The Graphics System

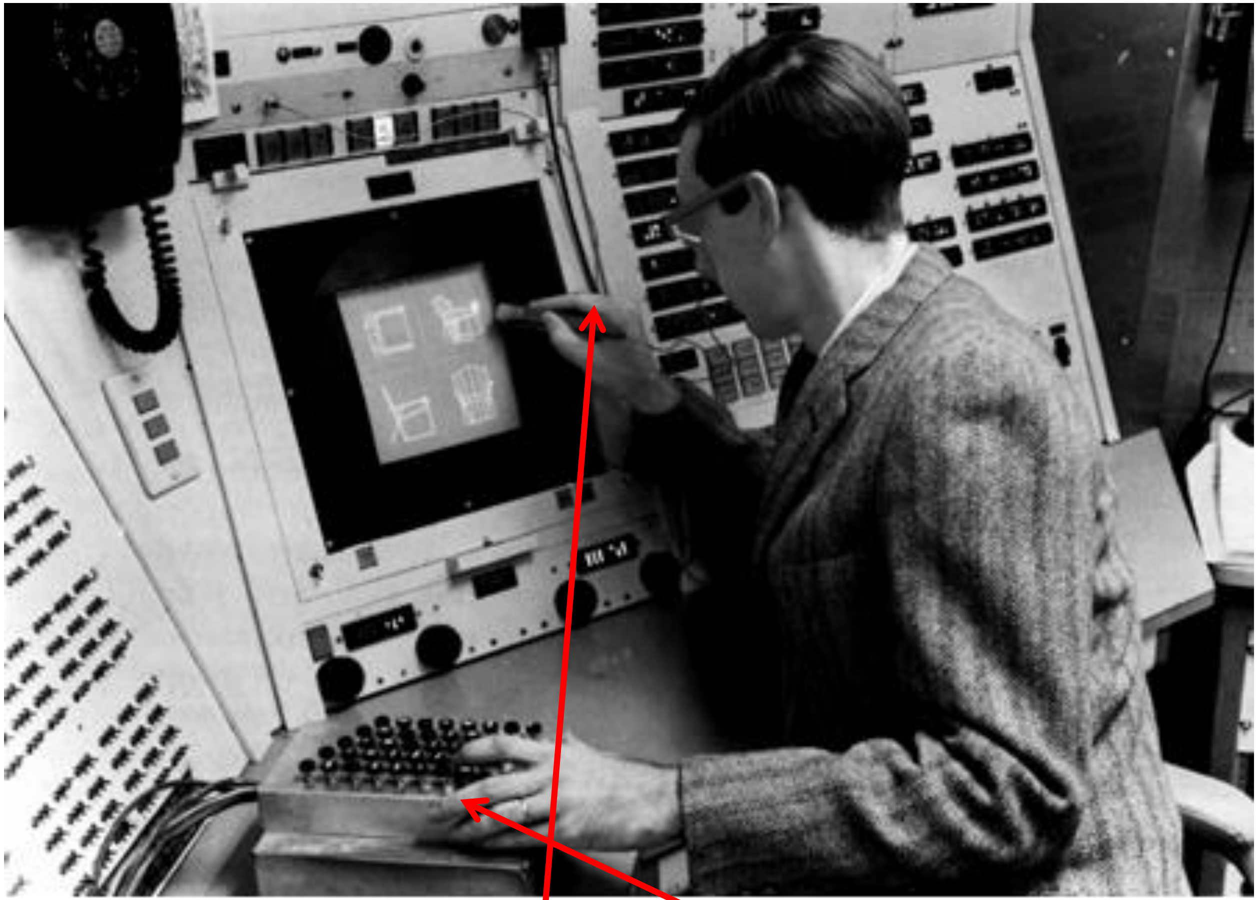
Fondamenti di Computer Graphics

Serena Morigi

serena.morigi@unibo.it

http://www.dm.unibo.it/~morigi/

From batch-mode to interactive mode



Human
Computer
Interaction

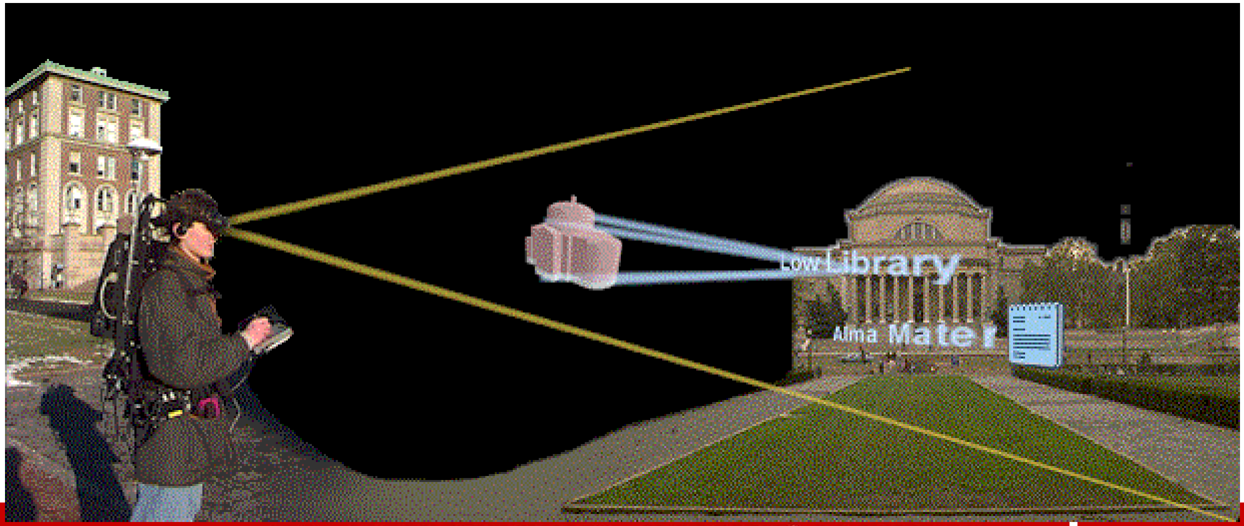
Sketchpad in 1963 . by Ivan Sutherland (MIT1963)
(CRT monitor, **light pen** and **function-key panel**)

Semi-immersive VR

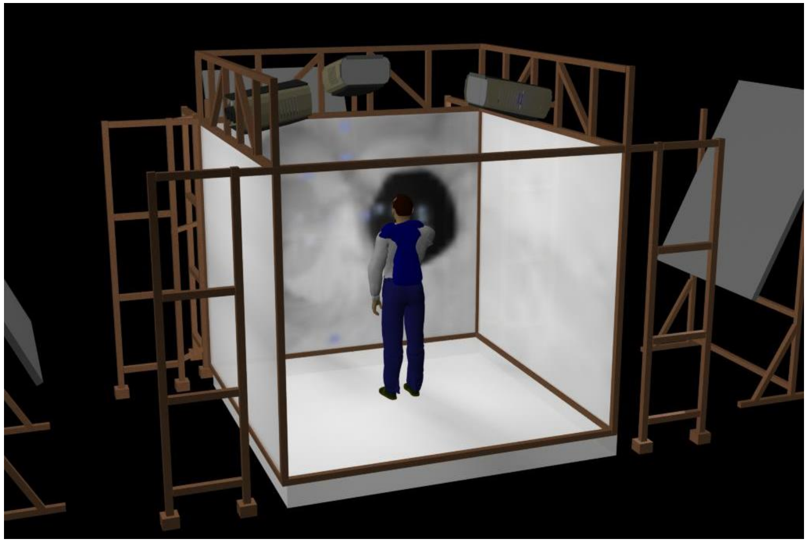
Fishtank VR on a monitor
small volume, head-tracked
interactive stereo

REAL-TIME (24fps)
+
3D VISION
+
3D INPUT (INTERACTIVITY)
=
IMMERSIVITY

augmented VR
(via video see-through optics)



Fully immersive VR: Graphics is Pervasive
(via Head-mounted Displays HMD, Cave [180 George St.])



VR replaces the real world with a simulated one

Cave Painting

Use feet for navigation, freeing hands for other uses

Virtual Reality refers to computer technologies that use software to generate realistic images, sounds and other sensations that replicate a real environment (or create an imaginary setting), and simulate a user's physical presence in this environment.

Augmented Reality

Augmented Reality blends what the user sees in their *real* surroundings with digital content generated by computer software.

AR systems use a camera to capture the user's surroundings or some type of display screen which the user looks at :

- Google Glasses
- Microsoft's Hololens
- Magic Leap

Layar su Android

- LightWeight Augmented Reality
- Marker based Augmented Reality
- Markerless Augmented Reality

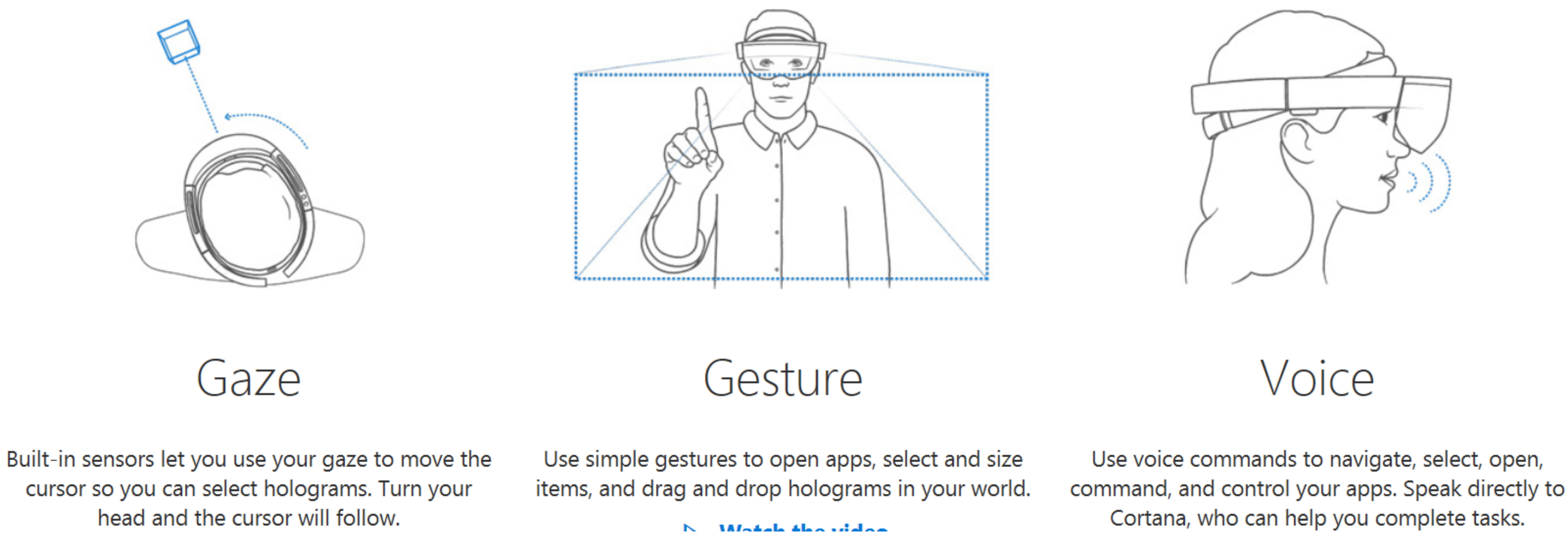
Augmented Reality Glasses

- **Google Glass born 2013- dead 2016**

- **Mixed Reality : HoloLens**

  Blend Holograms with the Real World

It uses a high-definition stereoscopic 3D optical head-mounted display, and spatial sound to allow for augmented reality applications, with a natural user interface that the user interacts with through gaze, voice, and hand gestures



A hologram is a photographic recording of a light field, (3D photography) and it is used to display a fully 3D image of the holographed subject

Mixed Reality

Rokid's Project Aurora smartglasses **(2018)**

Unlike expensive AR headsets like Microsoft HoloLens, Project Aurora doesn't have its own computer built in. Being consumer-oriented the headset connects via USB to a users mobile device and has two displays for 3D content, full-motion 6DOF (6 degrees of freedom, and spatial navigation (SLAM).

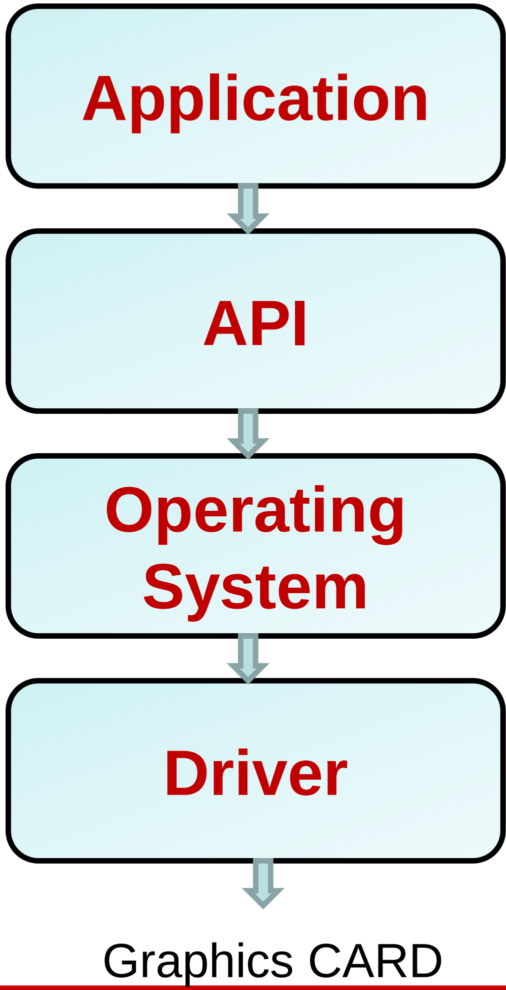
The graphics system

Graphics system = Hardware devices + graphics software to produce images

Hardware devices rapresent the power of the graphic system: the resources to capture, process and produce new images.

The **software** make it possible: manage hardware resources to manipulate graphic data.

Software (layers)



3D Game/Application

DirectX /OpenGL

Win/MaC OS etc..

Card Dependent Driver

Hardware

- **Devices (periferiche):**
  - Interactive Input devices
    - data gloves/glasses
    - Mouse
    - joystick, trackball
    - scanner 2D/3D
  - Display
  - plotter, printer
  - force feedback device, AR glasses, VR helmet
  - Digital camera

RASTER devices

- **Graphics cards (schede grafiche)**

RASTER devices

A raster device is composed of a rectangular matrix of samples or pixels. (2D scanner, digital camera, monitor, printer)

pixel (picture element).

- The **resolution** properly refers to the <u>pixel density</u>, the number of pixels *per* unit distance or area, not *total* number of pixels.

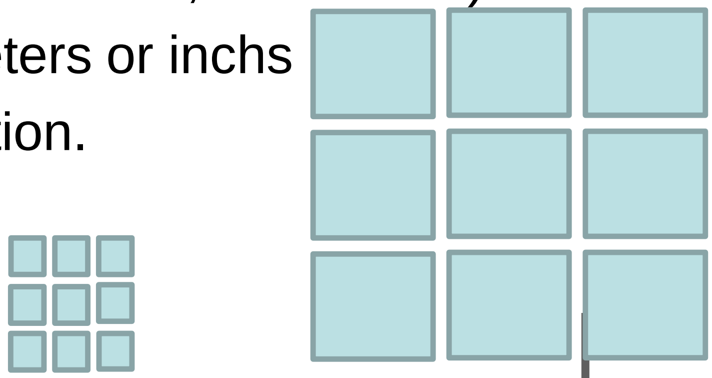
<u>video</u> inch - pixel per inch ppi - or centimeter - ppc

<u>printer/scanner</u> dots per inch dpi

Example: 300 dpi means 300x300, or 90,000 dots per square inch

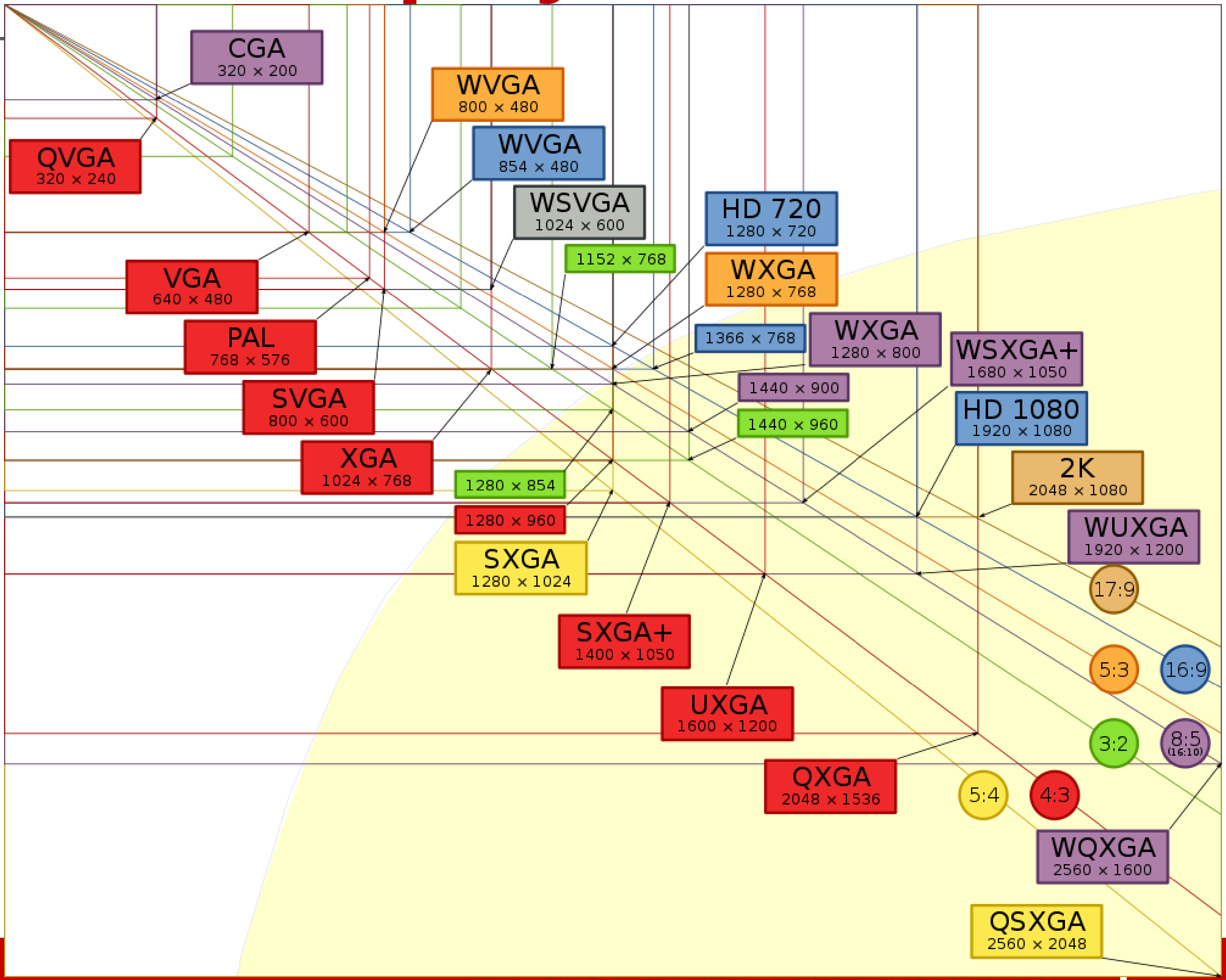
- The **display resolution** is usually used to mean *pixel dimensions,* the number of pixels in each dimension (e.g., 1920×1200, 1024×768)
- The **physical size** is wide x height in centimeters or inchs

physical size = display resolution/resolution.



