

SUPSI

Introduzione

Fondamenti di Multimedia Processing

Tiziano Leidi

21.09.2018

Multimedia

Nella sua connotazione più generica, con multimedia si intende l'utilizzo e la combinazione di differenti forme di contenuti come:

- testo,
- suono,
- immagini,
- video,
- e contenuti interattivi.

Multimedia

Oggi, praticamente tutto ciò che è multimedia è diventato digitale. L'acquisizione, il trasporto, il processing e l'archiviazione di dati multimediali rappresentano sfide importanti per molti ambiti professionali.

Multimedia processing

Nel contesto di questa lezione, ci riferiremo principalmente al processing di segnali audio e video con applicazioni pratiche nei domini del riconoscimento vocale e della visione computerizzata.

Fondamenti di multimedia processing

L'obiettivo del corso è quello di fornire le conoscenze di base per cominciare a muoversi nell'immenso universo del multimedia processing dal punto di vista dello ingegnere e sviluppatore software.

Fondamenti di multimedia processing

- 15 settimane, 4 ore di lezione e laboratorio
- Contenuti teorici, esercitazioni, approfondimenti, ...

Docenti:

- Tiziano Leidi: tiziano.leidi@supsi.ch
- Diego Frei: diego.frei@supsi.ch
- Yari Mattei: yari.mattei@supsi.ch
- Loris Grossi: loris.grossi@supsi.ch

Contenuti

- Tecniche e formati per l'immagazzinamento, la codifica e la compressione di dati audio, immagini e video.
- Protocolli per il networking e lo streaming audio/video.
- Architetture software per il processing audio/video, frameworks e altri strumenti di sviluppo.
- Tecniche di audio processing, image processing e video processing.
- Introduzione alla visione computerizzata e alle tecnologie per il riconoscimento vocale.
- Programmazione ed elaborazione di flussi multimediali con processori multi-core e GPU.
- Introduzione alle tecniche di machine learning per l'analisi di flussi audio/video.

Particolarità della lezione

- Lezione con materiale didattico in parte in italiano, in parte inglese.
- Le lezioni verranno presentate in italiano.
- Parecchio spazio verrà lasciato alle attività di laboratorio.
- È una lezione che avete scelto voi, ... cercate di trasformarla in un vostro spazio di sperimentazione!

Risorse

- Slides delle lezioni.
- Eventuali articoli come approfondimenti.
- Tools e frameworks di sviluppo e relativa documentazione.
- Materiale delle esercitazioni.

Le risorse verranno messe a disposizione o referenziate da iCorsi:

Corsi SUPSI / DTI / Bachelor in Informatica / Fondamenti di Multimedia Processing

<https://www.icorsi.ch/course/view.php?id=6117>

Valutazione

- Presenza e partecipazione in classe.
- Risultati delle esercitazioni.
- Esame finale.

Sorgenti delle informazioni

Le informazioni contenute nelle slides hanno le seguenti origini:

- Scritte direttamente dal docente.
- Copiate (ed eventualmente rimaneggiate) da Wikipedia o altri siti internet.
- Copiate (ed eventualmente rimaneggiate) da libri, papers e articoli scientifici, documentazione di tools/frameworks.

SUPSI

Nozioni di base sul suono

Fondamenti di Multimedia Processing

Tiziano Leidi

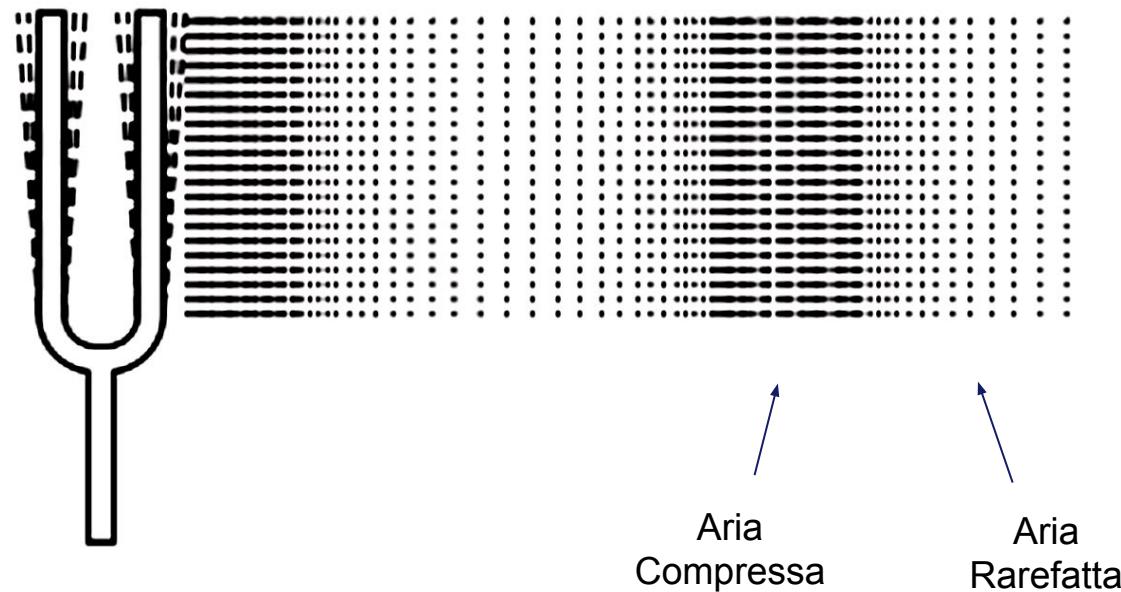
21.09.2018

Cos'è il suono

Il suono è una forma di energia. È la versione “aerea” della vibrazione.

Sulla terra, vibrazione e suono sono legati in maniera inscindibile.

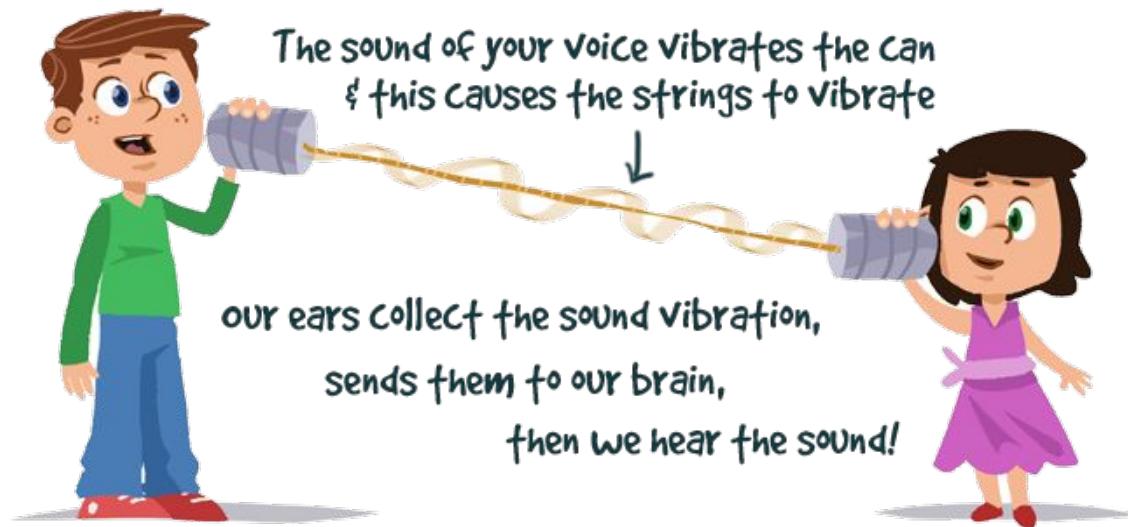
I suoni vengono generati da corpi in oscillazione, ad esempio a causa di urti, e trasmessi da mezzi elastici (principalmente fluidi).



Cos'è il suono

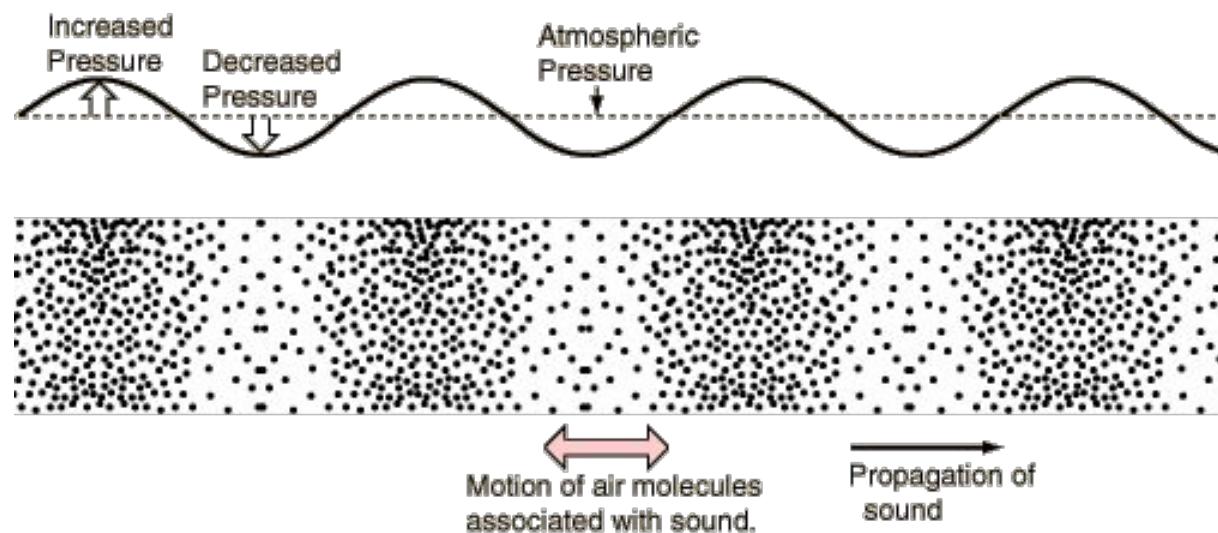
Sentire suoni è una delle capacità che fa parte dei 5 sensi di molti esseri viventi, inclusi gli esseri umani.

Le vibrazioni sonore si propagano nell'aria (o in altri mezzi elastici, come l'acqua), raggiungendo l'apparato uditivo dell'orecchio, che li percepisce e li interpreta.



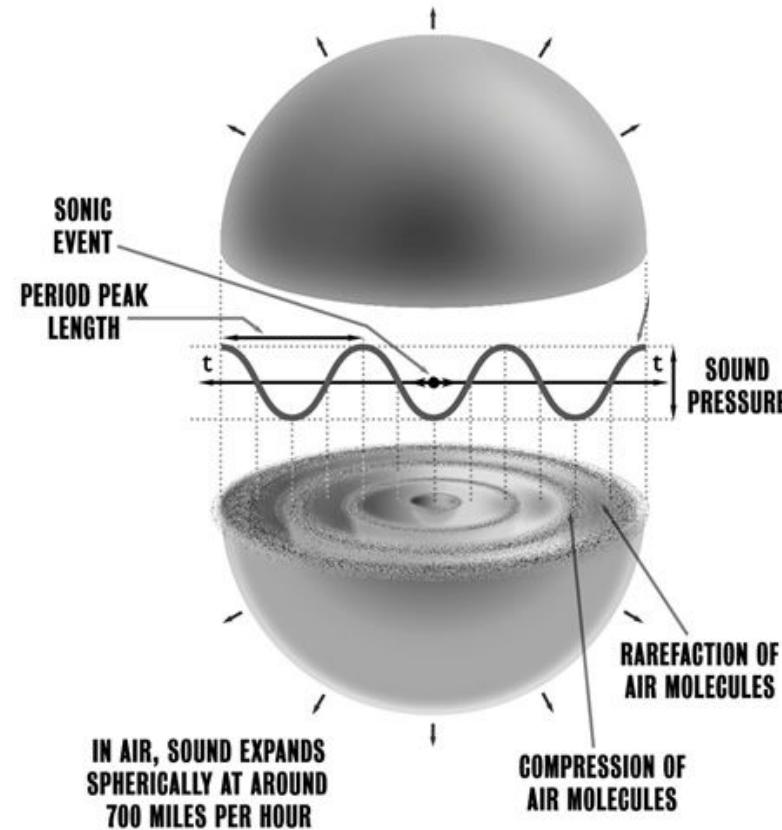
Propagazione dell'energia sonora

Le vibrazioni sonore vengono trasmesse dalle particelle d'aria (o del mezzo di propagazione in questione), che scontrandosi con quelle a loro prossime, generano delle “onde” di pressione.



Propagazione dell'energia sonora

La rappresentazione abituale del suono è quella di un'onda e si parla di onde sonore. Ma la trasmissione è differente da quella di un'onda del mare:



Velocità di propagazione

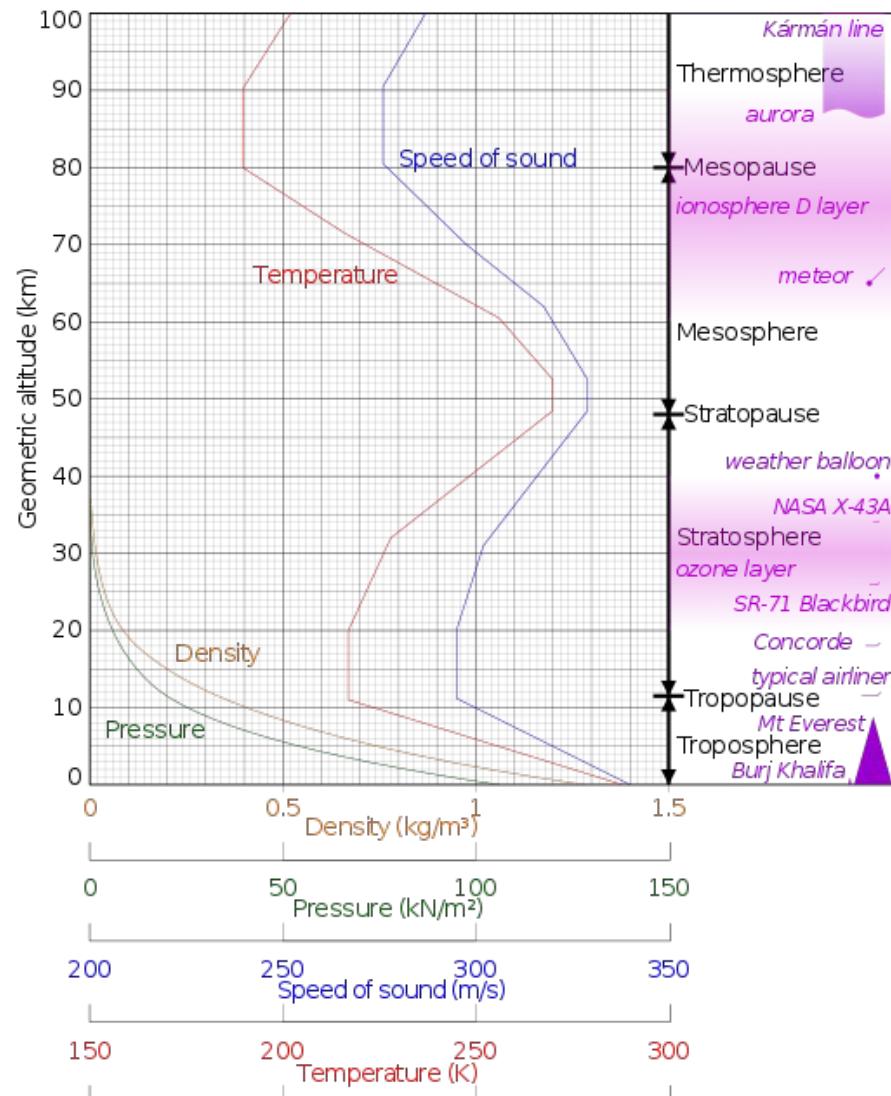
La velocità di propagazione del suono è determinata dal mezzo di trasmissione e dalla temperatura:

- In aria secca a 0°C la velocità è di 340.27 m/s, mentre a 20°C è di 343 m/s = 1235 km/h (mach 1).
- Nell'acqua, il suono viaggia 4.3 volte più veloce che nell'aria.
- Nel vuoto, ... non c'è suono.

Oggetti che si muovono a velocità maggiori di mach 1 viaggiano a velocità supersonica.



Velocità di propagazione



Velocità di propagazione

<https://www.khanacademy.org/science/physics/mechanical-waves-and-sound/sound-topic/v/speed-of-sound>

Energia sonora

L'unità di misura dell'energia sonora è il joule (J).

Il mezzo di propagazione (aria, acqua, ...) funge da accumulatore dell'energia, sia potenziale che cinetica.

$$W = W_{\text{potential}} + W_{\text{kinetic}} = \int_V \frac{p^2}{2\rho_0 c^2} dV + \int_V \frac{\rho v^2}{2} dV$$

V è il volume d'interesse

p è la pressione sonora

v è la velocità delle particelle

ρ_0 è la densità del mezzo in assenza di suono

ρ è la densità locale del mezzo

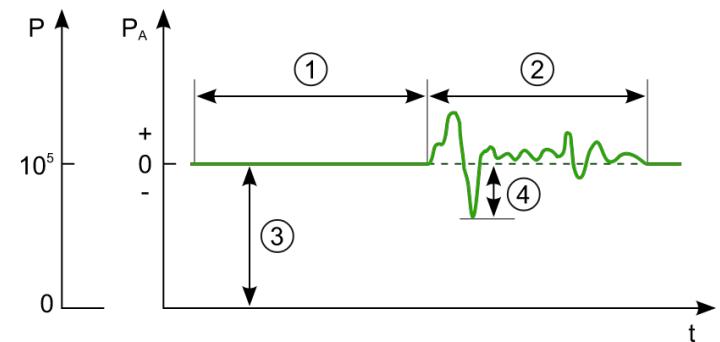
c è la velocità del suono



Pressione sonora

La pressione sonora, detta anche pressione acustica, è rappresentata dalla variazione di pressione esercitata dal suono rispetto alla pressione atmosferica ambientale.
L'unità di misura della pressione sonora è il pascal (Pa).

$$p_{\text{total}} = p_{\text{stat}} + p,$$



- | | |
|--------------------|----------------------------|
| p | è la pressione sonora |
| p_{total} | è la pressione totale |
| p_{stat} | è la pressione atmosferica |

- | | |
|----|-----------------------|
| 1. | silenzio |
| 2. | suono udibile |
| 3. | pressione atmosferica |
| 4. | pressione sonora |

Pressione sonora effettiva (RMS)

Esistono più varianti di rappresentazione della pressione sonora. Quella più utilizzata è la pressione sonora effettiva, comunemente chiamata pressione RMS (da Root Mean Square).

Si tratta di un valore medio della pressione istantanea calcolato su un intervallo di tempo definito.

$$P = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} p(t)^2 dt}$$

La media RMS viene anche detta media quadratica, un valore statistico utilizzato in vari ambiti, ad esempio della fisica e dell'elettronica.

Livello di pressione sonora (SPL)

A causa dell'ampio intervallo di percezione dell'orecchio umano, l'utilizzo del valore assoluto di pressione sonora risulta scomodo. Viene quindi utilizzato il livello di pressione sonora.

È un valore relativo calcolato come il logaritmo del rapporto fra una data pressione sonora e una pressione sonora di riferimento.

Come pressione di riferimento si usa $20 \mu\text{Pa}$ per la propagazione nell'aria e $1 \mu\text{Pa}$ in altri mezzi di propagazione. Si tratta del limite di capacità di udito dell'essere umano (che equivale più o meno al suono di una zanzara a 3 metri di distanza).

Decibels (dB) SPL

Il livello di pressione sonora si misura in decibels (dB) SPL:

$$L_p = 20 \log_{10} \left(\frac{p_{\text{rms}}}{p_{\text{ref}}} \right) \text{ dB}$$
$$L_p = \ln \left(\frac{p}{p_0} \right) \text{ Np} = 2 \log_{10} \left(\frac{p}{p_0} \right) \text{ B} = 20 \log_{10} \left(\frac{p}{p_0} \right) \text{ dB}$$

P_{rms}

P_{ref}

Np è il neper
B è il bel
dB è il decibel

Una pressione RMS di 1 Pa corrisponde ad un livello di 94 dB SPL.

Esempi

	Sound Pressure RMS	SPL
	Pa	dB
Auditory threshold at 1 kHz	2×10^{-5}	0
Breathing	6.32×10^{-5}	10
Normal Conversation	$2 \times 10^{-3} - 2 \times 10^{-2}$	40 – 60
Television at home	2×10^{-2}	approx. 60
Hearing Damage limit	$2 \times 10^{-2} - 2 \times 10^{-1}$	60 – 80
Jack hammer at 1 m	$2 \times 10^{-1} - 6.32 \times 10^{-1}$	80 – 90
Hearing damage (possible)	20	approx. 120
Vuvuzela horn at 1 m	20	120
Threshold of pain	63.2	130
Jet engine at 30 m	632	150
Stun grenades	6,000–20,000	170–180
Theoretical limit for undistorted sound	101,325	194
Shockwave (distorted sound waves)	>101,325	>194

Potenza sonora e livello di potenza sonora

Un altro valore spesso utilizzato è quello della potenza sonora (in inglese sound power). Rappresenta l'energia acustica emessa da una sorgente. È un valore assoluto.

A differenza della pressione, è indipendente dalla distanza e, ad esempio, dalle dimensioni della stanza. È la potenza totale emessa dalla sorgente in tutte le direzioni.

Il livello di potenza sonora (L_w) viene misurato in dB SWL. Viene ad esempio utilizzato per misurare la quantità di rumore prodotto dagli utensili di lavoro.



Livello di potenza sonora

$$L_w = 10 \log\left(\frac{W}{W_{ref}}\right) (dB)$$

L_w = sound power level

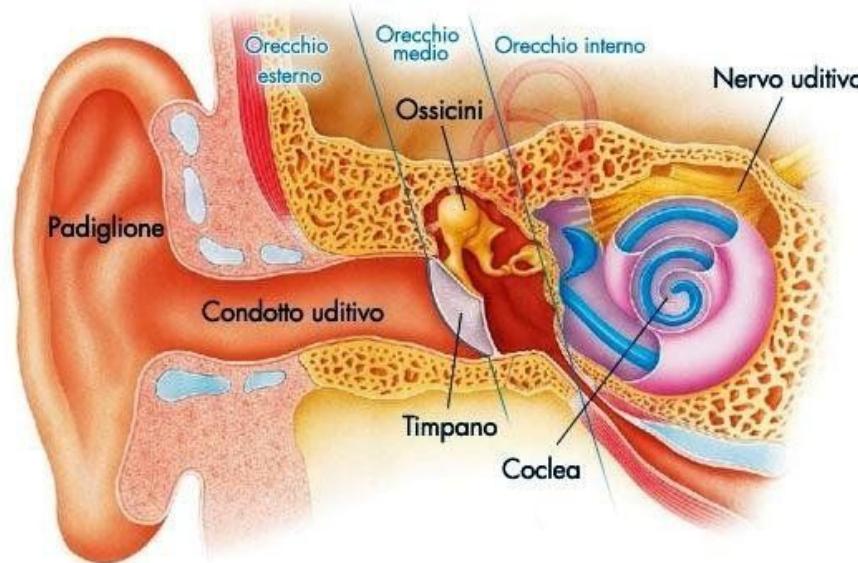
W = power of the sound

$W_{ref} = 10^{-12}$ W (a power level suited to human perception)

sound	L_w	
threshold of hearing	0	
background in sound studio	20	
whisper at 2m	40	quiet
conversation at 1m	50	
office environment	50	
restaurant/store	60	noisy
pneumatic drill	90	
punch press	110	very loud
truck horn	110	
propeller airplane	120	intolerable
propeller cargo airplane	140	
saturn V rocket	195	

Il sistema uditivo

Il sistema uditivo (più semplicemente udito) è il primo dei cinque organi di senso a svilupparsi nel feto e a favorire il contatto con l'ambiente esterno. Il sistema comprende sia gli organi sensoriali (orecchie e altri organi preposti alla percezione e alla traduzione del suono), sia le parti uditive del sistema sensoriale.



Il sistema uditivo

https://www.youtube.com/watch?v=TzdxCWli3_M

Psicoacustica

La psicoacustica è la scienza che studia la percezione del suono. Più specificamente, studia la risposta psicologica e fisiologica associata al suono (rumore, voce, musica, ...).

L'orecchio interno esegue una moltitudine di trattamenti del segnale quando converte le onde sonore in stimoli neurali. È quindi capace di filtrare e trasformare l'informazione.

Psicoacustica

Effetti psicoacustici intervengono, ad esempio:

- Nella localizzazione delle sorgenti sonore,
- nella riconoscimento dei timbri dei suoni e delle note,
- negli effetti di mascheramento.

In varie situazioni si riesce ad ingannare l'orecchio approfittando degli effetti psicoacustici. Si pensi ad esempio alla stereofonia o alla compressione psicoacustica usata ad esempio nei formati come l'MP3.

Tipologie di suoni

Un suono può essere causato da un singolo evento (ad esempio una percussione), detto transiente, o da una vibrazione ripetuta, come ad esempio quella di un diapason. Ci sono suoni periodici e aperiodici.

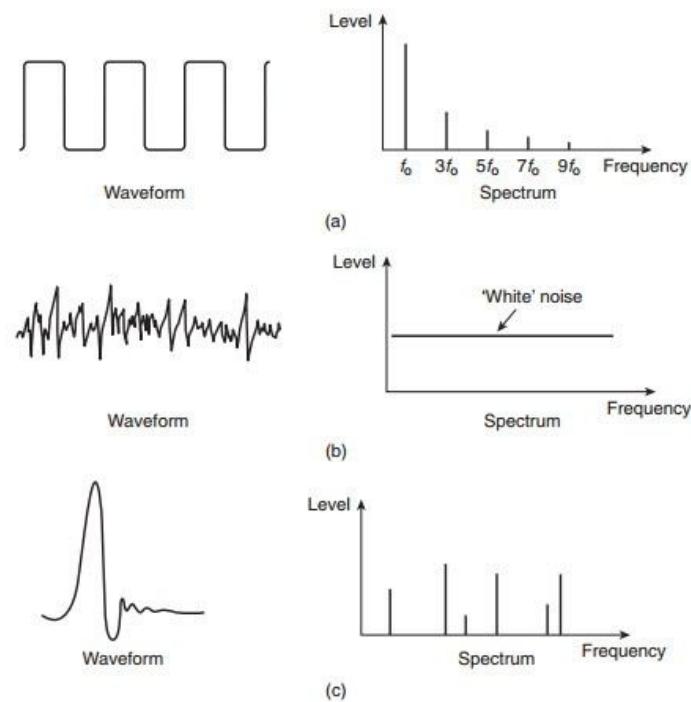


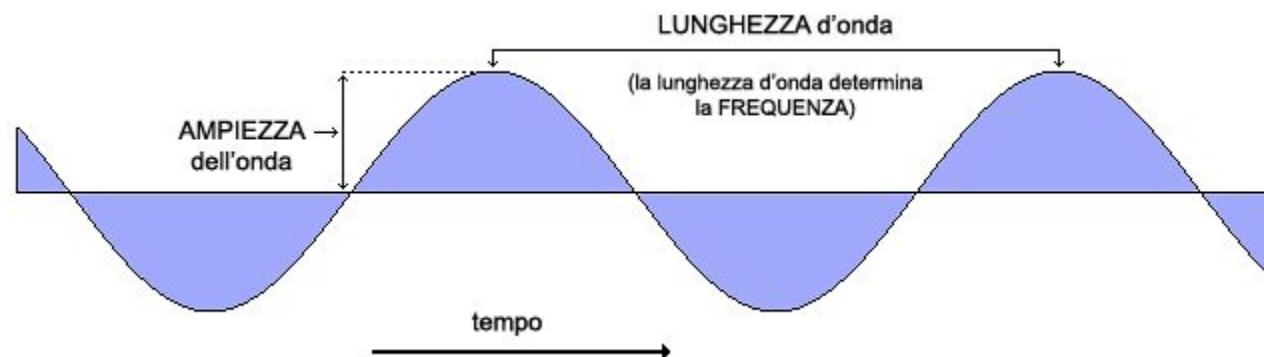
Figure 1.3: (a) A periodic signal repeats after a fixed time and has a simple spectrum consisting of fundamental plus harmonics. (b) An aperiodic signal such as noise does not repeat and has a continuous spectrum. (c) A transient contains an anharmonic spectrum.

Aampiezza, lunghezza d'onda e frequenza

Due delle caratteristiche principali delle onde sonore sono:

- ampiezza (amplitude): determina l'intensità del suono.
- lunghezza d'onda (wavelength) e frequenza (frequency): per i suoni periodici determinano l'altezza (pitch) del suono.

La frequenza è proporzionale alla lunghezza d'onda. Per una maggiore lunghezza d'onda si avrà una minore frequenza e viceversa.



Periodo e frequenza

La lunghezza d'onda di un suono ne determina il periodo: è il tempo necessario per completare un ciclo di vibrazione.

Più il periodo è lungo, più la frequenza (e di conseguenza la nota) sarà bassa. Più il periodo è breve, più la frequenza (e di conseguenza la nota) sarà alta.

Il periodo T viene misurato in secondi (s).

La frequenza è definita come $1/T$ ed è misurata in Hertz (Hz).

La frequenza misura il numero di oscillazioni.

440 Hz significa 440 oscillazioni al secondo.

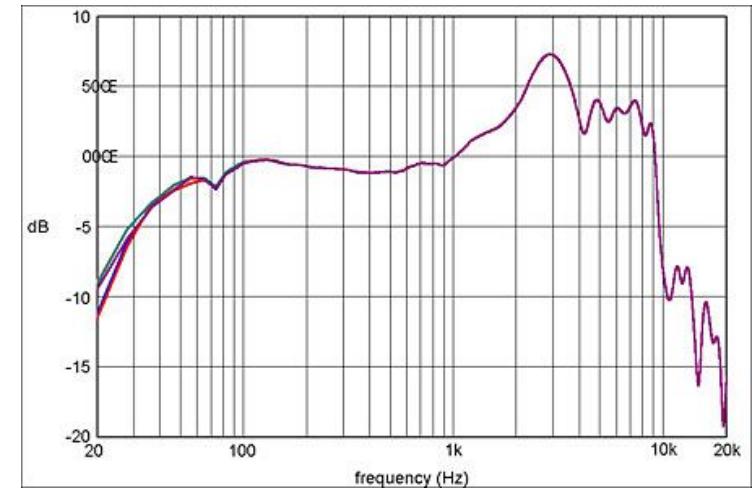
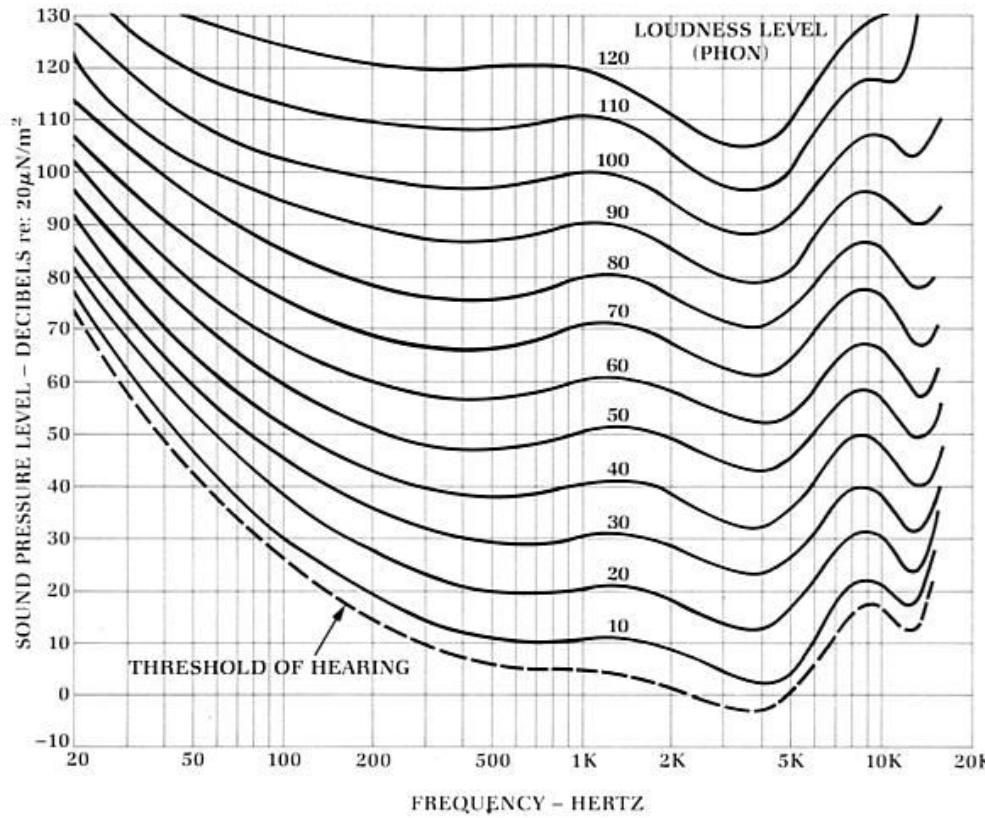
Frequenze udibili

L'orecchio umano è capace di percepire suoni nell'intervallo fra i 20 Hz e i 20 kHz. Il limite massimo decresce con l'invecchiamento. La maggior parte degli adulti non sente sopra ai 16 kHz.

Le frequenze sotto i 20 Hz possono essere percepite solo per vibrazione tramite il corpo, attraverso il suolo o per contatto con oggetti.

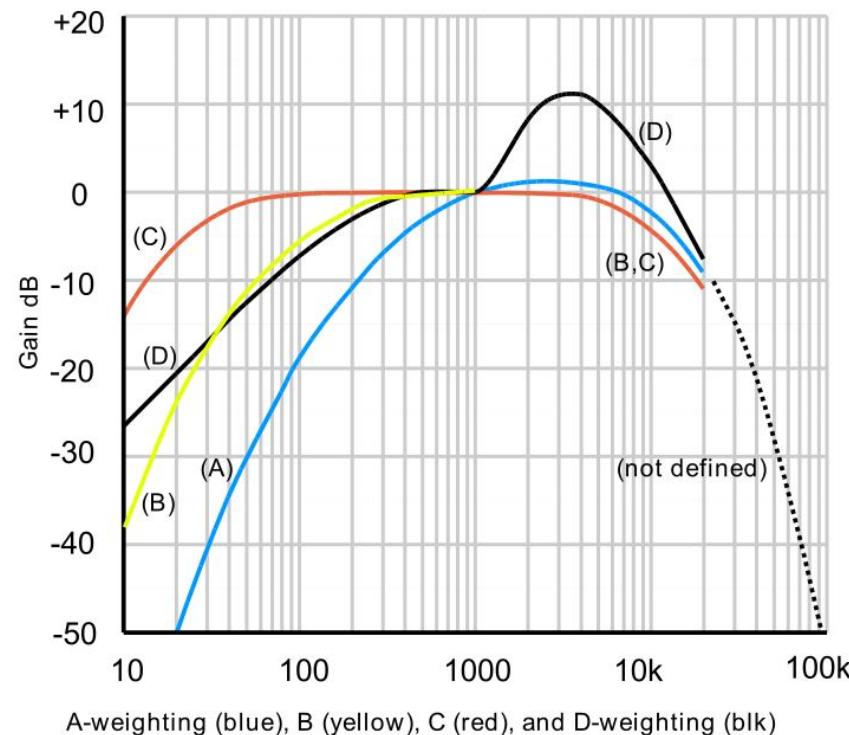
Frequenze udibili

L'orecchio umano non percepisce il livello sonoro in maniera uguale a tutte le frequenze. Fra i 3 e i 4 KHz è dove si sente meglio.

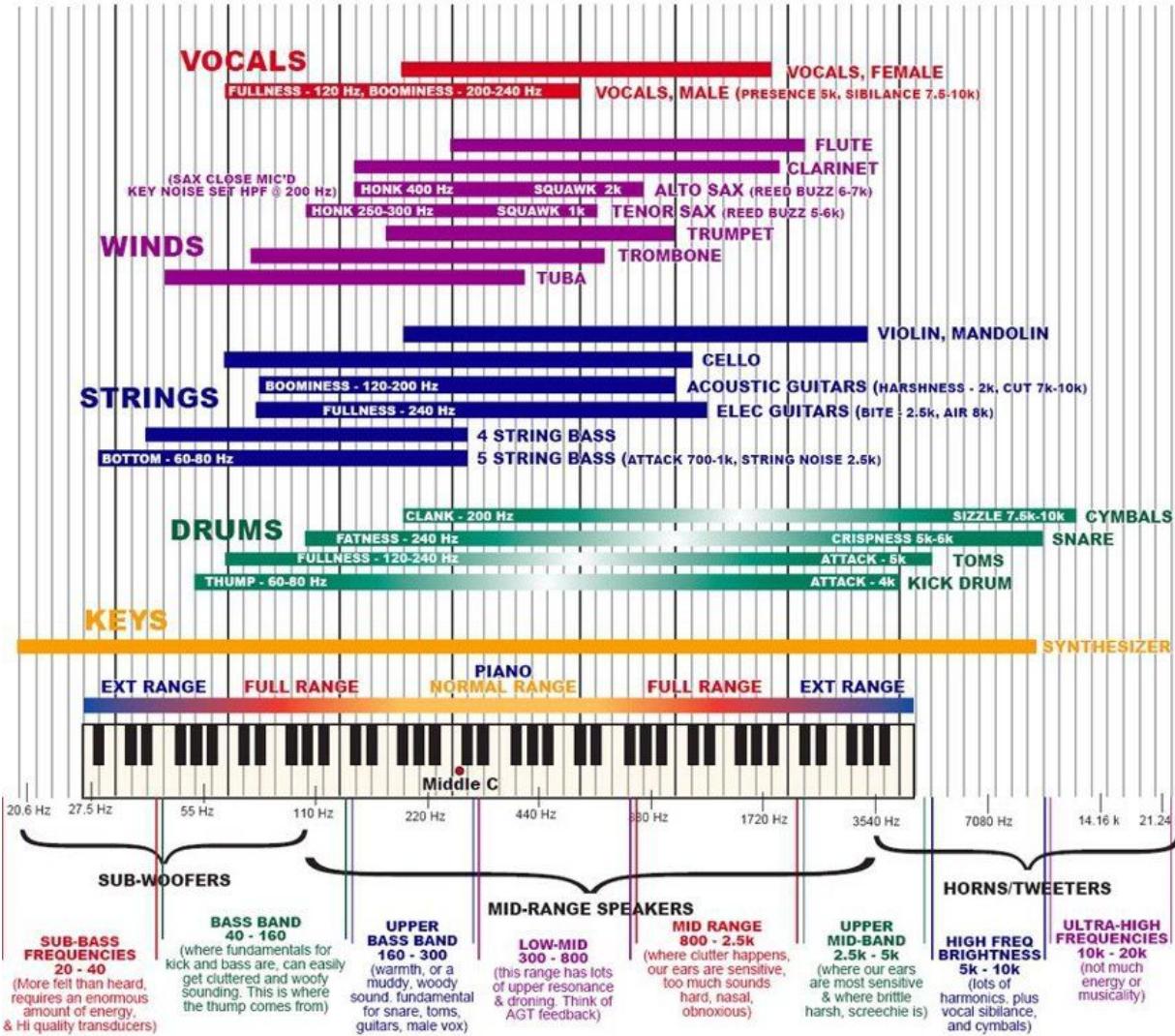


A-weighting (dB_A)

Per adattare le misure di livello sonoro alla differente percezione che l'essere umano ha delle frequenze vengono usate delle pesature. La più comune è l'A-weighting utilizzata nel dB_A.



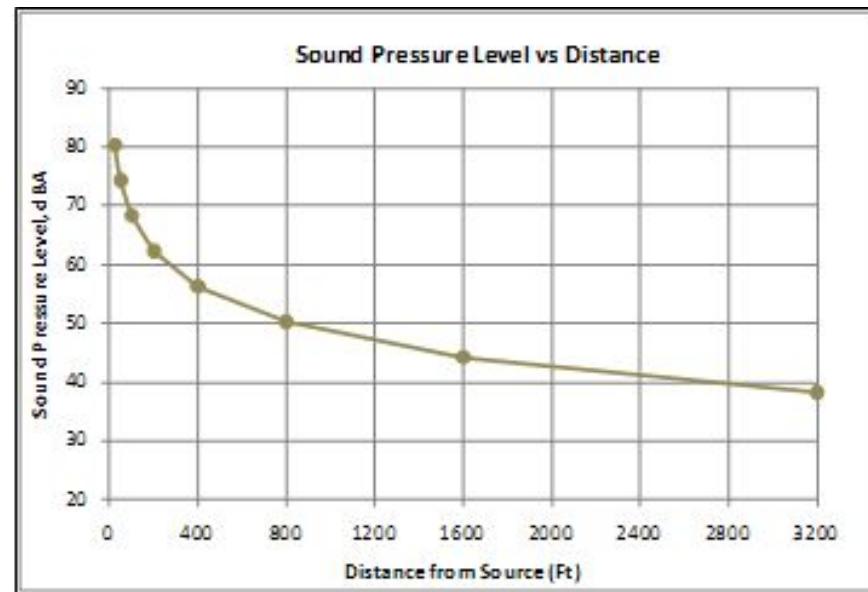
Frequenze udibili



Dispersione del suono

Come ogni forma di energia, anche l'energia sonora si disperde nell'ambiente. Il fattore di dispersione è proporzionale alla distanza.

Source Sound Power Level (LwA) = 106 dBA	
Distance, Feet	Sound Pressure Level, dBA
25	80
50	74
100	68
200	62
400	56
800	50
1600	44
3200	38

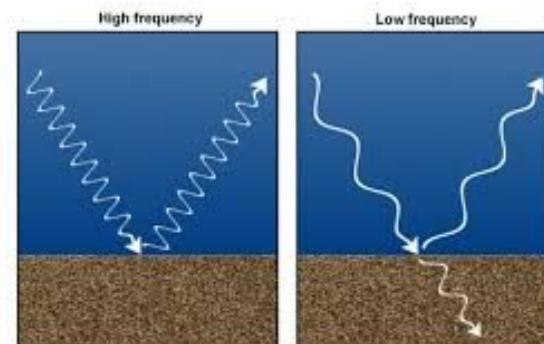


Propagazione delle frequenze sonore

Le onde sonore non si propagano in maniera identica per tutte le frequenze.

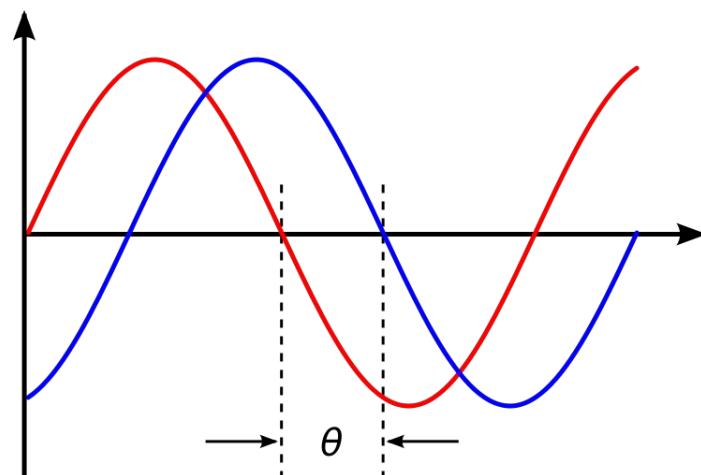
Le frequenze acute contengono poca energia e quindi si disperdono e vengono assorbite più facilmente dagli oggetti nell'ambiente.

Anche le riflessioni contro le barriere (ad esempio i muri) dipendono dalle frequenze. Le frequenze alte vengono più facilmente riflesse, mentre quelle basse riescono ad attraversare le barriere.



Fase

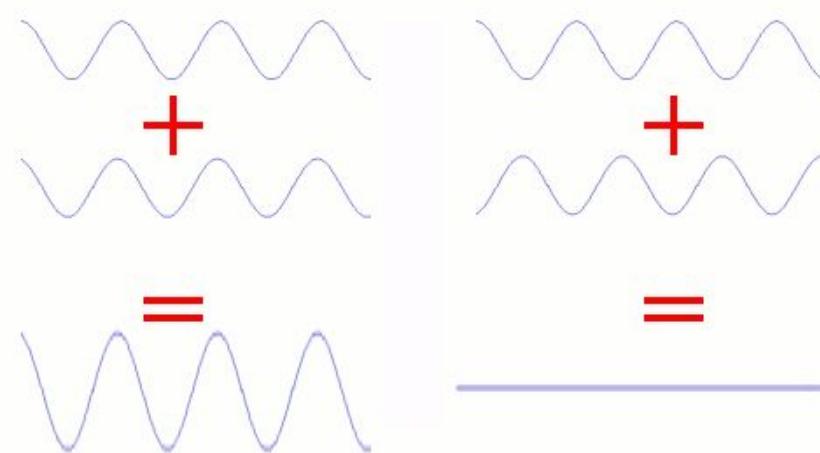
Un'altra proprietà importante delle onde sonore è la fase. Varia a dipendenza dell'istante in cui l'onda comincia la propria oscillazione. Onde sonore con la stessa frequenza possono avere una differenza di fase: è la distanza tra due punti al medesimo frangente di oscillazione delle due onde.



Fase

Le differenze di fase sono inudibili, ma possono avere forte effetto quando i suoni vengono combinati.

Ad esempio, se si somma un'onda sinusoidale con la sua esatta opposta (cioè un'onda a 180° di fase), si ottiene silenzio.



Timbro

Il timbro di un suono ne determina il carattere sonoro. Strumenti diversi producono suoni di timbro differente. Il timbro viene anche chiamato colore del suono.

Il suono con il timbro più semplice è quello prodotto da un sintetizzatore di sinusoide:

$$y(t) = A \sin(2\pi ft + \varphi) = A \sin(\omega t + \varphi)$$

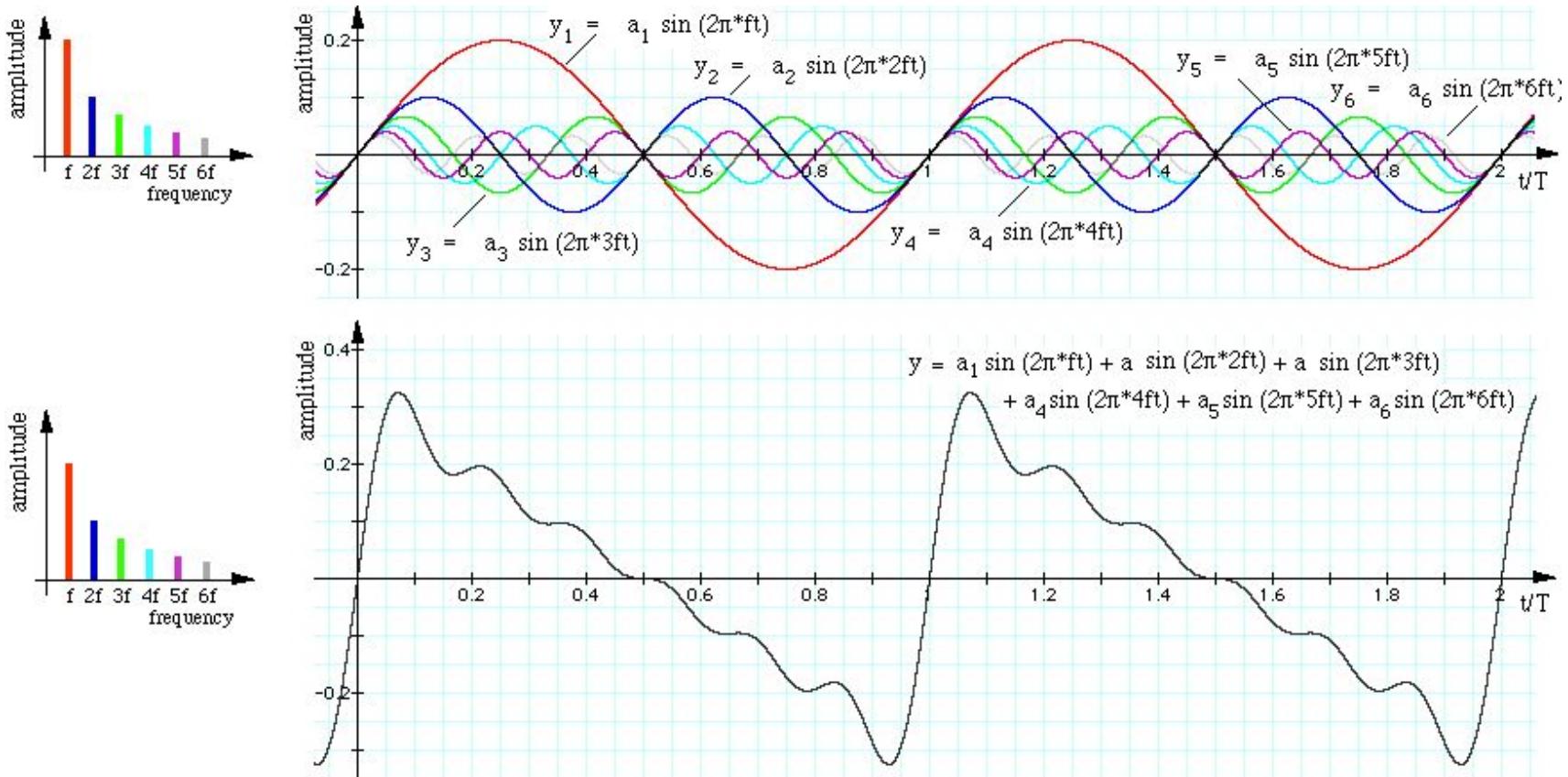
A:	ampiezza
f:	frequenza
$\omega = 2\pi f$:	frequenza angolare
phi:	fase

Quello più complesso è quello del rumore bianco.

La percezione del timbro è determinata da due caratteristiche fisiche: lo spettro e l'inviluppo.

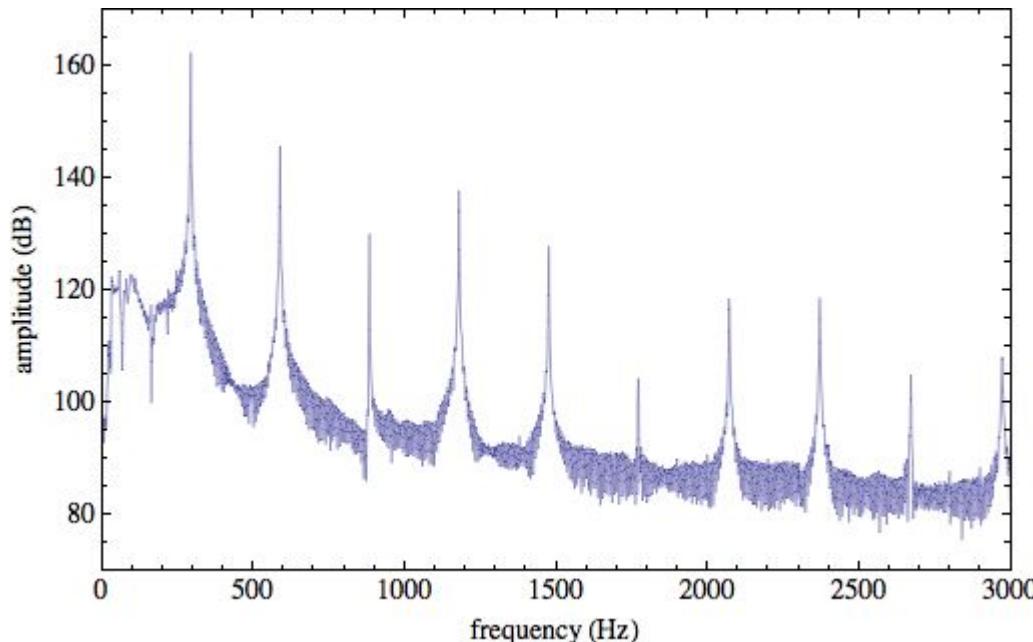
Spettro sonoro (delle frequenze)

Ogni suono può essere modellato come combinazione di onde sinusoidali a differenti frequenze.



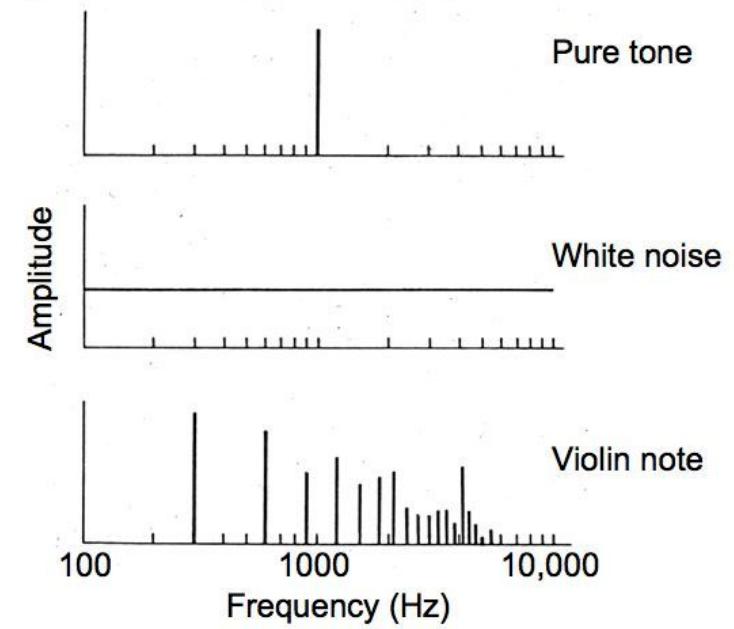
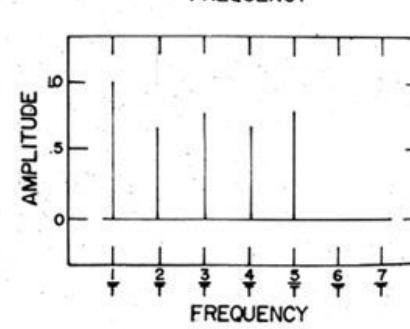
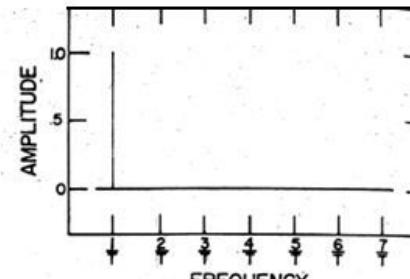
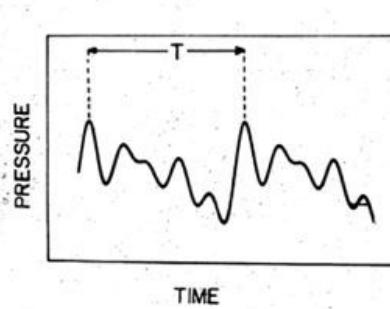
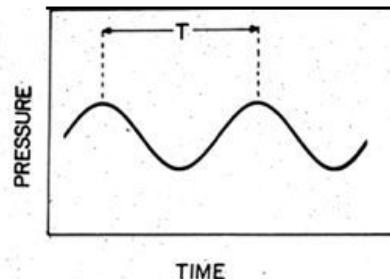
Spettro sonoro

Se un suono contiene frequenze che sono multipli di una frequenza “fondamentale”, l’unione di questi suoni viene percepita come un unico suono la cui frequenza è quella fondamentale e il cui carattere è determinato dalle frequenze e ampiezze coinvolte.

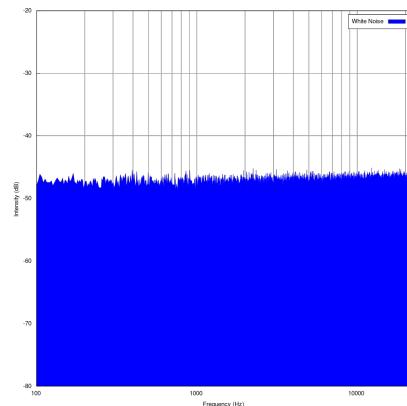


Spettro sonoro

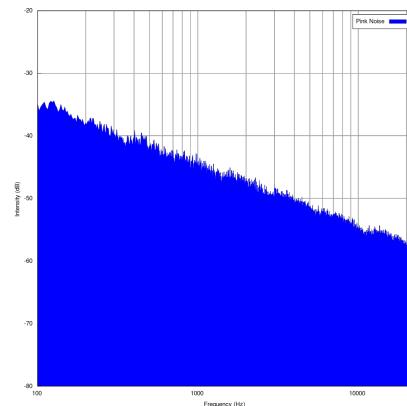
Per un tono puro (una sinusoide) il grafico è formato da una semplice linea verticale, per un suono musicale da una serie di linee in corrispondenza alle frequenze fondamentali e alle armoniche, per un rumore lo spettro è costituito da una banda di frequenze.



