



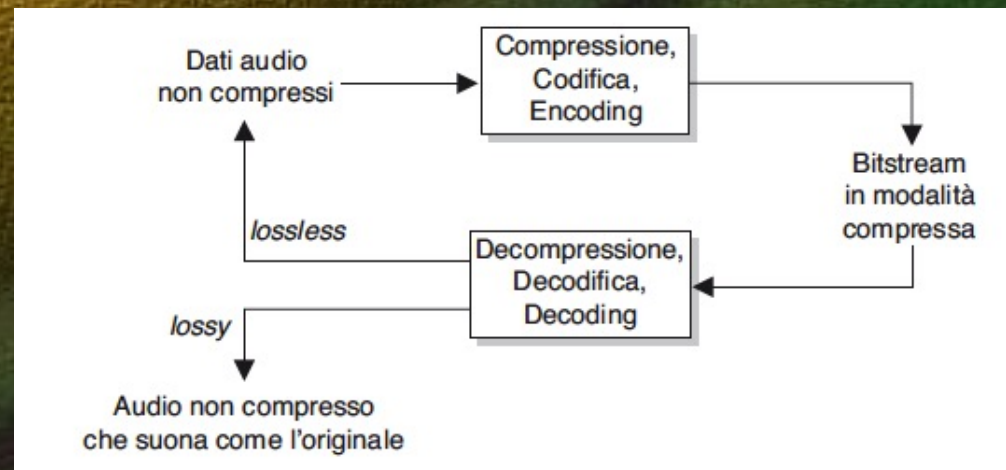
Tecnologie digitali per il suono e l'immagine 2020/21

Vincenzo Lombardo
Corso di Laurea in DAMS
Università di Torino

Mutuato in parte da Elaborazione audio e musica
(Laurea Magistrale di Informatica)

Compressione del suono

Come risparmiare spazio e tempo lavorando sulla forma d'onda



La compressione audio

L'arte di minimizzare le risorse per i dati audio

- ridurre la memoria occupata
- ridurre i costi di trasmissione
- Obiettivo: buona qualità in confronto all'audio non compresso

Riproduzione trasparente

- Audio che anche “orecchie” sensibili non riescono a distinguere dall'originale
- Numero minimo di bit per riproduzione trasparente?

Audio digitale

- Vantaggi
 - Alta fedeltà delle copie del segnale
 - Robustezza
 - Gamma dinamica estesa
- Svantaggio: alto tasso di trasferimento dati
 - Campionamento: 44.1 kHz (CD), 48 kHz (DAT)
 - Quantizzazione: PCM lineare a 16 bit
 - Molto spazio occupato
 - Ampia banda occupata sulla trasmissione

Esempio: 1 min stereo, qualità CD

- Parametri

- campionamento = 44,100 c/s
- quantizzazione = 16 bit = 2 byte
- canali = 2

- Memoria = $44.100 \text{ c/s} * 2 \text{ can} * 2 \text{ byte/c} * 60 \text{ s}$
~ 10 Mbyte

- Tempo = $10 \text{ MB} * 8 \text{ bit/byte} / (2 \text{ Mbps} * 60 \text{ s})$
~ 40 sec

Inoltre ...

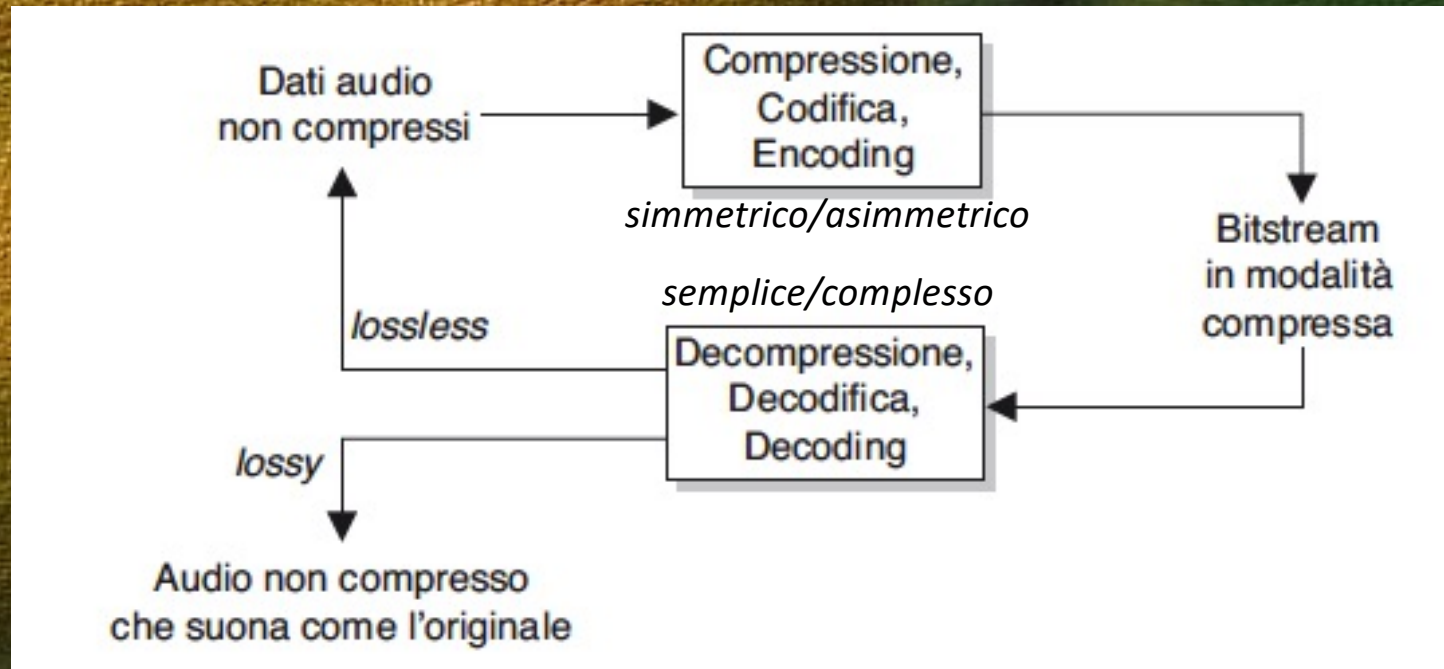
- Problemi di banda per applicazioni multimediali
- Accesso e trasferimento dati sempre più veloce, ma ...

... immagini, video, testo, e audio di alta qualità sincronizzati tutti assieme

Due livelli di applicazioni

- Bassa qualità
 - Telefonia
- Alta qualità
 - colonne sonore per giochi
 - memorie allo stato solido per la musica
 - audio su Internet
 - broadcast di audio digitale (Radio e TV)

Schema generico co/dec





Schemi di compressione semplici

Compressione del silenzio

- Utile per intensità vicine allo 0
- Silenzio = sequenze di campioni di intensità nulla

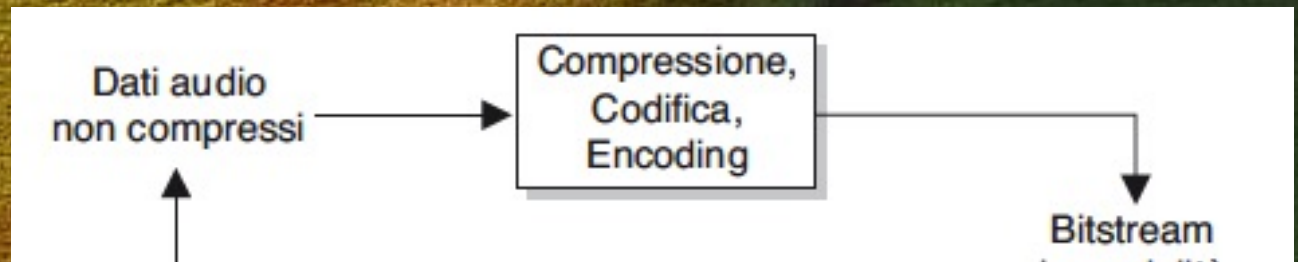
RLE (Run Length Encoding)

- Compressione lossless
- Codifica un run

Esempio:

- “ddddddhhhhhhhyyyyyyy” → “7d8h9y”

RLE encoder

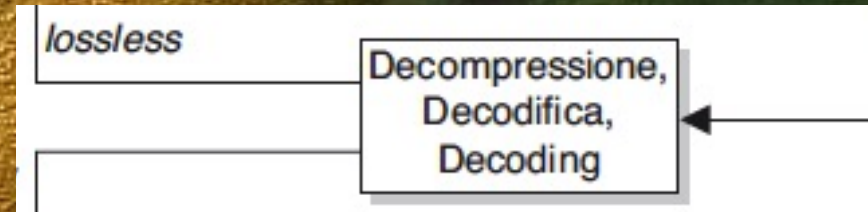


- Init stream;
- Init valore attuale x a un valore impossibile X;
- Init contatore a 0
- While not end-of-stream do
 - Leggi prossimo valore y
 - if x = y then
 - Incrementa contatore di 1
 - Else
 - Scrivi <contatore, x>;
 - init contatore a 1;
 - Valore attuale x diventa y

“ddddddhhhhhhhhhyyyyyyy” → “7d8h9y”

RLE decoder

- `inizializza lo stream;`
- `Init contatore a 0`
- `While not end-of-stream do`
 - `Leggi prossima coppia <c,x>`
 - `For c volte do`
 - `Scrivi x`

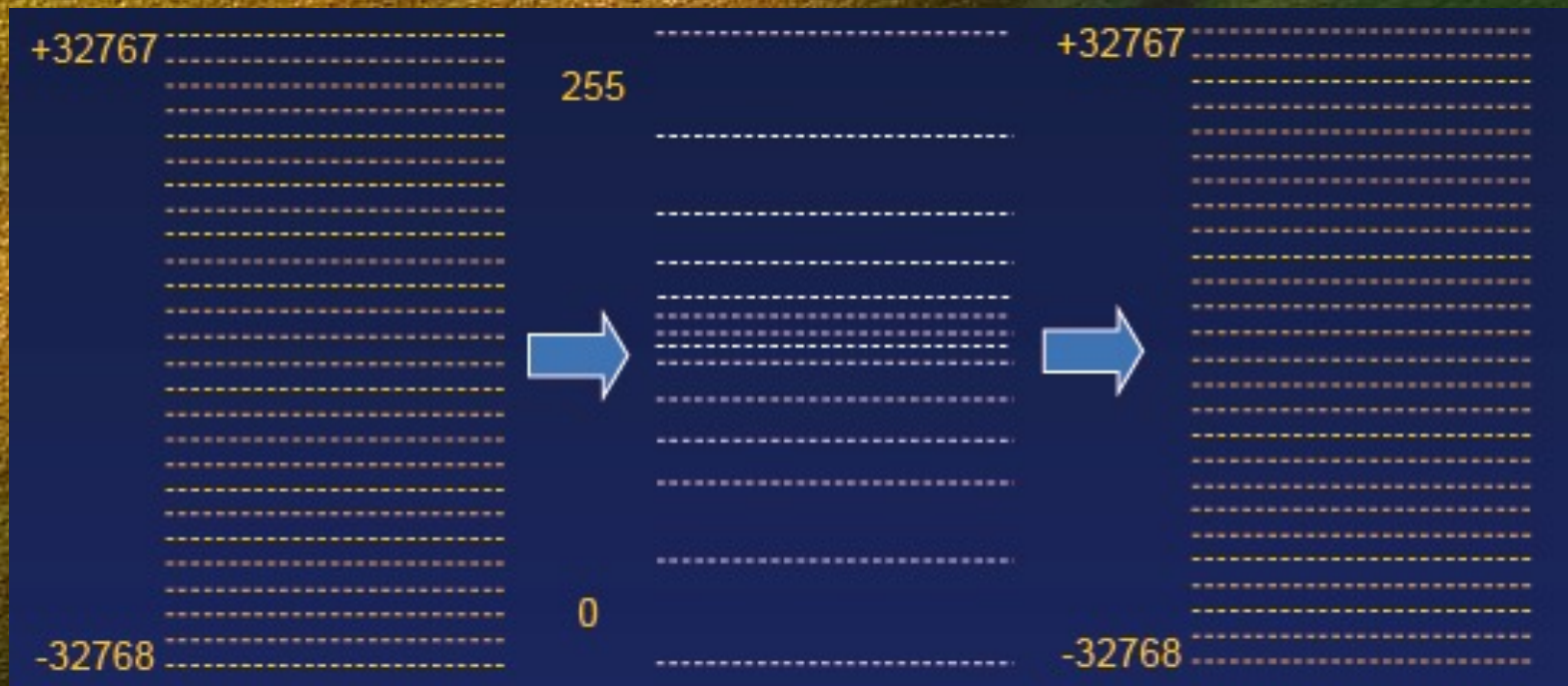


`"ddddddhhhhhhhyyyyyyy" ← "7d8h9y"`

Compressione del silenzio (lossy): RLE + soglie

- Due casi
 - soglia di intensità sonora sotto la quale il segnale è interpretato come silenzio
 - minimo numero di campioni abbastanza forti che fungano da chiusura di una sequenza di silenzio
- Esempio: 15 deboli + 2 forti + 13 deboli

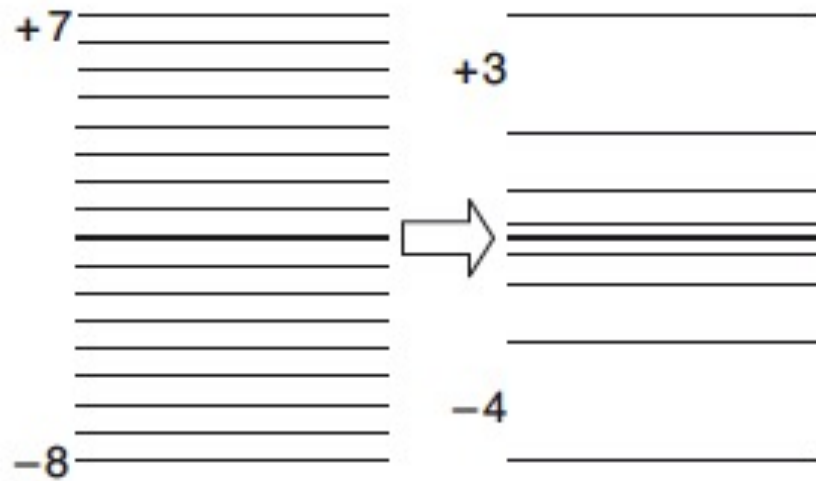
ITU-T G.711, μ -law e A-law



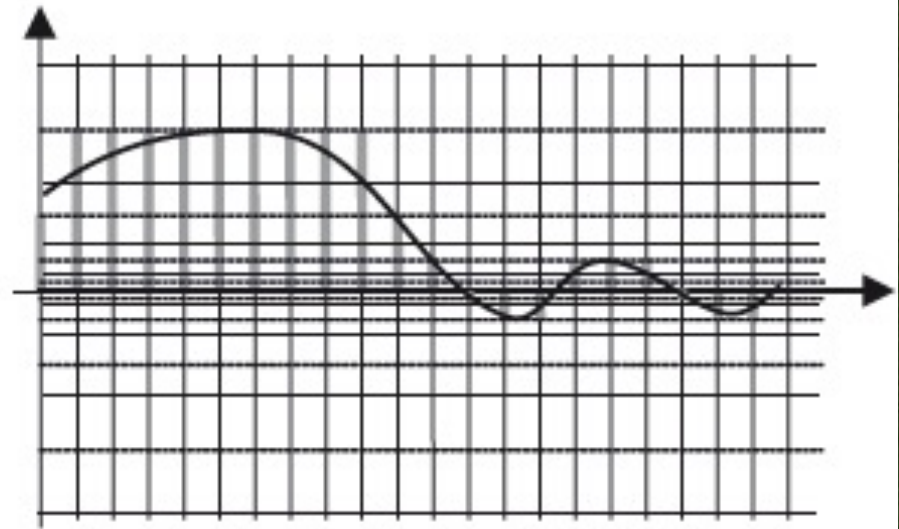
μ -law

- Quantizzazione logaritmica
- Gamma dinamica di 14 bit con una codifica a 8-bit
- Per i servizi voce ISDN in Nord America e Giappone
- Semplice il computo della codifica

Principio μ -law

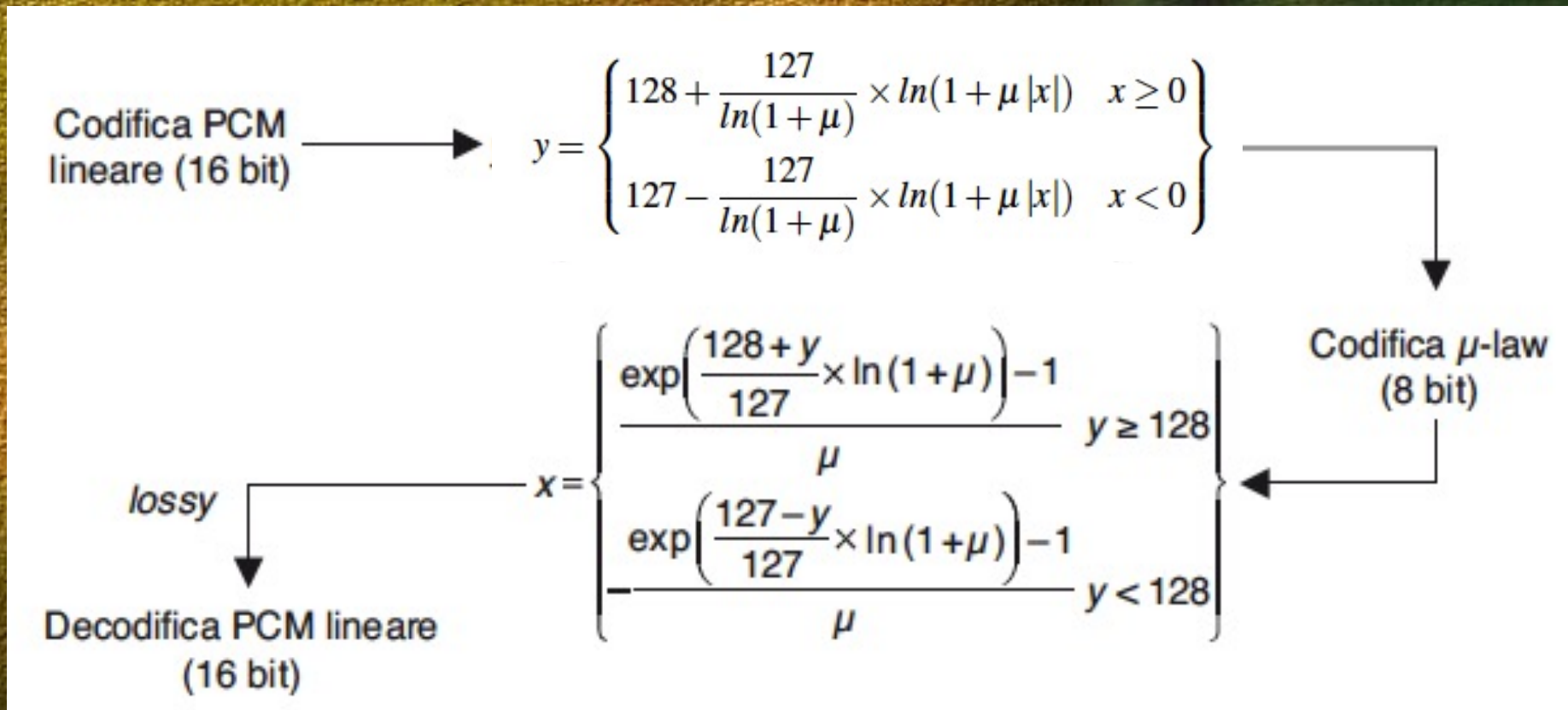


(a)



