Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

A Study on Sampling and Quantization Techniques of Image Processing

Ms.A.Pritha[1], D.Santhi Jeslet[2],

[1]Assistant Professor, [2]Head, [1,2]Department of Computer Science, M.G.R.College, Hosur, TN, India

Abstract: *An image is a 2D rectilinear array of pixels.An image is a two-dimensional function f(x,y), where x and y are the spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x ,y) is called the intensity of the image at that level. If x, y and the amplitude values of f are finite and discrete quantities, we call the image a digital image. A digital image is composed of a finite number of elements called pixels, each of which has a particular location and value. Digitization implies that a digital image is an approximation of a real scene .This paper describes about the methods of processing the image. Finally, Dithering and Halftonic techniques were discussed to solve the errors of gray scale image processing.*

Keywords: *Digitization, Pixels, Electromagnetic Energy Spectrum, Image Acquisition ,Illumination Source, Sensoring Image.*

I. INTRODUCTION

Often the domain and the range of an original signal $x(t)$ are modeled as contin-uous. That is, the time (or spatial) coordinate t is allowed to take on arbitrary real values (perhaps over some interval) and the value $x(t)$ of the signal itself is allowed to take on arbitrary real values (again perhaps within some interval). As mentioned previously in Chapter XX, such signals are called *analog* signals. A continuous model is convenient for some situations, but in other situations it is more convenient to work with *digital* signals — i.e., signals which have a discrete (often finite) domain and range. The process of digitizing the domain is called *sampling* and the process of digitizing the range is called *quantization*.

Most devices we encounter deal with both analog and digital signals. Digital signals are particularly robust to noise, and extremely efficient and versatile means for processing digital signals have been developed. On the other hand, in certain situations analog signals are sometimes more appropriate or even necessary. For example, most underlying physical processes are analog (or at least most conveniently modeled as analog), including the human sensor motor systems. Hence, analog signals are typically necessary to interface with sensors and actuators. Also, some types of data processing and transmission are most conveniently performed with analog signals. Thus, the conversion of analog sig-nals to digital signals (and vice versa) is an important part of many information processing systems.In this paper, we consider some of the fundamental issues and techniques in converting between analog and digital signals. For sampling, three fun-damental issues are (i) How are the discrete-time samples obtained from the continuous-time signal?; (ii) How can we reconstruct a a continuous-time signal from a discrete set of samples?; and (iii) Under what conditions can we recover the continuous-time signal exactly? For quantization, the three main issues we consider are (i) How many quantization levels should we choose?; (ii) How should the value of the levels be chosen?; and (iii) How should we map the values of the original signal to one of the quantization levels?

II. METHODOLOGY

A. Sampling

1. Sampling a Signal :

Figure 1 shows an analog signal together with some samples of the signal. The samples shown are equally spaced and simply pick off the value of the underlying analog signal at the appropriate times. If we let T denote the time interval between samples, then the times at which we obtain samples are given by nT where $n = \ldots, -2, -1, 0, 1, 2, \ldots$. Thus, the discrete-time (sampled) signal $x[n]$ is related to the continuous-time signal by

$$x[n] = x(nT).$$

It is often convenient to talk about the sampling frequency f_s. If one sample is taken every T seconds, then the sampling frequency is $f_s = 1/T$ Hz. The sampling frequency could also be stated in terms of radians, denoted by ω_s. Clearly, $\omega_s = 2\pi f_s = 2\pi/T$.

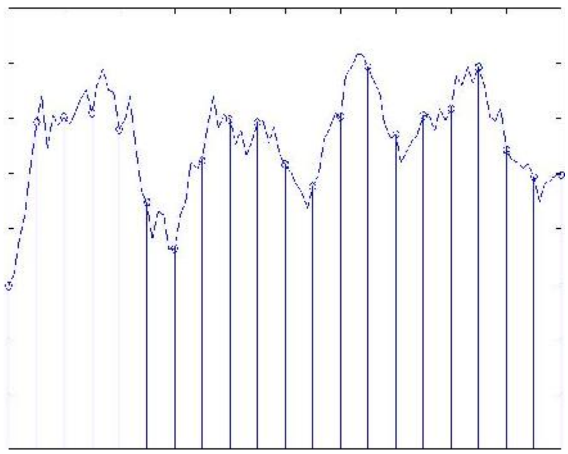


Figure 1: Sampling A Analog Signal.