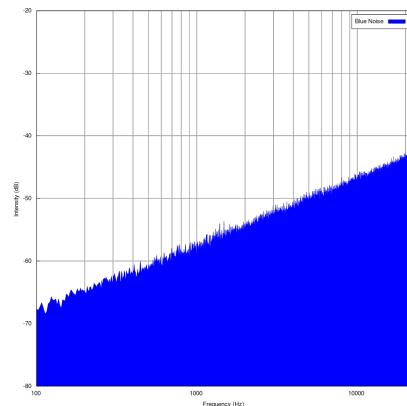
Spettro del rumore



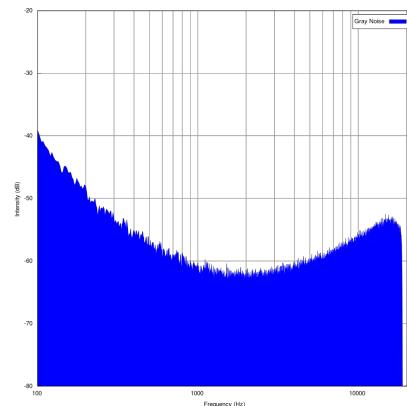
Rumore bianco



Rumore rosa



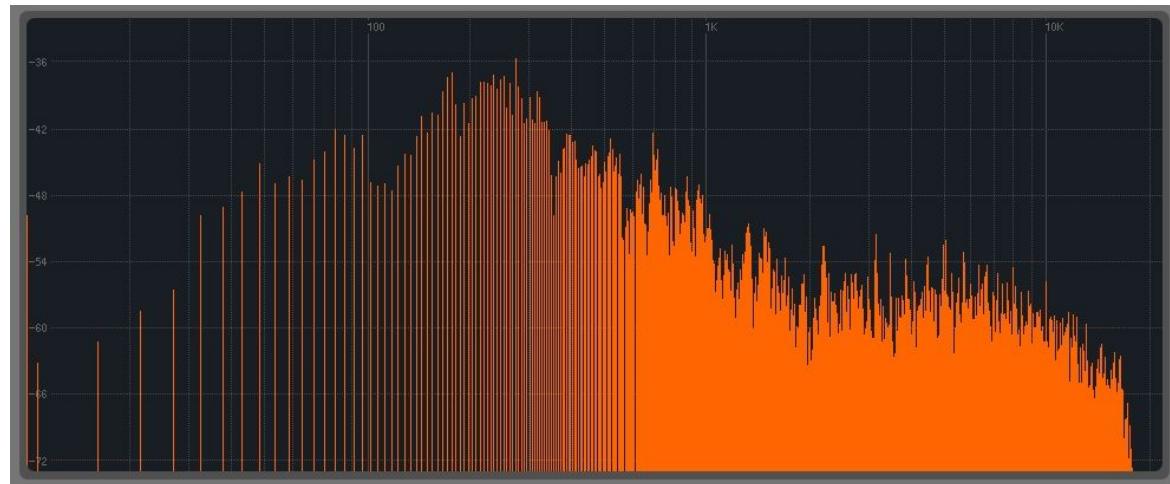
Rumore blu



Rumore grigio

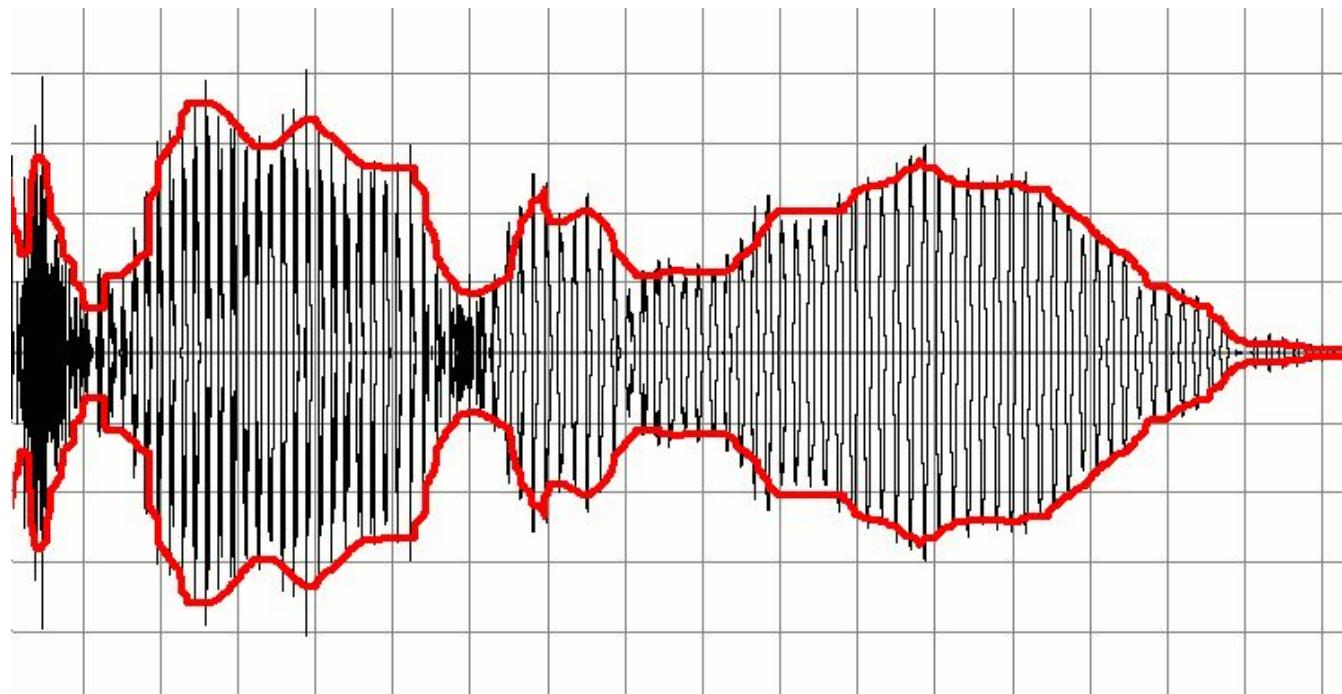
Spettro sonoro

Lo spettro sonoro può essere visualizzato e utilizzato nell'analisi di un suono o rumore: vi si riportano i livelli sonori delle singole sinusoidi.



Inviluppo

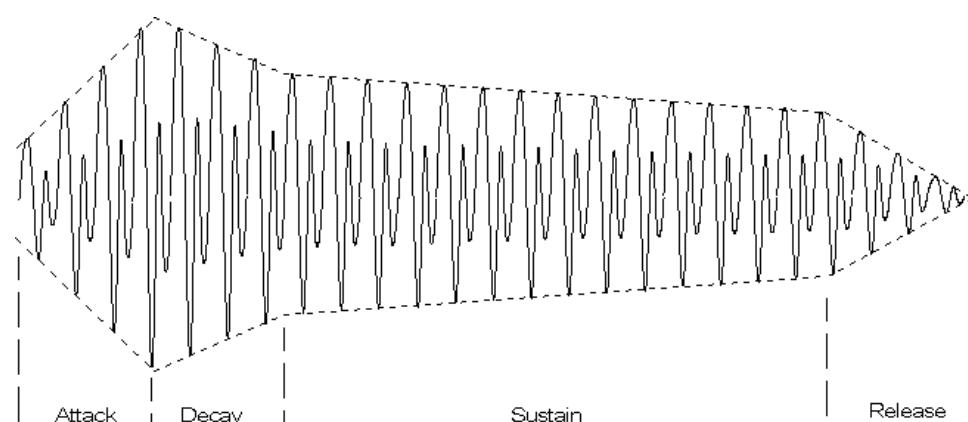
L'inviluppo è l'andamento dell'ampiezza di un suono nel tempo.
Ogni timbro sonoro ha un suo particolare inviluppo.



Inviluppo

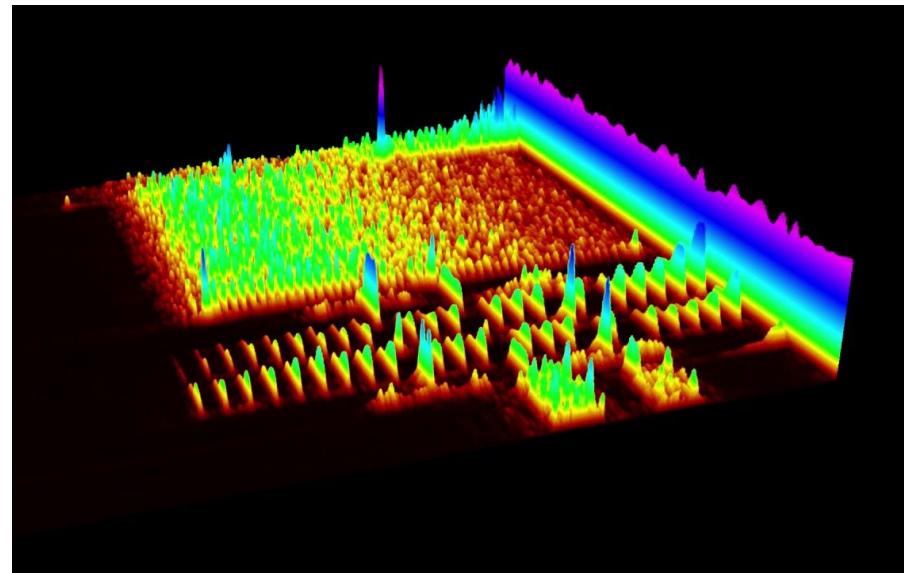
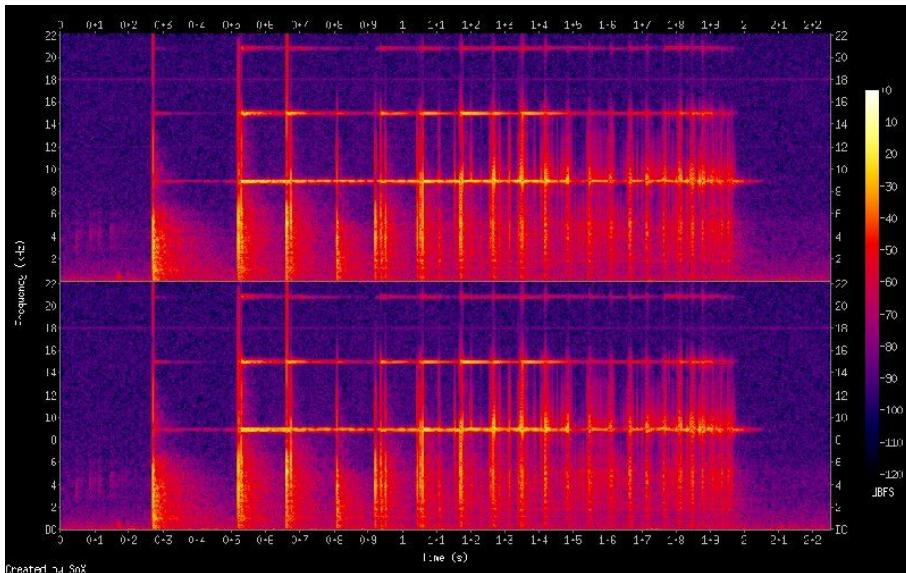
L'inviluppo di un suono è solitamente composto da 4 fasi, dal momento in cui viene originato, al momento in cui si estingue:

- Attacco (Attack): l'ampiezza raggiunge, a partire da zero, il suo valore massimo.
- Decadimento (Decay): dopo l'attacco, parte dell'energia iniziale viene persa e l'ampiezza diminuisce.
- Sostegno (Sustain): l'ampiezza mantiene un livello quasi costante per un certo tempo.
- Rilascio (Release): l'ampiezza decresce fine ad estinguersi completamente.



Spettrogramma

Lo spettrogramma è la rappresentazione grafica dell'intensità di un suono in funzione sia del tempo, che della frequenza. È lo strumento migliore per osservare il timbro di un suono. Ne esistono due varianti: bidimensionale e tridimensionale.



Direzione e distanza del suono

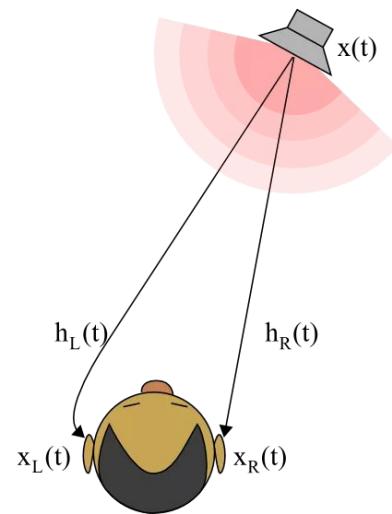
Oltre a riconoscere l'intensità e l'altezza dei suoni, il sistema uditivo degli esseri umani è capace di distinguerne la direzione di provenienza.

Questa capacità viene chiamata ascolto binaurale (binaural hearing).

Inoltre, l'orecchio umano è capace di stimare la distanza della sorgente sonora.

Direzione del suono

Sul piano orizzontale, la direzione viene identificata soprattutto sfruttando la differenza tra l'istante di arrivo del suono ad un orecchio, rispetto a quell'altro. Per le frequenze sotto gli 800 Hz, per cui la metà della lunghezza d'onda è maggiore della distanza fra le orecchie (21.5 cm), è possibile un'identificazione inequivocabile basata sulle differenze di fase.



Direzione del suono

Da 200 Hz in su, viene inoltre sfruttata la leggera differenza di livello sonoro, causata dall'effetto di mascheramento della testa. Questa differenza non è udibile sotto i 200 Hz.

A partire da 1600 Hz in su, la lunghezza d'onda è minore della distanza fra le orecchie e quindi le differenze di fase non possono più essere sfruttate. Viene utilizzata praticamente solo la differenza di livello sonoro, che fortunatamente è più marcata per le alte frequenze.

Per le frequenze alte, in ambienti riverberanti alcune informazioni di direzione possono venir estrapolate anche dalle riflessioni iniziali del suono sulle pareti.

Direzione del suono

Per quanto concerne le frequenze basse sotto gli 80Hz, l'orecchio non ha grandi capacità di riconoscere la direzione di provenienza. Le differenze di fase sono troppo piccole per venire riconosciute.

Inoltre, ci sono casi in cui l'orecchio può entrare in confusione (ad esempio suono proveniente frontalmente o da dietro). In queste situazioni, il cervello attiva comportamenti istintivi, in cui viene ruotata leggermente la testa lateralmente per permettere l'identificazione della direzione.

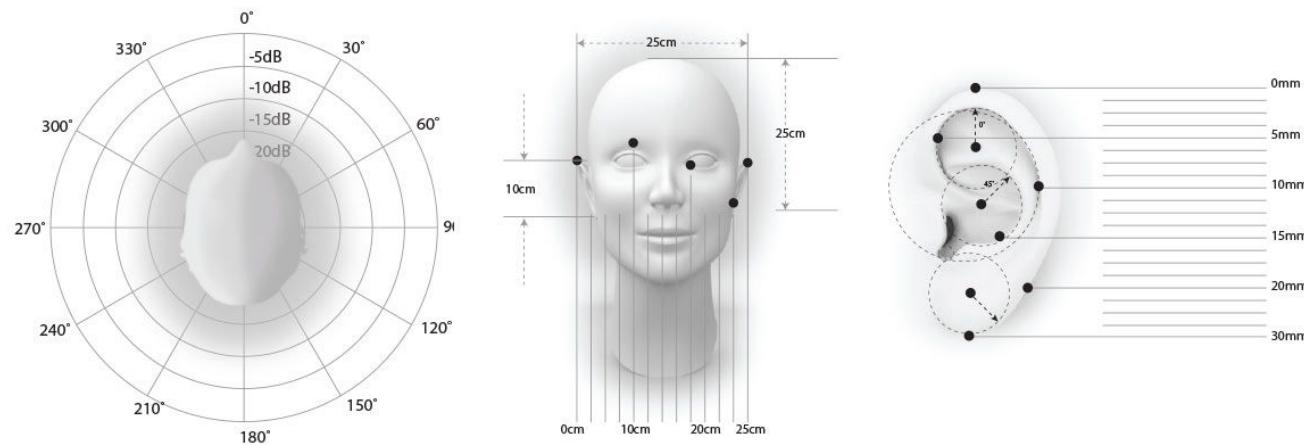
Head related transfer function (HRTF)

L'ascolto binaurale non è l'unica capacità sfruttata per l'identificazione della direzione del suono. Ad esempio per distinguere suoni provenienti da sopra rispetto a suoni provenienti dal basso, viene sfruttata la cosiddetta head related transfer function (HRTF).

Quando il suono si scontra con l'ascoltatore, la dimensione e forma della testa, delle orecchie, del canale delle orecchie, la densità della testa, dimensione e forma delle cavità nasali e orali, modificano il suono e quindi la percezione ottenuta, intensificandone alcune frequenze e attenuandone altre.

Head related transfer function (HRTF)

In generale, l'HRTF intensifica le frequenze fra i 2 - 5 kHz con una risonanza primaria (il suono viene rinforzato a causa delle riflessioni) di +17 dB at 2,700 Hz. Ma la risposta di frequenza è più complessa e ha impatto su spettro più ampio, variando in maniera importante fra persona e persona.



Distanza del suono

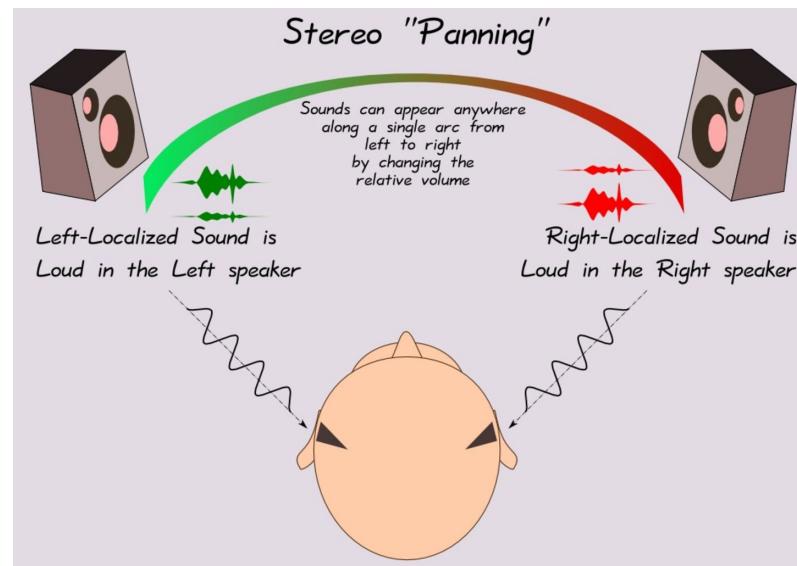
Per la stima della distanza, vengono soprattutto sfruttati i riverberi (prodotti dall'ambiente), in particolare l'informazione di predelay (il tempo che intercorre fra l'arrivo del suono diretto e le prime riflessioni).

A contribuire è anche lo spettro di frequenze del suono d'arrivo. Le frequenze acute contengono poca energia e quindi si disperdonano più facilmente. Vengono anche più facilmente filtrate da eventuali ostacoli.

Stereofonia

L'ascolto stereofonico è un inganno psicoacustico in cui vengono utilizzate due sorgenti sonore (correttamente posizionate), con una tecnica di riproduzione in cui i medesimi contenuti sonori vengono riprodotti con differenti ampiezze.

Quest'inganno permette di posizionare sorgenti sonore virtuali nello spazio compreso fra i due altoparlanti.



SUPSI

Audio analogico vs. audio digitale

Fondamenti di Multimedia Processing

Tiziano Leidi

28.09.2018

Audio analogico vs. audio digitale

Oggi, praticamente tutta l'industria audio (sia professionale che consumer) è passata al digitale.

La transizione è in atto da decine di anni, ... ma l'evoluzione non è ancora finita.

I vantaggi dell'audio digitale sono innumerevoli, in particolare in termini di qualità sonora che si può ottenere e mantenere, sia all'interno della catena di produzione sonora, sia per l'immagazzinamento e distribuzione dei dati di lavorazione e dei prodotti finiti.

Audio analogico vs. audio digitale

Le soluzioni audio analogiche non sono però completamente scomparse.

Alcune parti della sequenza di produzione (soprattutto quelle vicine ai microfoni e agli altoparlanti) mantengono, per necessità o per vantaggio, la loro natura analogica.

Audio analogico - cenni storici

La tecnologia audio analogica ha cominciato a svilupparsi (principalmente per la registrazione sonora) nella seconda metà del 19esimo secolo con l'invenzione di sistemi meccanici come il fonautografo e il fonografo.

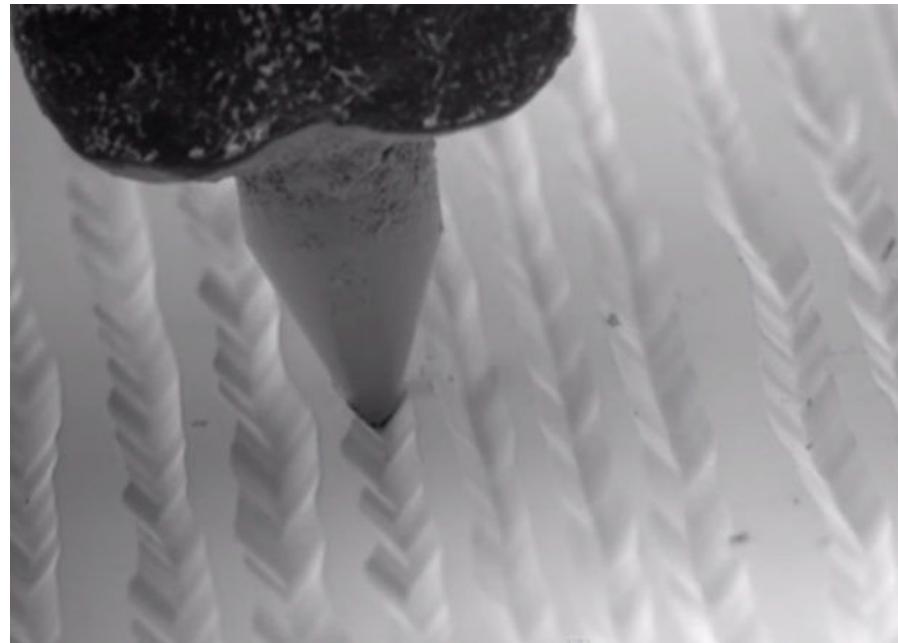


Attorno agli anni '20 sono state poi introdotte le prime soluzioni elettroniche, ad esempio per la registrazione con microfoni e con nastri.



Audio analogico

I metodi di registrazione audio analogica immagazzinano il segnale audio direttamente all'interno (o sulla superficie) dei supporti di registrazione, modificandone la struttura fisica.



Audio analogico

Le soluzioni elettroniche, si avvalgono di:

- trasduttori in grado di convertire i segnali sonori in segnali elettrici (microfoni)
- mezzi per la memorizzazione dei segnali elettrici (ad esempio supporti magnetici)
- trasduttori in grado di convertire i segnali elettrici in suoni (altoparlanti)

L'elettronica ha anche permesso la realizzazione di sintetizzatori sonori programmabili in grado di generare onde acustiche di diversa intensità, frequenza e timbro.

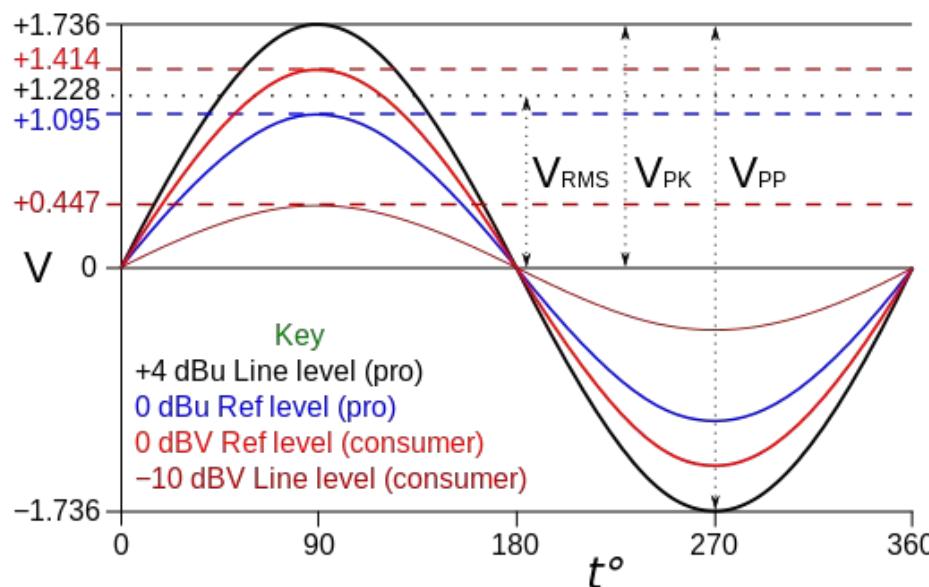
Audio analogico

Un segnale audio analogico è un segnale continuo, per cui la caratteristica variabile nel tempo, che nel caso dell'audio è solitamente il voltaggio istantaneo, rappresenta una caratteristica fisica anch'essa variabile nel tempo, come ad esempio la pressione delle onde sonore.



Audio analogico

Vengono utilizzati vari livelli di segnale elettrico per rappresentare i segnali audio. Grossa differenza esiste fra i segnali amplificati e quelli “low-power”. Inoltre, per i segnali “low-power” esistono diverse norme in ambito professionale e consumer.



V_{PK} : Picco

V_{PP} : Picco a Picco

V_{RMS} : Root Mean Square

dBV e dBu

dBV e dBu sono unità di misura usate per le intensità di voltaggio. Come per i dB SPL si tratta di valori relativi a specifici valori di riferimento.

dBV: voltaggio RMS relativo a 1 V

dBu: voltaggio RMS relativo a $\sqrt{0.6}$ V ≈ 0.7746 V ≈ -2.218 dBV

Range dinamico

Il range dinamico descrive il rapporto di ampiezza fra l'onda sinusoidale più forte possibile (non distorta) e il rumore di fondo (rumore rms) del sistema in uso.

Il range dinamico viene spesso confuso con il signal-to-noise ratio (SNR), che è la differenza fra il livello medio del segnale e il livello medio del rumore di fondo.

Come abbiamo visto, il range dinamico dell'orecchio umano è attorno ai 140 dB, variabile a dipendenza della frequenza. La fascia tra i 120 dB e i 140 dB è detta soglia del dolore.

Il range dinamico della musica in una sala concerti è tipicamente di 80 dB, mentre quello di un discorso è tipicamente attorno ai 40 dB.

Range dinamico di segnali analogici

Nei segnali audio analogici, il range dinamico dipende dal voltaggio massimo concesso e dal livello del rumore di fondo.

Ad esempio con un massimo di 5 V (rms) e con un rumore di fondo di 10 μ V (rms) il range dinamico è di 500000:1, quindi 114 dB.

Vari processi contribuiscono al rumore di fondo di un sistema analogico. Può provenire dal microfono, dal preamplificatore, dai cavi, etc.

Livelli e headroom

I livelli di voltaggio specificati dagli standards vengono detti livelli nominali. Sono livelli per cui gli apparecchi sono stati disegnati in modo da lavorare in maniera efficiente, sia in termini di range dinamico che di headroom disponibili.

I circuiti elettrici degli apparecchi limitano il range dinamico. A un lato vi è il segnale massimo producibile senza distorsione, all'altro lato il rumore interno prodotto dalle componenti elettroniche.

L'headroom è la differenza fra il livello nominale e il livello massimo, oltre il quale il segnale viene compromesso dalla distorsione.

VU meters

I VU meters vengono utilizzati per rappresentare i livelli di intensità sonora negli apparecchi come i mixer. L'unità di misura è il Volume Unit (VU).

In situazione normali, 0 VU permette un headroom di almeno 18 dB senza distorsione significativa.

Solitamente, negli apparecchi professionali $0 \text{ VU} = +4 \text{ dBu}$, mentre negli apparecchi consumer $0 \text{ VU} = -10 \text{ dBV}$.

Trasduttori

Un trasduttore (transducer) è una device elettronica che converte energia da una forma all'altra.

Nell'audio vengono principalmente utilizzati:

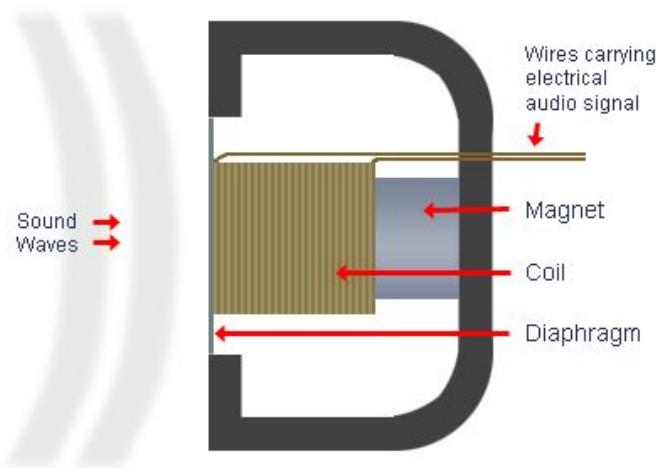
- Microfono (microphone), ovvero un trasduttore elettroacustico in input che converte un'onda sonora in un segnale elettrico.
- Altoparlante (loudspeaker), un trasduttore elettroacustico in output che esegue l'operazione opposta del microfono.

Il principio base del funzionamento di microfoni e altoparlanti è simile. Avete già provato a collegare un paio di cuffie all'ingresso di un convertitore analogico-digitale per utilizzarle come microfono?

Microfoni

Esistono più tipi di microfoni, costruiti sfruttando differenti tecnologie per l'acquisizione di segnali sonori.

L'esatto meccanismo di funzionamento varia a seconda del tipo di tecnologia utilizzata.



Avremo occasione di approfondire le conoscenze sui microfoni nel corso delle prossime lezioni.

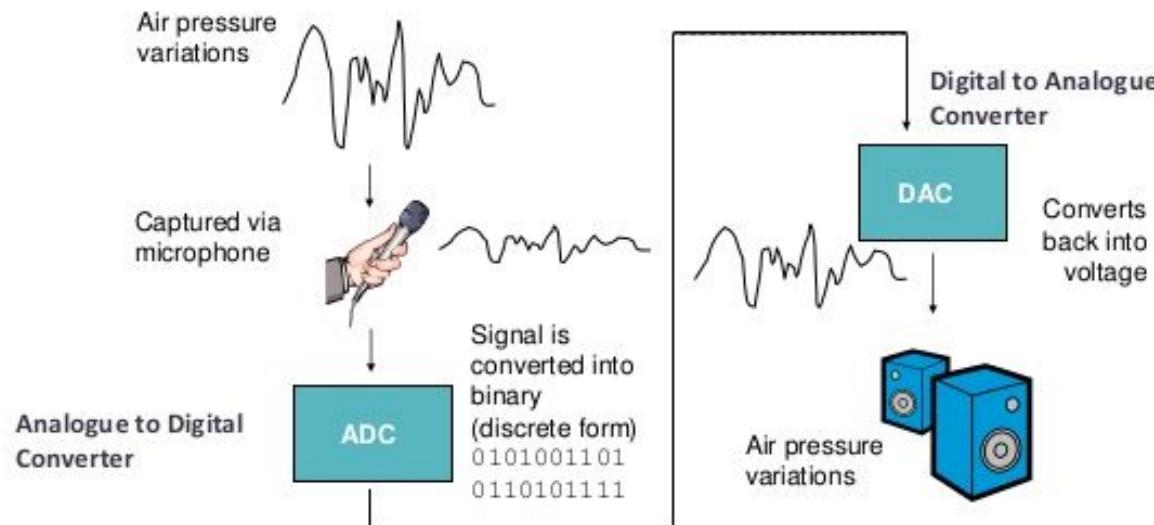
Altoparlanti

Anche le tipologie di altoparlanti sono molteplici. Anche nel caso degli altoparlanti, l'esatto meccanismo di funzionamento varia a dipendenza del tipo di tecnologia utilizzata.

<https://www.youtube.com/watch?v=ln7RMmtb4gw>

Audio digitale - cenni storici

La tecnologia per l'audio digitale ha cominciato a svilupparsi negli anni '70 ed '80, principalmente con l'avvento dei PC. A partire dagli anni '90 ha poi rimpiazzato buona parte della tecnologia audio analogica per la registrazione sonora, ma anche per la produzione sonora, l'audio engineering e le telecomunicazioni.



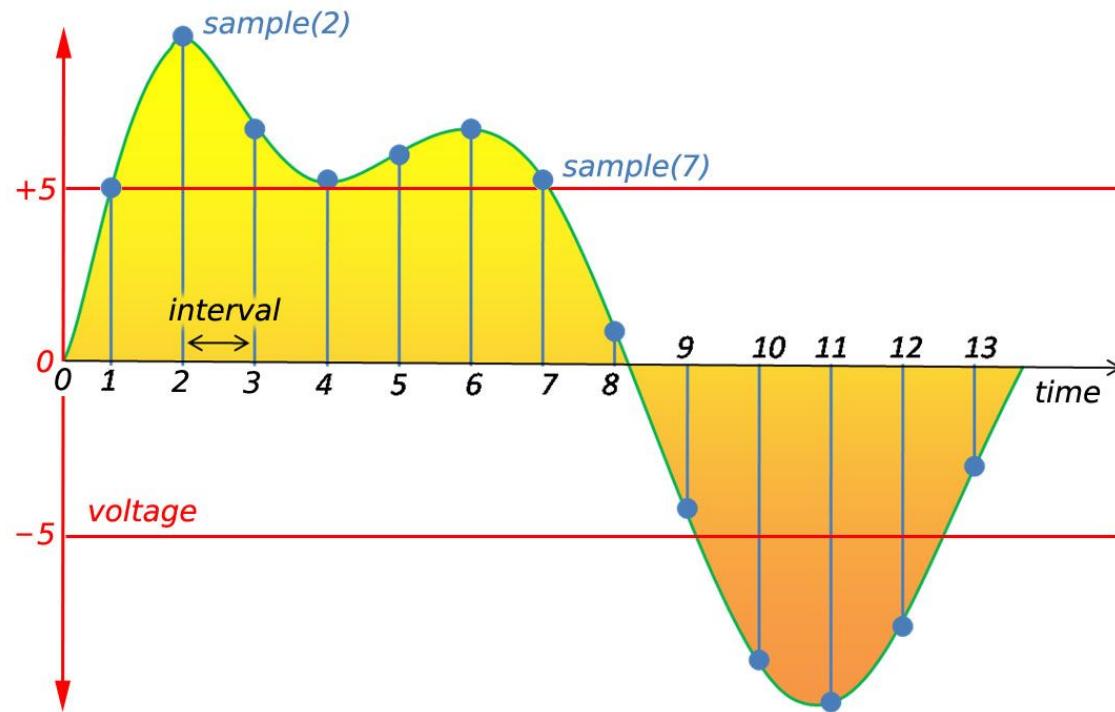
ADC and DAC

ADC (Analog to Digital Converter): è un sistema che converte un segnale analogico in un segnale digitale. Di norma, il segnale digitale prodotto riflette l'andamento del voltaggio della corrente in entrata. La conversione include il processo di quantizzazione (introduce di conseguenza piccoli errori).

DAC (Digital to Analog Converter): è un sistema che esegue l'operazione inversa dell'ADC. Il processo di conversione può degradare il segnale. In particolare, il problema che può intervenire è quello della distorsione armonica.

Conversione analogico-digitale

Nella conversione analogico digitale, viene utilizzato il processo di campionatura per acquisire, a intervalli regolari, dei campioni (samples) del segnale elettrico.



Pulse code modulation

La tecnica di conversione più utilizzata è quella di pulse code modulation (PCM): ad essere campionata a intervalli regolari è l'ampiezza del segnale analogico. Ogni campione viene quantizzato al valore più vicino all'interno del range dei possibili valori digitali.

Esistono quindi due caratteristiche principali di un segnale audio:

- La quantità di bits utilizzati (bit depth): influenza la qualità sonora in termini di dinamica massima acquisibile.
- La frequenza di campionamento (sample rate): influenza la qualità sonora in termini di frequenza massima acquisibile.

Range dinamico di segnali digitali

Nei segnali digitali il range dinamico massimo ottenibile è limitato dell'errore di quantizzazione e quindi dipende dal numero di bits utilizzati nella conversione.

bits	8	12	16	18	20	24
dB	48	72	96	108	120	144

Di conseguenza la dinamica massima ottenibile con un segnale audio in qualità CD è di 96 dB.

Esistono però algoritmi di noise shaping (dithering) capaci di spostare la percezione del range dinamico a 120 dB, approfittando della risposta in frequenza dell'orecchio umano.

Range dinamico di segnali digitali

In informatica 16 bits (fixed point) equivalgono al tipo di dato “short”. Il range di valori è quindi tra –32'768 e 32'767.

Oggi, 16 bits o 24 bits (fixed point), vengono utilizzati principalmente per l'immagazzinamento dei dati su files.

Per il processing, il formato più utilizzato è quello dei 32 bits virgola mobile, che permette range dinamici più ampi, avvalendosi della rappresentazione con mantissa ed esponente (ad esempio 0.1×10^{-7}).

Decibels full-scale

Per misurare l'ampiezza dei segnali digitali viene utilizzata l'unità di misura dB_{FS} .

È un'unità di misura che mette in relazione l'ampiezza del segnale con l'ampiezza del picco massimo permesso da quel sistema (detto il full-scale).

Oltre alla soglia del full-scale interviene il clipping, una forma di distorsione molto aggressiva.

Al full-scale è assegnato il valore di 0 dB_{FS} , quindi tutti i valori più piccoli del massimo sono negativi.

Ad esempio, un segnale che ha metà dell'ampiezza massima, avrà un'ampiezza di $-6 \text{ dB}_{\text{FS}}$.

VU meters

Negli apparecchi digitali, 0 VU ha normalmente un valore di $-18\text{ dB}_{\text{FS}}$ (18 dB sotto al full scale).

Per le registrazioni a 24-bit (soprattutto per i master) l'allineamento è solitamente a $-24\text{ dB}_{\text{FS}}$ per permettere un headroom aggiuntivo, da sfruttare come margine di manovra nel momento della conversione al formato di distribuzione.

Sample rate

Il sample rate è la frequenza di acquisizione dei campioni. Viene misurato in Hz ed è proporzionale alla durata dell'intervallo che intercorre tra l'acquisizione di un campione e quello successivo. Convertire con un sample-rate di 44.1 KHz significa acquisire 44100 campioni ogni secondo.



Sample rate

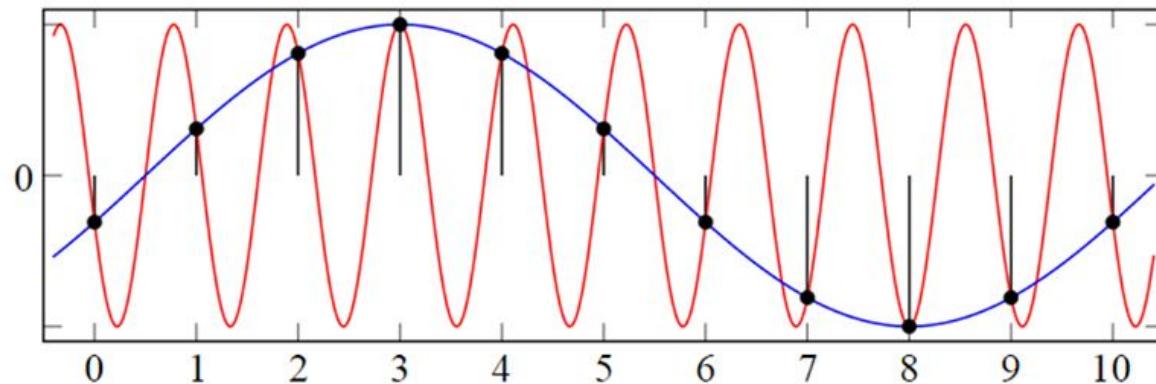
La frequenza massima rappresentabile con un segnale digitale è limitata dal sample rate utilizzato nella la conversione.

Il massimo raggiungibile è la frequenza equivalente alla metà del sample rate. Questa frequenza è della frequenza di Nyquist (in onore di Harry Nyquist).

Ad esempio, con un sample rate di 44.1 KHz è possibile campionare segnali con frequenze fino ad un massimo di 22050 Hz.

Il teorema di Nyquist-Shannon

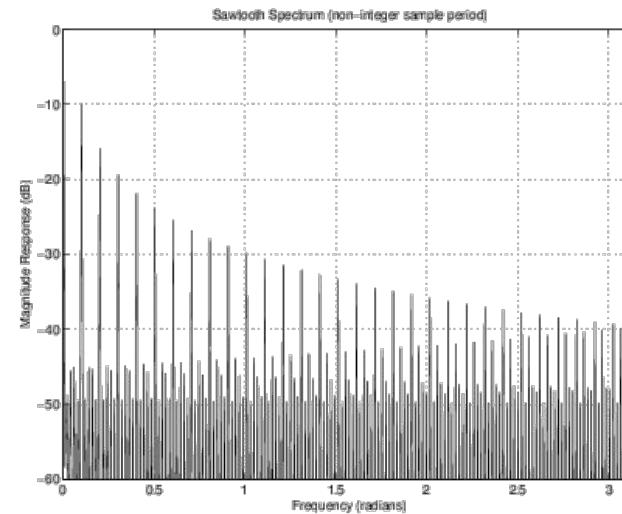
Il teorema di Nyquist-Shannon detto anche sampling theorem, stabilisce le condizioni sufficienti di sample-rate (il doppio della frequenza massima che è necessario rappresentare) per permettere ad una sequenza discreta di campioni di catturare l'informazione di un segnale continuo con una larghezza di banda definita.



Aliasing

Se si campionano frequenze oltre alla frequenza di Nyquist interviene un fenomeno detto aliasing.

Si osserva un effetto di folding. Le frequenze oltre la frequenza massima vengono riflesse all'interno dello spettro, generando frequenze inesistenti.



È possibile evitare l'aliasing applicando un filtro (passa bassa) al segnale. Va applicato prima della conversione.

Jitter

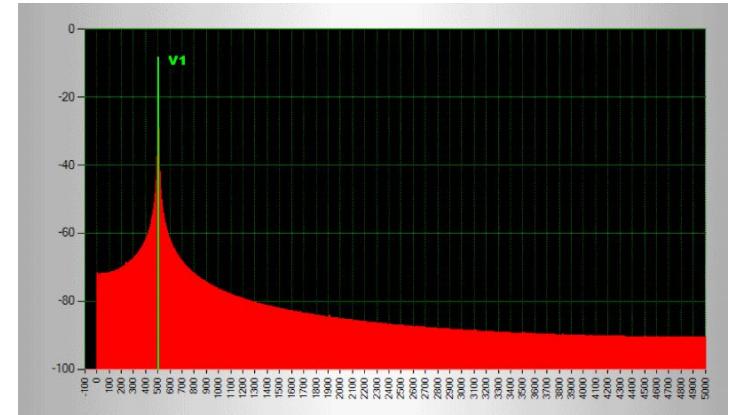
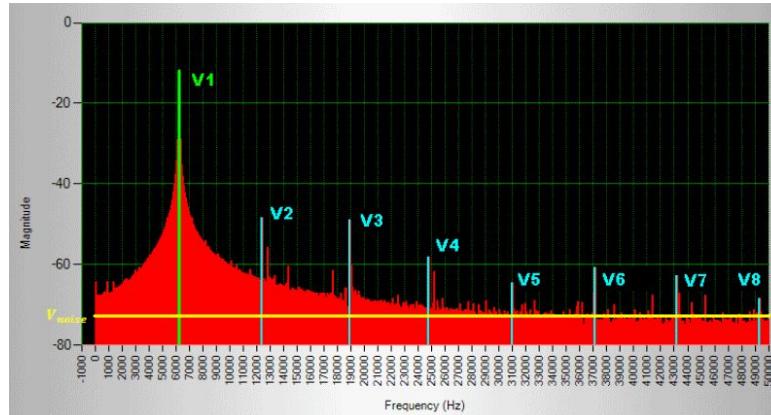
Nel processo di conversione esiste un ulteriore fattore che determina la qualità del segnale risultante.

Il sampling jitter è una lieve variazione nella precisione dei clock dei convertitori ADC o DAC. Questa lieve alterazione può modificare lo spettro delle fasi del segnale, riducendo conseguentemente la qualità di riproduzione, in particolare per quanto concerne i transienti del segnale.

Total Harmonic Distortion (THD)

La Total Harmonic Distortion (THD), è il valore RMS degli armonici prodotti dai convertitori (ADC o DAC) relativi al livello RMS di un segnale input sinusoidale vicino al full-scale. Ad esempio, assumendo un segnale input di frequenza f , la non linearità del convertitore produce le frequenze armoniche $2f$, $3f$, $4f$, ... non presenti nel segnale originale. Generalmente queste frequenze armoniche distorcono il segnale di output.

THD+N è la THD con aggiunto il rumore di fondo (noise).



Formati audio per files

I formati audio per files permettono di immagazzinare dati audio su hard-disks e altre memorie persistenti. Vengono utilizzati anche per lo streaming di dati su internet.

Il formato di audio coding è la soluzione utilizzata per immagazzinare i dati. Può essere di tipo RAW (vengono immagazzinati direttamente i dati in formato PCM) oppure compresso per ridurre la taglia del file.

Anche se è possibile immagazzinare i dati RAW senza ulteriore informazioni, solitamente vengono utilizzati formati container o formati specifici audio, con metadati contenenti informazioni sul contenuto del file (come il numero di bits e il sample-rate dei dati PCM).

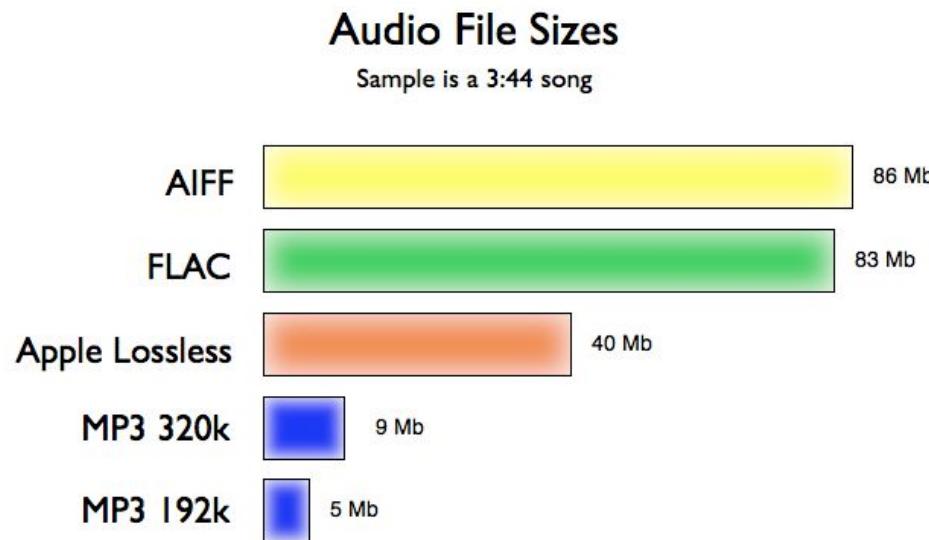
Formati audio per files

Ci sono tre maggiori tipologie di formati audio per files:

- Formati non compressi come .wav, .aiff o .au. Contengono direttamente i dati in formato PCM, preceduti da un header. In alcuni casi possono contenere direttamente i dati RAW senza alcun header.
- Formati con compressione lossless, come .flac, .ape (Monkey's Audio) o .wv (WavPack). Contengono i dati PCM, ma compressi con strumenti simili al .zip.
- Formati con compressione lossy, come .mp3, .aac, .opus. Sfruttano forme di compressione psicoacustica per ridurre la dimensione dei files.

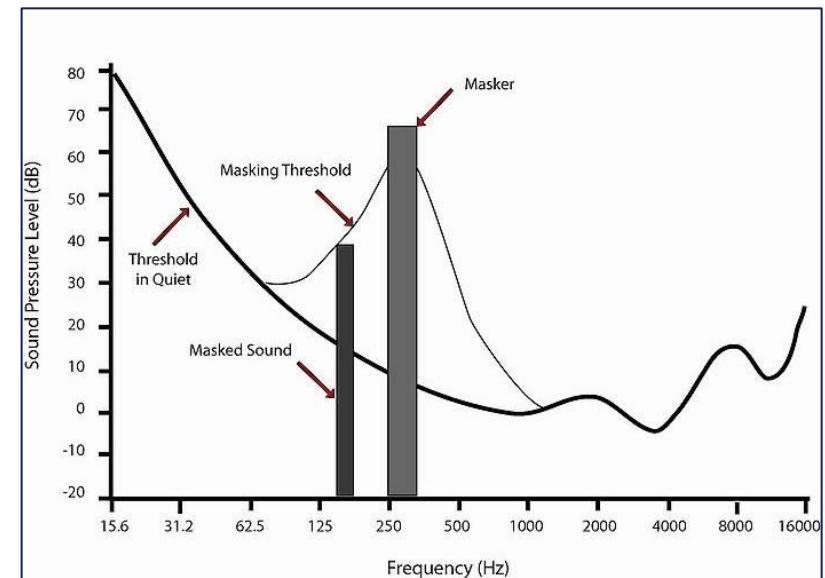
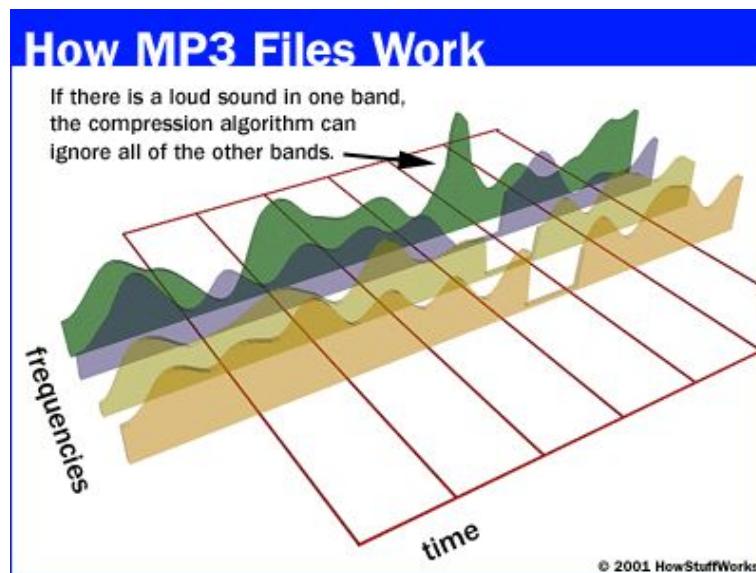
Formati audio per files

- PCM, 16 bits, 44.1 KHz, mono: 16 bits per campione \times 44'100 campioni al secondo = 705'600 bits per secondo.
- PCM, 16 bits, 44.1 KHz, mono: 16 bits per campione \times 44'100 campioni al secondo \times 2 canali = 1'411'200 bits per secondo.
- MP3 256: comprime in al massimo 256'000 bits per secondo, indipendentemente dal numero di canali.



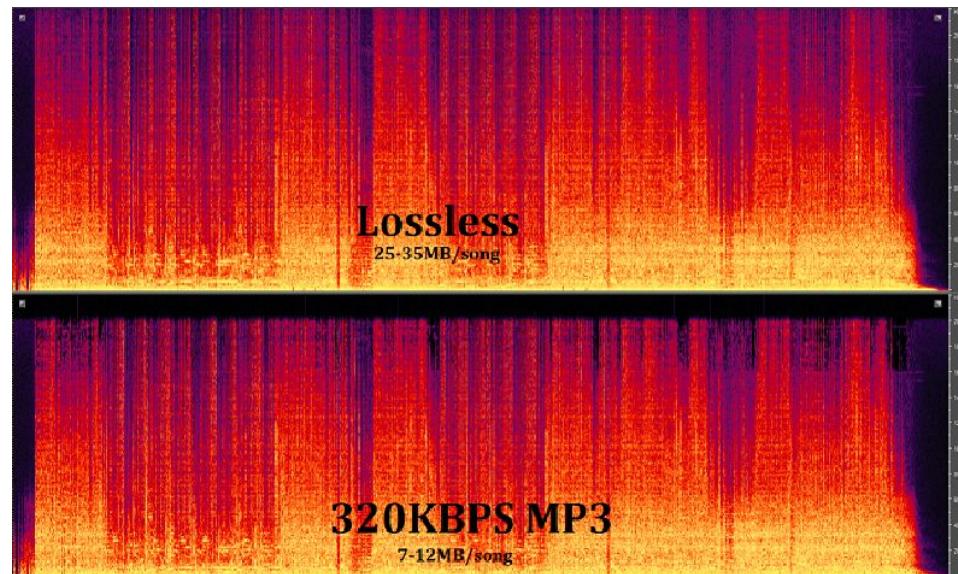
Compressione psicoacustica

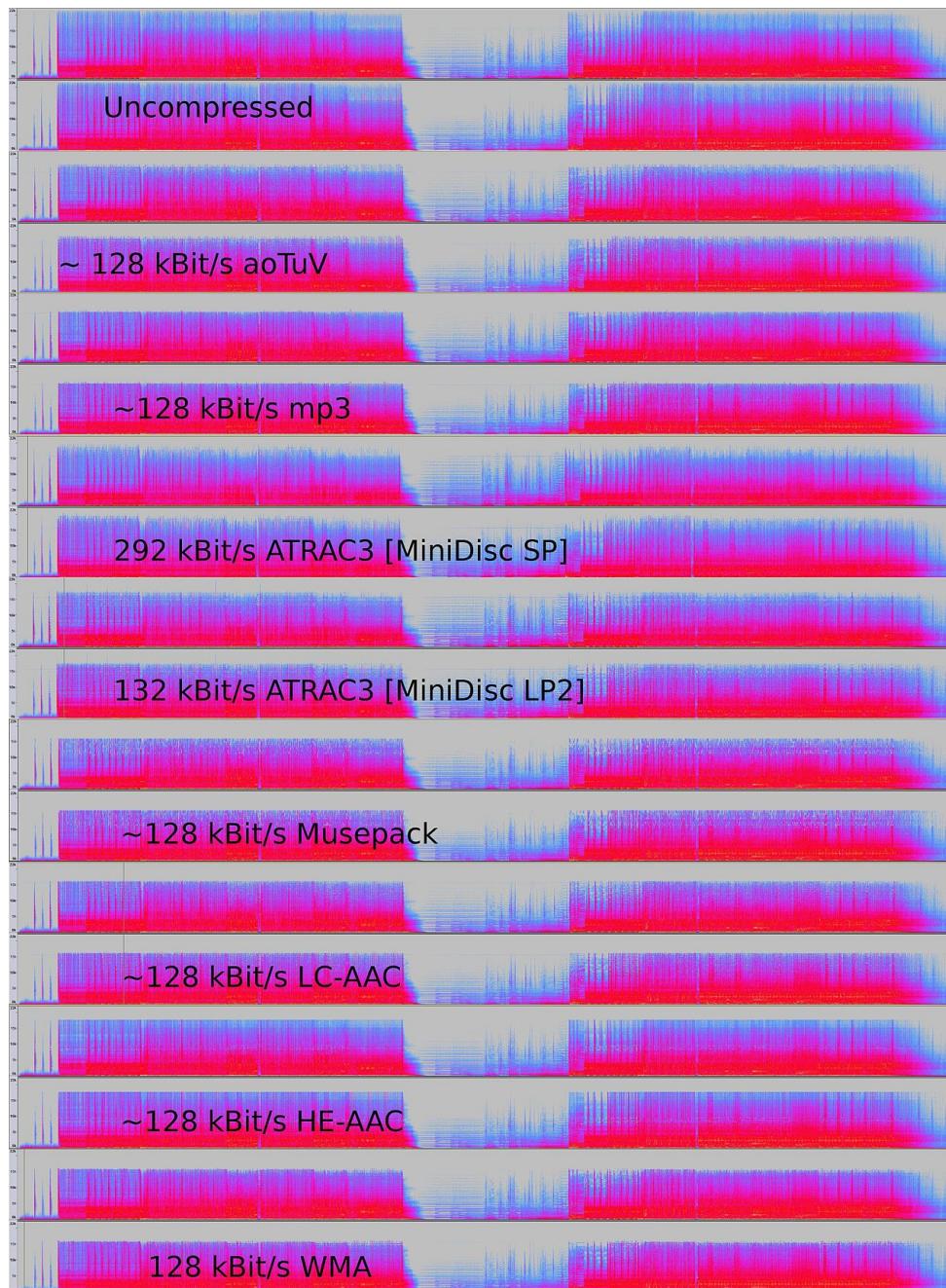
La compressione psicoacustica (anche chiamata perceptual coding) lavora riducendo o approssimando la precisione di alcune componenti sonore considerate sotto la capacità di udito delle persone.



Compressione psicoacustica

I difetti che vengono attribuiti alla compressione psicoacustica sono: forme di distorsione, timbro metallico, perdita di profondità e di immagine 3D, riduzione della ricchezza sonora e suono complessivamente impastato.





