Ch 3 - Intensity Transformations

- Overview
- Linear Map
- Inverse Map
- Thresholding
- Windowing
- Log Transformation
- Power Law Transformation
- Contrast Stretching
- Intensity Slicing
- Bit Plane Images
- Conclusion

Overview

- Goal is to modify pixel intensity to improve the visibility of objects of interest in an image
- General form of operation:

```
For (int y = 0; y < Ydim, y++)
  For (int x = 0; x < Xdim; x++)
    Output[y][x] = Map( Input[y][x] )
```

- Different Map functions give different effects
- Slope of curve determines enhancement effect

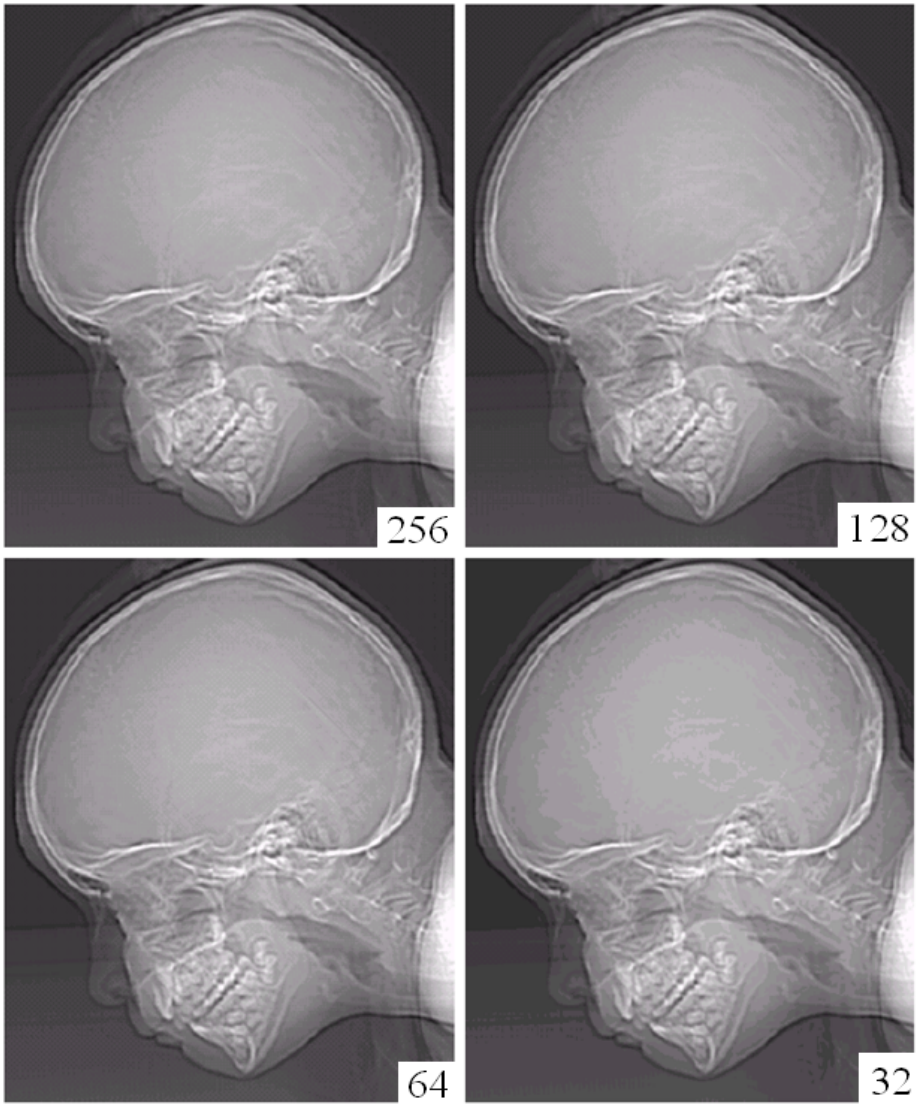
Linear Map

- A linear map function is used to compress or stretch the intensity range of pixels in an image
- Typically used prior to data storage or display
- Map(i) = A * i + B
- Example:
  - Input range = $[I_{min}..I_{max}]$
  - Output range = $[O_{min}..O_{max}]$
  - $Map(i) = (O_{max}–O_{min}) * (i–I_{min}) / (I_{max}–I_{min})+O_{min}$

