# LABORATORIO DI REALTÀ AUMENTATA

Claudio Piciarelli

Università degli Studi di Udine
Corso di Laurea in Scienze e Tecnologie Multimediali

# Introduction to 3D graphics

- Raster images
- Vector images
- From 2D to 3D
- Transformations
- Projections

# Raster images

La maggior parte delle fotografie con cui abbiamo a che fare

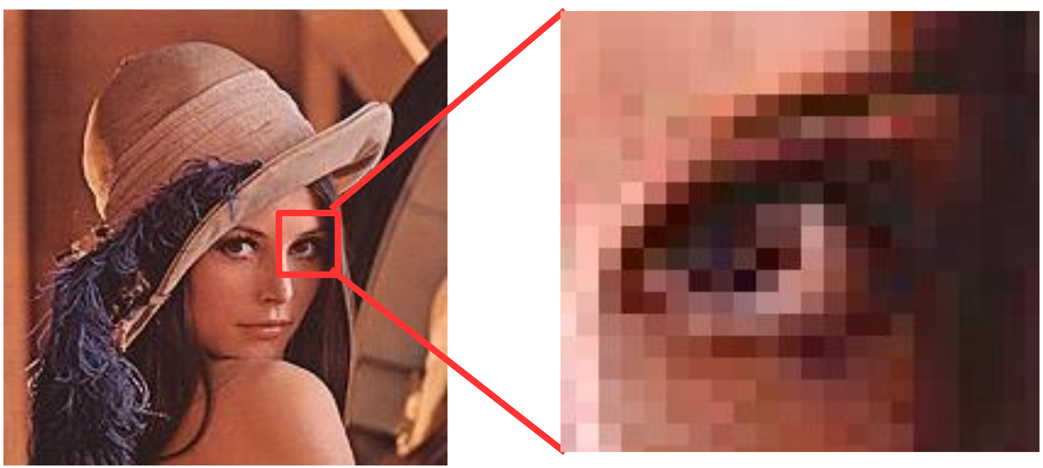
=> composte da una griglia di PIXEL colorati

# Raster images

→ lo svantaggio è che non contengono INFORMAZ. SEMANTICHE sull'imm. stessa

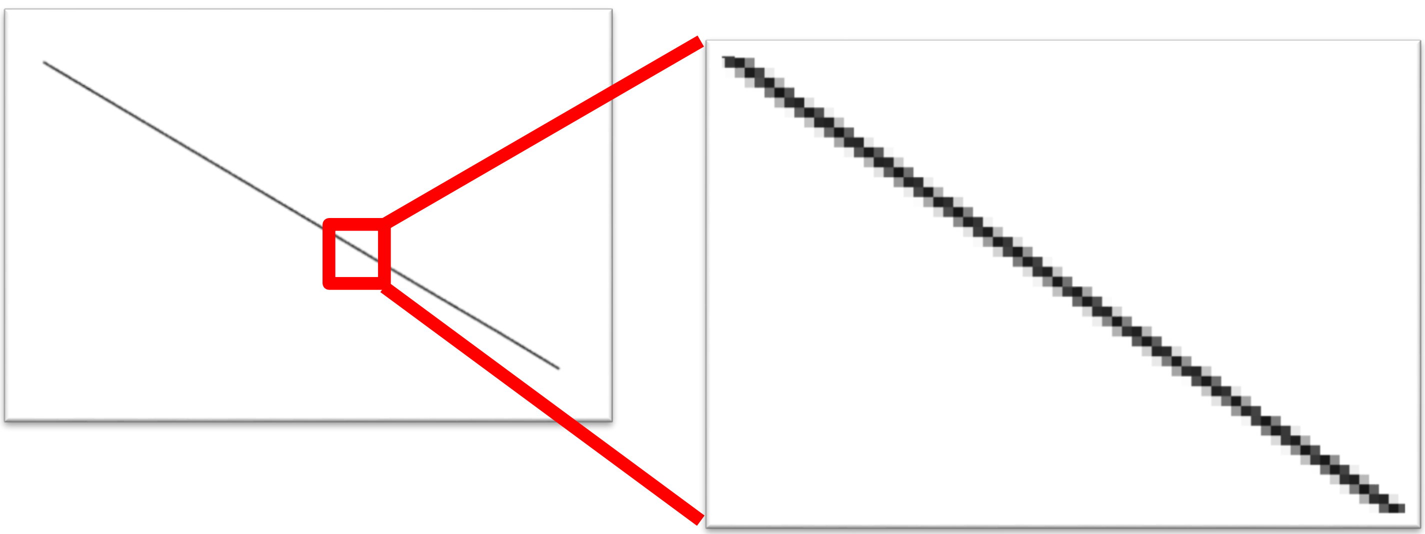
significato dell'immagine (richiede interpretaz.)

- Raster images are composed of a grid of colored squares (the pixels)
- Pictures acquired with a digital cameras, each single frame of a digital video, the vast majority of web images… all of them are raster images



# Semantic expressiveness

- The problem with raster images is their lack of semantic expressiveness. => il computer non conosce la semantica di un'immagine ma la vede solo come un'insieme di pixel
- This line is actually just a bunch of gray pixels

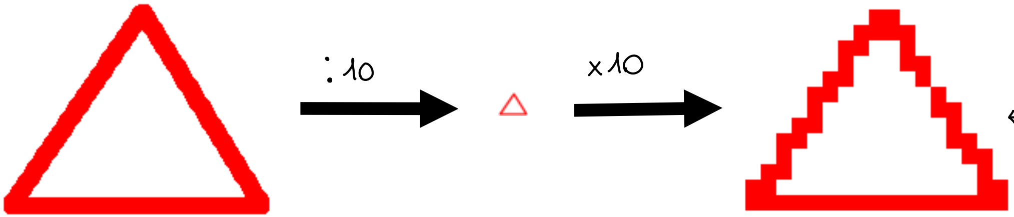


# Information loss

→ quando manipolo l'immagine perdo informazione

es. se prendo un'immagine, la rimpicciolisco di 10 e la reingrandisco di 10 non avrò mai la risoluzione iniziale

- The lack of semantic information has some drawbacks. What happens if you shrink and then enlarge an image?




- Information loss!

# Information loss

- Rotating by 90° and rotating 9 times by 10° do not lead to the same result

- Only rotations of 90° (or multiples) can be handled without information loss

IMMAGINI
- RUSTER
- VETTORIALI

# 2D vector graphics

modo alternativo per preservare le informazioni geometriche dell'immagine

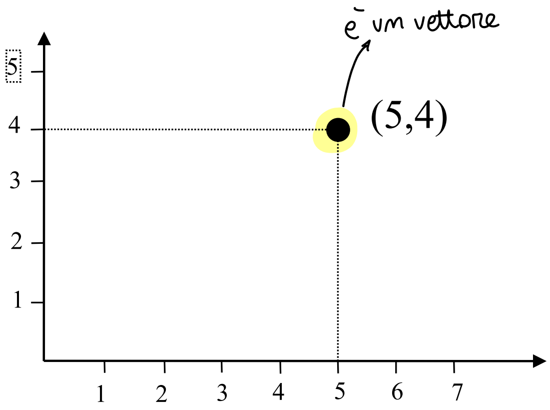
⇒ GRAFICA VETTORIALE

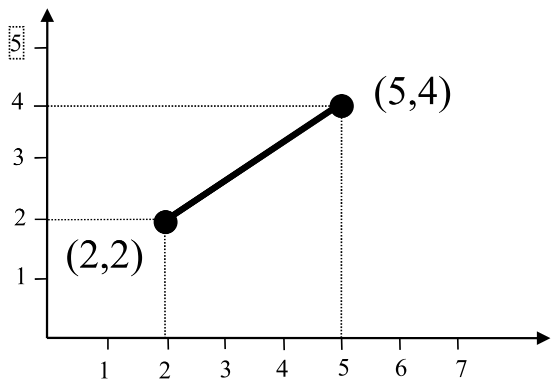
al posto che salvare un insieme di pixel salvo un'insieme di informazioni geometriche

- punti (vertici)
- linee (lati)

# 2D vector graphics

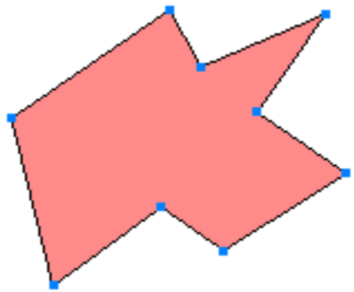
- Vector graphics relies on a geometric representation of images, using points (vertices) and segments (edges)

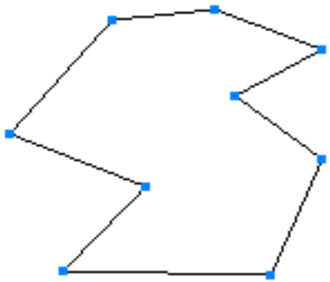


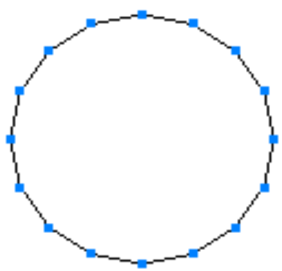



# Vertices, edges and polygons

- Vertices and edges are the basic tools to build up polygons







i font si basano sulla grafica vettoriale

# 2D vector graphics representation

- A vector draw is no more composed of a set of pixels. Rather, it is described by two sets, the vertices and edges

