essere facilmente manipolata, come si vede dalla figura 20 .

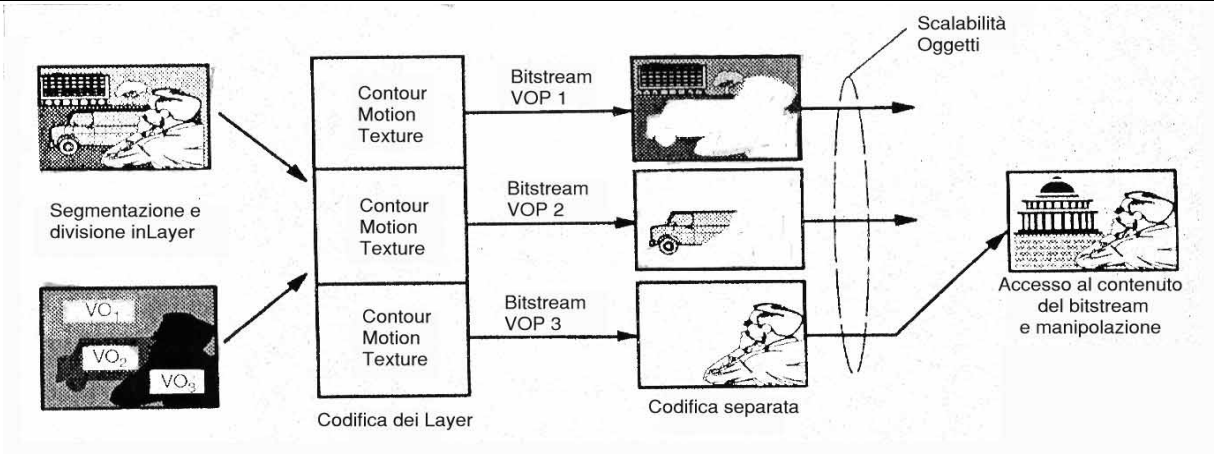


Figura 20: Codifica ad oggetti in MPEG-4. Come si può vedere, oltre ad accedere al contenuto del bitstream è possibile anche manipolarlo.

5.2.2 CODIFICA E TRASMISSIONE DEI VOP

Innanzitutto, viene codificata la forma di ogni VOP, le dimensioni ed il texture presente: il procedimento usato in questo ultimo caso è uguale allo standard MPEG-1 con macroblocchi di grandezza 8x8.

Ogni singolo VOP viene successivamente inscritto in un rettangolo formato dal minimo numero di blocchi di dimensioni 16x16 pixel sia in verticale che in orizzontale, e ad ogni pixel viene assegnato un valore rappresentante la trasparenza (viene generato in tal modo il *canale alfa*). Questi valori, poi, vengono codificati tramite DCT in maniera simile a quanto già visto per MPEG-1.

Anche la tecnica di trasmissione dei VOP segue concetti già spiegati: infatti, esistono VOP di tipo I, B e P. Il vettore di moto viene calcolato a partire da macroblocchi di 16x16 o 8x8 pixel di dimensione esattamente come accade in MPEG-1 e MPEG-2, ma solo nel caso in cui detti macroblocchi siano completamente inclusi nel VOP; altrimenti, viene utilizzata una tecnica diversa che consiste nel calcolare il vettore di moto solo per i pixel appartenenti al VOP e non in tutto il macroblocco (algoritmo di tipo SADCT, Shape-Adaptive DCT), migliorando così l'efficienza in codifica[Ebrahimi].

Tra i vari VOP che compongono un'immagine, quello dello sfondo riveste una grande importanza: in genere, è fisso o si sposta a causa del movimento di una telecamera, e per questo motivo sono state studiate svariate tecniche di codifica, tra le quali quella di tipo *sprite*. Consideriamo ad esempio la ripresa di una partita di tennis. Questa può essere divisa in due parti principali: lo sfondo ed il tennista. Analizzando tutto il filmato, è possibile recuperare lo sfondo per intero, dato che la telecamera ed il tennista si spostano ogni volta, e di conseguenza parti precedentemente non visibili in un dato frame risultano visibili in un altro. Così, il frame 1 mostrerà una parte dello sfondo, il frame 10 un'altra ancora e così via, fino ad arrivare ad esempio al frame 200 che mostra l'ultima parte dello sfondo che ci manca. A quel punto, sovrapponendo tutte le parti dello sfondo potremo creare un nuovo frame in cui c'è il solo sfondo per intero senza il tennista. In fase di codifica, allora, è sufficiente trasmettere all'inizio del filmato, e solo per una volta, il frame con l'intera immagine dello sfondo; successivamente, frame dopo frame, vengono inviate le coordinate della telecamera e del tennista, la rotazione, la scala, la prospettiva dello sfondo. Il decoder dunque calcolerà la nuova posizione e forma dello sprite e vi sovrapporrà l'immagine del tennista, precedentemente codificata come VOP (figura 21). In ogni frame otto parametri sono sufficienti per descrivere correttamente le

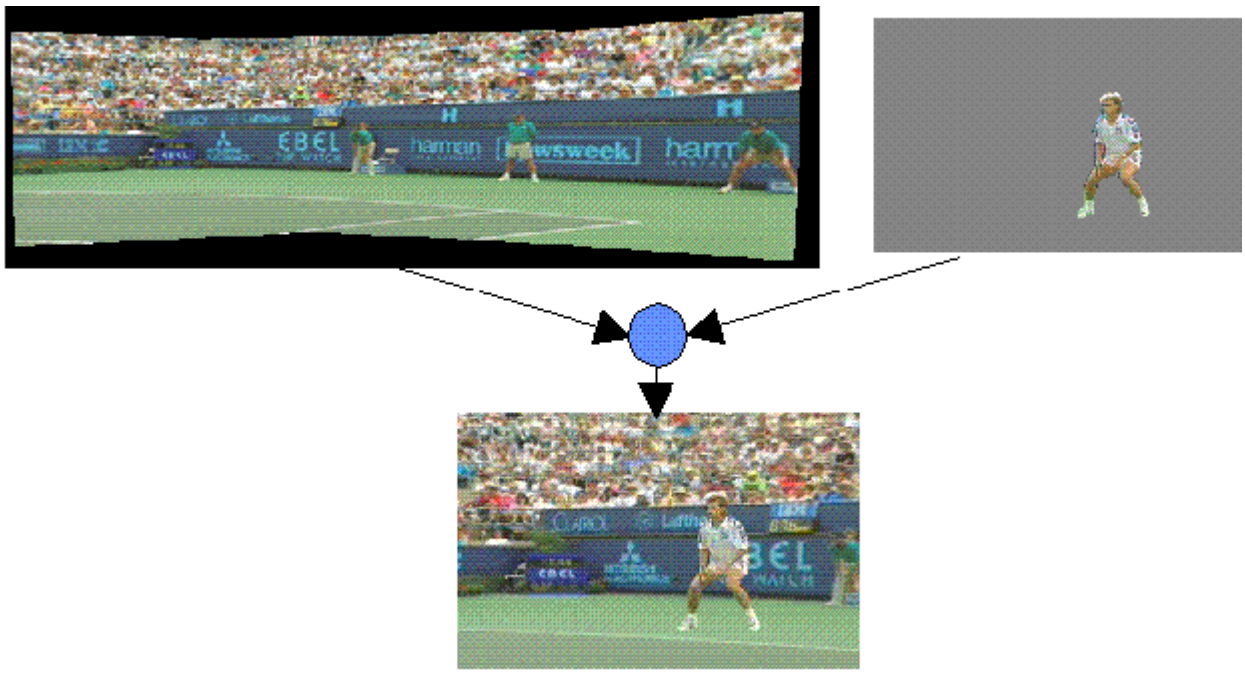


Figura 21: Codifica a sprite dello sfondo in una partita di tennis. Il frame finale è dato dalla sovrapposizione dello sprite "sfondo" con lo sprite "tennista".

informazioni necessarie, anche se a volte ne bastano meno [Lee97].

Non sempre però è necessario esaminare tutto il filmato per recuperare l'immagine dello sfondo: soprattutto nelle videoconferenze, lo sfondo è trasmesso per intero all'inizio del filmato (quando ancora non è iniziata la conferenza), e quindi la codifica dello sprite è molto semplificata. Se poi, come spesso accade, lo sfondo è una parete fissa, la banda usata per trasmettere questo sprite è molto bassa perché il numero di coordinate da trasmettere è molto piccolo.

5.2.3 TRASMISSIONE A BANDA STRETTA E LARGA

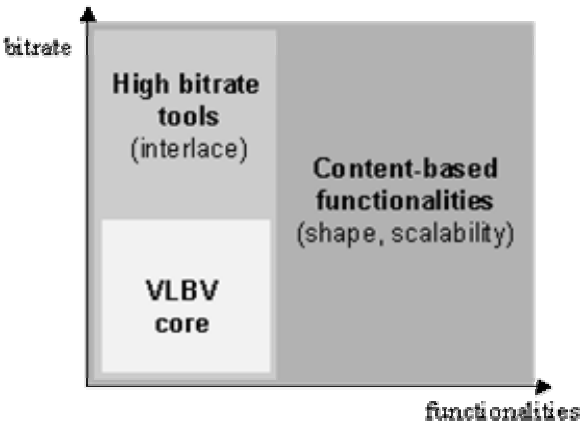


Figura 22: Tool di codifica del video in MPEG-4

Lo standard MPEG-4 ha come primario obiettivo la trasmissione di sequenze audiovisive a bassa banda. Nella figura 22 possiamo vedere come viene suddiviso il modello di codifica video a seconda delle funzionalità richieste e del bit rate disponibile. In basso a sinistra, il *VLVB (Very Low Bit rate Video)* fornisce tool ed algoritmi per applicazioni operanti fra 5 e 64 Kb/s con varie risoluzioni video (come i formati QCIF o CIF) e bassi frame rate (da 0 Hz per immagini fisse a 15 Hz). Tipiche applicazioni sono la codifica di immagini rettangolari (stile MPEG-1 e MPEG-2) con grande efficienza e robustezza all'errore e le operazioni di

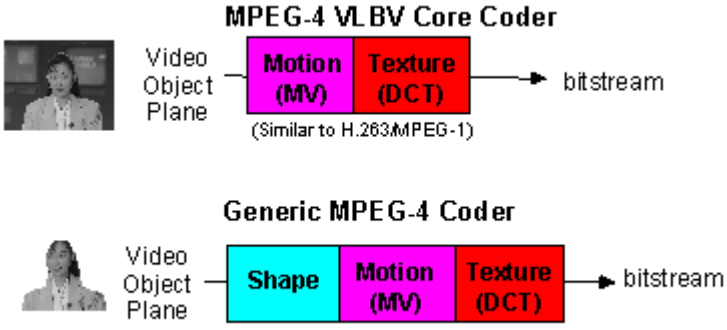


Figura 23: Differenza fra la codifica VLVB di MPEG-4 e quella generica

RA/FF/FR (Random Access/Fast Forward/Fast Reverse) per la creazione di database multimediali.

Per bit rate fino a 4 MHz, invece, troviamo il modello HBV (*Higher Bit rate Video)*, con risoluzioni via via maggiori fino alla qualità di tipo CCIR-601, ma con gli stessi tool ed algoritmi del modello VLVB [Sikora97c][ISO-N2995].
Per entrambi i modelli, poi, esistono funzionalità addizionali, ossia codifica di oggetti di forma arbitraria. Come si vede dalla figura 23, quindi, MPEG-4 è compatibile con gli standard precedenti: infatti, un'immagine rettangolare non è altro che un solo VOP dalle stesse dimensioni del frame video.

5.2.4 PROFILI E LIVELLI IN MPEG-4

La parte Video di MPEG-4 è divisa in profili e livelli diversi, a seconda delle applicazioni cui è rivolta:

- *Simple profile*: Ammette la codifica I e P dei VOP, e si suddivide nei seguenti livelli:
  - Livello 1 (64 Kbit/sec, formato QCIF);
  - Livello 2 (128 Kb/s, formato CIF);
  - Livello 3 (384 Kb/s).

  L'applicazione principale è la videotelefonia: tutti e tre i livelli soddisfano le richieste minime necessarie per un buon utilizzo di questo servizio, ovvero permettere all'utente di riconoscere distintamente una faccia e l'emozione che esprime, di capire distintamente la voce e di leggere il labiale senza problemi.
- *Core profile*: Viene ammessa anche la codifica B dei VOP e la scalabilità temporale. Si suddivide in:
  - Livello 1 (384 Kb/s, formato CIF);
  - Livello 2 (2 Mb/s. formato CIF).
- *Main profile*: Contiene i profili precedenti e li estende. Si divide in:
  - Livello 1, per bande fino a 2Mb/s;
  - Livello 2: per bande fino a 15 Mb/s, con codifica di tipo CCIR-601;
  - Livello 3: fino a 38.4 Mb/s, in formato CCIR-601 progressivo ad alta definizione.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

5.2.5 SCALABILITÀ DEL PROTOCOLLO MPEG-4

Come abbiamo visto, MPEG-4 è stato disegnato per essere supportato da un grande numero di ricevitori con diverse capacità di risoluzione e bande di trasmissione. Il concetto di scalabilità, sia spaziale che temporale. già introdotto in MPEG-2, viene qui ripreso ed ampliato.
La scalabilità spaziale consiste nel trasmettere tre layer, ognuno dei quali con un VOP a differenti risoluzioni. La versione con bassa risoluzione viene codificata come layer base, mentre le risoluzioni via via maggiori vengono ricampionate verso l'alto e compensate con un layer di *prediction error*, generato dalla differenza tra il frame calcolato e quello originale. In tal modo, se il ricevitore non è in grado di decodificare il segnale alla massima risoluzione, si baserà solo sul layer di base. Questo tipo di codifica nasce per soddisfare determinate esigenze di banda, anche variabile.
La scalabilità temporale si basa su quella spaziale, dato che in un bitstream sono supportati diversi frame rate: in questo modo è per esempio possibile codificare lo sfondo (magari fisso) con un frame rate minore, ed il VOP principale (una persona che parla) ad un frame rate maggiore. Anche in questo caso, quindi, si possono soddisfare svariate esigenze di banda senza compromettere di molto la qualità della trasmissione.

5.2.6 GESTIONE DEGLI ERRORI

Uno degli scopi di MPEG-4 è la robustezza dei segnali in presenza di ambienti a forte tasso di errore. Il crescente sviluppo delle comunicazioni cellulari ha spinto MPEG-4 a studiare tool ed algoritmi specificamente progettati per questo tipo di sistemi.
I risultati di MPEG-4 possono sostanzialmente essere divisi in tre grandi aree o categorie:

- Risincronismo;
- Recupero dei dati;
- Correzione dell'errore (*error concealment*).

I tool di risincronismo, come dice il nome, cercano di ristabilire la sincronia tra il decoder ed il flusso dei dati dopo aver rilevato uno o più errori. Generalmente, i dati tra il punto di sincronismo prima dell'errore ed il primo punto in cui il sincronismo viene ristabilito, vengono persi.

La tecnica di risincronismo adottata da MPEG-4 è simile a quella già presente nello standard H.261 e H.263. In questi standard, viene definito un Group Of Blocks (GOB) che contiene una o più file di macroblocchi. Ogni GOB è preceduto da un header che contiene le informazioni utili al decoder per identificare un GOB ed attivare così il risincronismo.

La tecnica usata dal GOB è la seguente: dopo aver codificato un certo numero di macroblocchi, viene inserito un marcatore di sincronismo nel bitstream. Il problema maggiore è che la codifica avviene a bit rate variabile, e quindi i marcatori non si trovano equamente distanziati l'uno dall'altro. In presenza di aree ad alto movimento di immagini, dunque, sarà più facile trovare degli errori, e di conseguenza correggerli.

In MPEG-4, invece, il marcatore non viene inserito in base al numero di macroblocchi, bensì al numero dei bit contenuti nei dati; in tal modo, tutti i marcatori sono equamente spaziati tra di loro.