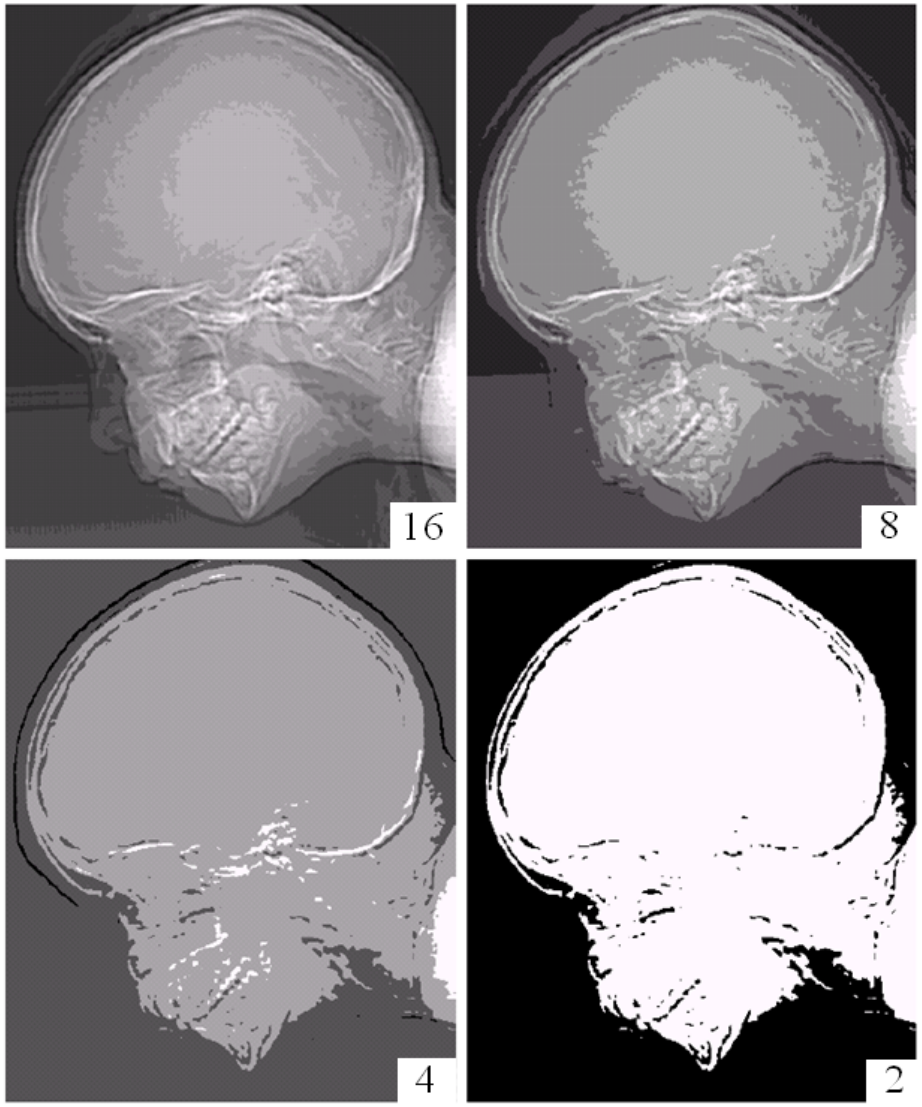
Linear Map

- When the linear map function has A < 1 the output range is smaller input range
- When pixels are stored in integers this results in quantization and loss of information
- Example:
  - Input range [0..255]
  - Map(i) = i / 2 yields 128 intensity levels
  - Map(i) = i / 4 yields 64 intensity levels
  - Map(i) = i / 8 yields 32 intensity levels