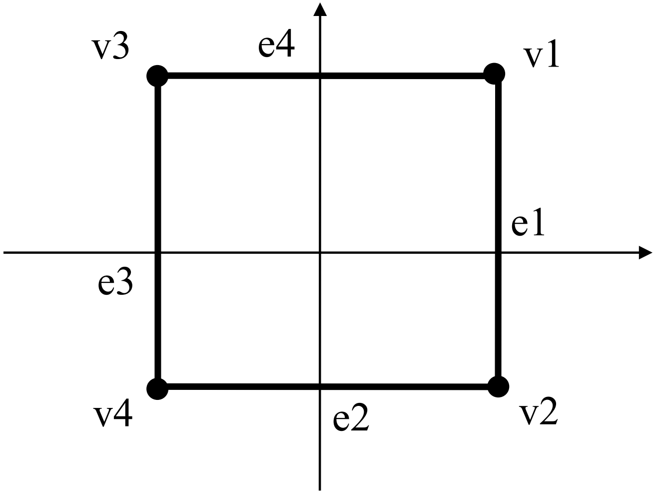
e se devo, es., fare delle modellazioni faccio delle operaz. matematiche sui vertici (es. divido per 10)

Vertici →

$$V = \{ v1(1,1), v2(1,-1), v3(-1,1), v4(-1,-1) \}$$

Lati →

$$E = \{ e1(v1,v2), e2(v2,v4), e3(v4,v3), e4(v3,v1) \}$$



# Curves in vector graphics

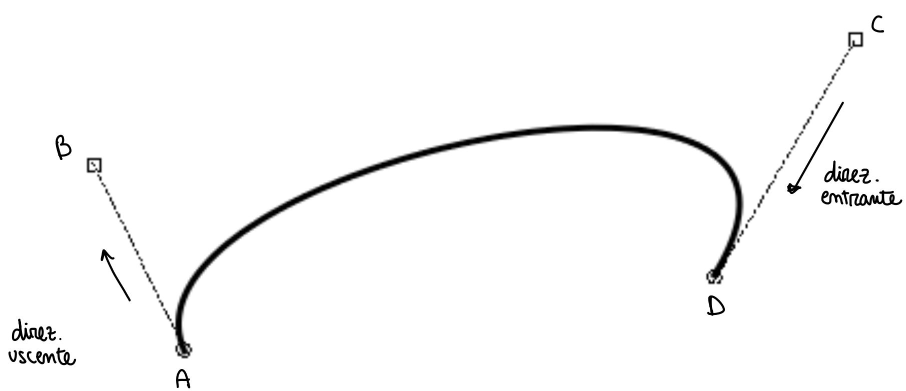
- What about curves?
- Two possible approaches:
  - Approximate the curve with segments

  - Give a mathematical description of what a curve is

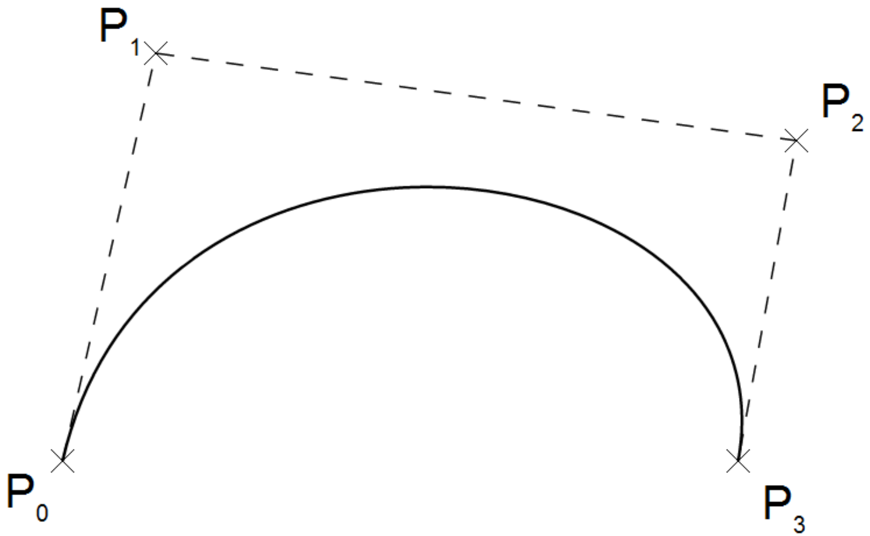
# Bézier Curves

→ mi servono 4 punti

- Mathematical description of a curve based on its geometric properties
- You need to define a start point, an end point, and the tangent directions of the curve in those points



# Bézier curves



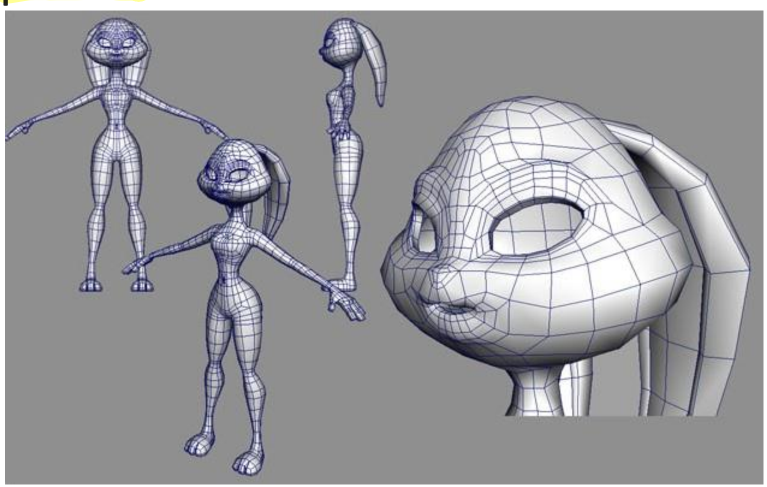

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ t \in [0,1].$$

Parametric equation of a cubic Bézier curve

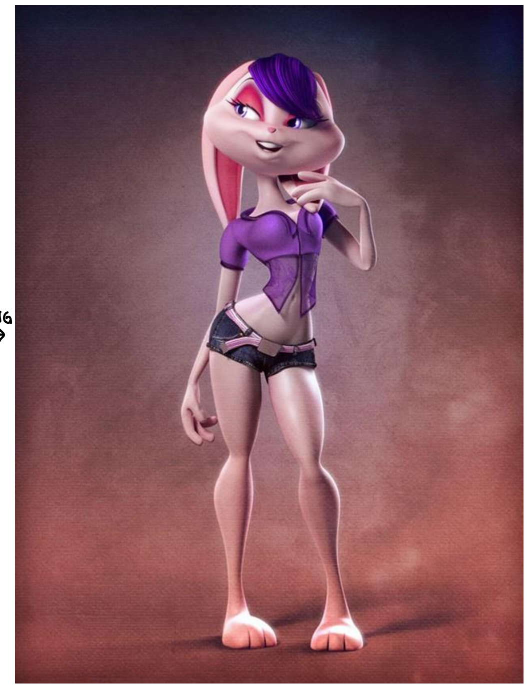
# 3D vector graphics

# 3D graphics

- We have seen that working on vector graphics allows to manipulate images without information loss
- This is why 3D artists work on **vector** models, which are converted to raster images only at the end of the creation process