SUPSI

Tecniche base di audio processing

Fondamenti di Multimedia Processing

Tiziano Leidi

05.10.2018

Processing di segnali audio

L'audio signal processing comprende lo studio, lo sviluppo e l'utilizzo di tecniche (di signal processing) per modificare in maniera intenzionale segnali audio.

In passato, si utilizzavano processori audio analogici di tipo elettronico (rari casi, come il phaser, utilizzavano tecniche meccaniche). Oggi, vengono impiegati soprattutto processori audio digitali.

Dal punto di vista del trattamento dei segnali, l'avvento del digitale ha introdotto molte nuove possibilità, ampliando le possibili forme di processing realizzabili.

Processing di segnali audio

Le tecniche di audio processing più comuni comprendono:

- equalizzazione e impiego di altri tipi di filtri
- modulazioni di ampiezza o di frequenza
- compressione, espansione e limiting
- aggiunta di delay e riverberi
- chorus, flanger, phaser
- pitch-shift e time-stretching
- effetti audio 3D

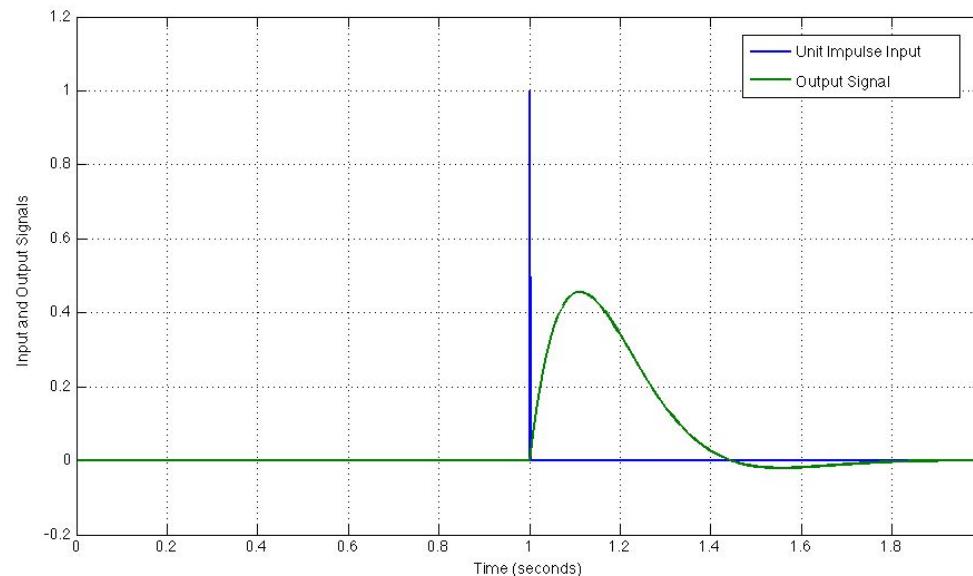
Alcuni dei possibili effetti vengono ottenuti con dei semplici trucchi.
Altri richiedono l'impiego di tecniche avanzate di signal processing.
Discuteremo l'impiego dell'FFT e dei filtri FIR/IIR.

Processing di segnali audio

Per poter operare nell'ambito del processing di segnali audio sono necessari alcuni strumenti di analisi ...

Risposta impulsiva (impulse response)

La risposta impulsiva di un sistema è una funzione (del tempo) che coincide con l'output del sistema se sollecitato con un impulso. Un impulso è un segnale idealmente istantaneo e di massima ampiezza (teoricamente infinita). In ambito analogico viene detto Dirac delta function, mentre in ambito digitale Kronecker delta function.



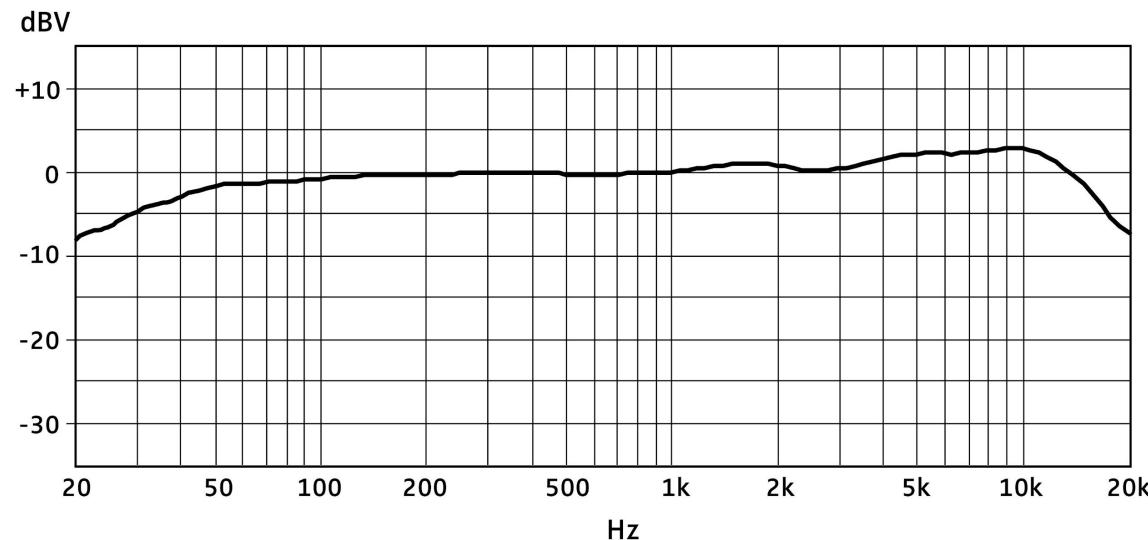
Risposta in frequenza

La risposta in frequenza è una misura quantitativa dello spettro di output di un sistema, utilizzata per caratterizzare la dinamica del sistema.

Solitamente la si misura sollecitando il sistema con uno sweep di ampiezza costante o con un segnale con ampio spettro di frequenze (come il rumore bianco).

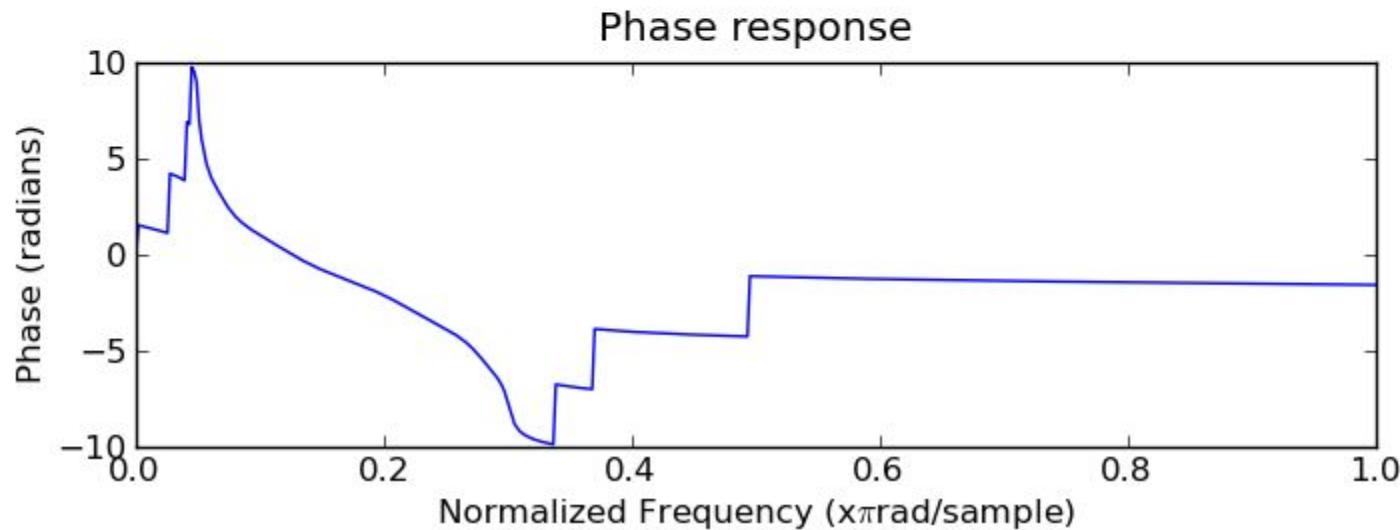
Risposta in frequenza

Per un sistema audio, una risposta in frequenza uniforme (piatta), significa assenza di alterazioni sonore (colorazione o distorsione). Ad esempio, un amplificatore audio ad alta fedeltà ha di norma una risposta in frequenza piatta fra i 20 Hz e i 20,000 Hz, con tolleranze di soli ± 0.1 dB nel range delle frequenze medie.



Risposta in fase

In maniera simile alla risposta in frequenza è possibile rappresentare la risposta in fase: mette in relazione le differenze di fase fra il segnale di input e quello di output di un sistema.



Trasformata di Fourier

I segnali possono essere continui o campionati, ma sono in ogni caso composti da numeri reali. Però nel signal processing si lavora spesso con numeri complessi.

Alcune caratteristiche dei segnali vengono meglio comprese se si opera nel dominio delle frequenze, dove ogni singola frequenza (sinusoide) è definita dalla sua ampiezza e dalla sua fase.

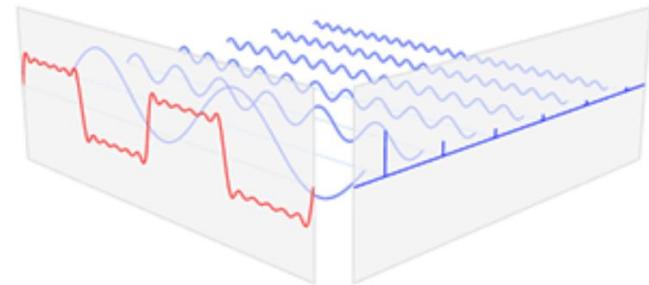
Ampiezza e fase vengono combinati dagli ingegneri in un unico numero complesso: $a \cos(\phi) + i a \sin(\phi)$.

Quest'approccio introduce la trasformata di Fourier ...

Trasformata di Fourier

La trasformata di fourier scomponete una funzione del tempo (quindi un segnale) nelle frequenze che la compongono. Il risultato è una funzione di numeri complessi, il cui valore assoluto rappresenta l'ampiezza di ogni frequenza del segnale, mentre l'argomento complesso è la differenza di fase della sinusoide di quella determinata frequenza.

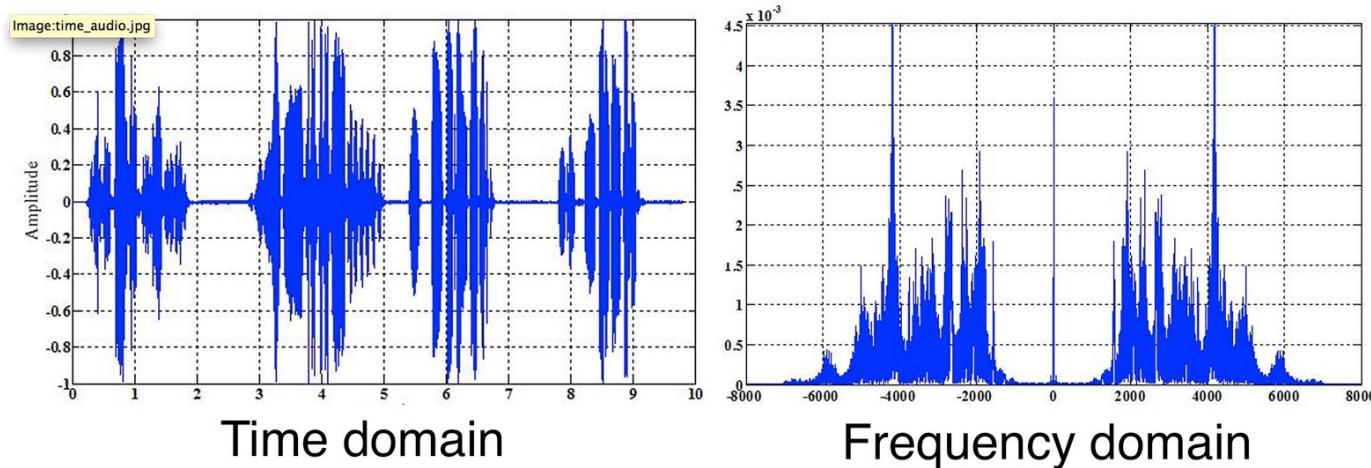
$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$$



x rappresenta il tempo (in secondi), mentre la variabile ξ rappresenta la frequenza (in hertz)

Trasformata di Fourier

La trasformata di fourier permette quindi di passare dal dominio del tempo al dominio delle frequenze.

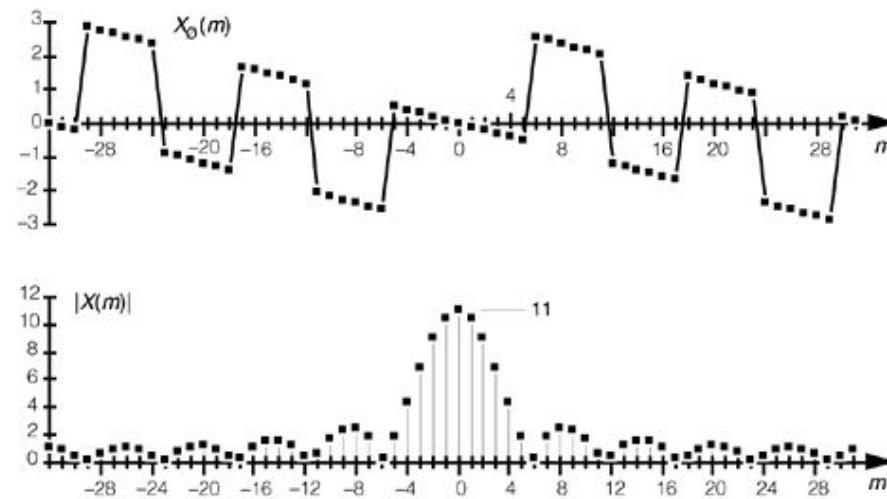


La sua inversa viene chiamata sintesi di Fourier:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

Trasformata discreta di Fourier

In digitale, per il calcolo della trasformata di Fourier viene utilizzata la trasformata discreta di Fourier - discrete Fourier transform (DFT) - che converte: una sequenza finita di campioni di una funzione, in una sequenza (della stessa lunghezza) di campioni della trasformata di Fourier. L'intervallo tra ogni frequenza è il reciproco della durata della sequenza in input.

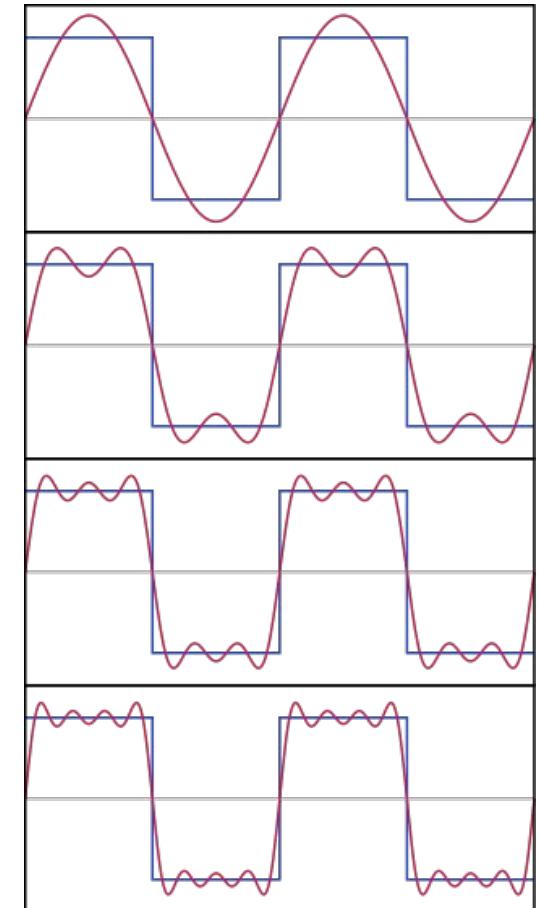


Trasformata discreta di Fourier

L'inversa della DFT è una serie di fourier, che usa i campioni ottenuti tramite la DFT come coefficienti delle sinusoidi delle frequenze:

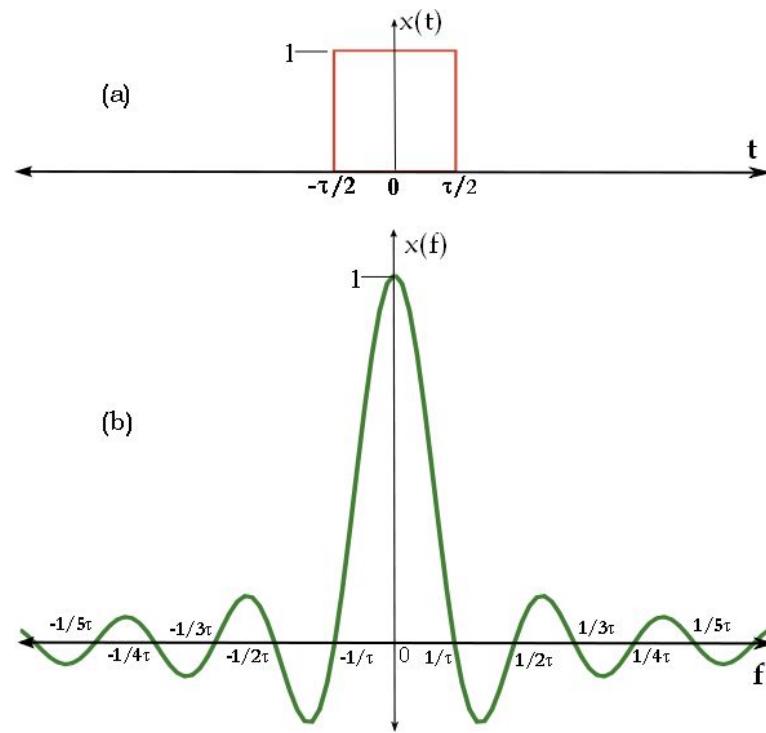
$$s_N(x) = \frac{A_0}{2} + \sum_{n=1}^N A_n \cdot \sin\left(\frac{2\pi nx}{P} + \phi_n\right)$$

Per il calcolo della DFT viene normalmente impiegato l'algoritmo di Fast Fourier Transform (FFT), che è particolarmente efficiente in termini di tempi di calcolo.



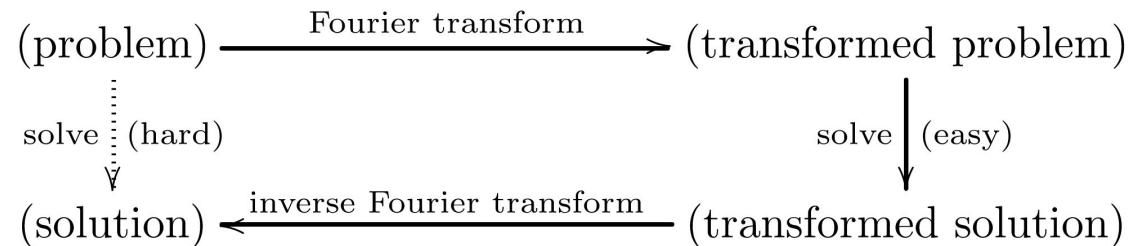
Trasformata di Fourier

Segnali che sono localizzati nel tempo hanno trasformate di Fourier che sono distribuite nel dominio delle frequenze e viceversa.



Trasformata di Fourier

Le operazioni lineari in uno dei due domini (tempo o frequenza) hanno corrispondenti operazioni nell'altro dominio, che sono spesso più facili da eseguire.



Ad esempio la convoluzione (un'operazione spesso utilizzata in audio processing) nel dominio del tempo corrisponde alla moltiplicazione nel dominio delle frequenze.

Trasformata di Fourier

Concretamente, ogni sistema dinamico lineare stazionario - linear time-invariant system (LTI) - come ad esempio un filtro, può essere espresso in maniera relativamente semplice come operazione sulle frequenze.

Dopo l'esecuzione delle operazioni si torna nel dominio del tempo per ottenere il risultato finale.

Una delle possibili implementazioni di filtri audio è quindi quella di utilizzare l'algoritmo di FFT ed eseguire il filtraggio direttamente nel dominio delle frequenze.

Filtr: sistemi LTI

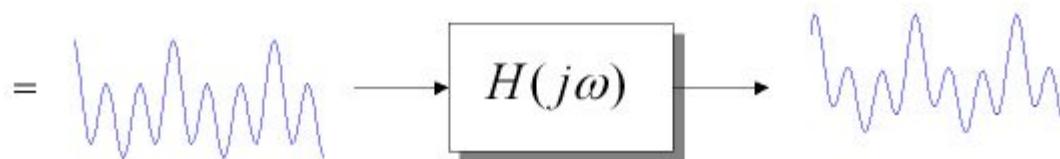
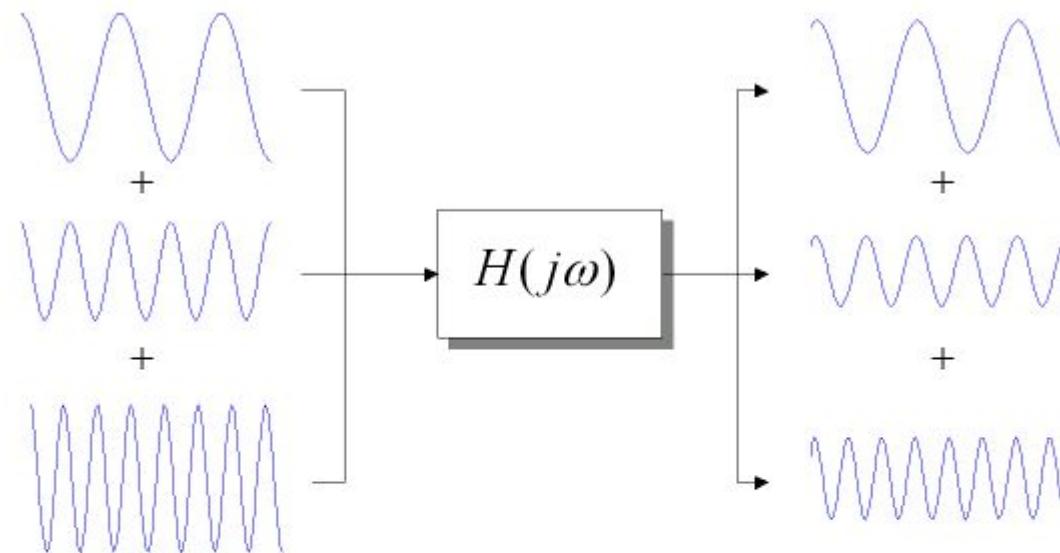
Linearità significa che, se il sistema ha due o più inputs e due o più corrispondenti outputs ad ogni combinazione lineare degli inputs

$$a_1 v_1 + \dots + a_n v_n$$

corrisponde una combinazione lineare degli outputs.

Tempo-invariante (time invariant), significa che non è importante quando viene fatto partire il sistema. Per un determinato input, risulterà comunque sempre il medesimo output.

Filtri: sistemi LTI



Filtri: sistemi LTI

Nella realtà, non esiste nessun sistema che è veramente LTI. Alcune non-linearità possono essere introdotte ad esempio a causa di variazioni di temperatura o altri fattori esterni. Molti sistemi possono però essere modellati per semplicità come sistemi LTI, trascurando le eventuali non-linearità.

I sistemi LTI sono importanti perché possono essere risolti con metodologie di signal processing. Invece, i sistemi che non sono LTI sono rappresentati da equazioni differenziali non lineari, molte delle quali non sono risolvibili.

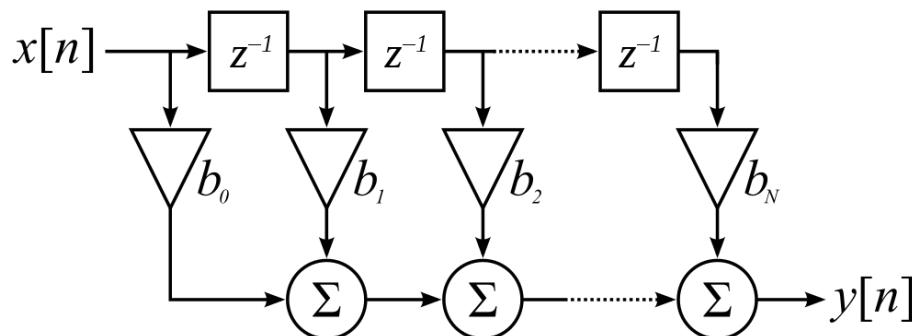
è possibile realizzare filtri con l'FFT (come descritto precedentemente. Alternative più efficaci sono i filtri FIR e IIR.

Filtrati FIR

Un filtro con finite impulse response (FIR) è un filtro la cui risposta impulsiva (o risposta a qualsiasi input di lunghezza finita) è di durata finita, perché l'output va automaticamente a 0 in un tempo finito.

Ogni valore della sequenza di output è una somma pesata dei valori di input più recenti.

$$y[n] = b_0 x[n] + b_1 x[n - 1] + \cdots + b_N x[n - N] = \sum_{i=0}^N b_i \cdot x[n - i]$$



$x[n]$: è il segnale di input

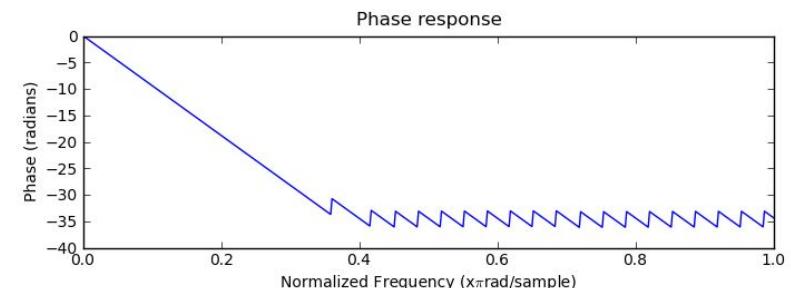
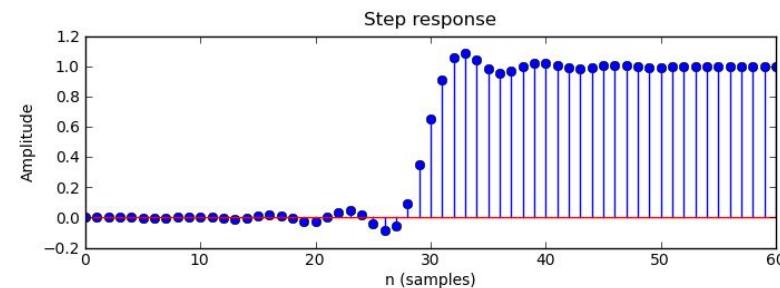
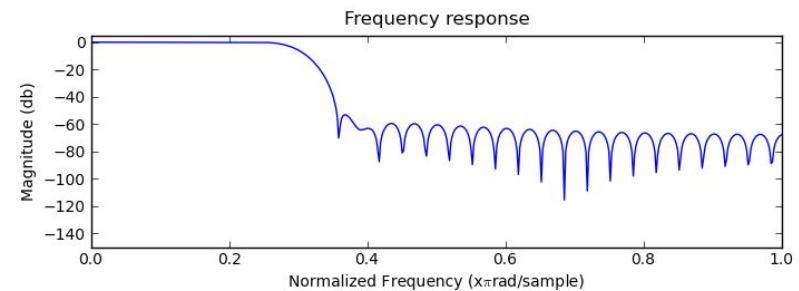
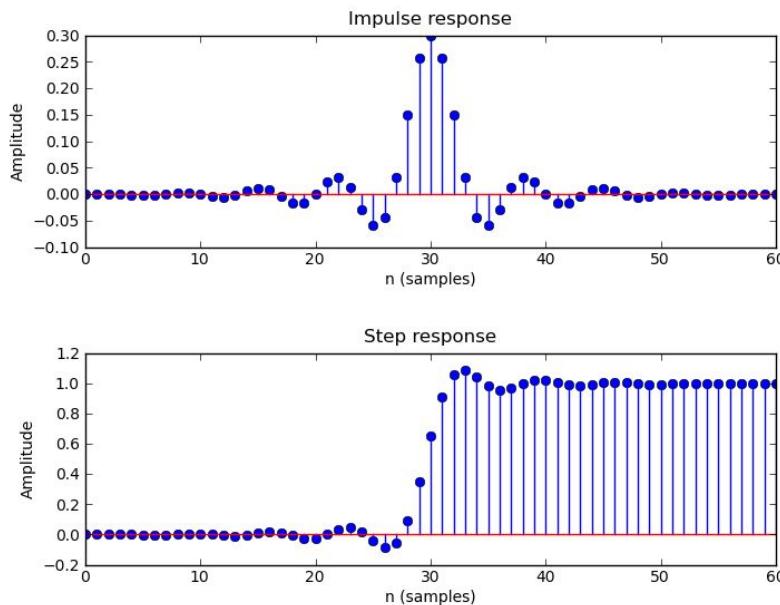
$y[n]$: è il segnale di output

N : è l'ordine del filtro

b_i : sono i coefficienti del filtro

Filtrи FIR

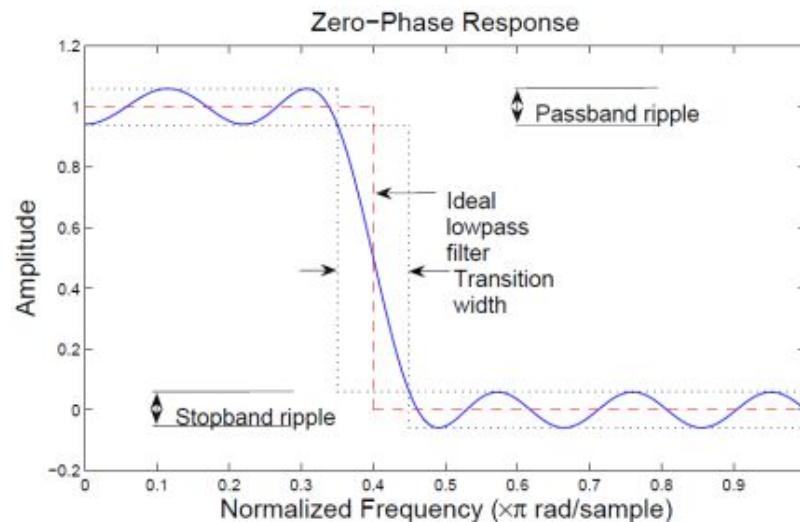
La risposta impulsiva di un filtro FIR di ordine N dura esattamente $N + 1$ campioni (dal primo elemento diverso da 0 all'ultimo elemento diverso da 0) prima che l'output vada a 0.



Filtri FIR

I filtri FIR possono garantire una risposta di fase lineare, caratteristica interessante ad esempio nei filtri di crossover degli altoparlanti.

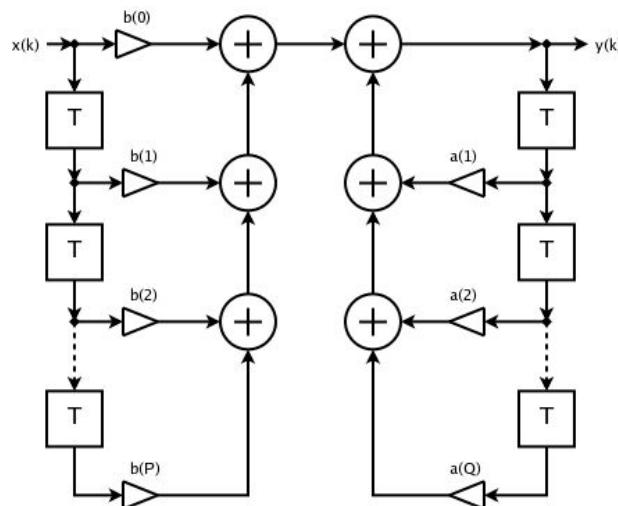
come controparte richiedono una quantità di calcoli maggiore rispetto ai filtri IIR, soprattutto per i tagli nelle basse frequenze. Inoltre è difficile rendere i filtri FIR parametrizzabili in tempo reale. Il calcolo dei coefficienti è un'operazione tipicamente complessa.



Filtr IIR

I filtri IIR sono filtri con risposta impulsiva infinita. A differenza dei filtri FIR hanno un feedback (una parte ricorsiva del filtro). Per questo motivo la risposta in frequenza dei filtri IIR è migliore di quella dei filtri FIR dello stesso ordine.

$$y[n] = \frac{1}{a_0} (b_0 x[n] + b_1 x[n-1] + \cdots + b_P x[n-P] - a_1 y[n-1] - a_2 y[n-2] - \cdots - a_Q y[n-Q])$$



$x[n]$: è il segnale di input

$y[n]$: è il segnale di output

P: è l'ordine della parte in feedforward

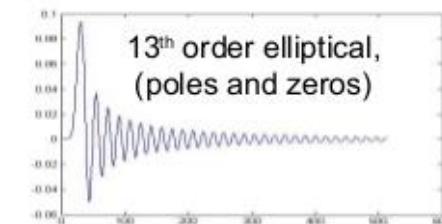
b_i : sono i coefficienti di feedforward

Q: è l'ordine della parte in feedback

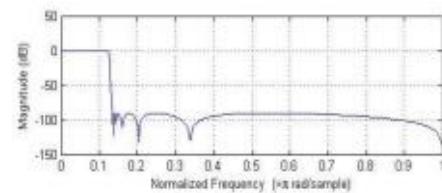
a_i : sono i coefficienti di feedback

Filtr IIR

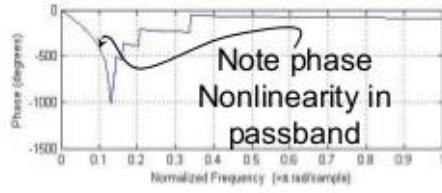
I filtri IIR sono più efficienti e possono essere facilmente parametrizzati in tempo reale. La risposta di fase però non è lineare. Inoltre, se non vengono disegnati correttamente, possono diventare instabili a causa dell'anello di feedback.



Impulse
Response
Notice similar
length



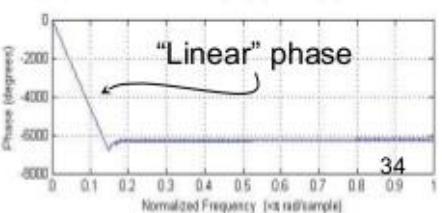
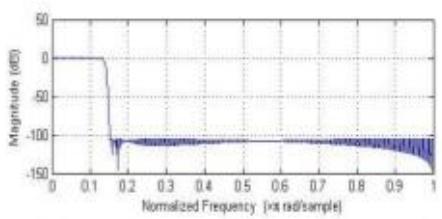
Magnitude
Response
Notice similar frequency
response



Phase
Response
DIFFERENT
PHASE
RESPONSE



512 point
Symmetric
FIR

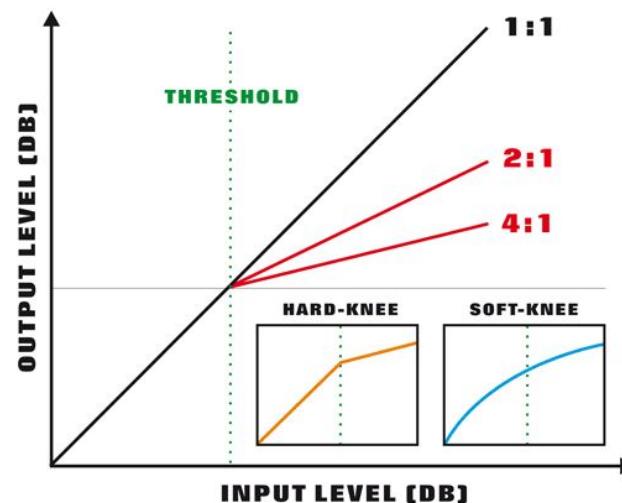


"Linear" phase
34

Processori di dinamica

Le trasformazione eseguite dai processori di dinamica come compressori e limiter sono di principio tutte non-lineari. Quindi, non è possibile utilizzare gli strumenti del signal processing. Le possibilità di trattamento sono quindi relativamente limitate.

Nella maggior parte dei casi questi tipi di processori mantengono in memoria un certo numero di dati recenti che usano per adattare dinamicamente la dinamica (quindi l'inviluppo) del segnale d'uscita.



SUPSI

Nozioni di base sull'immagine

Fondamenti di Multimedia Processing

Tiziano Leidi

19.10.2018

Immagine

Con immagine si intende una rappresentazione (in forma di artefatto) della percezione visiva umana di un soggetto o di un ambiente. Un'immagine può essere bidimensionale come una foto, oppure tridimensionale come un ologramma.

Al lato pratico, le immagini catturano la luce.

Possono essere generate con dispositivi ottici (videocamere, microscopi, telescopi) oppure dall'essere umano per il tramite dell'occhio.

Luce

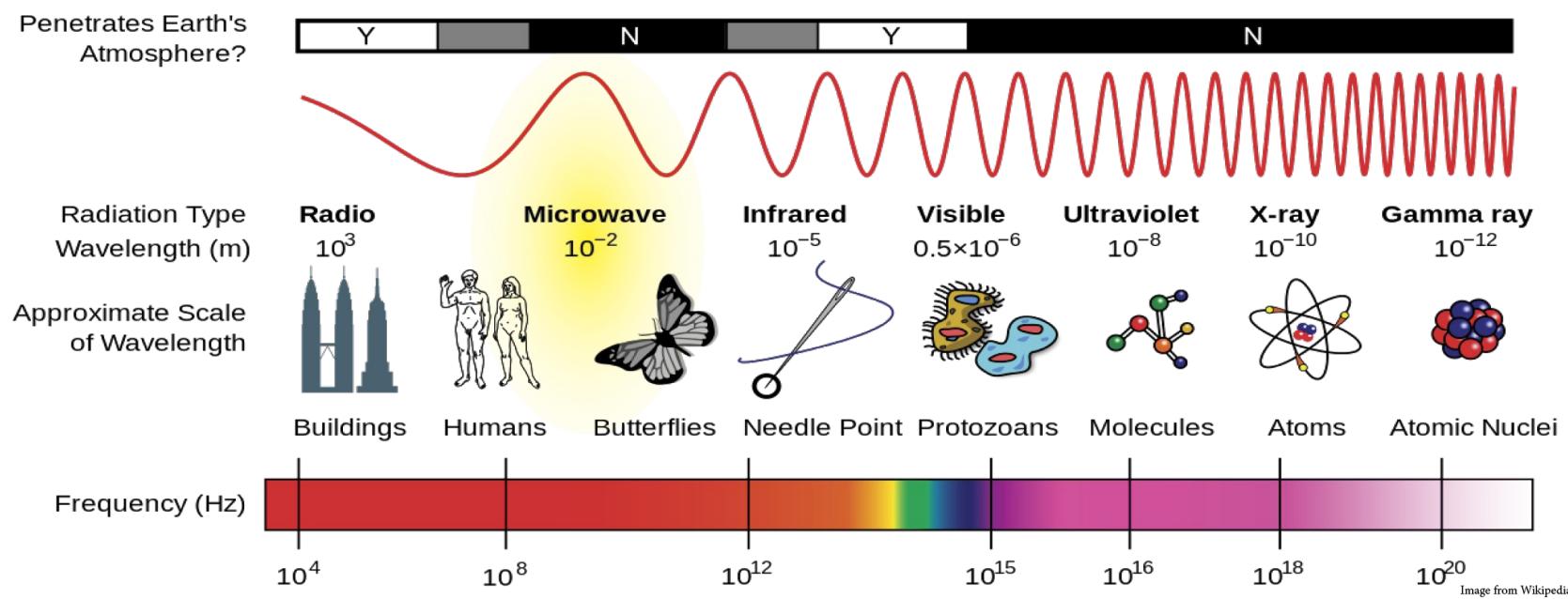
La luce è una radiazione elettromagnetica (EMR) in una certa porzione dello spettro elettromagnetico. Con luce si intende solitamente la luce visibile, quella percepita dal senso della vista.

La luce visibile è definita con lunghezza d'onda tra i 400 e i 700 nm, tra gli infrarossi e gli ultravioletti. La frequenza corrispondente è tra i 430 e i 750 terahertz (THz).

La sorgente principale di luce sulla terra è il sole.

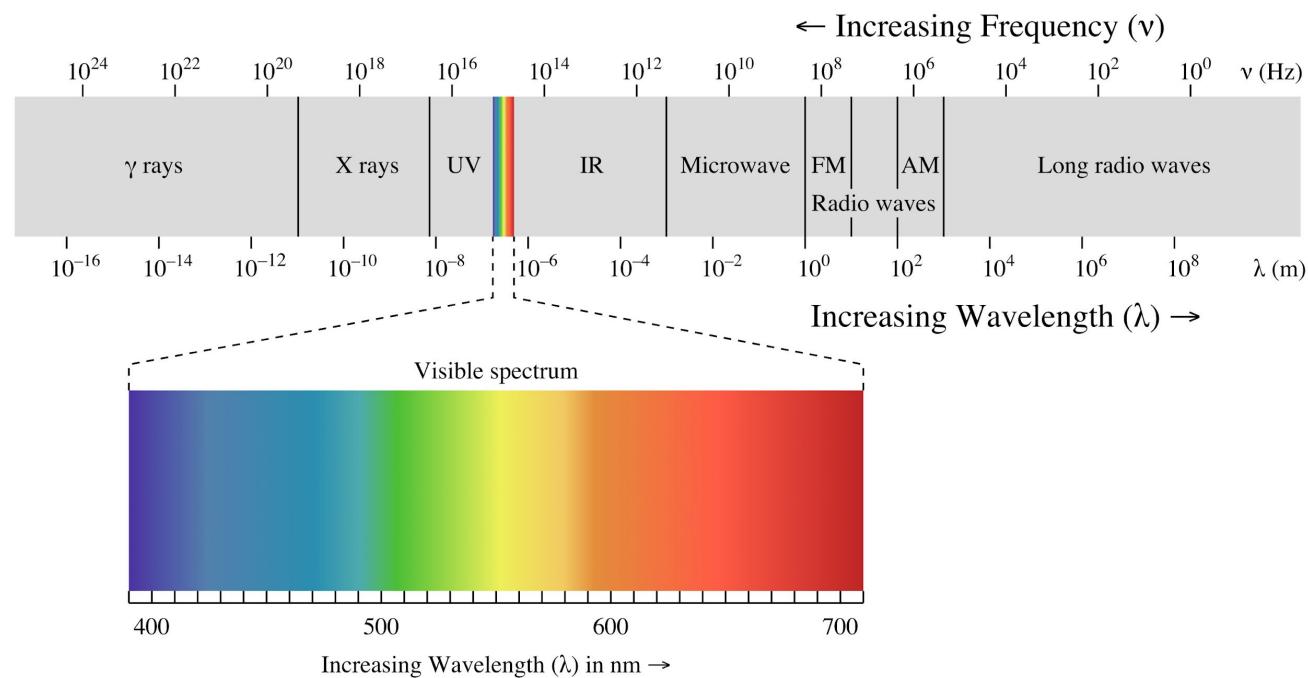
EMR

In generale, l'EMR è classificata per lunghezza d'onda in radio, micro onde, infrarosso, luce visibile, ultravioletto, raggi X e raggi gamma.



Luce visibile

Come ogni tipo di luce, la luce visibile viene emessa e assorbita in piccoli pacchetti chiamati fotoni, che esibiscono proprietà sia delle onde che delle particelle. Questa caratteristica viene definita dualità onda-particella.



Ottica

Lo studio della luce e della sua interazione con la materia è chiamata ottica. L'osservazione e studio dei fenomeni ottici come l'arcobaleno e l'aurora boreale offre molte informazioni sulla natura della luce.



Luce

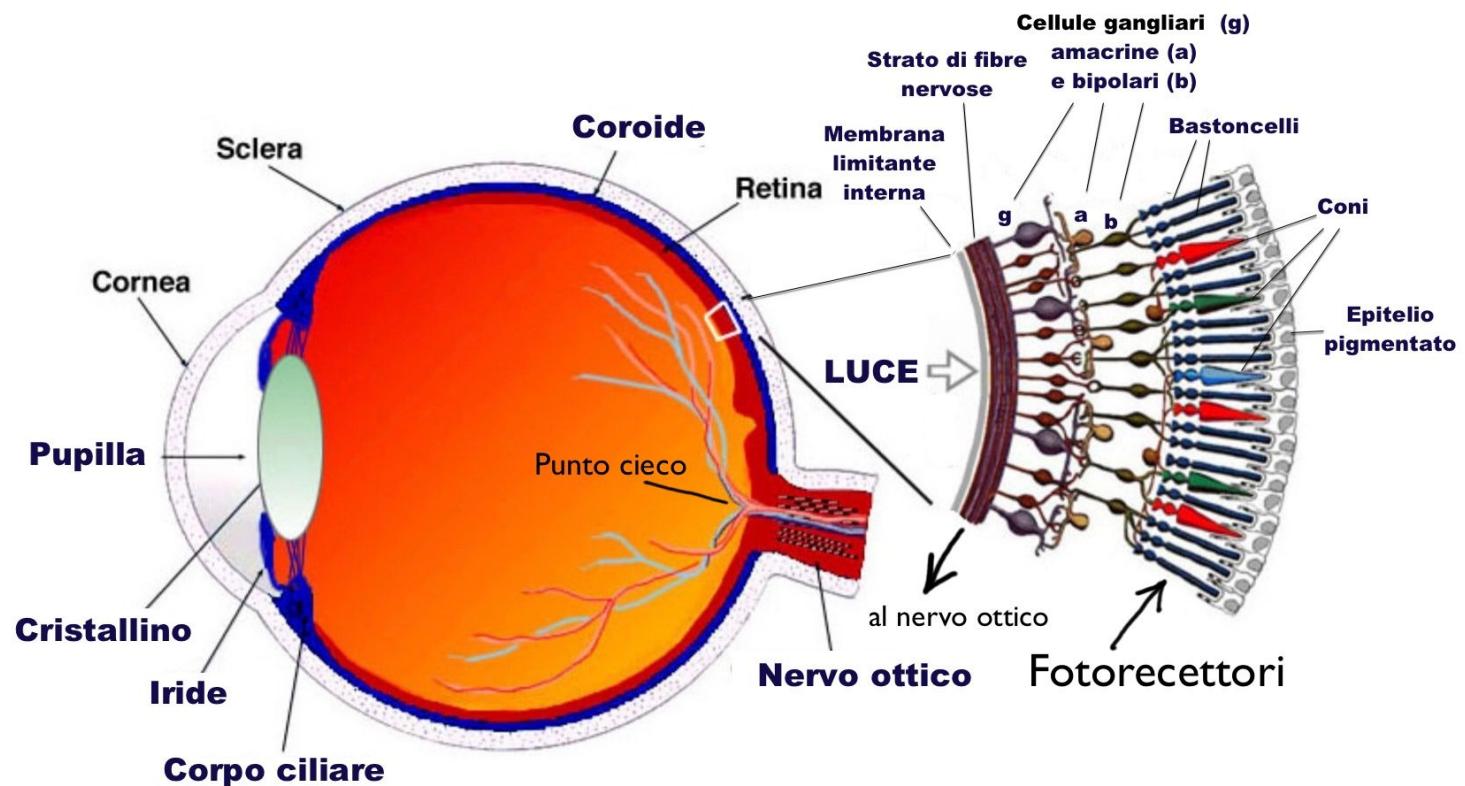
<https://www.youtube.com/watch?v=IXxZRZxafEQ>

Sistema visivo

Per vedere, gli occhi processano la luce. Il senso della vista comincia con la luce che attraversa la cornea, responsabile dei tre quarti della messa a fuoco. Segue l'attraversamento del cristallino che aggiusta la messa a fuoco.

Il risultato è un'immagine chiara proiettata su un foglio di fotorecettori chiamati retina, che fanno parte del sistema nervoso centrale, ma rimangono delocalizzati sul fondo dell'occhio.

Sistema visivo



Sistema visivo

<https://www.youtube.com/watch?v=JcDggCkhyis>

Grandezze radiometriche

Esistono una moltitudine di grandezze utilizzate per misurare l'intensità o potenza della luce. La prima categoria è quella delle grandezze radiometriche (relative alla radiazione elettromagnetica) la cui misurazione è compito della radiometria.

Fra queste:

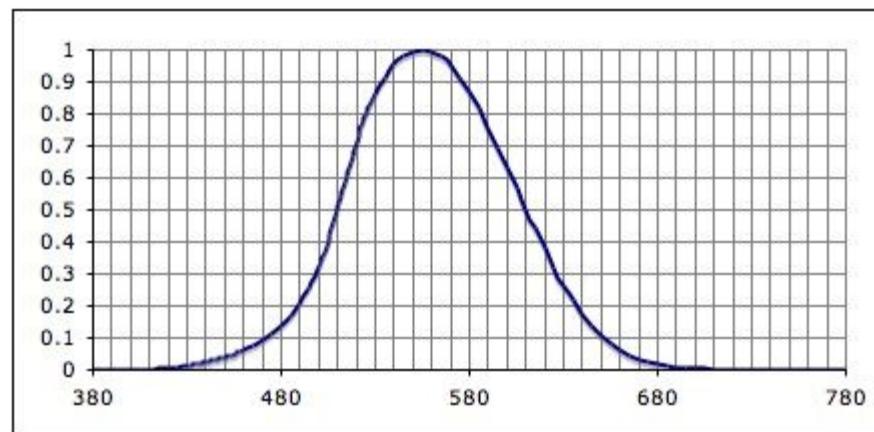
- l'energia radiante, misurata in Joule (J),
- Il flusso radiante, misurato in Watt (W),
- la radianza, misurata in watt allo steradiante per metro quadrato.

Lo steradiante è l'angolo solido: il corrispondente tridimensionale dell'angolo radiante

Grandezze fotometriche

Le grandezze radiometriche non sono direttamente utilizzabili nello scienzia della vista. Il sistema visivo umano presenta una sensibilità non uniforme alle diverse lunghezza d'onda.

Vengono quindi utilizzate le grandezze fotometriche, che quantificano l'emissione luminosa partendo dalle grandezze radiometriche, ma mediante pesatura con la curva di risposta spettrale dell'occhio umano.



Grandezze fotometriche

Fra le grandezze fotometriche troviamo:

- il flusso luminoso, misurato in lumen,
- l'intensità luminosa, misurata in candela,
- l'illuminamento, misurato in lux.

Ad esempio, il flusso luminoso è la misura della potenza della luce che viene percepita dall'essere umano.

Flusso luminoso

Source	Luminous flux (lumens)
37 mW "Superbright" white LED	0.20
15 mW green laser (532 nm wavelength)	8.4
1 W high-output white LED	25–120
Kerosene lantern	100
40 W incandescent lamp at 230 volts	325
7 W high-output white LED	450
6 W COB filament LED lamp	600
18 W fluorescent lamp	1250
100 W incandescent lamp	1750
40 W fluorescent lamp	2800
35 W xenon bulb	2200–3200
100 W fluorescent lamp	8000
127 W low pressure sodium vapor lamp	25000
400 W metal-halide lamp	40000

Adattamento all'intensità luminosa

Il range d'intensità luminose a cui l'occhio umano è capace di adattarsi è enorme, nell'ordine dei 10^{10} . Il sistema visivo umano non riesce a lavorare su tutto questo range simultaneamente, il range d'intensità che l'occhio riesce a coprire simultaneamente è relativamente piccolo, di conseguenza deve reagire alle eventuali grosse variazioni adattandosi progressivamente.

Immagini in scala di grigi

Un'immagine greyscale è un'immagine in cui viene rappresentata solo l'intensità luminosa. Immagini di questo tipo sono anche conosciute come immagini in bianco e nero o monocromatiche. Sono composte esclusivamente da differenti tonalità di grigio fra il nero (l'intensità minore) e il bianco (l'intensità maggiore).



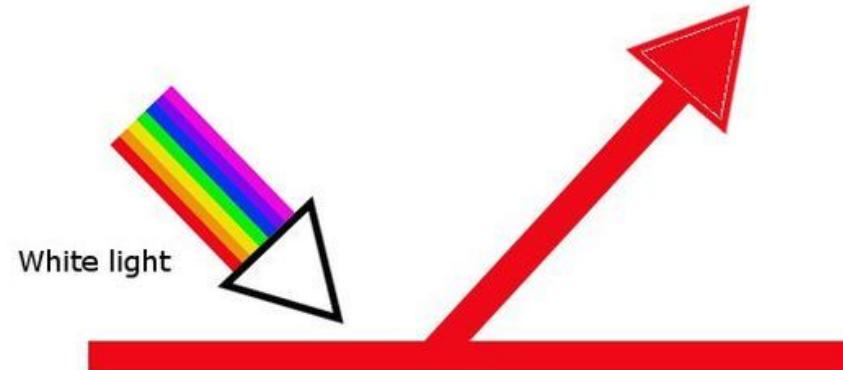
Colori

Il riconoscimento dei colori è una capacità della percezione visiva umana, che permette di classificare la luce visibile in sotto categorie chiamate ad esempio rosso, blu, giallo, verde, arancione o viola.

La percezione dei colori avviene grazie alla stimolazione dei coni dell'occhio da parte delle EMR nello spettro della luce visibile.

Colori

I colori vengono generati dagli oggetti in funzione della loro capacità di riflettere le varie lunghezze d'onda presenti nella luce visibile. Questa capacità di riflessione è governata dalla proprietà fisiche dei differenti oggetti.



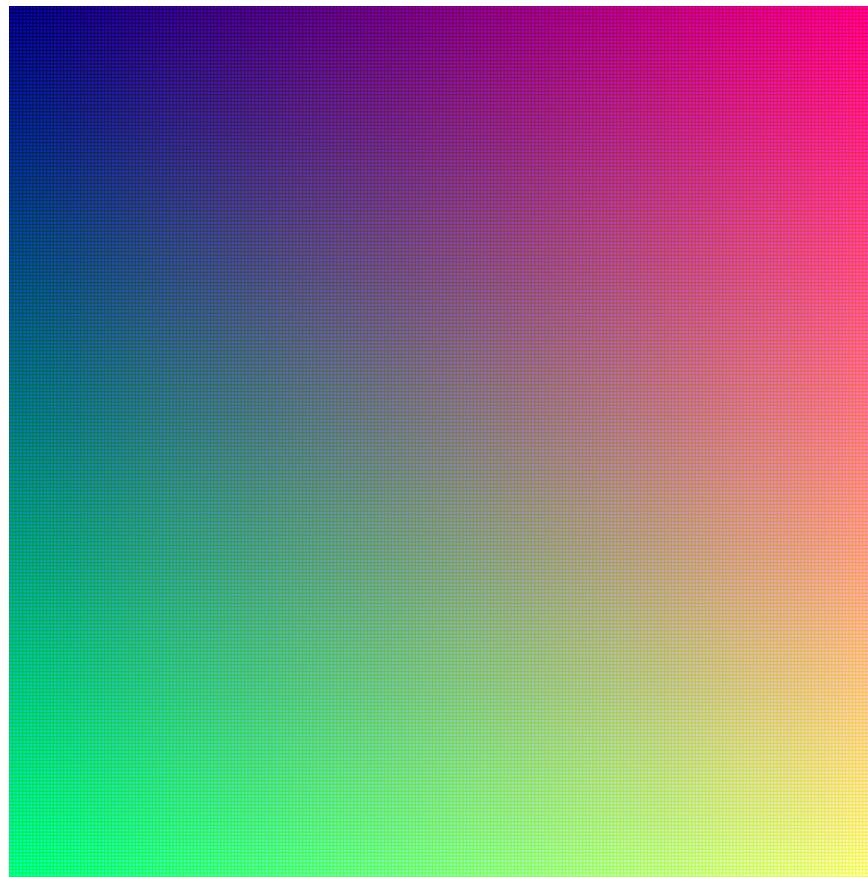
Colori

Color, wavelength, frequency and energy of light

Color	λ (nm)	ν (THz)	ν_b (μm^{-1})	E (eV)	E (kJ mol $^{-1}$)
Infrared	>1000	<300	<1.00	<1.24	<120
Red	700	428	1.43	1.77	171
Orange	620	484	1.61	2.00	193
Yellow	580	517	1.72	2.14	206
Green	530	566	1.89	2.34	226
Blue	470	638	2.13	2.64	254
Violet	420	714	2.38	2.95	285
Near ultraviolet	300	1000	3.33	4.15	400
Far ultraviolet	<200	>1500	>5.00	>6.20	>598

Colori

L'occhio umano può distinguere fino a 10 milioni di diversi colori.



