Buffers

Ed Angel
Professor Emeritus of Computer Science
University of New Mexico

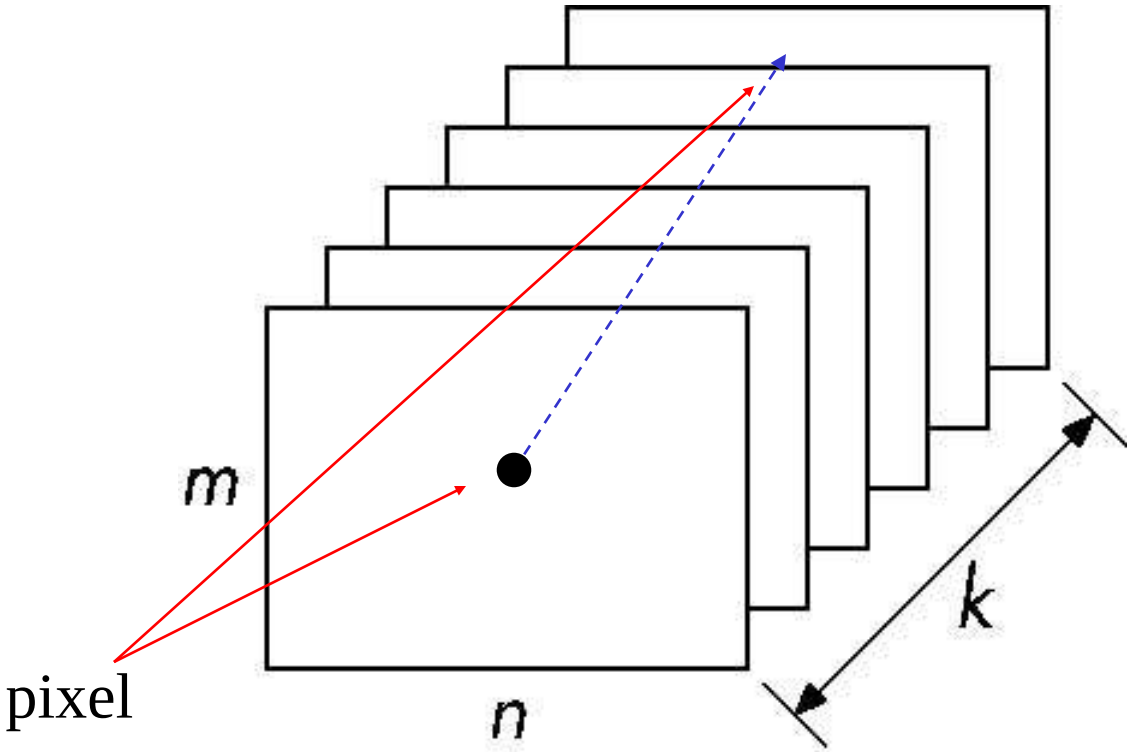
Objectives

- Introduce additional OpenGL buffers
- Learn to read from buffers
- Learn to use blending