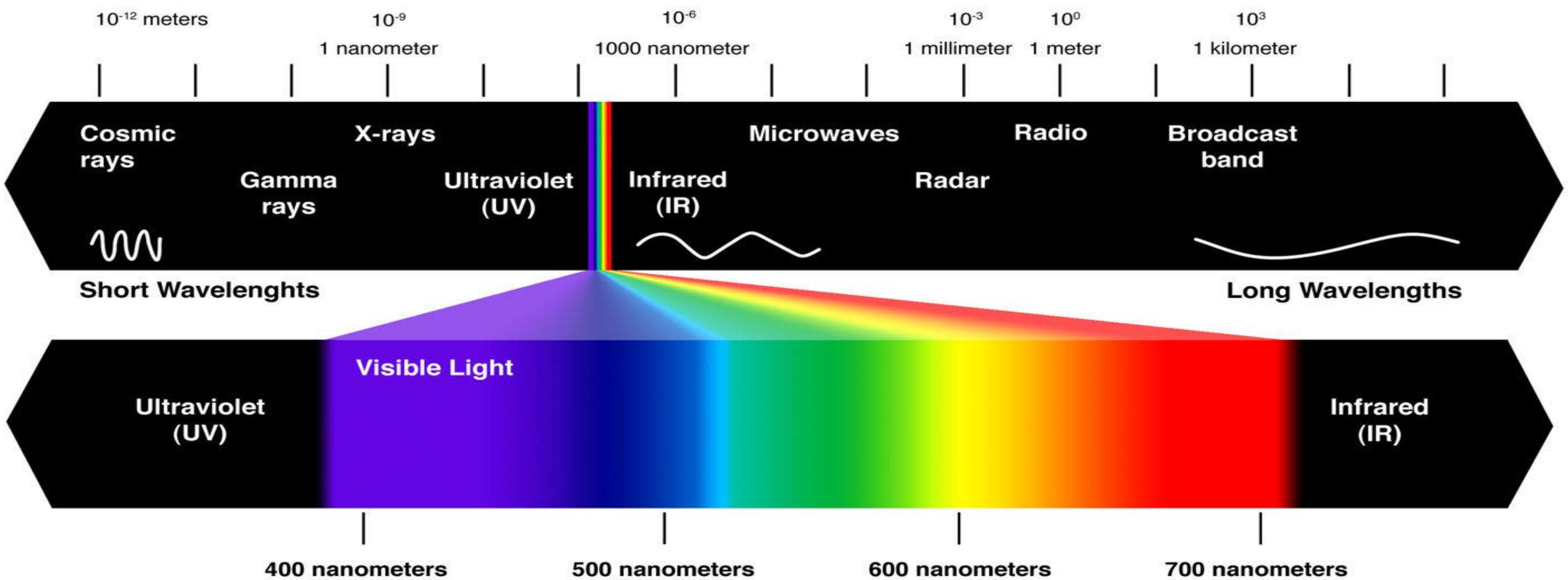
Standard Display Resolutions

Widescreen
eXtended
Super
Ultra
High Definition

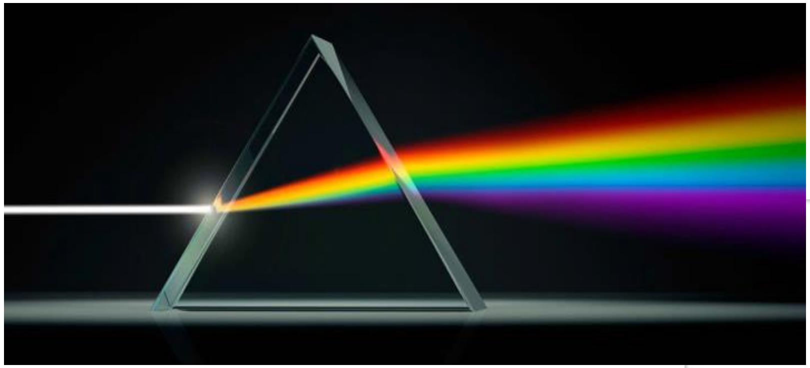


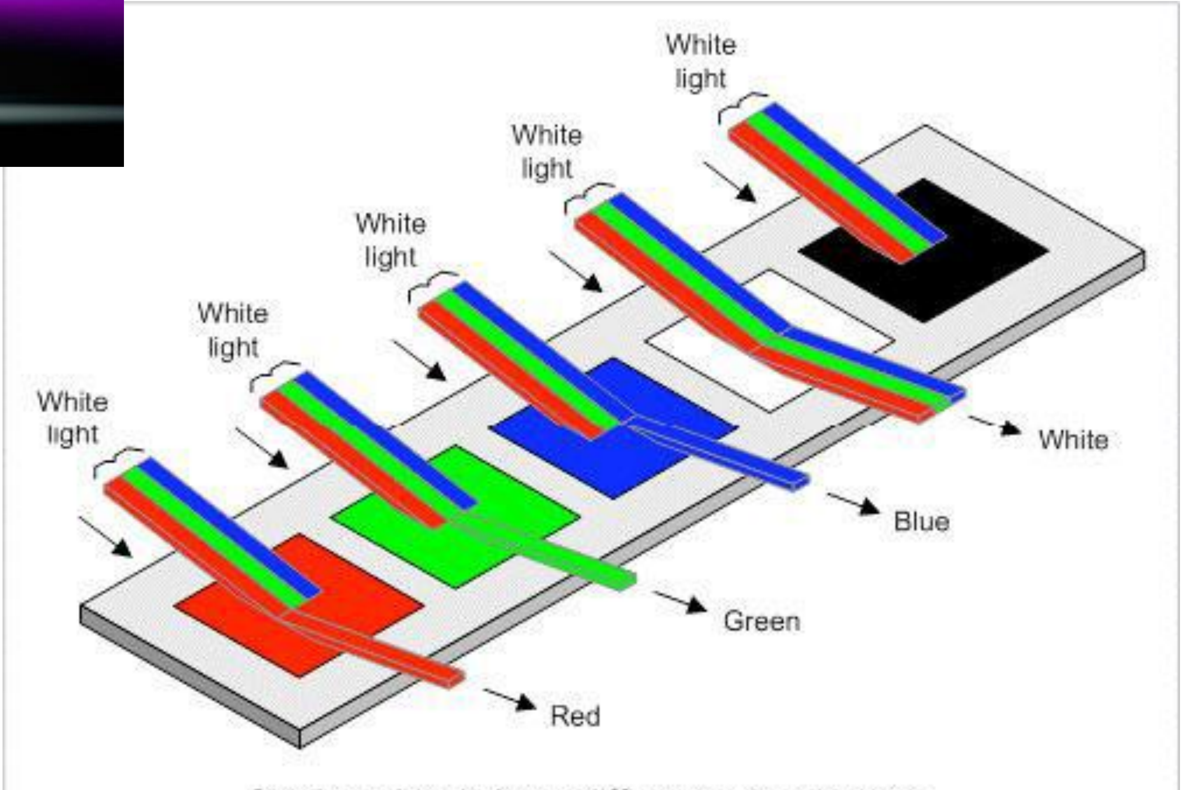
Light and Color

- Most light is not visible!
- Frequencies visible by human eyes are called ”visible spectrum”
- These frequencies what we normally think of as “color”



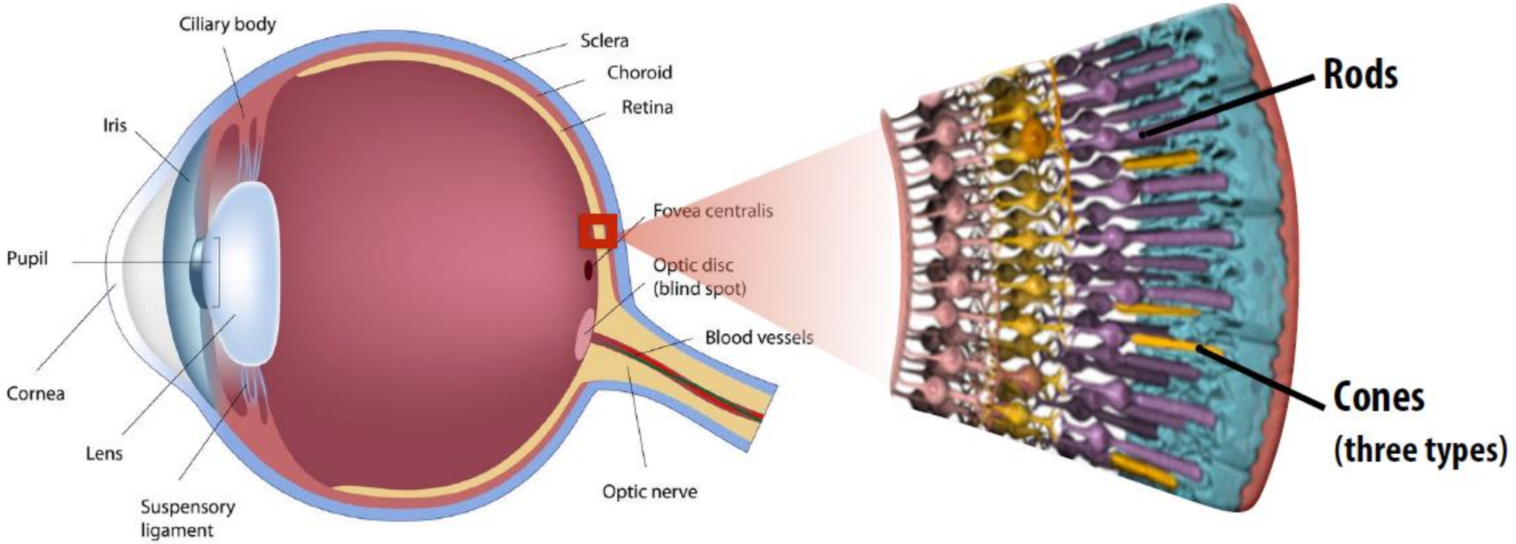
Reflecting and Absorbing Light





Shining white light on different colored paints

The eye's photoreceptor cells: rods & cones



- **Human Eye:** The retina has two kinds of sensors:
  Rods: only measure light intensity
  3 different cones: measure three colors (red, green, blue)
- **Trichromatic Theory**
  Since the human eye works with only 3 signals (ignoring rods), we work with 3 signals for images, printers, and displays.

Color Models

A **color model** is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values of *color components* :

[1.] **RGB color model**(red, green, blu)

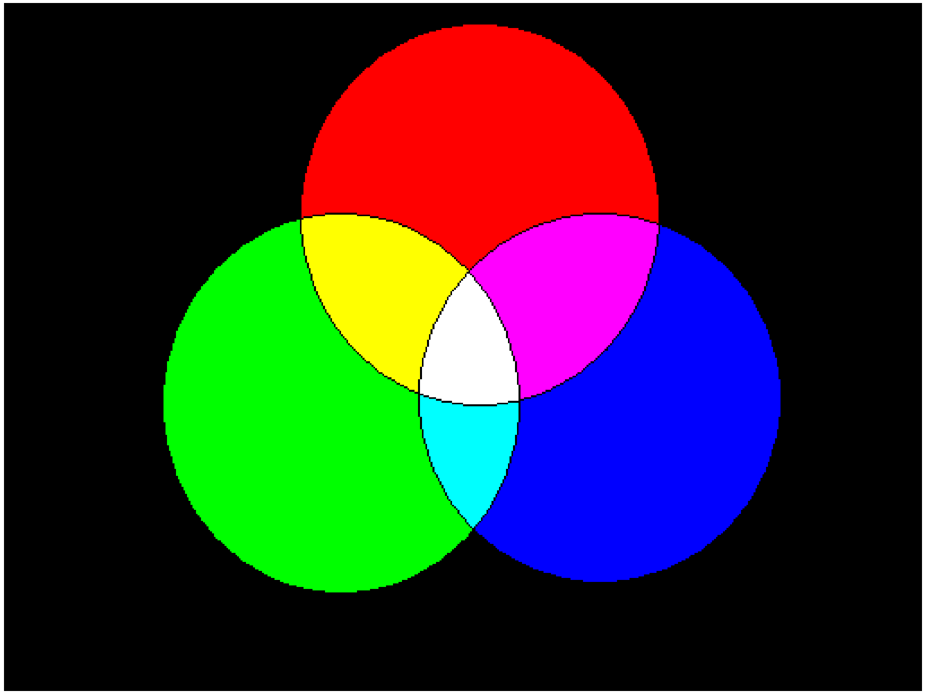
Uses additive color mixing: color components are added together to create a color from out of the darkness.

RGBA is RGB with an additional channel, alpha, to indicate transparency. (video)

[2.] **CMY color model**(cyan, magenta, yellow)

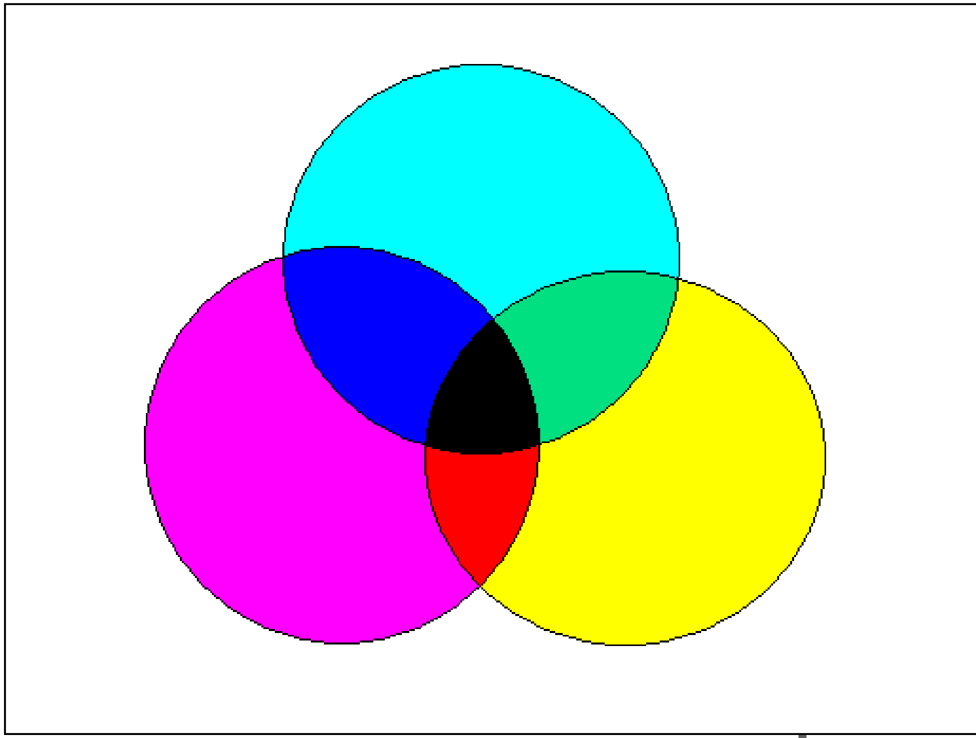
Uses subtractive color mixing One starts with a white substrate (canvas, page, etc), and uses ink to subtract color from white to create an image (printers).

$$C\ M\ Y = [1\ \ 1\ \ 1] - [R\ G\ B]$$



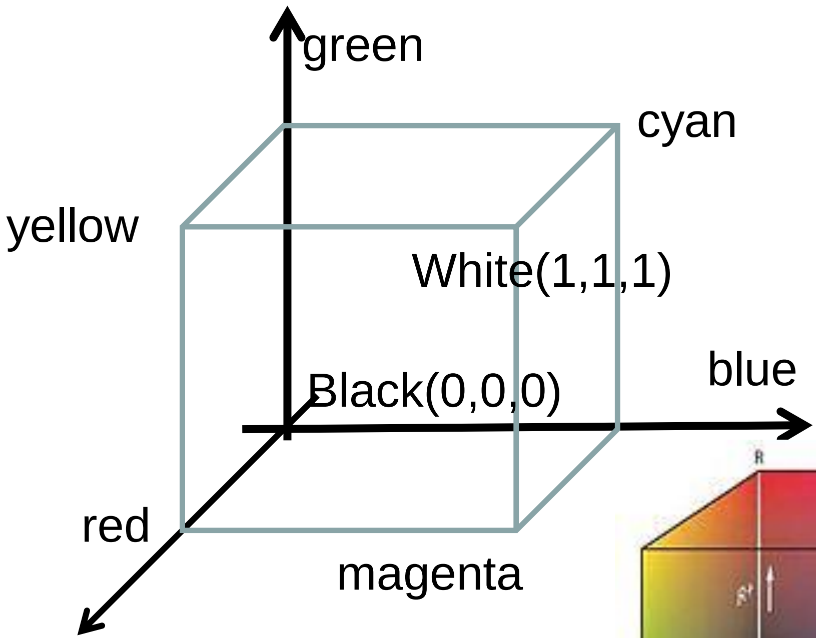
additive color mixing
(**sintesi additiva)**

subtractive color mixing
(**sintesi sottrattiva)**

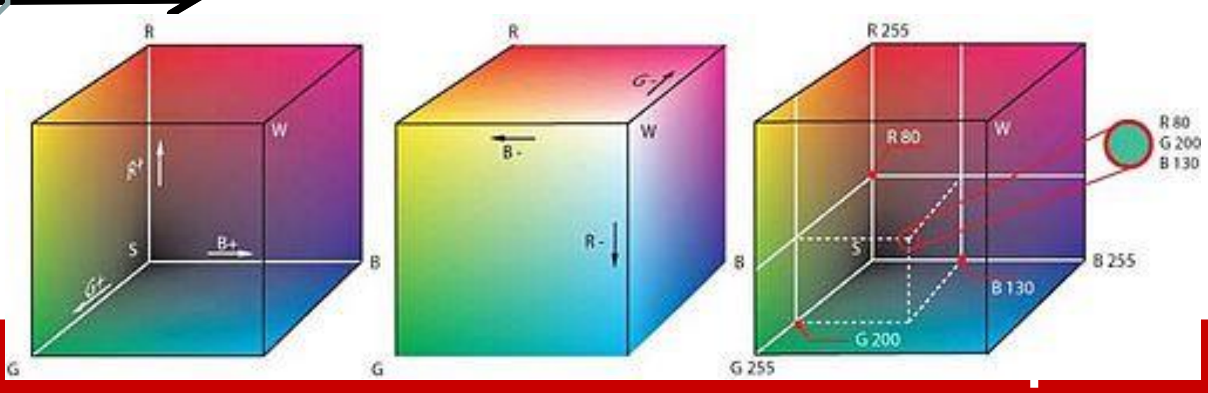


Color Model: RGB

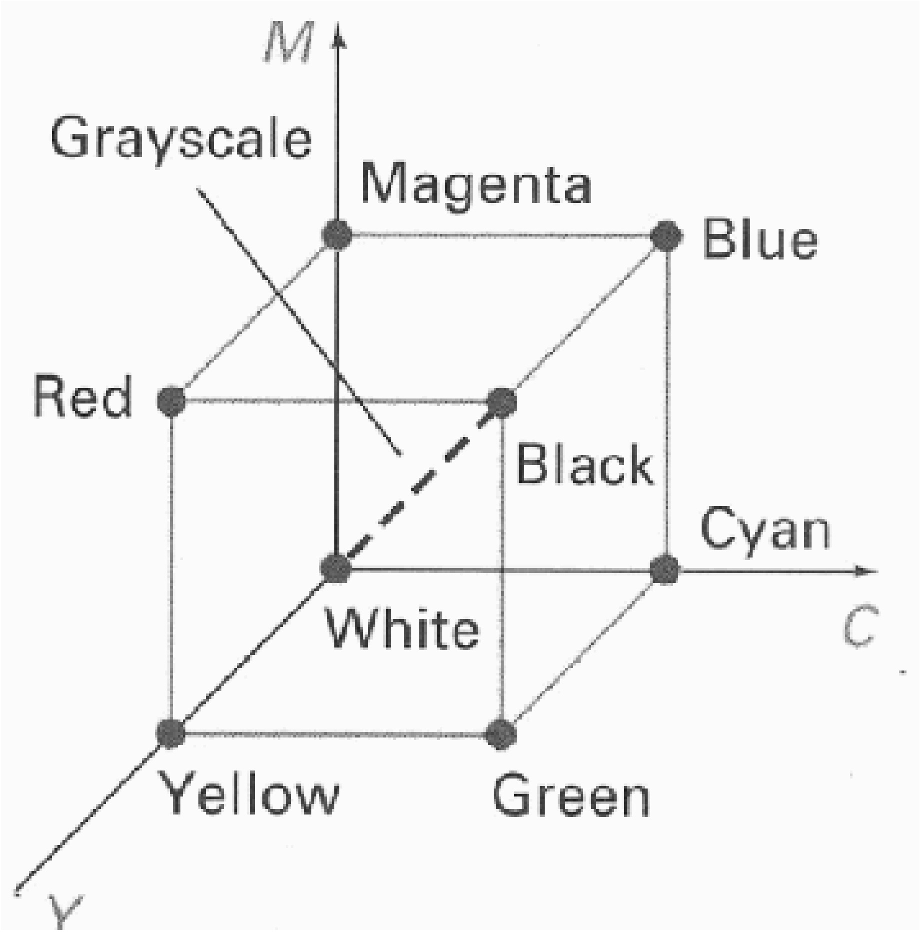
To create a three-dimensional representation of a color space, we can assign the amount of **blue** color to the representation's X axis, the amount of **green** to its Y axis, and the amount of **red** to its Z axis. The resulting 3-D space provides a unique position for every possible color that can be created by combining those three pigments.



A color is a point in a 3D cube with components: red, green e blue.



Color Model: CMY



The greyscale spectrum lies on the line joining the black and white vertices

Color space:

HSI Model (Hue, saturation, Intensity)

- **Hue** (tinta): (*tonalità, colore dominante così come viene percepito dall'osservatore*); describes the color itself
- **Saturation**: (*saturazione, purezza del colore o quantità di luce bianca miscelata con una certa tinta*); signals how much the color is polluted with white color
- **Intensity**: *(luminanza)*,

relative lightness or darkness

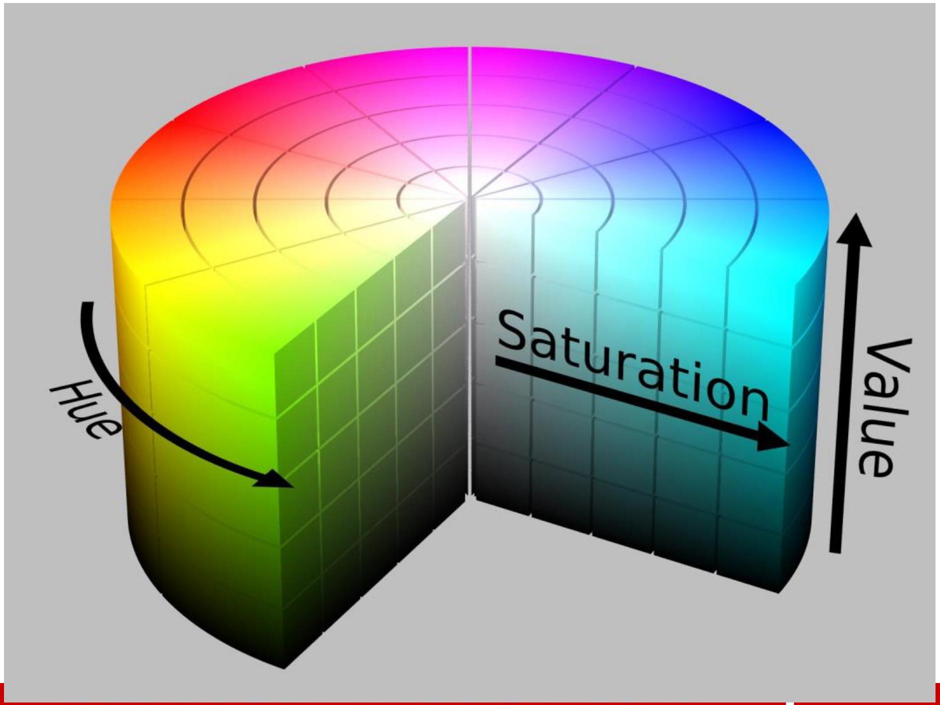
of a particular color

range is between [0,1]

0 means black,

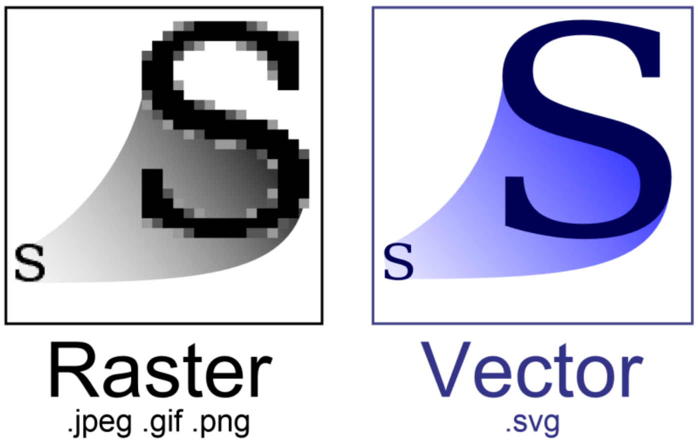
1 means white

Hue + saturation = Chromaticity



RASTER and VECTOR IMAGES

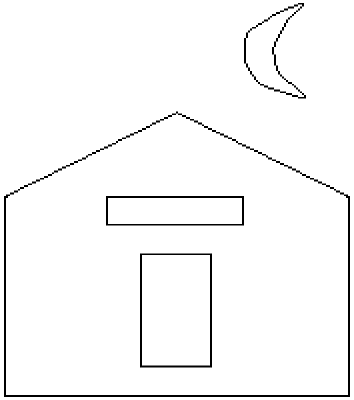
- **RASTER IMAGE** represented by a pixel grid. A bitmap file contains pixel values (.bmp, .gif);
- **VECTOR IMAGE** represented by a set of instructions (lines, circles, arcs, box,..), a vector image file contains commands and data. (.dxf,.svg)



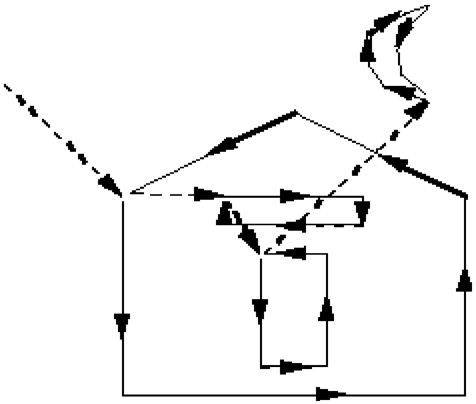
Each representation is used for different applications.

