1

Fundamentals of Digital Image Display

When we venture into knowledge and science, we do so only to return better equipped for living.
Goethe

Whether the Universe's fundamentals are continuous or discrete is a question for philosophers, and up to now there has been little sign of full agreement on the matter. Even scaling down the question to address only the field of technology, the analog and digital approaches are very often two sides of the same coin, and looking at one or the other is a matter of choice and efficiency. A human being can very well be seen as a *source*, a *processor* and a *receiver* of information (in the sense of reducing the uncertainty). The image is often the way that information is carried from the source to the destination, and because of this a very narrow but well-defined field of today's technology deals with creating and displaying it.

Compared with other technological fields, image display is a relatively young one, its history being about 50 years, among which the largest part can be categorized as being based on analog methods. In very recent years, digital techniques of image display have been the tools that have allowed various electronic devices to interconnect and the digital era we are in has impacted this field in a way that was hard to predict even a few years ago. The image display process is considered to be digital if at least one essential stage from the source to the final image is performed in the digital domain, but that does not mean that all of the stages are digital, and in fact some of them are and will remain in the analog domain. Based on these considerations, the field of *digital image display* can be defined as a set of discrete algorithms applied to video data for the purpose of producing as an end result a physical image that has a good quality psychologically, in the sense of pleasing the human eye.

Digital Image Display: Algorithms and Implementation G. Berbecel
 ISBN: 0-470-84921-5

1.1 The Human Visual System: A Communication Channel

Looking at the human body as an entity which communicates with the outside world through a number of communication channels, an engineering approach allows us to determine that the one with the highest ability is the visual channel, its ability being much higher than the sum of all the other four channels together.

From the earliest stages of civilization there has been an attempt to communicate visually, and the entire history of mankind can be traced following the evolution of the ways in which people have tried to communicate ideas between themselves and from one generation to another.

The first method we know about was cave painting, then, as many alphabets were developed, paper became the main basis for capturing information. The ability to take pictures using a camera was a big step forward, but the really revolutionary development was *film*.

Film creation and reproduction were the first big challenges to understanding how the human eye works, its strengths and weaknesses, and how to exploit them to create the impression of moving pictures. Fortunately the eye was proving to be accustomed to static images, or images that change slowly, the speed of change being related to what people can achieve moving by their own force. There was no need for the human eye to distinguish between rapidly changing images because there were no such fast changing images for a long period of time. This specific weakness was exploited by the technology creating the impression of motion by presenting the eye with a sequence of static pictures that change faster than a threshold value.

On the evolutionary scale, *television* was another challenge for the eye in dealing with remotely transmitted moving video sequences, and the computer-generated image was the final test of the eye's ability to deal with something which had never existed before, artificially created images.

From the very beginning, the eye has had to deal with images representing objects at various distances, and the brain has had the task of making the necessary corrections such that the perception is always the same. When a flying bird passes by and moves away, at no point in time does one get the impression of the bird becoming smaller in reality, and that is because the brain adjusts the perceived size according to the distance, based on some *a priori* knowledge. This is called *image rescaling*. It represents a processing function consisting of changing the perceived size of a physical image, and it is a function that the eye does perfectly. In the era of artificially created images produced by electronic devices, there is a new need to move the image rescaling function outside of the human brain, to do it electronically. This need clearly appears when an image is generated for a specific type of display, and is submitted to a different one.

Although mathematically the required function is simple (just a dilation which corresponds to an image resizing), in practice it is extremely difficult because the electronic device has to produce a result that must be perceptually the same as the one created by the brain in the natural process of image resizing. Competing with a human function which has been specially created and polished over a long time has always been quite a challenge.

1.2 A Simplistic Model of the Eye

Because of the complexity of the eye and because the available data is just partial, a complete model of the eye is not available. In a simple interpretation a bi-dimensional

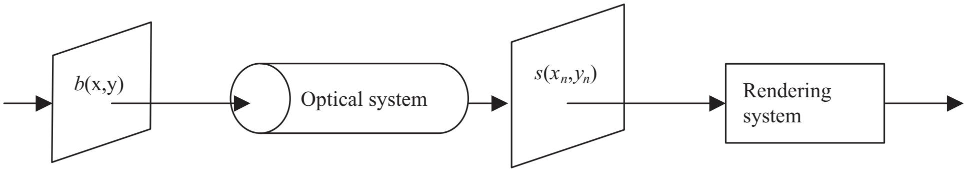

Figure 1.1 A simplified model showing the eye as a communication system

distribution of electromagnetic energy in the range 350–700 nm, which can be described using a continuous function $b(x, y)$, propagates through an optical system and hits an area of photoreceptors that translate the incident energy into electrical signals. These signals are then sent to the brain where an 'image' is created from the discrete data provided by the photoreceptors. The word image was written with quotation marks to suggest that it is in fact a perceptual entity, a projection into our system of knowledge. Figure 1.1 shows the suggested simplified model. The role of the optical system is to control the amount of blurring, which is believed to depend on the geometric coordinates x and y. The level $s(x_n, y_n)$ is the place where two important things happen: first, a conversion from continuous to discrete representation is performed and, second, a conversion from an incoming electromagnetic wave to an internal electrical signal is achieved. In the last stage, the rendering system translates the discrete set of electrical values into a perceived image.

It is worth noting that although the system model is simple and does not contain high-performance elements, the overall performance is excellent because of the corrections performed at the rendering level. The trade-off between the physical components and the processing done in the brain makes it possible to achieve high-quality results.

The usual way to characterize such a system is to study the impulse response that is known as a *point spread function* (PSF). The combinations of all sources of dispersion produce a PSF that is close to a Gaussian with a variance of about 1.5 times the distance between two receptors. It is this PSF that allows us to explain the relatively poor resolution of the eye (an angle of approximately one minute).

When two light pulses are close enough in space such that at the retinal level (the sampling level where the signal is represented as discrete) the centers of the two Gaussian curves are close enough (closer than twice the variance), the processing done by the brain is not able to distinguish them, and they appear as a single spot (Figure 1.2). It is clear that there is a direct relationship between the eye's resolution and the distance between the

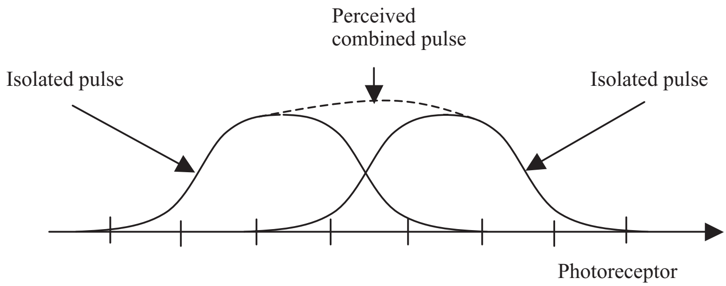

Figure 1.2 An illustration of how two pulses that are too close spatially will be perceived as a single one because of the spatial resolution of the photoreceptors

receptors on the retina. Besides this one-dimensional aspect, there is also a two-dimensional one: the resolution is significantly increased if the pattern changes from isolated points to straight lines. The eye is able to correlate patterns very well, and the eye's resolution for straight lines is about five times higher than for dots (this is sometimes called 'vernier's case'; Fairchild, 1998). The explanation of this is the bi-dimensional processing performed by the eye, and the orientation toward detecting bi-dimensional patterns. The natural environment, the one the eye has evolved with, mainly consists of patterns and contours, and the eye has trained itself to do well in these specific cases.