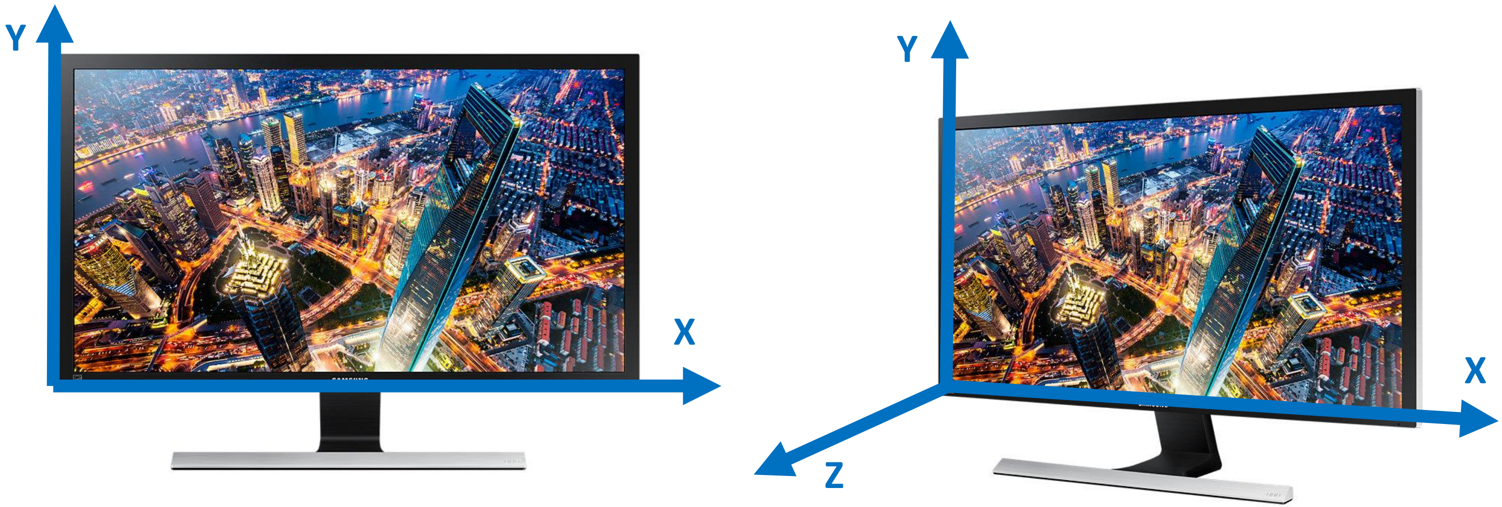


RENDERING

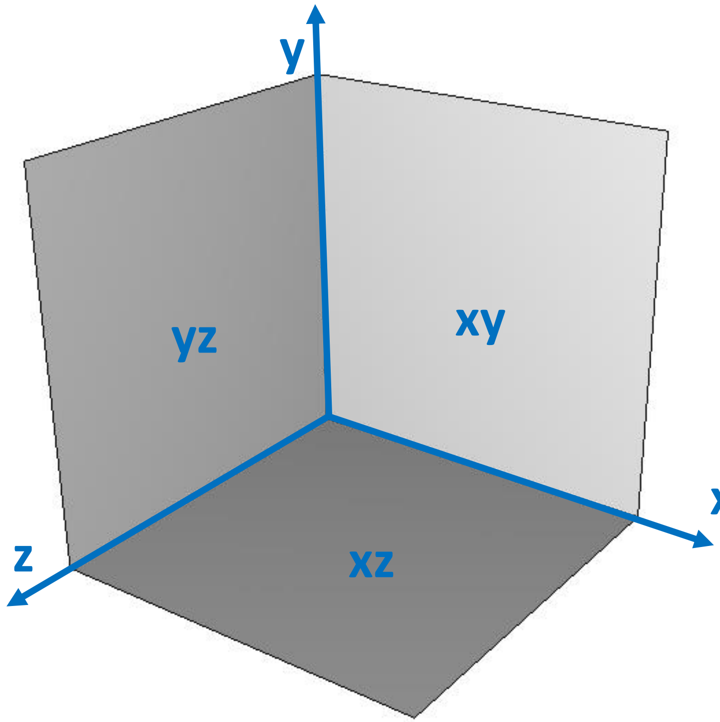


http://www.3dtotal.com/tutorial/359-making-bunny-photoshop-maya-by-carlos-ortega-character-animal-bunny-rabbit

# From 2D to 3D




- The third dimension “exits” from the screen

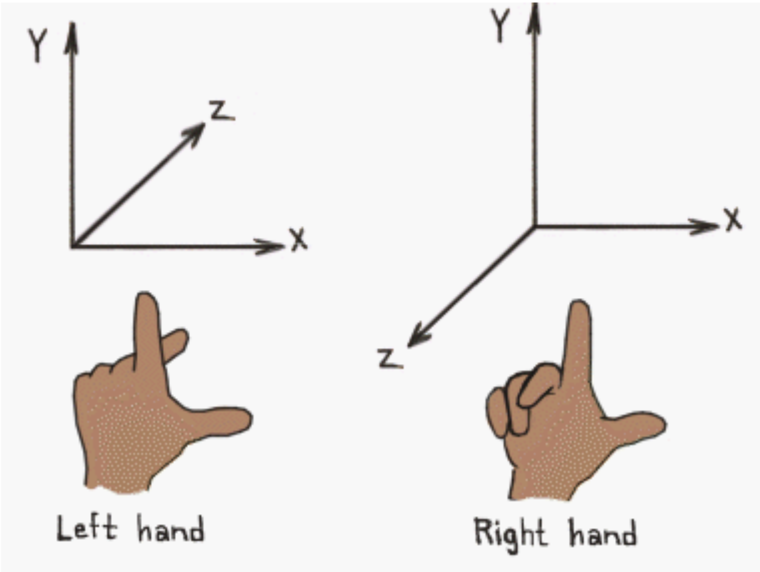



- The three axes define the three main planes

# Reference systems

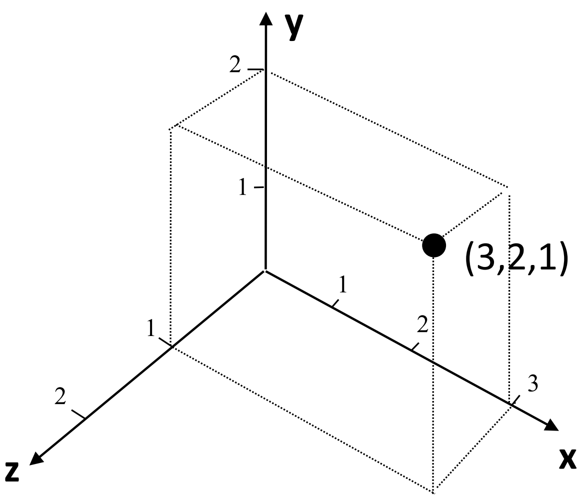
- The three axes define a reference system. Not all the systems are equal! Need to define:
  - Which axis aims upward (or downward)
  - If the system is left-handed or right-handed

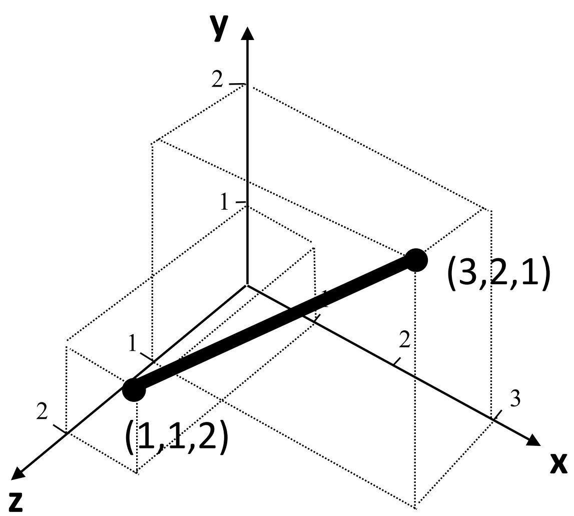
→ non ci sono trasformaz. per passare da left-handed a right-handed



# 3D image representation

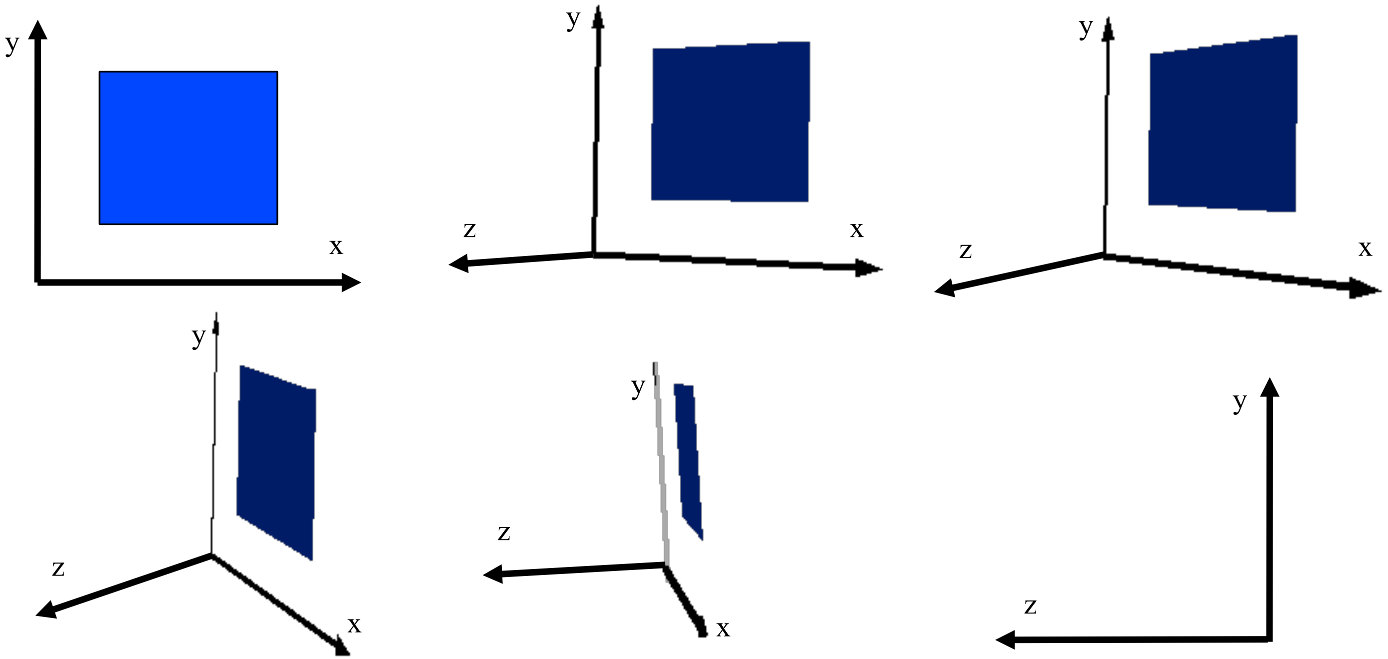
- Still vertices and edges, this time in a 3D space



