

## La rappresentazione dell'informazione

## Digitalizzare l'informazione

- *Digitalizzare*: rappresentare l'informazione per mezzo di cifre (ad es: da 0 a 9)
- Limitazioni delle cifre
  - rappresentazione alternativa: ogni insieme di simboli potrebbe rappresentare un numero telefonico, a patto che il tastierino numerico sia etichettato in modo opportuno

7-2

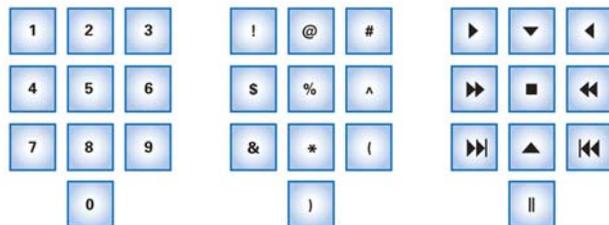


Figura 6.1 Tre diversi assegnamenti di simboli per il tastierino di un telefono.

7-3

## Ordinamenti

- Un vantaggio dell'uso dei numeri per la codifica è la possibilità di ordinare gli elementi
- Per utilizzare altri simboli, dobbiamo definire un *ordinamento* dei simboli stessi
  - dal valore più piccolo al più grande

7-4

## Codificare con i dadi

- Consideriamo una rappresentazione con i dadi
  - ogni dado ha sei facce e le configurazioni delle facce possono essere utilizzate per rappresentare l'informazione digitale

7-5

## Codificare con i dadi (cont.)

- Consideriamo la rappresentazione dell'alfabeto con i dadi
  - 26 lettere, solo 6 configurazioni su un dado
  - Utilizziamo più configurazioni per rappresentare ogni lettera
  - Quante configurazioni sono richieste?
    - 2 dadi producono 36 combinazioni ( $6 \times 6$ )
    - 3 dadi ne producono 216 ( $6 \times 6 \times 6$ )
    - n dadi ne producono  $6^n$

7-6

## Codificare con i dadi (cont.)

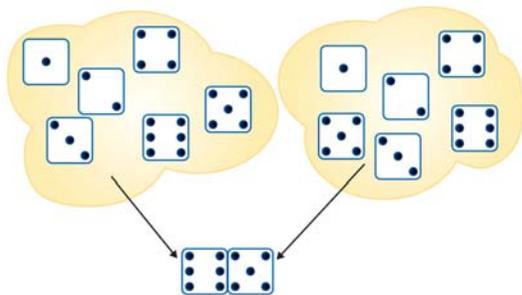


Figura 6.2 Una sequenza di due dadi consente di rappresentare  $6 \times 6 = 36$  combinazioni diverse.

7-7

## Codificare con i dadi (cont.)

- Chiamiamo simbolo ogni configurazione prodotta da una coppia di dadi:
  - 1 e 1 == A
  - 1 e 2 == B
  - 1 e 3 == C
  - 1 e 4 == D
  - 1 e 5 == E
  - 1 e 6 == F
  - 2 e 1 == G
  - ecc.

7-8

## Codificare con i dadi (cont.)

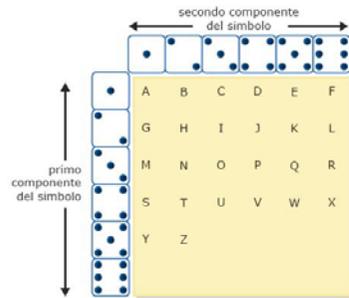


Figura 6.3 Un assegnamento iniziale di lettere ai simboli formati con le coppie di dadi.

7-9

## Estendere la codifica

- 26 lettere dell'alfabeto sono state rappresentate; 10 spazi sono rimasti non associati
- Questi spazi possono essere usati per le cifre arabe
- Cosa ci occorre per includere la punteggiatura? Abbiamo solo 36 spazi in due dadi. In che modo possiamo evitare di ricorrere a un sistema a tre dadi?

