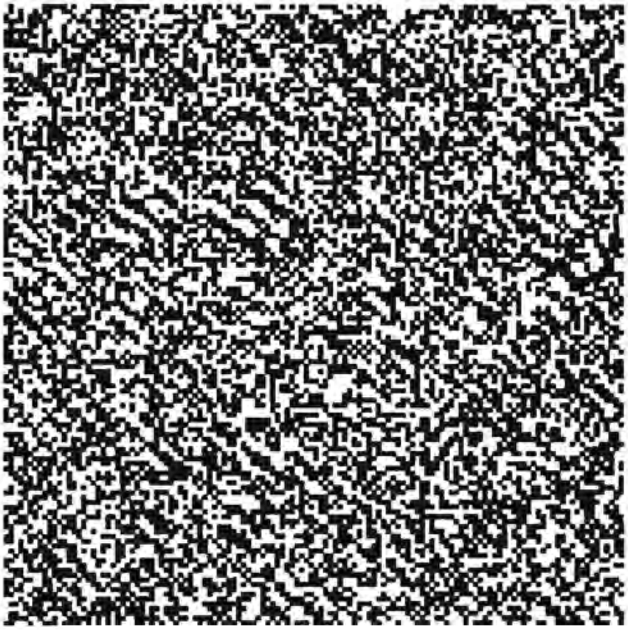
Computer-generated Fourier holograms based on pulse-density modulation

DOI:
10.6100/IR417117

Document status and date:
Published: 01/01/1994

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

COMPUTER-GENERATED FOURIER HOLOGRAMS BASED ON PULSE-DENSITY MODULATION



P. VAN DEN BULCK

COMPUTER-GENERATED FOURIER HOLOGRAMS BASED ON PULSE-DENSITY MODULATION

COMPUTER-GENERATED FOURIER HOLOGRAMS BASED ON PULSE-DENSITY MODULATION

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof. dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op dinsdag 24 mei 1994 om 16.00 uur

door

PIERRE MARIE THEODOOR VAN DEN BULCK

geboren te Hulst

druk: wibro dissertatiedrukkerij, helmond.

Dit proefschrift is goedgekeurd
door de promotoren
prof. dr.-ing. H.J. Butterweck
prof. dr. ir. J.P.M. Schalkwijk
en de copromotor
dr. ir. M.J. Bastiaans

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Bulck, Pierre Marie Theodoor van den

Computer-generated Fourier holograms based on
pulse-density modulation / Pierre Marie Theodoor van den
Bulck. - Eindhoven : Technische Universiteit Eindhoven
Proefschrift Eindhoven. - Met lit. opg.
ISBN 90-386-0283-9
Trefw.: computerhologrammen / neurale netwerken.

Contents

Summary

Often, an output device with binary output is used for the realization of computer-generated holograms. This requires a transformation of the calculated hologram transmittance function into a binary function. For binary holograms based on pulse-density modulation, this function is assumed to consist of nonoverlapping identical binary pulses. The desired optical properties are approximately obtained by an appropriate modulation of the local density of the pulses. For Fourier holograms a small difference between the spectra of the original and the binary hologram is required in a given frequency band.

Continuous pulse-density modulation allows a free positioning of the pulses. For one-dimensional signals we apply an integration concept in order to determine the positions of the pulses, either individually or for small groups of pulses simultaneously. For the various methods the approximation error as a function of the pulse density is estimated. Computer simulations verify the theory for the lower-order methods. Reformulating the integration concept in differential form, one can derive a set of nonlinear partial differential equations for two-dimensional pulse-density modulation. Two-dimensional pulse-density signals with a relatively small number of pulses have been obtained by solving these equations numerically. However, in its present implementation this approach requires too much computation for a large number of pulses. In a concluding section the relation of continuous pulse-density modulation with known methods for the clustering of points has been discussed.

For discrete pulse-density modulation the positioning of the pulses is restricted to fixed raster points. The problem of determining where to place the pulses is now reduced to deciding for the individual raster points whether to place a pulse or not. In a natural way the integration concept developed for continuous pulse-density modulation then passes into error diffusion. In order to apply error diffusion on a two-dimensional raster, a processing order of the raster points is introduced. The quantization errors introduced at previous raster points are weighted by diffusion coefficients and taken into account during the decision for a certain raster point under consideration. A linear model for error diffusion shows how the diffusion coefficients can be employed in order to obtain a small deviation of the spectra within the given frequency band. Unstable behaviour of the recursive error diffusion system is avoided by a proper choice of the diffusion coefficients. A method for the determination of appropriate and stable diffusion coefficients is presented. Applying

the error diffusion system with the calculated diffusion coefficients results in binary holograms with a small deviation in the given frequency band.

The generalization of error diffusion which is not recursively computable leads to a Hopfield neural network. The interconnection weights and the thresholds of the neurons in this network are determined by the original hologram and the frequency band used. The binary hologram is found as the outputs of the neurons. Given an initial pulse distribution, the neuron outputs are updated until some final solution is reached. However, despite the additional amount of computation, compared to error diffusion, the results are rather disappointing. Since the network performs a local search during updating, the obtained pulse distribution is optimal with respect to its local neighbourhood. A Boltzmann machine, which is the combination of a Hopfield network with simulated annealing, allows escaping from local optimal solutions, such that better solutions can be found. An efficient implementation of the Hopfield network is possible due to the special structure of the interconnection weights. This structure is also exploited for the presented Boltzmann machine with parallel updating. This leads to substantially improved binary holograms with a reasonable computational effort.

Chapter 1

Introduction

The design of computing and signal processing systems based on parallel architectures is receiving more and more attention. For the realization of parts of such systems (e.g. interconnections) optics offers interesting design potentialities. An important component of optical systems is the holographic element, which serves as a wavefront-shaping device. The conventional way to realize a hologram is the photographic recording of an interference pattern. With a more recent technique the desired pattern is generated by means of a digital computer. Digital holography offers possibilities going beyond those afforded by classical holography. Moreover, computer-generated holograms are relatively small and light, and are suitable for mass production. In Chapter 2 of this thesis, a concise introduction to classical holography is presented, followed by the basic principles of computer-generated Fourier holograms. A more detailed discussion of computer-generated holograms can be found in recent review articles by Lee (1978), Dallas (1980) and Bryngdahl and Wyrowski (1990).

The calculated hologram transmittance is realized by means of an output device, such as an e-beam lithograph or a plotter. Since most output devices generate binary output, the transmittance function of the hologram has to be transformed (quantized) into a binary signal. In this thesis we investigate how pulse-density modulation can be applied in order to obtain binary holograms. The transmittance function is assumed to consist of nonoverlapping identical binary pulses, while the desired properties of the hologram are obtained by an appropriate modulation of the density of the pulses. For Fourier holograms the spectrum of the original hologram should equal the spectrum of the binary hologram in a given frequency band. In general, this is only approximately achieved with pulse-density modulation. The necessary quality measures for the binary hologram are discussed in Chapter 2.

Pulse-density modulation is subdivided in continuous pulse-density modulation and discrete pulse-density modulation. With continuous pulse-density modulation, treated in Chapter 3, the positioning of the pulses is not restricted, except that overlap of the pulses is forbidden. A one-dimensional pulse-density signal is obtained by means of an integration concept (Eschbach and Hauck, 1987). Basically, in its graphical representation the positive hologram transmittance function is represented as a sum of nonoverlapping equal-area 'slices'. Each slice is represented by a binary pulse of the same area, where the pulse position is determined by the specific form of the slice. The result is a signal with a local pulse density which is proportional to the amplitude of the transmittance function. Variations on this concept, where the positions of small groups of pulses are determined simultaneously, are also considered in Chapter 3. Pulse-density modulation based on the integration concept is closely related to known methods for numerical integration. Due to this relation the approximation error of the various methods can be easily estimated.

A straightforward extension of the integration concept for two-dimensional signals is not possible. However, reformulating the one-dimensional problem in a differential form, one can derive a set of nonlinear partial differential equations for the two-dimensional case. The pulse positions of the two-dimensional pulse-density signal are then obtained by solving this set of equations numerically. Chapter 3 is concluded with a short discussion of the Linde-Buzo-Gray algorithm and the Kohonen neural network in the context of the vector quantization problem, which is related to pulse-density modulation.

For discrete pulse-density modulation pulses are restricted to be positioned at fixed raster points only. Under this condition one-dimensional continuous pulse-density modulation leads to a modulation technique known as error diffusion (Floyd and Steinberg, 1976). The determination of the binary hologram is now reduced to a decision problem: for each raster point we have to decide whether a pulse is placed or not. With error diffusion the raster points are processed sequentially, introducing an error in each decision. Basically, the error of the previous raster points is diffused to the raster point under consideration, and there influences the decision to be made. Due to the recursive nature of error diffusion, stability is an important issue. In order to apply this concept to two-dimensional signals an appropriate order of processing of the points in the two-dimensional raster is introduced. Error diffusion is employed in Chapter 4 to obtain binary holograms.

With error diffusion the raster points are processed recursively. However, this is not a basic requirement, and the finiteness of the number of raster points admits other solutions. In Chapter 5 we consider an iterative approach as given by the sequential updating law of a discrete-time Hopfield neural network. In addition, a parallel updating law is discussed. The performance of error diffusion and that of the Hopfield neural network is compared in Chapter 5. Since the Hopfield network performs a local search during updating, a locally optimal binary hologram is found. With a Boltzmann machine, which combines the updating law of a Hopfield neural network with simulated annealing, it is possible to escape from local optimal solutions and to find better pulse distributions. Also in Chapter 5 the relation between Hopfield's neural network and other iterative techniques (direct binary search, the iterative Fourier-transform algorithm, projections on convex sets) is discussed.

Chapter 2

Computer-generated holograms

2.1 Holography

In 1948 D. Gabor invented the method of holography (Gabor, 1948). He realized that it is possible to encode both amplitude and phase information of an optical wave (emanating from some coherently illuminated object) in an interference pattern. When this pattern is recorded in photographic material, the resulting hologram holds all information about the object. With proper illumination of the hologram a virtual or real image of the original object appears. Since a monochromatic coherent light source is needed to make a hologram, it was not before the invention of the laser in the 1960's that holograms of reasonable size could be realized. One possible way to realize a hologram with a coherent light source is shown in Figure 2.1. An expanded laser bundle is split into two bundles by means of a half-transmitting mirror. The reflected bundle illuminates an object from which, due to reflection

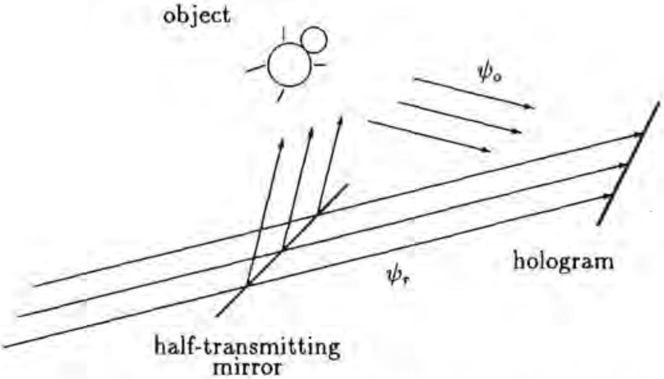


Figure 2.1: Optical recording of a hologram. In the hologram the interference pattern $|\psi_o + \psi_r|^2$ is recorded.

and refraction, a secondary wave called the 'object bundle' $\tilde{\psi}_o(x,y,z,t)$ emanates. This bundle interferes with the directly transmitted 'reference bundle' $\tilde{\psi}_r(x,y,z,t)$. Particularly, this holds true at the photographic plate which records the time average of the squared total field $(\tilde{\psi}_o+\tilde{\psi}_r)^2$. Due to the assumed harmonic time dependence with frequency ω we have

$$\begin{aligned}\tilde{\psi}_o(x,y,z,t) &= \mathrm{Re}\left[\psi_o(x,y,z)e^{-i\omega t}\right]\\ \tilde{\psi}_r(x,y,z,t) &= \mathrm{Re}\left[\psi_r(x,y,z)e^{-i\omega t}\right] ,\end{aligned} \tag{2.1}$$

where $\psi_o(x,y,z)$ and $\psi_r(x,y,z)$ denote the complex field distributions of the object and reference bundle, respectively. Superposition of $\tilde{\psi}_o$ and $\tilde{\psi}_r$ then implies superposition of ψ_o and ψ_r, and the photographic process implies recording of the squared absolute value of the resulting complex field $(\psi_o+\psi_r)$. More specifically, the silver density of the exposed and developed photographic material is determined by

$$|\psi_o+\psi_r|^2 = (\psi_o+\psi_r)(\psi_o^*+\psi_r^*) = |\psi_o|^2+|\psi_r|^2+\psi_o\psi_r^*+\psi_r\psi_o^* . \tag{2.2}$$

In a linear approximation the silver density is proportional to $|\psi_o+\psi_r|^2$, while for an ideal 'positive' film the transparence is proportional to this function. The latter approximation is assumed to be valid in our further considerations.

Under certain conditions, the interference pattern stored in the photographic plate holds all information about the original object, meaning that with proper illumination of the photographic plate we must be able to reconstruct the original object. In this reconstruction step we illuminate the hologram with the reference field used in the recording procedure. Under the assumption that the (transmission) hologram is thin, it modulates the complex amplitude of the incident reference wave, resulting in a field

$$|\psi_o+\psi_r|^2\psi_r = (|\psi_o|^2+|\psi_r|^2)\psi_r+\psi_o|\psi_r|^2+\psi_o^*\psi_r^2 \tag{2.3}$$

behind the hologram. Besides a zero-order diffraction term $(|\psi_o|^2+|\psi_r|^2)\psi_r$, the hologram reconstructs a field distribution $\psi_o|\psi_r|^2$ in the first-order diffraction, which under the (legitimate) approximation $|\psi_r|^2 = 1$ is identical to the field that originated from the original object. The last term $\psi_o^*\psi_r^2$ denotes the twin image of the object in the first negative diffraction order. When these contributions can be spatially separated, the original object appears as shown in Figure 2.2.

Soon one realized that an alternative to the recording procedure in optical holography is to calculate and realize the interference pattern using a digital computer. This was the start of digital holography. Given a mathematical description of the desired object, we calculate the wavefront field which has to be recorded in the hologram. In general the result will be complex and has to be transformed by a (first) coding procedure into a real signal in order to make realization as a transparency possible. One way to achieve this is to simulate the recording process of optical holography. However, with digital holography we are able to manipulate signals

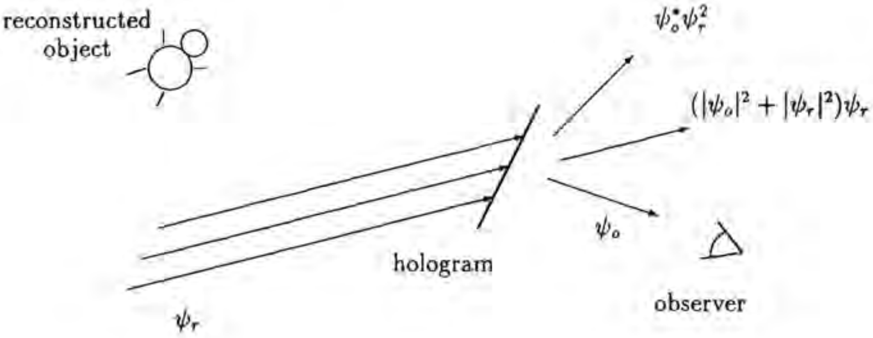


Figure 2.2: Reconstruction of the object. With proper illumination of the hologram the original object is observed.

in ways that have no counterpart in optical holography. This, and the fact that the object does not have to exist physically, is a major advantage of digital holography. After having calculated the hologram transmittance, an output device, e.g. a laserprinter, is used in order to realize the transparency. Since most output devices are able to generate binary output only, the continuous transmittance has to be transformed into a binary signal by means of a second coding procedure. When the binary hologram is realized and used in the reconstruction procedure, it will generate a reconstruction that deviates from the desired object due to the procedures involved in the calculation of the hologram. This thesis is particularly concerned with the effects of the second coding procedure, i.e. quantization, with the final aim to calculate binary holograms that generate a good approximation of the desired object.

2.2 Computer-generated Fourier holograms

Since holograms store the entire information content of the original object, it is possible to generate 3-dimensional images with holograms. In addition to this application, holograms are also applied in optical systems as subsitutes for conventional optical elements, such as lenses, and for the storage and filtering of optical signals. The last functions are difficult or even impossible to achieve with conventional optical elements. The holograms discussed in this thesis are supposed to belong to this category.

We restrict ourselves to the calculation of holograms used in the optical system shown in Figure 2.3. This system consists of a lens with focal distance f surrounded by free space. An input signal is applied in the front focal plane of this system and transformed into an output signal in the back focal plane. The input signal is realized by illuminating a hologram placed in the input plane with a (coherent) plane wave. The hologram generates in the output plane a signal which we call (in analogy with optical holography) 'the reconstruction' [1]. For this reason we will call the input plane the hologram plane and the output plane the reconstruction plane.

Without imposing any restrictions on the input signal (related to the physical realization of the hologram), we analyze the output signal $\psi_o(x_o, y_o)$ measured in the output plane when a possibly complex input signal $\psi_i(x_i, y_i)$ is applied in the input plane. Under the assumption that the smallest spatial period occurring in $\psi_i(x_i, y_i)$ is large compared to the wavelength λ of the incident wave, we are allowed to use scalar diffraction theory to calculate the field distribution in the output plane. Using the Fresnel approximation and the notion of a 'thin lens' acting as a phase modulator with a quadratic phase characteristic, we find for the output signal (Goodman, 1968)

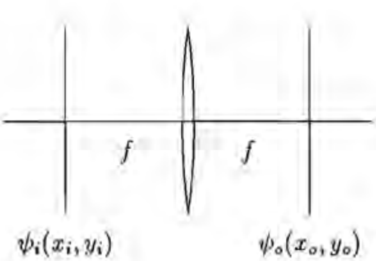


Figure 2.3: Optical Fourier transform. The hologram in the input plane is illuminated with a (coherent) plane wave.

$$\psi_o(x_o, y_o) = -\frac{ik}{2\pi f} e^{ik2f} \iint \psi_i(x_i, y_i) e^{-ik\frac{x_o x_i + y_o y_i}{f}} \mathrm{d}x_i \mathrm{d}y_i \ , \tag{2.4}$$

with $k = 2\pi/\lambda$ the wave number [2]. Introducing the normalized coordinates

$$x = \frac{x_i}{\sqrt{\lambda f}} \qquad y = \frac{y_i}{\sqrt{\lambda f}} \tag{2.5}$$

for the front focal plane and

$$u = \frac{x_o}{\sqrt{\lambda f}} \qquad v = \frac{y_o}{\sqrt{\lambda f}} \tag{2.6}$$

[1] The term 'reconstruction' is thus used for the signal itself, whereas the original meaning of this word deals with the process of reconstructing.

[2] Unless stated otherwise, all integrations and summations in this thesis extend from $-\infty$ to $+\infty$.

for the back focal plane, we find

$$\psi_o(\sqrt{\lambda f}u, \sqrt{\lambda f}v) = -ie^{ik2f} \iint \psi_i(\sqrt{\lambda f}x, \sqrt{\lambda f}y) e^{-i2\pi(ux+vy)} \mathrm{d}x\mathrm{d}y \; . \tag{2.7}$$

For convenience we define $\psi(x,y)$ and $\Psi(u,v)$ according to

$$\begin{aligned} \psi(x,y) &= \psi_i(\sqrt{\lambda f}x, \sqrt{\lambda f}y) \\ \Psi(u,v) &= ie^{-i2kf}\psi_o(\sqrt{\lambda f}u, \sqrt{\lambda f}v) \; , \end{aligned} \tag{2.8}$$

and observe that, due to (2.7), these two signals form a Fourier transform pair:

$$\begin{aligned} \Psi(u,v) &= \iint \psi(x,y) e^{-i2\pi(ux+vy)} \mathrm{d}x\mathrm{d}y \\ \psi(x,y) &= \iint \Psi(u,v) e^{\, i2\pi(xu+yv)} \mathrm{d}u\mathrm{d}v \; . \end{aligned} \tag{2.9}$$

Since the signal generated in the back focal plane is proportional to the Fourier transform of the input signal applied in the input plane, the holograms under consideration are called 'Fourier holograms'. In the remainder of this thesis we will use the proposed notation with normalized coordinates to denote signals in the input and output plane. Corresponding physical signals and physical dimensions can be found by 'denormalization' using the above equations (2.5) and (2.6). The normalization constant has a typical value $\sqrt{\lambda f} \approx 2.5$ mm for $\lambda = 500$ nm and $f = 10$ m.