(b)

Schema di codifica μ -law

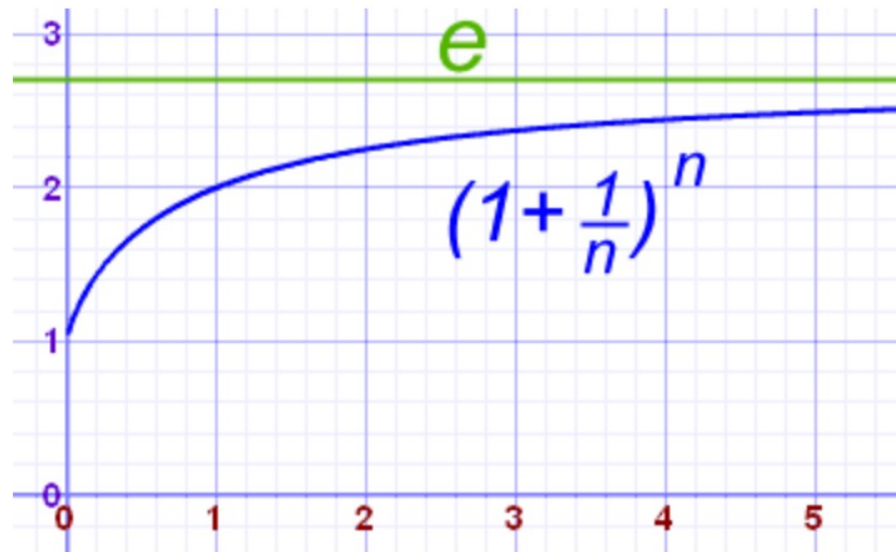


\ln = logaritmo naturale, in base $e = 2,7182818284590452353602874713527\dots$

\exp = esponenziale naturale, $e^n = (2,7182818284590452353602874713527\dots)^n$

Numero e

ln = logaritmo naturale, in base e = 2,7182818284590452353602874713527...



n	$(1 + 1/n)^n$
1	2,00000
2	2,25000
5	2,48832
10	2,59374
100	2,70481
1.000	2,71692
10.000	2,71815
100.000	2,71827

Dettagli caso positivo

$$y = \begin{cases} 128 + \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x \geq 0 \\ 127 - \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x < 0 \end{cases}$$

- Range variabili: x da 0 a 32767 (normalizzato da 0 a 1), y da 128 a 255
 - x=0 (silenzio SIGNED) → y=128 (silenzio UNSIGNED)
- Andamento logaritmico dei valori
 - A 32768 (=128x128) valori x, corrispondono 128 y, se fosse quantizzazione uniforme 128 x per 1 y
 - vicino al silenzio (x vicino a 0): pochi valori x (< 128) corrispondono a 1 y
 - lontano dal silenzio (x vicino a 32767): molti valori x (> 128) corrispondono a 1 y
 - Logaritmo \ln : logaritmo base e = 2,718281... = $\sum(1+1/n)^n$
 - $0 \leq \ln(1+\mu|x|) / \ln(1+\mu) \leq 1$ dà una quota di 127, che si comporta in modo logaritmico

Operazioni di conversione

	Campione originale	x in (-1,+1)	Nuovo campione
	-32768	-1	0

$y = \begin{cases} 128 + \frac{127}{\ln(1+\mu)} \times \ln(1+\mu x) & x \geq 0 \\ 127 - \frac{127}{\ln(1+\mu)} \times \ln(1+\mu x) & x < 0 \end{cases}$	-32100	-0,9796142	0
	-32000	-0,9765625	1

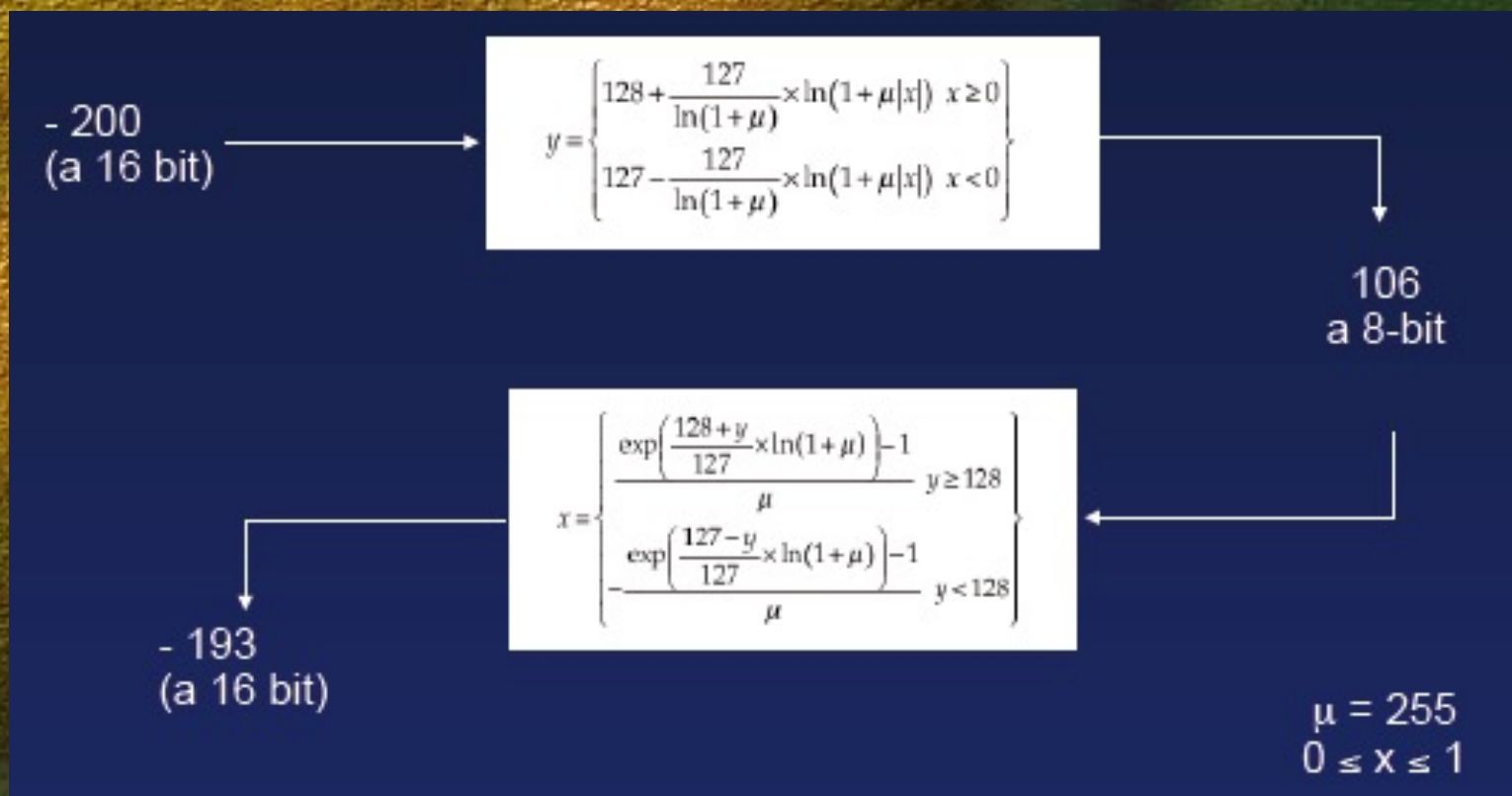
	-200	-0,00610351	106
	-100	-0,0030517	113

	0	0	128
$x = \begin{cases} \frac{\exp\left(\frac{128+y}{127} \times \ln(1+\mu)\right) - 1}{\mu} & y \geq 128 \\ \frac{\exp\left(\frac{127-y}{127} \times \ln(1+\mu)\right) - 1}{\mu} & y < 128 \end{cases}$
	100	0,0030517	141
	200	0,00610351	149

	32000	0,9765625	254
	32100	0,9796142	255

	32767	1	255

Lossy μ -law



The background of the slide features a diagonal gradient. The upper-left portion is a bright, shimmering gold, which transitions through a dark, almost black diagonal band to a deep, forest green on the lower-right. The overall texture is slightly grainy, suggesting a digital or printed surface.

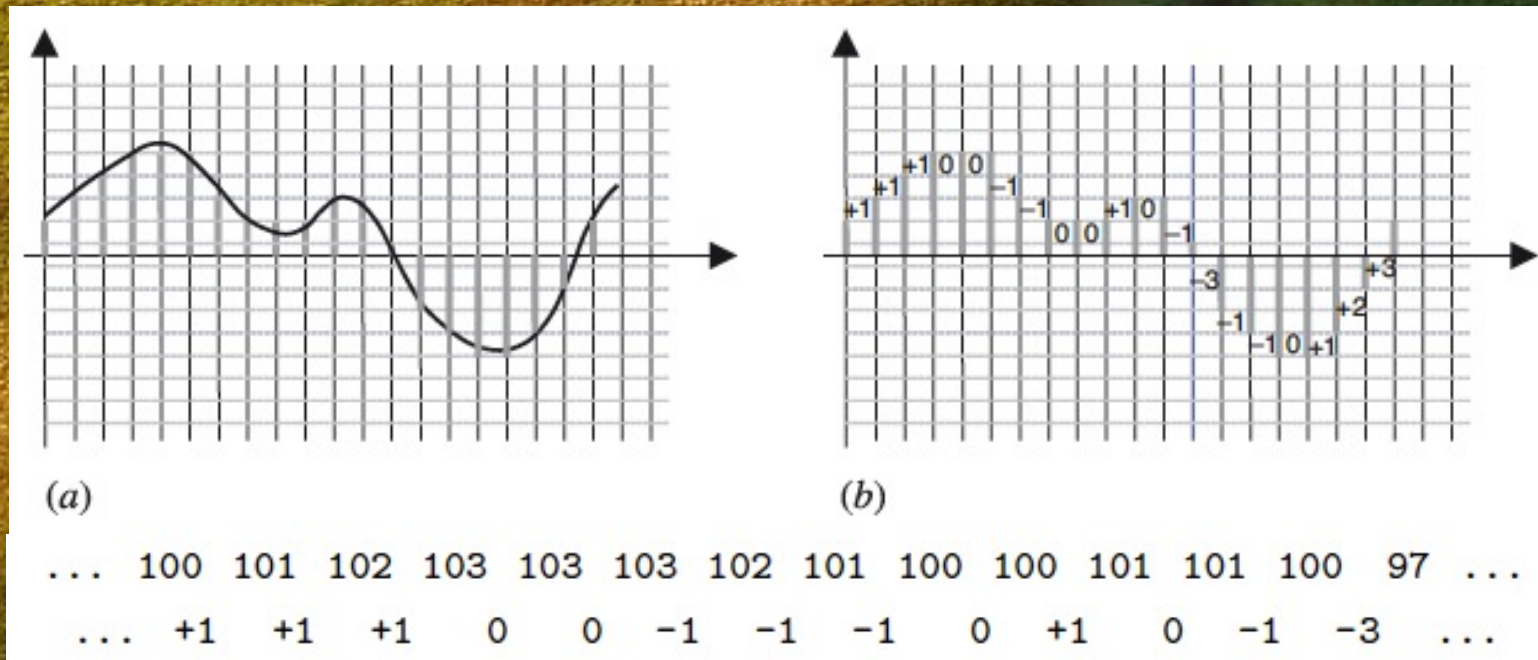
Compression ADPCM

ADPCM = Adaptive Differential PCM

CCITT G.721, G.723, e ITU-T G.726

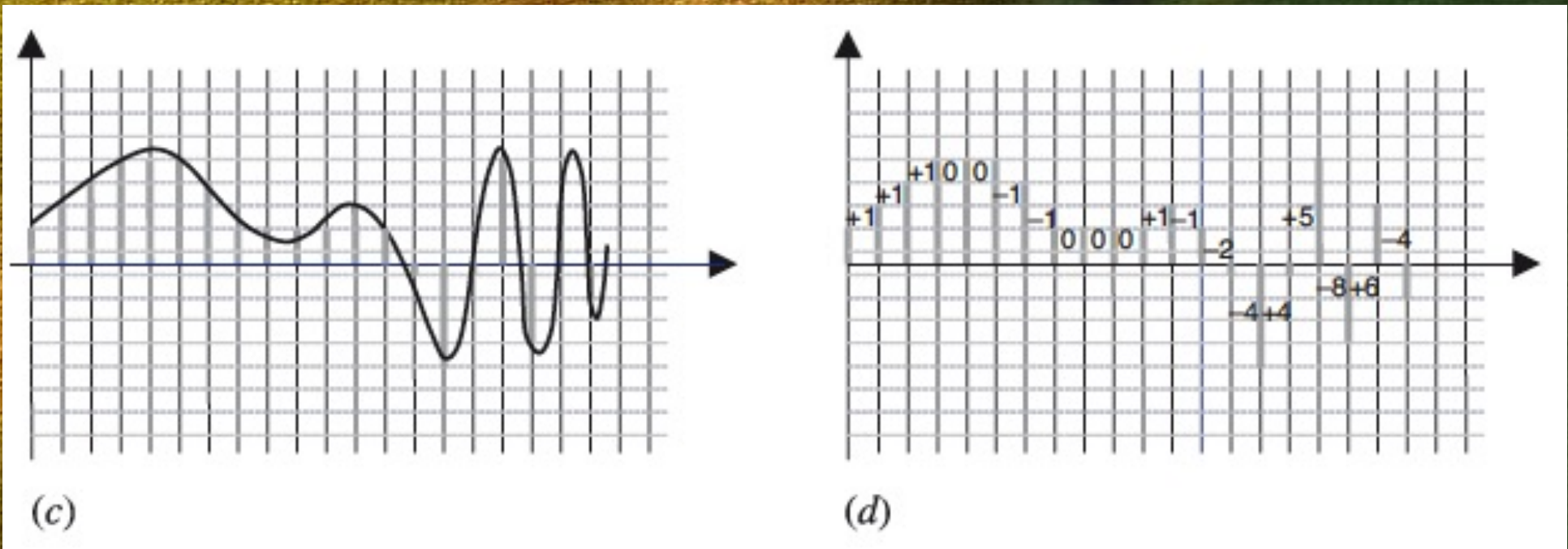
- Standard per la compressione di dati vocali
 - CCITT G.721 (ADPCM a 32 Kbps)
 - CCITT G.723 (ADPCM a 24 e 40 Kbps)
- metodo comune di compressione
- buon compromesso tra
 - velocità di elaborazione
 - tasso di compressione
 - decodifica di qualità

Codifica della differenza (DPCM)



- Ridondanze temporali tra i campioni
- Differenza tra due campioni a x-bit con meno di x bit
- Si memorizza la differenza (non il campione)

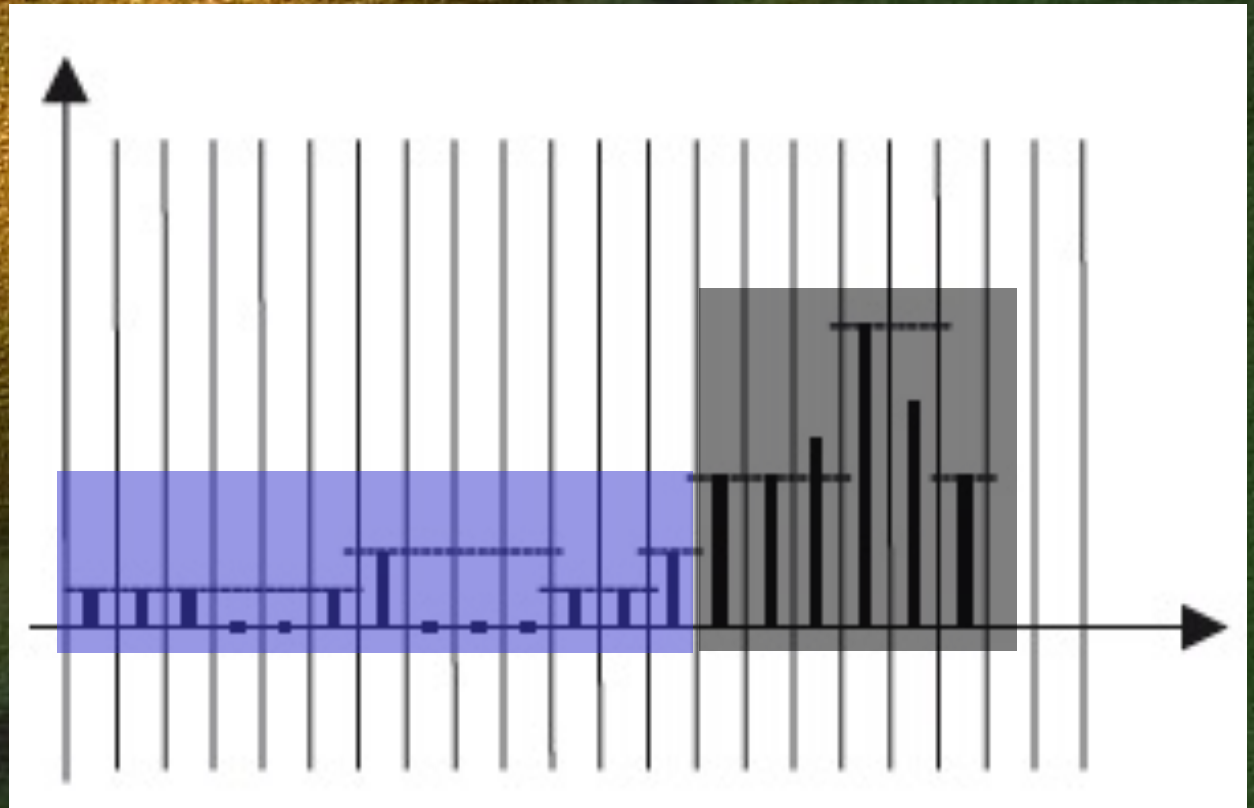
Problema Slope Overload



- Differenze elevate (alte frequenze) non si possono rappresentare con pochi bit
- Errori introdotti porterebbero distorsioni sulle alte frequenze

