Introduction to Computer Graphics

Ming-Hwa Wang, Ph.D.
COEN 148/290 Computer Graphics
Department of Computer Engineering
Santa Clara University

Mandatory Basics

- an image is a two-dimensional array of picture element (pixel); the resolution of an image refers to how closely packed the pixels are when displayed
- continuous images (real images) vs. discrete images (computer graphics on raster display)
  - there is no such thing as a smooth curve in a computer image, the illusion is achieved mostly by using large quantities of pixels (sometime with anti-aliasing)
- color space:
  - red, green, blue (RGB) color space - internal representation
    - color palette using color lookup table (LUT): 256 levels per RGB component requires 8 bits or a byte, LUT converts a color number (color index value or pseudo color) into its RGB color
  - hue, intensity (or value, or lightness), saturation (HSV or HLS) color system - human representation
- computer graphics: the process of converting a mathematic or geometric description of an object (the model) into a two-dimensional projection (a visualization) that simulate the appearance of a real object
  - limitations: computer graphics can't do depth of field and motion blur as cameras can, and cameras do not have near and far clipping plans as computer graphics have
  - a de facto approach: starting with a very detailed geometric description, applying a series of transformations, imitating reality by making the object look solid and real (rendering)
    - not sufficient for animation, need synthesis methods and modeling methods
  - the main elements of a graphics system: application program ↔ process with interaction → frame store → video controller → raster display
  - photo-realism: make a graphics image of an object or scene indistinguishable from a TV image or photograph
  - computer graphics is not an exact science, involve much simplification in the original mathematical model so that it can be implemented as a computer graphics algorithm
- persistence of vision: human's eyes continue to perceive light for a short time after the light has gone away (also the phosphor persistence for CRT)

Hardware Basics: interactive output technologies

- monitor
  - type:
    - cathode ray terminal (CRT)
    - flat panel (LCD or plasma): thinner, lighter, less power, crisper display
    - head mounted display: with optics to feel like large screens, separate displays for eyes to see 3D, can sense your head position/orientation and with position-sensitive input device to make virtual reality possible
  - raster scan: the dot that can be lit up is rapidly swept across the monitor face in a pattern
  - size: measured in screen diagonal length
  - aspect ration: ratio of width / height
  - triads: 3 colors are arranged in a group, and triads are arranged in a hexagonal pattern so that each triad has six neighbors
  - dot pitch: the smallest detail a monitor can possibly show or resolve, the distance from any one of a triad to each of the six neighbors
  - scan rate or refresh rate: horizontal scan rate is the rate at which individual scan lines are drawn (15-100 KHz), and the vertical scan rate indicates how often the entire image is refreshed (60-70 Hz)
  - interlacing (alternating even field and odd field): refreshes only every other scan line each vertical pass, and requires only half as many scan lines to be drawn, reduce hardware cost but flicker quite noticeably
- hard copy output
  - printer
    - inkjet: 300-700 DPI (dot per inch), slow, inexpensive, quality ok
    - laser printer: faster, better quality, more expensive
    - wax transfer: quality even better, slower, more expensive
    - dye sublimation: ultimate image quality, and very expensive
  - photograph: film recorder
- display controller (video card or graphic card)
  - the video random access memory (VRAM) bitmaps is the heart of any display controller, it is where the pixels are kept, and the bitmaps divides the remaining display controller into the drawing "front end" and the video "back end"
    - drawing front end: off-load processor by receiving drawing commands from the processor, and the pixels are "drawn" by writing the new values into the bitmaps, most display controllers also allow the processor to directly read/write the pixels in the bitmaps
    - video back end: interprets the pixel values into their colors and creates the video signals that drive the monitor
  - types:
    - 2D display controller or graphics user interface (GUI) engine
    - 2½D display controller: allowing the color to vary across the object being drawn using bilinear interpolation, dithering, and Z buffering
    - Full 3D display controller: do transformation, lighting, and other advanced effects

