Computer Graphics Hardware and Software

Lecture Notes,
CEng 477

What is Computer Graphics?

- Different things in different contexts:
  - pictures, scenes that are generated by a computer.
  - tools used to make such pictures, software and hardware, input/output devices.
  - the whole field of study that involves these tools and the pictures they produce.
- Use of computer to define, store, manipulate, interrogate and present pictorial output.

- How pictures are represented in computer graphics?
- How pictures are prepared for presentation?
- How interaction within the picture is accomplished?

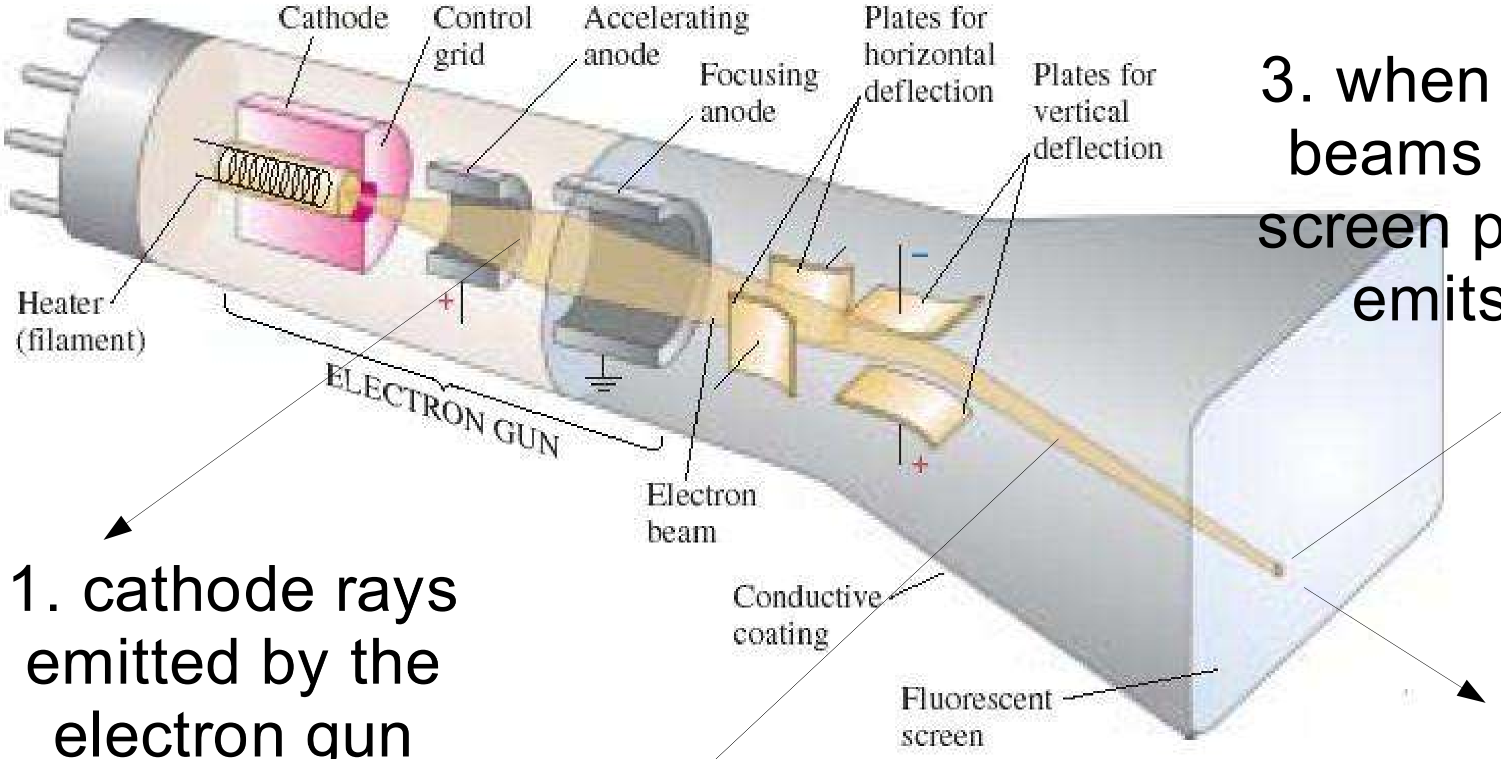
Computer Graphics Applications

- Art, entertainment, and publishing
  - movie production, animation, special effects
  - computer games
  - World Wide Web
  - Book, magazine design, photo editing
- CG and Image processing (syntesis vs. analysis)
- Simulations (education, training)
- CAD architectural, circuit design etc.
- Scientific analysis and visualization
- Graphical User Interfaces

