

Disclaimer: il seguente documento contiene degli appunti rielaborati, pertanto questo non sostituisce né il testo consigliato dal docente né tantomeno il materiale didattico fornito.

IL FORMATO AUDIO

MPEG: motion picture-coding experts group

L'MPEG è uno standard definito dalle organizzazioni ISO e IEC. I file MPEG definiscono più specificatamente: uno standard di compressione, decompressione, elaborazione, codifica.

Gli standard forniti da questo formato valgono per: file video, audio e contenuti multimediali.

MPEG si è evoluto nel corso degli anni, si osservino i suoi diversi formati che si sono susseguiti negli anni:

MPEG-1

Standard pensato per trasmettere le onde stereofoniche a 300 kbps (esso sfrutta quindi fino a 2 soli canali).

da 32 a 224 kbps -> compressione che questo standard era in grado di offrire

33, 44.1, 48 kHz -> tassi di campionamento che questo formato era in grado di offrire

MPEG-2

Questo formato consente di trasferire dati ad un bitrate maggiore (6 mbps circa), infatti esso impiegava 5 canali di trasmissione.

MPEG-3

Pensato per la TV-HD, però non è mai stato rilasciato. Il suo funzionamento è stato inglobato dalla versione precedente.

MPEG-4

Consente di modificare rumori principali e di sottofondo. Grazie a questo standard è stato infatti possibile focalizzarsi unicamente sui suoni in primo piano e definirli in maniera migliore.

NOTA-> gli MPEG fino al 7 si focalizzarono sul miglioramento dell'audio in generale (sia di fondo che principale)

MPEG-7

Questo standard definisce delle informazioni grazie alle quali è possibile cercare uno specifico audio all'interno di un database. Questo formato è stato necessario negli anni in cui internet esplose per consentire alla gente di individuare in rete le tracce audio volute.

MPEG-7 definisce solo questa agevolazione nella ricerca, non definisce degli standard circa la compressione, il campionamento e il filtraggio dei segnali audio; a causa di ciò venne creato MPEG-47, ossia l'unione fra il formato MPEG-7 e MPEG-4, in questa maniera gli standard per il trattamento dei segni audio furono integrati in MPEG-7 che offriva tali funzionalità mai viste. MPEG-7 fu implementato mediante XML.

MPEG-21

Questo standard fu realizzato al fine di definire un ambiente di lavoro nel quale sviluppare le applicazioni musicale. Esso definisce la tecnologia per: scambiare, commerciare e consumare elementi digitali.

MPEG-D

Diviso in 3 parti: MPEG surround; Spatial Audio Object Coding; Unified Speech and Audio Coding

Osservate tutte queste versioni che si sono susseguite negli anni (la prima vista nel 1988, l'ultima vista al 2007) ci sono alcune regole da tenere a mente:

1. Ogni volta che si sviluppa una nuova versione di MPEG, essa deve essere retro-compatibile con le precedenti; ciò significa che i file che adottano la versione nuova di MPEG devono funzionare sui dispositivi sui quali funzionava anche la versione precedente.

2. L'implementazione di una versione MPEG è libera per qualsiasi utente, tuttavia ogni versione sviluppata deve adottare i tassi di compressione, campionamento e decodifica definiti dagli standard.
3. Per utilizzare l'originale algoritmo di compressione del formato MPEG deve versare un pagamento a chi lo ha sviluppato per primo (istituto Fraunhofer).

MPEG-1

Utilizza 3 livelli (layers) di compressione, il terzo (che è anche il più esterno) è l'**mp3** layer.

-layer_1:

È il più interno; su questo vengono costruiti gli altri 2 layer.

In questo livello, dalla traccia audio che si vuole salvare in MPEG-1, vengono prelevati 384 campioni (quindi se la traccia audio ha N campioni, da questa se ne prelevano sempre 384).

Sui campioni prelevati si esegue una trasformata veloce di Fourier (FTT), passando dal dominio del tempo a quello delle frequenze, il segnale è più facile da comprimere perché i suoni diversi (una voce, un fruscio, un rumore...) sono più distinguibili rispetto al tempo.

Ottenuto questo spettro delle frequenze di 384 campioni, esso viene filtrato/quantizzato individuando 32 bande di frequenza uniformi (le bande sono 32 sempre per uno standard). Visto che ci sono 384 campioni per 32 bande, significa che per ogni banda sono catturati 12 campioni. Le 32 bande sono realizzate con un filtro polifasico.

