Images

Continuous phenomenon \( f(x, y) \)

Set of measurements (grid) \( \rightarrow \) storage as an array

Reconstruction (in display)

Sampling

Re-sampling (change the set of samples)

Other operations

Windowing operations (crop, treat rectangle as image)

Point Operations - 1 image vs. multiple images

- Brightness / contrast
- Levels curves
- Histogram equalization
- Color operations / color twists
  - Desaturation, color to gray (naive)

Multi-image \( g\left(f_1(x, y), f_2(x, y)\right) \)

\( \alpha \) channel (still per pixel)

\[ \alpha f_1(x, y) + (1 - \alpha) f_2(x, y) \]

\( \alpha = f_1(x, y) \)

Other operations - mashing, under, atop, ....

Feathering, cloning, ....
Area Operations

\[ f(x, y) = \text{function of the neighborhood around } x, y \]

Size/shape of neighborhood (often square, odd size)

What if neighborhood exceeds border? \( x \pm k, y \pm h \)

- Zero (or constant)
- Clamp (border)
- Wrap
- Mirror

Examples:

- \( \max, \min \)
- Average
- Weighted average

Note: Weights are an image (size of neighborhood)

1D version w/ signals \( \left[ \frac{1}{3} \ 1 \ rac{1}{3} \right] \)

Convolution = sliding weighted average

Dealing w/ edges

- Infinite zeros \( \Rightarrow \) signal "grows"
- Border handling
What to do w/ Convolution

Blur - various (learn more later) different kernels
   effects of averaging - radial symmetry/gaussians
   window size
   kernel shape

"Unsharpen" - really doesn't unblur
   can't recover what was lost
   subtract neighborhood
   \[
   \begin{bmatrix}
   -\frac{1}{2} & 2 & \frac{1}{2}
   \end{bmatrix}
   \begin{bmatrix}
   -a & 1+2a & a
   \end{bmatrix}
   \]
   just "details" - subtract average
   \[
   \begin{bmatrix}
   -\frac{1}{2} & 1 & -\frac{1}{2}
   \end{bmatrix}
   \begin{bmatrix}
   -a & 2a & -a
   \end{bmatrix}
   \]
   unsharp mask - subtract a blurred version
   doesn't create details - just enhanced what's there.

Directional Smear / Strobe (copy) / ....

Fun Facts about Convolution -- requires flip
   \[f*(g*h) = (f*g)*h\] associative
   \[f*g = g*f\] commutative

Separability