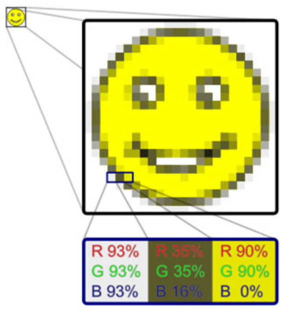
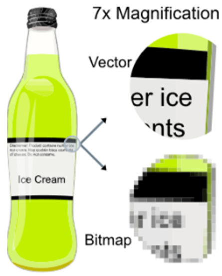
Chapter three

- Raster Images
- Raster Devices
- Images, Pixel and Geometry
- RGB Color
- Alpha Compositing
- Image Storage
- Summary

Raster Images

- **Raster** graphic or bitmap **image** is a dot matrix data structure that represents a generally rectangular grid of pixels, viewable via a bitmapped display (monitor), paper, or other display medium.
- There are other ways of describing images besides using arrays of pixels. A **vector image** is described by storing descriptions of shapes—areas of color bounded by lines or curves—with no reference to any particular pixel grid.



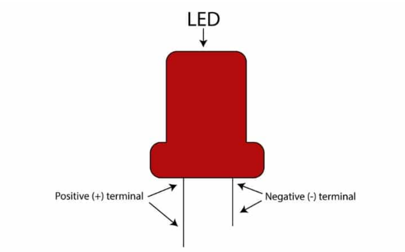


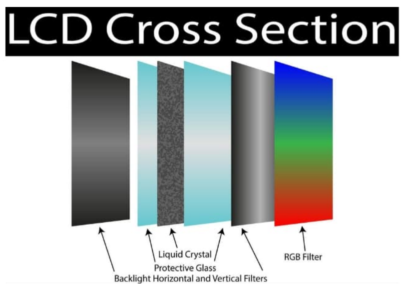
Raster Devices

- A few familiar raster devices can be categorized into a simple hierarchy:
  - Output
    - – Display
      - * Transmissive: liquid crystal display (LCD)
      - * Emissive: light-emitting diode (LED) display
    - – Hardcopy
      - * Binary: ink-jet printer
      - * Continuous tone: dye sublimation printer
  - Input
    - – 2D array sensor: digital camera
    - – 1D array sensor: flatbed scanner

Raster Devices-cont.

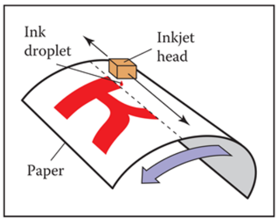
- Displays
  - Current displays, including televisions and digital cinematic projectors as well as displays and projectors for computers, are nearly universally based on fixed arrays of pixels. They can be separated into :
    - **emissive** displays, which use pixels that On Off On + – Anode Cathodes LEDs. The operation of a light-emitting diode (LED) display. directly emit controllable amounts of light, and
    - **transmissive displays**, in which the pixels themselves don’t emit light but instead vary the amount of light that they allow to pass through them.

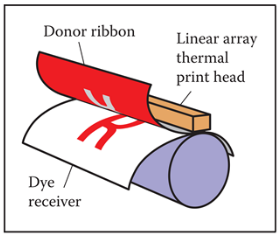




Raster Devices-cont.

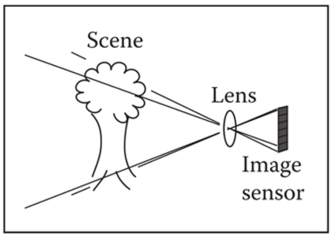
- Hardcopy Devices
  - In printing, pigments are distributed on paper or another medium so that when light reflects from the paper it forms the desired image. Printers are raster devices like displays, but many printers can only print binary images—pigment is either deposited or not at each grid position. An examples:
    - An **ink-jet printer** is an example of a device that forms a raster image by scanning. An ink-jet print head contains liquid ink carrying pigment, Paper Inkjet head Ink droplet. The operation of an ink-jet printer. which can be sprayed in very small drops under electronic control.
    - **Thermal dye printer** is an example of a continuous tone printing process, meaning that varying amounts of dye can be deposited at each pixel—it is not all-or-nothing like an ink-jet printer .A donor ribbon containing colored dye is pressed between the paper, or dye receiver, and a print head containing a linear array of heating elements, one for each column of pixels in the image.

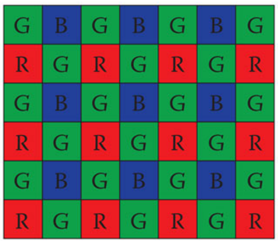




Raster Devices-cont.