7-10

## Creare nuovi simboli con il carattere di escape

- Possiamo usare l'ultima configurazione di dadi come simbolo di "escape". Non corrisponde ad alcun carattere legale, quindi non lo useremo mai nel normale processo di digitalizzazione
- Indica che la digitalizzazione sta "sfuggendo" alla rappresentazione base

7-11

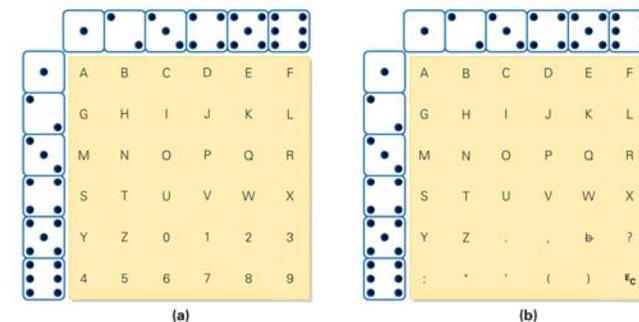


Figura 6.4 Due rappresentazioni complete che usano coppie di dadi come simboli (nota: il simbolo  $\text{¢}$  indica lo spazio).

7-12

## La rappresentazione binaria dell'informazione

- Le configurazioni fondamentali usate nell'IT scaturiscono dall'incontro tra mondo fisico e mondo logico
- Nel mondo fisico la più importante forma di informazione è la presenza o assenza di un fenomeno
- Nel mondo logico, sono importanti i concetti di vero e falso

7-13

## La rappresentazione PandA

- *PandA* è l'abbreviazione per "presenza e assenza"
- È *discreta* — il fenomeno è presente o non lo è (vero o falso). Non esiste alcuna gradazione di valori

7-14

## Un sistema binario

- Due configurazioni—Presente e Assente—fa di PandA un *sistema binario*
- L'unità base di PandA è il "*bit*" (abbreviazione di *binary digit*)

7-15

Presenza	Assenza
Vero	Falso
1	0
Acceso	Spento
+	-
Sì	No
Favorevole	Contrario
Yang	Yin
Lisa	Bart

Tabella 6.1 Alcune possibili interpretazioni delle due configurazioni PandA.

7-16

## I bit nella memoria del computer

- La memoria è organizzata come una lunghissima sequenza di bit (posizioni in cui si può impostare la presenza/assenza di un fenomeno e successivamente rilevarla)
- Analogia: un sentiero come memoria
  - un sentiero costituito da blocchi quadrati rappresenta la memoria e un ciottolo è presente o assente
  - se su un quadrato c'è un ciottolo, il valore è 1, se il ciottolo non c'è il valore è 0

7-17



Figura 6.5 Un pezzo di sentiero di cemento può codificare una sequenza di bit (1010 0010).

7-18

## Codifiche PandA alternative

- Esistono altri modi per codificare due stati per mezzo di fenomeni fisici
  - possiamo usare pietre su tutti i quadrati, ma nere per uno stato e bianche per l'altro
  - possiamo usare più pietre di due colori per quadrato, dicendo che più la preponderanza di nere significa 0 e la preponderanza di bianche significa 1

7-19

## Combinare più bit

- Finché abbiamo solo due configurazioni, dobbiamo combinarle in una sequenza per creare abbastanza simboli per codificare l'informazione
- Nella rappresentazione PandA, con 2 configurazioni e sequenze di lunghezza  $n$ , possiamo creare  $2^n$  simboli

7-20

$n$	$2^n$	simboli
1	$2^1$	2
2	$2^2$	4
3	$2^3$	8
4	$2^4$	16
5	$2^5$	32
6	$2^6$	64
7	$2^7$	128
8	$2^8$	256
9	$2^9$	512
10	$2^{10}$	1024

Tabella 6.2 Il numero dei simboli che si possono esprimere con un numero variabile di bit.

7-21

## Numerazione posizionale nel caso decimale

- Per trovare la quantità espressa da un numero decimale ogni cifra va moltiplicata per il valore della sua posizione: alla fine si sommano tutti i risultati

$10^3$	$10^2$	$10^1$	$10^0$	Valori delle posizioni decimali
1	0	1	0	Cifre del numero decimale
$1 \times 10^3$	$0 \times 10^2$	$1 \times 10^1$	$0 \times 10^0$	Moltiplica ogni cifra per il valore della posizione
1000	0	10	0	Somma tutto per ottenere 1010 decimale

Tabella 9.1 Il numero decimale 1010 rappresenta mille e dieci:  $1000 + 10$ .