Graphic Primitives

- graphics primitives: can be draw directly into bitmaps
  - point primitives are the smallest thing we can draw (pixels)
  - vector primitives
    - line: artifact – diagonal lines are 29% fewer pixels than horizontal or vertical lines
    - polyline: more efficient than an equivalent list of independent lines
  - polygon primitives
    - convex polygons: most graphics systems draw simple convex polygons directory
      - triangles: always convex and simple
    - concave polygons: easier to decompose into triangles
    - triangle strip or Tstrip: only 1 point is needed to specify each new triangle, often used to draw curve surfaces
      - if each triangle is drawn individually, the shared corner may compute 6 times, mesh and strip avoid/reduce this overhead
    - quad mesh
- complex shapes can be drawn by lots of simple primitives

Modeling

- modeling: the process of describing an object or scene so that we can eventually draw it (and makes it easy to draw)
  - most real objects are opaque and we can only see their surface, surfaces are often easier to describe than volumes, most 3D rendering hardware can draw only surfaces
- model types
  - explicit surface model: an outright list of all surface elements, or patches, that make up a surface
    - polygons: the predominant method for modeling surface using tessellation, need find appropriate tessellation level and make them smooth
    - curved patches: model small regions of our curved surface with patches of the convenient mathematical functions, can get a closer fit to the intended surface with fewer patches, more compact and accurate, but need to decompose into polygons for rendering on hardware
      - biquadratic or bicubic patches
    - spline patches: a curved patch where the mathematical function is governed by a set of control points, the patch takes on a smooth shape based on the position of these control points
      - nonuniform rational b-spline (NURBS), B-spline, beta spline, etc.
  - implicit surface: described by a mathematical function that can answer whether any point you ask about is to one side, or the other side, or on a surface; simpler but need eventually surface points will have to found so that the surface can be drawn
    - isosurfaces: 3D contour
    - potential function: builds surfaces by modeling lots of little blobs together, the blobs are spheres when far apart, and they bleed into each other when close together
  - constructive solid geometry (CSG): objects are created by combining simple 3D solids (spheres, boxes, cones, and cylinders) using only 3 Boolean operations: OR (or union), AND (or intersection), NOT (for minus), and XOR
  - space subdivision: divide space into voxels (or volume elements); the number of voxels goes up with the cube of resolution, but not subdivide all the space at the same resolution; it is homogenous, i.e., you don't need detailed information in any region of space that contains the same thing throughout
    - octrees: always broken in the middle along each of the 3 major axes, and all voxels are always rectangular solids for storing 3D volumetric data
      - quadtrees are 2D variations of octrees
    - binary space partition (BSP) trees: if a block of space isn't homogenous, it's broken into two blocks, this process continues until either they are homogenous or a resolution limit is reached; in regular BSP trees, all blocks are rectangular solids; blocks can be cut in two by any arbitrary plane, which require less subdivision, but more difficult to describe and the resulting blocks are more difficult to manipulate
  - procedure models: described by a method, or procedure, for producing primitives when necessary instead of the primitives themselves, and it is more compact than an explicit list of primitives; when the required level of detail isn't know when the model is created, the procedure can be performed after as necessary to achieve the desired level of detail
    - fractals: functions that have infinite detail, e.g., Mandelbrot set function
    - graftals: objects that are "grown" according to a strict recipe
    - particle systems: the object is drawn as the tracks of large numbers of individual particles; each particle has a start location and a trajectory
- modeling issues
  - level of detail: trade-off between realistic pictures and computing resources/storages
  - suitability: model choices

Lighting

a scene includes the objects and all the other stuff the computer needs to actually make a picture

- light: our brains get many subtle shape cues from how the brightness of an object varies across its surface; shape perception works because light doesn't come from all direction equally (otherwise, whiteout or disorienting effect make features difficult to distinguish, e.g., in cloudy slow)

- directional light: specified with only a direction to the light and an intensity, which apply everywhere in the scene, e.g., sunlight
- point light: light intensity falls off with the square of the distance from the light source, but in computer graphics, it just falls off with the distance
- spotlights: a light with shade or reflector around it so that it shines only in a cone
- ambient light: the light is scattered about by bouncing off other objects; most rendering methods don't handle scatter light (except radiosity), so we pretend there is a constant low-level illumination everywhere to prevent areas from appearing complete dark (not for shape perception)

