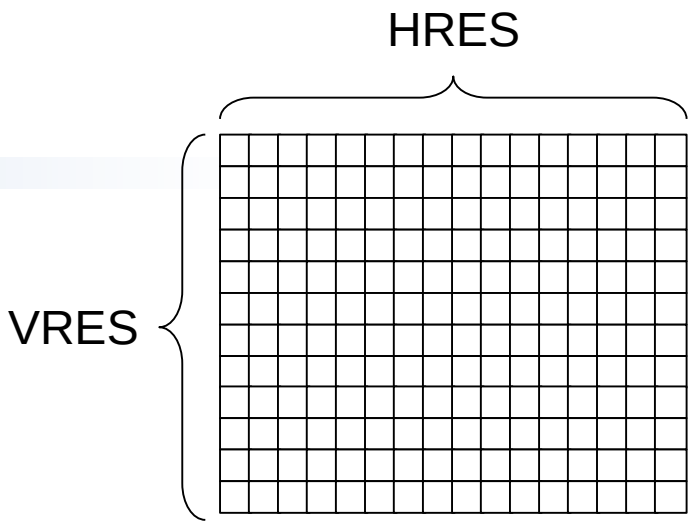
Image Resolutions

CS418 Computer Graphics

John C. Hart

Raster Images



- 2-D Array of Color Values
- Spatial Resolution: HRES × VRES
- Color Resolution (bits per pixel)

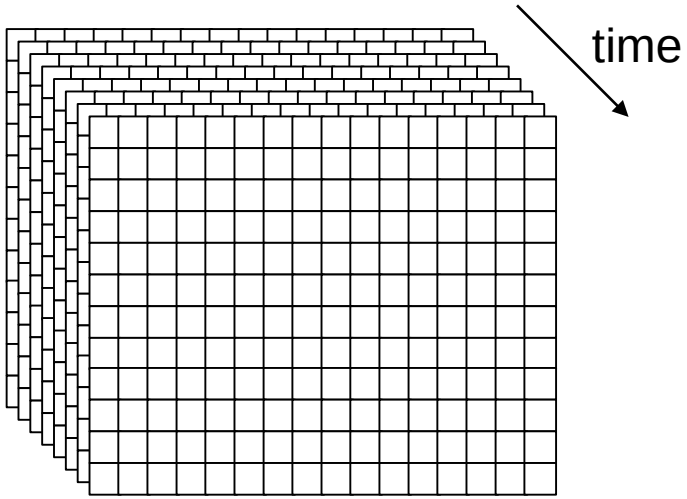
1 bpp: 0 1

8 bpp: = 1 1 1 0 1 1 0 1

24 bpp: = #FF6D55

Raster Images

- 2-D Array of Color Values
- Spatial Resolution: HRES × VRES
- Color Resolution (bits per pixel)
- Temporal Resolution (frames per sec.)



Raster Images

- 2-D Array of Color Values
- Spatial Resolution: HRES × VRES
- Color Resolution (bits per pixel)
- Temporal Resolution (frames per sec.)
- Interlace

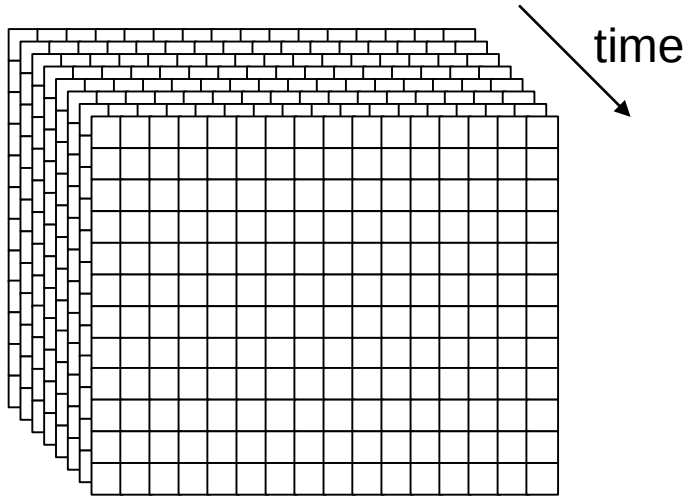
  Odd scan lines (one field)