Display (Video Display Device)

- Most CG on video monitors
- Still most popular: Cathode Ray Tube (CRT)
- Other popular display types:
  - Liquid Crystal Display
  - Plasma display
  - Field Emission Displays
  - Digital Meromirror Devices
  - Light Emitting Diodes
  - 3D display devices (hologram or page scan methods)

CRT



1. cathode rays emitted by the electron gun

2. focusing and deflection

3. when electron beams contact screen phosphor emits light

4. light fades, redraw required in a small period (refresh)

CRT types

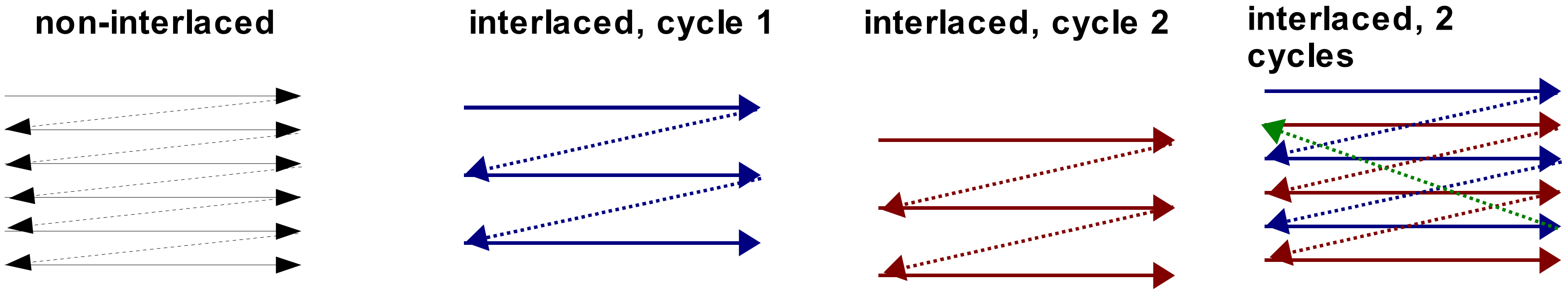
- Direct View Storage Tubes (not CRT, no need for refresh, pictures stored as a permanent charge on phosphor screen)
- Calligraphic refresh CRT (line drawing or vector random scan, need refreshing)
- Raster-scan (point by point refreshing)
- **Refresh rate:** # of complete images (frames) drawn on the screen in 1 second. Frames/sec.
- **Frame time:** reciprocal of the refresh rate, time between each complete scan. sec/frame

Vector Scan

- Picture definition is stored as a set of line-drawing commands in a refresh buffer.
- to display a picture, the system cycles through the set of commands in the buffer
- Designed for line drawing applications (CAD)

Raster Scan

- Screen is a regular grid of samples called **pixels** (**pic**ture **el**ement)
- Screen is refreshed line by line



non-interlaced

interlaced, cycle 1

interlaced, cycle 2

interlaced, 2 cycles

- Interlacing: Avoid flickering affect for small refresh rates. interlaced 50Hz: actually 25Hz

- **resolution:** a 2D term that measures the number of scan-lines and the number of pixels on each line (maximum number of points that can be displayed without overlap on a CRT)
- **black and white** display only binary pixels.
- **intensity** of a pixel can be achieved by the force of electron beam (gray scale)
- **color** display?

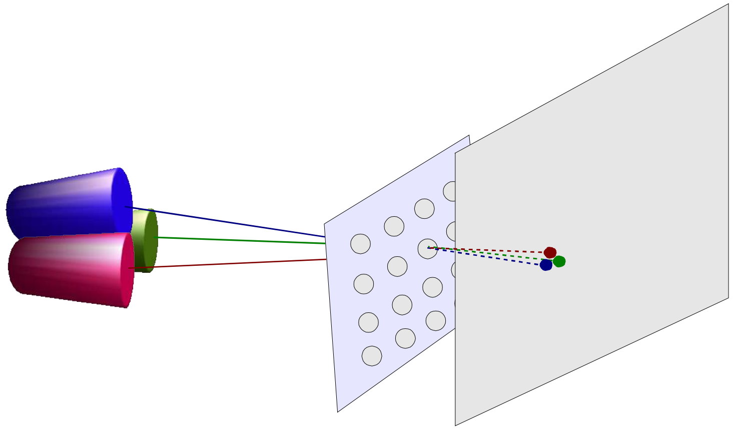
Color Displays

- Beam penetration method:

  special phosphors emitting different colors for different intensity of electron. Slow, limited colors.

