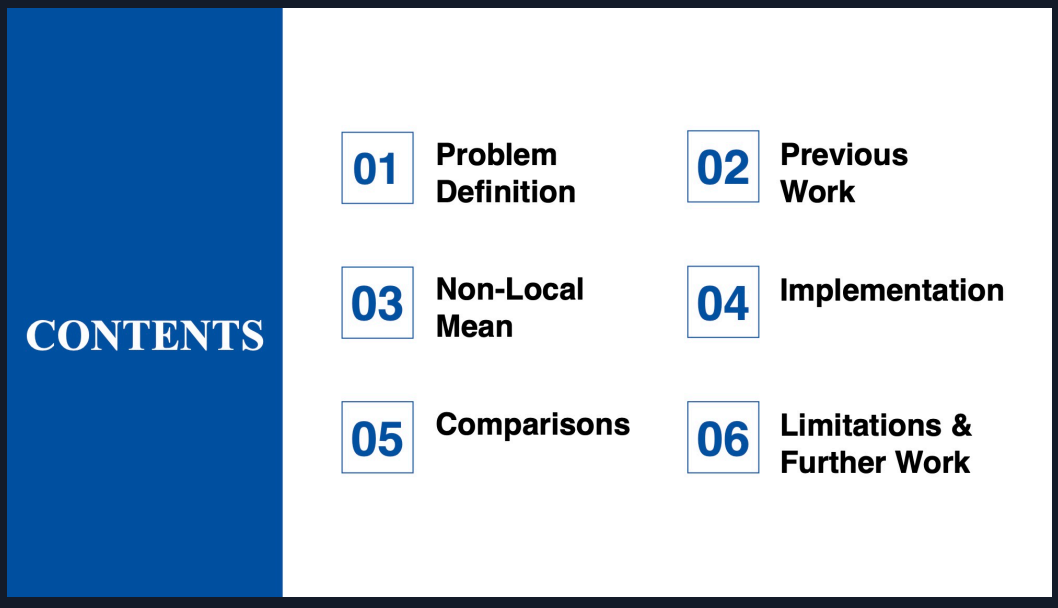
H1 Non-local Mean Algorithm

H2 Self Introduction

Hi I am Jinny, and this is jefferson, We are gonna introduce Non-local Mean filtering Algorithm for image denoising.

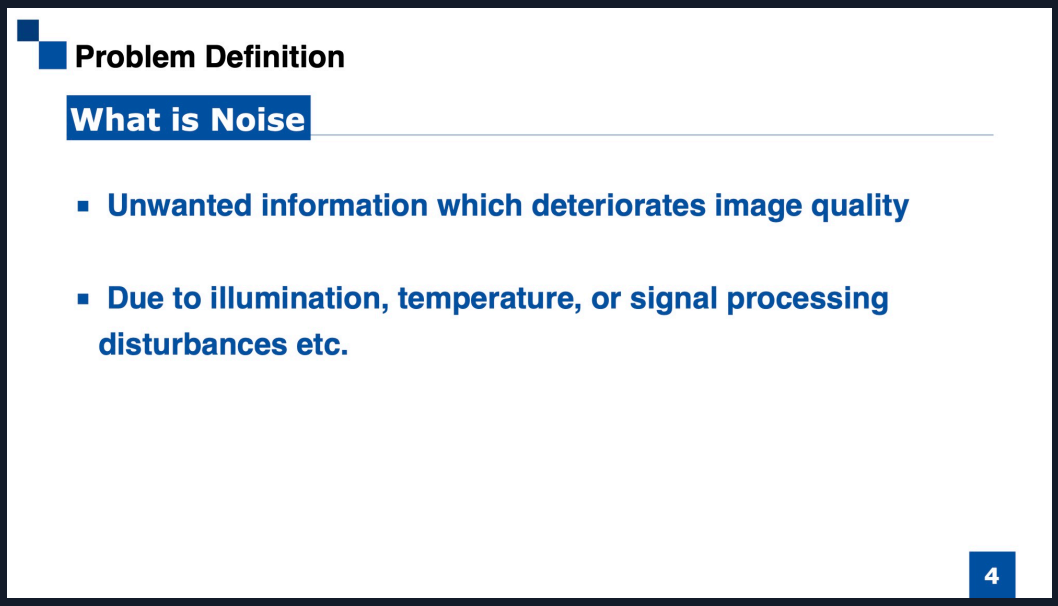


- So here's our agenda today:
- we will first give u a bref introduction about our problem space, the image noise actually -
    - and the background information about the image denoising method.

