Image Processing II

Michael A. Wirth, Ph.D.
University of Guelph
Computing and Information Science
Image Processing Group

Image Combination

- Arithmetic combination is applied on a pixel-by-pixel basis.
  - The two images must have comparable dimensions.
  - If not then image1 w1× w2, image2 h1× h2
  - The new image will have dimensions w × h

$$w = \min(w_1, w_2)$$

$$h = \min(h_1, h_2)$$

Image Math

- Useful for masking and compositing of images
- Two types of image combination:

  arithmetic (image math) → grayscale images

  logical (boolean) → binary images

Image Addition

- Image addition superimposes information
  - Pixels in the resulting image have values in the range 0-510
  - Normalize the resulting image
    - divided by two → image averaging or converted to 16-bit
  - Primarily used for noise removal
- "Alpha blending"
  - Give more emphasis to one image than the other

$$g(x,y) = \alpha f_1(x,y) + (1-\alpha) f_2(x,y)$$

  - When α=0.5, g(x,y) becomes a simple, even-weighted average

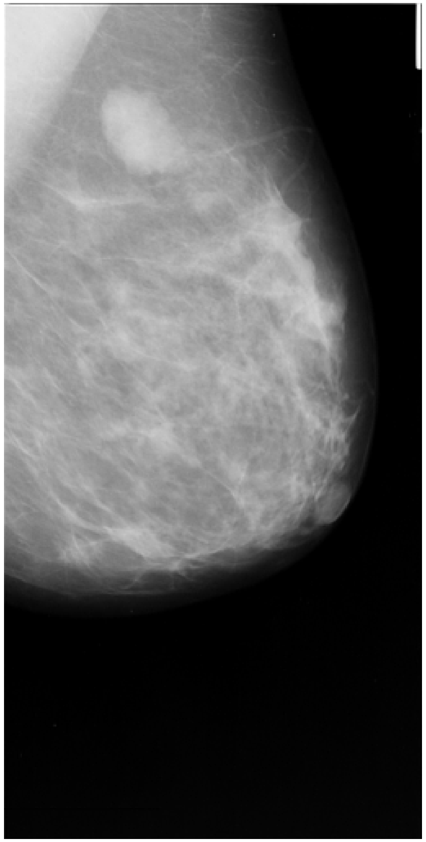
Image Addition

  - Every pixel can have its own α stored in a separate α-channel
- Dynamically rescale the result

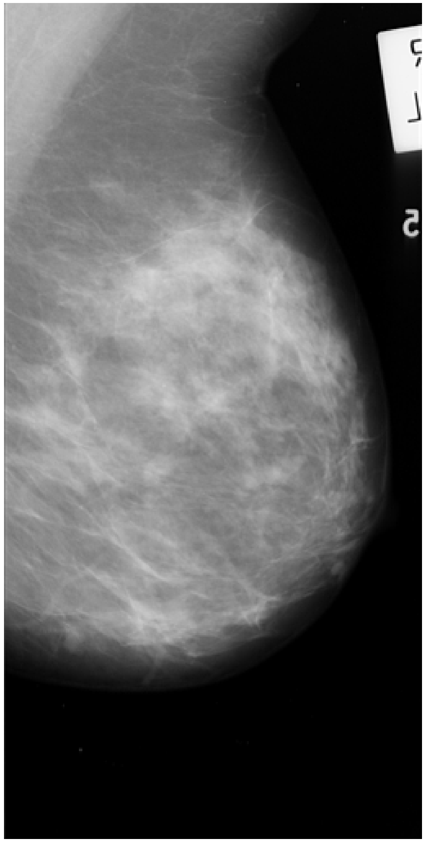
$$g'(x,y) = 255 \times (g(x,y)/\min(g))/(\max(g) - \min(g))$$

- Sometimes addition is handled directly by changing the central value of the kernel

Image Addition



+



=

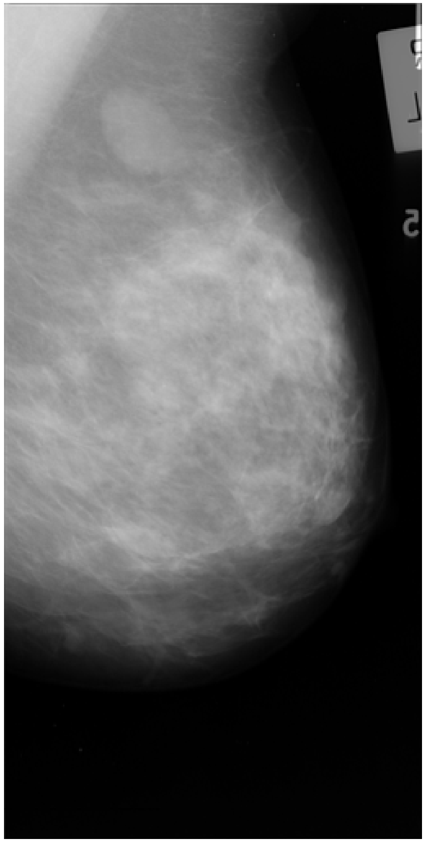


Image Subtraction

- Image subtraction calculates the differences between images
  - Used primarily for change detection
- Pixels in the resulting image have values in the range –255 to +255

$$g(x,y) = \left|f_1(x,y) - f_2(x,y)\right|$$

- Changes will be indicated by pixels in the difference image which have non-zero values.
  - The difference image will contain only features that change

Image Subtraction

- Sensor noise, slight intensity changes, and various other factors result in small differences which are of no significance.
- It is usual to apply a threshold to the difference image.
- Object motion can be measures through subtraction

    e.g. track the motion of cells in response to chemical cues.