- Shadow mask method:

  3 electron guns + a shadow mask grid. Intensities of 3 colors result in an arbitrary color pixel. (most TVs and monitors)



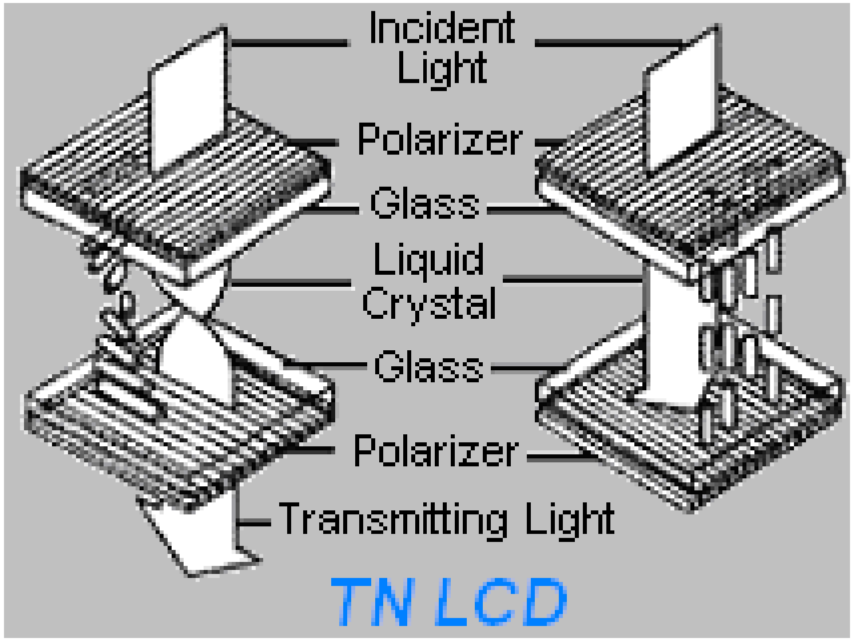
- black and white: 1 bit per pixel.
- gray scale: 1 byte per pixel (256 gray levels)
- true color: 3 bytes=24pits per pixel (2^{24} colors)
- indexed color frame buffer: each pixel uses 1 byte, an index entry in a colormap table matching the color to the actual color.

Vector vs Raster Scan

- raster scan monitors:
  - inexpensive
  - filled areas, patterns
  - refresh process is independent (constant for any complex scene)
- vector scan monitors:
  - Smooth lines. no need for scan conversion: lines to pixels. (raster scan solution antialiasing)
  - sometimes memory and CPU efficient 1000 lines:
    Vector scan: 2000 endpoints and 1000 operations
    Raster scan: whole frame buffer 1000 scan conversions.