Adaptive PCM (ADPCM)

- Differenze grandi
- Passi di quantizzazione grandi
- Differenze piccole
- Passi di quantizzazione piccoli

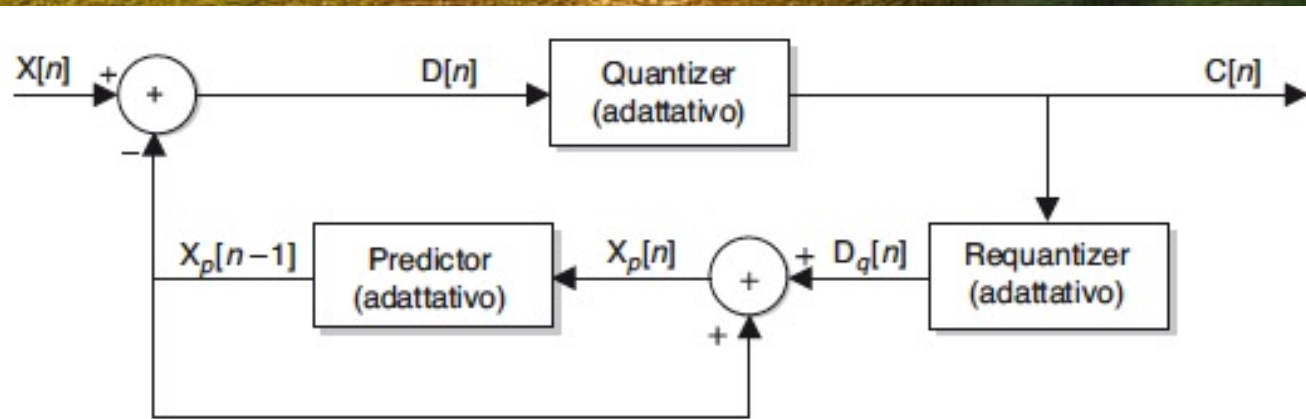


La predizione

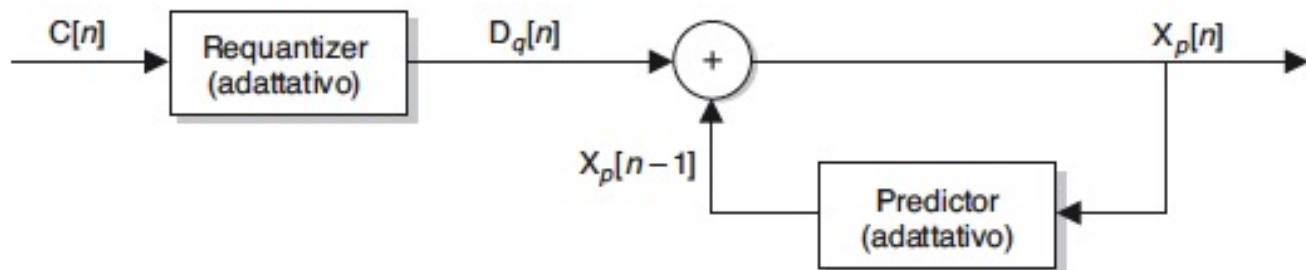
- campione precedente usato come base di partenza
- si sfrutta la conoscenza della porzione precedente del segnale
- Esempio: +1 e -1

```
... 100 101 102 103 103 103 102 101 100 100 101 101 100 97 ...  
... 100 101 102 103 104 103 102 101 100 101 100 101 100 99 ...  
... 0 0 0 0 -1 0 0 0 0 -1 +1 0 0 -2 ...
```

Schema ADPCM



(a)



(b)

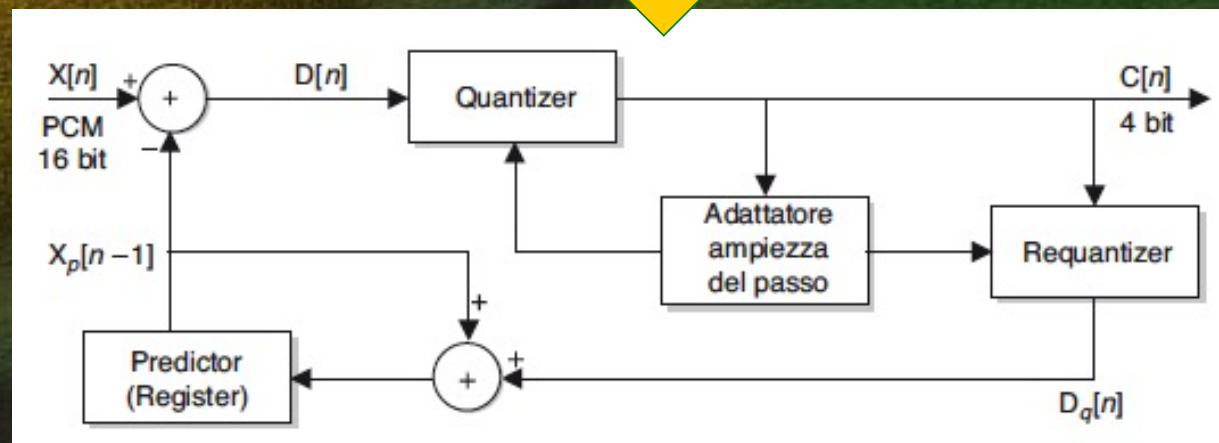
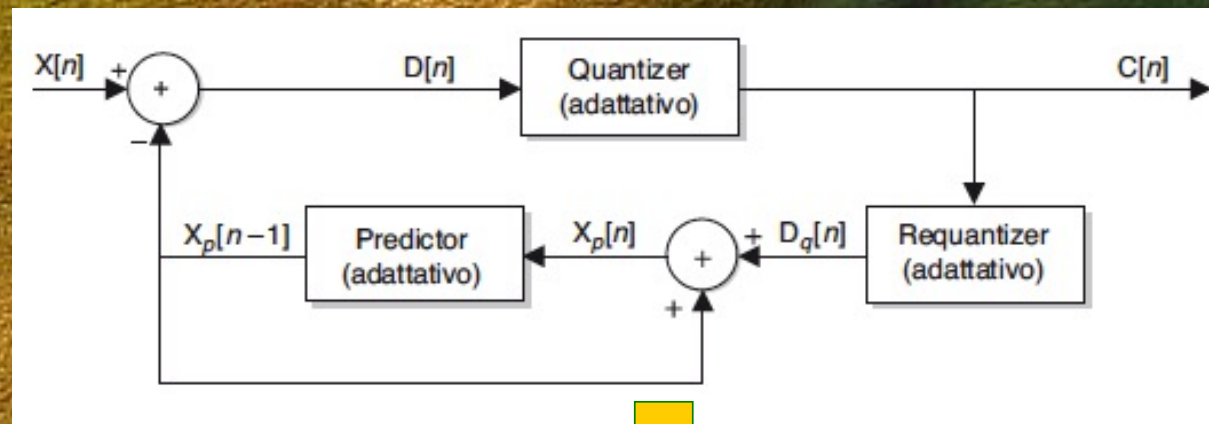
Funzionalità aggiuntive

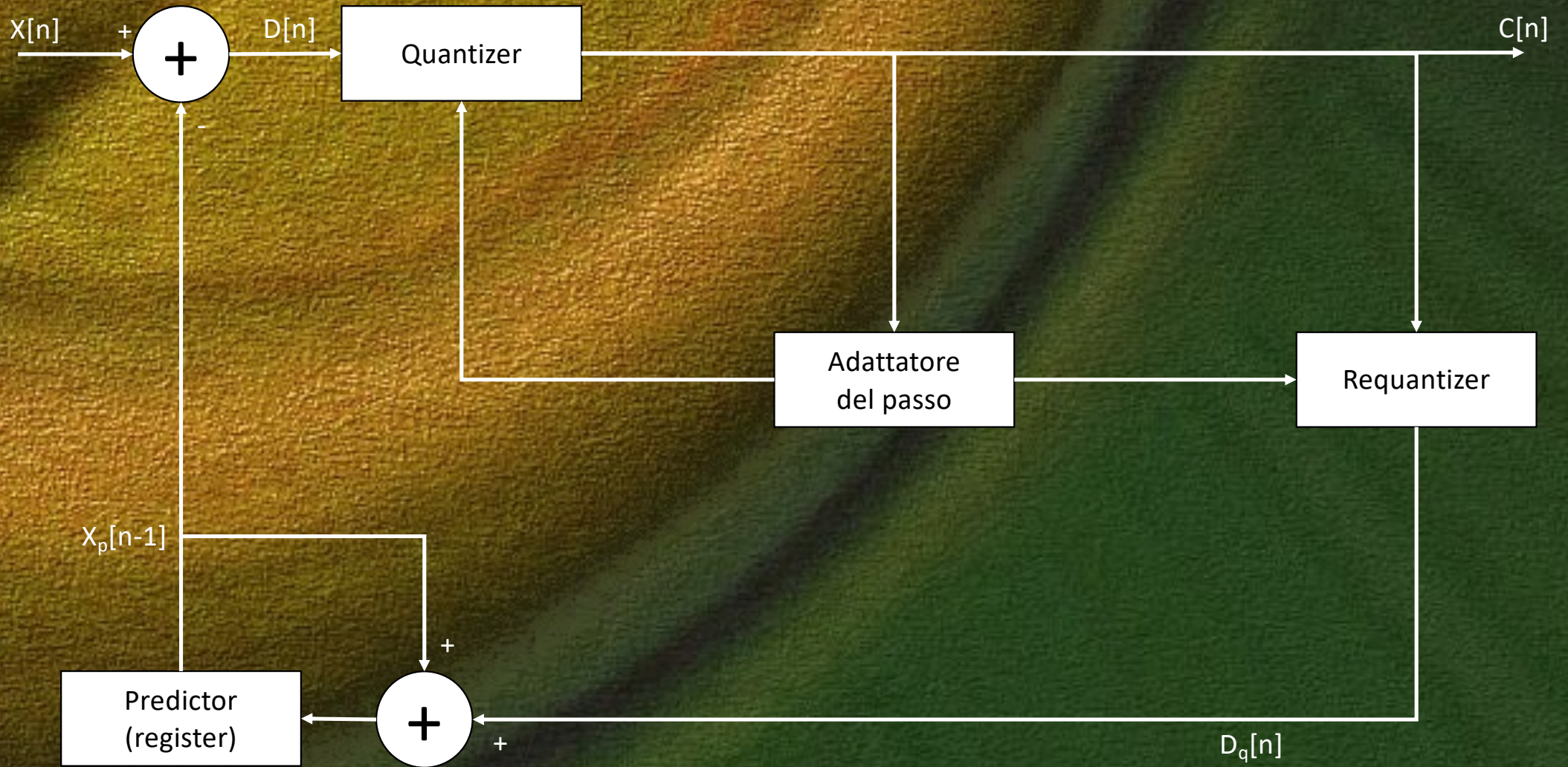
- Formattazione del bitstream $C[n]$
 - Utile nel recupero degli errori di decodifica
 - Problema propagazione dell'errore
 - Ridistribuzione pseudocasuale della sequenza
- Informazioni aggiuntive
 - indicare i livelli di quant. di Predictor e/o Quantizer
 - fornire informazioni contestuali al decoder per recupero errori o accesso diretto al bitstream

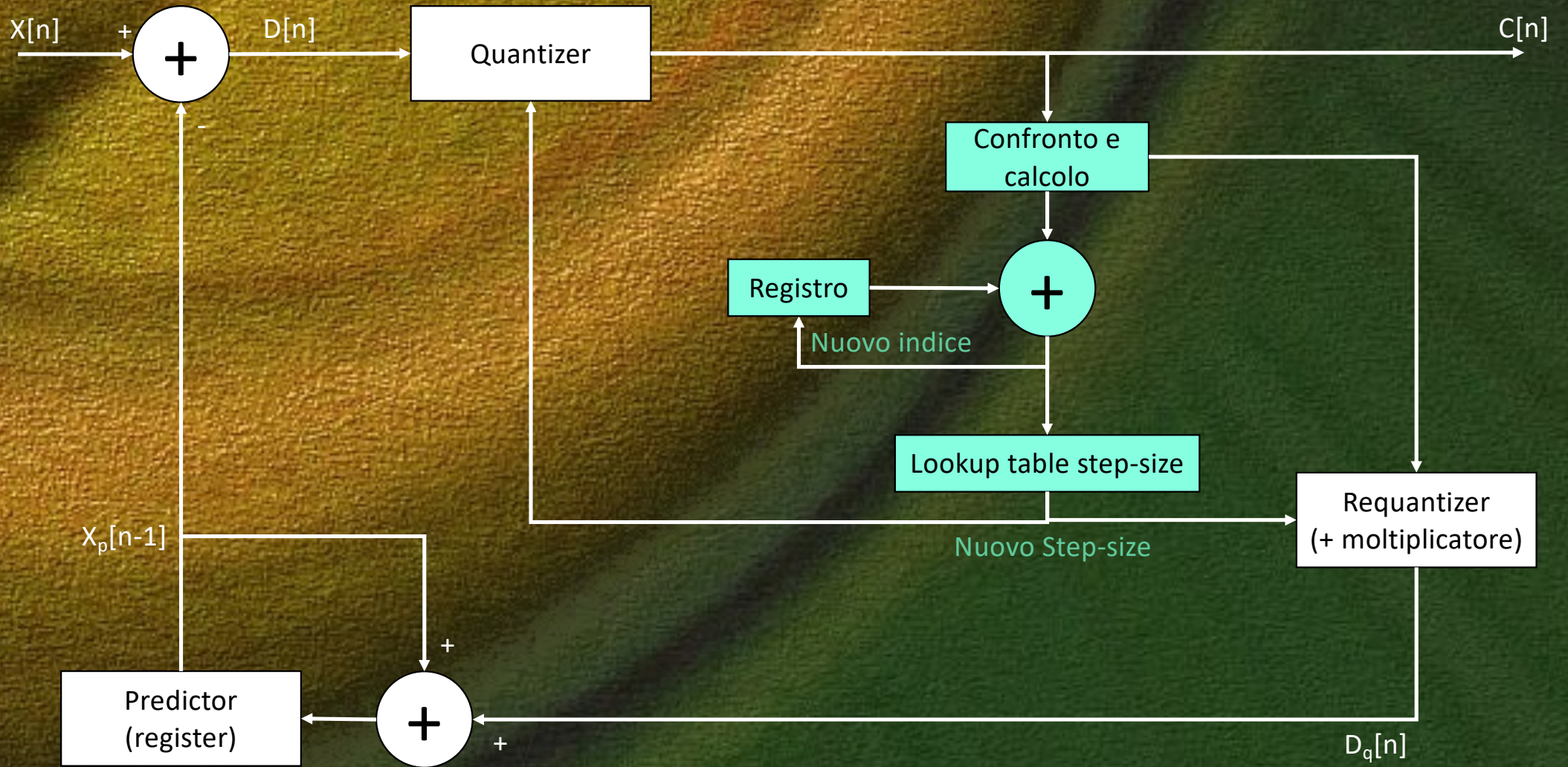
IMA ADPCM

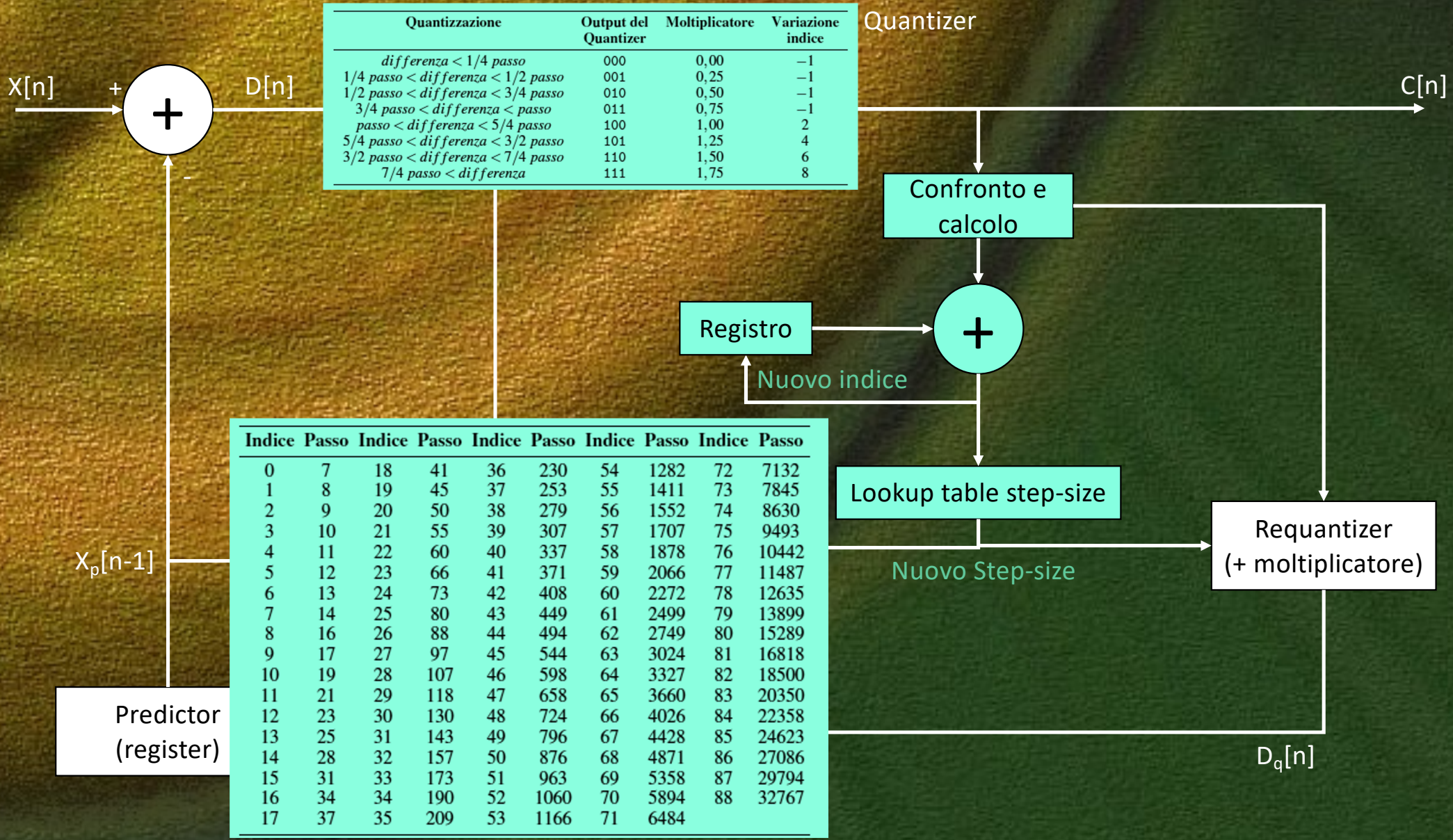
- Predictor non adattativo
- Passo di quantizzazione aumenta in modo logaritmico
- Variante IMA/DVI ADPCM: comprime da 16-bit a 4 bit
- Variante Microsoft ADPCM: da usare con i file WAV (da 16 a 4 bit)