The mechanism of vision is complex and it is not within the scope of this book to explain it in detail, but instead it is analyzed from a pure image processing perspective, and, where possible, image processing interpretations are performed. Based on the fact that the eye performs differently at low levels of luminance compared with high levels, it has been observed that there are at the retinal level two kinds of receptors: *rods*, which are responsible for the low luminance levels, and *cones*, which are responsible for high levels of luminance. Numerically, there are many more rods than cones (approximately 130 million rods compared with about 7 million cones; Fairchild, 1998).

It is easy to understand that in similar conditions, the larger the number of receptors, the better the spatial resolution, which is a direct consequence of the sampling theorem. Apparently there is a contradiction, because the resolution at low luminance levels is smaller than that at high luminance levels (this is not justified by considering only the density of the receptors), and the explanation comes from the way these receptors are connected: a cone processing cell is connected to one cone only, while a rod processing cell is connected to multiple rods. The eye has adapted itself to be sensitive at low luminance levels and in doing so it averages the responses over a whole area by taking input from many rods, but the price is paid in decreasing the spatial resolution. This is equivalent to low pass filtering and downsampling where only the low frequency component matters.

The cones are responsible for perceiving colors, and they achieve this by differentiating three types of cones corresponding to three zones of the visible spectrum. The three types of cones are called red (R), green (G) and blue (B). Their sensitivity as well as the sensitivity of the rods is depicted in Figure 1.3 (which was created by interpreting the data shown in Figure 1.4(a) and (b) in Fairchild, 1998). The sensitivity of each color component is shown in Figure 1.3, and the overall spectral luminous efficiency of the rods (*scotopic* vision) and cones (*photopic* vision) is also depicted. Roughly, the peak values are 440 nm for B, 550 nm for G and 590 nm for R. Scotopic vision comes into effect at low luminance levels; for higher luminance levels it is saturated and photopic vision is applied.

The curve describing photopic vision has a significant practical implication when a specific display medium is chosen, because in a three-chromatic approach there are three types of light generator, one for each color component. Based on their individual wavelengths, their position on the photopic curve will allow the calculation of the weights for each color component.

This new classification of red, green and blue cones makes room for another clarification of the ratio of these three types of cones and their distribution on the retina. The ratio of red, green and blue cones is 40 : 20 : 1, as shown in Fairchild (1998), and the important point regarding their distribution is that there is an area on the retina almost completely unpopulated by such receptors, called the 'blind spot'. The output of these three types of cones has to be fed into ganglions; again, the eye does this in a smart way, taking a linear combination of the three fundamental components: a luminance signal (Y), a yellow–blue

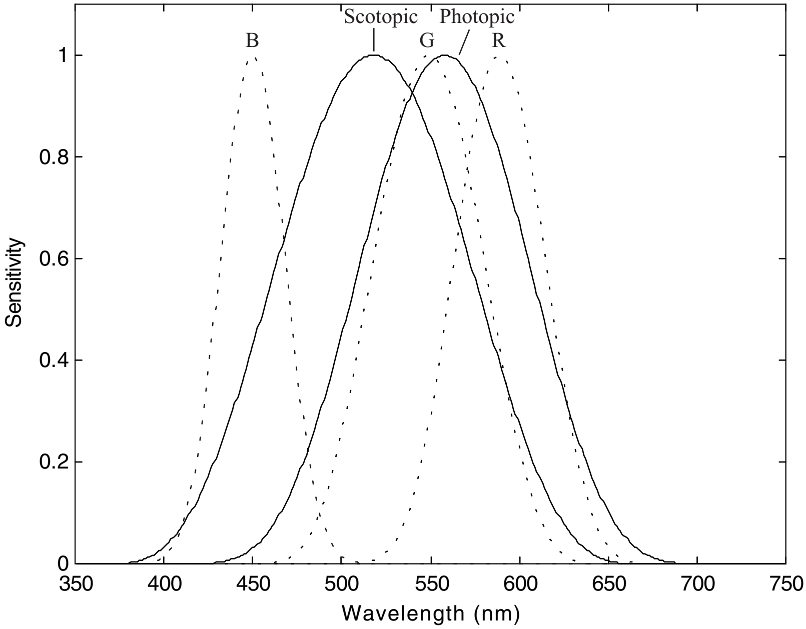

Figure 1.3 The sensitivity of cones and rods

signal (U), and a red–green signal (V). The reason for doing so is that each component has a different frequency bandwidth: that for Y is larger, and those for U and V are smaller. In image processing language, this is a *multiresolution architecture* where on one channel the information is sent with high spatial resolution, and on the other channels the information is de-correlated and they are just enhancement channels. This model allows us to explain why the visual resolution is high on luminance only, and relatively low on color. Doing some calculations of the physical dimensions of the elements forming the optical system in the eye, and using the calculated distance between two cones, the angular resolution calculated is about one minute of arc, which is in accordance with experiments. This simple result has a tremendous effect on the viewing distance for each type of display, and was one of the factors taken into account when the number of lines in the television system was chosen. Besides this purely static model, the time when the image is perceived has a significant impact. For example, a pure gray detail that is blinking at a given frequency can appear colorful. Also, a technique employed in dithering is to spread the error from the luminance channels across the color channels, which through an appropriate modulation in time can be perceived as luminance. The eye's behavior has to be understood when an artificial image is created on some sort of medium. One important characteristic is *Weber's law*, which states that if the luminance is denoted by Y, and the minimum perceived variation is ΔY, then

$$\Delta Y/Y = \text{constant} \tag{1.1}$$

This constant is important in choosing the number of quantization bits in a real system. The least significant bit will be the smallest increment possible to be achieved, and that increment still has to be below the limit of visibility. Another important factor is the contrast of a given display medium, which is defined as the ratio between the highest level of achievable luminance and the lowest level of luminance. For example, in the case of plain text on paper, the white portion reflects about 82% of the energy, and the black portion reflects about 2%, so the contrast is about 41, which leads to the conclusion that six bits of precision are enough in the printing of a document. In the case of an electronic display, for a constant of about 2% in Weber's law, we can conclude that six bits are again enough to have an apparently continuous variation of a gray level from black to white.

