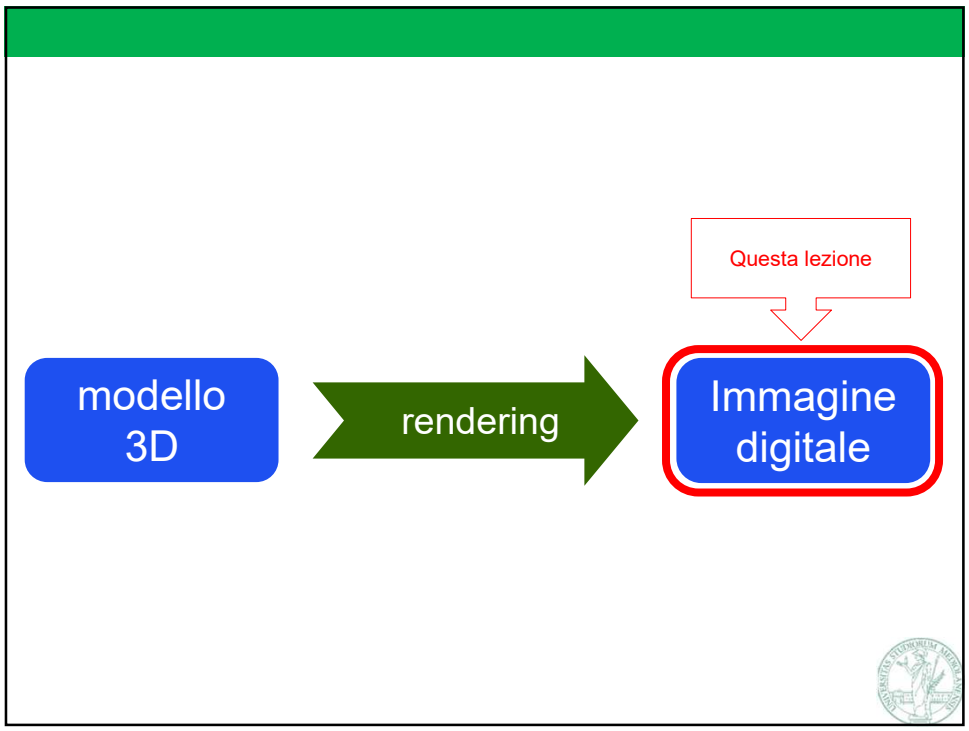
Marco Tarini - Computer Graphics 2019/2020
Università degli Studi di Milano

Immagini digitali

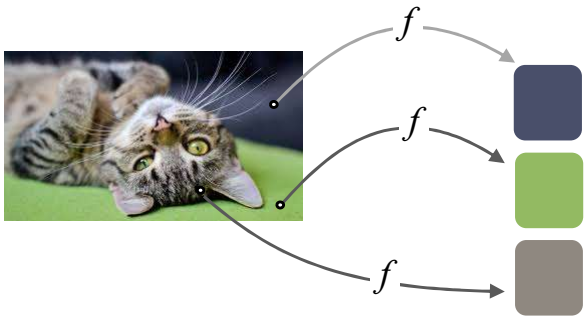
5



6

Immagine

- ✓ Un assegnamento di un colore ad ogni punto di una regione piana rettangolare
- ✓ Quindi una funzione f da (x,y) a (colore)



- ✓ Come rappresento digitalmente un'immagine?