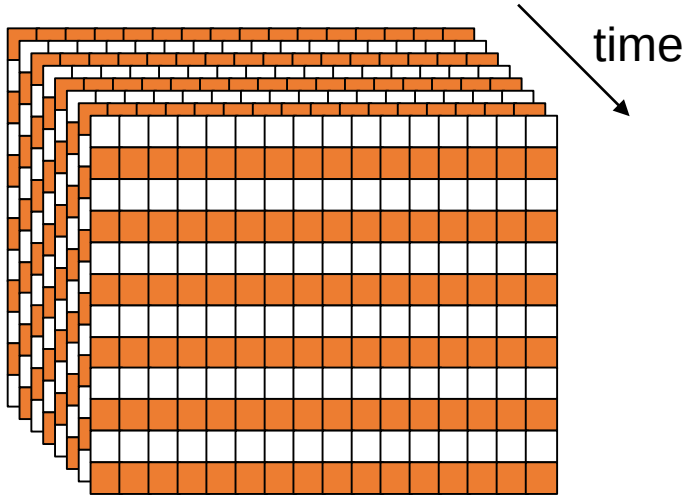
  Even scan lines (other field)

  Repeat

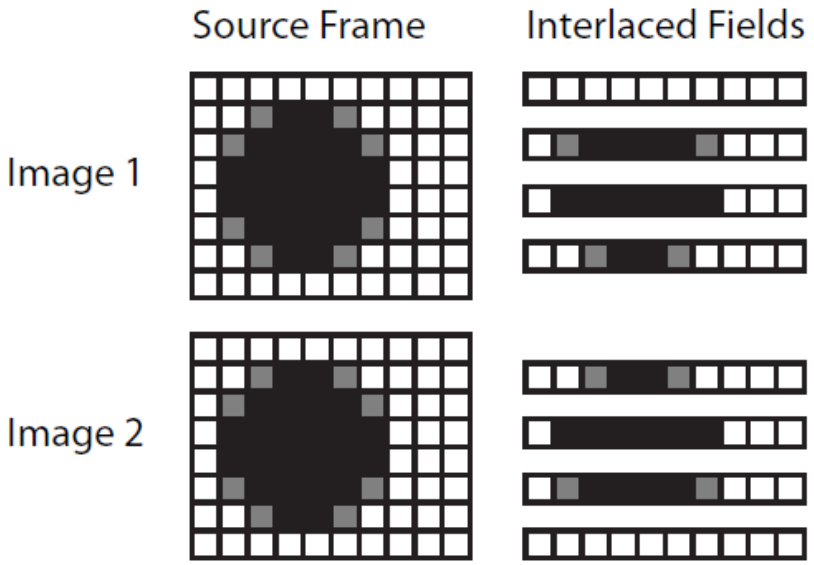
  Doubles temporal resolution