The following examples indicate the relevance of Fourier holograms. In Figure 2.4 Fourier holograms are applied in order to generate optical interconnections between input and output plane (Jenkins et al., 1984). In the input plane a number of holograms is placed, individually illuminated with a laser source. Each hologram is designed to generate light-spots with given intensities in the output plane, where detectors are placed. When the laser sources are uncorrelated, every detector sums the received light from the uncorrelated laser sources weighted in intensity. In this way we have realized optical interconnections between the input and the output plane. This system realizes an optical matrix-vector multiplication $\boldsymbol{b} = C\boldsymbol{a}$, where

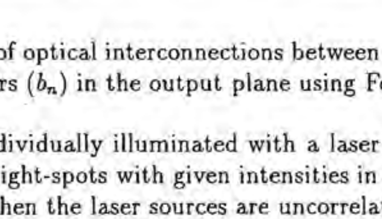


Figure 2.4: Realization of optical interconnections between laser sources (a_n) in the input plane and detectors (b_n) in the output plane using Fourier holograms.

$\boldsymbol{a} = (a_1, \ldots, a_N)^T$ denotes the intensities of the laser sources and $\boldsymbol{b} = (b_1, \ldots, b_N)^T$ denotes the intensities measured by the detectors. Due to the addition in intensity, the (real) matrix elements are restricted to $C_{nm} \geq 0$. It is also possible to let the laser sources originate from one main source. In that case the addition is coherent and the matrix elements C_{nm} can be complex. Optical interconnections are for example found in optical systems for signal processing, optical computers and in optical implementations of neural networks (Keller and Gmitro, 1993).

In a second example, 'spatial filtering', the optical setup of Figure 2.3 is extended with an additional lens. In the resulting $4f$-system, shown in Figure 2.5, the computer-generated hologram is placed in the intermediate plane and acts as the transfer function of a linear shift-invariant system. The underlying function from which it is derived by means of a Fourier transformation, plays the role of a 'point spread function' (impulse response). With this optical system real-time pattern recognition can be realized.

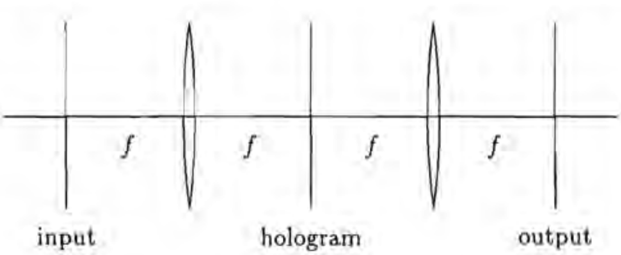


Figure 2.5: A Fourier hologram in a $4f$ system acts as the transfer function of a linear shift-invariant system.

In both applications the desired signal $\Psi(u, v)$ in the output plane is given. The accompanying input signal in the hologram plane is approximately realized by illuminating a binary hologram. While the incident wave in Figure 2.4 is planar, this is not the case in Figure 2.5. In the remainder of this thesis, however, we assume that the incident wave is planar and has unit amplitude. The design procedure from a desired object to a binary hologram consists of a number of steps as shown in Figure 2.6. Given $\Psi(u, v)$ in the output plane, henceforth designated as 'the object', we calculate the associated complex amplitude $\psi(x, y)$ by means of the inverse Fourier transformation. Since this transformation and the subsequent coding is performed with the aid of a digital computer, only a finite number of samples $\psi[n_1, n_2]$ is calculated. The consequences of this limitation are considered later on in this section. The next step in Figure 2.6 is to transform the calculated complex signal samples $\psi[n_1, n_2]$ into real signal samples $\phi[n_1, n_2]$ by means of the coding operator $\mathcal{C}_1$. This will be the subject of Section 2.3. The real signal samples are then quantized into binary samples $b[n_1, n_2]$ using a coding operator $\mathcal{C}_2$. The output

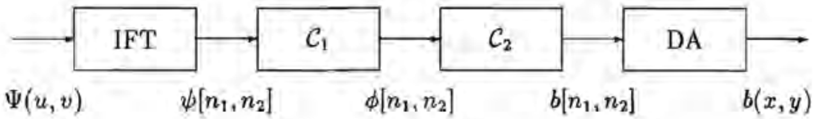


Figure 2.6: The operations involved in the calculation and realization of a computer-generated Fourier hologram.

device performs the digital-to-analog conversion of this binary signal to the physical hologram. Some aspects considering quantization and realization of the hologram transmittance are considered in Section 2.4. Finally the binary hologram is used in an optical system to generate the reconstruction.

We end this section with a discussion of the implications involved in the calculation of the complex transmittance of the hologram. The complex transmittance $\psi(x,y)$ is calculated by means of the inverse Fourier transformation

$$\psi(x,y) = \iint \Psi(u,v) e^{i2\pi(xu+yv)} \mathrm{d}u\mathrm{d}v \ . \tag{2.10}$$

To make physical realization possible, the hologram must have finite size. The complex transmittance $\psi(x,y)$ is therefore required to vanish outside the domain

$$\mathbb{H} = \left\{ (x,y) \in \mathbb{R}^2 \mid -\tfrac{1}{2}\Delta_x < x \leq \tfrac{1}{2}\Delta_x, -\tfrac{1}{2}\Delta_y < y \leq \tfrac{1}{2}\Delta_y \right\} \ , \tag{2.11}$$

corresponding to a hologram with finite size $\Delta_x \times \Delta_y$. This means that the object $\Psi(u,v)$ has to be bandlimited. As stated before, we calculate $\psi(x,y)$ in a finite number of sampling points in $\mathbb{H}$, which are assumed to be equidistant. Without considering numerical implications for the moment, this is achieved by evaluation of (2.10) for $x = \frac{2n_1+1}{2N_1}\Delta_x$ and $y = \frac{2n_2+1}{2N_2}\Delta_y$ with $n_1 = -\frac{1}{2}N_1, \ldots, \frac{1}{2}N_1 - 1$ and $n_2 = -\frac{1}{2}N_2, \ldots, \frac{1}{2}N_2 - 1$. (This implies that N_1 and N_2 are even.) The distance between neighbouring sampling points (the 'sampling distance') equals $X = \Delta_x/N_1$ for the x-coordinate and $Y = \Delta_y/N_2$ for the y-coordinate. The optimal choice for the sampling distance is the result of a compromise between computational effort and loss of information. To settle this problem we consider the Fourier transform of the discrete signal $\psi[n_1,n_2]$, defined according to

$$\Psi_d(\theta_1,\theta_2) = \sum_{n_1 \in \langle N_1 \rangle} \sum_{n_2 \in \langle N_2 \rangle} \psi[n_1,n_2] e^{-i2\pi(\theta_1 n_1 + \theta_2 n_2)} \ . \tag{2.12}$$

The summation is over $n_1 = -\frac{1}{2}N_1, \ldots, \frac{1}{2}N_1 - 1$ and $n_2 = -\frac{1}{2}N_2, \ldots, \frac{1}{2}N_2 - 1$, denoted by $\langle N_1 \rangle$ and $\langle N_2 \rangle$. The subscript d is introduced to avoid confusion between the Fourier transform $\Psi_d(\theta_1,\theta_2)$ of a discrete signal and the Fourier transform $\Psi(u,v)$ of a continuous signal. The Fourier transform $\Psi_d(\theta_1,\theta_2)$ is periodic according to $\Psi_d(\theta_1,\theta_2) = \Psi_d(\theta_1 + m_1, \theta_2 + m_2)$, with $m_1, m_2 \in \mathbb{Z}$. We take the fundamental interval according to $-\frac{1}{2} < \theta_1 \leq \frac{1}{2}$ and $-\frac{1}{2} < \theta_2 \leq \frac{1}{2}$, each denoted by $\langle 1 \rangle$. The discrete signal $\psi[n_1, n_2]$ arises from sampling $\psi(x, y)$, according to

$$\psi[n_1, n_2] = \psi(n_1 X + \tfrac{1}{2} X, n_2 Y + \tfrac{1}{2} Y) . \tag{2.13}$$

In Appendix A it is shown that the Fourier transforms $\Psi_d(\theta_1, \theta_2)$ and $\Psi(u, v)$ are then related through

$$\Psi_d(\theta_1, \theta_2) = \frac{1}{XY} e^{i\pi(\theta_1 + \theta_2)} \sum_{m_1} \sum_{m_2} (-1)^{m_1} (-1)^{m_2} \Psi\left(\frac{\theta_1 + m_1}{X}, \frac{\theta_2 + m_2}{Y}\right) . \tag{2.14}$$

The frequencies $\theta_1 = Xu$ and $\theta_2 = Yv$ are normalized with respect to the sampling frequencies X^{-1} and Y^{-1}. According to (2.14) the Fourier transform of $\psi[n_1, n_2]$ consists of the original object $(m_1 = 0, m_2 = 0)$ plus shifted replicas of the original object. The additional factors $e^{i\pi(\theta_1 + \theta_2)}$, $(-1)^{m_1}$ and $(-1)^{m_2}$ arise from our choice of the sampling points.

The bandlimited object $\Psi(u, v)$ has infinite support and therefore overlap of the replicas (aliasing) will always occur when the transmittance function $\psi(x, y)$ is sampled. As a consequence, reconstruction of the original object from the samples $\psi[n_1, n_2]$ is impossible. However, we assume that the desired object $\Psi(u, v)$ has its main contribution within a region $|u| \leq \frac{1}{2}\Delta_u$, $|v| \leq \frac{1}{2}\Delta_v$ and decreases rapidly for larger spatial frequencies. In this case aliasing can be made small by choosing the sampling distances $X \ll \Delta_u^{-1}$ and $Y \ll \Delta_v^{-1}$. For our computer-generated holograms we have used 4 times oversampling in both directions: $X = \frac{1}{4}\Delta_u^{-1}$ and $Y = \frac{1}{4}\Delta_v^{-1}$.

Often the desired object $\Psi(u, v)$ is not known analytically but in a sampled form. Due to the band-limitation of $\Psi(u, v)$ such a description preserves the entire information provided that the sampling raster is dense enough. In concrete terms, sampling of $\Psi(u, v)$ at $u = k_1 U$, $v = k_2 V$ with $k_1, k_2 \in \mathbb{Z}$ is permitted if $U \leq \Delta_x^{-1}$ and $V \leq \Delta_y^{-1}$. This also means that we should be able to calculate the samples $\psi[n_1, n_2]$ from the sampled values $\Psi(k_1 U, k_2 V)$. This is shown in the following way. We introduce the discrete signal

$$\Psi[k_1, k_2] = \Psi_d(k_1/N_1, k_2/N_2) . \tag{2.15}$$

This signal is the result of taking $N_1 \times N_2$ samples in each period of $\Psi_d(\theta_1, \theta_2)$. In terms of u and v this means that we have used the maximal sampling distance $U = \Delta_x^{-1}$ and $V = \Delta_y^{-1}$. Moreover, we define $\hat{\psi}[n_1, n_2]$ according to

$$\hat{\psi}[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{k_1 \in \langle N_1 \rangle} \sum_{k_2 \in \langle N_2 \rangle} \Psi[k_1, k_2] e^{i2\pi\left(\frac{n_1 k_1}{N_1} + \frac{n_2 k_2}{N_2}\right)} \tag{2.16}$$

According to (2.16) we find that $\hat{\psi}[n_1, n_2]$ is also periodic with period length N_1 and N_2 for both coordinates. The inverse formula of (2.16) reads

$$\Psi[k_1, k_2] = \sum_{n_1 \in \langle N_1 \rangle} \sum_{n_2 \in \langle N_2 \rangle} \hat{\psi}[n_1, n_2] e^{-i2\pi \left(\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2} \right)} , \tag{2.17}$$

where the summation is again over one period. The discrete signals $\Psi[k_1, k_2]$ and $\hat{\psi}[n_1, n_2]$ form a discrete Fourier transform pair, and (2.17) and (2.16) are known as the discrete Fourier transformation (DFT) and the inverse discrete Fourier transformation (IDFT), respectively.

In a straightforward way it can be shown that $\hat{\psi}[n_1, n_2]$ and $\psi[n_1, n_2]$ are related according to

$$\hat{\psi}[n_1, n_2] = \sum_{m_1} \sum_{m_2} \psi[n_1 - m_1 N_1, n_2 - m_2 N_2] . \tag{2.18}$$

This equation states that due to the sampling in the reconstruction plane a periodic repetition takes place in the hologram plane, possibly leading to aliasing. In our case, however, $\psi[n_1, n_2]$ is a finite two-dimensional sequence consisting of $N_1 \times N_2$ samples. Thus overlap of the shifted replicas does not occur and we have $\psi[n_1, n_2] = \hat{\psi}[n_1, n_2]$ for $n_1 \in \langle N_1 \rangle$ and $n_2 \in \langle N_2 \rangle$.

Given the samples $\Psi(k_1 U, k_2 V)$ we use (2.14) in order to calculate $\Psi[k_1, k_2] = \Psi_d(k_1 U, k_2 V)$. By means of the inverse discrete Fourier transformation (2.16) we then find the periodic sequence $\hat{\psi}[n_1, n_2]$ and therefore $\psi[n_1, n_2]$. Due to the (non-avoidable) aliasing it is impossible to reconstruct the original samples $\Psi(k_1 U, k_2 V)$. For this reason we use the discrete Fourier transform pair $\hat{\psi}[n_1, n_2] \leftrightarrow \Psi[k_1, k_2]$ for the calculation of Fourier holograms.

We note that the assumption that the object is concentrated within a window of finite dimensions $\Delta_u \times \Delta_v$ in the reconstruction plane is in contradiction with the requirement of bandlimitation due to the finite size $\Delta_x \times \Delta_y$ of the hologram. This is the reason for the appearance of fluctuations in the reconstruction plane, known as speckle. In some applications (e.g. display applications) the intensity of the object is of interest only. We then have the freedom to multiply the intensity of the object by an arbitrary phase distribution. This gives rise to a better use of the limited dynamical range of the photographic material of the hologram. In general, the resulting (complex) transmittance, however, is not bandlimited, and needs a larger hologram size than available. If the original hologram size is chosen, part of the required transmittance is replaced by zero. This leads to speckle in the reconstruction plane. In order to avoid this effect a bandlimited phase distribution can be applied (Wyrowski and Bryngdahl, 1988). With discrete objects (cf. Figure 2.4), the speckle is known to appear between the sample points and does not affect the object samples. Since a periodic repetition of the hologram in the input plane leads to sampling of the reconstructed signal in the output plane, this provides another means to attenuate the speckle (Lesem et al., 1968). The phase can then be chosen freely. With the exception of Chapter 3 we have assumed the object to be discrete and we have multiplied the object's intensity by a random phase distribution.

2.3 Determination of the real transmittance

Under the assumption that the hologram placed in the input plane of the Fourier transformation lens is illuminated by a unit-amplitude plane wave, the hologram transmittance function has to be the inverse Fourier transform of the object function $\Psi(u,v)$, yielding a complex-valued function $\psi(x,y)$. For amplitude holograms the complex $\psi(x,y)$ has to be transformed into a positive real signal $\phi(x,y)$ that can be directly realized as the transmittance function of a transparency. Therefore a (first) coding step $\mathcal{C}_1$ is necessary, cf. Figure 2.6. Although the coding operator $\mathcal{C}_1$ applies to the sampled signal $\psi[n_1,n_2]$, we choose a continuous treatment to elucidate the basic transition from complex to real transparence functions. The discrete case is treated further down. We remark that by varying the refractive index or the thickness of the transparent material we achieve phase modulation. For a phase hologram $\psi(x,y)$ has to be transformed into a complex signal with unit amplitude. For both kinds of hologram amplitude and phase information of the calculated complex transmittance are encoded in the transparency. This justifies the name (computer-generated) hologram.

Here we confine ourselves to the encoding of amplitude holograms. First the complex amplitude $\psi(x,y)$ is multiplied by a complex 'carrier' $e^{i2\pi(u_o x+v_o y)}$. In a next step the *real part* of this modulated complex signal is taken. This results in a real transmission function $\mathrm{Re}\left[\psi(x,y)e^{i2\pi(u_o x+v_o y)}\right]$. (In some sense this corresponds to the procedure followed in optical holography where a complex carrier was *added*, cf. (2.2)). Finally a positive real bias function $\beta(x,y)$ is added to obtain a positive transmission function. The desired $\phi(x,y)$ thus assumes the form

$$\phi(x,y) = |\psi(x,y)|\cos[2\pi(u_o x + v_o y) + \arg\psi(x,y)] + \beta(x,y)\,. \tag{2.19}$$

An obvious choice is a constant value [3] $\beta(x,y) = \beta_o\mathrm{rect}(x/\Delta_x, y/\Delta_y)$ such that $\min\phi(x,y) = 0$, but alternatives can be considered (Burch, 1967). (With the non-constant bias $\beta(x,y) = 1 + |\psi(x,y)|^2$ for $(x,y) \in \mathbb{H}$ we can exactly simulate the recording procedure (2.2) used in optical holography.) In the remainder of this thesis we further assume that the transmittance $\phi(x,y)$ of the amplitude hologram satisfies the condition $0 \leq \phi(x,y) \leq 1$, which is imposed by the physical requirement

[3] The two-dimensional rectangular function is defined as $\mathrm{rect}(x,y) = \mathrm{rect}(x)\mathrm{rect}(y)$ with

$$\mathrm{rect}(x) = \begin{cases} 1 & |x| < 1/2 \\ 1/2 & |x| = 1/2 \\ 0 & |x| > 1/2 \end{cases}\,. \tag{2.20}$$

of passivity. This condition amounts to an amplitude bound of the reconstruction in the output plane.

The Fourier transform of $\phi(x,y)$ with a constant bias $\beta(x,y) = \beta_o$ for $(x,y) \in \mathbb{H}$ reads

$$\Phi(u,v) = \tfrac{1}{2}\Psi(u-u_o, v-v_o) + \tfrac{1}{2}\Psi^*(-u-u_o, -v-v_o) + \beta_o\Delta_x\Delta_y \mathrm{sinc}(\Delta_x u, \Delta_y v) \quad (2.21)$$

Thus, due to the modulation on a complex carrier the object $\Psi(u,v)$ is (spatially) shifted over (u_o, v_o). In addition, a twin object $\Psi^*(-u-u_o, -v-v_o)$ belonging to the conjugate input signal appears. This is the result of taking the real part of the complex transmittance. The constant bias finally gives rise to a 'sinc-peak' [4] in the origin of the output plane. Under our (previous) assumption that the (unmodulated) object is concentrated near the origin in the output plane, for spatial frequencies $|u| \leq \frac{1}{2}\Delta_u$, $|v| \leq \frac{1}{2}\Delta_v$, we can avoid overlap of the original object with its twin image by choosing $u_o > \frac{1}{2}\Delta_u$ and/or $v_o > \frac{1}{2}\Delta_v$. The two-dimensional sinc-function in (2.21) has small dimensions (size $\Delta_x^{-1} \times \Delta_y^{-1}$) and, as such, is of minor concern. A possible configuration in the output plane satisfying this condition is shown in Figure 2.7. We refer to the rectangular window $\mathbb{F}$ defined according to

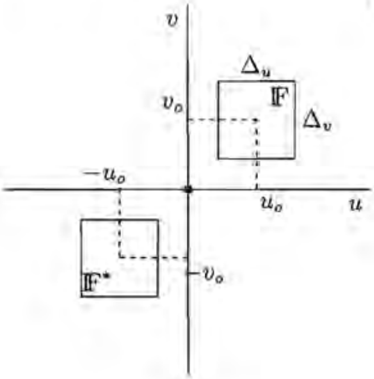


Figure 2.7: A possible configuration in the reconstruction plane where overlap of the object window $\mathbb{F}$ and the twin object window $\mathbb{F}^*$ is avoided.

$$\mathbb{F} = \{(u,v) \in \mathbb{R}^2 \mid -\tfrac{1}{2}\Delta_u < u - u_o \leq \tfrac{1}{2}\Delta_u, -\tfrac{1}{2}\Delta_v < v - v_o \leq \tfrac{1}{2}\Delta_v\} \quad (2.22)$$

as the object window, since in this window the (shifted) object appears. The twin object occurs in a twin object window denoted by $\mathbb{F}^*$.