Il marcatore di sincronismo contiene tutte le informazioni necessarie per decodificare il primo macroblocco che segue, come per esempio il numero sequenziale ed il livello di quantizzazione; oltre a questo, contiene un Header Extension Code (HEC) che attiva ulteriori processi di risincronismo nel caso che l'header del VOP sia corrotto.

Per maggiore sicurezza, è stato anche introdotto un secondo metodo chiamato "intervallo di sincronismo". In pratica, per evitare che un VOP sbagliato venga codificato come VOP Start Code (che indica l'inizio di un VOP), gli Start Code vengono inseriti solo ad intervalli regolari. In tal modo, se un VOP Start Code è al di fuori di questo intervallo, non verrà considerato valido.

Nelle versioni precedenti di MPEG, invece, se un header di un certo tipo si corrompeva per assumere casualmente un codice identico ad un header di un altro tipo, questo errore non veniva riconosciuto.

Una volta recuperato il sincronismo, il secondo passo consiste nel recuperare i dati, che generalmente verrebbero scartati. Una tecnica usata è il RVLC (Reverisble Variable Length

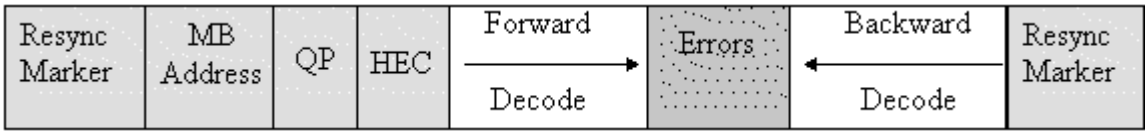


Figura 24: Tecnica di correzione degli errori in MPEG-4

Coding), che permette di recuperare parte degli errori tramite decodifica di tipo Forward e Backward, come indicato nella figura 24.

Quando gli errori non si possono eliminare, una tecnica molto importante è quella di correggerli. Per esempio, a basse risoluzioni è possibile "eliminare" un errore ricopiando i blocchi del frame

precedente, creando così l'illusione di una certa continuità nelle immagini. Questa tecnica, d'altronde, è usata anche in MPEG-2.

In MPEG-4, l'approccio è quello di dividere l'oggetto in movimento dal texture. Viene così introdotto un nuovo marcatore di sincronismo tra le informazioni inerenti il movimento e quelle inerenti il texture. In caso di errore, dunque, il texture viene scartato, ed il VOP successivo viene recuperato da quello precedente mediante compensazione di moto[ISO-N2203].

In questo modo, gli errori che non potevano essere recuperati dagli algoritmi di *data recovery* vengono recuperati a partire dai frame precedenti. L'errore permane, ma è molto meno visibile.

5.3 CODIFICA DEL VIDEO SINTETICO

La codifica del video sintetico in MPEG-4 si divide in due grandi categorie: la codifica di animazioni facciali e quella di animazioni del corpo. Inoltre, viene gestito il texture mapping 2D e 3D come pure il mesh 2D e 3D. La codifica si basa sullo standard VRML 2.0 (Virtual Reality Markup Language) con alcune migliorie.

5.3.1 ANIMAZIONE FACCIALE

In MPEG-4 esiste un oggetto, denominato Viso (*Face*), già pronto per essere renderizzato ed animato. Inizialmente, dopo la creazione, la faccia assume un'espressione neutra, e questa può essere modificata a piacere grazie a parametri quali FDP (Facial Definition Parameter – Parametro di definizione facciale) e FAP (Facial Animation Parameter – Parametro di animazione facciale). Per default, le labbra si toccano, la lingua è orizzontale e piatta e tocca il punto d'unione delle labbra stesse. Il viso iniziale deve essere costruito in base ad un modello geometrico conforme ad MPEG-4, perché lo standard ISO/IEC definisce solo come muoverlo ed animarlo, e non ne definisce forma, colore, aspetto. Pertanto, esistono vari modelli predefiniti, come *JOE*, creato allo CSELT[ISO-14496-5], *IST*, creato invece nell'*Instituto Superior Técnico* di Lisbona, o quello tutto italiano, sviluppato presso i laboratori di Genova, con i suoi Sarah, Mike ed Oscar[Pockaj].

5.3.2 FAP

I 68 diversi FAP esistenti, alcuni dei quali evidenziati in tabella, forniscono le molteplici espressioni che può assumere il viso, come tristezza, allegria, sconcerto ecc. Ad esempio, il FAP 51 definisce

come abbassare la metà del labbro inferiore, il FAP 25 come alzare verso l'alto l'occhio destro e così via.

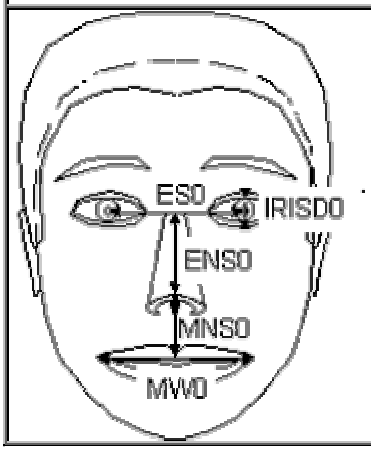
#	Nome del FAP	Descrizione	Unità
1	viseme	Insieme di valori utili a determinare l'unione di due visemi per questo frame (p.es. pbm, fv, th)	ND
2	expression	Serie di valori che determinano l'unione di due espressioni	ND
3	open_jaw	Spostamento verticale della mascella	MNS
12	raise_l_cornerlip	Spostamento verticale dell'angolo interno sinistro del labbro	MNS
14	thrust_jaw	Spostamento in profondità della mascella	MNS
18	depress_chin	Movimento di compressione verso l'alto del mento (come nel caso di tristezza)	MNS
25	pitch_l_eyeball	Orientamento verticale della pupilla sinistra	AU
29	dilate_l_pupil	Dilatazione della pupilla sinistra	IRISD
38	squeeze_r_eyebrow	Spostamento orizzontale del sopracciglio destro	ES
39	puff_l_cheek	Spostamento orizzontale della guancia sinistra	ES
43	shift_tongue_tip	Spostamento orizzontale della punta della lingua	MW
51	lower_t_midlip_o	Spostamento verticale della metà esterna superiore del labbro	MNS
63	raise_nose	Spostamento verticale della punta del naso	ENS
65	raise_l_ear	Spostamento verticale dell'orecchio sinistro	ENS
66	raise_r_ear	Spostamento verticale dell'orecchio destro	ENS

Tabella 5: Esempi di possibili FAP

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Con il nome di "viseme" si definisce una classe di 15 ulteriori FAP che definiscono i fonemi da associare ad un movimento labiale e facciale. Ad esempio, vengono definiti fonemi quali *p,b,m* per parole come *pane, burro, mamma*, oppure *z, S, dZ* per parole come *casa, scimmia, giallo.*
Le unità di misura espresse da ciascuna funzione si basano sulle definizioni del FAPU (*Facial Animation Parameters Unit* – Unità parametriche di animazione facciale), che rappresentano

Tabella 6: Descrizione dei parametri facciali

Descrizione		**Valore in FAPU**
	IRIS Diameter (Diametro dell'iride, IRISD0 nella faccia neutra)	IRISD = IRISD0 / 1024
	Eye Separation – Distanza fra gli occhi	ES = ES0 / 1024
	Eye – Nose Separation (Distanza occhi-naso)	ENS = ENS0 / 1024
	Mouth – Nose Separation (Distanza bocca-naso)	MNS = MNS0 / 1024
	Mouth Width (Grandezza della bocca)	MW = MW0 / 1024
	Angular Unit (Unità angolare)	AU = 10^{-5} rad

frazioni di distanze predefinite. La rotazione è invece espressa in radianti. Non esiste un limite a queste unità di misura; pertanto, è possibile ad esempio far spalancare la bocca in maniera ben superiore alle normali caratteristiche umane e rendere così il modello utile per effetti speciali quali cartoni animati o scherzi.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

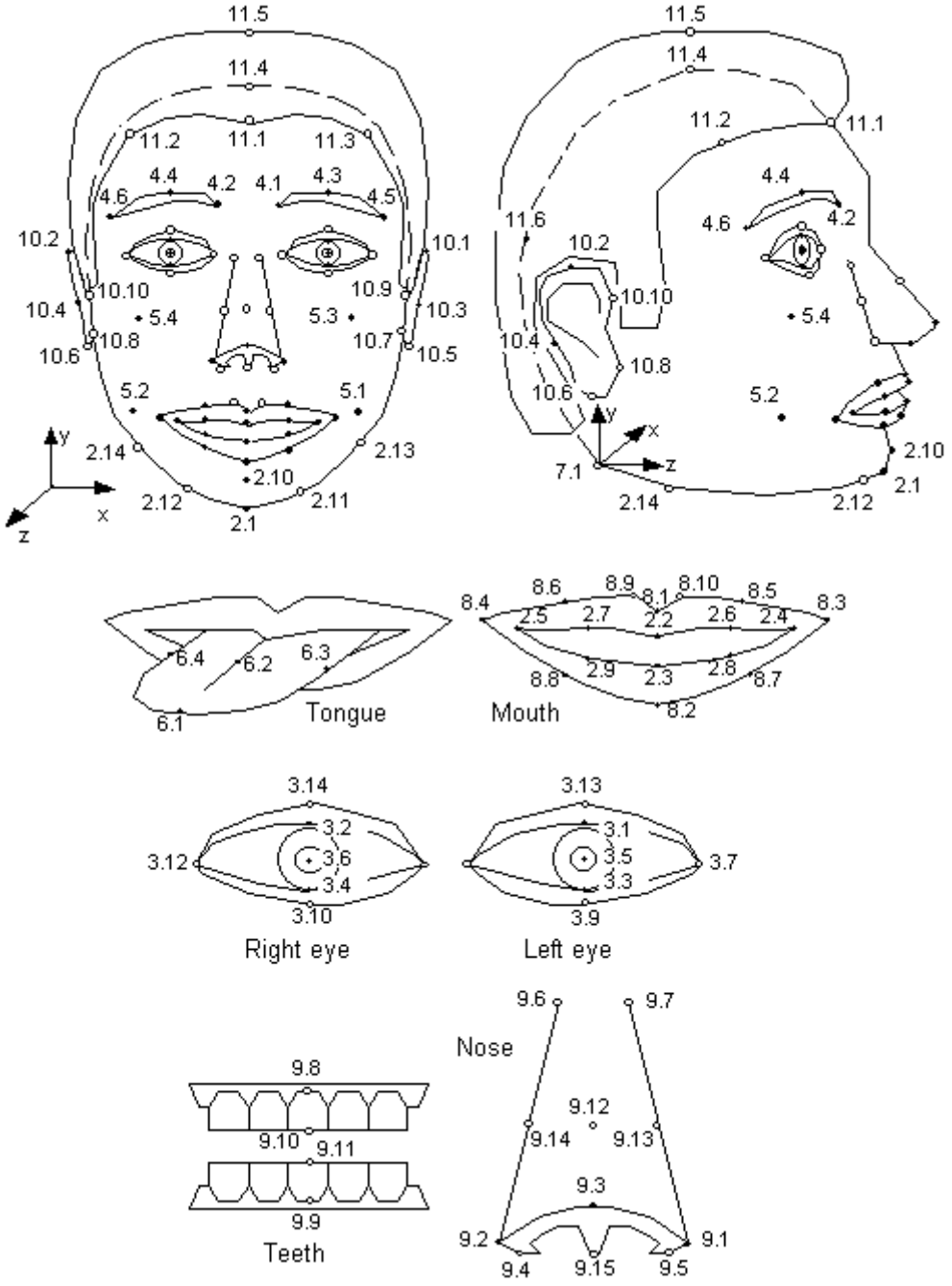


Figura 15: FDP

5.3.3 FDP

Le FDP invece sono dei complessi sistemi di parametri definiti sia per la calibrazione del viso (ovvero, l'adattamento del modello geometrico ad un viso reale), sia per permettere lo scaricamento di un intero modello facciale tra l'encoder ed il decoder.
Sono stati definiti 84 *punti caratteristici*, come si vede dalla tabella seguente, divisi per categoria (viso, lingua, denti ecc.) e per funzione: alcuni vengono usati solo per l'animazione, mentre tutti possono essere usati per la calibrazione. Questi punti caratteristici vengono divisi in gruppi a seconda della parte del viso cui appartengono. Ogni modello facciale deve avere come minimo questi punti
caratteristici per poter essere considerato conforme allo standard MPEG-4.
Per poter animare un viso artificiale, dunque, bisogna seguire i seguenti passi:

- Costruire un modello e definire i punti FDP;
- Definire per ogni FAP come si deve muovere il punto FDP correlato (ad esempio: per la funzione "alza il labbro", dire di quanto si alza e quali punti spostare);

Alternativamente, si può operare uno scaricamento del modello facciale tra encoder e decoder, per poi ricostruirlo tramite il protocollo BIFS, in questo caso coincidente con lo standard VRML. Le varie scene sono collegate da nodi di tre tipi: uno gerarchico, uno di trasformazione 3D (come rotazione, scala, traslazione); uno di definizione di attributi (mesh 2D/3D, colore, texture).

5.3.4 LIVELLI NELLA CODIFICA VIDEO SINTETICA

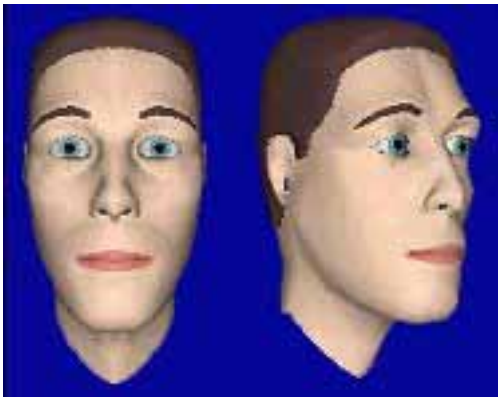


Figura 26: Modello geometrico

MPEG-4 ha stabilito una suddivisione in livelli anche per la codifica video sintetica:

- Al livello 1, è possibile animare un solo modello facciale con una bit rate massima di 16Kb/s, renderizzata a 15 Hz di frame rate massimi;
- Al livello due, fino a 4 facce possono essere animate contemporameamente, a patto che la bit rate totale non superi 32Kb/s ed il rendering non oltrepassi i 60Hz di rate.

5.3.5 CALIBRAZIONE FACCIALE

Per capire meglio in cosa consista la calibrazione facciale, vediamo un esempio:
Prima di tutto, bisogna preparare un modello geometrico, come "Oscar" ad esempio, creato nei laboratori dell'Università di Genova e caratterizzato da 2346 poligoni e 1228 vertici. Poi, a partire dal set FDP di Claude, generato da un'immagine reale, si può modificare ed adattare il modello (*calibrazione*). Infine, viene applicato il texture mapping e si ottiene il risultato finale. A questo punto, finita la calibrazione, si procede all'animazione generando tutte le espressioni volute.
L'intera procedura è descritta dalle figure 26-30. Nell'appendice è anche possibile trovare un filmato dimostrativo.