- Then we will talk about some typical smoothing filters algorithm with their performance.
- We will then focus on the Non-local Mean Algorithm, giving you definition, formula and consistency.
- We will also show your our own implementation of non-local algorithm with a demo -
  - and the result comparision with other classical local smoothing algorithm.
- Finally we will have a short discussion about the limitation of the Non-local Mean algorithm
  - and insignts of future works.

OK Now let's start from the problem - Noise.

H2 Problem - Noise

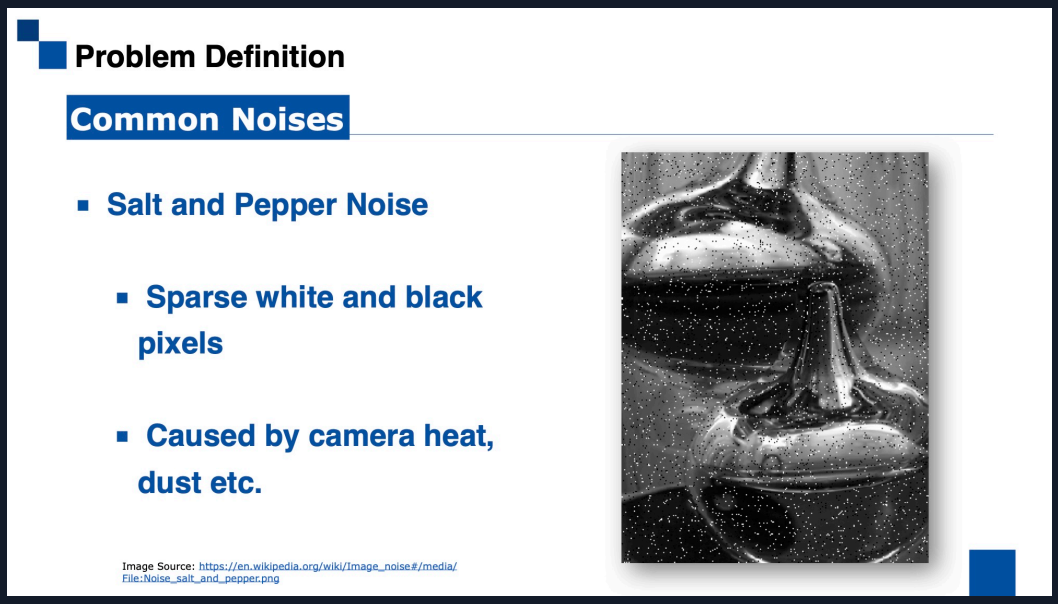


H3 What is Noise In Image?

- **Briefly Speaking, they are Unwanted information which deteriorates image quality**
- Images taken by digital cameras or conventional film cameras will pick up noise from a variety of sources like by a device's mechanism (illumination, temperature) or signal processing disturbances
- Noise can be random or with different frequency distribution
    - for example gaussian noise is with normal frquency distrbution

