Computação Visual e Multimédia

10504: Mestrado em Engenharia Informática

Chap. 2 — Light, Color, and Images

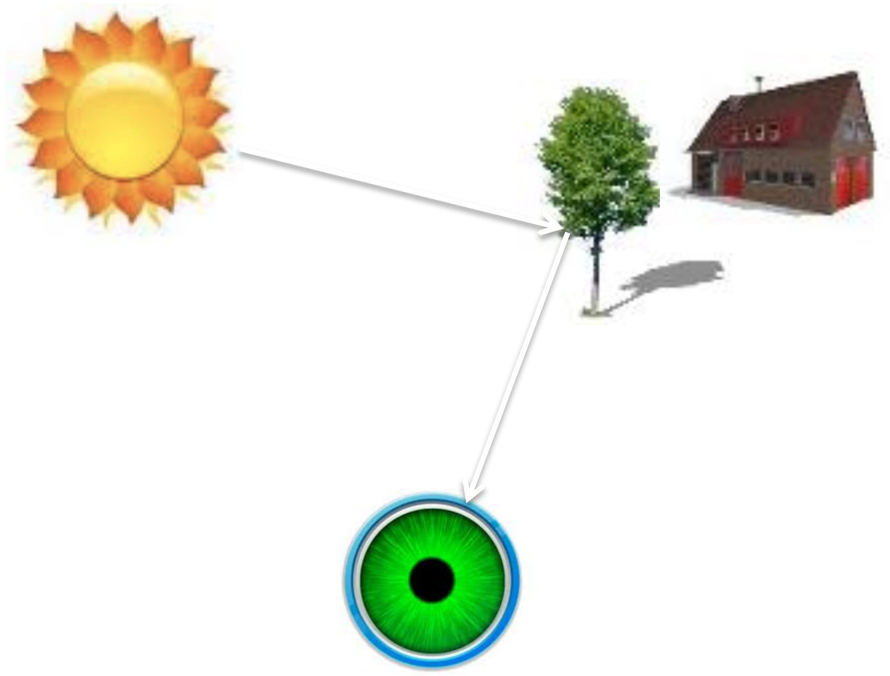
Module Introduction

Outline

....

- Light & Color
- Image Formation
- Image Digitization
- Image Representation
- Image File Formats
- Programming with images.

Eye's image formation: overview



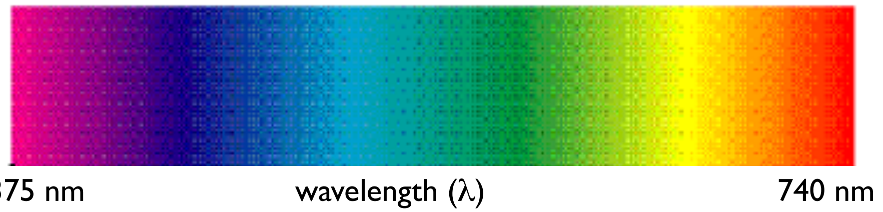
Involved entities:

- Light source
- 3D scene (with objects)
- Eye/retina

Steps of image formation:

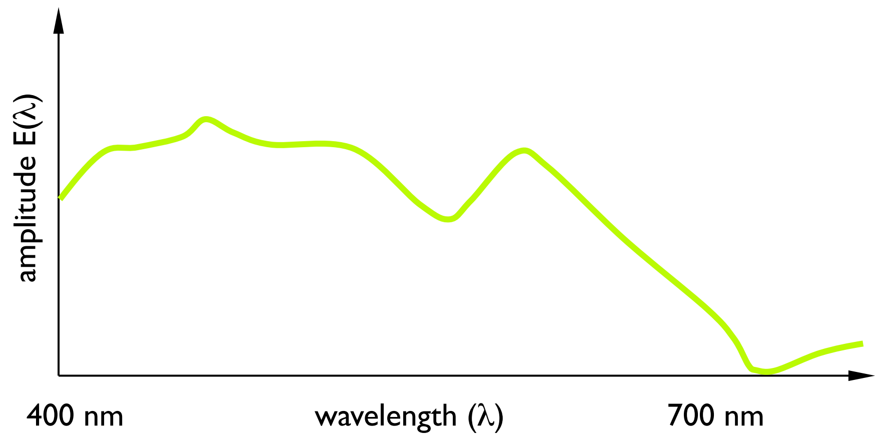
- The scene is illuminated by a light source.
- The scene reflects light towards the eye.
- The eye/retina senses it via bio-photoreceptors: rods and cones.

What is light?

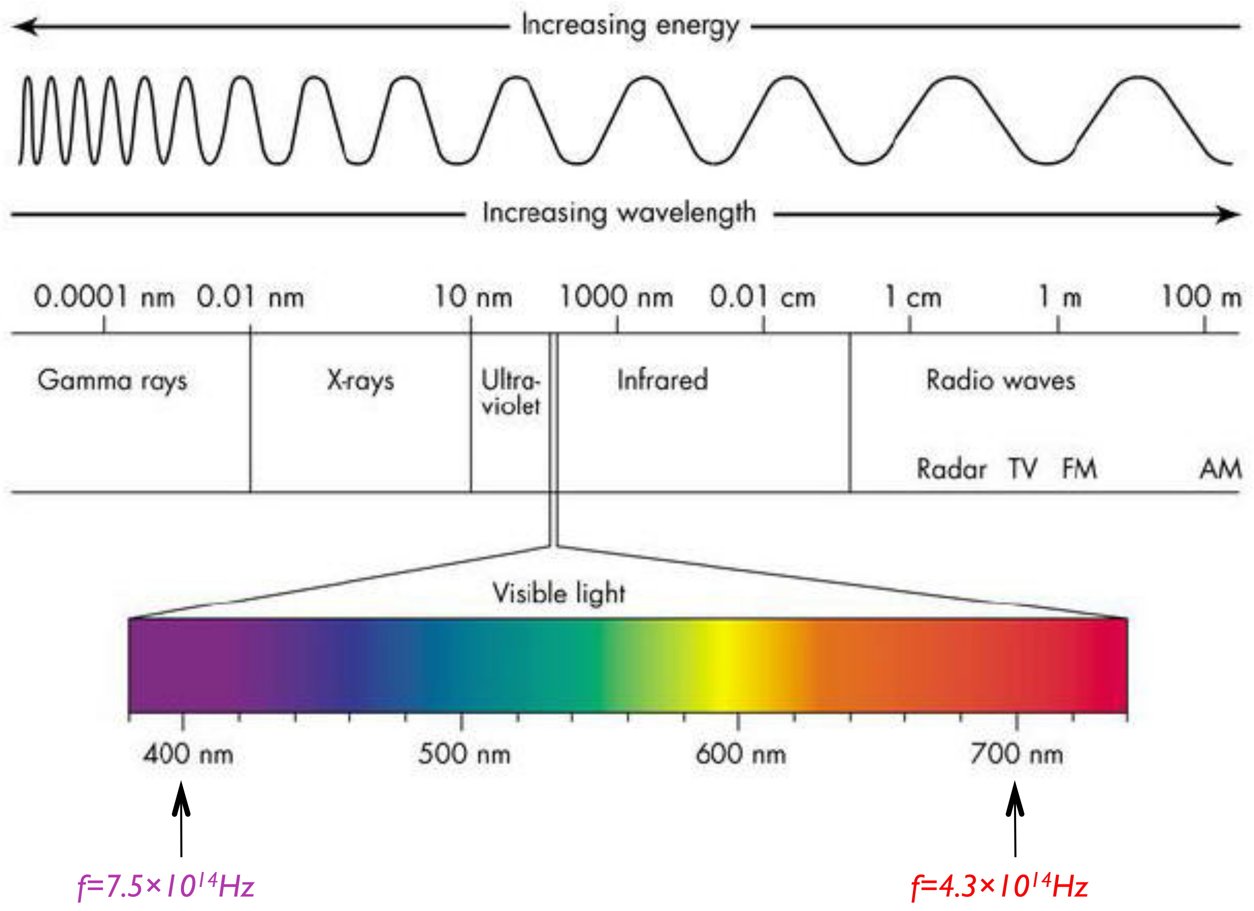


Definition:

- Light is an electromagnetic wave.
- It is the **visible** portion of the electromagnetic (EM) spectrum
- It occurs between wavelengths of approximately 400 and 700 nanometers.
- Spectral color bands: magenta, blue, green, yellow, orange, red
- Distinct colors correpond to distinct wavelenghts



Electromagnetic spectrum



Light velocity: $c = 3\times10^8\ m/sec = \lambda.f$

Light energy: $E = h.f$

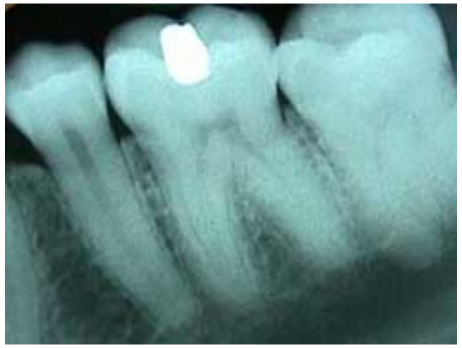
where h is the Planck constant

Short wavelengths

http://science.hq.nasa.gov/kids/imagers/ems/xrays.html

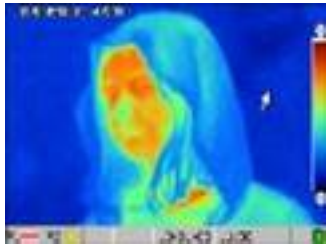
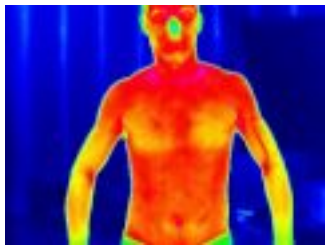
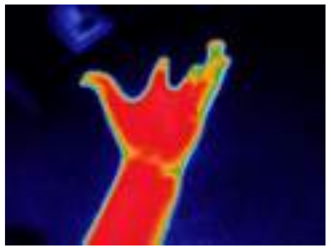
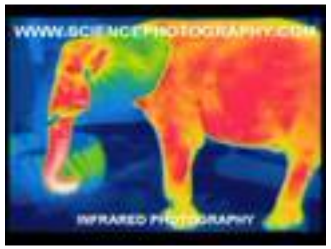
Note that:

- Different wavelengths of radiation have different properties.
- The **x-ray** region of the spectrum, it carries sufficient energy to penetrate a significant volume or material
- X-ray technology has allowed us to see inside the human body since 1895 (Wilhelm Roentgen).