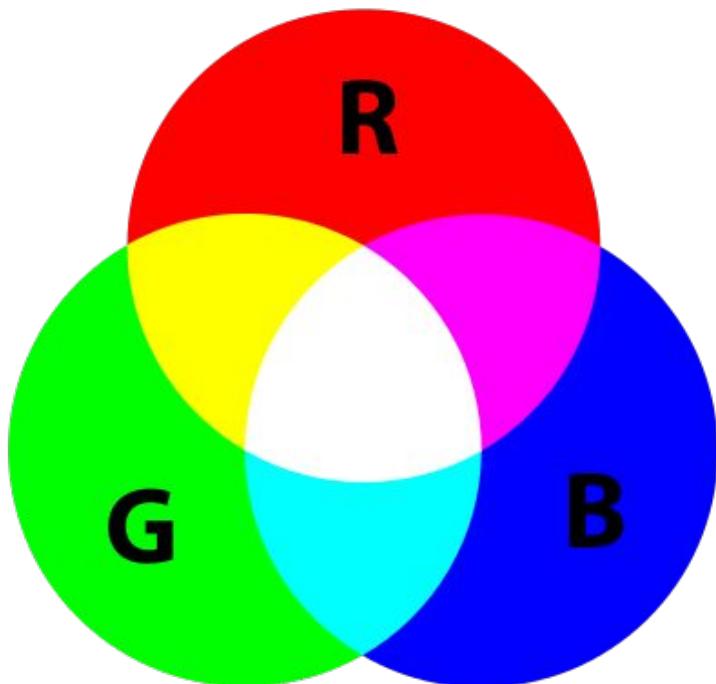
Modelli dei colori

Definendo uno spazio dei colori, è possibile identificarli numericamente con delle coordinate.

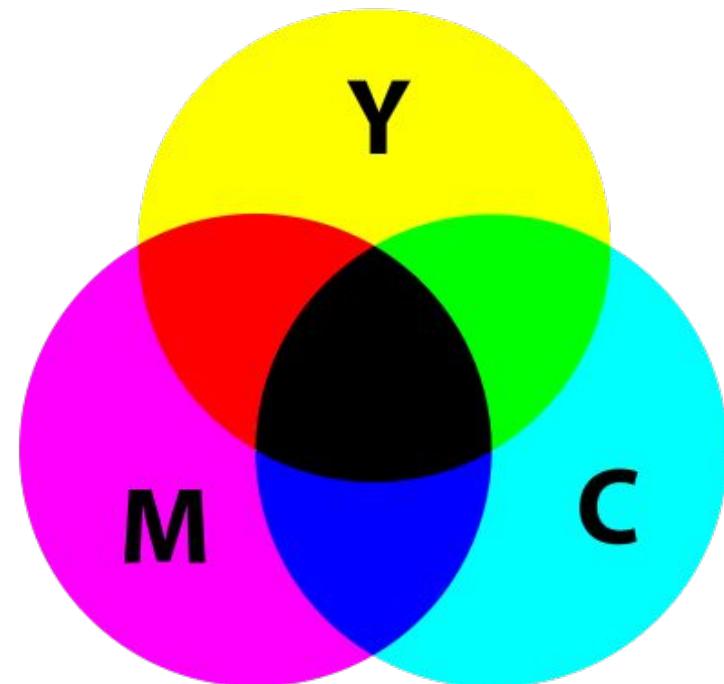
Ad esempio, lo spazio dei colori RGB è basato sui tre tipi di coni dell'occhio umano che rispondono alle tre bande di luce con i picchi tra 564–580 nm (rosso), 534–545 nm (verde) e 420–440 nm (blu).

In altri spazi dei colori possono esserci più di tre dimensioni, come ad esempio nel modello CMYK dove una delle dimensioni rappresenta la vivacità di colorazione (colorfulness).

Modelli dei colori



Additiva



sottrattiva

Modello RGB

Il modello dei colori RGB è basato sulla teoria che tutti i colori visibili possono essere creati sommando i tre colori rosso (R), verde (G) e blu (B).

Combinando rosso e verde nella stessa quantità si ottiene il giallo (Y), con verde e blu il ciano (C) e con rosso e blu il magenta (M). Si generano quindi i colori del modello CMY.

Parametri della percezione dei colori

Lavorare con i modelli dei colori tramite coordinate è complicato. Di conseguenza, per facilitarne l'utilizzo, i modelli dei colori definiscono vari parametri come ad esempio:

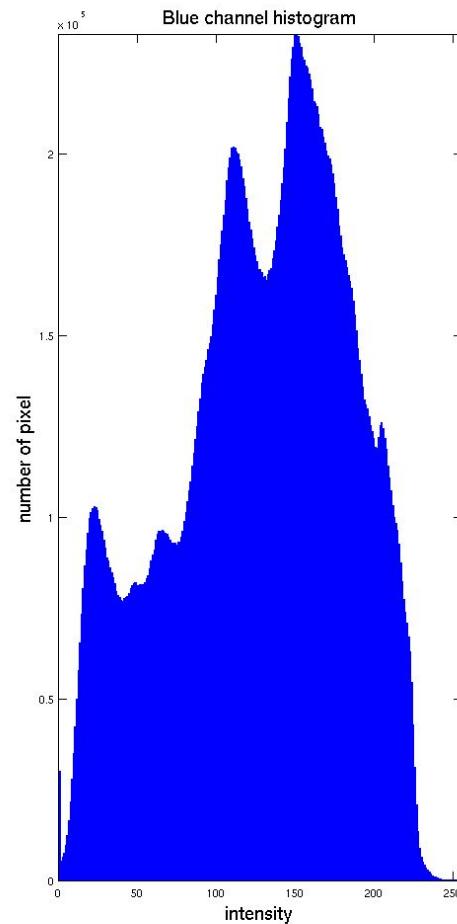
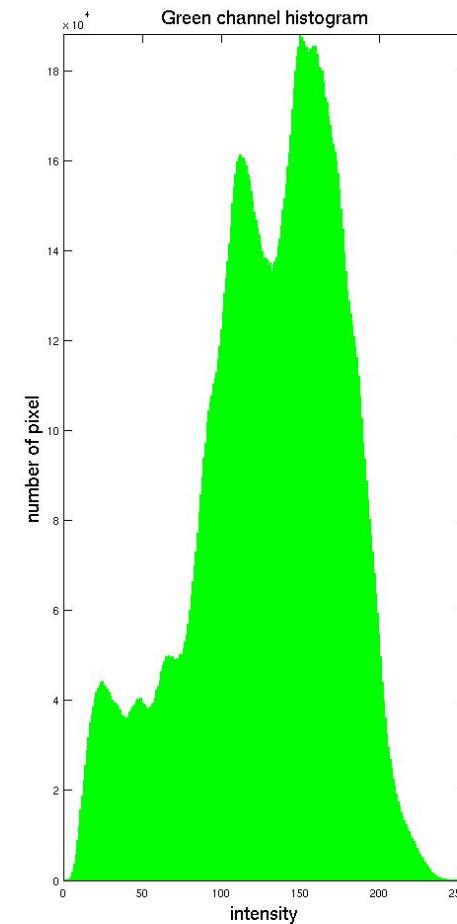
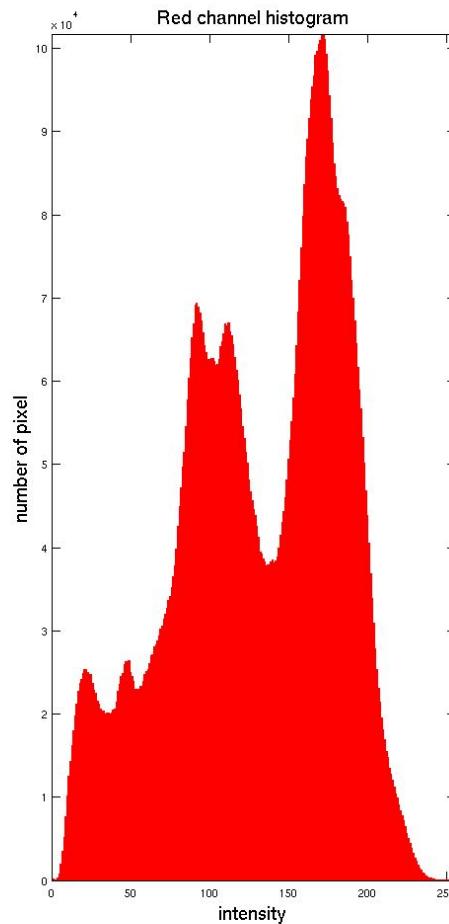
- Hue: rappresenta la gradazione o varietà di colore.
- Colorfulness, saturation e chroma (da non confondere con video chroma): rappresentano l'intensità cromatica.
- Brightness e lightness: rappresentano la percezione di intensità luminosa.
- Contrast: rappresenta la differenza in intensità luminosa o colore all'interno dell'immagine.

Istogramma dei colori

Nella fotografia e nell'image processing, l'istogramma dei colori è una rappresentazione della distribuzione dei colori. Per le immagini digitali rappresenta il numero di pixel in ogni range di colore.

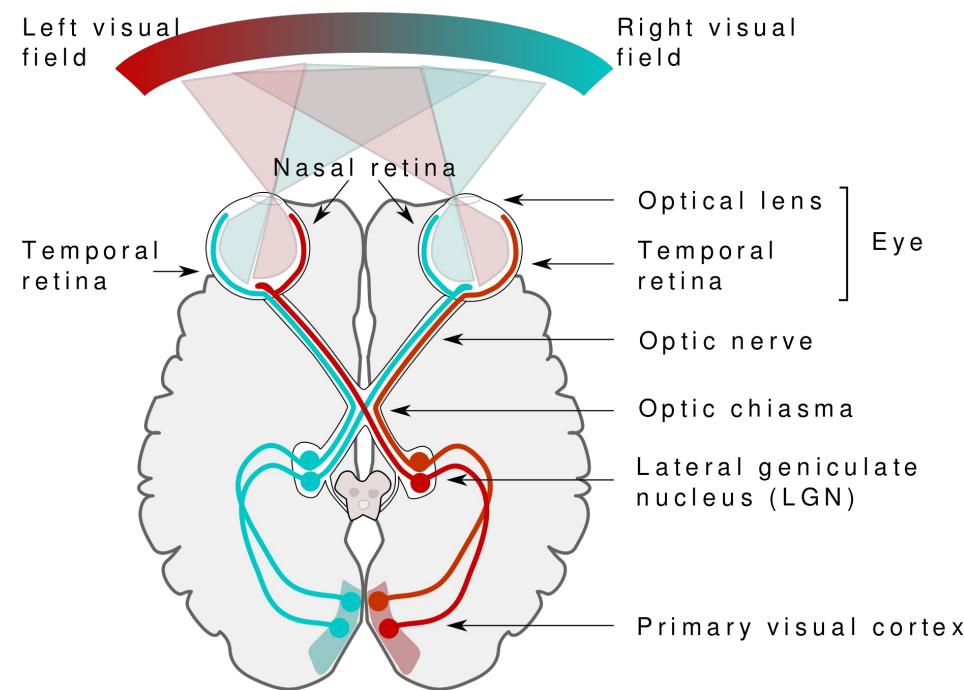
L'istogramma può essere realizzato per ogni modello dei colori, anche se il termine è più comune per gli spazi tridimensionali come RGB o HSV. Per le immagini monocromatiche si usa il termine istogramma di intensità.

Istogramma dei colori



Visione binoculare

La visione binoculare, anche detta visione stereoscopica o stereopsi, è una caratteristica del sistema visivo, propria di alcune specie animali, quali l'essere umano.



Visione binoculare

La visione binoculare avviene in un campo ristretto rispetto alla normale visione bioculare (come ad esempio quella di molto pesci). Nella visione binoculare entrambi gli occhi partecipano alla visione dello stesso soggetto o della stessa area di campo.

Visione binoculare

La parte del singolo campo visivo monoculare dell'occhio destro, va a sommarsi alla stessa parte visibile anche dal singolo campo visivo monoculare dell'occhio sinistro, formando appunto una immagine binoculare.

L'acutezza visiva è maggiore rispetto alla visione monoculare (più del doppio). Il campo visivo è di circa 95° orizzontali e 80° verticali.

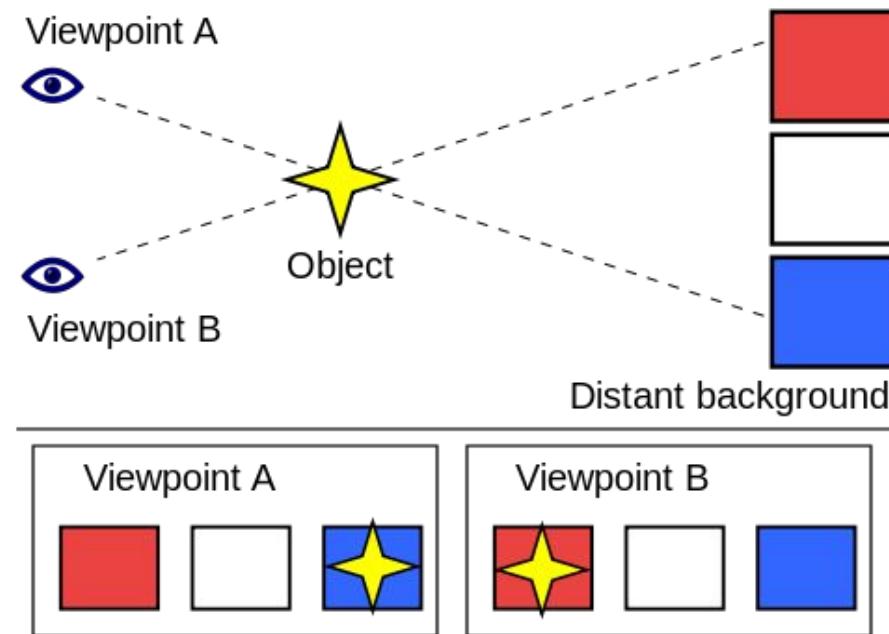
Disparità binoculare

Gli occhi umani sono separati orizzontalmente da una distanza interpupillare di circa 50–75 mm. Di conseguenza, ogni occhio ha una visione leggermente diversa del mondo circostante.

In ogni momento, gli occhi fissano un singolo punto nello spazio. A causa del differente punto di osservazione occhi, gli altri punti nello spazio hanno posizioni diverse sulla retina. La disparità binoculare è la differenza di posizione fra i punti di proiezione. Viene normalmente espressa in gradi (angolo visivo).

Parallasse

Il parallasse è la differenza della posizione apparente di un oggetto visto da due differenti punti di vista.



Percezione della distanza

La percezione della distanza è la capacità di vedere il mondo in 3 dimensioni riuscendo a valutare la distanza degli oggetti. La percezione della distanza deriva da una moltitudine di fattori suddivisi in:

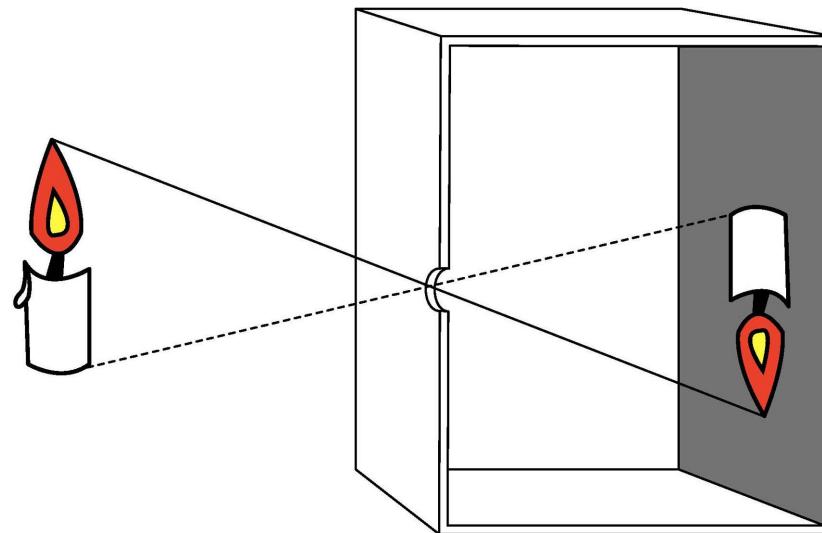
- fattori binoculari: sono principalmente la stereopsi, i movimenti di convergenza degli occhi, la disparità e l'utilizzo del parallasse.
- fattori monoculari: sono ad esempio la dimensione e la granularità dell'oggetto, ma anche l'utilizzo del parallasse di movimento.

Fotografia

La fotografia è la scienza, arte e pratica di creare immagini che durano nel tempo registrando la luce chimicamente con materiale fotosensibile (ad esempio con film fotografico), o elettronicamente con sensori.

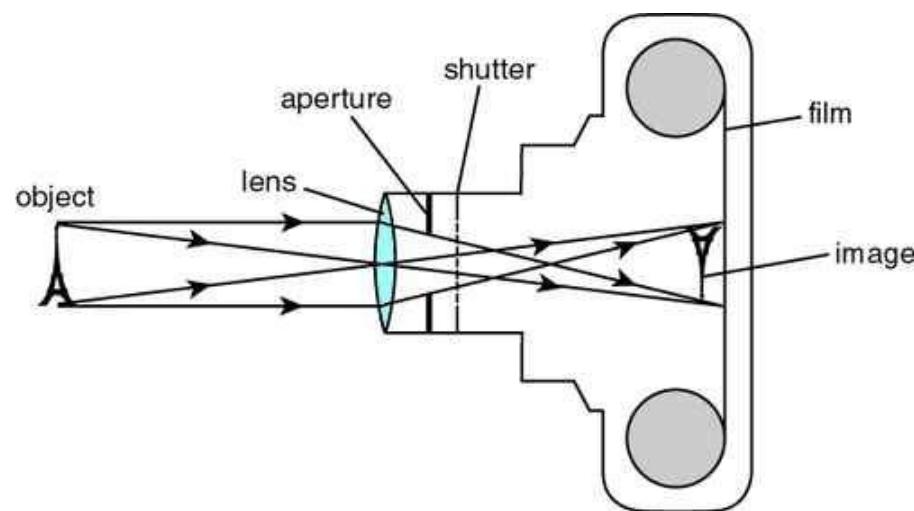
Fotografia

Una lente viene tipicamente utilizzata per la messa a fuoco della luce riflessa o emessa dagli oggetti, in modo da catturarla per un determinato periodo di esposizione, sulla superficie fotosensibile all'interno di una camera.



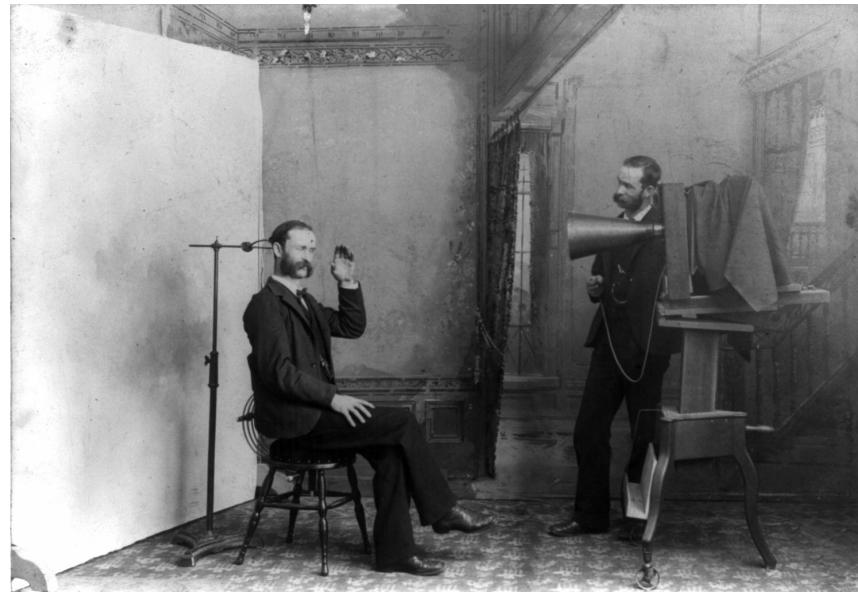
Film fotografico

Il risultato dell'esposizione di un film fotografico è un'immagine invisibile latente, che viene successivamente sviluppata in un'immagine visibile, negativa o positiva a dipendenza dell'utilizzo e del processo di sviluppo utilizzato. L'immagine negativa viene normalmente utilizzata per creare immagini positive ingrandite su carta.



Fotografia, cenni storici

La storia della fotografia ha origini nell'antichità con la scoperta di due principi fondamentali: quella della proiezione di un'immagine all'interno di una camera oscura e il fatto che alcune sostanze possono venire permanentemente alterate dall'esposizione alla luce.



Fotografia, cenni storici

La tecnologia negli anni si è evoluta, con forte accelerazione introdotta dall'avvento delle soluzioni digitali.

Si è passati dall'utilizzo di film fotografico per foto in bianco e nero, successivamente a colori, a tecniche di fotografia digitale ad altissima risoluzione, a 360° o di tipo stereoscopico.

SUPSI

Immagine digitale

Fondamenti di Multimedia Processing

Tiziano Leidi

26.10.2018

Immagine digitale

Un'immagine digitale è una rappresentazione numerica (binaria) di un'immagine bidimensionale. Ne esistono di due tipi: vettoriale o raster (dette anche bitmap).

Solitamente il termine immagine digitale si riferisce alle immagini raster.

Immagine vettoriale

La grafica vettoriale è fondata sull'utilizzo di vettori per la visualizzazione. Vengono quindi usati dei poligoni per rappresentare le immagini.

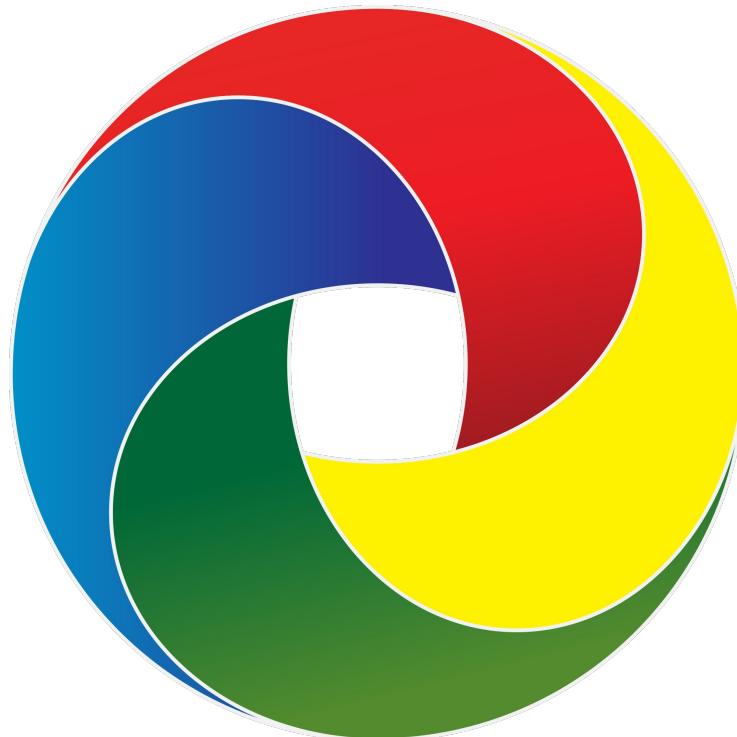
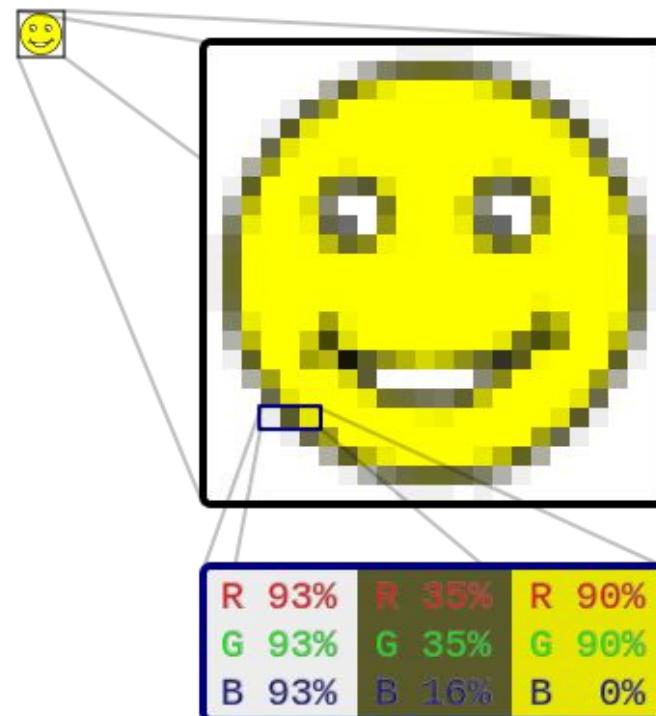


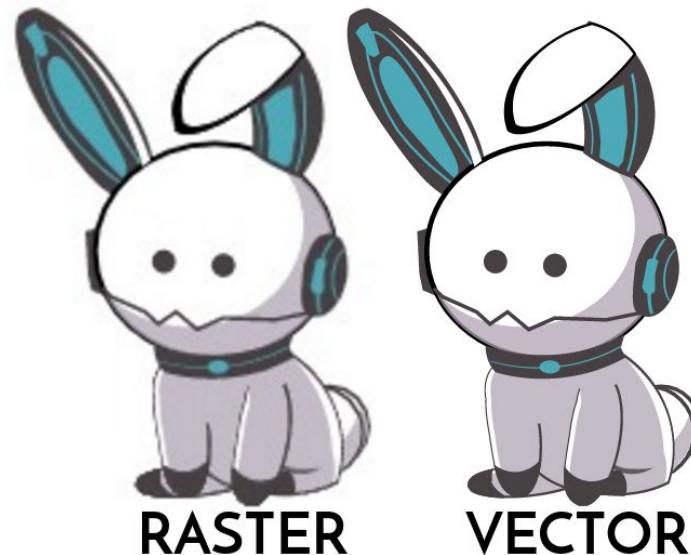
Immagine raster

Nella grafica raster l'immagine viene vista come una scacchiera e ad ogni elemento della scacchiera, chiamato pixel, viene associato uno specifico colore.



Vettoriale vs. raster

Il vantaggio delle immagini vettoriali è che sono indipendenti dalla risoluzione. Invece, le immagini raster non possono essere adattate alla risoluzioni senza perdita di qualità. Le immagini vettoriali hanno però l'inconveniente di essere più laboriose da produrre.



Camera digitale

Le camere digitali (foto o video) permettono l'acquisizione di immagini raster.

Il principio di funzionamento è simile a quello delle fotocamere per film fotografico: un sistema ottico, di norma una lente con diaframma variabile, permette la messa a fuoco della luce, che viene catturata da un sensore di luce elettronico.

All'interno del sensore, una carica elettrica viene prodotta per ogni pixel, successivamente processato elettronicamente ed immagazzinato (ad esempio su file).

Sensori CCD o CMOS

Oggi vengono impiegate 2 tipologie di sensori per le immagini: i semiconductor charge-coupled devices (CCD) e i complementary metal–oxide–semiconductor (CMOS).

Un sensore CCD ha un unico amplificatore per tutti i pixels, mentre in un CMOS è disponibile un amplificatore per ogni pixel.

La maggior parte delle camere digitali ha sensori CMOS perché offrono migliori prestazioni, sono più veloci e consumano meno.

Invece, sensori CCD vengono ancora utilizzati per le camere digitali a basso costo.

Sensori CCD o CMOS

<https://www.youtube.com/watch?v=MubI-KUOAPU>

Campionatura e quantizzazione

Se si vuole convertire un'immagine analogica in digitale è necessario discretizzare l'immagine. Strumenti come i frame grabber permettono di eseguire queste operazioni.

L'immagine deve essere discretizzata sia spazialmente (sampling), sia in termini di intensità di luce (quantization). Sono quindi 2 i fattori che influenzano la digitalizzazione:

- Il sampling-rate: determina la risoluzione spaziale,
- Il livello di quantizzazione: determina il numero di tonalità di grigio o livelli di colore che vengono acquisiti.

Campionatura e quantizzazione

Un'immagine è una funzionale dello spazio:

$$f : \mathbb{R}^m \longrightarrow \mathbb{R}, m = 2, 3$$

che associa ad ogni posizione spaziale un valore d'intensità di grigio, ...

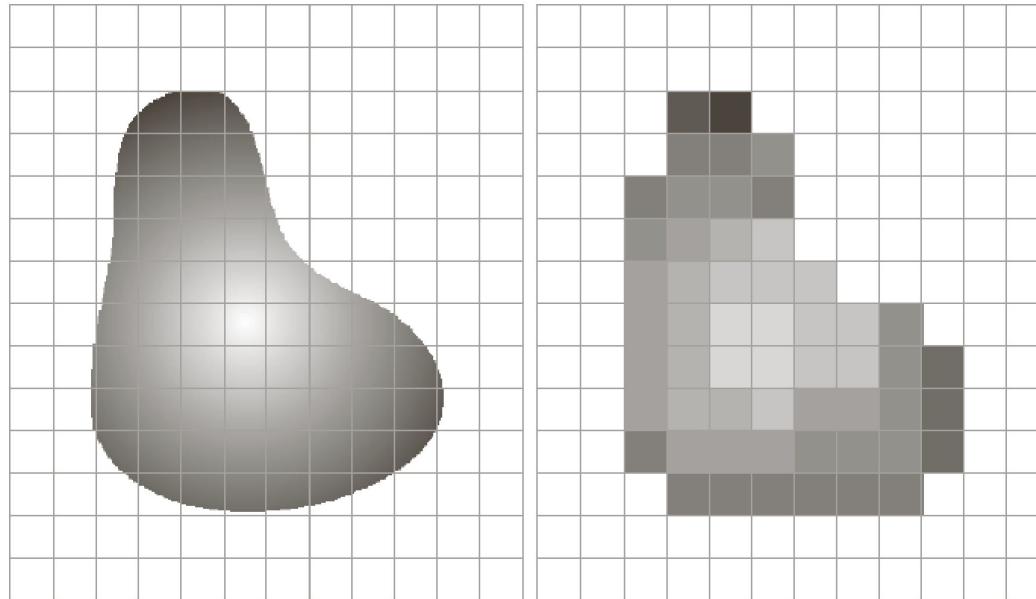
$$f : \mathbb{R}^m \longrightarrow \mathbb{R}^3, m = 2, 3$$

oppure i valori di un colore per un determinato modello dei colori.

Campionatura

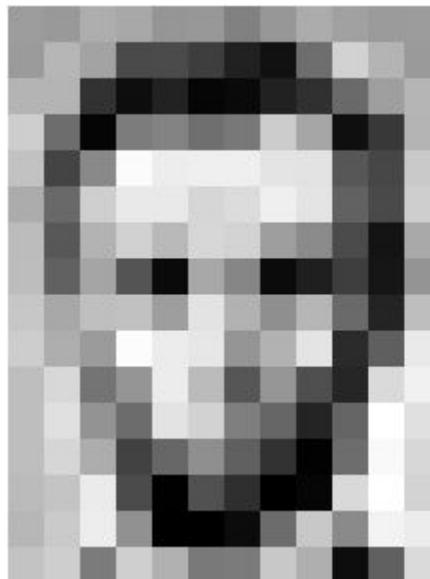
Un'immagine fa parte del mondo reale, quindi è analogica e la sua funzione è continua.

Il processo di campionatura la discretizza nello spazio, trasformandola in una matrice di pixel (voxel nel caso 3D).



Quantizzazione

Il processo di quantizzazione si occupa invece della discretizzazione dell'intensità luminosa, restituendo un valore nel caso delle tonalità di grigio, rispettivamente tre o più valori nel caso di un modello dei colori.



157	153	174	168	150	152	129	151	172	161	155	166
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	84	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	157	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	167	86	150	79	38	218	241
190	224	147	108	227	210	127	102	95	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Risoluzione spaziale

La risoluzione spaziale è una misura del più piccolo dettaglio discernibile in un'immagine digitale. Per misurarla vengono utilizzate unità di misura come il dots per inch (DPI).

Affermare che un'immagine ha una risoluzione ad esempio di 1024x1024 pixels non ha un effettivo significato, se la risoluzione non è messa in relazione ad un'unità di misura spaziale.

Risoluzione di intensità

La risoluzione d'intensità si riferisce al più piccolo dettaglio discernibile nel livello di intensità di grigio o colore. Per le immagini grayscale vengono d'abitudine utilizzati 8 bits (256 valori), mentre per le immagini a colori si usano 16 bits. L'utilizzo di 32 bits è raro.

Colori nelle schede video

Le schede video dei primi PC supportavano solo 16 colori fissi. Successivamente sono state introdotte schede capaci di mostrare 256 colori simultaneamente, ognuno dei quali poteva essere scelto da un pool di 16 milioni di colori.

Le attuali schede video dedicano 24 bits ad ogni pixel e sono quindi capaci di mostrare 16 milioni di colori senza restrizioni.

24 bit (16 million colors) è chiamato True Color.

Aliasing

Come per l'audio, anche nella campionatura di immagini ci si può confrontare con il problema dell'aliasing: l'introduzione di distorsione o artefatti durante la ricostruzione di un segnale che, prima di venir campionato, conteneva oggetti più piccoli della metà del sampling-rate (Nyquist).

Nelle immagini, la frequenza è in relazione alla dimensione strutturale. Parti piccole hanno frequenza alta.



Aliasing

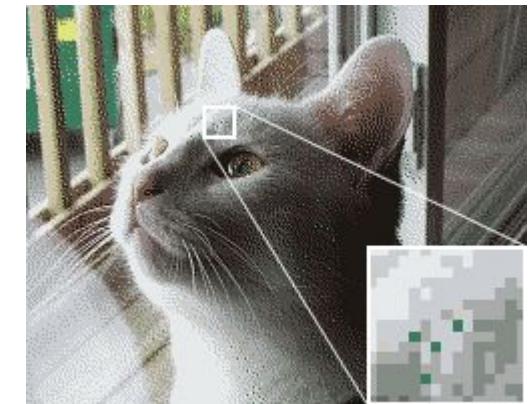
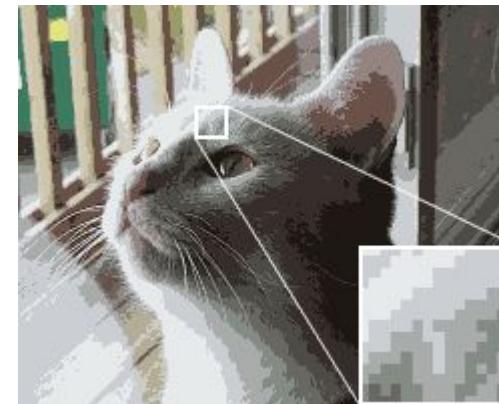
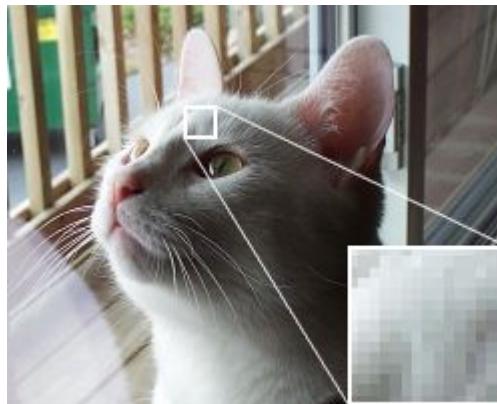
Nel caso delle immagini si tratta di aliasing spaziale (a differenza di quello temporale che si può presentare per i segnali audio e video). Per evitarlo vanno applicati filtri di anti-aliasing.



Errori di quantizzazione e dither

Anche il processo di quantizzazione introduce errori, in forma di rumore, che può presentarsi come patterns a larga scala (ad esempio bande di colore).

Il dither è una forma di rumore introdotta intenzionalmente per rendere casuale l'errore di quantizzazione.



Interpolazione

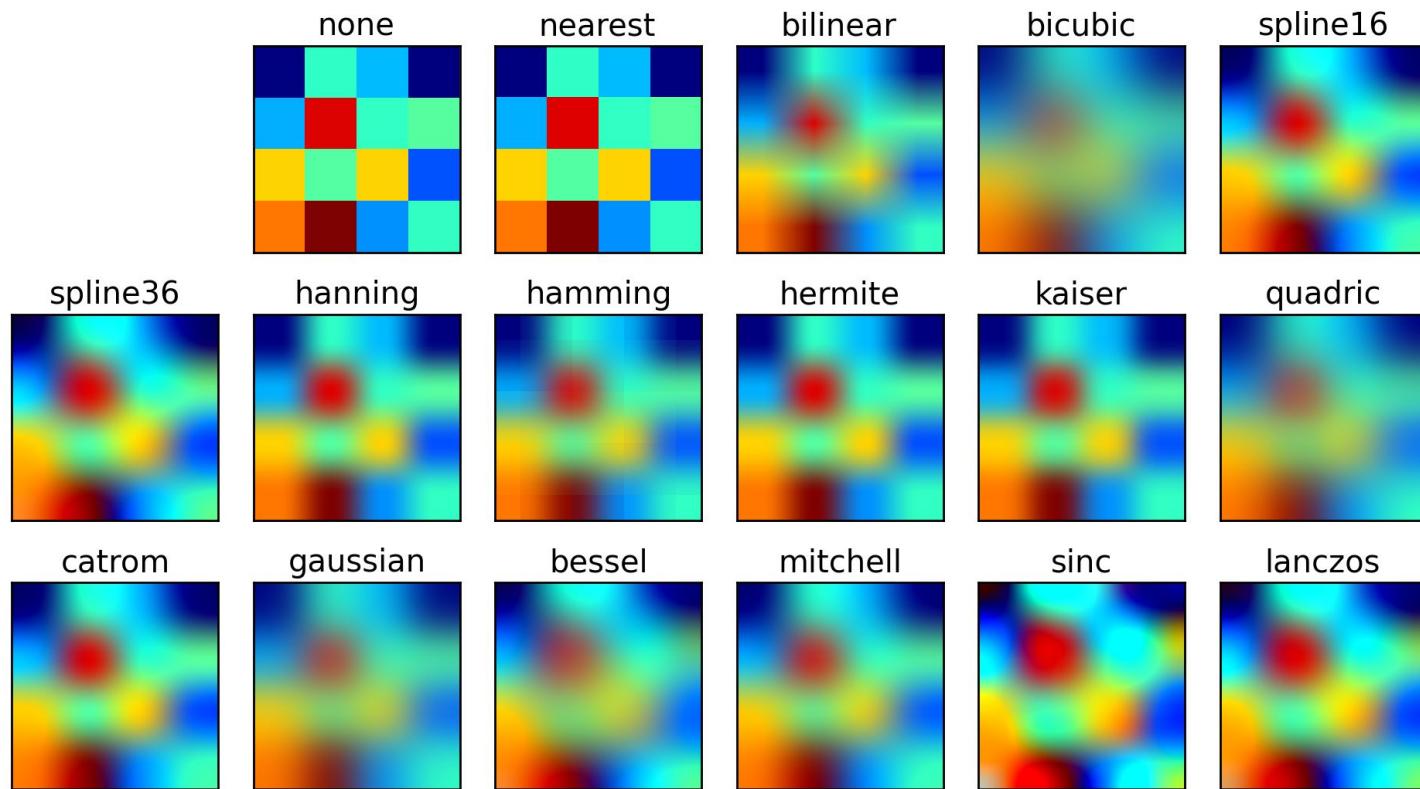
L'interpolazione è un'operazione spesso utilizzata per le immagini per eseguire zoom, rotazioni, e altre correzioni geometriche.

L'interpolazione viene eseguita utilizzando informazione conosciuta in determinate posizioni per stimare i valori di posizioni sconosciute.

Esistono vari metodi di interpolazione, dalla nearest neighbour interpolation a forme più complesse, che sfruttano ad esempio le splines.

<http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>

Interpolazione



Relazioni fra pixels

Esistono alcune relazioni fra pixel che sono state definite, come ad esempio:

- Vicinanza (neighborhood): ogni pixel ha 2 vicini verticali, 2 vicini orizzontali e 4 vicini diagonali
- Adiacenza (adjacency): due pixels sono adiacenti se sono vicini e se rispettano determinati vincoli di prossimità di grigio o colore

Esistono anche altre proprietà che possono essere presenti fra pixels come la connectivity, le paths, le regions e i boundaries.

Distanza e operazioni fra pixels

Inoltre, esistono misure di distanza tra pixel come l'euclidean distance, la city-block distance e la chess-board distance.

Infine, sono state definite operazioni fra pixel come somma, sottrazione, moltiplicazione, divisione, AND, OR e complemento.

Formati file per immagini digitali

I principali formati file per le immagini raster sono .gif, .jpg, .tiff, .png e .bmp.

I files per le immagini digitali possono essere di dimensioni molto grandi, quindi, come per l'audio, vengono applicate tecniche di compressione per ridurne la taglia. La compressione può essere di tipo lossless o lossy.

Riduzioni di taglia dei files sono possibili anche sulla base del numero di colori contenuti nelle immagini. Nei vari tipi di formati vengono usati differenti tipi di approcci per la codifica dei colori.

Header e container

Come nel caso dei formati file audio, anche nel caso delle immagini i file sono composti da una prima parte di metà informazione, contenente le caratteristiche dell'informazione salvata, seguita dagli effettivi dati dell'immagine.

Anche nel caso delle immagini si possono usare dei formati container flessibili e generici. Il formato TIFF è in verità un formato container specifico per le immagini.

GIF

Il formato .gif crea una tabella fino a 256 colori da un pool di 16 milioni. Se l'immagine ha meno di 256 colori, l'algoritmo è capace di riprodurre l'immagine in maniera esatta. Per i casi in cui l'immagine contiene più di 256 colori, alcuni algoritmi usano il colore più vicino, mentre altri sfruttano tecniche come l'“error diffusion” per minimizzare l'errore.

Inoltre, GIF sostituisce eventuali patterns (ampie zone di colore uniforme) con delle abbreviazioni (come bianco x 20).

Di conseguenza, GIF è lossless solo per immagini con 256 colori o meno. Per immagini True Color può perdere il 99.998% dei colori.

JPG

JPG è un formato ottimizzato per la fotografia e per immagini contenenti molti colori simili in zone contigue. Può raggiungere livelli di compressione molto alti pur mantenendo qualità d'immagine molto elevate.

JPG lavora scartando informazione che l'occhio umano non dovrebbe notare. Salva l'informazione come colore a 24 bit. Il tipo di compressione è lossy, il livello di compressione è adattabile.

TIFF, PGN e BMP

TIFF è un formato flessibile che può essere sia lossless che lossy, ma viene principalmente utilizzato senza nessun tipo di compressione. Eventuali dettagli dell'algoritmo utilizzato vengono salvati con il file.

PNG è un formato lossless. Riduce la taglia del file cercando dei patterns che può sfruttare per semplificare il salvataggio dell'informazione. La compressione è di conseguenza perfettamente reversibile.

BMP è un formato proprietario non compresso inventato da Microsoft.

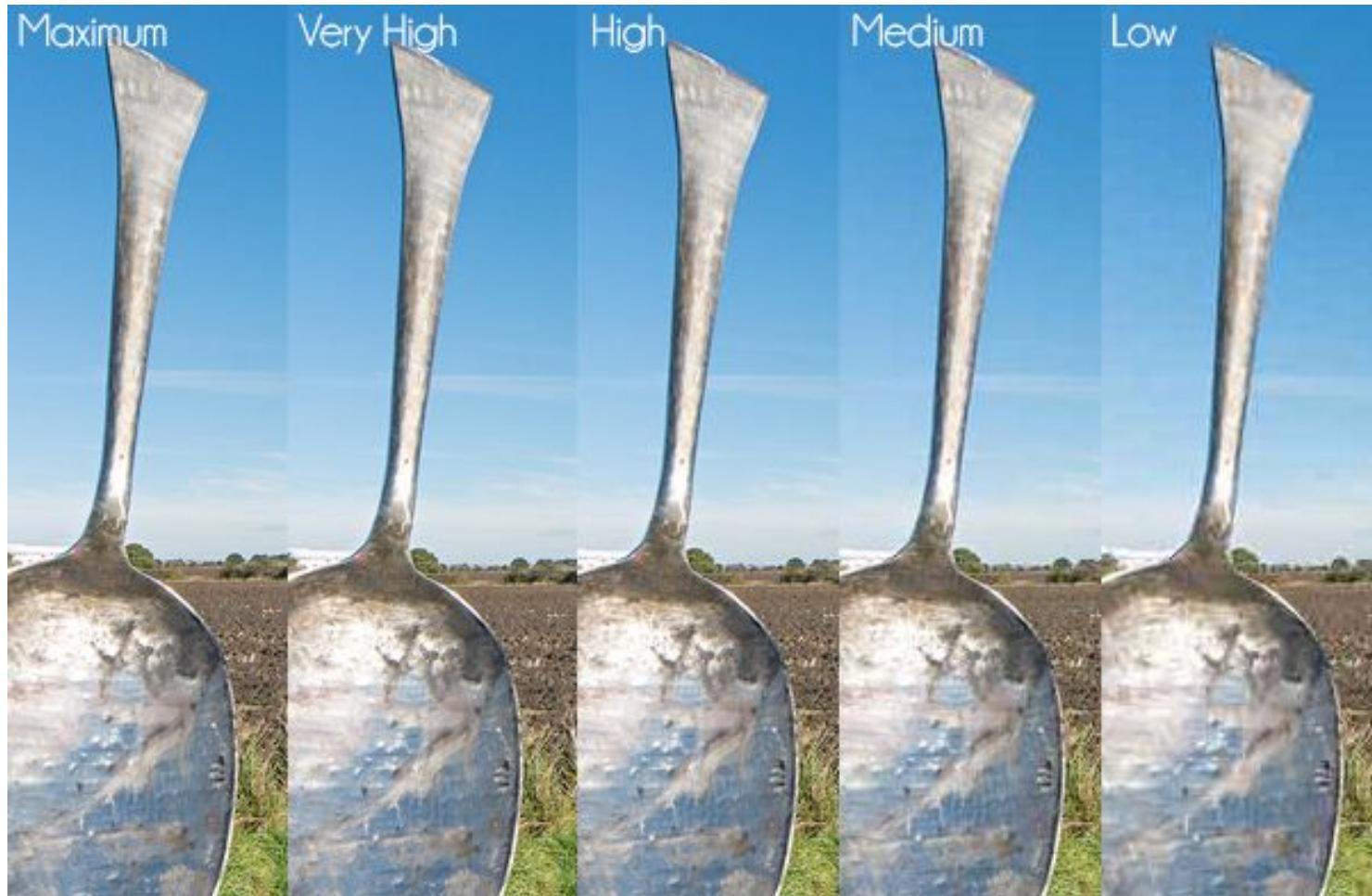
Confronto di dimensione file

File type	Size
Tiff, uncompressed	901K
Tiff, LZW lossless compression (yes, its actually bigger)	928K
JPG, High quality	319K
JPG, medium quality	188K
JPG, my usual web quality	105K
JPG, low quality / high compression	50K
JPG, absurdly high compression	18K
PNG, lossless compression	741K
GIF, lossless compression, but only 256 colors	286K

GIF vs. JPEG



Qualità di compressione JPEG



Strumenti matematici per l'Image processing

Per eseguire operazioni di digital image processing vengono sfruttati alcuni strumenti matematici di base, come:

- Array e matrici
- Operazioni lineari vs. nonlineari
- Operazioni aritmetiche
- Operazioni su set e operazioni logiche
- Operazioni spaziali
- Operazioni vettoriali e matriciali
- Trasformazioni
- Metodi probabilistici

Operazioni di tipo array vs. matriciali

Le immagini digitali raster possono essere viste come matrici. Quindi su di esse è possibile eseguire sia operazioni di tipo array, sia matriciali.

Ad esempio, il prodotto di due matrici può essere inteso come il prodotto array (ogni elemento delle due matrici viene moltiplicato), oppure come l'effettivo prodotto matriciale.

Di norma, nel digital signal processing si intendono le operazioni come operazioni di tipo array.

Operazioni lineari vs. nonlineari

Anche nel caso dell'image processing si fa distinzione fra operazioni lineari e nonlineari.

Si consideri due immagini A e B. Per un filtro lineare, come la media sia ha che:

$$F_m(A + \lambda B) = F_m(A) + \lambda F_m(B)$$

Questa proprietà non viene invece soddisfatta dai filtri non lineari, come ad esempio per la mediana.

Linear image processing

Le operazioni di linear image processing sono basate, come per il signal processing convenzionale, su due operazioni principali: la convoluzione e la trasformata di Fourier.

Nell'image processing la convoluzione è l'operazione più importante perchè le immagini hanno l'informazione codificata nel dominio spaziale invece che in quello delle frequenze.

Filtri lineari possono ad esempio venir utilizzati per:
accentuare gli spigoli degli oggetti, ridurre il rumore di fondo,
correggere l'illuminazione, ecc.

Queste operazioni vengono eseguite per convoluzione tra
l'immagine originale e un kernel di filtro, producendo
l'immagine filtrata.

Analisi di Fourier

L'analisi di Fourier è meno utile per le immagini. Mentre per un segnale audio la trasformata di Fourier trasforma la dubbia rappresentazione nel dominio del tempo in una spettro delle frequenze facile da capire, la trasformata di Fourier di un'immagine converte l'informazione chiara del dominio spaziale in una forma poco intellegibile nel dominio delle frequenze.

SUPSI

Tecniche base di image processing

Fondamenti di Multimedia Processing

Tiziano Leidi

09.11.2017

Digital image processing

Esistono una moltitudine di tecniche di digital image processing.

Le più comuni si possono suddividere in:

- Trattamento dei colori dell'immagine
- Ricostruzione e ristrutturazione dell'immagine
- Segmentazione
- Operazioni di morfologia matematica
- Riconoscimento di forme e oggetti

Equalizzazione dell'istogramma

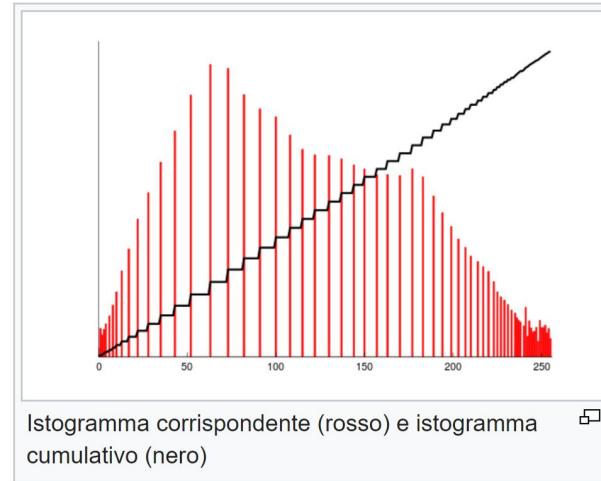
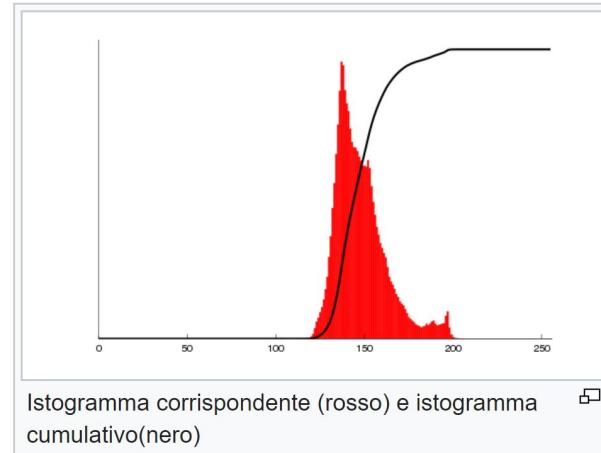
L'equalizzazione dell'istogramma è un metodo dell'image processing per l'adeguamento del contrasto.

L'obiettivo è di ridistribuire le intensità nell'istogramma, fornendo, come conseguenza, maggiore contrasto a zone che hanno basso contrasto locale.

Può essere applicato sia a immagini in scale di grigio, sia a immagini a colori.

L'equalizzazione dell'istogramma può produrre risultati poco realistici nella fotografia. È molto efficace nel caso di immagini scientifiche.

Equalizzazione dell'istogramma



Normalizzazione dell’immagine

La normalizzazione è un processo affine all’equalizzazione dell’istogramma, in cui vengono modificate le intensità dei pixels (estendendone il range) per ottenere una migliore occupazione del range dinamico disponibile.

Viene anche chiamato “contrast stretching” o “histogram stretching”, più in generale anche “dynamic range expansion”.

Filtri digitali tramite convoluzione o DFT

Tecniche di convoluzione o di DFT possono essere utilizzate per eseguire varie operazioni sulle immagini, in particolare operazioni di sfocatura (blur) e messa a fuoco (sharpening).

Il filtraggio può essere eseguito nel dominio spaziale tramite convoluzione, applicando dei kernels, o nel dominio delle frequenze, tramite la trasformata di Fourier (DFT), modificando specifiche regioni di frequenze.

Gaussian blur

Nell'image processing una sfocatura gaussiana (anche detta Gaussian smoothing) si ottiene applicando all'immagine una funzione gaussiana tramite convoluzione.

Viene spesso utilizzata per ridurre il rumore e la presenza di dettagli nelle immagini.



box blur

Un box blur (anche conosciuto come box linear filter) è un filtro lineare nel dominio spaziale, in cui ogni pixel dell'immagine risultante ha un valore equivalente alla media dei suoi vicini nell'immagine di partenza.

Si tratta di un filtro low-pass, che può essere scritto come matrice determinante 3x3 moltiplicata per 1/9. I box blurs vengono spesso utilizzati per approssimare i Gaussian blur.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening

Un'immagine è a fuoco se ha buona risoluzione e un buon contrasto sui contorni (edges). Più l'immagine ha risoluzione, meglio può essere messa a fuoco. Per quanto concerne il contrasto dei bordi, si tratta di un fattore soggettivo. Può essere migliorato applicando filtri che migliorano il contrasto dei contorni.

Sharpening



Convoluzione

Nel trattamento d'immagine, la convoluzione è il processo di aggiungere ogni elemento ai propri vicini, moltiplicato per i valori specificati in un kernel di dimensione 3x3.

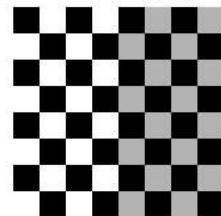
Tramite questa semplice operazione, si implementa una forma di convoluzione matematica.

Va sottolineato che la moltiplicazione da eseguire è un array multiplication, non una moltiplicazione matriciale tradizionale.

Kernel

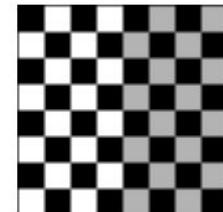
I kernel usati nella convoluzione vengono anche detti matrici di convoluzione o maschere. Possono essere utilizzati per implementare una moltitudine di filtri, incluse alcune soluzioni per la detezione dei contorni.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



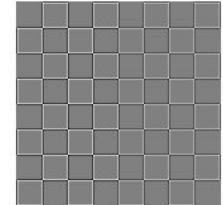
Identità

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Originale

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Highpass

3x3 mean blur

Laplacian
edge detection

Normalizzazione del kernel

La normalizzazione dei valori di un kernel è definita come la divisione di ogni elemento del kernel per la somma di tutti gli elementi del kernel, in modo che la somma di tutti i valori all'interno del kernel dia 1.

Questa soluzione garantisce che i pixels siano in media luminosi nell'immagine risultante quanto lo sono nell'immagine originale.

Detezione dei contorni

Il riconoscimento dei contorni (edge detection) ha lo scopo di marcare i punti in cui l'intensità luminosa cambia bruscamente. L'operazione genera immagini contenenti molte meno informazioni rispetto a quelle originali, eliminando la maggior parte dei dettagli non rilevanti alla descrizione della forma e delle caratteristiche strutturali e geometriche degli oggetti.



Trasformata di Fourier 2D

Visto che le immagini contengono dati nel dominio dello spazio (si tratta di uno spazio bi-dimensionale) invece che in quello del tempo, per l'image processing viene solitamente utilizzata la trasformata di Fourier di tipo 2D.



Trasformata di Fourier 2D

Come nell'audio, anche la trasformata di Fourier 2D sfrutta funzioni armoniche per eseguire la decomposizione spettrale:

$$\begin{aligned} F(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy, \\ f(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv \end{aligned}$$

F è il risultato (lo spettro) della trasformata di Fourier dell'immagine f .

L'operazione inversa permette di ricostruire l'immagine a partire dallo spettro.

Trasformata di Fourier 2D

I parametri u e v , vengono chiamate frequenze spaziali.

$F(u,v)$ è complesso: $F(u, v) = F_R(u, v) + jF_I(u, v)$.

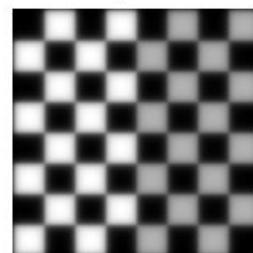
Il valore assoluto è lo spettro di magnitude: $|F(u, v)|$

In maniera analoga si può definire la DFT 2D da applicare alle immagini raster nell'ambito del digital image processing.

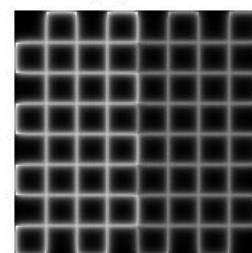
Trasformata di Fourier 2D

Il fatto che la modellazione eseguita dalla DFT sfrutti segnali periodici può presentare delle complicazioni nel caso delle immagini, visto che non presentano necessariamente periodicità.