[4] The two-dimensional sinc-function is defined as $\mathrm{sinc}(x,y) = \mathrm{sinc}(x)\mathrm{sinc}(y)$, with $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$; $\mathrm{sinc}(u,v)$ is the two-dimensional Fourier transform of $\mathrm{rect}(x,y)$.

In the above we have considered the transformation of a *continuous* complex signal $\psi(x,y)$ into a real signal $\phi(x,y)$, while the coding operator C_1 transforms a *discrete* complex signal $\psi[n_1,n_2]$ into a real signal $\phi[n_1,n_2]$. In analogy with continuous signals we find

$$\phi[n_1,n_2] = \mathrm{Re}\left[\psi[n_1,n_2]e^{i2\pi(\theta_{1o}n_1+\theta_{2o}n_2)} + \beta_o\right] , \tag{2.23}$$

with a Fourier transform

$$\Phi_d(\theta_1,\theta_2) = \tfrac{1}{2}\Psi_d(\theta_1-\theta_{1o},\theta_2-\theta_{2o}) + \tfrac{1}{2}\Psi_d^*(-\theta_1-\theta_{1o},\theta_2-\theta_{2o}) + \beta_o\frac{N_1N_2\mathrm{sinc}(N_1\theta_1,N_2\theta_2)}{\mathrm{sinc}(\theta_1,\theta_2)}e^{-i\pi(\theta_1+\theta_2)} . \tag{2.24}$$

Again we assume the discrete signal to be scaled according to $0 \le \phi[n_1,n_2] \le 1$. The Fourier transform of the signal $\phi[n_1,n_2]$ is periodic and consists of two-dimensional repetitions of the configuration shown in Figure 2.7. With sampling distances X and Y the repetition in the frequency variables u and v is over a distance X^{-1} and Y^{-1}, respectively. Due to the oversampling ($X^{-1} = 4\Delta_u$, $Y^{-1} = 4\Delta_v$) overlap of the windows does not occur.

2.4 Quantization and realization of the transmittance

To realize the hologram transmittance as a transparency we use an output device with certain characteristics. This means that we must adapt our signal to the properties of the output device. Most of the used devices, such as a plotter, laserprinter, laserwriter or e-beam writer generate binary output. The continuously-valued hologram samples have thus to be transformed into binary-valued samples.

In the early days holograms were drawn with a pen plotter, followed by a photographic reduction step to scale the hologram to a proper size. The smallest achievable dot size for plotters is large compared to the pen's elementary steps. When the raster points of the plotter are addressed individually, overlap of the dots can occur. Since the nonlinear effects of such an event are difficult to analyze, a *cell-oriented* (Dallas, 1980) approach has been proposed (Brown and Lohmann, 1966). To this end the hologram is divided in cells and within each cell the (sampled) signal value is represented by a transparent inner cell. The area of the inner cell is modulated according to the signal amplitude, while the signal phase is coded in the position of the inner cell (Figure 2.8). Both position (phase) and area (amplitude) can be realized with high accuracy on a plotter device.

Later on output devices were developed that made individual addressing of raster points possible. The laser writer with a raster period of 10 μm and a dot with size 10 μm was constructed for writing hologram distributions directly in holographic film. Today the e-beam lithograph is also applied to write holograms. This device

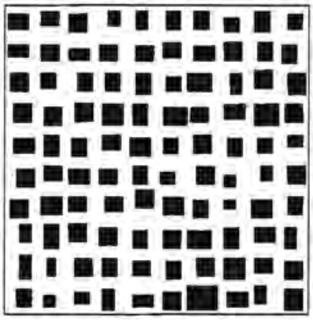


Figure 2.8: Cell-oriented coded computer-generated hologram. The transparent inner cells are shown black.

has been developed for the production of masks for the integrated circuit fabrication and can work in the sub-micron region. With e-beam lithography the number of addressable points is very large so that holograms with a large space-bandwidth product can be obtained. For such devices new kinds of algorithms with a *point-oriented* approach were developed which are also the subject of the present thesis.

The signal generated by the output device consists of shifted versions of a given basic pulse. We therefore have to try to encode the information of the *real* input signal $\phi(x, y)$ in a varying pulse density. All techniques studied in this thesis are based on this pulse-density modulation principle. In general the positioning of the pulses is restricted to certain places only. We then speak about discrete pulse-density modulation. In Chapter 3, however, we leave this restriction (except for overlap of the pulses) and consider the case of continuous pulse-density modulation. The output signal is then modeled according to

$$b(x,y) = \sum_{m=1}^{M} s(x - x_m, y - y_m) \; , \tag{2.25}$$

where the basic pulse $s(x, y)$ assumes the value 1 inside a small region $(x, y) \in \sigma$ and the value 0 outside σ. Given the real signal $\phi(x, y)$, or in sampled form $\phi[n_1, n_2]$, the pulse positions (x_m, y_m) have to be calculated.

With continuous pulse-density modulation we have to calculate *where* each pulse is desired. For discrete pulse-density modulation, which is considered in Chapter 4 and Chapter 5, this problem is reduced to deciding for each raster point *whether* a pulse is placed or not. We assume that the allowed pulse positions define a rectangular raster with periods X and Y for the x- and y-directions, respectively, and model the output signal according to

$$b(x,y) = \sum_{n_1 \in \langle N_1 \rangle} \sum_{n_2 \in \langle N_2 \rangle} b[n_1, n_2] \mathrm{rect}(\frac{x}{X} - n_1 - \tfrac{1}{2}, \frac{y}{Y} - n_2 - \tfrac{1}{2}) \; . \tag{2.26}$$

The elementary pulse is the rectangular pulse $\mathrm{rect}(x/X, y/Y)$ with pulsewidth $X \times Y$. Since the raster distance equals the pulse-width in both dimensions overlap does not occur. The binary hologram consists of $N_1 \times N_2$ cells with a binary transmittance $b[n_1, n_2]$ and has a total size $\Delta_x = N_1 X$, $\Delta_y = N_2 Y$. The binary number $b[n_1, n_2] \in \{0, 1\}$ denotes whether the cell on raster point (n_1, n_2) is opaque and thus blocks the incident light ($b = 0$), or the cell is transparent and thus transmits the incident light ($b = 1$).

With discrete pulse-density modulation the coding operator $\mathcal{C}_2$ transforms (quantizes) the real samples $\phi[n_1, n_2]$, continuous in amplitude, into the binary samples $b[n_1, n_2]$. In order to analyze the properties of this nonlinear mapping we interpret $\mathcal{C}_2$ as the addition of coding noise $e[n_1, n_2]$, i.e. $b[n_1, n_2] = \phi[n_1, n_2] + e[n_1, n_2]$. Due to the linearity of the Fourier transformation we find that $B_d(\theta_1, \theta_2)$ (the Fourier transform of $b[n_1, n_2]$) consists of the original object (and twin image) $\Phi_d(\theta_1, \theta_2)$ and a disturbance $E_d(\theta_1, \theta_2)$, the Fourier transform of the coding noise. For convenience we call the last contribution simply coding noise. It will be clear from the context whether we are referring to coding noise in the hologram plane or to coding noise in the reconstruction plane. In order to disturb the original object as little as possible, $\Phi_d(\theta_1, \theta_2)$ and $E_d(\theta_1, \theta_2)$ should be spatially separated. Due to the oversampling in the hologram plane this is possible to some degree. Although the mapping is a purely deterministic operation, a statistical approach will appear advantageous in the design of the coding operator $\mathcal{C}_2$ that achieves this goal.

Some of the algorithms for the coding operator $\mathcal{C}_2$ originate from the field of image processing, where similar problems exist. For instance, the grey-tones in images printed in newspapers or by fax-apparatus are simulated with binary dot structures. For this technique, called halftoning, numerous algorithms such as dithering and error-diffusion have been developed. In particular the error-diffusion halftoning algorithm, considered in Chapter 4, has been applied with succes to calculate binary holograms. As computers became faster, more complex coding algorithms were developed. These algorithms, such as direct binary search (Seldowitz et al., 1987), the iterative Fourier-transform algorithm (Wyrowski and Bryngdahl, 1989), Hopfield's neural network (Just and Ling, 1991) and the Boltzmann machine, are also point-oriented but work in an iterative way. Hopfield's neural network and the Boltzmann machine are considered in Chapter 5 of this thesis.

In Figure 2.9a we have shown an example of a binary hologram calculated with a Hopfield neural network. In Figure 2.9b the fundamental interval of the (calculated) reconstruction of the binary hologram is shown. (For the results in this thesis we show the fundamental interval $-\frac{1}{2}X^{-1} < u \leq \frac{1}{2}X^{-1}$ and $-\frac{1}{2}Y^{-1} < v \leq \frac{1}{2}Y^{-1}$ only.) The coding noise is almost completely located outside the object window and the twin object window, where we can clearly observe the original object and the twin object. The (large) sinc-peak in the reconstruction of the hologram is not shown.

Finally, the output device transforms the binary samples $b[n_1, n_2]$ into a contin-

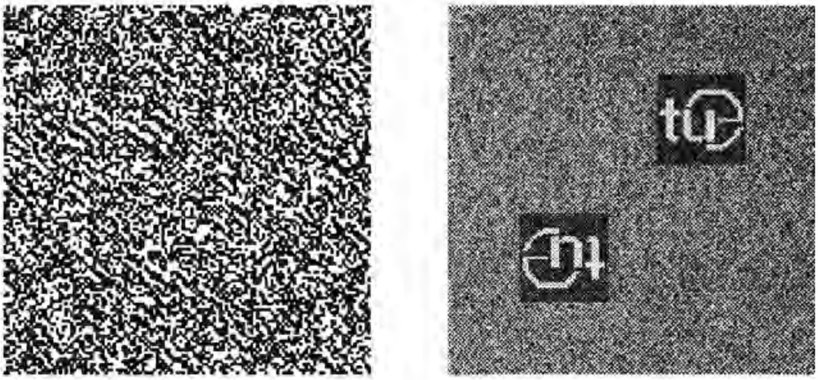


Figure 2.9: a. Example of a binary point-oriented Fourier hologram. b. The calculated reconstruction of the binary hologram. The sinc-peak in the reconstruction is not shown.

uous signal $b(x,y)$, i.e. (2.26) with a Fourier transform

$$B(u,v) = XY\mathrm{sinc}(Xu,Yv)e^{-i\pi(Xu+Yv)}B_d(Xu,Yv)\,. \tag{2.27}$$

Using (2.14) we find for the reconstruction of the hologram

$$B(u,v) = \mathrm{sinc}(Xu,Yv)\sum_{m_1}\sum_{m_2}(-1)^{m_1}(-1)^{m_2}\Phi(u+\frac{m_1}{X},v+\frac{m_2}{Y}) + \\ XY\mathrm{sinc}(Xu,Vy)e^{-i\pi(Xu+Vy)}E_d(Xu,Vy)\,. \tag{2.28}$$

We conclude that the binary hologram reconstructs shifted versions of the original object (and the twin object). Due to the rectangular interpolation of $b[n_1,n_2]$ the higher-order repetitions in the spectrum, which are not shown in Figure 2.9, are attenuated by the factor $\mathrm{sinc}(Xu,Yv)$.

2.5 Quality measures

Although the coding noise is shifted outside the object window, some noise will disturb the original object. This distortion is expressed in a signal-to-noise ratio (SNR). In some applications the binary hologram has to generate a desired amplitude and phase distribution in the reconstruction. In that case the signal-to-noise ratio has to express both amplitude and phase errors and therefore we define

$$\mathrm{SNR} = \frac{\iint_{\mathbf{F}}|\Phi(u,v)|^2\mathrm{d}u\mathrm{d}v}{\iint_{\mathbf{F}}|B(u,v)-\Phi(u,v)|^2\mathrm{d}u\mathrm{d}v}\,. \tag{2.29}$$

In other applications we are interested only in the amplitude (intensity) of the object. An appropriate definition of the signal-to-noise ratio then reads

$$\mathrm{SNR} = \frac{\iint_{\mathbb{F}} |\Phi(u,v)|^2 \mathrm{d}u\mathrm{d}v}{\iint_{\mathbb{F}} \left(|B(u,v)| - |\Phi(u,v)|\right)^2 \mathrm{d}u\mathrm{d}v} . \tag{2.30}$$

We remark that a definition for the signal-to-noise ratio different from (2.29) is used in literature. The coding noise is then taken $e[n_1, n_2] = b[n_1, n_2] - \lambda\phi[n_1, n_2]$ rather than $e[n_1, n_2] = b[n_1, n_2] - \phi[n_1, n_2]$. The (complex) scaling constant λ is taken such that the coding noise $E(u, v)$ is orthogonal to $\Phi(u, v)$, according to

$$\iint_{\mathbb{F}} E(u,v)\Phi^*(u,v)\mathrm{d}u\mathrm{d}v = 0 . \tag{2.31}$$

The accompanying value for λ reads

$$\lambda = \frac{\iint_{\mathbb{F}} B(u,v)\Phi^*(u,v)\mathrm{d}u\mathrm{d}v}{\iint_{\mathbb{F}} |\Phi(u,v)|^2 \mathrm{d}u\mathrm{d}v} . \tag{2.32}$$

One could say that part of the coding noise contributes to the object. The alternative signal-to-noise ratio is then found by replacing $\Phi(u, v)$ by $\lambda\Phi(u, v)$ in (2.29). Also, (2.30) is sometimes used with $|\Phi(u, v)|$ replaced by $\lambda|\Phi(u, v)|$. For the real constant λ we then find

$$\lambda = \frac{\iint_{\mathbb{F}} |B(u,v)||\Phi(u,v)|\mathrm{d}u\mathrm{d}v}{\iint_{\mathbb{F}} |\Phi(u,v)|^2 \mathrm{d}u\mathrm{d}v} . \tag{2.33}$$

In this thesis, however, we require that λ equals 1.

Maximizing the signal-to-noise ratio (2.29) of the amplitude-phase optimization (AP) problem also gives rise to a large (but suboptimal) signal-to-noise ratio (2.30) for the amplitude-only optimization (AO) problem. (The other way around is in general not true.) Since (2.29) is mathematically more tractable than (2.30) it will form the starting point for further analysis.

The coding noise which is generated outside the object windows represents part of the light that is not used to form the desired object. In order to quantify the power contained in the object we introduce the hologram efficiency η. This efficiency is the product of the transmission efficiency η_t and the diffraction efficiency η_d. With binary amplitude holograms part of the incident light power P_i is blocked by the opaque cells. The transmission efficiency, defined according to

$$\eta_t = \frac{P_o}{P_i} = \frac{1}{\Delta_x\Delta_y} \iint_{\mathbb{H}} |b(x,y)|^2 \mathrm{d}x\mathrm{d}y , \tag{2.34}$$

expresses the percentage of the light power transmitted by the binary hologram. Only part of the transmitted power P_o is used to generate the object. This is expressed by the diffraction efficiency

$$\eta_d = \frac{\iint_{\mathbb{F}} |B(u,v)|^2 \mathrm{d}u\mathrm{d}v}{\iint |B(u,v)|^2 \mathrm{d}u\mathrm{d}v} , \tag{2.35}$$

where the integration in the denominator extends over the entire uv-plane. Due to Parseval's theorem the total transmitted power equals the total power in the reconstruction plane and therefore we find for the hologram efficiency

$$\eta = \eta_t \eta_d = \frac{1}{\Delta_x \Delta_y} \iint_{\mathbb{F}} |B(u,v)|^2 \mathrm{d}u\mathrm{d}v \ . \tag{2.36}$$

An estimate for the upper limit of the efficiency can easily be given (Wyrowski, 1990). Under the assumption that one half of the cells in the binary amplitude hologram are opaque, only half of the incident light power is transmitted. One half of the transmitted noise power goes to the dc-peak, the remaining part is divided among the two windows in the best case. The theoretical maximum of the efficiency thus equals $\eta = 0.125$.

The efficiency of an amplitude hologram is rather small due to the blocking of the light by the opaque cells. A bleaching procedure transforms the binary amplitude hologram into a binary phase hologram, which does not suffer from this problem. An alternative way to generate a phase hologram is to etch a calculated phase distribution in a glass substrate. This makes it possible to realize quantized phase holograms with more than two levels. We remark that by means of a modification, the techniques considered in this thesis can be adjusted to the calculation of multilevel phase holograms (see for example (Weissbach et al., 1989)). With multilevel phase holograms (number of levels ≥ 3) the transformation of the complex signal $\psi(x,y)$ to a real signal $\phi(x,y)$ is not necessary. As a result, the twin object does not exist.

Chapter 3

Continuous pulse-density modulation

3.1 Introduction

In this chapter we consider the approximation of a given signal by a signal consisting of a number of nonoverlapping identical pulses, whose positions have to be determined. We assume that these pulses can be placed with infinite precision and for this reason we call this kind of approximation continuous pulse-density modulation.

Although actual output devices are not able to position pulses continuously in space, we still feel that this approach makes sense. Using a continuous-space approach we hope to acquire a better understanding of the problem and use the obtained insights for the discrete case, where pulses are restricted to fixed positions.

The starting point for continuous pulse-density modulation is a continuous real signal $\phi(x,y)$, with $0 \leq \phi(x,y) \leq 1$. According to the previous chapter, $\phi(x,y)$ represents the transmittance function of an amplitude hologram. When the input signal is not known analytically but is given in sampled form, a continuous signal is generated by interpolation. The approximating signal

$$b(x,y) = \sum_{m=1}^{M} s(x - x_m, y - y_m) \approx \phi(x,y) \tag{3.1}$$

is a set of shifted replicas of the elementary two-dimensional pulse $s(x,y)$, which is nonzero only in a region $(x,y) \in \sigma$. To avoid overlap of the pulses, the distance between neighbouring pulses has a lower bound. The set of pulses can be considered as a set of two-dimensional Dirac-functions convolved with the elementary pulse, according to

$$b(x,y) = s(x,y) * \sum_{m=1}^{M} \delta(x - x_m, y - y_m) . \tag{3.2}$$

Introducing the Fourier transform $S(u,v)$ of the elementary pulse $s(x,y)$ we find for the Fourier transform of $b(x,y)$:

$$B(u,v) = S(u,v) \sum_{m=1}^{M} e^{-i2\pi(ux_m+vy_m)} . \tag{3.3}$$

When we consider both amplitude and phase errors, the quality of the approximation is expressed as the signal-to-noise ratio (2.29) defined in Chapter 2. In order to maximize the signal-to-noise ratio we try to minimize the denominator

$$\iint_{\mathbb{F}} |B(u,v) - \Phi(u,v)|^2 \mathrm{d}u\mathrm{d}v . \tag{3.4}$$

Obviously, only spatial frequencies within the object window $\mathbb{F}$ are involved in the error. Stated otherwise, (3.4) represents the noise power in the object window, due to the pulse-density approximation. According to the previous chapter, $\mathbb{F}$ is centered around (u_o, v_o) and has size $\Delta_u \times \Delta_v$. Within this object window the desired object is located. For convenience we define the frequency weighting function

$$|A(u,v)|^2 = \begin{cases} 1 & (u,v) \in \mathbb{F} \cup \mathbb{F}^* \\ 0 & \text{elsewhere} \end{cases} , \tag{3.5}$$

and write for the noise power

$$P = \iint |A(u,v)|^2 |B(u,v) - \Phi(u,v)|^2 \mathrm{d}u\mathrm{d}v . \tag{3.6}$$

We remark that both the object window and the twin object window are taken into account in (3.6). This does not imply any restriction since the real signals $\phi(x,y)$ and $b(x,y)$ have a symmetric Fourier transform. In order to express the noise power P in the spatial domain we apply Parseval's theorem

$$\iint |G(u,v)|^2 \mathrm{d}u\mathrm{d}v = \iint |g(x,y)|^2 \mathrm{d}x\mathrm{d}y \tag{3.7}$$

to (3.6). We then find that the noise power is expressed in the spatial domain according to

$$P = \iint |a(x,y) * [b(x,y) - \phi(x,y)]|^2 \mathrm{d}x\mathrm{d}y , \tag{3.8}$$

with $a(x,y)$ the inverse Fourier transform of $A(u,v)$.