Because your bones and teeth are dense and absorb more X-rays than your skin does, silhouettes of your bones or teeth are left on the X-ray film, while your skin appears transparent. Metal absorbs even more X-rays - can you see the filling in the image of the tooth?

Long wavelengths

Source | http://www.youtube.com/watch?v=_WP2XwBhmAk

On the other hand:

- Copious quantities of **infrared** (IR) radiation are emitted from warm objects (e.g., locate people in total darkness).
- "Synthetic aperture radar" (SAR) imaging techniques use an artificially generated source of **microwaves** to probe a scene.
- SAR is unaffected by weather conditions and clouds (e.g., has provided us images of the surface of Venus).

Synthetic Aperture Radar image of Washington, D.C.

Source | http://www.sandia.gov/RADAR/sar.html

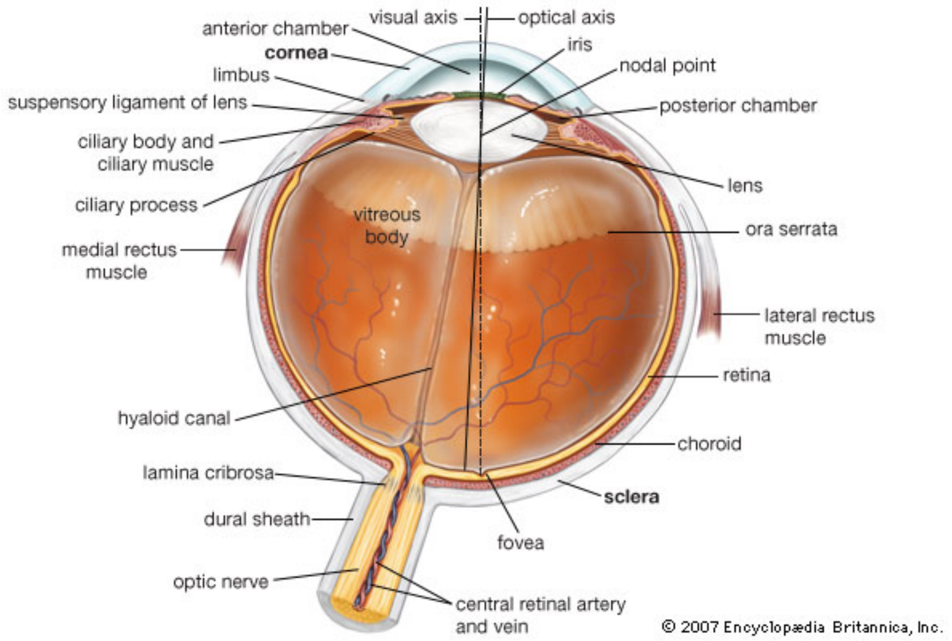
Image formation in the eye

http://faculty.washington.edu/chudler/eyecol.html

How does it work?:

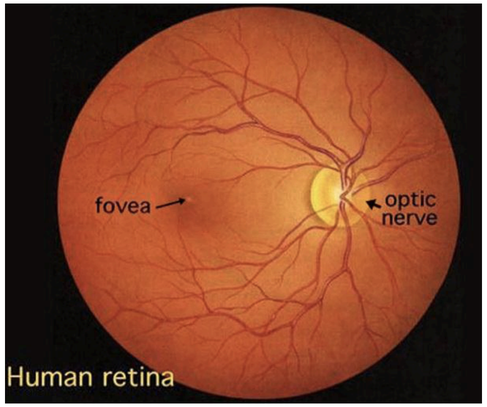
– Light enters the eye through the transparent cornea, passes through the aqueous humor, the lens, and the vitreous humor, where it finally forms an image on the retina.

– The retina itself is a complex of photoreceptors: rods and cones. When stimulated by light, they produce electrical signals that are transmitted to the brain via the optic nerve.



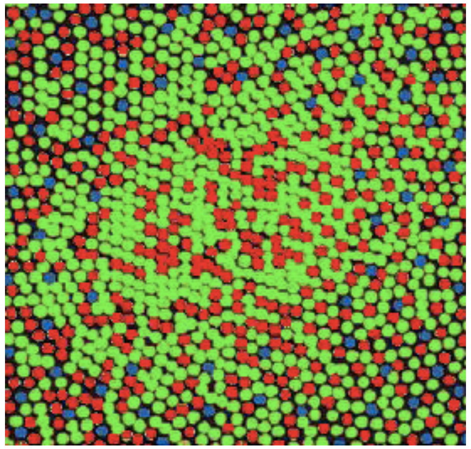
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT1/node2.html

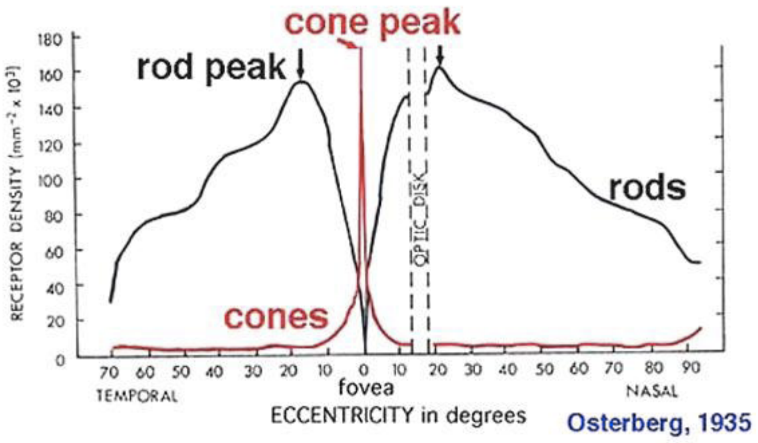
Eye's light photosensors



How does it work?:

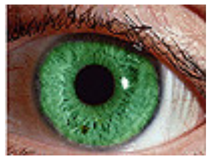
- The human fovea has a diameter of about 1.0 mm with a high concentration of cone photoreceptors.
- The center of the fovea is the foveola – about 0.2 mm in diameter – where only cone photoreceptors are present and there are virtually no rods. The central fovea consists of very compact cones, thinner and more rod-like in appearance than cones elsewhere.
- Starting at the outskirts of the fovea, however, rods gradually appear, and the absolute density of cone receptors progressively decreases.





http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html

Spectral response of the eye photosensors



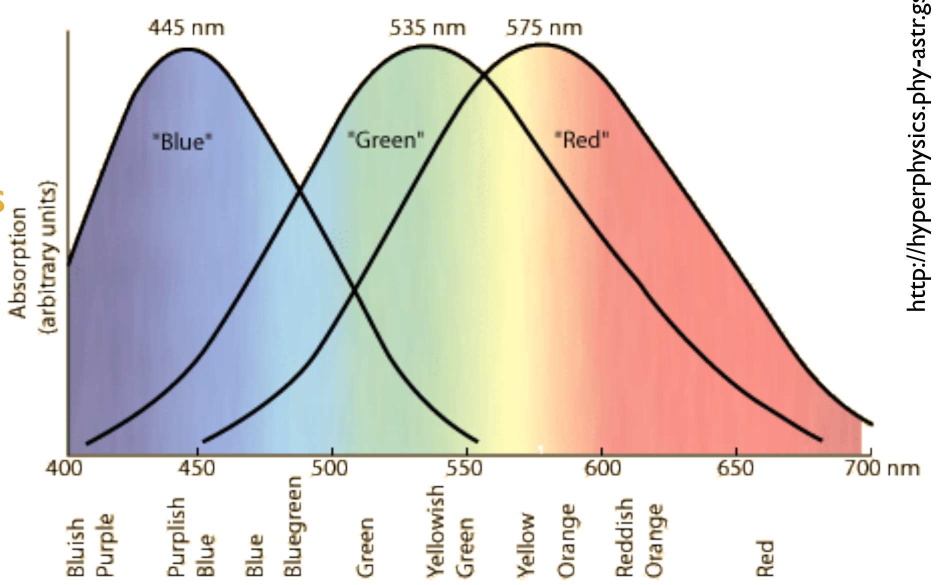
Cones:

- <u>Color-sensitive</u> (6 to 7 millions).
- By population, about 64% of the cones are red-sensitive, about 32% green sensitive, and about 2% are blue sensitive.
- They measure the light frequency (color).

Rods:

- <u>Luminosity-sensitive</u> (75 to 150 millions).
- They measure the light intensity (luminosity).