Linear Map

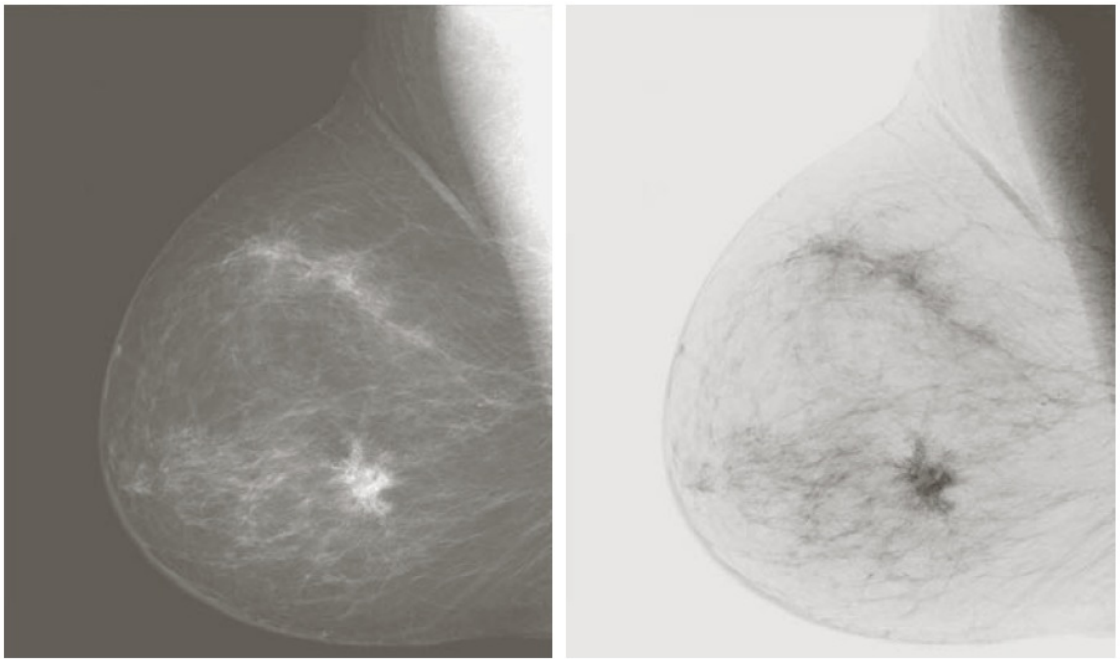


Linear Map



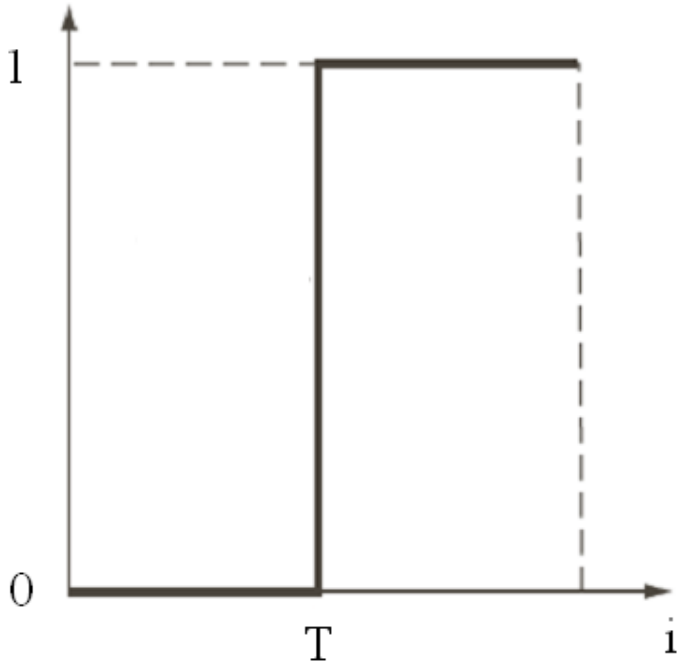
Inverse Map

- An inverse linear map will convert a negative image into a positive image (and vice versa)
- Map(i) = 255 – i



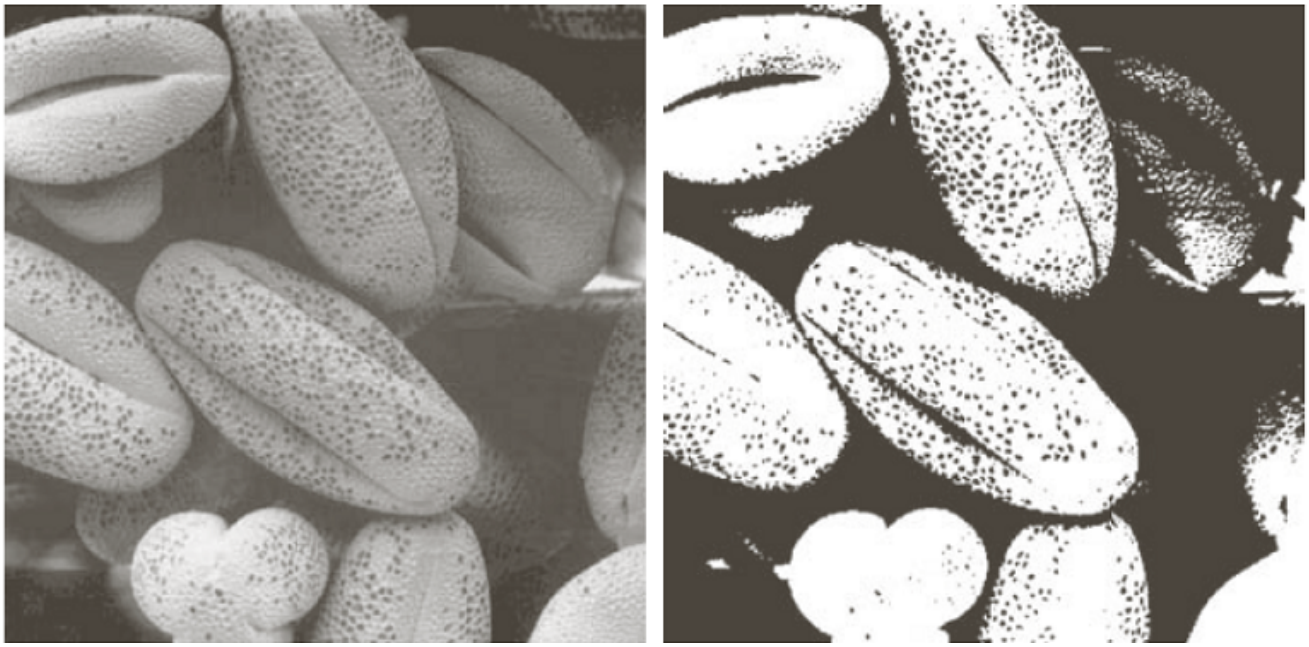
Thresholding

- An intensity transformation which produces a black and white output image
- Map(i) = 0 if i < T
  1 if i >= T



