Abstract

Interest in digital image processing methods stems from two principal application areas: improvement pictorial information for human interpretation; and processing of image data for storage, transmission, and representation for autonomous machine perception.

Digital image processing refers to processing digital images by means of a digital computer. Note that digital image is composed of finite number of elements, each of which has a particular location and value.

There will be disturbances in every aspect digital image processing has no exception. Digital image processing also suffers from some sort of disturbances from external sources and others. This is going to be discussed in this aspect.

As we know that the images could be in the form of analog signals there is a need to convert these signals to digital form which can be done by plotting the image using different transfer functions which are explained here under. A transfer function maps the pixel values from the CCD (Charge coupled device) to the available brightness values in the imaging software all the images so far have been plotted using linear transfer functions.

Filter masks and other manipulations have also discussed in this aspect in order to make the image filter and get a clear cut form of the image.

Introduction

What is digital image processing?

A digital image is picture which is divided into a grid of “pixels” (picture elements). Each pixel is defined by three numbers(x, y, z), and displayed on a computer screen.
The first two numbers give the x and y coordinates of the pixel, and the third gives its intensity, relative to all the other pixels in the image. The intensity is a relative measure of the number of photons collected at that photosite on the CCD, relative to all the others, for that exposure.

The clarity of a digital image depends on the number of “bits” the computer uses to represent each pixel. The most common type of representation in popular usage today is the “8-bit image”, in which the computer uses 8 bits, or 1 byte, to represent each pixel.

This yields $2^8$ or 256 brightness levels available within a given image. These brightness levels can be used to create a black-and-white image with shades of gray between black (0) and white (255), or assigned to relative weights of red, green, and blue values to create a color image.

The range of intensity values in an image also depends on the way in which a particular CCD handles its ANALOG TO DIGITAL (A/D) conversion. 12-bit A/D conversion means that each image is capable of $2^{12}(4096)$ intensity values.

If the image processing program only handles $2^8(256)$ brightness levels, these must be divided among the total range of intensity values in a given image.

Below histogram shows the number of pixels in a 12-bit image that have the same intensity, from 0 to 4095.

Suppose u have software that only handles 8-bit information you assign black and white limits, so that all pixels with values to the left of the lower limit are set to 0, while all those to the right of the upper limit are set to 255. This allows u to look at details within a given intensity range.
So, a digital image is a 2-dimensional array of numbers, where each number represents the amount of light collected at one photosite, relative to all the other photosites on the CCD chip.

It might look something like....

- 98 107 145 126 67 93 154 223 155 180 232 250 242 207 201
- 72 159 159 131 76 99 245 211 165 219 222 181 161 144 131
- 156 110 114 91 70 128 321 296 208 193 191 145 422 135 138

By this we can guess the brightest pixel in the “image”...

Correcting the raw image

Every intensity value contains both signal and noise. Your job is to extract the signal and eliminate the noise!

But before that the sources of noise should be known. They can be -

a) The Dark Current:

Since electrons in motion through a metal or semiconductor create a current, these thermally agitated electrons are called the Dark Current.

...so, to eliminate thermal electrons, the CCD must be COOLED as much as possible. The cooler one can make one’s CCD, the less dark current one will generate. In fact, the dark current decreases roughly by a factor of 2 for every 7°C
drop in temperature of the chip.

At – 100°C the dark current is negligible.

When you process an image correctly, you must account for this dark current, and subtract it out from the image. This is done by taking a “closed shutter” image of a dark background, and then subtracting this dark image from the “raw” image you are observing.

The exposure time of the dark image should match that of the image of the object or starfield you are viewing.