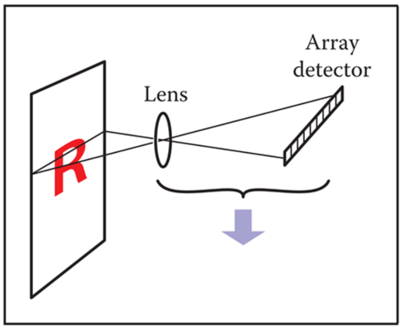
- Input Devices
  - Raster images have to come from somewhere, and any image that wasn't computed by some algorithm has to have been measured by some raster input device, most often a camera or scanner.
    - A **digital camera** is an example of a 2D array input device. The image sensor Lens Image sensor Scene. The operation of a digital camera. in a camera is a semiconductor device with a grid of light-sensitive pixels. Two common types of arrays are known as **CCDs** (charge-coupled devices) and **CMOS** (complimentary metal–oxide–semiconductor) image sensors.





Raster Devices-cont.

- Input Devices-cont.
  - A **flatbed scanner** also measures red, green, and blue values for each of a grid of pixels, but like a thermal dye transfer printer it uses a 1D array that sweeps across the page being scanned, making many measurements per second.



Images, Pixels, and Geometry

- In the physical world, images are functions defined over two-dimensional areas—almost always **rectangles**. So we can abstract an image as a function

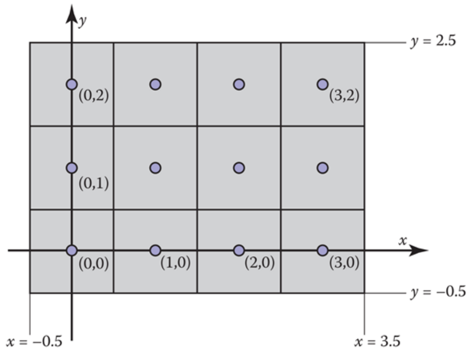
$$I(x, y): R \rightarrow V,$$

where $R \subset \mathbb{R}^2$ is a rectangular area and V is the set of possible pixel values.

- The simplest case is an idealized grayscale image where each point in the rectangle has just a brightness (no color), and we can say $V = \mathbb{R}$ (the nonnegative reals).
- An idealized color image, with red, green, and blue values at each pixel, has $V = (\mathbb{R}^+)^3$

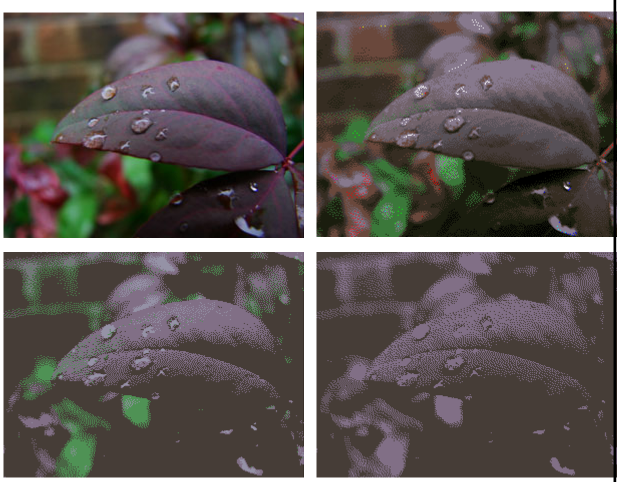
Images, Pixels, and Geometry-cont.

- A pixel from a camera or scanner is a measurement of the average color of the image over some small area around the pixel.
- A display pixel, with its red, green, and blue subpixels, is designed so that the average color of the image over the face of the pixel is controlled by the corresponding pixel value in the raster image.