Scanners, printers (raster devices) plotters (vector device)

- Vector Image

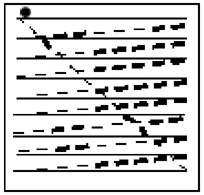


Ideal Drawing

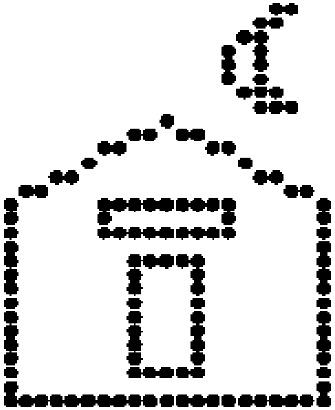


Vector Drawing

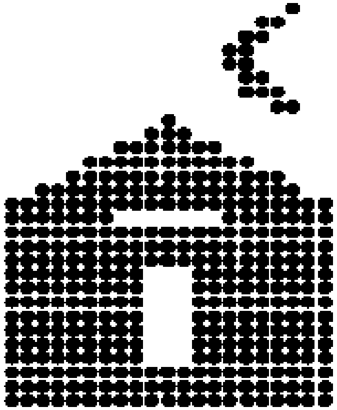
- Raster Image



Raster



Outline primitives



Filled primitives

Raster Graphics

- photos and paints
- pro: effects similar to those of traditional painting and graphics (brush, airbrush, pencil, charcoal).
- versus:
  - SCALING zoooming produces ‘pixelization’ effects
  - EDITING to (move, modify, cancel) part of an image select pixels and move, context independent
  - ROTATION is an approximation

Vector Graphics

- Commands (points, lines, arcs, Bézier curves) to describe objects, needs a language (PostScript is well known)
- Can contain bitmaps inside
- The printer receives the image description (e.g. in Postscript) and the image is printed using the best resolutions
- Versus:
  - No smooth colors
- Pro:
  - SCALING good quality
  - OVERLAPPING objects

Scalable Vector Graphics (.svg)

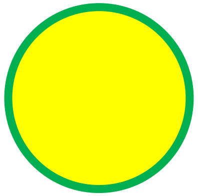
SVG specification is an open standard defined in XML text file and used to define graphics for the Web

SVG is a language for describing 2D graphics in XML

The HTML <svg> element (introduced in HTML5) is a container for SVG graphics.

Represent a yellow circle

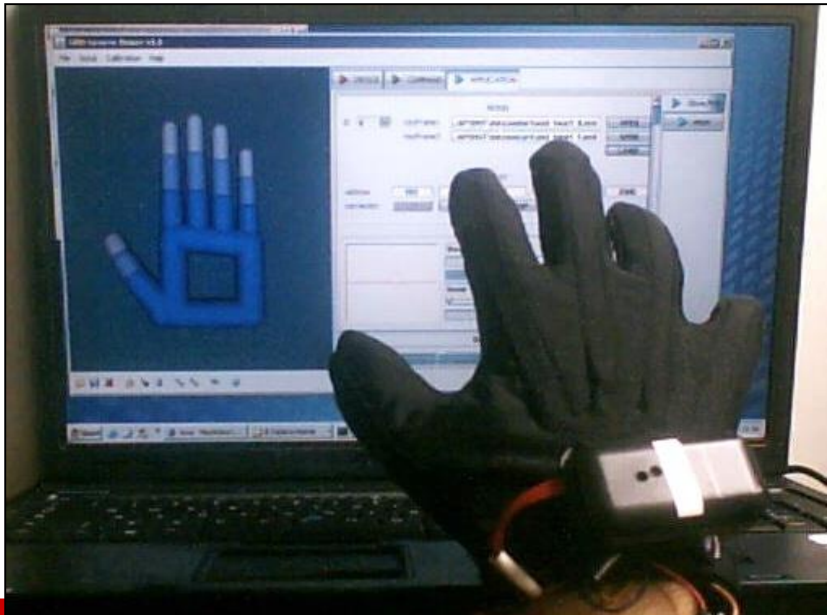
```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<svg width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />
</svg>
```

Input 3D devices

- 3D scanner , digitizers, spaceball, glove, tablet, pen..
- Capture a 3D location (x,y,z)
- Capture other properties like (e.g. color)
- Control switch/menu by actions (gesture recognition)

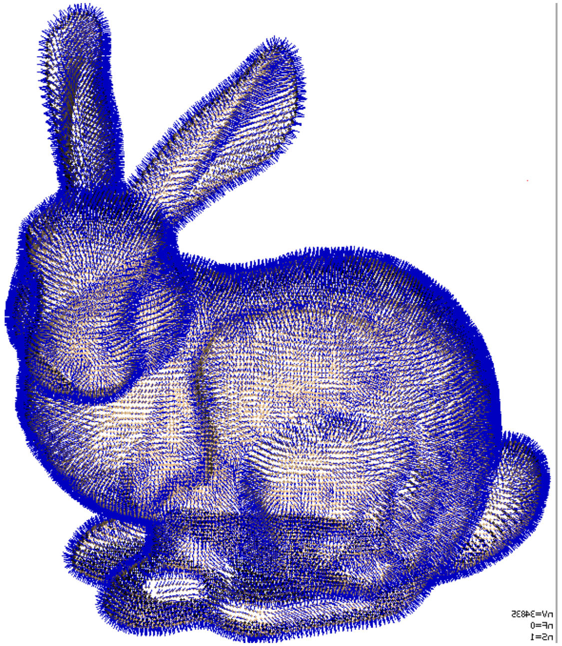


3D Scanner

Acquisition of a 3D point cloud on the surface of a physical object (relative position and color),

Software reconstruction of a virtual digital model of the real object.

Physical Model



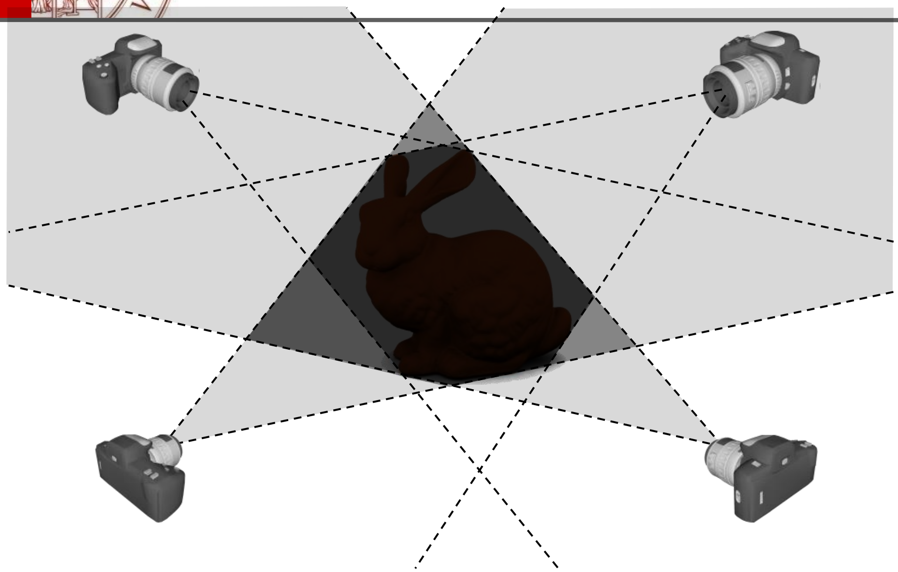
Virtual model

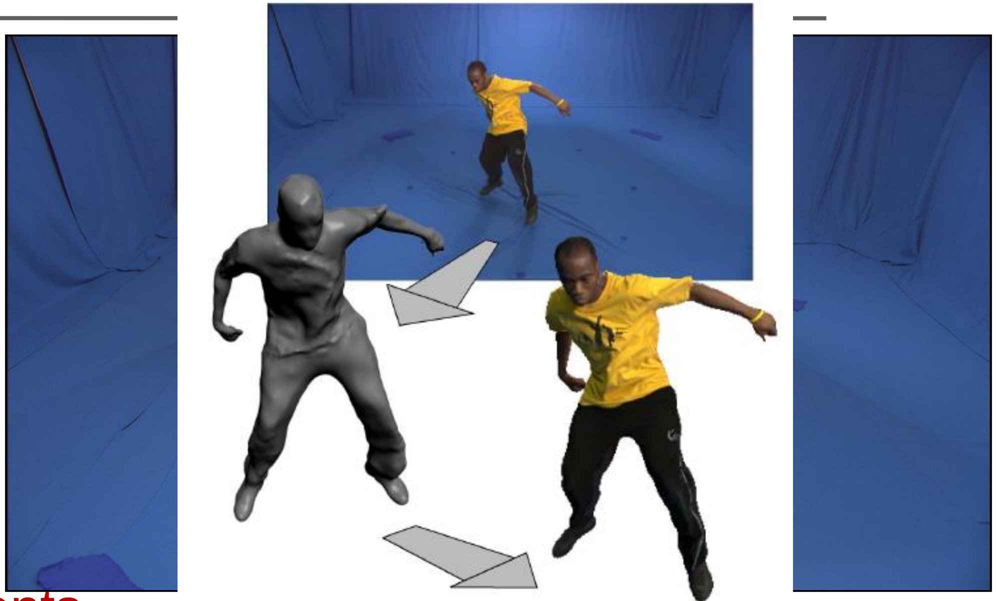
Taxonomy of 3D Scanning: Direct Measurements

Contact — Direct Measurements
(**digitizer**, coordinate measuring machines (CMM), AFM)

Non-Contact
- Passive
- Active

Taxonomy of 3D Scanning:
Shape-from-Silhouettes



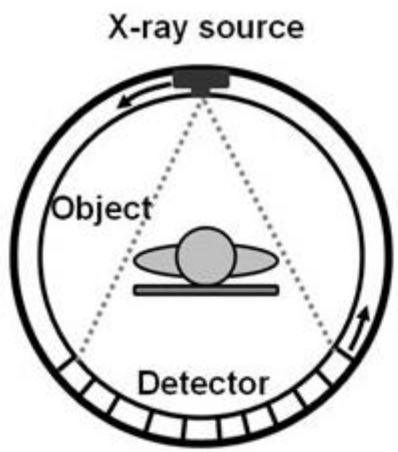


Contact —— Direct Measurements
(digitizer, coordinate measuring machines (CMM), AFM)

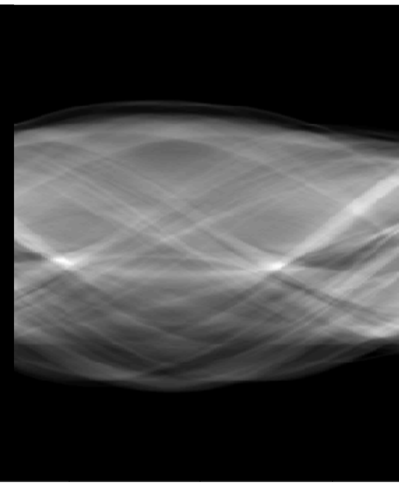
Non-Contact
- Passive — Shape-from-X
  (stereo/**multi-view**, silhouettes, motion)
- Active

J. Starck and A. Hilton. Surface Capture for Performance-Based Animation. *IEEE Computer Graphics and Applications*, 2007

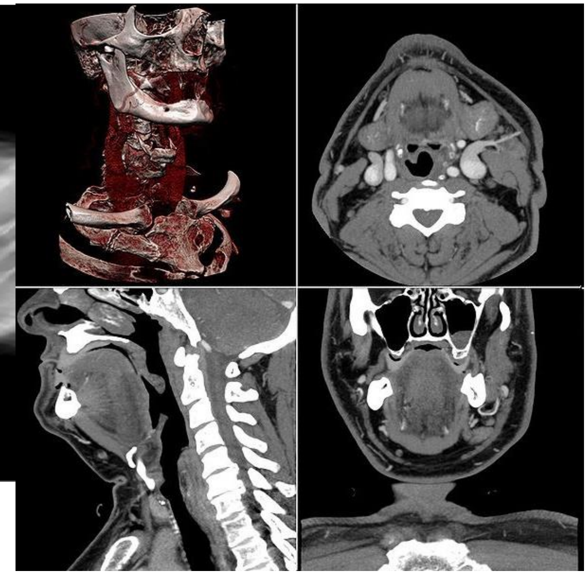
Taxonomy of 3D Scanning:
Computed Tomography (CT)



Density Function



Parallel/Fan-beam Projections



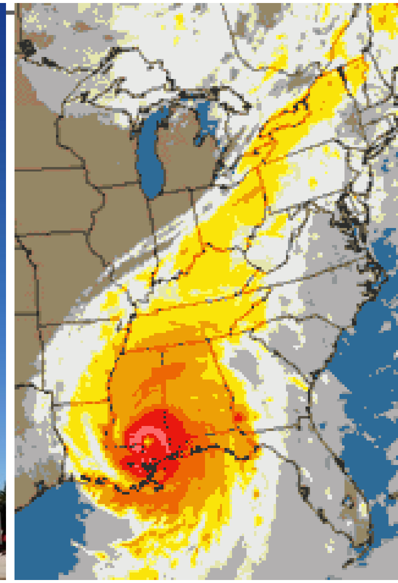
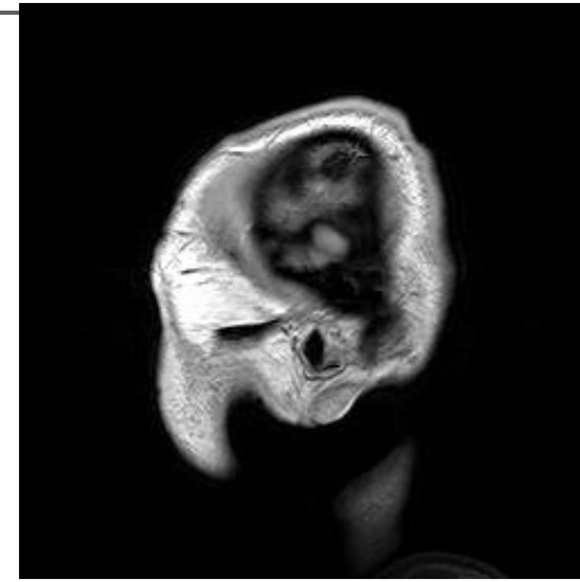
Contact — **Direct Measurements**
(digitizer, coordinate measuring machines (CMM), AFM)

Non-Contact
- Passive — **Shape-from-X**
  (stereo/multi-view, silhouettes, motion, texture, etc.)
- Active
  - Transmissive
    - **Computed Tomography (CT)**
    - **Transmissive Ultrasound**
  - Reflective

Taxonomy of 3D Scanning:
Non-optical Active Methods

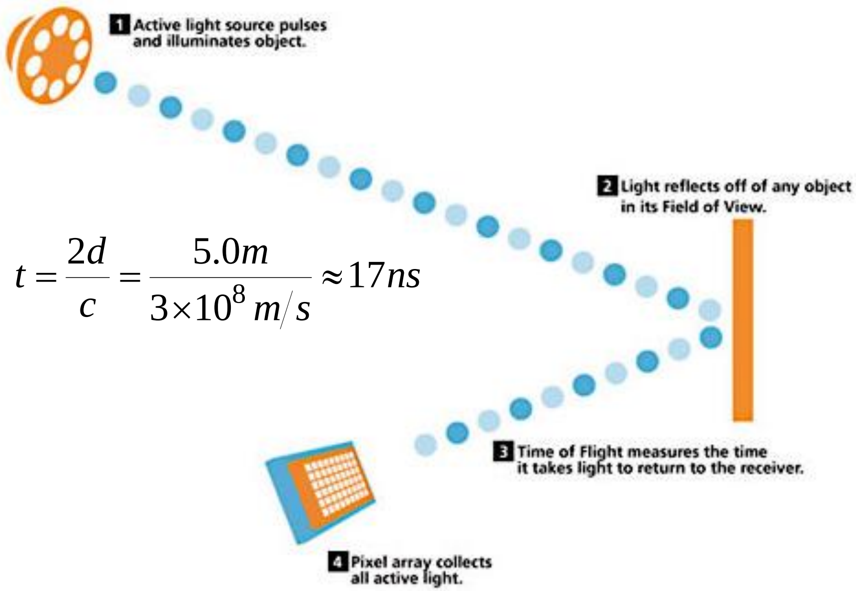




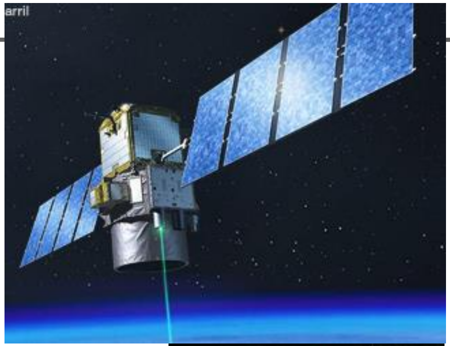



- Contact — Direct Measurements
  - (digitizer, coordinate measuring machines (CMM), AFM)
- Non-Contact
  - Passive — Shape-from-X
    - (stereo/multi-view, silhouettes, motion, texture, etc.)
  - Active
    - Transmissive
      - Computed Tomography (CT)
      - Transmissive Ultrasound
    - Reflective — Non-optical Methods
      - (**reflective ultrasound**, radar, sonar, **MR**

Taxonomy of 3D Scanning: Time-of-Flight



$$t = \frac{2d}{c} = \frac{5.0m}{3\times10^{8}\, m/s} \approx 17ns$$



Cloud Aerosol Lidar

Kinect device

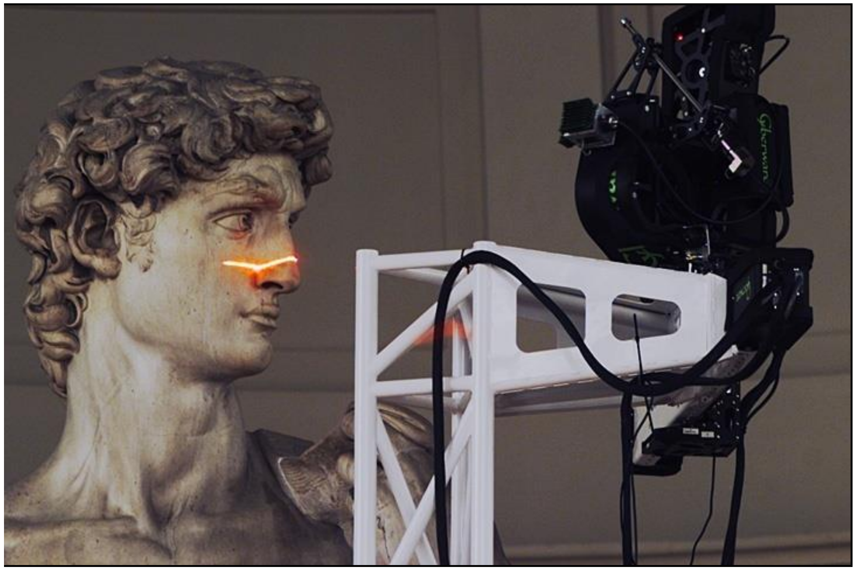
Contact — **Direct Measurements** (digitizer, coordinate measuring machines (CMM), AFM)

Non-Contact

- Passive — **Shape-from-X** (stereo/multi-view, silhouettes, motion)
- Active
  - Transmissive
    - **Computed Tomography (CT)**
    - **Transmissive Ultrasound**
  - Reflective
    - **Non-optical Methods** (reflective ultrasound, radar, sonar, MRI)
    - **Time-of-Flight**