  Halves spatial resolution

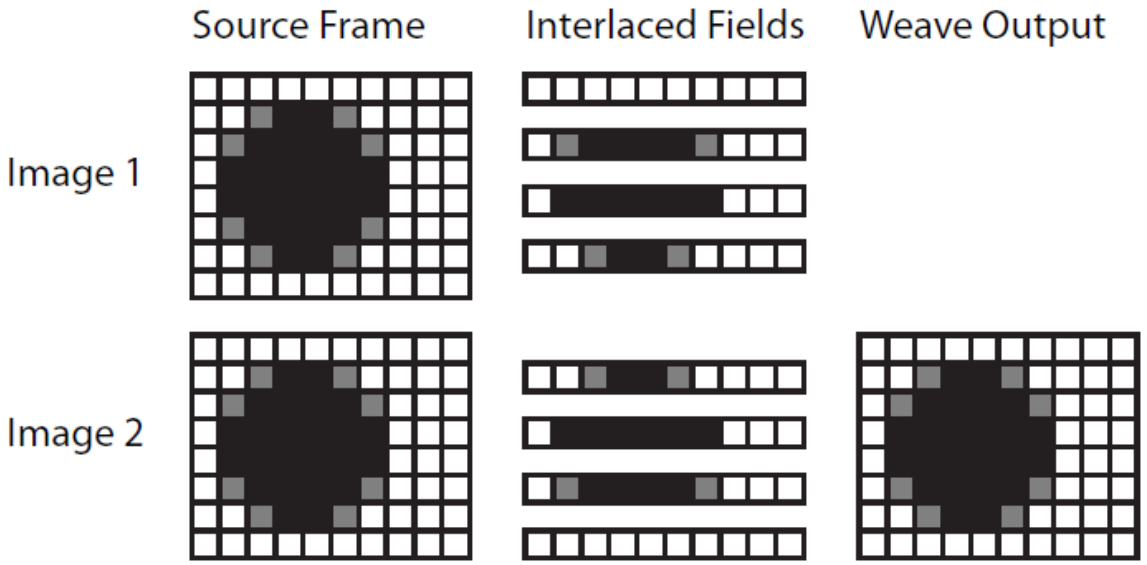




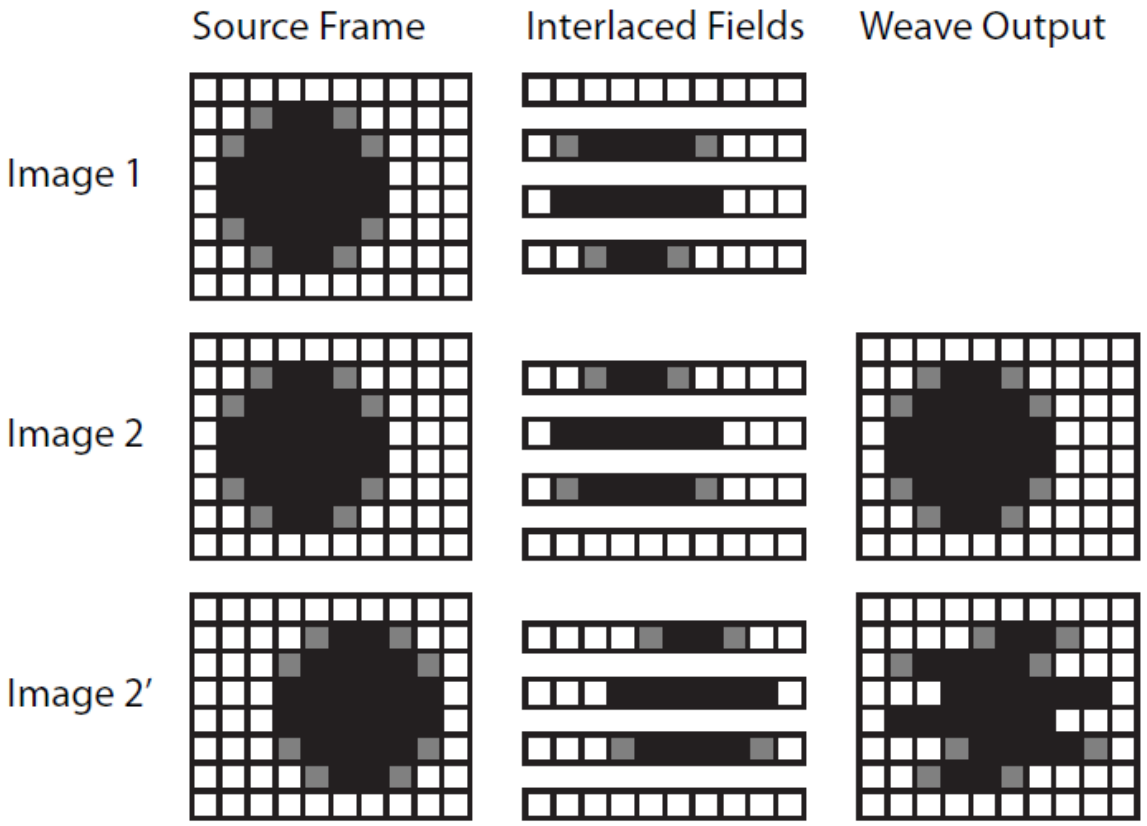
Interlaced v. Progressive Scan



Interlaced v. Progressive Scan