In maniera simile alla convoluzione, La DFT 2D viene utilizzata per applicare varie tipologie di filtri.



Lowpass



Highpass

Denoise

Immagini acquisite con camere digitali o convertite da film fotografico convenzionale, contengono rumore introdotto da una moltitudine di fattori, come ad esempio i circuiti elettrici, rispettivamente i sensori in caso di illuminazione insufficiente o eccessi di temperatura.

Molto frequentemente il processing di queste immagini richiede la rimozione (almeno parziale) di questo rumore, sia per questioni estetiche, ma anche per questioni prettamente pratiche, come nel caso della computer vision.

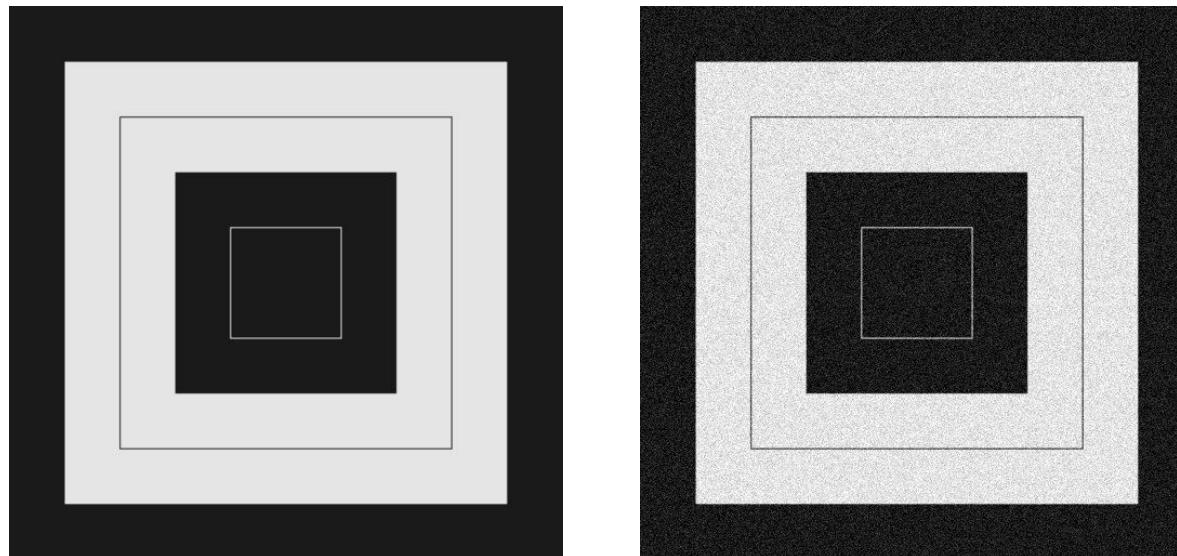
Rumore salt-and-pepper

Il rumore di tipo salt-and-pepper, anche conosciuto come rumore impulsivo, causa disturbi improvvisi e netti all'interno dell'immagine. Si presenta di norma come occorrenze sparse di pixels neri o bianchi. Un metodo efficace per la sua riduzione è l'utilizzo di un filtro mediano.



Rumore gaussiano

Il rumore gaussiano ha una distribuzione statistica di tipo normale. Un caso speciale è quello del rumore bianco, in cui i valori sono tutti indipendenti e distribuiti in maniera identica. Una delle tecniche convenzionali per la rimozione del rumore gaussiano è quella della media tramite convoluzione.



Restoration: Wiener filter

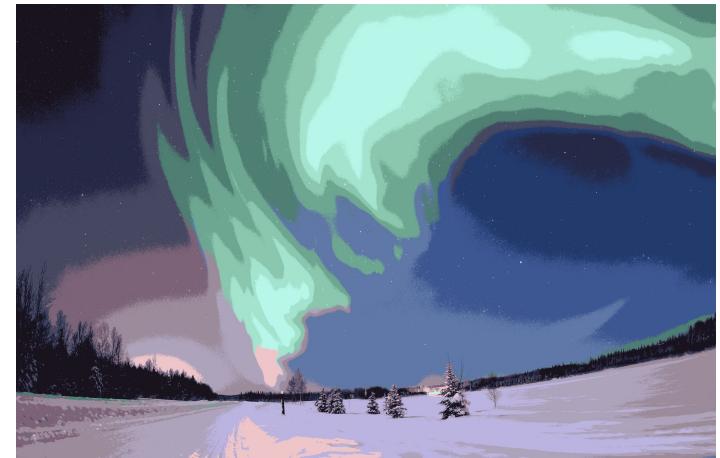
I filtri più tradizionali eseguono la separazione tra segnale e rumore a condizione che questi occupino diverse bande di frequenza. Il filtro di Wiener supera questa limitazione con un approccio statistico.

Si assume di avere conoscenza delle caratteristiche spettrali del segnale originale e del rumore, e si cerca il filtro LTI il cui risultato sia "il più vicina possibile" al segnale originale (minimizzando una prefissata misura dell'errore).

Questo filtro è spesso utilizzato nel processo di deconvoluzione.

Segmentazione

La segmentazione è il processo in cui si suddivide un'immagine digitale in più segmenti (sets di pixel). L'obiettivo è quello di semplificare/modificare la rappresentazione dell'immagine in qualcosa che sia più facile e sensato da analizzare. La segmentazione viene normalmente utilizzata per identificare forme e contorni.



Thresholding

Il metodo più semplice per eseguire una segmentazione è chiamato metodo di sogliatura (thresholding).

Nella sua versione più essenziale, un'immagine in scale di grigi viene convertita in un'immagine con pixels solo bianchi o neri. Il valore usato come discriminante è la soglia (threshold).



Thresholding

Esistono molte tecniche utilizzate per il thresholding, come ad esempio il thresholding bilanciato tramite istogramma, il maximum entropy method, l'Otsu's method (maximum variance) e il k-means clustering.

Thresholding basato sull'istogramma

I metodi di thresholding che sfruttano l'istogramma, sono normalmente efficaci rispetto agli altri metodi, perché necessitano di un solo passaggio di processing sui pixels.

Prima di cominciare il processing, è necessario acquisire un istogramma dell'immagine. Picchi e valli dell'istogramma vengono utilizzati per identificare i clusters all'interno dell'immagine. Può funzionare sia sulla base delle intensità di grigio, sia sulla base dei colori.

K-means

L'algoritmo di K-means sfrutta una tecnica iterativa per partizionare l'immagine in K clusters.

L'algoritmo di base è il seguente:

- Scegli K centri dei clusters (a caso o in base ad un'euristica).
- Assegna ogni pixel dell'immagine al cluster che minimizza la distanza fra il pixel e il centro del cluster.
- Ricalcola il centro del cluster posizionandolo nel punto medio di tutti i pixels del cluster.
- Ripeti il passo 2 e il passo 3 fino a convergenza (nessun pixel cambia cluster).

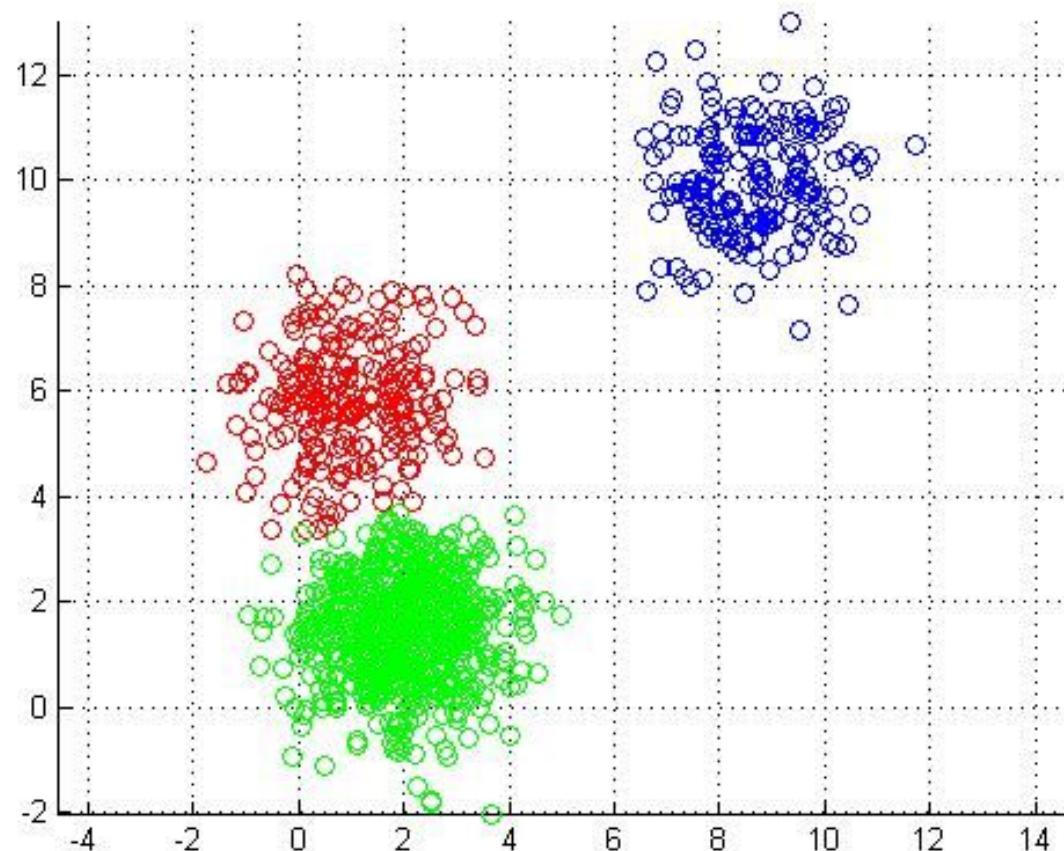
K-means

La distanza viene normalmente calcolata come la differenza assoluta o al quadrato fra un pixel e il centro del cluster.

La differenza può essere basata sulla posizione, l'intensità, il colore, la texture o una combinazione pesata di questi fattori.

È garantito che l'algoritmo converga, ma non è detto che restituisca la soluzione ottimale. La qualità della soluzione dipende dal set iniziale di clusters e dal valore di K.

K-means



Morfologia matematica

La morfologia matematica è una teoria e tecnica per l'analisi e il processing di strutture geometriche fondata sulla teoria degli insiemi, dei reticoli, sulla topologia e sulle funzioni random.

La morfologia matematica viene principalmente applicata alle immagini digitali, ma può essere impiegata anche sui grafi, sulle mesh di superficie, sui solidi e su molte altre strutture spaziali.

Morphological image processing

La morfologia matematica sta alla base delle tecniche di morphological image processing, che consistono nell'impiego di alcuni operatori per trasformare le immagini (solitamente di tipo binario, ma si può applicare anche ad immagini in scale di grigio). I quattro operatori principali sono: erosioni, dilatazione, apertura e chiusura.

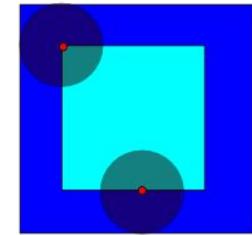
L'idea di base della morfologia binaria è di testare l'immagine con una struttura semplice predefinita, per verificare se questa struttura è compatibile o incompatibile con le forme presenti nell'immagine. La struttura di test viene chiamata structuring element.

Erosione

$$A \ominus B = \{z \in E | B_z \subseteq A\} = \bigcap_{b \in B} A_{-b}$$

$$B_z = \{b + z | b \in B\}, \forall z \in E$$

Traslazione di B tramite il vettore z

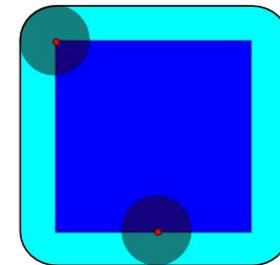


In uno spazio euclideo E , L'erosione dell'immagine A con lo structuring element B è la combinazione di tutti i punti che può raggiungere il centro di B quando si muove all'interno di A .

Esempio: l'erosione di un quadrato di dimensione 10 con un disco di raggio 2 è un quadrato di dimensione 6.

Dilatazione

$$A \oplus B = \bigcup_{b \in B} A_b = \bigcup_{a \in A} B_a$$

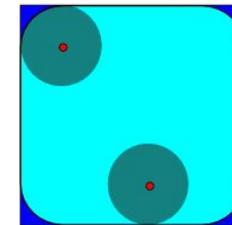


La dilatazione dell'immagine A con lo structuring element B è la combinazione di tutti i punti che può raggiungere B quando il centro di B si muove all'interno di A .

Esempio: la dilatazione di un quadrato di dimensione 10 con un disco di raggio 2 è un quadrato di dimensione 14 con gli angoli arrotondati.

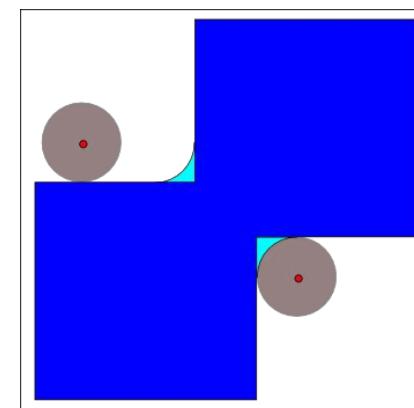
Apertura e chiusura

$$A \circ B = (A \ominus B) \oplus B$$



L'apertura è la traslazione dello structuring element B all'interno di A .

$$A \bullet B = (A \oplus B) \ominus B$$



La chiusura è il complemento della traslazione dello structuring element B all'esterno dall'immagine A .

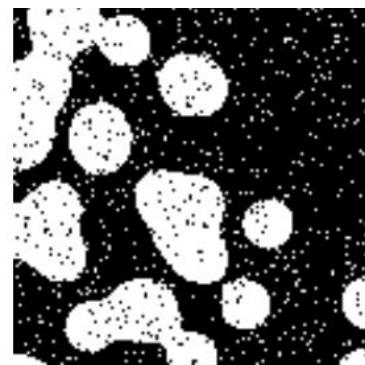
Morphological image processing

L'erosione permette di rendere più regolari i bordi poco precisi, riducendo la dimensione delle forme. Permette anche di individuare eventuali buchi all'interno delle forme.

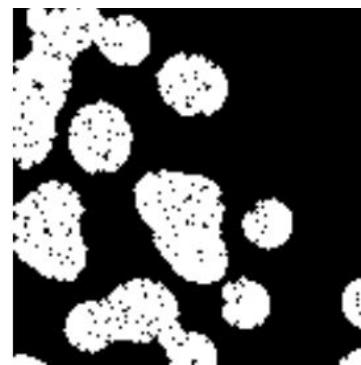
La dilatazione permette di rendere più spesse le forme fini e di coprire eventuali buchi.

Morphological image processing

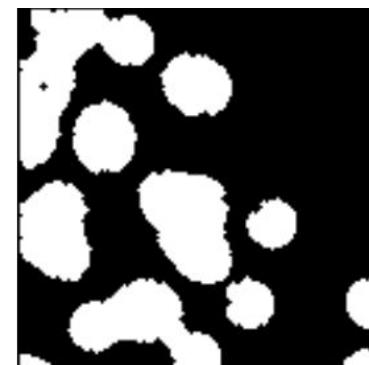
L'apertura permette di ristrutturare le figure rimuovendo le imperfezioni sui bordi e i piccoli buchi. La controparte è che arrotonda le forme. La chiusura permette di isolare le forme rimuovendo eventuale rumore di fondo.



Originale



Apertura



Chiusura

SUPSI

Introduction to GPUs and CUDA

Tiziano Leidi

Fondamenti di Multimedia Processing
Bachelor in Ingegneria Informatica

Multimedia Processing

Today, multimedia contents are practically all in digital form. The acquisition, transfer, modification and archiving of multimedia information represents a big challenges in many application domains.

Innovative technologies are continuously required to allow processing the growing amount of multimedia data at real-time. Multimedia processing is constantly hungry of computing power.

Microprocessors

- Miniaturization of components is now reaching the molecular level.
- The smallest "linewidth" (width of feature on the chip surface) shrank from **two microns** in 1980 to less than **a tenth micron** (100 nanometers) a quarter-century later.
- If viewed in cross section, the thickness of horizontal layers of material deposited on the silicon surface is currently less than **1 nanometers**.
- *The width of a human hair is about 100 microns and the width of a molecule is about 1 nanometer.*

Semiconductor manufacturing processes

10 µm – 1971

6 µm – 1974

3 µm – 1977

1.5 µm – 1982

1 µm – 1985

800 nm – 1989

600 nm – 1994

350 nm – 1995

250 nm – 1997

180 nm – 1999

130 nm – 2001

90 nm – 2004

65 nm – 2006

45 nm – 2008

32 nm – 2010

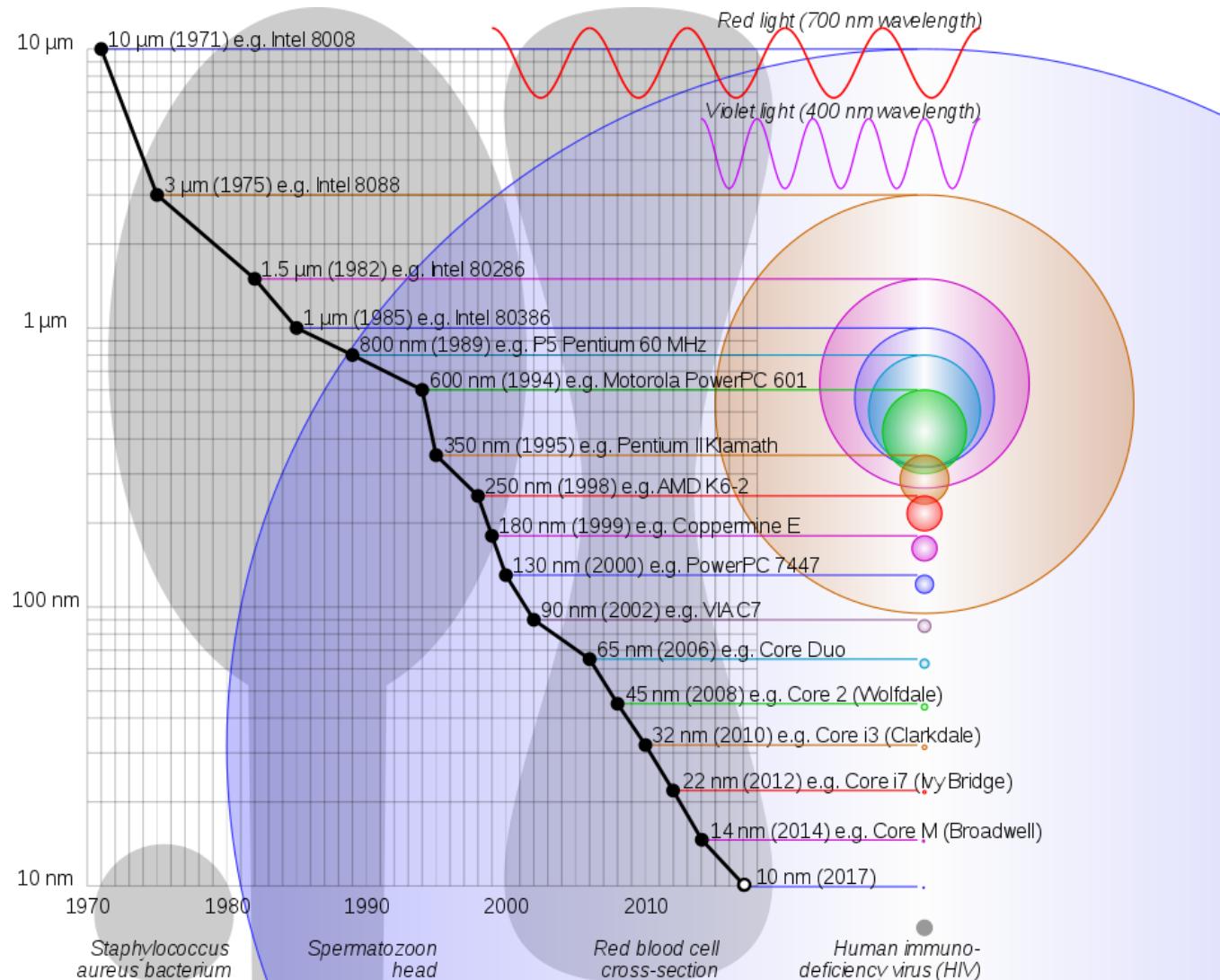
22 nm – 2012

14 nm – 2014

10 nm – 2016

7 nm – 2018

5 nm – 2020

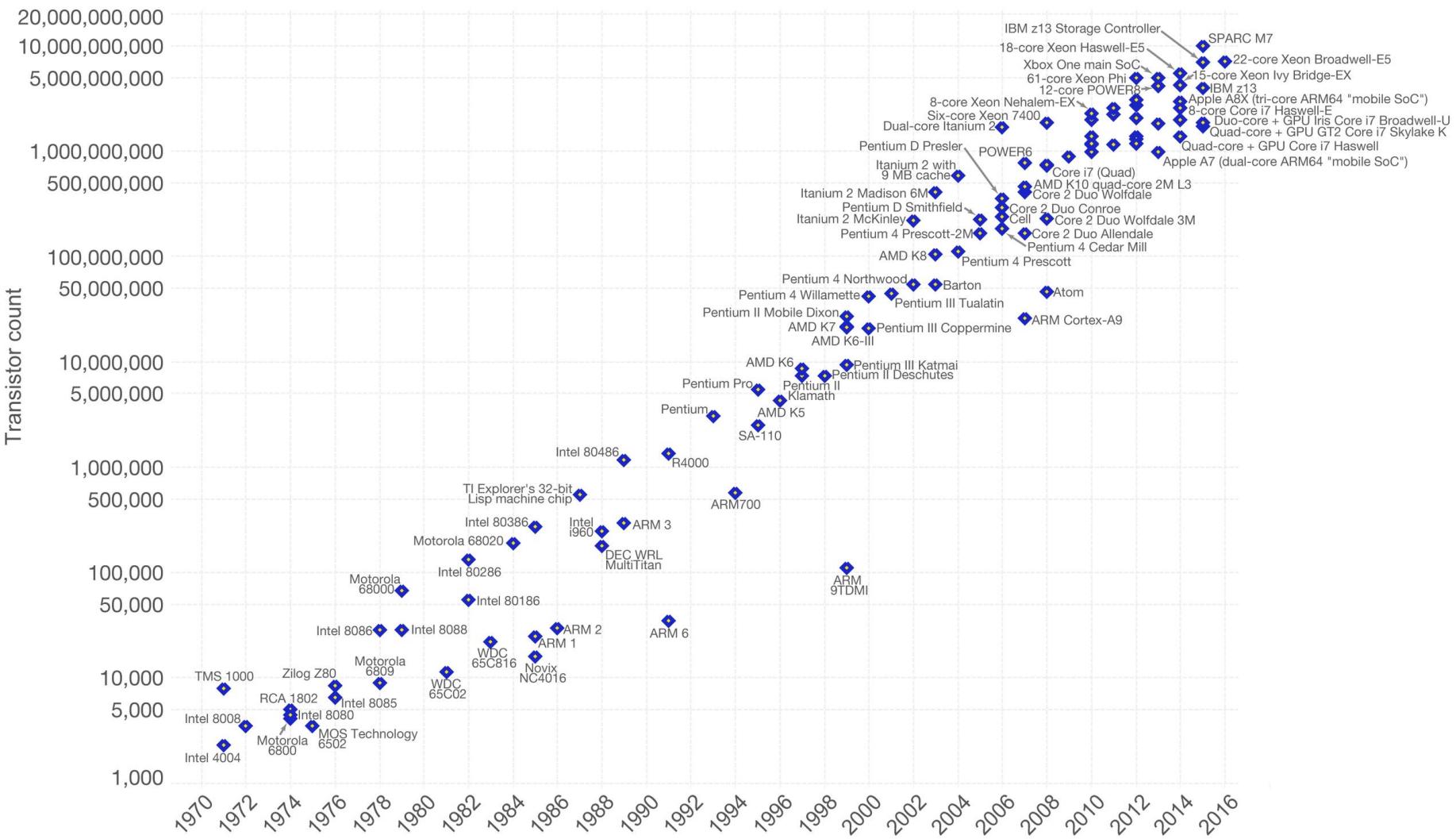


More's Law

- “The number of transistors in integrated circuits doubles approximately every 18 months”.
- Law described by Gordon E. Moore (Intel), based on the period from 1958 to 1965. Trend “for at least 10 years”.
- Today, used by semiconductor industry for long-term planning.
- Influences features of digital devices: speed of computers, memory sizes, number of pixels, ...



More's Law



Technical Obstacles

- **The memory wall:** increasing gap between processor and memory speeds. Pushes cache sizes larger in order to mask the latency of memory. Then, the memory bandwidth becomes the bottleneck.
- **The ILP wall:** the increasing difficulty of finding enough parallelism in a stream of instructions to keep a single-core processor busy.
- **The power wall:** the trend of consuming exponentially increasing power with each factorial increase of operating frequency.

Chip-Level Multiprocessors

- While manufacturing technology improves (smaller size of individual gates), **physical limits of semiconductors** have become a major design concern. These physical limits can cause significant heat dissipation, energy consumption and data synchronization problems.
- **Clock rates** increased by orders of magnitude in the decades of the late 20th century, from several megahertz in the 1980s to several gigahertz in the early 2000s.
- Then, as the rate of clock speed improvements slowed, increased use of parallel computing in the form of **multi-core and many-core processors** has been pursued to improve overall processing performance.

Hardware Acceleration

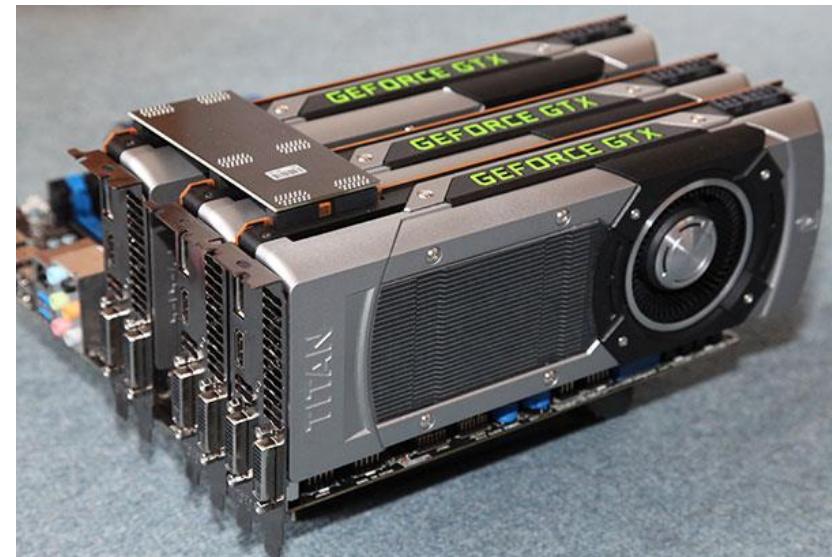
Hardware acceleration is the use of computer hardware to **perform specific functions faster** than is possible with software on a general-purpose CPU.

Depending upon **granularity**, hardware acceleration can vary from a small functional unit to a large functional block.

Support provided **SIMD execution** (SSE, AVX) and **floating-point units (FPUs)** may also be considered a form of HW acceleration.

Hardware Acceleration

- The hardware that performs the acceleration, when in a separate unit from the CPU, is referred to as a **hardware accelerator**.
- Hardware accelerators are designed for **computationally intensive** tasks.
- Examples of hardware accelerators are **GPUs**.

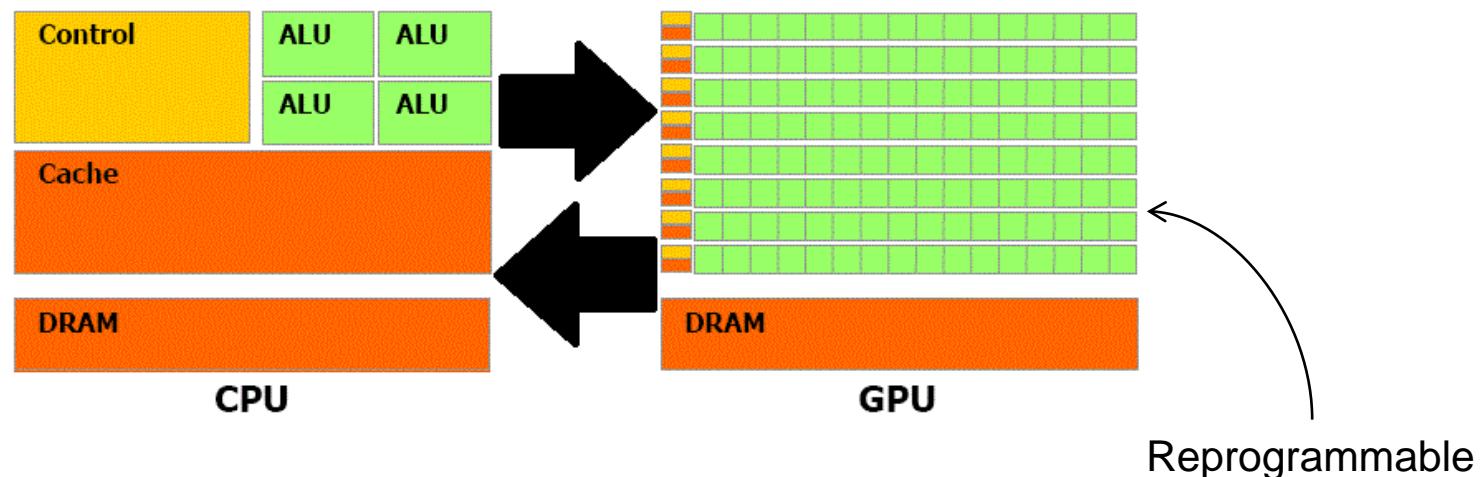


Heterogeneous Computing

Heterogeneous computing refers to systems that use more than one kind of processor at the same time.

These systems usually incorporates **specialized processing capabilities** to handle particular tasks (for example CPU + GPU).

Heterogeneous Computing



Types of Hardware Accelerators

- **GPUs:** initially designed for manipulating images for output to a display. Today, their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large blocks of data is done in parallel.
- **MICs:** multiprocessors developed by Intel (Xeon Phi), leverage x86 legacy by creating a x86-compatible multiprocessor architecture that can utilize existing parallelization software tools (OpenMP, Cilk).
- **DSPs:** specialized processors with an architecture optimized for digital signal processing. Large number of mathematical operations performed quickly and repeatedly on a series of data samples.
- **FPGAs:** integrated circuits designed to be configured by a customer or a designer after manufacturing.

GPUs

GPUs have been developed as a consequence of the strong request of faster graphic cards coming **from the gaming industry.**

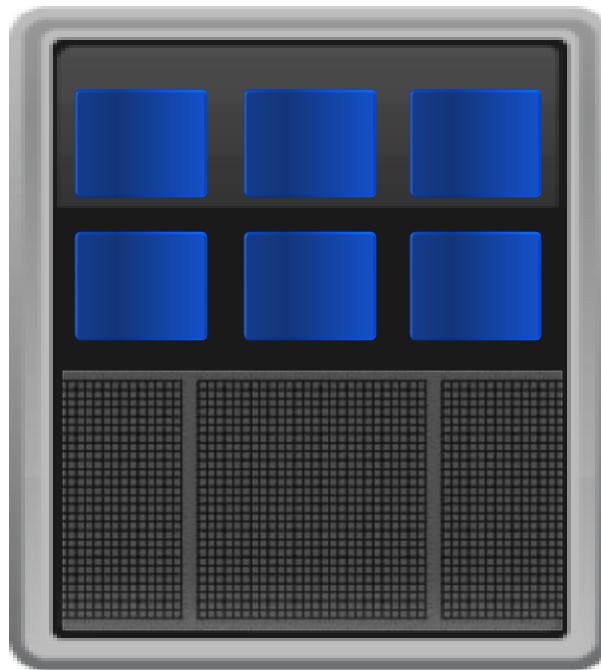
Then, people realized that the computing power provided by GPUs could also be exploited for different tasks than computer graphics: for example for scientific simulation, data mining and obviously for multimedia processing.

GPUs

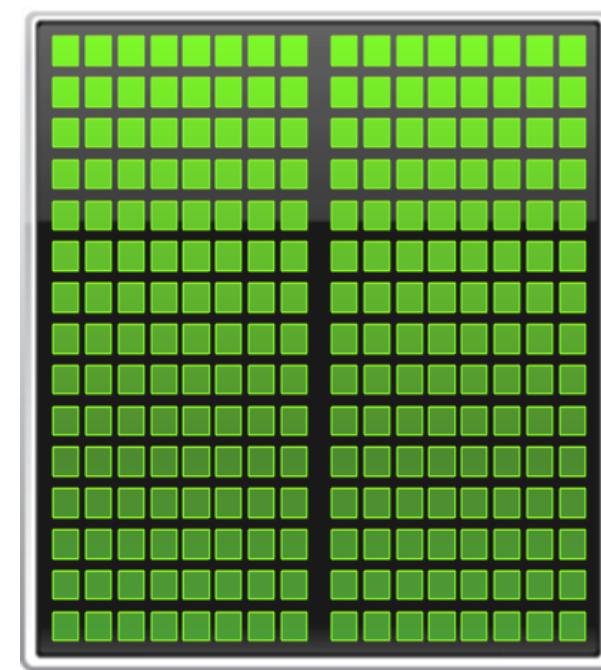
- GPUs are specialized processors with **dedicated memory** and with **many-core design optimized for SIMD floating point operations** (often required for rendering graphics).
- GPUs devote proportionally more transistors to arithmetic logic units and **less to caches and flow control** in comparison to CPUs (no branch prediction, no speculative execution, ...). GPUs also typically have **larger memory bandwidth** compared to CPUs.
- GPUs have a parallel throughput architecture that emphasizes executing **many concurrent threads slowly**, rather than executing a single thread very quickly.

Parallel Processing with GPUs

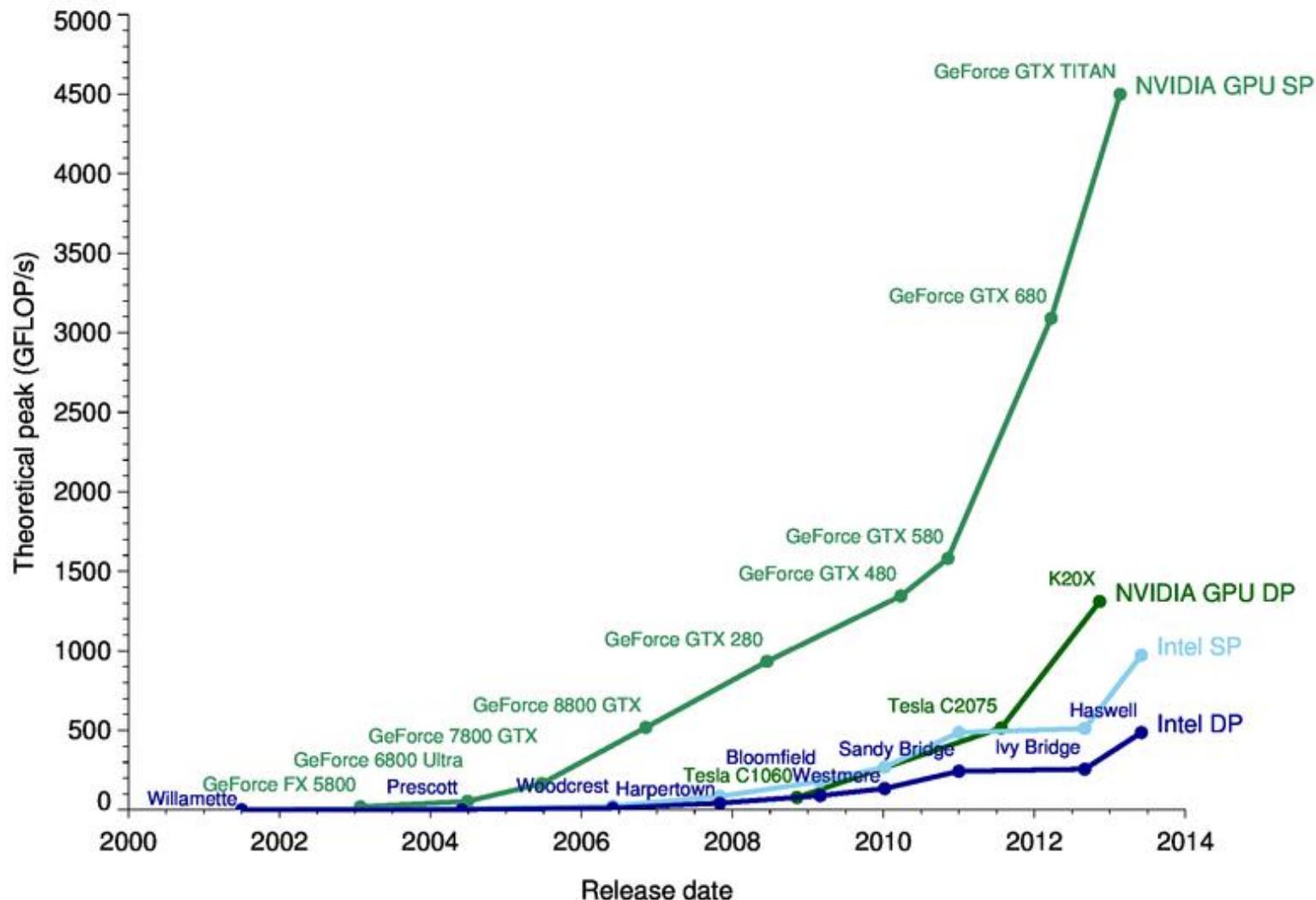
CPU



GPU



CPUs vs. GPUs – Floating Point Operations



Examples

Nvidia GeForce RTX 2080 Ti – 999\$

- 12 nm, 18.6 billion of transistors
- 4352 CUDA Cores (Turing Architecture)
- Memory: 11 GB GDDR6, 616 GB/sec
- Proc. power (GFLOPS): 13447 single, 420 double, 26895 half

AMD Radeon RX Vega 64 Liquid – 699\$

- 14 nm
- 4096 Stream Processors
- Memory: 8 GB HBM2, 484 GB/sec
- Proc. power (GFLOPS): 13738 single, 859 double, 27476 half

Examples

Nvidia Quandro GV100 – 9999\$

- 12 nm
- 5120 CUDA Cores (Volta Architecture)
- Memory: 32 GB HBM2, 870 GB/sec
- Proc. power (GFLOPS): 14800 single, 7400 double

AMD Radeon Pro SSG – 4600\$

- 14 nm
- 4096 Stream Processors (Vega Core)
- Memory: 16 GB HBM2, 484 GB/sec
- Proc. power (GFLOPS): 12290 single, 768 double, 24600 half

Examples

Nvidia Tesla V100 – 10664\$

- 12 nm
- 5120 CUDA Cores (Volta Architecture)
- Memory: 16 GB HBM2, 900 GB/sec
- Proc. power (GFLOPS): 14028 single, 7014 double

Nvidia Tegra X1

- ARM Cortex-A57 + ARM Cortex-A53 (4 + 4 cores)
- 256 CUDA Cores (Maxwell Architecture)
- Memory: 8 GB LPDDR3/4, 26.6 GB/sec
- Proc. power (GFLOPS): 512 single, 1024 half

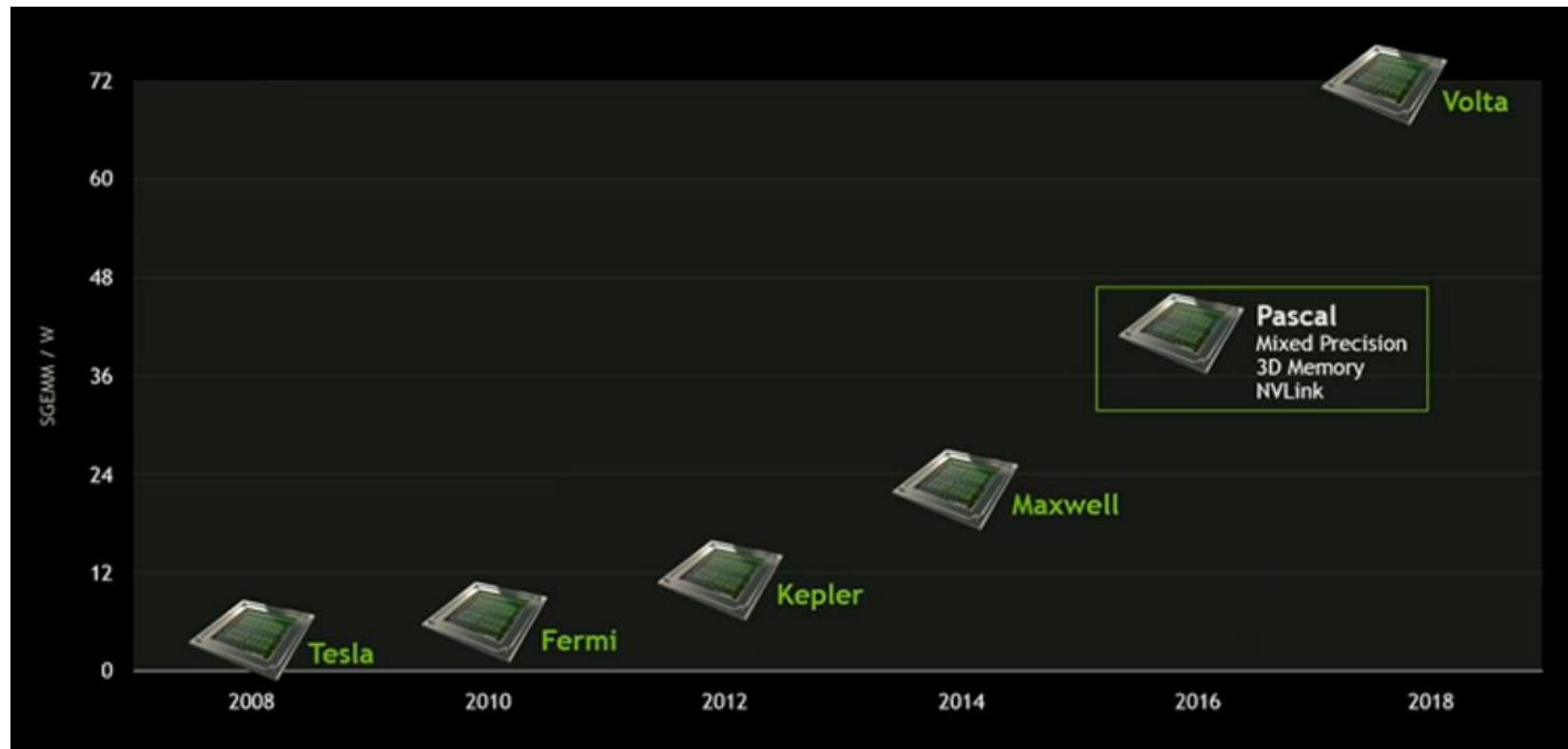
Benchmarks

User rating %	Value %	Avg. bench %	Mkt. share %	Age months
66 90 th	80 58 th	194 52 ↔ 235 100 th	0.76 96 th	1 99 th
70 93 rd		190 133 ↔ 215 100 th	0 24 th	11+ 97 th
70 93 rd		156 53 ↔ 185 100 th	0.05 66 th	19+ 93 rd
148 99 th	79.4 57 th	155 68 ↔ 177 99 th	3.71 99 th	20 93 rd
54 72 nd		151 127 ↔ 164 99 th	0 24 th	24+ 91 st
57 82 nd	93.1 84 th	150 61 ↔ 169 99 th	1.06 97 th	1 99 th
60 86 th		146 44 ↔ 181 99 th	0.03 42 nd	27+ 90 th
58 83 rd	102 98 th	134 111 ↔ 145 99 th	0.57 94 th	1 100 th
52 56 th		132 92 ↔ 152 99 th	0 24 th	18+ 94 th
60 86 th		124 71 ↔ 139 98 th	0.01 42 nd	15+ 96 th

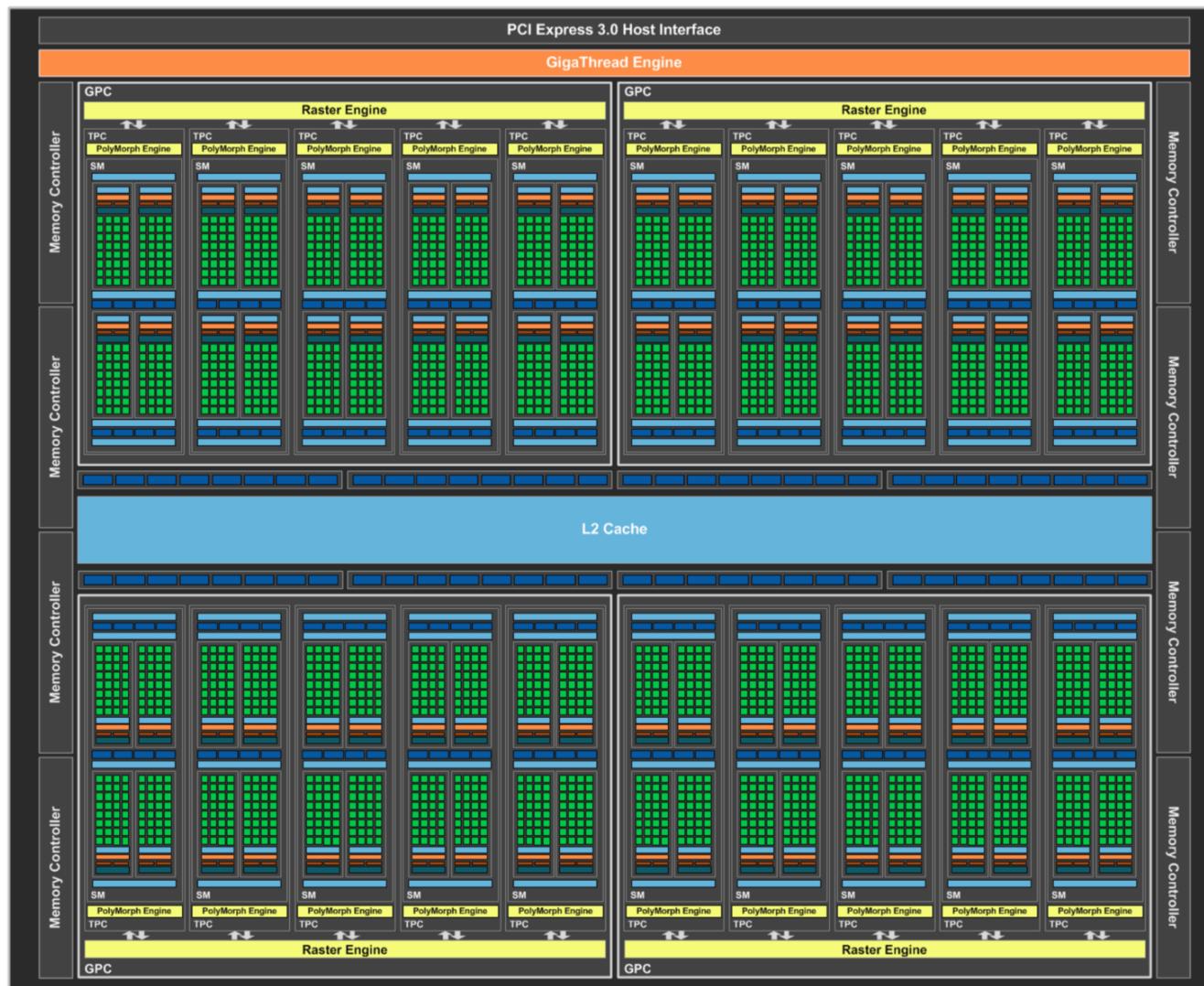
<http://gpu.userbenchmark.com/Comparison?>

<https://www.futuremark.com/hardware/gpu>

Nvidia Architecture Evolution



Pascal Architecture



Turnig Architecture



Tensor Cores

NVIDIA Tesla GPUs are powered by **tensor cores**, a revolutionary technology that delivers groundbreaking deep learning performance.

Tensor cores can accelerate **large matrix operations**, used when training neural networks, and perform mixed-precision matrix multiply and accumulate calculations in a single operation.

For example, a tensor core multiplies two 4×4 FP16 matrices, and then adds a third FP16 or FP32 matrix to the result by using fused multiply–add operations, and obtains an FP32 result that could be optionally demoted to an FP16 result.

Ray Tracing

The Turing architecture introduced the first consumer products capable of **real-time ray-tracing**, which has been a longstanding goal of the computer graphics industry.

The ray-tracing performed by the cores can be used to produce **reflections, refractions and shadows**, replacing traditional raster techniques such as cube maps and depth maps.

Instead of replacing rasterization entirely, however, the information gathered from ray-tracing can be used to **augment the shading with information that is much more photo-realistic**, especially in regards to off-camera action.

GPUs

GPUs have been developed as a consequence of the strong request of faster graphic cards coming **from the gaming industry.**

Then, people realized that the computing power provided by GPUs could also be exploited for different tasks than computer graphics: for example for scientific simulation, data mining and obviously for multimedia processing.

GPUs

Use **massively parallel approach** to hide latency:

- A lot more threads than cores
- Very low task switching overhead
- Threads are not swapped in/out, registers are mapped to threads for entire lifetime

GPUs

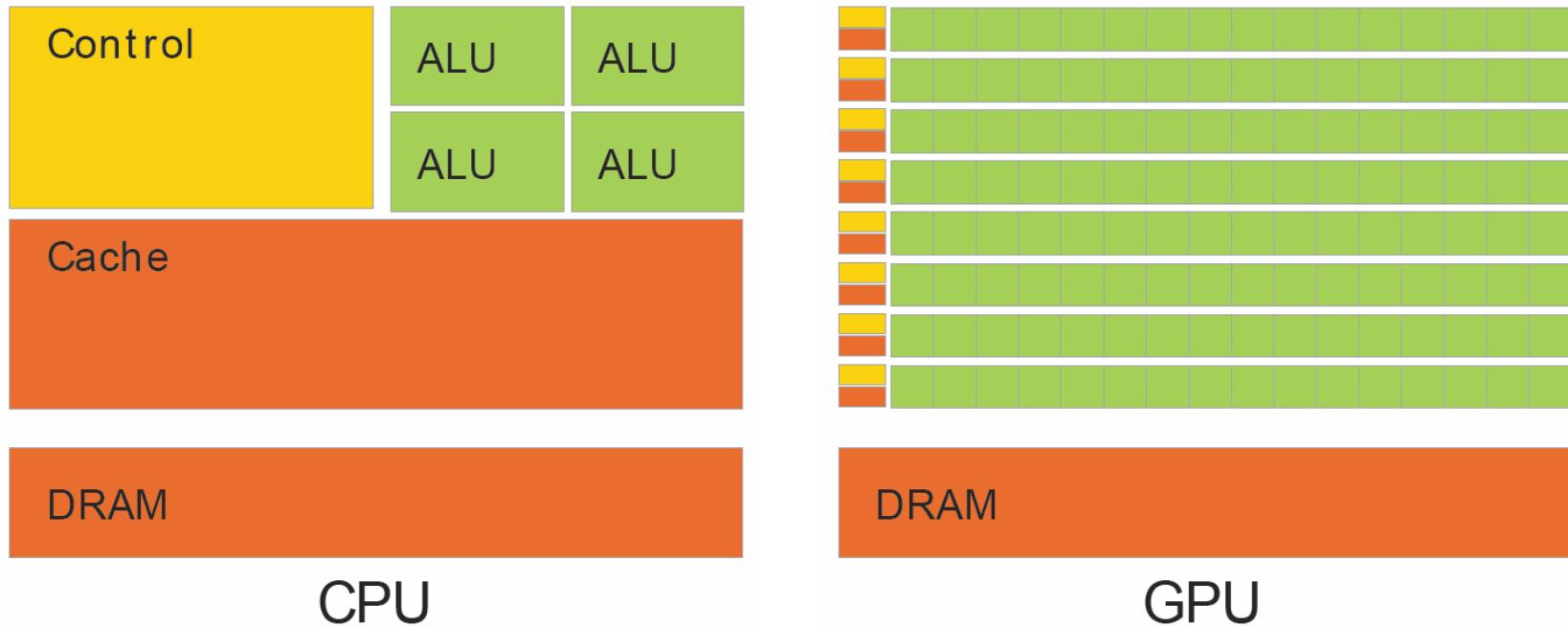
Even with massive bandwidth, memory is very slow relative to compute.

GPUs hide memory latency through fast thread context switch.

As a result of the parallel architecture:

- Memory transfers and computes overlap
- Hide memory latency with compute and compute with memory transfers

CPU vs. GPU



Host: the CPU and its memory (host memory)

Device: the GPU and its memory (device memory)

GPU Details

Main hardware features:

- Multi-core design
- SIMD core optimized for floating point
- Dozens of multi cores per card
- User controlled caches per multi core (shared memory)
- Read only 2D optimized caches (textures)
- Real L1/L2 caches on from the Fermi generation

Parallel Processing with GPUs

Programming solutions:

- **CUDA Toolkit** (C/C++, Fortran, Python): Nvidia
- **OpenCL**: open standard, AMD, Nvidia, Intel, Apple, ...
- **Vulkan API**: open standard, AMD, Nvidia, Intel, Apple, ...
- **DirectCompute**: part of the Microsoft DirectX APIs
- **OpenACC**: collection of compiler directives to specify loops and regions of code in standard C, C++ and Fortran to be offloaded from a host CPU to an attached accelerator
- **C++ AMP** (Accelerated Massive Parallelism): library implemented on DirectCompute, with language extensions
- **OpenMP 4.5**: adds support for hardware accelerators

CUDA Tools Ecosystem



Accelerated Solutions

GPUs are accelerating many applications across numerous industries.

[Learn more >](#)



Numerical Analysis Tools

GPU acceleration for applications with high arithmetic density.

[Learn more >](#)



GPU-Accelerated Libraries

Application accelerating can be as easy as calling a library function.

[Learn more >](#)



Language and APIs

GPU acceleration can be accessed from most popular programming languages.

[Learn more >](#)



Performance Analysis Tools

Find the best solutions for analyzing your application's performance profile.

[Learn more >](#)



Debugging Solutions

Powerful tools can help debug complex parallel applications in intuitive ways.

[Learn more >](#)



Key Technologies

Learn more about parallel computing technologies and architectures.

[Learn more >](#)



Accelerated Web Services

Micro services with visual and intelligent capabilities using deep learning.

[Learn more >](#)

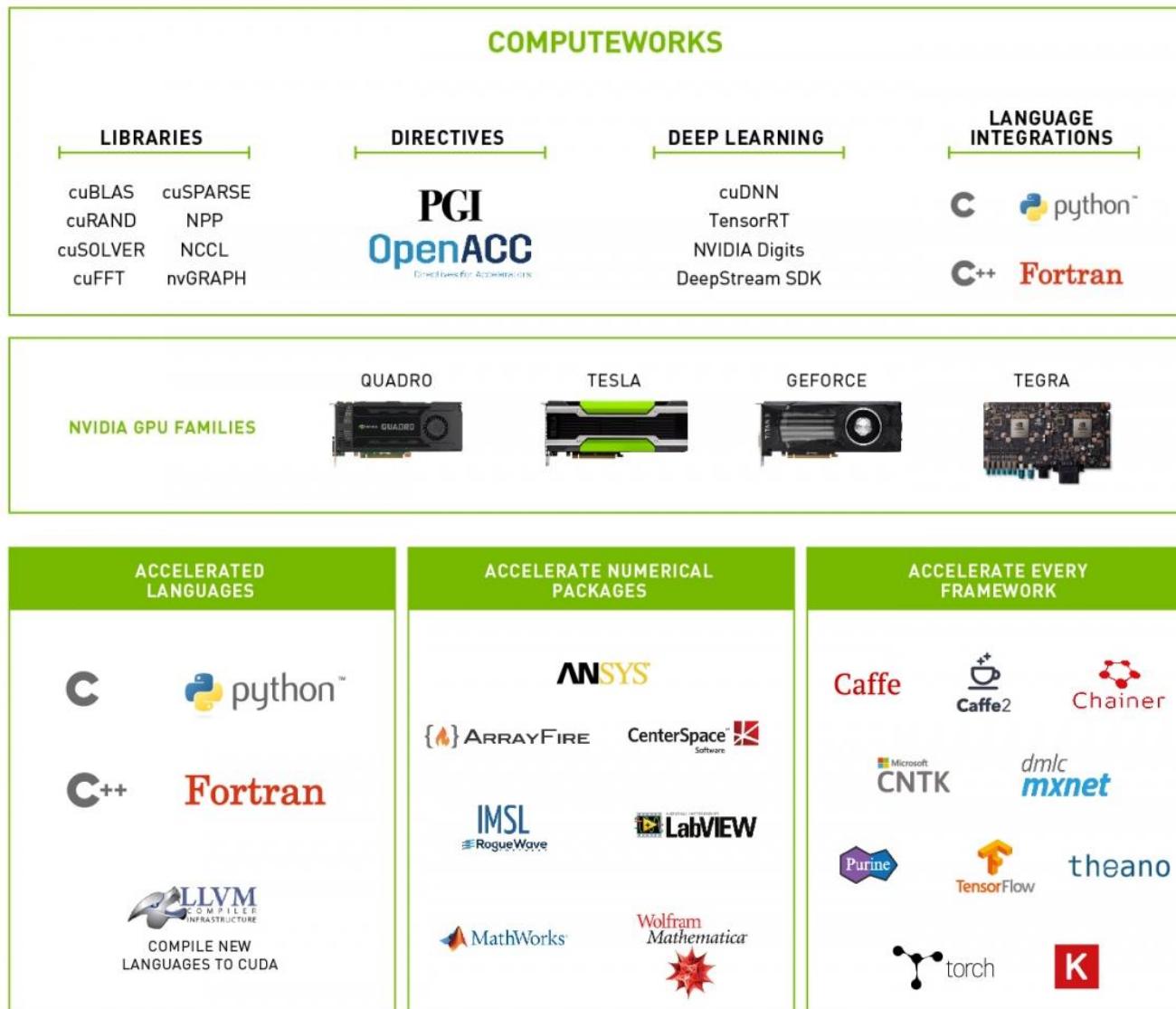


Cluster Management

Managing your cluster and job scheduling can be simple and intuitive.