Camera or Viewing

- viewing
  - eye point (view point or camera point): the point to look from
  - view direction or gaze direction: the direction to look into
  - lookat point: a point that is to appear in the center of the picture
- projections
  - perspective projection: distant objects are drawn smaller than near ones; all objects get projected onto the image along a line from the eye point; our brain naturally interpret depth from the perspective projection scaling
  - flat or orthographic projection: objects are projected flat onto the image without any change in size (when preserving dimension is important); it is the same as perspective projection with the eye point infinitely far back

Calculate the light-object interaction

- local or direct reflection model: only consider the interaction of an object with a light source
  - Phong: usually ends up with an object reflecting more light than it receives, can't do shadows and have to be calculated by a separate 'add-on' algorithm or texture mapping
- global reflection model: consider light reflects from one object and evaluate the interaction between objects (specular interaction, diffuse interaction, shadows)
  - ray tracing: attends to perfect specular reflection, sharp refraction
  - radiosity: models diffuse interaction (common in man-made interiors), softly-lit

Rendering: converting a scene into 2D grid of pixels (image)

- rendering methods:
  - wire-frame rendering: only the edges are drawn just like stick-figure drawing; faster (because drawing lines in hardware) so useful for quick previews (or mechanical drawing need wire-frame only); old calligraphic controllers only can draw lines on the screen but don't have bitmap or pixels
  - Z-buffer rendering: only the unoccluded surfaces are drawn by maintaining an additional value (the Z-value) in each bitmap pixel beyond the color; before a new value is written into a pixel, the pixel's Z-value is compared to the new Z value; if the new Z-value represent a closer distance to the eye point, the new value is written into the pixel; otherwise, the pixel is left alone
  - ray tracing:
    - in Z-buffering, we render one object at a time until we run out of objects, and we might access any of the bitmap pixels for each object; in ray tracing, we render one pixel at a time until we run out of pixels, and we might access any of the objects for each bitmap pixel; also ray tracing images can have shadow (help human to know the relative position of objects) and reflection
    - ray casting or non-recursive ray tracing: for each pixel, a ray is shot from the eye point in the view direction for that pixel; the pixel is then set to the color of whatever the ray bumps into
    - recursive ray tracing: in theory, the original ray (the eye ray) and additional rays (launched when a reflective or transparent object is hit) might keep launching more rays indefinitely; in practice, new rays are not launched when their importance to the original pixel color is small enough to ignore, or when an arbitrary recursion is reached
    - speed issues:
      - too slow, not good for real-time situations
      - too complicate, not good available on hardware
      - it does its work per pixel instead of per object, but pixels greatly outnumber objects for most of today's scene (good for Z-buffer rendering though)
      - for a fixed-size bitmap, ray tracing may actually be faster for images of very complex scene
    - speedup techniques:
      - object hierarchy ray tracing: clumps objects into groups, then clumps groups into bigger groups (the bounding volumes); instead of checking a ray against every object in the group, the ray is checked against the bounding volume first; if it doesn't intersect the bounding volume, the there is no need to go any further with any object or subgroup
      - space subdivision ray tracing: the scene space is subdivided into little regions, and each region contains a list of the objects found within that region; rays are traced from their start points through each of the regions in their path, and intersection checks are performed only on objects listed in the regions; most schemes in use are adaptive, i.e., they subdivide more finely in areas that requires more detail; adaptive octree is a good choice
  - radiosity:
    - because ray tracing follows thin shafts of light (the rays), it deals well with light coming from point source; radiosity excels at diffuse reflection or secondary scatter

- in radiosity rendering, as in Z-buffer rendering, the scene is decomposed into a pile of polygons, and those polygons will ultimately be rendered using a Z buffer
- the radiosity algorithm first computes what portion of the light from each polygon reaches each others; this form factor (i.e., "coupling factor") is determined between every two polygons in the scene; then set final visible colors or each polygon by first setting the illumination onto each polygon to 0 (black), the only polygons that wouldn't appear black now are the light sources; next, the light from the bright polygons causes other polygons to be lit, this, in turn, shine on other polygon; this process of lighting up polygons continues until things settle down; finally the polygons are drawn by using Z-buffer renderer