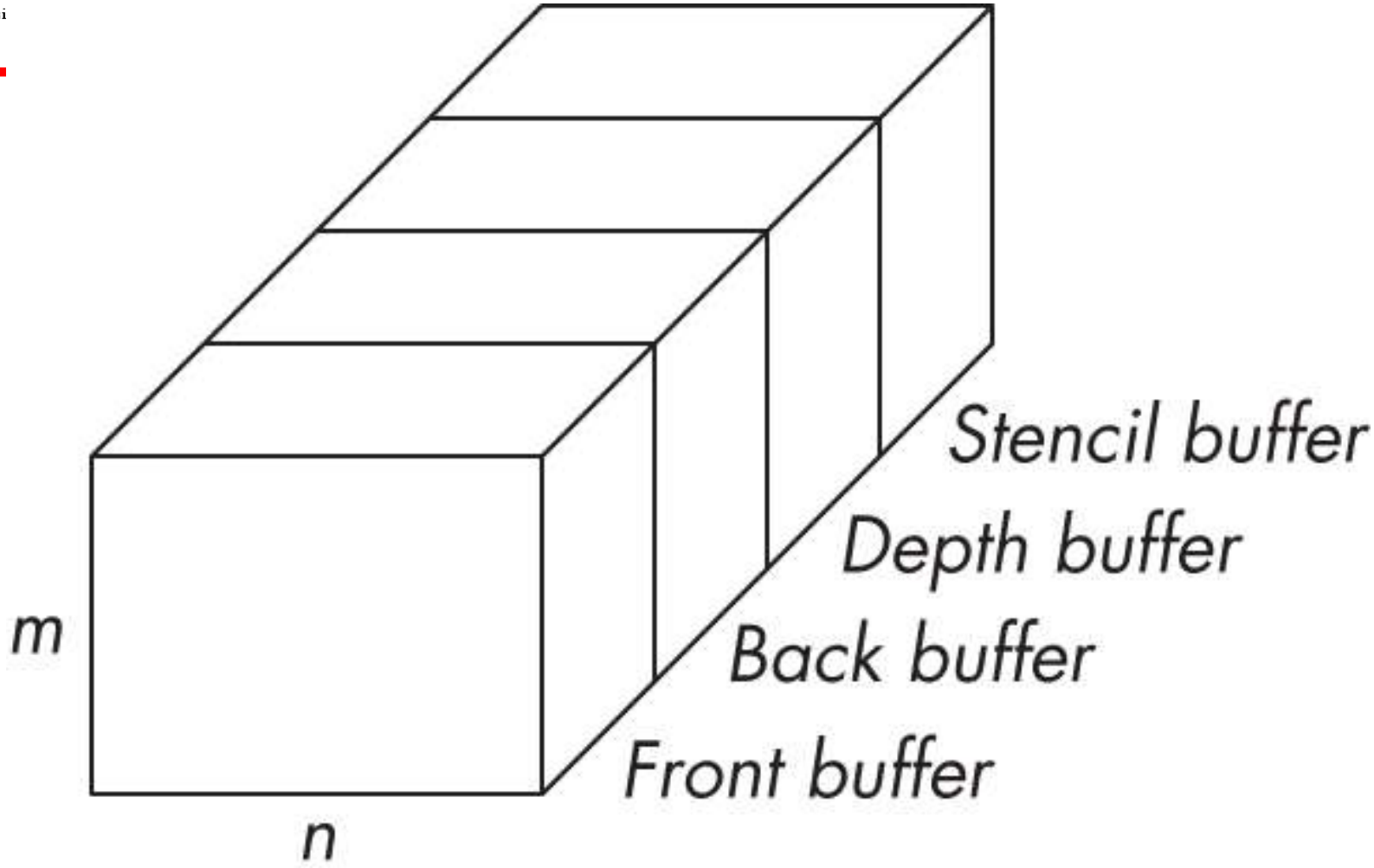
The University of New Mexico

Buffer

Define a buffer by its spatial resolution (n x m) and its depth (or precision) k, the number of bits/pixel