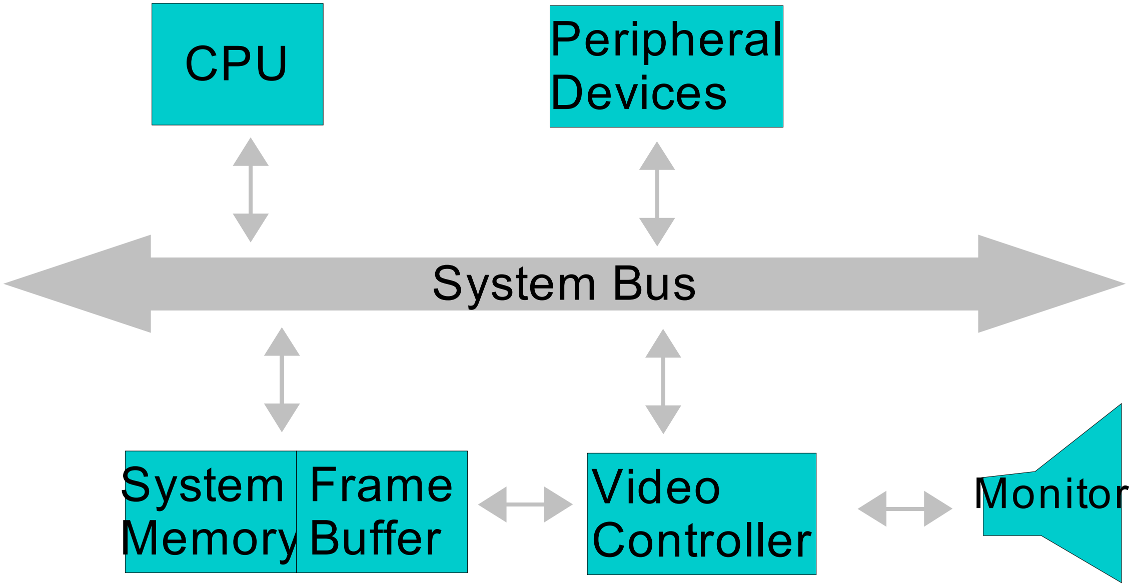
LCD Displays

- Thinner and lighter. No tube and electron beams.
- Blocking/unblocking light through polarized crystals.
- A matrix of LC cells one for each pixel.
- No refresh unless the screen changes.
- Color 3 cells per pixel.



Simple Raster Display System

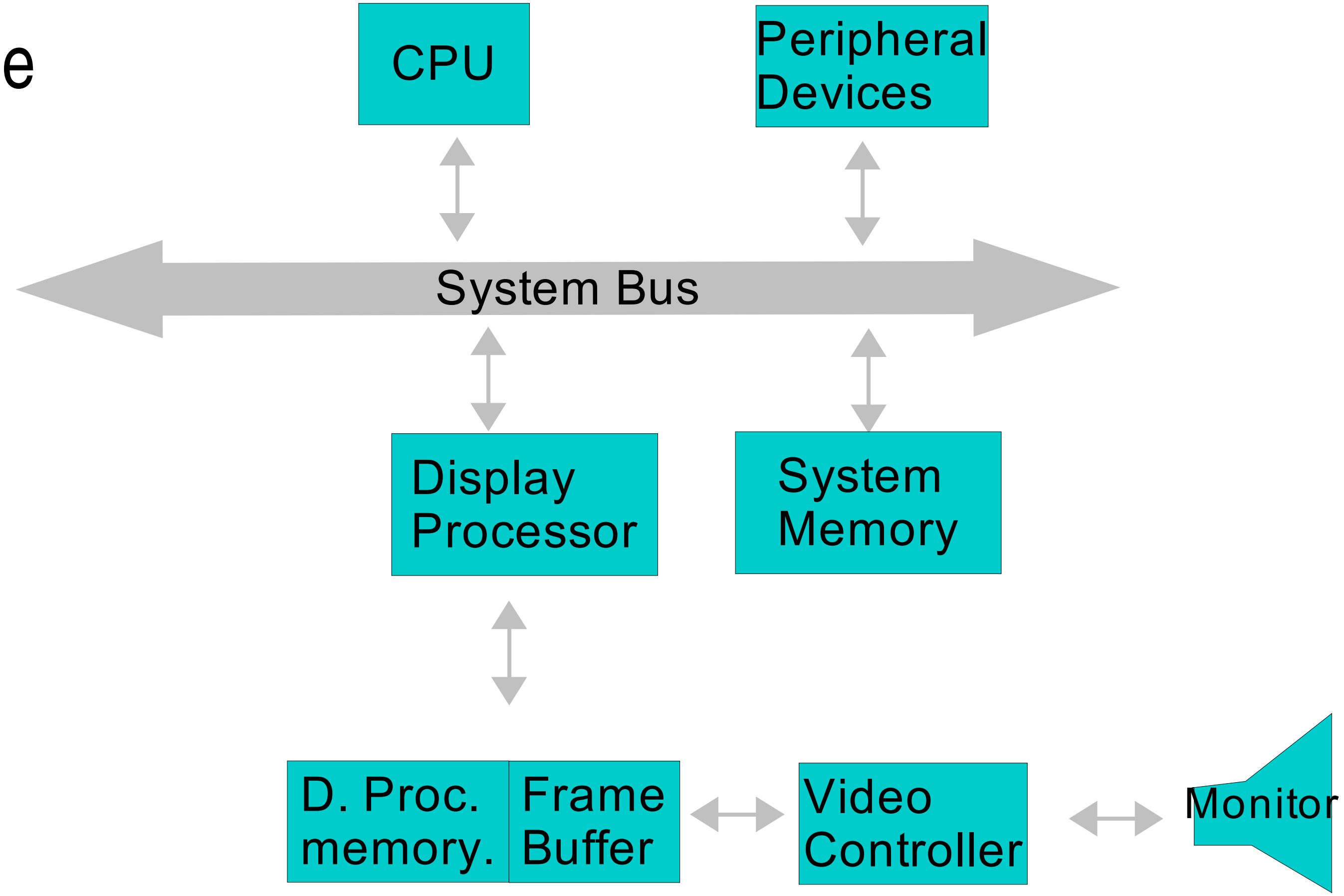
- Frame buffer: stored pixel map of screen
- Video controller just refreshes the frame buffer on the monitor periodically.



