

Histograms

- A histogram plots how many times(the frequency) of each intensity in an image occurs.
- Helps avoid taking overexposed pictures.
- Easier to detect types of processing previously applied to an image
- Widely used in image processing.
- a common practice is to normalize the histogram function by the number of pixels in an image.
 - This helps us to remove bias.
 - The sum of all components of a normal histogram is equal to 1.

Histograms and Contrast

- Lower number of pins far apart = high contrast.
- High Contrast = very clear details **BUT**
- High Contrast = loss of details sometimes.
- We can get higher contrast by grouping pixels into discrete levels.

Histograms and Dynamic Range

- high dynamic range = very bright and very dark parts in an image.
- Dynamic range may exceed number of available bits to represent pixels
 - Solution :
 - Capture multiple images of the same scene.
 - Combine them.

Color image Histogram

- Two types of histogram
- Intensity histogram : image converted to grayscale and the histogram of the grayscale is taken
- Individual color channel histogram : 3 histograms for red, green, blue.
- Provide useful information like dynamic range, brightness, contrast.
- No information about distribution of colors.
- Different images can have the same histogram.

Histogram Equalization/Normalization

- The aim is to create uniform(flat) distribution of histogram. ie, the probability is $(1/L-1)$ to get a pixel value.
- Apply a point operation that changes histogram into a uniform distribution.
- Spreading out the frequencies in an image is a simple way to improve a dark or washed image.
- Can be expressed as
 - r_{k} : input intensity.
 - s_{k} : processed intensity.
 - k : range of intensity.
 - L : maximum intensity.
- $s = T(r)$, where $T(r)$ is a transformation to equalize a histogram, should be :
 - i. monotonically increasing : ie, if $x > y$, then $T(x) \geq T(y)$, So we don't get negative images after the transformation is applied.

- ii. Bounded by $[0, L-1]$: So the output is in the same range as the input.
 note that, given that T is strictly monotonically increasing, ie if $x > y$, then $T(x) > T(y)$, there is a $T^{-1}(s)$ that returns r .
- $s = T(r) = (L-1) \int_0^r P(r') dr'$
- There should be a bunch of integrations and probability equations to prove this, look at the course book for the details.
 Long story short, its a probability thing, where we look at all values to determine the transformation of a certain value.
- Histogram Equalization distributes the gray level to reach the maximum gray level white because the cumulative distribution function equals 1 when $0 \leq r \leq L$.
- Some cases :
 - Low Contrast Image Histogram : Narrow and centered toward the middle of grayscale. we distribute the histogram to a wider range, improving the quality
 - We do this by adjusting the probability density function of the image so that the probability is spread evenly.

Histogram Matching

- In histogram matching, we can specify the shape of the target histogram we want the output image to have. This is especially useful for domains where histogram equalization will result in a distorted image.
- We use the same method used in deriving the transformation function of the histogram equalization to find the transformation function for the desired histogram, however:
 - We need $P(s)$ in mathematic form and the ability to express s in terms of r .
 - doesn't result in a uniform histogram.
 - Its a trial and error process.
 - its a mapping operation, but is **not 1-to-1**.
 - Typically, we chose the mapped value closest to the original to produce less deformation.
 - The functions have to be increasing, because we dont want to flip pixel values.
- The process is as follows **INSERT IMAGE HERE**
- Algorithm :
 - Apply histogram equalization to the input, we gets $T(r)$.
 - Apply histogram equalization of the target, we get $s = G(z)$.
 - Obtain the inversed transformation function
 - Obtain the output by applying the function to the image pixel values.
$$z = G^{-1}(s) = G^{-1}(T(r))$$

or we can use **Discrete Formulation** :

- $$s_k = T(r_k) = (L-1) \sum_{j=0}^k P(r_j)$$
- find s_k for r , and get an approximate of $T(r)$.
 - find s_k for z , and get an approximate of $G(z)$.
 - Approximate the inverse function.
 - Obtain the output by applying the function to the image pixel values.