7-22

## Numerazione posizionale nel caso binario

- Funziona nello stesso modo del decimale, tranne che la sua base è 2
- Dato un numero espresso in binario, possiamo trovare la rappresentazione decimale corrispondente moltiplicando le cifre per il valore della posizione e sommando tutti i risultati

$2^3$	$2^2$	$2^1$	$2^0$	Valori delle posizioni binarie
1	0	1	0	Bit del numero binario
$1 \times 2^3$	$0 \times 2^2$	$1 \times 2^1$	$0 \times 2^0$	Moltiplica ogni bit per il valore della posizione
8	0	2	0	Somma tutto per ottenere 10 decimale

Tabella 9.2 Il numero binario 1010 rappresenta il decimale dieci:  $8 + 2$ .

7-23

## La rappresentazione esadecimale

Perché usare le cifre esadecimali? Scrivere una sequenza di bit è lungo, noioso ed è facile commettere errori

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Le 16 cifre (o *hex*) possono essere rappresentate perfettamente dai 16 simboli delle sequenze di 4 bit
- Convertire da hex a binario e viceversa:
  - data una sequenza di bit, si prendono i bit 4 per volta e si scrive la cifra esadecimale corrispondente
  - data una cifra hex, si scrive il gruppo di 4 bit a essa associato

7-24

simbolo	binario	hex	simbolo	binario	hex
AAAA	0000	0	PAAA	1000	8
AAAP	0001	1	PAAP	1001	9
AAPA	0010	2	PAPA	1010	A
AAPP	0011	3	PAPP	1011	B
APAA	0100	4	PPAA	1100	C
APAP	0101	5	PPAP	1101	D
APPA	0110	6	PPPA	1110	E
APPP	0111	7	PPPP	1111	F

Tabella 6.3 I sedici simboli della rappresentazione PandA a quattro bit.

7-25

## Digitalizzare il testo

- Originariamente la rappresentazione binaria—1 e 0—è stata utilizzata per la codifica dei numeri e dei caratteri
- Oggi si digitalizzano anche i suoni, i video e altri tipi di informazione
- Per codificare il testo, quali simboli devono essere inclusi?
  - anche se desideriamo rendere piccola la lista dei simboli onde usare meno bit, non vogliamo escludere caratteri importanti

7-26

## Assegnare i simboli alla codifica

- 26 lettere minuscole e 26 lettere maiuscole, 10 cifre, 10 segni aritmetici, 20 segni di interpunzione (spazi inclusi) e 3 caratteri non stampabili (a capo, tabulazione, backspace) = 95 caratteri, che rappresentano la lingua Inglese
- Per 95 simboli sono necessari 7 bit
- Una vecchia rappresentazione a 7-bit è il codice *ASCII* (*American Standard Code for Information Interchange*)

7-27

## Extended ASCII: un codice a 8 bit

- A metà degli anni '60 divenne chiaro che il codice a 7 bit non bastava per rappresentare i caratteri dei linguaggi diversi dall'inglese
- IBM ha esteso il codice ASCII a 8 bit e 256 simboli
- Chiamato "*Extended ASCII*", la prima metà è costituita dall'originale codice ASCII (con uno 0 aggiunto all'inizio di ogni gruppo di bit)
- Permette di esprimere la maggior parte dei linguaggi occidentali e include molti altri simboli utili

7-28

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0010	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	°
1000	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
1001	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
1010	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
1011	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figura 6.6 Extended ASCII, l'American Standard Code for Information Interchange.

7-29

## La codifica ASCII di un numero telefonico

- Il numero telefonico 888 555 1212 come sarà rappresentato nella memoria di un computer?
- Codifica ogni cifra in un byte ASCII

7-30

## L'alfabeto NATO per le telecomunicazioni

- Il codice per le telecomunicazioni è volutamente inefficiente, affinché le lettere siano intelligibili anche in presenza di rumore

A	Alpha	H	Hotel	O	Oscar	V	Victor
B	Bravo	I	India	P	Papa	W	Whiskey
C	Charlie	J	Juliet	Q	Quebec	X	X-ray
D	Delta	K	Kilo	R	Romeo	Y	Yankee
E	Echo	L	Lima	S	Sierra	Z	Zulu
F	Foxtrot	M	Mike	T	Tango		
G	Golf	N	November	U	Uniform		

Tabella 6.4 L'alfabeto NATO è progettato per non essere minimo.