$S_3(\lambda)$ $S_2(\lambda)$ $S_1(\lambda)$



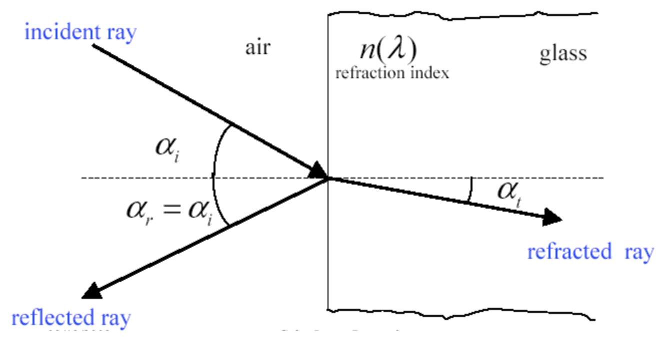
http://hyperphysics.phy-astr.gsu.edu/hbase/vision/colcon.html

$S_1(\lambda)$ is the absorption response curve of red cones

$S_2(\lambda)$ is the absorption response curve of green cones

$S_3(\lambda)$ is the absorption response curve of blue cones

3D scenes and objects



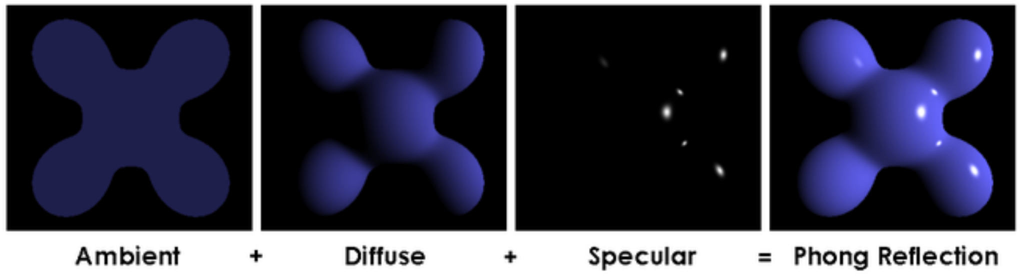
http://www.hunterlab.com/pdf/color.pdf

Light source:

- A light source usually emits light that appears to be white. When light is dispersed by a prism it is seen to be made up of all visible wavelengths.

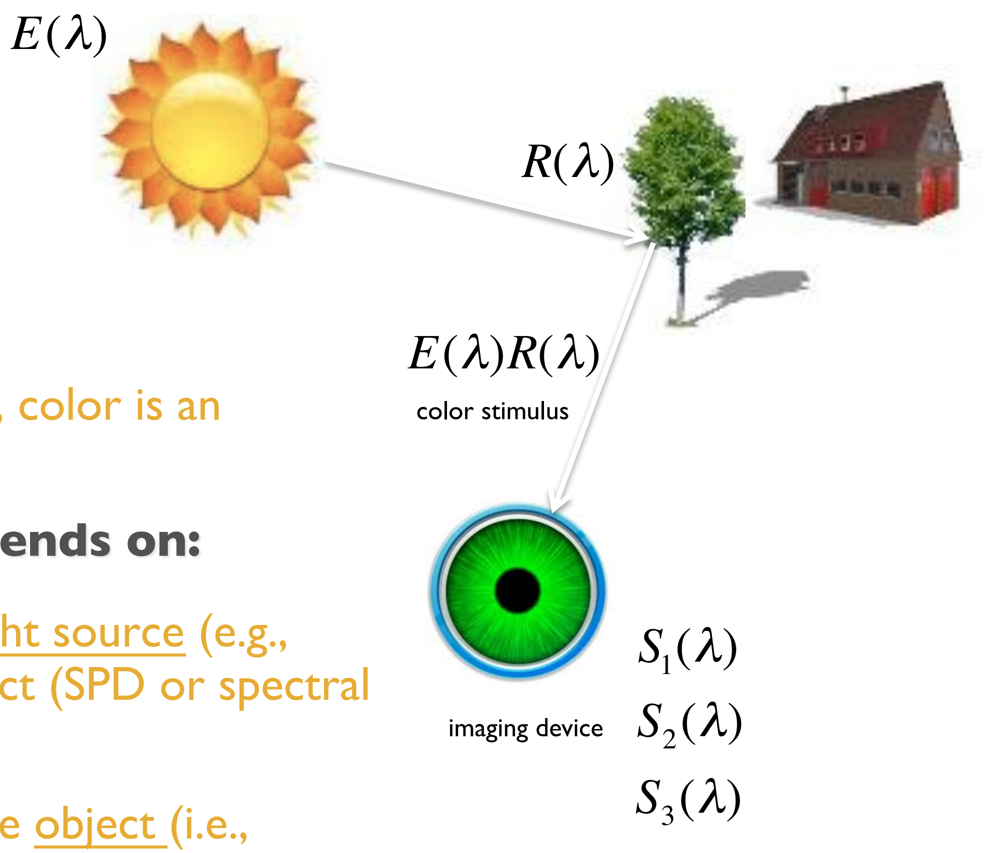
Object:

- *Objects change light.*
- Colorants such as pigments or dyes, in the object, selectively absorb some wavelengths of incident light while reflecting or transmitting others (Snell's law).
- The amount of reflected or transmitted light at each wavelength can be quantified. This is a spectral curve of the object's color characteristics.



http://en.wikipedia.org/wiki/Phong_shading

What is color?



Definition:

– Similar to a texture or shape, color is an attribute of an object.

As observed before, color depends on:

– The spectral properties of light source (e.g., sun) that illuminates the object (SPD or spectral power distribution E(λ)).

– The spectral properties of the object (i.e., reflection R(λ)).

– The spectral properties of the photoreceptors of the imaging device (e.g., eye or camera).

Thus:

– Color is defined by 3 values: $$\rho_n = \int_{\substack{visible\\spectrum}} E(\lambda)R(\lambda)S_i(\lambda)d\lambda, \qquad n = 1,2,3$$

http://www.hunterlab.com/pdf/color.pdf

http://en.wikipedia.org/wiki/Phong_shading

Camera's image formation: overview

Camera (artificial eye):

- Camera replaces eye in image formation.
- The role of the camera in machine vision is analogous to that of the eye in biological systems.

Steps of image formation:

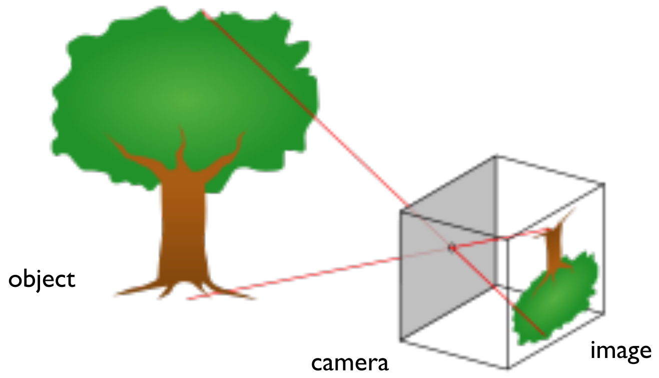
- The scene is illuminated by a single source.
- The scene reflects radiation towards the camera.
- The camera senses it via chemicals on film.

Pinhole camera

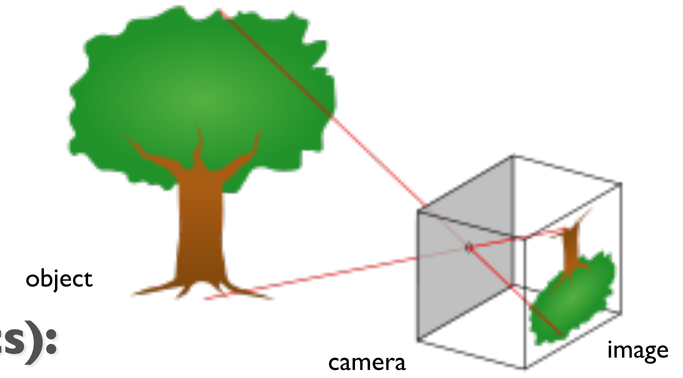
How does it work?:

- A pinhole camera is a simple camera without a lens and with a single small aperture – effectively a light-proof box with a small hole in one side.
- Light from a scene passes through this single point, called “pinhole”, and projects an inverted image on the opposite side of the box.
- This is the simplest device to form an image of a 3D scene on a 2D surface
- The human eye in bright light acts similarly, as do cameras using small apertures.

http://en.wikipedia.org/wiki/Pinhole_camera



Simplified pinhole camera

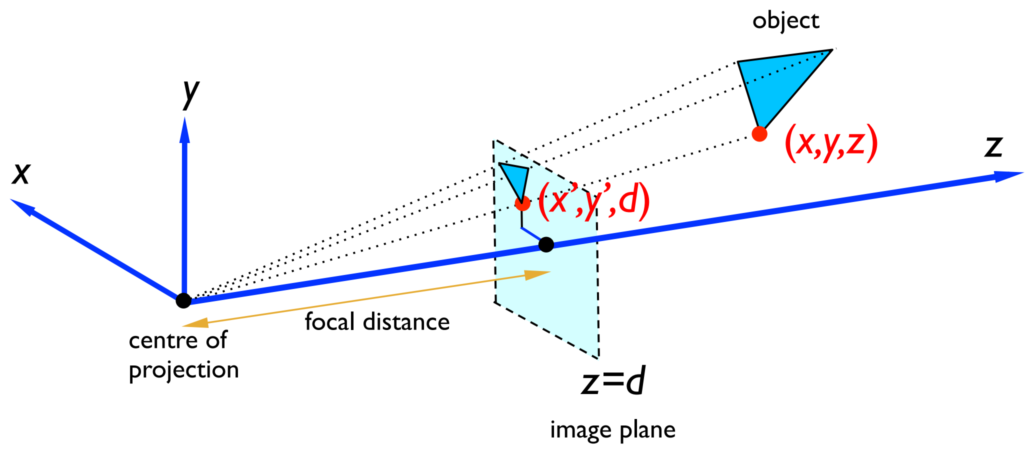


http://en.wikipedia.org/wiki/Pinhole_camera

Perspective geometry (computer graphics):

- Perspective geometry provides a 3D to 2D mapping without inverting the image.
- That is, the perspective geometry provides a simplified model of the pinhole camera by placing the image plane between the focal point of the camera and the object.