# Polygons

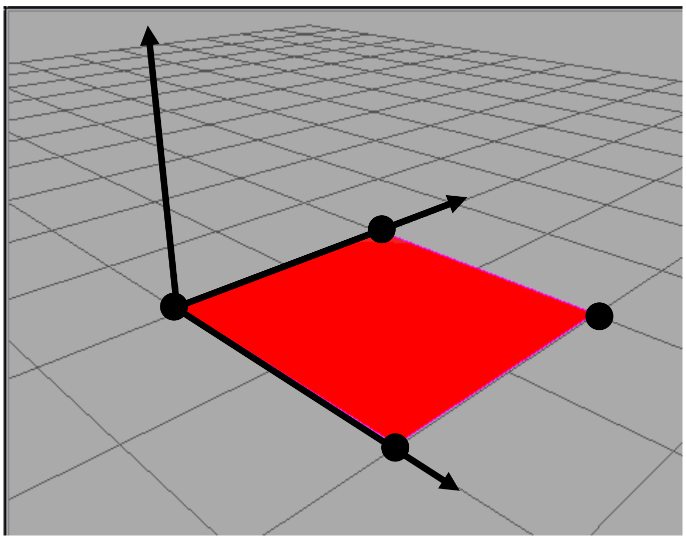
- Polygons are still 2D surfaces, they have no depth

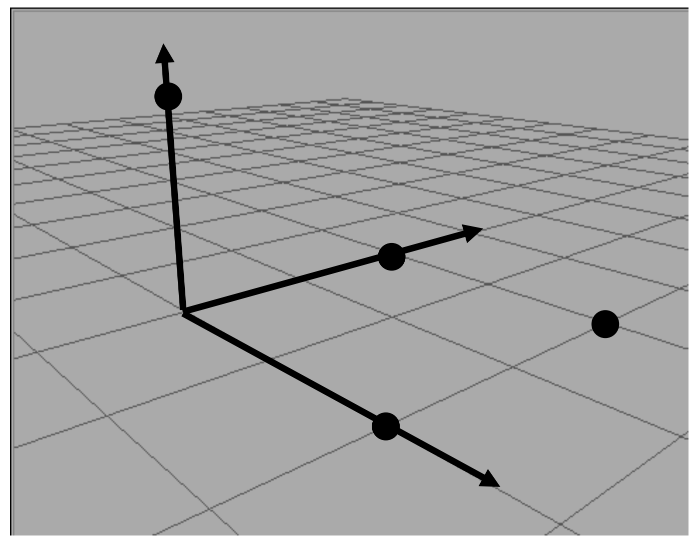


# Polygons in 3D

infatti es. dati 2 punti riesco a disegnare SEMPRE una retta passante, dati 3 punti non sempre riesco a disegnare una retta passante ⇒ questo vale nel 2D. nel 3D il limite è 3, con 4 non riesco più

- Warning: not all sets of vertices and edges define a polygon. Polygons are flat surfaces, and if the vertices are > 3 the surface passing through all of them may not exist



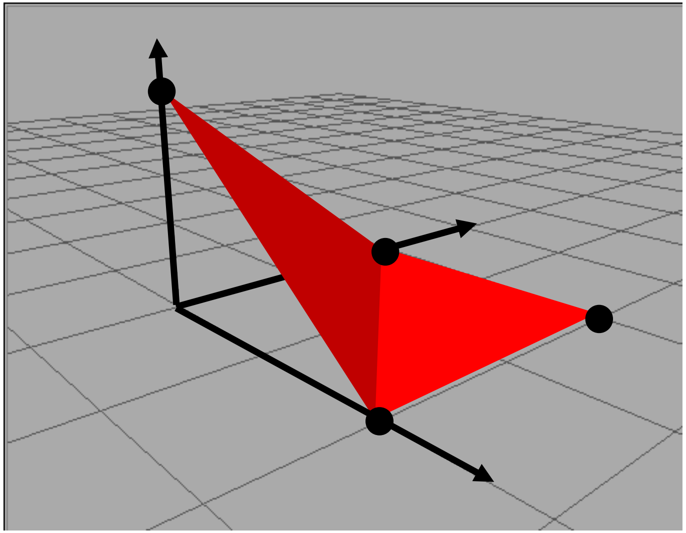


(2D equivalent: it is not always possible to draw a straight line across 3 or more points)

# Triangles

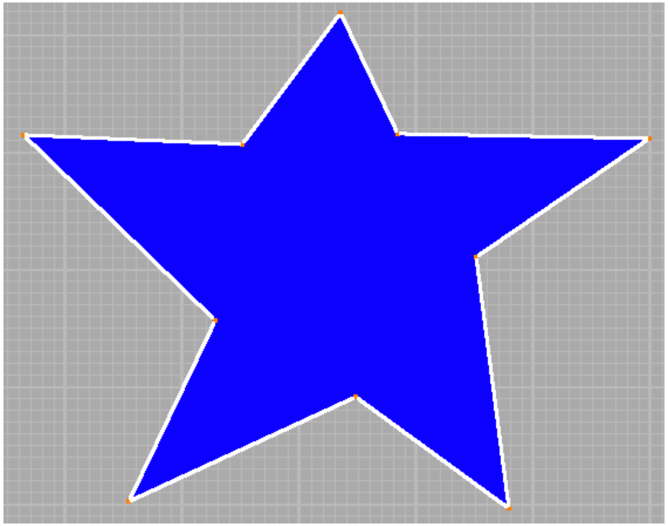
=> ogni poligono è composto da tanti triangoli

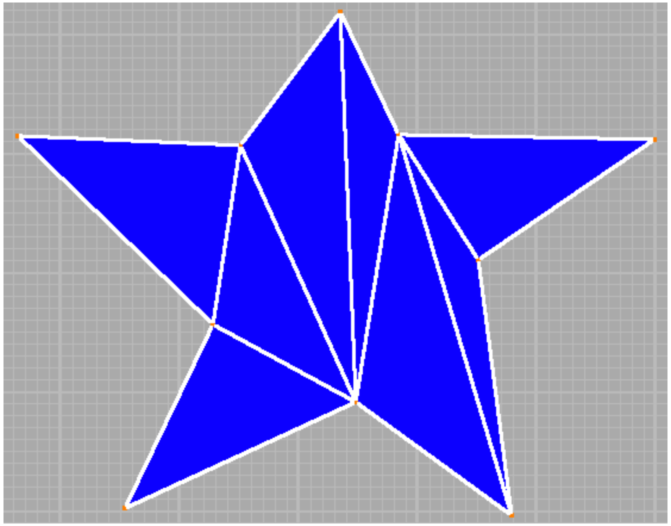
- However, three vertices always define a polygon (a triangle). This is why triangles are so important in 3D graphics



# Triangulation

- Any generic polygon can be decomposed in a set of triangles. The decomposition is not unique, but the number of resulting triangles is fixed





Curiosity: if the polygon has n vertices, the triangulated version has n-2 triangles and n-3 new edges. No new vertices are added

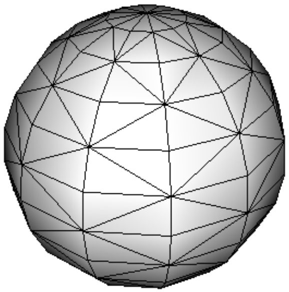
# Triangulation: pros and cons

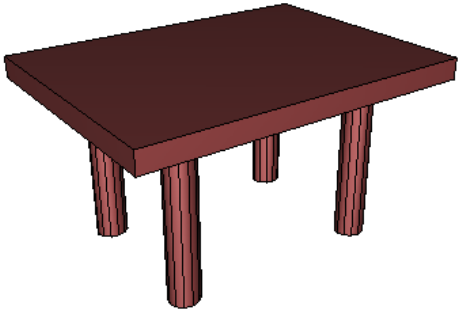
- Pros: no more impossible shapes due to vertices not lying on the same plane

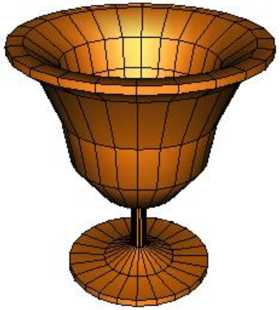
- Cons: the image complexity increases

# Polyhedra → insiemi di poligoni uniti da lati e vertici

- A polyhedron is set of polygons held together by shared vertices and edges
- E.g. the result of triangulation is a polyhedron
- Polyhedra can describe 3D shapes