Local Histogram Processing

We can apply histogram operations to certain sections of an image rather than a whole image. this is especially useful for filtering, blurring, coloring, etc certain parts of an image. **We specify a neighborhood ($m \times n$) around (x,y) , construct the histogram, and any transformation functions on it.**

TODO INSERT MISSING LECTURE

Smoothing Spatial Filter

Average Smoothing

We have a mask matrix size of $n \times n$ with values $(1/n)$ and a pixel neighborhood $n \times n$. The value of each pixel at the center of $n \times n$ is given by

$$f(p) = \sum_{n} \sum_{m} \text{filter}[n][m] * \text{neighborhood}[n][m]$$

Weighted Smoothing

Same as above, except the mask can have different values (x/y) so as long as they add up to 1.

This way we can give more weight to the pixels at the center or the side.

The Identity Mask

The identity mask is a mask that has a value of 1 at the center and 0 everywhere else. applying this mask results in no changes to the image.

Smoothing highlights **gross details**, that is the larger details in an image. This could be useful in segmentation.

Smoothing Spatial filters can be used to remove noise and make images clearer (for example, when there is salt-and-pepper noise).

Sharpening Filters

Sharpening is the opposite of smoothing. The purpose is to highlight transitions in intensity. It is useful in fields such as medical imaging, feature extraction, feature enhancement, and industrial inspection.

Image blurring is done by pixel averaging. Averaging is analogous to integration. Therefore, it is logical to conclude that sharpening is analogous to differentiation.

we are concerned about the behaviour of 1st and 2nd derivatives in the areas of constant intensity, onset and end of discontinuities, and intensity maps.

Properties of the First Derivative

1. Zero in areas of constant intensity.
2. Nonzero at the onset of a step and a ramp.
3. Nonzero along the intensity ramp.
4. $df/dx = f(x+1) - f(x)$

Properties of the Second Derivative

1. Zero in areas of constant intensity
2. Nonzero at the onset of a step and a ramp.
3. zero along the intensity ramp.
4. $d^2 f/dx^2 = df(x+1)/dx - df(x)/dx$

Laplacian Image Sharpening

We have the equation

$$\Delta^2 f(x,y) = f(x+1,y) - f(x-1,y) + f(x,y+1) - f(x,y-1) - 4f(x,y)$$

To find the 2nd derivative using the partial derivatives of X and Y.

Computing the Laplacian doesn't produce a sharpened image. It produces images that have the grayish edge lines and other discontinuities over a black background.

It is common practice to scale the laplacian image to [0-255]

If the definition (the neighborhood we are working in) has a negative center coefficient, we subtract the laplacian from the image. Otherwise, we add the laplacian to the image.

$$g(x,y) = f(x,y) + c[\Delta f(x,y)]$$

This can be applied in a single step by

$$g(x,y) = 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

Unsharp Masking & High-Boost Filtering

It is a process used in the printing industry to sharpen images

Steps :

1. Blur the original image
2. Subtract the blurred image from the original (mask)
3. Add the (weighted portion) mask to the original

ie, We extract the gross details then emphasize them to a certain degree. This is given by this pair of equations.

$$g_{\text{mask}}(x,y) = f(x,y) - f'(x,y)$$

$$g(x,y) = f(x,y) + k \cdot g_{\text{mask}}(x,y), k \leq 1$$

Choosing $k > 1$ results in **High-Boost Filtering**, which could result in losing detail depending on the domain and the image itself.

Sharpening Using the first derivative

Δf is a vector that points towards the greater rate of change. it is given by $|\partial_x f - \partial_y f|$

Gradients (Δ) are used widely in industrial inspection because it produces thicker edges in the result which makes it easier for machines to detect artifacts.

Non-Linear Spatial Filters (Order-static)

These are filters whose response is based on the original value of the pixel at the center of the neighborhood. An Example of this is the **median filter**. These are especially useful in removing salt-and-pepper noise. What is special about the median filter is that it doesn't add new intensity values to an image.

A similar filter is the **averaging filter**, which is less effective than the median filter in noise removal and may add new pixel values.