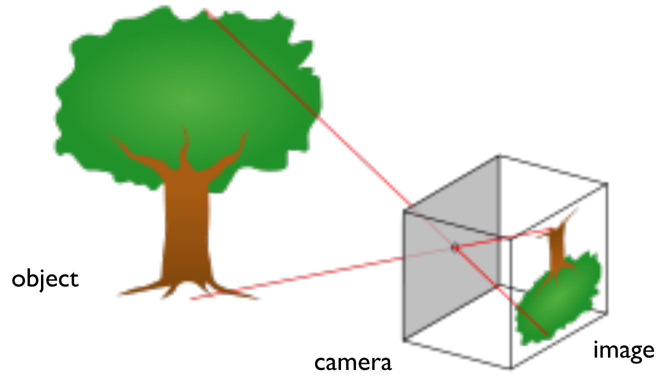
$$\frac{x'}{x} = \frac{y'}{y} = \frac{z'}{z} = \frac{d}{z}$$



$$\begin{cases} x' = x\dfrac{d}{z} \\ y' = y\dfrac{d}{z} \\ z' = d \end{cases}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Simple lens model



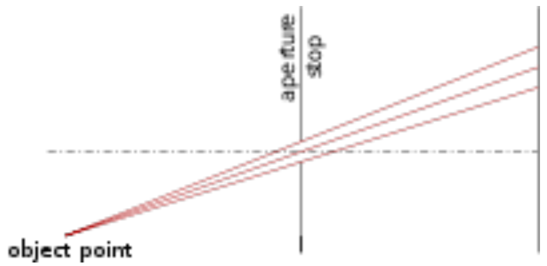
Pinhole camera:

– A pinhole camera is not able to produce real images.

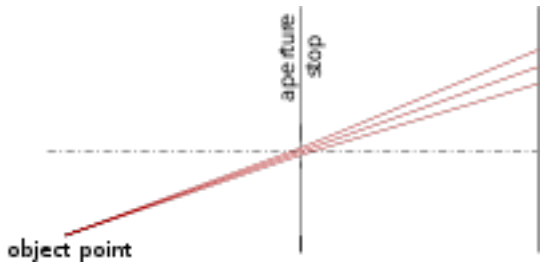
Lens:

– A lens is necessary to get a sharp focus of the image.

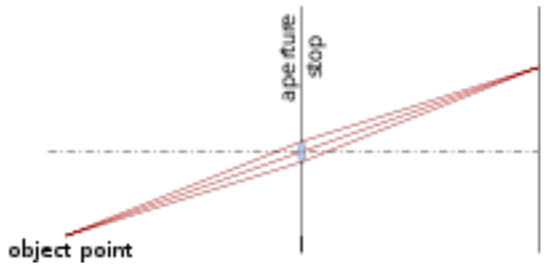
– A lens gathers light over an finite aperture (pinhole).



With a large pinhole, the image spot is large, resulting in a blurry image.



With a small pinhole, light is reduced and diffraction prevents the image spot from getting arbitrarily small.



With a simple lens, much more light can be brought into sharp focus.

http://en.wikipedia.org/wiki/Camera_lens

Simple lens model (cont'd)

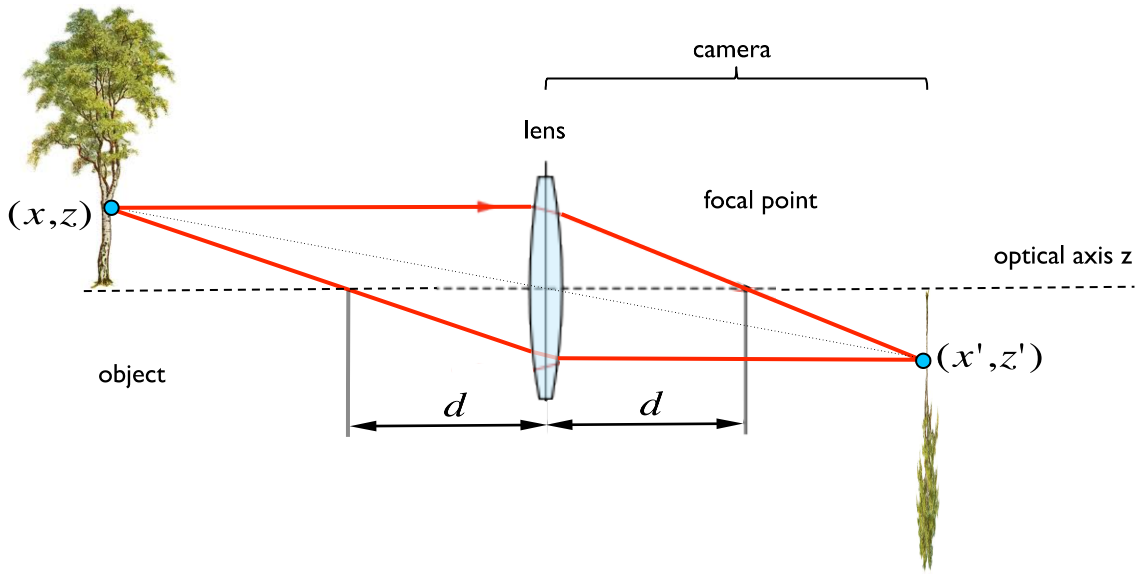
Lens:

- A lens is necessary to get a sharp focus of the image.
- By similar triangles in front of the lens, we have: $\frac{x'}{d} = \frac{x}{z-d}$
- By similar triangles behind the lens, we have: $\frac{x}{d} = \frac{x'}{z'-d}$

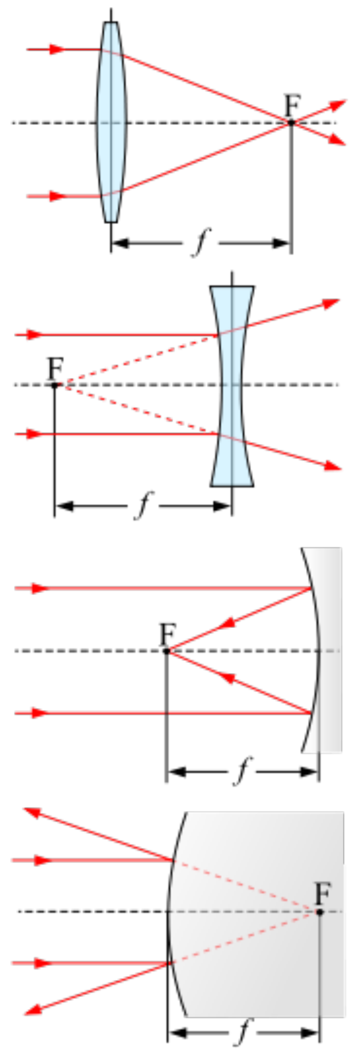
Lens formula:

$$\frac{1}{z} + \frac{1}{z'} = \frac{1}{d}$$

- This formula is obtained from those two two equations.

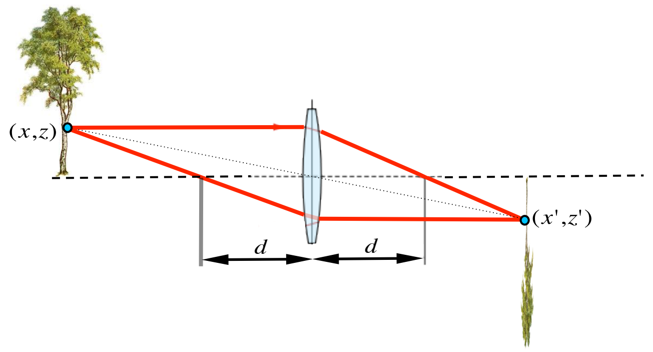


The focal point *F* and focal length *f* of a positive (convex) lens, a negative (concave) lens, a concave mirror, and a convex mirror.



http://en.wikipedia.org/wiki/Focal_length

Simple lens model (cont'd)



Lens formula: $\frac{1}{z}+\frac{1}{z'}=\frac{1}{d}$

– It can be re-written as: $z'=\frac{zd}{z-d}$

– Similarly, we get $x'=\frac{xd}{x-d}$ and $y'=\frac{yd}{y-d}$

Projective transform:

– In matrix form:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & -d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

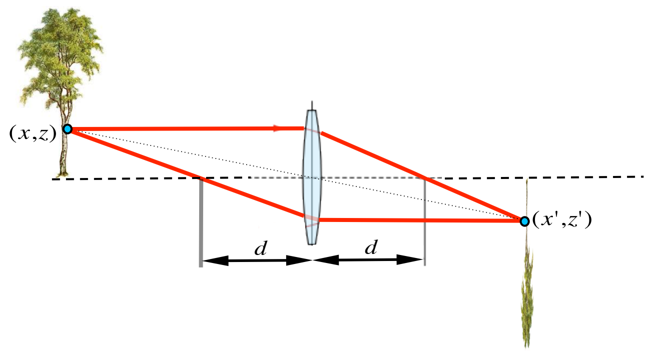
– This means that surface points (x,y,z) and their conjugate image points (x',y',z') are related by a projective transform.