[Learn more >](#)

Nvidia ComputeWorks



CUDA Parallel Computing Platform

Programming Approaches

Libraries

“Drop-in”
Acceleration

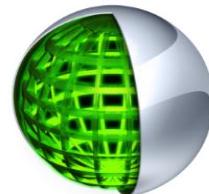
OpenACC
Directives

Easily Accelerate
Apps

Programming
Languages

Maximum Flexibility

Development Environment



Nsight IDE (Eclipse, VS)
Linux, Mac and Windows
GPU Debugging and
Profiling

CUDA-GDB
debugger
NVIDIA Visual
Profiler

Open Compiler Tool Chain



Enables compiling new languages to CUDA
platform, and CUDA languages to other
architectures

Libraries: Easy Acceleration

- **Ease of use:** using libraries enables GPU acceleration without in-depth knowledge of GPU programming.
- **“Drop-in”:** many GPU-accelerated libraries follow standard APIs, thus enabling acceleration with minimal code changes.
- **Quality:** libraries offer high-quality implementations of functions encountered in a broad range of applications.
- **Performance:** libraries are tuned by experts.

Use of Libraries

Step 1: Substitute library calls with equivalent in library:

saxpy (...) -> cublasSaxpy (...)

Step 2: Manage data locality

- with CUDA: cudaMalloc(), cudaMemcpy(), etc.
- with CUBLAS: cublasAlloc(), cublasSetVector(), etc.

Step 3: Rebuild and link the CUDA-accelerated library

nvcc myobj.o -l cublas

OpenACC

- Simple compiler hints
- Compiler parallelizes code
- Works on many-core GPUs & multi-core CPUs
- **Easy:** Directives are the easy path to accelerate compute intensive applications.
- **Open:** OpenACC is an open GPU directives standard, making GPU programming straightforward and portable across parallel and multi-core processors.
- **Powerful:** GPU directives allow complete access to the massive parallel power of a GPU.

C Version (Sequential)

```
while (error > tol && iter < iter_max) {  
    error = 0.0;  
  
    for(int j = 1; j < n-1; j++) {  
        for(int i = 1; i < m-1; i++) {  
  
            Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +  
                                  A[j-1][i] + A[j+1][i]);  
  
            error = max(error, abs(Anew[j][i] - A[j][i]));  
        }  
    }  
  
    for(int j = 1; j < n-1; j++) {  
        for(int i = 1; i < m-1; i++) {  
            A[j][i] = Anew[j][i];  
        }  
    }  
  
    iter++;  
}
```

Iterate until converged

Iterate across matrix elements

Calculate new value from neighbors

Compute max error for convergence

Swap input/output arrays

OpenACC Version

```
#pragma acc data copy(A), create(Anew)
while (error > tol && iter < iter_max) {
    error = 0.0;

#pragma acc kernels
for(int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {
        Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                               A[j-1][i] + A[j+1][i]);
        error = max(error, abs(Anew[j][i] - A[j][i]));
    }
}

#pragma acc kernels
for(int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {
        A[j][i] = Anew[j][i];
    }
}

iter++;
}
```

Copy A in and out at start/end. Allocate Anew on accelerator

Iterate until converged

Execute GPU Kernel for loop

Calculate new value from neighbors

Compute max error for convergence

Execute GPU Kernel for loop

What is CUDA

The Compute Unified Device Architecture (CUDA) is a parallel computing platform and programming model invented by NVIDIA that enables harnessing the power of graphics processing units (GPUs).

From its introduction in 2006, CUDA has been widely deployed, supported by an installed base of over 300 million CUDA-enabled GPUs in notebooks, workstations, compute clusters and supercomputers.

Today, GPU-accelerated applications are available for astronomy, biology, chemistry, physics, data mining, manufacturing, finance, ...

GPU Details

The NVIDIA GPU architecture is built around a **scalable array of multithreaded *Streaming Multiprocessors (SMs)***.

A multiprocessor is designed to execute hundreds of threads concurrently. To manage such a large amount of threads, it employs a unique architecture called **SIMT** (Single-Instruction, Multiple-Thread).

The instructions are pipelined to leverage **instruction-level parallelism within a single thread, as well as thread-level parallelism** extensively through simultaneous hardware multithreading.

However, unlike CPU cores, **instructions are issued in order** and there is no branch prediction and no speculative execution.

CUDA Programming Model

The GPU is seen as a compute device to execute a portion of an application that:

- Has to be executed many times
- Can be isolated as a function
- Works independently on different data

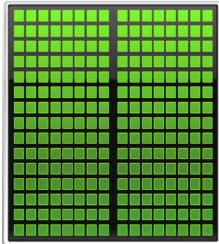
Such a function can be compiled to run on the device. The resulting program is called a **kernel**.

GPU Kernels

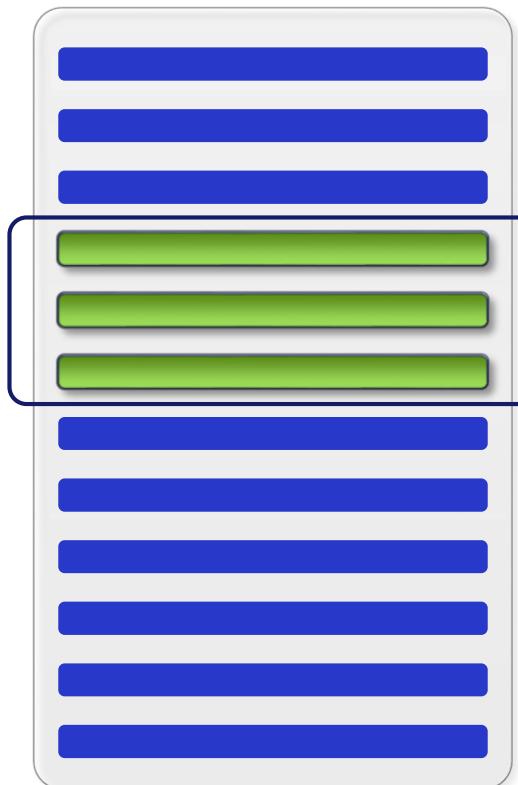
Application Code

Compute-Intensive Functions

GPU

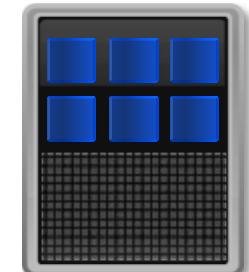


Use GPU to Parallelize



Rest of Sequential CPU Code

CPU

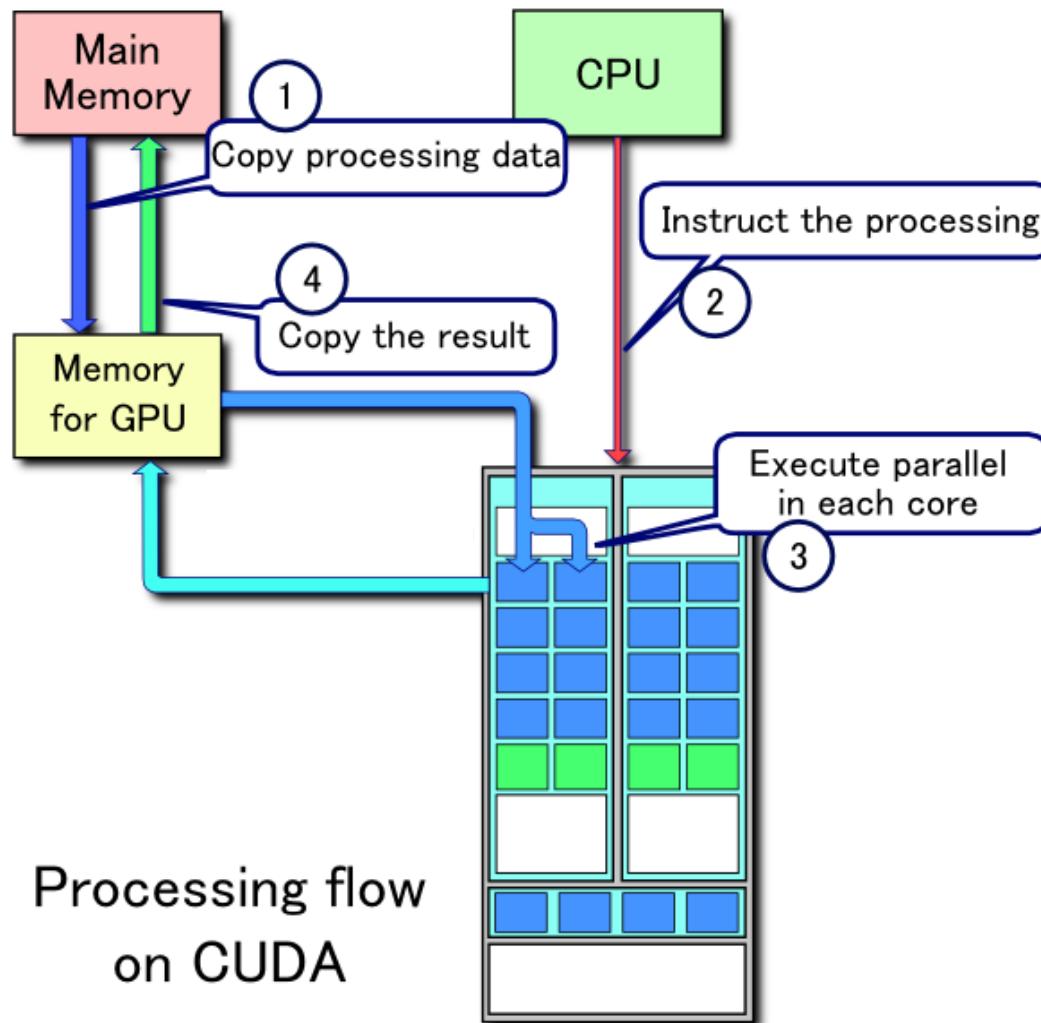


+

What is a Kernel

- Calculation on the GPU is done via **kernels**.
- A kernel is a function callable from the host and executed on the CUDA device.
- **Run simultaneously by many threads in parallel.**
- Each kernel does as little work as possible.

GPU Processing Flow



Unified Memory

Unified memory is a memory management system that simplifies GPU development by providing a **single memory space directly accessible by all GPUs and CPUs** in the system, with automatic page migration for data locality.

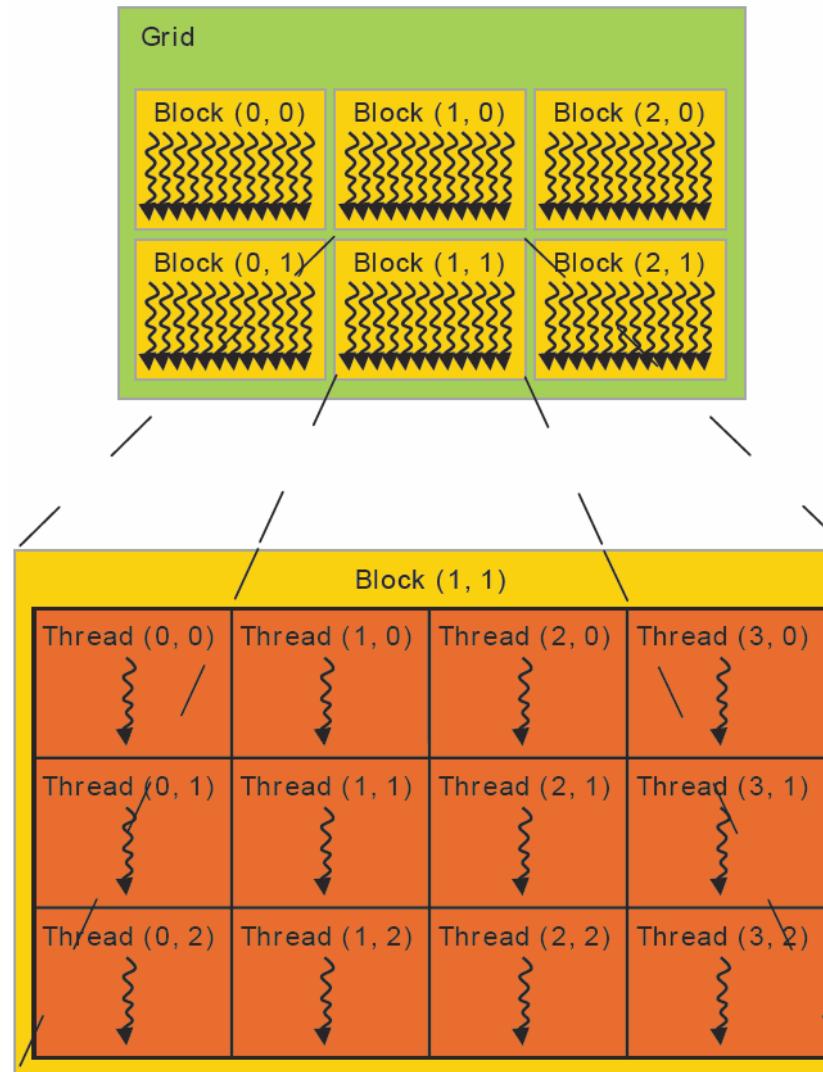
Migration of pages allows the accessing processor to benefit from L2 caching and the lower latency of local memory. Moreover, migrating pages to GPU memory ensures GPU kernels take advantage of the very high bandwidth of GPU memory (e.g. 720 GB/s on a Tesla P100).

Unified Memory

Page migration is all **completely invisible** to the developer: the system automatically manages all data movement for you.

Starting from the Pascal GPU architecture unified memory is more powerful, thanks to larger virtual memory address space and page migration engine, enabling true virtual memory demand paging.

Kernel Execution



Kernel Execution

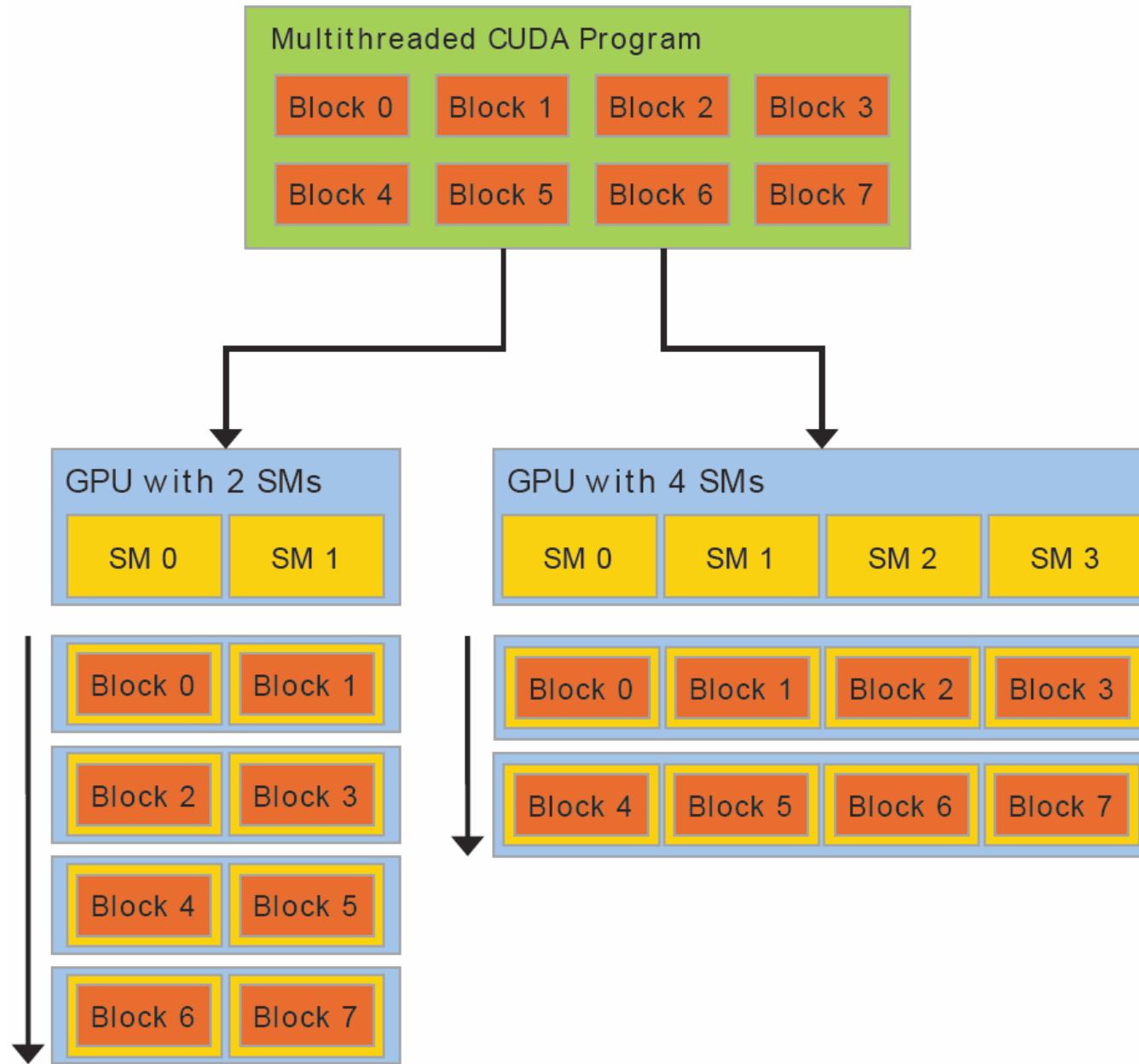
When a CUDA program on the host CPU invokes a **kernel grid**, the blocks of the grid are distributed to multiprocessors with available execution capacity. **The threads of a thread block execute concurrently on one multiprocessor**, and multiple thread blocks can execute concurrently on one multiprocessor.

As thread blocks terminate, new blocks are launched on the vacated multiprocessors.

The batch of threads that executes a kernel is organized as a **grid of thread blocks**.

Grids of Thread Blocks

- Limited number of threads in a block.
- Allows larger numbers of thread to execute the same kernel with one invocation.
- Blocks identifiable via block ID.
- Blocks can be one- or two-dimensional arrays.



Kernel Execution

The multiprocessor creates, manages, schedules, and executes threads in groups of 32 parallel threads called warps.

Individual threads composing a warp start together at the same program address, but they have their own instruction address counter and register state and are therefore free to branch and execute independently.

The way a block is partitioned into warps is always the same; each warp contains threads of consecutive, increasing thread IDs with the first warp containing thread 0.

Blocks are not swapped out until they finish all their work.

Kernel Execution

A warp executes one common instruction at a time, so **full efficiency is realized when all 32 threads of a warp agree on their execution path.**

If threads of a warp diverge via a **data-dependent conditional branch**, the warp **serially executes each branch path taken**, disabling threads that are not on that path.

Branch divergence occurs only within a warp; different warps execute independently regardless of whether they are executing common or disjoint code paths.

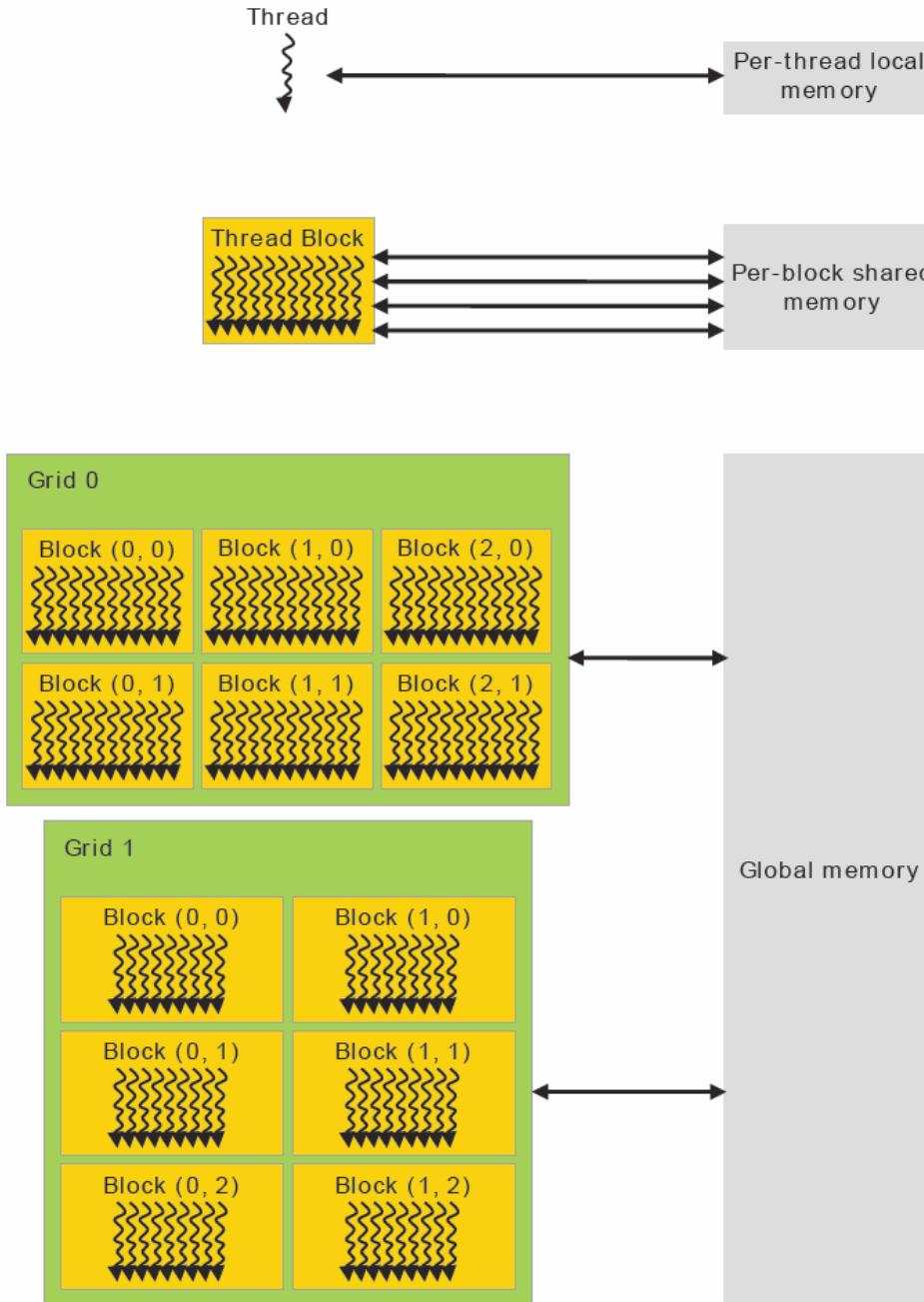
Kernel Execution

If an **atomic instruction** executed by a warp reads, modifies, and writes to the same location in global memory for more than one of the threads of the warp, **each read/modify/write to that location occurs and they are all serialized, but the order in which they occur is undefined.**

SIMT vs. SIMD

Compared to SIMD, SIMT enables programmers to write **thread-level parallel code for independent, scalar threads, as well as data-parallel code for coordinated threads.**

For the purposes of correctness, the programmer can essentially ignore the SIMT behavior; however, **substantial performance improvements can be realized by taking care that the code seldom requires threads in a warp to diverge.**



SUPSI

Programming CUDA

Tiziano Leidi

Sviluppo di Applicazioni High-Performance
Bachelor in Ingegneria Informatica

C for CUDA

CUDA provides several **additions to the C/C++ environment**:

- C language extensions to enable heterogeneous programming
- mathematical functions of the C/C++ standard library (for both host and device)
- intrinsic functions (only supported on the device)
- CUDA runtime API and CUDA driver API
- Environment variables

C/C++ for the device code has some limitations compared to the full standard. For example it does not support run time type information (RTTI) and exception handling.

Example: Device Code

A simple kernel to add two vectors of integers:

```
__global__ void add(int *a, int *b, int *c) {  
    int index = threadIdx.x + blockIdx.x * blockDim.x;  
    c[index] = a[index] + b[index];  
}
```

CUDA Kernel

- The function type qualifier `__global__` declares a function as being an executable kernel on the CUDA device.
- This function can only be called from the host.
- All kernels **must be declared with a return type of void.**

Example: Host Code

```
#define N (2048*2048)
#define THREADS_PER_BLOCK 512
int main(void) {
    int *a, *b, *c;                      // host copies of a, b, c
    int *d_a, *d_b, *d_c;                  // device copies of a, b, c
    int size = N * sizeof(int);

    // Alloc space for device copies of a, b, c
    cudaMalloc((void **) &d_a, size);
    cudaMalloc((void **) &d_b, size);
    cudaMalloc((void **) &d_c, size);

    // Alloc space for host copies of a, b, c
    // and setup input values
    a = (int *)malloc(size); random_ints(a, N);
    b = (int *)malloc(size); random_ints(b, N);
    c = (int *)malloc(size);
```

Example: Host Code

```
// Copy inputs to device
cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);

// Launch add() kernel on GPU
add<<<N/THREADS_PER_BLOCK, THREADS_PER_BLOCK>>>(d_a, d_b, d_c);

// Copy result back to host
cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);

// Cleanup
free(a); free(b); free(c);
cudaFree(d_a); cudaFree(d_b); cudaFree(d_c);
return 0;
}
```

Function Type Qualifiers

The available function type qualifiers are:

device

- Executed on the device
- Callable from the device only

global

- Executed on the device
- Callable from the host only

host

- Executed on the host
- Callable from the host only

Calling the Kernel

- The host calls the kernel by specifying the name of the kernel plus an **execution configuration**:

```
Func<<< Dg, Db, Ns >>>(parameter) ;
```

- The configuration includes three things:
 - The number of threads in a group - **the block geometry**
 - The number of groups - **the grid geometry**
 - The number of bytes in shared memory that is dynamically allocated (optional) in addition to the statically allocated memory

dim3

- To pass the grid and block dimensions in a kernel invocation an integer vector type is used: **dim3**.
- dim3 has 3 elements x, y and z.
- In C code, dim3 can be initialized as:

```
dim3 grid = { 512, 512, 1 };
```

- In C++ code, dim3 can be initialized as:

```
dim3 grid( 512, 512, 1 );
```
- Not all the 3 elements need to be provided. Any element not provided during initialization is initialized to 1.

Calling the Kernel

Dimensions:

- Dg is of type **dim3**:
 $Dg.x * Dg.y$ = number of blocks being launched;
- Db is of type **dim3**:
 $Db.x * Db.y * Db.z$ = number of threads per block;
- Ns is of type **size_t**.

Built-in variables:

- $gridDim$ is of type **dim3** - dimensions of the grid.
- $blockIdx$ is of type **uint3** - block index within the grid.
- $blockDim$ is of type **dim3** - dimensions of the block.
- $threadIdx$ is of type **uint3** - thread index within the block.

Threads and Blocks

Thread 0

Thread 1

Thread 2

Thread 3

```
c[0] = a[0] + b[0];
```

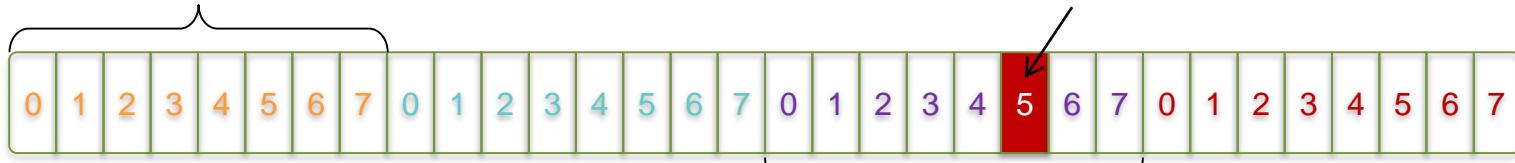
```
c[1] = a[1] + b[1];
```

```
c[2] = a[2] + b[2];
```

```
c[3] = a[3] + b[3];
```

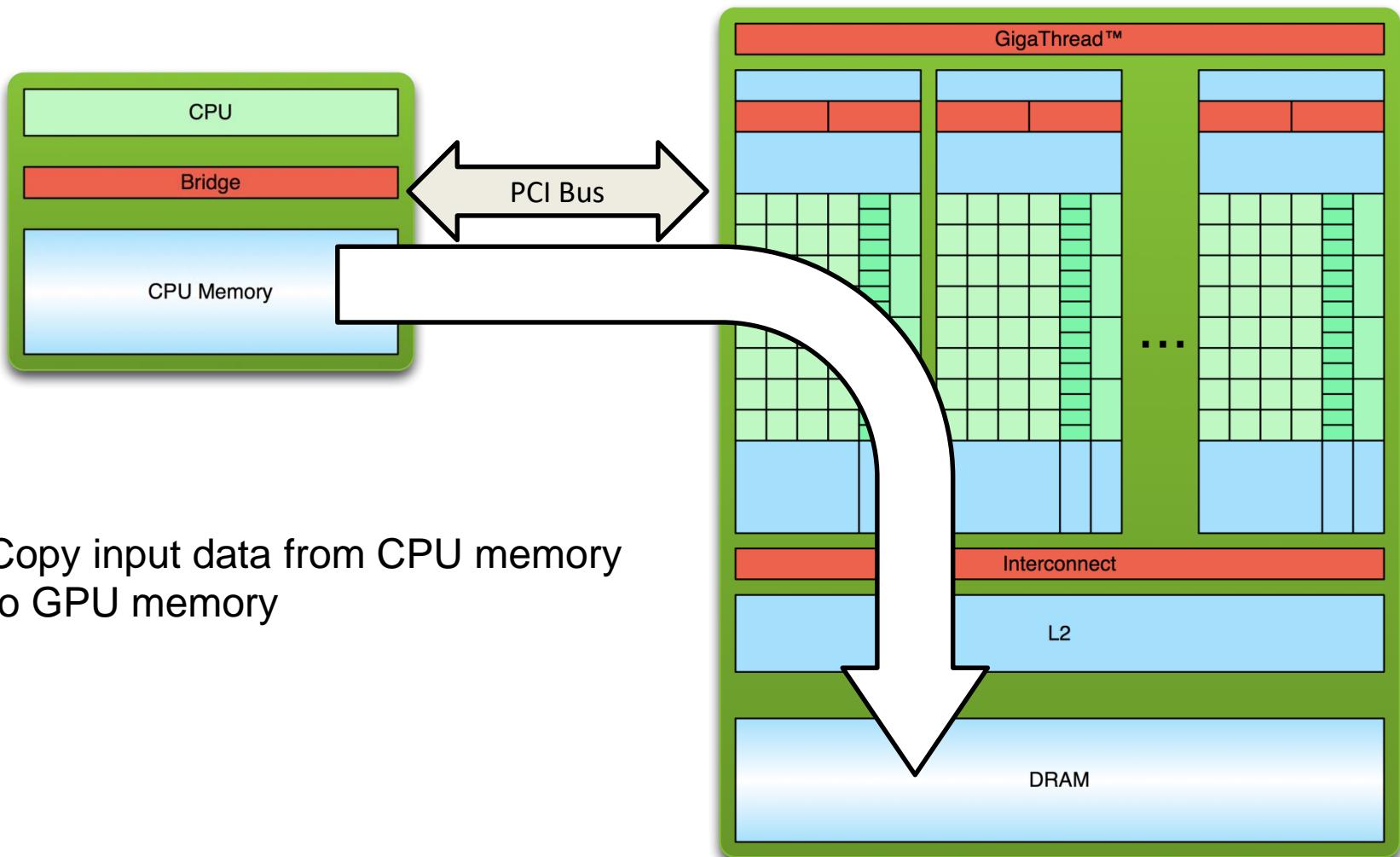


M = 8

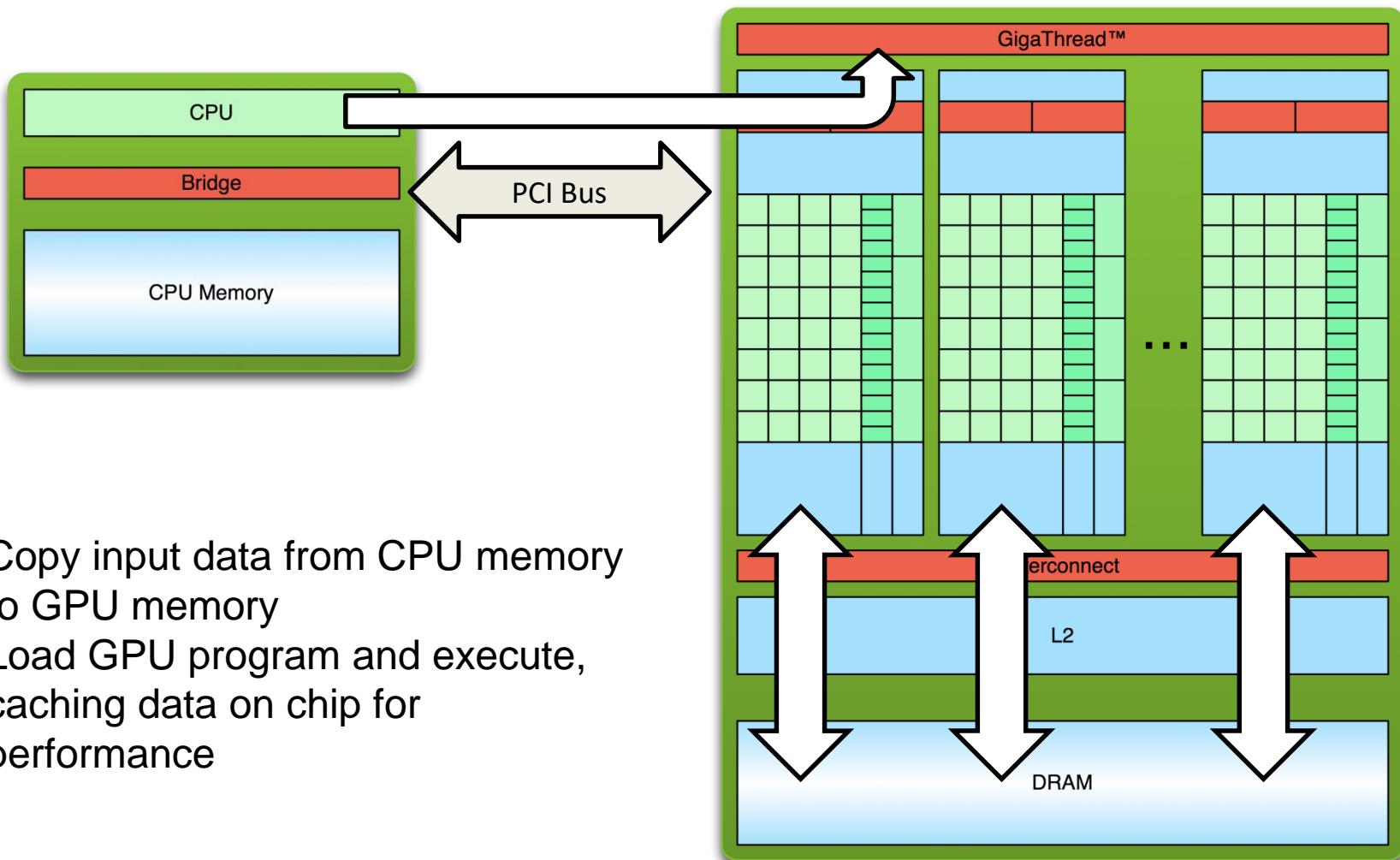


blockIdx.x = 2

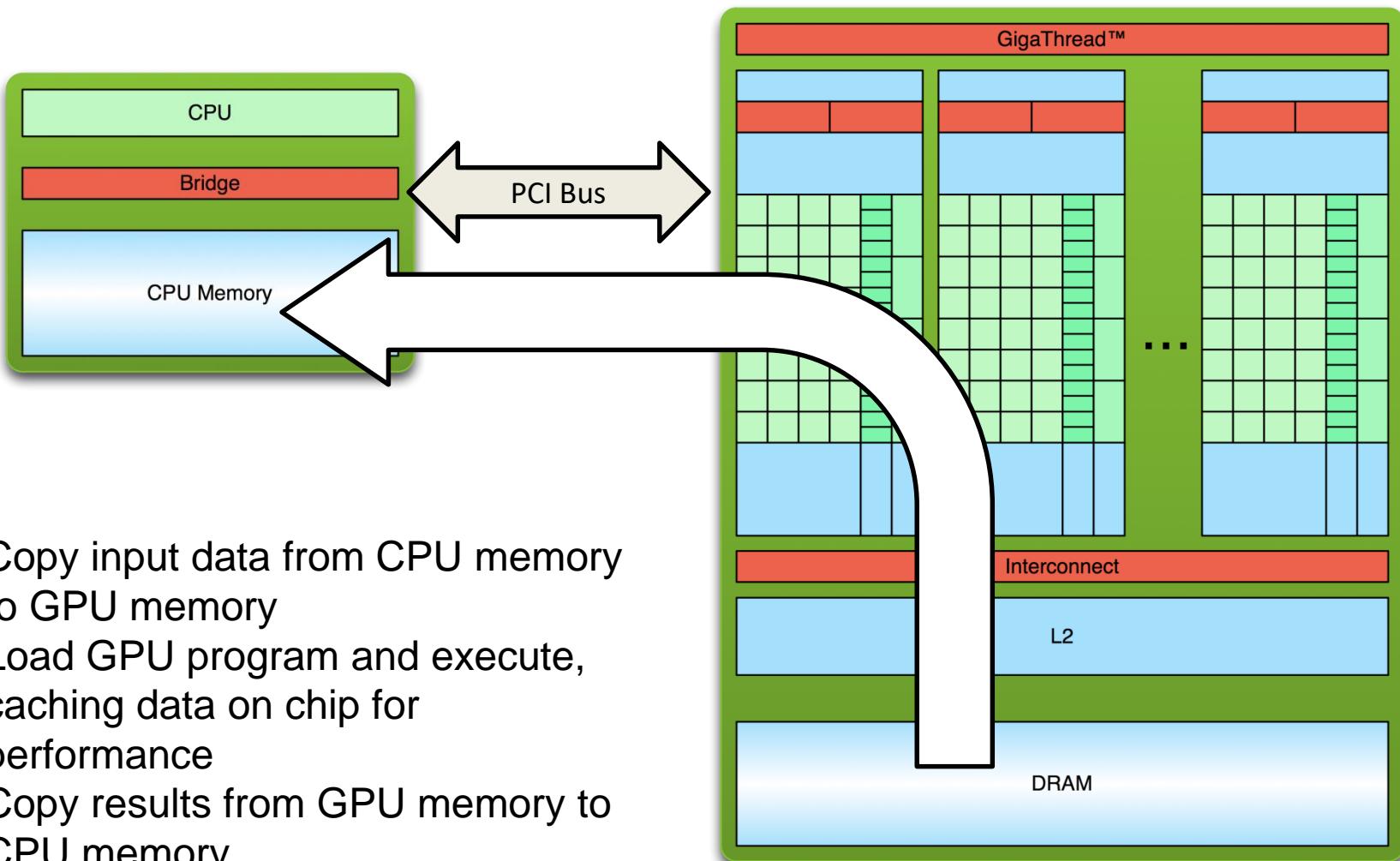
Processing Flow



Processing Flow



Processing Flow

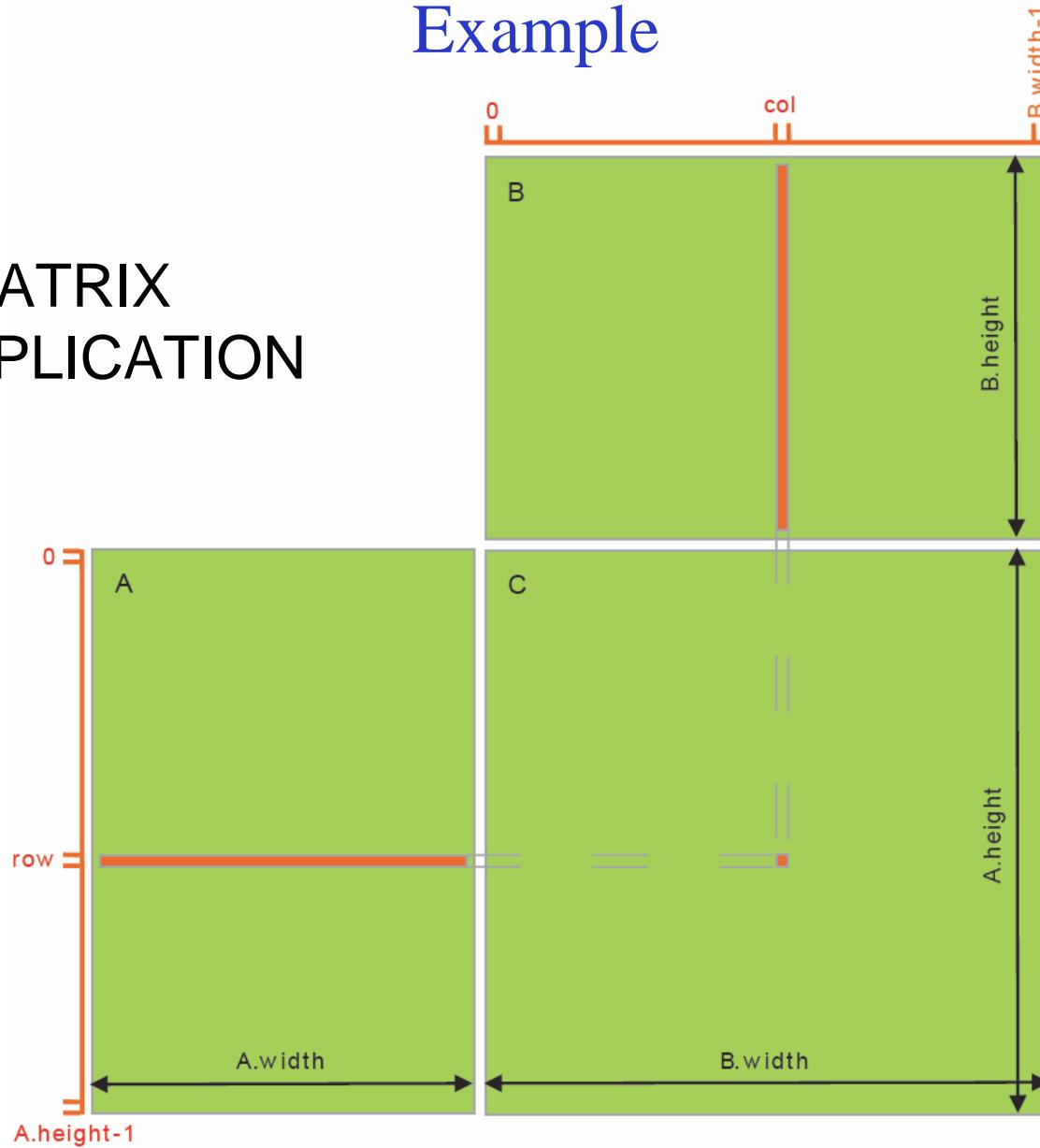


Compilation

- CUDA C language programs have **the suffix ".cu"**.
- Can be compiled directly to executable or to object code.
- Can be later linked to objects created by different compilers (e.g. integrated into visual studio).

Example

MATRIX MULTIPLICATION



Example

```
// Matrices are stored in row-major order:  
// M(row, col) = *(M.elements + row * M.width + col)  
typedef struct {  
    int width;  
    int height;  
    float* elements;  
} Matrix;  
// Thread block size  
#define BLOCK_SIZE 16  
  
// Forward declaration of the matrix multiplication kernel  
__global__ void MatMulKernel(const Matrix, const Matrix, Matrix);  
  
// Matrix multiplication - Host code  
// Matrix dimensions are assumed to be multiples of BLOCK_SIZE  
void MatMul(const Matrix A, const Matrix B, Matrix C) {
```

Example

```
// Load A and B to device memory
Matrix d_A;
d_A.width = A.width; d_A.height = A.height;
size_t size = A.width * A.height * sizeof(float);
cudaMalloc(&d_A.elements, size);
cudaMemcpy(d_A.elements, A.elements, size, cudaMemcpyHostToDevice);
Matrix d_B;
d_B.width = B.width; d_B.height = B.height;
size = B.width * B.height * sizeof(float);
cudaMalloc(&d_B.elements, size);
cudaMemcpy(d_B.elements, B.elements, size, cudaMemcpyHostToDevice);
// Allocate C in device memory
Matrix d_C;
d_C.width = C.width; d_C.height = C.height;
size = C.width * C.height * sizeof(float);
cudaMalloc(&d_C.elements, size);
```

Example

```
// Invoke kernel
dim3 dimBlock(BLOCK_SIZE, BLOCK_SIZE);
dim3 dimGrid(B.width / dimBlock.x, A.height / dimBlock.y);
MatMulKernel<<<dimGrid, dimBlock>>>(d_A, d_B, d_C);
// Read C from device memory
cudaMemcpy(C.elements, Cd.elements, size, cudaMemcpyDeviceToHost);
// Free device memory
cudaFree(d_A.elements);
cudaFree(d_B.elements);
cudaFree(d_C.elements);
}
```

Example

```
// Matrix multiplication kernel called by MatMul()
__global__ void MatMulKernel(Matrix A, Matrix B, Matrix C) {
    // Each thread computes one element of C
    // by accumulating results into Cvalue
    float Cvalue = 0;
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;
    for (int e = 0; e < A.width; ++e)
        Cvalue += A.elements[row * A.width + e] * B.elements[e * B.width + col];
    C.elements[row * C.width + col] = Cvalue;
}
```

Memory Management

CUDA provides a simple API for handling device memory:

- `cudaMalloc()`, `cudaFree()`, `cudaMemcpy()`
- similar to the C equivalents `malloc()`, `free()`, `memcpy()`

Host and device memory are separate entities.

Device pointers point to GPU memory:

- May be passed to/from host code
- May not be dereferenced in host code

Host pointers point to CPU memory:

- May be passed to/from device code
- May not be dereferenced in device code

Variable Type Qualifiers

The available variable type qualifiers are:

device

- Resides in **global memory space**
- Has the **lifetime of an application**
- Is accessible **from all the threads** within the grid and from the host through the runtime library

constant

- Resides in **constant (global) memory space**
- Has the **lifetime of an application**
- Is accessible **from all the threads** within the grid and from the host through the runtime library

Variable Type Qualifiers

shared

- Resides in the **shared memory space** of a thread block
- Has the **lifetime of the block**
- Is only accessible **from all the threads within the block**

Register Memory

Registers are:

- The **fastest form of memory** on the GPU
- Reside on the multi-core
- Only accessible by the thread
- Have the **lifetime of the thread**

Shared Memory

Shared memory:

- Can be as fast as registers when there are no bank conflicts or when reading from the same address
- Accessible by any thread of the block from which it was created
- Has the lifetime of the block

Global Memory

Global memory:

- Potentially **150x slower** than register or shared memory
- Watch out for uncoalesced reads and writes
- Accessible from either the **host or device**
- Has the **lifetime of the application**

Local Memory

Local memory:

- A potential performance bottleneck
- Resides in global memory
- Can be 150x slower than register or shared memory
- Only accessible by the thread
- Has the lifetime of the thread

Used automatically by CUDA for:

- register overflow
- arrays addressed by parameters in register memory
- possibly misaligned global memory reads

Constant Memory

Constant memory:

- Cached access to global memory
- Read-only
- Limited to 64kb for the application
- Shared by all threads
- Shared by instruction cache
- Cache is 8kb

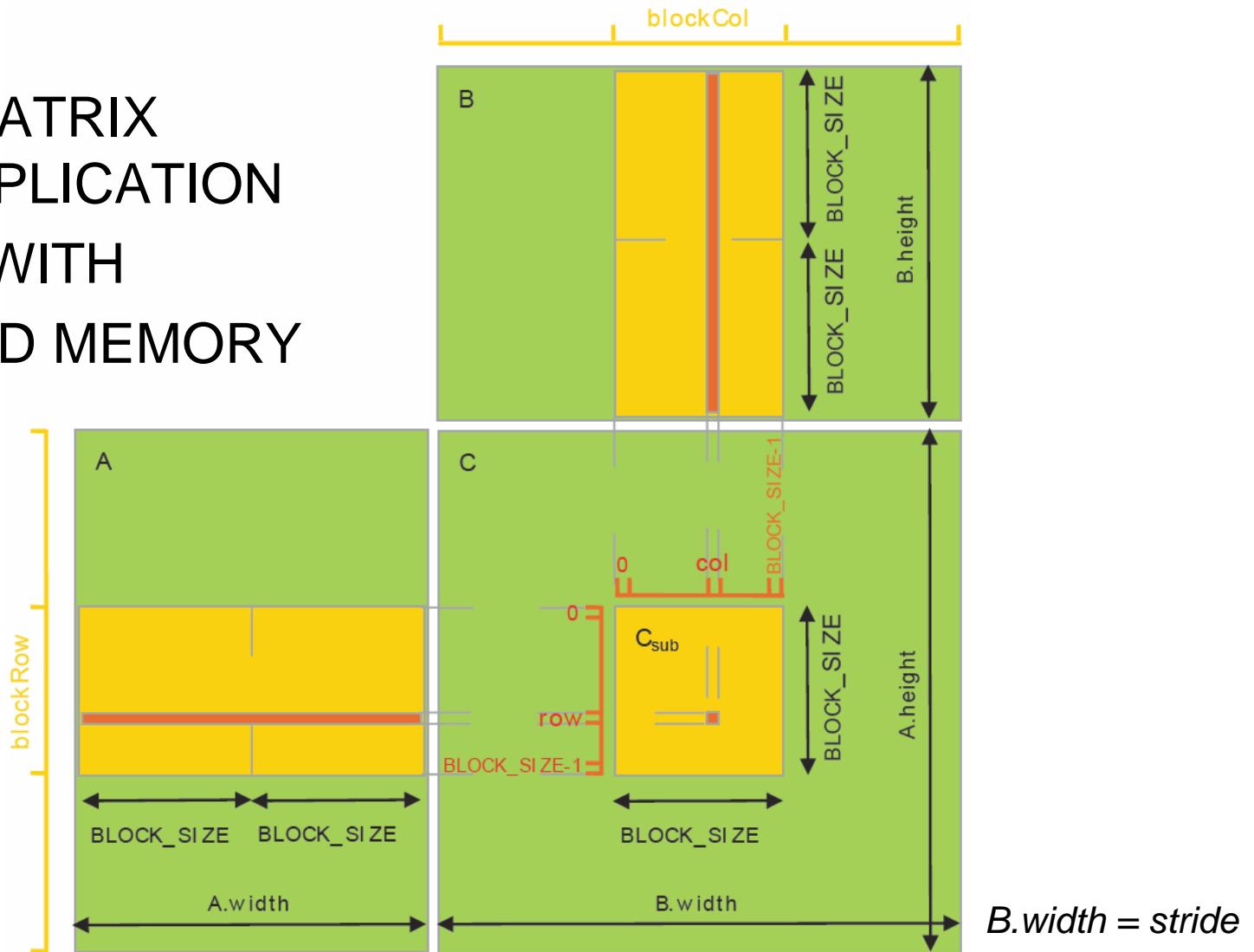
Texture Memory

Texture memory:

- Cached access to global memory
- Read-only
- L1 shared by multiple processors (depending on generation), L2 by all threads
- Optimized for 2D access
- L1 cache is 6kb to 8kb (depending on processor)

MATRIX MULTIPLICATION WITH SHARED MEMORY

Example



Example

```
// Matrices are stored in row-major order:  
// M(row, col) = *(M.elements + row * M.stride + col)  
typedef struct {  
    int width;           Larghezza della matrice di partenza,  
    int height;          necessaria per andare a capo  
    int stride;          anche nel caso delle sottomatrici  
    float* elements;  
} Matrix;  
  
// Get a matrix element  
__device__ float GetElement(const Matrix A, int row, int col) {  
    return A.elements[row * A.stride + col];  
}  
  
// Set a matrix element  
__device__ void SetElement(Matrix A, int row, int col, float value) {  
    A.elements[row * A.stride + col] = value;  
}
```

Example

```
// Get the BLOCK_SIZExBLOCK_SIZE sub-matrix Asub of A that is
// located col sub-matrices to the right and row sub-matrices down
// from the upper-left corner of A
__device__ Matrix GetSubMatrix(Matrix A, int row, int col) {
    Matrix Asub;
    Asub.width = BLOCK_SIZE;
    Asub.height = BLOCK_SIZE;
    Asub.stride = A.stride;
    Asub.elements = &A.elements[A.stride * BLOCK_SIZE * row + BLOCK_SIZE * col];
    return Asub;
}
// Thread block size
#define BLOCK_SIZE 16
// Forward declaration of the matrix multiplication kernel
__global__ void MatMulKernel(const Matrix, const Matrix, Matrix);
```

Example

```
// Matrix multiplication - Host code
// Matrix dimensions are assumed to be multiples of BLOCK_SIZE
void MatMul(const Matrix A, const Matrix B, Matrix C) {
    // Load A and B to device memory
    Matrix d_A;
    d_A.width = d_A.stride = A.width; d_A.height = A.height;
    size_t size = A.width * A.height * sizeof(float);
    cudaMalloc(&d_A.elements, size);
    cudaMemcpy(d_A.elements, A.elements, size, cudaMemcpyHostToDevice);
    Matrix d_B;
    d_B.width = d_B.stride = B.width; d_B.height = B.height;
    size = B.width * B.height * sizeof(float);
    cudaMalloc(&d_B.elements, size);
    cudaMemcpy(d_B.elements, B.elements, size, cudaMemcpyHostToDevice);
```

Example

```
// Allocate C in device memory
Matrix d_C;
d_C.width = d_C.stride = C.width; d_C.height = C.height;
size = C.width * C.height * sizeof(float);
cudaMalloc(&d_C.elements, size);

// Invoke kernel
dim3 dimBlock(BLOCK_SIZE, BLOCK_SIZE);
dim3 dimGrid(B.width / dimBlock.x, A.height / dimBlock.y);
MatMulKernel<<<dimGrid, dimBlock>>>(d_A, d_B, d_C);

// Read C from device memory
cudaMemcpy(C.elements, d_C.elements, size, cudaMemcpyDeviceToHost);

// Free device memory
cudaFree(d_A.elements);
cudaFree(d_B.elements);
cudaFree(d_C.elements);
}
```

Example

```
// Matrix multiplication kernel called by MatMul()
__global__ void MatMulKernel(Matrix A, Matrix B, Matrix C) {
    // Block row and column
    int blockRow = blockIdx.y;
    int blockCol = blockIdx.x;
    // Each thread block computes one sub-matrix Csub of C
    Matrix Csub = GetSubMatrix(C, blockRow, blockCol);
    // Each thread computes one element of Csub
    // by accumulating results into Cvalue
    float Cvalue = 0;
    // Thread row and column within Csub
    int row = threadIdx.y;
    int col = threadIdx.x;
```

Example

```
// Loop over all the sub-matrices of A and B that are required to compute Csub
// Multiply each pair of sub-matrices together and accumulate the results
for (int m = 0; m < (A.width / BLOCK_SIZE); ++m) {
    // Get sub-matrix Asub of A
    Matrix Asub = GetSubMatrix(A, blockRow, m);
    // Get sub-matrix Bsub of B
    Matrix Bsub = GetSubMatrix(B, m, blockCol);
    // Shared memory used to store Asub and Bsub respectively
    __shared__ float As[BLOCK_SIZE][BLOCK_SIZE];
    __shared__ float Bs[BLOCK_SIZE][BLOCK_SIZE];
    // Load Asub and Bsub from device memory to shared memory
    // Each thread loads one element of each sub-matrix
    As[row][col] = GetElement(Asub, row, col);
    Bs[row][col] = GetElement(Bsub, row, col);
    // Synchronize to make sure the sub-matrices are loaded
    // before starting the computation
    __syncthreads();
```

Example

```
// Multiply Asub and Bsub together
for (int e = 0; e < BLOCK_SIZE; ++e)
    Cvalue += As[row][e] * Bs[e][col];
    // Synchronize to make sure that the preceding
    // computation is done before loading two new
    // sub-matrices of A and B in the next iteration
    __syncthreads();
}

// Write Csub to device memory
// Each thread writes one element
SetElement(Csub, row, col, Cvalue);
}
```

Syncthreads

- Threads within a block can cooperate by sharing data through some shared memory and by synchronizing their execution to coordinate memory accesses.
- Synchronization points are specified in the kernel by calling `__syncthreads()`: a barrier at which all threads in the block must wait before any is allowed to proceed.
- For efficient cooperation, the shared memory is expected to be a low-latency memory near each processor core and `__syncthreads()` is expected to be lightweight.