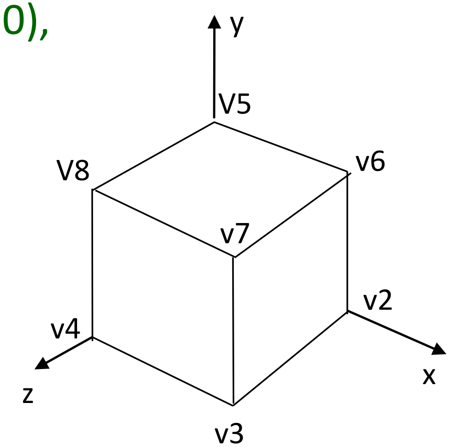


VOXEL: equivalente 3d di un pixel, quindi è un cubettino → questa rappresentazione serve perchè "riempe" il poliedro, a differenza della classica rappresentazione (POLIGON.). Usata in medicina, es. TAC che fa un'immagine voxel che può poi essere sezionata

# Representation of 3D polyhedra

- Described by a set of **vertices**, one of **edges** and one of polygons (**faces**)

V = {v1(0,0,0), v2(1,0,0), v3(1,0,1), v4(0,0,1), v5(0,1,0), v6(1,1,0), v7(1,1,1), v8(0,1,1)}



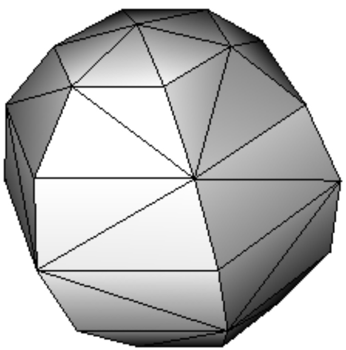

E = {e1(v1,v2), e2(v2,v3), e3(v3,v4), e4(v4,v1), e5(v5,v6), e6(v6,v7), e7(v7,v8), e8(v8,v5), e9(v1,v5), e10(v2,v6), e11(v3,v7), e12(v4,v8)}

P = {p1(e1,e2,e3,e4), p2(e5,e6,e7,e8), p3(e3,e12,e7,e11), p4(e1,e9,e5,e10), p5(e4,e12,e8,e9), p6(e2,e11,e6,e10)}

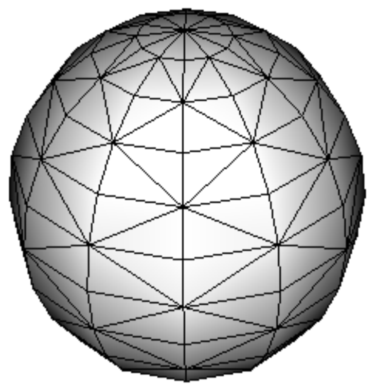
# Curved surfaces

come gestisco le superfici curve?

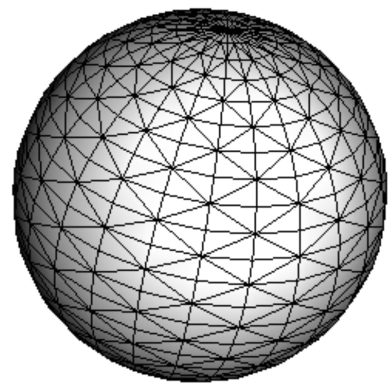
- Two approaches:
  - Mathematical description (e.g. NURBS)
  - Polyhedron approximation



32 vertices, 60 faces



134 vertices, 264 faces



554 vertices,1104 faces

Curiosity: in any polyhedron without holes, the number of vertices, edges and faces are related according to the Euler formula: faces = edges – vertices + 2

# Transformations

# Transformations

- Working on vector models allow us to manipulate (transform) 3D objects without information loss
- Basic transformations:
  - Translations
  - Rotations
  - Scale changes

→ trasformaz. che posso descrivere matematicamente

!! una trasformaz. che applico ad un oggetto 3d è fatta solo sui VERTICI
i lati vengono da se

che sono dei vettori

- Mathematically described by algebraic operations (matrix-vector multiplications)

↳ PRODOTTO MATRICE VETTORE

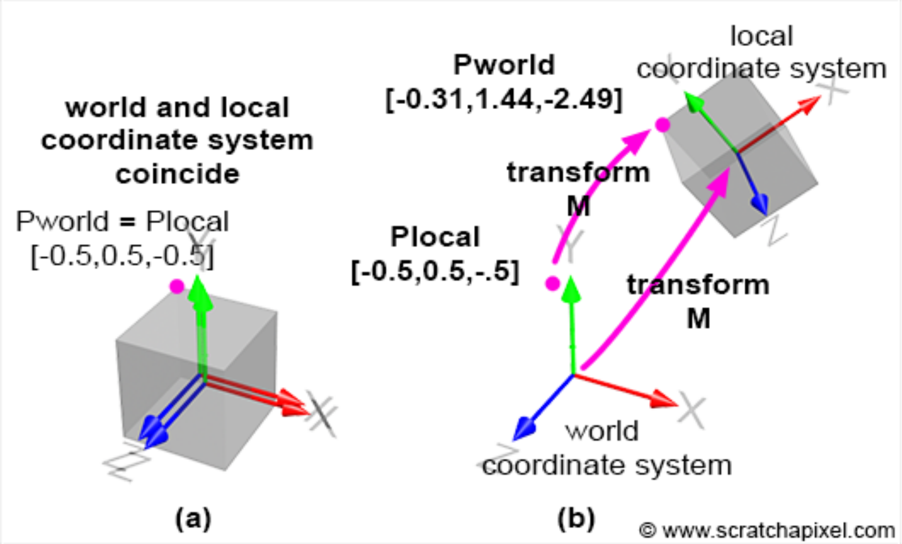
$$\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

# Local and global transformations

relative all'oggetto stesso

- Local: relative to a reference system that moves with the object (**object coordinates**)
- Global: relative to a fixed reference system (**world coordinates**)

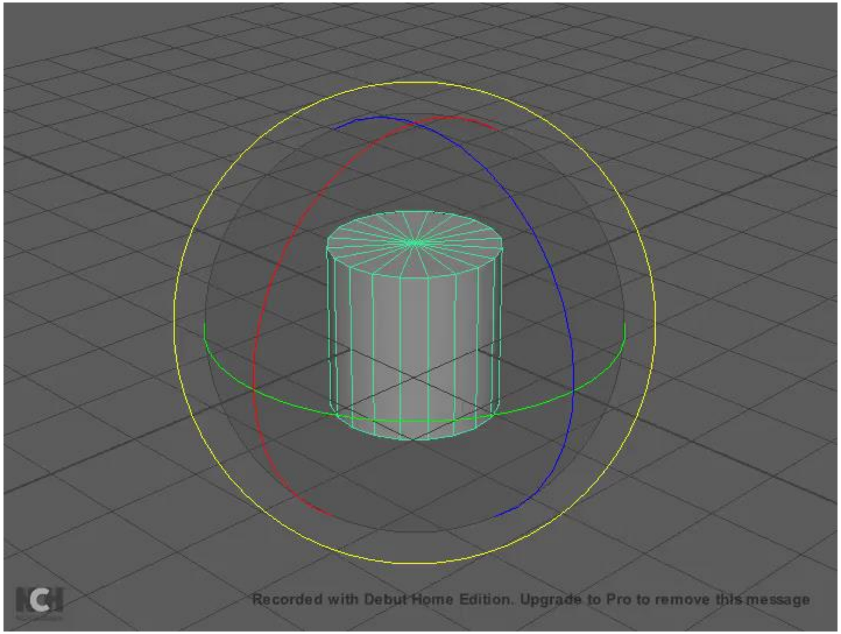
relative ad un sistema di riferimento fissato



# Pivots

→ il punto che non si sposta durante la trasformazione
CENTRO di una trasformazione

- The pivot is the center of a transformation, the point that does not move even when the objet is rotated/scaled



# Projections

i modelli 3d noi li vogliamo mostrare su uno schermo che è 2D, raster per la precisione

Come rappresento un'immagine 3D su uno schermo 2D? → PROIEZIONI

# Projections

- In order to visualize a 3D image on a 2D screen, we must drop a dimension
- The process of creating 2D representations of 3D images is called projection

- Basic idea: the projection plane is like a transparent plane between the object and the observer
- The intersections of ray lights from the object to the observer with the plane define the projection

→ una PROIEZIONE è PROSPETTICA se l'osservatore è posto a distanza FINITA dal piano di proiezione e dall'oggetto