Schema IMA





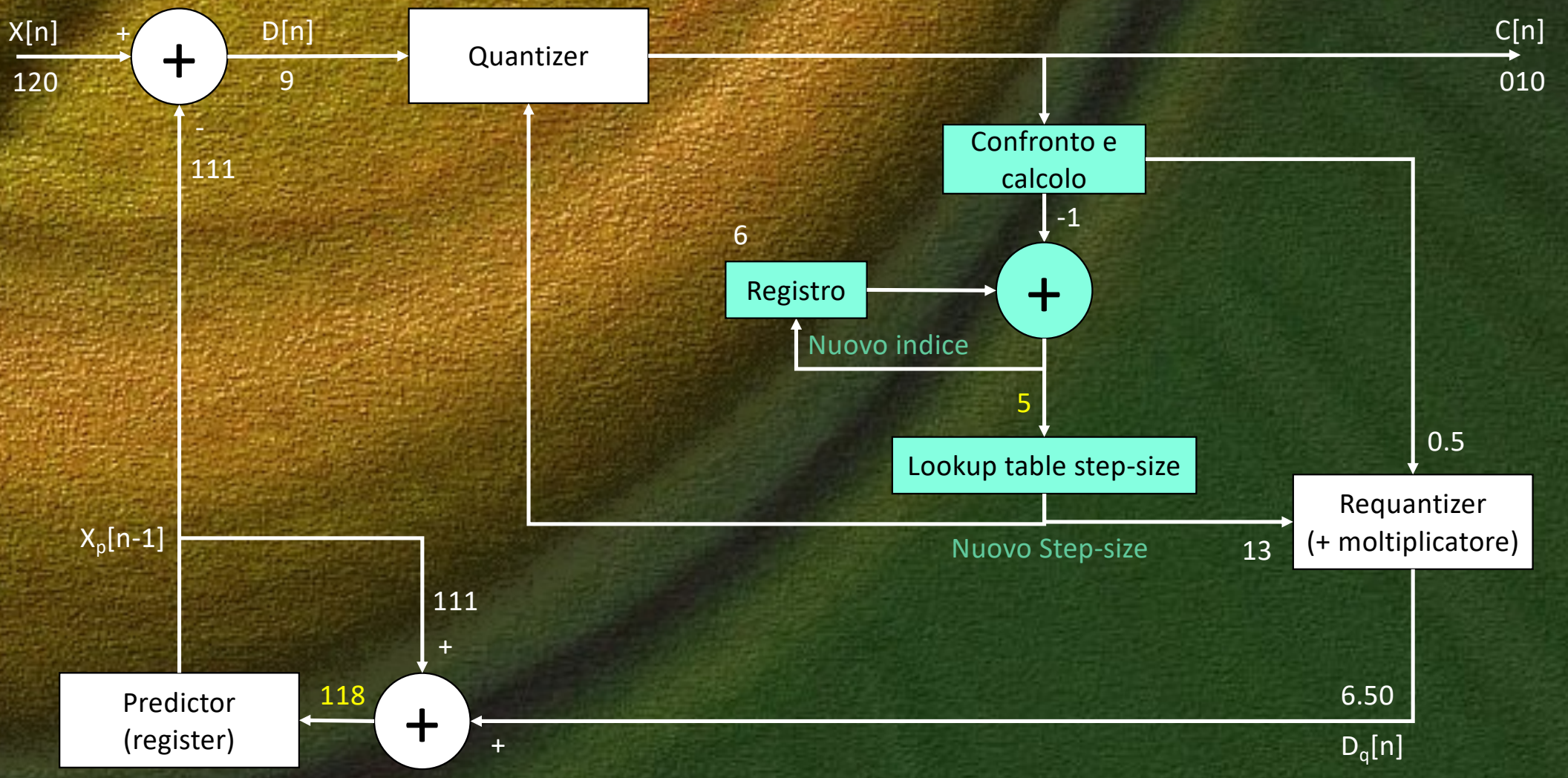




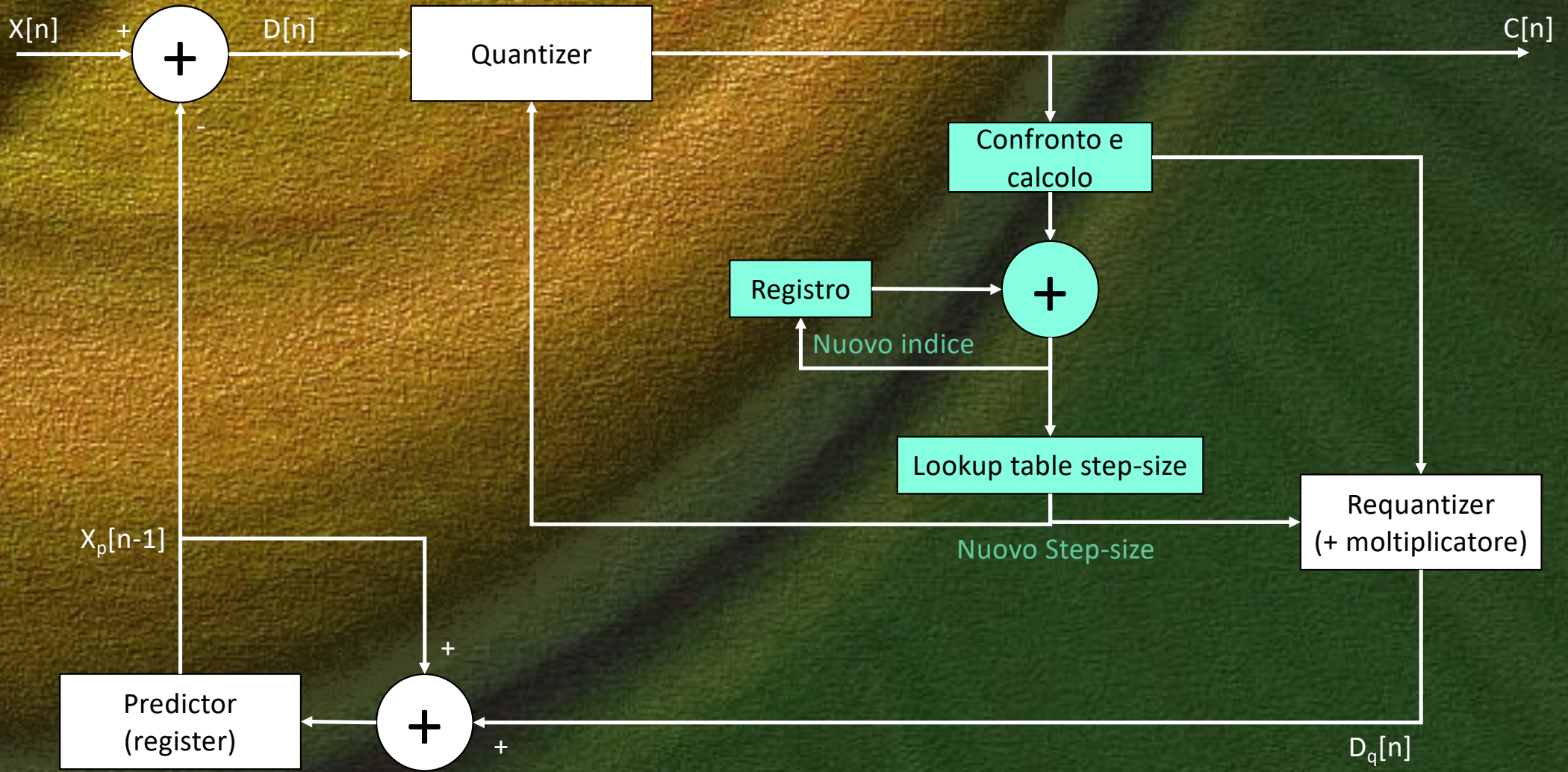
Quantizzazione	Output del Quantizer	Moltiplicatore	Variazione indice
$diff erenza < 1/4$ passo	000	0,00	-1
$1/4$ passo < $diff erenza < 1/2$ passo	001	0,25	-1
$1/2$ passo < $diff erenza < 3/4$ passo	010	0,50	-1
$3/4$ passo < $diff erenza <$ passo	011	0,75	-1
passo < $diff erenza < 5/4$ passo	100	1,00	2
$5/4$ passo < $diff erenza < 3/2$ passo	101	1,25	4
$3/2$ passo < $diff erenza < 7/4$ passo	110	1,50	6
$7/4$ passo < $diff erenza$	111	1,75	8

Indice	Passo	Indice	Passo	Indice	Passo	Indice	Passo	Indice	Passo
0	7	18	41	36	230	54	1282	72	7132
1	8	19	45	37	253	55	1411	73	7845
2	9	20	50	38	279	56	1552	74	8630
3	10	21	55	39	307	57	1707	75	9493
4	11	22	60	40	337	58	1878	76	10442
5	12	23	66	41	371	59	2066	77	11487
6	13	24	73	42	408	60	2272	78	12635
7	14	25	80	43	449	61	2499	79	13899
8	16	26	88	44	494	62	2749	80	15289
9	17	27	97	45	544	63	3024	81	16818
10	19	28	107	46	598	64	3327	82	18500
11	21	29	118	47	658	65	3660	83	20350
12	23	30	130	48	724	66	4026	84	22358
13	25	31	143	49	796	67	4428	85	24623
14	28	32	157	50	876	68	4871	86	27086
15	31	33	173	51	963	69	5358	87	29794
16	34	34	190	52	1060	70	5894	88	32767
17	37	35	209	53	1166	71	6484		

2	112	12	0/7	$3/2 \times 7 < 12 < 7/4 \times 7$	110	1,5	6	6	10,50	111
3	120	9	6/13	$1/2 \times 13 < 9 < 3/4 \times 13$	010	0,5	-1	5	6,50	118
4	127	9	5/12	$3/4 \times 12 < 9 < 12$	011	0,75	-1	4	9,00	127



3	120	9	6/13	$1/2 \times 13 < 9 < 3/4 \times 13$	010	0,5	-1	5	6,50	118
4	127	9	5/12	$3/4 \times 12 < 9 < 12$	011	0,75	-1	4	9,00	127
5	131	4	4/11	$1/4 \times 11 < 4 < 1/2 \times 11$	001	0,25	-1	3	2,75	130



13	99	-14	4/11	$5/4 \times 11 < 12 < 3/2 \times 11$	101	1,25	4	8	-13,75	99
14	54	-45	8/16	$7/4 \times 16 < 45$	111	1,75	8	16	-28,00	71
15	20	-51	16/34	$3/2 \times 34 < 51 < 7/4 \times 34$	110	1,5	6	22	-51,00	20

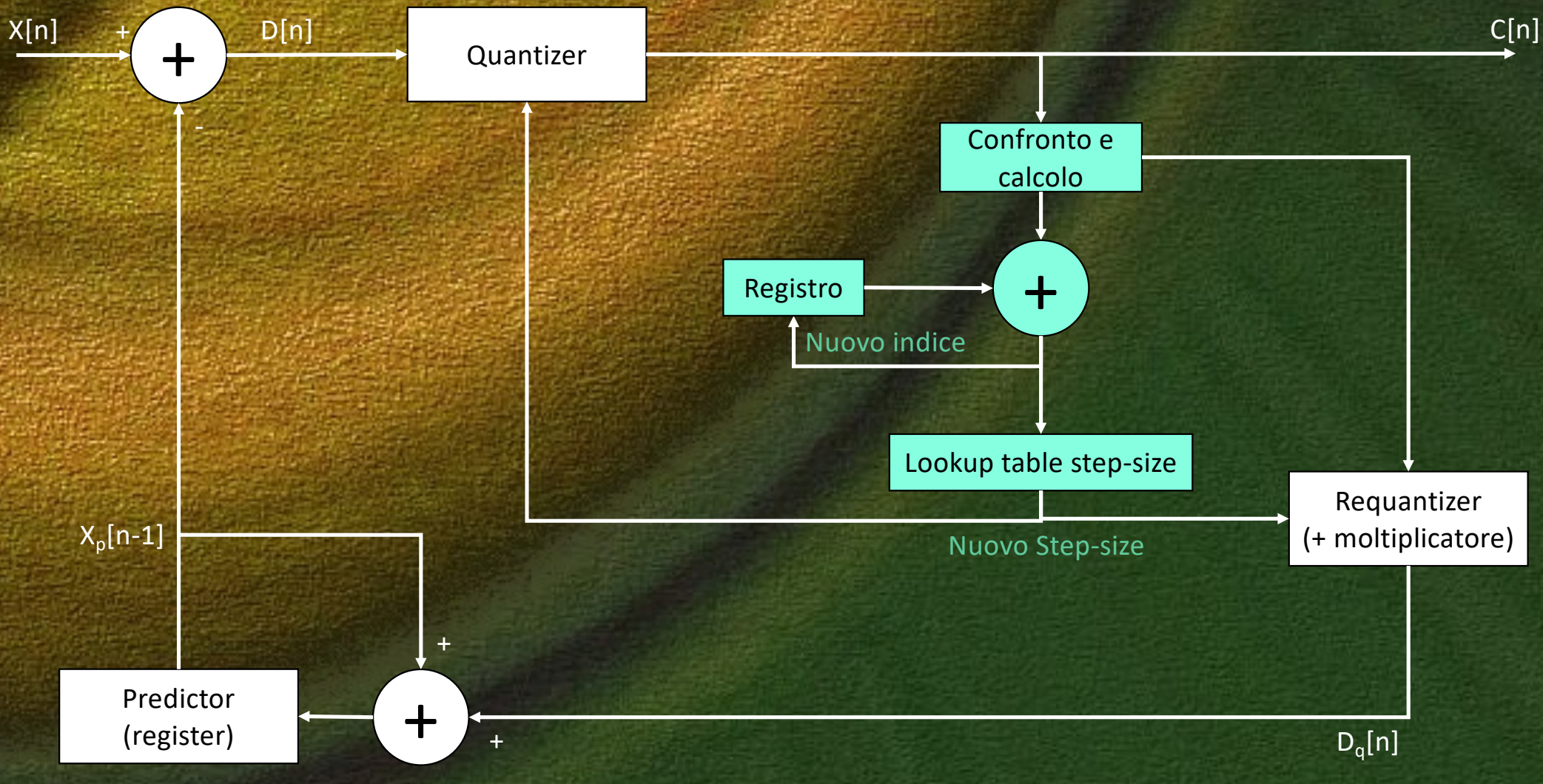


Tabella dei passi di quantizzazione

Indice	Passo	Indice	Passo	Indice	Passo	Indice	Passo	Indice	Passo
0	7	18	41	36	230	54	1282	72	7132
1	8	19	45	37	253	55	1411	73	7845
2	9	20	50	38	279	56	1552	74	8630
3	10	21	55	39	307	57	1707	75	9493
4	11	22	60	40	337	58	1878	76	10442
5	12	23	66	41	371	59	2066	77	11487
6	13	24	73	42	408	60	2272	78	12635
7	14	25	80	43	449	61	2499	79	13899
8	16	26	88	44	494	62	2749	80	15289
9	17	27	97	45	544	63	3024	81	16818
10	19	28	107	46	598	64	3327	82	18500
11	21	29	118	47	658	65	3660	83	20350
12	23	30	130	48	724	66	4026	84	22358
13	25	31	143	49	796	67	4428	85	24623
14	28	32	157	50	876	68	4871	86	27086
15	31	33	173	51	963	69	5358	87	29794
16	34	34	190	52	1060	70	5894	88	32767
17	37	35	209	53	1166	71	6484		

Adattatore dell'ampiezza del passo

Quantizzazione	Output del Quantizer	Moltiplicatore	Variazione indice
<i>differenza < 1/4 passo</i>	000	0,00	-1
<i>1/4 passo < differenza < 1/2 passo</i>	001	0,25	-1
<i>1/2 passo < differenza < 3/4 passo</i>	010	0,50	-1
<i>3/4 passo < differenza < passo</i>	011	0,75	-1
<i>passo < differenza < 5/4 passo</i>	100	1,00	2
<i>5/4 passo < differenza < 3/2 passo</i>	101	1,25	4
<i>3/2 passo < differenza < 7/4 passo</i>	110	1,50	6
<i>7/4 passo < differenza</i>	111	1,75	8

IMA ADPCM: riassunto

- Fattore di compressione di bit-per-campione:
 - 4 a 1 (da 16 a 4 bit)
 - bit-rate < 400 kbps
- Passi di quantizzazione superiori e bit-rate ridotti:
 - CCITT G.721: 16 o 32 kbps
 - CCITT G.723: 24 kbps

La compressione di tipo percettivo

MP3 & Co.