The type of sampling mentioned above is sometimes referred to as “ideal” sampling. In practice, there are usually two non-ideal effects. One effect is that the sensor (or digitizer) obtaining the samples can’t pick off a value at a single time. Instead, some averaging or integration over a small interval occurs, so that the sample actually represents the average value of the analog signal in some interval. This is often modeled as a convolution – namely, we get samples of $y(t) = x(t)\quad h(t)$, so that the sampled signal is $y[n] = y(nT)$. In this case, $h(t)$ represents the impulse response of the sensor or digitizer. Actually, sometimes this averaging can be desirable. For example, if the original signal $x(t)$ is changing particularly rapidly compared to the sampling frequency or is particularly noisy, then obtaining samples of some averaged signal can actually provide a more useful signal with less variability. The second non-ideal effect is noise. Whether averaged or not, the actual sample value obtained will rarely be the exact value of the underlying analog signal at some time. Noise in the samples is often modeled as adding (usually small) random values to the samples.

Although in real applications there are usually non-ideal effects such as those mentioned above, it is important to consider what can be done in the ideal case for several reasons. The non-ideal effects are often sufficiently small that in many practical situations they can be ignored. Even if they cannot be ignored, the techniques for the ideal case provide insight into how one might deal with the non-ideal
effects. For simplicity, we will usually assume that we get ideal, noise-free samples.

2. Reconstructing a Signal

We now consider the reverse problem, namely how to construct a continuous-time signal given discrete-time samples. Suppose we are given a set of samples $x[n]$ that we know came from some continuous-time signal $x(t)$. We also assume we know the sampling rate T , so that we know $x(nT) = x[n]$. How can we recover $x(t)$ for other values of t? Unless we have some additional knowledge or make some assumptions, this problem clearly has many solutions. Figure 2 shows some discrete-time sam-ples and some possible continuous-time functions from which these samples could have been obtained. We have no way to know for sure what the val- ues of the original signal are at times other than nT . However, there are some simple estimates we might make to approximately reconstruct $x(t)$.

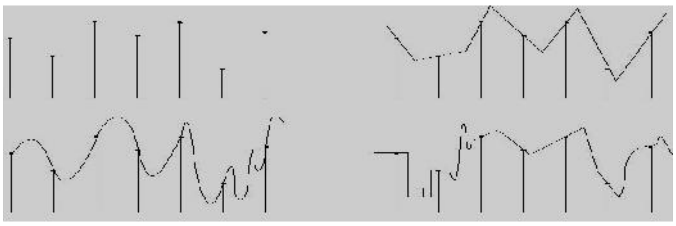


Figure 2: Possible Continuous-Time Functions Corresponding To Samples

Figure 3: Zero-Order Hold Reconstruction

The first estimate one might think of is to just assume that the value at time t is the same as the value of the sample at some time nT that is closest to t. This *nearest-neighbor* interpolation results in a piecewise-constant (staircase-like) reconstruction as shown in Figure 4.

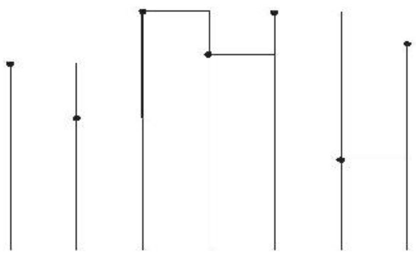


Figure 4: Nearest - Neighbour Reconstruction

Actually, instead of nearest-neighbor interpolation, most devices implement a similar type of interpolation known as zero-order-hold interpolation shown in Figure 3. This is one of the most widely used methods and is easy to implement. As with nearest-neighbor interpolation, this results in a piecewise-constant reconstruction, but the discontinuities are at the sample points instead of between sample points. In other words, the reconstruction is obtained by passing the discrete-time samples, as a set of impulses, into an appropriate system. In this case, the system has a very simple impulse response and can be easily implemented. Likewise, nearest method is linear interpolation shown in Fig 5.With this method the ,the reconstruction is a continuous function that just connects sample values with straight lines.