Image Subtraction

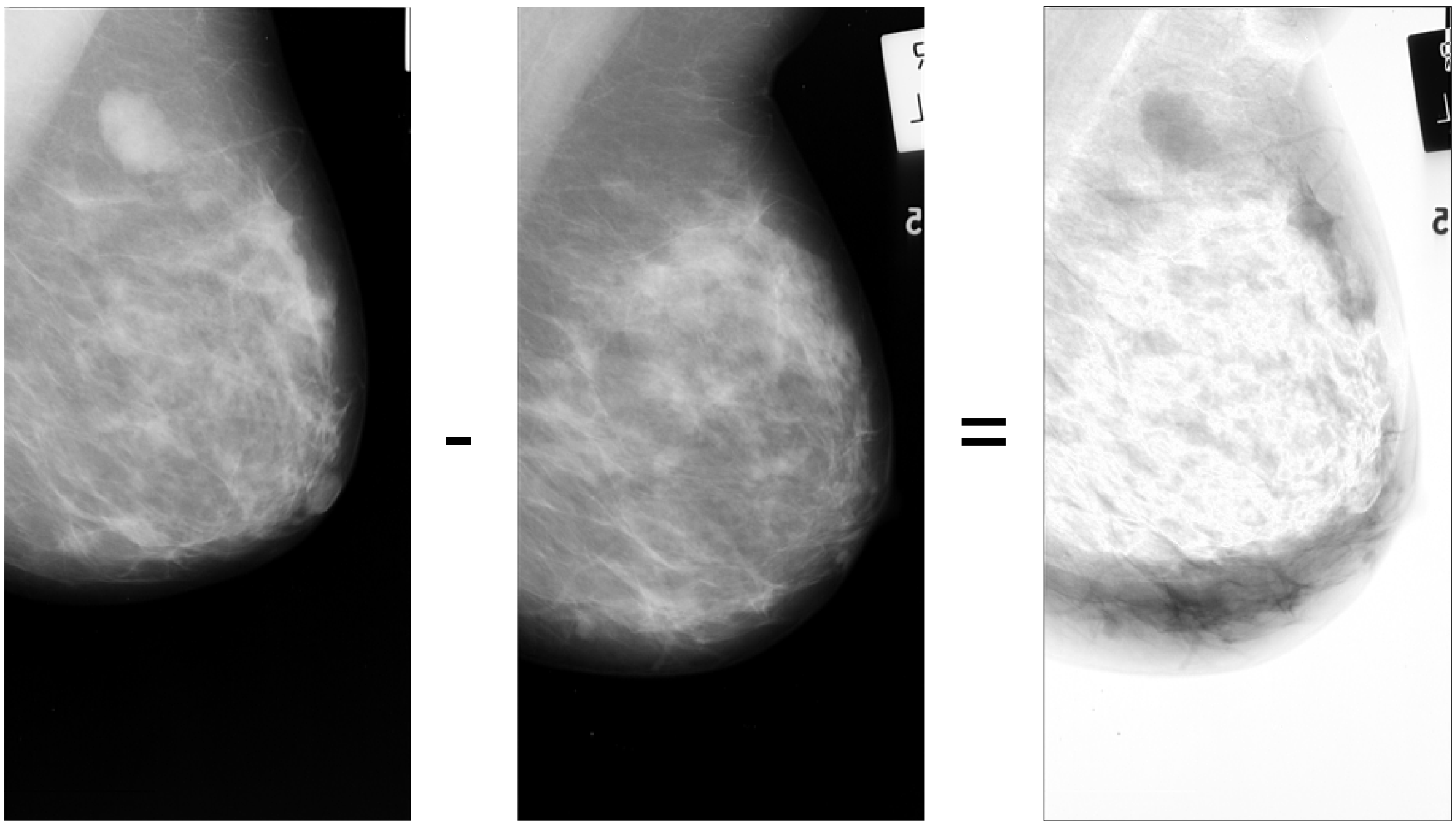


Image Division

- Image division is used for removing backgrounds when linear detectors or cameras are used.
  - For meaningful results use floating-point arithmetic
  - Produces a ratio image in which the pixels should be rescaled and rounded → normalise

$$g(x,y) = f_1(x,y) / f_2(x,y)$$

  - Pixels of 0 intensity are removed from f_2, adding a constant of unity to produce

$$f_2 \equiv f_2 + C, \qquad C = 1$$

Image Division

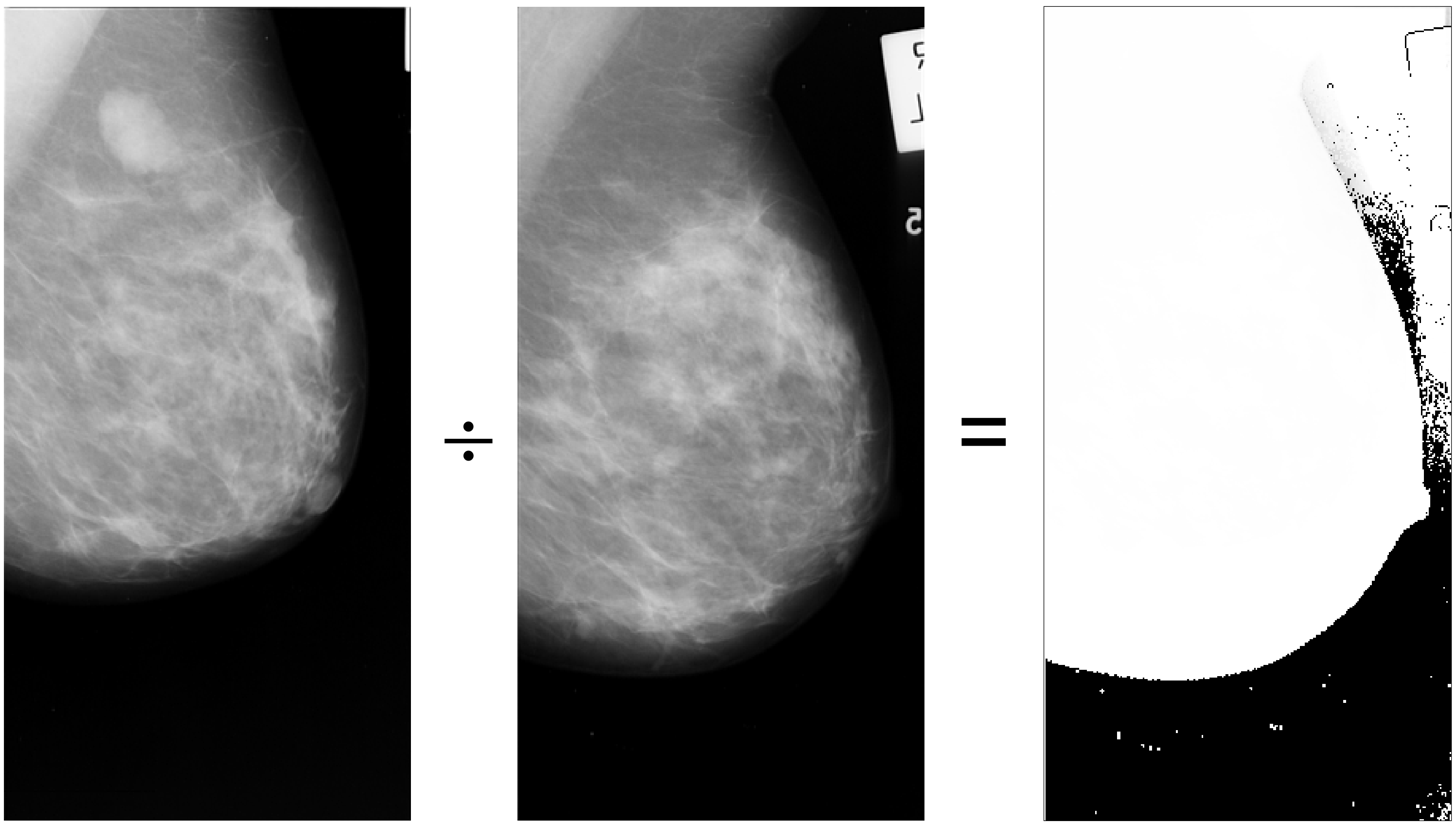


Image Multiplication

- Image multiplication is used for superimposing information

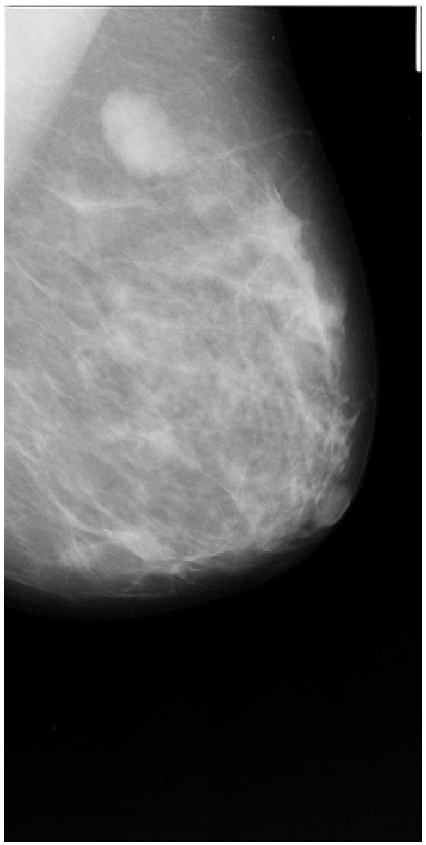
$$g(x,y) = f_1(x,y) \times f_2(x,y)$$

  - Results in an extreme range of values: 0→255 becomes 0→ 65,000
  - Loss of precision in rescaling