7-31

## Perché "byte"?

- Perché BYTE si scrive con la Y?
- Gli ingegneri dell'IBM stavano cercando di definire una quantità di memoria tra un bit (*bocconcino, assaggio*) e una parola (in generale lunga 32 bit). Sembrava appropriato usare il termine *bite* (boccone) per il gruppo di 8 bit, ma per evitare errori si decise di cambiare la i con la y

7-32

## Rappresentare l'informazione multimediale

### I colori nero e bianco

- Ogni intensità RGB è rappresentata da un byte
  - l'intensità più bassa è 0000 0000
  - la più alta è 1111 1111
    - l'intervallo dei valori va da 0 a 255
- Il nero corrisponde all'assenza di colore; il bianco è dato da tutti i componenti a intensità massima

7-35

### Digitalizzare i colori

- Rappresentazione binaria dei colori RGB
  - un colore sul monitor è specificato da tre componenti: rosso, verde e blu
    - l'intensità di un colore è rappresentata da una quantità (da 0 a 255)
- Numeri binari e numeri decimali
  - il numero delle cifre è la base di un sistema
  - i numeri binari sono composti da due sole cifre, 0 e 1
  - i numeri decimali da 10 cifre, da 0 a 9

7-34

### Modificare un colore con un'addizione

- Quali colori rappresentano questi valori?  
1100 1000 1100 1000 1100 1000
- Ogni byte corrisponde al valore decimale 200. Il colore è rappresentato da RGB (200, 200, 200).
  - In HTML, scritto in esadecimale è #C8C8C8
  - ha una quantità uguale di rosso, verde e blu, è più vicino al bianco che al nero (grigio di media intensità)
  - tutti i colori con componenti RGB uguali sono il nero, il bianco e il grigio
- Aumentare l'intensità con un'addizione binaria
  - per ottenere un grigio più chiaro, occorre modificare i valori in modo che si avvicinino al bianco

7-36

## Un colore più chiaro: sommare con il riporto

- L'addizione binaria è simile all'addizione decimale
    - il procedimento va da destra a sinistra, sommando le cifre nelle varie colonne
    - talvolta il risultato può essere espresso con una sola cifra ( $1+0=1$ )
    - talvolta la somma non può essere espressa da una sola cifra ma occorre fare un riporto nella colonna a sinistra
- ( $1+1=10$ , abbiamo scritto 0 e riportato 1)

7-37

## Overflow

- Dato che i computer utilizzano sequenze di bit di dimensioni prefissate, cosa succede quando c'è il riporto del bit più a sinistra?
- *errori di overflow*
  - i computer segnalano quando i calcoli che stanno eseguendo danno origine a un overflow; sta al programmatore trovare il modo di recuperare

7-38

## Elaborare una rappresentazione

- Cambiare i colori di una foto della luna
  - immaginiamo di aver digitalizzato una foto in bianco e nero della luna
  - nel computer, i pixel della foto formano una lunga sequenza di terne di byte RGB. Che valori hanno?
    - ci sono solo i neri, bianchi o grigi
  - e se volessimo fare una versione colorata?

7-39

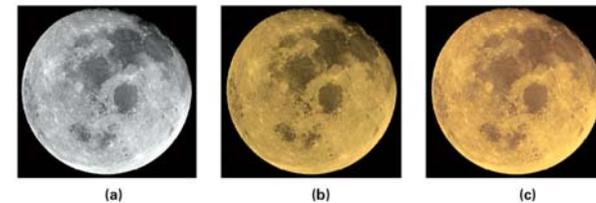


Figura 9.2 Tre fotografie della luna: (a) l'originale in bianco e nero; (b) dopo la colorazione; (c) dopo aver messo in risalto le parti più illuminate.