5.4 ANIMAZIONE DEL CORPO

Figura 27: Modello reale

In maniera simile all'animazione facciale, anche per il corpo vengono definiti due parametri: il primo è il BDP (Body Definition Parameter – Parametro di definizione del corpo), che serve a trasformare il corpo di default in uno particolare, con una propria forma, superficie e texture. C'è poi il BAP (Body Animation Parameter – Parametro di animazione del corpo), per produrre posture ed animazioni, costituito da 296 parametri diversi. Tra questi, troviamo gli angoli di rotazione di piedi, anca, ginocchia, bacino, spina dorsale, spalle, clavicole, gomiti, polsi e dita. Il corpo contiene fino a 186 gradi di libertà, 25 per le sole mani[bap]. Questo può dare un'idea delle svariate posture che può assumere l'oggetto Corpo. Il modello inoltre può essere costruito con diversi livelli di definizione ed è così

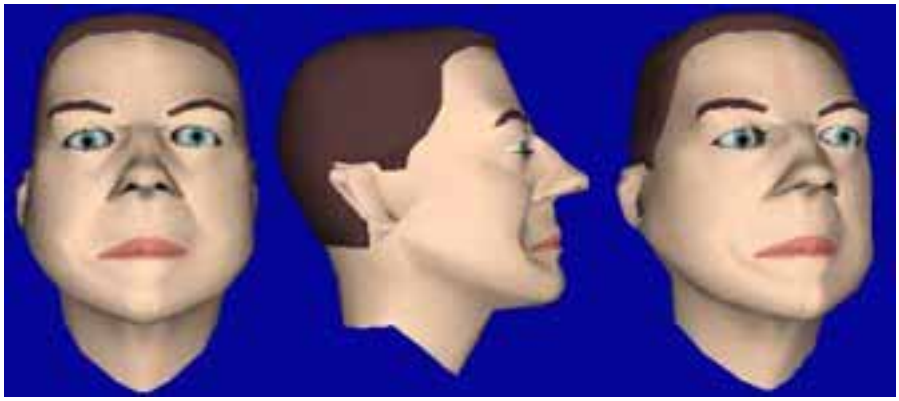


Figura 28: Calibrazione del modello

Figura 29: Calibrazione

scalabile in complessità.
I BAP, come i FAP, vengono quantizzati e codificati con uno schema predittivo simile alla codifica video: il frame successivo viene calcolato a partire da quello precedente e compensato con l'errore di predizione (dato dalla differenza fra il frame vero e quello calcolato). Anche la quantizzazione può essere variata, ed il modello risulta quindi scalabile in bit rate.

Una volta costruito tramite modello geometrico, l'oggetto Corpo (*Body*) contiene un generico corpo umano in una posizione di default, caratterizzato da vari poligoni con mesh 3D, pronto per essere renderizzato. In sostanza, il corpo si trova in piedi, con le braccia distese lungo il corpo, le palme delle mani rivolte verso l'interno.

Figura 30: Animazione finale

Non appena riceve le coordinate e le istruzioni dai set BDP e BAP, il corpo assume colore, forma e posizione volute.

5.5 CODIFICA SNHC

Un mesh 2D è una divisione di una regione in tanti tasselli poligonali, i cui vertici vengono definiti come *nodi*. In MPEG-4, vengono considerati solo *mesh triangolari*, i cui tasselli sono dei triangoli. Mentre l'immagine si muove, i punti del bordo e quelli interni si spostano, e di conseguenza anche i triangoli, a seconda del movimento, si stirano e si comprimono.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

La codifica del mesh 2D consiste nel trasmettere, frame per frame, le coordinate di questi punti in modo da potervi inserire il texture in maniera corretta. La codifica e la trasmissione di questi dati segue, ancora una volta, la tecnica predittiva. Nel caso di mesh triangolari, il calcolo non è molto complesso e la tecnica di codifica utilizzata è generalmente quella di mappatura affine, che ha bisogno di sei coefficienti diversi. La relazione che sussiste tra le coordinate (x,y) e le coordinate $\underline{x},\underline{y}$ al tempo t sono[Tekalp]:

$$\underline{x} = ax + by + c$$
$$\underline{y} = dx + ey + d$$

con a..e parametri del modello affine.

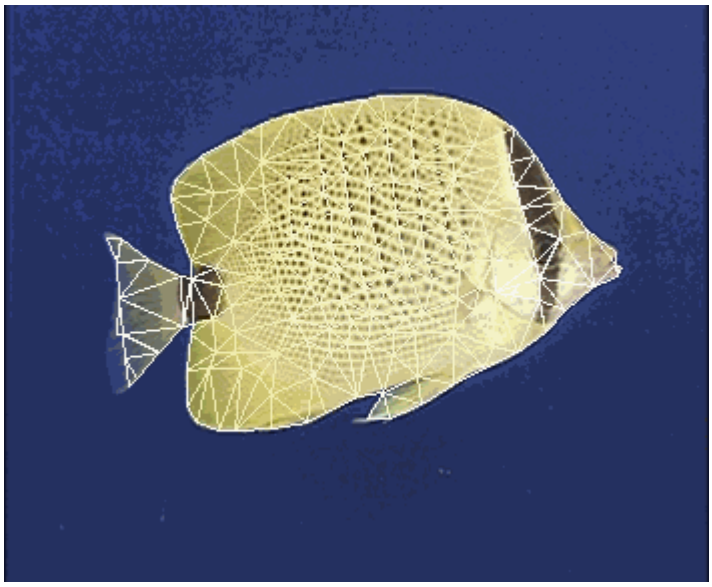


Figura 31: Mesh 2D

Figura 32: Codifica SHNC mediante sovrapposizione di scritte ad un'immagine

Tramite questa mappatura, si possono controllare anche valori quali riflessi e deformazioni di taglio, e preservare linee rette. Questo metodo si applica molte bene a tutte le immagini dai movimenti dolci e continui, perché ogni singolo triangolo vincola tutti quelli adiacenti, mentre immagini con forti discontinuità non si prestano a questo tipo di codifica.

Grazie alla rappresentazione con mesh 2D, è possibile non solo gestire il texture, ma anche unire immagini naturali con quelle artificiali (generate dal computer) per creare particolari effetti. In tal caso, bisognerà curare il sincronismo tra i due tipi di immagini, ma grazie a questa tecnica è possibile un editing avanzato mediante il quale si possono variare sfondi o colori degli oggetti, come pure aggiungere dei testi e così via. La figura 32 evidenzia uno dei tanti esempi possibili. La capacità di poter riunire oggetti audiovisivi sia naturali che artificiali, così come quella di

rappresentare codifiche ibride di oggetti, prende nome di SNHC (Synthetic and Natural Hybrid Coding).

5.6 CODIFICA AUDIO

Nello standard MPEG-4, anche la codifica audio, come quella video, si può suddividere in due grandi aree:

- Codifica di audio naturale;
- Codifica di audio sintetico.

La prima gestisce tutti gli algoritmi ed i codec necessari alla trasmissione di voce e musica in una banda compresa tra i 2Kb/s ed i 64 Kb/s, mentre la seconda sfrutta e migliora standard già esistenti, ad esempio il MIDI, per poter ricreare un suono sintetizzato, come quello di uno strumento musicale o la stessa voce umana. Entrambe le aree devono poter interagire fra loro, e dunque MPEG-4 ha studiato un meccanismo per implementare le funzioni di tipo SNHC anche per l'audio.

5.6.1 CODIFICA AUDIO NATURALE

Così come le applicazioni video possono essere viste come contenitori di oggetti (*content-based*), MPEG-4 ha come obiettivo anche quello di fornire multipli layer anche nei bitstream audio, per poter poi gestire e manipolare i suoni come se fossero "oggetti" diversi. In tal modo, ad esempio, la codifica multilayer di un concerto per violino permette di estrarre, a richiesta, il suono di ogni singolo strumento, per poi eventualmente cambiarlo. Alternativamente, è possibile ascoltare il concerto con o senza il primo violino. MPEG-4 ha standardizzato la codifica di audio naturale a bit rate compresi fra 2 Kb/s e 64 Kb/s.
Al fine di ottenere la massima compressione e la migliore qualità audio in tutto il range della banda, sono stati definiti tre tipi di codifica:

- *Codifica Parametrica*: opera tra i 2 ed i 4 Kb/s nel caso di voce campionata a 8 kHz, e tra i 4 ed i 6 Kb/s per musica campionata a 8 o 16 kHz;
- *Codifica CELP (Code Excited Linear Predictive)*: supporta audio campionato ad 8 e 16 kHz e viene usata tra i 6 ed i 24 Kb/s;
- *Codifica T/F (Time to Frequency)*: Per bande fino ai 64 Kb/s.

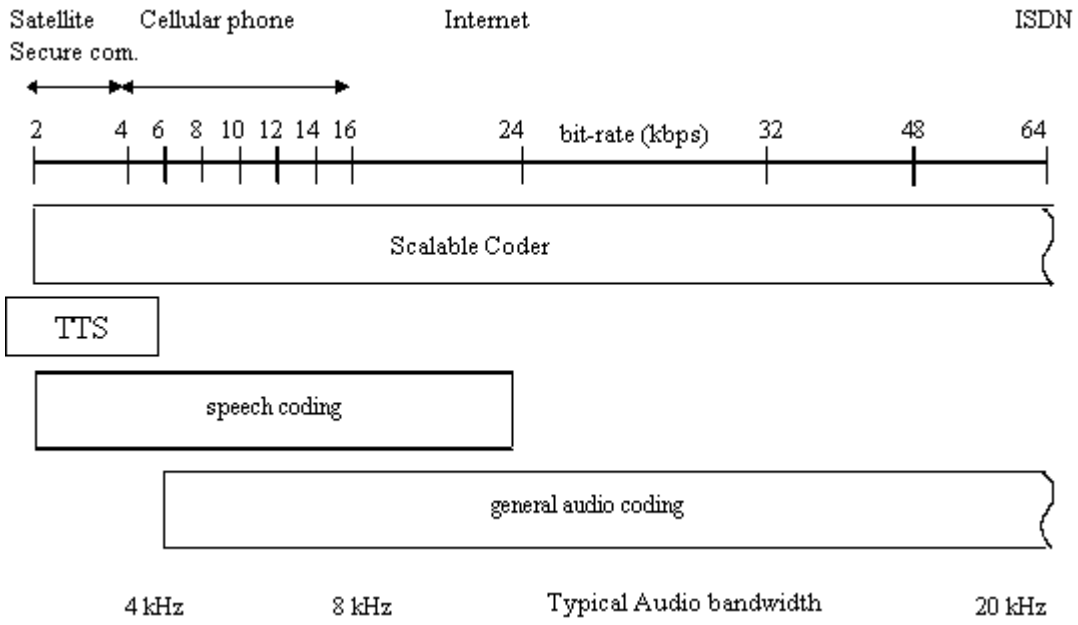


Figura 33: Codifica audio naturale in MPEG-4

In realtà, i range sono molto più ampi, e, quindi, apparentemente non viene seguito il concetto di *one system – one tool*: osserviamo infatti nella figura 33 una sovrapposizione dei tre tool di codifica audio. Questo però è necessario al fine di permettere un passaggio graduale fra una banda e l'altra.

Le altre caratteristiche supportate dai tool di codifica di MPEG-4 sono:

- Cambio di velocità, che permette di cambiare la scala del tempo senza alterare il tono;
- Cambio di tono, senza cambiare la scala dei tempi;
- Scalabilità della bit rate, che permette la riduzione della bit rate del segnale mantenendone l'intelligibilità. Il downscaling può essere fatto sia durante la trasmissione, sia dall'encoder a priori;
- Scalabilità della larghezza di banda, visto come caso particolare del punto precedente, che permette di scartare una parte dello spettro di frequenze durante la trasmissione o la codifica;
- Scalabilità della complessità dell'encoder, per permettere a diversi encoder di generare bitstream validi ed intelligibili;
- Scalabilità della complessità del decoder, per permettere a vari decoder di decodificare correttamente il bitstream;
- Robustezza all'errore, per evitare o correggere distorsioni udibili caratterizzate da un errore di trasmissione.

Vediamo adesso nei dettagli i diversi tipi di tools.

5.6.2 CODIFICA PARAMETRICA

MPEG-4 non stabilisce esattamente quale codec usare, bensì il tipo di codec. Nell'ambito della codifica parametrica, tre codificatori diversi sono stati proposti, e sono l'HVXC (Harmonic Vectro eXcitation Coding), l'HLIN (Harmonic and Individual Line plus Noise) – presente solo nella Versione 2 dello standard - ed il FS016 (un tipo di CELP)[Quackenbusch98][ISO-N2203].
L'HVXC e l'HLIN lavorano "in coppia", in quanto il primo permette la codifica di segnali vocali a 2 e 4 Kb/s in modo scalabile e la decodifica a bit rate variabile ad una media tipica di 1.5 Kb/s; il secondo è studiato per codificare segnali audio non vocali, come la musica, a 4 Kb/s ed oltre. Entrambi supportano il cambio di tono e velocità anche durante la decodifica, come richiesto dallo standard.
Alternando i due decoder tramite fading del volume o mixando il loro output, si può generare una codifica di tipo parametrico.
Il codec FS1016 invece è uno standard federale americano. Utilizza l'algoritmo CELP ed opera a 4.8 Kb/s.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

5.6.3 CODIFICA CELP

Questo codec lavora tra i 6 ed i 24 Kb/s e supporta un segnale di ingresso campionato a 8 o 16 kHz, e la scalabilità della bit rate. Nel caso di un segnale ad 8 kHz, possono essere aggiunti fino a tre layer di qualità (enhancement layers) a passi di 2 kHz l'uno. In questo modo, si possono aggiungere 2, 4 o 6 Kb/s al layer di base a seconda della grandezza di banda e della capacità del decoder. E' supportata anche la bit rate variabile.
Se il segnale invece è stato campionato a 16 kHz, si può decodificare il segnale usando solo una parte del bitstream: infatti, è supportata la scalabilità nella complessità del codec. La complessità può essere cambiata anche in tempo reale, così da soddisfare eventuali congestioni nella rete o carico variabile sul processore.
Il codec CELP supporta anche la scalabilità in larghezza di banda, se il segnale sorgente è a 8 kHz, grazie ad un tool di estensione della banda che converte il segnale a 16 kHz.

5.6.4 CODIFICA T/F

Per quanto riguarda la codifica Time to Frequency, a 64 Kb/s viene utilizzato il codec MPEG-2 AAC. A bande inferiori, invece, possono essere usati altri tool, come il BASC (Bit-sliced Arithmetic Coding) ed il TwinVQ (Transform-domain Weighted Interleaved Vector Quantization).
Il BASC fornisce una scalabilità molto fine in bit rate, a passi di 1 Kb/s in un range compreso fra 16 e 64 Kb/s. Il formato BASC (presente solo in MPEG-4 Versione 2) è una miglioria della codifica AAC, e la conversione da AAC a BASC è una procedura abbastanza facile.
Il TwinVQ invece è un codec particolarmente robusto agli errori, ed è indicato per sistemi che richiedono una forte scalabilità in bit rate.

5.6.5 RISULTATI DEI TEST DI VERIFICA

I test di verifica dei vari codec (FS1016, HXCV, CELP a vari bit rate, MPEG-2 Layer III 24 Kb/s, AAC e TwinVQ) sono stati condotti sia in Europa che in Giappone, e le lingue testate sono state inglese, tedesco, svedese e giapponese. I test si sono svolti presso la NTT (Nippon Telegraph & Telephone), la FhG (Fraunhofer Gesellschaft) e la NRC (Nokia Research Center), con persone di madrelingua e con finlandesi (nel caso NRC) che conoscevano alcune lingue straniere. I risultati sono stati i seguenti [ISO-N2424]:

- Codifica parametrica: Valore compreso tra 2.5 e 3.2, con prevalenza dell'HXVC sull'FS016;
- Codifica CELP: Valori compresi tra 2.7 e 3.8 per i CELP,sempre migliori di MPEG-2 AAC;
- Codifica T/F: Valori compresi fra 3.5 e 3.7 per i codec di tipo VQ, leggermente migliori di MPEG-2 AAC a bit rate inferiori a 64 Kb/s

Per quanto riguarda la comprensione del testo, i punteggi più alti sono stati sempre assegnati alla lingua svedese.

5.7 CODIFICA AUDIO SINTETICO

Anche per quanto riguarda l'audio sintetico, è supportato il concetto dello streaming di tipo multilayer. Le applicazioni principali che riveste il campo della codifica sintetica sono:

- Conversione di un file di testo in parlato (TTS – *Text To Speech*);
- Linguaggio di trasmissione di musica generata da computer (SAOL - *Structured Audio Orchestra Language*).