Un filtro polifasico è in grado di campionare su differenti fasce di frequenza; in particolare, quello adoperato per questo formato, campiona su frequenze di: 33kHz, 44.1kHz e 48kHz. La grandezza della banda e il tempo di campionamento sono fattori che dipendono sempre dal tasso di campionamento adottato per svolgere le operazioni. Qualora il tempo di fra un campione ed un altro non sia superiore ai 2 millisecondi, si potrebbero verificare dei pre-echi.

Le 32 bande di frequenza utilizzate sono uniformi (ossia tutte della stessa grandezza), però Bark aveva scoperto che le bande di frequenza dell'organo di Corti sono non-uniformi (più sono elevate le frequenze, minore è la banda di frequenza che la contiene).

Alla luce di questi fatti, l'impiego di 32 bande uniformi fa sì che le frequenze più basse siano meno accurate (questo problema è già stato discusso quando si è trattata la quantizzazione). Infatti in MPEG-1, quello delle basse frequenze, era un problema irrisolto che verrà invece affrontato nelle versioni successive.

Per fare fronte a questa problematica, assumendo che ogni quanto fosse ampio 2db, si applica la compansion ad ogni banda di frequenza:

6 bit per l'ordine di grandezza + 4 bit per la classe di riquantizzazione

6 bit utilizzati per implementare una codifica a blocchi, arrivando ad ottenere un range dinamico di 128db.

4 bit (ossia 16 valori rappresentabili da 0 a 15) utilizzati per codificare 12 campioni, ogni campione verrà poi salvato su un bit:

bit 0 -> per le frequenze nulle (i silenzi)

bit 1 -> non utilizzato

bit 2 fino a 14 -> qui viene salvato uno dei 12 campioni contenuti nella banda

bit 15 -> lasciato libero (previdenza; esso può tornare utile per versioni future del formato)

La quantità di bit utilizzati per salvare i campioni di una banda, dipende dal rapporto segnale-rumore calcolato nella banda stessa. Ciò significa che per bande diverse si avrà un diverso numero di bit, diventa perciò necessario inserire un header in cima alla sequenza di bit associati ad una stessa banda, in maniera tale da agevolare il processo di decoding.

Durante la fase di decoding si utilizza un banco di filtri PCM.

-layer_2:

Ingloba il primo layer migliorandolo.

La banda di risoluzione della FFT aumenta. I tassi di campionamento di cui dispone lo standard vengono aumentati e vengono introdotti i cosiddetti "semi-tassi", ossia dei tassi di campionamento il cui valore è a metà di quelli originariamente definiti. I blocchi di campioni prelevati dal segnale nel dominio del tempo, da 384 passano a 1152 (sono 3 blocchi del livello 1 di di MPEG-1: mp1). La banda di frequenze è suddivisa in 3 regioni riquantizzate utilizzando più bit a seconda dell'ampiezza della frequenza che vi ci passa.

-layer_3:

Ingloba il secondo layer migliorandolo. Non utilizza più la FFT, ma una trasformata discreta del seno, ciò partiziona le frequenze in bande discrete, non più uniformi, aumentando la precisione del contenuto del file.

La quantizzazione qui è non-uniforme, nello specifico vengono adottate quella a codifica μ -law e A-law. Il bitrate è variabile fra i diversi blocchi.

Altre tipologie di formato sono:

-MPEG-AAC: incluso nel formato MPEG-4, esso supporta fino a 48 canali audio.

-Dolby AC-3: formato che applica una compressione di tipo percettivo, la casa Dolby (la produttrice di questo formato) lo ha sviluppato a partire da quello AAC.

-WMA: formato più performante di mp3, sviluppato dalla Microsoft

-FLAC: dispone dell'algoritmo di compressione più potente in assoluto; è ottimizzato per la compressione della voce

MIDI

Sigla che sta per: Musical Instrument Digital Interface

È un protocollo standard utilizzato nelle comunicazioni tra dispositivi hardware (computer, sintetizzatori...) a livello applicativo.

MIDI definisce: le regole per la comunicazione (aspetto tipico di un protocollo); l'interfaccia attraverso la quale fare avvenire questa comunicazione (a livello hardware); il formato dei dati scambiati (SMF: Standard MIDI File).

Implementando MIDI sui dispositivi è possibile mettere in comunicazione hardware di natura differente; MIDI offre inoltre delle velocità di trasmissione fissate. Il protocollo MIDI offre fra i suoi servizi: elevata performance musicale; una codifica universale dei dati; delle funzionalità riguardanti la trasmissione/ memorizzazione/ codifica/ decodifica dei dati.