Taxonomy of 3D Scanning:
Triangulation with Laser Striping

WholeBody WP4 CyberWare

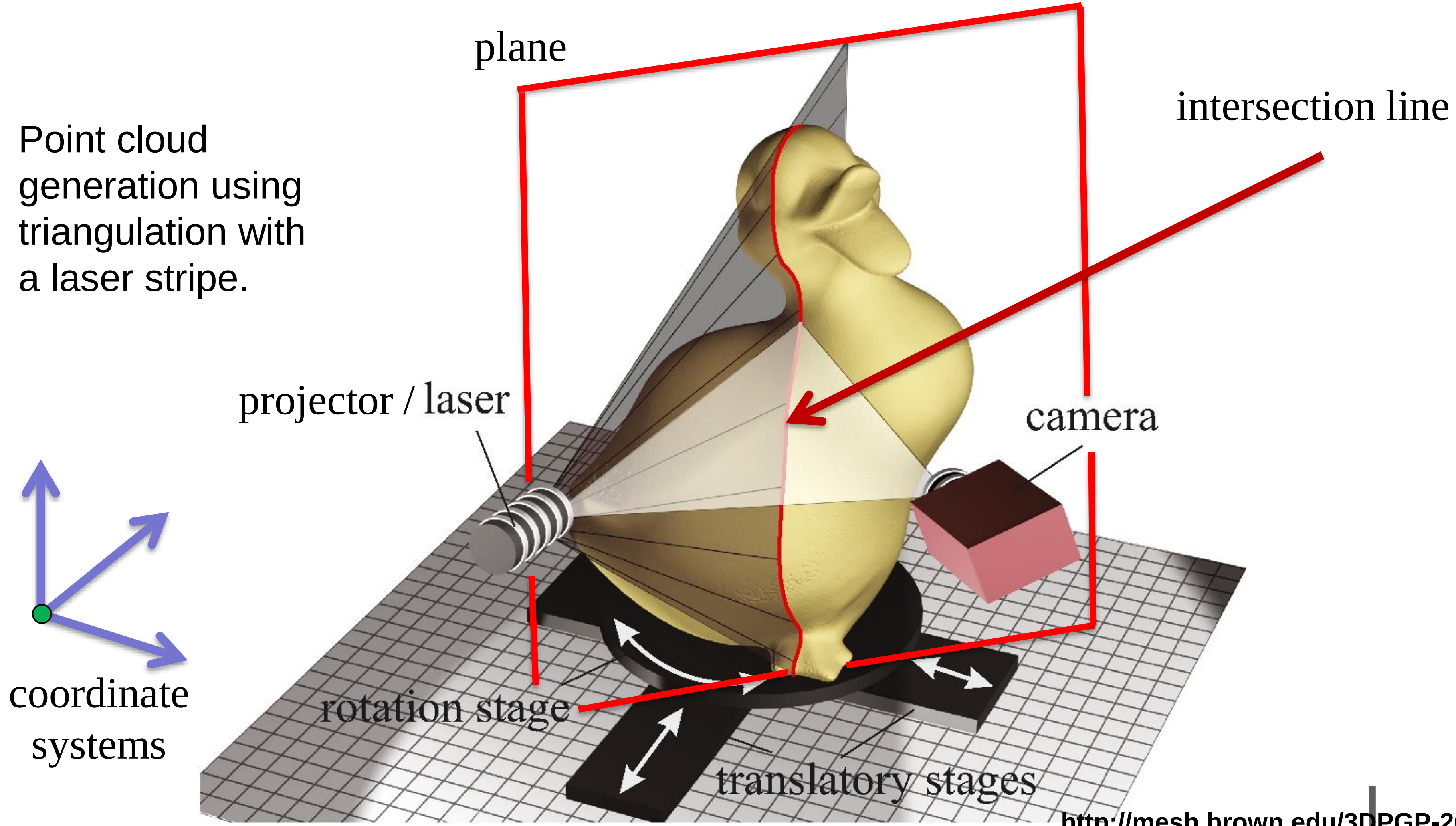


M. Levoy et al. The Digital Michelangelo Project: SIGGRAPH,2000

- Contact — Direct Measurements
  (digitizer, oordinate measuring machines (CMM), AFM)
- Non-Contact
  - Passive — Shape-from-X (stereo/multi-view, silhouettes, motion)
  - Active
    - Transmissive
      - Computed Tomography (CT)
      - Transmissive Ultrasound
    - Reflective
      - Non-optical Methods
      - Time-of-Flight
      - Triangulation
        (laser striping and structured lighting)

3D Triangulation

Point cloud generation using triangulation with a laser stripe.



http://mesh.brown.edu/3DPGP-2009/

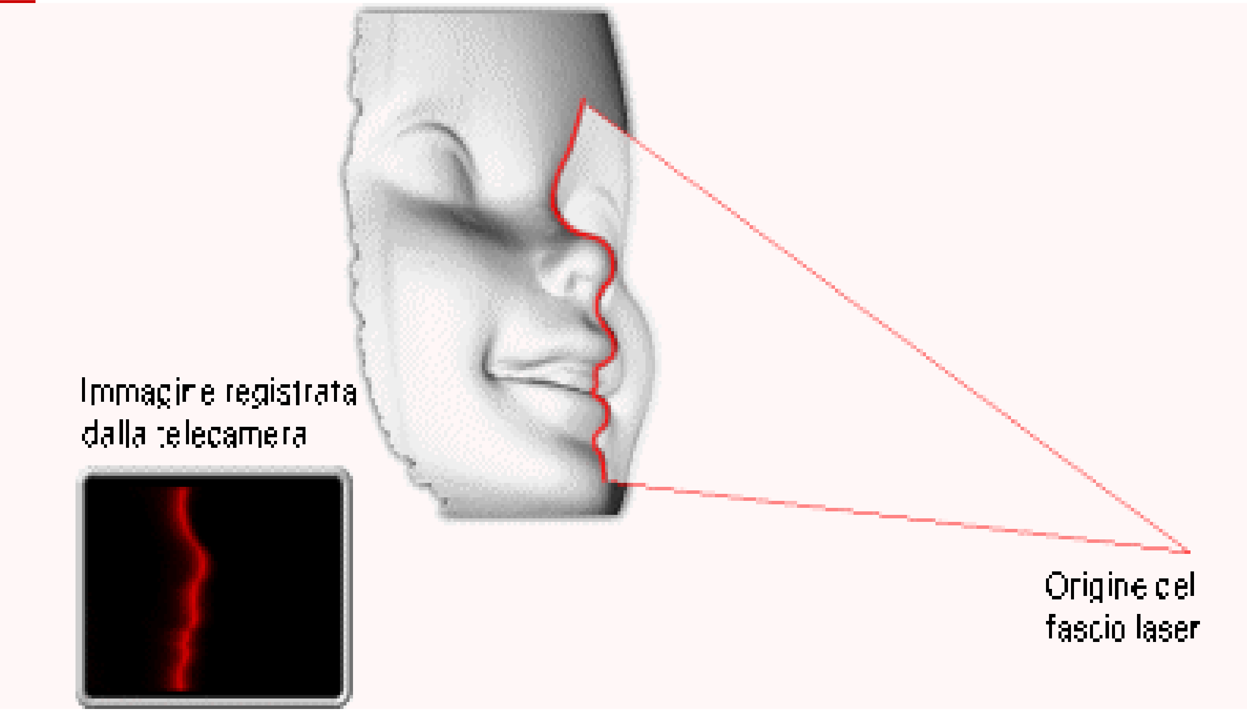
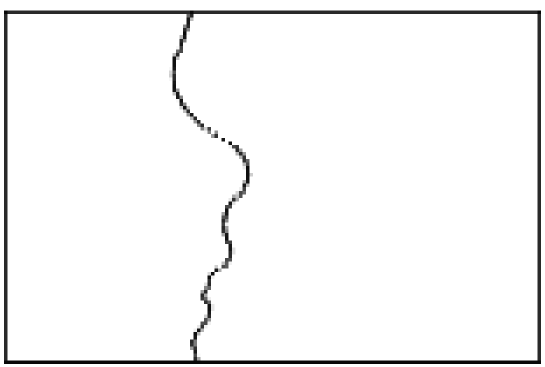
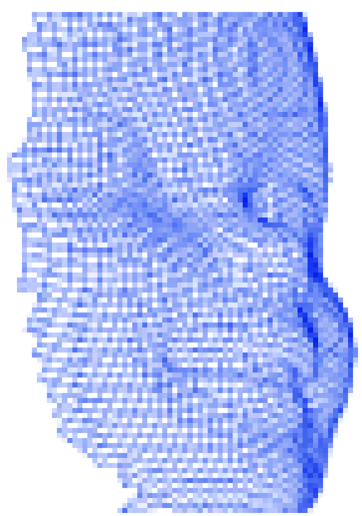
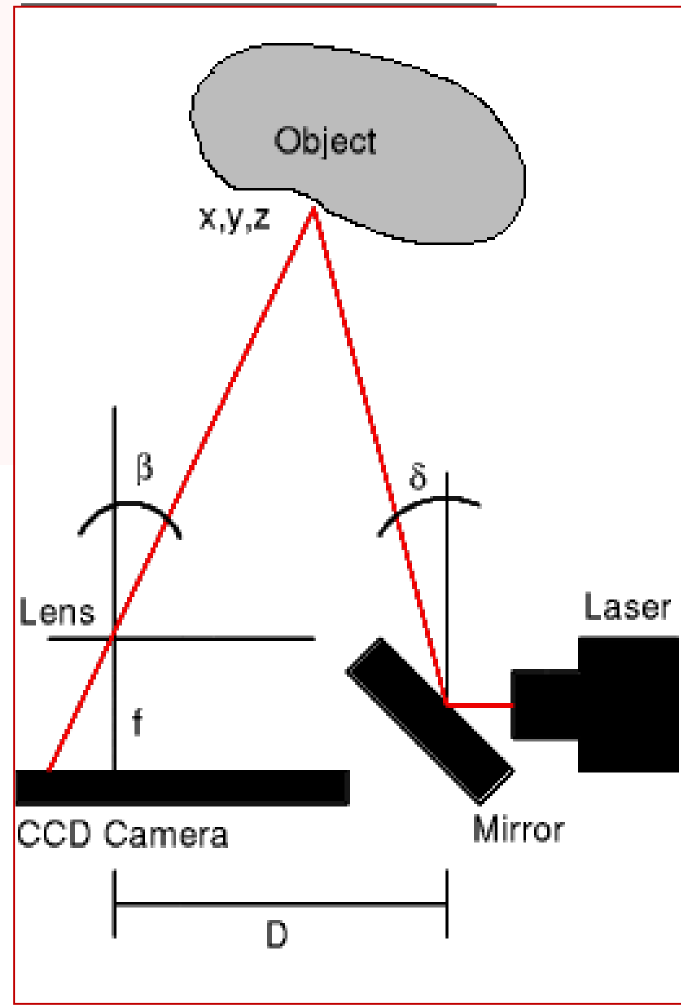


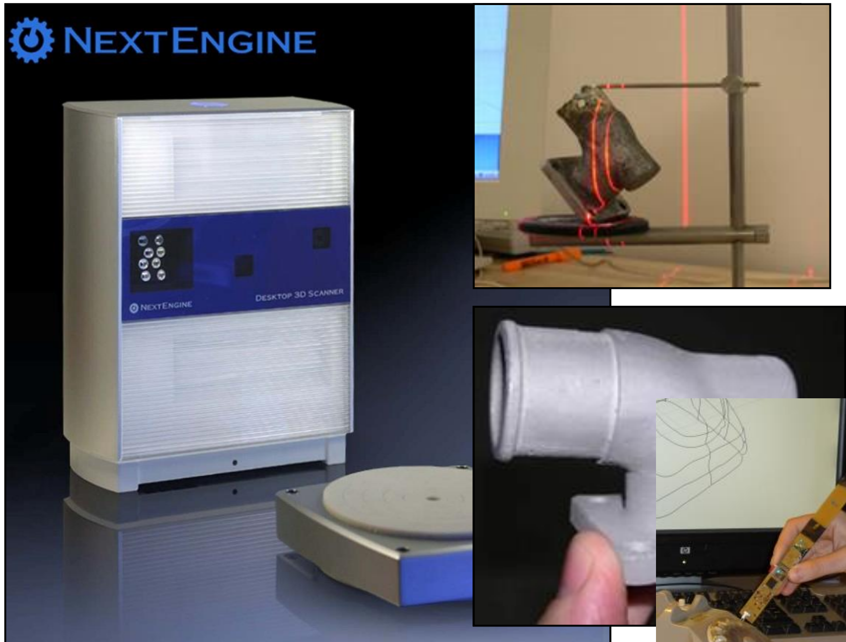
Immagine registrata
dalla telecamera
Origine del
fascio laser






Object
x,y,z
β
δ
Lens
Laser
f
CCD Camera
Mirror
D

High performance Graphics Lab
University of Bologna



Smart pen

Microscribe digitizer

Scanner laser 3D:
Minolta VIVID 900

Data glove

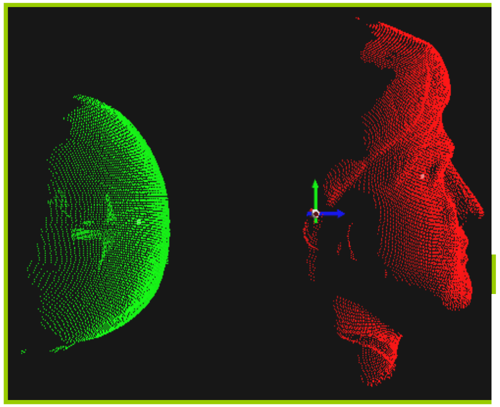
PIX-30 (Roland)

Features/Limitations

Accuracy, resolution, velocity of acquisition, reliability and cost
Cost varies (NextEngine ~$3,000 USD, others more expensive)

The 3D Scanning Pipeline

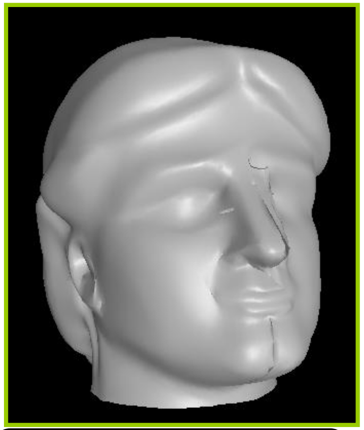
Scanner laser 3D:
Minolta VIVID 900



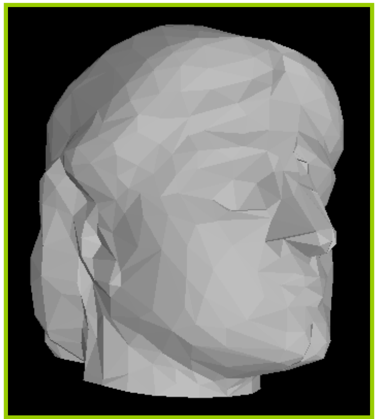
Multiple scans must be merge to cover the whole object surface

8 RANGE IMAGE
cloud of points
98503

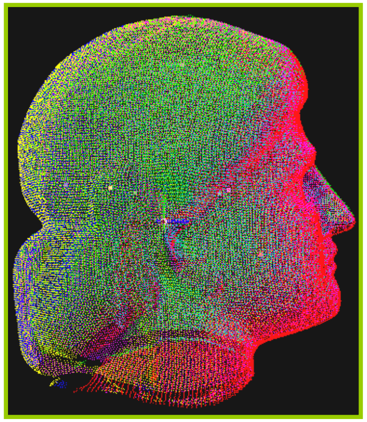
Data Capture → Align 3D Point Clouds → Reconstruct 3D Point Cloud(s) → Extract Isosurface (Polyhedral Mesh) → Surface Reconstruction



Surface Reconstruction

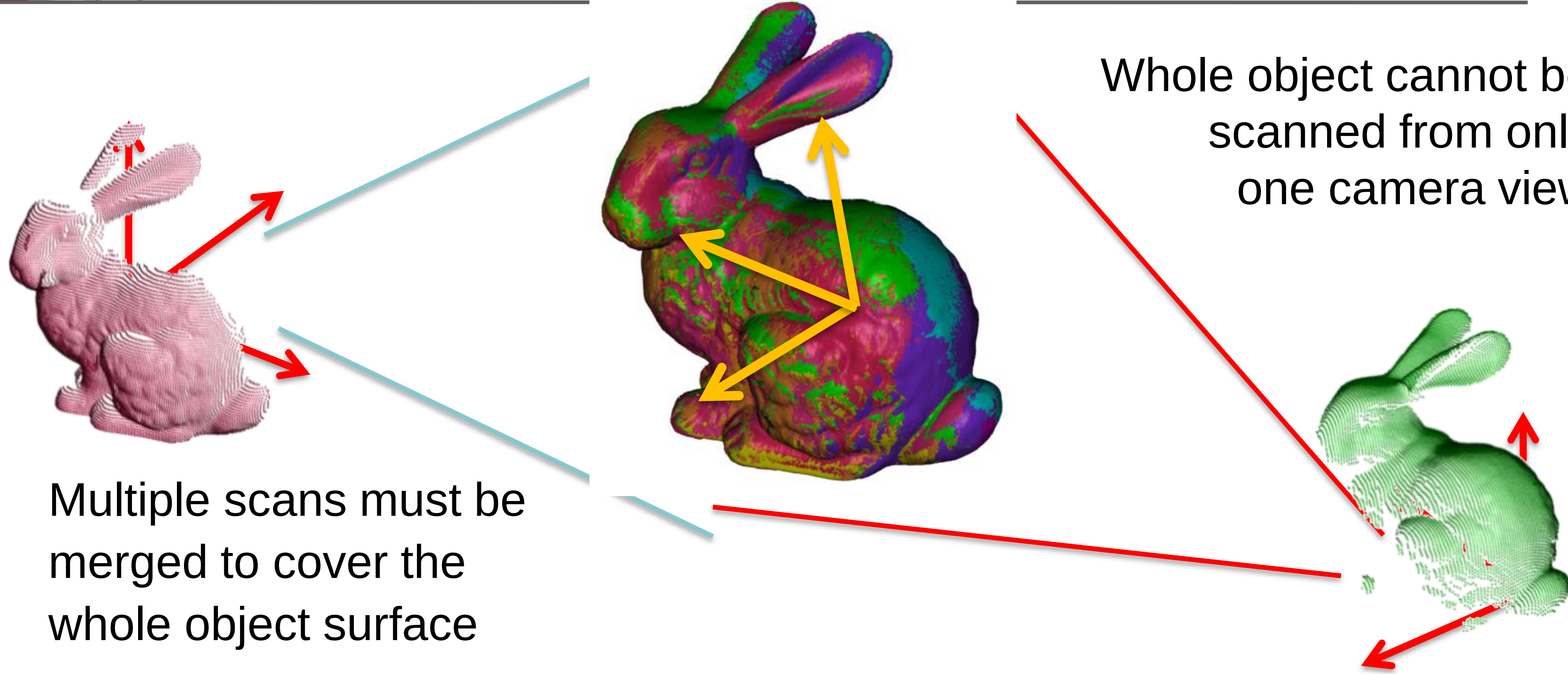


Extract Isosurface (Polyhedral Mesh)



Reconstruct 3D Point Cloud(s)

Merging Multiple Point Cloud Scans



Whole object cannot be scanned from only one camera view

Multiple scans must be merged to cover the whole object surface

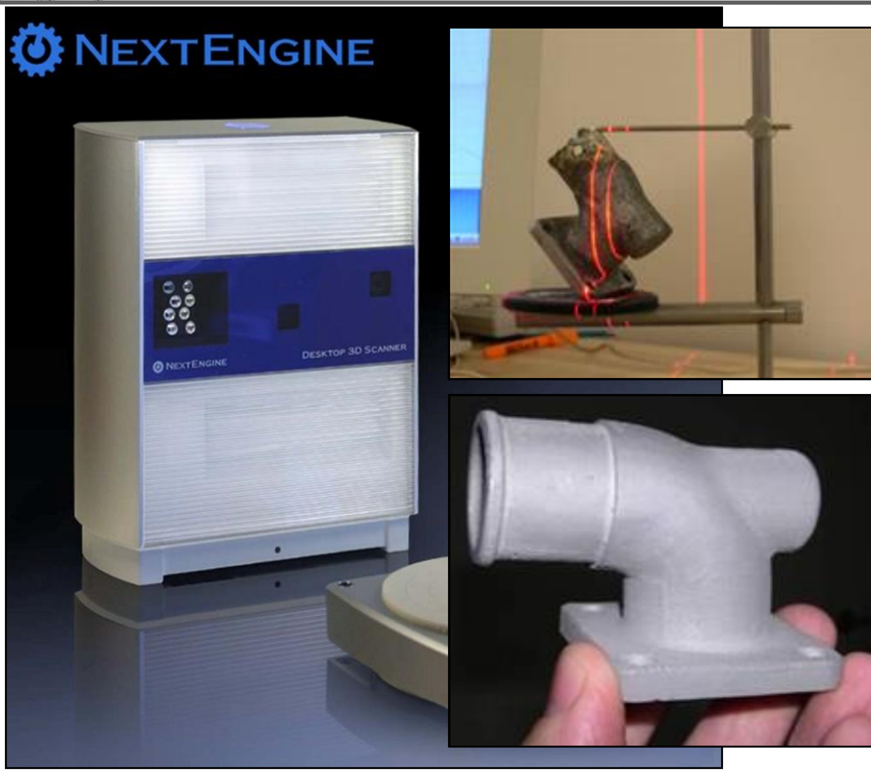
- Each point cloud scan is generated with respect to a **different local camera coordinate system**
- Relative position and orientation of each scan with respect to a **global coordinate system** must be determined to produce a single merged point cloud (Iterative Closest Point Algorithm (ICP))

Complex Models May Require 100s of Scans

http://www.research.ibm.com/pieta

Commercial 3D Scanners



In general the technical parameters to be taken into account in the evaluation of this kind of instruments are basically four:

accuracy (due to measurement error), maximum density sampling (**resolution**), **the velocity** and **reliability** of acquisition

Scanner 3D PIX-30 (Roland)

- tecnologia piezoelettrica
- volume di scansione in cm (X,Y,Z)=(30.5, 20.3, 6.0)
- passo di scansione in mm (X,Y,Z)=(0.05, 0.05, 0.025)
- può acquisire con precisione anche oggetti fragili e soffici
- velocità di scansione in mm/sec (X,Y,Z)=(30,30,9)

RASTER SCAN DISPLAY SYSTEM

Any graphics system with raster display contains three main components:

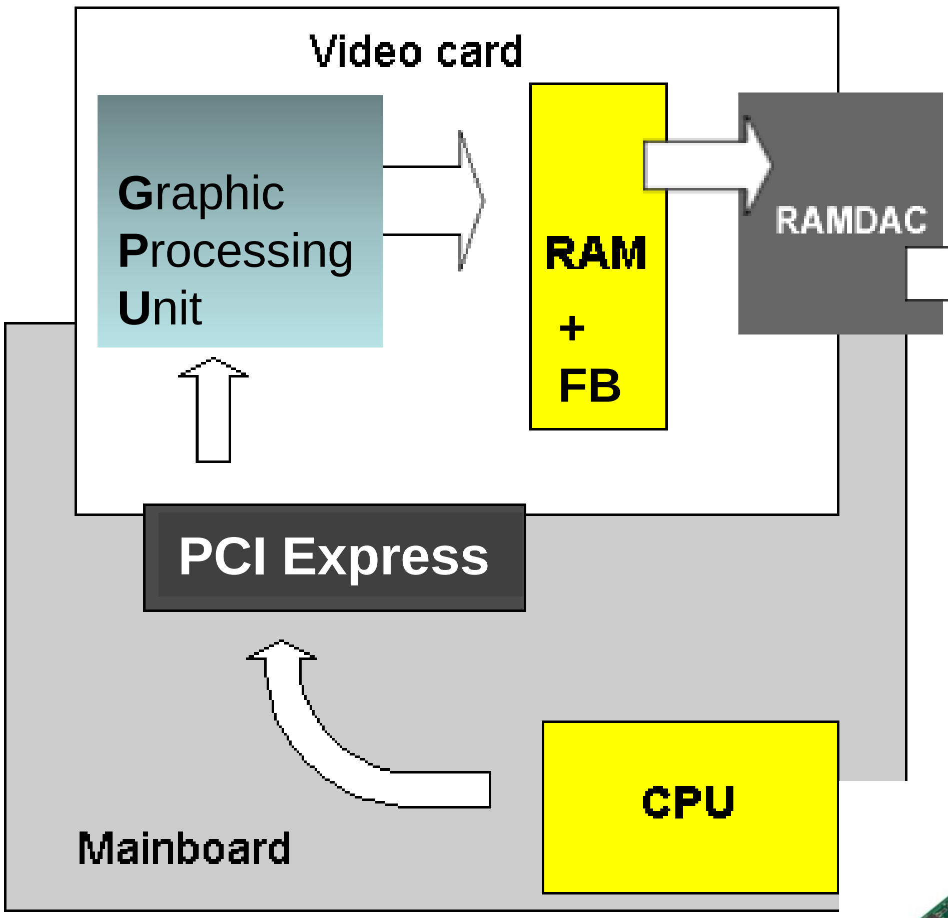
[1.] **Graphics card** (video card, *scheda grafica*) **G**raphics **P**rocessing **U**nit. Inside there is a special RAM memory called Frame Buffer

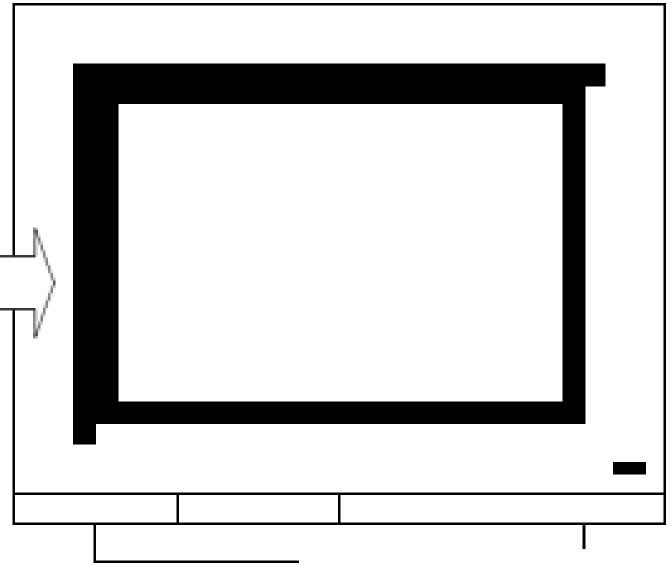
[2.] **MONITOR**

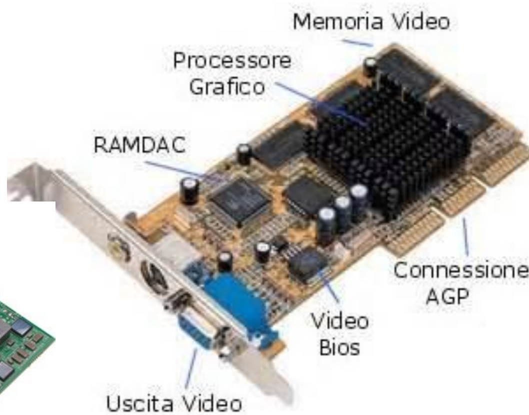
connected to the graphics card via cable

[3.] **device driver:** a program allowing operating systems or higher-level computer programs to interact with the video card

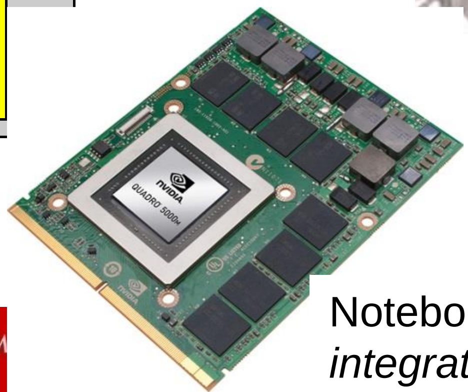
Video/Graphics card







Dedicated



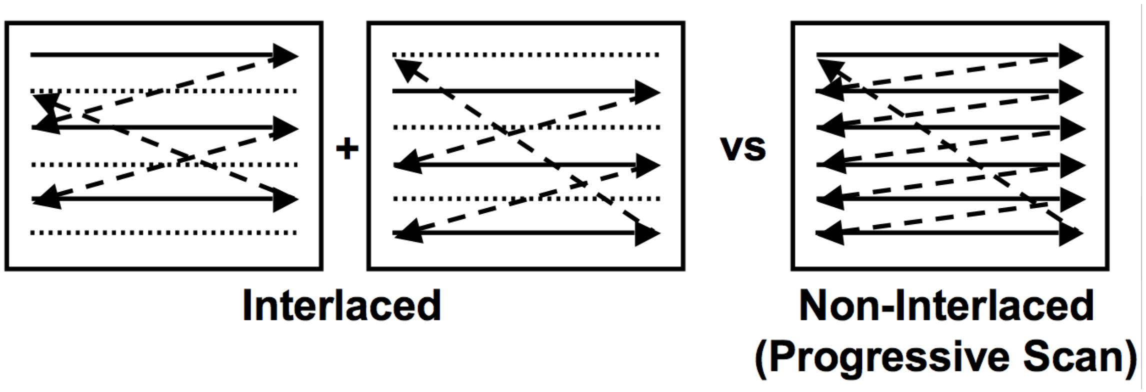
Notebook:
integrated graphics

Monitor Technology

- CRT, cathode ray tube
- LCD, liquid crystal display
- PDP, plasma display peripheral

Display scan

Refresh rate, or vertical frequency, measured in Hertz (Hz) represents the number of frames displayed on the screen per second. Too few, and the eye will notice the intervals in between and perceive a flickering display.
The world-wide accepted refresh rate for a flicker-free display is 70Hz and above (preferably 75 Hz or more).



Frame rate, also known as **frame frequency** and **frames per second** (**FPS**), is the frequency (rate) at which an imaging device produces unique consecutive images called frames. Frame rate standards in the TV and digital cinema business: 24fps, 25fps, and 30fps

Interlace scan (CRT/Plasma) is a technique for doubling the perceived frame rate of a video display without consuming extra bandwidth. The interlaced signal contains two fields of a video frame captured at two different times. One field contains all the odd lines in the image, the other contains all the even lines.

FRAME BUFFER (FB)

- FB is a RAM memory which stores images just before visualizing them to the display.

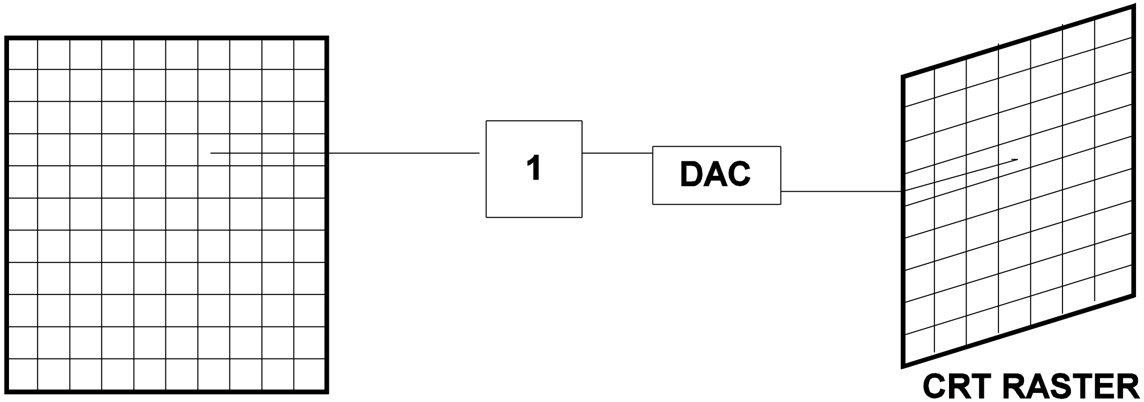
- It is a buffer collection - color, stencil (masks), accumulation (blend, add, combine images), double buffering, Z buffer (depth v_z/v_w), stereo –

- COLOR FRAME BUFFER:

  contains the color components for each pixel
- Its matrix shape reproduces the grid structure of the raster display; each element (x,y) contains info for the corresponding pixel at location (x,y) in the display

- ## Monochromatic Video

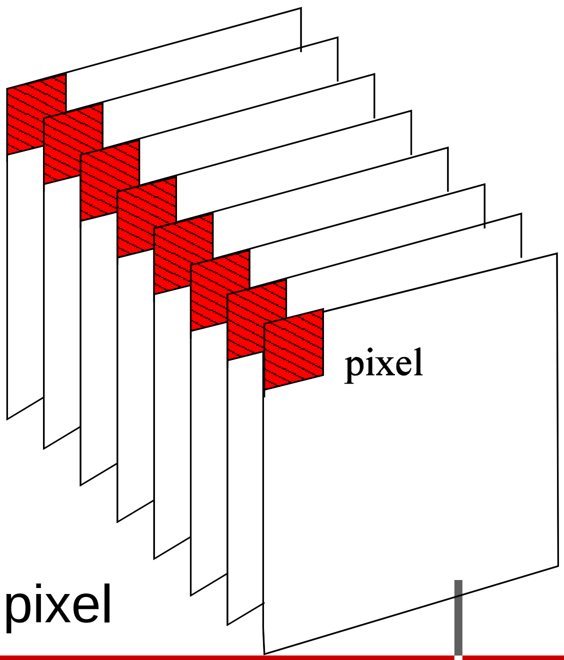
FB with a single plane (1 bit per pixel)



- ## Color Video

(true color model, colormap model)

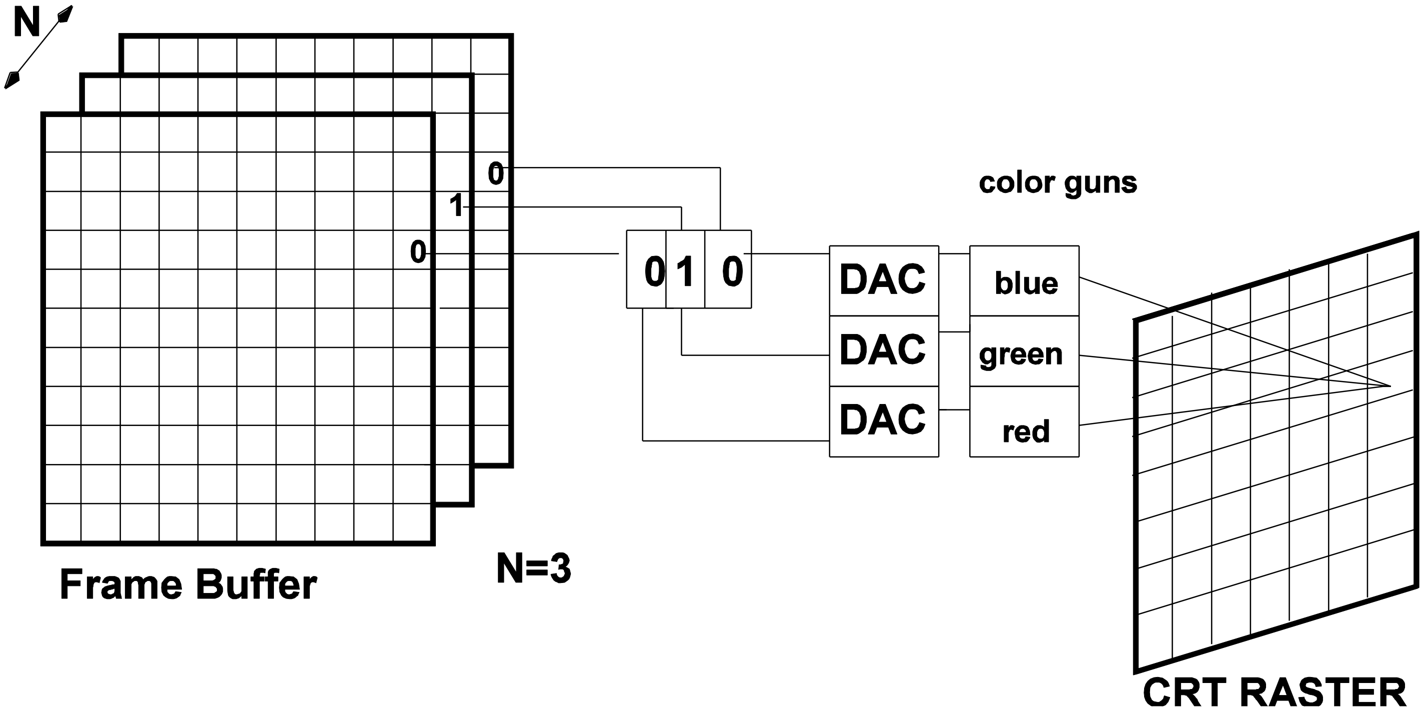
FB with N planes , that is N bit per pixel



E.g. Color FB with 8 planes: 256 different colors per pixel

True-color model

FB contains the color components for each pixel. Thus a FB with N planes stores 2^N different color values



3 planes for 8 different colors

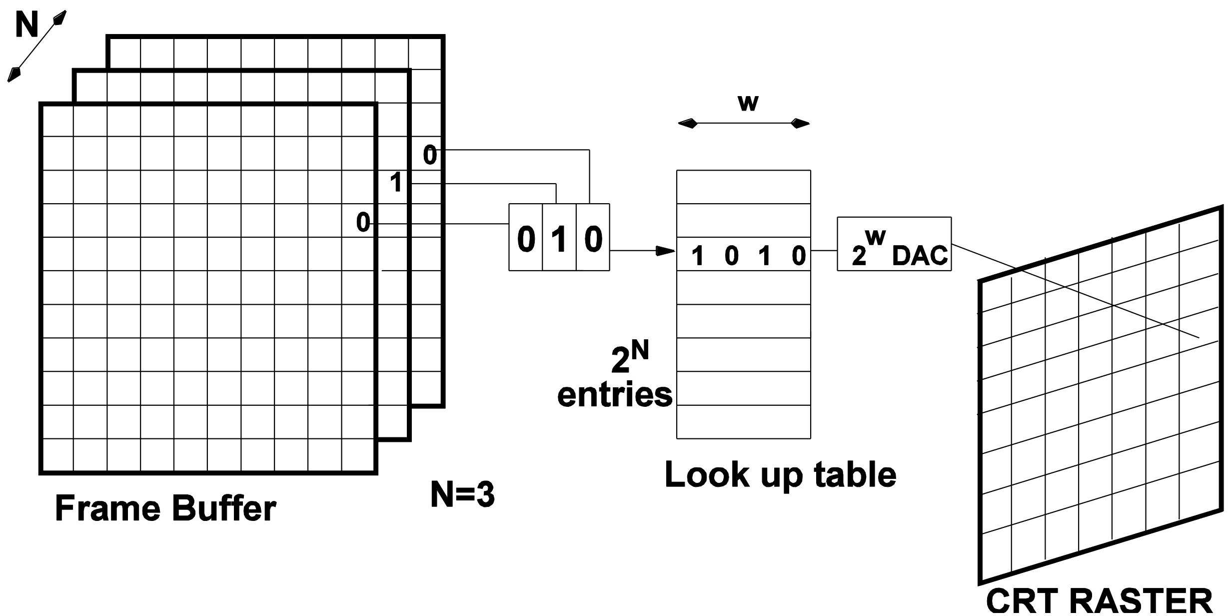
In general: 32 bit per pixel (8+8+8+8 RGBA)

Pseudocolor

FB contains the addresses to a color table, named *Look Up Table* o *colormap* which stores 2^N items each containing w bit.

Thus the colormap stores 2^w different color values, simultaneously, but only 2^N different colors are available on FB.

Different colormaps can be loaded



Example:

FB with 3 bit-planes ,

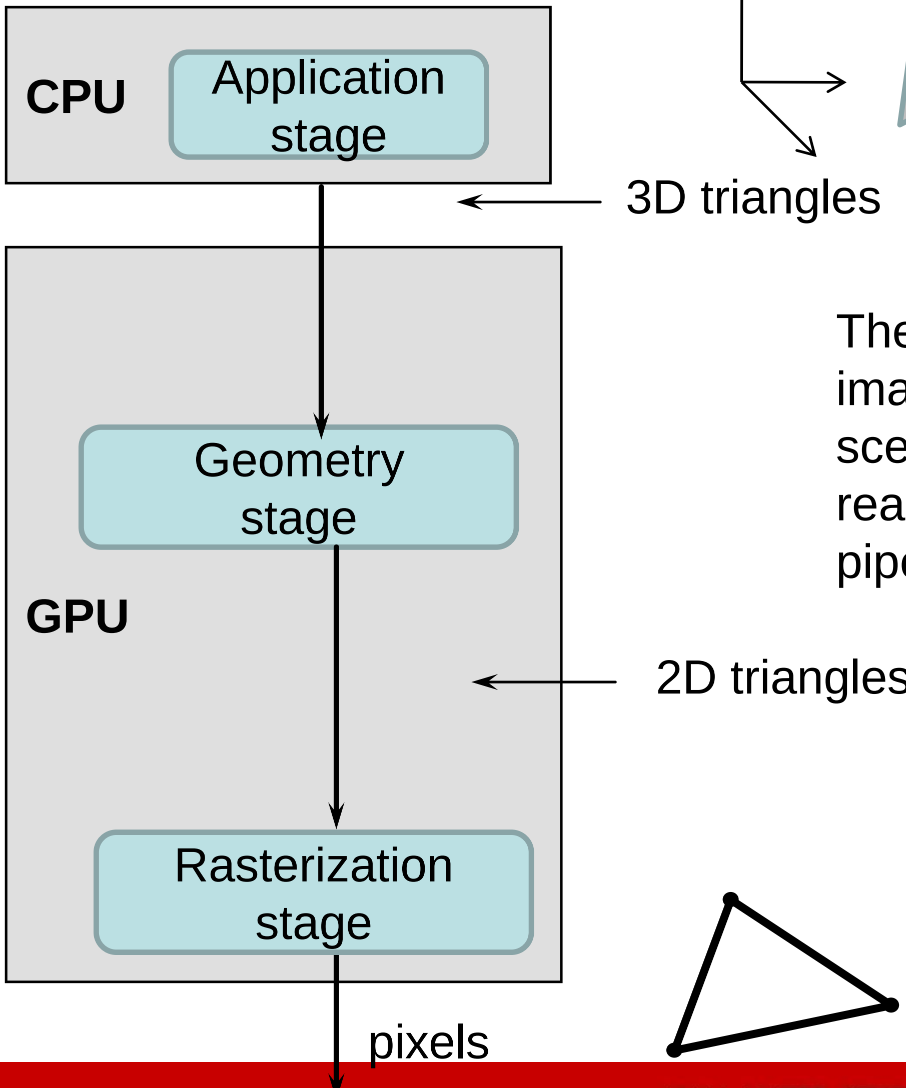
8 different colors pick up from a range of 16 available colors (2^4)

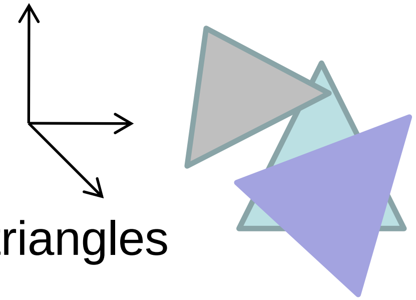
Real-time graphics

- **CPU**: general-purpose computer ('60)
- **VGA** (Video Graphic Array) hardware controller (DPU) (anni '80): special-purpose graphics system
- **Graphics Hardware Unit** ('90): pipeline graphics system (special-purpose VLSI circuits) Silicon Graphics International (SGI) and Evans Sutherland design expensive multichip.
- **GPU** (Graphics Processor Unit) (end'90): single chip GPU, cheaper, in PC and console for video game

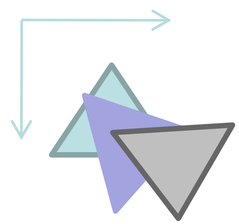
- So far, 5 GPU generations. Towards the offline rendering system.