      short noise with Poisson frequency distribution

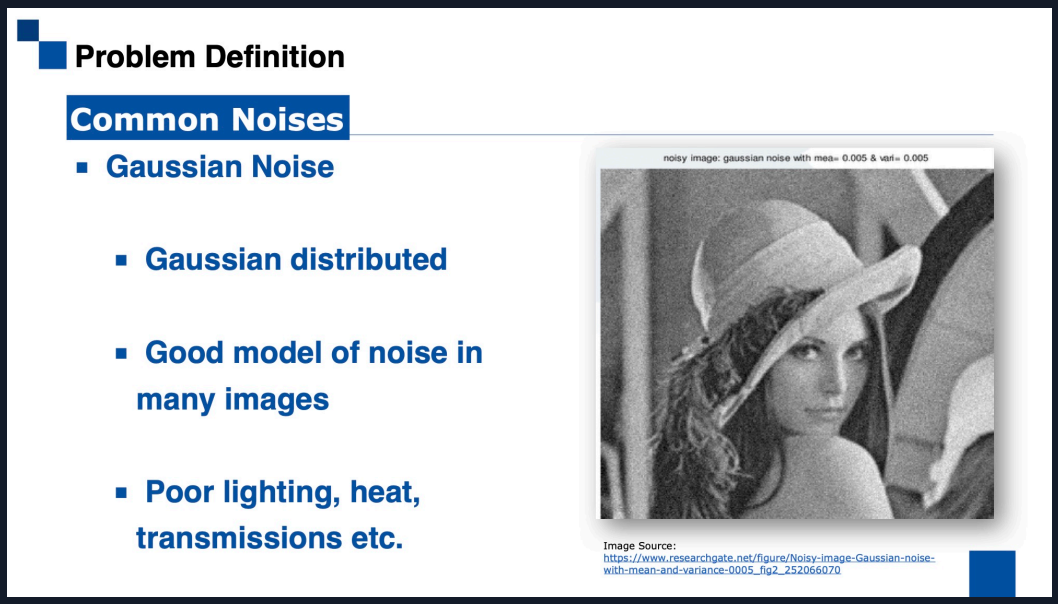
      Now let's see some common noise type



H3 Noise Type

H4 1. Salt and Pepper Noise

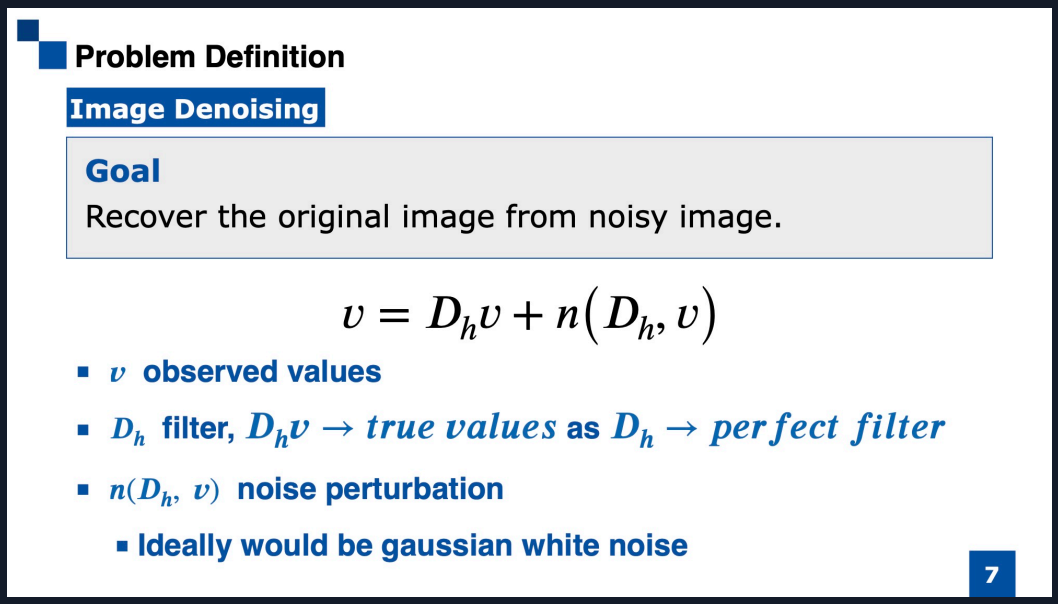
- to short with, they are sparse light and dark disturbances
    - As you can see in example picture, there are some pixels that are very different in color or intensity from their surrounding pixels
    - so when viewed, the image contains dark and white dots, hence in term salt and pepper
- And they are usually caused by
    - flecks of dust inside the camera
    - the camera is overheated



And Next common noise type is :

H4 2. Gaussian Noise

- so As the name implies, they are Noises with Gaussian frequency distribution
  - that is to say, if each pixel in the image will be changed from its original value by a small amount
  - then the values the noise can take on are Gaussian-distributed
- Notice that Gaussian noise - a good model for representing noise frequency in image
  - since by central limit theorem , the sum of different noises tends to approach a Gaussian distribution (
- And this kind of noise are usally caused
  1. during acquisition - that is, caused by sensor
     - if you take photo in a poor light, high temperature, etc
  2. or caused by electronic circuit transmission