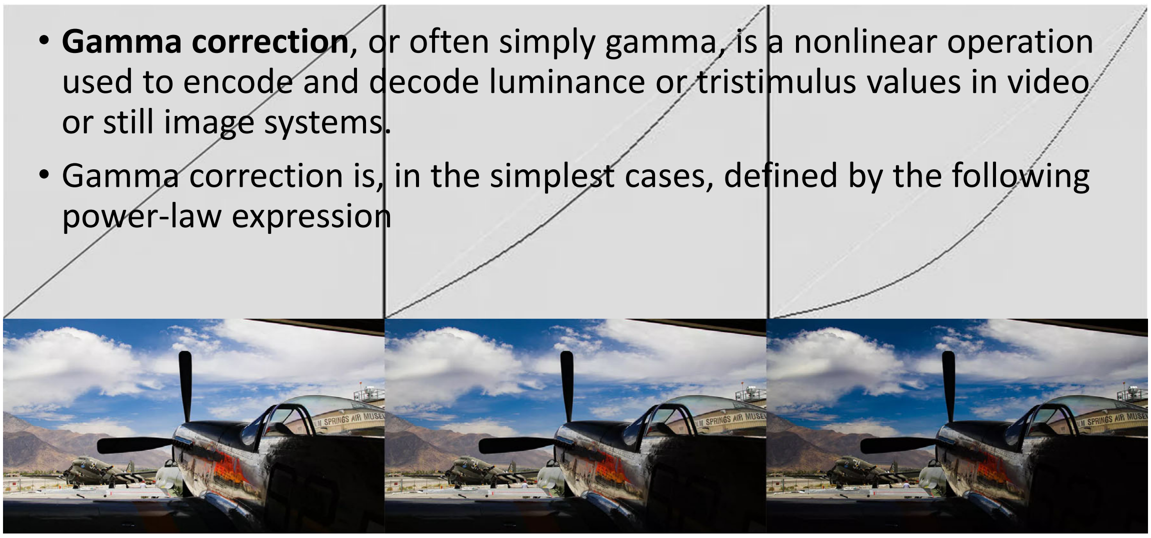
Images, Pixels, and Geometry-cont.

- The values of pixels in terms of real numbers, representing intensity (possibly separately for red, green, and blue).
- Images should be arrays of floating-point numbers, with either one (for **grayscale**, or **black** and **white**, images) or three (for RGB color images)
- 32-bit floating point numbers stored per pixel. This format is sometimes used, when its precision and range of values are needed, but images have a lot of pixels and memory and bandwidth for storing and transmitting images are invariably scarce.



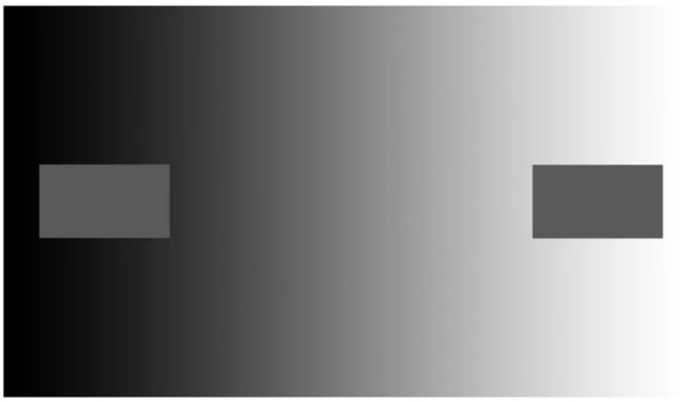
Gamma correction

- **Gamma correction**, or often simply gamma, is a nonlinear operation used to encode and decode luminance or tristimulus values in video or still image systems.
- Gamma correction is, in the simplest cases, defined by the following power-law expression



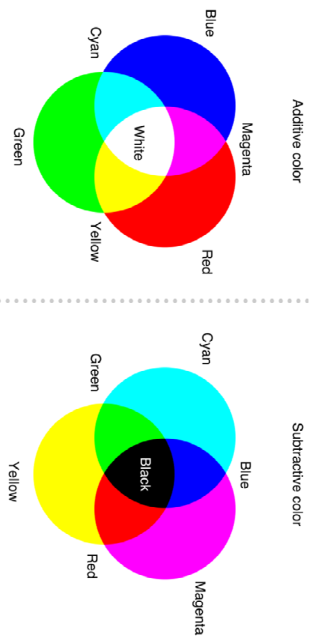
Gamma correction-cont.

- Which rectangle is darker and which is lighter?



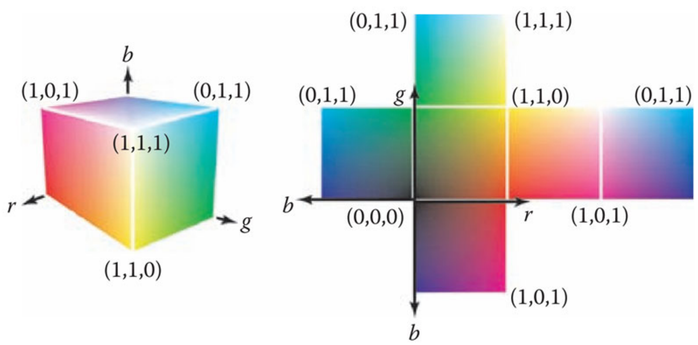
RGB Color

- Most computer graphics images are defined in terms of red-green-blue (RGB) color. RGB color is a simple space that allows straightforward conversion to the more familiar additive mixing that happens in displays. the controls for most computer screens.
- In grade school you probably learned that the primaries are red, yellow, and blue, and that, e.g., yellow + blue = green. This is subtractive color mixing, which is fundamentally different from



RGB Color-cont.