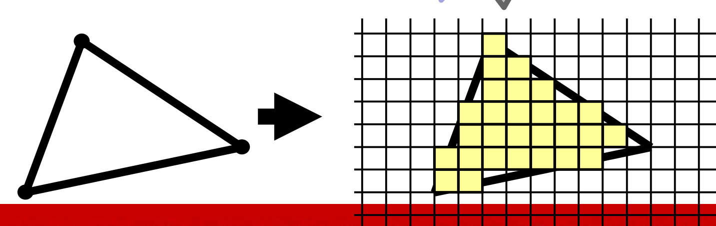
Rendering pipeline: a functional overview



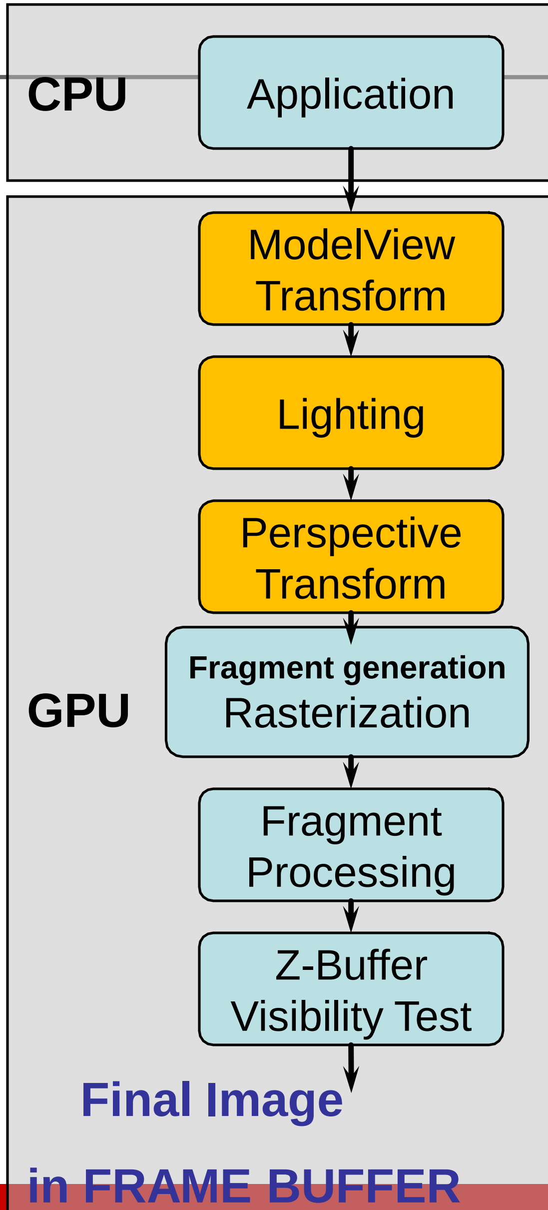


The task is to synthesize an image from a description of a scene—60 times per second for real-time graphics - by a pipeline of specialized stages.





Rendering Pipeline



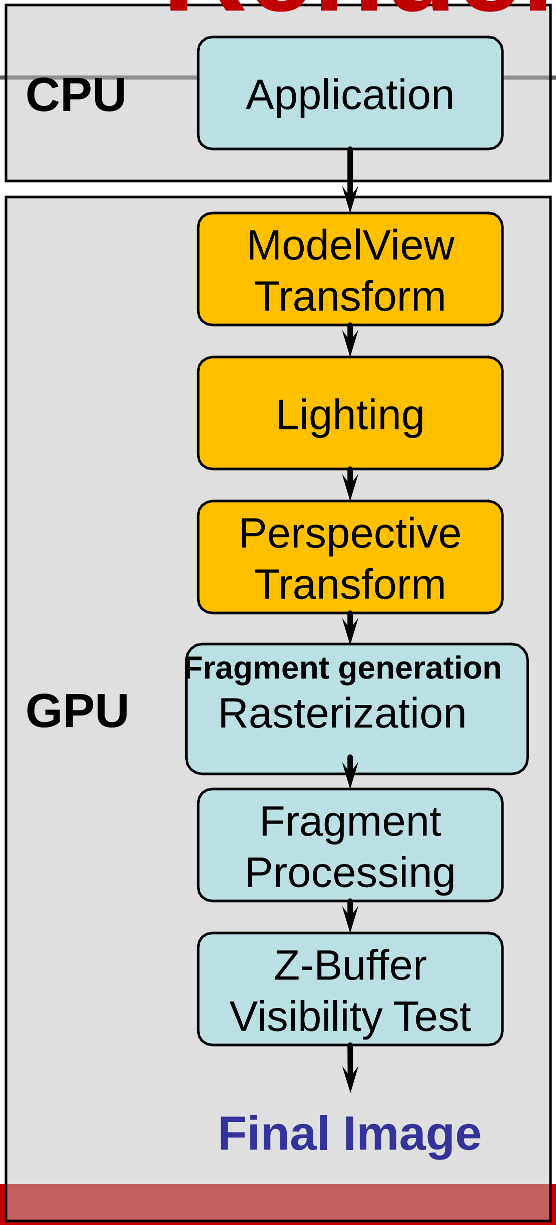
3D API commands (OpenGL o DIRECT3D)

← ***Vertices***

Most real-time graphics systems assume that everything is made of triangles, and they first carve up any more complex shapes, such as quadrilaterals or curved surface patches, into triangles.

The developer uses a computer graphics library (such as OpenGL or Direct3D) to provide each triangle to the graphics pipeline one vertex at a time.

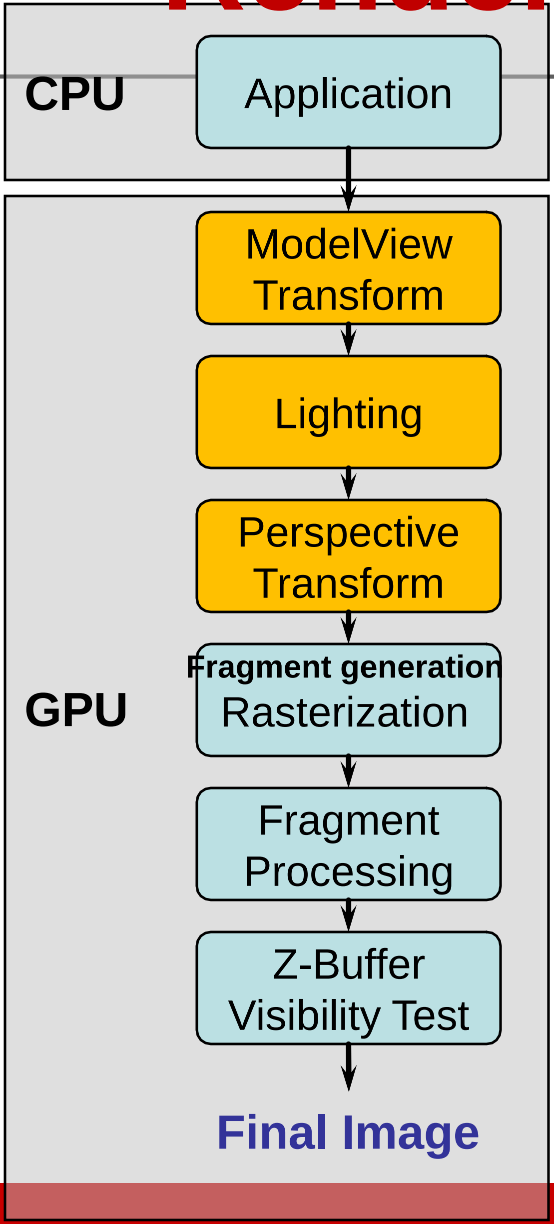
Rendering Pipeline



World Space Vertices

the GPU must first transform all objects into a common coordinate system.

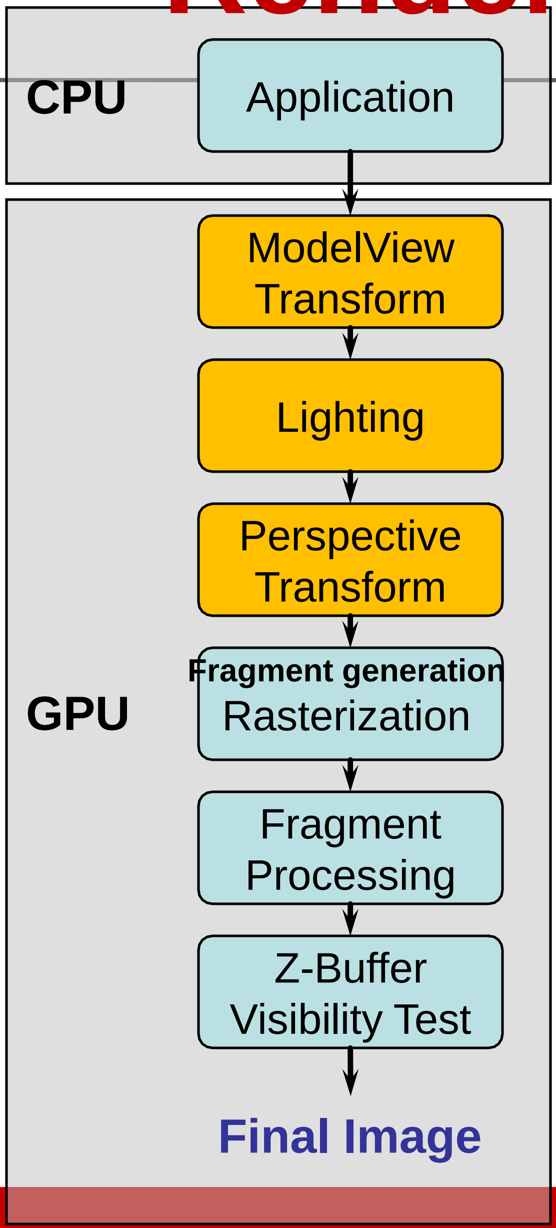
Rendering Pipeline



Lit, World Space Vertices

Once each triangle is in a global coordinate system, the GPU can compute its color based on the lights in the scene.

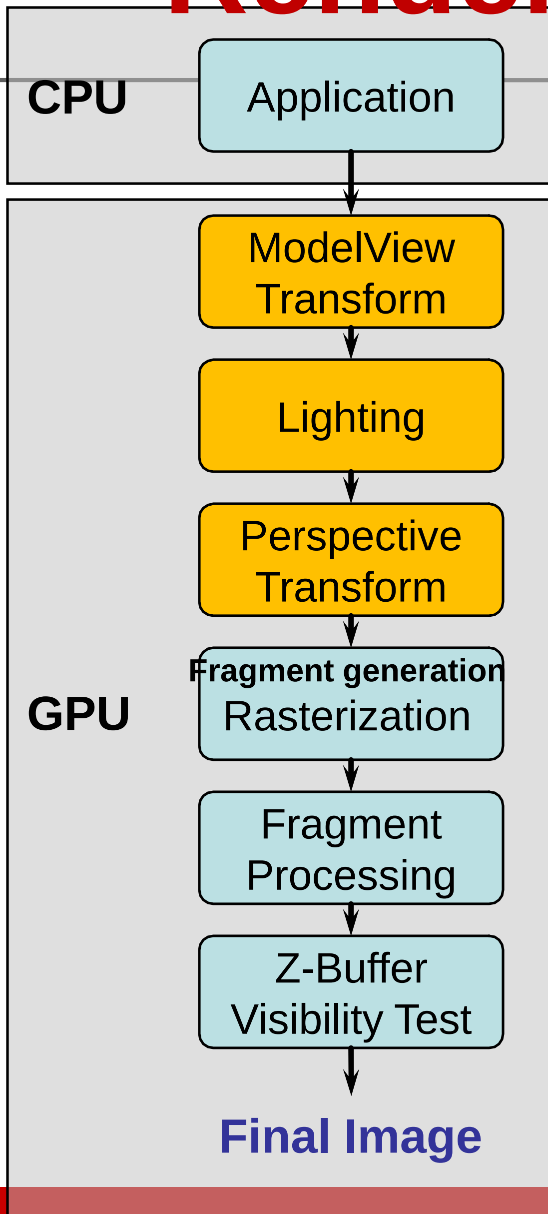
Rendering Pipeline

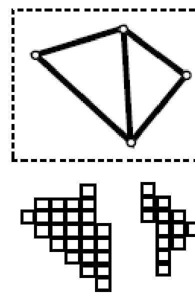


2D Projected Vertices

Next each colored 3D triangle is projected onto the virtual view plane.

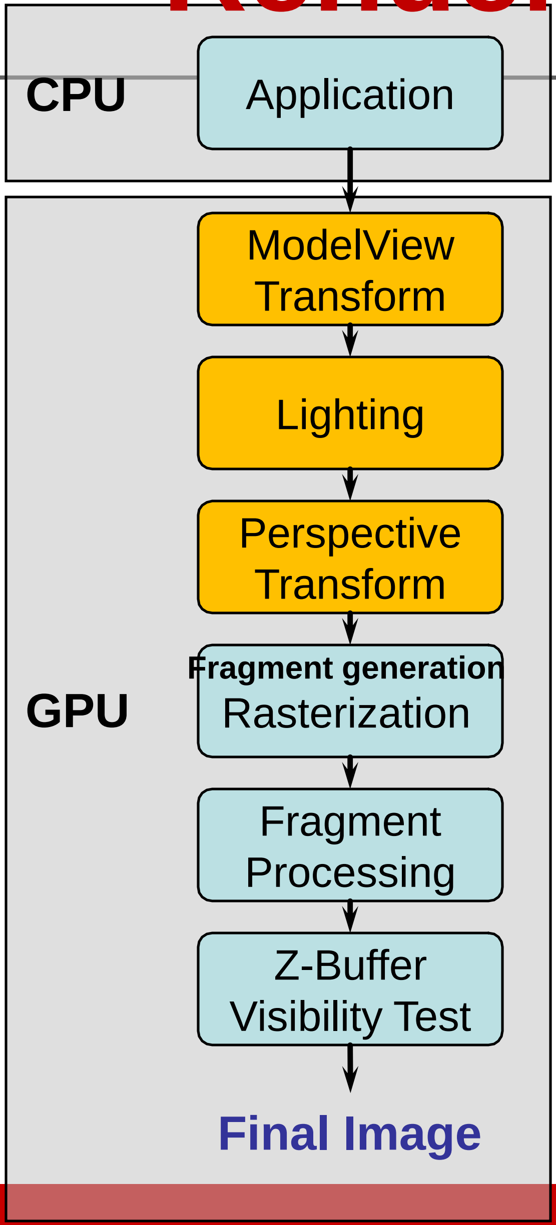
Rendering Pipeline



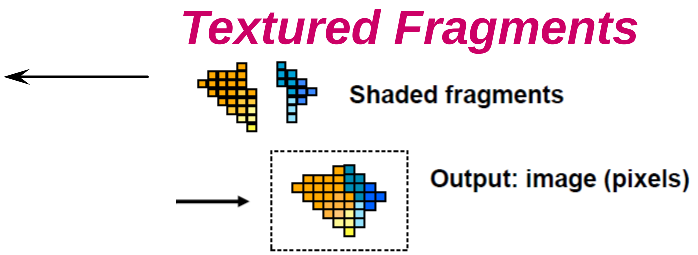


Each visible screen-space triangle overlaps some pixels on the display; determining these pixels is called rasterization

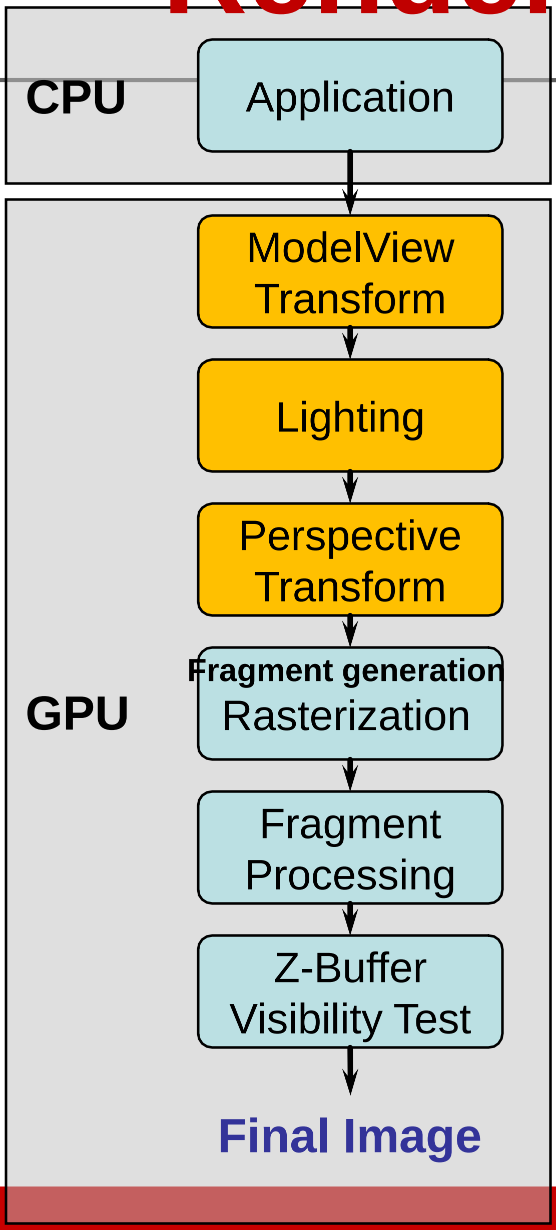
Rendering Pipeline



The actual color of each pixel can be taken directly from the lighting calculations,
but for added realism,
images called textures are often draped over the geometry to give the illusion of detail.



Rendering Pipeline



In most scenes, some objects obscure other objects.
All modern GPUs provide a depth buffer, a region of memory that stores the distance from each pixel to the viewer.

Textured, Visible Fragments

Output: image (pixels)

Graphic card generations

GPUs have evolved from a **hardwired implementation** of the graphics pipeline to a programmable computational substrate that can support it. Fixed-function units for transforming vertices and texturing pixels have been subsumed by a unified grid of processors, or **shaders**, that can perform these tasks and much more.

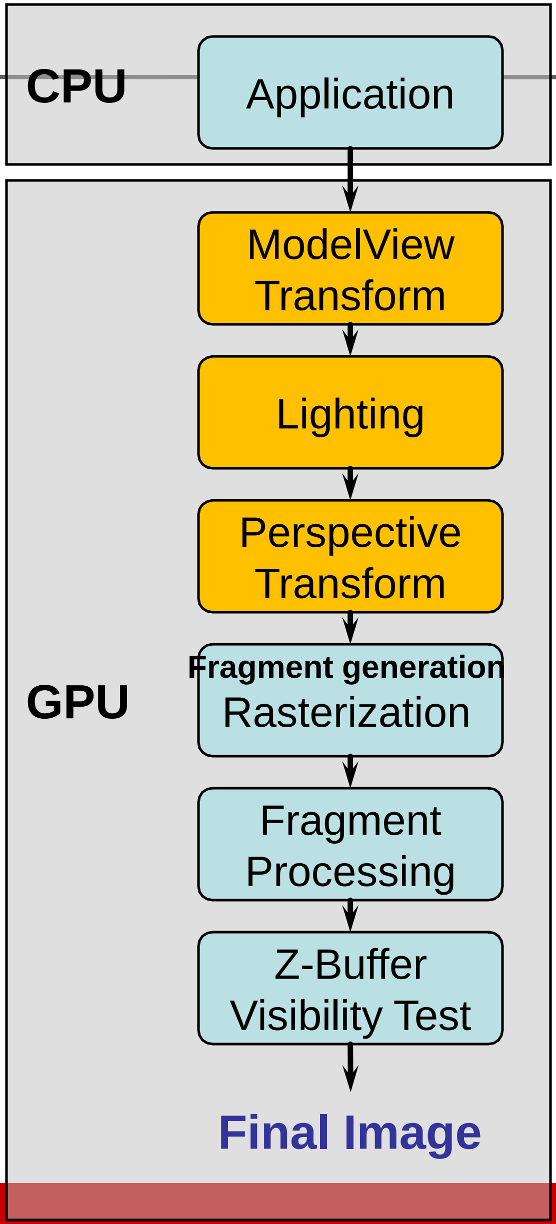
This evolution has taken place over several generations by gradually replacing individual pipeline stages with increasingly programmable units.

GPU with Fixed function pipeline

- I generation (->1998) single chip GPU. TNT (NVIDIA), RAGE (ATI), Voodoo3 (3dfx), vertex transformation on CPU, pixel processing in GPU, limited set of math operations on pixels.