Our goal is to determine the positions (x_m, y_m) of the individual pulses contained in $b(x,y)$ such that (3.8) is minimized. In order to derive an expression for the explicit dependence of the noise power on the pulse positions, we introduce the filtered input signal $\tilde{\phi}(x,y) = a(x,y) * \phi(x,y)$ and the filtered pulse-density signal $\tilde{b}(x,y) = a(x,y) * b(x,y) = \sum_{m=1}^{M} \tilde{s}(x - x_m, y - y_m)$ with $\tilde{s}(x,y) = a(x,y) * s(x,y)$. This allows us to write for (3.8)

$$P = \iint \left[\tilde{b}^2(x,y) - 2\tilde{b}(x,y)\tilde{\phi}(x,y) + \tilde{\phi}^2(x,y)\right] \mathrm{d}x\mathrm{d}y . \tag{3.9}$$

In a straightforward way the following identities can be derived:

$$\iint \bar{b}^2(x,y)\mathrm{d}x\mathrm{d}y = \sum_{m=1}^{M}\sum_{n=1}^{M} \hat{s}(x_m - x_n, y_m - y_n) \tag{3.10}$$

and

$$\iint \bar{b}(x,y)\bar{\phi}(x,y)\mathrm{d}x\mathrm{d}y = \sum_{m=1}^{M} \hat{\phi}(x_m, y_m) \ , \tag{3.11}$$

with $\hat{s} = \bar{s} \star \bar{s} = s * \hat{a} * s$ and $\hat{\phi} = \bar{s} \star \bar{\phi} = s * \hat{a} * \phi$. The autocorrelation function

$$\hat{a}(x,y) = a(x,y) \star a(x,y) = \iint a(\xi - x, \eta - y)a(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \tag{3.12}$$

is the inverse Fourier transform of the frequency weighting function $|A(u,v)|^2$. Using these simplifications we find for the noise power

$$P = \sum_{m=1}^{M}\sum_{n=1}^{M} \hat{s}(x_m - x_n, y_m - y_n) - 2\sum_{m=1}^{M} \hat{\phi}(x_m, y_m) + \iint \bar{\phi}^2(x,y)\mathrm{d}x\mathrm{d}y \ . \tag{3.13}$$

The first term in (3.13) represents the mutual interaction of the pulses, while the influence of the (external) input signal is represented by the second term. The last term is independent of the pulse positions and can be omitted in the minimization of P. The necessary conditions for a pulse distribution to be optimal are found by differentiation of (3.13) with respect to the pulse positions and setting the derivatives equal to zero:

$$\text{grad } P = \mathbf{o} \ . \tag{3.14}$$

A number of gradient-search algorithms can be constructed based on the optimality conditions (3.14). Using such an algorithm will in general result in a pulse distribution with a local minimal noise power.

We remark that minimization of the energy in the present form does not guarantee that the pulses will not overlap. Moreover, it is possible (and probable) that the pulses are shifted outside the domain $\mathbb{H}$. Both problems are avoided by the introduction of penalty functions that become large whenever the pulses approach each other or the boundary.

Due to the interaction between the pulses, an iterative procedure is needed in order to determine the (locally) optimal pulse distribution. In the next section, we consider an approach for one-dimensional signals where this interaction is left out. In this way we are able to determine each pulse position separately. Also in Section 3.2 we consider similar (non-iterative) methods to determine pulse positions in blocks (pairs, triples). In Section 3.3 we discuss how this approach, which is based on numerical integration, is applied in two dimensions. Next, in Section 3.4 we discuss some clustering techniques that are related to pulse-density modulation. Finally, we consider in Section 3.5 how continuous pulse-density modulation can be adapted in order to obtain a pulse train with pulses at fixed positions.

3.2 One-dimensional continuous pulse-density modulation

3.2.1 Introduction

In this section we consider continuous pulse-density modulation for one-dimensional signals. A positive real signal $0 < \phi(x) \leq 1$ with a Fourier transform $\Phi(u)$ is defined on the interval $-\frac{1}{2}\Delta_x \leq x \leq \frac{1}{2}\Delta_x$. We assume that $\phi(x) = 0$ outside this interval. In the one-dimensional case the approximating signal is a pulse train

$$b(x) = \sum_{m=1}^{M} s(x - x_m) \, , \tag{3.15}$$

in which the replicas of the elementary pulse $s(x)$ are not allowed to overlap each other. For the elementary pulse we take the rectangular pulse $\mathrm{rect}(x/\sigma)$, of width σ, and write

$$b(x) = \mathrm{rect}(\frac{x}{\sigma}) * \sum_{m=1}^{M} \delta(x - x_m) \, . \tag{3.16}$$

In this equation $*$ stands for one-dimensional convolution and $\delta(x)$ is the one-dimensional Dirac function. The Fourier transform of the pulse train

$$B(u) = \sigma \mathrm{sinc}(\sigma u) \sum_{m=1}^{M} \mathrm{e}^{-i2\pi u x_m} \, , \tag{3.17}$$

must approximate the Fourier transform $\Phi(u)$ within the object window

$$\mathbb{F} = \left\{ u \in \mathbb{R} \mid |u| \leq \tfrac{1}{2}\Delta_u \right\} \, . \tag{3.18}$$

Here we take the object window centered around the origin, and we remark that in this case the twin object window $\mathbb{F}^*$ coincides with the object window $\mathbb{F}$. (As a consequence we can realize objects with an even distribution $|\Phi(u)|^2$ only.) Expressed in the frequency domain the approximation error for a one-dimensional signal becomes

$$P = \int_{\mathbb{F}} |B(u) - \Phi(u)|^2 \mathrm{d}u \, . \tag{3.19}$$

For a high pulse density the width σ of the nonoverlapping pulses has to become very small, which allows the approximation [1] $\sigma \mathrm{sinc}(\sigma u) \approx \sigma \cdot 1 = h$. This approximation is allowed as long as $\sigma \ll 2\Delta_u^{-1}$. In the spatial domain this means that the rectangular pulse is replaced by a Dirac pulse with equal area, and the approximating signal becomes the impulse train

$$b(x) = h \sum_{m=1}^{M} \delta(x - x_m) \, . \tag{3.20}$$

[1] We introduce h on account of dimensionality reasons. Note that $[\delta(x)] = [x]^{-1}$.

We still have to avoid overlap of the original rectangular pulses, and therefore the distance between the successive pulse positions must not be smaller than σ.

Since $\mathbb{F}$ is centered around the origin we try to equate the spectra $\Phi(u)$ and $B(u)$ for very small spatial frequencies u. Particularly, for $u = 0$ the value of $\Phi(u)$ equals the total area

$$A = \int_{-\frac{1}{2}\Delta_x}^{\frac{1}{2}\Delta_x} \phi(x)\mathrm{d}x \tag{3.21}$$

of the signal $\phi(x)$ and the same consideration holds for $B(u)$. This means that the total area of $b(x)$ has to equal the total area of $\phi(x)$. For M pulses with area h this requirement is fulfilled [2] if $h = A/M$. However, merely considering $u = 0$ yields not more than a strictly local error measure in the frequency domain. As a result the error measure in the spatial domain is only of a global nature and gives us no information where to place the pulses. The positioning of the pulses can thus be used to achieve a satisfactory approximation for other frequencies $u \in \mathbb{F}$ as well, that is, in the vicinity of $u = 0$.

Although this does not lead to the exact minimum of the approximation error, we discuss a simple approach where the pulse positions are calculated independently. For that purpose we write $\phi(x)$ as a sum of partial signals $\phi_m(x)$ with partial Fourier transforms $\Phi_m(u)$:

$$\phi(x) = \sum_{m=1}^{M} \phi_m(x) \leftrightarrow \Phi(u) = \sum_{m=1}^{M} \Phi_m(u) \, . \tag{3.22}$$

In the same way the pulse train $b(x)$ is written as a sum of partial signals $b_m(x)$ with partial Fourier transforms $B_m(u)$:

$$b(x) = \sum_{m=1}^{M} b_m(x) \leftrightarrow B(u) = \sum_{m=1}^{M} B_m(u) \, . \tag{3.23}$$

The obvious choice for the partial signals $b_m(x)$ is $b_m(x) = h\delta(x - x_m)$ and we find for the accompanying partial Fourier transforms $B_m(u) = he^{-\imath 2\pi u x_m}$. Next each partial signal $\phi_m(x)$ is approximated by one partial signal $b_m(x)$ and thus by one pulse. This means that each partial signal should have an area h. One possible choice to achieve this reads

$$\phi_m(x) = \begin{cases} \phi(x) & x \in [\chi_{m-1}, \chi_m] \\ 0 & \text{elsewhere} \end{cases} \, . \tag{3.24}$$

In this way the total interval $[-\frac{1}{2}\Delta_x, \frac{1}{2}\Delta_x]$ is divided in subintervals [3] $[\chi_{m-1}, \chi_m]$ for $m = 1, \ldots, M$ with $\chi_0 = -\frac{1}{2}\Delta_x$ and $\chi_M = \frac{1}{2}\Delta_x$. The interval boundaries χ_m are taken such that the partial signals $\phi_m(x)$ have equal area

$$\int \phi_m(x)\mathrm{d}x = \int_{\chi_{m-1}}^{\chi_m} \phi(x)\mathrm{d}x = h \, . \tag{3.25}$$

[2] In the remainder of this chapter we assume that h divides A.

[3] Note that the pulse positions are indicated with x_m and the interval boundaries with χ_m.

With this approach every partial signal $\phi_m(x)$ is approximated by a Dirac-pulse $b_m(x) = h\delta(x - x_m)$ having exactly the same area, where the pulse position x_m depends on $\phi(x)$ for $x \in [\chi_{m-1}, \chi_m]$ only. This concept was introduced by Eschbach and Hauck (1987). Although the exact pulse positions still have to be determined, we expect that $x_m \in [\chi_{m-1}, \chi_m]$. Summing up all the partial approximations gives the desired pulse train $b(x)$. The area of $b(x)$ equals the area of $\phi(x)$ and thus we have $B(0) = \Phi(0)$. Moreover, by increasing the number of pulses we can force the total approximation error on $\mathbb{F}$ to zero, as will be shown later.

In each subinterval the input signal $\phi(x)$ has an area $h = A/M$ and exactly one pulse with the same area is placed. Since the subintervals are narrow in regions where $\phi(x)$ takes large values, a high pulse-density will be the result. In regions where the input signal takes small values, the pulse-density will be low. This is in accordance with our intuition.

In order to calculate the interval boundaries χ_m we introduce the function

$$g(x) = \int_{-\frac{1}{2}\Delta_x}^{x} \phi(s)\mathrm{d}s \ . \tag{3.26}$$

with boundary values $g(-\frac{1}{2}\Delta_x) = 0$ and $g(\frac{1}{2}\Delta_x) = A$. Under our assumption that $\phi(x)$ is positive, $g(x)$ is a monotonically increasing function which allows inversion, i.e. $x = x(g)$. Equation (3.26) reads in differential form $\phi(x)\mathrm{d}x = \mathrm{d}g$. The coordinate transformation $g(x)$ maps the interval $[-\frac{1}{2}\Delta_x, \frac{1}{2}\Delta_x]$ in the x-domain onto $[0, A]$ in the g-domain. Dividing the interval $[0, A]$ in sub-intervals $[\gamma_{m-1}, \gamma_m]$ with equal length $\gamma_m - \gamma_{m-1} = h$ then implies a division of the interval $[-\frac{1}{2}\Delta_x, \frac{1}{2}\Delta_x]$ in the desired subintervals $[\chi_{m-1}, \chi_m]$, as shown in Figure 3.1. The boundaries of the subintervals are $\chi_m = x(\gamma_m)$ with $\gamma_m = mh$; the boundaries of the total domain are $\chi_0 = x(\gamma_0) = x(0) = -\frac{1}{2}\Delta_x$ and $\chi_M = x(\gamma_M) = x(A) = \frac{1}{2}\Delta_x$.

Applying the coordinate transformation in the expression of the Fourier transform of $\phi(x)$ yields

$$\Phi(u) = \int_{-\frac{1}{2}\Delta_x}^{\frac{1}{2}\Delta_x} \phi(x)e^{-i2\pi ux}\mathrm{d}x = \int_0^A e^{-i2\pi ux(g)}\mathrm{d}g = \sum_{m=1}^{M} \int_{\gamma_{m-1}}^{\gamma_m} e^{-i2\pi ux(g)}\mathrm{d}g = \sum_{m=1}^{M} \Phi_m(u) \ . \tag{3.27}$$

Next, for each partial Fourier transform $\Phi_m(u)$ the function $x(g)$ is approximated by a constant $x_m = x(g_m)$. The result is the Fourier transform

$$B(u) = \sum_{m=1}^{M} B_m(u) = h \sum_{m=1}^{M} e^{-i2\pi ux_m} \tag{3.28}$$

of the desired pulse train. The pulse positions depend on the approximation $x_m \approx x(g)$ used in each sub-interval.

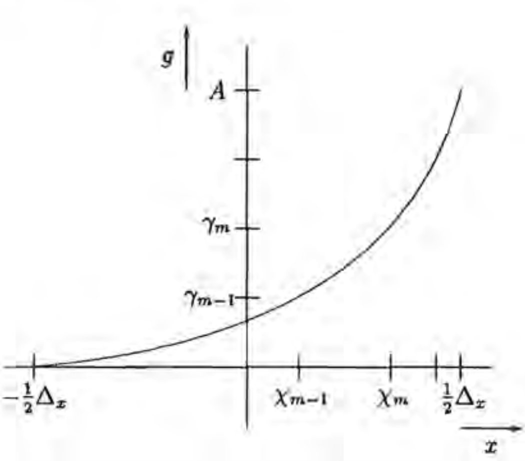


Figure 3.1: The input signal $\phi(x)$ defines the coordinate transformation $g(x)$ which maps the equal-length intervals $[\gamma_{m-1}, \gamma_m]$ on the non-equal-length intervals $[\chi_{m-1}, \chi_m]$.

We note that when the coordinate transformation is *not* applied the approximation leads to a sampling problem. In that case we have

$$\Phi(u) = \int_{-\frac{1}{2}\Delta_x}^{\frac{1}{2}\Delta_x} \phi(x) e^{-i2\pi ux} dx = \\ \sum_{m=1}^{M} \int_{\chi_{m-1}}^{\chi_m} \phi(x) e^{-i2\pi ux} dx \approx X \sum_{m=1}^{M} \phi(x_m) e^{-i2\pi u x_m} = B(u) \ . \quad (3.29)$$

The subintervals $[\chi_{m-1}, \chi_m]$ are now of equal length $X = \Delta_x/M$ and again within each interval one pulse is placed. The result $B(u)$ of the above approximation is the Fourier transform of

$$b(x) = X \sum_{m=1}^{M} \phi(x_m) \delta(x - x_m) \ . \quad (3.30)$$

When we choose the middle of each interval $x_m = \frac{1}{2}(\chi_{m-1} + \chi_m)$ for the pulse positions, the pulse train consists of equidistant Dirac pulses with varying area. This approximation can be considered as the result of sampling the input signal $\phi(x)$ with a constant sampling distance $X = \Delta_x/M$. Shannon's sampling theorem states that when bandlimited signals are sampled with a sufficiently high sampling rate, perfect reconstruction of the original signal from the sampled signal is possible. This is because $B(u)$ equals the original $\Phi(u)$ within $\mathbb{F}$, and thus the approximation error P vanishes. The varying pulse area and equidistant pulse positions for ordinary sampling are exchanged for a constant pulse area and non-equidistant pulse positions for pulse-density modulation. A perfect reconstruction is now only possible in the limiting case $h \to 0$.

Since pulse-density modulation appears to be related to numerical integration, we investigate in the next subsections which known numerical integration methods are applicable to generate equal-area pulse trains. In general, numerical integration of a function $f(g)$ defined on the interval $[a, b]$ is based on

$$\int_a^b f(g)\mathrm{d}g = c_0 f(g_0) + c_1 f(g_1) + \dots c_K f(g_K) + R(f) . \tag{3.31}$$

The integration nodes $g_0, \dots, g_K$ and weights $c_0, \dots, c_K$ are chosen such that the remainder $R(f)$ vanishes for a certain class of functions, which are polynomials in our case. Expanding $f(g)$ in a Taylor series this means that the numerical integration becomes exact for a certain truncation of this series. For pulse-density modulation, where we have $f(g) = \mathrm{e}^{-i2\pi u x(g)}$, we require the integration weights to be identical. Only in that case equal-area (Dirac) pulses are generated.

3.2.2 Gauss' integration methods

With Gauss' 1-point integration method ('Gauss-1') we find for the integration node $g_0 = \frac{1}{2}(a + b)$ and for the integration weight $c_0 = (b - a)$. Applying these results in the approximation of the partial signal $\Phi_m(u)$ on the interval $[\gamma_{m-1}, \gamma_m]$ we have

$$\Phi_m(u) = \int_{\gamma_{m-1}}^{\gamma_m} e^{-i2\pi u x(g)}\mathrm{d}g \approx h e^{-i2\pi u x(g_m)} = B_m(u) , \tag{3.32}$$

where $g_m = \frac{1}{2}(\gamma_{m-1} + \gamma_m)$. Obviously, we have $g_m - \gamma_{m-1} = \gamma_m - g_m = \frac{1}{2}h$, which is identical to

$$\int_{\gamma_{m-1}}^{g_m} \mathrm{d}g = \int_{g_m}^{\gamma_m} \mathrm{d}g = \tfrac{1}{2}h . \tag{3.33}$$

After applying the inverse coordinate transformation $x(g)$ we find in the space domain

$$\int_{\chi_{m-1}}^{x_m} \phi(x)\mathrm{d}x = \int_{x_m}^{\chi_m} \phi(x)\mathrm{d}x = \tfrac{1}{2}h . \tag{3.34}$$

We conclude that with Gauss' 1-point integration formula the pulse position x_m is given as the (normalized) median value of the input signal on each interval $[\chi_{m-1}, \chi_m]$. As a consequence, the area of $\phi(x)$ between two successive pulse positions also equals h, and we have

$$h = \int_{x_{m-1}}^{x_m} \phi(x)\mathrm{d}x \leq \int_{x_{m-1}}^{x_m} \mathrm{d}x = (x_m - x_{m-1}) \cdot 1 . \tag{3.35}$$

Using $h = \sigma \cdot 1$ we find $\sigma \leq x_m - x_{m-1}$, which states that with Gauss' 1-point integration method, overlap of the pulses cannot occur.

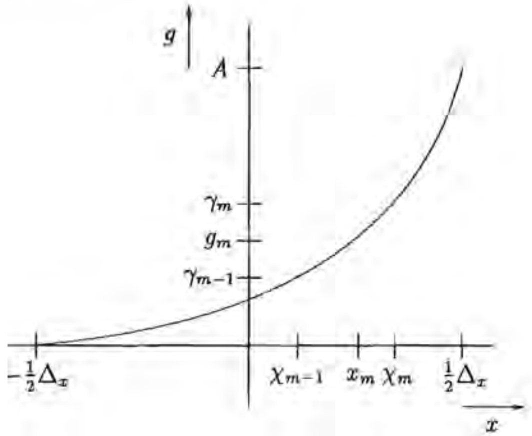


Figure 3.2: With Gauss' 1-point integration formula the pulse position is $x_m = x(g_m)$ with $g_m = \frac{1}{2}(\gamma_{m-1} + \gamma_m)$.