Thresholding

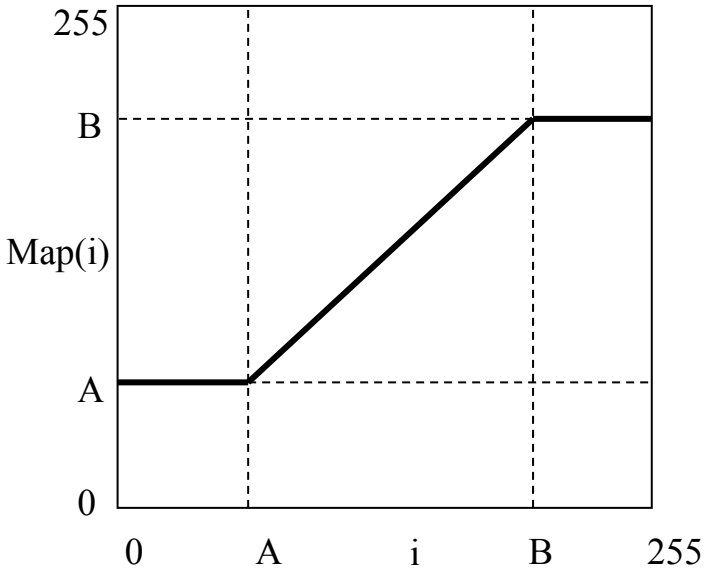
- By selecting the threshold T we can separate objects of interest from the background



Windowing

- When objects of interest have intensity values in the range [A..B] we can exclude all pixels outside this range to enhance object visibility

- Map(i) = A if (i < A)
  B if (i > B)
  i otherwise



Windowing

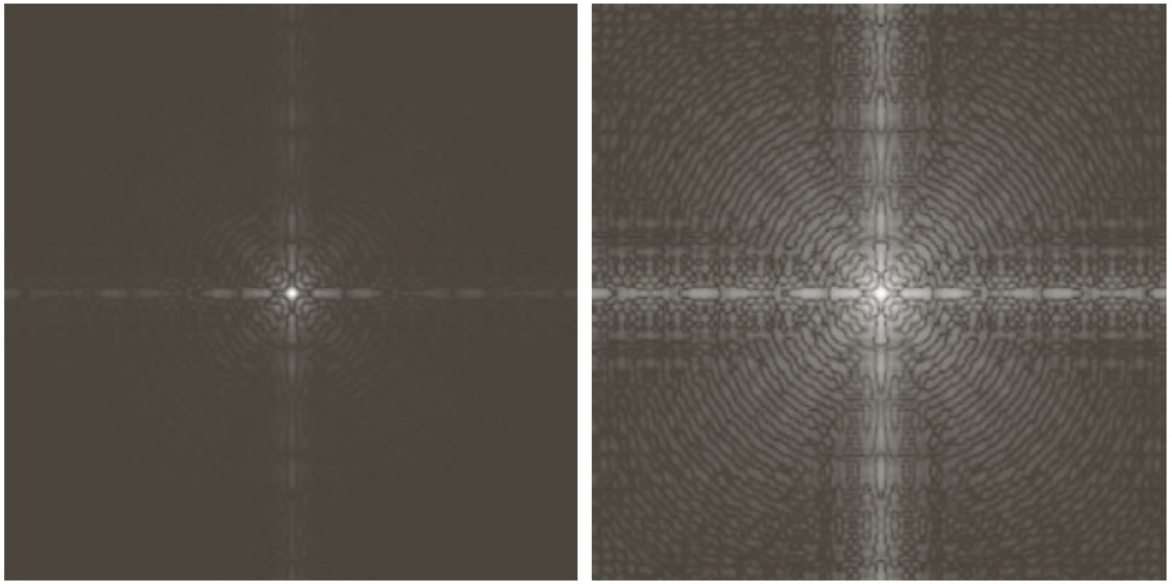
Windowing

Log Transformation

- The log function can be used to reduce the intensity range in an image
- The inverse log function can be used to expand the intensity range in an image
- Map(i) = c * log(1 + i)
- Example:
  - Input range [a..b]
  - Output range [0..255]
  - Map(i) = 255 * log(1 + i - a) / log(1 + b - a)