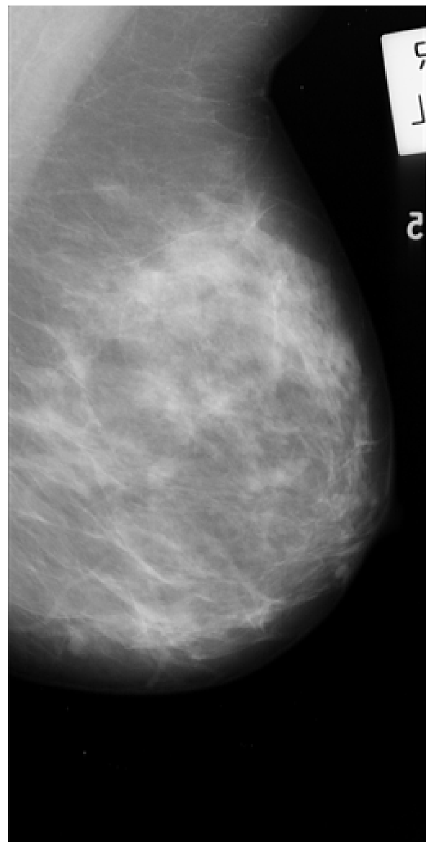
  e.g. Combine edge and direction information from Sobel edge detection

  e.g. Add fluorescence or other emission images to a reflection or transmission image

Image Multiplication



×



=

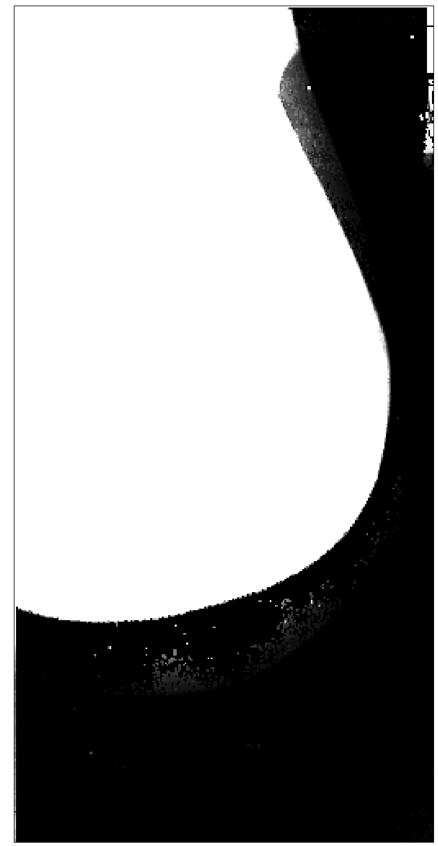


Image Minimum & Maximum

- Image combination using min (or max) involves retaining the darker (or lighter) intensity values at each location

$$g(x,y) = \min(f_1(x,y), f_2(x,y))$$

e.g. To build up a confocal scanning light microscope (CSLM) image with greater depth of field