In fact, the one who regularly take CCD image keep files of dark current exposures that match typical exposure times of images they are likely to take, such as: 10,20,45,60or 300 seconds, which are updated regularly, if not nightly.

b) The Bias Correction

CCD cameras typically add a bias value to each image they record. If you know that the same specific bias value has been added to each pixel, you can correct for this by subtracting a constant from your sky image.

c) Pixel – to – pixel Sensitivity variation

Another source of noise is the inherent variation in the response of each pixel to incident radiation. Ideally, if your CCD is functioning properly, there should be no variation in pixel value when you measure a uniformly-illuminated background. However, nothing is perfect, and there usually is some slight variation in the sensitivity of each photosite, even if the incident radiation is totally uniform.

This can be accounted for by taking a picture of a uniformly bright field, and dividing the ray image by this “flat” field – a process called flat fielding. The length of time to expose the flat image should be enough to saturate the pixels to the 50% or 75% level.

One must take 4 pictures before beginning to process the image. One needs 4 images to create a “noiseless” image of the sky.

1) The original;
2) A dark exposure of the same integration time as your original;
3) A flat exposure;
4) And another dark exposure, of the same integration time as your flat exposure!

\[
\text{Final image} = \frac{\text{raw image} - \text{dark}_{\text{raw}}}{(\text{Flat} - \text{dark}_{\text{flat}})}
\]

(Don’t forget to subtract your dias correction form each image)

So... the correct raw image taken with any CCD one must

- Subtract the bias correction form each image;
- Subtract the dark current image from the raw image;
- Subtract the dark current image form the flat field image;
- Divide the dark – subtracted raw image by the dark- subtracted flat image.

3 ways of displaying your image to Enhance Certain Features

Once you have dark – subtracted and flat – fielded your image, there are many techniques you can use to enhance your signal, once you have eliminated the noise. These manipulations fall into two basic categories:

1) CHANGING THE WAY IN WHICH THE INFORMATION IS PLOTTED ON YOUR SCREEN

These methods are basically mapping routines, and include:

- Limiting the visualization thresholds within the histogram
- Plotting the image using different transfer functions
- Histogram equalization

2) MATHEMATICAL METHODS OF MASSAGING YOUR DATA

These methods employ various matrix multiplications, Fourier transformations, and convolutions, and we will address them in the next section.

**Limiting the visualization thresholds within the histogram**

We already saw that the histogram function shows you the distribution of brightness values in an image, and the number of pixels within the same brightness value.

In the below histogram shown, most of the useful information is contained between the user-defined limits. The peak of intensities on the lower end could possibly be some faint feature, which could be enhanced in a variety of ways...
By changing the visualization limits in the histogram, the user can pre-
deﬁne the black and white levels of the image, thus increasing the level of detail
available in the mid-ranges of the intensities in a given image.

Some examples of histogram limitation is used to examine different features

![Histogram Limitation](image)

**Plotting the image using different transfer functions**

A transfer function maps the pixel values from the CCD to the available brightness
values in the imaging software all the images so far have been plotted using linear
transfer functions

**…but you can also use non-linear scaling**

Human eyes see a wide range of intensities because our vision LONGARITHMICALLY
scaled.

When you plot digital images logarithmically, it allows you to see a broader range of
intensities, and can give a more “natural” look... as if you could see the object with
your naked eyes.

**…or, you could use a Power Law scaling**

Fractional powers enhance low intensity features, while powers greater than 1 enhance
high intensity features.
HISTOGRAM EQUILIZATION

It means of flattening your histogram by putting equal numbers of pixels in each "bin", it serves to enhance mid-range features in an image with a wide range of intensity values. When you equalize your histogram, you distribute your 4096 intensity from your CCD equally among the intensity values available in your software. This can be particularly useful for bringing out features that close to the sky background, which would otherwise be lost.

After you have corrected your raw image so that you are confident that what you are seeing really come form incident photons and not electronic noise of your CCD. You may still have unwanted components in your image.

It is time now to perform mathematical operations on your signal which will enhance certain features, remove unwanted noise, smooth rough edges, or emphasize certain
boundaries.