Interlaced v. Progressive Scan



Interlaced v. Progressive Scan

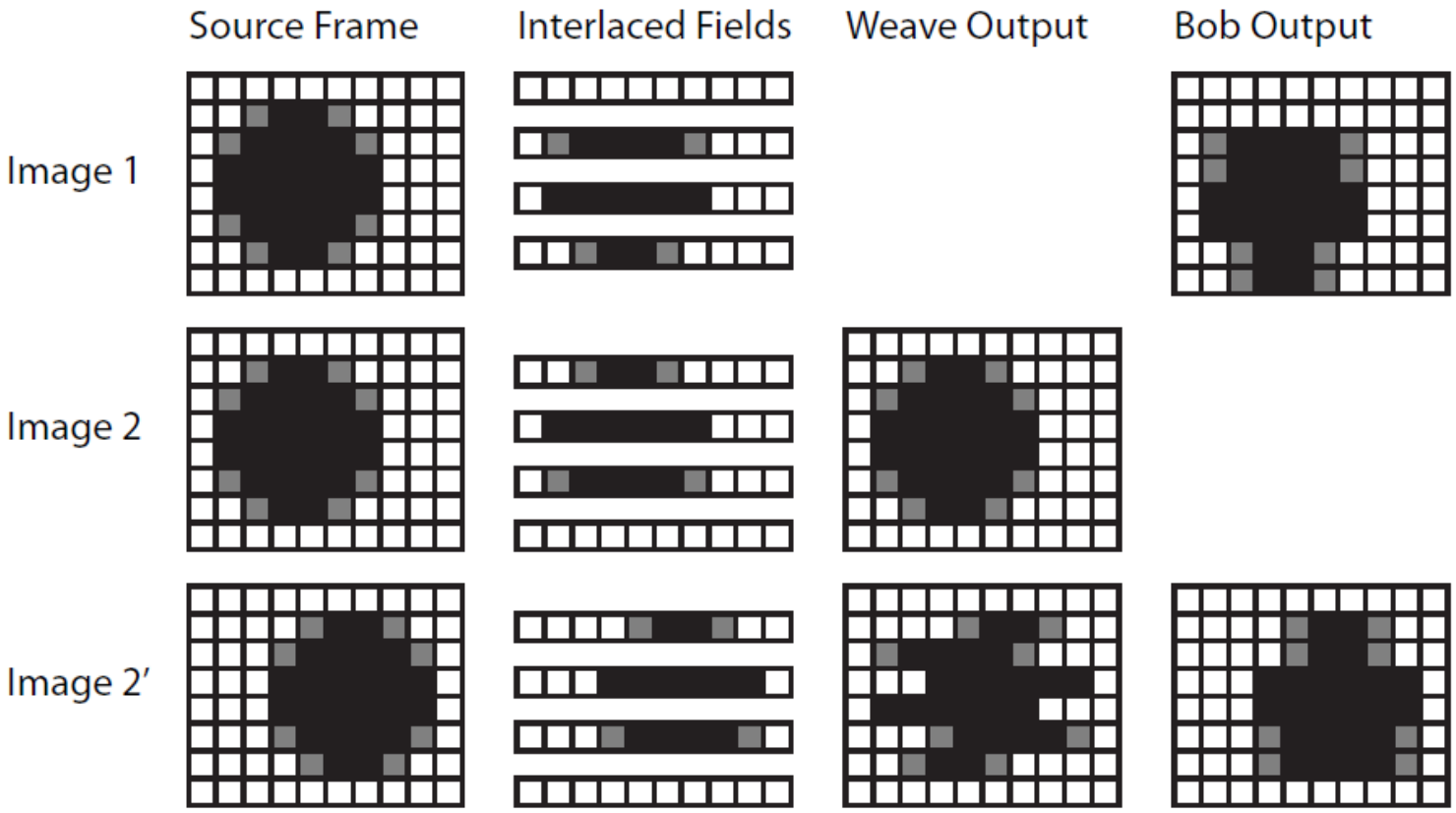
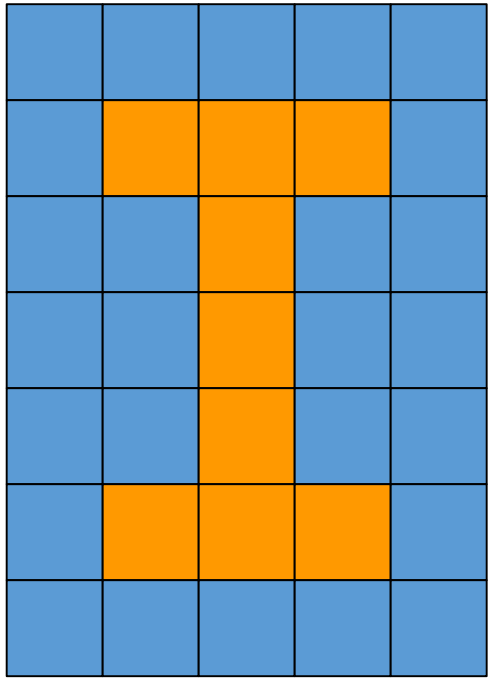