- The RGB color cube in 3D and its faces unfolded. Any RGB color is a point in the cube.



Alpha Compositing

- Often we would like to only partially overwrite the contents of a pixel. A common example of this occurs in compositing, where we have a background and want to insert a foreground image over it.
- For **opaque** pixels in the foreground, we just replace the background pixel. For **entirely transparent** foreground pixels, we do not change the background pixel. For **partially transparent** pixels, some care must be taken:

$$\boldsymbol{c} = \alpha \boldsymbol{c}_f + (1 - \alpha)\boldsymbol{c}_b$$

Alpha Compositing-cont.

- An example of compositing using the previous Equation . The **foreground** image is in effect cropped by the α channel before being put on top of the **background** image. The resulting composite is shown on the bottom.

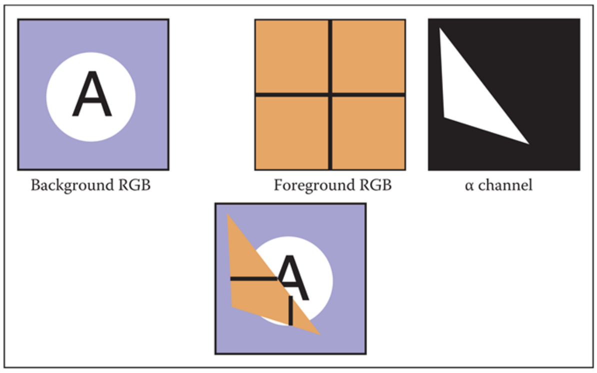


Image Storage

- Most RGB image formats use **eight** bits for each of the red, green, and blue channels. This results in approximately three megabytes of raw information for a single million-pixel image.
- To reduce the storage requirement, most image formats allow for some kind of compression. At a high level, such compression is either **lossless** or **lossy**.

Image Storage-cont.

- Popular image Storage formats include:
  - **jpeg**. This lossy format compresses image blocks based on thresholds in the human visual system. This format works well for natural images.
  - **tiff**. This format is most commonly used to hold binary images or losslessly compressed 8- or 16-bit RGB although many other options exist.
  - **ppm**. This very simple lossless, uncompressed format is most often used for 8-bit RGB images although many options exist.
  - **png**. This is a set of lossless formats with a good set of open source management tools.
  - **svg.** Scalable vector graphics for store vectors data

Summary

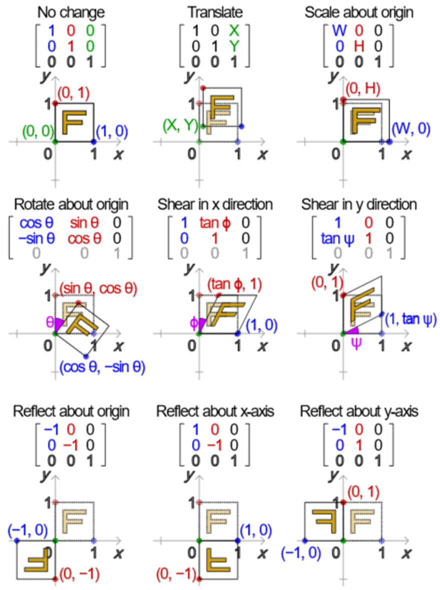
- Raster Images are based on each color pixel while vector are based on math equations.
- Raster Devices are into two main categories: Output and Input.
- Images are set of pixel arrays.
- Most computer graphics images are defined in terms of red-green-blue (RGB) color.
- Alpha Compositing is about most important piece of information needed to blend a foreground object over a background object.
- Image Storage can differ form one extention to another depending on size and compression.

Chapter four

- Transformation Matrices
- 2D Linear Transformations
- Scaling
- Shearing
- Rotation
- Reflection
- 3D Linear Transformations
- Summary

Transformation Matrices

- Geometric transformations like **rotation**, **translation**, **scaling**, and **projection** can be accomplished with **matrix multiplication**, and the **transformation matrices** used to do this are the subject of this chapter.



2D Linear Transformations

- We can use a 2 × 2 matrix to change, or transform, a 2D vector:

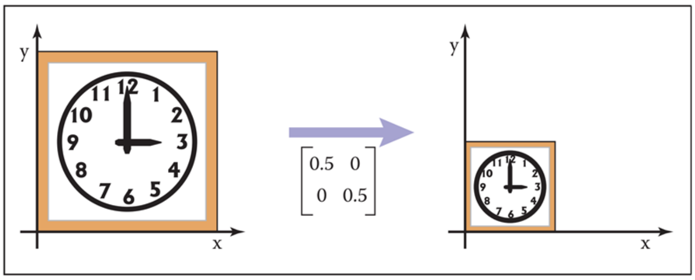
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{bmatrix}.$$

- By this simple formula we can achieve a variety of useful transformations, depending on what we put in the entries of the matrix, as will be discussed in the following sections.