- shading: the process of eventually coming up with pixel color values using RGB color space
  - direct evaluation (figuring it out the hard way): the Phong lighting model
    - knowing which way the surface is facing (the N vector), where the light is coming from (the L vector), and where we are looking from (E vector); from N and L, we can compute reflection (the R vector)
    - shading normal: the normal vector used for the purpose of computing the apparent color
    - geometric normal: the normal vector of the polygon we are drawing
    - facet shading: the geometric normal is also used as the shading normal; in this shading, each patch is planar, one shading normal applies to an entire patch
    - smooth shading: the original surface's normal vectors on each vertex were used as the shading normal; the vertex color values are smoothly blended in the interior of each polygon; the edge still shows the individual polygon because the silhouette edge is dependent only on the as-drawn geometry, not the shading normal vector
    - visual properties or surface properties
      - emissive color: the color the object appears in the dark, as if it were emitting light; it is not used often
      - diffuse color: as light hits a spot on an object, the diffuse color is reflected equally in all direction; it is the matte color (not shiny)
      - specular color: the color is reflected mostly around the reflection vector; it is the shinny highlight; the diffuse and specular surface properties are most often used together
        - specular exponent: controls how tightly the specular reflection is bunched around the reflection vector; a value of 0 is diffuse color, and a value of infinite is pure mirror reflection; the specular color contribution is proportional to the dot product of the E and R vectors raised to the power of the specular exponent
  - interpolation (fudging it for most of the pixels): compute the real object color only at the vertices of each polygon, then fill in the interior pixels based on the color at vertices
    - flat shading: the color values from the polygon vertices are averaged, and all the polygon's pixels are set to this fixed (flat) color value; very simple and even supported by low-end 2D display controllers, but the individual polygons are quite visible in the resulting picture
    - Gouraud or linear or bilinear (bi means 2D) shading: linear means the rate of change is held constant; linear shading is the most common shading technique used with Z-buffer rendering
    - Phong shading: instead of interpolating the color value for each pixel in a primitive, the shading normal vector is interpolated, then a separate apparent color computation is performed for each pixel; it has better quality result than bilinear interpolation
- Mach bands
  - whenever there is a sudden change in the color's rate of change, the optical illusion, Mach bands, is observed; the common solution is to use more and smaller polygons in modeling
- compositing: composite or blend different images into one (frequently done in moviemaking and television production)
  - overlay where not zero (the bargain-basement compositing method): image B is simply written over image A whenever image B doesn't have all 0 pixel values
  - alpha-buffered rendering
    - alpha (or transparency) value: opacity fraction, 0 means fully transparent (invisible), and the maximum value (255) means fully opaque
    - alpha buffer: the alpha values for all the pixels in an image or bitmap
    - alpha-buffered rendering: overlay a new image onto an existing one; only the overlay image have alpha values, the starting and resulting images are fully opaque; or all 3 images have alpha values
  - chroma keying: the resulting image is taken from the overlay whenever the overlay is (or isn't) a particular hue
    - blue screen: heavily used in television weather reports; blue was the chosen because it's the hue most opposite to skin color, and the reporter isn't wearing anything blue
- antialiasing:
  - aliasing (the jaggies): smooth lines in some digit images are jagged and look more like stair steps (individual pixels)
  - antialiasing: to avoid or reduce aliasing by setting pixels near the edge to some in some intermediate value
    - just ignore them because antialiasing requires lots of computation and is rarely supported in hardware
    - a simple way – sub-pixels: starts with a description of the image at a higher resolution than the final image by using multiple pixels for each final pixel (filtering from DSP)