5.7.1 TTS

Se ci rechiamo alla stazione dei treni, ci rendiamo conto di come gli annunci si susseguano senza sosta, e di come siano in parte costanti, in parte variabili (in caso, ad esempio, di ritardi, gli orari di partenza ed arrivo cambiano). In una situazione del genere, una codifica di tipo TTS permetterebbe all'operatore di scrivere semplicemente le variazioni sulla tabella di marcia, e di affidare all'instancabile voce sintetica di un computer il compito di parlare; la stessa cosa può essere pensata negli aeroporti, dove addirittura gli annunci vengono detti in più lingue.
Software di questo tipo già esiste, ma MPEG-4 ha fatto molti passi in avanti: non solo voleva migliorare la modulazione delle parole, che al momento attuale risultano spesso incorrelate tra di loro, e quindi la naturalezza dell'espressione, ma anche aggiungere la componente video, sincronizzando il parlato con la codifica video sintetica.
Si proponeva anche la possibilità di sopprimere uno o più canali audio o modificare intonazione, pronuncia e timbro della parlata.
Proprio in questo ambito è stato studiato l'encoder e decoder TTS (Text To Speech), per permettere la conversione da semplice testo a parlato. In ingresso, viene fornita una stringa contenente del testo, nonché parametri quali sesso, età, velocità della parlata, durata dei silenzi e volume di voce. Il TTSI (Text To Speech Interface) interpreta tutti i dati provenienti in ingresso e li converte in uno stream audio. Contemporaneamente, viene identificato ogni fonema e convertito in uno dei FAP di tipo "viseme", tenendo conto anche dei fonemi precedenti e successivi in modo da creare una certa uniformità nel movimento facciale associato (vedi tabella ^?^). In tal modo, vengono generati due layer diversi, uno per l'audio ed uno per il video, che poi possono essere sincronizzati e trasmessi contemporaneamente.

Testo originale	Traduzione in fonemi
Once upon a time a child was born	w-uh-n-s-@-p-o-n-@-t-a-i-m-@-ch-a-i-l-d-w-o-z-b-oo-n

Tabella 7: Esempio di traduzione in fonemi a partire da una frase in inglese

5.7.2 SAOL

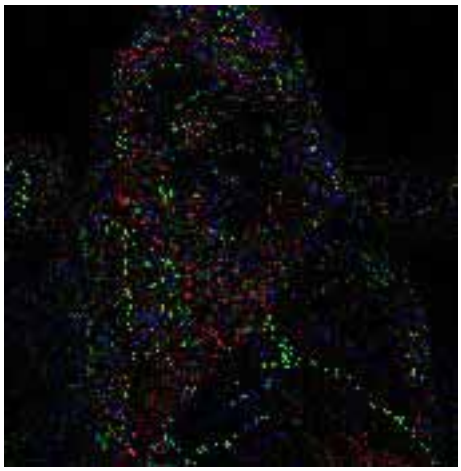
Un altro campo che riveste particolare interesse nel campo dell'audio sintetico è la trasmissione in maniera efficiente di musica generata al computer. A partire dal 1996 iniziarono gli studi a tal proposito, grazie alla collaborazione di Hughes Electronics che vide un demo del prodotto Csound.
Il Csound è un linguaggio usato per descrivere sintetizzatori di suono, e sulle basi di Csound e di un altro programma, NetSound, e con gli occhi puntati allo standard MIDI, nacque, tra il 1996 ed il 1997, SAOL, l'attuale standard di MPEG-4.
SAOL significa Structured Audio Orchestra Language ed è particolarmente ottimizzato per descrivere algoritmi di sintesi ed effetti digitali, anche se qualunque tipo di algoritmo può essere codificato in SAOL sotto forma di diagramma di flusso[Saol][Saol_2].
Con questo linguaggio innanzitutto si descrive il numero di strumenti che compongono l'orchestra (*orchestra chunk*); per ogni singolo strumento, poi, si descrive un algoritmo che lo modella, come per esempio la forma d'onda, l'ampiezza, il timbro, ma anche filtri digitali, interpolatori e così via.
Seguono poi i dati veri e propri, ovvero alcuni suoni già campionati da usare nel processo di sintesi, e gli *events*, che decidono quale strumento suonare, quando, con quale tempo ecc. In tal modo, non serve campionare un'intera scala musicale, ma solo il "la", ad esempio. Il la# od il la*b* possono essere ricostruiti con il processo di sintesi, e così il "sol" ed il "si": basta variare la frequenza d'uscita.
La cosa interessante di SAOL, che senza dubbio lo differenzia dal MIDI, è che i suoni non sono standardizzati. Ad esempio, se un file MIDI richiede lo strumento "pianoforte", questo avrà un suono diverso a seconda che il PC abbia una scheda audio con o senza wave table, o un sintetizzatore esterno collegato all'uscita MPU-401. Nel caso invece di SAOL, il brano verrà suonato indipendentemente dalla piattaforma, perché il suono è incluso nel file. Questo naturalmente permette l'invenzione di suoni nuovi, nonché la modifica di quelli già esistenti, ed è quello che gli autori si aspettano quando creano dei nuovi brani.
In ricezione, viene dapprima decodificato l'*orchestra chunk*; successivamente, i dati in arrivo vengono letti ed interpretati, nonché sincronizzati, da parte dello *scheduler*. Ogni strumento così emette un suono, e la somma di questi crea l'orchestra originale.
Naturalmente, SAOL da solo non basta perché mancano ancora le note del brano da suonare. Di questo si occupa SASL (Structured Audio Score Language), che appunto contiene le note, la loro durata (croma, biscroma ecc.), il volume e così via. La struttura di SASL è più leggera di quella

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

MIDI perché non ammette parametri come la ripetizione di battute o loop, ad esempio, ma occupa molto meno larghezza di banda.

Per poter migliorare la qualità del brano da riprodurre, si possono usare funzioni di mix e di post-produzione, come ad esempio specificare come mescolare la voce principale con il sottofondo musicale, introdurre un tremulo o uno sfumato. AudioBIFS è deputato a questo scopo e, come BIFS, è basato su un grafo i cui nodi rappresentano ciascuno una funzione particolare, come ad esempio "Mixa N canali", "Aspetta N secondi", "Esegui il suono", "Copia/Incolla" (per i loop) e "Applica il suono ad un oggetto 2D/3D". Per meglio chiarire quest'ultimo punto, c'è da dire che ad ogni nodo è associato un solo suono, che può essere codificato in modo diverso. Dunque, il basso potrà seguire una codifica MPEG-4 HILN, il piano una di tipo CELP e così via, e questo per ottimizzare le risorse di banda. Grazie alle peculiarità di AudioBIFS, si può associare un suono ad un oggetto, creando effetti speciali come il rumore di un auto mentre questa si allontana a video. Per uno studio dettagliato del formato dei files di saol e sasl, si veda la Bibliografia [Saol_3].

Figure 34-36: Apparentemente non esiste nessuna differenza fra l'immagine originale (a sinistra) e quella con il Watermarking (a destra). In realtà, uno XOR fra le due immagini mostra ciò che l'occhio umano non è in grado di percepire (sotto)



5.8 DMIF: DIRITTI DI PROPRIETÀ IN MPEG-4

Lo standard MPEG-4 fornisce anche dei meccanismi di protezione per salvaguardare il copyright e per permettere ad esempio la distribuzione a pagamento on-line di film o canzoni in formato MPEG-4.

A tal scopo, è stata standardizzata l'interfaccia IPMP (Intellectual Property Managment & Protection – Protezione e gestione della proprietà intellettuale), divisa in IPMP-D (IPMP-

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Descriptors – Descrittori IPMP) e IPMP-ES (IPMP Elementary Streams). In tal modo, tutti gli oggetti MPEG-4 protetti hanno un IPMP-D loro associato che fornisce informazioni su come gestirne e proteggerne il contenuto[ISO-N2614].

Oltre a ciò, MPEG-4 fornisce anche un meccanismo per identificare gli oggetti protetti tramite l' Intellectual Property Data Set (IPI Data Set). Questo identifica il contenuto o tramite sistemi standardizzati internazionalmente (p.es. ISBN, ISO ecc.) o tramite coppie di chiavi/valore (ad esempio, <Cantante/Elton John>). Così, se un utente vuole ascoltare un brano da un database a pagamento, in fase di decodifica vengono inizializzati BIFS, i descrittori degli oggetti (VOP,VOL, ecc.) e l'IPMP-ES. A questo punto, il client ed il server negoziano un protocollo di autenticità a chiavi e stabiliscono un canale sicuro attraverso il quale possono essere trasmesse le chiavi di decodifica. Le chiavi vengono trasmesse grazie all'IPMP-ES mentre la mappatura del contenuto e la sua associazione sono a carico degli IPMP-Ds. Finalmente, lo stream MPEG-4 viene decrittato.

Il processo appena descritto presuppone che le informazioni inerenti la crittazione dei dati vengano codificate in uno stream diverso da quello degli AVO, e che vi sia un puntatore a tutti gli oggetti sotto copyright. Un'altra possibilità, detta *Watermarking*, è quella di proteggere l'oggetto (immagine, suono, video) inserendo i dati IPMP nell'oggetto stesso tramite una chiave segreta. L'oggetto risulta impercettibilmente modificato, come si può vedere dalle figure 34-36, ma senza la decodifica della chiave risulta impossibile una corretta identificazione da parte del decoder.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

6 MPEG-4 FASE 2

Lo standard MPEG-4 Versione 2, nato nel dicembre 1999, è una versione ampliata rispetto a quella precedente e completamente compatibile verso il basso. Le modifiche apportate sono[ISO-N3075]:

- Introduzione di MPEG-J (supporto Java);
- Miglioramento di BIFS ed AudioBIFS, con nuove funzioni quali ad esempio supporto per gli accessi di tipo multiutente e calcolo degli echi di un suono a causa di riflessioni sulle pareti di una stanza;
- VRMLScript,
- Definizione di un formato file MP4;
- Maggiore robustezza nella codifica video naturale e codifica di immagini stereoscopiche;
- Codifica di mesh 3D;
- Maggiore numero di tools nella codifica audio.

6.1 MPEG-J

MPEG-J definisce le API per poter visualizzare i formati MPEG-4 tramite Java. L'applicazione viene trasmessa come ES separato dal resto di MPEG-4, consentendo a MPEG-J di accedere alle varie componenti del lettore MPEG-4 oltre alle funzionalità di base proprie di Java. Non è supportato il download del decoder.
Le varie API permettono di controllare il grafo ed i nodi, di cambiarli, aggiungerli o toglierli.

6.2 FORMATO DEL FILE MP4

Il formato MP4 è stato concepito per contenere una presentazione di tipo MPEG-4 in maniera flessibile ed estensibile in modo da facilitarne lo scambio, l'editing e la presentazione ai mezzi di comunicazione. La rappresentazione può essere sia "locale" che remota, tramite trasmissione via FlexMux indipendente dal protocollo di trasmissione. Il formato si basa sul protocollo QuickTime® di Apple Computer Inc.

Il file è composto da molte unità, dette *atomi* che descrivono indici, durate e puntatori ad altri atomi in maniera gerarchica; la presentazione vera e propria può essere contenuta nello stesso file MP4 oppure referenziata tramite un URL.
Una novità consiste nel fatto che il formato MP4 fornisce "suggerimenti" all'applicazione su come trasmettere i vari pacchetti tramite i FlexMux ed ottimizzare così le risorse di banda.

6.3 CODIFICA DI MESH 3D

MPEG-4 ha studiato una serie di tool atti alla trasmissione di oggetti 3D codificati con proprietà quali colori, texture, ombreggiature. In particolare, lo standard supporta la scalabilità del tipo LOD (Level Of Detail), permettendo rappresentazioni via via più semplici del mesh a seconda della distanza dall'osservatore, scalabilità spaziale e trasmissione di mesh 3D progressiva.

6.4 AUDIO IN MPEG-4 VERSION 2

I vari tool aggiunti in questa versione di MPEG-4 comprendono[ISO-N2803][ISO-N2604]:

- *Metodi di correzione di errore*: sono stati introdotti nuovi algoritmi per la codifica AAC, ed altri per il recupero di errori di trasmissione;
- *Contesto ambientale*: Questi nuovi tool permettono di aggiungere sempre più realismo ad una scena MPEG-4 creata tramite AudioBIFS. Le nuove proprietà introdotte comprendono la descrizione di un ambiente (come ad esempio un teatro) tramite parametri quali direzionalità del suono, caratteristiche dei materiali circostanti (in termini di riflessione e trasmissione), velocità del suono, elasticità delle pareti (con conseguente eco).
- *Migliore codifica*:
  - Viene ridotto il ritardo di codifica/decodifica, migliorando in tal modo le comunicazioni in tempo reale. Il nuovo codec deriva dall'AAC ed opera a frequenza massima di campionamento di 48 Khz;. La lunghezza del frame è al massimo di 480 campioni, contro i 960 dell'AAC, e viene migliorato il problema dei pre-eco;
  - La codifica BSAC (*Bit-Sliced Arithmetic Coder*), che deriva dall'AAC, fornisce scalabilità in bit rate a passi di 1 Kb/s per ogni canale audio;
  - Viene introdotto l'HLIN, in grado di cambiare velocità di riproduzione e di timbro tramite un nuovo segnale parametrico, eliminando i precedenti algoritmi

- Combinazioni con i precedenti codec (come HVXC) sono possibili (vedi ad esempio la già citata interazione tra HLIN ed HVXC);
- Nella codifica CELP, viene migliorata la bit rate comprimendo i silenzi. A tal scopo viene impiegato un rilevatore di attività vocale.

7 MPEG-7

Al giorno d'oggi, sono disponibili sempre più informazioni audiovisive da fonti dislocate in tutto il mondo. Le informazioni possono essere rappresentate in varie forme, e possono essere immagini ferme, video, grafica, modelli 3D, audio, parlato. Mentre le informazioni audio e video vengono usate generalmente dall'uomo, c'è un numero sempre maggiore di casi in cui le informazioni audiovisive vengono create, scambiate, ricevute e riusate da apparati elettronici. Scenari possibili sono ad esempio il riconoscimento dell'immagine (sorveglianza, telecamere intelligenti ecc.), conversione (da testo a immagine, da testo a parola ecc.) o ricezione (motori di ricerca utili a selezionare solo le informazioni che ci interessano).
La rappresentazione allora di tutte le fonti audiovisive create dall'uomo non può fermarsi ad una sola forma di codifica, sia essa planare (come MPEG-1 e MPEG-2) o ad oggetti (come MPEG-4), ma deve saper andare al di là, deve cioè in qualche modo essere interpretata, capita. In tal modo, un videoregistratore potrebbe essere programmato per registrare tutti i programmi sportivi della giornata tramite informazioni di carattere audiovisivo.
MPEG-7 si prefigge lo scopo di descrivere i dati del contenuto multimediale. Lo standard non studia un'applicazione in particolare, ma supporterà il più vasto numero di applicazioni possibili.

7.1 STRUTTURA DI MPEG-7: CODIFICA TRAMITE DESCRITTORI

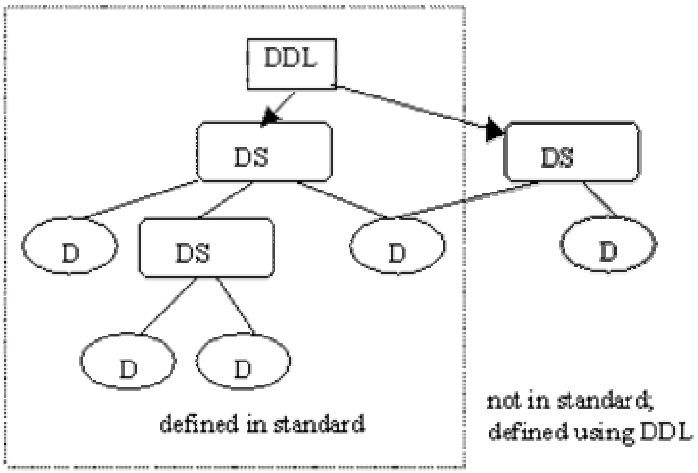


Figura 37: DDL e DS in MPEG-7. Come si vede, i DS possono essere definiti nello standard o definiti come nuovi tramite i DDL

Lo studio di MPEG-7 iniziò nell'ottobre 1996, con il titolo ufficiale di "*Multimedia Content Description Interface*", e dovrebbe diventare standard nel 2001. L'idea era quella di estendere le capacità limitate o fuorvianti di soluzioni proprietarie nell'identificazione di un contenuto (si pensi che alla parola

"automobile" Altavista™ risponde con qualche milione di URL, rendendo la ricerca quasi impossibile), e per fare ciò era necessario introdurre dei nuovi *descrittori (Descriptors)*. Questi descrittori possono anche essere raggruppati tra di loro in strutture predefinite, chiamate *schemi descrittivi (Description Schemes)*; la creazione di nuove strutture avviene tramite uno speciale linguaggio chiamato *linguaggio di definizione della descrizione (Description Definition Language, DDL)*. In tal modo, la ricerca del materiale può avvenire in maniera molto più rapida ed efficiente[ISO-N2861]. Infine, MPEG-7 include anche una *rappresentazione codificata di una descrizione*, ad esempio per un accesso facilitato o per efficienza di memorizzazione (vedi figura 37).

