EE416: Introduction to Image Processing and Computer Vision

Il Yong Chun

Department of Electrical and Computer Engineering, the University of Hawaiʻi, Mānoa

November 3, 2021

5 Image enhancement and spatial filtering

Another application of the fundamental tools developed thus far is **image enhancement**.

What does it mean to "enhance" an image? The purpose of image enhancement [1, Ch. 8] is to "improve" images in terms of

- the visual appearance for human interpretation [1, Ch. 7], or
- the suitability for subsequent computer processing.

$$\text{original input image} \rightarrow \boxed{\text{image enhancement}} \rightarrow \text{output (improved) image} \rightarrow \begin{cases} \text{eye} \\ \text{algorithm} \end{cases}$$

The meaning of "improved" is very application dependent, so it can be difficult to form general theoretical frameworks here. Furthermore, many of the methods are nonlinear, thereby precluding use of the analytical tools developed in previous chapters, although most basic enhancement methods are shift invariant.

So most image enhancement work is qualitative, one exception being **image denoising**. Indeed, Lim [1, p. 452] describes image enhancement as "an algorithm that is simple and ad hoc..."

What are typical ways in which one "enhances" an image? Adjusting brightness and contrast, sharpening, noise smoothing, red eye removal, etc.

Somewhat related to image enhancement is the problem of **image restoration**, where we want to estimate an original image $f[m, n]$ given a related degraded image $g[m, n]$, as follows:

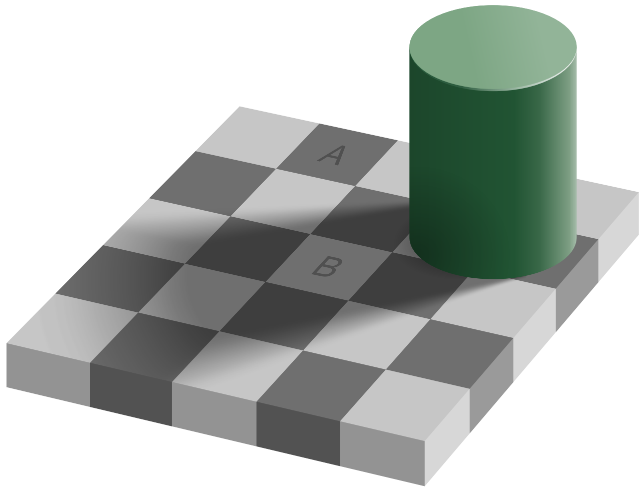
$$f[m,n] \rightarrow \boxed{\text{degradation}} \rightarrow g[m,n] \rightarrow \boxed{\text{restoration}} \rightarrow \hat{f}[m,n]$$

In this problem, our goal is to make $\hat{f}[m, n]$ as "close" to the original $f[m, n]$ as possible. We can quantify "close" and develop "optimal" methods. By comparison, in image enhancement we are often not trying to recover an original; instead we are deliberately modifying the image values to "improve" some characteristics, such as reducing degradations or accentuating certain features.

5.1 Contrast adjustment

Most grayscale displays accept 8-bit inputs in the range 0–255. Often we cannot see image "details" because the displayed grayscale values of an image are "too similar". In particular, the human visual system is nonlinear [1, Ch. 7], and responds differently to intensity contrast depending on the local background intensity.

Example: Here is an illustration of how challenging it can be to try to optimize images for human perception. (Published by Edward H. Adelson of MIT in 1995 [wiki].)



Which patch has higher image intensity values: A or B? $\boxed{??}$

Often human perception of an image can be enhanced by a simple memoryless[1] or point-wise intensity transformation $\mathcal{T} : \mathbb{R} \to \mathbb{R}$

$$g[m,n] = \mathcal{T}(f[m,n]).$$

5.1.1 Linear mapping

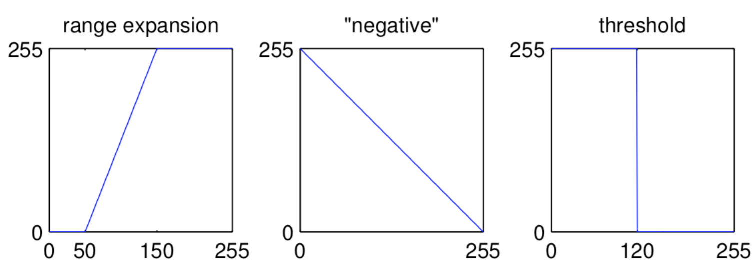
Piecewise linear amplitude adjustment.

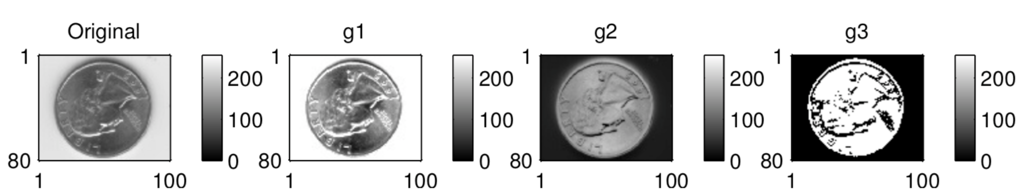
$$g[m,n] = \begin{cases} 0, & f[m,n] < a \\ \frac{255}{b-a}\left(f[m,n] - a\right), & a \le f[m,n] \le b \\ 255, & f[m,n] > b \end{cases} \qquad \text{Graph:}\boxed{??}$$

This is often used to map an image to the range $(0, 255)$ for display. Performed with Matlab's imagesc function. The clim option of imagesc adjusts a and b.