Image Minimum

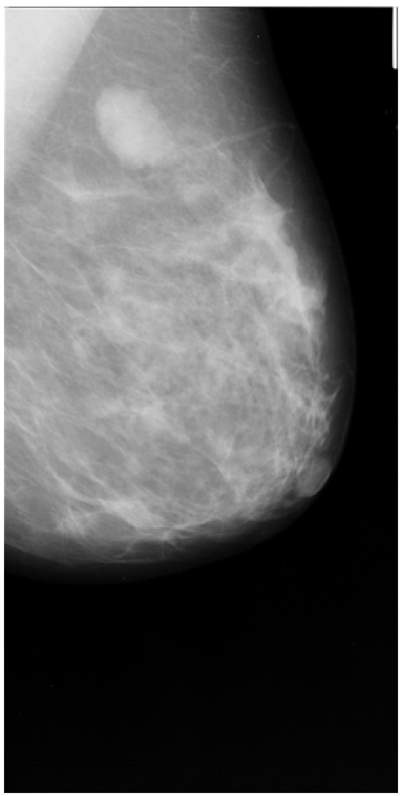
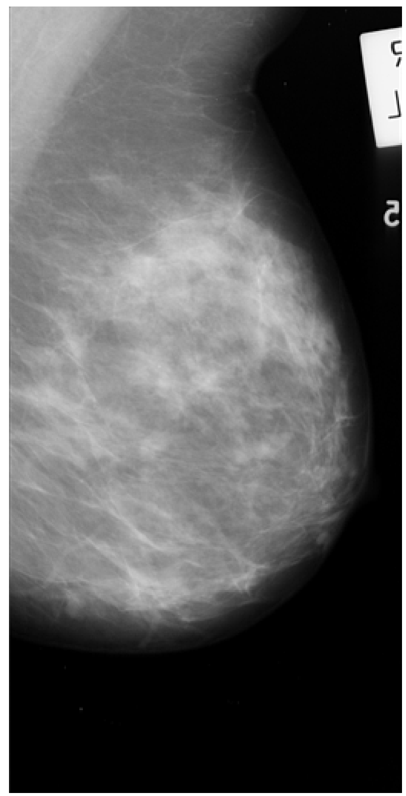
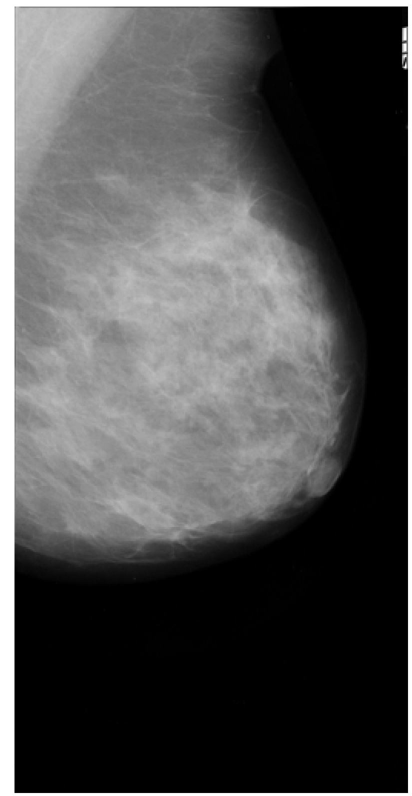
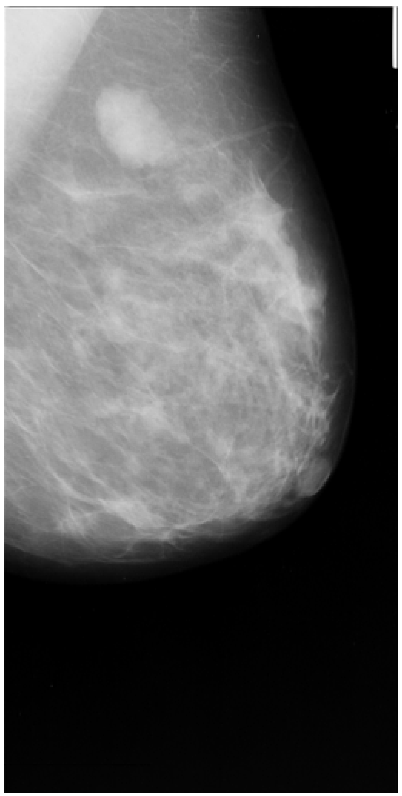
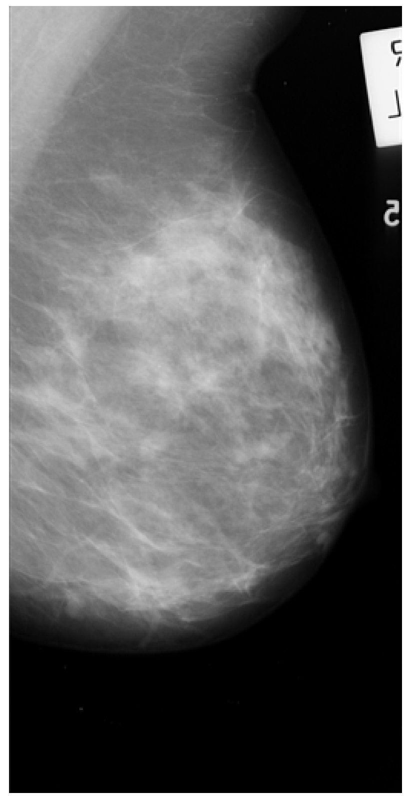
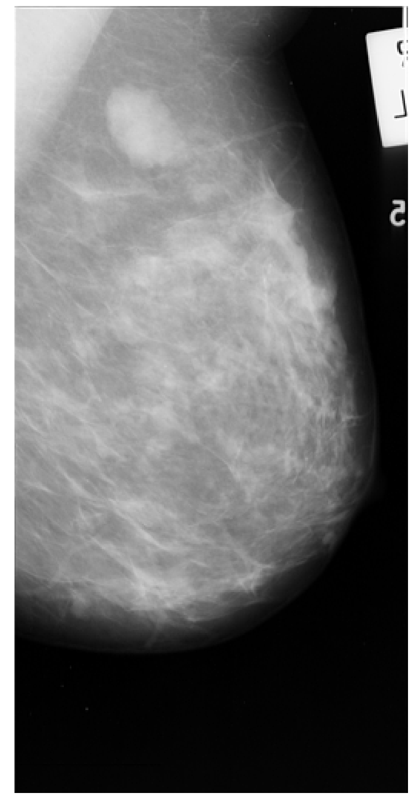
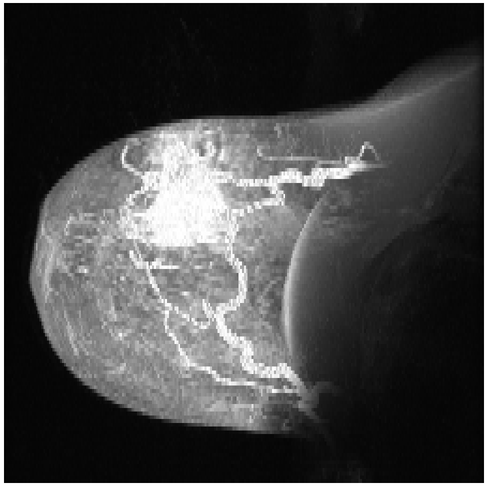
min ( ) = 

Image Maximum

max ( ) = 

Additional Effects

- Combining *n* images
  - Shifting each image slightly before performing combination, produces a perspective view of the surface.
  - Use the max function to combine images
    - e.g. maximum intensity projections



Logical Combination

- Logical or boolean combinations are usually applied to binary images

$$g(x,y) = f_1(x,y) \odot f_2(x,y)$$

- Operations
  - AND, OR, XOR, NOT

AND

- The pixel at location (x,y) is 1 if it is 1 in both images $f_1(x,y)$ and $f_2(x,y)$.
  - All pixels common to both images

$$g(x,y) = (f_1 \text{ AND } f_2) = 1$$
$$\text{if } f_1(x,y) = f_2(x,y) = 1$$

OR

- The pixel at location (x,y) is 1 if it is 1 in either of the images $f_1(x,y)$ or $f_2(x,y)$.

$$g(x,y) = (f_1 \text{ OR } f_2) = 1$$
$$\text{if } f_1(x,y) = 1 \text{ OR } f_2(x,y) = 1$$

XOR

- Exclusive-OR
- The pixel at location (x,y) is 1 if it is 1 in either of the images $f_1(x,y)$ or $f_2(x,y)$, but not if it is 1 in both.

$$g(x,y) = (f_1 \text{ XOR } f_2) = 1$$
$$\text{if } f_1(x,y) = 1 \text{ AND } f_2(x,y) = 0,$$
$$\text{or } f_1(x,y) = 0 \text{ AND } f_2(x,y) = 1$$

Rank Filtering

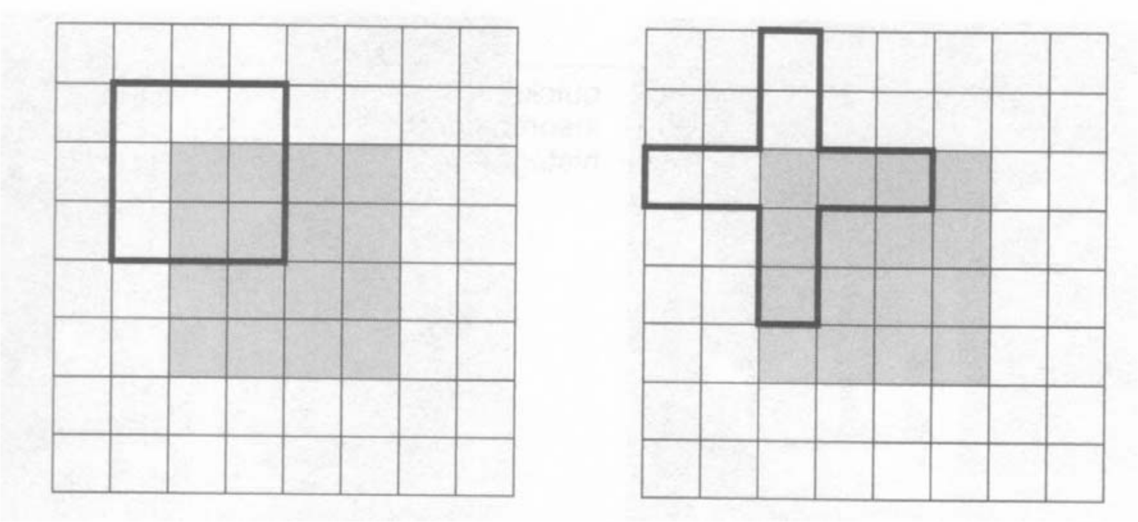
- Non-linear filters known collectively as "order statistic" filters or rank filters
- Compile a list of intensity values in the neighborhood of a given pixel, sort this list into ascending order, then select a value from a particular position in the list to use as the new value for the pixel.