I descrittori di MPEG-7 non dipendono dal modo in cui l'oggetto è codificato o memorizzato, e quindi bene si adattano a formati da MPEG-1, MPEG-2 o MPEG-4 (a cui si è ispirato, riprendendo il concetto di "content-based"), a PCM od analogici, da filmati DVD a semplici rappresentazioni cartacee.

A seconda delle applicazioni e del contesto delle stesse, spesso vengono richieste informazioni e descrizioni diverse, e questo implica che uno stesso materiale possa essere descritto in vari modi. Così, a basso livello una scena può essere codificata a partire dalle caratteristiche dei vari oggetti presenti (una palla, una bambina, un cane), dal loro vettore di movimento, dai suoni (abbaiare, ridere); ad alto livello, interviene invece una codifica semantica della scena (*C'è una bambina che gioca a palla con il cane; la bambina ride ed il cane abbaia*).

Oltre a questo tipo di informazioni, ne possono essere aggiunte anche altre[ISO-N3158]:

- *Formato*: per esempio, JPEG, MPEG-2 ecc. Questo serve per sapere se l'utente è in grado o meno di poter decodificare il dato multimediale;
- *Accessibilità*: Copyright, prezzo;
- *Classificazione*: include generi come ad esempio Western, Azione, Commedia; Country, Pop, Folk; oppure controlli sulla censura (V.M. 14, V.M. 18 ecc.);
- *Collegamenti ad altri materiali interessanti*: per un facile accesso a dati simili (ad esempio, una canzone Country di Sherrié Austin potrebbe avere un link a quella di Kenny Chesney);
- *Contesto*: Può anche essere utile un contesto nel caso di eventi generici (ad esempio, nelle Olimpiadi del 1996, il contesto potrebbe essere "*salto con l'asta*" o "*100 metri piani*").

in molti casi è preferibile l'aggiunta di testo in modo da permettere un'informazione più esauriente, anche se, a differenza della lista sopra descritta, questo non è indipendente dalla lingua. Un esempio potrebbe essere il nome dell'autore, di un regista, di una città.

I dati di MPEG-7 possono essere direttamente associati al materiale audiovisivo, come pure trovarsi in un database lontano; devono essere pertanto studiati meccanismi biunivoci per consentirne il reperimento.

Per meglio chiarire il concetto, la figura 38 mostra una ipotetica catena di tipo MPEG-7. Le ellissi rappresentano delle azioni (codifica o decodifica, ad esempio), mentre i rettangoli simboleggiano

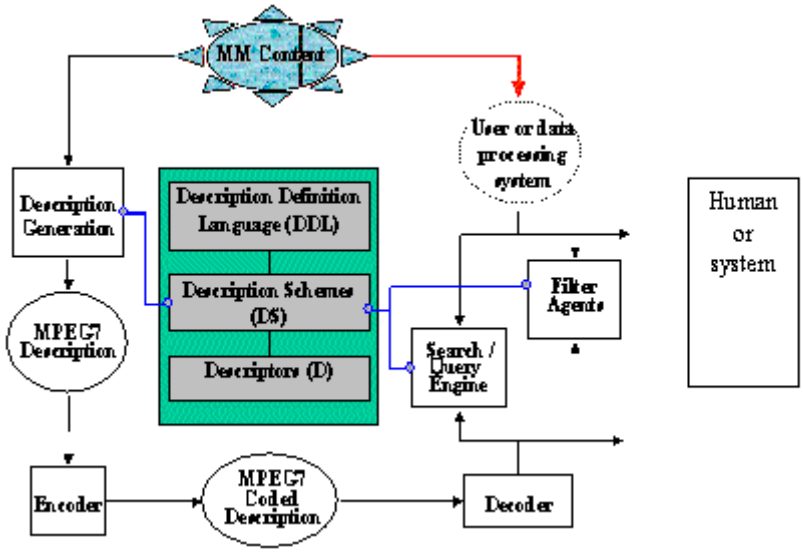


Figura 38: Relazioni e collegamenti tra le varie parti di MPEG-7

elementi statici, come una descrizione. Naturalmente, altri parametri, come il testo, possono essere aggiunti, ma questo è solo un esempio.

7.2 AREE DI INTERESSE

Sono molte le aree di interesse che potrebbero beneficiare dello standard MPEG-7. Alcune di queste includono[ISO-N2327][ISO-N2328]:

- *Biblioteche digitali* (catalogo immagini, canzoni…)
- *Elenchi multimediali* (pagine gialle, elenchi telefonici…)
- *Servizi di broadcast* (radio, TV…)

Le potenziali applicazioni potrebbero quindi riversarsi su settori quali:

- *Giornalismo* (ad esempio, per cercare discorsi di uomini politici tramite il loro testo, la loro voce, la loro immagine);

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

- *Informazioni turistiche* (per avere informazioni su una città a partire da una foto);
- *Servizi culturali* (gallerie d'arte, musei…);
- *Intrattenimento* (ricerca di un gioco);
- *Servizi di investigazione* (ricognizione facciale);
- *Sorveglianza* (controllo sul traffico, antifurti…);
- *Acquisti* (per cercare un vestito di un determinato tipo o colore);
- *Architettura, design*;
- *Archivi* per filmati, video, radio.

MPEG-7 non standardizza il modo in cui questi dati dovranno essere usati per rispondere alle domande dell'utente, e dunque non dà nessuna indicazione su come creare i motori di ricerca. In linea di principio, comunque, il materiale potrà essere richiesto usando video, musica, parlato e qualunque altra forma di tipo multimediale. Esempi possibili di domande sono:

- *Musica*: Si suonano delle note e si ottiene in risposta la canzone ricercata o quelle che più presentano un'analogia;
- *Grafica*: Si disegnano delle linee a video e si ottiene in risposta una serie di immagini contenenti immagini, diagrammi o logo simili;
- *Immagini*: Si definiscono degli oggetti, compresi colori e texture, e si ottengono in risposta vari esempi con i quali costruire l'immagine voluta;
- *Movimento*: Dato un gruppo di oggetti ed una descrizione del loro movimento, si riceve in risposta una lista di oggetti con le stesse caratteristiche;
- *Scenario*: A partire da una descrizione di uno scenario, si ottengono scenari simili;
- *Voce*: Usando un estratto di una canzone di Mina, si ottiene un database di tutti gli album della cantante o dei suoi videoclip.

7.3 LO STANDARD MPEG-7

Lo standard MPEG-7 è suddiviso nelle seguenti parti[ISO-N3158]:

- *Systems*: Sono tutti i tool necessari all'efficiente trasporto e memorizzazione delle descrizioni MPEG-7, nonché alla sincronizzazione tra il contenuto e le descrizioni. Include inoltre i tool relativi alla protezione di proprietà intellettuale;

- *Description Definition Language*: descrive il linguaggio per definire nuovi Description Schemes e forse nuovi altri descrittori;
- *Audio*: Sono i descrittori (D) e gli schemi dei descrittori (DS) che si riferiscono solo a descrizioni di tipo audio;
- *Video*: Sono i descrittori e gli schemi dei descrittori che si riferiscono solo a descrizioni di tipo video;
- *Generic Entities and Multimedia Definition Schemes (MMDS)*: Sono i descrittori e gli schemi dei descrittori che si riferiscono a caratteristiche generiche ed a descrizioni multimediali;
- *Reference Software*: Tratta l'implementazione software delle parti rilevanti dello standard MPEG-7;
- *Conformance*: Linee guida e procedure per verificare che le implementazioni MPEG-7 create siano conformi allo standard.

Analizziamo ora le parti in modo più appofondito.

7.4 SYSTEMS

Sono tutti i tool necessari all'efficiente trasporto e memorizzazione delle descrizioni MPEG-7, nonché alla sincronizzazione tra il contenuto e le descrizioni. Include inoltre i tool relativi alla protezione di proprietà intellettuale. Al momento non sono stati ancora definiti i tool.

7.5 DESCRIPTION DEFINITION LANGUAGE (DDL)

I tool principali usati per implementare le descrizioni in MPEG-7 sono i DDL, i D ed i DS[ISO-N3201]:

- I *descrittori* (D) associano una serie di possibili valori ad una certa caratteristica dell'oggetto da catalogare;
- Gli *schemi dei descrittori* (DS) sono dei modelli di oggetti multimediali e specificano quali descrittori possono essere usati per definirli, e le interazioni tra i descrittori stessi o altri DS.
- Il *linguaggio di definizione della descrizione* (DDL) definisce in questo contesto le regole sintattiche per specificare gli schemi di descrizione e la loro interpretazione.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Quando questo linguaggio iniziò ad essere studiato, molte soluzioni erano già presenti, ma nessuna sembrava particolarmente efficace né stabile, e così il gruppo DDL decise di crearne uno completamente nuovo. Dopo un po', il consorzio W3C, già autore del linguaggio HTML, iniziò lo sviluppo di un linguaggio simile, dal nome XML (Extensible Markup Language), ed il gruppo DDL vi si affiancò. Il vantaggio fu l'interscambio tra i tue tipi di linguaggi, ma questa "unione di intenti" portò anche a dei problemi. Infatti, i tempi di sviluppo di DDL dipendevano da quelli di XML, e quindi dal consorzio W3C; inoltre, il linguaggio XML è di proprietà di W3C, e questo rende problematico creare delle estensioni specificatamente per MPEG-7. La soluzione finale è stata quella di lavorare in parallelo, per poi mantenere due standard comunque diversi tra di loro. Tra gli scopi di DDL c'è quello quindi di rimanere indipendenti da XML, ma di esserne in ogni caso compatibili. Inoltre, verranno definite solo le funzionalità di base, per poi estenderle in futuro e solo quando sia davvero necessario.
Il linguaggio XML (Extensible Markup Language) è un linguaggio molto simile ad HTML, con la differenza che i vari *tag* (i campi racchiusi tra i simboli "<" e ">") possono essere definiti dall'utente. Un esempio di documento in XML è il seguente:

```
<letter>
<header>
<name>Sig. Mario Rossi</name>
<address>
<street>Via Indipendenza, 10</street>
<city>Bologna</city>
</address>
</header>
<text>Caro Sig. Bianchi,…</text>
</letter>
```

In questo esempio, i tag <letter> e </letter> definiscono l'inizio e la fine di una lettera, così come <address> e </address> dichiarano che all'interno di questi tag è presente un indirizzo.
Oltre a queste caratteristiche, XML prevede un tipo di definizione di documento, chiamato DTD (*Document Type Definition*), che fornisce attributi specifici per determinati tag. Ad esempio, in

<book title=”La divina commedia” type=”compilation”>
<novel num=”1”>…</novel>
<novel num=”2”>…</novel>
</book>

si definisce un libro (tag <book>) e si aggiungono degli attributi, quali il titolo ed il tipo di libro; e così il tag <novel> contiene il numero dei vari capitoli (1,2).

Lo scopo principale di DDL è quello di definire i DTD sia per i Description Shemes che per i Descriptors, controllare la sintassi del linguaggio e definire una maniera facile di scrivere le definizioni di schema (tramite ciò che sarà l’*Attribute Group*). Attualmente, lo studio è ancora in corso[ISO-N2730], e continuerà fino al meeting del Febbraio 2001 all’Universita di Parigi.
I *tag* finora introdotti ed approvati sono i seguenti[ISO-N3201]:

- <schema>
- <attribute>
- <attrGroup>
- <DSType>
- <DType>

Vediamone una breve descrizione :

7.5.1 <SCHEMA>

Contiene la versione, l’URI (Uniform Resource Identifier) che identifica lo schema, dal nome *targetNamespace*, ed altri URI (chiamati *xmlns*) che puntano a DDL esterne. Ad esempio,

<schema targetNamespace=http://www.mpeg7.org/GenericDS.ddl version=”1.0”
xlms=http://www.mpeg7.org/mpeg7ddl>
…
…
</schema>

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

7.5.2 <ATTRIBUTE>

Questo tag specifica gli attributi di una determinata variabile. Questa può essere un intero, un valore booleano di tipo *true*|*false*, od un altro tipo di dato definito altrove. Ad esempio,

<attribute name=”Variabile1” datatype=”integer” value=”30”>

Vari tipi di *datatype* sono stati definiti. Alcuni tra questi sono “string”, “boolean”, “binary”, “uri”, “language” ecc.

7.5.3 <ATTRGROUP>

Riunisce vai tag di tipo <attribute> tra loro all’interno di una definizione DS o D:

<attrGroup name=”Gruppo di riferimento”>
<attribute ….>
<attribute …>
</attrGroup>

7.5.4 <DSTYPE>

Questo *tag* definisce la struttura e la sintassi per nuovi tag. Può includere altri DS e D, localizzati altrove, nonché altri DSType, a patto che la definizione non entri in un loop ricorsivo infinito. Ad esempio, il codice

<DSType name=”Nuovo DS”>
<DSTypeRef=”v1” type=”integer”>
<DSTypeRef=”h1” type=”Secondo DS”>
</DSType>

permette l’introduzione dei nuovi tag *<v1>…</v1>* e *<h1>…</h1>*.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

7.5.5 <DType>

In maniera molto simile a <DSType>, questo tag definisce nuovi Descriptors. Al momento non è ben specificata la differenza tra D e DS, ed MPEG-7 si propone di chiarire il concetto in futuro.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

7.6 AUDIO

Il campo dell'audio, può essere diviso in tre grandi filoni sottoposti a studio da parte di MPEG-7: uno riguarda i tool di descrizione degli effetti sonori, uno quelli di descrizione degli strumenti e l'ultimo il riconoscimento vocale.

- Descrizione degli effetti sonori: Al 50° incontro tenutosi a Maui nel dicembre 1999 sono stati definiti i parametri oggettivi inerenti lo spazio e la metrica, ed adesso bisogna studiarne la validità mediante test soggettivi.
- Tool di descrizione degli strumenti: in questo campo, sono stati definiti parametri quali armoniche, sostenuti ed altro ancora, ma bisogna aspettare il responso umano prima di poterli integrare in MPEG-7.
- Riconoscimento vocale: Sono stati creati con successo programmi in lingua inglese in grado di riconoscere la voce e di tradurla in fonemi. Ora si sta studiando la versione tedesca.

7.7 VIDEO

I descrittori di tipo MPEG-7 inerenti il video si possono dividere in quattro grandi categorie:

- Colore
- Texture
- Forma
- Movimento

Ognuna di queste categorie presenta sofisticati descrittori che permettono di identificare meglio l'oggetto richiesto.

7.7.1 DESCRITTORI DEL COLORE

- *Codifica di colore (Color Space)*: questo descrittore definisce le componenti di colore di una certa applicazione, sia essa un'immagine od un istogramma (inteso come un singolo frame o somma di frame). Le componenti attualmente definite sono RGB (per i computer), YUV (per i

video), HSV, HMMD (molto simili alla percezione umana del colore), Monocromatica e trasformazione lineare riferita all'RGB.