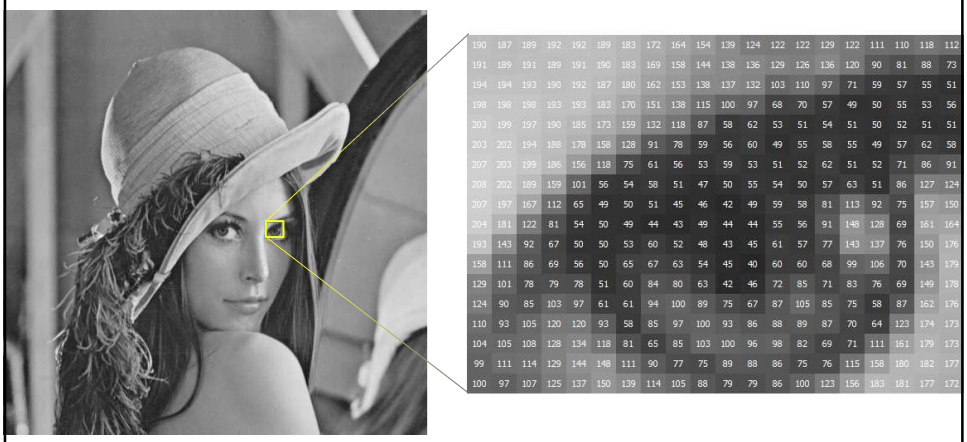
7

Immagini digitali

- ✓ **Rappresentazioni vettoriali**: un set di primitive 2D sovrapposte, quali:
  - ⇒curve parametriche
  - ⇒triangoli , poligoni (2D), cerchi,
  - ⇒zone contornate da curve di Bèzier (2D)
  - ⇒testo (in ascii, ad una pos, associato ad un font)
  - ⇒*etc.*
  - ⇒ciascuna primitiva associata ad un colore
- ✓ **Rappresentazioni rasterizzate**: una griglia regolare 2D di campioni di **colore**, detti "pixel" (da "picture element")

11

Raster Image: gray scale image

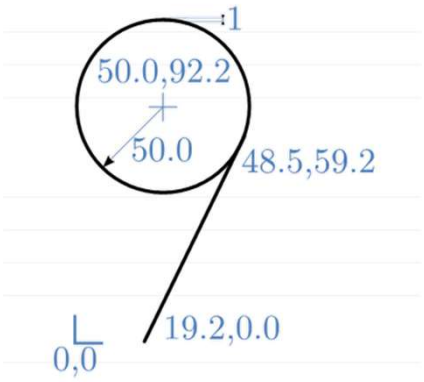


Ogni pixel un livello di grigio
(memorizzato con un numero da 0: nero a 255: bianco)

12

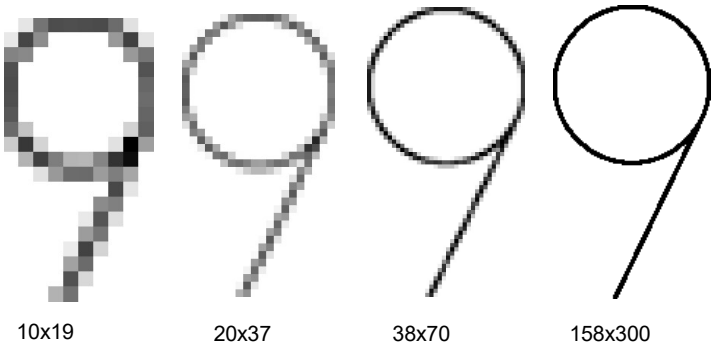
Immagine vettoriale: esempio (inventato)

```
NUMBER_OF_PRIMITIVES 2

CIRCLE
center 50.0, 92.2
radius 50.0
fill_color 1.0, 1.0, 1.0
line_color 0.0, 0.0, 0.0
line_thickness 1pt

SEGMENT
endpoint_one 19.2, 0.0
endpoint_two 48.5, 59.2
line_color 0.0, 0.0, 0.0
line_thickness 1pt
```



13

Raster Images: resolution



14

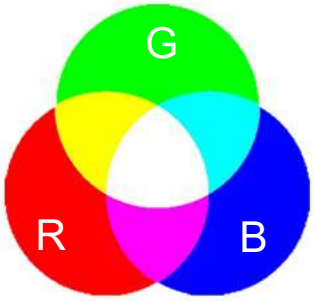
Come rappresento digitalmente un colore?