SUPSI

Video processing e computer vision

Fondamenti di Multimedia Processing

Tiziano Leidi

14.12.2017

Cinematografia

La cinematografia è la scienza e l'arte della fotografia in movimento, in cui la luce visibile viene registrata in maniera ripetuta, chimicamente, per il tramite di un materiale fotosensibile, o elettronicamente, per il tramite di un sensore.



L'impiego di tecniche cinematografiche è frequente in molti ambiti sia commerciali, che scientifici o dell'intrattenimento.

Videocamere

Nelle videocamere, in maniera analoga alle fotocamere, una lente viene utilizzata per mettere a fuoco ripetutamente la luce riflessa dagli oggetti in immagini sulla superficie fotosensibile all'interno della camera.

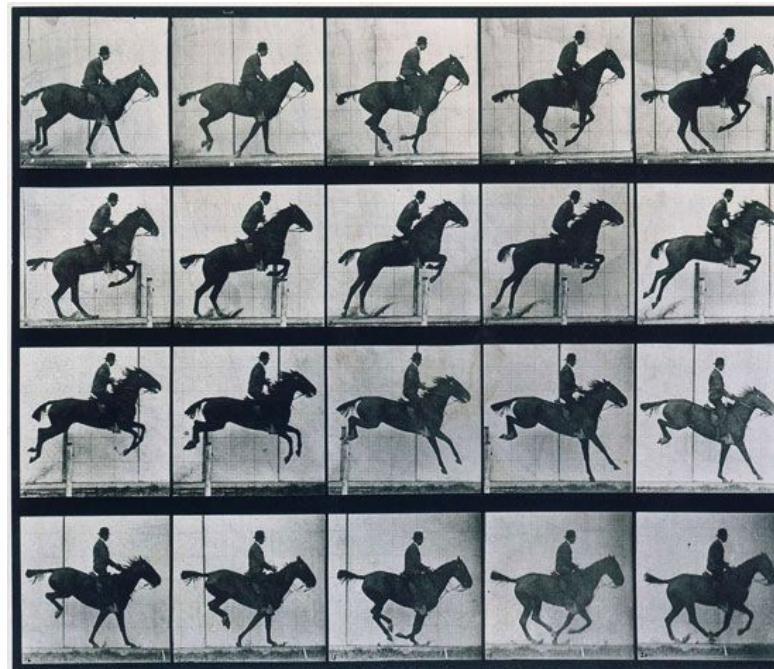
Vengono registrate sequenze di immagini.

Nel caso dell'emulsione fotografica, si tratta di una serie di immagini latenti invisibili sulla pellicola, che viene poi sviluppata chimicamente.

Nel caso d'impiego di sensore elettronico, viene prodotta una scarica elettrica per ogni pixel di ogni immagine, che viene successivamente processata elettronicamente e salvata all'interno di un file video.

Riproduzione

Le immagini presenti sulla pellicola sviluppata o all'interno del file video vengono successivamente riprodotte in rapida sequenza e proiettate su uno schermo cinematografico o su televisore/monitor, creando l'illusione del movimento.



Proiettore cinematografico

Un proiettore cinematografico è un dispositivo opto-meccanico per riprodurre film di immagini in movimento proiettandole su uno schermo.

Secondo la teoria del fenomeno beta-phi, il cervello ricostruisce un'esperienza di movimento apparente se gli viene presentata una sequenza di immagini statiche quasi identiche ad un frame rate maggiore di 10/12 immagini al secondo.



Formati dei film

I proiettori vengono classificati in base alla dimensione del film utilizzato.

I formati più tipici sono:

- per i film amatoriali: 8mm, super 8 e 9.5 mm
- per il cinema: 35mm, 70mm

Oggi esistono inoltre proiettori cinematografici digitali.

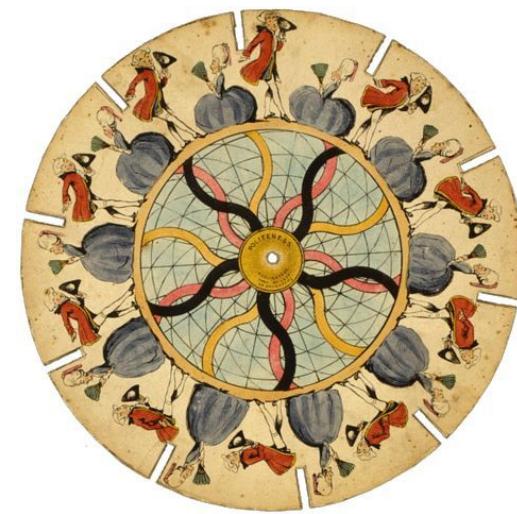
I produttori autorizzati sono: Sony, Barco, Christie and NEC. Sony utilizza la tecnologia proprietaria SXRD, mentre le rimanenti la tecnologia di Digital Light Processing (DLP) sviluppata da Texas Instruments (TI).

Cenni storici

I primi tentativi risalgono agli anni 1830, in cui sequenze di immagini in movimento venivano create utilizzando dischi rotanti.

L'invenzione della prima camera capace di registrare sequenze di immagini continue è del 1845, utilizzata per la meteorologia.

È a partire dal 1873 che si cominciano a vedere i primi casi di cinematografia vera e propria, applicata negli anni che seguono sia a sperimentazioni scientifiche, sia nell'ambito dell'intrattenimento.



Cenni storici

A Parigi nel dicembre del 1895, Louis and Auguste Lumière perfezionarono il cinematografo, un'apparecchiatura per registrare, stampare e riprodurre films.

I fratelli Lumière furono i primi a presentare proiezioni di film a pagamento ad un pubblico più grande di una singola persona.

Nel 1896, i primi cinema furono aperti in Francia (Parigi, Lione, Bordeaux, Nizza, Marsiglia), Italia (Roma, Milano, Napoli, Genova, Venezia, Bologna, Forlì), Bruxell e Londra.

In seguito, l'evoluzione fu massiccia, con l'introduzione del film a colori e sonoro fra il 1900 e il 1920 e l'avvento del digitale a partire dagli anni '80.

Funzionamento e storia delle videocamere

https://www.youtube.com/watch?v=_zpZ63_U-ws

Televisione

La televisione è un dispositivo che combina un tuner, un display e degli altoparlanti per permettere la visione di trasmissioni televisive.

I primi modelli avevano un funzionamento di tipo meccanico, sfruttando un tubo a neon, un disco rotante e un vetro d'ingrandimento. In seguito la tecnologia si sviluppò principalmente su tubi a raggi catodici.



Cenni storici

A livello sperimentale, le prime televisioni furono sviluppate alla fine degli anni '20, ma raggiunsero il mercato solo dopo diversi anni. È con la fine della seconda guerra mondiale che il broadcasting televisivo in bianco e nero divenne popolare negli Stati Uniti e in Inghilterra.

Negli anni '50, la televisione era lo strumento principale per influenzare l'opinione pubblica.

Nella metà degli anni '60 fu introdotto il broadcasting a colori.

A partire dagli anni '90 l'evoluzione accelerò con l'introduzione della televisione digitale.

Nel 2013, il 79% delle case del nostro pianeta aveva una televisione.

Cenni storici

THE EVOLUTION OF TELEVISION



**1928
OCTAGON**

General Electric made the Octagon in 1928 as part of their experimental TV program in Schenectady, New York. The first TV drama, the Queen's Messenger, was produced in September of that year by GE.



**1930
BAIRD**

The Baird Televisor was made by Plessey in England from 1930 through the early 30s. It was the first television receiver sold to the public.



**1936
EMYVISOR & COSSOR**

In the year 1936 two T.V was invented the first was the Emyvisor which picture in black & white, the next TV was the Cossor which not only show picture in black and white but also in color.



**1938
MARCONI**

Here's a 1938 Marconi 707 Television & All Wave Radio Receiver. Measuring 26" x 19" x 19" and weighing more than 100 pounds, the set was actually considered to be somewhat compact in its day, and though its 7 inch screen would be regarded as minuscule by contemporary standards, in 1938 it was not insubstantial.



**1939
RCA**

RCA introduced television to the American public at the 1939 World's Fair. Before the fair, they published a brochure for their dealers to explain television.



**1946
RCA**

The RCA 630TS television became an immediate hit when it was introduced in 1946, right after World War II.



**1948
MOTOROLA**

Motorola's "Golden View" was the most popular 7-inch television in the late 1940s and early 1950s. It came in both tabletop and portable cabinets and it was one of the cheapest TVs available at the time.



**1949
RAYTHEON**

The Raytheon M-1101 is an American TV set manufactured in Raytheon's Belmont Radio plant in Chicago on October 1949, the CRT face was more or less masked to give a rectangular appearance), this style of TV is known as "porthole", like the "windows" on a ship.



**1953
SHARP**

Sharp started producing as the first Japanese television in mass production. The 14-inch TV was the standard in the first Japanese households for years. With its wooden frame, it precisely met the design aesthetic taste of the fifties.



**1958
PHILCO**

Is this the ultimate TV? Love it or hate it, the Philco Predicta television is unarguably one of the design icons of the 20th Century.



**1962
MEIDENSHA**

Meidensha TV's were really contemporary in style and design. The wooden frame and high voltage tubes were considered as a great combo back then.



**1973
PHILCO-FORD**

1973 Philco-Ford - Model 8450ETG - One of the last 'vacuum tube' sets. It was in this time period that the American television set industry migrated to a transistorized TV chassis.



**1998
SONY**

The Sony TV was created in 1998, it was the first television that had a built in VCR and DVD player. The Sony TV had better picture and a lot more channel with color.



**2007
SAMSUNG**

Samsung emerged as one of the largest flat panel TV producer worldwide. Samsung also introduced a ten-millimeter thick only, 40-inch LCD television panel for the first time too.



**2014
SAMSUNG**

Samsung started selling commercial curved smart TVs. In the IFA2014, Samsung also displayed the first bendable TV with 5,120 x 2,160 resolutions.

Supporti per il salvataggio

L'evoluzione negli anni di vari tipi di supporti per il salvataggio di filmati e degli associati registratori/riproduttori video ha reso possibile l'immagazzinamento di materiale cinematografico e televisivo per la riproduzione a casa.

I formati più conosciuti: Betamax, cassette VHS, DVD e dischi Blu-ray ad alta definizione.

Tecnologie per display

- CRT (Cathode Ray Tube): è un tubo sottovuoto contenente uno o più emettitori di elettroni combinato con uno schermo fluorescente. Per creare le immagini accelera e deflette i fasci di elettroni sullo schermo.
- DLP (Digital Light Processing): è una tecnologia per video proiettori che sfrutta micro-specchi digitali. Viene normalmente impiegata per i proiettori utilizzati negli ambiti commerciali o dell'insegnamento.
- PDP (Plasma Display Panel): è un pannello piatto che utilizza piccole celle contenenti gas ionizzato caricato elettricamente. Utilizzato principalmente per grossi schermi.

Tecnologie per display

- LCD (Liquid Crystal Display): è un pannello piatto che utilizza cristalli liquidi modulati dalla luce. I cristalli liquidi non emettono direttamente la luce. Per produrre la luce vengono quindi utilizzati dei riflettori o delle backlight (tipicamente delle luci fluorescenti nelle LCD-TV o dei LED nelle LED-TV).
- OLED (Organic Light-Emitting Diode): è una tecnologia che per emettere la luce sfrutta LED, che come elemento elettro-luminescente hanno uno strato di materiale organico capace di produrre differenti tipi di luce in risposta alla sollecitazione elettrica ricevuta.

Risoluzione

Nel cinema digitale, la risoluzione è normalmente rappresentata dalla quantità di pixel orizzontali. Ad esempio: 2K (2048×1080 quindi 2.2 megapixels) o 4K (4096×2160 quindi 8.8 megapixels).

Risoluzione

Nelle televisioni e nei monitor le risoluzioni più frequenti sono:

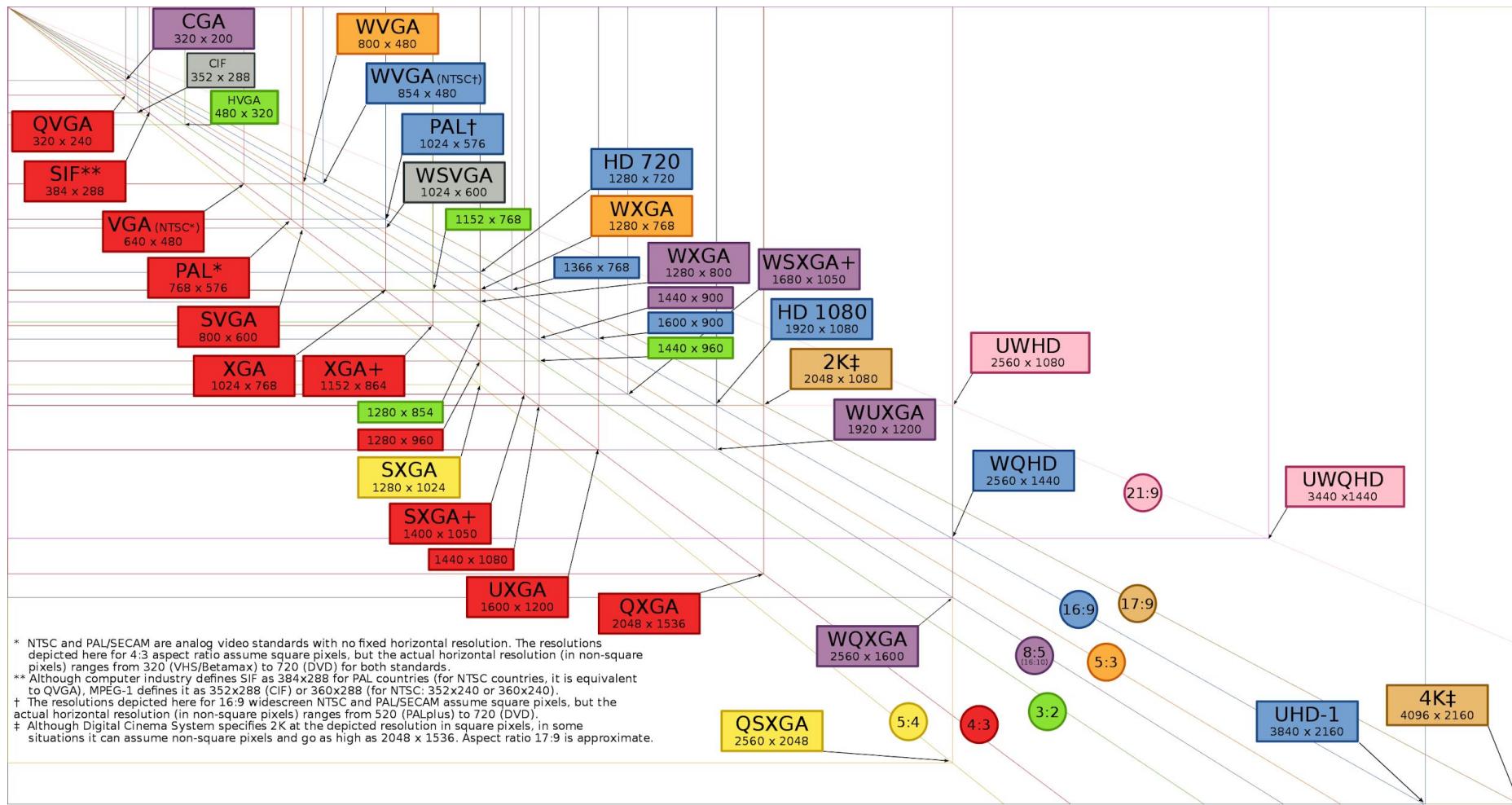
- SD (Standard Definition): 576i, sistema PAL/SECAM Europeo con 576 linee interlacciate; 480i, sistema NTSC Americano con 480 linee interlacciate. Le videocassette VHS possono essere considerate SD, hanno una risoluzione approssimativa di 480i/576i×360.
- HD (High Definition): 720p - 1280×720p progressive scan quindi 921'600 pixels; 1080i - 1920×1080i interlacciato quindi 1'036'800 pixels; 1080p - 1920×1080p quindi 2'073'600 pixels.
- UHD (Ultra High Definition): 4K - 3840×2160 progressive scan; True 4K - 4096×2160; 8K - 7680×4320; True 8K - 8192×4320.

Aspect ratio

L'aspect ratio di un'immagine descrive la proporzione fra la sua larghezza e la sua altezza.

- 4:3 è il formato utilizzato dall'invenzione della cinematografia, poi nei televisori e nei monitor. I film a 35 mm del cinema muto erano in 4:3.
- 16:9 è il formato standard internazionale dell'HDTV e del DVD.
- 21:9 è un formato cinematografico recente utilizzato oggi negli ultra-wide monitors.

Risoluzione e aspect ratio



Frame rate

Il frame rate, espresso in frames al secondo (FPS), è la frequenza con cui le immagini vengono mostrate in un filmato. La sensibilità temporale della visione umana varia a dipendenza delle caratteristiche dello stimolo visivo ed è differente per ogni individuo.

Di norma, le immagini vengono mostrate a velocità costante. Nel cinema la velocità tradizionalmente impiegata è 24 frames per secondo, nella televisione PAL/NTSC il framerate è di 25/30. Oggi in diversi contesti si usano frame-rate più alti.

Video processing

In maniera simile all'image processing, nel video processing si utilizzano tecniche di filtraggio dei segnali applicandole a streams di tipo video.

Le tecniche di video processing vengono impiegate in svariati ambiti, compresi quelli della televisione e del cinema e comprendono, fra le altre: modifica dell'aspect ratio; zoom digitale; aggiustamento di brightness, contrast, hue, saturation, sharpness, gamma; conversione del frame rate; aggiustamento del colore; riduzione del rumore; tecniche di upscaling; e molto altro.

Video processing

Nel video processing vengono impiegate una moltitudine di tecniche dell'image processing.

Tutte le tecniche attuate a livello di singola immagine vengono dette tecniche di intra-frame processing.

Invece, le tecniche che sfruttano l'informazione temporale esistente fra più di una immagine in sequenza vengono dette tecniche di inter-frame processing.

Formati di codifica e compressione video

I formati di codifica e compressione video sono formati di rappresentazione dei dati usati per il salvataggio e nella trasmissione di contenuti video digitali.

Esempi di formati video sono: MPEG-2 Part 2, MPEG-4 Part 2, H.264 (MPEG-4 Part 10), HEVC e RealVideo RV40.

I formati sono definiti in una specifica. Un'implementazione hardware o software di un formato video è un video codec.

Esempi di video codec sono Xvid e OpenH264, che sono due delle varie implementazioni di MPEG-4 Part 2, rispettivamente H.264.

Formati container per il multimedia

I contenuti video codificati in un particolare formato video vengono normalmente combinati con una stream audio tramite un formato container per il multimedia come AVI, MP4, FLV, RealMedia, Matroska, o QuickTime.

Di conseguenza, l'utente non manipola files di tipo H.264 ma piuttosto .mp4, che è un container MP4 con video codificato in H.264 e audio di norma in formato AAC.

Compressione video

I filmati video vengono generalmente compressi utilizzando compressioni di tipo lossy, capaci di comprimere maggiormente dei compressori lossless.

Esistono compressori disegnati esplicitamente per compressione lossy rispettivamente lossless, mentre alcuni formati video come il Dirac e l'H.264 permettono entrambe le tipologie.

Una sottoclasse di formati di codifica video relativamente semplice è quella dei formati intra-frame, in cui la compressione viene applicata esclusivamente alle singole immagini senza approfittare della correlazione tra immagini successive. Un esempio è il Motion JPEG.

MPEG

Il Moving Picture Experts Group (MPEG) è un gruppo di lavoro di autorità che è stato formato da ISO e IEC per stabilire uno standard per la compressione e la trasmissione audio e video.

Lo standard MPEG consiste in una moltitudine di parti, ognuna delle quali copre un determinato aspetto della specifica.

Negli anni lo standard si è evoluto, di conseguenza alcune parti possono rivelarsi in parziale sovrapposizione con quelle precedenti.

H.264 - MPEG-4 Part 10

H.264 anche chiamato MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) è un formato standard orientato a blocchi e basato sulla compensazione del movimento per la compressione video. Dal 2014 è uno dei formati più utilizzati per la registrazione e la compressione di contenuti video. Supporta risoluzioni fino a 4096×2304 , incluso il 4K UHD.

Inter-frame prediction

I filmati video contengono molta informazione ridondante spaziale, ma soprattutto temporale. Negli algoritmi di compressione come H.264 viene sfruttata questa ridondanza con tecniche di inter-frame prediction, in particolare avvantaggiandosi della correlazione tra immagini vicine nella sequenza.

Inter-frame prediction



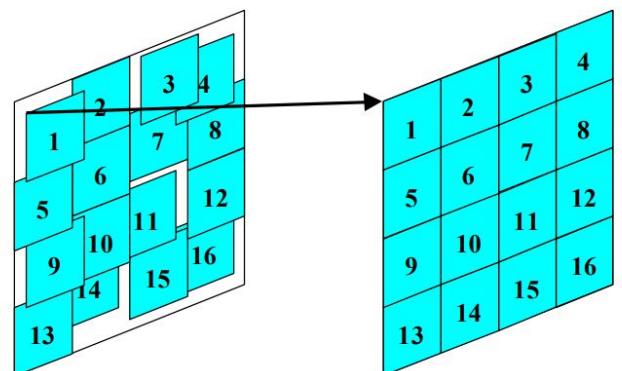
Previously Coded Frame
(Reference Frame)



Current Frame
(To be Predicted)

Block-matching overview:

- 1) Split current frame into 16x16-pixel blocks
- 2) Find best match for each block from prior frame



Reference Frame

Predicted Frame

Inter-frame prediction

Prediction of
Current Frame



Prediction Error
(Residual)



Motion JPEG

Il Motion JPEG (M-JPEG o MJPEG) è un formato di compressione video in cui ogni frame è compresso separatamente come immagine JPEG.

Viene principalmente utilizzato nelle videocamere, in particolare nelle webcam.

M-JPEG usa uno schema di compressione intraframe. È quindi meno computationally intensive delle soluzioni che applicano interframe prediction.

Motion JPEG

Le compressioni come MPEG2 e H.264/MPEG-4 AVC, possono raggiungere rapporti di compressione di 1:50 o anche meglio. M-JPEG ha un'efficienza limitata a 1:20 o meno, ma necessita di meno risorse di processing (tempo CPU e memoria).

Inoltre, l'M-JPEG tollera cambiamenti veloci nella videotream, cambiamenti per cui gli schemi di compressione interframe possono presentare perdite di qualità inaccettabili.

Computer Vision

Le soluzioni di computer vision hanno l'obiettivo di emulare artificialmente le capacità del sistema visivo umano.

La computer vision comprende tecnologie e metodi per acquisire, modificare ed analizzare singole immagini o sequenze di immagini digitali, con lo scopo ultimo di estrapolare informazioni per poter prendere decisioni in maniera automatica (possibilmente senza supervisione umana).

Computer Vision

I principali campi d'applicazione sono:

- ambito medico: ha lo scopo di produrre diagnosi medica di un paziente. I dati sono immagini di microscopi, immagini a raggi X, immagini acquisite con tomografia, angiografia, ultrasuoni.
- industria: l'informazione viene estratta con lo scopo di supportare un processo manifatturiero.
- applicazioni militari: individuazione di soldati e veicoli nemici, guida di missili.
- veicoli autonomi: includono sottomarini, robots con ruote, auto, camion, droni e altri veicoli aerei.

Attività della computer vision

- Riconoscimento: determinare se una classe di oggetti, caratteristiche o attività è presente. Risolto in maniera robusta dall'essere umano, ma non dalla computer vision (caso generale: oggetti arbitrari in situazioni arbitrarie).
- Identificazione: riconoscere un'istanza individuale di un oggetto. Esempi: faccia di una persona, impronta digitale.
- Detection: individuare condizioni specifiche. Esempio: cellule o tessuti abnormali nelle immagini medicali.
- Riconoscimento ottico di caratteri (OCR): identificare caratteri in testo stampato o scritto a mano.
- Image retrieval di contenuti: trovare tutte le immagini con un determinato contenuto comune all'interno di un set.

Attività della computer vision

- Stima di posizione e orientamento: identificare la posizione o l'orientamento di un oggetto in relazione alla posizione della camera. Esempio d'applicazione: soluzione di computer vision a supporto di un braccio automatizzato.
- Stima del movimento: identificare la direzione e la velocità degli oggetti.
- Tracking: seguire i movimenti degli oggetti.
- Ricostruzione della scena: data una o più immagini di una scena, rispettivamente un video, ricostruire un modello 3D (semplificato o più complesso) della scena.

Struttura di un sistema di computer vision

Un sistema di computer vision è composto da:

- acquisizione dell'immagine: produzione di immagini digitali tramite sensori o altri tipi di strumenti (ad esempio radar). Le immagini o i filmati possono essere di tipo bidimensionale o 3D.
- Pre-processing: prima dell'estrazione dell'informazione, può essere necessario trattare le immagini per la rimozione del rumore, l'adeguamento del contrasto, l'adattamento della scala di rappresentazione, ecc.
- Estrazione delle feature: estrazione di caratteristiche a vari livelli di complessità. Ad esempio: linee, contorni, spigoli, blob di punti, forme, textures, movimento.

Struttura di un sistema di computer vision

- Detection/Segmentation: decisione su quali punti o regioni dell'immagine sono rilevanti per i successivi passi di processing.
- Processing di tipo high-level: a questo punto le immagini sono state segmentate in piccoli set di dati, ad esempio set di punti o regioni, supposti di contenere un oggetto o caratteristica. Il processing successivo si occupa di:
 - verificare che i dati soddisfino determinate assunzioni.
 - stima di determinati parametri specifici (posizione, dimensione, ...).
 - classificazione degli oggetti in categorie.

Background subtraction

La tecnica di background subtraction, anche conosciuta come foreground detection, ha l'obiettivo di estrarre elementi in primo piano (ad esempio esseri umani o automobili) all'interno di un filmato.

Viene utilizzata principalmente nel caso di acquisizione di immagini da videocamere statiche.

Il principio di base è quello di identificare l'oggetto in movimento confrontando il frame corrente con un frame di riferimento, di norma chiamato "background image", o "background model".

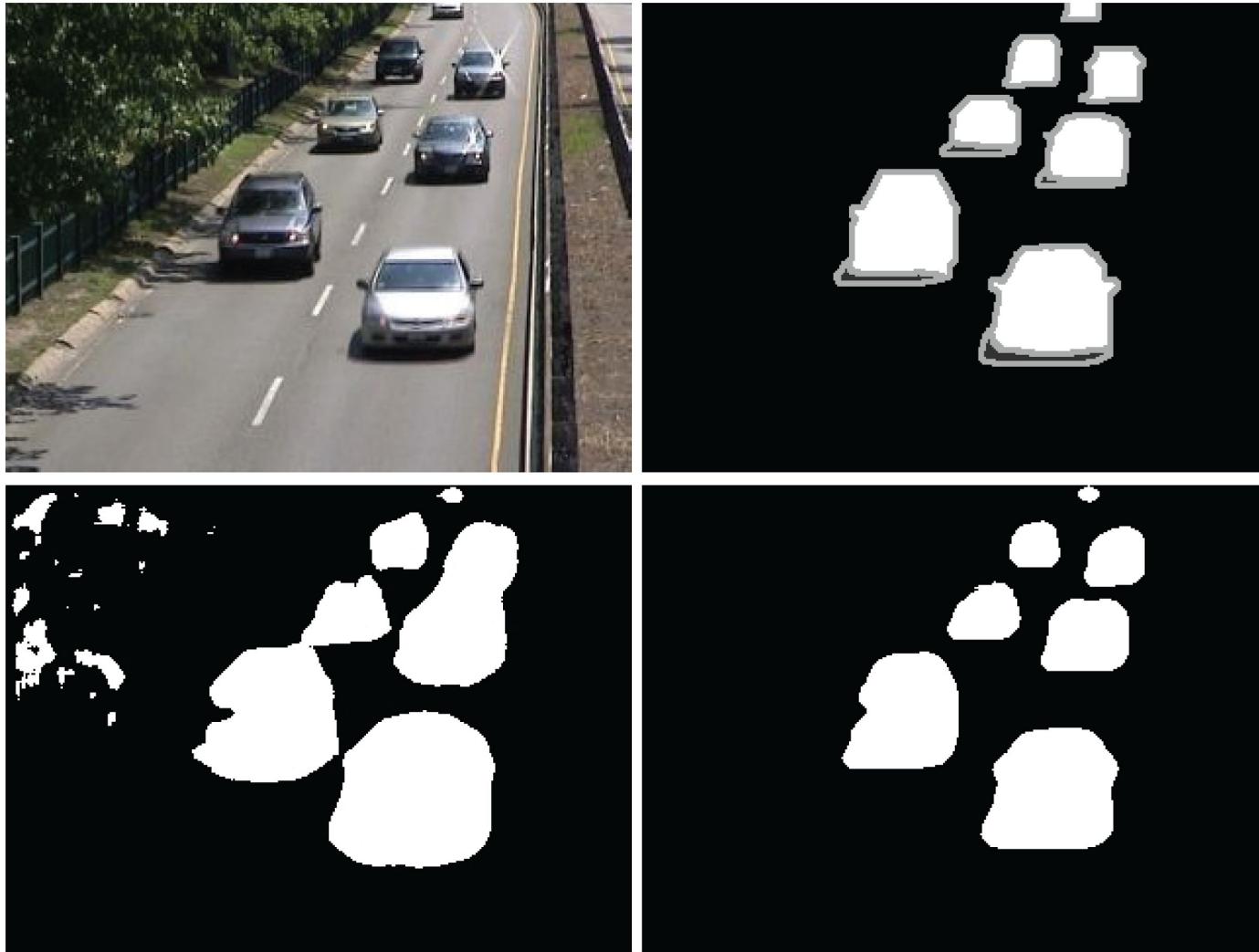
Background subtraction

La tecnica di background subtraction si basa su un'ipotesi che spesso non è applicabile ad ambienti reali. Un algoritmo di background subtraction robusto dovrebbe essere capace di gestire cambi di luce, effetti del vento o della pioggia, elementi di ingombro e modifiche della scena a lungo termine.

Esistono varie tecniche di background subtraction:

- Frame differencing
- Mean filter
- Gaussian average

Background subtraction



Frame Differencing

Segmenta gli oggetti utilizzando la sottrazione d'immagine per ogni pixel in $I(t)$ con il corrispondente pixel nella stessa posizione dell'immagine di background B .

$$P[F(t)] = P[I(t)] - P[B]$$

L'immagine risultante presenterà intensità luminosa solo per i pixels che hanno subito modifiche nei due frames.

Quest'approccio funziona solo se i pixel di background sono statici. Solitamente sulle immagini risultanti viene applicato un threshold.

$$|P[F(t)] - P[F(t + 1)]| > \text{Threshold}$$

Video motion tracking

Lo scopo delle tecniche di video tracking è di associare gli oggetti in frames video consecutivi. L'associazione può rivelarsi complessa se gli oggetti si muovono rapidamente rispetto al frame rate.

Un'altra situazione che accresce la complessità del problema è nel caso di cambi di direzione. In queste situazioni possono essere impiegati modelli di moto che descrivono e quindi vincolano i cambiamenti di moto accettabili.

Video motion tracking

Ci sono due componenti principali di un sistema di visual tracking:

- rappresentazione e localizzazione del target,
- filtraggio e associazione dei dati.

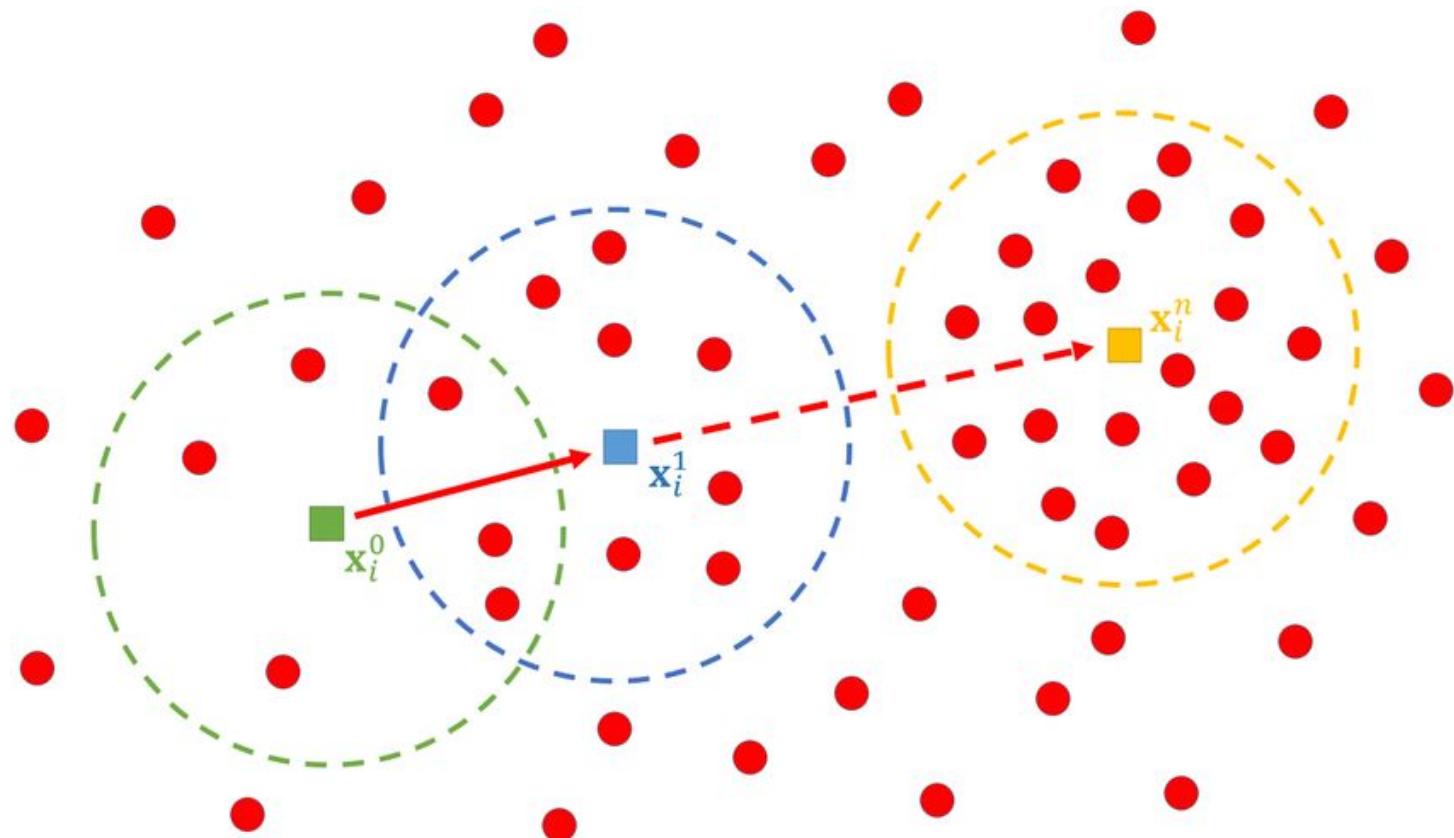
Per la prima parte vengono usate tecniche di kernel-based tracking, come il mean shift, o di contour tracking. Per la seconda parte vengono applicati Kalman filter o Particle filter.

Mean Shift

Il mean shift è una tecnica di analisi non parametrica per identificare i maxima (detti modes) di una funzione di densità. Viene detto algoritmo di mode-seeking.

Ha un comportamento iterativo che massimizza una misura di similitudine. Viene impiegato in prevalenza per l'analisi dei clusters.

Mean Shift



Mean Shift

La variante più semplice impiegata per il visual tracking crea una confidence map della nuova immagine, basandosi sugli histogrammi dei colori dell'oggetto presente nell'immagine precedente. Utilizza l'algoritmo di mean shift per identificare il picco in prossimità della vecchia posizione dell'oggetto.

La confidence map è una funzione di densità di probabilità. Ad ogni pixel della nuova immagine viene assegnata la probabilità che pixels di quel colore siano presenti nell'oggetto.

L'idea viene estesa in varianti come il kernel-based object tracking e il CAMshift.

SUPSI

Audio e video streaming

Fondamenti di Multimedia Processing

Tiziano Leidi

11.01.2018

Trasmissione di contenuti multimediali

La trasmissione di contenuti multimediali su lunghe distanze è una delle sfide tecnologiche con cui l'essere umano si sta confrontando da tempo.

Le tecniche impiegate in passato sono innumerevoli. Oggi la convergenza è nella direzione di soluzioni di streaming audio/video che sfruttano Internet come canale di comunicazione.

Telefono

Il telefono è stato il primo degli strumenti che ha reso possibile la comunicazione di un segnale audio attraverso una lunga distanza.

Il telefono converte suono, di norma la voce umana, in un segnale elettrico trasmesso via cavo (o qualche altro mezzo di trasmissione) ad un altro telefono che si occuperà della riproduzione.



Cenni storici

Prima dell'invenzione dei telefoni elettromagnetici, esistevano dei dispositivi meccanico-acustici capaci di trasmettere la voce (e la musica) su distanze maggiori di quelle possibili per propagazione diretta. Sfruttavano tubi di trasmissione e altre soluzioni fisiche.

Il telefono è emerso dall'evoluzione del telegrafo elettrico. Varie persone contribuirono alla sua invenzione, inclusi Antonio Meucci, Elisha Grey e Thomas Edison.

Cenni storici

Nel 1876, Alexander Graham Bell fu il primo a depositare un brevetto negli Stati Uniti per un dispositivo capace di generare in tempo reale a distanza una replica chiaramente intellegibile della voce umana.

In seguito l'evoluzione fu ampia, sia in termini tecnologici che di modalità di utilizzo. Fra i principali cambiamenti vanno sicuramente citati la telefonia mobile e gli smartphone, ma anche la telefonia over IP.

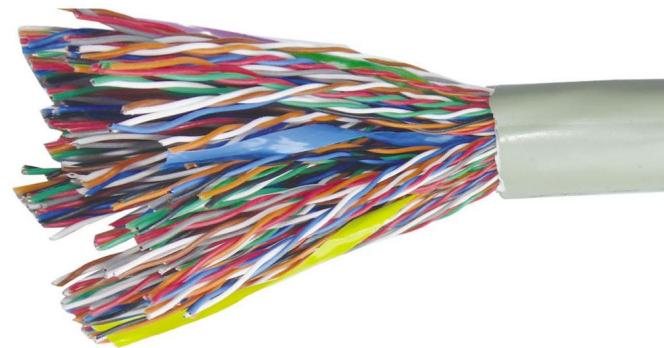
Linee telefoniche

Una linea telefonica è un singolo cavo fisico (o altro mezzo di comunicazione) che collega un telefono alla rete telefonica.

Ogni linea è di norma associata ad un numero di telefono.

Oggi le linee telefoniche vengono usate soprattutto per la comunicazione Internet tramite Digital Subscriber Line (DSL).

Di norma, i cavi sono in rame (in alcuni casi alluminio) che in passato venivano appesi a pali del telefono in coppie distanziate o come doppino. Le linee odierne sono sotterranee.



Cavi di tipo twisted pair

UTP Category	Description
CAT1	Up to 1Mbps, Old Telephone Cable
CAT2	Up to 4Mbps, Token Ring Networks
CAT3	Up to 10Mbps, Token Ring & 10BASE-T Ethernet
CAT4	Up to 16Mbps, Token Ring Networks
CAT5	Up to 100Mbps, Ethernet, Fast Ethernet, Token Ring
CAT5e	Up to 1Gbps, Ethernet, Fast Ethernet, Gigabit Ethernet
CAT6	Up to 10Gbps, Gigabit Ethernet, 10G Ethernet(55 meters)
CAT6a	Up to 10Gbps, Gigabit Ethernet, 10G Ethernet(55 meters)
CAT7	Up to 10Gbps, Gigabit Ethernet, 10G Ethernet(100 meters)

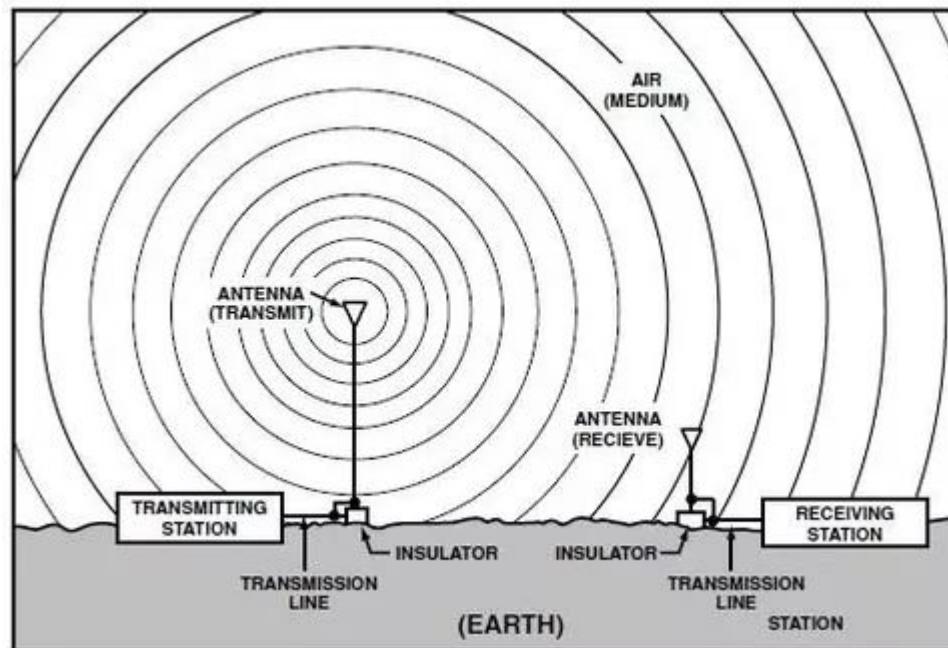
Trasmissione radio

La trasmissione radio è la tecnologia che sfrutta le onde radio per trasmettere informazione (voce, suono, ...), modulando sistematicamente le proprietà di onde elettromagnetiche trasmesse nello spazio. Come proprietà vengono sfruttate: ampiezza, frequenza, fase, lunghezza di impulso.

Quando le onde incontrano un conduttore elettrico, il campo oscillante induce una corrente alternata nel conduttore. L'informazione contenuta può essere estratta, ricostruendo il segnale originale.

Trasmissione radio

Per la trasmissione e la ricezione vengono utilizzate delle antenne: array di conduttori connessi elettricamente ai trasmettitori o ai ricevitori.



Trasmissione radio

Per le trasmissioni ad onde radio viene sfruttato il range dello spettro fra i 3 kHz e i 300 GHz.

Name	Frequency (Hz) (Wavelength)	Photon energy (eV)
Gamma ray	> 30 EHz (0.01 nm)	124 keV - 300+ GeV
X-Ray	30 EHz - 30 PHz (0.01 nm - 10 nm)	124 eV to 120 keV
Ultraviolet	30 PHz - 750 THz (10 nm - 400 nm)	3.1 eV to 124 eV
Visible	750 THz - 428.5 THz (400 nm - 700 nm)	1.7 eV - 3.1 eV
Infrared	428.5 THz - 300 GHz (700 nm - 1 mm)	1.24 meV - 1.7 eV
Microwave	300 GHz - 300 MHz (1 mm - 1 m)	1.24 μ eV - 1.24 meV
Radio	300 MHz - 3 kHz (1 m - 100 km)	12.4 feV - 1.24 meV

Cenni storici

L'idea della comunicazione senza fili risale agli anni intorno al 1830 in cui venivano condotti esperimenti di telegrafia wireless mediante trasmissione sul suolo, l'acqua o addirittura i binari del treno.

Nel 1894 l'inventore Italiano Guglielmo Marconi costruì il primo telegrafo senza fili, basato sulla tecnologia di trasmissione ad onde radio (trasmissione di onde Hertziane nell'aria), che ebbe successo commerciale.

A partire da quel momento la tecnologia radio venne sfruttata per una moltitudine di impieghi, inclusa la trasmissione di suono e video per radio e televisione.

Broadcasting

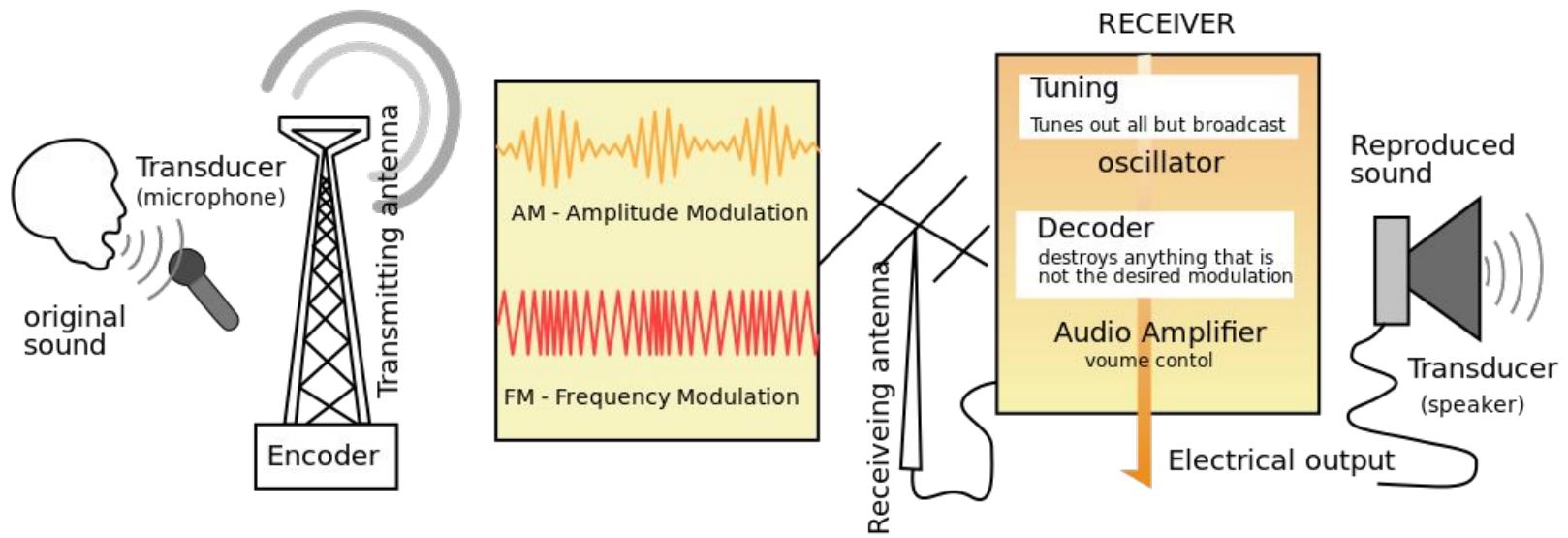
Il broadcasting è la distribuzione di contenuti audio o video ad un audience delocalizzata tramite un mezzo di comunicazione di massa, tipicamente usando le onde radio, in un modello uno a molti.

Le soluzioni di telecomunicazione precedenti al broadcasting erano caratterizzate da un modello uno a uno (ad esempio il telefono).

Il broadcasting ha avuto i suoi inizi con la radio AM, popolare a partire dagli anni 20 grazie all'introduzione dei trasmettitori e ricevitori radio a valvole.

Radio broadcasting

Con radio broadcasting si intende la trasmissione di contenuti audio (voce e musica) tramite tecnologia radio. Mentre in passato la trasmissione veniva fatta solo tramite onde radio, oggi può passare anche via cavo o più genericamente in Internet.



AM e FM

Le stazioni radio AM sono state le prime a venire sviluppate. AM significa modulazione d'ampiezza, una metodologia di broadcasting tramite onde radio in cui viene variata l'ampiezza del segnale trasmesso in corrispondenza delle variazioni d'ampiezza del segnale sonoro. Viene principalmente usata la banda delle onde medie.

Le stazioni radio FM sfruttano invece la modulazione di frequenza nella banda VHF. La modulazione di frequenza è stata inventata da Edwin Howard Armstrong per ovviare al problema di radio-frequency interference (RFI), che produce rumore nella ricezione AM.

AM e FM

Band	Frequency range	Wavelength range
Extremely low frequency (ELF)	< 3 kHz	>100 km
Very low frequency (VLF)	3 - 30 Hz	10 - 100 km
Low frequency(LF)	30 - 300 kHz	1 - 10 km
Medium frequency (MF)	300 kHz - 3 MHz	100m - 1km
High frequency (HF)	3 - 30 MHz	10 - 100m
Very high frequency (VHF)	30 - 300 MHz	1 - 10m
Ultra high frequency (UHF)	300 MHz - 3 GHz	10cm - 1m
Super high frequency (SHF)	3 - 30 GHz	1 - 10cm
Extremely high frequency (EHF)	30 - 300 GHz	1mm - 1cm

Televisione

Nel broadcasting televisivo, oltre al suono vengono trasmesse anche immagini in movimento. Oggi la televisione è diventata un mass media per l'intrattenimento, le news, la politica, la pubblicità, ecc.

La programmazione viene distribuita da differenti stazioni televisive (dette canali) con licenza di distribuire in una determinata banda di frequenza. All'inizio la trasmissione ad onde radio era l'unico mezzo sfruttato, progressivamente sostituito dall'avvento della televisione via cavo.

Oggi esistono alcune alternative come la televisione digitale terrestre, la televisione via satellite e la televisione over IP.

Televisione via cavo

La televisione via cavo è un sistema di distribuzione della programmazione televisiva tramite segnali a radiofrequenza trasmessi via cavi coassiali in sostituzione delle onde radio. Inizialmente la trasmissione era di tipo analogico, poi si è passati al digitale.

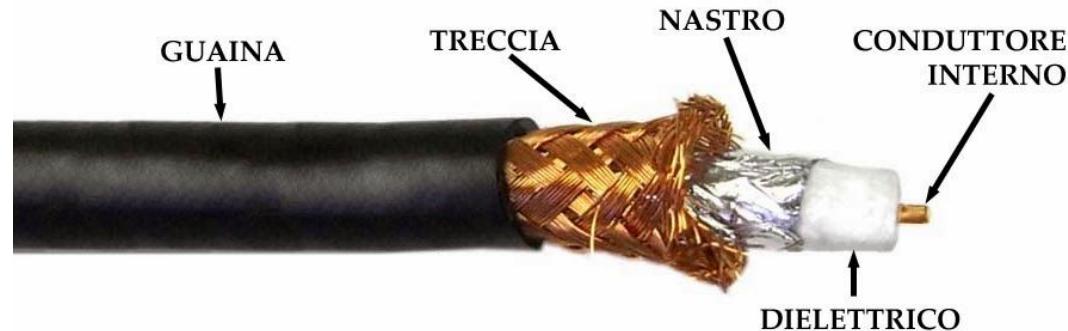
Oggi il cavo viene sfruttato anche per servizi alternativi come quelli telefonici, la radio FM e l'high-speed Internet.

Alcuni canali televisivi sono nati specifici per la televisione via cavo (e via satellite), come ad esempio: HBO, MTV, Cartoon Network, Eurosport e CNN.

Cavo coassiale

Un cavo coassiale è composto da un conduttore interno circondato da uno strato tubolare d'isolazione. Si chiama coassiale perchè il conduttore e la schermatura condividono un medesimo asse geometrico. Ha capacità di trasmissione 80 volte maggiore a quella di un cavo twisted-pair.

Le dimensioni di un cavo coassiale vengono controllate per fornire uno spazio di conduzione costante, indispensabile per il funzionamento efficace della linea di trasmissione.



Televisione via satellite

La televisione via satellite è un servizio che distribuisce la programmazione televisiva direttamente alle case dei telespettatori da un satellite di telecomunicazione che orbita attorno alla terra.

Il segnale viene ricevuto con un'antenna parabolica. Mentre in passato si usavano grosse parabole per la trasmissione di segnali analogici, i sistemi moderni sono digitali e trasmettono nella banda di frequenze fra i 12 e i 18 GHz. Richiedono quindi una parabola più piccola di un metro di diametro.

Digitale terrestre (radio)

Il broadcasting radio digitale è emerso inizialmente in Europa e poi si è diffuso in molte regioni del mondo. Il sistema utilizzato più comunemente è il Digital Audio Broadcasting (DAB) di cui ne esistono alcune varianti, incluso il DAB+. La radio analogica (AM e FM) è però ancora utilizzata. Inoltre, la competizione della radio IP è forte.

Nel broadcasting digitale, il segnale analogico viene digitalizzato e compresso con formati come l'MP2, poi trasmesso con schemi di modulazione digitale. L'obiettivo è di aumentare il numero di programmi presenti in una determinata banda di spettro e di migliorarne la qualità.

Digitale terrestre (televisione)

In maniera simile, la televisione digitale terrestre (DTTV o DTT) sfrutta le onde radio terrestri per trasmettere segnali TV digitali. Rispetto all'analogico, i vantaggi sono innumerevoli, inclusi un miglior sfruttamento dello spettro di frequenze e migliore qualità di trasmissione.

Esistono quattro formati principali: ATSC, DVB, ISDB e DTMB. Dal 2010 questi formati hanno sostituito praticamente ovunque i sistemi analogici NTSC, PAL e SECAM.

Radio e televisione Internet

La radio e la televisione Internet sono forme di distribuzione digitale che sfruttano Internet come mezzo di trasmissione. Questo tipo di approccio presenta diversi vantaggi rispetto a soluzioni più tradizionali come il digitale terrestre, la via cavo e la via satellite.

Per la telefonia, la radio e la televisione in Internet esistono soluzioni specifiche over IP:

- il voice over Internet protocol (VoIP),
- il radio over Internet protocol (RoIP),
- l'Internet Protocol Television (IPTV).

Radio e televisione over IP

Oggi, molti distributori radio e TV hanno sostituito le soluzioni tradizionali con quelle over IP mediante set-top boxes o altro equipaggiamento simile.

Le soluzioni over IP hanno inoltre permesso l'apparizione di nuove forme di distribuzione, principalmente di tipo subscription-based sempre su set-top boxes o direttamente sugli smart TV.

Un esempio nell'ambito audio è Spotify, mentre in quello video alcuni esempi sono Netflix, Amazon Video e Google Play Movies.

Tipologie di soluzioni di streaming

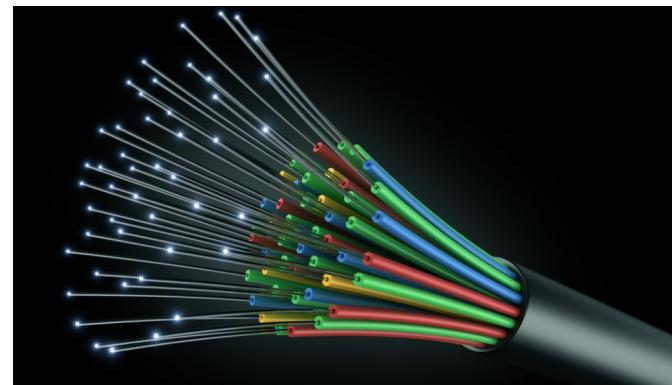
Le soluzioni di streaming audio/video possono essere classificate in tre gruppi principali:

- live media (televisione): permette di accedere alla programmazione dal vivo, con o senza interattività.
- time-shifted media: permettono di accedere alla programmazione passata.
- video on demand (VOD): permette di accedere ad un catalogo di contenuti disponibili per lo streaming.

Fibra

La fibra ottica è la nuova generazione di supporto fisico per la comunicazione. Ogni elemento della fibra ottica (piccoli tubi di vetro) è ricoperto da uno strato di plastica ed introdotto in un tubo protettivo.

La capacità di trasmissione è quella della luce, 26'000 volte maggiore di quella del cavo twisted-pair. È leggera e robusta in termini di trasmissione, è però più complessa da installare rispetto ai cavi twisted pair o coassiali.



Streaming

Il termine "streaming" ha cominciato a venire utilizzato negli anni '90 come migliore descrizione di video on demand su IP. Oggi, "streaming" viene utilizzato per tutte le situazioni in cui contenuti audio/video vengono trasmessi tramite internet a un computer, uno smartphone, un set-top box o una smart TV.