OpenGL Frame Buffer

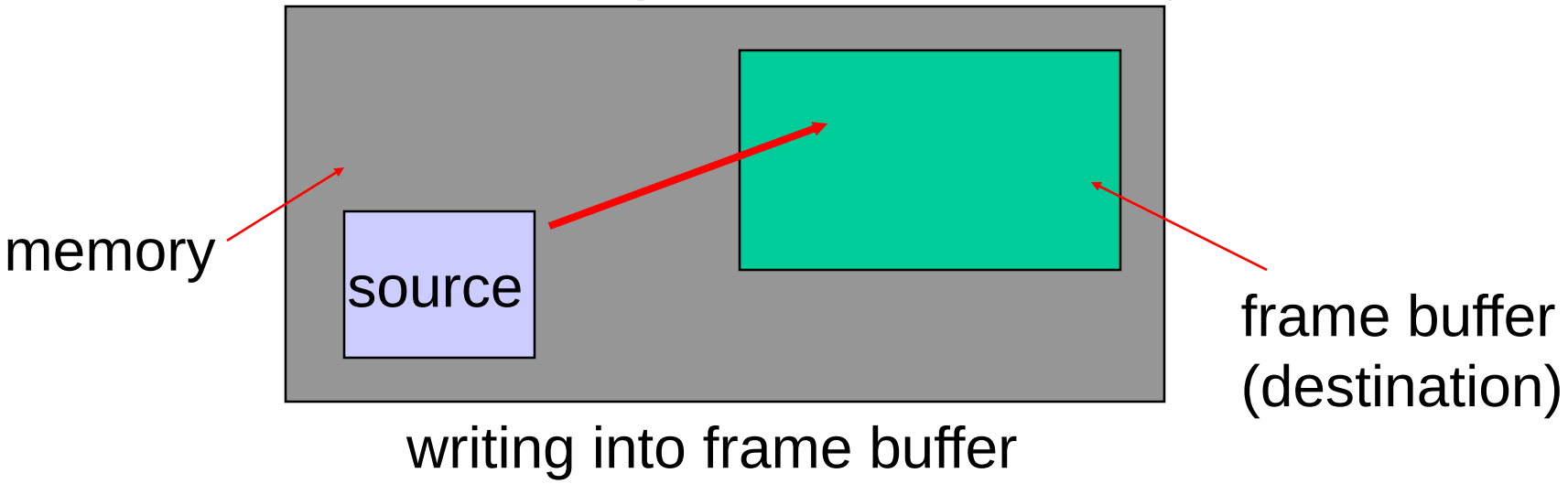


OpenGL Buffers

- Color buffers can be displayed
  - Front
  - Back
  - Auxiliary
  - Stereo
- Depth
- Stencil
  - Holds masks
- Most RGBA buffers 8 bits per component
- Latest are floating point (IEEE)