        - box filtering: average all the sub-pixels within a final pixel; reasonably fast, but quality is medium
      - a better way - convolution: the bunch of sub-pixels need to be weighted (filter kernel or convolution kernel) and summed to produce each output pixel; more computation but better quality
  - temporal aliasing or strobing (aliasing in time): each frame of an animation is one fixed image throughout, any motion in the animation is really a sequence of frozen steps; to have each animation frame represent a small range of time, and fast-moving objects would then appear blurred
- texture mapping: replacing a parameter to the apparent color calculation with an external value; the user defines texture coordinates across the object during modeling; the texture coordinates are interpolated across each polygon instead of the final color values; texture mapping adds significant visual detail to the image without the need for modeling complex geometry
  - texils: the pixels in the texture map
  - other forms of texture mapping
    - bump mapping: texture-mapping small perturbations of the shading normal vector; it can make an object appear to be rougher than it really is geometrically
    - 3D texture or solid texture: the texture is a function instead of an explicit table of values (like an image)
- dithering: a technique for trading off spacial resolution to get more color (or gray-scale) resolution; the apparent color or gray-scale was increased at the expense of adding a dither pattern, or graininess, to the image; the noticeable graininess in the image is a disadvantage of dithering; the dither pattern can be a tile of any size or random patterns
  - a black-and-white dithering example by treating each clump of 2x2 pixels as a unit; set 0, 1, 2, 3, or 4 of the pixels to white and the rest to black; when view as an aggregate, each clump now can have five different gray levels

Animation

images are animated by showing many of them in rapid succession to give the illusion of a moving image; frame rate: NTSC video standard 30 frames/sec (due to interlacing, it actually shows a new half-image every 1/60 seconds), PAL or SECAM video standard 25 frames/sec, movie 24 frames/sec (to decrease flicker, each frame is shown twice)

- key frames (WYSIWYG method): the positions and other parameters of scene elements are exactly defined for certain key frames; the parameter are then interpolated to obtain values for the in-between frames (may not result in the exact motion you had in mind)
  - linear interpolation: changes each parameter by a fixed step each frame; produces very jerky motion that seems to suddenly start, stop, and change direction
  - cubic interpolation: avoids sudden starts, stops, or changes in direction
- parametric (rule-based method): provide rules (based on input parameters, particularly the animation time value) for how to determine each animated value at any point in time
- inverse kinematics and dynamics (physics Knurd method): by giving constraints (rules about how the system functions), the computer actually calculates some of the motion automatically; inverse means that the computer is figuring out what the objects must have done to achieve an end result you specify

Saving the Pixels and Image File Issues

bitmap are written to image files; standard image file formats make it easier to store, transmit, and exchange digital images

- common image file strategies
  - order: all the pixels for one horizontal scan line are written before going on to next scan line (scan-line or top-down order), and left-to-right order within a scan line
  - color: either palette (pseudo) color or true color
- compression: since the color value of most pixels is close to the color value of adjacent pixels, most images have a great deal of redundant information, which make them good candidate for compression algorithms
  - lossless compression: preserves all image data; you can compress and uncompress an image as many times as you want and still end up with the same original pixel values
    - runlength compression/coding: reduce storage for successive pixels that have exactly the same value; works reasonably well on computer-generated images, especially ones that have large areas of background with the same value horizontally; doesn't work well on scanned images because they often have slight differences between adjacent pixels
    - LZW (Lempel-Ziv and Welch) compression: identity and compressing repeating patterns in the data; the patterns grow each time they are referenced; LZW works well on a variety of images, including scanned and dithered images
  - lossy compression: lose some (least like to be noticed by human visual system) of the data during compression and thus achieve higher compression ratios; the color information is blurred more (yielding higher compression) than the intensity information; the higher the compression ratio, the more compression artifacts become visible in the image
    - joint photographic experts group (JPEG) for still digital color image: works on individual images to compress redundant information from pixel to pixel
    - motion picture experts group (MPEG) for motion picture: to compress redundant information from frame to frame; human are more tolerant of some kinds of artifacts in animated images than in still images; higher compression ratios than JPEG
- image file format
  - tag image file format (TIFF) for importing images into application

- graphics interchange format (GIF) for downloading images via a modem
- portable pixel map (PPM) is the only format OpenGL supported

Conference

- ACM/SIGGRAPH (the Association for Computing Machinery's Special Interest Group on Graphics) http://siggraph.org
- IEEE Computer Graphics and Applications
- Visual Computer