# CS 543: Computer Graphics
# Lecture 9 (Part III): Raster Graphics Part 3

Emmanuel Agu

# Manipulating Pixmaps

- Pixmap = rectangular array of numerical values
- Pixmap copied to frame buffer = rendered
- Change frame buffer entry = onscreen picture changes
- Each pixel location has fixed number of bits (color depth)
- Example: if color depth is *b* bits, can store up to $2^b$ values

# Manipulating Pixmaps

- Operations of interest:
  - Copying pixmaps
    - glReadPixels: frame buffer to off-screen memory
    - glDrawPixels: pixmap to frame buffer
    - glCopyPixels: frame buffer to frame buffer
    - memCopy: off-screen to off-screen
  - Comparing pixmaps
  - Representing and coloring regions in pixmap

## Manipulating Pixmaps

- Data types for pixmaps
  - Bitmap: 1 bit, on or off
  - Gray scale: one byte, values 0-255
  - RGB: 3 bytes (red, green, blue)
  - RGBA: 4 byte (red, green, blue, alpha)
- Declaration of RGB triple:

```
class RGB{
  public: unsigned char r, g, b;
};
```

# RGBpixmap Class

- OpenGL convention: pixmap (bottom to top, left to right)
- Add draw, read and copy methods (which use openGL)

```
Class RGB{
  public: unsigned char r, g, b;

  RGBpixmap( ); // constructor
  void setPixel(int x, int y, RGB color);
  RGB getPixel(int x, y);
  void draw( ){ glDrawPixels(nCols, nRows, GL_RGB,
                      GL_UNSIGNED_BYTE, pixel);
  void read( ){glReadPixels(x, y, nCols, nRows, GL_RGB,
                            GL_UNSIGNED_BYTE, pixel);
```

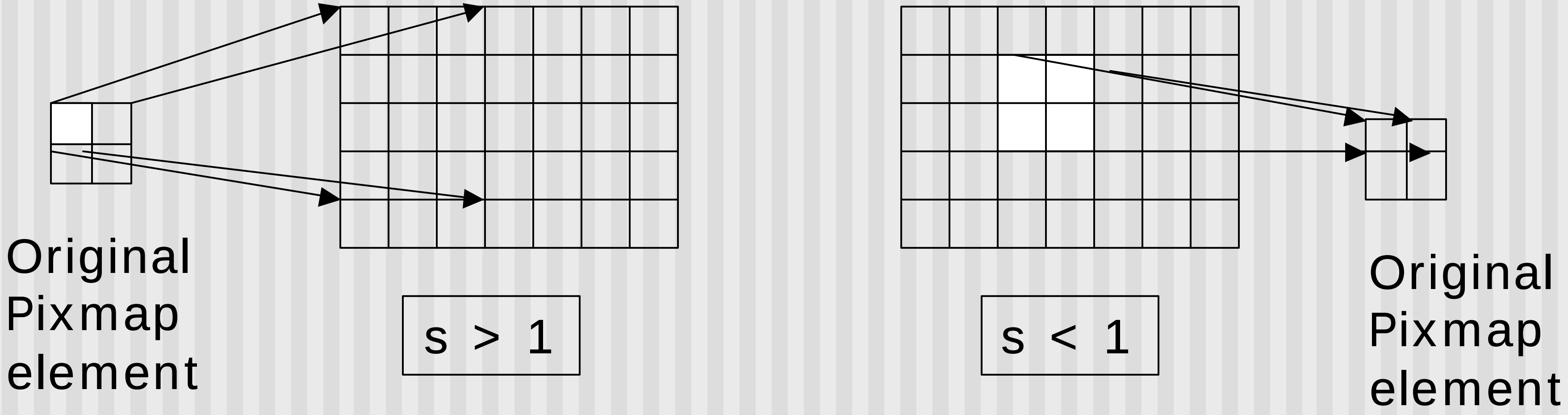
## RGBpixmap Class

```
// ….. contd.

  void copy( ){ glCopyPixels(.. Parameters..);

  int readBMPFile(char *fname);
  void writeBMPFile(char *fname);
};
```

Note: refer to Hill fig. 10.3 for full RGBPixmap declaration

# Scaling and Rotating Images

- Scaling: want a pixmap that has s times more pixels in x, y
  - s > 1: enlargement
  - s < 1: reduction (information is lost!)



- openGL scaling:
  - **`glPixelZoom(float sx, float sy)`**
  - Sets scale factors for drawing pixmaps
  - Note: pixmaps not scaled, pictures drawn are scaled

## Scaling and Rotating Images

- glPixelZoom(float sx, float sy)
  - Sets scale factors for subsequent glDrawPixels command
  - Scaling is about current raster position, pt.
  - Pixel row r and column c of pixmap
  - Drawn as rectangle with bottom left current screen coordinates
  - Draws (pt.x + sx*r, pt.y + sy.c)
- 90, 180 and 270 degree rotations:
  - Copy one pixmap to another doing matrix transposes
- General rotations:
  - affine transform of pixmap points to get new pixmap

## Combining Pixmaps

- Two pixmaps A and B combined pixelwise to form third pixel C
- i.e. C[i][j] = A[i][j] $\otimes$ B[i][j]
- Averaging:
  - C[i][j] = ½ *(A[i][j] + B[i][j])
- Subtraction:
  - C[i][j] = A[i][j] - B[i][j]
- Generalized weighting:
  - C[i][j] = (1-f).A[i][j] + f.B[i][j]

## Combining Pixmaps

- Generalized weighting:
  - C[i][j] = (1-f).A[i][j] + f.B[i][j]
- Example:
  - A = (14, 246, 97), B = (82, 12, 190), f = 0.2
  - C = (27, 199, 115) = 0.8 A + 0.2 B
- Question: How to dissolve image A into B?
- Raster demo!!

# Alpha Channel and Image Blending

- Even more generalized weighting = blending/compositing
- Blending:
  - draw partially transparent image over another
  - Add 4th component, alpha value (A) to RGB
  - Interpretation: alpha specifies how opaque each pixel is
  - Transparent (A = 0), Total opacity (A = 255)
  - Alpha most frequently used in scaling colors
- Alpha channel: series of alpha values in a pixmap

```
class RGB{
      public: unsigned char r, g, b,a;
};
```

# References

- Hill, chapter 10