1.3 The Image Rescaling Problem

Mathematically this problem is simple to define. For a given three-dimensional distribution of luminance $b(x, y, t)$ we have to determine a dilated version $s(x, y, t)$ such that:

$$s(x, y, t) = b(k_x x, k_y y, k_t t)$$

for any x, y, t where $b(x, y, t)$ is defined, k_x, k_y, k_t being constants. Graphically the problem is shown in Figure 1.4, where a parallelepiped of input data is mapped onto another parallelepiped of output data, the mapping being described by the equation given above. In reality, though, the luminance is not known on the entire parallelepiped, but on a discrete grid only, which means that only $b(x_n, y_n, t_n)$ is known. This is indeed what happens at the retinal level, a process called sampling, where the incoming distribution of luminance is measured and translated into signals in some discrete physical places only.

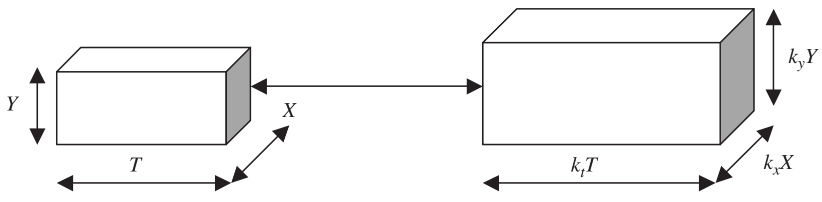

Figure 1.4 The mapping function associated with image rescaling

In the case of artificially created images the discrete aspect is also present: indeed, for a cathode ray tube (CRT) display the image is created from a collection of lines which together produce a field or a frame; for a fully digital device each picture element is individually controlled, and refreshed after a given period of time. Moved into the discrete domain, the rescaling problem has to be redefined, and it is no longer a simple one. The difficulty comes from the fact that when one picture element from the input space found at (x_n, y_n, t_n) on the grid is mapped onto the output space, its counterpart $(k_x x_n, k_y y_n, k_t t_n)$ is not always on the output space grid, so it cannot be displayed. Only the picture elements found on the grid can be displayed. The input and output spaces have a different number of nodes and a one-to-one mapping is not always possible. This is why the discrete problem is different from the continuous one, and much more difficult to solve. This problem is part of a well-known field of mathematics called *function interpolation*, which will be the tool we will utilize. But the interpolation process by itself is rarely enough because the ultimate judge is

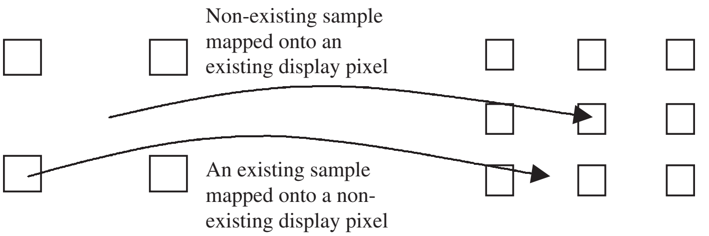

Figure 1.5 Two examples of the worst mapping case of an input image grid onto an output display grid

the human eye, which is accustomed to doing this process by itself in a fashion which satisfies its own needs, and with support from its natural computation engine, the human brain. Figure 1.5 gives two examples where there is not enough information for data mapping onto the display and interpolation is required. If the mathematical description given above is not enough to clearly demonstrate that image rescaling is one and the same thing as signal interpolation, the next clarification hopefully will do it. It is sometimes not straightforward to understand that by increasing the number of samples, the image size increases as well. In order to visualize that, it is enough to accept that the physical size of a display element is the same in both the input and output spaces, in which case more samples in the output means a larger image size. It is worth noting that the assumption of the picture element being the same size in both spaces is for explanatory purposes, and the real size of the picture element has no impact on correctly recognizing the process of image rescaling. In the rest of the book, rescaling will have the meaning of changing the number of samples without any assumption about the size of the picture element.

1.4 Digital Image Representation

In order to address the image rescaling problem properly, the image representation has to be clarified first. An image is nothing but a special case of a signal so the general theory of digital signal representation applies. The main concepts will be described in the one-dimensional case, and later the results will simply be extended to two dimensions. Let $x(t)$ be a continuous signal defined for any real value of t, satisfying the conditions:

$$\int_{-\infty}^{\infty} |x(t)|\,\mathrm{d}t < \infty \qquad \int_{-\infty}^{\infty} |x(t)|^2\,\mathrm{d}t < \infty \tag{1.2}$$

In that case the signal is guaranteed to have a Fourier transform that is defined as:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,\mathrm{d}t \tag{1.3}$$

In accordance with the *sampling theorem*, if the signal is sampled with a sampling period T, this period has to be small enough such that:

$$T < \frac{\pi}{\Omega} \quad \text{with} \quad X(\omega) = 0 \quad \text{for} \quad |\omega| > \Omega \tag{1.4}$$

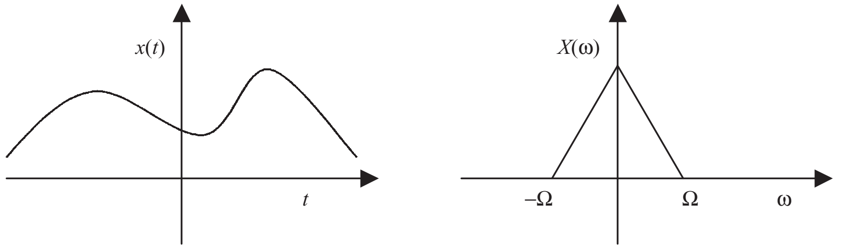

Figure 1.6 Time domain and frequency domain representation of the signal $x(t)$

A graphical representation of a signal $x(t)$ and its Fourier transform is shown in Figure 1.6. One important aspect is that during this transform, the energy is preserved. The frequency representation is more convenient to work with because the interval when the signal is not zero becomes finite (this is called signal support).

In a digital system only discrete numbers can be handled, so the signal $x(t)$ will be sampled with a sampling period T, resulting in the samples:, ..., $x[-2T]$, $x[-T]$, $x[0]$, $x[T]$, $x[2T]$, ...

