

Rappresentazione dell'informazione

Notazione posizionale: valore cifra dipendente dalla posizione
(sistema decimale)

Numeri naturali

b (base)	2	8	10	16
insieme S	0, 1	0,1,2,3,4,5,6,7	0,1,2,3,4, 5,6,7,8,9	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
$c_i \in [0, b-1]$				

numero N (posizionale) = $c_{n-1} c_{n-2} \dots c_0$ con $c_i \in S$

Interpretazione in base decimale di un numero in base b

$$N_{10} = \sum_{i=0}^{n-1} c_i b^i, \text{ dove } n \equiv \text{numero cifre}$$

Es. $N_2 = 101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_{10}$
 $N_8 = 101_8 = 1 \cdot 8^2 + 0 \cdot 8^1 + 1 \cdot 8^0 = 65_{10}$
 $N_{10} = 101_{10} = 1 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0 = 101_{10}$

Conversione numero naturale da base 10 a base 2

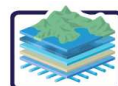
1. Numero iniziale diviso (divisione intera) ripetutamente per la base 2 fino a quando il risultato è 0;
2. i resti prodotti dalle divisioni intere presi in ordine inverso costituiscono la codifica cercata.

Osservazione: l'algoritmo converge sempre



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Esempio: $N_{10} = 26$

resti

base $\rightarrow 2$ \downarrow

26		0 - significativo
13		1
6		0
3		1
1		1+ significativo

0 \leftarrow fine

base 8

26		2
3		3
0		fine

$$26_{10} = 11010_2 = 32_8 = 1A_{16}$$

Conversioni utili

Binario \longleftrightarrow ottale

$$26_{10} = (0)11.010 = 32$$

Binario \longleftrightarrow esadecimale

$$26_{10} = (000)1.1010 = 1A$$

Rappresentazione dei numeri naturali nel calcolatore

L'unità atomica di rappresentazione è la cifra binaria (base 2) chiamata bit (BInary digiT).

Un numero naturale binario è memorizzato in una parola di memoria composta da N bit (cifre).



Impossibile rappresentare tutti i numeri naturali: ∞ richiede ∞ bit

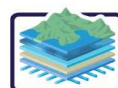


La rappresentazione finita riduce quindi l'intervallo rappresentabile



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Intervallo di rappresentabilità

Dati N bit quanti e quali numeri naturali si possono rappresentare?

2^N combinazioni, ossia tutti i numeri nell'intervallo $[0, 2^N-1]$

N	2^N
10	1024 (1K)
16	65536(64K)
20	$\cong 1.000.000(M)$
32	$\cong 4$ miliardi (G)

Corrispondenza tra combinazione binaria e numero per N=16

0000 0000 0000 0000 = 0

0000 0000 0000 0001 = 1

0000 0000 0000 0010 = 2

...

0000 0000 0001 1010 = 26

1111 1111 1111 1111 = 65535

Osservazioni:

- l'intervallo di rappresentabilità è finito
- tutti i numeri all'interno dell'intervallo sono rappresentabili

Aritmetica binaria

Somma

00001 ←riporti

011001 + 25+

100001 = 33=

111010 58

Sottrazione

00100 ←prestiti

101001 - 41 -

100101 = 37 =

000100 4

Prodotto

010 x 2 x

101 = 5 =

010

000

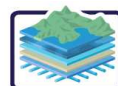
010

01010 10

Aumenta il num. di bit necessari ->



POLITECNICO
DI MILANO



SpatialDBgroup

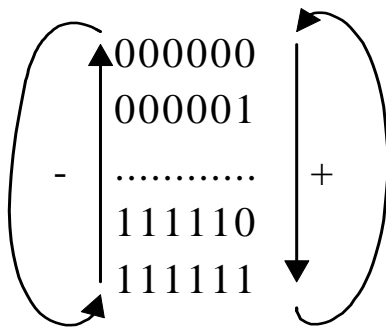
Politecnico di Milano – DEI –
Prof. Mauro Negri