In our approach we have determined one pulse position x_m in one interval $[\chi_{m-1}, \chi_m]$, but we might as well try to determine two adjacent pulse positions x_{2m-1}, x_{2m} in a double interval $[\chi_{2m-2}, \chi_{2m}]$ or three adjacent pulse positions x_{3m-2}, x_{3m-1} and x_{3m} in a triple interval $[\chi_{3m-3}, \chi_{3m}]$, and so on.

To determine the pulse positions in pairs, we combine two adjacent intervals and use Gauss' 2-points integration formula (Abramowitz and Stegun, 1970) ('Gauss-2') as a numerical approximation for

$$\Phi_{2m}(u) + \Phi_{2m-1}(u) = \int_{\gamma_{2m-2}}^{\gamma_{2m}} e^{-i2\pi u x(g)} \mathrm{d}g \ . \tag{3.36}$$

The resulting approximation is then

$$B_{2m}(u) + B_{2m-1}(u) = h \left[e^{-i2\pi u x(g_{2m})} + e^{-i2\pi u x(g_{2m-1})} \right] \ , \tag{3.37}$$

where the integration nodes are given by

$$g_{2m,2m-1} = \frac{\gamma_{2m} + \gamma_{2m-2}}{2} \pm \frac{\gamma_{2m} - \gamma_{2m-2}}{2\sqrt{3}} = \gamma_{2m-1} \pm \frac{h}{\sqrt{3}} \ . \tag{3.38}$$

Combining more intervals and using Gauss' higher-order integration formulas is not possible since the integration weights are not equal, resulting in a non-equal area pulse train. However, using Chebyshev's integration formula it is possible to use higher-order methods.

3.2.3 Chebyshev's integration methods

With Chebyshev's integration method the integration nodes in (3.31) are determined under the constraint that all weights are equal. This allows us to simultaneously determine the positions of K ($3 \leq K \leq 9$, with the exception of $K = 8$) Dirac pulses in K intervals. We assume that K divides M and use Chebyshev's K-points ('Cheby-K') integration formula for $m = 1, \ldots, M/K$ on the interval $[\gamma_{Km-K}, \gamma_{Km}]$ (Abramowitz and Stegun, 1970), according to

$$\sum_{k=0}^{K-1} \Phi_{Km-k}(u) = \int_{\gamma_{Km-K}}^{\gamma_{Km}} e^{-i2\pi u x(g)} \mathrm{d}g \approx h \sum_{k=0}^{K-1} e^{-i2\pi u x(g_{Km-k})} = \sum_{k=0}^{K-1} B_{Km-k}(u)\,. \tag{3.39}$$

The integration nodes are $g_{Km-k} = \frac{1}{2}(\gamma_{Km} + \gamma_{Km-K}) + \frac{1}{2}\xi_k K h$, with ξ_k (for $k = 0, 1, \ldots, K-1$) the K zeros of the polynomial part of

$$\xi^K \exp\left[-K\left(\frac{1}{2\cdot 3\xi^2} + \frac{1}{4\cdot 5\xi^4} + \frac{1}{6\cdot 7\xi^6} + \ldots\right)\right]\,. \tag{3.40}$$

We remark that Chebyshev's 1- and 2-points integration formulas yield the same results as Gauss' 1- and 2-points integration formulas, respectively.

Example

We determine the integration nodes for Chebyshev's 2-points integration formula ($K = 2$) and show that the result agrees with Gauss' 2-points formula. For $k = 0, 1$ the integration nodes are given by

$$\begin{aligned} g_{2m} &= \tfrac{1}{2}(\gamma_{2m} + \gamma_{2m-2}) + \tfrac{1}{2}\xi_0 2h = \gamma_{2m-1} + \xi_0 h \\ g_{2m-1} &= \tfrac{1}{2}(\gamma_{2m} + \gamma_{2m-2}) + \tfrac{1}{2}\xi_1 2h = \gamma_{2m-1} + \xi_1 h\,. \end{aligned} \tag{3.41}$$

To find ξ_0, ξ_1 we have to consider the polynomial part of

$$\xi^2 \exp\left[-2\left(\frac{1}{2\cdot 3\xi^2} + \frac{1}{4\cdot 5\xi^4} + \frac{1}{6\cdot 7\xi^6} + \ldots\right)\right] \tag{3.42}$$

and determine its zeros. Using the Taylor expansion $e^x = 1 + x + \frac{1}{2}x^2 + \ldots$ we find

$$\xi^2 + \xi^2\left[-2\left(\frac{1}{2\cdot 3\xi^2} + \ldots\right)\right] + \tfrac{1}{2}\xi^2\left[-2\left(\frac{1}{2\cdot 3\xi^2} + \ldots\right)\right]^2 + \ldots\,, \tag{3.43}$$

with a polynomial part $\xi^2 - \frac{1}{3}$. The zeros $\xi_0 = 1/\sqrt{3}$, $\xi_1 = -1/\sqrt{3}$ indeed give rise to the integration nodes of Gauss' 2-points integration formula.

As a second example the integration nodes for Chebyshev's 3-points integration formula ($K = 3$) are calculated. For $k = 0, 1, 2$ the integration nodes are given by

$$\begin{aligned} g_{2m} &= \tfrac{1}{2}(\gamma_{3m} + \gamma_{3m-3}) + \tfrac{1}{2}\xi_0 3h \\ g_{2m-1} &= \tfrac{1}{2}(\gamma_{3m} + \gamma_{3m-3}) + \tfrac{1}{2}\xi_1 3h \\ g_{2m-2} &= \tfrac{1}{2}(\gamma_{3m} + \gamma_{3m-3}) + \tfrac{1}{2}\xi_2 3h\ . \end{aligned} \tag{3.44}$$

In this case ξ_0, ξ_1 and ξ_2 are the zeros of the polynomial part of

$$\xi^3 + \xi^3\left[-3\left(\frac{1}{2\cdot 3\xi^2} + \ldots\right)\right] + \tfrac{1}{2}\xi^3\left[-3\left(\frac{1}{2\cdot 3\xi^2} + \ldots\right)\right]^2 + \ldots\ , \tag{3.45}$$

i.e. the zeros of $\xi^3 - \frac{1}{2}\xi$. For $K = 3$ we thus find the integration nodes in (3.44) with $\xi_0 = 1/\sqrt{2}, \xi_1 = 0, \xi_2 = -1/\sqrt{2}$.

3.2.4 Integration methods based on moments

The above integration methods are based on exact integration of a number of terms (polynomials) in the Taylor series expansion of $e^{-i2\pi u x(g)}$ in g. The integration error (caused by the next term) depends on the spatial frequency [4] u. As a result, the approximation in the frequency domain is good in the vicinity of $u = 0$, and the errors are 'shifted' towards higher frequencies.

In the present subsection we consider an alternative way to achieve this goal. We recall that we have to approximate the partial Fourier transforms

$$\Phi_m(u) = \int_{\gamma_{m-1}}^{\gamma_m} e^{-i2\pi u x(g)}\mathrm{d}g = \int \phi_m(x)e^{-i2\pi u x}\mathrm{d}x \tag{3.46}$$

by

$$B_m(u) = he^{-i2\pi u x_m} = \int b_m(x)e^{-i2\pi u x}\mathrm{d}x\ . \tag{3.47}$$

In the vicinity of $u = 0$ we can write $\Phi_m(u)$ and $B_m(u)$ in a Taylor series expansion

$$\begin{aligned} \Phi_m(u) &= \Phi_m(0) + \Phi'_m(0)u + \ldots \\ &= \int \phi_m(x)\mathrm{d}x - i2\pi u \int x\phi_m(x)\mathrm{d}x + \ldots \end{aligned} \tag{3.48}$$

and

$$\begin{aligned} B_m(u) &= B_m(0) + B'_m(0)u + \ldots \\ &= h - i2\pi x_m h u + \ldots\ . \end{aligned} \tag{3.49}$$

[4] A comparable situation occurs in a Taylor series expansion for $\cos(2\pi u x)$ in x. The higher the frequency u the more terms we need in order to achieve a desired accuracy.

Due to (3.25) the zero-order terms in the two Taylor series expansions (3.48) and (3.49) are equal. With an appropriate choice of the pulse position we can also equate the first-order terms in the expansions and obtain $B'(0) = \Phi'(0)$. The solution reads

$$x_m = \frac{1}{h}\int_{\chi_{m-1}}^{\chi_m} x\phi(x)\mathrm{d}x = \frac{1}{h}\int x\phi_m(x)\mathrm{d}x \,. \tag{3.50}$$

The obtained pulse position x_m is the normalized first-order moment of $\phi_m(x)$ (the 'center of gravity') and for this choice of x_m we have the equalities

$$\begin{aligned} \int \phi_m(x)\mathrm{d}x &= \int b_m(x)\mathrm{d}x = h \\ \int x\phi_m(x)\mathrm{d}x &= \int x b_m(x)\mathrm{d}x = h x_m \,. \end{aligned} \tag{3.51}$$

In this way the zero- and first-order moments of $\phi_m(x)$ and $b_m(x)$ are equated, hence the name 'moment-1' method.

We might as well try to equate the second-order derivative for $u = 0$ by considering also the second-order moment ('moment-2'). Again we combine two adjacent intervals and determine two pulse positions x_{2m-1}, x_{2m} in the double interval $[\chi_{2m-2}, \chi_{2m}]$ by equating the zero-, first- and second-order moments. In that case we approximate

$$\Phi_{2m}(u) + \Phi_{2m-1}(u) = \int_{\gamma_{2m-2}}^{\gamma_{2m}} e^{-i2\pi u x(g)}\mathrm{d}g \tag{3.52}$$

by

$$B_{2m}(u) + B_{2m-1}(u) = h\left[e^{-i2\pi u x(g_{2m})} + e^{-i2\pi u x(g_{2m-1})}\right] \,. \tag{3.53}$$

We have to solve x_{2m} and x_{2m-1} (and eliminate h) from the set of equations

$$\begin{aligned} \int \left[\phi_{2m}(x) + \phi_{2m-1}(x)\right]\mathrm{d}x &= \int \left[b_{2m}(x) + b_{2m-1}(x)\right]\mathrm{d}x = 2h \\ hm_1 = \int x\left[\phi_{2m}(x) + \phi_{2m-1}(x)\right]\mathrm{d}x &= \\ &\int x\left[b_{2m}(x) + b_{2m-1}(x)\right]\mathrm{d}x = h(x_{2m} + x_{2m-1}) \\ hm_2 = \int x^2\left[\phi_{2m}(x) + \phi_{2m-1}(x)\right]\mathrm{d}x &= \\ &\int x^2\left[b_{2m}(x) + b_{2m-1}(x)\right]\mathrm{d}x = h(x_{2m}^2 + x_{2m-1}^2) \,. \end{aligned} \tag{3.54}$$

The solution reads

$$x_{2m,2m-1} = \frac{m_1 \pm \sqrt{2m_2 - m_1^2}}{2} \,. \tag{3.55}$$

Using Schwarz' inequality

$$|\int p(x)q(x)\mathrm{d}x|^2 \leq \int |p(x)|^2\mathrm{d}x \int |q(x)|^2\mathrm{d}x \tag{3.56}$$

with $p(x) = x\sqrt{\phi(x)}$ and $q(x) = \sqrt{\phi(x)}$, we have the property

$$\begin{aligned}(hm_1)^2 &= \left[\int x[\phi_{2m}(x) + \phi_{2m-1}(x)]\mathrm{d}x\right]^2 \\ &< \left[\int x^2[\phi_{2m}(x) + \phi_{2m-1}(x)]\mathrm{d}x\right]\left[\int [\phi_{2m}(x) + \phi_{2m-1}(x)]\mathrm{d}x\right] \\ &= (hm_2)(2h) \ ,\end{aligned} \tag{3.57}$$

which implies that the two positions x_{2m} and x_{2m-1} are real and non-coincident. The equality sign in Schwarz' inequality holds only if $p(x) = cq^*(x)$ with c an arbitrary real constant; this condition is not satisfied here. We remark that the extension to higher-order moments may not be possible, because in that case there is no guarantee that the solutions are real and non-coincident.

3.2.5 Error analysis for the integration methods

In the previous subsections we have proposed several methods to calculate the pulse positions. For all methods we expect that the approximation error will decrease to zero when the number of pulses is continuously increased. In this section we will analyze this error behaviour. Since all methods are based on numerical integration it is natural to use the error analysis known from numerical integration (Davis and Rabinowitz, 1984). Considering the approximation error in the spatial domain, we try to minimize

$$P = \int |a(x) * [b(x) - \phi(x)]\,|^2 \mathrm{d}x \ . \tag{3.58}$$

Under the assumption that $A(u) = \mathrm{rect}(u/\Delta_u)$ we have $a(x) = \Delta_u \mathrm{sinc}(\Delta_u x)$. For convenience we define the filtered signals $\tilde{\phi}(x) = a(x) * \phi(x)$ and $\tilde{b}(x) = a(x) * b(x)$. Minimizing (3.58) then implies minimizing the signal power of $\tilde{e}(x) = \tilde{b}(x) - \tilde{\phi}(x)$. To determine $\tilde{\phi}(x)$ we use the coordinate transformation $g(x)$ followed by a numerical approximation:

$$\tilde{\phi}(x) = \int_{-\frac{1}{2}\Delta x}^{\frac{1}{2}\Delta x} a(x-s)\phi(s)\mathrm{d}s = \int_0^A a(x - s(g))\mathrm{d}g \approx h\sum_{m=1}^{M} a(x - s(g_m)) = \tilde{b}(x) \ , \tag{3.59}$$

where $s(g)$ is the solution of the differential equation $\mathrm{d}s/\mathrm{d}g = \phi^{-1}(s)$. By definition the filtered error signal $\tilde{e}(x)$ is then the approximation error caused by the numerical integration. When Gauss' 1-point integration rule is used to determine the pulse position $x(g_m)$ in the interval $[\gamma_{m-1}, \gamma_m]$, we can write for the approximation error

$$\int_{\gamma_{m-1}}^{\gamma_m} a[x - s(g)]\mathrm{d}g = ha[x - s(g_m)] + \frac{h^3}{24}\frac{\mathrm{d}^2}{\mathrm{d}g^2}a[x - s(g)]|_{g=\xi_m} \ , \tag{3.60}$$

method	order	method	order
Gauss-1	M^{-2}	Cheby-5	M^{-6}
Gauss-2	M^{-4}	Cheby-6	M^{-6}
moment-1	M^{-2}	Cheby-7	M^{-8}
moment-2	M^{-4}	Cheby-8	——
Cheby-3	M^{-4}	Cheby-9	M^{-10}
Cheby-4	M^{-6}		

Table 3.1: Error behaviour for the various integrations methods.

with $\xi_m \in (\gamma_{m-1}, \gamma_m)$. After summing up the contributions of all intervals we find for the total filtered error signal

$$\tilde{e}(x) = \frac{h^3}{24} \sum_{m=1}^{M} \frac{\mathrm{d}^2}{\mathrm{d}g^2} a(x - s(g))_{g=\xi_m} = \frac{h^2}{24} \int_0^A \frac{\mathrm{d}^2}{\mathrm{d}g^2} a(x - s(g)) \mathrm{d}g + O(h^4) \ . \quad (3.61)$$

The left-hand summation in (3.61) can be considered as a Riemann sum for the integral on the right-hand side. Interchanging differentiation and integration we find

$$\tilde{e}(x) = \frac{h^2}{24} \frac{\mathrm{d}}{\mathrm{d}g} \left(a[x - s(A)] - a[x - s(0)]\right) + O(h^4). \quad (3.62)$$

With the chain rule for differentiation

$$\frac{\mathrm{d}}{\mathrm{d}g} a(x) = \frac{\mathrm{d}}{\mathrm{d}x} a(x) \cdot \frac{\mathrm{d}x}{\mathrm{d}g} = \frac{a'(x)}{\phi(x)} \quad (3.63)$$

we finally find

$$\tilde{e}(x) = \frac{h^2}{24} \left[\frac{a'(x + \frac{1}{2}\Delta x)}{\phi(-\frac{1}{2}\Delta x)} - \frac{a'(x - \frac{1}{2}\Delta x)}{\phi(\frac{1}{2}\Delta x)} \right] + O(h^4) \ . \quad (3.64)$$

This equation shows that the filtered error signal decreases quadratically to zero when the (average) pulse-density is increased, since $h = A/M$. We remark that the given derivation is only valid for signals $\phi(x) > 0$. A more general derivation is given in (Jagerman, 1966), which is valid for signals $\phi(x) \geq 0$. For this class of signals the approximation error does not decrease as fast as $O(M^{-2})$. However, with the application of amplitude holograms we can always add a small constant in order to ensure that $\phi(x) > 0$, and therefore our error analysis is appropriate.

A similar analysis can be carried out for the other methods. The results for the various numerical integration methods are summarized in Table 3.1. We conclude that with the higher-order methods a smaller approximation error can be achieved. However, the higher-order methods require $a(x - s(g))$ to admit differentiations of sufficient order, meaning that $\phi(x)$ should be sufficiently smooth. If $\phi(x)$ does not fulfill this constraint, the approximation error will not decrease as fast as shown in Table 3.1. A notable fact is that with the Chebyshev integration methods the approximation error does not decrease monotonically with increasing order.

3.2.6 Computer simulation results

Using the various pulse-density modulation methods, we have generated pulse trains for a given input signal $\phi(x)$ and calculated the approximation error as a function of the number of pulses M. To generate a continuous low-pass signal $\phi(x)$, a sequence of uniformly distributed noise samples is filtered by a discrete low-pass filter with cut-off frequency $\theta = \frac{1}{20}$. From the output of the digital filter we take a sequence of $2N+1$ successive samples $\phi[n]; n = -N, \ldots, N$. This discrete signal is converted to a continuous signal by means of linear interpolation using the linear spline [5] trian(x).

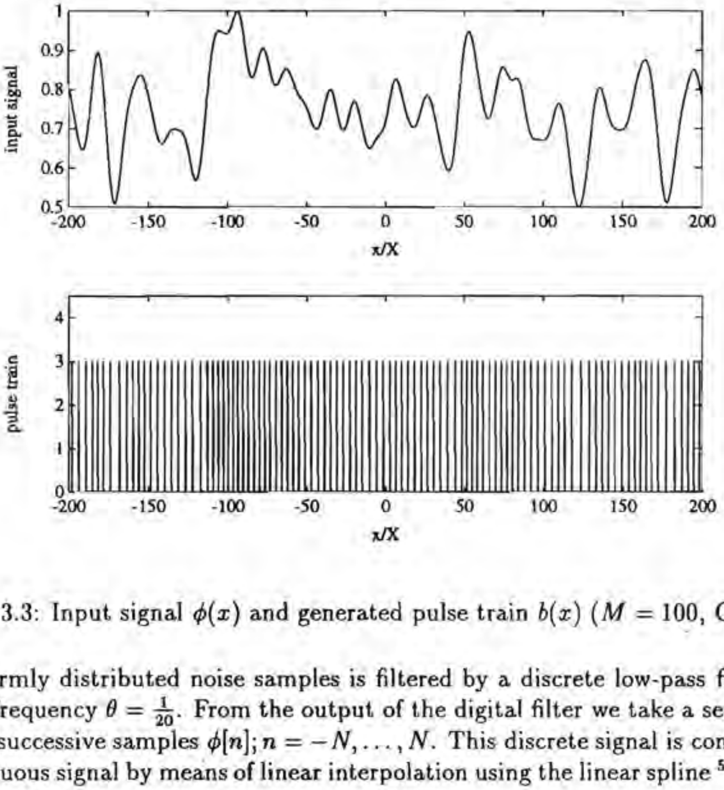


Figure 3.3: Input signal $\phi(x)$ and generated pulse train $b(x)$ ($M = 100$, Gauss-1).