Log Transformation

- Log transformations often used to display the Fourier transform of an image

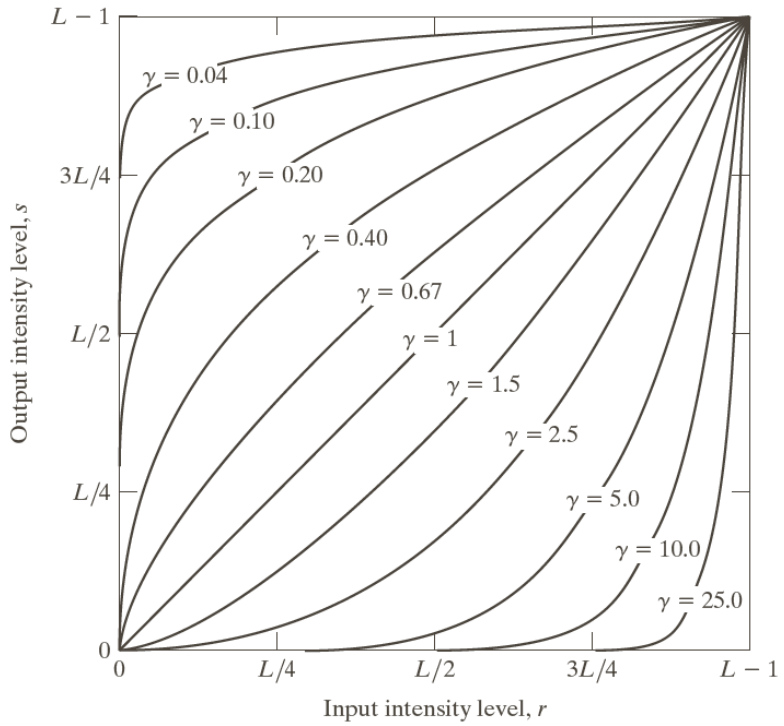


Power Law Transformation

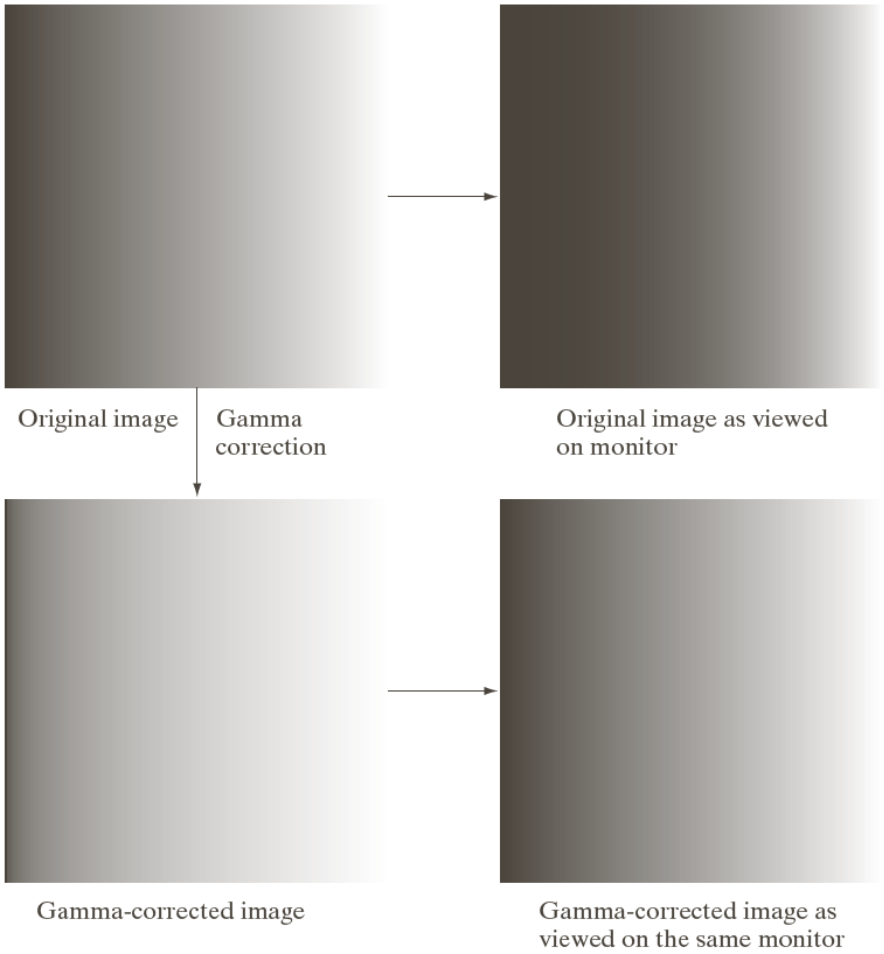
- A power law function (gamma function) stretches and compresses intensities to improve object visibility in an image
- $Map(i) = c * i^{\gamma}$
- Example:
  - Input range [a..b]
  - Output range [0..255]
  - $Map(i) = 255 * (i-a)^{\gamma} / (b-a)^{\gamma}$

Power Law Transformation

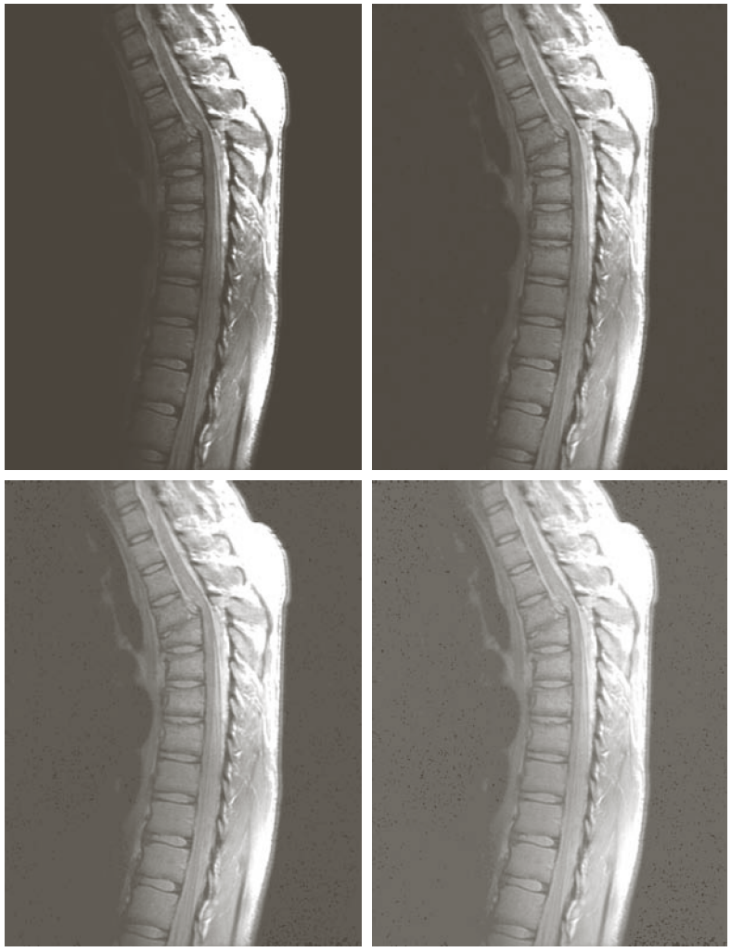
- We get different effects by changing power γ
- Slope > 1 enhanced, slope < 1 de-enhanced