- II generation (1999-2000) GeForce2 (NVIDIA), Radeon 7500 (ATI), Savage3D (S3), transform and lighting of vertices is done in hardware as well (uses the **fixed function pipeline**)

Fixed Function Pipeline



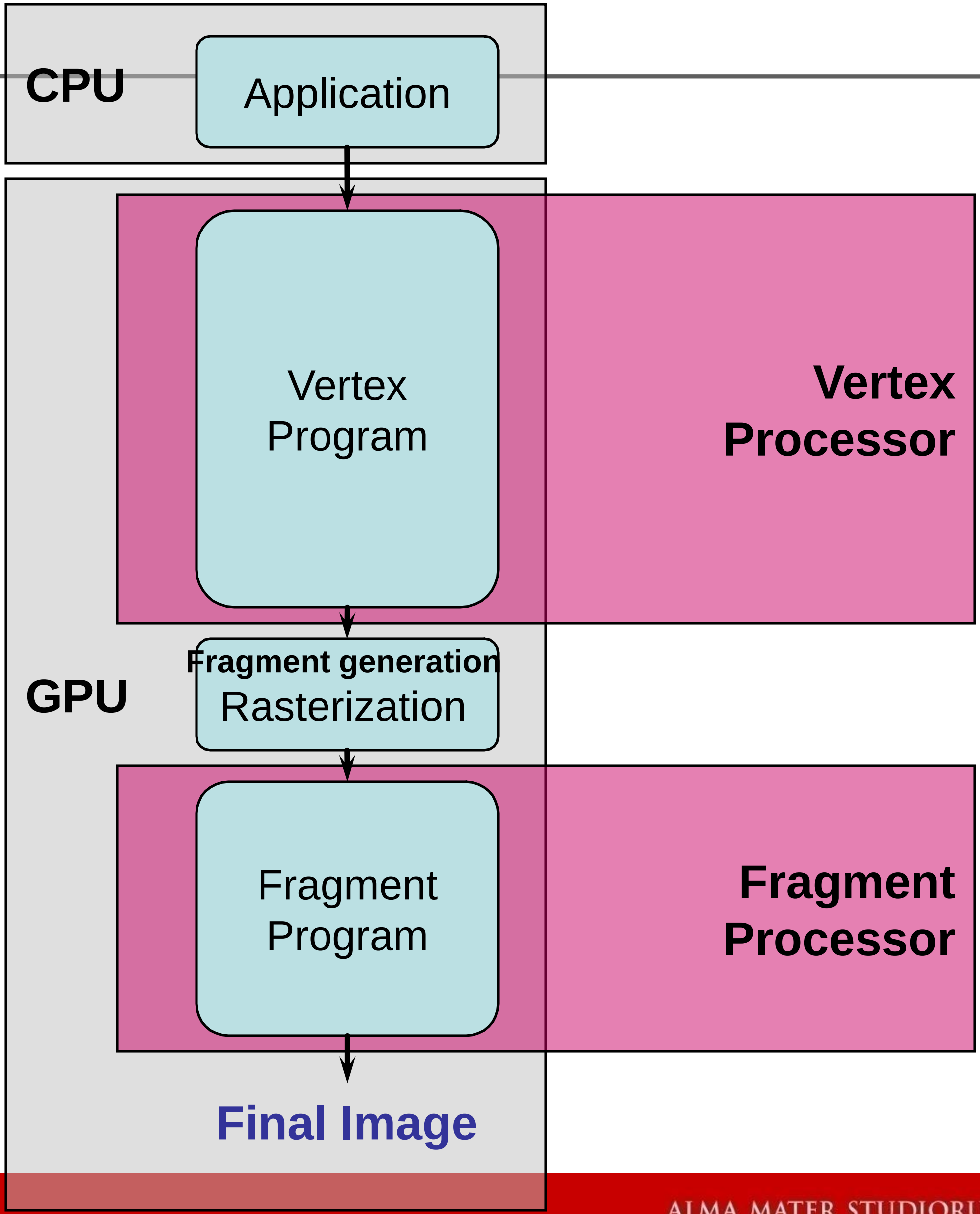
Hardwired implementation of the graphics pipeline

PRO:
improved performance

CONTRO:
the programmer had limited control over how the hardware created the final image. To do non-standard effects, like cartoon shading, required a lot of hackery.

off-line rendering
vs.
real-time rendering

Programmable Pipeline

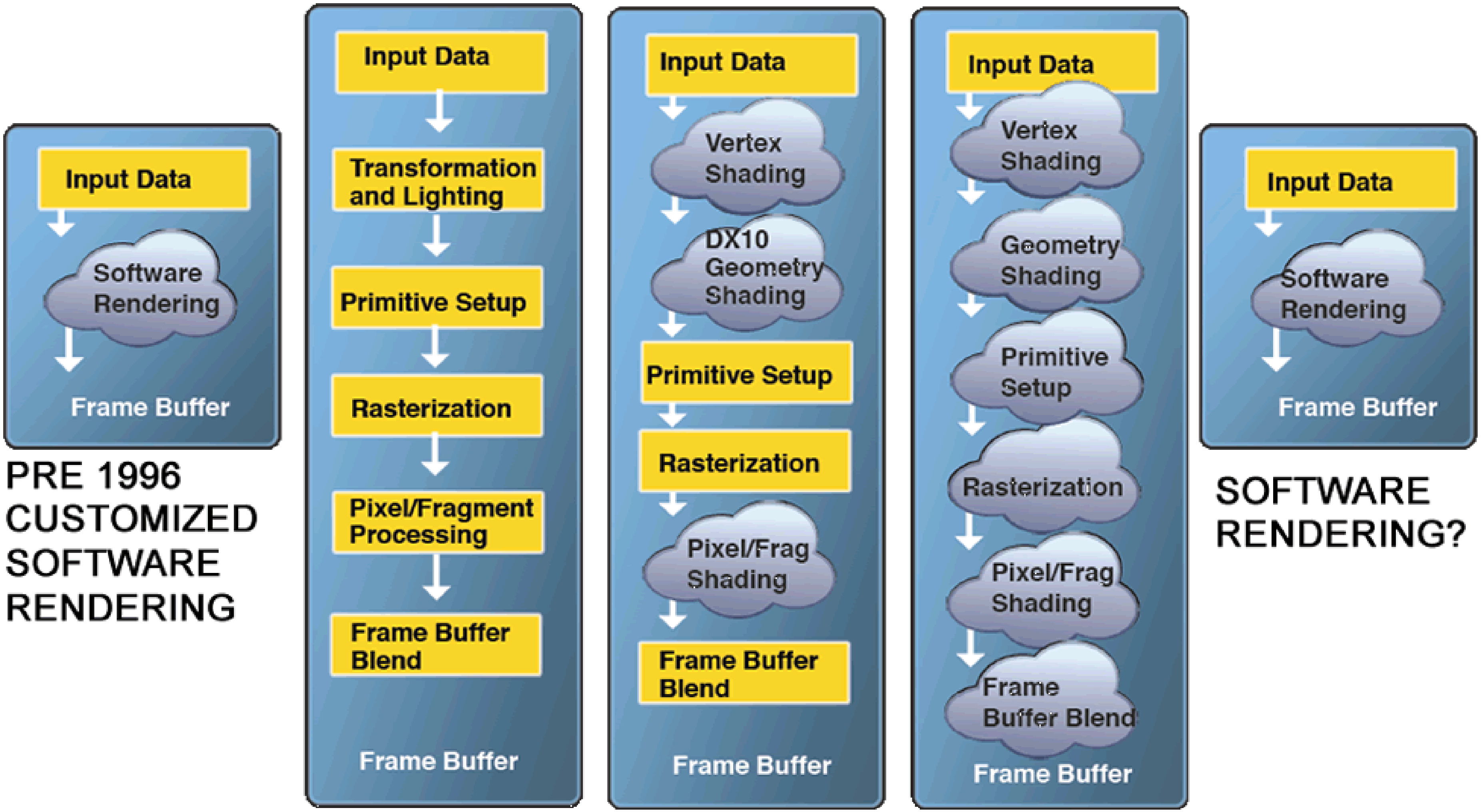


The programmer simply sends **data** to the card and then can write a program to interpret the data and create an image.

Graphic card generations

GPU with Programmable pipeline

- **III generation (2001)** GeForce3,GeForce4 (NVIDIA), Radeon 8500 (ATI), introduced **programmable vertex shaders**, graphics card lets programmers download assembly programs to control vertex transformation and lighting, No pixel programmability.
- **IV generation (2002-2006)** GeForce FX family (NVIDIA), Radeon 9700 (ATI),Quadro4 XGL (NVIDIA), NVIDIA released Cg. **Vertex and pixel programmability**. Increased use of lighting effects such as bump mapping and shadowing. Use of 32-bit floating point
- **V generation (now) GPGPU General-Purpose GPU**: **unified shaders**, introduced in console videogames, XBOX360, GeForce 8800 (128 proc.); 32 bit floating point throughout the pipeline, GeForce 6+,
- 2010: INTEL (Larrabee), NVIDIA GeForce GTX285, AMD Radeon HD 4890



PRE 1996 CUSTOMIZED SOFTWARE RENDERING

PRE 2001

DX10

NO FIXED FUNCTION?

SOFTWARE RENDERING?

Programmable Pipeline

- **Vertex shader programs** take as input per vertex information (object space position, object space normal, etc.) and per frame constants (perspective matrix, modeling matrix, light position, etc.). They produce some of the following outputs: clip space position, diffuse color, specular color, transparency, texture coordinates, and fog coordinates.

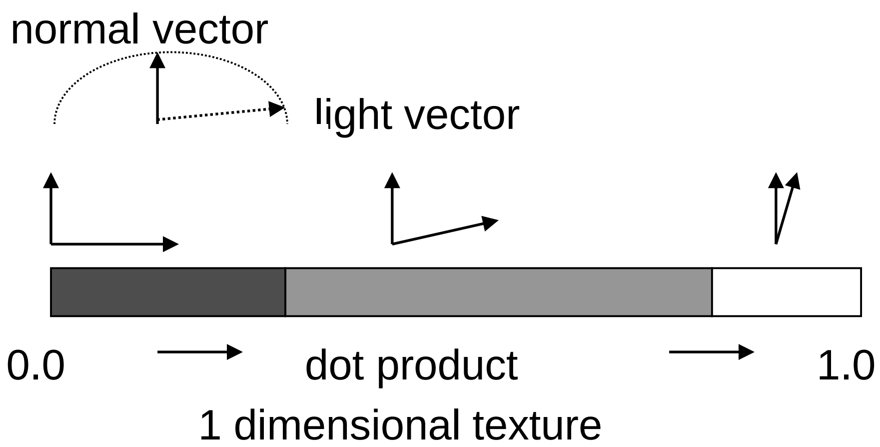
- **Pixel shader programs** take as input the outputs from the vertex shader program and texture maps. They produce a final color and transparency as output. They are often called fragment shaders.
  - Per-Pixel Lighting
  - Environment Mapping, Bump Mapping
  - NPR (Non Photorealistic Rendering)

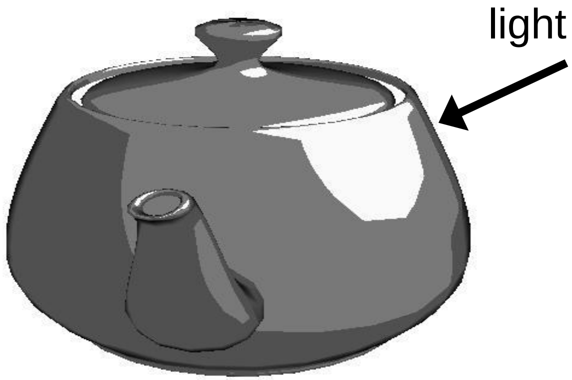
Example: Cartoon Shading

Cartoon shading is a cheap and neat looking effect used in video games

Instead of using traditional methods to light a vertex, use the dot product of the light vector and the normal of the vertex to index into a 1 dimensional "texture" (A texture is simply a lookup function for colors – nothing more and nothing less)

Instead of a smooth transition from low intensity light (small dot product) to high intensity light (large dot product) make the 1 dimensional texture have sharp transitions

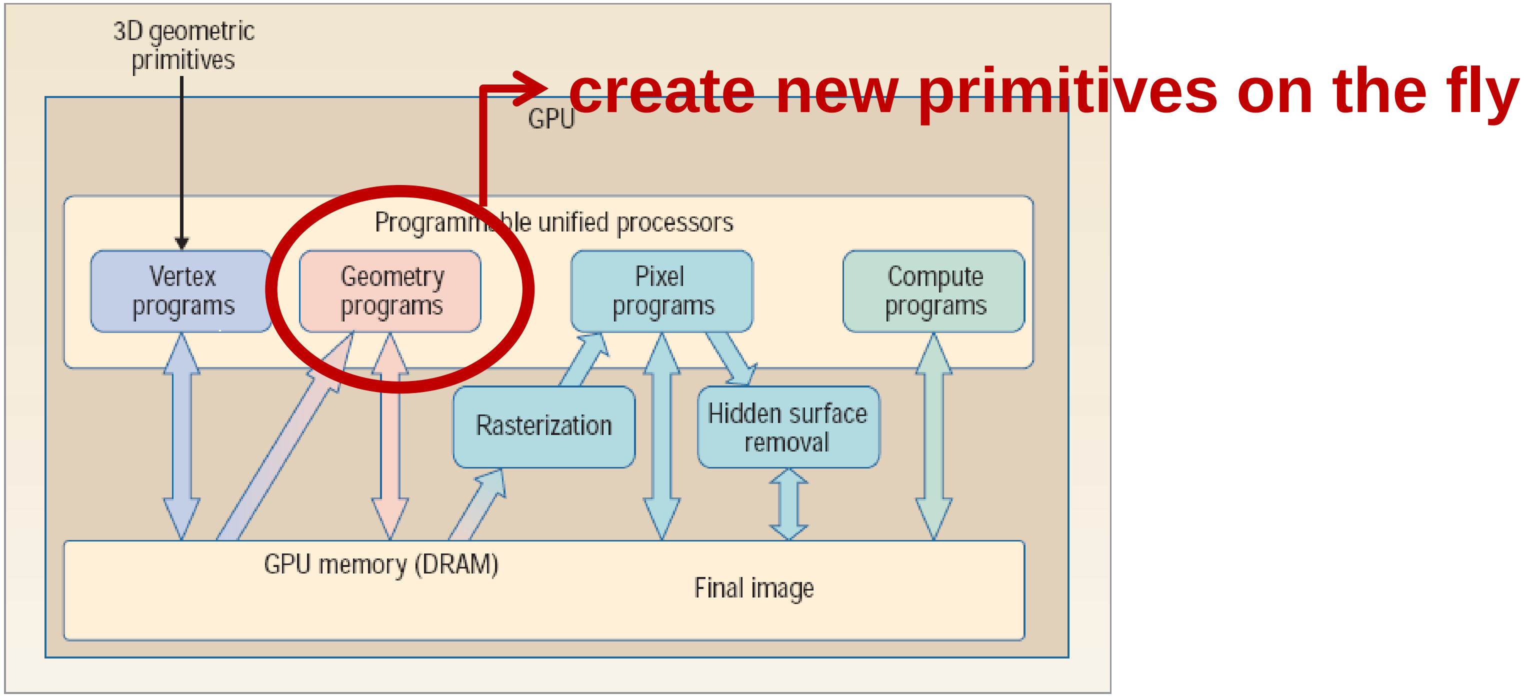




Languages for shader programs

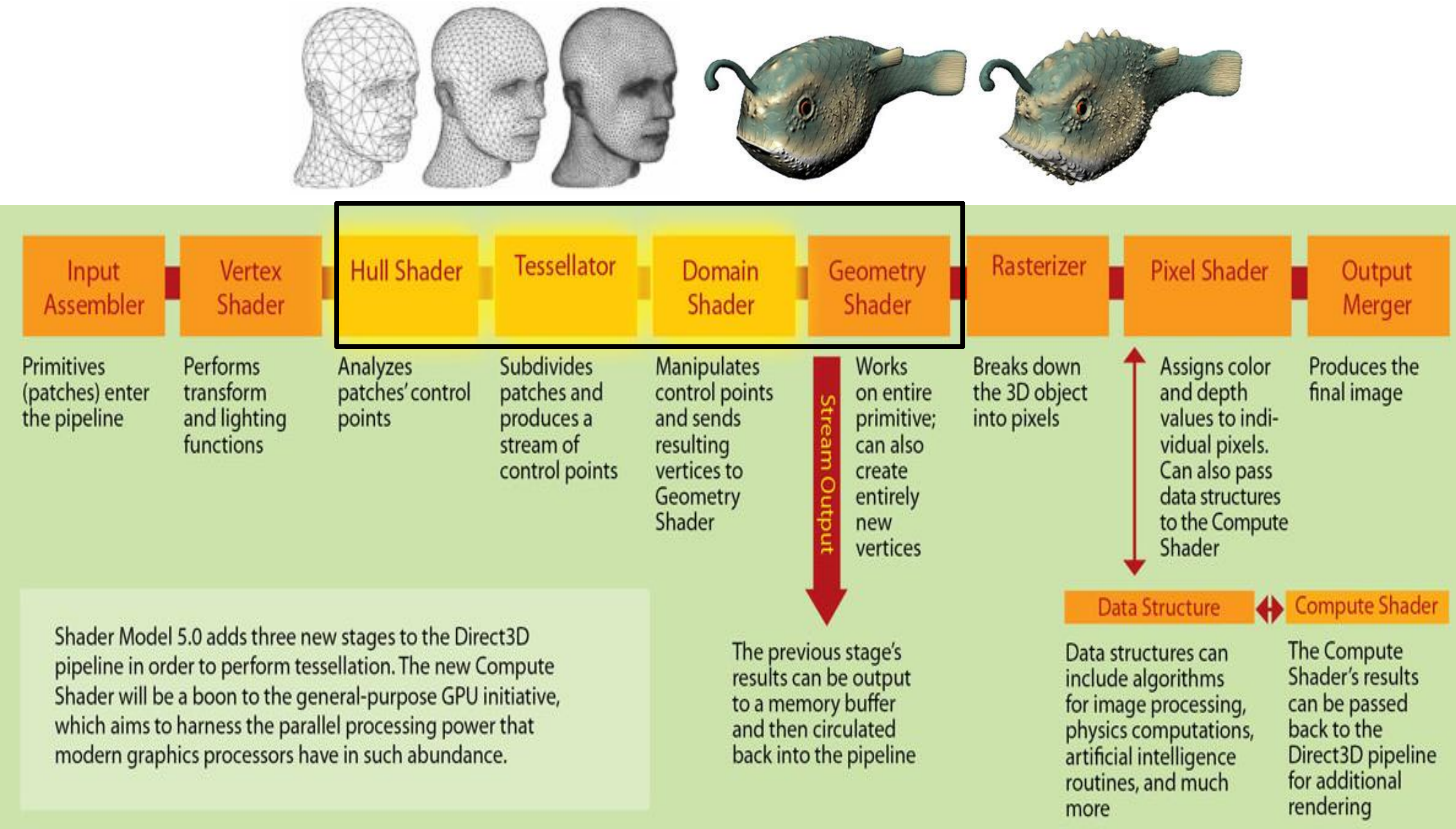
- Programming Languages
  - Cg from NVIDIA: Cg is a C-like language that the graphics card compiles into a program
    The program is run once per-vertex **and/or** per-pixel *on the graphics card*
  - RenderMonkey from ATI
  - HLSL from Microsoft in DirectX
  - GLSL from SGI in OpenGL (>2.0)

V generation: unified shaders and GPGPU



- Provides one large grid of data-parallel floating-point processors
- Vertices, triangles, and pixels recirculate through a set of programmable processors
- Demand for the various shaders in a single image can vary unpredictably. A unified shader architecture can allocate a varying percentage of its pool of processors to each shader type.

Shader model pipeline

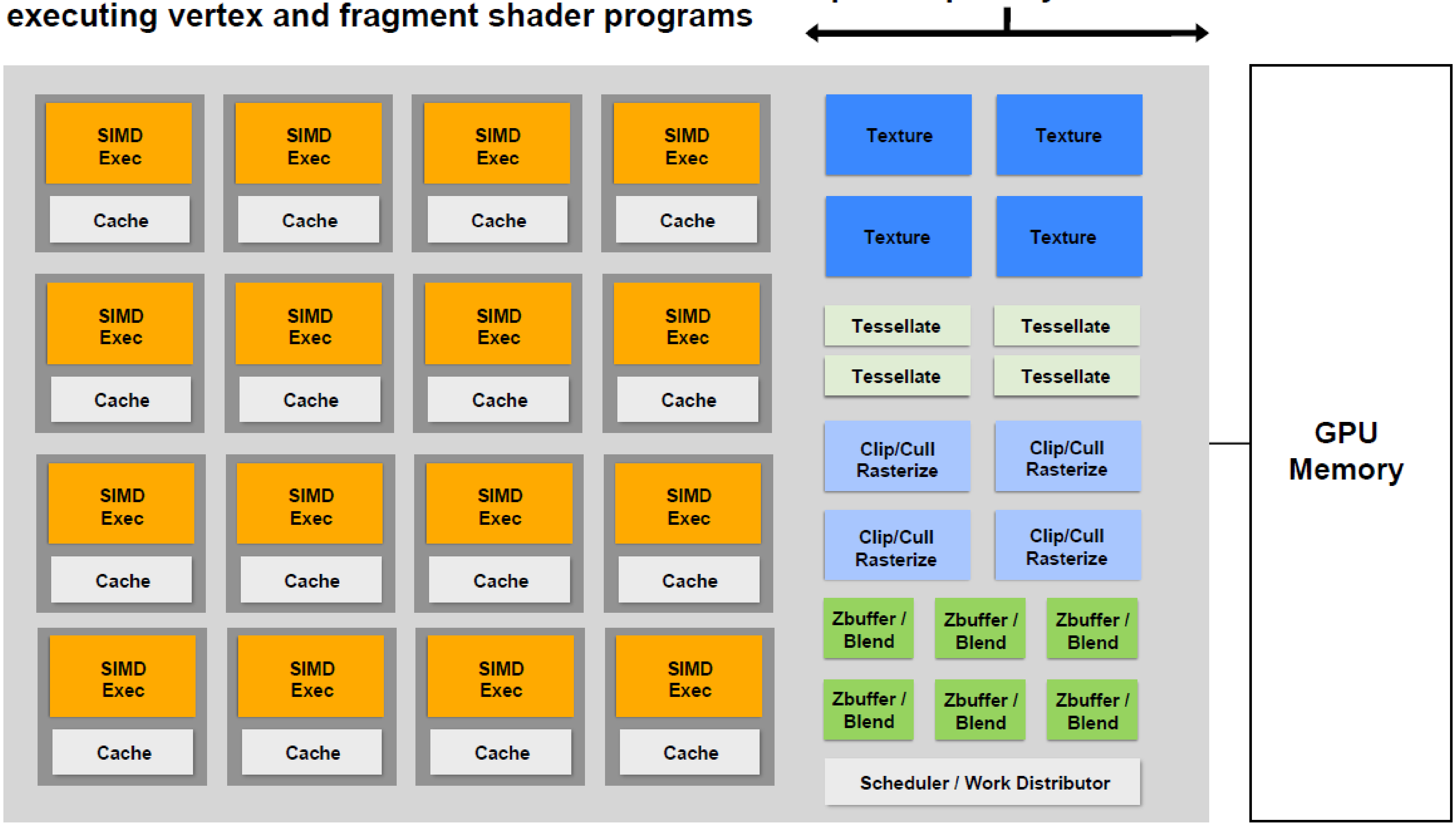


V generation: GPU multi-core Architecture

What's in a GPU?