Figure 5: First Order Hold Reconstruction

3. The Sampling Theorem

Sampling Theorem, also called the Shannon Sampling Theorem or the Shannon-Whitaker-Kotelnikov Sampling Theorem, after the researchers who discovered the result. This result gives conditions under which a signal can be exactly reconstructed from its samples. The basic idea is that a signal that changes rapidly will need to be sampled much faster than a signal that changes slowly, but the sampling theorem for-malizes this in a clean and elegant way. It is a beautiful example of the power of frequency domain ideas.

Sampling where $\omega_s >> \omega$

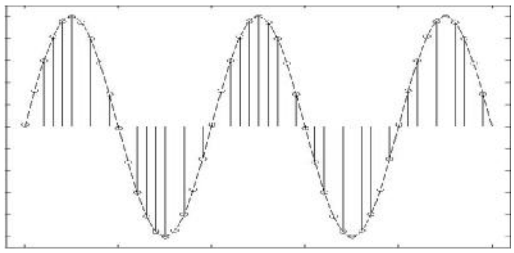


Figure 6: Sampling Sinunoid At High Rate

To begin, consider a sinusoid $x(t) = \sin(\omega t)$ where the frequency ω is not too large. As before, we assume ideal sampling with one sample every T seconds. In this case, the sampling frequency in Hertz is given by $f_s = 1/T$ and in radians by $\omega_s = 2\pi/T$. If we sample at rate fast compared to ω, that is if $\omega_s >> \omega$ then we are in a situation such as that depicted in Figure 6. In this case, knowing that we have a sinusoid with ω not too large, it seems clear that we can exactly reconstruct $x(t)$. Roughly, the only unknowns are the frequency, amplitude, and phase, and we have many samples in one period to find these parameters exactly.

Sampling where $\omega_s << \omega$

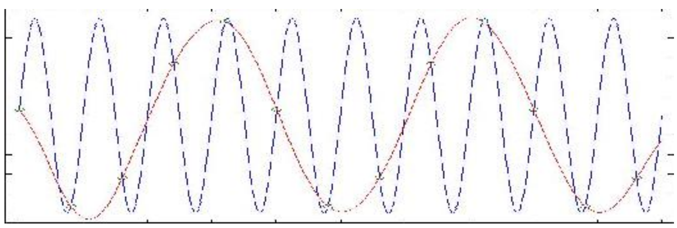


Figure 7: Sampling Sinunoid At Too Slow Of A Rate

In the Figure 7 the number of samples is small with respect to the frequency of sinusoid. This phenomenon is aliasing, if the sampling rate is too low.**Figure 8 Sampling too slowly causes aliasing.**

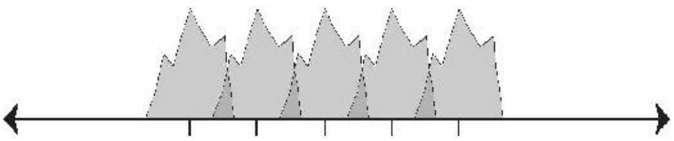
In Figure 8, high frequency components alias as lower frequencies and corrupt the unshifted copy of X (ω). In this figure, the copies are shown individually, with the overlap region simply shaded more darkly. What actually happens is that these copies add together so that we are unable to know what each individual copy looks like. Thus we are unable to recover the original X (ω), and hence cannot reconstruct the original $x(t)$.

III. QUANTIZATION

A. Uniform Quantization:

Quantization makes the range of a signal discrete, so that the quantized signal takes on only a discrete, usually finite, set of values. Unlike sampling (where we saw that under suitable conditions exact reconstruction is possible), quantization is generally irreversible and results in loss of information. It therefore introduces distortion into the quantized signal that cannot be eliminated. One of the basic choices in quantization is the number of discrete quantization levels to use. The fundamental trade in this choice is the resulting signal quality versus the amount of data needed to represent each sample. With L levels, we need $N = \log_2 L$ bits to represent the different levels, or conversely, with N bits

We can represent $L = 2^N$ levels.