Image File Formats

Format	Fidelity	Complexity	Use
BMP	Uncompressed	**Simple**, Microsoft	Easy to process, windows icons
PPM	Uncompressed	**Simple**, open, dated	Easy to process, unix icons
GIF	Compressed Lossless (LZW)	**Only 256 colors**	Charts, graphs, diagrams, text
JPG (Exif)	Compressed **Lossy** (DCT)	Blocky, edges ring	Photographs
TIFF	Compressed Lossless (LZW)	**Flexible** but complex structure	Fax, scanning, artwork
PNG	Compressed Lossless (zlib)	**Flexible** but complex structure	Distributing images
SVG	Uncompressed	Flexible	**Vector/Line Art**

Color Palettes

- Store all RGB colors used in any image pixel in a table
- Store index to color in each pixel to compress data size



=

97 97 217	97 97 217	97 97 217	97 97 217	97 97 217
97 97 217	255 153 0	255 153 0	255 153 0	97 97 217
97 97 217	97 97 217	255 153 0	97 97 217	97 97 217
97 97 217	97 97 217	255 153 0	97 97 217	97 97 217
97 97 217	97 97 217	255 153 0	97 97 217	97 97 217
97 97 217	255 153 0	255 153 0	255 153 0	97 97 217
97 97 217	97 97 217	97 97 217	97 97 217	97 97 217

or

0	0	0	0	0
0	1	1	1	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	1	1	1	0
0	0	0	0	0

where:

0=

1=

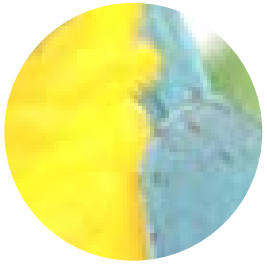
Reducing Color

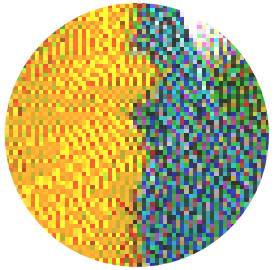
- How to convert a full color image into a paletted image
- “Posterize” picks color in the palette closest to full color pixel
- Dithering adds colors by combining palette colors of neighboring pixels

N. Damera-Venkata, B.L. Evans & V. Monga
Color Error Diffusion Halftoning,
IEEE Signal Processing 20(4), 51-58, July 2003.

Reducing Color

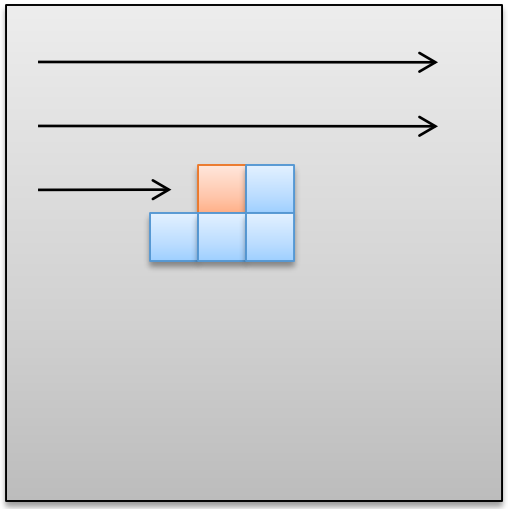
- How to convert a full color image into a paletted image
- “Posterize” picks color in the palette closest to full color pixel
- Dithering adds colors by combining palette colors of neighboring pixels