Power Law Transformation



Power Law Transformation



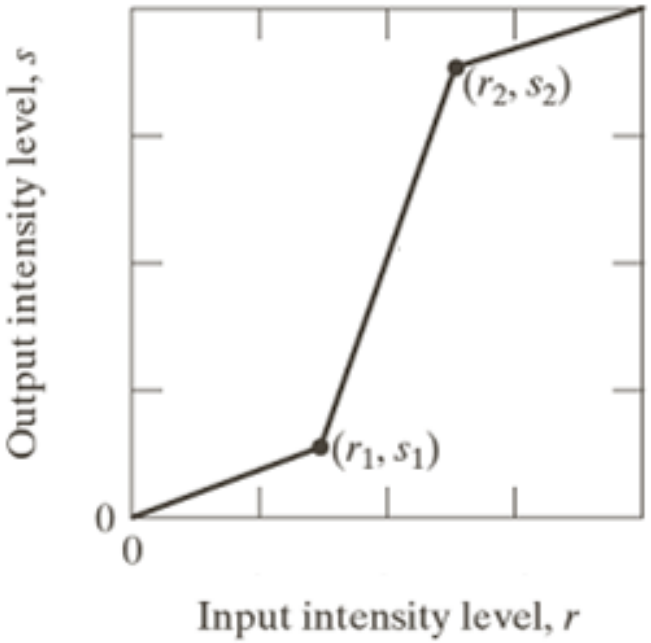
Power Law Transformation

Contrast Stretching

- Sometimes we know the intensity range of objects of interest, and we want to enhance this range without losing all information about intensities outside this range
- Contrast stretching uses a piecewise linear Map(i) function to accomplish this goal

Contrast Stretching

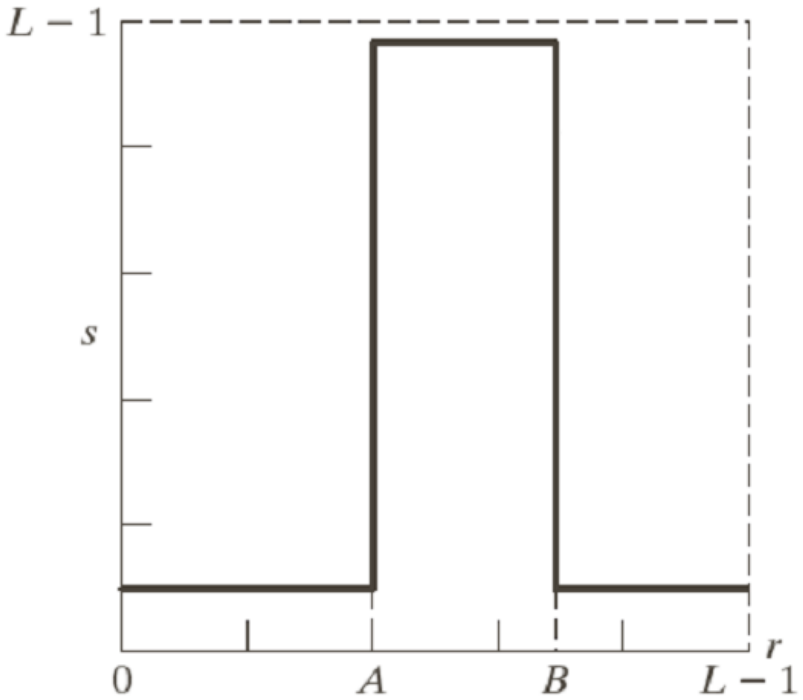
- User specifies points (r1, s1) and (r2, s2)
  - Input [0..r1] is mapped to output [0..s1]
  - Input [r1..r2] is mapped to output [s1..s2]
  - Input [r2..255] is mapped to output [s2..255]