The simplest type of quantizers are called *zero memory* quantizers in which quantizing a sample is independent of other samples. The signal amplitude is simply represented using some finite number of bits independent of the sam-ple time (or location for images) and independent of the values of neighboring samples. Zero memory quantizers can be represented by a mapping from the amplitude variable x to a discrete set of quantization levels $\{r_1, r_2..., r_L\}$.

The mapping is usually based on simple thresholding or comparison with certain values, t_k. The t_k are called the transition levels (or decision levels), and the r_k are called the reconstruction levels. If the signal amplitude x is between t_k and t_{k+1} then x gets mapped to r_k.

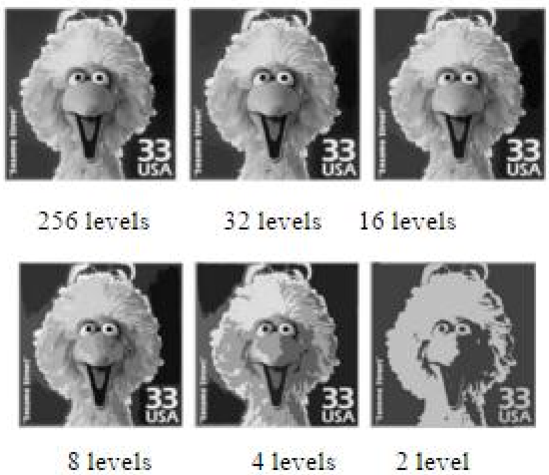


Figure 9: Quantizing A Gray-Level Image

Figure 9 shows an example of the effects of reducing the number of bits to represent the gray levels in an image using uniform quantization. As fewer quantization levels are used, there is a progressive loss of spatial detail. Further-more, certain artifacts such as *contouring* (or *false contouring*) begin to appear. These refer to artificial boundaries which become visible due to the large and abrupt intensity changes between consecutive gray levels. Using uniform quantization, images we usually start seeing false contours with 6 or fewer bits per pixel.ie., about 64 or fewer gray levels. In audio signals, we can often hear distortion noise about 8 bits per sample or 256 amplitude levels. These figures depend on particular image or signal.

B. Non Uniform Quantization

It is natural to expect the amount of distortion introduced depend on the quantizer and we can try to minimize the distortion by choosing a "good" quantizer

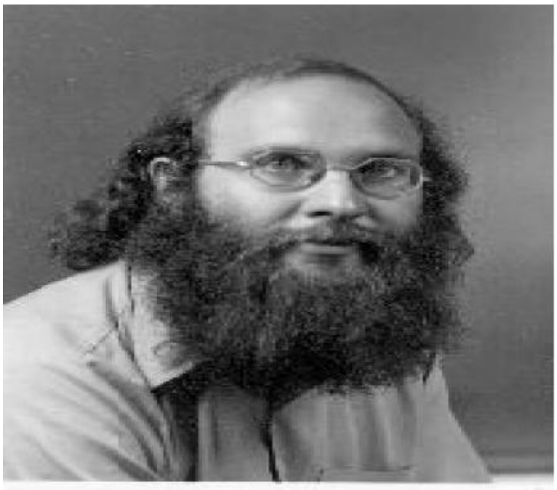
In Fig 10 the original image of 256 8 level 4 levels 2 level gray levels is shown and near by the Histogram shows the distribution of intensity Levels. The both Uniform quantized and Non-uniform quantized image with equal

Figure 10: Original image,256 gray levels
4 levels is represented clearly.