Moving from the *continuous time* domain to the *discrete time* domain is always difficult, and special attention will be paid to it in this case. The mathematical result which is relevant for this process is the sampling theorem, which states that if the sampling period T is smaller than a given threshold, that is:

$$T_N < \frac{\pi}{\Omega} \tag{1.5}$$

then the continuous signal $x(t)$ can be perfectly reconstructed from its sample using the reconstruction formula:

$$x(t) = \sum_{-\infty}^{\infty} x[kT] \frac{\sin\left(\frac{\pi}{T}(t - kT)\right)}{\frac{\pi}{T}(t - kT)} \tag{1.6}$$

It is said in this case that the sequence $\{x[n]\}$ is a representation of the signal $x(t)$ in discrete space. If the sequence $\{x[n]\}$ is a perfect substitution for $x(t)$, there must be a direct relationship between the discrete sequence $\{x[n]\}$ and $X(\omega)$. Indeed, it can easily be shown that if a function $X_p(\omega)$ is defined based on this, such that:

$$X_p(\omega) = \sum_{-\infty}^{\infty} x[kT] e^{-j\omega kT} \tag{1.7}$$

then there is an interesting relationship between $X_p(\omega)$ and $X(\omega)$ given by:

$$X_p(\omega) = \frac{1}{T} \sum_{-\infty}^{\infty} X\left(\omega - k\frac{2\pi}{T}\right) \tag{1.8}$$

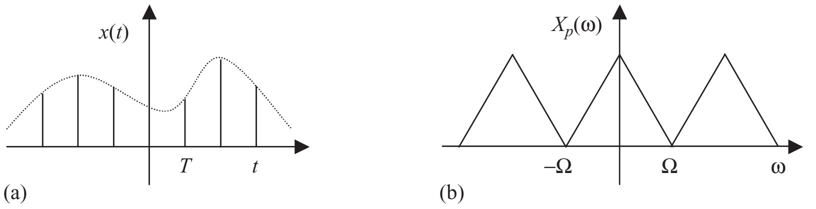

Figure 1.7 (a) The sampled signal and (b) the periodic spectrum

Equation (1.8) has a very simple interpretation, which will appear in some other cases throughout the book: by sampling the continuous signal with a sampling period T, the function $X_p(\omega)$ created with its sample is exactly generated by replicating the spectrum $X(\omega)$ with a period which is $(2\pi/T)$, as Figure 1.7 suggests. How the sampled signal is modeled has an impact on the correctness of the statement that 'the spectrum of the sampled signal results from replicating the spectrum of the continuous signal'. If the sampled signal is treated as just being a collection of discrete numbers $\{x[n]\}$ then the Fourier transform cannot be $X_p(\omega)$ because of the energy issue: $\{x[n]\}$ has finite energy and $X_p(\omega)$ has infinite energy, so Parceval's theorem is not preserved. If instead the sampled signal is seen as a sequence of Dirac distributions (delta functions) multiplying the sample $x[n]$, then the Fourier transform of such an entity is indeed the function $X_p(\omega)$, and in that case the energy requirement is satisfied. If this is not carefully addressed it could be slightly confusing because when multiplying with the train of delta functions, the energy of the resulting signal is no longer finite, and therefore does not represent the spectrum of the signal $x(t)$. For the purpose of this book, this issue is minor, and was raised here just for the sake of completeness.

$X_p(\omega)$ is just a tool to give us access to $X(\omega)$ through the discrete sequence represented by the signal samples. Also it is important to mention that in this replication scenario there is nothing to prevent us doing the replication of the spectrum as a result of sampling in the time domain even if the sampling period is smaller than T_n, which is the *Nyquist sampling period*, and the only thing that will happen will be that the replicated versions of the spectrum will overlap, making the reconstruction of $x(t)$ impossible. Based on this, the sampling theorem can easily be understood as allowing the perfect reconstruction of the continuous signal from its sample as long as the spectra do not overlap as a result of sampling. The final note about this sampling process is that, based on $X_p(\omega)$, the samples can be modified in such a way to produce a desired spectrum. The tool to achieve this is called convolution, and the process itself is called digital filtering.

The convolution process can be defined simply as starting with two discrete sequences $\{x[n]\}$ and $\{h[n]\}$ and, based on them, defining a third, $\{y[n]\}$, such that:

$$y[n] = \sum_k x[k]h[n-k] \tag{1.9}$$

It is a straightforward result of the above definition that its *associated periodic function* is:

$$Y_p(\omega) = X_p(\omega)H_p(\omega) \tag{1.10}$$

The practical value of equation (1.9) is that for a given discrete sequence $\{x[n]\}$, using a specific discrete sequence $\{h[n]\}$, the periodic associated function of the convolution can be adjusted to have a specific shape. If the aforementioned discrete sequence $\{x[n]\}$ is a representation of a continuous signal $x(t)$, through this operation the resulting discrete sequence $\{y[n]\}$ is the representation of another continuous signal $y(t)$ whose spectrum has been tailored in accordance with the shape of $H_p(\omega)$, which is the associated periodic function of the sequence $\{h[n]\}$, called a digital filter. For explanatory purposes only, the discrete sequence $\{h[n]\}$ is the representation of a continuous signal $h(t)$ called an impulse response. The sequence $\{h[n]\}$ shares the same name, *digital impulse response*.

With this tool of discrete convolution, the signal is manipulated entirely in the discrete domain, and the conversion to the continuous time domain is performed only if necessary. This is a very convenient approach for those systems that are inherently continuous in time. For example, a signal displayed on a CRT can be manipulated in such a way: the signal is sampled, digitally filtered and converted back into continuous space.

Equation (1.6) expressed a signal $x(t)$ as a sum of products between some numbers $x[n]$ and a unique function called the sinc( ) function, and this is crucial for defining the signal representation. To apply it to the image case, each number of the discrete sequence $\{x[n, m]\}$ can be seen as a weight of a shifted version of the bi-dimensional version of sinc(,). There are other possible ways of representing the signal, by taking other functions instead of sinc( ). One very good example is the Wavelet representation, which allows much greater flexibility in choosing the weighting functions with special properties that would impact the entire method of signal representation. Because the focus in this book is not on signal representation, the one that is chosen uses the sinc(,) function.

1.5 The Digital Signal Rescaling Problem

For simplicity, the problem will be formulated in the one-dimensional case: given a discrete sequence $\{x[n]\}$ representing the samples of a continuous signal $x(t)$ with $x[n] = x(nT)$, let us calculate another discrete sequence $\{y[n]\}$ with $y[n] = x(\alpha T)$. The process is illustrated in Figure 1.8.

Theoretically the problem has a simple solution: from the discrete sequence $\{x[n]\}$ the continuous signal $x(t)$ can be retrieved based on relationship (1.6), and later on the value of the signal at any position can be found. The technical problem with this is that a sample $y[k]$

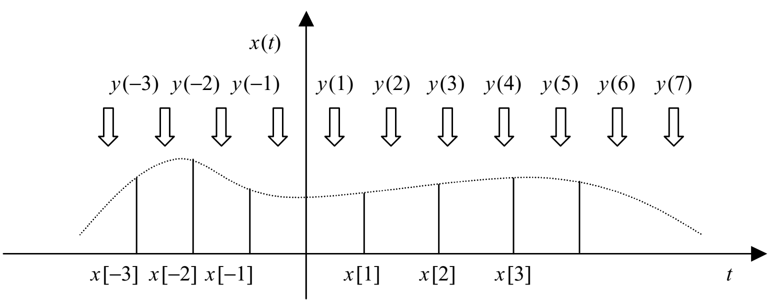

Figure 1.8 Graphical description of the digital rescaling problem, where $x[\ ]$ is the input and $y[\ ]$ is the output

will be dependent on all the input samples $x[n]$, which makes the calculation difficult. It would be even more difficult to determine the samples of the interpolation function $\sin(x)/x$ as results from the following formula:

$$y(n\alpha T) = \sum_{-\infty}^{\infty} x[kT] \frac{\sin\left(\frac{\pi}{T}(n\alpha T - kT)\right)}{\frac{\pi}{T}(n\alpha T - kT)} \tag{1.11}$$

In practice this is difficult to implement and alternatives must be found.