⇓

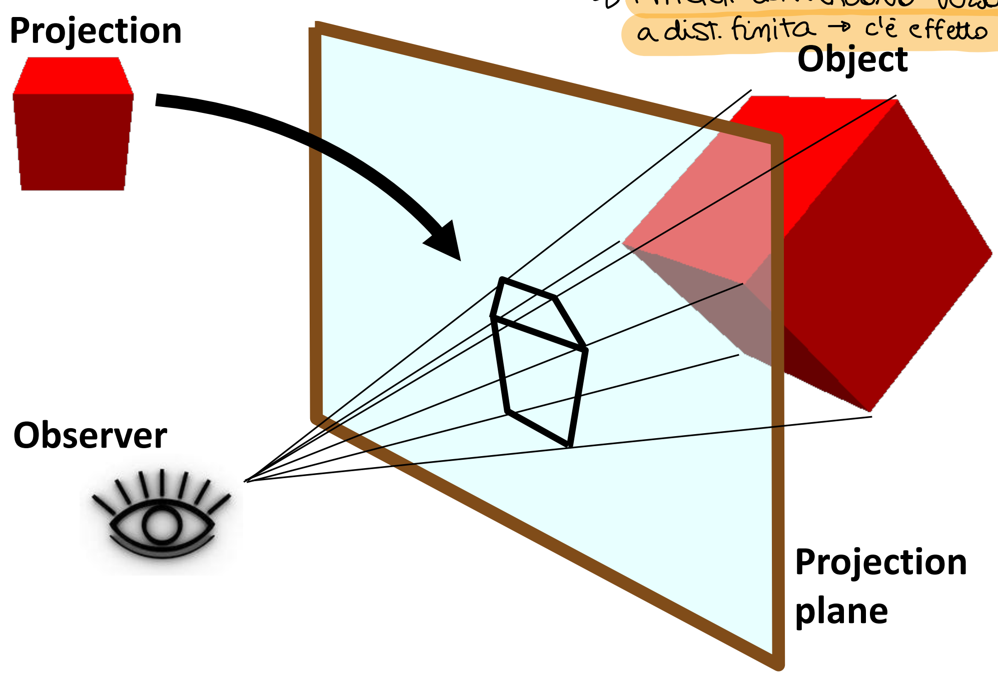
qualsiasi esempio reale

↳ una conseguenza è la DISTORSIONE PROSPETTIVA

# Perspective projections

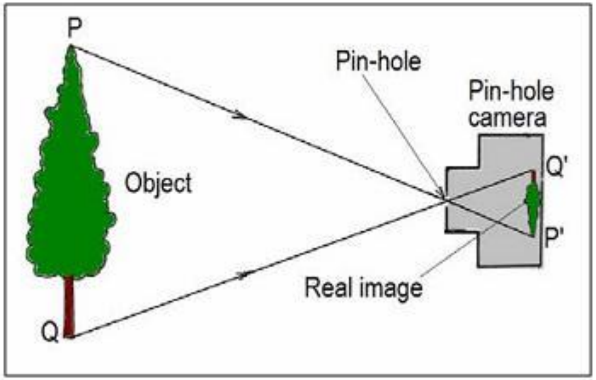
- In perspective projections, the observer is placed at finite distance from the projection plane

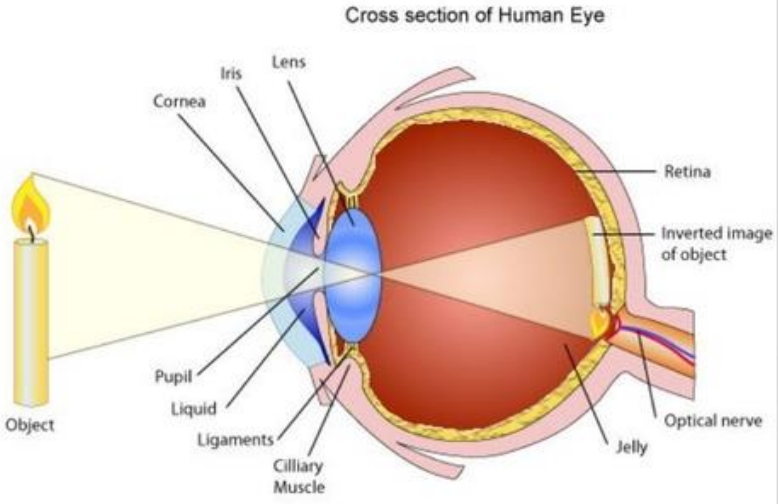
↳ i RAGGI CONVERGONO verso uno SPETTATORE a dist. finita → c'è effetto DISTORSIONE PROSPETTIVA



# Perspective projections

- This is what happens in cameras or eyes



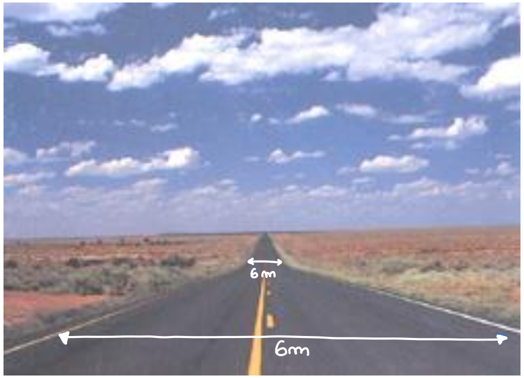



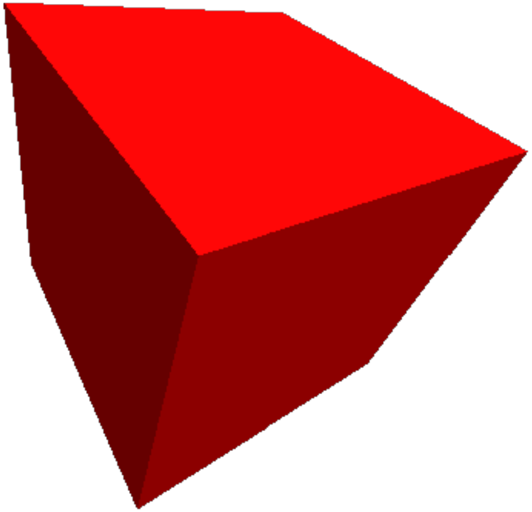
# Perspective distortion

vedo le cose lontane più piccole

- In perspective projections, perspective **distortion** occurs: lines that are parallel in the 3D world, seem to converge towards a *vanishing point*

linee parallele sembrano convergere nella proiezione

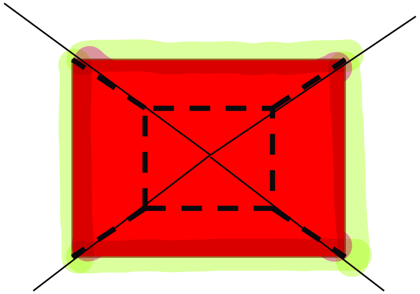


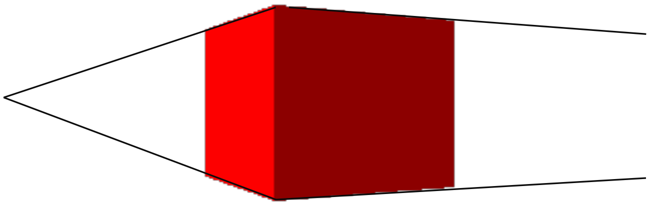



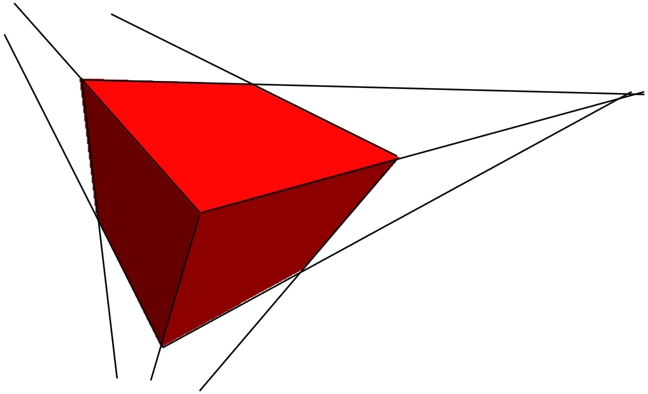
# Vanishing points

→ nella proiezione due linee rimangono parallele se sono parallele al piano di proiezione

- Parallelism is preserved only for lines parallel to the projection plane. Thus, depending on the object shape and position, there can be multiple vanishing points