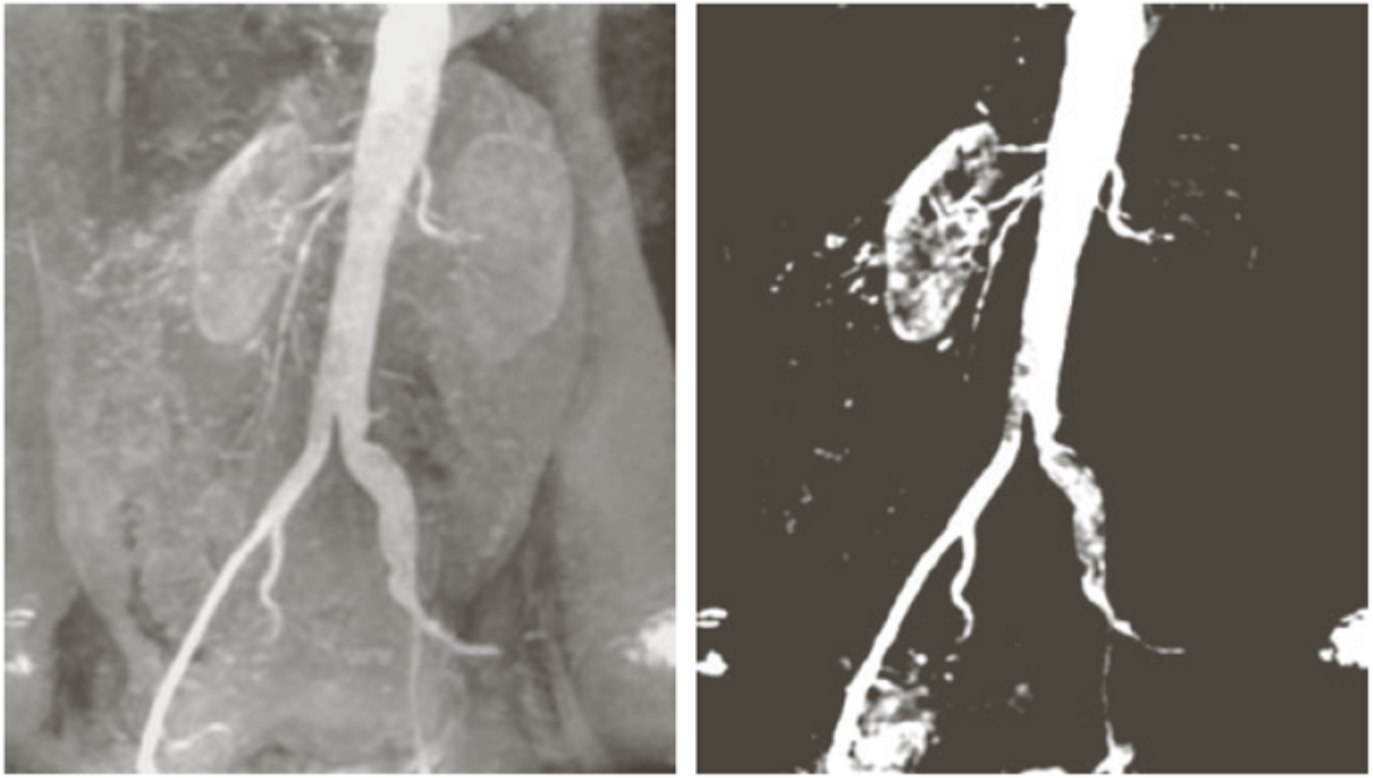
Contrast Stretching

Intensity Slicing

- One way to emphasize pixels with intensity values [A..B] is to make these pixels white, and map intensities outside this range to black
- This is very similar to intensity thresholding