– Note that the conjugate transform is its own inverse i.e. the conjugate point of the conjugate point is just the original point, since:

$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & -d \end{bmatrix} \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & -d \end{bmatrix} \equiv \begin{bmatrix} d^2 & 0 & 0 & 0 \\ 0 & d^2 & 0 & 0 \\ 0 & 0 & d^2 & 0 \\ 0 & 0 & 0 & d^2 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

http://www.cim.mcgill.ca/~langer/557/lecture21.pdf

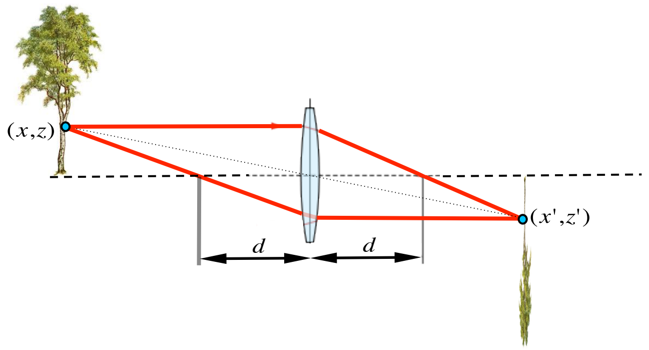
Camera optics



Camera Optics:

- In practice, the aperture must be larger to admit more light
- Lenses are placed to in the aperture to **focus** the bundle of rays from each scene point onto the corresponding point in the image plane
- Optical parameters of the lens
  - lens type, focal length, field of view
- Photometric parameters
  - type, intensity, and direction of illumination
  - reflectance properties of the viewed surfaces
- Geometric parameters
  - type of projections
  - position and orientation of camera in space
  - perspective distortions introduced by the imaging process

Camera calibration



Camera calibration:

- It is a process that allows us to deduce 3D geometric information from an image, one must determine the parameters that relate the position of a point in a scene to its position in the image.
- This is a cumbersome process of estimating the intrinsic and extrinsic parameters of a camera.
- There are 4 intrinsic camera parameters: two are for the position of the origin of the image coordinate frame, and two are for the scale factors of the axes of this frame.
- There are 6 extrinsic camera parameters: three are for the position of the center of projection, and three are for the orientation of the image plane coordinate frame.
- However, recent advances in computer vision indicate that we might be able to eliminate this process altogether.

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT1/node2.html

Range cameras

OPTEX 3D range image camera

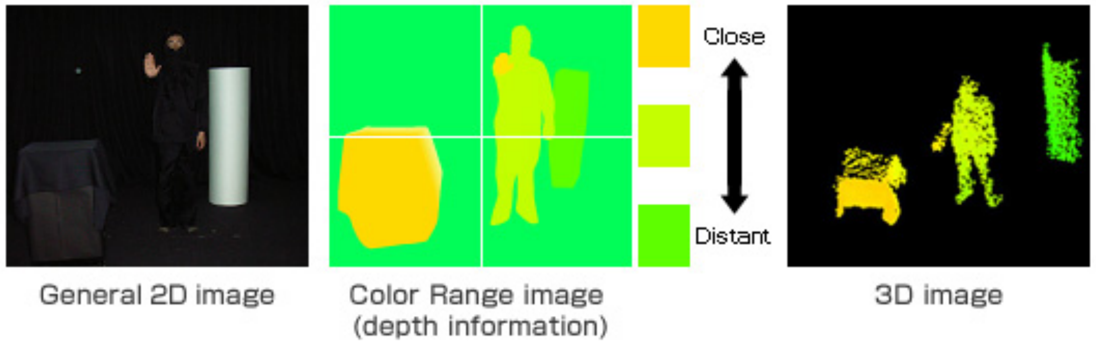
Definition:

– An array of pixels (sometimes color) plus an array of distances to the objects in the scene (depth information for each pixel) at framerate = depth images.

Other names:

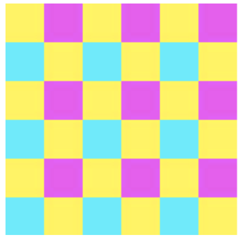
– Depth cameras, flash lidar, time-of-flight (ToF) camera, and RGB-D camera. The underlying sensing mechanisms are equally varied: range-gated ToF, RF-modulated ToF, pulsed-light ToF, and projected-light stereo.

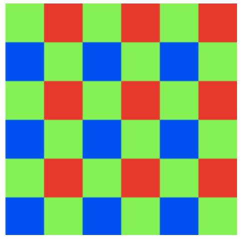
– PrimeSense supplies the 3D sensing technology to Project Natal for the XBox 360.



http://www.optex.co.jp/e/product/3d.html

CCD (Charged-Coupled Device) cameras


CMY Bayer Pattern


RGB Bayer Pattern

http://www.modernimaging.com/Kodak_DCS-620x_Technology.htm

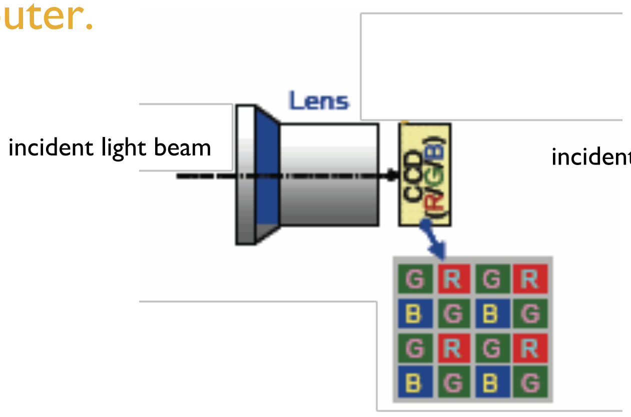
Working principle:

– Tiny solid state cells (photosensors) convert light energy into electrical charge.

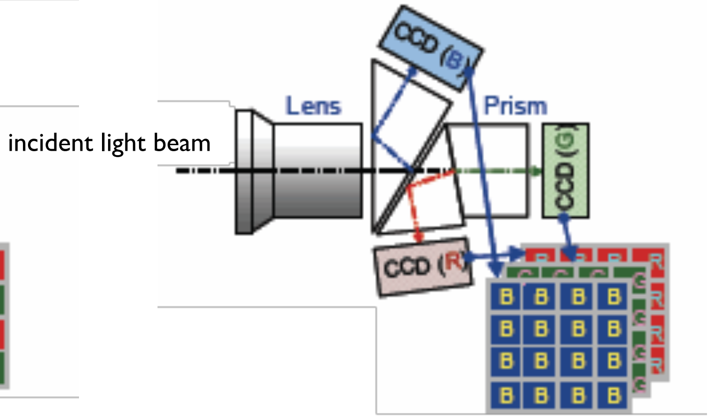
– An analog-to-digital converter (ADC) then turns each pixel's value into a digital value by measuring the amount of charge at each photosite and converting that measurement to binary form.

Image:

– The image plane acts as a digital memory that can be read row by row by a computer.



1 chip CCD camera

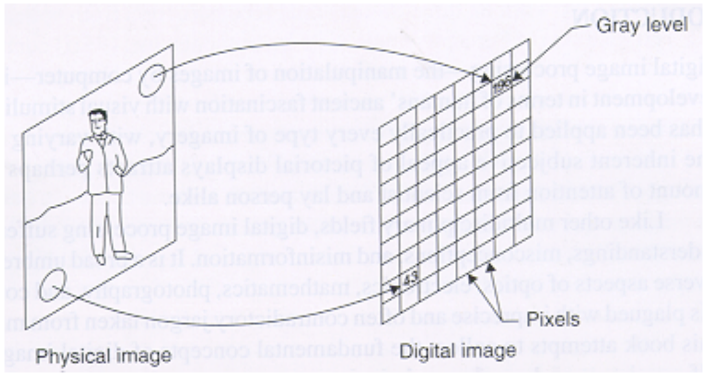


3 chips CCD camera

http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera2.htm

http://www.nec.com/global/solutions/security/technologies/3chip_em_ccd.html

Image digitization



http://css.engineering.uiowa.edu/~dip/LECTURE/ImageProperties2.html

Sampling:

- An image captured by a sensor is expressed as a continuous function f(x,y) of two co-ordinates in the plane.
- Image digitization means that the function f(x,y) is sampled into a matrix with M rows and N columns.
- Sample points are called **pixels**.

Quantization:

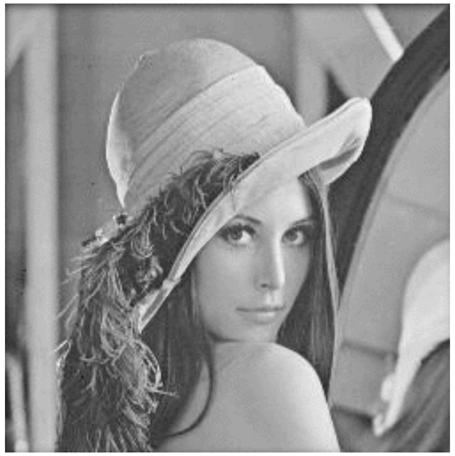
- The image quantization assigns to each continuous sample an integer value.
- The continuous range of the image function f(x,y) is split into K intervals (recall the levels in the previous chapter).
- Quantization = **number of bits per pixel**.

Overall:

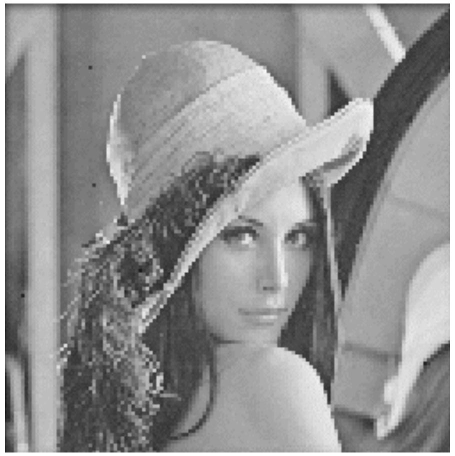
- The finer the sampling (i.e., the larger M and N) and quantization (the larger K) the better the approximation of the continuous image function f(x,y).