Un dispositivo hardware in cui l'implementazione di MIDI è essenziale è il sequencer. Questo apparecchio è dotato di una memoria programmabile ed il suo utilizzo riguarda la registrazione ed esecuzione delle tracce audio; consente anche di editarle.

Bisogna però attenzione che MIDI non è responsabile della qualità dei suoni sui quali agisce, questa dipende solo e soltanto degli apparecchi che si interfacciano mediante MIDI (e anche dalla sorgente stessa del suono, oggetto o musicista che sia).

Si osservi adesso cosa si trova all'interno di un dispositivo per fare sì che esso sia in grado di riprodurre una traccia sonora:

I canali si occupano di dare un determinato timbro ai suoni da riprodurre, in questa maniera è come se si stesse digitalizzando il vero e proprio strumento musicale (ci possono essere fino a 16 canali, tutti i messaggi indirizzati ad uno specifico canale, verranno tradotti con il medesimo timbro).

la distinzione del contenuto di un audio è realizzata mediante le tracce, queste sono presenti in quantità da 1 ad N e sono capaci di individuare (per esempio) i diversi strumenti presenti in uno stesso brano musicale.

-Patch: implementano all'interno del dispositivo il concetto di timbro, possono esserci fino a 128 di questi.

-Banchi: consentono di incrementare il numero di patch disponibili, infatti 128 è un numero assai basso attraverso il quale non sarebbe possibile rappresentare ogni singolo timbro esistente in natura. I banchi bisogna immaginarli come collezioni di 128 patch ciascuno.

Da quanto detto fino ad ora si evince che i canali e i patch sono legati. Questi 2 oggetti sono legati da un concetto di "molti ad uno" -> ad un canale vengono associati più patch.

La traccia viene associata ad un canale i-esimo il quale poi riproduce un certo timbro andando ad individuare il patch necessario a dare quel timbro particolare, recuperandolo dal banco di patch.

Più tracce possono entrare all'interno di un canale contemporaneamente, ma un canale può produrre solo un timbro alla volta (esempio: se entrano 2 tracce in un canale in contemporanea, il canale può riprodurle sfruttando un solo timbro, ossia selezionando un solo patch).

MIDI suona le note col seguente procedimento:

una nota viene codificata con 2 messaggi, uno di inizio (note-on) e uno di fine (note-off); un parametro chiamato "velocity" caratterizza la rapidità con la quale avviene la riproduzione dall'inizio alla fine. Le eventuali variazioni di pressione nella riproduzione della nota, sono implementate dall'aftertouch. L'altezza della nota è invece realizzata mediante il pitch (un valore a scelta fra 128; essi sono più sufficienti se si pensa al fatto che un pianoforte dispone di 88 tasti).

I dispositivi MIDI hanno un loro clock per ordinare i messaggi. Il segnale di clock non usa i secondi come unità temporale bensì i "tick" (anche detti PPQ: parti-per-quarto) -> facendo in questa maniera si ottiene un'unità di tempo più elastica. (variano assumendo i valori delle potenze/ dei multipli del 2 partendo da 24 fino a $4096=2^{12}$).

BPM -> beat-per-minuto, indica il numero di quarti in un minuto

$60\text{sec} / N \text{ BPM} = \text{durata di un beat in secondi (Bs)}$

$Bs / N \text{ PPQ} = \text{durata di un tick in secondi (Ts)}$

esempio:

Siano dati 120BPM e 24PPQ; quanto dura un tick in questo sistema?

$Bs = 60 / 120 = 0.5\text{sec}$

$Ts = 0.5 / 24 = 0.02\text{sec}$

Nei diversi sistemi MIDI, l'ampiezza del PPQ è chiamata "division" (nell'esempio di prima essa era pari a 24); ad una division maggiore, corrisponde una maggiore risoluzione temporale della traccia audio, questo perché sarà possibile distinguere un maggiore numero di campioni all'interno di uno stesso quarto, quindi noni provenienti da sorgenti diverse/ di basse intensità stanno maggiormente distinguibili.

I messaggi MIDI sono sequenze di 10 bit; tuttavia il primo e l'ultimo bit sono usati come segnalino per indicare la fine e l'inizio dei dati riguardanti una precisa parola logica (una nota ad esempio), quindi i veri e propri dati sono contenuti negli 8 bit del tronco del messaggio.