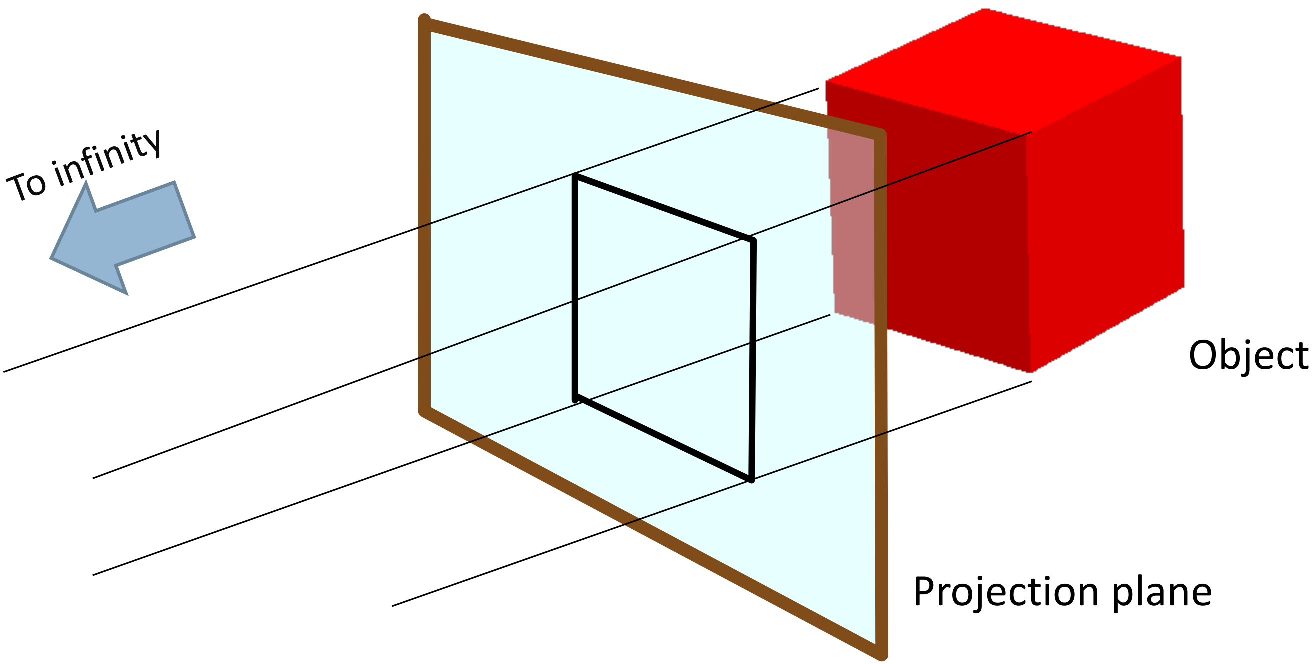




# Axonometric projections

→ distanza ∞, non c'è distorsione prospettica (si mantiene il parallelismo)

- In **axonometric projections**, the observer is placed at **infinity** and the light rays are **parallels**

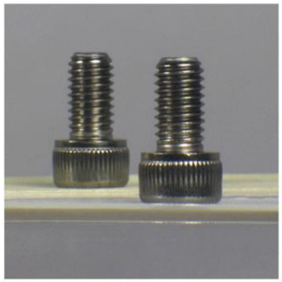


# Axonometric projections

- Examples from machine vision, where special lenses (called telecentric lenses) are often used to achieve axonometric projections

PROIEZIONE ASSONOMETRICA

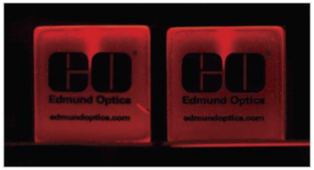
PROIEZIONE PROSPETTICA



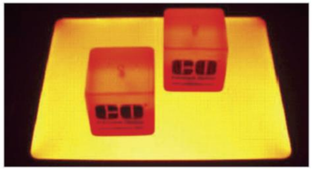
PROSP.

FIXED FOCAL LENGTH LENS


TELECENTRIC LENS ASSON.

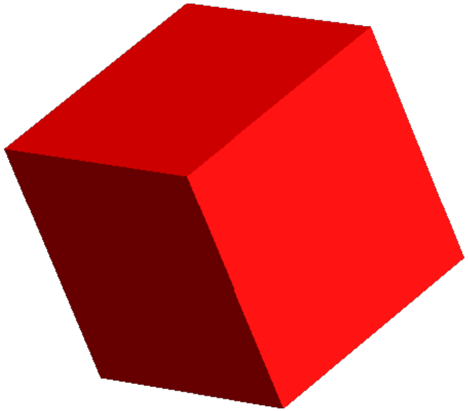



SETUP



# Axonometric projections

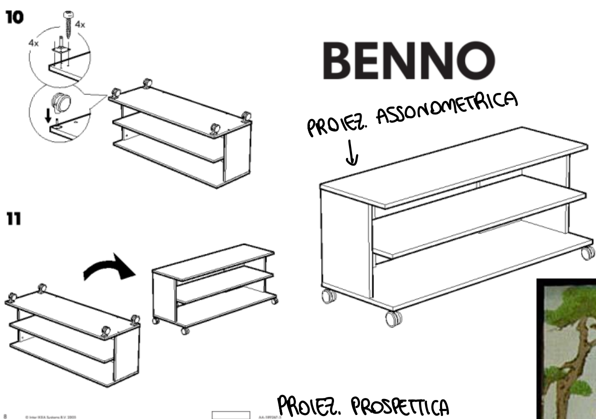
- Parallel lines in 3D are parallel in axonometric projections too.
- There is no perspective distortion. Objects with the same 3D size have the same 2D appearance



# Axonometric projections

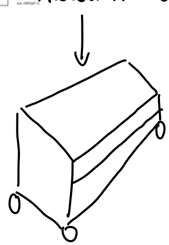
→ ad es. usate nei cataloghi IKEA

- Useful when it is important to clearly show spatial relations, because of the lack of distortion



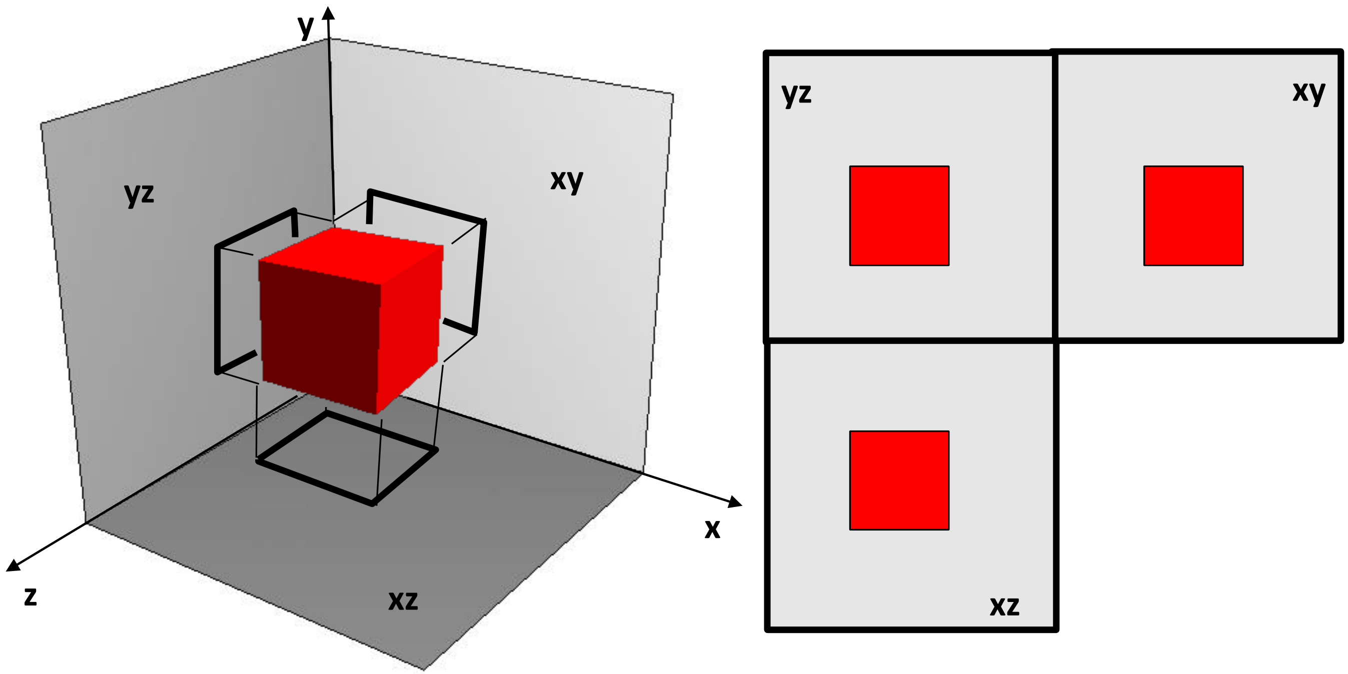

→ se fosse usata la PROIEZ. PROSPETTICA avrei che il lato più vicino risulta più grande e (nel caso in cui i due lati non sono uguali) potrebbe creare confusione

PROIEZ. PROSPETTICA



# Orthogonal projections

- Orthogonal (or orthographic) projections are just axonometric projections on the three main planes XY, XZ and YZ