Image sampling (example)

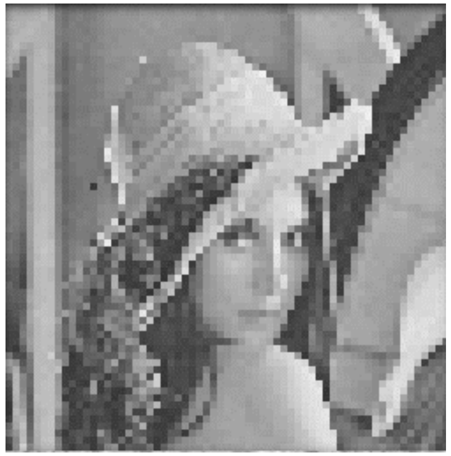
original image



sampled by a factor of 2



sampled by a factor of 4



sampled by a factor of 8

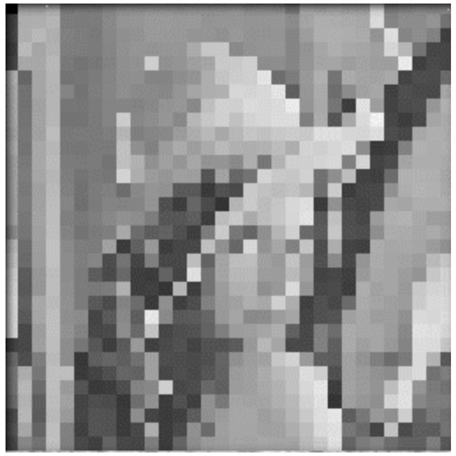
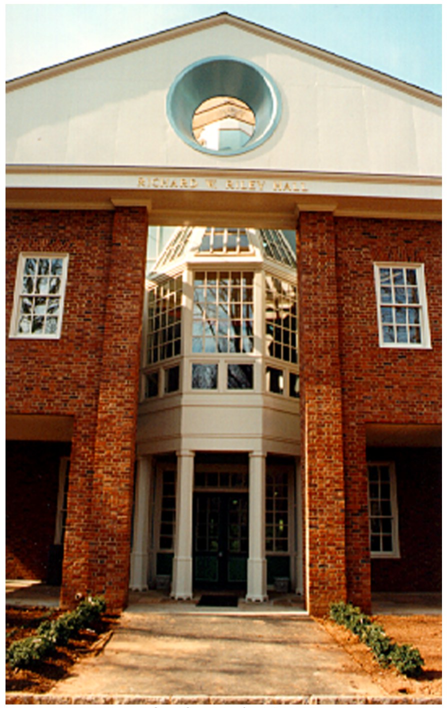


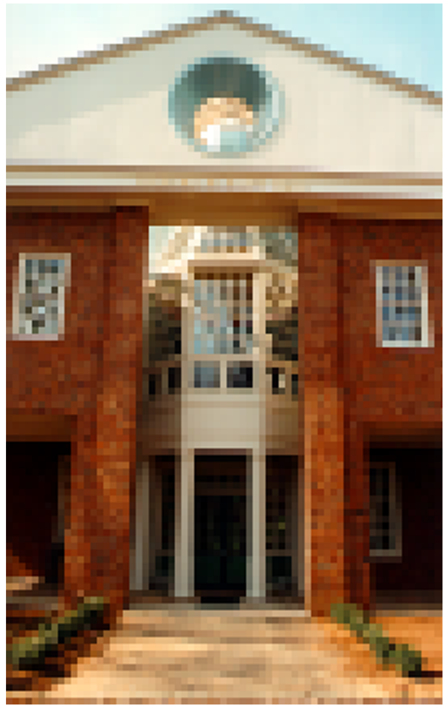
Image resolution

Definition:

– Resolution is a measurement of sampling density.

– Resolution of bitmap images give a relationship between pixel dimensions and physical dimensions. The most often used measurement is ppi (pixels per inch)


272 × 416


136 × 208

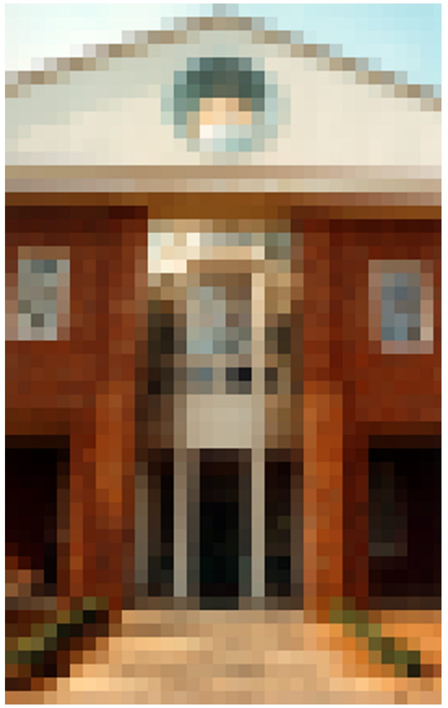
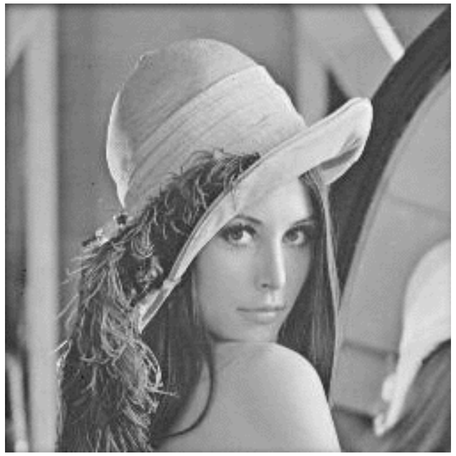
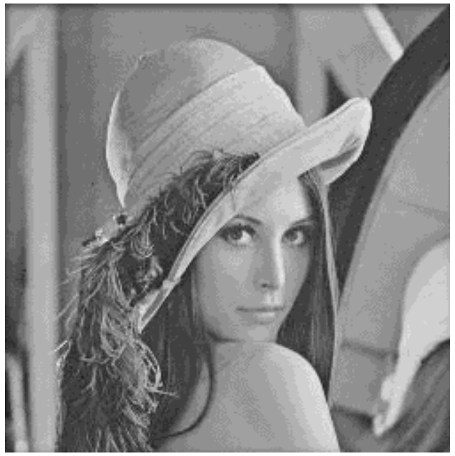

68 × 104

Image quantization (example)

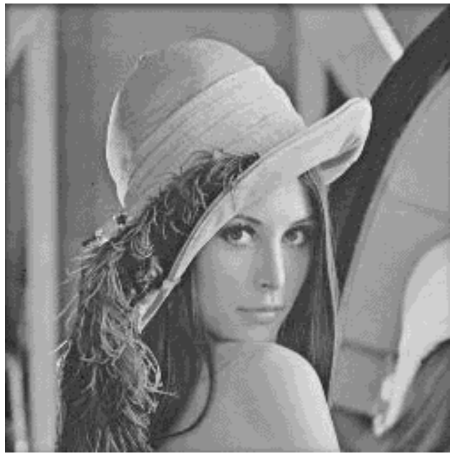
256 gray levels (8 bits/pixel)



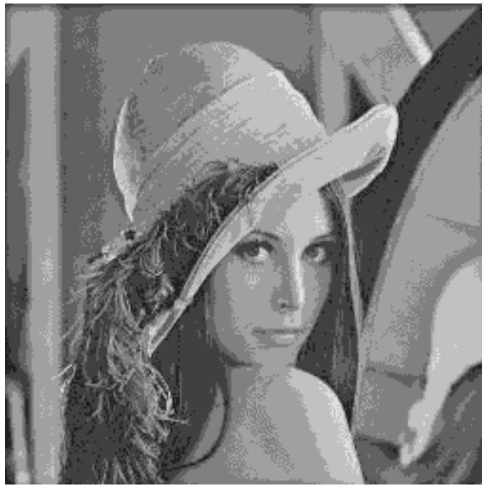
32 gray levels (5 bits/pixel)



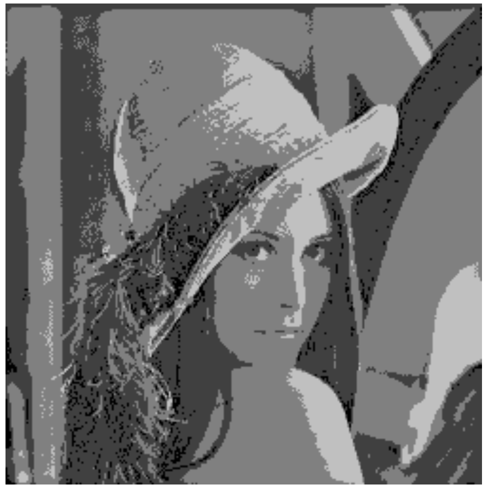
16 gray levels (4 bits/pixel)



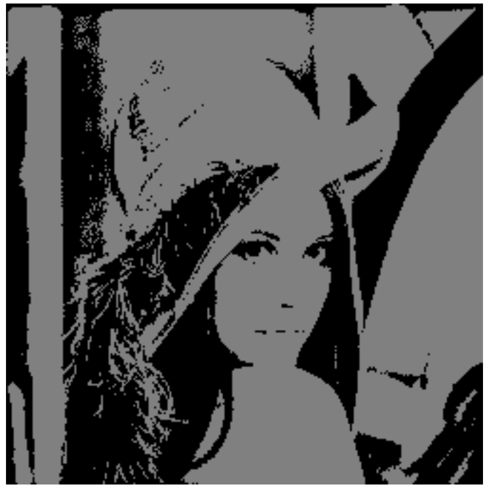
8 gray levels (3 bits/pixel)



4 gray levels (2 bits/pixel)



2 gray levels (1 bits/pixel)



Digital image representations

color depth = number of bits

Black-and-white image:

– Quantization bits per pixel: 1 (2 levels).