Alternate terminology: **window** $= b - a$ and **level** $= \frac{b+a}{2}$ used routinely in medical imaging.

Example:





[1] A system whose output $g(x,y)$ at any point (x,y) only depends on the input image $f(x,y)$ at the same location (x,y) could be called memoryless, if we stretch the English usage of "memory" somewhat. For example, a detector that converts amplitude to intensity (with no other effects) via the relation $g(x,y) = |f(x,y)|^2$ is memoryless. Another term that is perhaps more apt in imaging is pointwise.

5.1.2 Nonlinear mapping

Such piecewise-linear adjustments are insufficient in many cases, and nonlinear operations can be helpful.

To compensate for nonlinearities in I/O devices like photographic film, scanners, printers, monitors, and the eye, one often applies **gamma correction** to an image prior to display:

$$g[m,n] = (f[m,n])^{\gamma}$$

where $\gamma \in \mathbb{R}$ is a tuning parameter. Matlab's `cmgamma` function is related to this operation.

Input and output nonlinearities in imaging systems.

$$\begin{array}{c}\mathbf{l}\text{: Incoming light intensity} \\ \text{[Vidicon tube for image capturing]}\end{array} \rightarrow \boxed{\text{A/D}} \xrightarrow[\text{Digital image transmission or storage}]{\text{digitized signal } \mathbf{x}} \boxed{\text{D/A}} \rightarrow \begin{array}{c}\tilde{\mathbf{l}}\text{: Displayed light intensity} \\ \text{[Display device]}\end{array}$$

- $x[m,n]$ generally takes values from 0 to 255.
- Vidicon tubes have (had) nonlinear input/output relationship:

$$x = 255 \left(\frac{l}{l_{\text{in}}}\right)^{1/\gamma_i}$$

  where l_{in} is the maximum input and γ_i is a parameter of the input device.
- The cathode ray tube (CRT) has the inverse input/output relationship:

$$\tilde{l} = l_{\text{out}} \left(\frac{x}{255}\right)^{\gamma_o}$$

  where l_{out} is the maximum output and γ_o is a parameter of the output device.

Gamma correction.

- The input/output relationship for this imaging system is then

$$\tilde{l} = l_{\text{out}} \left(\frac{x}{255}\right)^{\gamma_o} = \boxed{??}$$

  So we have that $\dfrac{\tilde{l}}{l_{\text{out}}} = \boxed{??}$
- If $\gamma_i = \gamma_o$, then $\tilde{l} = \dfrac{l_{\text{out}}}{l_{\text{in}}} l$
- The signal x is said to be *gamma corrected* because it is predistorted to display properly on the CRT.

To read: [T1, §5-1.3], [T2, §3.2.Log Transformations–Power-Law (Gamma) Transformations].

5.1.3 Histogram-based contrast enhancement

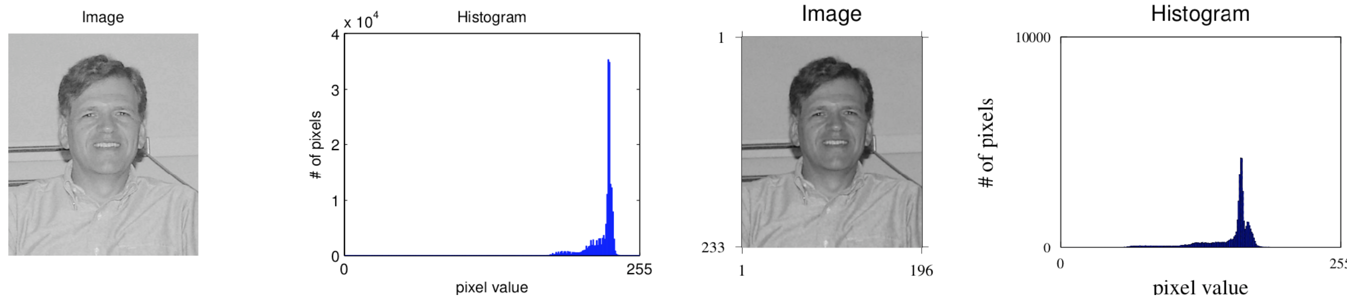
Mathematical definition.

Mathematically, given observations $X_1, \ldots, X_L$, a **histogram** is a function that counts the number of observations that fall into each of K disjoint sets $\mathcal{B}_k$ known as **bins**:

$$\underbrace{h_k = \sum_{l=1}^{L} \mathbb{I}_{\{X_l \in \mathcal{B}_k\}},}_{\#\text{ of } X_i \text{ values in the } k\text{th bin}} \qquad k = 1, \ldots, K, \quad \sum_{k=1}^{K} h_k = L.$$

In 1D, usually the bins are disjoint intervals, e.g., $\mathcal{B}_k = (t_{k-1}, t_k]$. This is what Matlab's `hist` command does, choosing $K = 10$ bins by default.