Principi psicoacustici applicati

Perché gli schemi percettivi

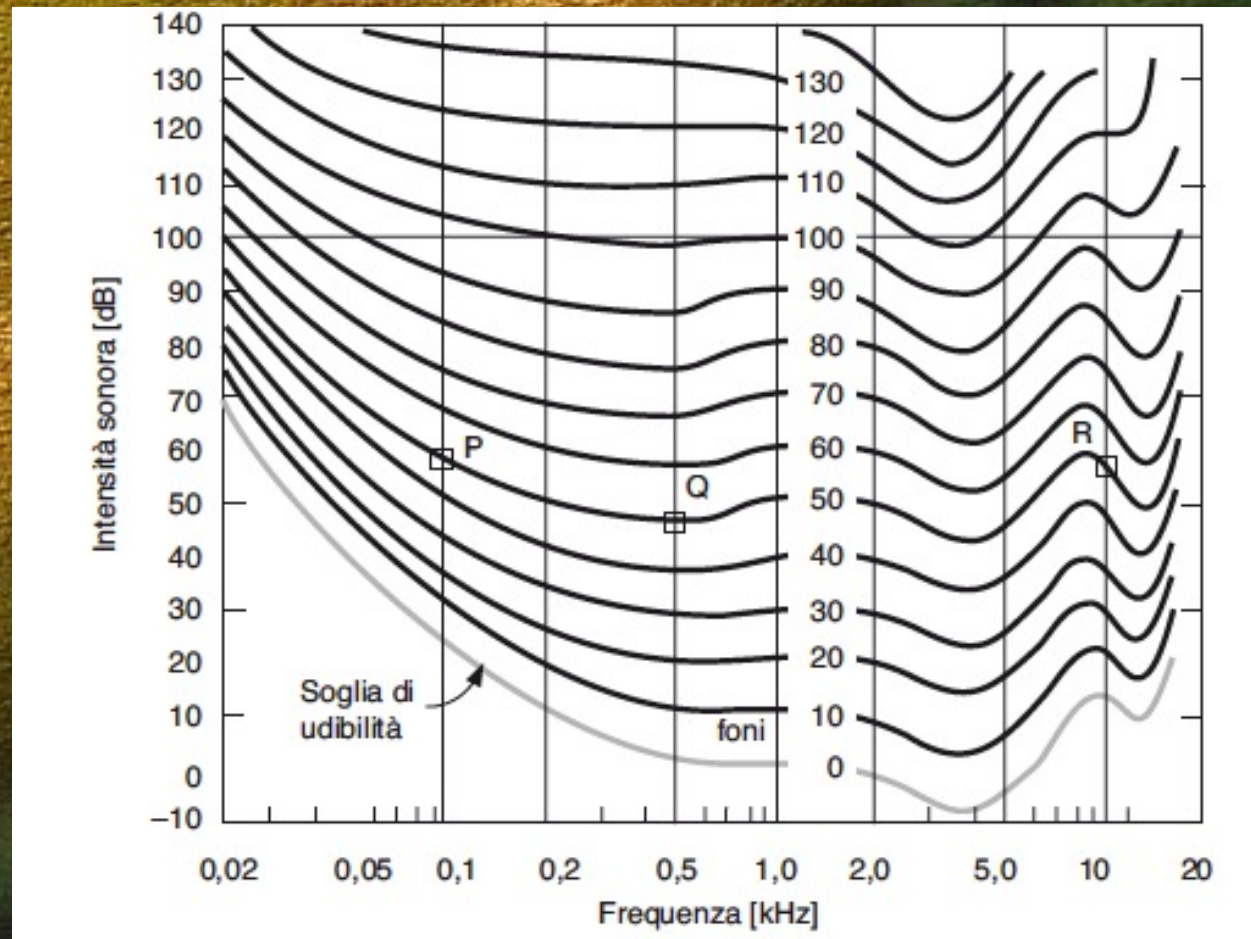
- Schemi semplici non producono rilevanti risparmi di spazio e tempo (peggioramento qualità)
- Eliminare parti del segnale che il nostro apparato uditivo non percepirebbe
- Codifiche di tipo lossy (con perdite), bit-rate molto bassi (0,5 bit/campione sul parlato)

- Dibattito sulla qualità

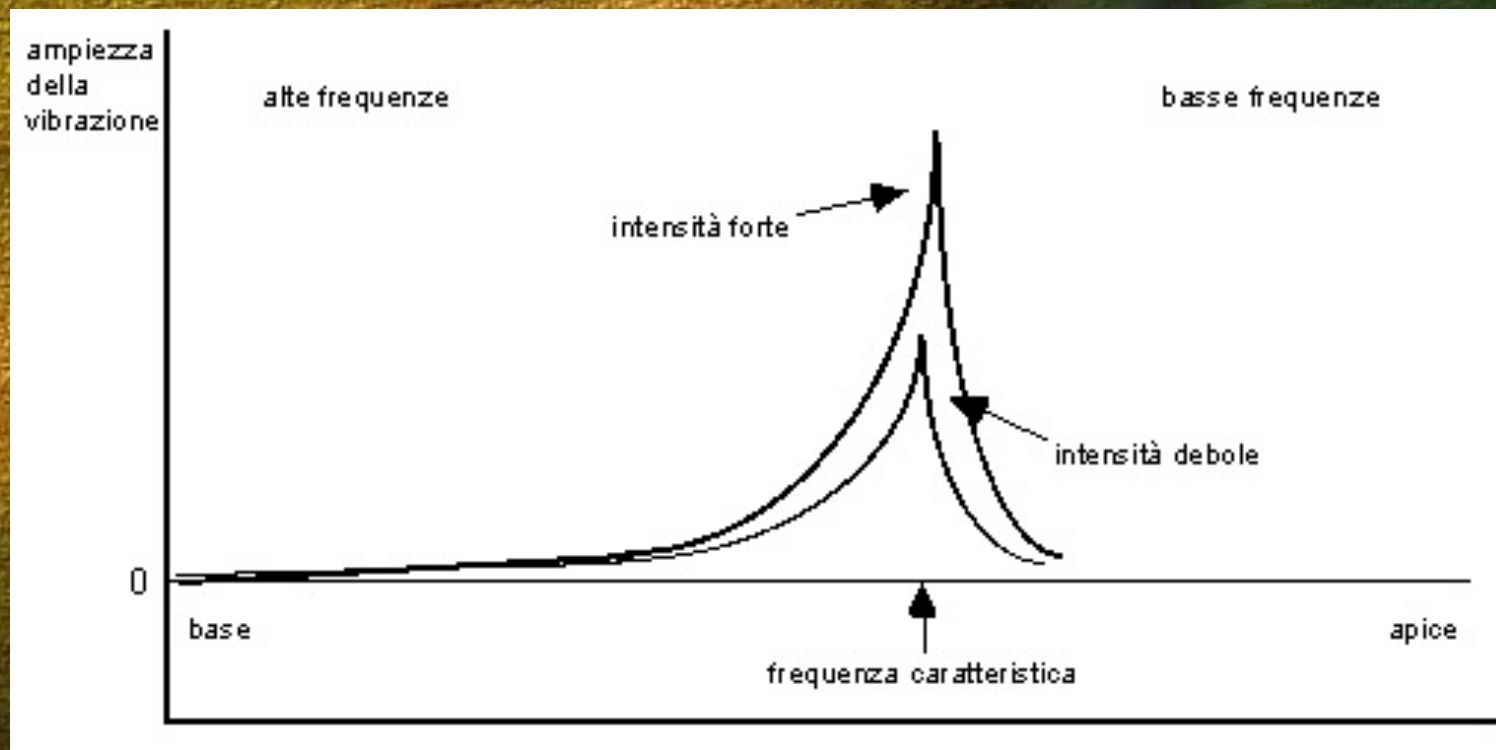


Principi psicoacustici

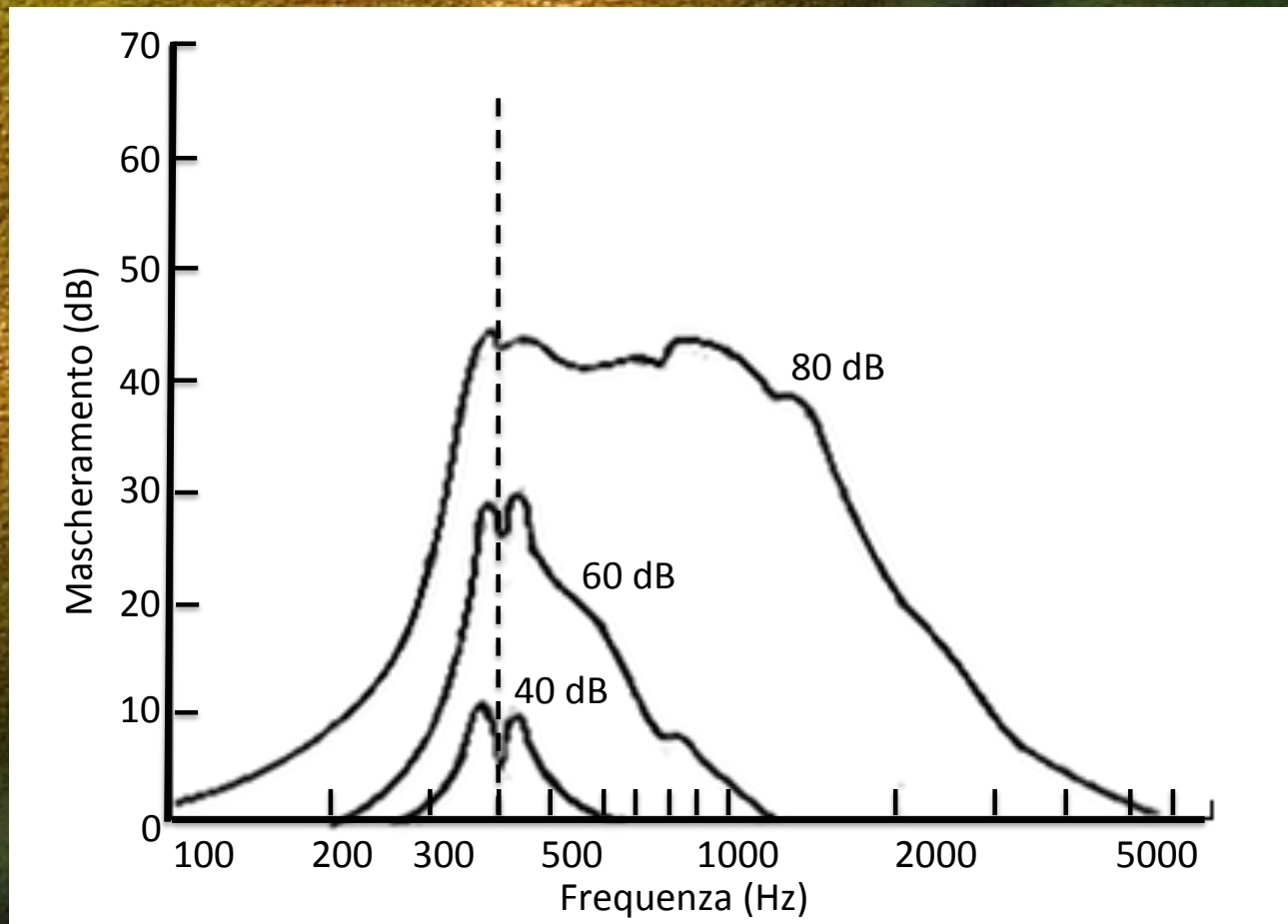
Soglia assoluta di udibilità



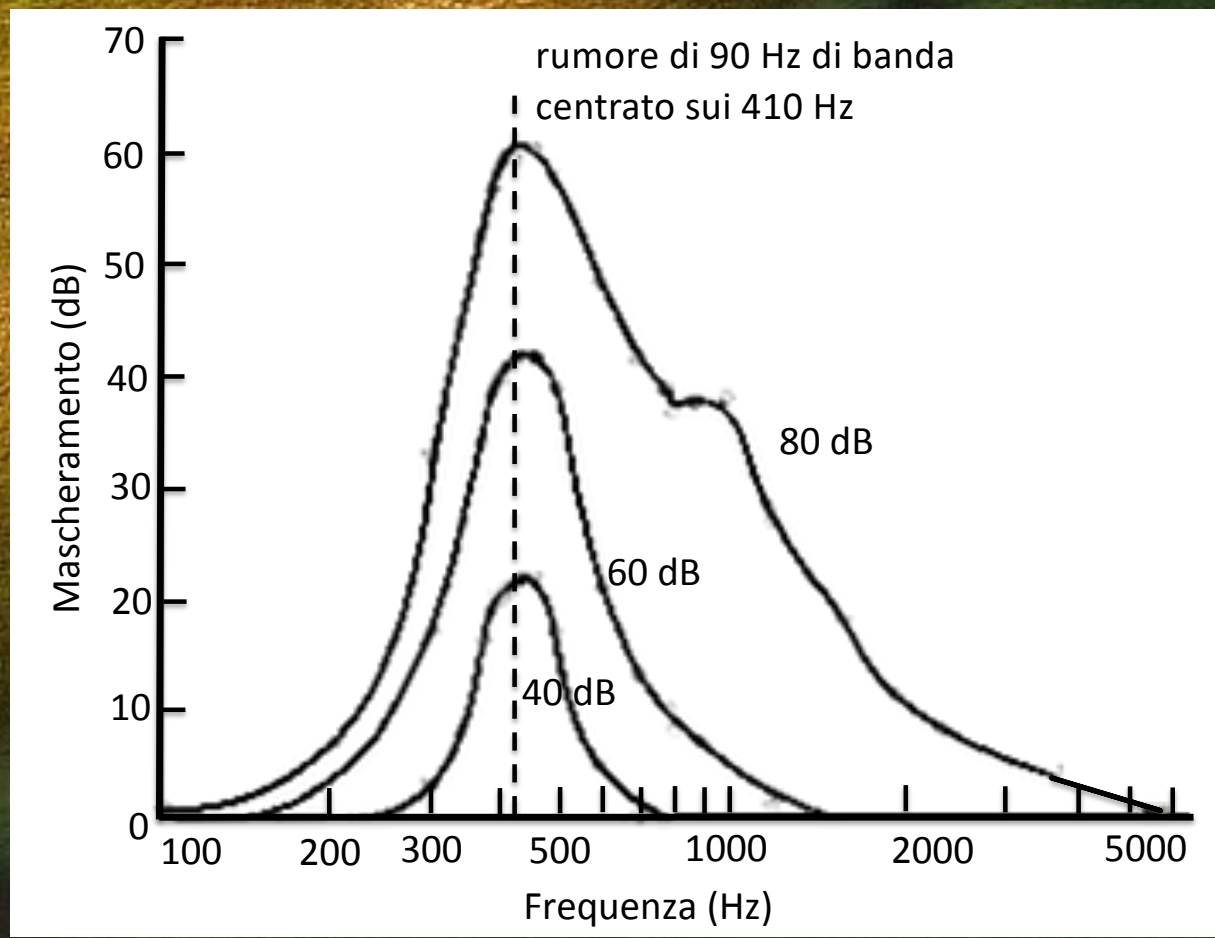
Mascheramento



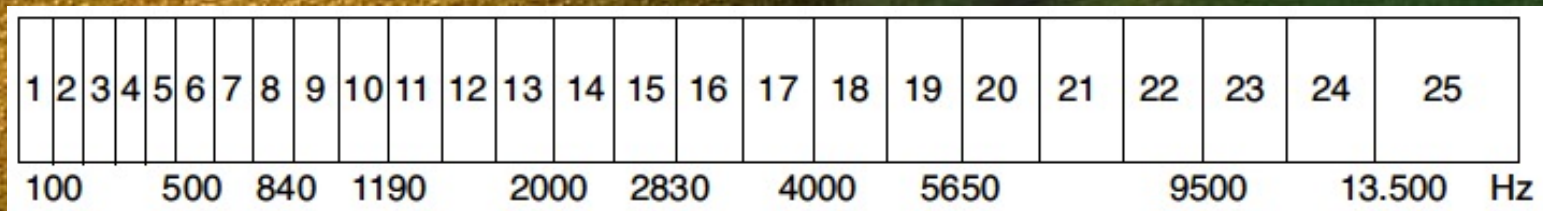
Il mascheramento tonale



Il mascheramento non tonale



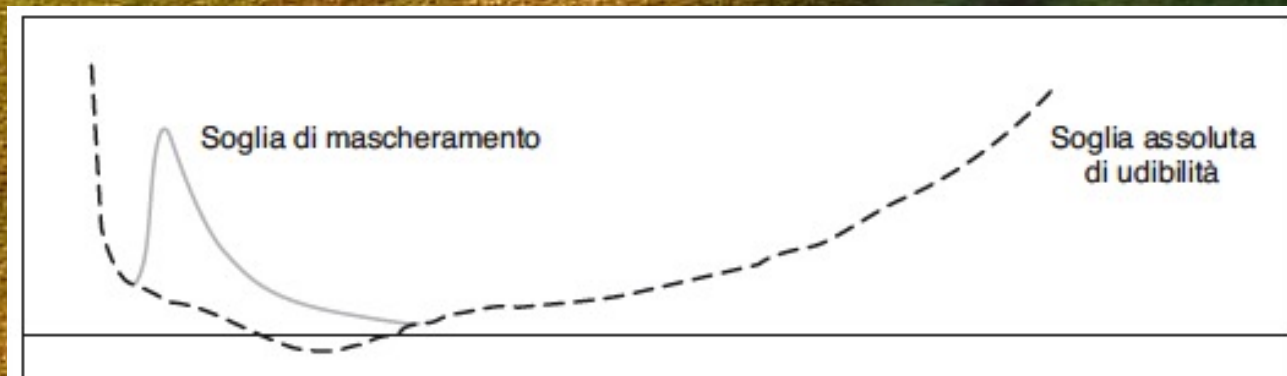
Bande critiche



Bande critiche di Fletcher

Numero di banda	Centro della banda	Estremi della banda e estensione
1	60	fino a 100 (80)
2	150	100-200 (100)
3	250	200-300 (100)
4	350	300-400 (100)
5	450	400-500 (100)
6	550	500-600 (100)
7	655	600-710 (110)
8	775	710-840 (130)
9	920	840-1000 (160)
10	1095	1000-1190 (190)
11	1300	1190-1410 (230)
12	1545	1410-1680 (270)
13	1840	1680-2000 (320)
14	2190	2000-2380 (380)
15	2605	2380-2830 (450)
16	3095	2830-3360 (530)
17	3680	3360-4000 (640)
18	4380	4000-4760 (760)
19	5205	4760-5650 (890)
20	6175	5650-6720 (1.050)
21	7360	6720-8000 (1.280)
22	8750	8000-9500 (1.500)
23	10.400	9500-11.300 (1.800)
24	12.400	11.300-13.500 (2.200)
25	16.700	da 13.500 (6.500)