[5] The function trian(x) is defined as

$$\text{trian}(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{3.65}$$

The resulting polygon

$$\phi(x) = \sum_n \phi[n]\text{trian}\left(\frac{x - nX}{X}\right) \tag{3.66}$$

is set to zero outside the interval $[-\frac{1}{2}\Delta_x, \frac{1}{2}\Delta_x]$. Finally a positive constant is added and the result is scaled according to $0 < \phi(x) \leq 1$. In Figure 3.3 we show a polygon ($N = 200$), with a total number of 401 samples.

In addition a pulse train with $M = 100$ pulses, resulting from Gauss' 1-point integration method is shown. We observe that the local pulse density of the pulse train is modulated by the local amplitude of the input signal $\phi(x)$: large amplitudes of $\phi(x)$ cause a high pulse density of $b(x)$, while small amplitudes cause a low pulse density. Next, both signals $\phi(x)$ and $b(x)$ are filtered by an ideal low-pass filter $A(u)$ with cut-off frequency $\frac{1}{2}\Delta_u = \frac{1}{20}X^{-1}$. The accompanying output signals $\tilde{\phi}(x)$ and $\tilde{b}(x)$ are shown in Figure 3.4. Even for $M = 100$, the approximation with pulse-density modulation is quite good. (Note that with a cut-off frequency $\theta = \frac{1}{20}$ we have $\phi(x)$ oversampled with a factor 10, and therefore $M = 100$ is only about 2.5 times the minimum number of samples.) This is expressed more adequately by considering the noise power P of the error signal $\tilde{e}(x)$. For the calculation of P we

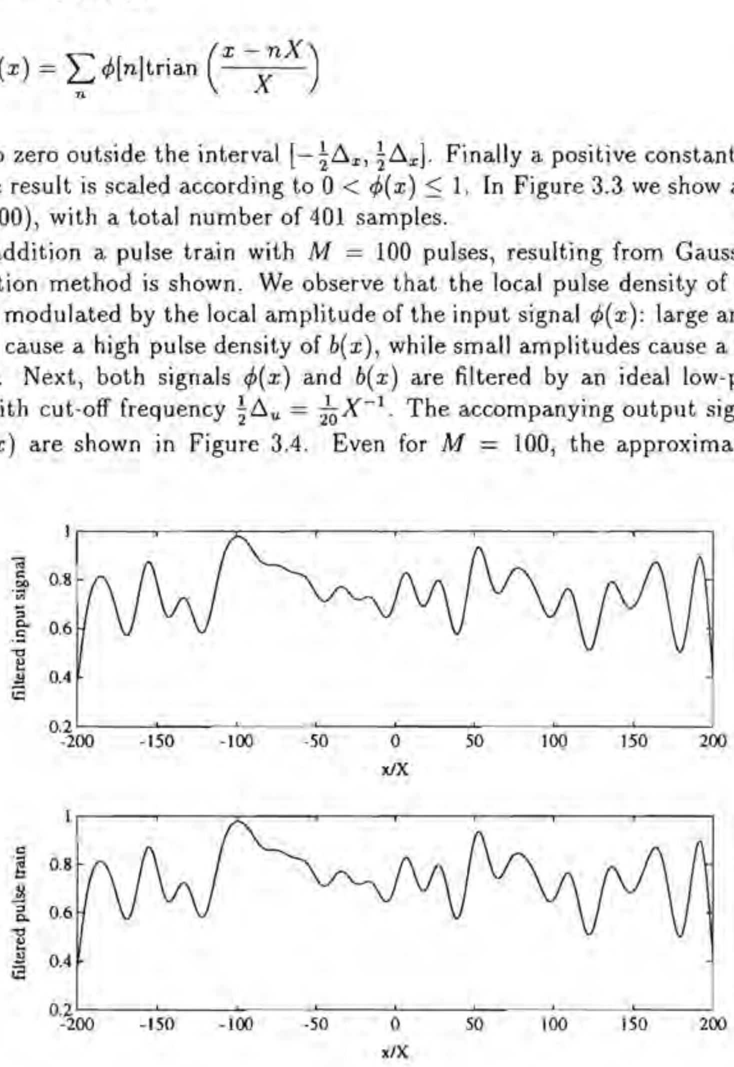


Figure 3.4: Filtered input signal $\tilde{\phi}(x)$ and filtered pulse train $\tilde{b}(x)$.

have used the approximation

$$P \approx \frac{1}{2N+1} \sum_{n=-N}^{N} \tilde{e}^2[n] \tag{3.67}$$

for (3.58). Numerical results for the various integration methods as a function of the total number of pulses are given in Figure 3.5. In order to make a comparison with the theoretical expectation possible, we have shown $P_{\text{rms}} = \sqrt{P}$. In agreement with the given theoretical analysis the Gauss-1 integration method introduces an error of $O(M^{-2})$. Also the results of the moment-1, Gauss-2, the moment-2 and the Cheby-3 integration methods verify the theoretical values of Table 3.1. For the higher-order methods (we have shown only Cheby-4), however, the experimental results are not in agreement with the theoretical values. We have already noted that the higher-order methods impose more severe demands on the smoothness of the continuous signal. In our case, $\phi(x)$ is generated using linear interpolation. Therefore, even the second-order derivative of $\phi(x)$ does not exist. This probably causes problems for the higher-order integration methods, and higher-order splines should be used for the interpolation. Of course, this requires extra computation. Although moment-1 takes the first-order moment into account its does not perform as good as Gauss-1. Furthermore we note that we have added a relatively large constant to the input

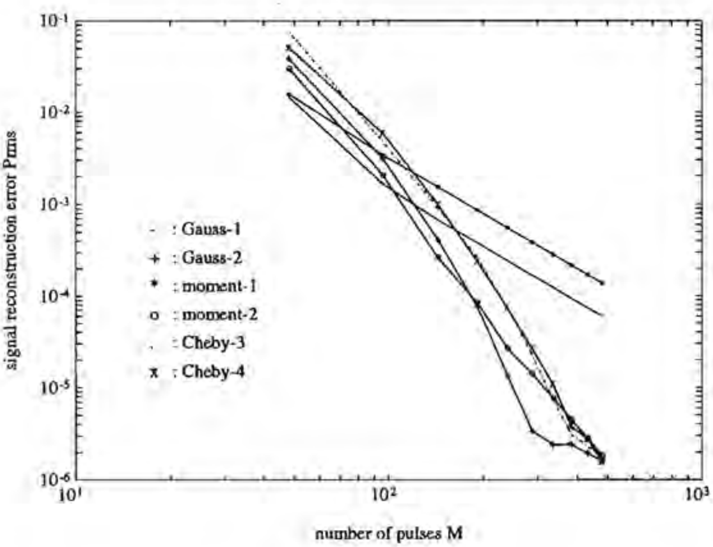


Figure 3.5: Error P_{rms} as function of total number of pulses for various integration methods.

signal, leading to a small efficiency of the hologram. For smaller constants the difference in performance between higher- and lower-order methods disappears.

3.2.7 Conclusion

In this section we have discussed the approximation of a given signal by a pulse-density signal for the one-dimensional case. While the original problem is formulated as a multidimensional minimization problem, we have considered an approach which makes it possible to determine the positions of small groups of pulses independently. This leads to several methods for various group-sizes, all based on numerical integration. For a large number of pulses, the approximation error is forced to zero. However, only the lower-order methods turn out to be of practical interest.

3.3 Two-dimensional continuous pulse-density modulation

3.3.1 Introduction

In the previous sections we have proposed several methods to determine the pulse positions for one-dimensional signals. All methods are based on dividing the (input) signal in equal-area parts, leading to a division of the (one-dimensional) domain in subintervals. This approach can be generalized to the two-dimensional case, where the two-dimensional domain $\mathbb{H}$ has thus to be divided in M nonoverlapping cells C_m with equal 'area'

$$h = \iint_{C_m} \phi(x,y)\mathrm{d}x\mathrm{d}y \; . \tag{3.68}$$

Again, the total area of $\phi(x,y)$ is assumed to equal $A = Mh$.

While for the one-dimensional case the solution of the division-problem is unique, an infinite number of solutions exists in the two-dimensional case. As a result a number of different approaches have been proposed. In (Eschbach and Hauck, 1987) a sequential algorithm for the determination of the cells is described. Starting in one corner of the domain $\mathbb{H}$ the signal samples (the input signal is assumed to be discrete) are processed in a well-defined order and grouped into clusters. Each cluster (or cell) is represented by one pulse. Variations on this concept can be found in (Eschbach, 1990) and in (Koppelaar, 1992) where also a 'recursive domain division' algorithm is considered.

Here, we carry the generalization of the one-dimensional approach further by means of the introduction of a two-dimensional coordinate transformation $f = f(x,y)$, $g = g(x,y)$ analogous to the one-dimensional coordinate transformation

$g = g(x)$. In this way the coordinates (x, y) are transformed to a new set of coordinates (f, g). With the infinitesimal relation $\phi(x,y)\mathrm{d}x\mathrm{d}y = \mathrm{d}f\mathrm{d}g$ we find for the Fourier transform of $\phi(x, y)$ in terms of the new coordinates

$$\Phi(u,v) = \iint \phi(x,y)e^{-i2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y = \iint e^{-i2\pi[ux(f,g)+vy(f,g)]}\mathrm{d}f\mathrm{d}g \ . \qquad (3.69)$$

Next, the fg-domain is divided in $M = M_1M_2$ cells $\tilde{C}_{m_1m_2} = [(m_1-1)\sqrt{h}, m_1\sqrt{h}] \times [(m_2-1)\sqrt{h}, m_2\sqrt{h}]$:

$$\Phi(u,v) = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \iint_{\tilde{C}_{m_1m_2}} e^{-i2\pi[ux(f,g)+vy(f,g)]}\mathrm{d}f\mathrm{d}g \ . \qquad (3.70)$$

Applying a numerical approximation within each cell we find

$$B(u,v) = h \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \mathrm{e}^{-i2\pi[ux(f_{m_1},g_{m_2})+vy(f_{m_1},g_{m_2})]} \ . \qquad (3.71)$$

The result is the Fourier transform of a set of two-dimensional pulses with equal area h:

$$b(x,y) = h \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \delta(x - x_{m_1m_2}, y - y_{m_1m_2}) \ . \qquad (3.72)$$

For one-dimensional signals the interval boundary χ_m is the solution of $g(x) = mh$ with $m = 0, \ldots, M$. In two dimensions the cells $C_{m_1m_2}$ in the xy-domain are formed by the coordinate curves $f(x,y) = m_1\sqrt{h}$ and $g(x,y) = m_2\sqrt{h}$. In case Gauss' 1-point integration method is applied to determine the pulse position within each cell the solution is given as the intersection of the coordinate curves $f(x,y) = (m_1 - \frac{1}{2})\sqrt{h}$ and $g(x,y) = (m_2 - \frac{1}{2})\sqrt{h}$.

In differential notation the coordinate transformation $f = f(x,y)$, $g = g(x,y)$ reads

$$\begin{aligned} \mathrm{d}f &= \frac{\partial f}{\partial x}\mathrm{d}x + \frac{\partial f}{\partial y}\mathrm{d}y \\ \mathrm{d}g &= \frac{\partial g}{\partial x}\mathrm{d}x + \frac{\partial g}{\partial y}\mathrm{d}y \ . \end{aligned} \qquad (3.73)$$

The determinant of this set of equations, also known as the Jacobian (Kreyszig, 1983), follows from $\mathrm{d}f\mathrm{d}g = \phi(x,y)\mathrm{d}x\mathrm{d}y$ as

$$f_xg_y - f_yg_x = \phi(x,y) \ . \qquad (3.74)$$

For convenience we have introduced the shorthand notation f_x for $\partial f/\partial x$ etc. Due to the fact that $\phi(x,y) \neq 0$, the mapping of (x,y) onto (f,g) is one-to-one.

In order to determine the coordinate transformation we have to solve the nonlinear partial differential equation (3.74). However, the problem is underdetermined, so an additional equation needs to be formulated. A reasonable choice is to take the set of curves $f(x,y) = f_{m_1}$ and $g(x,y) = g_{m_2}$ orthogonal, leading to the differential equation

$$f_x g_x + f_y g_y = 0 \,. \tag{3.75}$$

According to (3.74) grad $f = (f_x, f_y)$ and grad $g = (g_x, g_y)$ form a cell with area $\phi(x,y)$ and (3.75) states that grad f and grad g are perpendicular. Finally, a set of boundary conditions has to be formulated in order to describe the mapping of the boundary of $\mathbb{H}$ in the (x,y)-plane onto the boundary in the (f,g)-plane.

Remark

One is tempted to solve (3.74), (3.75) using the Cauchy-Riemann equations $f_x = g_y$ and $f_y = -g_x$. However, it can be shown that this approach does not yield a solution for general signals $\phi(x,y)$. First, we define the analytic function $w(z) = f(x,y) + ig(x,y)$ and rewrite (3.74) and (3.75) into

$$|w'(z)|^2 = \phi(x,y) \,, \tag{3.76}$$

where the differentiation is carried out with respect to the complex variable $z = x + iy$. Next, we introduce $v(z)$ according to $w'(z) = e^{v(z)}$ and remark that for an analytic function $w(z)$ also $v(z)$ will be analytic. With (3.76) we find Re $v(z) = \frac{1}{2}\ln \phi(x,y)$. Since the real part of an analytic function is harmonic (this holds also for the imaginary part), we have

$$\text{div grad } \ln \phi(x,y) = 0 \,. \tag{3.77}$$

This is a severe restriction on the signal $\phi(x,y)$. We remark that (3.77) implies

$$\text{grad } \phi \cdot \text{grad } \phi = \phi \text{ div grad } \phi \,. \tag{3.78}$$

As an example we consider the particular case where $\phi(x,y) = x^2 + y^2$. Restriction (3.77) is then satisfied and we are able to use the Cauchy-Riemann equations to solve (3.74), (3.75). In this case we have for (3.76)

$$|w'(z)|^2 = |z|^2 \,. \tag{3.79}$$

The solution reads $w(z) = \frac{1}{2}(x^2 - y^2) + ixy + c$, with c a complex constant. Under the assumption that $c = 0$ the curves $f(x,y)$ and $g(x,y)$ are the hyperbolas

$$\begin{aligned} f(x,y) &= \tfrac{1}{2}(x^2 - y^2) \\ g(x,y) &= xy \,. \end{aligned} \tag{3.80}$$

The pulse positions are determined as the intersections of the coordinate curves $f(x,y) = (m_1 - \frac{1}{2})\sqrt{h}$ and $g(x,y) = (m_2 - \frac{1}{2})\sqrt{h}$, shown in Figure 3.6. The pulse density increases with increasing distance $r = \sqrt{x^2 + y^2}$. Note that the boundary of $\mathbb{H}$ is not considered in this solution.

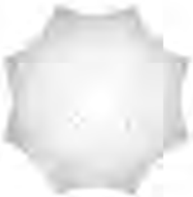


Figure 3.6: Pulse-density modulation for $\phi(x, y) = x^2 + y^2$ by means of the Cauchy-Riemann equations.

For a general input signal $\phi(x, y)$ we try to solve the problem numerically. It appears that dividing the problem in finding $f(x, y)$ and $g(x, y)$ followed by the determination of the pulse positions by calculating the intersections of the set of curves is inappropriate. Instead, we combine both sub-problems and solve them simultaneously. This is the subject of the next subsection.

3.3.2 Coordinate meshes with manageable density

In order to find the pulse positions by numerical means we follow the approach proposed by Christov (1982). This approach is concerned with the generation of a grid with a manageable density. Such grids are applied in numerical methods for solving partial differential equations. This way one is able to adapt the calculation precision to the structure of the underlying problem.

This approach is elucidated by considering the one-dimensional case first, where the coordinate transformation $g(x)$ was defined (cf. (3.26)) according to

$$\begin{aligned} &g_x(x) = \phi(x) \\ &g(-\tfrac{1}{2}\Delta_x) = 0 \; . \end{aligned} \tag{3.81}$$

The interval boundaries and pulse positions are easily determined if the accompanying inverse coordinate transformation $x(g)$ is known. According to (3.81) $x(g)$ is the solution of

$$\begin{aligned} &x_g(g) = \phi^{-1}(x) \\ &x(0) = -\tfrac{1}{2}\Delta_x \; . \end{aligned} \tag{3.82}$$

In order to determine the interval boundaries we solve (3.82) for $g_m = mh$. Since the area of $\phi(x)$ equals A we know (without solving (3.82)) that $x(A) = \frac{1}{2}\Delta_x$. The two constraints are the boundary conditions of a second-order differential equation, which is found by rewriting $x_g(g) = \phi^{-1}(x)$ into $\phi(x)x_g = 1$ followed by a differentiation with respect to g:

$$\begin{aligned} &(\phi(x)x_g)_g = 0 \\ &x(0) = -\tfrac{1}{2}\Delta_x \\ &x(A) = \tfrac{1}{2}\Delta_x \ . \end{aligned} \tag{3.83}$$

Due to our choice of the boundary conditions, the problems (3.82) and (3.83) are equivalent. Solving (3.83) numerically for $g_m = mh$ is thus an alternative way to determine the interval boundaries $\chi_m = x(g_m)$. To this end we replace the differentials by finite differences, according to

$$f_g \approx \frac{f_{m+\frac{1}{2}} - f_{m-\frac{1}{2}}}{h} \ . \tag{3.84}$$

This leads to the finite difference problem

$$\begin{aligned} &-\phi_{m-\frac{1}{2}}\chi_{m-1} + (\phi_{m-\frac{1}{2}} + \phi_{m+\frac{1}{2}})\chi_m - \phi_{m+\frac{1}{2}}\chi_{m+1} = 0 \\ &\chi_0 = -\tfrac{1}{2}\Delta_x \\ &\chi_M = \tfrac{1}{2}\Delta_x \ , \end{aligned} \tag{3.85}$$

with $m = 1, \ldots, M-1$. For the evaluation of $\phi_{m-\frac{1}{2}}$ and $\phi_{m+\frac{1}{2}}$ we use

$$\begin{aligned} &\phi_{m-\frac{1}{2}} = \phi(x_{m-\frac{1}{2}}) \approx \phi(\frac{x_{m-1} + x_m}{2}) \\ &\phi_{m+\frac{1}{2}} = \phi(x_{m+\frac{1}{2}}) \approx \phi(\frac{x_m + x_{m+1}}{2}) \ . \end{aligned} \tag{3.86}$$

The resulting set of $M-1$ equations in $M-1$ unknown variables is solved using Gauss elimination. After having determined the boundaries χ_m of the cells the pulse positions are found using an appropriate approximation.

The approach of Christov (1982) is the two-dimensional generalization of the previous boundary-value problem. Starting point is the set of equations

$$\begin{aligned} x_f y_g - y_f x_g &= \phi^{-1}(x,y) \\ x_f x_g + y_f y_g &= 0 \ , \end{aligned} \tag{3.87}$$

which can be easily derived from (and are equivalent to) (3.74) and (3.75). The boundary conditions are chosen such that the boundaries of $\mathbb{H}$ coincide with coordinate curves: $f(-\frac{1}{2}\Delta_x, y) = 0$, $f(\frac{1}{2}\Delta_x, y) = \sqrt{A}$, $g(x, -\frac{1}{2}\Delta_y) = 0$ and $g(x, \frac{1}{2}\Delta_y) = \sqrt{A}$. This way the entire domain $\mathbb{H}$ in the xy-domain is covered by the coordinate mesh and we have

$$\iint_{\mathbb{H}} \phi(x,y)\mathrm{d}x\mathrm{d}y = \int_0^{\sqrt{A}} \int_0^{\sqrt{A}} \mathrm{d}f\mathrm{d}g = A \ . \tag{3.88}$$

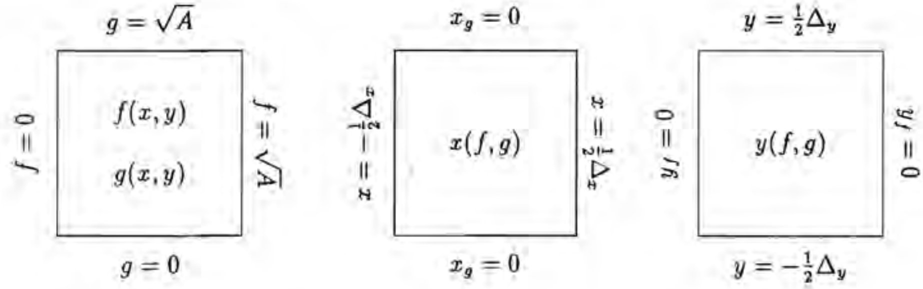


Figure 3.7: The boundary conditions for $x(f,g)$ and $y(f,g)$ are derived from the choice of the coordinate curves on the boundary in the xy-plane.