- Inexpensive
- Scan conversion of output primitives (lines, rectangles etc.) done by the CPU. Slow.
- As refresh cycle increases, memory cycles used by the video controller increases. Memory is less available to CPU.
- Solution: Graphics Display Processor

Graphics Display Processor

- Scan conversion, output primitives, raster operations (double buffering)
- Separete frame buffer



Computer Graphics Software

- Rendering Primitives
  - Models are composed of, or can be converted to, a large number of **geometric primitives**.
  - Typical rendering primitives directly supported in hardware include:
    - Points (single pixels)
    - Line segments
    - Polygons (perhaps simple, triangle, rectangle)

- Modeling primitives include these, but also
    - Piecewise polynomial (spline) curves
    - Piecewise polynomial (spline) surfaces
    - Implicit surfaces (quadrics, blobbies, etc.)
    - Other...
- Software renderer may support modeling primitives directly, or may convert them into polygonal or linear approximations for hardware rendering

Algorithms

- A number of basic algorithms are needed:
  - **Transformation:** Convert representations of models/primitives from one coordinate system to another
  - **Clipping/Hidden surface removal:** remove primitives and part of primitives that are not visible on the display
  - **Rasterization:** Convert a projected screen space primitive to a set of pixels.

- Advanced algorithms:
  - **Picking:** select a 3D obejct by clicking an input device over a pixel location.
  - **Shading and illumination:** Simulate the interaction of light with a scene.
  - **Animation:** Simulate movement by rendering a sequence of frames.

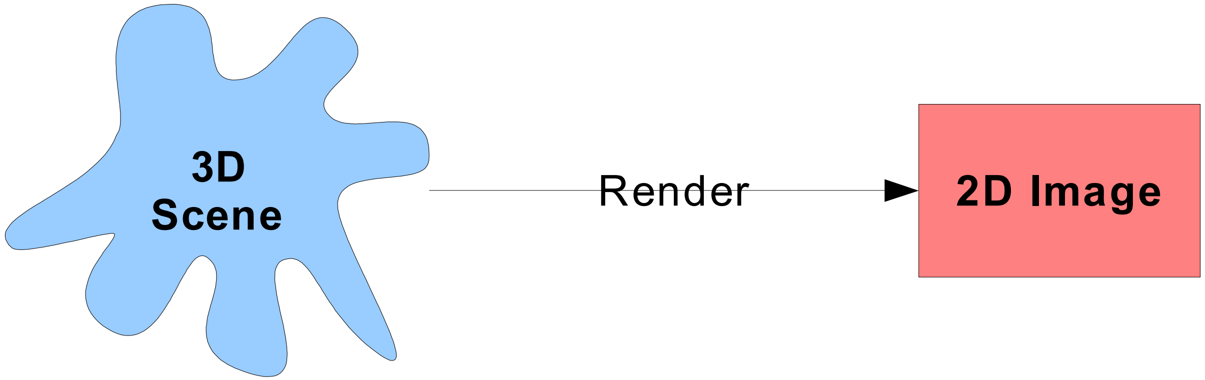
Application Programming Interfaces

- X11: 2D rasterization
- Postscript, PDF: 2D transformations, 2D rasterization
- Phigs+, GL, OpenGL, Direct3D: 3D pipeline
- APIs provide access to rendering hardware via conceptual model.
- APIs abstract the hardware implementations and algorithms in standard software calls.

- For 3D interactive applications, we might modify the scene or a model directly or just the change the attributes like viewing information.
- We need to interface to input devices in an event-driven, asynchronous and device independent fashion. APIs and toolkits are also defined for this task. GLUT, Qt, GTK, MFC, DirectX, Motif, Tcl/Tk.