Gray scale image:

– Quantization bits per pixel: 8 (256 levels).

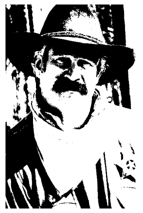
8-bit color image:

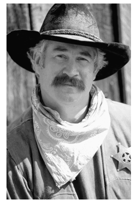
– Quantization bits per pixel: 8 (256 levels).

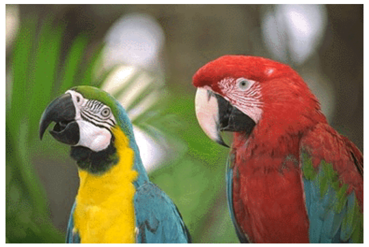
24-bit color image:

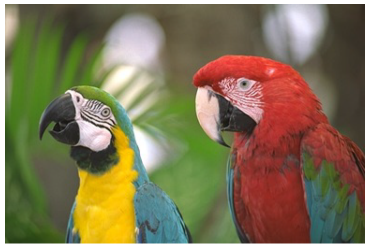
– Quantization bits per pixel: 24 (256x256x256=16,777,216 levels).

– 1 byte per color channel (RGB)


2 levels
(1 bit)


256 levels
(8 bits)


GIF format
(8 bits)


JPEG format
(24 bits)

Graphical representation of 24-bit “true color”

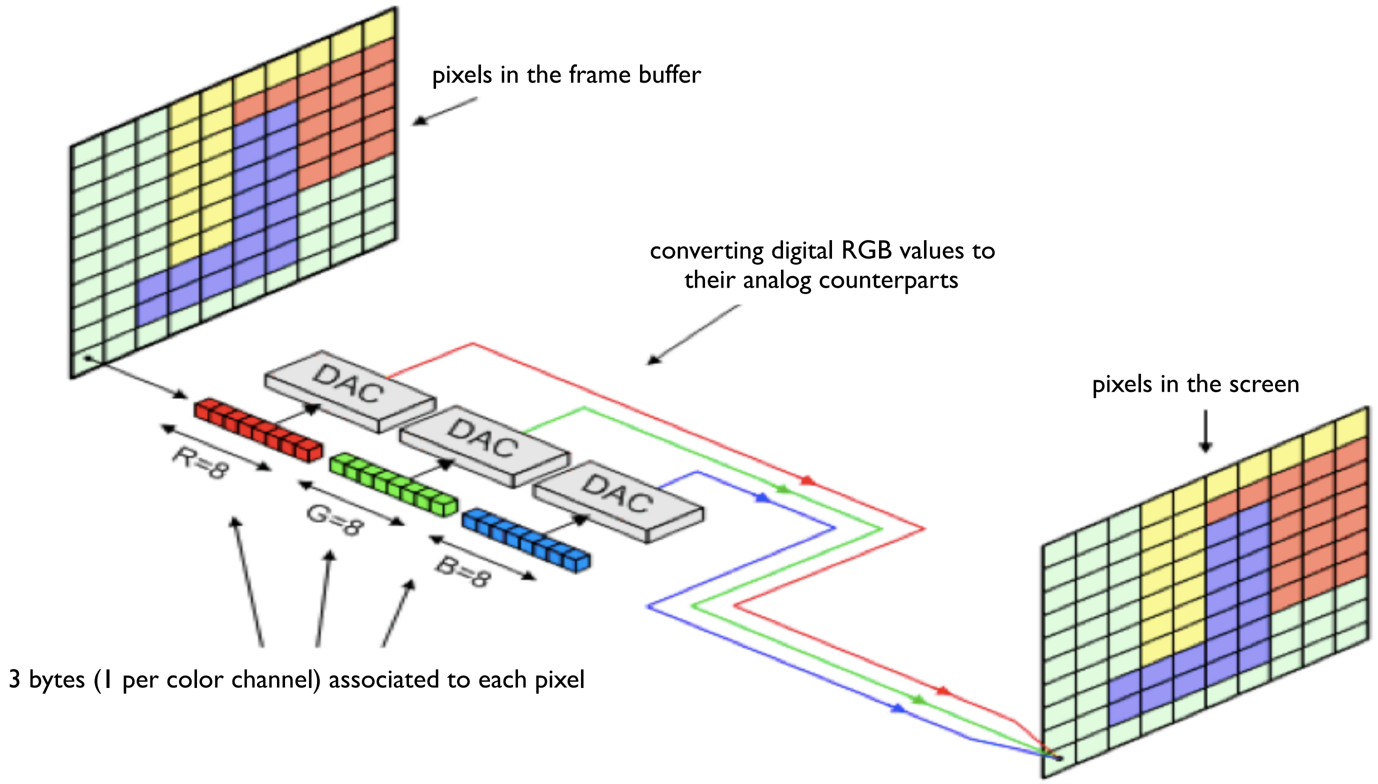
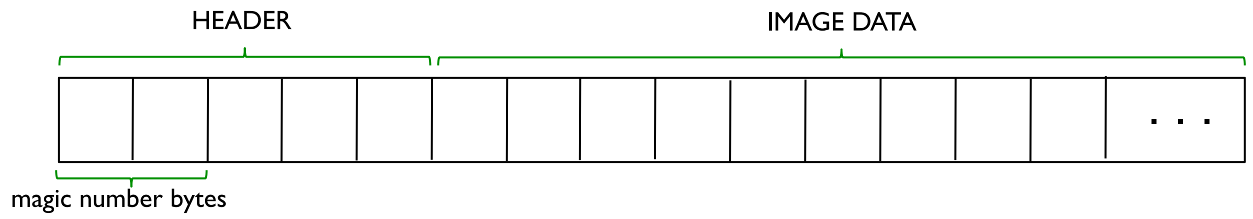

pixels in the frame buffer
converting digital RGB values to their analog counterparts
pixels in the screen
DAC
DAC
DAC
R=8
G=8
B=8
3 bytes (1 per color channel) associated to each pixel

Image file formats



Structure of an image file format:

- Many image formats adhere to the simple model shown above (line by line, no breaks between lines).
- The header contains at least the width and height of the image.
- Most headers begin with a **signature** or "magic number" - a short sequence of bytes for identifying the file format.

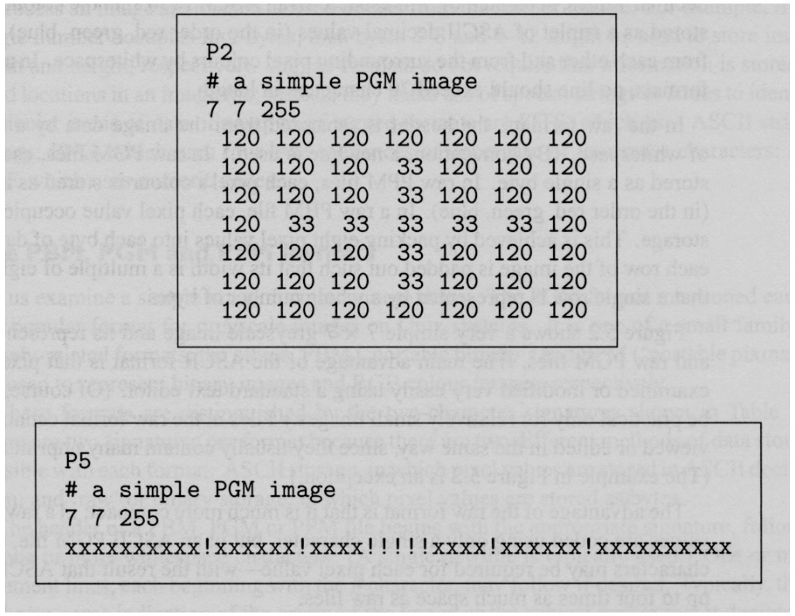
Common formats:

- GIF (Graphic Interchange Format)
- PNG (Portable Network Graphics)
- JPEG (Joint Photographic Experts Group)
- TIFF (Tagged Image File Format)
- PGM (Portable Gray Map)
- FITS (Flexible Image Transport System)

PGM format

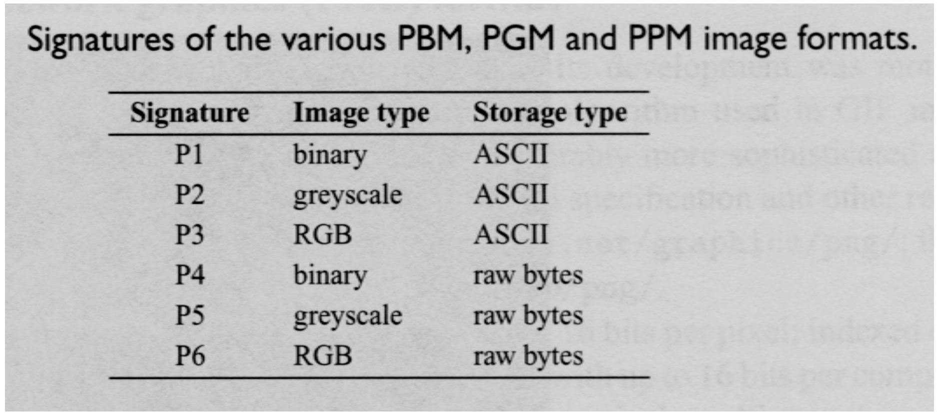
Main purpose:

- A popular format for grayscale images (8 bits/pixel)
- Closely-related formats are:
  - PBM (Portable Bitmap), for binary images (1 bit/pixel)
  - PPM (Portable Pixelmap), for color images (24 bits/pixel)



```
P2
# a simple PGM image
7 7 255
 120 120 120 120 120 120 120
 120 120 120  33 120 120 120
 120 120 120  33 120 120 120
 120  33  33  33  33  33 120
 120 120 120  33 120 120 120
 120 120 120  33 120 120 120
 120 120 120 120 120 120 120
```

```
P5
# a simple PGM image
7 7 255
xxxxxxxxxx!xxxxxx!xxxx!!!!!xxxx!xxxxxx!xxxxxxxxxx
```

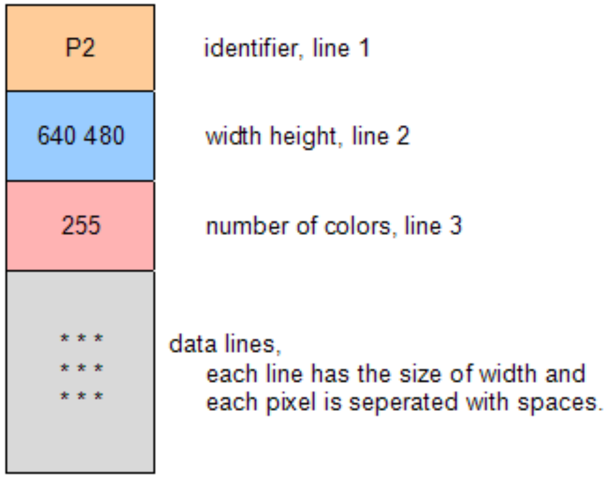


Signatures of the various PBM, PGM and PPM image formats.