Example: For an 8-bit-scale image $f[m, n]$ of size $M \times N$, typically $L = MN$ (total number of pixels), $K = 256$, and $\mathcal{B}_k = \{k - 1\}$ for $k = 1, \ldots, 256$, so the image histogram represents how many pixels have values $0, 1, \ldots, 255$. The following image has fairly poor contrast and its histogram is a clump of high values. Modifying the histogram can improve image appearance.



Statistical definition.

In statistics, often observations $\{X_1, \ldots, X_L\}$ are treated as **independent and identically distributed (i.i.d.) random variables** having **probability density function (pdf)** $f_X(x)$. Normalizing the histogram by L provides an estimate of the following probability:

$$\hat{\mathsf{p}}_k = \frac{h_k}{L} \approx p_k \triangleq \mathbb{P}\{X_l \in \mathcal{B}_k\} = \int_{\mathcal{B}_k} f_X(x) \, \mathrm{d}x.$$

This is an unbiased estimate: $\mathbb{E}[\hat{\mathsf{p}}_k] = p_k$. If we have discrete-valued random variables taking values $x_1, \ldots, x_K$ and we let $\mathcal{B}_k = \{x_k\}$, the p_k is simply the **probability mass function (PMF)** of the random variable and $\hat{\mathsf{p}}_k$ is an estimate of that PMF.

- A histogram is the simplest form of **density estimation**, i.e., estimating $f_X(x)$ from data.
- In most real-world images, the pixel values are not independent random variables because neighboring pixel values tend to be correlated. (We will consider such correlations in image processing methods in EE616.)
- However, for the purposes of contrast enhancement, we can imagine a probability experiment in which we toss all the pixel values into a hat and then select values from the hat at random. The result of this experiment will be a discrete random variable, whose probability mass function is the relative frequency of each image value, i.e., $p_k = h_k/L$.

Example: The histogram showed above, if normalized by $L = MN$, is such a PMF.

Histogram equalization.

- If the histogram is "too clustered" then often the image will be less appealing than if we "spread out" the image values to "more evenly" cover the range $[0, 255]$.
- Supposedly it is often visually pleasing to have a displayed image's histogram peak around 128 and fall off more or less Gaussian-like towards 0 and 255.
- Histogram-based methods for contrast enhancement usually involve a nonlinear transformation to achieve this more desirable "distribution" of pixel values.
- Matlab's `histeq` command defaults to a uniform target distribution, but allows a user-specified histogram.
- Typically this transformation is "more nonlinear" than simple gamma correction, and is adapted to the specific distribution of the input image.

To perform **histogram equalization** we must design a transformation $g = \mathcal{T}(f)$ that will have the effect of producing an output image $g[m,n]$ having (approximately) some desired histogram such as a (discrete) uniform histogram from an input image $f[m,n]$.

The problem of designing the transformation $\mathcal{T}$ is closely related to the "transformation of random variables" problem. (Given a pseudo-random number generator in a computer, typically producing uniform $[0,1]$ variates, how do we generate random variables with some other distribution?)

Histogram equalization: Continuous random variable. For simplicity, we first consider continuous random variables.

Definitions:

- **cumulate distribution function (cdf)** $F_X(x) = \mathbb{P}\{X \le x\}$
- pdf $f_X(x) = \frac{\mathrm{d}}{\mathrm{d}x} F_X(x)$ (roughly a picture of "relative frequency" (density))

For a continuous random variable, a histogram ideally would be the probabilities that a random variable falls into each of a set of intervals $t_0, \ldots, t_K$:

$$p_k = \mathbb{P}\{t_{k-1} < X \le t_k\} = \int_{t_{k-1}}^{t_k} f_X(x)\,\mathrm{d}x = F_X(t_k) - F_X(t_{k-1}).$$

A bar plot of p_k as a function of either k or t_k is an ideal histogram.

Suppose we have random variable X with pdf f_X and we want a transformation $Z = \mathcal{T}(X)$ such that Z has a desired pdf is $f_d(z)$, with corresponding cdf $F_d(z)$. Here is how we design $\mathcal{T}$.

Proposition 5.1. *For a continuous random variable X, the following new random variable*

$$Y = F_X(X)$$

is distributed uniformly between 0 and 1. Furthermore the following new random variable will have pdf $f_d(z)$:

$$\boxed{Z = F_d^{-1}(Y) = F_d^{-1}(F_X(X)) \triangleq \mathcal{T}(X).} \tag{1}$$

Proof. It is easy to prove the first result. For any $y \in [0,1]$,

$$F_Y(y) = \mathbb{P}\{Y \le y\} = \mathbb{P}\{F_X(X) \le y\} = \mathbb{P}\{X \le F_X^{-1}(y)\} = F_X(F_X^{-1}(y)) = y.$$

The proof uses that X is a continuous r.v. The following equalities give the second result:

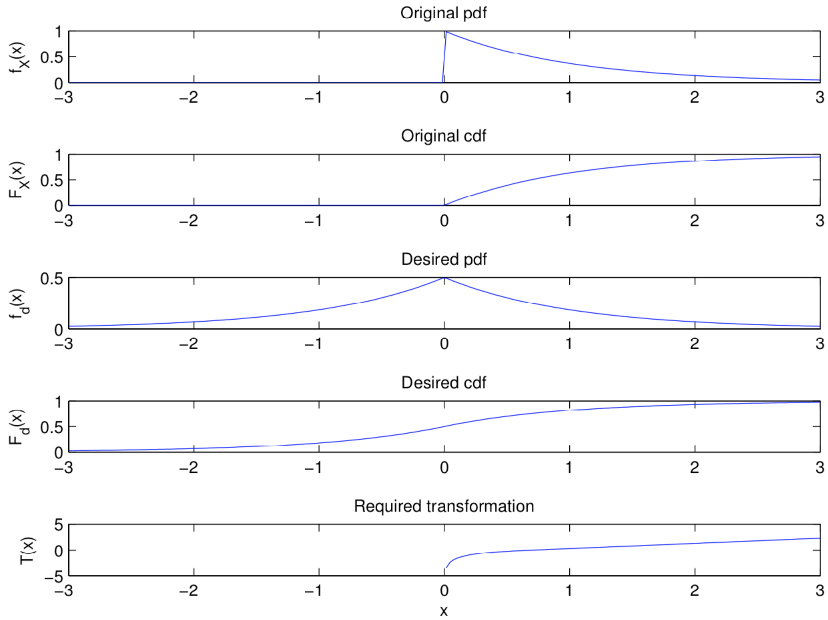
$$\mathbb{P}\{Z \le z\} = \mathbb{P}\{F_d^{-1}(F_X(X)) \le z\} = \mathbb{P}\{F_X(X) \le F_d(z)\} = \mathbb{P}\{X \le F_X^{-1}(F_d(z))\} = F_X(F_X^{-1}(F_d(z))) = F_d(z).$$

The proof assumes that there are no "flat regions" in F_d or F_X, i.e., Z and X are continuous random variables. (Note that cdfs of continuous random variables are strictly monotone over the range of the random variable.) $\square$

Ex. Suppose r.v. X has pdf $f_X(x) = e^{-x}\mathrm{step}(x)$ (exponential), but we desire $f_d(z) = \frac{1}{2}e^{-|z|}$ (Laplacian).

Because $F_d(z) = \begin{cases} \frac{1}{2}e^{z}, & z \le 0 \\ 1 - \frac{1}{2}e^{-z}, & z > 0 \end{cases}$, obtain $F_d^{-1} = \begin{cases} \log(2y) & 0 < y \le \frac{1}{2}, \\ \log \frac{1}{2(1-y)}, & \frac{1}{2} < y < 1. \end{cases}$ Thus, the required transformation is

$$\mathcal{T}(x) = F_d^{-1}(F_X(x)) = \boxed{??}$$



Note that the transformation function $\mathcal{T}$ is monotone increasing, i.e., $\mathcal{T}(t_2) \geq \mathcal{T}(t_1)$ for $t_2 > t_1$. This will always be the case for continuous random variables, because cdfs are strictly monotone over the range of the random variable.

Histogram equalization: Discrete random variable. For a discrete r.v., similar *general* concepts apply, but there are some differences in the details because the CDF of a discrete r.v. is not strictly monotone.

For simplicity, assume $k = 0, \ldots, K-1$. If $p(k)$ denotes the PMF of the input image $f[m,n]$, then its CDF is

$$F(x) = \sum_{k=0}^{K-1} p(k) \operatorname{step}(x-k),$$

where we carefully define the unit step function by $\operatorname{step}(t) = \begin{cases} 1, & t \geq 0 \\ 0, & \text{otherwise.} \end{cases}$

Suppose $p_d(k)$ denotes the desired PMF, with the corresponding CDF

$$F_d(x) = \sum_{k=0}^{K-1} p_d(k) \operatorname{step}(x-k).$$

Note that this CDF is not invertible. We rewrite the transformation relationship presented in (1) as follows:

$$\boxed{\mathcal{T}(x) = F_d^{-1}(F_X(x)) = \max\{y : F_d(y) \leq F_X(x)\},}$$

i.e., "for each x, choose the largest y such that $F_d(y) \leq F_X(x)$." This generalization makes $\mathcal{T}$ well-defined even for non-continuous r.v., where $F_d(\cdot)$ is not invertible. Similarly, in the discrete case, for each (input) value i, we should map it to an output value o for which

$$F_d(o) \approx F(i), \quad \text{i.e.,} \quad \sum_{k=0}^{o} p_d(k) \approx \sum_{k=0}^{i} p(k).$$