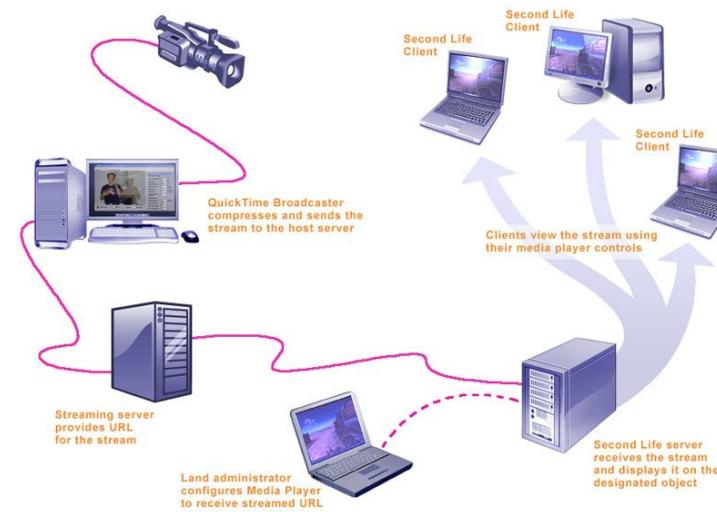
Nello streaming non è necessario scaricare il contenuto interamente prima di poterlo sentire/vedere. La riproduzione comincia "istantaneamente".

Servizi di streaming famosi sono ad esempio YouTube, Netflix, Spotify e Apple Music.

Live streaming

Con live streaming si intende lo streaming in tempo-reale mentre l'evento ha luogo, come nella televisione dal vivo.

Il live streaming è più complesso da realizzare. Necessita di videocamere e microfoni, encoder real-time per la digitalizzazione, uno strumento di publishing dei contenuti e una content delivery network (CDN) per distribuirli.



Challenges del multimedia streaming

Ci sono alcune challenges legate allo streaming su Internet di contenuti multimediali.

I principali problemi che possono insorgere sono quelli di packet-loss, packet-delay e network jitter.

In particolare, se la connessione non presenta sufficiente larghezza di banda, può capitare che i dati non arrivino per tempo, con conseguente degrado della qualità di ricezione.

Unicast vs. multicast

I protocolli unicast inviano una copia separata della stream dal server a ogni utente connesso. L'unicast è la norma per le connessioni Internet, ma la scalabilità non è ideale se molti utenti vogliono ricevere i medesimi contenuti contemporaneamente.

I protocolli multicast sono stati sviluppati per ridurre il carico sul server e sulla rete. Viene spedita un'unica stream dalla sorgente agli utenti. La disponibilità del multicast non è sempre garantita. Dipende dall'infrastruttura di rete disponibile. Inoltre routers e i firewalls possono ostacolarne il passaggio.

Content delivery network

Per ovviare alla moltitudine di ostacoli tecnici, una soluzione spesso adottata è quella di spedire la stream dalla sorgente ad un'infrastruttura cloud detta content delivery network (CDN), responsabile dell'eventuale transcoding e della distribuzione dei contenuti multimediali agli utenti.



LAN vs. WAN

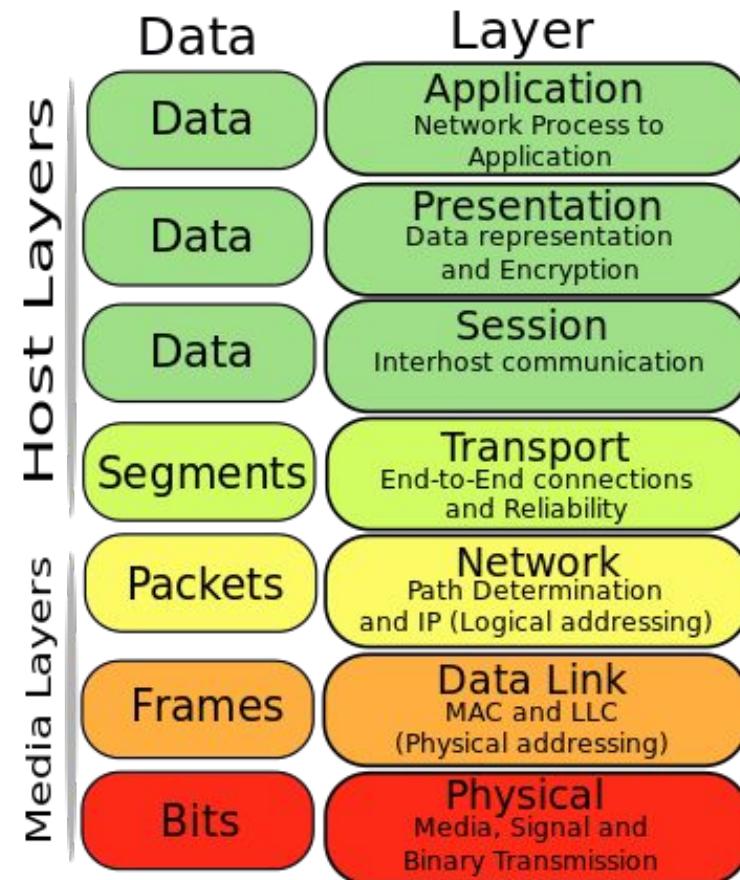
Esistono alcune soluzioni specifiche per l'audio o il video over IP su LAN. Nel caso dell'audio è possibile ad esempio utilizzare le tecnologie come Dante o Ravenna.

Lo streaming su WAN è più complesso a causa della presenza dei routers, dei firewalls e dei differenti Internet service providers (ISP) che riducono le possibilità di connessione.

In generale, il multicast su WAN è molto difficile da eseguire, perché bloccato dagli ISP.

In generale l'HTTP è l'unico canale di comunicazione garantito.

Modello OSI



Protocolli e formati per lo streaming

I protocolli e i formati comunemente utilizzati sono:

- Network Layer: IP
- Transport Layer: TCP, UDP
- Application Layer: HTTP, FTP, RTP (con RTCP), RTSP, RTMP (Adobe), RTMFP (su UDP), RTMPT (HTTP tunnel)
- Dynamic Adaptive Streaming (su HTTP): HLS (Apple), MPEG-DASH, HDS (Adobe), Smooth Streaming (Microsoft)

Ravenna

Ravenna è una tecnologia di streaming audio over IP pensata principalmente per utilizzo in LAN:

- Basata su RTP / RTCP
- Supporta sia unicast che multicast
- Sfrutta SDP o RTSP per la gestione della connessione e della stream
- La sincronizzazione viene mantenuta tramite PTPv2
- QoS basato su DiffServ

RTP

Il real-time transport protocol (RTP) permette la delivery sia di audio che di video over IP. Viene utilizzato nei sistemi di comunicazione e entertainment per la telefonia, teleconferenza, televisione e streaming Internet (incluso il WebRTC).

Di norma l'RTP viene utilizzato con UDP e in combinazione con l'RTP control protocol (RTCP), per mantenere controllo e sincronizzazione delle streams. L'RTP supporta anche il multicast.

RTP

L'RTP è pensato per lo streaming multimediale in real-time di dati da punto a punto. Fornisce funzionalità per la compensazione del jitter e per la detection della packet loss e dell' out-of-order delivery.

RTP è una delle fondamenta tecniche del Voice over IP. In questo contesto viene usato in combinazione con il session initiation protocol (SIP) per stabilire le connessioni all'interno della rete.

MPEG-DASH

Il dynamic adaptive streaming over HTTP (DASH), più comunemente conosciuto come MPEG-DASH, è una tecnica di streaming a bitrate adattivo che permette lo streaming Internet di contenuti multimediali a partire da web servers HTTP convenzionali.

È simile all'HTTP live streaming (HLS) di Apple, ma non è proprietario ed è uno standard internazionale.

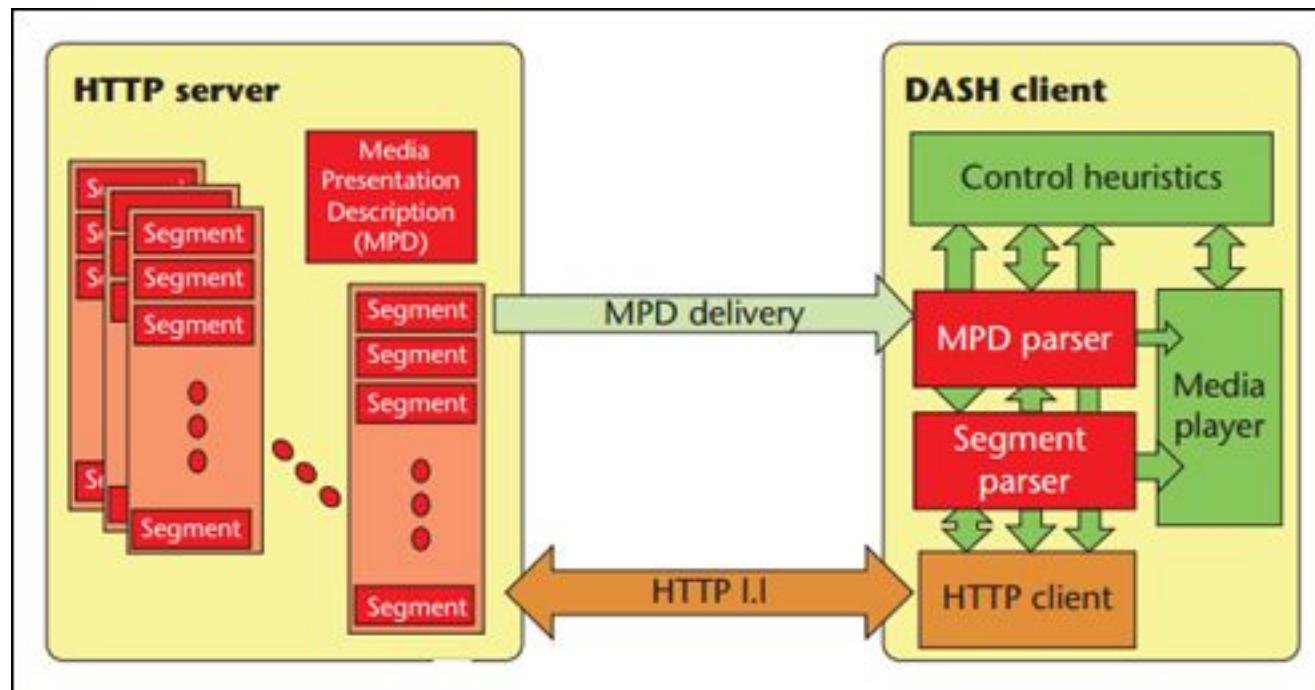
MPEG-DASH

Il funzionamento è il seguente: i contenuti multimediali vengono spezzati in una sequenza di segmenti (ad esempio di 10 sec) messi a disposizione in una varietà di bitrates differenti.

Sulla base delle condizioni di rete, il dispositivo di ricezione è responsabile della scelta del formato adeguato da scaricare e riprodurre. È anche responsabile di gestire eventuali pacchetti mancanti o out of order.

Il download è quindi un normale HTTP download.

MPEG-DASH



MPEG-DASH

MPEG-DASH può essere utilizzato da televisori, set-top boxes, computers, smartphones, tablets, etc.

A differenza di HLS, HDS, o Smooth Streaming, l'MPEG-DASH non ha vincoli di codec. In combinazione si può utilizzare qualsiasi formato audio e video.

SUPSI

GStreamer

Fondamenti di Multimedia Processing

Loris Grossi

11.01.2019

Cos'è?

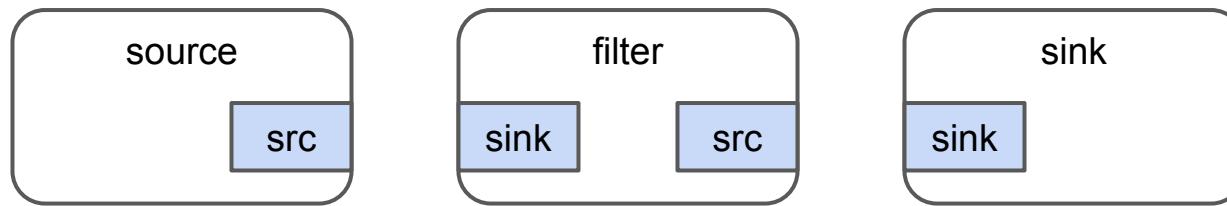
- Framework open source per la creazione di applicazioni multimediali
- Piattaforma altamente modulare sviluppata in C
- Multi-platform: Linux-kernel based, Android, MacOS, iOS, Windows
- Esistono dei bindings per vari linguaggi: ad esempio Python, C++, Perl, Ruby, C#, ...
- Utilizzabile anche da linea di comando

Cos'è?

- Utilizza un'architettura a plug-ins
- Molti elementi quali codecs, container formats, input e output drivers ed effetti sono forniti al momento dell'installazione
- Possibilità di sviluppare ulteriori plug-ins utilizzando il linguaggio C

Elemento

- Componente di base per la costruzione delle pipelines
- 3 elementi di base



- La comunicazione tra i vari elementi avviene attraverso dei pads
- Ogni elemento ha delle proprietà alle quali possono essere assegnati determinati valori

Pipeline

- Insieme di elementi collegati sequenzialmente
- È composta da almeno una source, almeno un sink e da uno o più elementi di tipo filter
- Può essere descritta con una lista di elementi separati dal carattere !



```
audiotestsrc ! audioconvert ! autoaudiosink
```

Elementi utili (audio)

Generazione segnali audio di test

audiotestsrc generatore di suoni (utilizzato per testare delle pipelines audio). Il tipo di suono è scelto utilizzando la proprietà `wave`

Audio adapters

audioconvert converte dei dati audio raw in vari formati

audioresample esegue una conversione di sample rate

Ascolto di audio

autoaudiosink seleziona automaticamente la device di riproduzione dell'audio

Elementi utili (video)

Generazione video di test

videotestsrc generatore di pattern video (utilizzato per testare delle pipelines video). Il tipo di pattern è scelto utilizzando la proprietà `pattern`

Video adapters

videoconvert converte da un color space ad un altro (ad esempio da RGB a YUV)

videorate esegue una conversione di sample rate (eliminando o duplicando dei frames)

videoscale ridimensiona il video

Visualizzazione dei media

autovideosink seleziona automaticamente la device di riproduzione del video

Elementi utili

Bins: elementi in grado di costruire dinamicamente una pipeline in base alle informazioni estratte dal media

playbin gestisce tutti gli aspetti di riproduzione di un media (dalla lettura, alla decodifica fino alla riproduzione)

uridecodebin decodifica i dati ricevuti da un URI fino ad ottenere dei dati raw

decodebin decodifica i dati in entrata fino ad ottenere dei dati raw

File input / output

filesrc legge i dati presenti in un file

filesink scrive i dati in input su di un file

Tools

gst-inspect-1.0 [PLUGIN | ELEMENT]

- Mostra le informazioni disponibili per il PLUGIN o l'ELEMENT selezionato
- Se eseguito senza PLUGIN o ELEMENT, mostra una lista di tutti gli elementi disponibili

The image shows two terminal windows side-by-side. The left window displays a list of available GStreamer elements: voaacenc, osxaudio, osxaudiosrc, osxaudiodeviceprovider, replaygain, rglimiter, rgvolume, soundtouch, bpmdetect, and rawparse. The right window shows detailed information for the 'autoaudiosink' plugin, including its factory details and plugin details.

```
loris:~ $ gst-inspect-1.0
voaacenc: AAC audio encoder
osxaudio: osxaudiosink: Audio Sink (OSX)
osxaudio: osxaudiosrc: Audio Source (OSX)
osxaudio: osxaudiodeviceprovider (GstDeviceProviderFactory)
replaygain: rganalysis: ReplayGain analysis
replaygain: rglimiter: ReplayGain limiter
replaygain: rgvolume: ReplayGain volume
soundtouch: pitch: Pitch controller
soundtouch: bpmdetect: BPM Detector
rawparse: unalignedaudioparse: unalignedaudioparse
```

```
loris:~ $ gst-inspect-1.0 autoaudiosink
Factory Details:
  Rank                  none (0)
  Long-name             Auto audio sink
  Klass                 Sink/Audio
  Description           Wrapper audio sink for automatically detected audio sink
  Author                Jan Schmidt <thaytan@noraizin.net>

Plugin Details:
  Name                  autodetect
  Description           Plugin contains auto-detection plugins for video/audio in
```

Tools

gst-launch-1.0 PIPELINE_DESCRIPTION

- Tool per creare ed eseguire una pipeline
- Nella forma più semplice, la PIPELINE_DESCRIPTION è una lista di elementi separati da !
- È possibile aggiungere delle proprietà ad ogni elemento property=value
- Utilizzato principalmente come tool di debugging
- Utilizzando l'opzione -v si attiva il supporto verbose

```
gst-launch-1.0 audiotestsrc ! audioconvert ! autoaudiosink
```

```
gst-launch-1.0 audiotestsrc wave=white-noise ! audioconvert ! autoaudiosink
```

Debugging

GST_DEBUG

- Variabile d'ambiente utilizzata per specificare il livello di debug desiderato

Level	Name	Description
0	none	No debug information is output.
1	ERROR	Logs all fatal errors.
2	WARNING	Logs all warnings.
4	INFO	Logs all informational messages.
5	DEBUG	Logs all debug messages.
6	LOG	Logs all log messages.

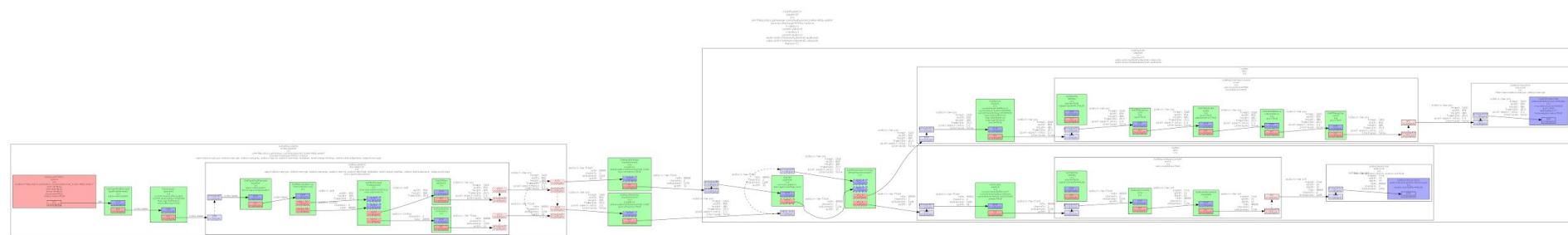
GST_DEBUG=5

GST_DEBUG=2, filesrc:4

Debugging

GST_DEBUG_DUMP_DOT_DIR

- Variabile d'ambiente utilizzata per specificare la path dove salvare il grafico della pipeline



SUPSI

Machine Learning per il Multimedia Processing

Fondamenti di Multimedia Processing

Tiziano Leidi

18.01.2019

Machine learning

Il machine learning è un campo dell'informatica con l'obiettivo di sviluppare funzionalità per rendere i computers capaci di imparare senza che li si programmi in maniera esplicita.

È l'evoluzione degli studi sul riconoscimento di pattern e sulle teorie di computational learning dell'intelligenza artificiale.

Il machine learning esplora lo studio e la realizzazione di algoritmi che imparano dai dati in modo da poter fare successivamente previsioni su di essi.

Machine learning

In varie situazioni gli algoritmi di machine learning si rivelano più efficaci di programmi scritti a mano, perché producono autonomamente un modello a partire da dati campione.

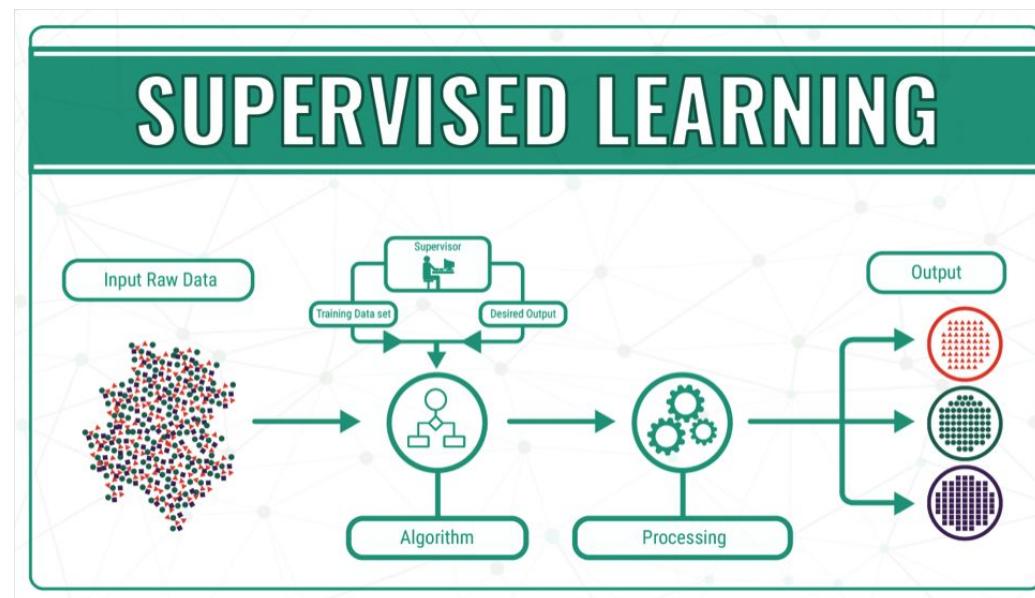
Viene impiegato soprattutto in domini applicativi in cui il design e la programmazione esplicita di algoritmi con buone performances è difficile o addirittura impossibile.

Machine learning

La realizzazione di buoni algoritmi di machine learning è però un compito complesso perché riconoscere i patterns può essere difficile e spesso i dati disponibili non sono sufficienti. Capita che molti programmi di machine learning non siano efficaci.

Supervised learning

Nel supervised learning all'algoritmo vengono forniti sia esempi di input, che i rispettivi dati di output, da un “educatore” dell'algoritmo. L'algoritmo cerca di scoprire la regola generale che associa gli inputs agli outputs.

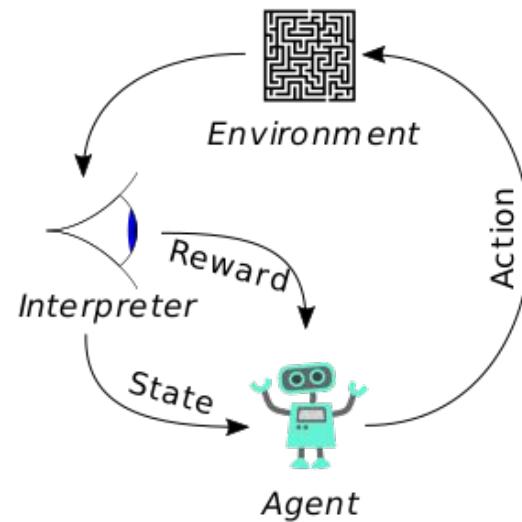


Supervised learning

L'apprendimento può anche continuare in fase di esecuzione dell'algoritmo, sulla base di feedback fornito dall'utente, nel qual caso si parla di active learning.

Reinforcement learning

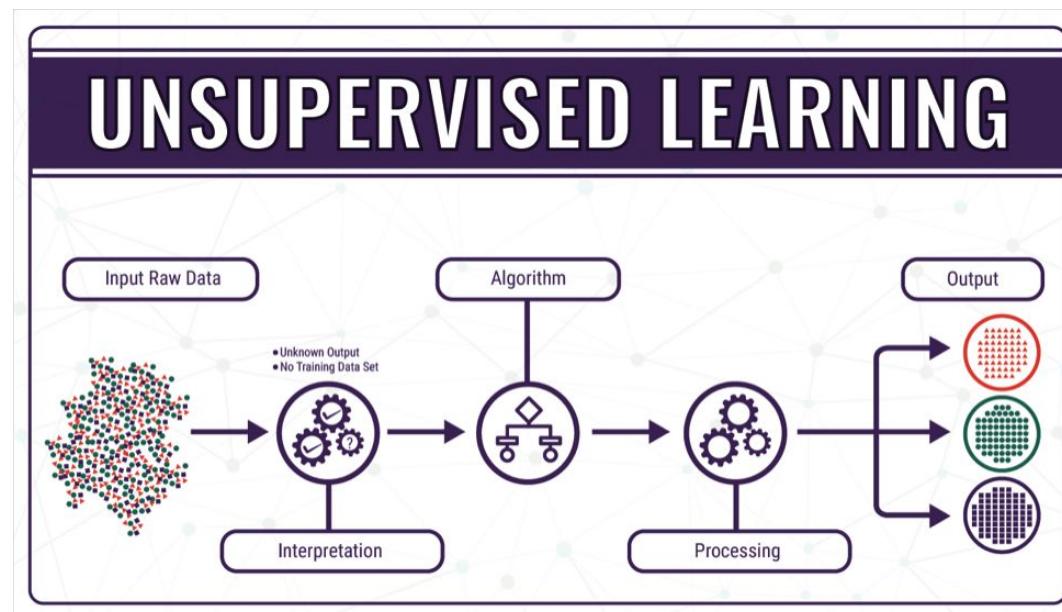
Nelle forme più estreme, come ad esempio nel reinforcement learning, come unica forma di feedback l'output viene approvato o respinto. Il reinforcement learning viene ad esempio sfruttato nei sistemi di guida automatica.



Unsupervised learning

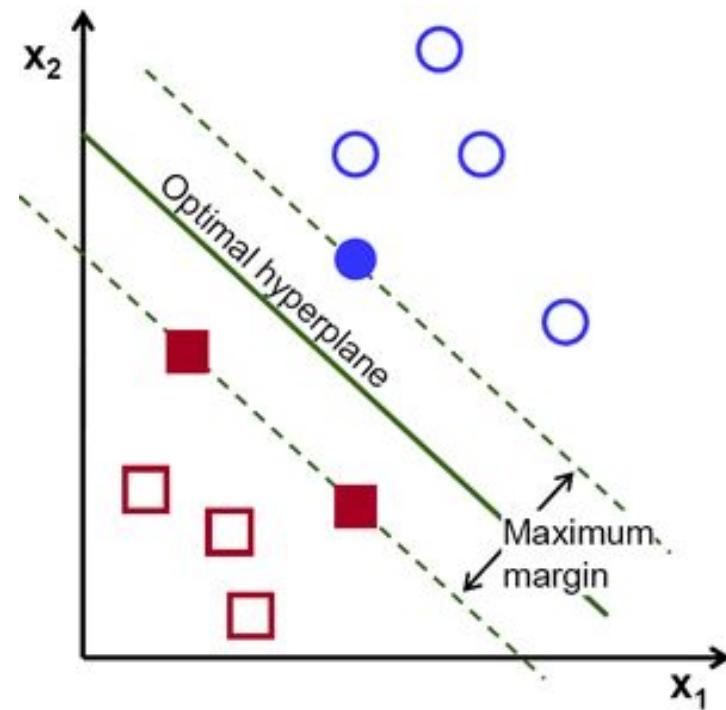
Nel unsupervised learning, nessuna informazione viene fornita all'algoritmo, che è costretto a scoprire in maniera autonoma la struttura dei dati in input.

È la forma più complessa di machine learning.



Esempio: support vector machine

Una support vector machine è un classificatore che divide lo spazio di input in due regioni separate da un margine lineare.



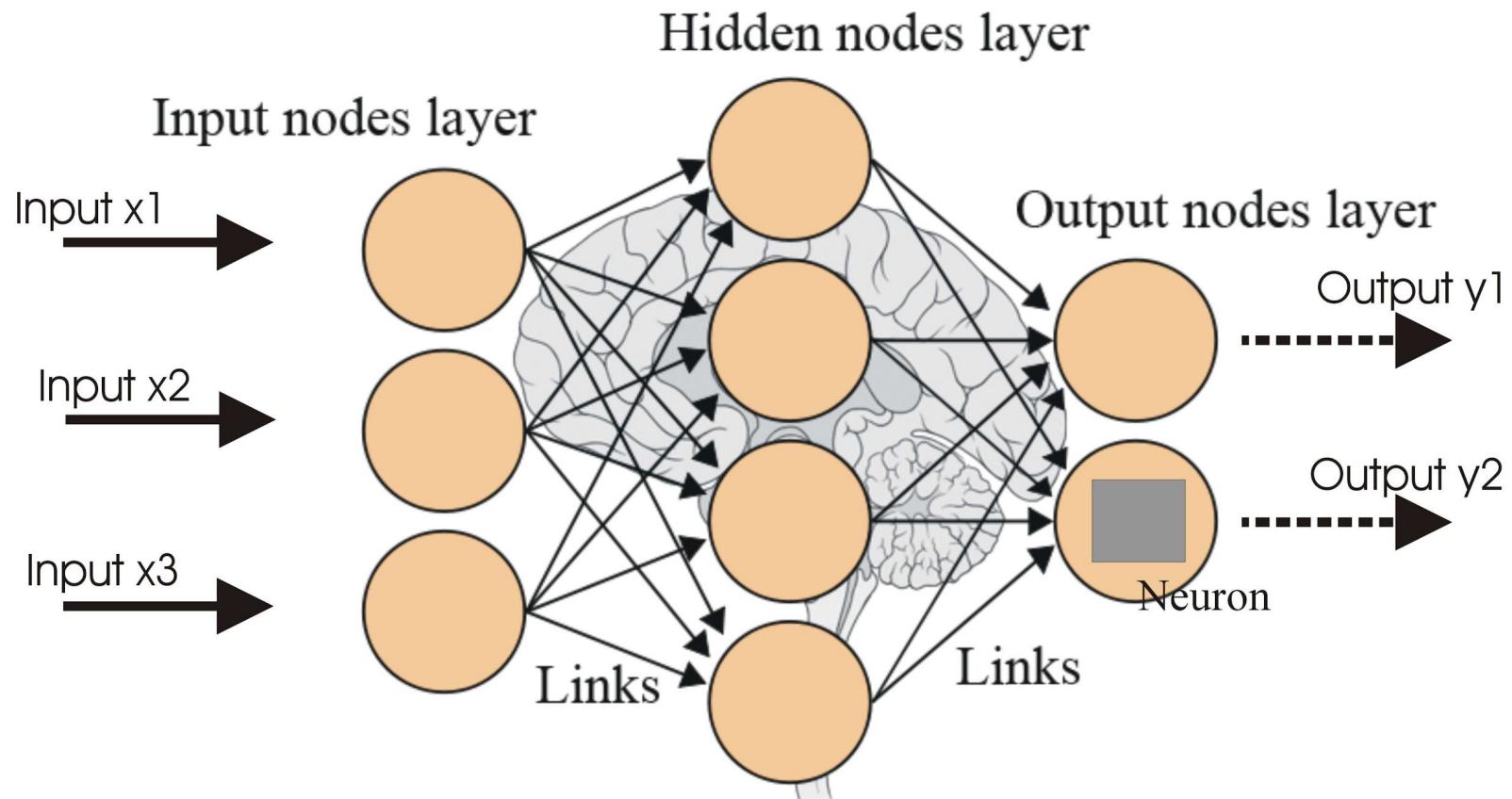
Reti neurali artificiali

Una delle famiglie più utilizzate di algoritmi di machine learning è quella delle reti neurali.

Una rete neurale è un algoritmo che prende ispirazione dalla struttura e dall'aspetto funzionale di reti neurali biologiche. I calcoli vengono strutturati in termini di gruppi di neuroni artificiali interconnessi, che processano collettivamente l'informazione.

Le neural networks utilizzate oggi sono strumenti non lineari per la modellazione di tipo statistico di dati con una certa complessità.

Reti neurali artificiali



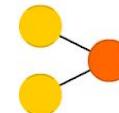
A mostly complete chart of

Neural Networks

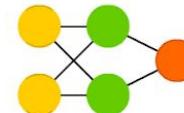
©2016 Fjodor van Veen - asimovinstitute.org

- (○) Backfed Input Cell
- (○) Input Cell
- (△) Noisy Input Cell
- (●) Hidden Cell
- (○) Probabilistic Hidden Cell
- (△) Spiking Hidden Cell
- (●) Output Cell
- (○) Match Input Output Cell
- (●) Recurrent Cell
- (○) Memory Cell
- (△) Different Memory Cell
- (●) Kernel
- (○) Convolution or Pool

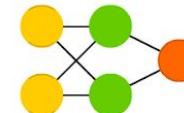
Perceptron (P)



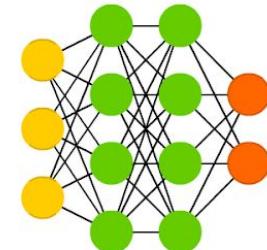
Feed Forward (FF)



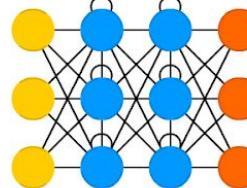
Radial Basis Network (RBF)



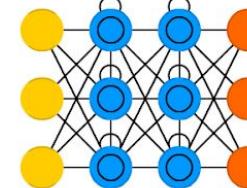
Deep Feed Forward (DFF)



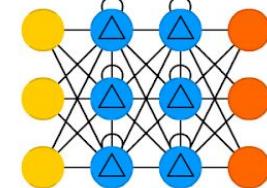
Recurrent Neural Network (RNN)



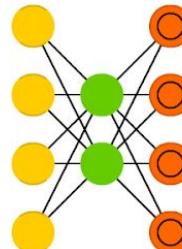
Long / Short Term Memory (LSTM)



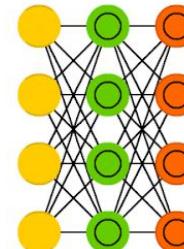
Gated Recurrent Unit (GRU)



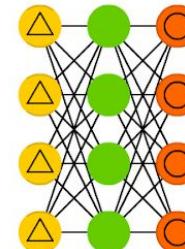
Auto Encoder (AE)



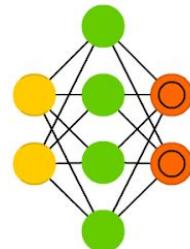
Variational AE (VAE)



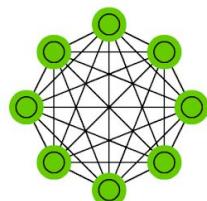
Denoising AE (DAE)



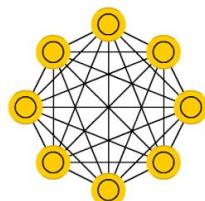
Sparse AE (SAE)



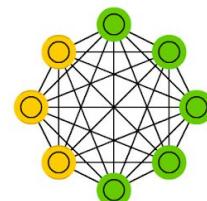
Markov Chain (MC)



Hopfield Network (HN)



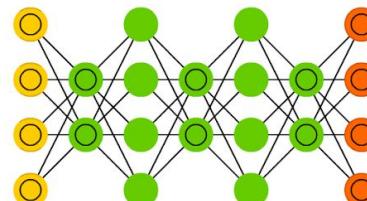
Boltzmann Machine (BM)



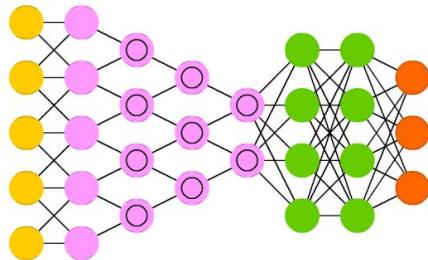
Restricted BM (RBM)



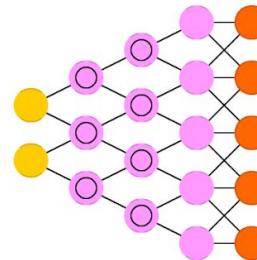
Deep Belief Network (DBN)



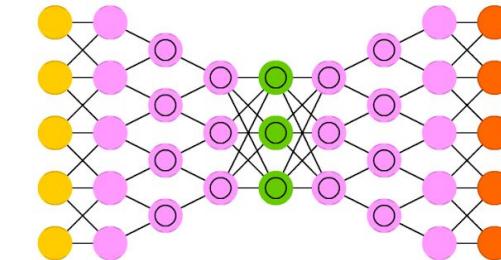
Deep Convolutional Network (DCN)



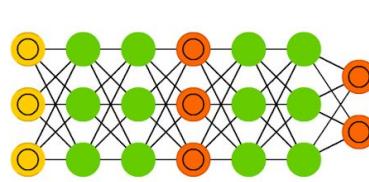
Deconvolutional Network (DN)



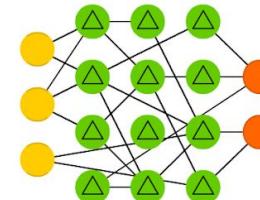
Deep Convolutional Inverse Graphics Network (DCIGN)



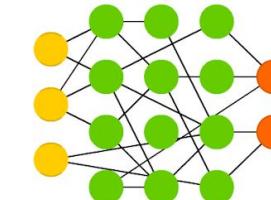
Generative Adversarial Network (GAN)



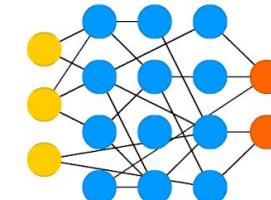
Liquid State Machine (LSM)



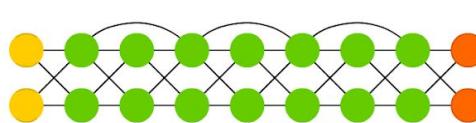
Extreme Learning Machine (ELM)



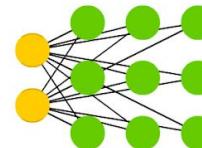
Echo State Network (ESN)



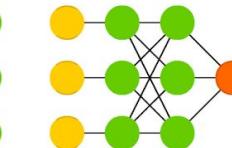
Deep Residual Network (DRN)



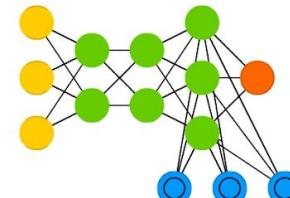
Kohonen Network (KN)



Support Vector Machine (SVM)



Neural Turing Machine (NTM)



Reti neurali artificiali - vantaggi

Il vantaggio principale delle reti neurali artificiali è la loro capacità di essere generalmente più efficaci degli altri algoritmi di machine learning.

Sono anche generalmente più semplici da allenare rispetto ad altri algoritmi di machine learning ed offrono più flessibilità di parametrizzazione.

Reti neurali artificiali - svantaggi

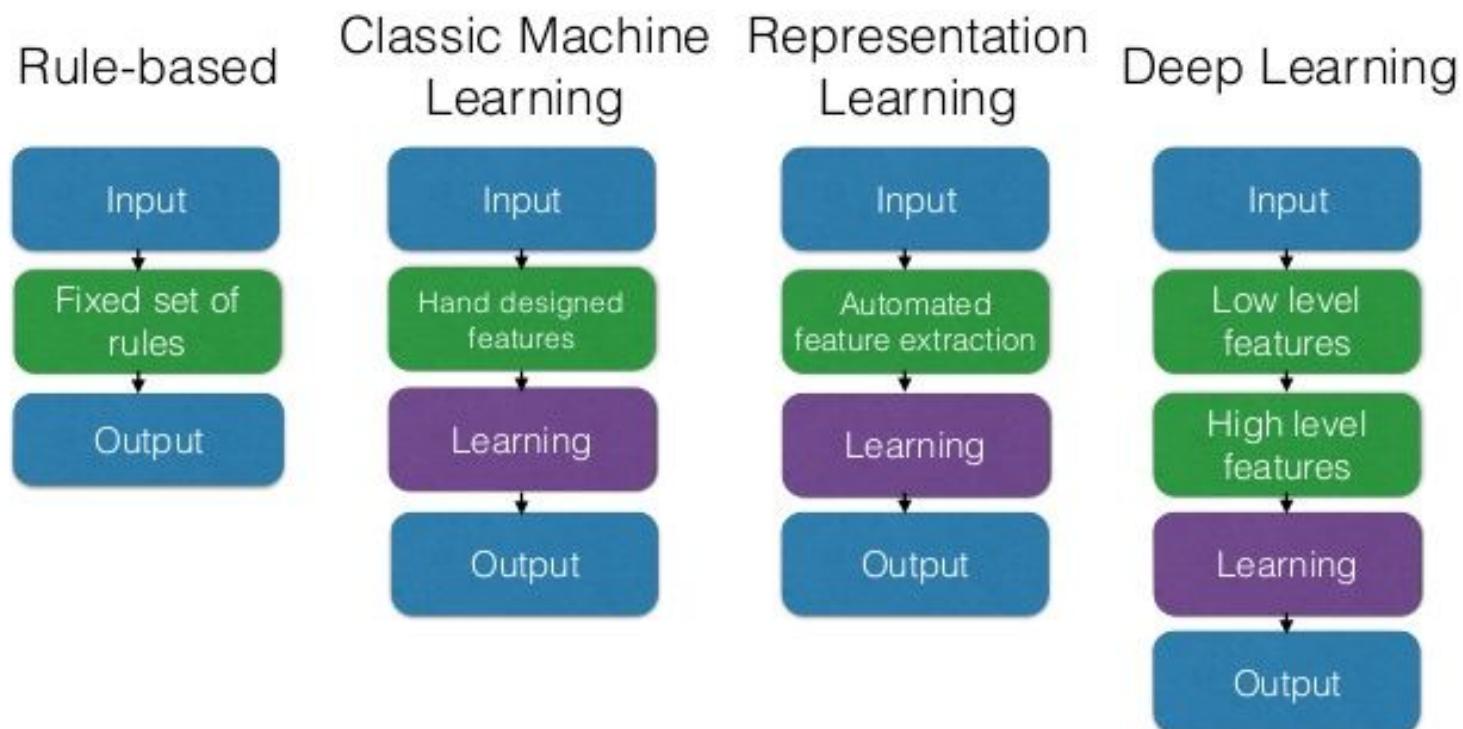
D'altro canto, la loro natura di tipo black-box può risultare svantaggiosa in casi in cui è necessario il controllo del comportamento dell'algoritmo (predictability).

Allo stesso tempo le reti neurali artificiali sono facilmente soggette ad overfitting.

Con pochi dati non identificano la caratteristica generica desiderata ma il caso specifico.

Per renderle capaci di generalizzare è necessario allenarle con data sets molto ampi e ben costruiti.

Evoluzione delle tecniche di ML



Deep learning

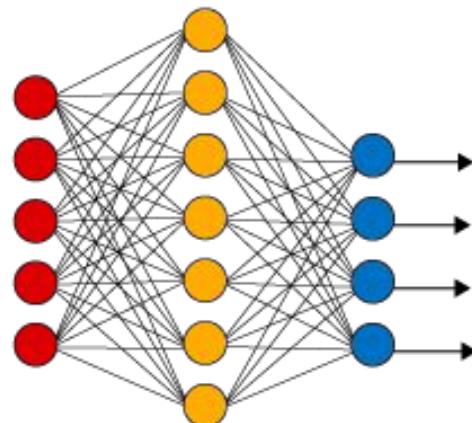
La crescita delle capacità di calcolo, la progressiva riduzione del costo dell'hardware e l'evoluzione delle GPUs degli ultimi anni, hanno contribuito allo sviluppo del concetto di deep learning.

Nel deep learning si cerca di emulare il processing che il cervello esegue ad esempio sulla luce e sul suono nelle capacità di visione e di udito.

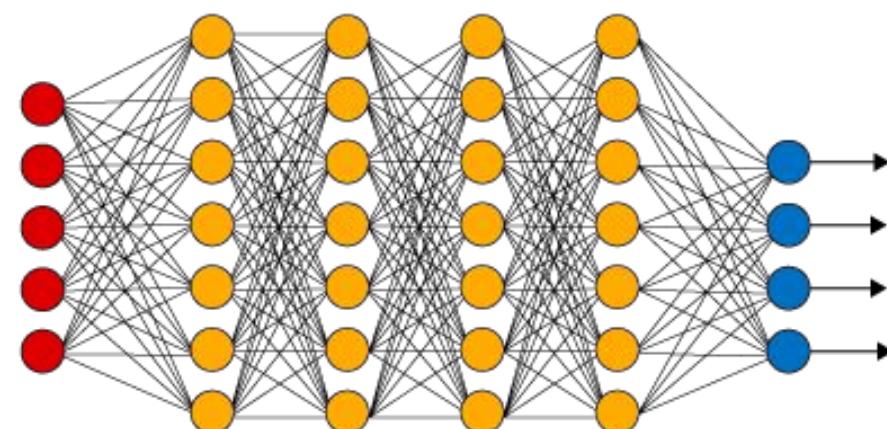
Deep learning

Nel deep learning vengono utilizzate reti neurali composte da layers multipli.

Simple Neural Network



Deep Learning Neural Network



● Input Layer

● Hidden Layer

● Output Layer

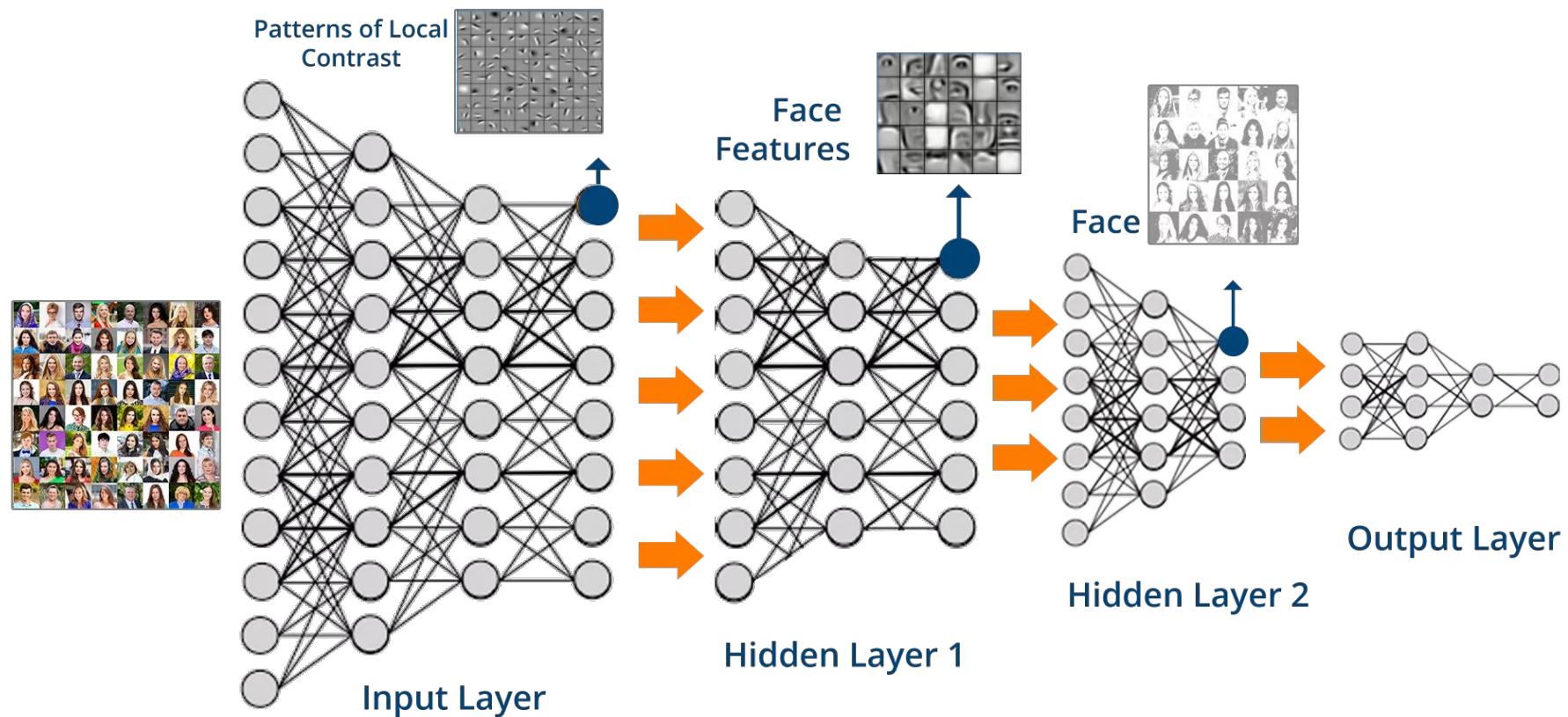
Deep learning

Nel deep learning, ogni layer impara a trasformare i dati di input in una rappresentazione ogni volta leggermente più astratta e composta.

Il termine deep si riferisce al numero di layers attraverso i quali i dati vengono trasformati.

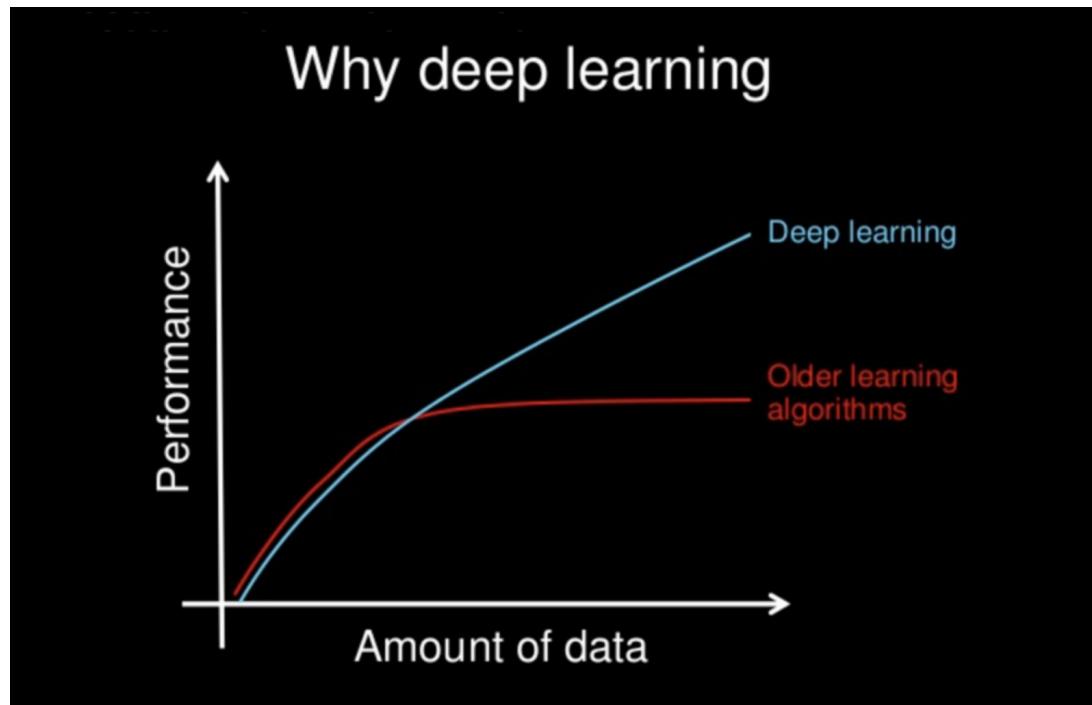
Nelle soluzioni end-to-end l'intero processo , dai sensori agli attuatori, viene eseguito da un'unica layered o recurrent neural network senza modularizzazione.

Deep learning



Deep learning

Il vantaggio degli algoritmi di deep learning è la capacità di continuare a migliorare le performances all'aumentare dei dati di apprendimento disponibili.

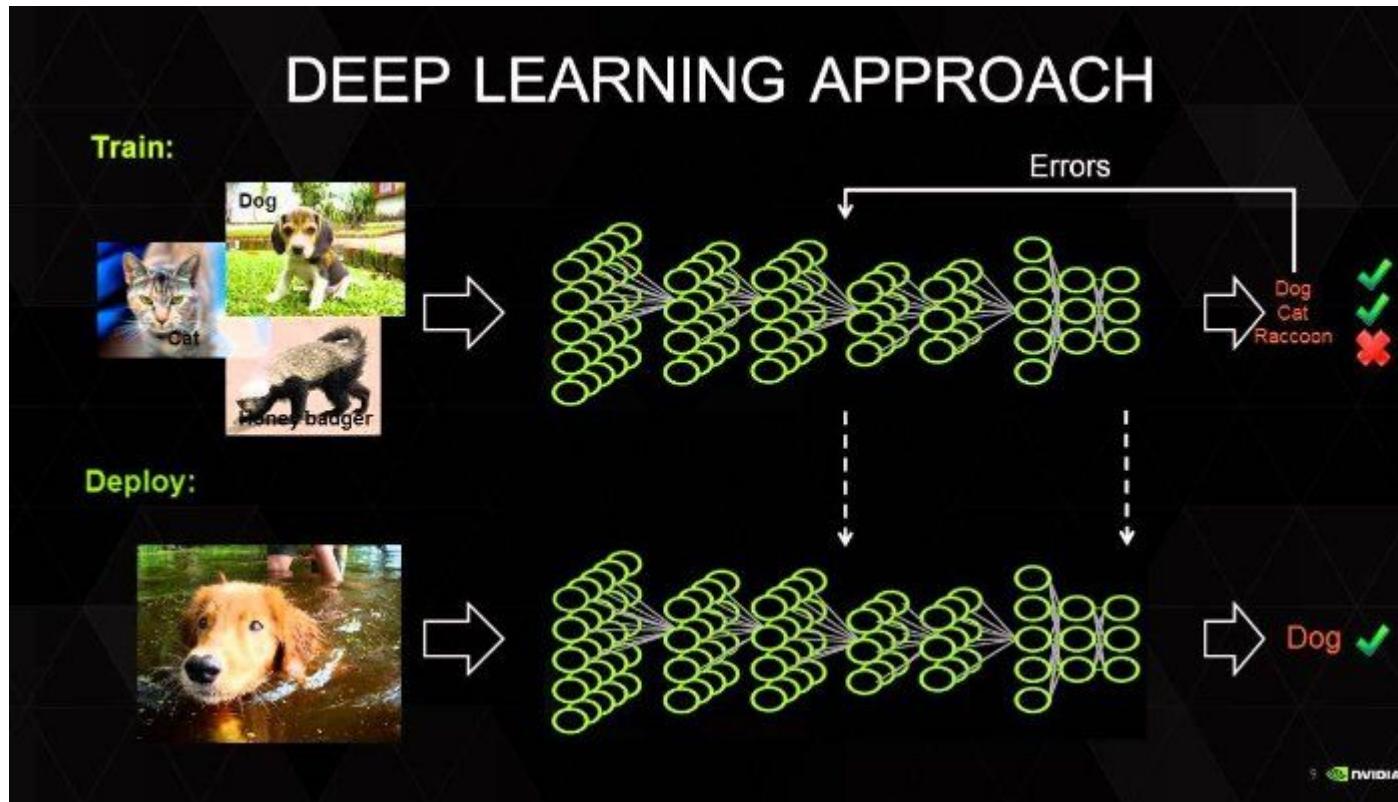


Deep learning

Per quanto concerne l'ambito del multimedia, oggi esistono applicazioni di successo soprattutto nella computer vision e nel riconoscimento vocale.

Deep learning

Ad esempio, il deep learning viene utilizzato con successo nella classificazione di immagini.

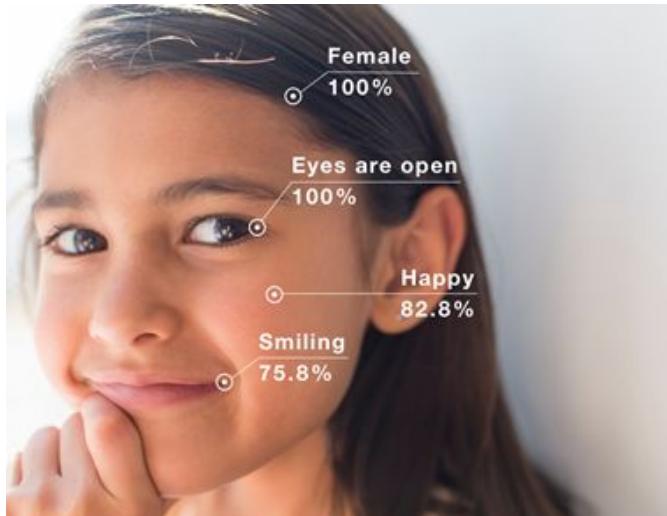


Esempio: Amazon Rekognition

Amazon Rekognition è un servizio della piattaforma AWS che permette di aggiungere funzionalità di analisi d'immagine alle applicazioni. Con Rekognition è possibile identificare oggetti, scene e facce. È anche possibile confrontare le facce.

L'API di Rekognition's permette l'accesso ad algoritmi di deep learning per la classificazione d'immagine. Utilizza modelli di tipo deep neural network.

Esempio: Amazon Rekognition



Esempio: Amazon Transcribe

Amazon Transcribe è un servizio della piattaforma AWS per il riconoscimento vocale automatico.

Facilita l'aggiunta di funzionalità di speech-to-text alle applicazioni.

Con Amazon Transcribe è possibile analizzare files audio, ottenendo come risultato files di testo contenenti la trascrizione.

È possibile utilizzare Transcribe anche in modalità streaming.

Esempio: Caffe

Caffe è un framework di deep learning sviluppato con l'obiettivo di essere veloce, modulare e facile da utilizzare. È realizzato dal Berkeley AI Research (BAIR) e da una comunità di contributors. Viene rilasciato sotto licenza di tipo BSD 2-Clause.