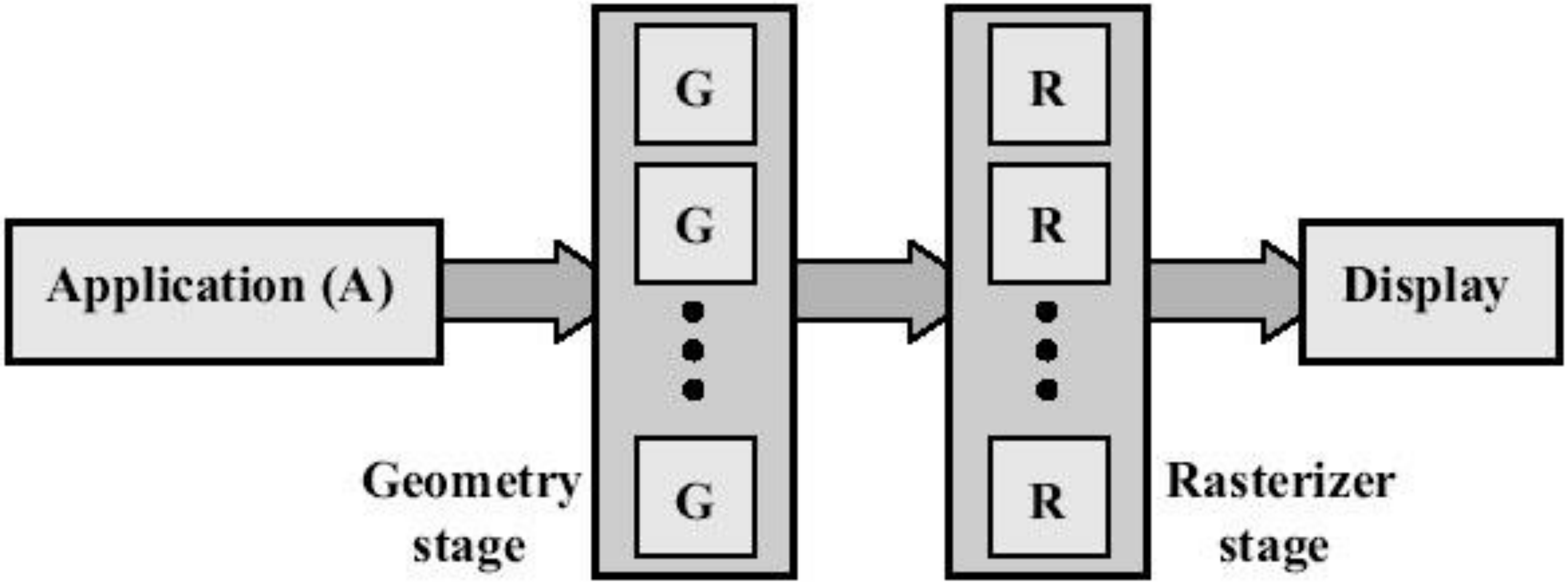
Modern GPUs offer ~2-4 TFLOPs of performance or executing vertex and fragment shader programs

T-OP's of fixed-function compute capability over here



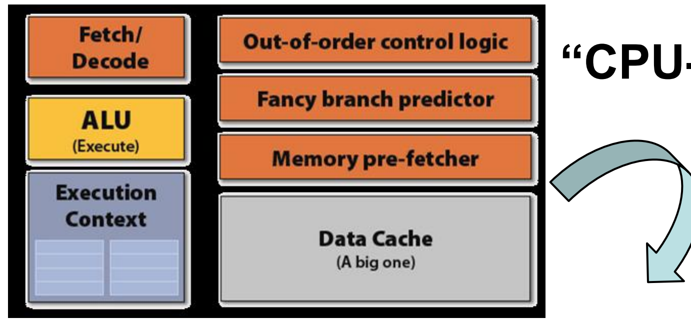
GPU Architecture

- Two major ways of getting better performance:
  - Pipelining
  - Parallellization
  - Combinations of these



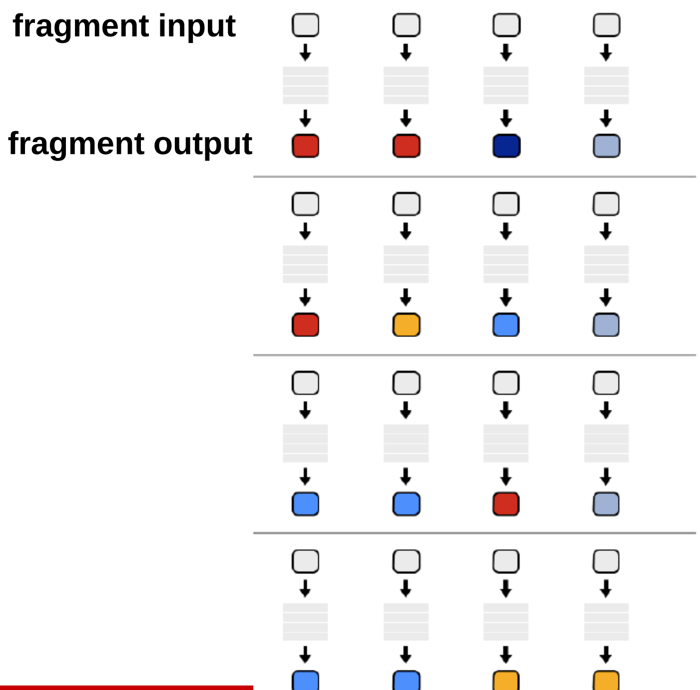
Idea # 1: remove control logic

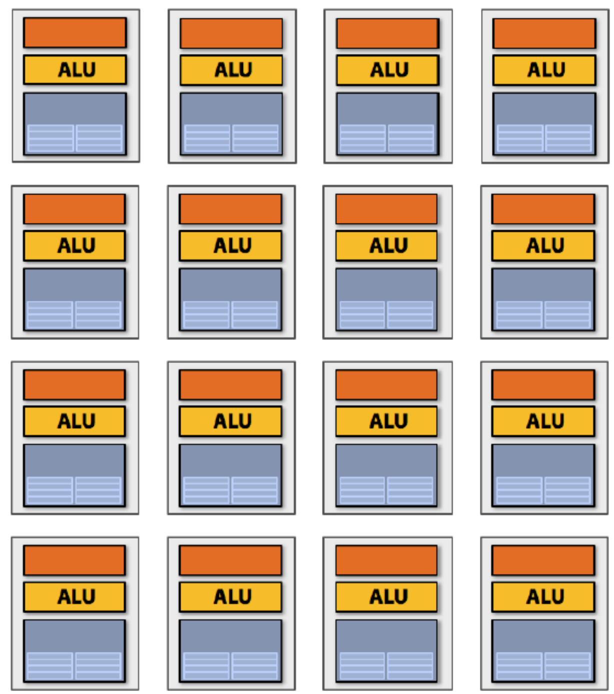
Remove components that help a single instruction stream run fast



"CPU-style" cores

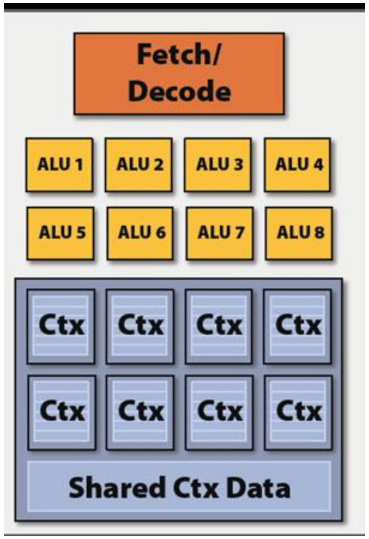
Sixteen cores
(sixteen fragments in parallel)





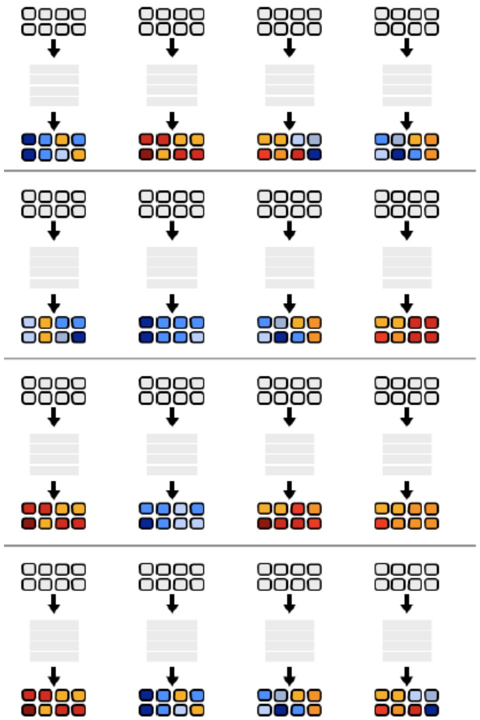
16 cores = 16 simultaneous instruction streams

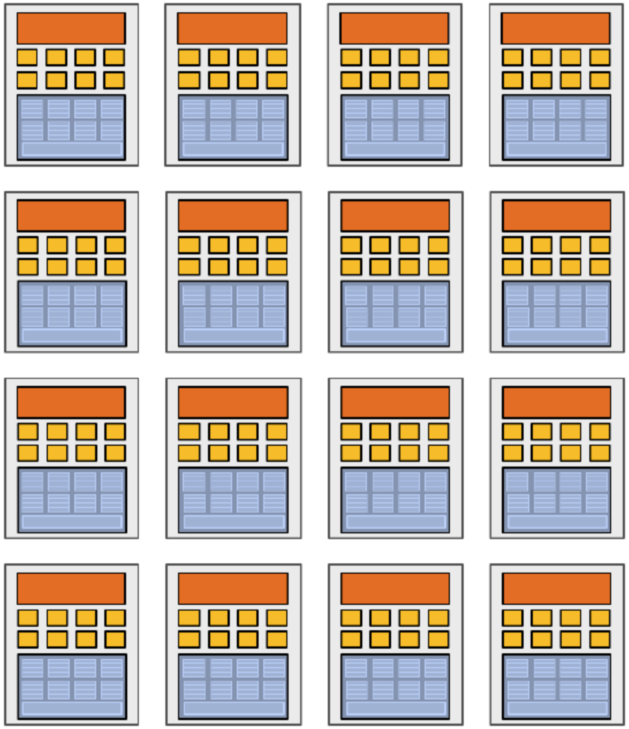
Idea #2: Add ALUs



Amortize cost/complexity of managing an instruction stream across many ALUs → **SIMD process**

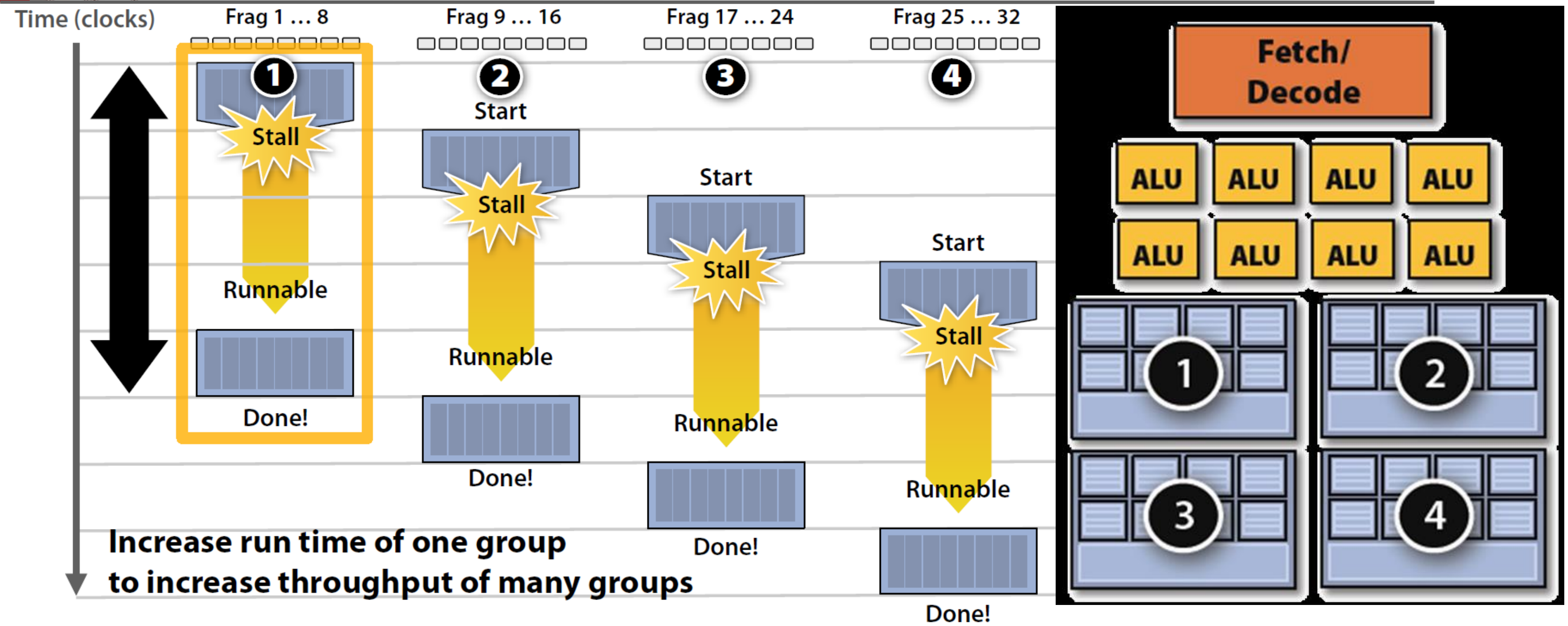
Processes 8 fragments using vector ops on vector registers





16 cores = 128 ALUs
= 16 simultaneous instruction streams

Idea #3: interleave fragments



Stalls occur when a core cannot run the next instruction because of a dependency on a previous operation.

Stall handle by context switch: Interleave processing of many fragments on a single core to avoid stalls caused by high latency operations.

NVIDIA GeForce GTX 280 (2009)

= single "physical" instruction stream fetch/decode (functional unit control)

= SIMD programmable functional unit (FU), control shared with other functional units. This functional unit may contain multiple 32-bit "ALUs"

= 32-bit mul-add unit

= 32-bit multiply unit

= execution context storag

= fixed function unit

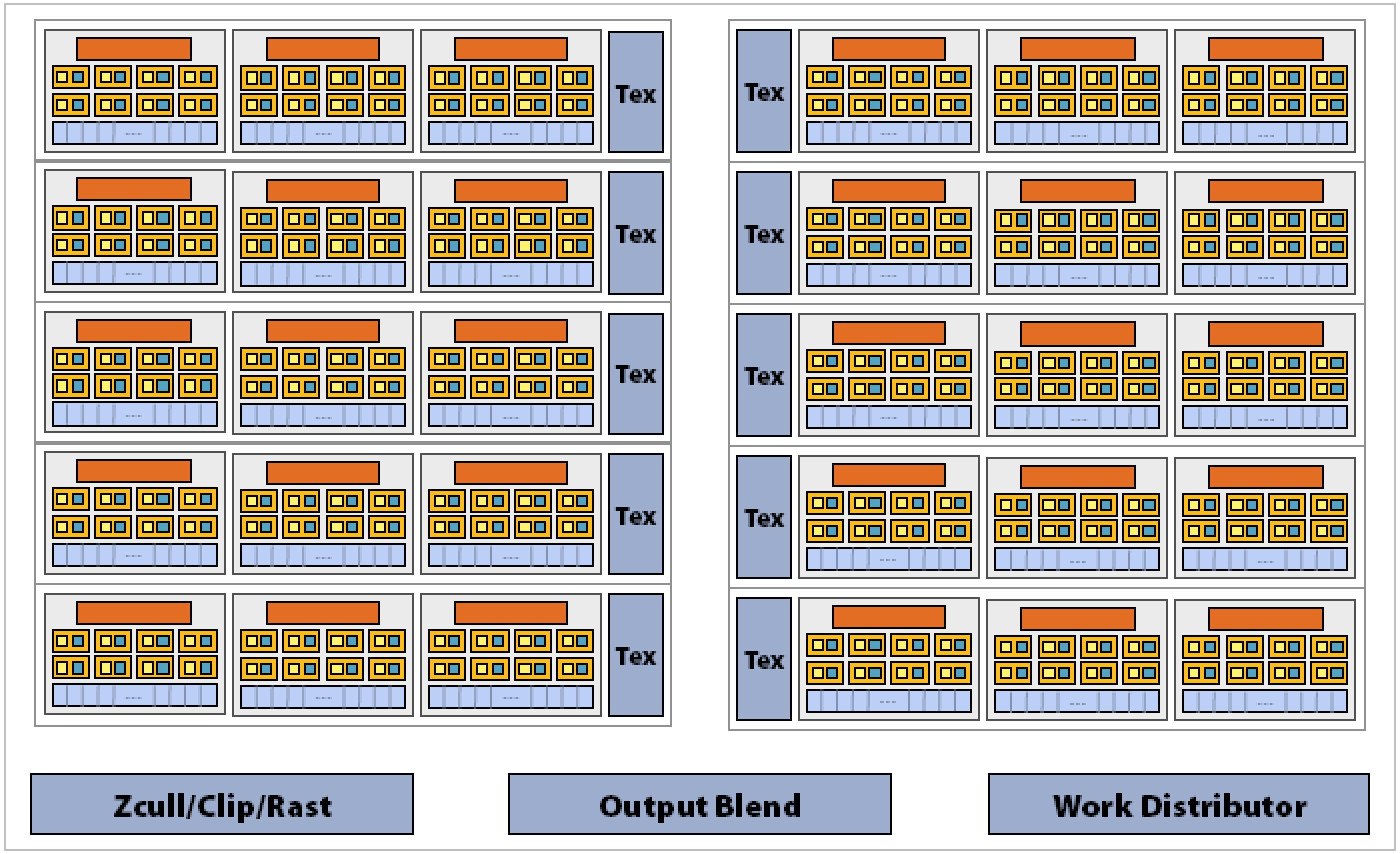
30 processing cores

8 SIMD functional units per core

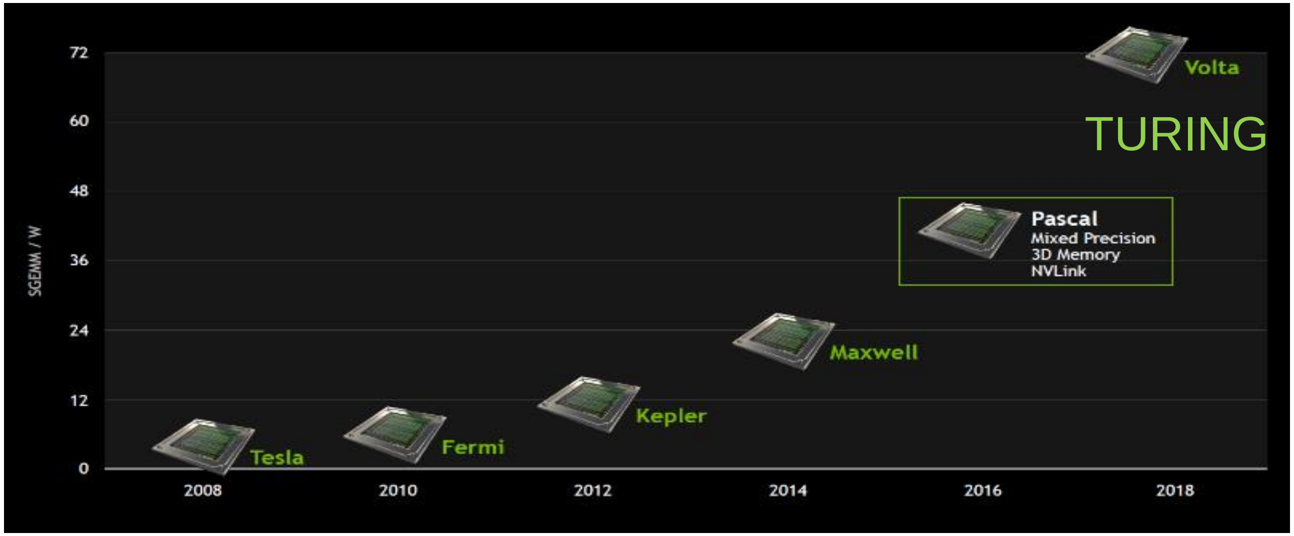
Best case: 240 mul-adds + 240 muls per clock

1.3 GHz clock

30*8*(2+1)*1.3 = 933GFLOPS



GPU roadmap (NVIDIA)



Tensor Cores

RT Cores +
Tensor Cores

- Pascal microarchitecture:
- An SM (streaming multiprocessor) consists of 64 CUDA cores, partitioned into two processing blocks, each having 32 single-precision CUDA Cores
- GDDR5 Unified memory – A memory architecture, where the CPU and GPU can access both main system memory and memory on the graphics card with the help of a technology called "Page Migration Engine".
- NVLink A high-bandwidth bus between the CPU and GPU, and between multiple GPUs estimated to provide between 80 and 200 GB/s.

Turing GPU Architecture

NVIDIA RTX 2080 e 2080 Ti

- (settembre 2018) **RTX** sta ad indicare il fatto che queste schede video sono in grado di **processare il Ray Tracing** in maniera nativa, ossia con risorse hardware dedicate. Questo permette quindi di avere **effetti di luce e riflessi** estremamente realistici.
  - **Turing SM Architecture is partitioned into**

- **CUDA CORES:**
  4 sub-cores ; each with
  16 INT32 cores,
  16 FP32 cores,
- **2 (AI) Tensor CORES,**
- **Ray Tracing (RT) CORES**

Framebuffer 11 GB di RAM GDDR6

GPGPU
(General Purpose GPU)

- GPU can be used for things other than graphics! GPU as a co-processor for general purpose computation
- **Why use the GPU?** GPU is more specialized than CPU, so can do what it does fast; Parallel, pipelined architecture
- Instead of pixels with color, think of a grid with a four-component vector at each cell.
- Instead of frames, think of time-steps.
- Instead of rendering equations, perform any computation you want.

GPGPU (General Purpose GPU)

- **OLD:** *GPGPU – trick the GPU into general-purpose*
- computing by casting problem as graphics
- Turn data into images (“texture maps”)
- Turn algorithms into image synthesis (“rendering passes”)

- **NEW:** GPU computing with CUDA (Compute Unified Device Architecture) provides a scalable data-parallel platform in a familiar environment C

GPGPU computing with CUDA and OpenCL

- These technologies allow specified functions called compute kernels from a normal C program to run on the GPU's stream processors. This makes C programs capable of taking advantage of a GPU's ability to operate on large buffers in parallel, while still making use of the CPU when appropriate.

- **NVIDIA® CUDA™** in NVIDIA GPUs to solve many complex computational problems in a fraction of the time required on a CPU.
- **Open Computing Language (OpenCL)** is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, digital signal processing (DSPs), field-programmable gate arrays (FPGAs).

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Serena Morigi
Dipartimento di Matematica
serena.morigi@unibo.it
http://www.dm.unibo.it/~morigi