- ✓Lo studio dei colori come fenomeno fisico, della sua misurazione e rappresentazione è il soggetto di alcune discipline
  - ⇒Colorimetria:
    studia la misurazione e rappresentazione del colore
  - ⇒Radiometria / Fotometria :
    ramo della fisica che studia della radiazione elettormagnetica (quale la luce è)
- ✓Ci accontentiamo di rispondere ad una domanda più semplice:
  - ⇒Cosa devo mandare ad un monitor per per riprodurre un dato colore in uno dei suoi pixel?

15

Sintesi additiva del colore

- ✓ Il pixel di un monitor riproduce un dato colore sovrapponendo 3 luci ad 3 date intensità: una rossa (R), una verde (G), una blu (B)
  - ⇒ Questa soluzione è giustificata da considerazioni sulla percezione umana (che ricadono nel dominio della colorimetria) la **teoria del tristimolo**
- ✓ Per determinare quale colore riprodurre dobbiamo scegliere tre valori di intensità (per R, G ,B)
  - ⇒ ciascuno scelto fra un massimo e un minimo (es fra 0.0 e 1.0, o fra 0 e 255)
  - ⇒ i valori min e max corrispondono rispettivamente alla massima e minima intensità della luce prodotta da tre emettitori (che insieme costituiscono un pixel sul monitor)



16

Come rappresento digitalmente un colore?

f → RGB = (0.28, 0.11 , 0.30)

f → RGB = (0.58, 0.73 , 0.38)

f → RGB = (0.56, 0.53 , 0.50)

RGB = (0, 0, 0) [nero]

RGB = (1, 1, 1) [bianco]

RGB = (1, 0, 0)

RGB = (0.00 , 0.69 , 0.98)

RGB = (1.00, 0.75 , 0.00)

RGB = (0.50, 0.50 , 0.50)

17

Caratteristiche delle immagini Raster

- ✓ **Resolution** = numero di linee e colonne
  - ⇒ Per esempio: 640 × 480 pixel
  - ⇒ Esprimibile anche in MegaPixel (es. 1000 × 1000 = 1MPixel)
- ✓ Ogni pixel ha *k* **canali**
  - ⇒ 1 canale = 1 valore scalare
  - ⇒ Immagine RGB = 3 canali: Red,Green,Blue
  - ⇒ Immagine toni di grigi (gray-scale) = 1 channel (gray level)
- ✓ Ogni canale è memorizzato con un dato numero di bit
  - ⇒ se 8: ho 2^8 = 256 valori possibili, min = 0 and max = 255
  - ⇒ Immagine B & W = il sono canale ha un 1 bit (0 = B e 1 = W)
- ✓ **image depth** (profondità): bit totali per un pixel
  - ⇒ per es: RGB con 8 bits per channel («true color») = 24 bits
  - ⇒ per es: B & W image (una «bitmap»): = 1 bit

18

Immagini: raggio dinamico

- ✓ Rapporto fra luminosità del punto più luminoso e quello più buio
- ✓ HDRI – High Dynamic Range Images:
  immagini con raggio dinamico elevato
  - ⇒ Richiedono più bit per canale

19

Immagini Raster e RAM

- ✓ Totale memoria: resX · resY · imageDepth
  - ⇒ pes es: una immagine 240×480 “true color”
    memoria = 240 · 480 · 3 bytes = 345.600 bytes
    cioè 240 · 480 · 24 bits = 2.764.800 bits
  - ⇒ per es: immagine 4000×2000 (8 MegaPixel) “true color”:
    4000 · 2000 · 3 = 24 MegaByte
- ✓ Compressione spesso necessaria
  - ⇒ Molti schemi di compressione sono usati
- ✓ Tipi di schemi di compressione:
  - ⇒ Lossless : non deteriorano il dato
  - ⇒ Lossy : più aggressivi (es 1:10) deteriorano il dato
  - ⇒ Texture compression: una compressione lossy di tipo speciale utilizzabile dalle immagini di texture