So the goal of our alogrithm is to

H2 Goal - Noise reduction

H3 Image Denoising

- recover the original image from a noisy measurement

H4 One Attempt Operator - Smoothing Filters

- using a spatial filter to smooth an image

H3 Denoising Operator

- a denoising operator D_h can be defined as this model

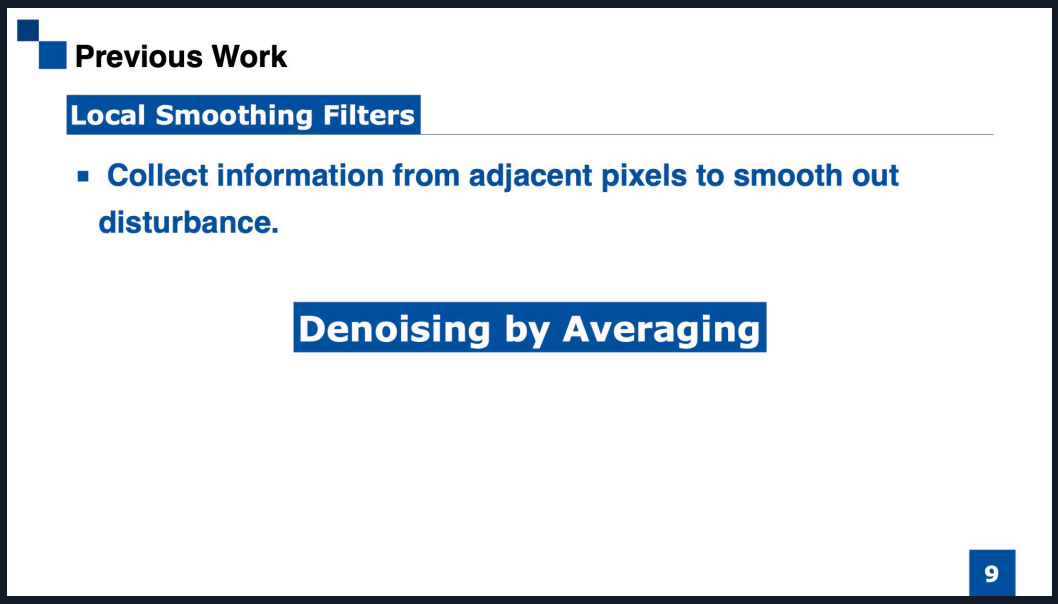
$$v = D_h v + n(D_h, v)$$

- where we denotes $v \to$ noisy image that is observed value, $h \to$ filtering parameter $\propto$ standard deviation of noise, $n() \to$ noise perturbation
- so a noisy image v after denoising by operator D will approach to the true value of the image
- And Ideally
  - the true value image should be smoother than the observed noisy image
  - and the noise perturbation should be a realization of a gaussian white noise
- which is a Special case of *Gaussian noise*
  - whose values at any pair of times are uncorrelated

Now we 've got a bref understanding about Smoothing Filters Algorithm

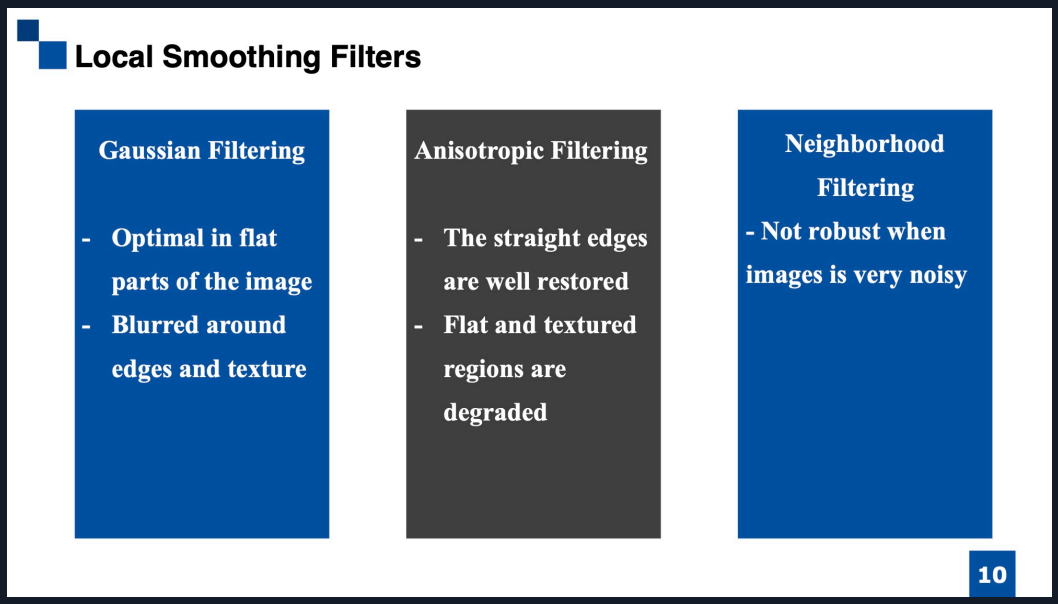
H2 Previous Work - Local Smoothing Filters