A reasonable criterion is

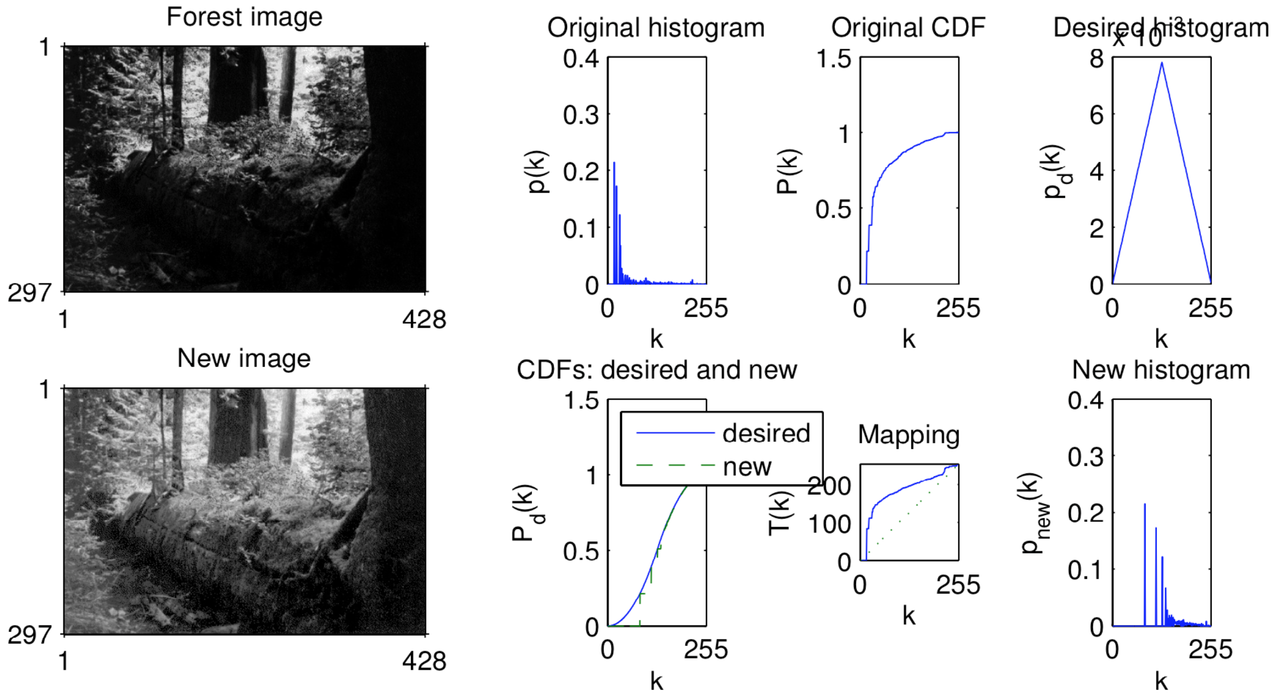
$$\mathcal{T}(i) = \max\left\{o : \sum_{k=0}^{o} p_d(k) \leq \sum_{k=0}^{i} p(k)\right\}.$$

This mapping is monotone, because as i increases the set expands. But $\mathcal{T}(i)$ is rarely strictly monotone.

For PMF/CDF arrays P and Pd of length K, this mapping is easily implemented in Matlab using the command

$$\mathsf{o} = \mathsf{sum}(\mathsf{Pd} \leq \mathsf{P}(i)) \tag{2}$$

Example: An image with poor contrast is "enhanced" by histogram equalization using (2).



To read: [T1, §5-3], [T2, §3.3.Histogram Equalization]

5.2 Image sharpening

5.2.1 Highpass filtering and unsharp masking

Because fine details like edges are often the principal source of high spatial frequency components, one can "sharpen" an image by "highpass filtering". For historical reasons related to photographic film development, this is also called unsharp masking. Originally this method was performed by analog methods using film and negatives! [wiki]

The natural analogy of the photographic processing approach is to scale up the original image $f[m,n]$ and then subtract a lowpass filtered version of image from itself:

$$g[m,n] = (1+\alpha) f[m,n] - \alpha h_l[m,n] \circledast f[m,n],$$

where $0 < \alpha$ and $h_l[m,n]$ is a lowpass filter.

- Equivalently,

$$g[m,n] = f[m,n] \circledast h[m,n], \quad h[m,n] = (1+\alpha)\delta[m,n] - \alpha h_l[m,n].$$

- A frequency response:

$$H(e^{\imath\mu}, e^{\imath\nu}) = 1 + \alpha - \alpha H_l(e^{\imath\mu}, e^{\imath\nu})$$

  Because $1 - H_l(e^{\imath\mu}, e^{\imath\nu})$ is a highpass filer, another interpretation that might be more intuitive is to see this as a "high boost" filter:

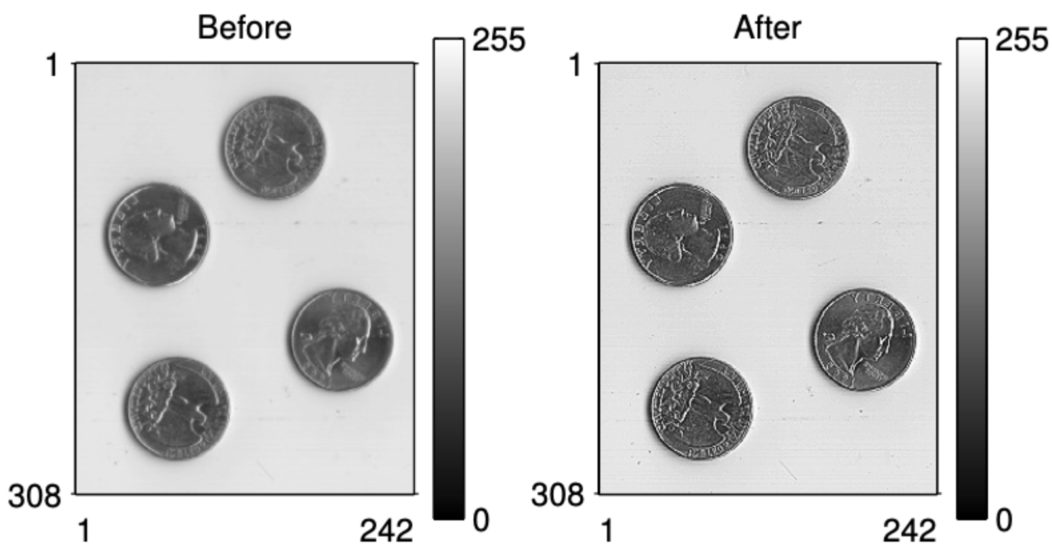
$$H(e^{\imath\mu}, e^{\imath\nu}) = 1 + \alpha(1 - H_l(e^{\imath\mu}, e^{\imath\nu}))$$

Example: The following is a typical sharpening filter. Note that it has DC response unity (so uniform regions are unchanged).

$$h[m,n] = \begin{bmatrix} 0 & -\alpha/8 & 0 \\ -\alpha/8 & \boxed{1+\alpha/2} & -\alpha/8 \\ 0 & -\alpha/8 & 0 \end{bmatrix} = (1+\alpha)\delta[m,n] - \alpha \begin{bmatrix} 0 & 1/8 & 0 \\ 1/8 & \boxed{1/2} & 1/8 \\ 0 & 1/8 & 0 \end{bmatrix}$$

What is this filter's frequency response at DC? $\boxed{??}$
See, also, Homework Problems 1–2 in Ch. 2.

Example: The following example used the above filter with $\alpha = 8$ for which $h[m,n] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \underline{5} & -1 \\ 0 & -1 & 0 \end{bmatrix}$ is particularly simple. The details in the letters on the coins become more visible after sharpening. It is not a more truthful representation of the reflectance image of these objects, but it may facilitate post-processing such as reading the letters for example.



Limitation: amplification of high spatial frequencies will tend to increase noise.

5.2.2 Homomorphic processing

In optical reflectance imaging, i.e., photography, the recorded signal is the product of the illumination times the object reflectance.

Sometimes the **illumination** of a scene causes large dynamic range changes in comparison to the variations of the object's **reflectance**, e.g., due to shadows of occluding structures. If the object properties are of greater interest than the illumination properties, then it can be desirable to reduce the contribution of the illumination variations in the image. Fortunately, this is often possible because often the illumination is spatially smoothly varying (low spatial frequencies).

Because the image is the product

$$f[m,n] = i[m,n]r[m,n],$$

it is natural to apply a logarithm to "separate" the two parts:

$$\log f[m,n] = \log i[m,n] + \log r[m,n].$$

Applying a lowpass filter to $\log f[m,n]$ should primarily pass the illumination component, where as a highpass filter should primarily pass the reflectance component

$$f[m,n] \to \boxed{\log} \to \boxed{\text{highpass from } \alpha \text{ at DC to } \beta \text{ at } \pi} \to \boxed{\exp} \to g[m,n].$$

By choosing $\alpha < \beta$, hopefully the low frequency illumination variations are reduced, highlighting the object. Note the use of a nonlinear transformation so that we could then apply LSI systems concepts!

Adaptive modification of local contrast and local luminance mean.
The preceding method can be viewed as a **two-channel processing** method. We attempt to separate the image into two components (reflectance and luminance previously), apply separate processing to each component (lowpass and highpass with different gains previously), and then add the two processed components back together again. Many algorithms have been based on variations of this theme.

5.3 Image denoising: Nonlinear filterting

Linear filter vs nonlinear filter.

- Linear filters
  - Tend to blur edges and other image detail.
  - Perform poorly to non-Gaussian noise.
  - Result from Gaussian image and noise assumptions
  - Images are not Gaussian
- Nonlinear filter
  - Can preserve edges
  - Very effective in removing impulsive noise
  - Result from non-Gaussian image and noise assumptions
  - Can be difficult to design

A significant portion of the image processing and computer vision literature is devoted to nonlinear methods that attempt to reduce noise while preserving important image features like edges without blurring them out. We will consider such methods soon.

Linearity and homogeneity.

Definition 5.2 (Linear system). *A system $y = F(x)$ is said to be* ***linear*** *if for all $\alpha, \beta \in \mathbb{R}$*

$$\alpha F(x_1) + \beta F(x_2) = F(\alpha x_1 + \beta x_2).$$

Any filter of the form $y_s = \sum_r h_{s,r} x_r$ is linear.

Definition 5.3 (Homogeneous system). *A system $y = F(x)$ is said to be* ***homogeneous*** *if for $\alpha \in \mathbb{R}$*

$$\alpha F(x) = F(\alpha x)$$

- This is much weaker than linearity.
- Homogeneity is a natural condition for scale invariant systems.