IV. DITHERING AND HALFTONING

Dithering, coarse quantization often results in the appear-ance of abrupt discontinuities in the signal. In images, these appear as false contours. One technique used to try to alleviate this problem is called *dither-ing* or *psuedorandom noise quantization*. The idea is to add a small amount of random noise to the signal *before* quantizing. Sometimes same noise values are stored and then subtracted for display or other purposes, although this is not necessary. The idea of adding noise intentionally might seem odd. But in the case of dithering, the noise actually serves a useful purpose. Adding noise before quantization has the effect of breaking up false contours by randomly assigning pixels to higher or lower quantization levels. This works because our eyes have limited spatial resolution. By having some pixels in a small neighborhood take on one quantized level and some other pixels take on a different quantized level, the transition between the two levels happens more gradually, as a result of averaging due to limited spatial resolution.

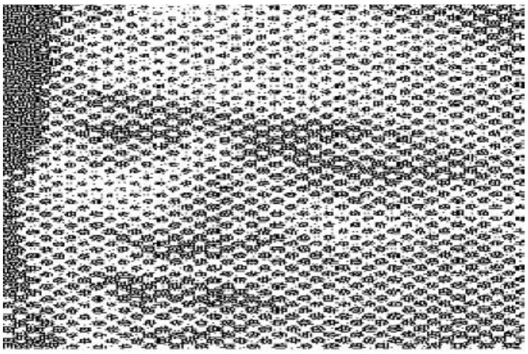
ORIGINAL IMAGE



DITHERING IMAGE

Halftoning, which refers to techniques used to give a gray scale rendition in images using only black and white pixels. The basic idea is that for an oversampled image, a proper combi-nation of black and white pixels in a neighbourhood can give the perception of gray scales due to spatial averaging of the visual system. For example, consider a 4 × 4 block of pixels. Certain average gray levels for the entire 4 × 4 block can be achieved even if the 16 pixels comprising the block are allowed to be only black or white. For example, to get an intermediate gray level we need only make sure that 8 pixels are black and 8 are white. Of course, there are a number of specific patterns of 8 black and 8 white pixels. A Common technique to avoid the appearance Of patterns is to select the particular arrangement of black and white pixels for a given gray scale rendition. It is called a "noise Matrix" or "halftone matrix" or "dither matrix".

CLASSICAL HALFTONE IMAGE



CONCLUSION

It would seem logical to emphasize the importance of Sampling and Quantization in image processing. This was a big breakthrough in science of processing a gray scale image, that moved to different levels of pixels per bits further it was possible to imagine. Now two methods of processing an image were able to exchange the reconstruction of an signal and to the nearest neighbouring image identification.In this paper we implemented Uniform and Non-uniform Quantization methods of image processing too. We have shown how Dithering and Halftoning techniques used to distribute errors among pixels and reduce the effects of quantization. Finally, we showed the original images and plotted graphs of image and signal processing ,reconstruction of images and Quantized images undsergone through results and also concluded with Dithering and Classical Halftoned images for the clear understanding of processing an image.

References

[1] **"**Handbook of Document Image Processing and Recognition"-Editors: Doermann, David, Tombre, Karl (Eds.)2014.

[2] Digital Image Processing, 3rd Edition by Rafael C. Gonzalez, Richard E. Woods

[3] S. Bayram, I. Avcubas, B. Sankur, and N. Memon, "Image manipulation detection," *J. Electron. Imag.*, vol. 15, no. 4, pp. 04110201–04110217, 2006.

[4] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1,pp. 101–117, Mar. 2008.

[5] H. Cao and A. C. Kot, "Manipulation detection on image patches using FusionBoost," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 992–1002, Jun. 2012.

[6] J. Fan, H. Cao, and A. C. Kot, "Estimating EXIF parameters based on noise features for image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 4, pp. 608–618, Apr. 2013.

[7] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[8]Annadurai s, Shanmugalakshmi R, "Fundamentals of digital image processing", Pearson Education 2007.

[9] N. Senthilkumaran, R. Rajesh, "Edge detection techniques for image segmentation – a survey of soft computing approaches", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.

[10] N.Senthilkumaran, R. Rajesh, "Brain image segmentation", international journal of wisdom based computing, vol. 1(3), December 2011.

[11] "Digital Image Processing " 1st Edition by Kenneth R. Castleman (Author) ,Sep2,2009