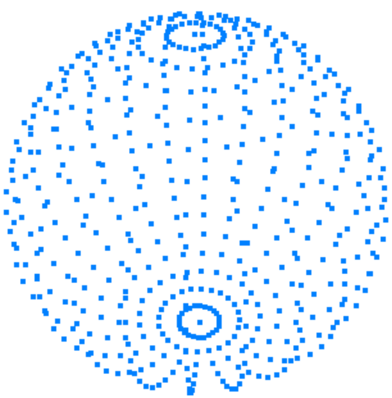


# Visualization of 3D objects

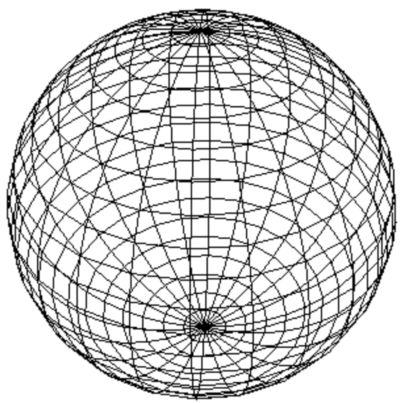
# What to show

- Vertices only (point cloud) 1
- Edges only (wireframe) 2
- Faces 3

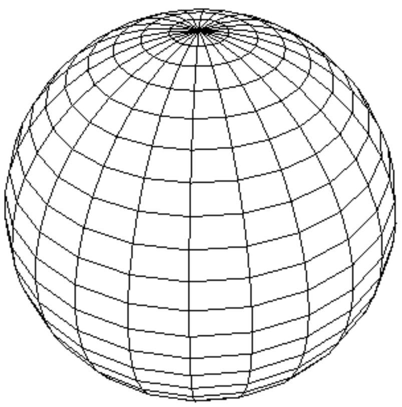
1



2

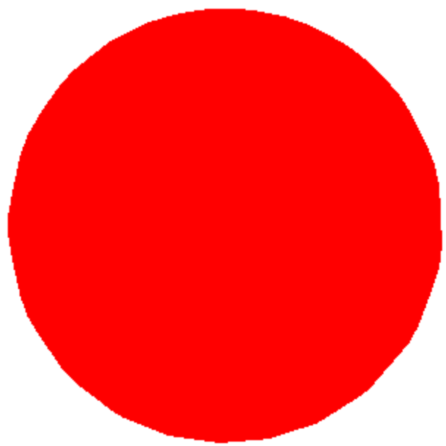


3



# Showing faces

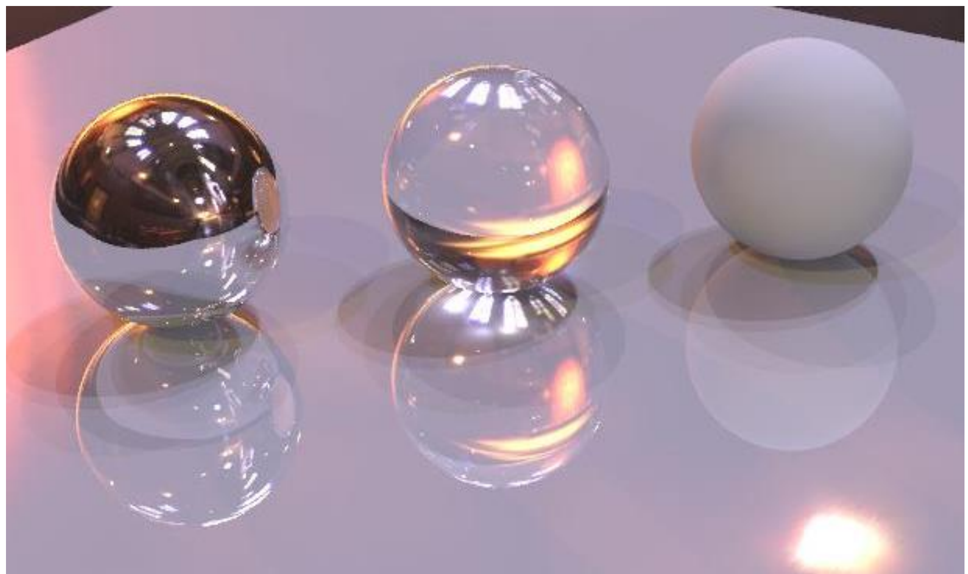
- In order to show the faces, they must have a color
- However, if all the faces were of the same color, the resulting image would be totally flat



# Materials

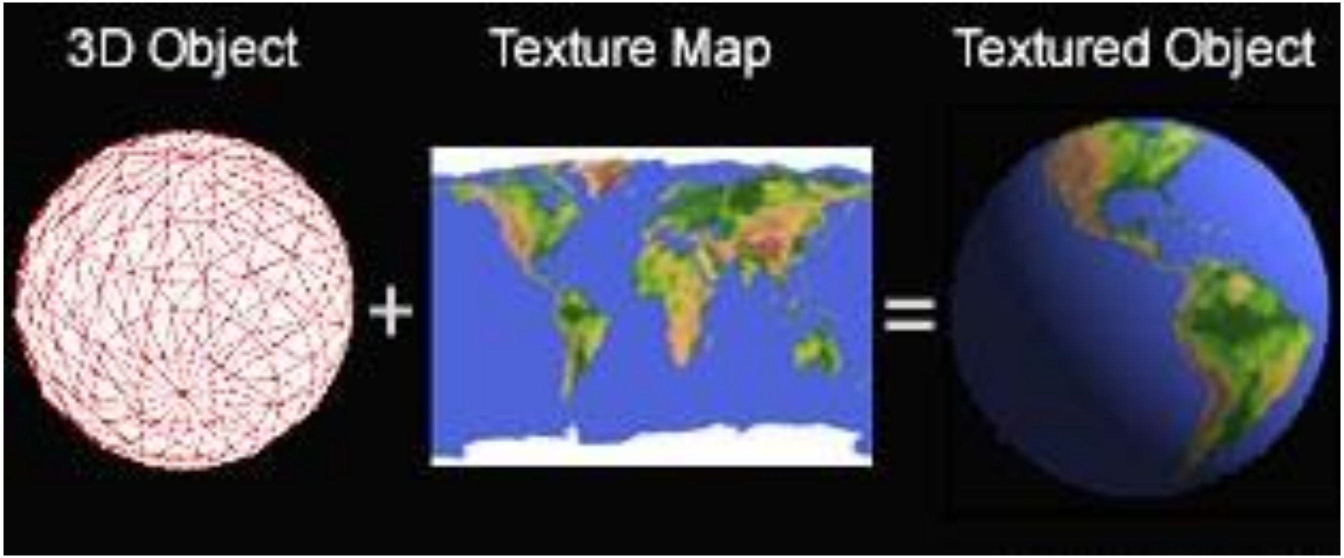
→ MATERIALE è l'insieme di cose che descrive come deve essere la resa visiva di un oggetto (deve avere geometria (forma) e resa visiva, sennò l'oggetto NON PUÒ ESSERE VISUALIZZATO)

- A material defines not only the color, but also how the surface reacts to light, thus modelling several properties affecting the final appearance:
  - Color
  - Transparency
  - Shininess
  - …



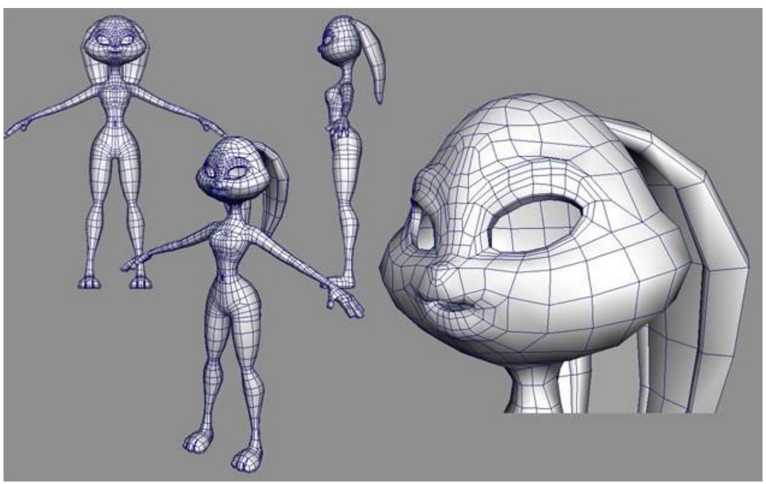
# Textures

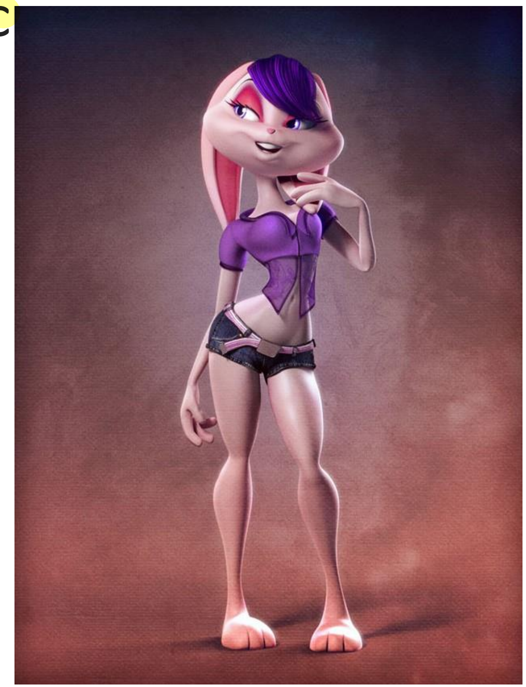
- Materials can be used to apply images (textures) on the surface of 3D objects



# Rendering

- The term **rendering** denotes the process of creating a final raster image from a 3D model and the corresponding materials, lights, etc
- A **render engine** is a software computing renders





http://www.3dtotal.com/tutorial/359-making-bunny-photoshop-maya-by-carlos-ortega-character-animal-bunny-rabbit