20

Pros and Cons

- ✓ Vector images:
  - ⇒ independent from the device resolution!
  - ⇒ well suited for: logos, diagrams, stylized drawings …
  - ⇒ can be very compact in space
  - ⇒ interpretation: difficult (a rendering required)
  - ⇒ resolution (number of primitive, of control points…) is adaptive
  - ⇒ when zoomed in: there may be little detail to look at, but it looks still good
- ✓ Raster images
  - ⇒ well suited for: natural images (e.g. photographs)
  - ⇒ quality depends on image resolution
  - ⇒ interpretation: direct (monitors can display them directly)
  - ⇒ when zoomed in: artifacts

22

Rarter Images: alcuni formati comuni (note)

- ✓ PNG (Portable Network Graphics):
  - ⇒ lossless compression (based on same algorithm as Zipped files)
  - ⇒ many formats, including with 3 or 4 channels
  - ⇒ good for synthetic images
- ✓ JPEG (Joint Photographic Experts Group):
  - ⇒ (typically) lossy compression (based on "DCT": discrete cosine transform)
  - ⇒ 3 channels of 8 bits each
  - ⇒ good for natural images (digital photography)
- ✓ JPEG 2000
  - ⇒ an advancement over Jpeg (didn't catch on as much)
- ✓ GIF (compuserve)
  - ⇒ strange *quirk* of the used image format
  - ⇒ used for tiny animations over two decades ('90s, 2000's)
  - ⇒ use will decline

23

Rarter Images: not so common formats

- ✓ TIFF
  - ⇒ can be lossless or lossy
  - ⇒ hi-dynamic range data (more than 8 bits per channel)
  - ⇒ used for hi-quality digital photography
- ✓ RAW
  - ⇒ a direct capture of a sensor (es camera CCD)
  - ⇒ non compressed
  - ⇒ non processed
- ✓ PNM (portable any map)
  - ⇒ no compression
  - ⇒ pixel values stored as ASCII (or binary)
  - ⇒ not very used
  - ⇒ but, useful: simple format, human readable values (ASCII numbers)
  - ⇒ and, trivial to parse (an importer / exporter can be written in any language in a few lines of code)

24

Vector Images: common formats

- ✓ SVG:
  - ⇒ developed by W3C
  - ⇒ a XML file describing the primitives
  - ⇒ based on Bèzier curves (including, filled ones)
  - ⇒ plus: basic shapes, text, colors, patterns …
  - ⇒ directly embeddable in Web pages (understood by all browser)
  - ⇒ example of Opensource editor / authoring tool: inkscape
- ✓ PostScript (PS):
  - ⇒ designed for printers
  - ⇒ set of instructions to send to a printer to print the image
  - ⇒ includes ink control
- ✓ Portable Document Format (PDF)
  - ⇒ by Adobe
  - ⇒ includes subsets of PS

25

Immagini digitali: metadata

- ✓ Molti formati di file per immagini digitali includono utilissimi metadati, per es
- ✓ Exif -- Exchangeable image file format
  - ⇒ date
  - ⇒ used camera settings
  - ⇒ thumbnail
  - ⇒ description
  - ⇒ copyright
  - ⇒ geolocation (GPS coordinates of capture)
  - ⇒ …

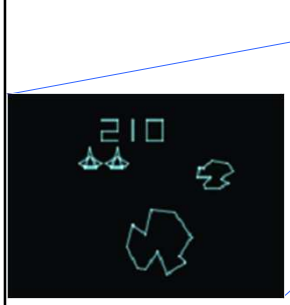
26

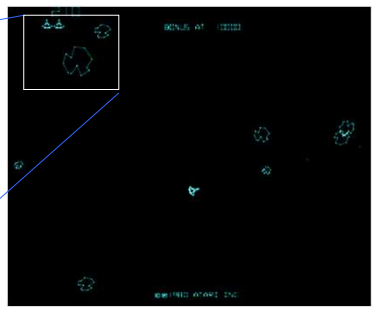
Rendering: su che *display hardware*?