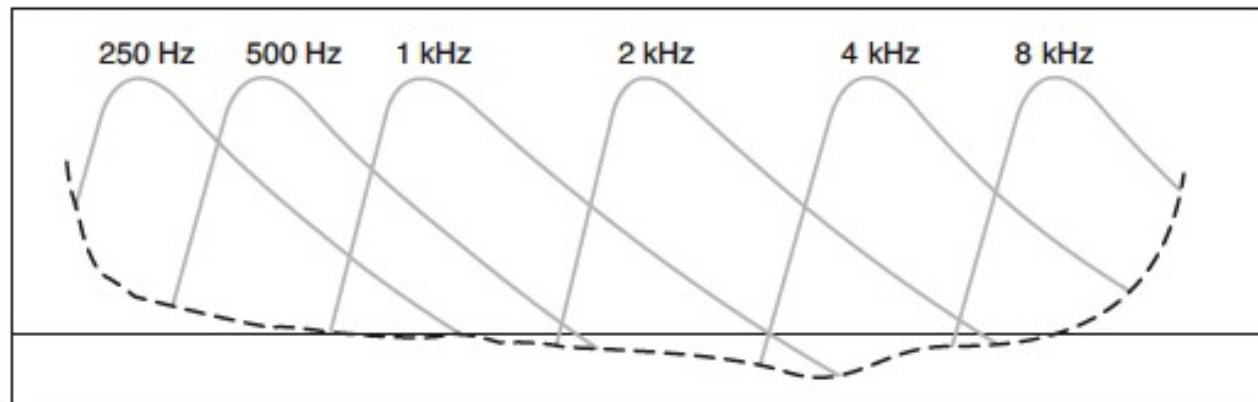
Soglie di mascheramento



(a)

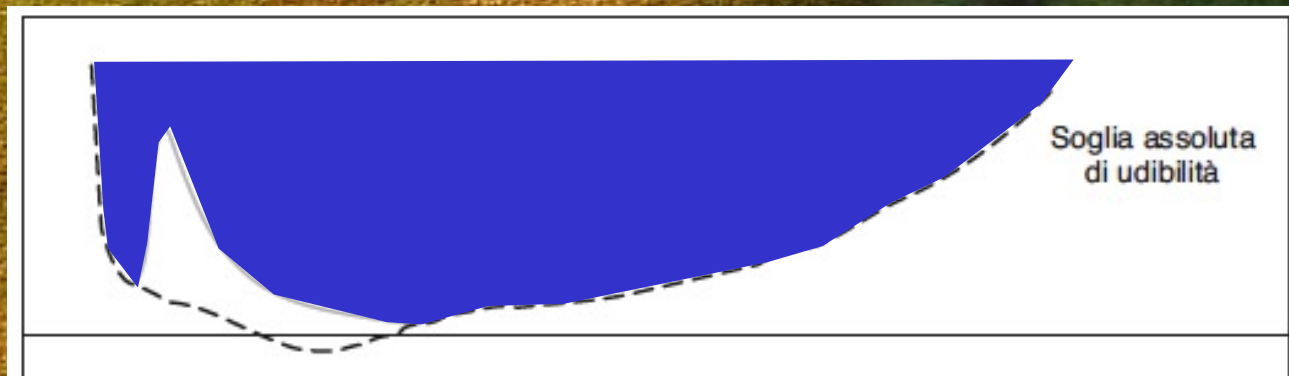
$$TH_N = E_T - 14,5 - B$$

$$TH_T = E_N - K$$



(b)

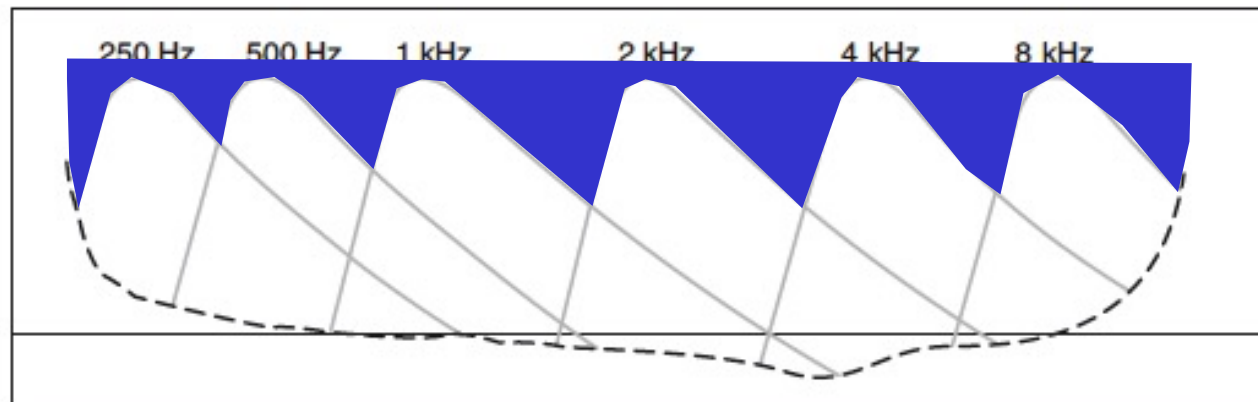
Suono effettivamente percepito



(a)

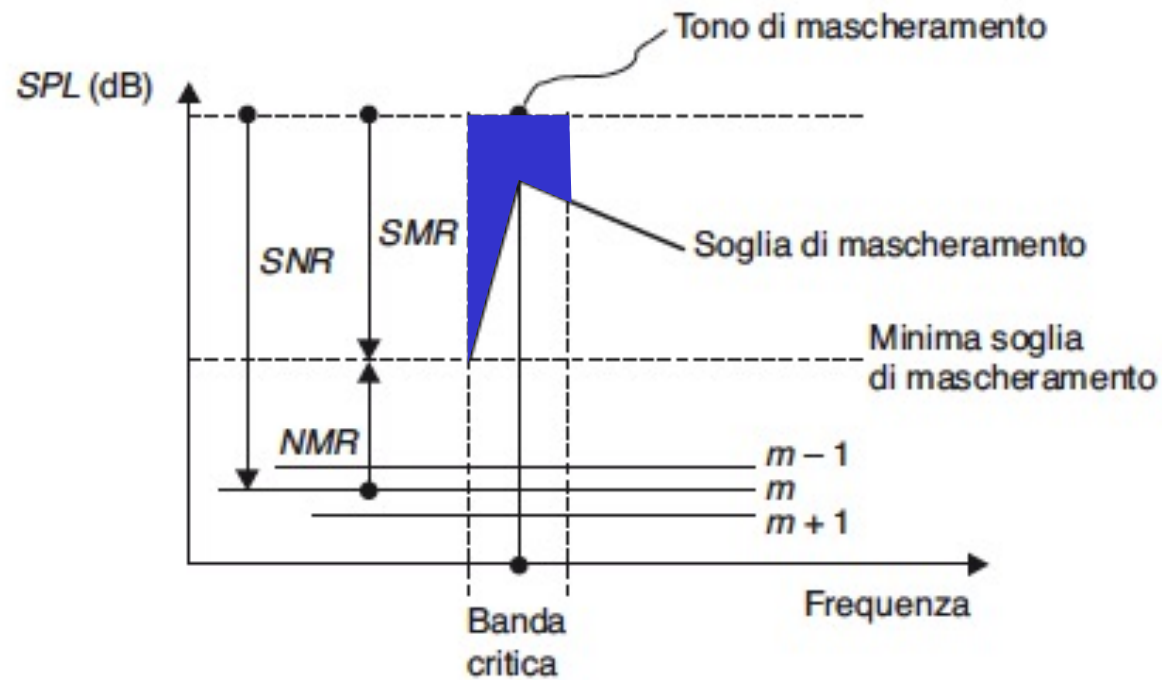
$$TH_N = E_T - 14,5 - B$$

$$TH_T = E_N - K$$



(b)

Soglie di mascheramento e SNR



$$SNR_{log} = \log\left(\frac{\text{max_ampiezza}}{\text{rumore_medio}}\right) = \log(\text{max_ampiezza}) - \log(\text{rumore_medio})$$

Tecniche di codifica

Codifica per blocchi (*block coding*)

Codifica del dominio trasformato (*transform coding*)

Codifica per sottobande (*sub-band coding*)

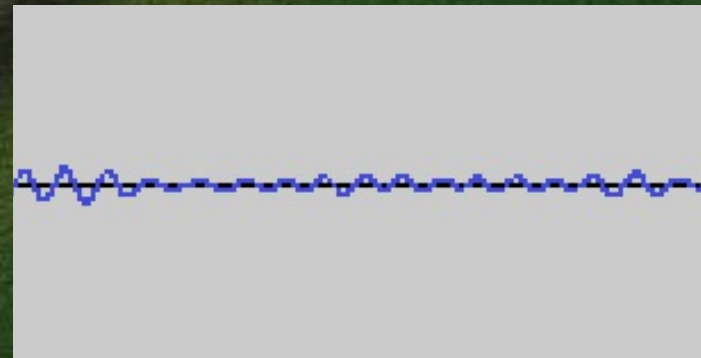
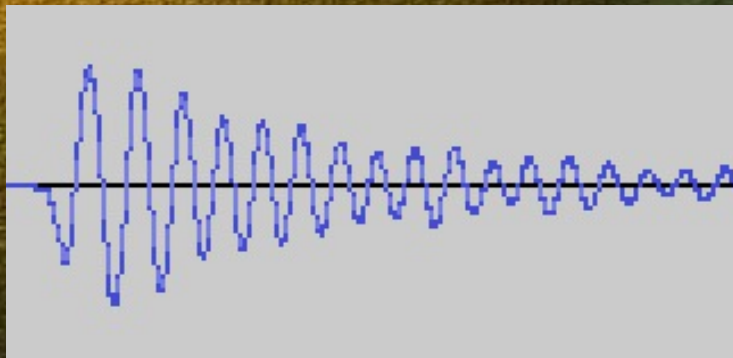
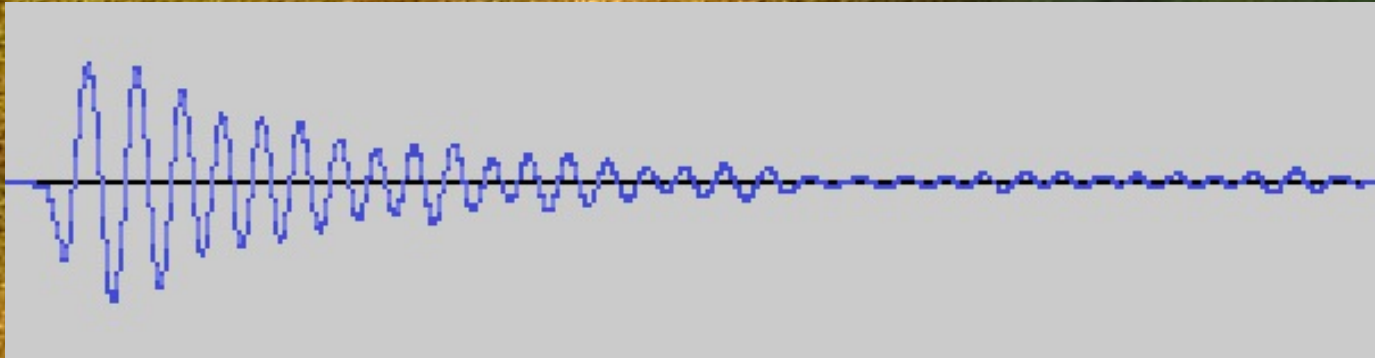
Codifica lossless (Huffman)

Codifica per blocchi (block coding)

5 - 6 - 5 - 4 - ... 2^y

- Codifica in virgola mobile: mantissa e esponente
 - Mantissa = valore da scalare dato l'esponente
 - Esponente come fattore di scala
- Motivazione: in un blocco, segnali audio reali non variano di ordine di grandezza → esponente (y)
- Risparmio: si rappresenta un solo esponente
- Metodo: calcolo del numero max di bit di esponente, e si mantiene per tutto il blocco
- Problema: pre-echi

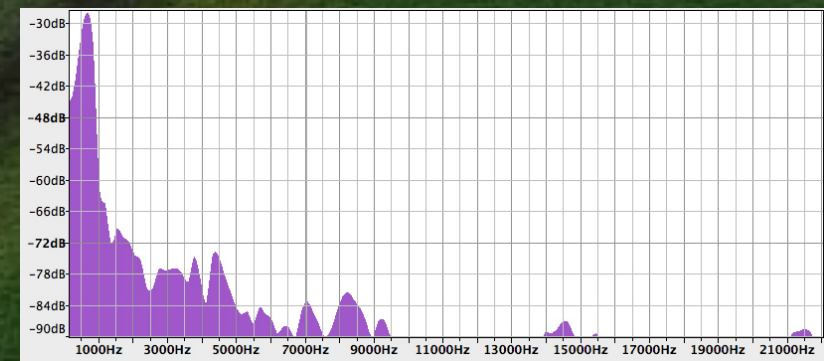
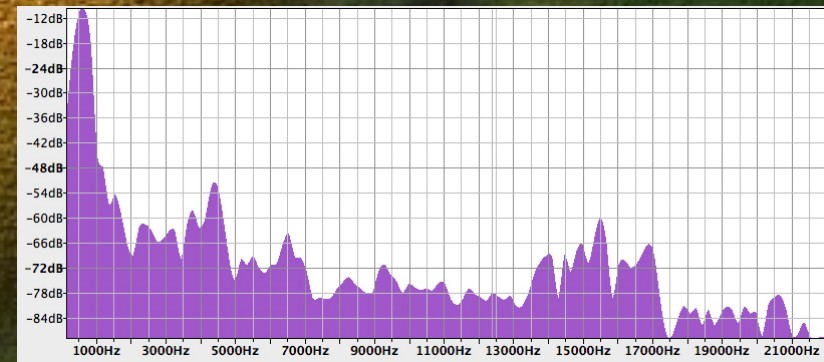
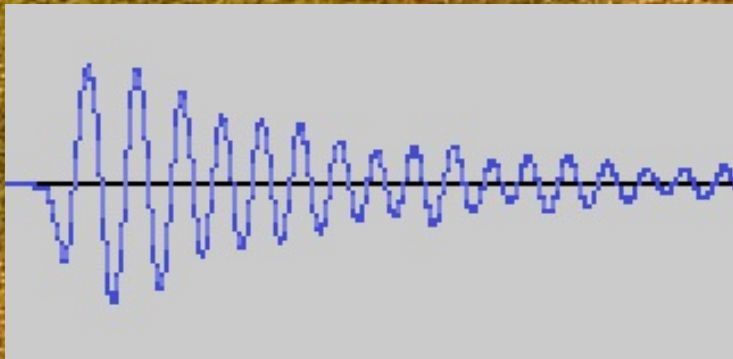
Block coding (esempio)



Codifica nel dominio della frequenza (transform coding)

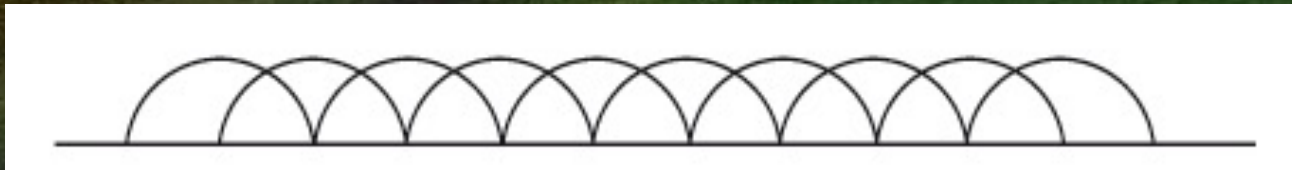
- Campioni di frequenza (e non di tempo)
- Motivazione: segnale audio cambia lentamente nel dominio della frequenza → tasso di campionamento inferiore
- Risparmio: maggiore ridondanza dei campioni → meno bit di quantizzazione
- Problema: Discriminare tra bande con diversa DR e mascheramento (SMR) → **subband coding**

Transform coding (Esempio)



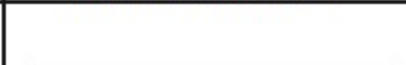







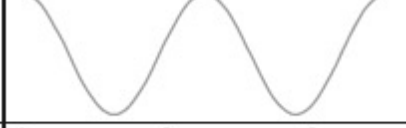


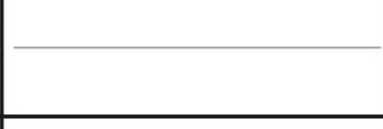
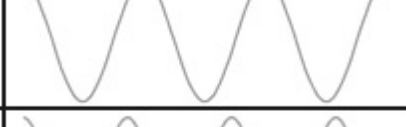
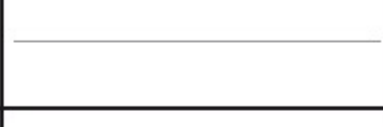


Transform coding: problema dei transitori

- “Finestrare” il segnale in segmenti molto brevi
- Problema: divisione netta può introdurre artifici agli estremi della segmentazione (pre-echi)
- Soluzione: sovrapposizione dei segmenti; ciascun campione appartiene a due finestre, con ruoli differenti