- *Colore dominante (Dominant Color)*: serve ad identificare un'immagine mediante pochi colori dominanti. Quantizzando un'immagine in modo da estrarne solo i colori dominanti, è possibile soddisfare le richieste dell'utente e mostrare, ad esempio, le bandiere che contengono il rosso in quantità rilevante.
- *Istogramma del colore (Color Histogram*): Rappresenta le caratteristiche del colore di un dato visivo. La definizione, così generica, rende questo descrittore molto flessibile ed utile per molte applicazioni.
- *Quantizzazione del colore (Color Quantization)*: Stabilisce il fattore di quantizzazione del colore.
- *Istogramma del colore dei GoF/GoP (GoF/GoP Color Histogram*): dato che i materiali audiovisivi non sono solo delle immagini fisse, ma possono anche essere dei filmati, questo descrittore può, ad esempio, calcolare la media dei *color histogram* di ogni frame (Group of Frame / Group of Pictures).
- *Istogramma della struttura di colore (Color-Structure Histogram)*: Il suo scopo è inteso esclusivamente per immagini fisse, e grazie a questo descrittore, che contiene la struttura locale del colore, si può fare un confronto tra due immagini per vedere se sono uguali o no.
- *Schema del colore (Color Layout)*: Cattura la distribuzione spaziale del colore tramite una DCT su quadrati di dimensioni 8x8 pixel, ed è utile non solo per le immagini, ma soprattutto per gli sketch.
- *Istrogramma binario della trasformata di Haar (Haar transformed Binary Histogram*): Un altro modo, molto compatto, di rappresentare il colore.

7.7.2 DESCRITTORI DEL TEXTURE

- *Luminance Edge Histogram Descriptor*: Questo descrittore riguarda la distribuzione spaziale di quattro angoli direzionali ed uno non direzionale, ed è molto utile per indicizzare e recuperare immagini naturali.
- *Homogeneous Texture*: Il texture è una caratteristica molto importante in un'immagine, perché in sostanza ne descrive il contenuto. Infatti, si può pensare di suddividere un'intera immagine in molti texture diversi, per poi classificarli. Se per esempio un utente richiede, a partire da una

foto aerea, tutti i parchi pubblici, ecco che il texture del parco diventa un descrittore discriminante, permettendo all'utente una risposta adeguata alla propria richiesta.

- *Texture Browsing Descriptor*: Mentre il precedente descrittore fornisce una descrizione quantitativa utile ad una ricerca molto accurata, questo è più utile a fornire una rappresentazione del texture più simile alla percezione umana. Così, a seconda che la ricerca venga fatta da un computer o da un uomo, verranno impiegati i corrispondenti descrittori.

7.7.3 DESCRITTORI DI FORME

- *Quadrato circoscritto (Object Bounding Box)*: Questo descrittore descrive il più piccolo rettangolo che circoscrive l'oggetto in questione, sia esso 2D o 3D, in modo che le facce / i lati di questo siano paralleli agli assi principali dell'oggetto.

- *Descrittori riguardanti il contenuto dell'oggetto*: La forma di un determinato oggetto può essere composta da una certa regione o da somme di regioni contenenti dei buchi. Questo descrittore deve rappresentare il contenuto dell'oggetto ed anche eventuali piccole imperfezioni dello stesso che lo rendono apparentemente diverso (come per esempio si può notare in figura 39) Per evitare che un utente usi per i propri prodotti dei logo protetti da copyright, o che voglia registrare un marchio già registrato da altri, il descrittore non solo tiene conto dell'effettiva forma dell'oggetto, ma anche di eventuali marchi registrati.

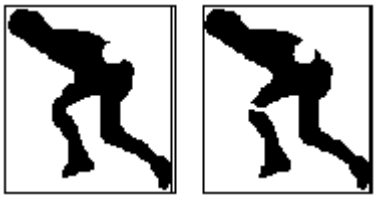


Figura 39: Piccole imperfezioni tra due figure uguali

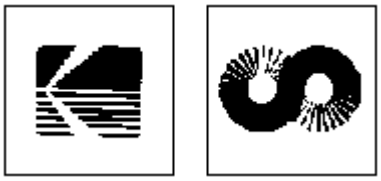


Figura 40: Esempi di logo

- *Descrittori riguardanti il contorno dell'oggetto*: La tecnica usata per rappresentare in maniera efficiente la forma di un oggetto si chiama CSS (*Curvature Scale Space* – Spazio della scala di curvatura) ed offre le seguenti caratteristiche:

Figura 41: Diverse forme, posizioni e scale devono identificare lo stesso oggetto

- Cattura molto bene le caratteristiche di un oggetto, e dei suoi simili;
- Riflette le proprietà visive umane;
- E' un sistema robusto a movimenti non rigidi e ad occlusioni parziali dell'immagine
- E' robusto a variazioni come zoom, movimenti di telecamere ed altro
- E' compatto

In tal modo, tutti gli oggetti mostrati nella figura 41 sono considerati validi nonostante le loro diverse forme, posizioni e scale.

7.7.4 DESCRITTORI DI MOVIMENTO

- *Movimento della telecamera (Camera Motion)*: Lo scopo di questo descrittore è quello di descrivere, per una certa sequenza, i movimenti effettuati dalla telecamera. Questi includono l'inquadratura fissa, la panoramica, la carrellata, la rotazione verso l'alto o verso il basso, lo zoom e così via. La descrizione di tali movimenti rende più facile identificare una certa scena richiesta dall'utente: ad esempio, se si cerca un primo piano, è probabile che l'azione precedente sia uno zoom seguito da un'inquadratura fissa, così come un panorama è caratterizzato da un'inquadratura panoramica.
- *Traiettoria di moto dell'oggetto (Object Motion Trajectory)*: Questo descrittore tiene conto della localizzazione spazio-temporale di un determinato oggetto. Tipicamente, si tratta di memorizzare i punti x,y,z,t ed, eventualmente, una o più funzioni di interpolazione che ne descrivano la traiettoria. Così facendo, è per esempio possibile attivare un sistema d'allarme nel caso in cui un oggetto attraversi un'area riservata o si muova a velocità troppo elevata. E' un descrittore indipendente dalla risoluzione spazio-temporale dell'oggetto (25Hz, 50Hz, CIF, SID, ecc) e quindi, anche esistendo diverse copie dello stesso oggetto, basta un solo set di descrittori per catalogarne il movimento; inoltre, è molto compatto e scalabile, riducendo in tal modo, a seconda delle esigenze, il numero dei punti da memorizzare e/o le funzioni di interpolazione.
- *Movimento parametrico dell'oggetto (Parametric Object Motion)*: I modelli di movimento parametrico sono già stati usati in MPEG-4 nella creazione di sprite e compensazione di moto. In MPEG-7, invece, vengono usati per descrivere degli oggetti con poche variabili. Tipicamente, viene ripreso il già visto concetto di modello affine, che permette il calcolo di rotazioni, traslazioni, ingrandimenti e riduzioni in maniera molto compatta; l'utente così potrà

facilmente ottenere delle risposte a domande quali “cerco un oggetto così fatto che trasla verso destra e poi si ingrandisce”.

- *Attività di moto (Motion Activity)*: Questo descrittore vuole catturare il concetto di “veloce” o “lento” così come viene percepito dall’uomo a seconda di una scena. Per esempio, “veloce” potrebbe includere sequenze come inseguimenti di auto o una partita di ping-pong; mentre “lento” potrebbe essere associato ad un telegiornale, un’intervista ecc. Naturalmente, possono anche essere aggiunti altri concetti, come “emozionante”, “triste”, “serio” e così via. Ad esempio, si potrebbe usare questo descrittore per riprendere le sequenze più interessanti di una gara di F1, come sorpassi, incidenti, sbandate mozzafiato ed altro ancora.

7.7.5 MULTIMEDIA DESCRIPTION SCHEMES (MMDS)

Lo scopo principale dei Multimedia Description Schemes, chiamati anche MMDS, è quello di descrivere, tramite DS e D, un documento audiovisivo. A tal fine, sono stati introdotti vari DS e D, divisi nelle seguenti 6 categorie che verranno esaminate in dettaglio:

- Segment DS;
- Semantic DS;
- Meta Informations DS;
- Media Informations DS;
- Summarization DS;
- Model DS;

7.7.6 SEGMENT DS

Questo descrittore nasce per definire un qualunque segmento di tipo audiovisivo tramite quattro sottoclassi che ne descrivono le proprietà. Il Segment DS può essere impiegato per rappresentare un documento audiovisivo completo sotto forma di albero.
Le quattro sottoclassi definite all’interno del SegmentDS sono:

- VideoSegmentDS: descrive una sequenza video;
- StillRegion: descrive un fermo immagine;
- MovingRegion: descrive una singola regione in movimento;

- AudioSegmentDS: descrive una sequenza audio;

7.7.7 SEMANTICDS

Definisce una struttura semantica che bene rappresenta un determinato oggetto audiovisivo. Le nozioni semantiche vengono suddivise in *oggetti*, con dimensioni spaziali, ed *eventi*, con dimensioni temporali, e funzionano come un indice in un libro: se per esempio in un oggetto AV compare *n* volte il concetto semantico di "telefono", il SemanticDS lo dichiara solo una volta per poi riferirsi all'oggetto "telefono" nel punto voluto.
Al momento, studi sono ancora in corso per definire questo tipo di descrittore.

7.7.8 MEDIA INFORMATIONS DS

Contiene tutti i tool specifici al tipo di immagazinamento dell'informazione. Per esempio, un concerto potrebbe essere registrato in MPEG-4 o su CD, ed è compito del Media Informations di mantenere tali informazioni. Questo descrittore contiene altri cinque sottocampi, che specificano l'identificazione del documento originale (*Media Identification DS*), la modalità di registrazione (*Media Format DS*), il tipo di codifica (*Media Coding DS*) e una descrizione su dove trovare il materiale (*Media Instance DS*).

7.7.9 META INFORMATIONS DS

Questo descrittore contiene un gruppo di DS e D che descrivono informazioni sul documento audiovisivo che non possono essere estratte automaticamente. I principali DS sono:

- PersonDS: Per la descrizione di persone (attore, regista…) o gruppi (squadra di calcio);
- Creation Meta Information DS: Per sapere da chi, quando e dove è stato creato il documento, perché è stato creato, in cosa consiste e dove trovare contenuti simili;
- Usage Meta Information DS: Informazioni sul copyright, su come usare il contenuto, ed informazioni di carattere finanziario

7.7.10 SUMMARIZATION DS

Questo descrittore serve a delineare il riassunto di un certo prodotto audiovisivo, mediante testo, immagini e possibilità di scorrere lungo il documento;

7.7.11 MODEL DS

Il Model DS fornisce un mezzo per descrivere l'analisi e la classificazione di materiale audiovisivo e la corrispondenza dello stesso con altri materiali e/o dati sintetici. A differenza degli altri, non è composto da vari DS, ma viene usato per descrivere due tipi di modelli:

- *Modello probabilistico*: viene usato per descrivere diverse funzioni statistiche e probabilistiche, tra cui il modello Gaussiano.
- *Modello analitico*: Fornisce un mezzo per descrivere proprietà di gruppi di oggetti, descrittori o classificatori in modo da definire dei concetti semantici. E' composto da vari DS, che descrivono funzioni come raggruppamento di oggetti, di descrittori, di concetti semantici in funzione di un database di esempi o di funzioni probabilistiche.

7.8 REFERENCE SOFTWARE

Questa parte di MPEG-7 consiste nello sviluppo di software di simulazione dei DS, D, DDL. Oltre a queste parti normative, è stata inclusa anche una parte non normativa, in quanto è necessario dover costruire un motore di ricerca ed un database dei dati per permettere da parte di un utente generico la ricerca delle informazioni audiovisive.
Le applicazioni del modello di sperimentazione (*eXperimentation Model, XM*) si suddividono in client e server. Nel caso server, viene implementato del software per estrarre i descrittori da un dato multimediale e poi scriverli in formato MPEG-7. Nel caso client, invece, viene effettuata la ricerca richiesta dall'utente a partire dal database di tipo MPEG-7 creato in precedenza.

7.9 CONFORMANCE

Questa parte di MPEG-7 fornisce linee guida e procedure per verificare che le implementazioni scritte in MPEG-7 siano compatibili con lo standard. Al momento, il Conformance Test non è ancora iniziato e verrà dunque definito in futuro.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

8 MPEG-21

Oggi come oggi non esiste nessun "quadro generale" che descriva tutte le attività di standardizzazione, già esistenti od in fase di progettazione, relative alla consegna di un prodotto elettronico.

Alla conferenza tenutasi a Maui (Hawaii) nel dicembre 1999, MPEG ha definito le prime idee sull'argomento, analizzando dal punto di vista del consumatore – e non da quello meramente tecnico – aspettative e bisogni.

Questa attività appartenente al gruppo SC29 è stata chiamata MPEG-21 ed ha come titolo "*Multimedia Framework*". Si propone di "raggiungere un comune accordo sull'infrastruttura relativa al multimediale per la consegna di contenuto elettronico con lo scopo di identificare dove siano richiesti nuovi standard appartenenti alla competenza di SC29".

Nella versione 0.2 del documento[ISO-N3162] sono stati individuati 12 "parametri chiave" legati alla consegna di dati multimediali. Questi parametri sono così suddivisi:

1. *Consegna tramite rete*: Bisogna tenere conto del rapporto qualità-prezzo (più grande è la larghezza di banda usata, e maggiore sarà il costo), dell'affidabilità della rete e degli eventuali errori in trasmissione;
2. *Qualità e flessibilità del servizio*: Il consumatore deve potersi fidarsi del contenuto che gli viene trasmesso e poter misurare in maniera oggettiva e soggettiva la qualità del servizio; inoltre, le informazioni spedite e ricevute devono essere protette contro un uso non autorizzato; le prestazioni richieste per un dato servizio devono essere descritte in termini comprensibili a tutti, e non con parole tecniche come "larghezza di banda"; l'utente deve poter accedere all'informazione voluta in qualunque modo, tempo e luogo.
3. *Qualità del contenuto*: Il contenuto dev'essere di alta qualità, non degradabile nel tempo (come nel caso dei nastri analogici);
4. *Qualità del contenuto (marche)*: Il prodotto può essere eventualmente firmato da ditte importanti per informare l'utente, in maniera soggettiva, che si tratta di un prodotto di qualità;
5. *Facilità d'uso*: L'utente deve poter capire l'interfaccia grafica (*User Interface – UI)* senza dover usare un manuale; l'interfaccia deve avere funzioni di aiuto/insegnamento, dev'essere semplice, intuitiva, indipendente dal linguaggio, conforme ad usi e costumi locali, portabile da dispositivo a dispositivo;

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

6. *Interscambio dei supporti fisici*: Il prodotto deve essere compatibile con tutte le marche del dispositivo di lettura/decodifica (es: tutti i produttori di CD-ROM); i formati devono essere compatibili verso il basso.
7. *Modelli di pagamento/abbonamento*: Si deve poter offrire materiale gratuito; ci può essere l'offerta di materiale in cambio di informazioni anziché in denaro; dev'essere possibile l'accesso al contenuto per un tempo limitato; il pagamento avviene solo alla consegna; l'utente si può abbonare a determinati servizi ottenendo degli sconti.
8. *Decodifica e visualizzazione su più piattaforme*: Il prodotto multimediale deve poter essere letto da dispositivi moderni o meno, deve quindi essere scalabile.
9. *Modalità di ricerca, filtri, recupero dei dati*: Bisogna creare dei database precisi, veloci, affidabili, che contengano il materiale richiesto e che lo localizzino in maniera efficiente e comprensiva; ci deve essere prova di autenticità del prodotto.
10. *Pubblicazione di materiali*: Anche il consumatore deve poter pubblicare un determinato prodotto per farsi conoscere; ci deve essere protezione nell'accessibilità dei dati.
11. *Diritti e doveri dei consumatori*: E' importante far capire ai consumatori ciò che possono fare e ciò che non possono fare una volta acquistati dei prodotti. I loro diritti/doveri devono essere standardizzati per ogni piattaforma onde evitare confusione. Bisogna preservare anche il copyright a causa della vulnerabilità dei prodotti digitali (come ad esempio il caso MP3).
12. *Privacy del consumatore*: In un ambiente di tipo *e-commerce*, le transazioni avvengono tramite carta di credito o firma elettronica. Questi dati, nonché informazioni personali quali generalità, domicilio, indirizzo di posta elettronica ecc. devono essere controllati dai fornitori di e-commerce per evitare fenomeni di pubblicità massiva (*spam*) o frodi (mediante uso illecito della carta di credito).