...and this brings us to our last topic for this module:

**Filters masks & other Mathematical Manipulations**

**The rhyme and reason**

Basically, any signal contains information of varying frequencies and phases. In digital signal enhancement, we attempt to accentuate the components of that signal which carry the information we want, and reduce to insignificance those components which carry the noise.

Audio equipment, such as your stereo or CD player, has filters which do this for one – dimensional, time – varying audio signal.

In digital image analysis we extend these techniques to 2 – dimensional signals which are spatially varying.

*In any case, the basic idea is the same:*

*Get the most out of your data*

*For the least amount of hassle!*

**Here’s how it Works:**

you create an “n X n” matrix of numbers, such as 3 X 3 or 5 X 5, and you move this across your image, like a little moving window, starting at the upper left corner (that’s 0,0,recall).

You “matrix multiply” this with the pixel values in the image directly below it to get a new value for the center pixel.

You move the window across your image, one pixel at a time, and repeat the operation, until you have changed the appearance of the entire image.

Here’s an example of one kind of filter:

```
  1 2 1
  2 4 2
  1 2 1
```

if you move this matrix across an image and matrix multiply along, you will end up
replacing the center pixel in the window with the weighted average intensity of all the points located inside the window.

HERE’S HOW TO DO IT:

\[
\begin{align*}
I_{11} & \quad I_{12} \quad I_{13} \quad I_{14} \quad I_{15} \ldots \\
I_{21} & \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25} \ldots \\
I_{31} & \quad I_{32} \quad I_{33} \quad I_{34} \quad I_{35} \ldots \\
I_{41} & \quad I_{42} \quad I_{43} \quad I_{44} \quad I_{45} \ldots \\
I_{51} & \quad I_{52} \quad I_{53} \quad I_{54} \quad I_{55} \ldots
\end{align*}
\]

\[
I_{22}' = \frac{1I_{11} + 2I_{12} + 1I_{13} + 2I_{21} + 4I_{22} + 2I_{23} + 1I_{31} + 2I_{32} + 1I_{33}}{1+2+1+2+4+2+1+2+1 \text{ OR } 16}
\]

Imagine your image in 3 dimensions, where the intensity is plotted as height. Large scale features will appear as hills and valleys, while small bright objects like stars will appear as sharp spikes. Some features will have steep gradients, while others shallow. Some features may have jagged edges, while others are smooth.

You can design little n X n windows to investigate and enhance these kinds of features of your image in the slides that follow, we will show examples of low Pass, high pass, edge detection, gradient detection, sharpening, blurring, bias filtering your image. Low – pass filters enhance the larger scale feature in your image.
High pass filters enhance the short period features in your image, giving it a sharper look.

Some examples of high pass filters:

\[
\begin{pmatrix}
0 & -1 & 0 \\
-1 & 20 & -1 \\
0 & -1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & -1 & 0 \\
-1 & 10 & -1 \\
0 & -1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & -1 & 0 \\
-1 & 10 & -1 \\
0 & -1 & 0
\end{pmatrix}
\]

Edge detection filters are used to locate the boundaries between regions of different intensities.

The “bias filter” makes an image look like a bas relief with shadows. This can be useful for examining certain details

You can also combine process – such as low pass, high pass, and image subtraction in a process called UNSHARP MASKING.

Unsharp masking consists of a 3 – step process:

1. Make a copy of the image where each pixel is the average of the group of pixels surrounding it, so that the large features are not disturbed, but the small ones are blurred( this is the unsharp mask)

2. The pixel values of the original image are multiplied by a constant (“A”), and then the pixel values of the unsharp mask are subtracted from this one or more times (“B”). In this way, the large features are not changed by much, but the small ones are enhanced.

3. Finally a low – pass filter is applied to the result.

You can also create your own filter to smooth or otherwise operate on your image.

**Bibliography**

1. Digital image processing by Rafael C. Gonzalez and Richard E. Woods
2. [www.physics.ucsb.edu](http://www.physics.ucsb.edu)