Median Filter

- Select the middle-ranked value from a neighborhood
  - For an $n \times n$ neighborhood, with n odd, the middle value is at position: $\left\lfloor \frac{n^2}{2} \right\rfloor + 1$
- Used to eliminate impulse noise
  - If the noisy pixels occupy less than half the area of the neighborhood some features of interest may not survive median filtering unscathed

Median Filter

- Median-filtering is non-specific
  - Any structure that occupies less than half of the filters neighborhood will tend to be eliminated
  - A result of the shape of the filter:



Median Filter

- The median filter does not reduce the brightness difference
  - used in neighborhood values, it is not an average.
- Does not shift boundaries

Minimum Filter

- The minimum filter is a rank filter in which the lowest (darkest) intensity value from the neighborhood is selected
  - Causes darker regions of an image to increase in size and dominate the lighter regions
  - Also known as *grayscale dilation*

Maximum Filter

- The maximum filter is a rank filter in which the highest (brightest) intensity value from the neighborhood is selected
  - Causes brighter regions of an image to increase in size and dominate the darker regions
  - Also known as *grayscale erosion*

Range Filter

- The range filter is a rank filter in which the difference between the maximum and minimum intensity values in a neighborhood is selected
  - An omnidirectional, non-linear edge-detector

Hybrid Median

- Edge-preserving median
  - In a 5×5 neighborhood, pixels are ranked into two different groups (*a* and *b*)
  - Median values from both groups are compared to the central pixel
  - The median of that set is the new pixel value

b		*a*		*b*
	b	*a*	*b*	
a	*a*	X	*a*	*a*
	b	*a*	*b*	
b		*a*		*b*

Median Filtering

- Repeated application of the median filter can cause *posterization*
  - Reducing the number of intensity values so that regions become uniform in intensity and edges between regions become abrupt
- Extremum filters replace the pixel value with either the maximum or minimum, whichever is closer to the mean value.

Mode Filter

- The mode of the distribution of intensity values in each neighborhood is the most likely value

  ≈ truncated median filter

  – For an asymmetric distribution the mode is the highest point.
- To calculate the mode filter:
  – Discard a few values from the neighborhood so that the median is shifted towards the mode.

Mode Filter

- For example:
  - In a 3×3 neighborhood, discard the two intensity values which are most different from the mean.
  - Rank the remaining seven.
  - Assign the median to the central pixel.
- Has the effect of sharpening steps.

Hybrid Filters

- Hybrids of linear and nonlinear filters
- α-trimmed Mean Filter
  - Sorts values from a neighborhood into ascending order, discards a certain number of these values from either end of the list and outputs the mean of the remaining values
  - If the ordered set of values is $f_1 \le f_2 \le \ldots \le f_{n^2}$ then the α-trimmed mean is:

$$\frac{1}{n^2 - 2\alpha} \sum_{i=\alpha+1}^{n^2-\alpha} f_i$$

Hybrid Filters

- The parameter α is the number of values removed from each end of the list.
  - It can vary between 0 and $\frac{n^2-1}{2}$
  - α=0 (mean filter), $\alpha = \frac{n^2-1}{2}$ (median filter) → in between a compromise between mean and median filters.

Adaptive Filters

- Properties of an image can vary spatially
- e.g. Gaussian random noise in an image
  - Normal smoothing is effective in homogeneous regions, adverse blurring effect in regions that are meant to be heterogeneous (due to the presence of edges)
- These effects can be minimized using an adaptive filter
  - Most compute local intensity level statistics within the neighborhood of a pixel and base their behavior on this information

Adaptive Filter

- Minimal Mean Square Error Filter

$$g(x,y) = f(x,y) - \frac{\sigma_n^2}{\sigma^2(x,y)}\left[f(x,y) - \bar{f}(x,y)\right]$$

  - σ_n^2 is an estimate of noise variance
  - $\sigma^2(x,y)$ is the intensity variance computed for the neighborhood centred on (x,y)
  - f(x,y) is the mean intensity value in that neighborhood

Frequency Filtering

- Frequency filtering is based on the frequencies of intensities or intensity variation in an image.
  - Convert an image into a spectrum of different frequency components and convert this spectral representation back into a spatial representation without any loss of information.
  - Process the image by manipulating its spectrum
  - Involves two aspects: amplitude and phase

Frequency Filtering

- The discrete Fourier transform (DFT)

$$F(u,v) = \tfrac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\left[\cos\left(\frac{2\pi(ux+vy)}{N}\right) + j\sin\left(\frac{2\pi(ux+vy)}{N}\right)\right]$$

or

$$F(u,v) = \tfrac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux+vy)/N}$$

- F(u,v) is a complex number
- Inverse Fourier transform

$$f(x,y) = \tfrac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{j2\pi(ux+vy)/N}$$

Frequency Filtering

- A forward transform of an N×N image yields an N×N array of coefficients
- Its real and imaginary parts are not informative in themselves.
  - More useful to think of the MAGNITUDE and PHASE of F(u,v)

$$F(u,v) = R(u,v) + jf(u,v) = \left|F(u,v)\right| e^{j\phi(u,v)}$$

  - R(u,v) and I(u,v) are th real and imaginary parts respectively

Frequency Filtering

- $|F(u,v)|$ is the magnitude, $\phi(u,v)$ is the phase

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

- Magnitudes correspond to the amplitudes of the basis images
  - array of magnitudes → amplitude spectrum
  - array of phases → phase spectrum

Frequency Filtering

- "Power spectrum" or spectral density
  - Simply the square of its amplitude spectrum

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

- Computational Considerations
  - $N \times N \rightarrow O(N^2)$ operation, with N^2 values of $F(u,v)$ to calculate $\rightarrow$ overall complexity is $O(N^4)$
  - Assume the multiplication of a complex number consumes 1 microsecond of CPU time
  - $\approx$70min for 256^2 image, 12days for 1024^2 image

Fast Fourier Transform (FFT)

- Separable transform
  - 1D FFT along each row → generates an intermediate image → 1D FFT down each column
- Complexity of the Fourier transform from $O(N^4)$ to $O(N^3)$
- Requires that both image dimensions are powers of 2.
  - Padding with zeros (does not affect the spectrum)