Come si può notare, molti punti sono già stati risolti dai protocolli precedenti: infatti, i punti 2) ed 8) si richiamano alla scalabilità; i punti 6) ed 8) sono in parte stati studiati in MPEG-4 e MPEG-2; il punto 9), si basa su MPEG-7; i punti 10), 11) e 12), infine, si riferiscono all'IPMP di MPEG-4.
Bisogna trovare i protocolli e gli standard mancanti, ed interfacciarli con quelli già esistenti ed è questo lo scopo del prossimo meeting di MPEG-21.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

9 CONCLUSIONI

In questa Tesi sono stati trattati la storia e lo sviluppo dello standard MPEG dalla sua nascita fino ai giorni d'oggi. Nel corso degli anni, MPEG ha dovuto affrontare diversi problemi e soddisfare le esigenze sempre più differenziate del pubblico, e la sfida è sempre stata vinta; MPEG è *lo* standard in assoluto e l'interesse di tutte le aziende del settore ne è la prova. Nelle applicazioni che richiedono, almeno in parte, un supporto hardware dedicato, MPEG non ha rivali ed infatti non esiste un film in formato CD-ROM che non segua lo standard MPEG-1, così come è impossibile trovare su DVD uno standard diverso da MPEG-2. La stessa televisione digitale si basa su ricevitori conformi allo standard nato allo CSELT, e non ad altri, e molte ditte hanno in questi ultimi anni creato molte schede nuove – basti citare le schede PCI/AGP per la decodifica del formato MPEG-2, le nuove schede per la compressione in MP3, il riproduttore di musica MP3 Rio© e sistemi di ripresa video ed audio in formato MPEG-4, come l'ultimo modello presentato al CeBit. Insomma, MPEG in questo settore non ha quasi rivali e nessuno sembra intenzionato a contrastarne il dominio. Solo negli USA il Dolby Stereo ha introdotto uno standard de facto tramite codifica in AC3 dei segnali audio, anche se sia negli Stati Uniti sia in Europa i ricevitori DVD sono compatibili sia con questo standard, sia con il suo "rivale" System 5.1.

Per contro, gli algoritmi implementati per la codifica e la decodifica sono molto complessi, e lo stato attuale della tecnologia non permette ottimi risultati anche nel campo del software puro. Riuscire a trovare un lettore di file in formato MPEG-2 è molto difficile; al momento esiste solo un programma in grado di farlo correttamente e lo stesso CSELT si è interessato al problema; i filmati in qualità MPEG-1 ormai non piacciono più al grande pubblico, e gli stessi siti Web preferiscono usare altri formati, quali ad esempio QuickTime® (più pesanti ma con migliore resa grafica); la decodifica di un bitstream audio/video in MPEG-4 è solo in parte possibile. Nel campo dell'audio, la codifica MP3 è ormai uno standard per Internet, e la decodifica avviene in tempo reale, ma la codifica in altri formati, come VQF o AAC, è lenta, e a volte non esistono software ancora adeguati allo scopo; i programmi per la gestione del TTS sono solo in coreano, mentre il decoder SAOL è lento e decodifica 4 minuti di musica all'ora su un Pentium III (operando invece in tempo reale tramite scheda hardware dedicata). Inoltre, bisogna dire che se qualunque PC è in grado di decodificare un file di tipo .WAV e quindi questo formato è ampiamente portabile, lo stesso non si può dire per tutti gli altri tipi di files appena citati, ognuno dei quali ha bisogno di un lettore

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

apposito. Non esiste ancora infatti un software in grado di gestire tutto ciò che ha costruito MPEG, e questo senza dubbio non ha favorito lo scambio.

Molto è stato fatto, ma molto resta ancora da fare, soprattutto nel settore software, anche se la qualità e la compressione dei bitstream sono sempre maggiori.

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

APPENDICE A: TABELLE COMPRESSIONI AUDIO E VIDEO

Qui di seguito viene riportato per completezza un confronto fra i rapporti di compressione dei formati MPEG sia per un segnale audio musicale (a partire da una codifica stereo 44,1 Khz), sia per un segnale video (a partire da una codifica di tipo CCIR-601).

Rapporto di compres-sione	WAVE	MPEG-1 Layer I 128 Kb/s	MPEG-1 Layer II 128Kb/s	MPEG-1 Layer III 128 Kb/s	MPEG-1 Layer II 64 Kb/s	MPEG-1 Layer III 64 Kb/s	MPEG-2 AAC 64 Kb/s	MPEG-4 VQF 64 Kb/s	MPEG-2 AAC 24 Kb/s	MPEG-4 VQF 24 Kb/s	SAOL	MP4	MIDI	VOTO
WAVE	1:1	0.07:1	0.07:1	0.07:1	0.04:1	0.04:1	0.05:1	0.04:1	0.02:1	0.02:1	0.01:1	0.05:1	0.001:1	5
MPEG-1 Layer I 128 Kb/s	16,6:1	1:1	1:1	1:1	0,75:1	0,75:1	0,77:1	0,75:1	0,29:1	0,29:1	0,22:1	0,79:1	0,01:1	ND
MPEG-1 Layer II 128Kb	16,6:1	1:1	1:1	1:1	0,75:1	0,75:1	0,77:1	0,75:1	0,29:1	0,29:1	0,22:1	0,79:1	0.01:1	2.3
MPEG-1 Layer III 128 Kb/	16,6:1	1:1	1:1	1:1	0,75:1	0,75:1	0,77:1	0,75:1	0,29:1	0,29:1	0,22:1	0,79:1	0.01:1	3.7
MPEG-1 Layer II 64 Kb/s	22:1	1,32:1	1,32:1	1,32:1	1:1	1:1	1.01:1	1:1	0,39:1	0,38:1	0,3:1	1,04:1	0.02:1	ND
MPEG-1 Layer III 64 Kb/s	22:1	1,32:1	1,32:1	1,32:1	1:1	1:1	1.01:1	1:1	0,39:1	0,38:1	0,3:1	1,04:1	0.02:1	ND
MPEG-2 AAC 64 Kb/s	21,6:1	1,30:1	1,30:1	1,30:1	0,98:1	0,98:1	1:1	0,98:1	0,38:1	0,37:1	0,29:1	1,03:1	0.02:1	3.4
MPEG-4 VQF 64 Kb/s	22:1	1,32:1	1,32:1	1,32:1	1:1	1:1	1.01:1	1:1	0,39:1	0,38:1	0,3:1	1,04:1	0.02:1	3.6
MPEG-2 AAC 24 Kb/s	56,6:1	3,41:1	3,41:1	3,41:!	2,57:1	2,57:1	2.62:1	2,57:1	1:1	0,97:1	0,76:1	2,7:1	0.05:1	ND
MPEG-4 VQF 24 Kb/s	58,4:1	3,52:1	3,52:1	3,52:1	2,65:1	2,65:1	2.37:1	2,65:1	1,03:1	1:1	0,79:1	2,78:1	0.05:1	ND
SAOL	74:1	4,45:1	4,45:1	4,45:1	3,36:1	3,36:1	3.42:1	3,36:1	1,30:1	1,26:1	1:1	3,52:1	0.06:1	5
MP4	21:1	1,26:1	1,26:!	1,26:!	0,95:1	0,95:1	0,97:1	0,95:1	0,37:1	0,36:1	0,28:1	1:1	0.02:1	5
MIDI	1139:1	68.7:1	68.7:1	68.7:1	51.8:1	51.8:1	52,1	51.8:1	20.1:1	19.5:1	15.6:1	54.3:1	1:1	5

Tabella 8: Confronto fra la compressione e la qualità di diverse codifiche audio

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Tutti i risultati della tabella sono sperimentali e si basano su compressioni effettuate tramite Xing™ ed il modello FhG per il protocollo MPEG-1, Yamaha TwinVQ Encoder™ per la codifica di tipo VQ, saolc.exe sviluppato al MIT per il saol ed il formato mp4, FAAC per la codifica in AAC.
Per quanto riguarda la codifica video, i risultati riassuntivi sono i seguenti:

Confronto	CCIR-601	MPEG-1	MPEG-2	MPEG-4 Sintetico	MPEG-4 Naturale
CCIR-601	1:1	0,05:1	0,05:1	0.01:1	0.04:1
MPEG-1	21:1	1:1	1:1	0.03:1	0.08:1
MPEG-2	21:1	1:1	1:1	0.03:1	0.08:1
MPEG-4 Sintetico	700:1	33:1	33:1	1:1	2.89:1
MPEG-4 Naturale	242:1	11.4:1	11.4:1	0.34:1	1:1

Tabella 9: Confronto fra le compressioni video di MPEG

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

APPENDICE B: SUONI ED ANIMAZIONI DI ESEMPIO

Nel CD-ROM si possono trovare i seguenti suoni ed animazioni:

Tipo di codifica	Audio	Video	Descrizione
MPEG-1	√	√	Filmato pubblicitario con Bugs Bunny
MPEG-4	√	√	Animazione facciale con sonoro
MPEG-4	√	√	Animazione facciale con effetti speciali
MPEG-2		√	Breve sequenza di ping-pong
MPEG-4		√	Animazione del corpo
MPEG-2 DVD	√	√	Trailer del film “Lost in space”
MPEG-2 DVD	√	√	Promo del Dolby Digital
MPEG 1		√	Nave della guardia costiera
MPEG-1		√	Nave della guardia costiera codificata ad oggetti in MPEG-4 e riconvertita in MPEG-1
MPEG-1		√	Ragazzo che parla
MPEG-1		√	Ragazzo che parla codificato ad oggetti in MPEG-4 e riconvertito in MPEG-1
MPEG-1		√	Filmato test usato nello studio della codifica in MPEG-4: interno di un ufficio
MPEG-1		√	Codifica ad oggetti del filmato precedente
MPEG-1		√	Filmato test usato nello studio della codifica ad oggetti in MPEG-4: presentatrice del telegiornale coreano
MPEG-1	√	√	Trailer del film “Small Soldiers”
MPEG-2		√	Panoramica su un giardino
MPEG-2		√	Donna che parla al telefono
MPEG-4	√		Codifica vocale in MPEG-4 effettuata allo CSELT
MPEG-4		√	Codifica video in MPEG-4 effettuata allo CSELT
MPEG-4		√	Video con rettangolo sintetico in rotazione
MPEG-4		√	Esempio di filmato interattivo: l’utente può spostare con il mouse un rettangolo presente sullo schermo
MPEG-4		√	Esempio di codifica video ad oggetti
WAVE	√		Sherrié Austin: “Lucky in Love”. Codifica 44100 Hz stereo (00:00”→ 00:10”)

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

MPEG-1	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo Layer I a 128 Kb/s - 44,1 Khz stereo (00:10”→00:20”)
MPEG-1	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo Layer II a 128 Kb/s, - 44,1 Khz stereo (00:20”→00:30”)
MPEG-1	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo Layer III a 128 Kb/s – 44,1 Khz stereo (00:30”→00:40”)
MPEG-1	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo Layer II a 64 Kb/s - 44,1 Khz stereo (00:40”→00:50”)
MPEG-1	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo Layer III a 64 Kb/s – 44,1 Khz stereo (00:50”→01:00”)
MPEG-2	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo AAC a 64 Kb/s, 22 Khz stereo (01:00 → 01:10”)
MPEG-4	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo TwinVQ a 64 Kb/s, 22 Khz stereo (01:10”→01:20”)
MPEG-4	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo AAC a 24 Kb/s, 16 Khz stereo (01:20”→01:30”)
MPEG-4	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo AAC a 24 Kb/s, 16 Khz stereo (01:30”→01:40”)
WAVE	√		Sherrié Austin: “Lucky in Love”. Codifica di tipo 44,1 Khz stereo (01:40”→01:50”)
MPEG-4 SAOL	√		Elpelele – Musica classica in formato MPEG-4
MPEG-4 SAOL	√		Battito di mani generati al computer
MIDI	√		Elpelele in formato MIDI
MPEG-4			Riproduzione di tutti i FAP per l’animazione facciale (file creato da Pockaj)
MPEG-4			Rappresentazione dei FAP caratteristici (file creato da Marceglia)
MPEG-4			Rappresentazione delle varie espressioni possibili tramite animazione facciale (file creato da Marceglia)

Tabella 10: Esempi video ed audio presenti nel CD-ROM

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

APPENDICE C: CONTENUTO DEL CD-ROM ALLEGATO

Lunghezza	Nome del file	Descrizione
	<CDROM>	
1.728.512	Evoluzione Degli standard MPEG.doc	*Tesi di Laurea di Luca Marceglia*
	<CDROM>\Audio\Mpeg-1	
160.081	Austin10-20.L1-128.mp3	*Codifica in MPEG-1 Layer I 128 Kb/s*
160.080	Austin20-30.L2-128.mp3	*Codifica in MPEG-1 Layer II 128 Kb/s*
160.067	Austin30-40.L3-128.mp3	*Codifica in MPEG-1 Layer III 128 Kb/s*
80.041	Austin40-50.L2-64.mp3	*Codifica in MPEG-1 Layer II 64 Kb/s*
80.025	Austin50-60.L3-64.mp3	*Codifica in MPEG-1 Layer III 64 Kb/s*
	<CDROM>\Audio\Mpeg-2	
81.409	austin60-70-22Khz.aac	*Codifica in MPEG-2 AAC 64 Kb/s*
31.168	austin90-100-16Khz.aac	*Codifica in MPEG-2 AAC 24 Kb/s*
	<CDROM>\Audio\Mpeg-4\Naturale	
80.180	Austin70-80.VQ-64.vqf	*Codifica in MPEG-4 VQF 64 Kb/s*
30.197	Austin80-90.VQ-24.vqf	*Codifica in MPEG-4 VQF 24 Kb/s*
	<CDROM>\Audio\Mpeg-4\Sintetico\saol	
7.720	claps.mp4	*File MP4 per applausi sintetici di MPEG-4*
2.391	claps.saol	*File SAOL per applausi sintetici di MPEG-4*
1.342	claps.sasl	*File SASL per applausi sintetici di MPEG-4*
40.401	elpelele.mid	*Elpelele in formato MIDI*
1.774.954	elpelele.mp4	*Elpelele in formato MPEG-4*
2.240.046	Elpelele.wav	*Conversione dal formato MPEG-4 (.MP4) a WAV di Elpelele (estratto)*
605.664	Note.wav	*Conversione da AIF a WAV dell'orchestra chunck di elpelele.mp4*
2.599	PC.mp4	*Files MP4, SAOL e SASL di suoni sintetici*
3.500	PC.saol	
30.323	PC.sasl	
395.226	samp_1.aif	*Singole note in formato AIFF estratte dall'orchestra chunk di elpelele.mp4*
119.354	samp_11.aif	
101.606	samp_13.aif	
303.754	samp_3.aif	
274.306	samp_5.aif	
265.888	samp_7.aif	
208.628	samp_9.aif	