One value of the theoretical method is to compare various interpolation algorithms by calculating the error between what a specific method would give and the ideal (theoretical) one.

But even the theoretical method has limitations when the discrete sequence is a result of a sampling process with a sampling period smaller than the Nyquist sampling period. This sub-Nyquist sampling rate produces an erroneous result in a system where a continuous signal exists, for example in a CRT display, but there are cases where the continuous signal does not exist at all, for example in a digital display, which displays a computer-generated signal. In such cases the signal is created digitally and is displayed digitally. Everything works just fine until rescaling is involved. To illustrate this, an example is shown in Figure 1.9, where a rectangular window signal is chosen for interpolation. The discrete sequence $\{x[n]\}$ is a rectangular pulse type, where the signal suddenly changes from minimum to maximum and vice versa, and when displayed, the eye will perceive a sharp change from black to white, for example. When rescaled digitally, the resulting sample will follow the envelope of the continuous-time signal which is represented by the $\{x[n]\}$ that is no longer an ideal pulse, because of the finite bandwidth and finite sampling rate involved. When displayed, the eye will reject the new shape as not being equivalent (rescaled version) to the previous one. The major difference between analog and digital display systems is that the digital system creates edges that are not possible in a finite bandwidth system. The explanation for this resides in the fact that in a digital display system, each individual picture element is driven independently, and acts as a signal by itself, and the meaning of a unique signal with a limited bandwidth is lost. This aspect poses difficult problems in a digital rescaling task because the theoretically simple problem of signal interpolation is replaced by the more challenging one of how to render the signal in such a way as to produce more

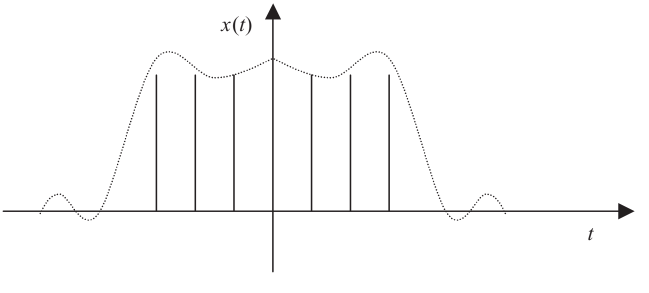

Figure 1.9 The continuous signal that is the counterpart of a rectangular window digital signal when interpolated theoretically

samples and at the same time appearing to the eye as similar to the initial version. This is why digital image rescaling is far from being a trivial task.

1.5.1 The Time Domain Approach to the Rescaling Problem: An Example

The rescaling problem is a good example of concurrent time domain and frequency domain solutions. There is a duality between the two domains, and almost any operation in one has its counterpart in the other. People prefer one or the other depending on their ability to handle time domain concepts or frequency domain concepts. Sometimes time domain tools are easier to manipulate; sometimes frequency domain apparatus is more effective. The approach in this book is to examine both of them. Figure 1.10 illustrates the concept of duality between the time and frequency domains: whatever is achieved in the processing stage in the time domain can be achieved in a processing stage in the frequency domain. Some of the functions are easier to perform in the frequency domain, but the cost of moving back and forth between domains is significant, and this is not the case with systems involved in digital image display. The processing is done in the time domain, but the algorithms are sometime generated to match the processing in the frequency domain.

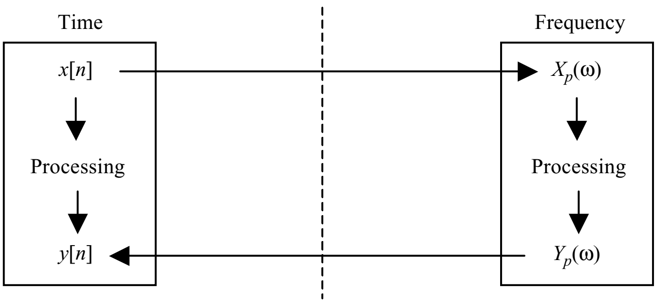

Figure 1.10 The two alternative paths to go from $x[n]$ to $y[n]$

To begin with, let us examine the simple task of doubling the number of samples of a given discrete signal $\{x[n]\}$. This process is called *upsampling*, and in this specific case would be an upsampling by a factor of 2. If the assumed sampling period for $\{x[n]\}$ was considered to be T, then the sampling period for the upsampling signal $\{y[n]\}$ would be $T/2$, and every second sample in $\{y[n]\}$ would coincide with a sample in $\{x[n]\}$; in other words $y[2n] = x[n]$.

The fact that some of the samples coincide is not a requirement, but it will be assumed to be true, unless clearly specified otherwise. So for any integer $n, y[2n] = x[n]$, $y[2n+2] = x[n+1]$, and the only one to be determined is $y[2n+1]$. Instead of using relationship (1.6), which would permit us to calculate $y[2n+1]$ exactly (which would be useless because the signal $\{x[n]\}$ could be aliased to begin with), it is recognized that the exact formula will take less and less contribution from the samples further away from the current interpolated one, and that is because of the shape of the $\sin(x)/x$ function, the interpolation function in the specific case of the signal representation chosen. The contribution from additional samples can be reduced as much as is wanted. For a desired output sample, the set of the input samples which are taken into consideration to generate it define a so-called

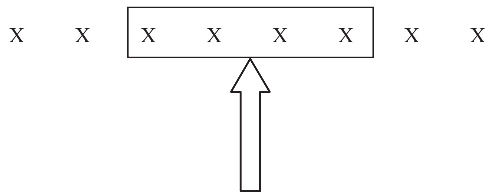

Figure 1.11 An illustration of the convolution window associated with a digital filtering process

convolution window which in the theoretical case is infinite, and which in practice can be reduced to any positive integer, including 1. In Figure 1.11 the convolution window which takes the closest w input samples to the desired output is illustrated (for illustration purposes $w = 4$).

For the simplest possible case the window length is chosen to be 1, and the output is chosen to be equal to its closest existing neighbor. For the case under study of upsampling by 2, there are two choices: $y[2n + 1] = x[n]$ or $y[2n + 1] = x[n + 1]$. If the output sample does not fall exactly in between, there is only one possibility when choosing the closest neighbor.