Osservazioni

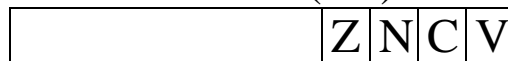
- possibile produzione di valori non rappresentabili (es. $65535 + 1$)
- Ogni operazione produce sempre un risultato

$$\begin{array}{r} 1111\ 1111\ 1111\ 111 \\ 1111\ 1111\ 1111\ 1111\ + \\ 0000\ 0000\ 0000\ 0001\ = \\ \hline (1)0000\ 0000\ 0000\ 0000 \end{array}$$

Circolarità nel calcolo



Registro di stato della CPU (SR)



Z = 1 se addizione = 0

N = 1 se risultato addizione è negativo

C = 1 se riporto sul bit + significativo in addizione (carry)

V = 1 se overflow

Un esempio più concreto?

.....

```
int main()
```

```
{int c, int v=1;
```

```
while (1)
```

```
{v=v+1000000; if (v<0)scanf("%d",&c); printf("\n%d",v);}
```

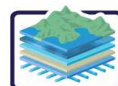
```
}
```

Quando viene superato 2.147.483.647 il programma produce numeri negativi



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Attenzione: se un'operazione A/B produce un numero reale si avrà un troncamento.

Errore assoluto è costante < 1

Errore relativo che diminuisce al crescere dei numeri.

Rappresentazione dei numeri interi

L'intervallo di rappresentabilità viene suddiviso tra numeri positivi e negativi

Esempi di intervalli f(hardware, linguaggio)

Ogni linguaggio, compilatore decide come gestire questi limiti.

Tipi C	Bit	Intervallo possibile (limits.h)
<i>short integer</i>	16	-32768 a 32767 <i>SHRT_MIN, SHRT_MAX</i>
<i>unsigned short integer</i>	16	0 a 65535 <i>USHRT_MAX</i>
<i>Integer</i>	32	-2147483648 a 2147483647 <i>INT_MIN, INT_MAX</i>
<i>unsigned integer</i>	32	0 a 4294967295 <i>UINT_MAX</i>
<i>long integer</i>	32	-2147483648 a 2147483647 <i>LONG_MIN, LONG_MAX</i>
<i>unsigned long integer</i>	32	0 a 4294967295 <i>ULONG_MAX</i>

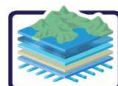
Convenzione codifica in modulo e segno (N bits)

- bit + significativo: 1 (negativi), 0 (positivi)
- valori rappresentabili: $-(2^{N-1}) \longleftrightarrow (2^{N-1})$
- doppia rappresentazione dello 0



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Esempio:

111(-3), 110(-2), 101(-1), 100(-0), 000(+0), 001(+1), 010(+2), 011(+3)

Codifica in complemento a 2

- si abbandona la regola del cambio base
- valori rappresentabili: $-(2^{N-1}) \longleftrightarrow (2^{N-1}-1)$
- primo bit mantiene il segno

Esempio:

100(-4), 101(-3), 110(-2), 111(-1), 000(+0), 001(+1), 010(+2), 011(+3)

Come si calcola

Numeri positivi: codifica binaria del valore assoluto

Numeri negativi: $|N_2| + (N_2) = 0$

ES. -3

$$\begin{array}{rcl} & 011 & (3) \\ & 101 & (-3) \\ & \text{-----} & \\ (1)000 & & \end{array} \quad \rightarrow N_2 = 2^N - |N_2|$$
$$\begin{array}{r} 1000- \\ 011 \\ \text{-----} \\ 101 \end{array}$$

Regole immediate per numeri negativi

1. complemento a 1 di $+N_2 + 1$ ($011 \rightarrow 100 + 1 \rightarrow 101$)
2. ricopiare bit – significativi di N_2 sino a primo 1 incluso e poi complemento a 1 dei rimanenti.