The value of the coordinate curves for the right and the upper boundary of $\mathbb{H}$ is arbitrary as long as the product equals the total area A. Our choice implies an equal number of pulses in the horizontal and vertical direction. Since the coordinate transformation is one-to-one the above choice for the coordinate curves at the boundary implies: $x(0,g) = -\frac{1}{2}\Delta_x$, $x(\sqrt{A},g) = \frac{1}{2}\Delta_x$, $y(f,0) = -\frac{1}{2}\Delta_y$ and $y(f,\sqrt{A}) = \frac{1}{2}\Delta_y$. Using (3.87) we find an additional set of 4 boundary conditions: $x_f(f,0) = 0$, $x_f(f,\sqrt{A}) = 0$ and $y_g(0,g) = 0$, $y_g(\sqrt{A},g) = 0$. In this way either the value or the derivative of both $x(f,g)$ and $y(f,g)$ is prescribed on the boundary in fg-domain. Figure 3.7 summarizes the derivation of boundary conditions.

In a straightforward way we can derive from the equations (3.87) the equivalent set of equations

$$\begin{aligned} \phi(x,y)(x_g^2 + y_g^2)x_f &= y_g \\ \phi(x,y)(x_g^2 + y_g^2)y_f &= -x_g \,. \end{aligned} \tag{3.89}$$

Note that this is a nonlinear generalization of the Cauchy-Riemann equations. An alternative (but not independent) set of equations is

$$\begin{aligned} \phi(x,y)(x_f^2 + y_f^2)y_g &= x_f \\ \phi(x,y)(x_f^2 + y_f^2)x_g &= -y_f \,. \end{aligned} \tag{3.90}$$

Next (3.89) and (3.90) are combined in order to achieve a form which is more convenient to solve numerically:

$$\begin{aligned} (\phi H_g^2 x_f)_f + (\phi H_f^2 x_g)_g &= 0 \\ (\phi H_g^2 y_f)_f + (\phi H_f^2 y_g)_g &= 0 \,. \end{aligned} \tag{3.91}$$

This result can be considered as the two-dimensional generalization of (3.83), where we now have the additional coordinate scaling factors $H_f^2 = x_f^2 + y_f^2$ and $H_g^2 = x_g^2 + y_g^2$.

In order to solve the problem stated by the set of equations (3.91) together with the boundary conditions, an approximation based on finite differences (similar to the one-dimensional case) is applied. The resulting set of equations is solved using the alternating direction implicit method (ADI) with Gauss elimination (Kreyszig, 1983). A set of 900 pulses obtained with Gauss' 1-point integration for the signal $\phi(x,y) = x^2 + y^2$ is shown in Figure 3.8. The density of the pulses is adapted to the local signal value of $\phi(x,y)$. When Figure 3.8 is viewed from a large distance (this is low-pass filtering) a radially increasing gray-tone level is observed. However, in contradiction with (3.75) the resulting coordinate curves for the input signal $\phi(x,y)$ are not orthogonal! In order to explain this we assume that an input signal, which is invariant under the exchange of the coordinates x and y, gives rise to a coordinate mesh possessing the same symmetry. (The coordinate curves are then related according to $f(x,y) = g(y,x)$). A closer inspection of the first quadrant in Figure 3.8 shows that in general an orthogonal coordinate mesh will not exist. The proposed method tries to minimize the orthogonality error, as can be shown by calculation of (3.75) during iteration.

Figure 3.8: Pulse density modulation for $\phi(x,y) = x^2 + y^2$, determined by numerically solving the generalized Cauchy-Riemann problem.

3.3.3 Discussion

One-dimensional pulse-density modulation based on a coordinate transformation can be generalized to the two-dimensional case. By means of the approach proposed by (Christov, 1982) a set of coordinate curves with a given density is determined. Although one of the basic assumptions of this method is the construction of an orthogonal mesh, the result turns out to be non-orthogonal. Since our objective is to obtain a coordinate mesh with a prescribed density (not necessarily orthogonal) we still feel this method is applicable.

While the one-dimensional pulse positions are determined independently, the problem has to be solved iteratively in the two-dimensional case. Due to the computational complexity, we have applied this method only for the determination of a small number of pulses for simple input functions. For hologram transmittances, where a large number of pulses is required in order to obtain a good approximation, our present implementation is too complex. In order to reduce the complexity one could think of gradually decreasing the step-size (increasing the number of pulses) during iteration.

3.4 Related topics

3.4.1 Introduction

In this section we discuss some techniques for clustering, which turn out to be related to pulse-density modulation. This relation is based on the consideration that when the positive input signal is normalized such that its total area equals unity we can regard the resulting $\phi(x,y)$ as the probability density function of the two-dimensional vector $\boldsymbol{x} = (x,y)^T$. From this point of view the Fourier transform $\Phi(u,v)$ equals the expectation of $e^{-i2\pi(ux+vy)}$:

$$\Phi(u,v) = \text{Exp}\left[e^{-i2\pi(ux+vy)}\right] = \iint \phi(x,y)e^{-i2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y \ , \tag{3.92}$$

also known as the characteristic function (Papoulis, 1965). Next, we consider a source which generates vectors $\boldsymbol{x}_n = (x_n, y_n)^T$ in accordance with the probability density function $\phi(x,y)$ defined on $\mathbb{H}$. We then find that the average

$$B(u,v) = \frac{1}{N}\sum_{n=1}^{N} e^{-i2\pi(ux_n+vy_n)} \tag{3.93}$$

equals the Fourier transform of the pulse-density signal $b(x,y)$ consisting of Dirac-pulses located at positions (x_n, y_n). In the limiting case $N \to \infty$ the average (3.93) approaches the ensemble average (3.92). Of course, finding a method to select vectors according to fixed probability density function is (for the continuous case, not for the discrete case) as difficult as the original pulse-density modulation problem.

While the above consideration requires a large number of vectors, we are particularly interested in a representation of the probability density function $\phi(x,y)$ with a finite number of 'representation' vectors $(x_m, y_m)^T$. To settle this problem a clustering of the generated vectors $(x_n, y_n)^T$ is introduced. This is briefly considered for vector quantization and the Kohonen network in the next sections. For a more detailed discussion and experimental results, see (Koppelaar, 1992).

3.4.2 Vector quantization

The mapping of a vector $\boldsymbol{x} \in \mathbb{D}$ onto a finite set of representation vectors $\boldsymbol{x}_m \in \mathbb{D}$ with $m = 1, \ldots, M$ is known as vector quantization. To this end the domain $\mathbb{D}$ is divided in M (connected) cells C_m, according to

$$\mathbb{D} = \bigcup_{m=1}^{M} C_m \text{ , where } C_n \cap C_m = \emptyset \text{ for } n \neq m \ . \tag{3.94}$$

All vectors within cell C_m are represented by $\boldsymbol{x}_m$. The mapping thus implies finding out to which cell a given input vector belongs. This leaves us the problem how to determine the optimal cells and their representation vectors.

The design of a non-uniform quantizer for the one-dimensional case, where the vectors become scalars and the cells become intervals, is due to Max (1960). In accordance with the notation of Section 3.2 the representation scalars are denoted by x_m, the interval boundaries by χ_m. Moreover, the probability density function is denoted by $\phi(x)$ defined on the domain $\mathbb{H}$. The optimal interval boundaries and representation vectors are determined by minimizing

$$\operatorname{Exp}[d(\hat{x}, x)] = \int d(\hat{x}, x)\phi(x)\mathrm{d}x \ , \tag{3.95}$$

with $d(\hat{x}, x)$ an appropriate distance function. In the one-dimensional case the mapping rule $x \rightarrow \hat{x}$ of the quantizer reads

$$\hat{x} = x_m \text{ if } \chi_{m-1} < x \leq \chi_m \ ; \ m = 1, \ldots, M \ . \tag{3.96}$$

In our case we have $\chi_0 = -\frac{1}{2}\Delta_x$ and $\chi_M = \frac{1}{2}\Delta_x$. It has been shown (Max, 1960) that for the (squared) Euclidean distance $d(\hat{x}, x) = |\hat{x} - x|^2$ the optimal solution satisfies

$$\begin{aligned} \chi_m &= \tfrac{1}{2}(x_m + x_{m+1}) \\ x_m &= \frac{\int_{\chi_{m-1}}^{\chi_m} x\phi(x)\mathrm{d}x}{\int_{\chi_{m-1}}^{\chi_m} \phi(x)\mathrm{d}x} \ . \end{aligned} \tag{3.97}$$

In order to determine the optimal solution an iterative algorithm is applied.

By means of (3.95) the quantization errors are given more weight in regions where $\phi(x)$ is large. As a result the optimal quantizer reserves more representation scalars x_m in regions where x is selected with a higher probability. Regarding the representation scalars as the positions of the pulses, the pulse density is thus adapted to the input signal $\phi(x)$. As for the first-order moment method of Subsection 3.2.4 the representation scalars are given as the center of gravity of each cell. Note, however, that the representation vectors are not equiprobable.

Of course, we can regard Gauss' 1-point integration method as the mapping rule of a scalar quantizer:

$$\hat{x} = x_m \text{ if } \frac{m-1}{M} < g(x) < \frac{m}{M} \ , \tag{3.98}$$

where $g(x)$ is defined according to (3.26). For a stochastic variable x with a probability density function $\phi(x)$, the quantizer output $\hat{x}$ will have a probability density function

$$b(x) = \frac{1}{M} \sum_{m=1}^{M} \delta(x - x_m) \ . \tag{3.99}$$

Obviously, the possible outputs of the quantizer are equiprobable and the quantizer can be regarded as a source with maximum entropy.

The design of a two-dimensional quantizer, i.e. a vector quantizer, is the generalization of the one-dimensional case. The optimal set of representation vectors minimizes

$$\mathrm{Exp}\,[d(\hat{\boldsymbol{x}}, \boldsymbol{x})] = \int d(\hat{\boldsymbol{x}}, \boldsymbol{x})\phi(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \ , \tag{3.100}$$

where the distance is assumed to be the (squared) Euclidean distance. Considering (3.97), the cells C_m are now defined according to

$$C_m = \{\boldsymbol{x} \in \mathbb{H} \mid d(\boldsymbol{x}_m, \boldsymbol{x}) \leq d(\boldsymbol{x}_n, \boldsymbol{x}),\ n \neq m\} \ . \tag{3.101}$$

The resulting cell-structure in Figure 3.9 is known as the Voronoi tessellation of the domain $\mathbb{H}$ in the two-dimensional plane (Voronoi, 1907). On the other hand, the representation vector is the center of gravity of the accompanying cell:

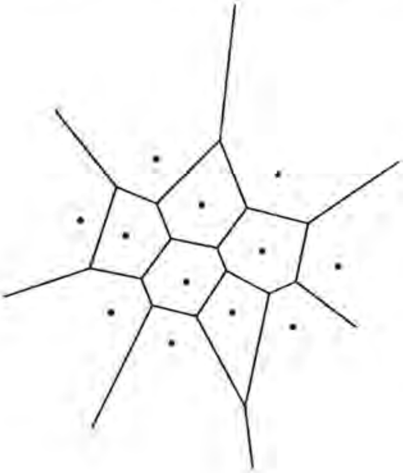


Figure 3.9: Voronoi tessellation of the two-dimensional plane. The representation vectors are shown as black dots.

$$x_m = \frac{\iint_{C_m} x\phi(x,y)\mathrm{d}x\mathrm{d}y}{\iint_{C_m} \phi(x,y)\mathrm{d}x\mathrm{d}y} \quad \text{and} \quad y_m = \frac{\iint_{C_m} y\phi(x,y)\mathrm{d}x\mathrm{d}y}{\iint_{C_m} \phi(x,y)\mathrm{d}x\mathrm{d}y} \ , \tag{3.102}$$

as follows from (3.97). Obviously, the representation vectors completely specify the cells (3.101), which in turn completely specify the representation vectors (3.102).

In contrast to the one-dimensional case an explicit mapping rule $\boldsymbol{x} \rightarrow \boldsymbol{x}_m$ can not be formulated. Instead, the representation vector $\boldsymbol{x}_m$ of a given vector $\boldsymbol{x}$ is found by means of

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_m \ \text{ if } \ d(\boldsymbol{x}_m, \boldsymbol{x}) \leq d(\boldsymbol{x}_n, \boldsymbol{x}) \ \text{ for all } n \neq m \ . \tag{3.103}$$

If a vector has equal distance to more than one decision vector (the boundary of a cell) we select the decision vector with the smallest index.

Based on the above considerations Linde, Buzo and Gray (1982) have proposed an iterative algorithm, known as the LBG-algorithm, in order to determine the optimal set of representation vectors. First, a training set of N vectors (with $N \gg M$) is generated in accordance with the probability function $\phi(x, y)$. In addition an initial set of representation vectors is assumed. For each training vector the accompanying representation vector is determined using (3.103), i.e. the training vectors are clustered. Next, the representation for each cluster (the set of vectors in the same cell) is determined by calculating the center of gravity. Given the new set of representation vectors the process is repeated (using the same set of training vectors) until a desirable solution is obtained.

We remark that in both the one-dimensional and the two-dimensional case the density of the obtained set of representation vectors (pulse positions) is matched to the probability density function (input signal). However, the representation vectors will have a non-equal probability

$$P[\boldsymbol{x}_m] = \int_{C_m} \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \ . \tag{3.104}$$

In terms of pulse-density modulation this means that the pulses have non-equal area.

3.4.3 Kohonen's neural network

The problem of determining a set of equiprobable representation vectors has been addressed by Kohonen (1984). This resulted in an algorithm which describes a specific type of neural network, known as the Kohonen neural network, which is briefly discussed.

A neural network (Figure 3.10) is built of a number of processing elements (often called neurons) arranged in a certain structure by means of connections. The connections maintain an instantaneous, uni-directional signal transport between the processing elements. Each processing element combines its incoming signals and generates one output signal, where the particular input-output relation is defined by a transfer function. In the neural network we can distinguish layers consisting of processing elements with the same transfer function.

Often, the first operation of the transfer function of a processing element is to weight and add the incoming signals. (Weights are stored in the processing element's local memory.) During a training session adaptation of the weights takes place according to a specific learning rule. In this way the network is given desired properties. For the Kohonen network, which belongs to the special class of networks capable of self-organization, the training is unsupervised. In this case only input signals are presented to the network. The network adapts its weights without any knowledge of desired output signals or without knowing how well it is performing, hence the name self-organizing network. The learning rule for such networks, proposed by Kohonen, is explained by means of the example shown in Figure 3.10. Two input signals x_1, x_2 are applied in the input layer, which distributes these signals unchanged to each of the M processing elements in the second layer. Processing element m $(m = 1, \ldots, M)$ weights its input signal x_i $(i = 1, 2)$ with w_{mi}. For convenience we introduce the weight vectors $\boldsymbol{w}_m = (w_{m1}, w_{m2})^T$ and the input vector $\boldsymbol{x} = (x_1, x_2)^T$. We assume that the input vector $\boldsymbol{x}$ is selected in accordance with a fixed probability density function $\rho(\boldsymbol{x})$. Next, each neuron in the upper layer calculates the Euclidean distance [6]

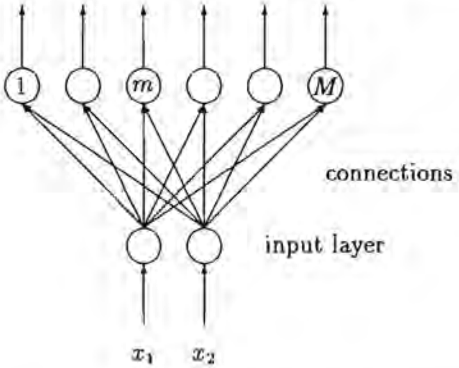


Figure 3.10: Architecture of the Kohonen neural network.

$$\mathrm{d}(\boldsymbol{w}_m, \boldsymbol{x}) = |\boldsymbol{w}_m - \boldsymbol{x}| \ . \tag{3.105}$$

The neuron with the smallest distance is called 'the winner', other neurons are 'the losers'. The main objective of Kohonen was to find a set of M equiprobable weight vectors, meaning that (given the probability density function ρ) each of the M neurons is selected as winner with equal probability $1/M$. To achieve this the following learning rule is proposed. During the training session a number of input vectors $\boldsymbol{x}$ are applied. For each input vector the processing element with the smallest distance (the winner) is allowed to adapt its weights according to

$$\boldsymbol{w}' = (1 - \alpha)\boldsymbol{w} + \alpha\boldsymbol{x} \ . \tag{3.106}$$

[6] Other distances can be used as well.

This equation states that the weight-vector $\boldsymbol{w}$ of the winner is shifted towards $\boldsymbol{x}$. The weights of the other neurons (the losers) remain unchanged. By slowly decreasing α to zero during the training the weight-vectors will converge to a stable configuration. For obvious reasons, this learning process is known as competitive learning. With the present learning law, however, equiprobability is achieved for specific probability density functions only. A solution for this problem is to build in a 'conscience mechanism' (Hecht-Nielsen, 1989). When a neuron is selected substantially more often than a fraction $1/M$ of the time it leaves the competition for a while, in order to give less fortunate neurons a chance to win.

At the end of the training the weight-vectors are distributed with a density proportional to the probability density function $\rho(\boldsymbol{x})$. Each neuron is selected with (approximately) the same probability. Consequently, if we generate input vectors $\boldsymbol{x} = (x, y)^T$ according to a fixed probability density function equal to the (normalized) input signal $\phi(x, y)$ the resulting distribution of the weight-vectors gives rise to the desired pulse-density signal $b(x, y)$. However, in order to train the network a large number of trials is necessary.

3.4.4 Discussion

In this section we have briefly discussed the relation between pulse-density modulation and vector quantization. For vector quantization the representation vectors act as the pulses. By means of the LBG-algorithm the optimal set of representation vectors is determined in order to minimize a well-defined function (3.100). However, since this object function seems incompatible with our desired object function (3.8), the applicability of the LBG-algorithm for pulse-density modulation is limited.

For the Kohonen network the weight vectors act as the pulses. Given a probability density function $\phi(x, y)$, the Kohonen network is able to distribute its weight vectors in such a configuration that each weight vector is equiprobable. The desired pulse-density signal is thus found as the probability density function of the weights. Provided that the number of weight vectors (pulses) is very large a good approximation can be obtained using the Kohonen network. We remark, however, that the Kohonen network is known to converge slowly.

3.5 From continuous to discrete pulse-density modulation

So far we have discussed pulse-density modulation without considering precision requirements. We did not restrict the pulse positions (except for overlap), hence the name continuous pulse-density modulation. Actual output devices, however, can place pulses with finite precision only. When this property is taken into account, the pulses are allowed to be placed on fixed positions only. In this case we have discrete pulse-density modulation.

Since discrete pulse-density modulation is a special case of continuous pulse-density modulation we can expect that discrete pulse-density modulation can be achieved by adjusting the known methods. For Gauss' 1-point integration method, the pulse positions were found by integrating the input signal $\phi(x)$ until the area h is reached. (For the first pulse the integration goes to $\frac{1}{2}h$.) Then a (Dirac) pulse with area h is placed and the process is repeated. With discrete pulse-density modulation we integrate $\phi(x)$ up to a certain position. If the area exceeds the threshold mh a pulse with area h is placed at this position. With the remaining area as starting-value the integration then proceeds till the next position with threshold $(m+1)h$. This results in an equal-area pulse-density signal consisting of Dirac-functions placed at fixed positions. We remark that this is not a discrete signal.