1.5.2 The Frequency Domain Approach to the Rescaling Problem: The Counterpart Example

Assuming a given discrete signal $\{x[n]\}$ and its associated periodic spectrum $X_p(\omega)$, let us determine the digital filter $\{h[n]\}$ that accomplishes the upsampling done above, and let us examine the function performed in the frequency domain, which means calculating the periodic function $H_p(\omega)$. The ultimate goal would be to see what happens when the two are multiplied to generate the spectrum of the result $Y_p(\omega) = X_p(\omega) \times H_p(\omega)$. It is the unique privilege of the frequency approach to give a clear meaning in terms of the spectrum of what happens during digital filtering. This information is not accessible from the time domain approach, and from this perspective the frequency domain solution is more complete, and easier to evaluate. The sequence of operations that achieves the upsampling by 2 has to be determined first in a logical way.

For a given discrete sequence $\{x[n]\}$, the periodic associated function is shown in Figure 1.12(a), being a replicated version of $X(\omega)$ with a period of $(2\pi/T)$, and the periodic associated function of the signal upsampled by 2 has to be a replicated version of $X(\omega)$ with a period of $(4\pi/T)$, as shown in Figure 1.12(d). The way to get from the spectrum in Figure 1.12(a) to the one in Figure 1.12(d) is to build an intermediate signal whose periodic associated function is shown in Figure 1.12(b) and filter that with a digital filter whose periodic associated function is shown in Figure 1.12(c). An exact understanding of the process described in Figure 1.12 is critical for comprehending the idea of *digital filtering* in the frequency domain. The signal in Figure 1.12(b) has to be generated somehow, by shrinking by a factor of 2 the spectrum of the signal in Figure 1.12(a). The filter shown in Figure 1.12(c) is selecting only the lower half of the spectrum, and it is deliberately labeled as having period 2π. There is no conflict between the period 2π of the filter and the period $2\pi/T$ of the signal, the reason being that when the digital filter is applied, both the signal and the digital filter will be sampled at the same sampling rate and the sampling rate no longer matters, so the scale can be chosen such that the period is 2π. This is called *normalized frequency*, and it corresponds to a sampling period of 1. The transition back from normalized

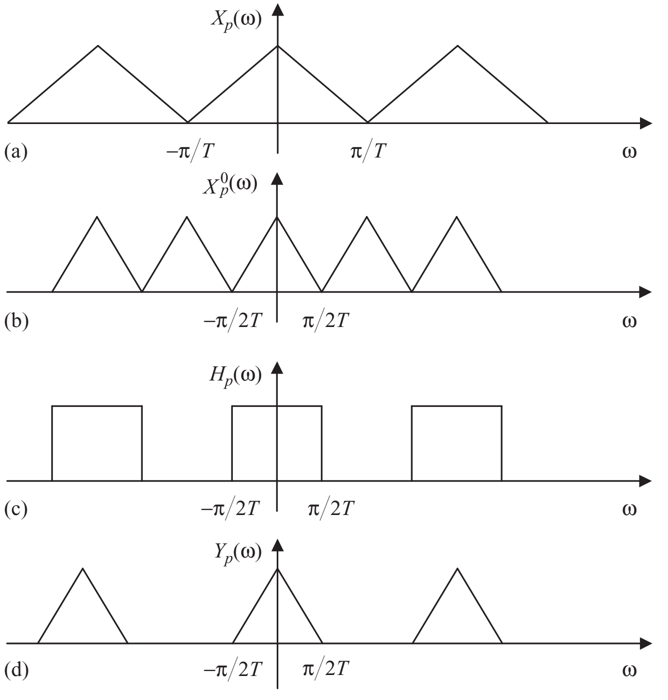

Figure 1.12 Upsampling by a factor of 2: (a) the periodic associated function of the input signal, (b) the shrunk by 2 version, (c) the filter's characteristic and (d) the periodic associated function of the upsampled signal

frequency to frequency is always possible, just taking into account the sampling rate. In the case of Figure 1.12(d), the sampling rate is $T/2$ and that fully justifies the labeling of the frequency axis.

The spectrum plotted in Figure 1.12(d) represents the periodic associated function of the signal upsampled by 2, because the spectra have been replicated with a period of $4\pi/T$, that is double the initial $2\pi/T$. The only remaining problem is to create the signal with a shrunk periodic associated function like the one in Figure 1.12(b), and to generate a filter with filter characteristic like the one shown in Figure 1.12(c). To produce the signal whose periodic associated function is represented in Figure 1.12(b), we will make the observation that when the spectrum shrinks twice, the time version of that signal has to suffer a dilation by 2 and the discrete sequence should be $\{x^0[n]\}$ with:

$$\begin{aligned} x^0[2n] &= x[n] \\ x^0[2n+1] &= 0 \end{aligned} \tag{1.12}$$

That means that between every sample of the original discrete sequence, a sample equal to zero will be inserted. Indeed, if we recall the definition of $X_p(\omega)$:

$$X_p(\omega) = \sum_{-\infty}^{\infty} x[kT]e^{-j\omega kT} \tag{1.13}$$

it is straightforward to observe that:

$$X_p^0(\omega) = \sum_{-\infty}^{\infty} x[kT]e^{-j2\omega kT} \tag{1.14}$$

and therefore

$$X_p^0(\omega) = X_p(2\omega) \tag{1.15}$$

The digital filter has to be a *low pass filter* (LPF) with the filter characteristic shown in Figure 1.12(c). The gain of the filter – that is, the value at frequency equal to 0 – has to be equal to 2. There are both theoretical and practical explanations for this normalization factor. The theoretical explanation is based on relationship (1.8) where the sampling period appears at the denominator, and when the sampling period becomes half of the original one, a factor of 2 appears at the nominator. The practical explanation is that through the process of filtering, one spectrum is removed, and in order to compensate for the energy lost, the remaining one has to be amplified by a factor of 2, because the power has to remain the same.

The digital filter that performs the closest neighbor selection is $\{h[n]\}$ with $h[0] = 1$; $h[1] = 1$, and zero otherwise. This will preserve the existing values and will take as interpolated values the closest sample to the right. There is another alternative to create the digital filter $\{h[n]\}$ with $h[-1] = 1$; $h[0] = 1$, and zero otherwise, which will select the closest sample on the left. Intuitively this is simple to implement, and before evaluating it, let's see how good such a filter is. The function $H_p(\omega)$ is given by:

$$H_p(\omega) = 1 + e^{-j\omega} \tag{1.16}$$

and the frequency characteristic is represented in Figure 1.13.