7-40

## Digitalizzare il suono

- Un oggetto produce suono vibrando all'interno di un mezzo come l'aria
  - Le vibrazioni si trasmettono nell'aria
  - Le onde di pressione sono emanate dall'oggetto e fanno vibrare i nostri timpani
  - L'intensità della pressione determina il volume
  - La *frequenza* (numero di onde al secondo) è l'altezza

7-41

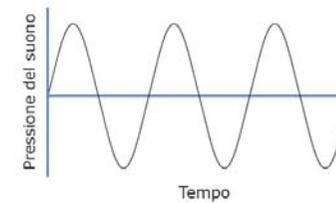


Figura 9.3 Un'onda sonora. L'asse orizzontale rappresenta il tempo, l'asse verticale la pressione del suono.

7-42

## Da analogico a digitale

- Per digitalizzare informazioni continue bisogna convertirle in bit
- È possibile esprimere con un numero binario la distanza dell'onda dall'asse (la quantità di pressione positiva o negativa)
- Quando dobbiamo eseguire le misure? Non possiamo registrare ogni punto dell'onda

7-43

## Campionamento

- Si prendono le misure a intervalli regolari
- Il numero di misurazioni al secondo è la *frequenza di campionamento*
  - maggiore è la frequenza, più accurata sarà la registrazione

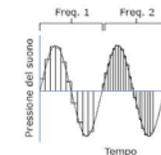


Figura 9.4 Due frequenze di campionamento: quella a destra è il doppio di quella a sinistra.

7-44

## Quale frequenza di campionamento?

- La frequenza di campionamento dovrebbe essere legata alla frequenza dell'onda
  - una frequenza troppo bassa potrebbe perdere dettagli che “si infilano” tra un campione e l'altro
  - *regola di Nyquist*: la frequenza di campionamento dev'essere almeno il doppio di quella dell'onda da registrare
    - dato che l'uomo può percepire suoni fino a 20.000 Hz, un campionamento di 40.000 Hz è sufficiente
    - la frequenza standard è 44.100 Hz

7-45

## ADC e DAC

- Il processo di digitalizzazione:
  - il suono è convertito dal microfono (*trasduttore*)
  - Il segnale entra in un *convertitore analogico-digitale* (ADC), che campiona l'onda a intervalli regolari e la passa alla memoria sotto forma di numeri binari
- Per riprodurre il suono, il processo è inverso
  - I numeri passano dalla memoria a un *convertitore digitale-analogico* (DAC), che ricrea l'onda elettrica più semplice che “passa” per tutti i punti rappresentati dal valore dei campioni
  - Il segnale entra poi in un altoparlante che lo converte in onda sonora

7-46

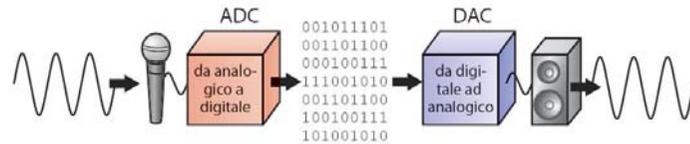


Figura 9.5 Uno schema della conversione da analogico a digitale e viceversa.

7-47

## Quanti bit per campione?

- Quanto dev'essere accurato un campione?
  - i bit devono rappresentare i valori sia positivi che negativi
  - più bit ci sono, più è accurato il campione
  - la rappresentazione digitale dei CD audio utilizza 16 bit (registra 65.536 livelli, la metà per i valori positivi e altrettanti per quelli negativi)

7-48

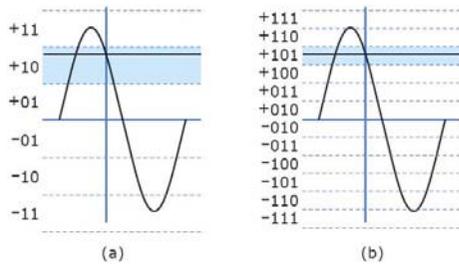


Figura 9.6 (a) Usando campioni a tre bit il valore letto sarà approssimato come +10. (b) Aggiungere un altro bit raddoppia l'accuratezza del campione.