H4 so what is a Local Smoothing Filters Algorithm



- **Briefly speaking, we collect information from adjacent pixels to smooth out disturbance**
- **moreover, local filters algorithm denoise by Averaging**
  - that is, take the mean value of a group of pixels surrounding a target pixel to smooth the image

Let's takle about some example algorithm's performance



H3 Gaussian Filtering

- Performence
  - optimal in harmonic flat parts of the image
  - blurred around edges and texture

H3 Anisotropic Filtering

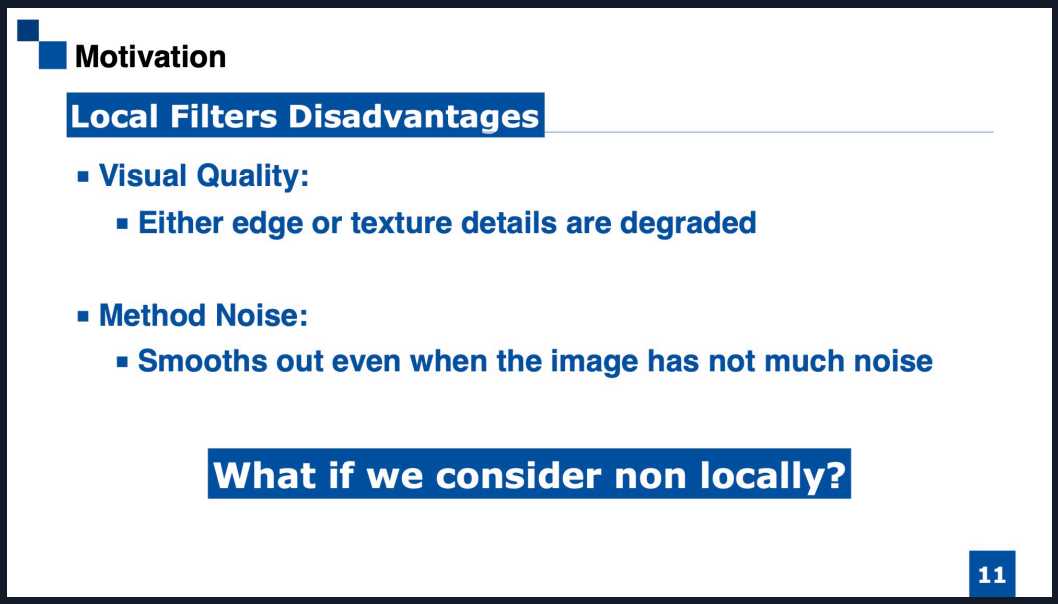
- Performence
  - the straight edges are well restored
  - flat and textured regions are degraded

H3 Neighborhood Filtering

- Performance
  1. the good thing is that the algorithm does not blur the edges
  2. comparing only grey level values in a single pixel is not so robust when these values are noisy

And we will show you the graphic detail later in comparison.

So as you can see, Though Local filters indeed solve the problem in some degree, there are some



H1 # common weekness of the local filters

to sum up with

- the edges and details of fine-scaled image will get blurred because they also correspond to the blocked high frequencies
- besides, image can be over smoothed when they are not very noisy

Then what is we consider ono locally?

Let's welcome jeff to talk more about this.