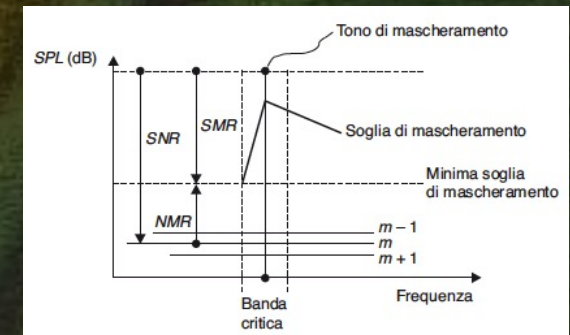


Transform coding (uso DCT)

Discrete
Cosine
Transform

#	Funzioni base della DCT	Coefficiente	Funzione risultante
0		0	
1		0,2	
2		0,8	
3		-0,5	
4		0,2	
5		0	
6		0	
7		0	

Codifica per sottobande (subband coding)



- Si divide lo spettro in bande da elaborare individualmente
- Vantaggio: le sotto-bande con DR ristretta si codificano con un minor numero di bit
- Numero di sotto-bande dipende dalle tecniche di compressione
 - massimo vantaggio dal fenomeno del mascheramento
 - nozione di banda critica, possibilmente restringendola

Compressione lossless: Algoritmo di Huffman

ALGORITMO DI HUFFMAN

1. Si selezionano due caratteri, x e y , con le frequenze minime
2. Si rimpiazzano x e y con un singolo carattere (fittizio) z , la cui probabilità è la somma delle probabilità di x e y
3. Si torna al passo 1 con un alfabeto ridotto, in cui z sostituisce x e y , finché l'alfabeto conta un solo elemento di frequenza 100%

Algoritmo di Huffman: esempio

~~A (60%)~~ ~~B (20%)~~ ~~C (8%)~~ ~~D (7%)~~ ~~E (5%)~~

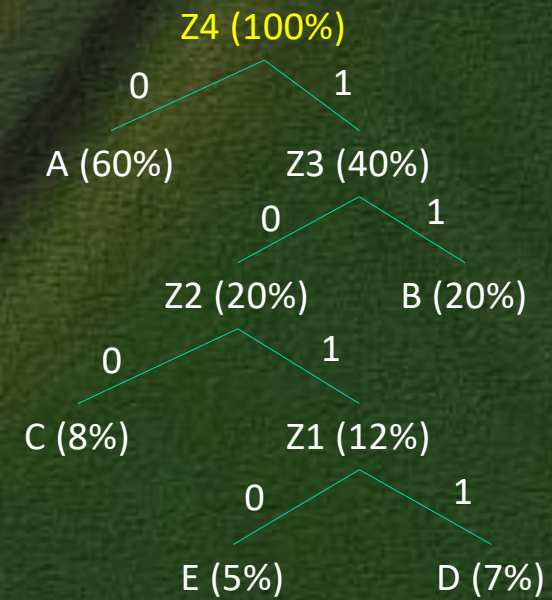
~~Z1 (12%)~~

~~Z2 (20%)~~

~~Z3 (40%)~~

Z4 (100%)

A 0
B 11
C 100
D 1011
E 1010



Algoritmo di Huffman: esempio

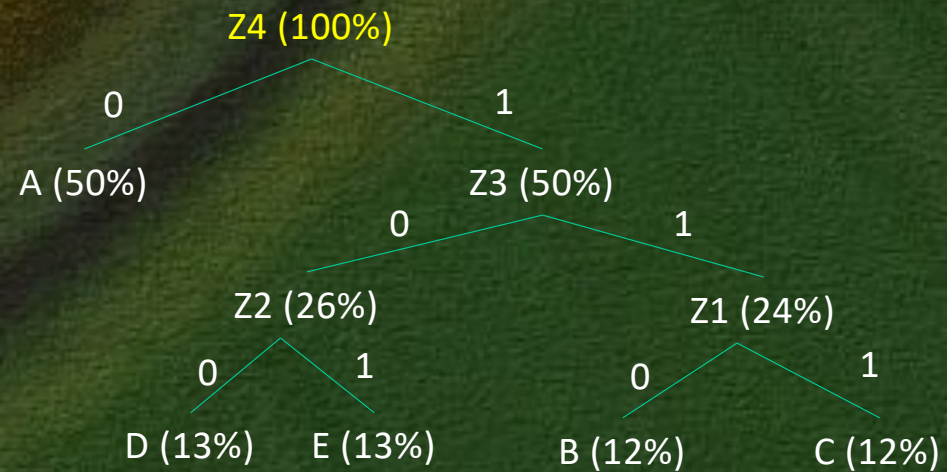
A (50%) B (12%) C (12%) D (13%) E (13%)

Alfabeto	Codifica
A	_____
B	_____
C	_____
D	_____
E	_____

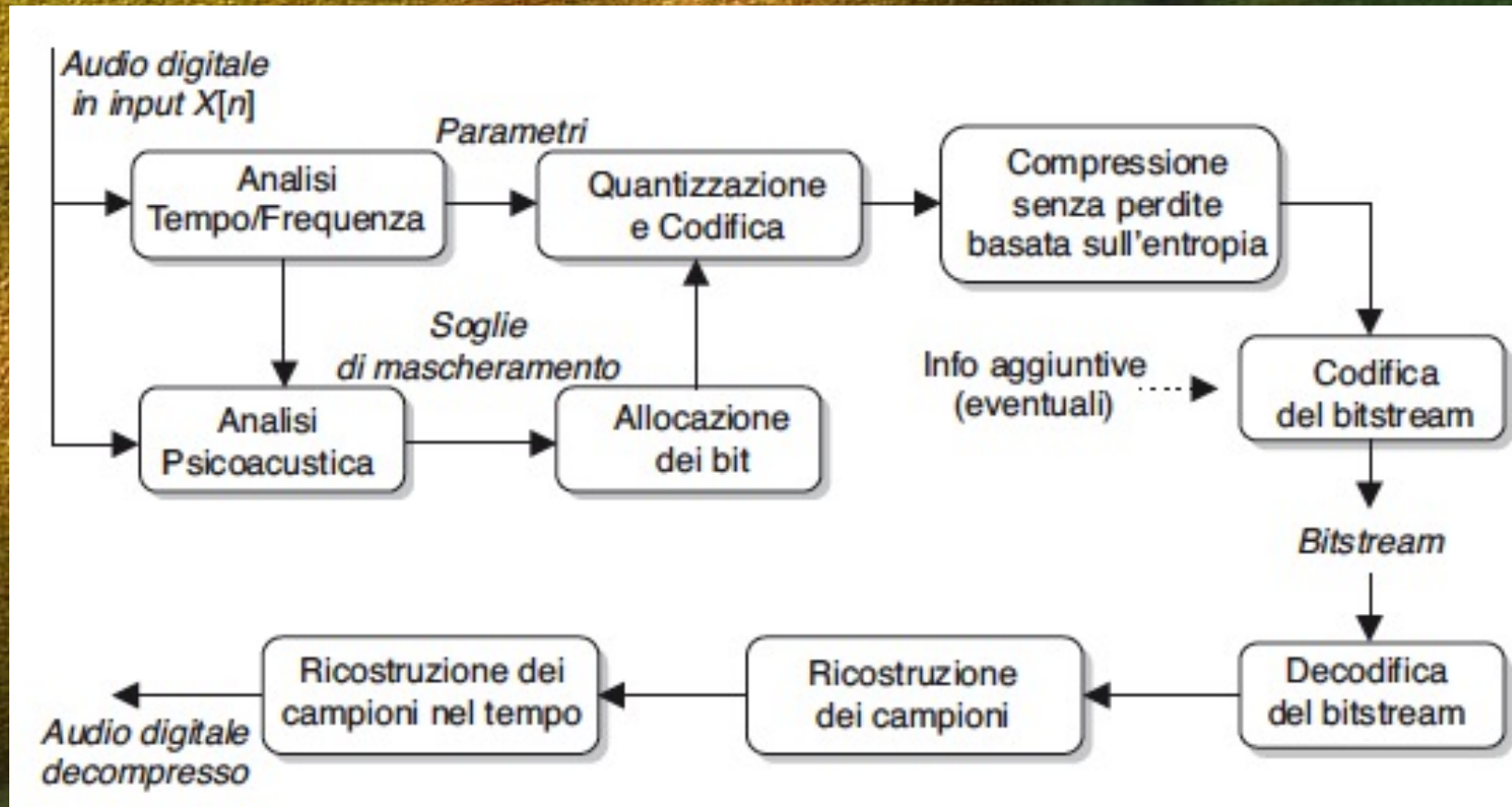
Algoritmo di Huffman: esempio

A (50%) ~~B (12%)~~ ~~C (12%)~~ ~~D (13%)~~ ~~E (13%)~~
 ~~Z1 (24%)~~ ~~Z2 (26%)~~
 Z3 (50%)

Alfabeto	Codifica
A	0
B	110
C	111
D	100
E	101



Schema generale di compressione percettiva





MPEG

Motion Picture coding Experts Group

Gruppo di lavoro di ISO/IEC per lo sviluppo di standard internazionali per

- ... compressione, decompressione, elaborazione ...
- ... e rappresentazione codificata di video, audio, combinazioni A/V.

Standard MPEG

- MPEG-1 (1992): codifica A/V a 1,5 Mbps
- MPEG-2 (1994): codifica A/V a 6 Mbps (surround + subwoofer)
- MPEG-3: per TV HD, assorbito da MPEG 2
- MPEG-4 (1999): A/V come oggetti in applicazioni multimediali
- MPEG-7 (2001): meta tag (XML), MPEG 47
- MPEG-21 (2001-): framework per ...
 - distribuzione e manipolazione di elementi digitali
 - Digital Rights Management – DRM
- MPEG-D (2007-): codifica dell'audio
 - MPEG Surround
 - Spatial Audio Object Coding
 - Unified speech and audio coding

<https://mpeg.chiariglione.org>

Principi MPEG

- retro-compatibilità: decoder MPEG2 ok per MPEG1
- min elementi normativi per uniformità rappresentazione e efficienza decodifica
 - formato dell'audio compresso
 - struttura dell'algoritmo di decodifica
- effettivo algoritmo di compressione MPEG audio lasciato alla creatività degli sviluppatori
- specifiche includono *encoder*, ricavato dai test

Diffusione MPEG

- Elettronica di consumo (con riproduttori portatili MP3 e MP4 – AAC)
- Telecomunicazioni (trasmissione audio su Internet)
- Multimedialità (inclusione di audio in prodotti multimediali off-line e on-line)

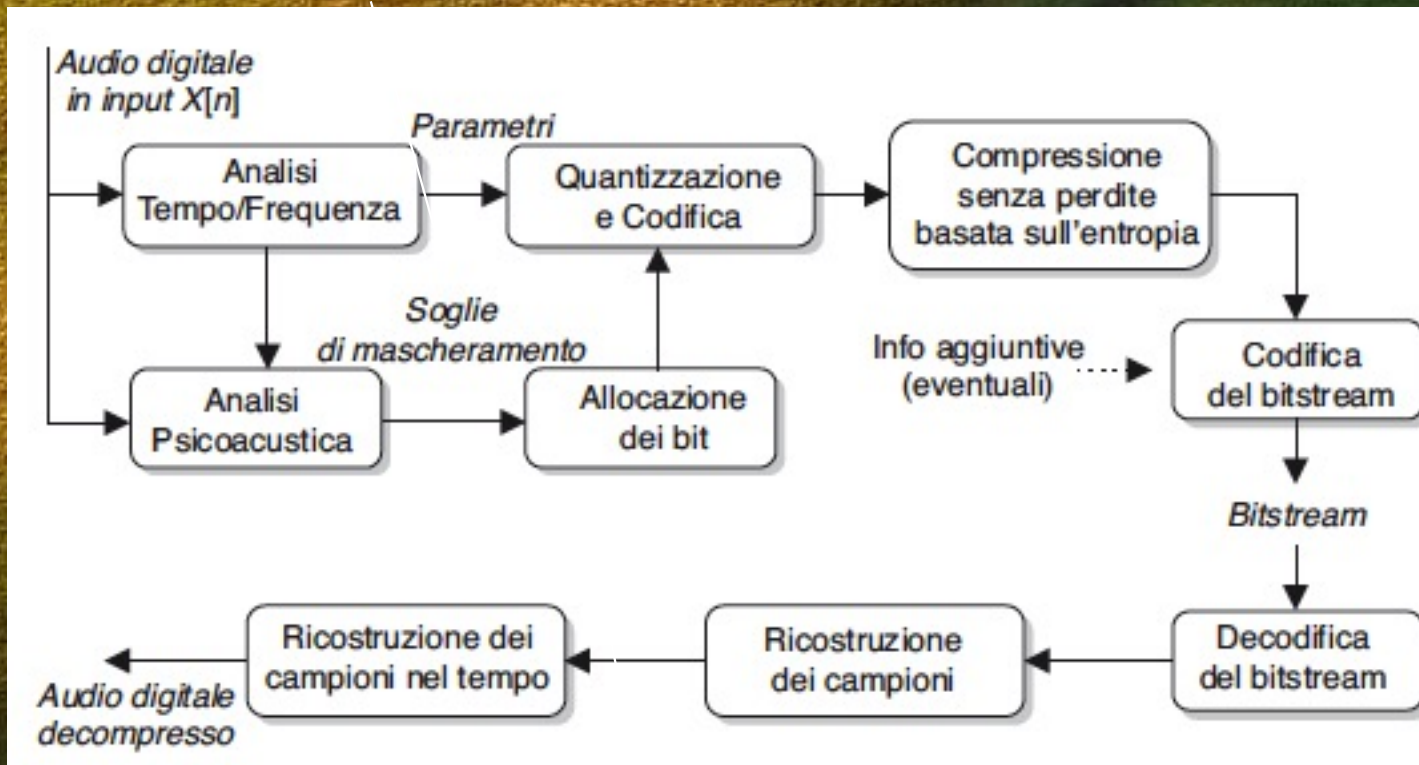
MPEG 1

- 1987 progetto Eureka (DAB), incl. Fraunhofer IIS e Uni Erlangen
- algoritmo molto potente ISO-MPEG Audio
- struttura di base comune su più layer: sottobande + modello psicoacustico
- tre livelli (layer) di compressione:
 - Layer I: bit-rate > 128 kbps, poca compressione
 - Layer II: bit-rate = 128 kbps
 - Layer III: bit-rate = 64 kbps, molta compressione

Schema generale MPEG

- segnale audio convertito nelle componenti spettrali
- banco di filtri (complessità diversa nei tre layer)

componenti spettrali
quantizzate e codificate



Compressione
lossless

scompatta
il bitstream

rumore di
quantizzazione
sotto soglia
di mascheramento
fattore di scala
per blocco
Compansion
Codici per subband

Da blocco di campioni di frequenza (subband)
A blocco di campioni audio

banco di filtri (modalità di sintesi) dalle sotto-bande

Modello psicoacustico e MPEG

- Presente solo nell'encoder, decoder semplice → Asimmetria MPEG, favorisce la riproduzione
- Nessun vincolo sul modello psicoacustico (qualità differisce)
- Implementazioni più accurate, FFT distinta
 - componenti spettrali tonali e non tonali
 - calcolo delle soglie di mascheramento globale

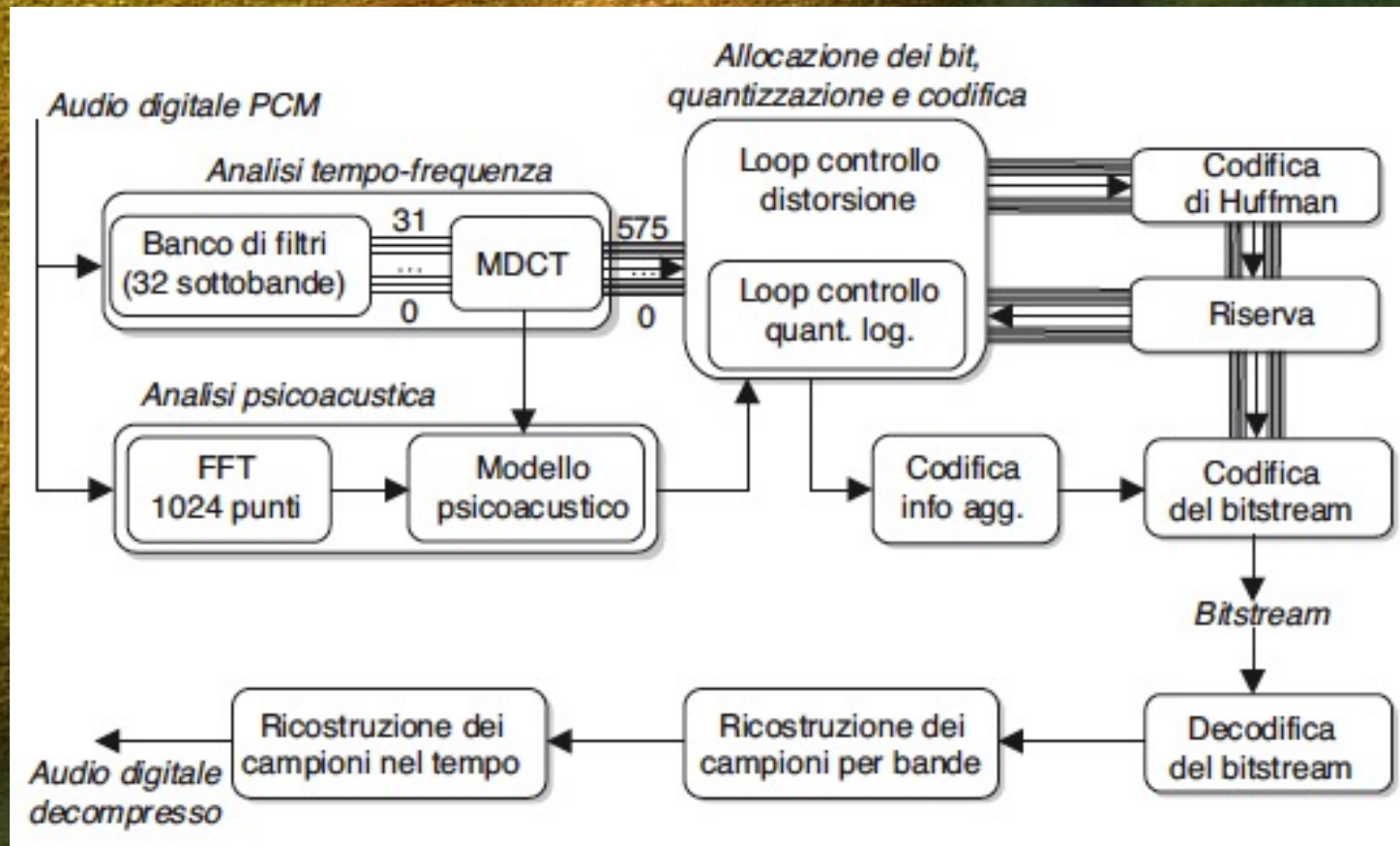
Modalità di compressione MPEG 1

- Più frequenze di campionamento (32/44,1/48)
- Bitstream compresso supporta mono, dual mono, stereo, joint stereo
- Bit rate da 32 a 224 kbps (compressione da 2,7 a 24 volte) – tassi fissi e variabili
- Supporta controllo e correzione errori e informazioni supplementari