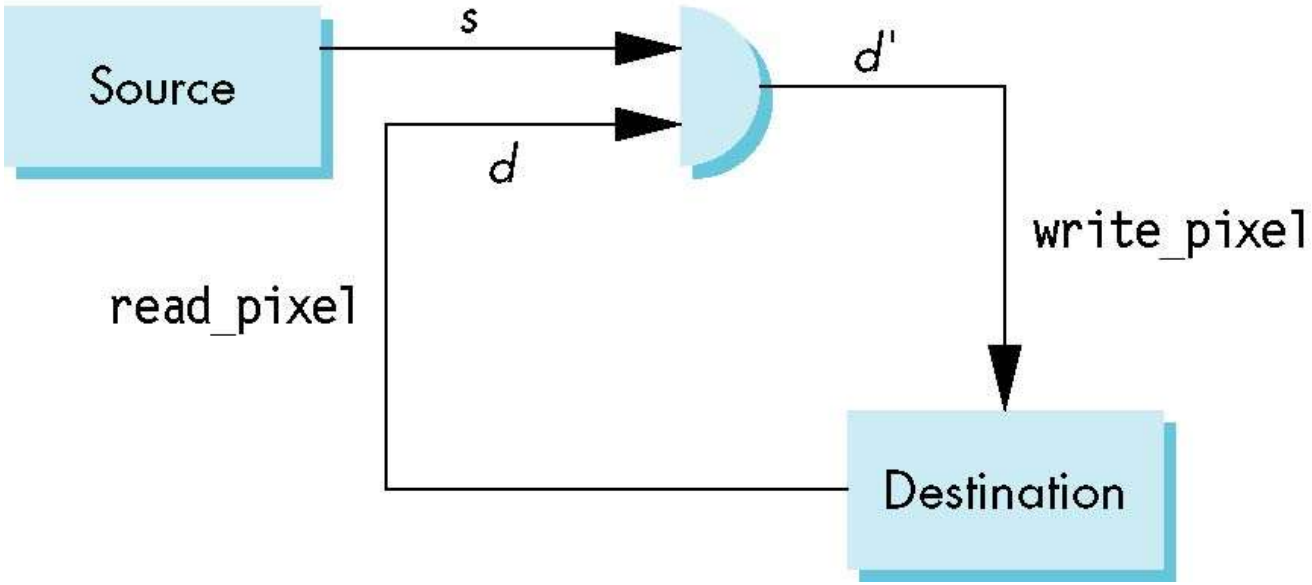
Writing in Buffers

- Conceptually, we can consider all of memory as a large two-dimensional array of pixels
- We read and write rectangular block of pixels
  - *Bit block transfer* (*bitblt*) *operations*
- The frame buffer is part of this memory



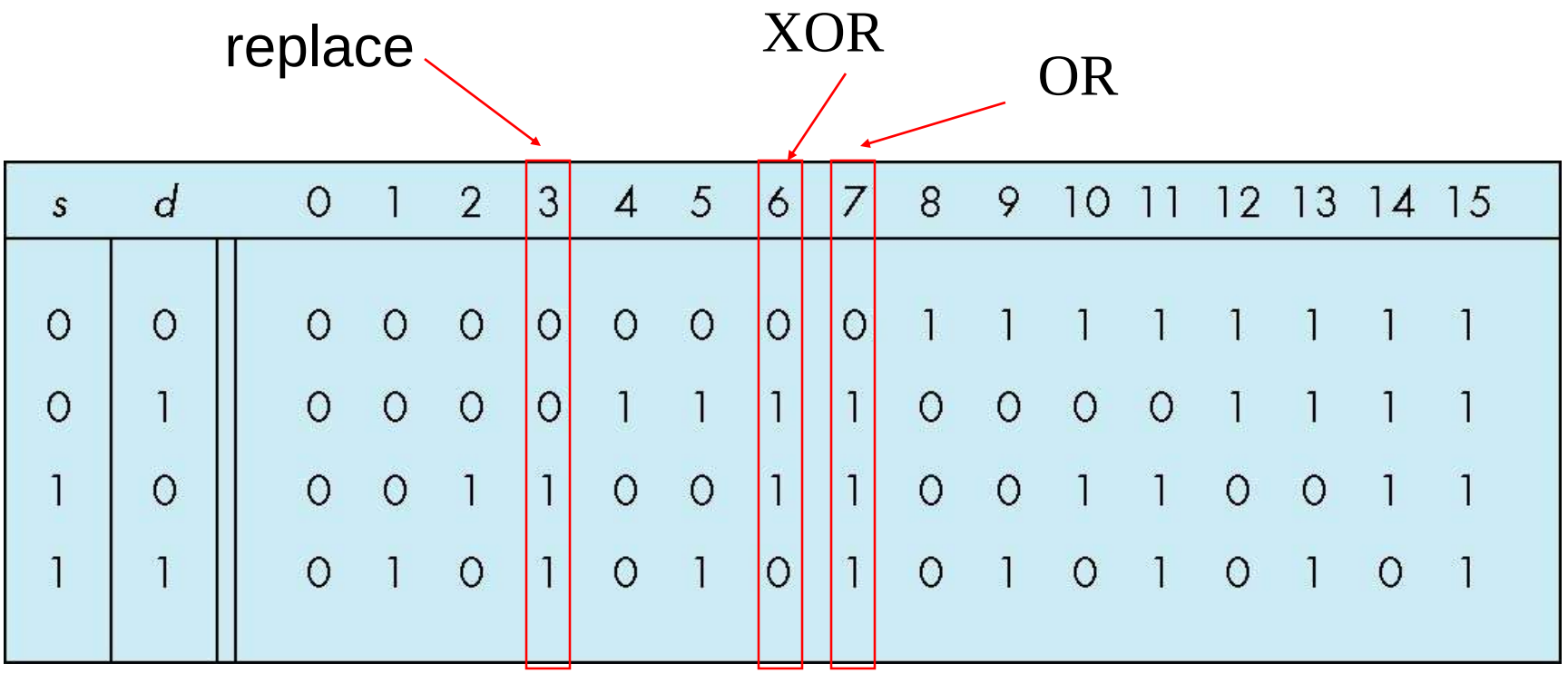
Writing Model

Read destination pixel before writing source



Bit Writing Modes

- Source and destination bits are combined bitwise
- 16 possible functions (one per column in table)