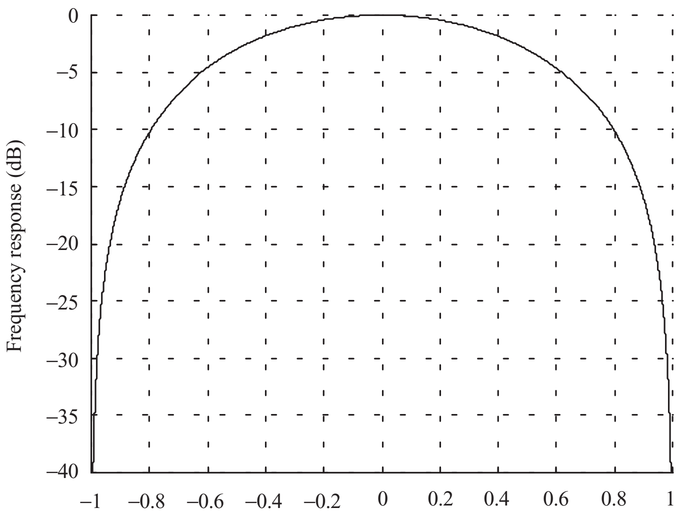

Figure 1.13 The frequency characteristic of the filter described by equation (1.16)

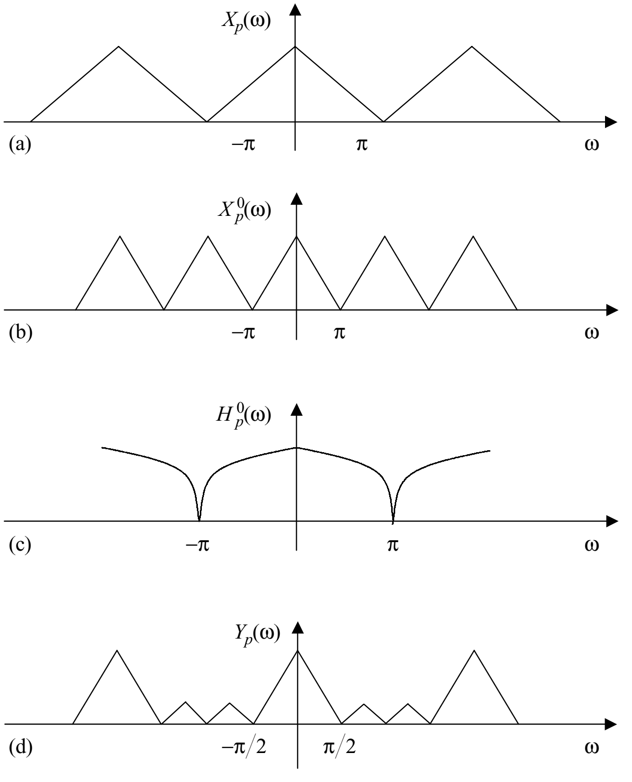

Figure 1.14 (a) The periodic associated function of the input signal, (b) the periodic associated function of the signal with zero inserted, (c) the filter's characteristic and (d) the periodic associated function of the output signal

The normalized frequency means that the limits -1 and 1 correspond to $-\pi$ and π. This is sometimes a more convenient representation, and it results from scaling the x axis by a factor of π. In the case of a digital filter the $[-\pi\ \pi]$ interval is a full period, and in the case of a sequence coming from a continuous signal after sampling, π corresponds to half of the sampling frequency. With the above filter the spectral representations of the diagrams in Figure 1.12 become the ones shown in Figure 1.14. There are two important aspects about Figure 1.14(d), when compared with Figure 1.12(d) representing the ideal case: first, there is unwanted attenuation between 0 and $\pi/2$, and there is also unwanted gain between $\pi/2$ and π. In terms of digital filters, the two bands are called *pass-band* and *stop-band*. To give a more intuitive sense of these deviations from the ideal case, an example will be selected. Consider the one-dimensional case of a ramp signal with a step of two consisting of eight samples: 2, 4, 6, 8, 10, 12, 14, 16. After the upsampling by 2 and filtering with the above filter the sequence will become 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14, 16, 16. This is not exactly what one would have expected to get as a result of interpolating a ramp, which would be 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, and this is because of the poor frequency characteristic of the filter involved. As an example of the visual impact of the interpolation filter on a real image, images of a ramp with a step of 32, interpolated with the above filter

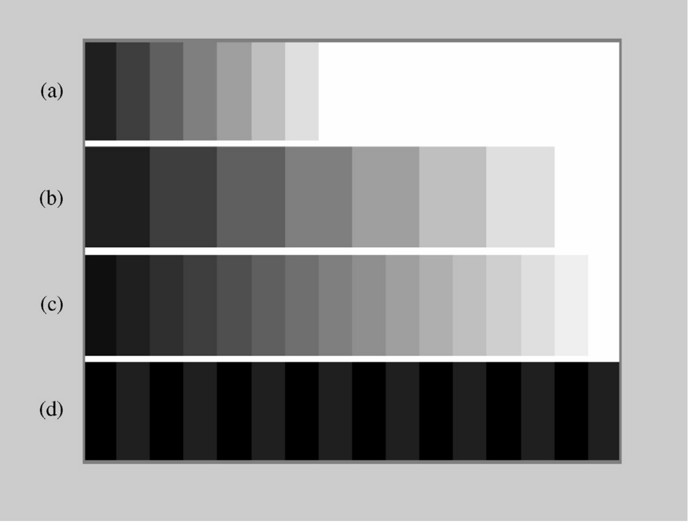

Figure 1.15 (a) A ramp image, (b) the image after pixel replication, (c) the image after linear interpolation, (d) the difference between the two

and with a filter closer to the ideal one, are shown in Figure 1.15. The filter characteristics for both filters involved in the example in Figure 1.15 are shown in Figure 1.16. In order to fully explain the differences between the two filters, the frequency representations of both signals (one-dimensional signals representing one line only) are shown in Figure 1.17.

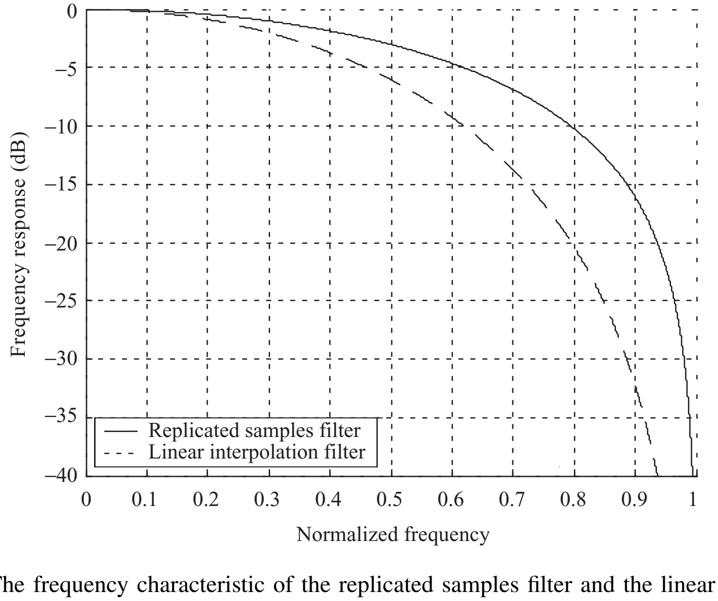

Figure 1.16 The frequency characteristic of the replicated samples filter and the linear interpolation filter