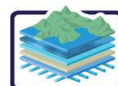
Osservazioni:

- unica rappresentazione dello 0 (complemento a 2)
- sottrazione realizzata come somma



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Torniamo all'overflow di $A + B$

Come vengono caricati i bit C e V

$A > 0, B > 0$

- $C = 0$

- se risultato ha primo bit a 0 \Rightarrow no overflow altrimenti $V = 1$

011 + 3

011 = 3

110 -2

$A > 0, B < 0$ (o viceversa)

$$A + (2^N - |B|) = 2^N - (|B| - A)$$

- $V = 0$

- $|B| > A \rightarrow C = 0$

010 + 2

101 = -3

111 -1

- $|B| \leq A \rightarrow C = 1$ corretto dopo eliminazione riporto

011 + (3)

011 + (3)

101 = (-3)

110 = (-2)

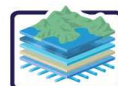
(1)000

(1)001



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

$A < 0, B < 0 \quad (2^N - |A|) + (2^N - |B|) = 2^N + 2^N - (|B| + |A|)$
 - C=1 riporto è strutturale e quindi non significativo
 - se risultato ha primo bit a 1 \Rightarrow no overflow altrimenti V=1

V=0 111 + -1
 101 = -3

 (1)100 -4

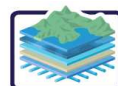
V=1 101+ (-3)
 101= (-3)

 010 \rightarrow +2



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Rappresentazione dei caratteri

- lettere maiuscole/minuscole A a .. Z z
- spazio
- cifre 0 9
- segni di interpunzione , : ; .
- simboli ! " # % @) < =
- caratteri di controllo per gestire la visualizzazione, la stampa, la trasmissione dei caratteri (*inizio riga, salto di riga, salto pagina*)

Codifica ASCII (American Standard Code for Information Interchange) basata su 8 bit

- da 0 a 32 caratteri speciali
- 65 A, 66 B, 67 C, 97 a, 98 b ordinamento per concetto di precede/segue e distanza tra due lettere e tra la stessa lettera minuscola e maiuscola (costante)
- codifiche nazionali nell'intervallo 128-255 \Rightarrow portabilità

Codifica Unicode (ISO 10646)

- 16 bit
- include alfabeti: latino, greco, arabo, ideogrammi cinesi,...

Osservazione: la doppia interpretazione dei caratteri in C

char b;