[bit_start - bit1 - bit2 - bit3 - bit4 - bit5 - bit6 - bit7 - bit8 - bit_stop]

-bit1: serve per distinguere la tipologia di dato contenuta all'interno del messaggio MIDI; se esso vale 1, la sequenza di 8bit (ossia un byte) è detta "status byte" e contiene dei messaggi di comunicazione; se esso vale 0, la sequenza è detta "data byte" e contiene dei dati.

-bit2 fino a bit4 = Nibble1

-bit5 fino a bit8 = Nibble2

Gli status byte si suddividono in:

-messaggi di canale: comunicano delle operazioni percettive che si vuole il dispositivo MIDI compia (riprodurre una nota, cambiare il timbro in uscita...)

-messaggi di sistema: comunicano delle operazioni temporali riguardanti i suoni in uscita (ad esempio di sincronizzazione)

Il Nibble1 si occupa di specificare, qualora il bit1 sia impostato ad 1, la tipologia dello status byte ricevuto (se di canale o di sistema).

Nibble1 è una sequenza di 3 bit: da 000 a 110 indica le diverse operazioni di canale; 111 indica invece un'operazione di timing (sincronizzazione).

Nibble2 è una sequenza di 4 bit: nel caso di messaggi di canale, serve ad indicare a quale canale è destinato il messaggio (infatti con 4 bit si possono rappresentare 16 numeri e, guarda caso, è stato detto che su un dispositivo possono stare fino a 16 canali); nel caso dei messaggi di sistema, il messaggio specifico rivolto all'interno sistema è codificato su questa sequenza di bit.

I messaggi di canale riguardano uno solo dei canali del sistema, mentre quelli di sistema tutto il dispositivo.

La differenziazione in Nibble1 e 2 c'è solo nel caso degli status byte; se a transitare sono dei dati, i 7 bit rimanenti (perché il bit1 sarà fissato a 0) saranno indiscriminatamente impiegati per la memorizzazione dei dati in transito.

Si osservino di seguito i diversi messaggi di canale associati alle diverse sequenze dei bit del nibble1:

[001] -> note on

Messaggio che comunica di suonare una nota attraverso un canale.

Assieme a questo messaggio vengono mandati 2 data byte: uno contenente il valore di velocity (avendo 7 bit a disposizione, questo è un valore che varia da 0 a 127); uno contenente il valore di pitch (avendo 7 bit a disposizione, questo è un valore che varia da 0 a 127).

[000] -> note off

Messaggio che comunica di smettere di suonare una nota.

Una maniera per ottimizzare la procedura riguardante l'arresto di emissione di una nota è: mandare un note on e mandare 2 data byte per la velocity, uno conterrà l'effettivo valore che indica quanto a lungo deve essere sostenuta la nota; l'altro avrà valore 0, ossia una nota che nasce e muore nell'istante in cui è stata generata.

[101] -> pressione di canale (aftertouch)

Messaggio per indicare il tremolo di una (in verità una media su quelle che attualmente lo stanno attraversando) nota in uscita da un canale.

Accompagnato da un data byte per il valore di variazione di pressione (da 0 a 127) che si vuole abbia la nota in uscita dal canale bersaglio.

[110] -> pitch bend change

Messaggio per indicare il vibrato di una nota in uscita da un canale.

Accompagnato da 2 data byte che insieme indicano il valore di variazione di frequenza che si vuole ottenere (128x128 valori).

[100] -> program change

Messaggi utilizzati per cambiare il timbro in uscita su un canale.

Accompagnato da un data byte che contiene un indice rappresentante il valore di timbro desiderato (da 0 a 127).

[011] -> controll change

Messaggi che si riferiscono a diversi eventi quali: controllo del fiato; pedale di sostegno; cambio del banco dei filtri...

Accompagnato da 2 data byte che trasportano un valore numerico dato dalla combinazione dei due messaggi (128x128 valori).

[010] -> chiave di pressione polifonica

Messaggi di sistema:

- MTC (MIDI Time Code): serve per settare un riferimento temporale assoluto
- Song Position Pointer: insieme di messaggi che indicizza una stessa canzone
- Song Select: seleziona una canzone presente nel sistema
- Tune Request: poco utilizzato, serve a verificare un'accordatura

Messaggi real-time:

- MIDI clock: serve per settare un riferimento temporale relativo
- Start/ continue/ stop: gestiscono le varie fasi di attivazione del sistema
- Active sensing: mantiene aperta la connessione fra 2 distinti sistemi MIDI
- System reset: reimposta la configurazione del sistema