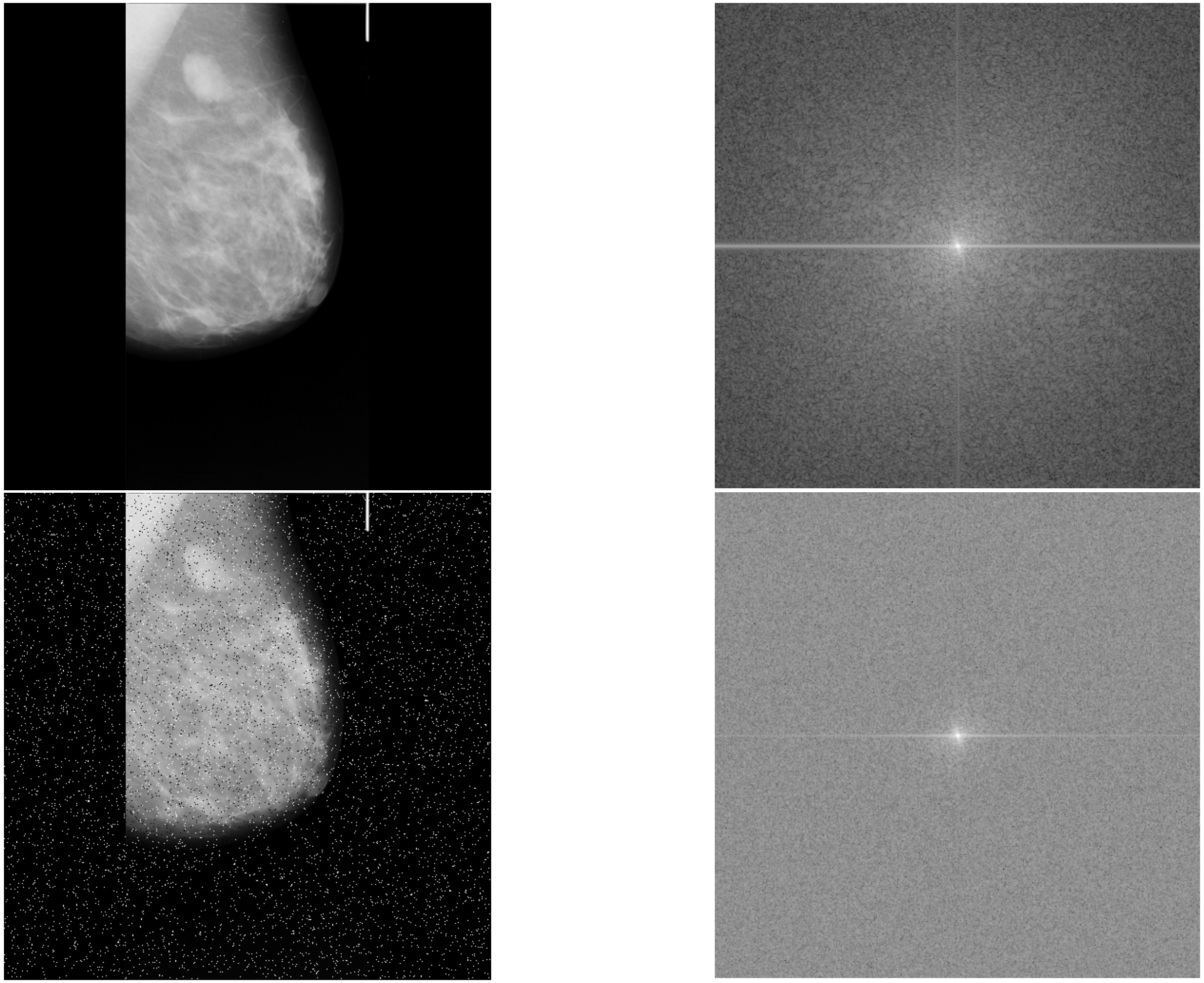
FFT

- Visualization of the amplitude spectrum
  - logarithmic mapping

$$\left|F(u,v)\right|' = C\log\left[\left|F(u,v)\right| + 1\right]$$

  - add one to $\left|F(u,v)\right|$ because the spectrum can be 0 in places, and the logarithm of zero is undefined

Example of FFT: Power Spectrums



FFT Filtering

- Can be expressed generally as the point-by-point multiplication of the spectrum by a filter transfer function

$$G(u,v) = F(u,v)H(u,v)$$

- Here G(u,v) is the filtered spectrum, F(u,v) is the spectrum and H(u,v) is the filter transfer function.
- Most filters are zero-phase-shift filters → they affect magnitude rather than phase

FFT Filtering

- Given a kernel, convolve that kernel with an image, or filter via the following procedure
  - Compute the Fourier transform of the image
  - Computer the Fourier transform of the kernel
  - Multiply the two transforms together
  - Compute the inverse Fourier transform of the product

Lowpass Filtering

- The simplest *lowpass filter* is a filter that "cuts off" all high frequency components of the Fourier transform that are at a distance greater than a specified distance from the origin.
  - Such a filter is called an *ideal lowpass filter*

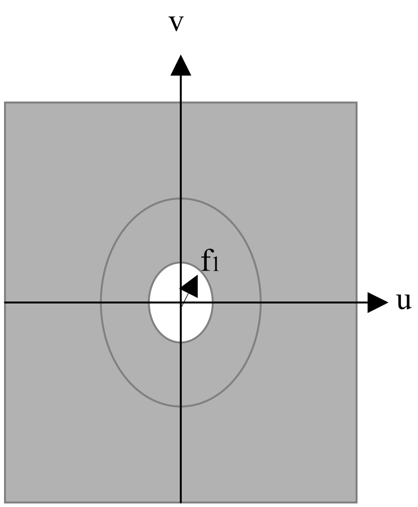
Ideal Lowpass Filter

- Frequency increases outward from the centre of a shifted spectrum.
  - Force F(u,v) to zero at some distance from the centre.

$$H(u,v) = \begin{cases} 1 & r(u,v) \le r_0 \\ 0 & r(u,v) > r_0 \end{cases}$$

  - r_0 is the filter radius and r(u,v) is the distance from the centre of the spectrum

$$r(u,v) = \sqrt{u^2 + v^2}$$



Butterworth Lowpass Filter

- Ideal filters can cause “ringing” – ripple-like effects.
- The Butterworth lowpass filter does not have a sharp discontinuity that establishes a distinct cutoff between passed and filtered frequencies.

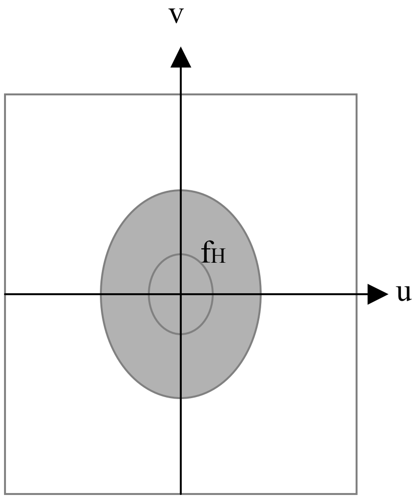
Highpass Filtering

- Image sharpening can be achieved through highpass filtering which attenuates the low-frequency components without disturbing the high-frequency information.