- ✓ Su un monitor, naturalmente!
- ✓ La tecnologia del monitor determina quale output sia richiesto dal rendering Real Time
- ✓ Es: su un CRT monitor (un tubo catodico) un fascio di elettroni «pennello» viene sparato contro una superficie coperta di materiale fosforescente
  - ⇒ **Display vettoriali**:
    il pennello viene pilotato liberamente, tracciando linee.
    Rendering = produrre il percorso del pennello
  - ⇒ **Display raster**:
    pennello spazza tutta la superficie,
    linea per linea («raster» per «raster»), un certo numero di volte al secondo (il «refresh rate»). Il percorso è sempre lo stesso, ma varia l'intensità del pennello
    Rendering = produrre l'intensità variabile del pennello
    Cioè il valore dei pixel. Cioè una immagine rasterizzata

28

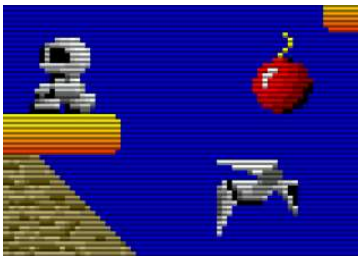
Rendering: su che *display hardware*?





Display vettoriale

1980 (Asteroids – Atari)



Display raster

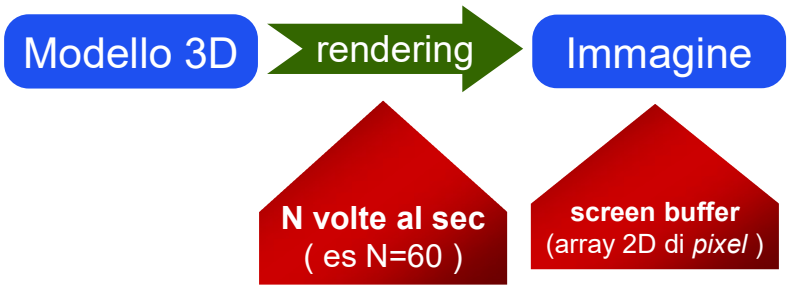
29

Screen Buffer (o frame buffer)

- ✓ I monitor moderni sono di tipo raster
  - ⇒ anche se non si basano più su tecnologia CRT ma LED o OLED (o altro)
  - ⇒ quindi: rendering per un monitor = produrre un'immagine rasterizzata
- ✓ **Screen buffer** = un buffer di memoria che memorizza l'immagine (rasterizzata) da riprodurre fisicamente dal monitor
  - ⇒ ad una locazione prefissata
  - ⇒ [fps] volte al secondo, il pixel fisici sul monitor vengono accesi / spenti secondo il valore attuale dello screen buffer ("refresh dello schermo")
  - ⇒ scrivere pixel sul monitor = cambiare il contenuto dello screen buffer (e aspettare un refresh dello schermo)
  - ⇒ effettuare un rendering = riempire lo screen buffer
- ✓ Le caratteristiche dello screen buffer sono determinate da quelle del monitor fisico:
  - ⇒ Risoluzione: tipicamente da ~800 a ~4000 pixel per lato
  - ⇒ Profondità: tipicamente 3 o 4 canali (R, G, B) 8 bit per canale
- ✓ Il rendering (se non è a schermo intero) può riempire solo una sottoparte dello screen buffer, detta il «**viewport**»
  - ⇒ se non è un rendering a schermo intero (full screen)

30

Real Time 3D Rendering



31

Real Time 3D Rendering: fill rate (esempio)

- ✓ 1 pixel = 32 bit = 4 bytes "pixel depth"
- ✓ screen buffer res = 1024 x 768 pixels "screen resolution"
- ✓ frame al secondo (fps) = 60 Hrz "frame rate"
- ✓ total = 4 x 1024 x 768 x 60 byte al sec "fill rate"

Fill Rate:

188 MegaBytes / sec

anche ignorando molti fattori, come l'overdraw
(necessità di sovrascrivere lo stesso pixel del buffer più volte durante il rendering)

32

GPU per Real-Time 3D Rendering

- ✓ Problema difficile, come si evince dai fill rate
  - ⇒fortunatamente, è anche massicciamente parallelizzabile
  - ⇒"embarrassingly parallel"
- ✓ Ingrediente *base* della soluzione:
  hardware specializzato: la GPU

33