Scaling

- The most basic transform is a scale along the coordinate axes. This transform can change length and possibly direction:

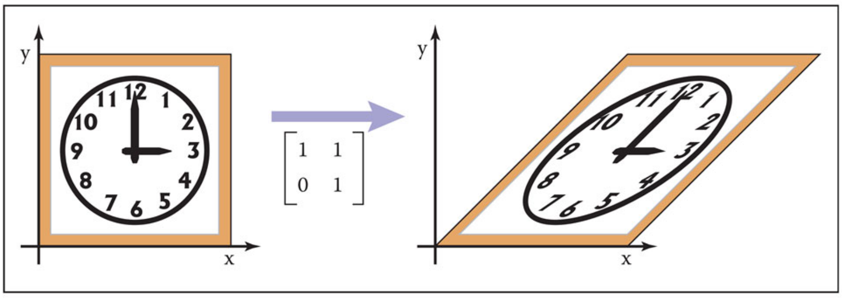
$$\text{scale}(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}.$$



Shearing

- A **shear** is something that pushes things sideways. The horizontal and vertical shear matrices are

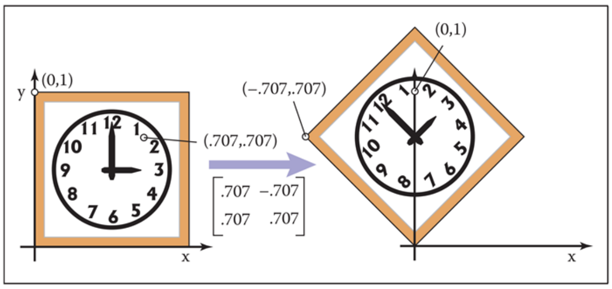
$$\text{shear-x}(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$



Rotation

- Suppose we want to rotate a vector a by an angle ϕ counterclockwise to get vector b

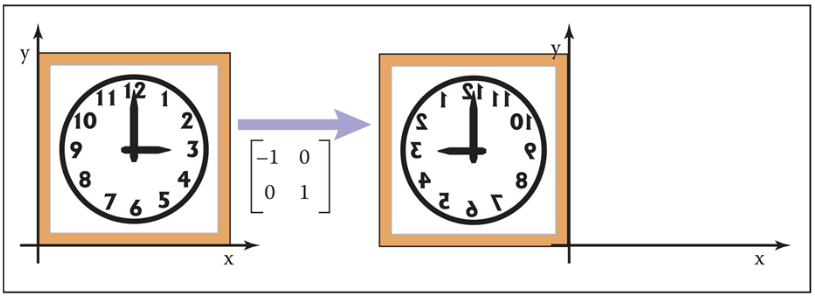
$$\text{rotate}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$



Reflection

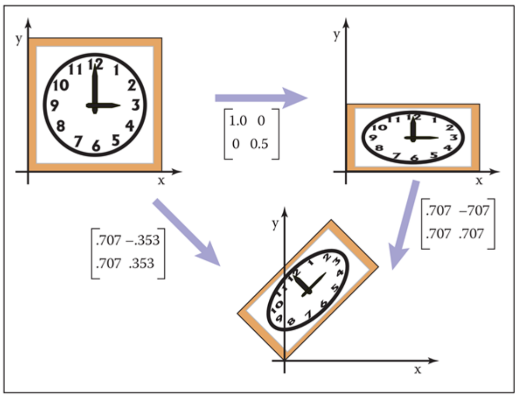
- We can reflect a vector across either of the coordinate axes by using a scale with one negative scale factor

$$\text{reflect-x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$



Multiple Transformation

- It is common for graphics programs to apply more than one transformation to an object.
- For example, we might want to first apply a scale S, and then a rotation R.



3D Linear Transformation

- The linear 3D transforms are an extension of the 2D transforms. For example, a scale along Cartesian axes is

$$\text{scale}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}.$$

Summery

- In linear algebra, linear transformations can be represented by matrices.
- The matrix associated with a stretch by a factor k along the x-axis is given by $\begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$
- For rotation by an angle θ clockwise about the origin the functional form is

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Summery-cont.

- Shearing (visually similar to slanting) $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
- Reflection in x-axis and y-axis

$$\text{reflect-y} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{reflect-x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

- 3D Transformation is an extension to 2D matrix.