Ideal Highpass Filter

$$H(u,v) = \begin{cases} 0 & r(u,v) < r_0 \\ 1 & r(u,v) \geq r_0 \end{cases}$$

- Has an inverted cylindrical shape
- Can cause ringing



Band Pass & Band Stop Filtering

- A band-pass filter passes a specific range of frequencies whilst suppressing others, a band-stop filter has the opposite effect.
- Band-stop filter

$$H_S(r) = \frac{1}{1 + \left[\Omega r / (r^2 - r_0^2\right]^{2n}}$$

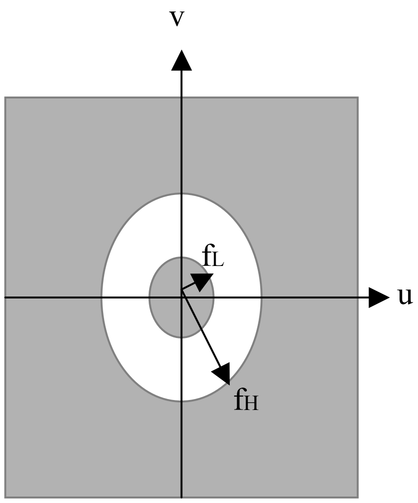
- $r = \sqrt{u^2 + v^2}$, r_0 is the radius

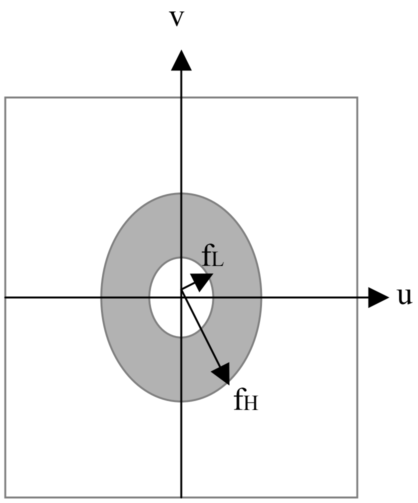
of the band centre Ω is the band width

Band Pass & Band Stop Filtering

- Band-pass filter

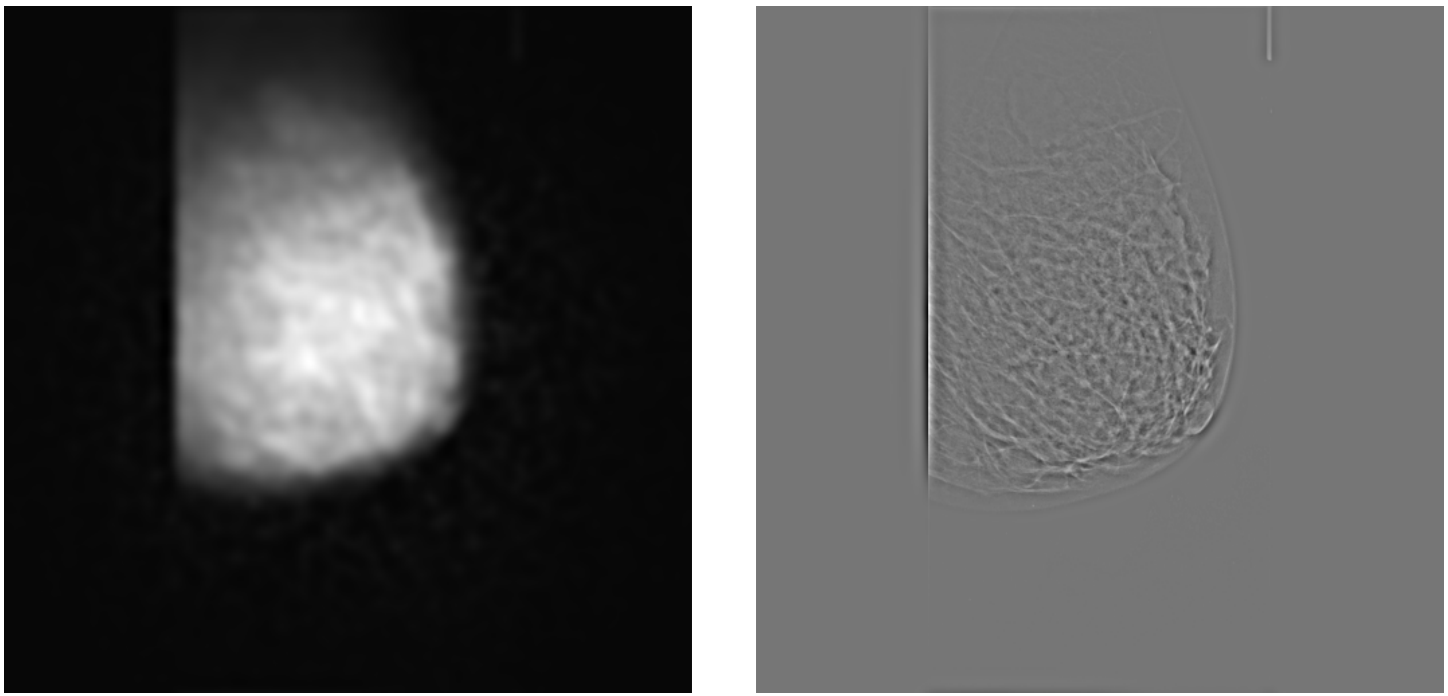
$$H_P(r) = 1 - H_S(r)$$





Example of FFT Filtering

- Low-pass & High-pass Butterworth (n=3)



Pseudocoloring

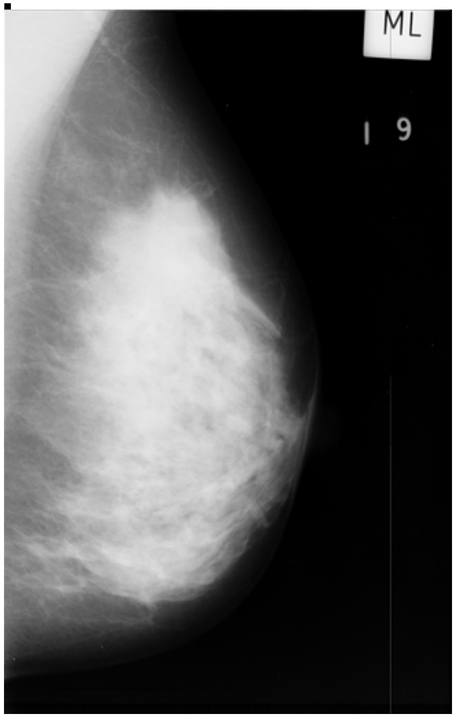
- *Pseudocoloring* (also called *false coloring*) consists of assigning colors to gray values based on a specified criterion.
  - The term *pseudo* is used to differentiate the process of assigning colors to grayscale images from the processes associated with true-color images.
  - The principal use of pseudocolor is for human visualization and interpretation.