5.3.1 Median filtering

The medical filter is given by

$$y_s = \text{median}\{x_{s+r} : r \in W\}$$

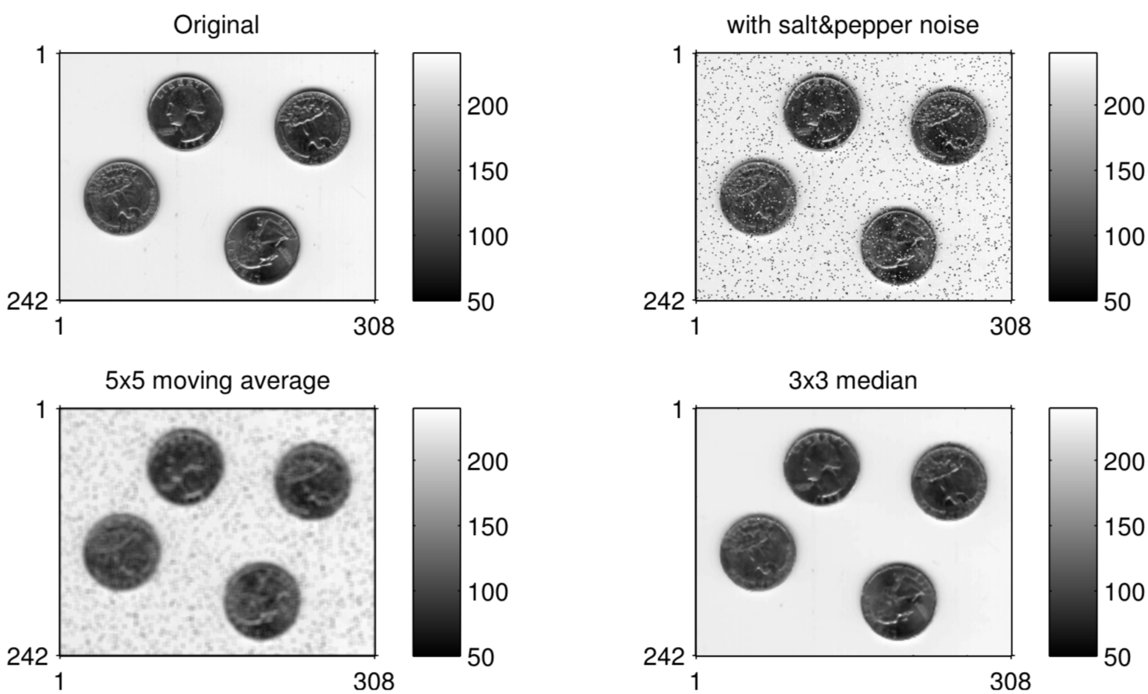
where W is a window with an odd number of points, e.g., three point window of $W = \{-1, 0, 1\}$.

- The median filter is linear $\boxed{??}$ homogeneous $\boxed{??}$
- What happens to impulsive noise (in an otherwise uniform region) when using a lowpass filter (like the moving average) and when using a median filter? Low pass filtering smears it out where as median filtering can remove the impulse completely.
- Using larger filter sizes means more reduction of impulsive noise values, but potentially more loss of fine signal values.

Example: Consider the 1D median filter with a 3-point window:

$x[m]$	0	0	1	1000	1	1	2	2
$y[m]$	?	0	1	$\boxed{??}$	1	1	2	?

Example: The following example uses Matlab's medfilt2 command to remove impulsive "salt and pepper" noise more effectively than a moving average filter.



Optimization viewpoint. Consider the median filter $y_s = \text{median}\{x_{s+r} : r \in W\}$, and the following functional:

$$F_s(\theta) \triangleq \sum_{r \in W} |\theta - x_{s+r}|$$

We now solve the following optimization problem

$$\underset{\theta}{\text{argmin}} F_s(\theta),$$

by differentiating $F_s(\theta)$:

$$\frac{\mathrm{d}F_s(\theta)}{\mathrm{d}\theta} = \frac{\mathrm{d}}{\mathrm{d}\theta}\sum_{r\in W}|\theta - x_{s+r}| = \sum_{r\in W}\mathrm{sign}(\theta - x_{s+r}) \triangleq f_s(\theta).$$

Note that this $\frac{\mathrm{d}F_s(\theta)}{\mathrm{d}\theta}$ expression only holds for $\theta \neq x_{s+r}$ for all $r \in W$ (remind that the absolute value function is not differentiable at 0). The solution falls at $\theta = x_{s^\star}$ such that

$$0 = \sum_{\substack{r\in W \\ r\neq(s^\star - s)}} \mathrm{sign}(\theta - x_{s+r})$$

In other words, $y_s = \underset{\theta}{\mathrm{argmin}} F_s(\theta)$.