scanf("%c",&b); z

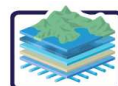
printf("%c\n",b); z

printf("%d\n", b); 122



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup

Rappresentazioni di immagini come bitmap

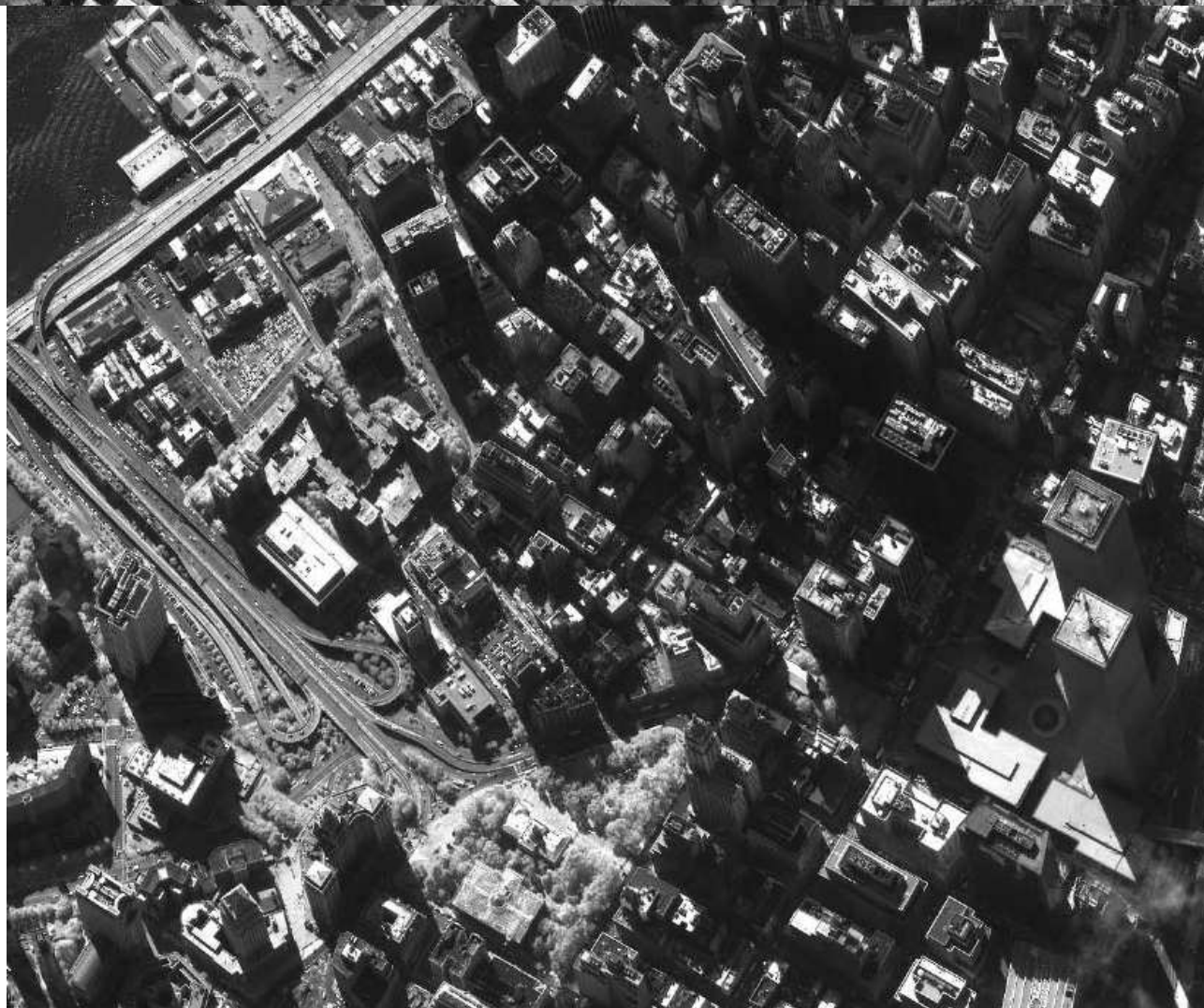


Immagine → digitalizzazione (cm x cm → punti x punti)
↑
risoluzione dpi(dot per inch - es. 1200 per fotogrammetria)

Numero bit x punto (pixel):

1. bianco/nero 1 bit x pixel
2. scala grigi (256 toni) 8 bit x pixel
3. colori a video: sistema RGB (256 toni per colore) 8 bit x 3 colori (24 bit)

Dimensioni: area(pixel * pixel)*bit x punto.

Es

schermo colori: $(1400 \times 1050) \times 3 \times 8 = 35.280.000$ bits ~ 4,4Mbytes

Distanza di distinguibilità

$2,56 \text{ cm (pollice)} \times 1/\text{dpi (ad es., 1200)} \cong 21\mu\text{m}$

Fotogrammetria: risoluzione al suolo

scala	scala media fotogramma	risoluzione al suolo
1000	4500	$21\mu\text{m} \times 4500 = 94500\mu\text{m} \cong 9,5\text{cm}$
2000	7000	$21\mu\text{m} \times 7000 = 147000\mu\text{m} \cong 14,7\text{cm}$
5000	13000	$21\mu\text{m} \times 13000 = 273000\mu\text{m} \cong 27,3\text{cm}$
10000	22000	$21\mu\text{m} \times 22000 = 462000\mu\text{m} \cong 46,2\text{cm}$