Graphics Rendering Pipeline

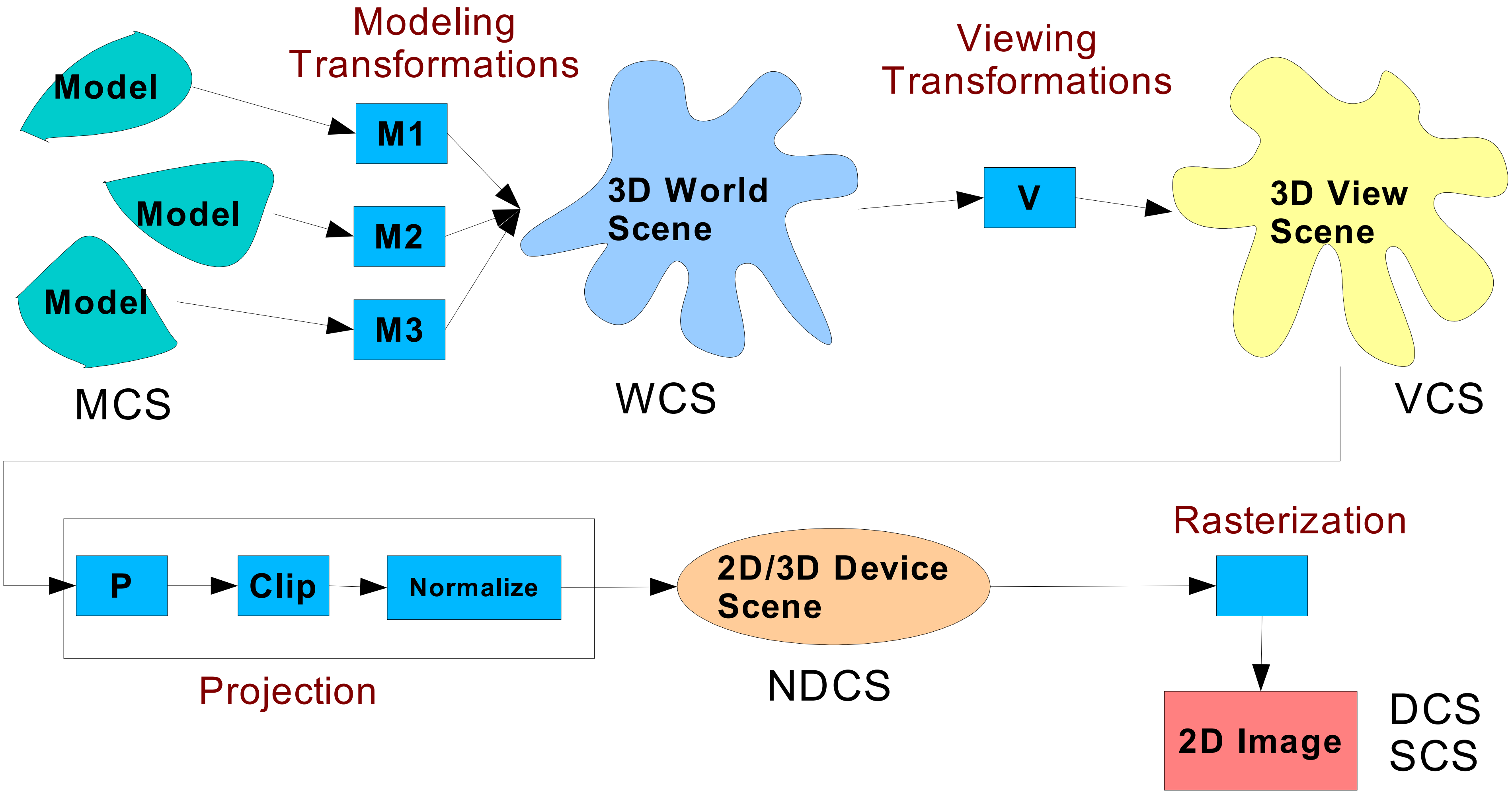
- **Rendering:** conversion from **scene** to **image**



- Scene is represented as a **model** composed of primitives. Model is generated by a program or input by a user.
- Image is drawn on an output device: monitor, printer, memory, file, video frame. Device independence.

- Typically rendering process is divided into steps called the graphics pipeline.
- Some steps are implemented by graphics hardware.
- Programmable graphics accelerator, GPU: programmable pipelines in graphics hardware

- The basic **forward projection pipeline:**


Modeling Transformations
Model
Model
Model
M1
M2
M3
MCS
3D World Scene
WCS
Viewing Transformations
V
3D View Scene
VCS
P
Clip
Normalize
Projection
2D/3D Device Scene
NDCS
Rasterization
2D Image
DCS
SCS