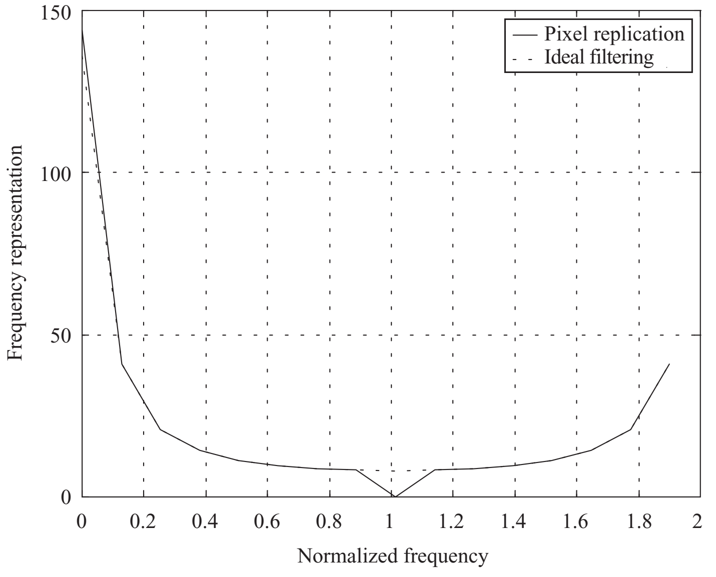

Figure 1.17 The spectrum of the output signal processed with a pixel replication filter and with a better filter

Fully explaining Figure 1.17 takes more effort than one would expect for such a simple case as the one under study. Indeed, apparently there is a paradox because the image band shown in Figure 1.15(c) has more high frequencies than that in Figure 1.15(b), generated by pixel replication, but the filter representing pixel replication has less attenuation in the stop-band, and one would expect greater content of high frequencies in Figure 1.15(b), exactly the opposite of what is seen. The explanation for this apparent paradox consists in the aliasing produced by the filter. The filter closer to the ideal filter has everywhere more attenuation than the pixel replication filter, and the signal is in fact low-pass filtered and looks like the one in Figure 1.15(c). The pixel replication filter, because of its smaller attenuation in the stop-band, will convert some of the low frequencies into high frequencies, and these are in fact exactly as shown in the lower band image, but with a negative sign, so the narrower bands vanish, giving the false impression that there are no high frequencies. When the linear filter is applied, this aliasing component is removed, and the high frequency component is no longer canceled, giving the appearance of the finer step for Figure 1.15(c). This is an excellent example of how aliasing manifests itself, and how it makes life difficult when it comes to giving an interpretation to the image in accordance with its native frequency content.

As shown in the above example, sometimes using a filter that produces aliasing works to our advantage, and it is the challenge facing the filter designer to come up with a digital filter that works best in most cases (for a given level of complexity).

1.6 A Bi-dimensional Example of Image Rescaling

An image is by definition a bi-dimensional entity. Going from one dimension to two dimensions is as important as going from dimension zero (a dot) to dimension one (a line), where a whole set of extra properties applies. As a simple case, there is always a straightforward extension from one dimension to two dimensions by just iterating the same approach vertically (column by column) and horizontally (line by line). Besides this there is also a whole set of processing methods specific to the bi-dimensional case. Throughout this book, a common element for both the hardware and software approaches is explored, and that common element is the algorithm that has to be implemented. As a general rule, hardware implementation is appropriate for real-time systems where the enormous number of calculations can only be performed by dedicated hardware, and software implementation is appropriate for non-real-time systems where the processing is performed off-line and the result is submitted for examination when it is ready. With the speed of available systems increasing rapidly, the boundary between software and hardware implementation is fading away, and more and more applications that used to require dedicated hardware can now be performed on a PC. This is why the description of the algorithm itself is the main task, and whenever possible descriptions of the software and hardware implementation are provided.

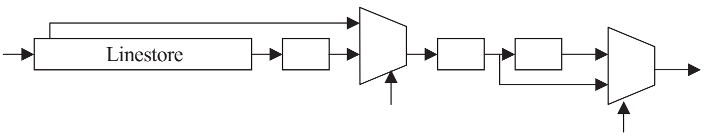

Figure 1.18 The hardware architecture of a pixel replication interpolation system

Hardware that performs the zero order interpolation (pixel replication algorithm) is illustrated in Figure 1.18. There is a linestore, a multiplexing circuit controlled by the output line clock (which is twice as fast as the input line clock), and a multiplexing circuit controlled by the output pixel clock (which is twice as fast as the input pixel clock). A software implementation is described by the following pseudo-code:

```
for each input line
{
    copy the line;
    {
        for each pixel on the line
        {

            output the pixel;
            output the pixel;
        }
    }
    copy the line;
```

```
    {
        for each pixel on the line
        {
            output the pixel;
            output the pixel;
        }
    }
}
```

Although both implementations perform the same function, there are very different restrictions for software and hardware solutions. For hardware, the main restriction is the amount of hardware involved, especially the memory. Whenever possible, the logic can be made more sophisticated if this saves memory. For software, memory is not a concern, but the speed of execution and the logic involved are the main optimization restrictions. Whenever the logic can be made simpler at the expense of more memory, or the execution faster by using a larger amount of memory, it is recommended to do so.

Figure 1.19 The image 'Couple'

To conclude this chapter describing pixel replication, the image in Figure 1.19 is scaled using the pixel replication algorithm, and the result is shown in Figure 1.20. This method of *image scaling* is the oldest, and is still implemented in many systems, principally because of its simplicity: there is very little hardware involved, just a linestore, a few registers for storing pixels, two multiplexing circuits and a simple control. There is no real processing, because the filter involved has coefficients equal to one. Most probably some system designers were not even aware of the existence of such a digital filter and its frequency characteristic. The approach was from the time domain, and as long as the number of pixels was doubled in each direction, everything was fine. One example where this algorithm is applied is in old projectors, because there was the need to rescale an image in real time. One can easily recognize this by looking at the edge of a projected image, after adjusting the keystone correction to compensate for the trapezoid that appears when the image is projected

Figure 1.20 The image 'Couple' after pixel replication

from a distance. Instead of a nice smooth edge, the edge is stair-stepped. The main artifact of this algorithm is its blocky appearance and a diagonal edge when scaled will look like a stair instead of looking smooth and continuous. It would be surprising to see anyone applying this algorithm today, and the decision to include it here was just because of its simplicity and because it was considered a good example of how image rescaling works.