Tecniche di compressione

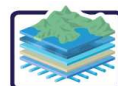
senza
TIFF

con perdita di informazione
JPEG



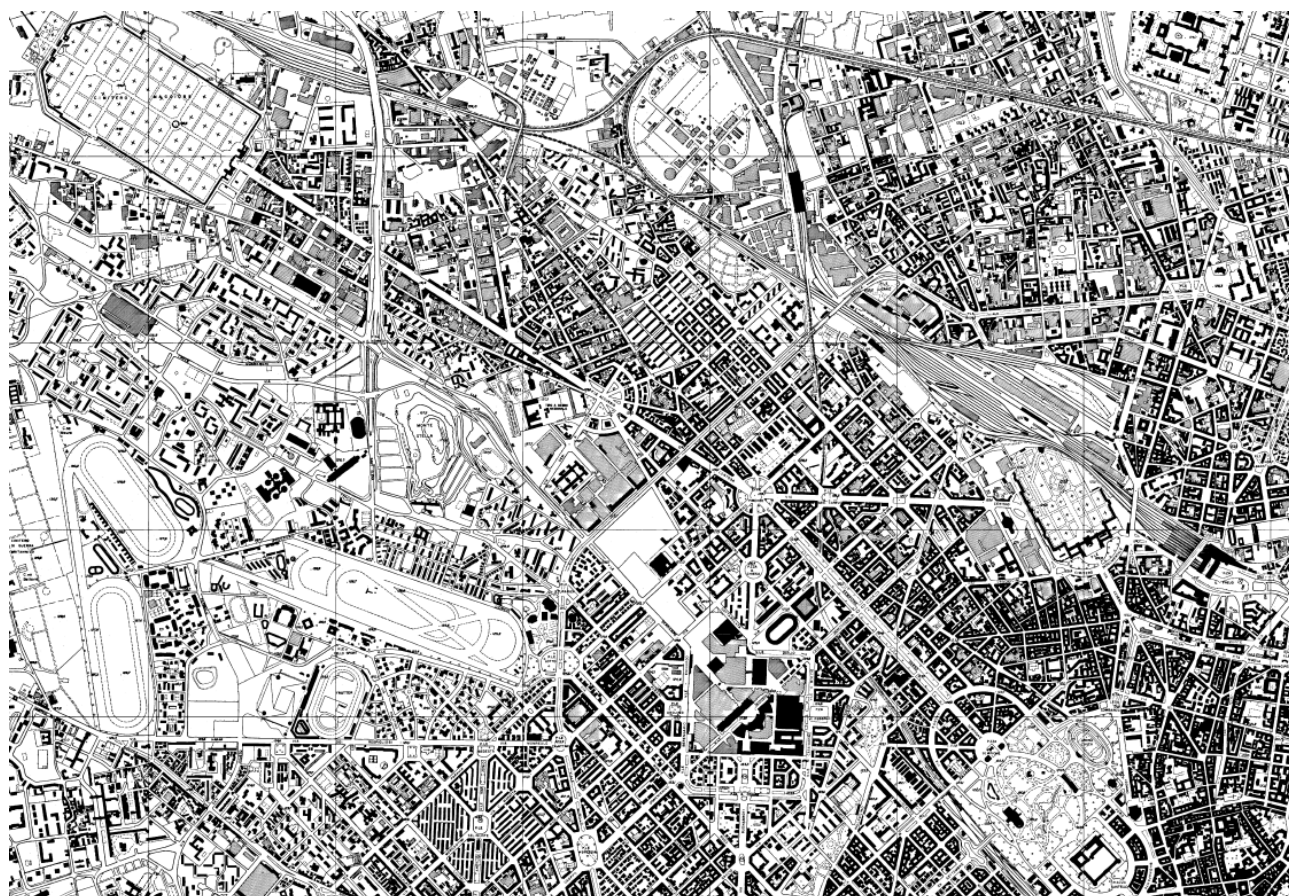
POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri

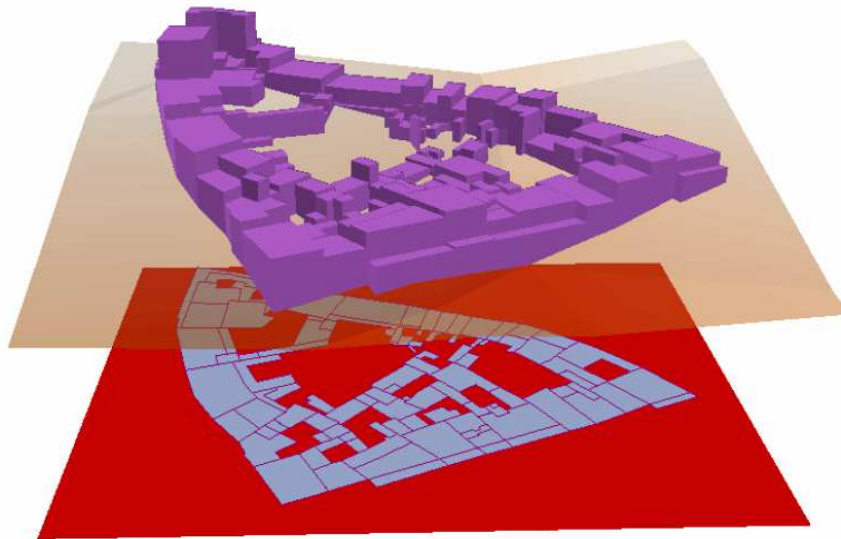


SpatialDBgroup

Dalle immagini alla sintesi vettoriale (cartografia)