Signature	Image type	Storage type
P1	binary	ASCII
P2	greyscale	ASCII
P3	RGB	ASCII
P4	binary	raw bytes
P5	greyscale	raw bytes
P6	RGB	raw bytes

PGM format (cont'd)

PGM ASCII storage



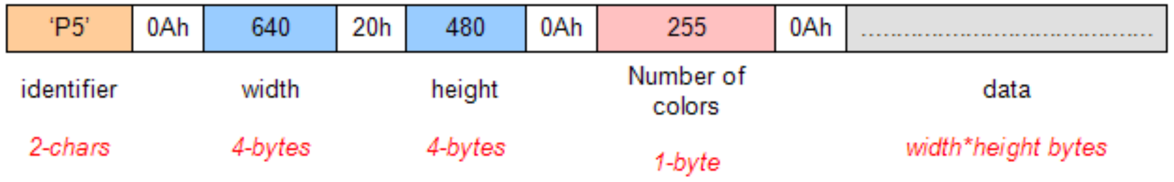
ASCII storage has the following advantages:

- Pixel values can be examined or modified very easily using a standard text editor.
- Files in raw format cannot be modified in this way since they contain many unprintable characters.

RAW storage has the following advantages:

- It is much more compact compared to the ASCII format.
- Pixel values are coded using only a single character!

PGM binary (raw) storage



Programming: example

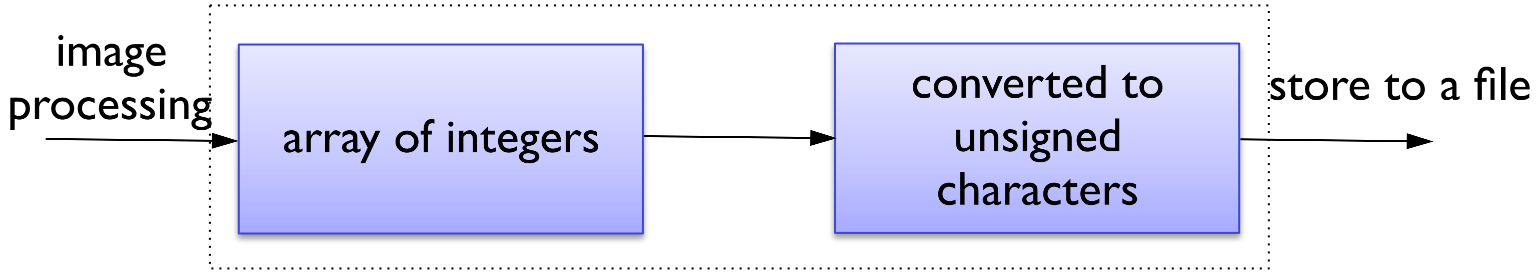
image.h

```
class ImageType
{
 public:
   ImageType();
   ~ImageType();

     // more functions ...

 private:
   int N, M, Q; //N:#rows, M:#columns
   int **pixelValue;
 };
```

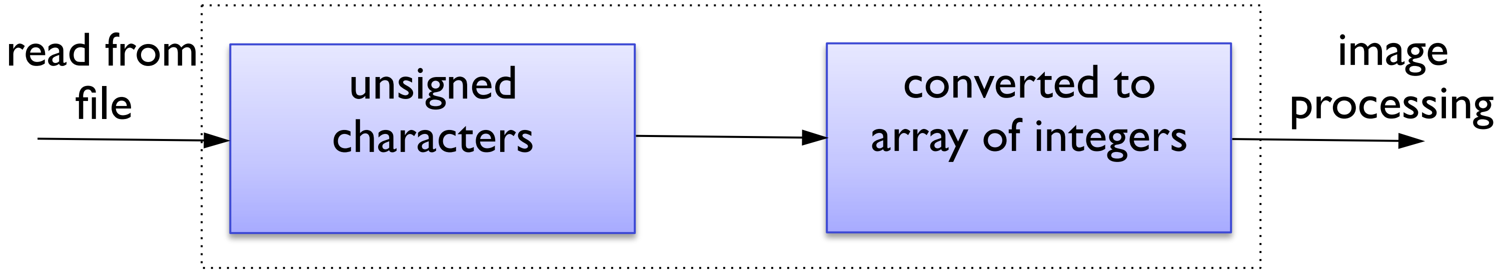
main.cpp

```
void readImageHeader(char[],int&,int&,int&,bool&);
void readImage(char[], ImageType&);
void writeImage(char[], ImageType&);

void main(int argc, char *argv[])
{
  int i, j;
  int M, N, Q;
  bool type;
  int val;
  int thresh;

     // read image header
  readImageHeader(argv[1], N, M, Q, type);

     // allocate memory for the image array
  ImageType image(N, M, Q);

     // read image
  readImage(argv[1], image);
  cout << "Enter threshold: ”; cin >> thresh;

     // threshold image
  for(i=0; i<N; i++)
    for(j=0; j<M; j++) {
      image.getVal(i, j, val);
      if(val < thresh)
        image.setVal(i, j, 255);
      else
        image.setVal(i, j, 0);
     }

     // write image
  writeImage(argv[2], image);

 }
```

Reading/Writing PGM images



writeimage.cpp



readimage.cpp

Writing a PGM image to a file

writeimage.cpp

```
void writeImage(char fname[], ImageType& image){
 int N, M, Q;
 unsigned char *charImage;
 ofstream ofp;

 image.getImageInfo(N, M, Q);

 charImage=(unsigned char*)new unsigned char [M*N];

    // convert the integer values to unsigned char
 int val;
 for(i=0; i<N; i++)
    for(j=0; j<M; j++){
     image.getVal(i, j, val);
     charImage[i*M+j]=(unsigned char)val;
    }

 ofp.open(fname, ios::out);
 if (!ofp) {
   cout << "Can't open file: " << fname << endl;
   exit(1);
 }
 ofp << "P5" << endl;
 ofp << M << " " << N << endl;
 ofp << Q << endl;

 ofp.write(reinterpret_cast<char*>(charImage),(M*N)*sizeof(unsigned char));

 if (ofp.fail()) {
   cout << "Can't write image " << fname << endl;
   exit(0);
 }
 ofp.close();
}
```

Reading a PGM image from a file

readimage.cpp

```
void readImage(char fname[], ImageType& image){
 int i, j;
 int N, M, Q;
 unsigned char *charImage;
 char header [100], *ptr;
 ifstream ifp;

 ifp.open(fname, ios::in);
 if (!ifp) {
   cout << "Can't read image: " << fname << endl;
   exit(1);
 }
    // read header
 ifp.getline(header,100,'\n');
  if ( (header[0]!=80) || (header[1]!=53) ) {   // 'P' or '5'
      cout << "Image " << fname << " is not PGM" << endl;
      exit(1);
  }

 ifp.getline(header,100,'\n');
 while(header[0]=='#')
   ifp.getline(header,100,'\n');

 M=strtol(header,&ptr,0);
 N=atoi(ptr);
 ifp.getline(header,100,'\n');
 Q=strtol(header,&ptr,0);

 charImage = (unsigned char *) new unsigned char [M*N];
 ifp.read( reinterpret_cast<char *>(charImage), (M*N)*sizeof(unsigned char));
 if (ifp.fail()) {
   cout << "Image " << fname << " has wrong size" << endl;
   exit(1);
 }
 ifp.close();
```

Reading a PGM image from a file (cont'd)

```
   //
   // Convert the unsigned characters to integers
   //

   int val;

   for(i=0; i<N; i++)
     for(j=0; j<M; j++) {
       val = (int)charImage[i*M+j];
       image.setVal(i, j, val);
     }

 }
```

How do I display an image from within my C++ program?

Recipe:

- Save the image into a file with a default name (e.g., tmp.pgm) using the WriteImage function.
- Put the following command in your C++ program:

```
system("gimp tmp.pgm");
```

- This is a system call !!
- It passes the command within the quotes to the Unix operating system.
- You can execute any Unix command this way ….

Image processing software

Examples:

- CVIPtools (Computer Vision and Image Processing tools.
- Intel Open Computer Vision Library.
- Microsoft Vision SDL Library.
- Matlab.
- Khoros.

For more information, see:

- **http://www.cs.unr.edu/~bebis/CS791E**
- **http://www.cs.unr.edu/CRCD/ComputerVision/cv_resources.html**

Summary:

….:

- Light & Color
- Image Formation
- Image Digitization
- Image Representation
- Image File Formats
- Programming with images.