Statistical analysis. Here is a statistical analysis of how much a median filter reduces noise compared to a moving average filter. Let $X_1, \ldots, X_N$ be i.i.d. random variables with pdf $f_X(x)$, mean μ, and variance σ^2. It is well known that the sample mean estimate of μ (which is analogous to a moving average filter) has a variance that diminishes with N:

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n, \qquad \mathrm{Var}\{\bar{X}\} = \frac{\sigma^2}{N}.$$

In fact, by the **central limit theorem**, $\bar{X}$ is asymptotically normal in distribution:

$$\sqrt{N}(\bar{X} - \mu) \to \mathcal{N}(0, \sigma^2).$$

Now, consider **the sample median** estimator $\hat{X} = \mathrm{median}(X_1, \ldots, X_N)$. Under the additional assumption that $f_X(x)$ is symmetric about μ, one can show that this estimator is asymptotically normal in distribution [2, p. 401]

$$\sqrt{N}(\hat{X} - \mu) \to \mathcal{N}\left(0, \frac{1}{4f_X^2(\mu)}\right).$$

In other words, for large N

$$\mathrm{Var}\{\hat{X}\} \approx \frac{1}{N4f_X^2(\mu)}.$$

Example: For $X_n \sim \mathsf{Uniform}(a, b)$, where $f_X(x) = \frac{1}{b-a}\mathbb{I}_{a<x<b}$, we have $\mathrm{Var}\{\bar{X}\} = \frac{(b-a)^2}{12N}$ and $\mathrm{Var}\{\hat{X}\} \approx \frac{(b-a)^2}{4N}$. The median has a larger variance than the mean for a uniform distribution, so it reduces noise less in the smooth regions of an image. But this penalty often may be acceptable when preserving edges is important.

5.3.2 Weighted median filtering

Optimization viewpoint. Consider the functional

$$F(\theta) = \sum_{r\in W} a_r|\theta - x_{s+r}|$$

We now obtain weighted median y_s by solving

$$y_s = \underset{\theta}{\mathrm{argmin}} F(\theta).$$

Differentiating, we have

$$\frac{\mathrm{d}F(\theta)}{\mathrm{d}\theta} = \frac{\mathrm{d}}{\mathrm{d}\theta}\sum_{r\in W} a_r|\theta - x_{s+r}| = \sum_{r\in W} a_r \mathrm{sign}(\theta - x_{s+r}) \triangleq f(\theta)$$

Note that this $\frac{\mathrm{d}F(\theta)}{\mathrm{d}\theta}$ expression only holds for $\theta \neq x_r$ for all $r \in W$.

$\boxed{\text{We now find } s^\star \text{ such that } f(\theta) \text{ is "nearly" zero.}}$

Computation of weighted median.

Step 1) Sort points in window W of size P.

- Let $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(P)}$ be the sorted values, called **order statistics**.
- Let $a_{(1)}, a_{(2)}, \ldots, a_{(P)}$ be the *corresponding* weights.

Step 2) Find $i^\star$ such that the following equations hold:

$$\begin{aligned} a_{i^\star} + \sum_{i=1}^{i^\star-1} a_{(i)} &\geq \sum_{i=i^\star+1}^{P} a_{(i)} \\ \sum_{i=1}^{i^\star-1} a_{(i)} &\leq \sum_{i=i^\star+1}^{P} a_{(i)} + a_{i^\star} \end{aligned}$$

Step 3) Then the value $x_{i^\star}$ is the weighted median value.

Homework (due by 11/4, 11:55 PM; Upload your solution to Laulima/Assignments)

Prob. 1. Assume continuous intensity values, and suppose that the intensity values of an image have the pdf

$$p_X(x) = \begin{cases} \frac{2x}{L-1}, & 0 \leq x \leq L-1, \\ 0, & \text{otherwise.} \end{cases}$$

a) Find the transformation function $\mathcal{T}$ that makes $y = \mathcal{T}(x)$ have a uniform pdf (i.e., histogram-equalized).
b) Find the transformation function $\mathcal{T}'$ that makes $z = \mathcal{T}'(y)$ (when applied to the histogram-equalized intensities, y) follow an image whose intensity pdf

$$p_Z(z) = \begin{cases} \frac{3z^2}{(L-1)^3}, & 0 \leq z \leq L-1, \\ 0, & \text{otherwise.} \end{cases}$$

c) Find the transformation function $\tilde{\mathcal{T}}$ that makes $z = \tilde{\mathcal{T}}(x)$ follow an image whose intensity pdf given in b).

Prob. 2. Consider the set of data $\{x_n : n = 1, \ldots, N\}$. We would like to estimate a "central value" using a method known as M-estimator. To do this, we compute the following function:

$$y = f(x) = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \rho(x_n - \theta),$$

where ρ is a function with the properties that $\rho(\Delta) \geq 0$ and $\rho(-\Delta) = \rho(\Delta)$
a) What is the value of y when $\rho(\Delta) = \Delta^2$?
b) What is the value of y when $\rho(\Delta) = |\Delta|$?
c) Derive an expression for computing y when $\rho(\Delta) = |\Delta|^{0.5}$. Express your estimator with a root-finding algorithm, e.g., $y = \text{root}_\theta\{\cdot\}$.
d) When $\rho(\Delta) = |\Delta|^{0.5}$, is the function $y = f(x)$ linear? Is it homogeneous? Justify your answers.

References

[1] J. S. Lim, *Two-dimensional Signal and Image Processing.* Upper Saddle River, NJ, USA: Prentice-Hall, 1990.

[2] P. Bickel and K. Doksum, *Mathematical Statistics.* Oakland, CA: Holden-Day inc., 1977.