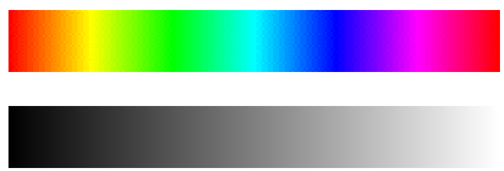
Pseudocoloring

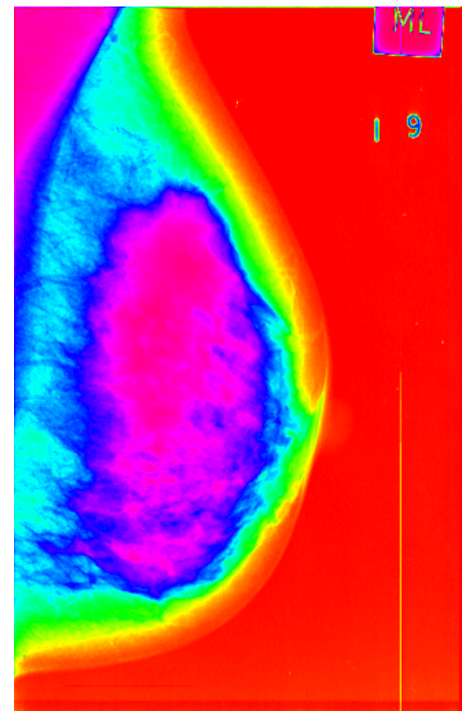
- One of the principal motivations for using color:
  - The human eye can detect only in the neighborhood of 30 grayscale intensity levels at any point in an image due to brightness adaptation.
  - However it can differentiate thousands of color shades and intensities.

Pseudocoloring

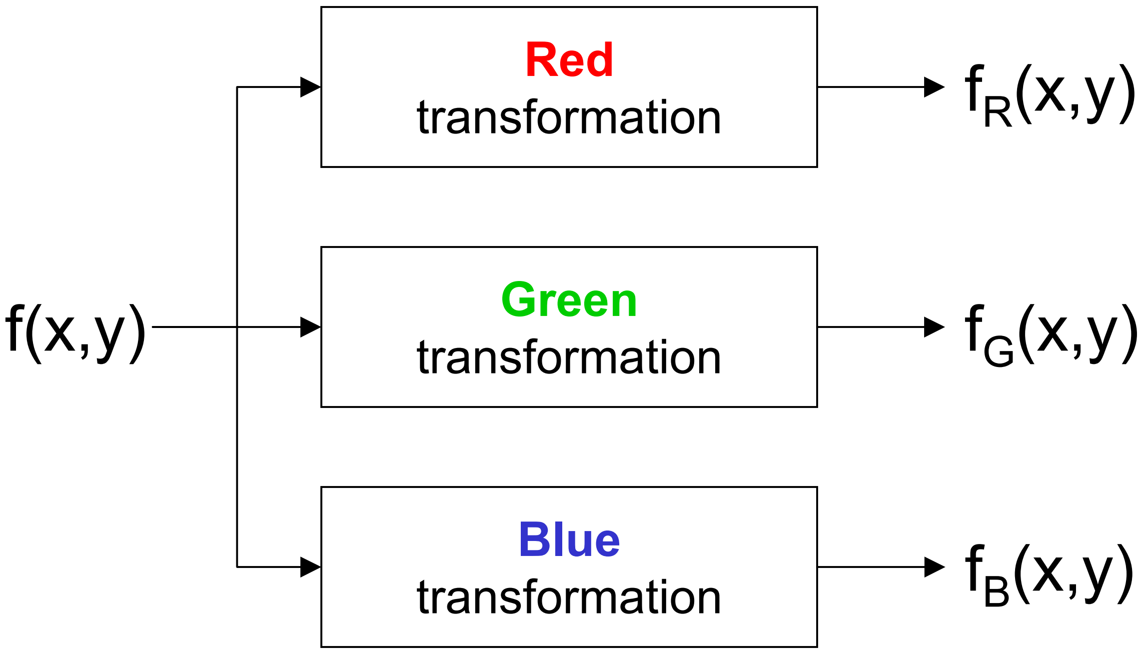
- Regions that appear of constant intensity in the grayscale image are really quite variable, as shown by the various colors in the RGB





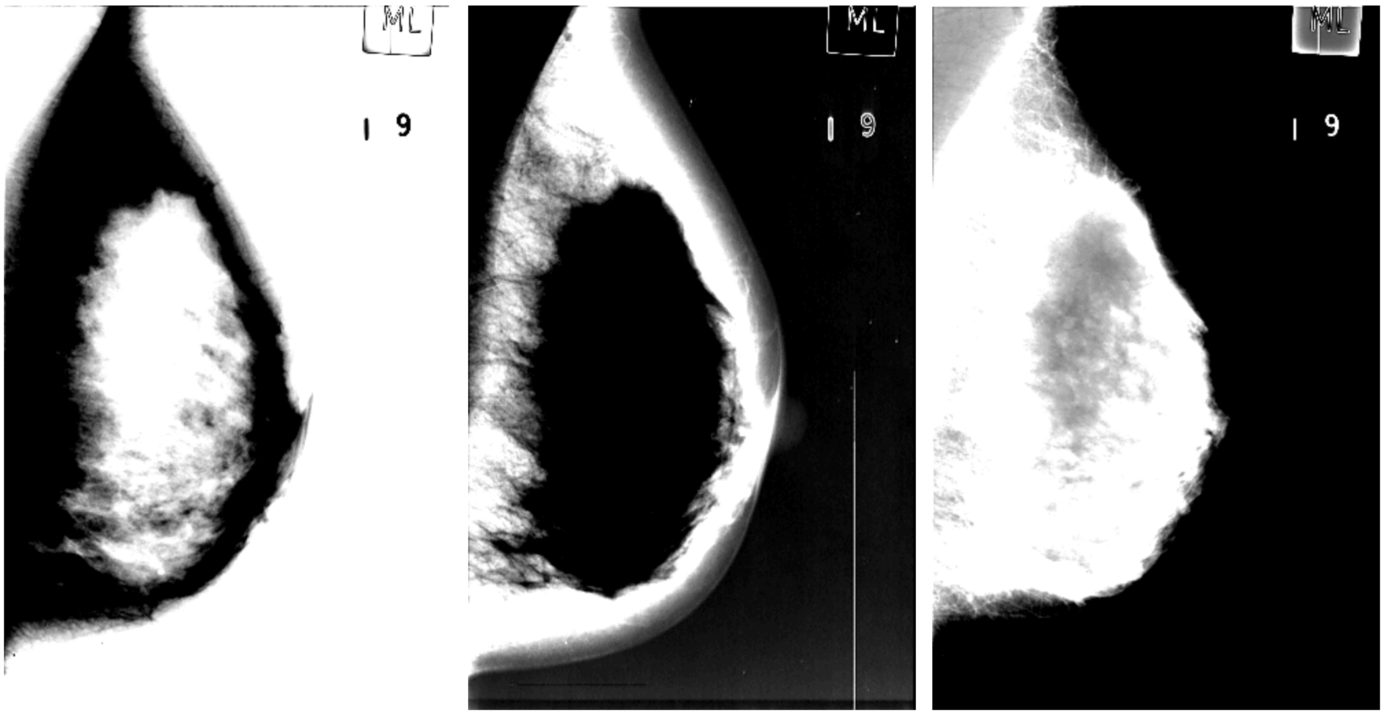


Gray-to-Color Transformation



Pseudocoloring

- Components of the RGB image.



(a) red (b) green (c) blue