s	d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

XOR mode

- Recall from Chapter 3 that we can use XOR by enabling logic operations and selecting the XOR write mode
- XOR is especially useful for swapping blocks of memory such as menus that are stored off screen

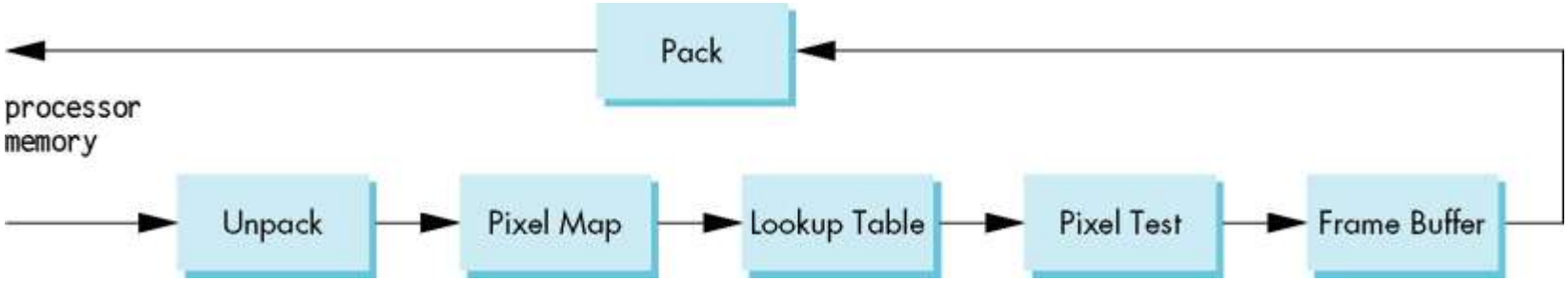
  If S represents screen and M represents a menu the sequence

  $$S \leftarrow S \oplus M$$
  $$M \leftarrow S \oplus M$$
  $$S \leftarrow S \oplus M$$

  swaps the S and M

The Pixel Pipeline

- OpenGL has a separate pipeline for pixels
  - Writing pixels involves
    - Moving pixels from processor memory to the frame buffer
    - Format conversions
    - Mapping, Lookups, Tests
  - Reading pixels
    - Format conversion



Buffer Selection

- OpenGL can read from any of the buffers (front, back, depth)
- Default to the back buffer
- Change with **`glReadBuffer`**
- Note that format of the pixels in the frame buffer is different from that of processor memory and these two types of memory reside in different places
  - Need packing and unpacking
  - Reading can be slow
- Drawing through texture functions

OpenGL Pixel Functions

```
glReadPixels(x,y,width,height,format,type,myimage)
```

start pixel in frame buffer

size

type of pixels

type of image

pointer to processor memory

```
GLubyte myimage[512][512][3];
glReadPixels(0,0, 512, 512, GL_RGB,
      GL_UNSIGNED_BYTE, myimage);
```

Deprecated Functionality

- glDrawPixels
- glCopyPixels
- glBitMap
- Replace by use of texture functionality, glBltFrameBuffer, frame buffer objects

Render to Texture

- GPUs now include a large amount of texture memory that we can write into
- Advantage: fast (not under control of window system)
- Using frame buffer objects (FBOs) we can render into texture memory instead of the frame buffer and then read from this memory
  - Image processing
  - GPGPU