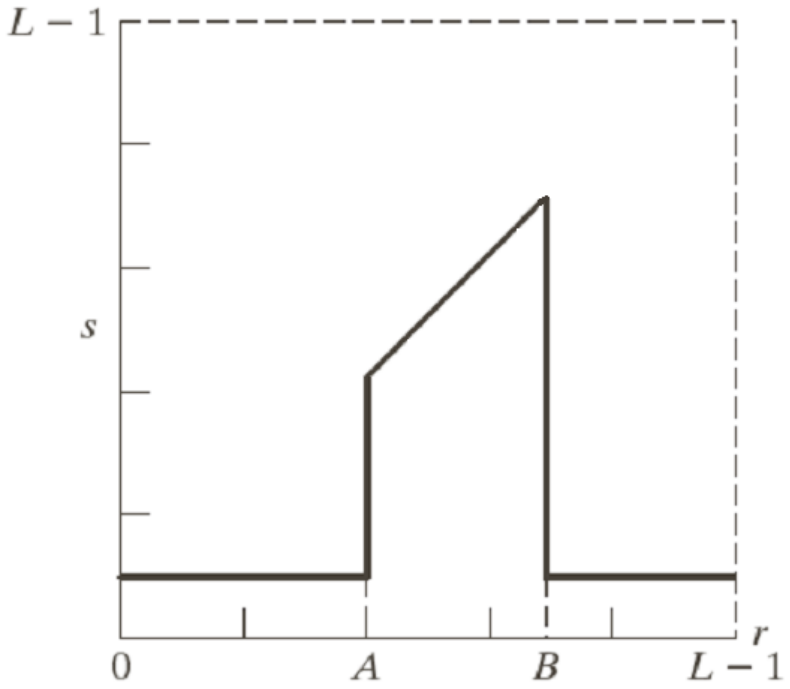


Intensity Slicing

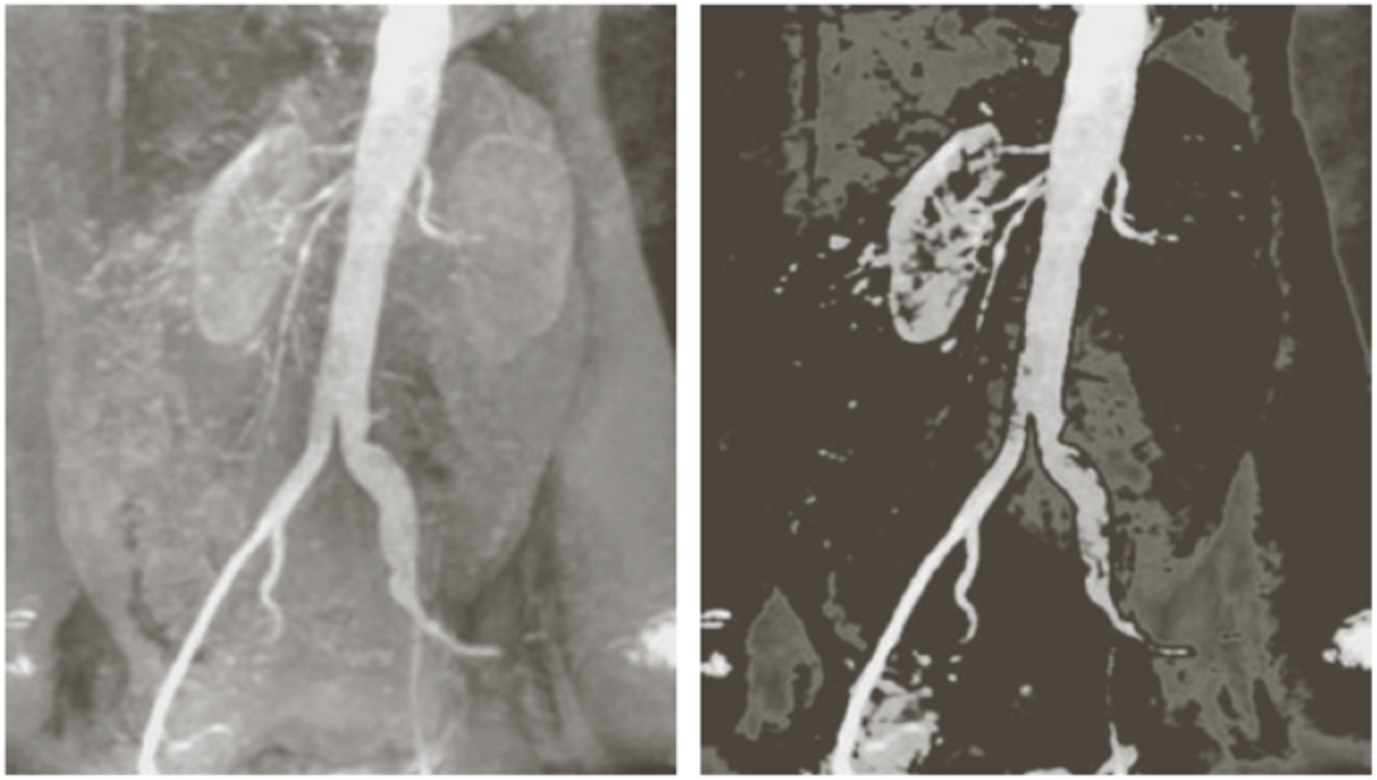


Intensity Slicing

- Another option is to leave intensities [A..B] unchanged and map intensities outside this range to black
- This reduces the amount of ‘background clutter’ in the image

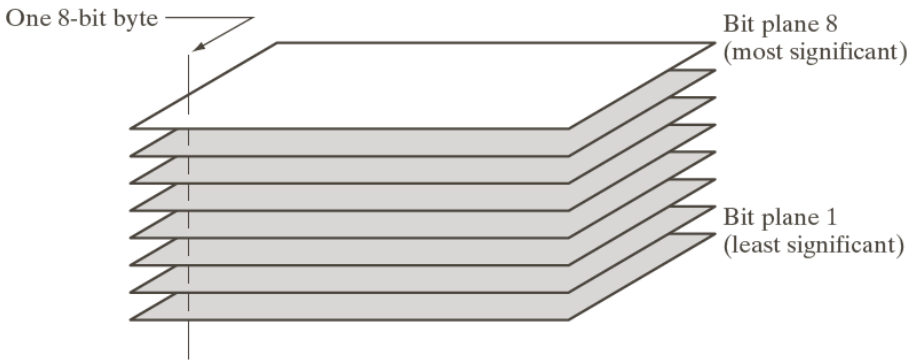


Intensity Slicing



Bit Plane Images

- Some image features can be visualized by looking at the bit plane images
- The most important details should be in the most significant bit plane, the least significant bit plane looks like noise



Bit Plane Images

Bit Plane Images

- Here we have images reconstructed using only the most significant bit planes

bits 7-8

bits 6-8

bits 5-8

Conclusion

- Intensity transformations can produce a wide variety of image enhancement effects
- The key to selecting a technique is understanding how each method modifies the intensity distribution in an image