7-49

## Vantaggi del suono digitale

- Possiamo eseguire delle elaborazioni
- Compressione MP3
  - un'applicazione è la compressione dell'audio digitale (riduzione del numero di bit necessari alla rappresentazione)
  - le frequenze che l'orecchio umano non può udire sono rimosse
  - un file MP3 in generale arriva a un fattore di compressione di 10:1
    - i requisiti sulla larghezza di banda sono inferiori. Si tratta di un formato popolare per la trasmissione via Internet
- Riprodurre una registrazione
  - i bit possono essere copiati senza perdere informazioni
  - l'originale e la copia sono esattamente uguali

7-50

## Immagini e video digitali

- Occorrerebbero 51 minuti per scaricare con un modem un'immagine a colori di 20 x 25 cm. digitalizzata a 300 pixel per pollice (dpi) senza alcuna compressione
- Com'è possibile vedere immagini a tutto schermo in pochi secondi navigando sul Web?
- Un tipico schermo di computer ha meno di 100 pixel per pollice
  - un'immagine digitalizzata a 100 dpi richiede poco più di un decimo della memoria
    - richiede sempre 5 minuti e mezzo per essere spedita
  - Soluzione: la compressione JPEG

7-51

## Compressione

- Cambiare la rappresentazione per usare un numero inferiore di bit per memorizzare o trasmettere informazioni
  - Esempio: i fax sono lunghe sequenze di 0 e 1 che codificano il contenuto di una pagina in bianco e nero. La codifica run length è utilizzata per specificare la lunghezza della prima sequenza di 0, seguita dalla sequenza di 1 ecc.
    - la compressione è *lossless*, cioè senza perdita – la rappresentazione originale può essere ricostruita perfettamente

7-52

## JPEG

- Utilizzata per le immagini fisse
- I nostri occhi non sono sensibili alle piccole variazioni di tonalità, ma sono sensibili alle piccole variazioni di luminosità
  - fornisce una descrizione meno accurata del colore
  - la compressione ottiene un rapporto 20:1 senza variazioni percepibili ai nostri occhi

7-53



Figura 9.7 Un dettaglio da un'immagine compressa mediante JPEG. (a) compressione 14:1; (b) compressione 140:1. Potete guardare le immagini originali all'indirizzo <http://www.aw.com/snyder>.

7-54

## Compressione MPEG

- La stessa idea alla base del JPEG, ma applicata ai filmati
- Come la compressione JPEG, è applicata a ogni frame
- Poiché in genere due immagini consecutive sono molto simili
  - la compressione MPEG memorizza solo “la differenza” tra fotogrammi consecutivi
  - ciò permette livelli di compressione elevatissimi

7-55

## Il riconoscimento ottico dei caratteri

- Leggere le targhe delle auto:
  - il computer deve capire dove si trova la targa nell'immagine
    - esamina gruppi di pixel cercando i bordi dove il colore cambia
    - cerca di identificare le *caratteristiche* peculiari
    - un *classificatore* le confronta con quelle delle lettere dell'alfabeto

7-56

## Tecnologia OCR

- Permette al computer di "leggere" caratteri stampati
  - applicazioni commerciali: le poste e le banche

7-57

## Ingannare i sensi: la realtà virtuale

- Creazione di un intero mondo alternativo
- Applica quest'idea a tutti i sensi e cerca di eliminare gli indizi che ci tengono legati alla realtà
- *Dispositivi aptici*
  - una tecnologia di input/output per il senso del tatto
  - un guanto aptico permette al computer di determinare la posizione delle nostre dita. Quando avviciniamo abbastanza le dita, il guanto le blocca dandoci la sensazione di aver afferrato un oggetto

7-58

## Il problema della latenza

- Uno dei problemi è che il sistema deve operare con velocità e precisione per sembrare naturale
- La *latenza* è il tempo impiegato dall'informazione per essere trasmessa
- Una latenza troppo lunga rovina l'illusione

7-59

## Il problema della banda passante

- La quantità di dati trasmessi per unità di tempo
- un aumento della banda passante significa una diminuzione della latenza

7-60

## I bit possono rappresentare diverse tipologie di informazione.

- Principio del mezzo universale:
  - per mezzo dei bit si può rappresentare ogni sorta di informazione discreta; i bit non hanno un significato intrinseco.
- I bit sono un mezzo universale
  - tutte le cose che possono essere rappresentate possono anche essere manipolate
- I bit non hanno preferenze
  - il significato dei bit deriva interamente dall'*interpretazione* che ne dà il computer attraverso il programma
- I bit non rappresentano necessariamente numeri
  - i bit possono essere interpretati come numeri, oppure no

7-61