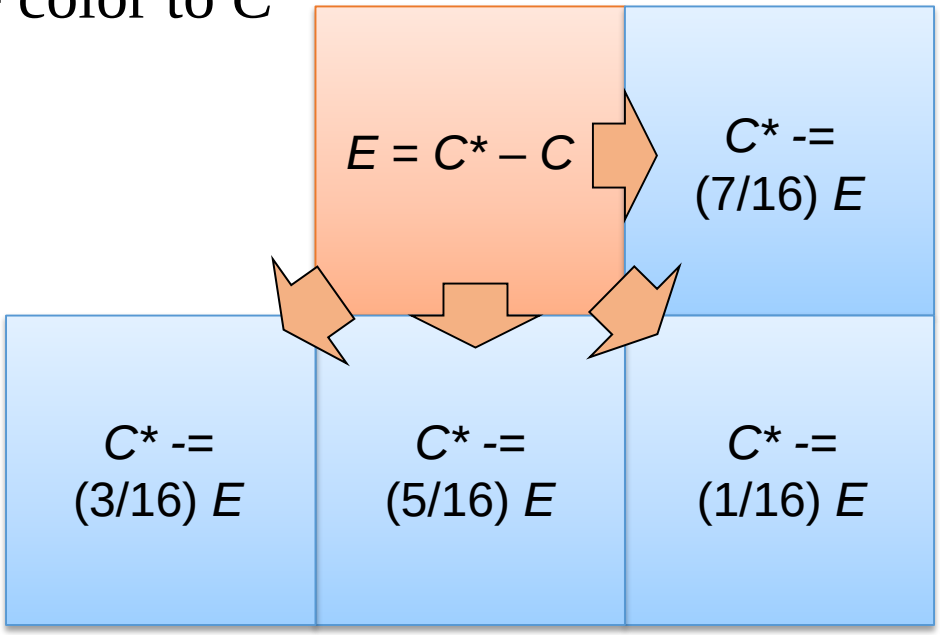
N. Damera-Venkata, B.L. Evans & V. Monga
Color Error Diffusion Halftoning,
IEEE Signal Processing 20(4), 51-58, July 2003.

Error Diffusion

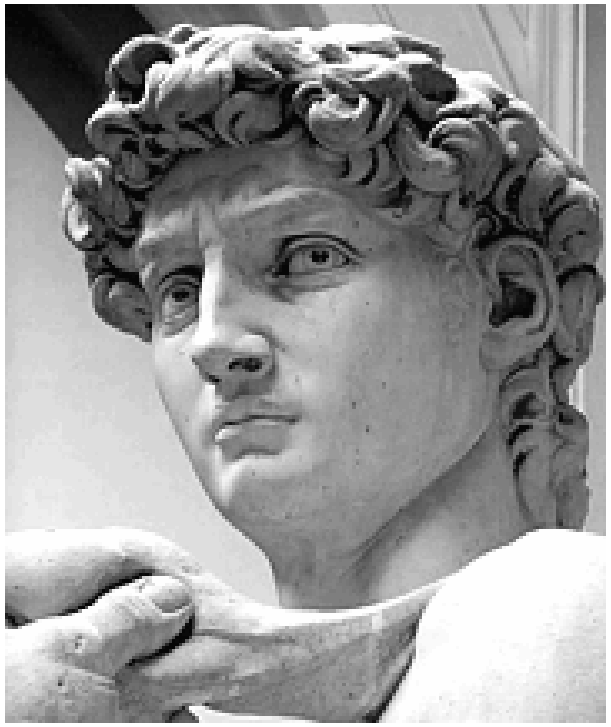


Floyd-Steinberg Algorithm

- Each pixel color $C(x,y)$ has an approximation $C^*(x,y)$
- Initialize $C^*(x,y) = \text{approx}(C(x,y))$ for all pixels
  - approx(C) returns closest palette color to C
- Approximation error: $E = C^* - C$
  - positive E: C^* too bright
  - negative E: C^* too dim
- Counteract error by changing brightness of unprocessed neighboring pixels



Dithered Images



Lego Mosaics