Codifica joint stereo

- Risparmio quando canali L e R non molto diversi
- Metodo M=S (mid/side) stereo: $L+R$ / $L-R$ (lossless)
- Metodo intensity stereo (low quality):
 - percezione stereo per frequenze > 2 kHz basata su inviluppo temporale del segnale (non direzione)
 - $L+R$ per sotto-bande > 2 kHz
 - differenze tra L e R solo sui fattori di scala

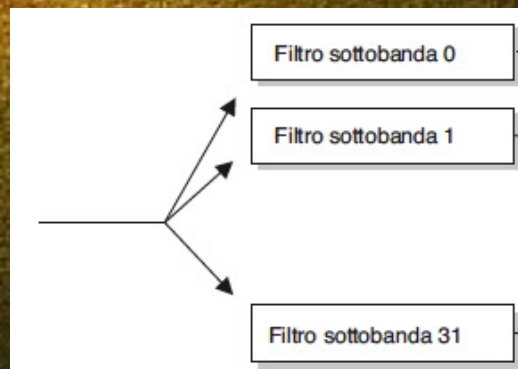
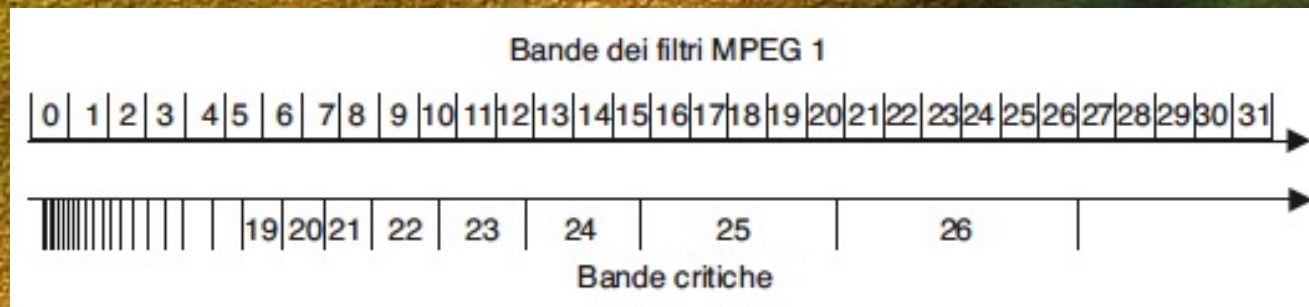
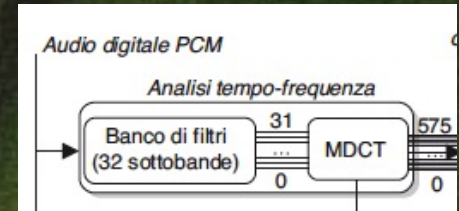
Schema MPEG – Layer III





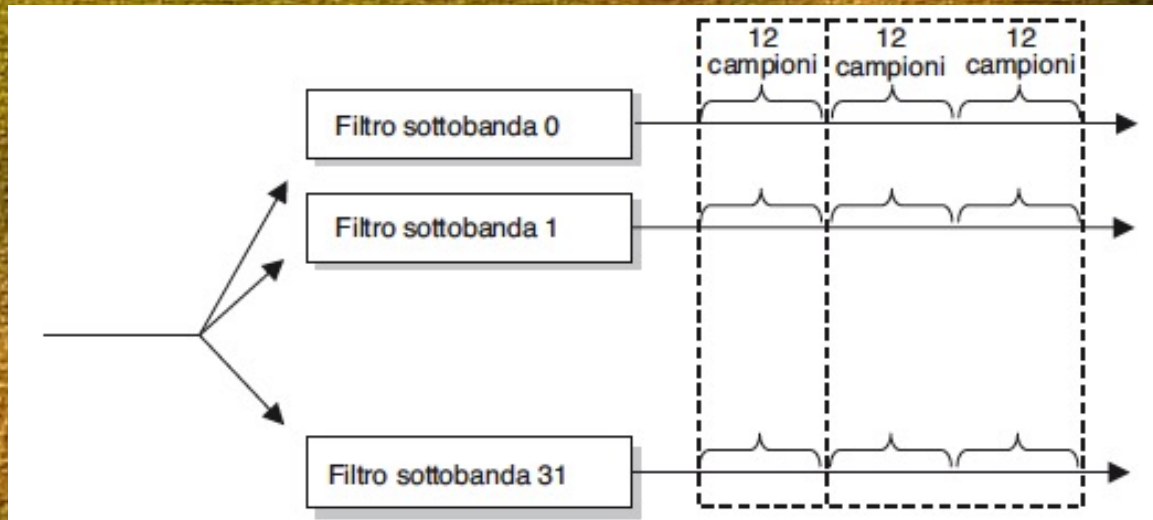
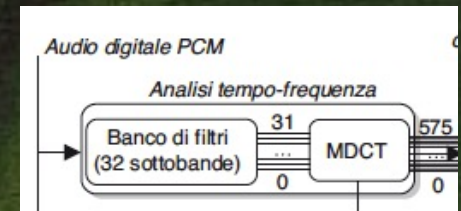
Elementi dello schema

Codifica subband: Filtro polifase



32 sottobande ugualmente spaziate

Block coding



Blocco temporale

a 48 kHz, 8 msec

384 campioni

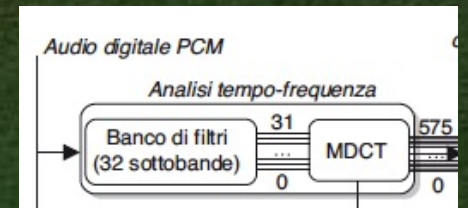
a 44,1 kHz, 8,7 msec

Blocchi (finestre, *frame*) di $(12 \times 32 =) 384 \times 3 = 1152$ campioni (36 per sotto-banda)

pre-mascheramento temporale, parola binaria ancora lunga

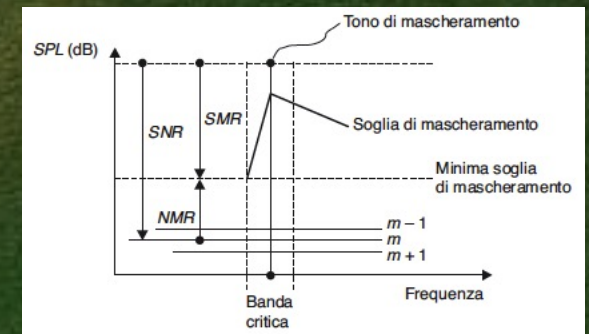
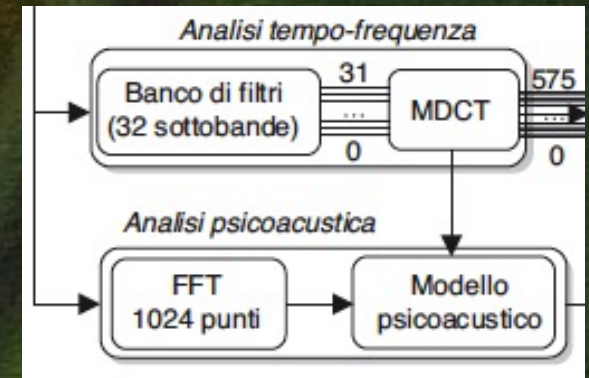
Layer III (MP3): Subband + transform coding

- In cascata ai filtri a 32 bande, modulo *MDCT*
 - Più elevata risoluzione frequenziale
 - Basse frequenze: partizione simile alle bande critiche
 - ...
 - risoluzione totale fino a 576 sottobande (32x18)
- Si differenzia il trattamento del segnale stazionario / transitori, si limitano i pre-echi



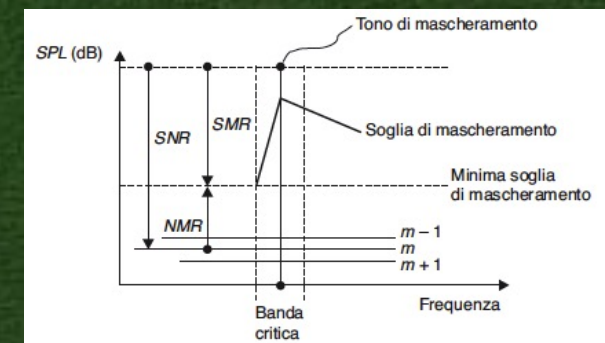
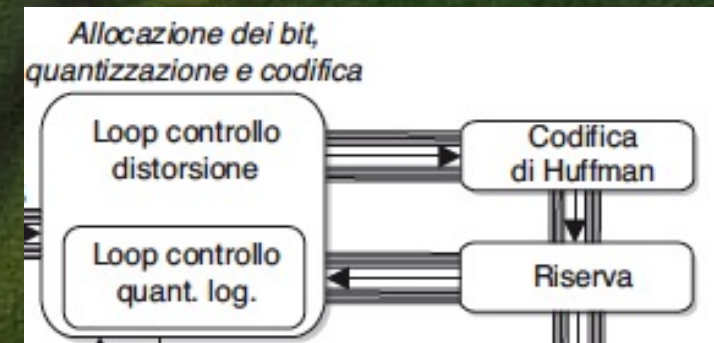
Modello psico-acustico

- FFT da 512 a 1024 punti
- Separazione del segnale in tonale e non tonale
 - “perceptual subband coding”
 - “perceptual noise shaping”
- Calcolo soglie mascheramento globale
- Calcolo rapporto SMR (Signal to Masker Ratio) per quantificare i dati da comprimere



Allocazione bit, quantizzazione, codifica

- Più cicli di quantizzazione e codifica
 - alloca i bit necessari per sottobanda (calcolo SMR)
 - quantizzazione non uniforme (III), compansion
 - sotto-bande raggruppate per fattore di scala
- Fattori
 - modello psicoacustico
 - bitrate in uscita
- Codici di Huffman per le 576
- Bitstream (III) soddisfa bit-rate variabili per sotto-banda
 - finestre di 1152 campioni, diversa lunghezza
 - “riserva” di bit

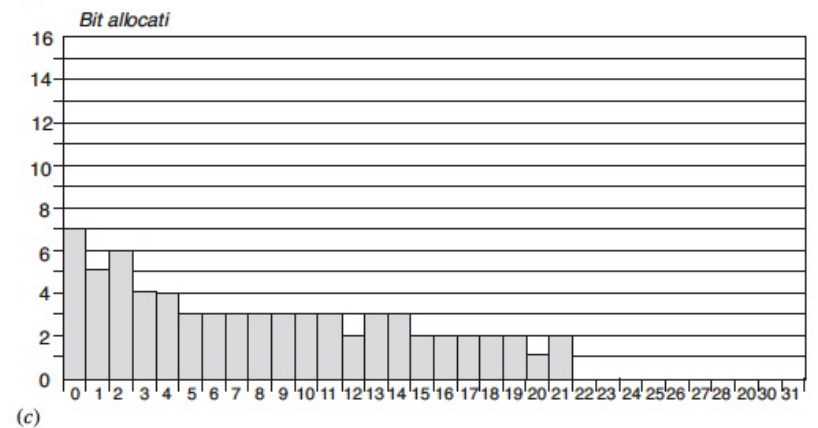
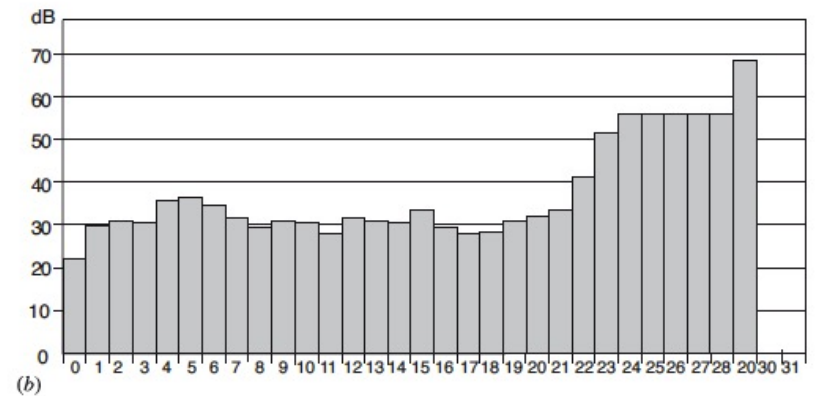
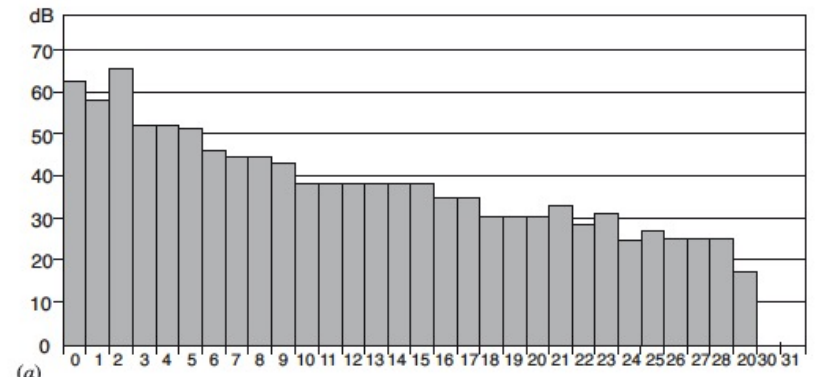


Esempio di allocazione

Spettro di ampiezza

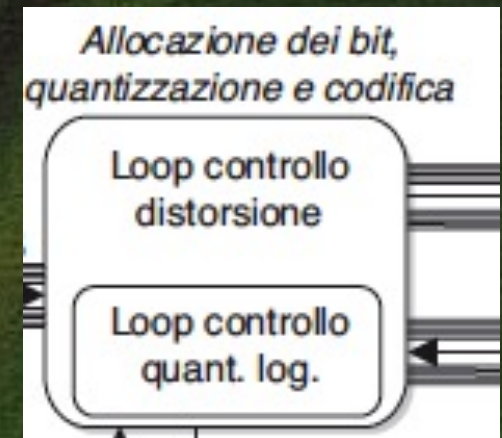
Soglie di mascheramento

Allocazione dei bit



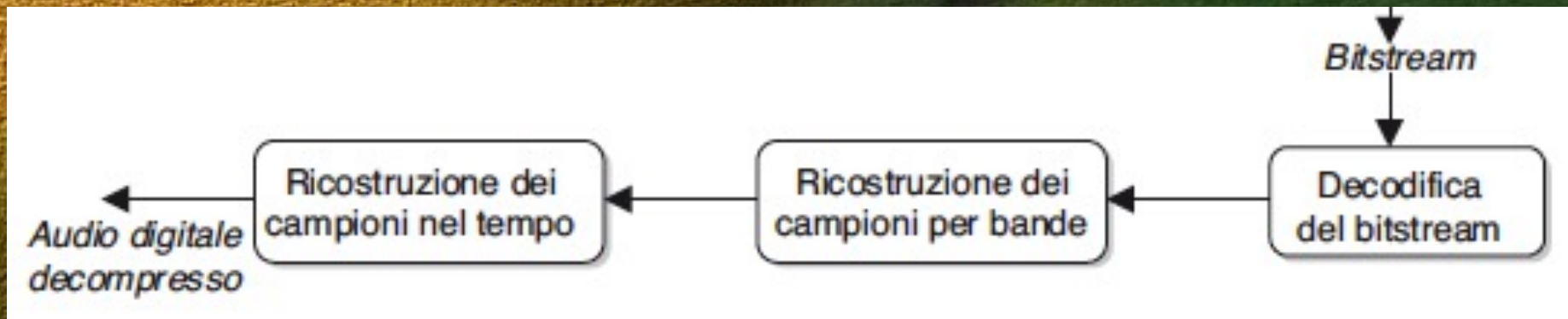
Loop e terminazione

- Due loop
 - Loop interno sui limiti del bitrate (RATE LOOP)
 - Loop esterno sul controllo del rumore per banda
- 3 casi di terminazione
 - In nessuna banda il rumore supera la soglia del mascheramento (no distorsione)
 - Iterare ancora comporta per qualche banda il superamento dei valori permessi
 - Prossima iterazione aumenta i bit per tutte le bande
- Gli encoder che lavorano in real time hanno anche limiti di tempo



Decodifica

- Sintetizza un segnale a partire dalle componenti spettrali codificate
- Non si ha più lo stesso segnale!!!
- Qualità dipende dal bit-rate



Performance di MP3

- qualità telefonica: 96:1 (2.5 kHz / mono / 8 kbps)
- meglio di AM radio: 24:1 (7.5 kHz / mono / 32 kbps)
- simile a FM radio: 26..24:1 (11 kHz / stereo / 56...64 kbps)
- quasi-CD: 16:1 (15 kHz / stereo / 96 kbps)
- CD: 14..12:1 (>15 kHz / stereo / 112..128kbps) prende approx. 1Mb/minute di spazio hard-disk

- Oltre: 8...4:1 per la musica acustica

L'encoder migliore

- La risposta dipende dalle esigenze
 - encoder veloci/lenti (> velocità, < fedeltà audio)
 - confrontare mp3 ottenuti da encoder diversi a parità di bit-rate
- Consiglio pratico (ovvio): creare MP3 con basso bitrate con encoder 'lenti'

Frontend MP3

- Interfaccia ai codec
- Parametro di qualità o bit-rate
- Alcuni frontend implementano funzioni per “normalizzare” il volume, o realizzano ID TAGS
- IDTAG è informazione (testuale, in genere) nel file di layer III (autore, titolo, etc ...)



Grazie dell'attenzione