409.656	saolc.exe	*Programma di conversione da files mp4 a AIFF sviluppato al MIT*
283.275	sax2.mp4	*Altri brani di esempio in MPEG-4*
334.241	scr1.mp4	
1.209	vowel.saol	*Files MP4, SAOL e SASL di suoni sintetici rappresentanti delle vocali (a-e-i-o-u)*
1.142	vowels.mp4	
1.057	vowels.sasl	
<CDROM>\Audio\Mpeg-4\Sintetico\tts		
2.587.631	mpeg4tts_execute.zip	*Programma principale per l'esecuzione del TTS*
13.172.537	tts-db1.zip	*Database in Coreano per la traduzioni in fonemi e visemi di un testo (Parte 1 e 2)*
13.388.664	tts-db2.zip	
<CDROM>\Audio\Wave		
1.764.044	Austin0-10.wav	*Campionamento a 44,1 Khz stereo in formato WAVE*
1.764.048	Austin100-110.wav	*Campionamento a 44,1 Khz stereo in formato WAV*
19.361.924	AustinCollage.wav	*Codifica del brano "Lucky in Love" di Sherrié Austin da vari formati MPEG-1, MPEG-2, MPEG-4 a WAVE*
<CDROM>\Documenti		
189.501	Ebrahimi.zip	*Touradj Ebrahimi, Caspar Horne, "MPEG-4 Natural Video Coding - An overview"*
315.061	FacialAnimation.zip	*Estratto dalla HomePage di Roberto Pockaj sull'Animazione Facciale*
1.532.407	MPEG-2 Tutorial.pdf	*MPEG-2: The basics of how it works*
347.085	mpeg-4_audio_v2_ver.zip	*"Report on the MPEG-4 Audio Version 2 Verification Test" - ISO/IEC JTC1/SC29/WG11/N3075, Dicembre 1999*
38.754	mpeg-4_ipmp.zip	*MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Applications Document" - ISO/IEC JTC1/SC29/WG11/N2614, Dicembre 1998*
561.083	mpeg-4__video_resilience.zip	*Report Of The Formal Verification Tests On MPEG-4 Video Error Resilience" – ISO/IEC JTC1/SC29/WG11/N2604, Dicembre 1998*
99.126	mpeg-7_c&o&tr.zip	*"MPEG-7: Context, Objectives and Technical Roadmap, V1.2" - ISO/IEC JTC1/SC29/WG11/N2861, Luglio 1999;*

131.629	mpeg-7_ddl.zip	*"DDL Working Draft 1.0" - ISO/IEC JTC1/SC29/WG11/N3201, Dicembre 1999*
181.470	saolc.pdf	*"About Structured Audio"*
104.960	Storia dell'MPEG.ppt	*Leonardo Chiariglione, Luglio 1999, "MPEG: A Balance of 10 years"*
358.920	Tekalp.zip	*Murat Tekalp and Jörn Ostermann, "Face and 2-D Mesh Animation in MPEG-4"*
3.150	Ttsi.zip	*Esempio di file di testo in inglese e traduzione in fonemi*
322.418	Using SI Tables.pdf	*Hewlett-Packard, "MPEG-2: The basics of how it works"*
2.098.096	W2203.zip	*"MPEG-4 Information technology Coding of audio-visual objects: Part 3: Audio" - ISO/IEC JTC1/SC29/WG11/N2203, Maggio 1998*
1.241.088	W2424.doc	*"Report on the MPEG-4 speech codec verification tests" – ISO/IEC JTSC1/SC29/WG11/N2424, Ottobre 1998*
720.607	W2503.pdf	*Come programmare il SAOL (tratto dal documento N2503)*
275.348	w2730.zip	*"Results of MPEG-7 Technology Proposal Evaluations and Recommendations" - ISO/IEC JTC1/SC29/WG11/N2730, Marzo 1999*
898.396	W2803.ZIP	*"MPEG-4 Audio Version 2" - ISO/IEC JTC1/SC29/WG11/N2803, Luglio 199*
558.427	W2804.zip	*"MPEG-4 Information technology Coding of audio-visual objects: Part 4: Conformance testing" - ISO/IEC JTC1/SC29/WG11/N2804, Marzo 1999*
80.384	W2805.doc	*"MPEG-4 Simulation Software" – ISO/IEC JTC1/SC29/WG11/N2805, Luglio 1999*
95.288	W3158.zip	*"Overview of the MPEG-7 Standard" - ISO/IEC JTC1/SC29/WG11/N3158, Dicembre 1999*

11.347.167	Xm.zip	*Experimentation Model di MPEG-7*
<CDROM>\Programmi		
104.775	faac061.zip	*Encoder FAAC (Freeware AAC) versione 0.61*
67.829	in_aac05.zip	*Plugin per Winamp per la decodifica dei files AAC*
3.826.256	powerdvd255trial.exe	*Lettore di files MPEG*
543.292	tvq-e215j.exe	*Yamaha TWIN-VQ Encoder (versione Giapponese)*
3.492.773	ueshw18.zip	*Ultimate Encoder 1.8*
1.685.239	vqe254b6e.exe	*Yamaha VQ Encoder/Decoder*
1.475.084	vqp251b1j.exe	*Yamaha VQ Player & Plugin*
354.028	winamp-vqf10.zip	*Plugin per WinAmp per la decodifica dei files VQF*
1.904.640	winamp25e_full.exe	*WinAmp 2.5e*
1.150.976	WinAsCelpDemo202.exe	*Encoder/Decoder CELP*
1.920.862	xme220t.exe	*Xing MPEG encoder*
<CDROM>\Video\MPEG-1		
1.372.516	Akiyo_O.mpeg	*Filmati in MPEG-1*
5.948.767	rrgen.mpg	
21.761.468	smallsoldiers.mpeg	
<CDROM>\Video\MPEG-2		
2.817.474	Fiori.m2v	*Filmati in MPEG-2 (solo video)*
825.079	tek7.m2v	
2.815.841	Telefono.m2v	
242.198	tennis.m2v	
<CDROM>\Video\MPEG-2\DVD		
16.789.504	CLIP04.VOB	*Filmati in formato DVD*
28.788.736	DDTrain.vob	
<CDROM>\Video\Mpeg-4\Naturale		
2.140.919	coast_guard.mpg	*Filmati in MPEG-1 che mettono a confronto la codifica planare con quella ad oggetti*
797.562	coast_lays.mpg	
1.115.857	foreman.mpg	
961.579	foreman_lays.mpg	
342.075	hall_lays.mpg	
3.414.733	hall_monitor.mpg	
<CDROM>\Video\Mpeg-4\Naturale\Player		
70.522	AudioOnly-z.mpg4	*Codifica audio in MPEG-4*
306	AudioOnly.txt	*Documentazione per il file precedente*
182.272	DecFrame.dll	
29.696	DecLib.dll	
31.232	DECVDO.DLL	
133.120	G723.DLL	
1.146.368	IM1-2D.exe	*Lettore di files MPEG-4 sviluppato allo CSELT*
1.393.152	MFC42D.DLL	
97.555	MPEG4Video-z.mpg4	*Filmato in MPEG-4*
273	MPEG4Video.txt	*Documentazione del file precedente*
81.408	MSVCIRTD.DLL	
373.248	MSVCRTD.DLL	
295	TestInterpolators-z.mpg4	*Rettangolo sintetico in movimento*
472	TestInterpolators.txt	*Documentazione*
189	TestTouchSensor-z.mpg4	*Filmato interattivo in MPEG-4*

980	trace.log	
1.938	trace.txt	
2.295.162	VideoOnly-z.mpg4	*Filmato sullo CSELT in MPEG-4*
62.976	YUV2RGB.DLL	
<CDROM>\Video\Mpeg-4\Sintetico		
54.264	allfaps.fap	*Rassegna di tutti i FAP disponibili*
13.528	Espressioni.fap	*Esempi di espressioni facciali (creato da Luca Marceglia)*
945.216	ges_q31.mpeg	*Esempio di animazione del corpo*
22.062	LucaMarcegliaFAPS.fap	*Esempi dei FAP più significativi (creato da Luca Marceglia)*
18.632.964	movie1.qt	*Animazioni facciali con audio in Quick Time*
3.442.068	wowwow.qt	
<CDROM>\Video\Mpeg-4\Sintetico\FACE		
18.648	ac.table	
2.260	anim.h	
5.674	Arith_Coder.c	
9.917	Arith_Decoder.c	
999	codec.h	
731	coder.h	
19.451	dc.table	
2.750	decoder.h	
818	editor.h	
1.068	face.h	
5.318	FaceIcon.bmp	
1.868	face_res.h	
15.872	fa_soft_donation.doc	*Documentazione dei files presenti in questa directory*
139.712	glu32.dll	
4.707	ISTface.dsp	
537	ISTface.dsw	
300.032	ISTface.exe	*Programma per la visualizzazione dei file .FAP sviluppato a Lisbona*
766	ISTface.ICO	
14.682	ISTface.rc	
41.537	ISTfacepoints	
58.965	ISTfacevertx	
14.751	IST_anim_file.c	
10.099	IST_ArithDec.c	
32.285	IST_coder.c	
17.617	IST_DCTdec.c	
20.784	IST_decoder.c	
15.396	IST_Editor.c	
1.580	IST_expressions	
17.991	IST_face.c	
48.216	IST_FAPanim.c	
1.702	IST_logo.bmp	
8.282	IST_logo_big.bmp	
7.381	IST_OGLface.c	
733.296	opengl32.dll	
885	README.txt	

3.548	resource.h	
298	run.table	

Tabella 11: Contenuto del CD-ROM allegato

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com

BIBLIOGRAFIA

[bap] "MPEG-4 Body Animation Page" - http://ligwww.epfl.ch/mpeg4

[Birkmaier] Craig Birkmaier, "Chiariglione and the birth of MPEG" – http://www.cselt.it/leonardo/press/leonardo_ieee/prof.html;

[Chiariglione96] Leonardo Chiariglione, "MPEG and multimedia communications" – http://www.cselt.it/ufv/leonardo/paper/isce96.htm, Agosto 1996;

[Chiariglione99] Leonardo Chiariglione, "MPEG: achievements and future projects" – IEEE 1999

[Ebrahimi] Touradj Ebrahimi, Caspar Horne, "MPEG-4 Natural Video Coding - An overview"

[erg] "MPEG-2 Transmission", http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/mpeg2-trans.html

[Fogg] Chad Fogg, -"MPEG-2 FAQ" - http://bmrc.berkeley.edu/research/mpeg/faq/mpeg2-v38/faq_v38.html

[hp] Hewlett-Packard, "Using SI Tables to create Electronic Program Guides" - http://www.tm.agilent.com/tmo/pia/component_test/PIAApp/Notes/English/MPEGpaper1.html

[hp_2] Hewlett-Packard, "MPEG-2: The basics of how it works" - http://www.tm.agilent.com/tmo/pia/component_test/PIAApp/Notes/pdf/MPEGtutorial1.pdf

[ISO-14496-5] "MPEG-4 Reference Software" - ISO/IEC 14496-5

[ISO-N1559] "DSM-CC FAQ Version 1.0" - ISO/IEC JTC1/SC29/WG11/N1559, 21 Febbraio 1997;

[ISO-N1909] "Overview of the MPEG-4 Version 1 Standard" - ISO/IEC JTC1/SC29/WG11/N1909, Ottobre 1997;

[ISO-N2203] "MPEG-4 Information technology Coding of audio-visual objects: Part 3: Audio" - ISO/IEC JTC1/SC29/WG11/N2203, Maggio 1998

[ISO-N2327] "MPEG-7 Requirements Document V .6" - ISO/IEC JTC1/SC29/WG11/N2327, Luglio 1998;

[ISO-N2328] "MPEG-7 Applications" - ISO/IEC JTC1/SC29/WG11/N2328, Luglio 1998;

[ISO-N2424] "Report on the MPEG-4 speech codec verification tests" – ISO/IEC JTSC1/SC29/WG11/N2424, Ottobre 1998

[ISO-N2604] "Report Of The Formal Verification Tests On MPEG-4 Video Error Resilience" – ISO/IEC JTC1/SC29/WG11/N2604, Dicembre 1998

[ISO-N2614] "MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Applications Document" - ISO/IEC JTC1/SC29/WG11/N2614, Dicembre 1998

[ISO-N2730] "Results of MPEG-7 Technology Proposal Evaluations and Recommendations" - ISO/IEC JTC1/SC29/WG11/N2730, Marzo 1999

[ISO-N2803] "MPEG-4 Audio Version 2" - ISO/IEC JTC1/SC29/WG11/N2803, Luglio 1999;

[ISO-N2804] "MPEG-4 Information technology Coding of audio-visual objects: Part 4: Conformance testing" - ISO/IEC JTC1/SC29/WG11/N2804, Marzo 1999

[ISO-N2805] "MPEG-4 Simulation Software" – ISO/IEC JTC1/SC29/WG11/N2805, Luglio 1999

[ISO-N2861] "MPEG-7: Context, Objectives and Technical Roadmap, V1.2" - ISO/IEC JTC1/SC29/WG11/N2861, Luglio 1999;

[ISO-N2995] "MPEG-4 Overview" - ISO/IEC JTC1/SC29/WG11/N2995, Ottobre 1999;

[ISO-N3075] "Report on the MPEG-4 Audio Version 2 Verification Test" - ISO/IEC JTC1/SC29/WG11/N3075, Dicembre 1999

[ISO-N3158] "Overview of the MPEG-7 Standard" - ISO/IEC JTC1/SC29/WG11/N3158, Dicembre 1999;

[ISO-N3162] "First ideas on defining a Multimedia Framework (version 0.2) - ISO/IEC JTC1/SC29/WG11/N3162, Dicembre 1999;

[ISO-N3201] "DDL Working Draft 1.0" - ISO/IEC JTC1/SC29/WG11/N3201, Dicembre 1999;

[Kate92] Ten Kate, "Matrixing of bit rate reduced audio signals" – Conference on Acoustic, Speech and Signal Processing – 1992

[LeBuhan96] Corinne Le Buhan , "Software-embedded data retrieval and error concealment scheme for MPEG-2 video sequences", Proceedings of SPIE Conference on Electronic Imaging, Digital Video Compression: Algorithms and Technologies 1996

[Lee97] Ming-Chieh Lee, Wei-ge Chen, Chih-lung Bruce Lin, Chuang Gu, Tomislav Markoc, Steven I. Zabinsky, Richard Szeliski, "A layered Video Object Coding System Using Sprite and Affine Motion Model" – IEEE Transactions on circuits and systems for video technology, Vol. 7, no.1, Febbraio 1997;

[Noll97] "Peter Noll, "MPEG Digital Audio Coding" – IEEE Signal Processing Magazine, Settembre 1997;

[Otholit] Otholit, "Sub-Band coding" – http://www.otholit.com/pub/u/howitt/sbc.tutorial.html;

[Pockaj] "Animazione Facciale" – Roberto Pockaj Home Page - http://www-dsp.com.dist.unige.it/~pok/RESEARCH/index.htm

[Quackenbusch98] Schuyler R. Quackenbush, "Coding of Natural Audio in MPEG-4" – IEEE 1998;

[Saol] "About Structured Audio" – http://www-edlab.cs.umass.edu/~anramani

[Saol_2] MIT Media Laboratory - "MPEG-4 Structured Audio" – http://sound.media.mit.edu/mpeg4;

[Saol_3] Eric D. Schreier, "External Documentation and release notes for saolc" – MIT Media Laboratory, Agosto 1999

[Schaefer98] R. Schäfer, "MPEG-4: a multimedia compression standard for interactive applications and services" – Electronics & Communication Engineering Journal, Dicembre 1998;

[Sikora97] Thomas Sikora, "MPEG Digital Video Coding Standards" – Digital Electronics Consumer Handbook, McGraw Hill, 1997;

[Sikora97a] Thomas Sikora, "…" – IEEE Signal Processing Magazine, Settembre 1997;

[Sikora97b] Thomas Sikora, "The stucture of the MPEG-4 Video coding Algorithm" – http://wwwam.hhi.de/mpeg-video/papers/sikora/fmpeg4vm.htm,1997;

[Sikora97c] Thomas Sikora, "MPEG-4 Very Low Bit rate Video" – IEEE International Symposium on Circuits and Systems, 1997;

[Steinmetz] Steinmetz e Nahrstedt, "Audio Compression" – Capitolo 6, http://www.cs.sfu.ca/undergrad/CourseMaterials/CMPT479/material/notes/Chap4/Chap4.3/Chap4.3.htm

[Tekalp] Murat Tekalp and Jörn Ostermann,
"Face and 2-D Mesh Animation in MPEG-4"

INDICE

Luca Marceglia
luca@pianetaluca.com
http://www.pianetaluca.com