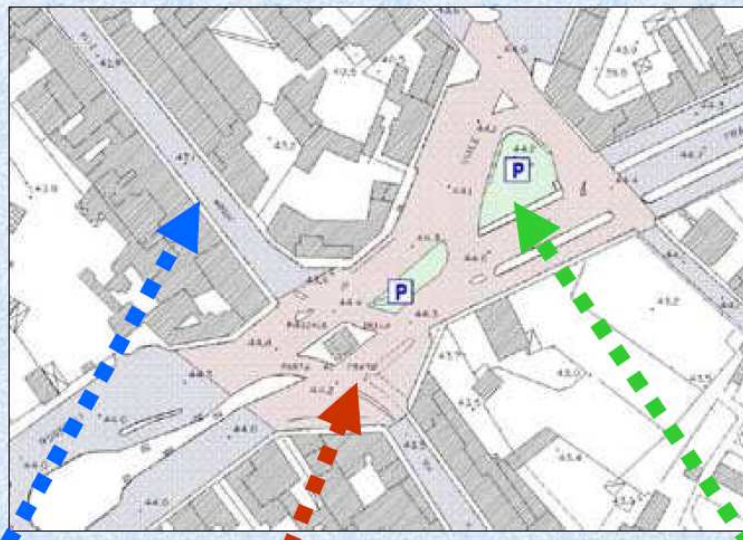


EDIFICIO e VOLUMETRIE



modellazione 3D

STRADE - AREA CIRCOLAZIONE VEICOLARE



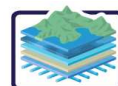
Tronco di
Carreggiata

Area Traffico
Strutturato
(es. INCROCIO)

Area a Traffico non
Strutturato
(es. PARCHEGGIO)



POLITECNICO
DI MILANO



SpatialDBgroup

Rappresentazione dei video

Video: sequenza di immagini

Sistema PAL: 720x576 pixel - 25 frame/sec.

$$(720 \times 576 \times 3 \times 8) \times 25 = (1,244 \text{ MB}) \times 25 = \sim 31 \text{ MB/sec}$$

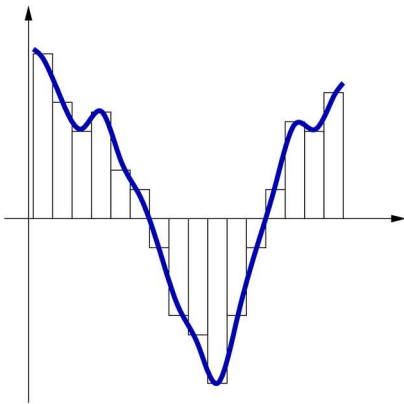
Film 133 minuti = 7980'' = ~ 247GB (29 DVD SSDL-8,5GB)

Tecniche di compressione variabile MPEG-2

- variazione tra fotogramma i-esimo e (i-1)-esimo
- velocità media 3,5Mb/sec

Film 133 minuti = 7980'' * 3,5 = ~28Gb = ~3,5GB (1,2GB restanti per sottotitoli, lingue) \Rightarrow 1 DVD SSSL-4,7GB)

Rappresentazione del suono



Frequenza campionamento = $2 \times 22.000 \text{ Hz} = \sim 44.100 \text{ campioni/sec}$

Numero bit per campione per canale: 16

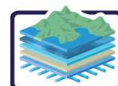
Dimensione = $44.100 \times 16 \times 2 = 1,4 \text{ Mb/sec} = 172 \text{ KB/sec}$

Voce (freq. Camp. $2 \times 4.000 \text{ Hz}$)



POLITECNICO
DI MILANO

Politecnico di Milano – DEI –
Prof. Mauro Negri



SpatialDBgroup