In general $\phi(x)$ is known in sampled form. This means that we have to interpolate the signal in order to generate the pulses. In the remainder of this section we consider a system, based on Gauss' 1-point integration formula that converts a discrete input signal directly to a binary output signal. For convenience we assume that $\phi(x)$ is defined on $x \in (-\infty, \infty)$. (This is in contrast with the finite-length assumption for the signals in the preceding of this chapter. Using infinite-length signals is merely a matter of notation, the derived results also hold for the original case.)

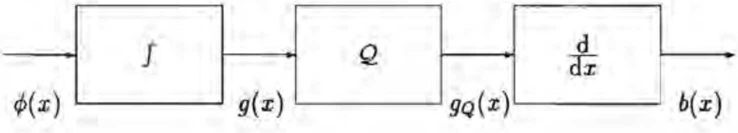


Figure 3.11: Equivalent block-diagram for Gauss' 1-point integration method.

An equivalent block-diagram for this integration method is shown in Figure 3.11. This system first integrates the input signal $\phi(x)$ resulting in

$$g(x) = \int_{-\infty}^{x} \phi(s)\mathrm{d}s \ , \tag{3.107}$$

which is the input for the quantizer $\mathcal{Q}$. The input-output relation of the quantizer, shown in Figure 3.12, is defined according to

$$\mathcal{Q}g = mh \ \text{ for } \ (m-\tfrac{1}{2})h \le g < (m+\tfrac{1}{2})h \ . \tag{3.108}$$

The quantization characteristic consists of a linear term and a remainder, according to $\mathcal{Q}g = g + \langle g \rangle$. The differentiation operator, which is the inverse operation of the integration, is applied to the quantized signal $g_Q(x) = \mathcal{Q}g(x)$. Since the step-size of the quantizer equals h, this results in the desired pulse train

$$b(x) = \frac{\mathrm{d}g_Q(x)}{\mathrm{d}x} \ . \tag{3.109}$$

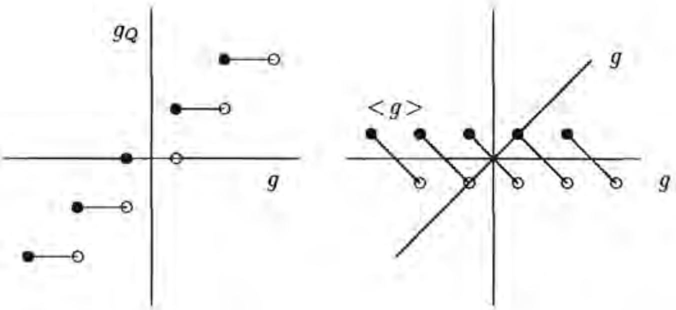


Figure 3.12: The input-output relation of the quantizer $\mathcal{Q}$ (shown on the left) consists of a linear term g and a remainder $\langle g\rangle$ (shown on the right).

In the limiting case $h \to 0$ the effect of the quantizer disappears and the output signal equals the input signal (after appropriate smoothing). This again shows that for an increasing average pulse-density the approximation error will decrease. Moreover, we conclude from Figure 3.11 that, since the quantizer represents an irreversible operation, exact reconstruction of the input signal from the output signal is impossible. (Except for the trivial case where $\phi(x)$ is constant.) According to Figure 3.12 the effect of the quantizer can be modeled as the addition of a signal $\langle g(x)\rangle$, that is, $g_Q(x) = g(x) + \langle g(x)\rangle$. As a consequence we find for the output signal $b(x) = \phi(x) + e(x)$ with $e(x) = \mathrm{d}\langle g(x)\rangle/\mathrm{d}x$. Since the differentiation operator acts as a high-pass filter the low-frequency content of $\langle g(x)\rangle$ is attenuated. This means that the low-frequency content of the pulse-density signal $b(x)$ (almost) equals the low-frequency content of the input signal $\phi(x)$.

Remark

It is possible to show that Figure 3.11 is the block-diagram for Gauss' 1-point integration method. To this end we first derive an equivalent expression for the pulse train

$$b(x) = h\sum_m \delta(x - x_m)\ . \tag{3.110}$$

Since each pulse position x_m is the (unique) solution of $g(x) = (m - \frac{1}{2})h$, we have (Bracewell, 1978)

$$\delta(g(x) - mh) = \frac{\delta(x - x_m)}{|g'(x_m)|} = \frac{\delta(x - x_m)}{\phi(x_m)}\ . \tag{3.111}$$

In combination with the sifting property (Bracewell, 1978) of the Dirac function we find

$$b(x) = h\phi(x)\sum_m \delta[g(x) - (m - \tfrac{1}{2})h]\ . \tag{3.112}$$

Next, using the scaling property $\delta[g-(m-\frac{1}{2})h] = h^{-1}\delta[g/h-(m-\frac{1}{2})]$ of the Dirac function and Poisson's summation formula (Bracewell, 1978) Equation (3.112) passes into

$$b(x) = \phi(x)\sum_{m}(-1)^m e^{-i2\pi\frac{mg(x)}{h}} . \tag{3.113}$$

(In this result we have used $e^{-i\pi m} = (-1)^m$.) This function is periodic in g (with period length h) and can therefore be written in a Fourier series. In a straightforward way the result

$$b(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left[g(x) + \sum_{m=1}^{\infty}\frac{(-1)^m h}{m\pi}\sin(2\pi\frac{mg(x)}{h})\right] \tag{3.114}$$

is found. The second term in (3.114) is the Fourier series (with coefficients $\frac{(-1)^m h}{\pi m}$) of the saw-tooth function $\langle g\rangle$ which together with the linear term g forms the quantization curve shown in Figure 3.12. In summary we have

$$b(x) = \frac{\mathrm{d}}{\mathrm{d}x}\mathcal{Q}\int_{-\infty}^{x}\phi(s)\mathrm{d}s , \tag{3.115}$$

this equation describes the system shown in Figure 3.11.

Figure 3.11 suggests replacement of the integrator and the differentiator by their discrete counterparts in order to acquire a system for discrete pulse-density modulation. The resulting system with a discrete input signal $\phi[n]$ and a discrete output signal $b[n]$ is shown in Figure 3.13. For the 'discrete integrator' we have taken the summation (Σ), with the transfer function $H(z) = (1-z^{-1})^{-1}$. In that case $g[n]$ is given by

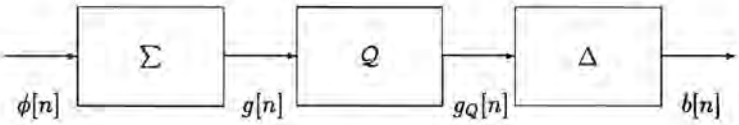


Figure 3.13: Block-diagram for discrete pulse-density modulation.

$$g[n] = \sum_{m=-\infty}^{n}\phi[m] . \tag{3.116}$$

Next, using the same quantizer as in the continuous case $g[n]$ is mapped to $g_Q[n]$. The differentiation is replaced by the first-order difference (Δ), which has a transfer function $H(z) = 1 - z^{-1}$. So, in the discrete case the last operator is the inverse of the first operator as well. For the output signal we find

$$b[n] = g_Q[n] - g_Q[n-1] \ . \tag{3.117}$$

Of course, the discrete system has to convert the input $\phi[n]$ to a binary-valued output $b[n]$. This is not guaranteed for an arbitrary value for the step-size h. However, for input signals with an amplitude range $[0,1]$ the increase in two successive samples of $g[n]$ cannot exceed unity. Taking the quantizer step-size equal to $h = 1$ results in an increase after quantization of either 0 or 1. The output signal is then binary-valued.

We remark the resemblance of Figure 3.13 with the class of coding systems known as analysis-synthesis systems. By means of an analysis operation (here: 'integration') a set of parameters is extracted from the input signal. In order to meet storage or transmission requirements the parameters are quantized and coded. After decoding the synthesis operation (here: 'differentiation') reconstructs an approximation of the input signal. An example of such a coding system is block transform coding.

We end this chapter with the derivation of an equivalent block-diagram for the discrete pulse-density modulation system. To this end we write for the output signal

$$\begin{aligned} b[n] &= g_Q[n] - g_Q[n-1] \\ &= \mathcal{Q}(g[n] - g_Q[n-1]) \ . \end{aligned} \tag{3.118}$$

This equation states that it makes no difference whether $g_Q[n-1]$ is subtracted before or after the quantization operation is applied. This is shown in Figure 3.14. The first block-diagram equals the block-diagram of Figure 3.13 with the summation and the finite difference shown in detail. In the second block-diagram $g_Q[n-1]$ is subtracted before quantization. The last block-diagram follows when both delay operators z^{-1} are combined. According to (3.118) the input of the quantizer reads $g[n] - g_Q[n-1]$, which can be written as $\phi[n] + g[n-1] - g_Q[n-1] = \phi[n] - \langle g[n] \rangle$. Under the assumption that $0 \leq \phi[n] \leq 1$ and $h = 1$ this implies that the amplitude-range of the input of the quantizer is $[-0.5, 1.5)$. As a result we are allowed to replace the quantizer by a threshold-device with an input-output relation

$$\mathcal{Q}s = \begin{cases} 1 & s \geq 0.5 \\ 0 & s < 0.5 \end{cases} \ . \tag{3.119}$$

The resulting system is then known as first-order (one-dimensional) error diffusion, which in a more general form is the subject of the next chapter.

3.6 Discussion

In this chapter we have discussed continuous pulse-density modulation, where the positions of the pulses in the approximating signal are not restricted to fixed positions. Finding the optimal set of pulses is a multidimensional optimization problem,

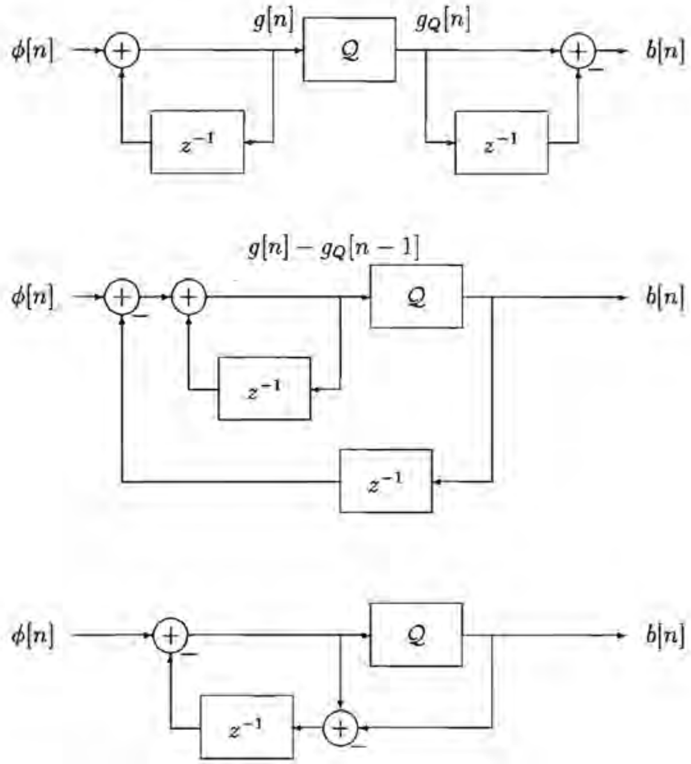


Figure 3.14: Derivation of an equivalent block-diagram for discrete-pulse density modulation. The result is known as first-order error-diffusion.

which can be solved by means of gradient search techniques. However, for a large number of pulses such an approach requires much computation.

Based on the considerations in (Eschbach and Hauck, 1987) we have discussed a simpler approach for the one-dimensional case. The input signal $\phi(x)$ is divided in partial signals with equal area, each approximated by one pulse with the same area. By combining several adjacent partial signals to be represented by a number of pulses, we developed several methods (all based on a coordinate transformation) to determine the position of the whole group. Although this approach does not lead to the optimal set of pulses, a good approximation can be obtained for a large pulse density.

Generalization of this concept to the two-dimensional case implies dividing the two-dimensional domain in cells of equal-area parts of the input signal $\phi(x, y)$. When the input signal is known in discrete form this problem is solved by means of a clustering of the signal samples. In general this is done recursively (Eschbach and Hauck, 1987; Koppelaar, 1992). In this chapter we have considered an approach which is based on a two-dimensional coordinate transformation (Jacobian). Given a continuous input function, a set of pulses is then determined following the method for the generation of coordinate meshes with manageable density, as proposed by Christov (1982). Although the results obtained for simple input functions are promising, a numerical implementation with reduced complexity has to be found in order to make this method feasible for a large number of pulses.

Finally, we have considered how (one-dimensional) continuous pulse-density modulation can be adapted for discrete signals. The resulting discrete method for pulse-density modulation is also known as first-order error diffusion. In the next chapter both higher-order one-dimensional and two-dimensional error diffusion are considered in more detail.

Chapter 4

Error diffusion

4.1 Introduction

The previous chapter was concluded with a recursive realization of discrete pulse-density modulation for one-dimensional signals. This system, which is also known as (first-order) error diffusion, transforms a discrete input signal $\phi[n]$ into a binary output signal $b[n]$. By means of a feedback loop the decision of the quantizer is influenced by errors caused by the quantization of previous signal samples. The result is (just as with the original continuous pulse-density modulation system in Figure 3.11) a high-pass filtering of the error caused by the quantizer. We remark that first-order error diffusion is also known in literature as $\Sigma\Delta$-modulation or $\Delta\Sigma$-modulation (Candy and Temes, 1992). This relation was first noted by Anastassiou (1989).

Two-dimensional error diffusion was originally introduced on a heuristic basis by Floyd and Steinberg (1976) for the transformation of digital gray-tone images into binary images (half-toning). The idea is to diffuse the error (introduced by the quantizer) to neighbouring samples (pixels) that have not been quantized yet. In this way binary images are obtained that resemble the original gray-tone images quite well. Error diffusion was first applied to the quantization of amplitude holograms by Hauck and Bryngdahl (1984). Later error diffusion has also been applied for the quantization of multilevel phase holograms (Weissbach et al., 1989).

With first-order error diffusion the unit delay z^{-1} is applied in the feedback loop. In the next section we consider error-diffusion systems with a more general feedback filter. Although our main interest is in two-dimensional error diffusion, we first restrict ourselves to the one-dimensional case. In this way a number of specifically two-dimensional problems are avoided in a first discussion. Moreover, one-dimensional error diffusion can be applied in the two-dimensional case as a suboptimal solution. Due to its recursive nature, the error-diffusion system can become unstable. In order to prevent such a situation, stability conditions are considered in Section 4.2. Next, a linear model is introduced in order to derive a relation between the feedback filter and the filtering of the error introduced by the quantizer. Based on this model a method for designing stable error-diffusion systems with desired filtering properties is proposed.

In Section 4.3 the one-dimensional theory of Section 4.2 is generalized for two dimensions. Some feedback filters designed for the quantization of computer-generated holograms are considered in Section 4.4. In Section 4.5 symmetrical error diffusion is introduced. We remark that the methods for the design of feedback filters for the calculation of binary amplitude and binary phase holograms as considered in this chapter are also applicable for multilevel phase holograms.

4.2 One-dimensional error diffusion

4.2.1 Internal and external error diffusion

In Figure 4.1 we show the block-diagram for error diffusion, where the unit delay in the feedback loop is replaced by a more general shift-invariant linear system. For symmetry reasons, we take the input signal $\phi[n]$ of this system bipolar (with $\max|\phi[n]| = 1$). The error-diffusion system transforms the input signal into the output signal $b[n] \in \{-1, 1\}$ which can be considered the (discrete) transmittance function of a binary phase hologram. The characteristic of the quantizer is given by

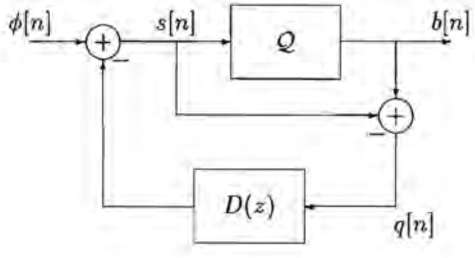


Figure 4.1: Block-diagram for one-dimensional (internal) error diffusion.

$$Q(s) = \begin{cases} +1 & \text{if } s \geq 0 \\ -1 & \text{if } s < 0 \end{cases}, \tag{4.1}$$

where $s[n]$ denotes the input signal of the quantizer. At every instant n the quantizer introduces a 'quantizer' error $q[n]$. By means of the feedback loop previous quantizer errors are taken into account in the decision of the quantizer. While for first-order error diffusion only the last quantizer error (with weight 1) is considered, the system in Figure 4.1 considers (at instant n) the weighted sum

$$\sum_{m=1}^{M} d[m]q[n-m] = d[n] * q[n] \,. \tag{4.2}$$

The error-diffusion coefficients $d[n]$ form the impulse response of a linear filter, referred to as the error-feedback filter. According to Figure 4.1 the error-diffusion system is described by the set of equations

$$\begin{aligned} s[n] &= \phi[n] - d[n] * q[n] \\ b[n] &= \mathcal{Q}s[n] \\ q[n] &= b[n] - s[n] \,. \end{aligned} \tag{4.3}$$

We assume that the error-feedback filter contains at least one elementary delay, in order to make a recursive realization possible (no delay-free loops). Moreover, we assume that the filter has an impulse response of finite length M (FIR-filter). Hence we have $d[n] = 0$ for $n \leq 0$ and $n > M$.

Instead of the *quantizer* error $q[n]$ we can also feed the *quantization* error $e[n] = b[n] - \phi[n]$ back to the input of the quantizer. This gives rise to the system shown in Figure 4.2. In this case the quantization error $e[n]$ is the input of the error-feedback filter with impulse response $c[n]$. For this filter the previous considerations (recursive computability, FIR) hold. Now, the error-diffusion system is described by the set of equations

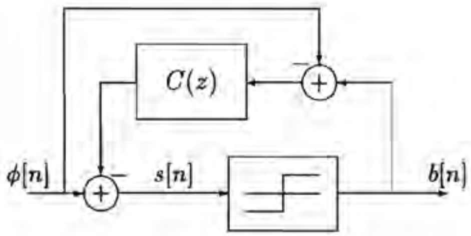


Figure 4.2: Block-diagram for one-dimensional (external) error diffusion.

$$\begin{aligned} s[n] &= \phi[n] - c[n] * e[n] \\ b[n] &= \mathcal{Q}s[n] \\ e[n] &= b[n] - \phi[n] \,. \end{aligned} \tag{4.4}$$

Note that in the first system the input signal of the feedback filter is an internal signal. For this reason we will refer to this system as internal error diffusion. In the second case the input signal of the feedback filter is formed at the terminals, hence the name external error diffusion.

Both systems consist of a linear part and a nonlinear element. In the study of linear systems the z-transformation has proved to be useful. The z-transform of the impulse response of a linear system is called the system function or transfer function. The system function of the feedback filter in the internal error-diffusion system is found according to

$$D(z) = \sum_{n=1}^{M} d[n] z^{-n} . \tag{4.5}$$

In the same way we introduce the system function $C(z)$ of the feedback filter in the external error-diffusion system. The two systems with internal and external error diffusion turn out to be equivalent provided that $C(z)$ and $D(z)$ are related according to

$$C(z) = \frac{D(z)}{1 - D(z)} , \tag{4.6}$$

which is equivalent with

$$D(z) = \frac{C(z)}{1 + C(z)} . \tag{4.7}$$

Example

For first-order error diffusion the system function of the error-feedback filter equals $D(z) = z^{-1}$. According to (4.6) we find for the system function of the external error-diffusion system

$$C(z) = \frac{z^{-1}}{1 - z^{-1}} . \tag{4.8}$$

This is the system function of a discrete integrator (delayed over one sample). Obviously, first-order error diffusion takes all previous quantization errors $e[n]$ with equal weight into account.

We remark that an FIR-filter $D(z)$ is transformed into a filter $C(z)$ with an impulse response of infinite length (IIR-filter) and vice versa. The general rational function $C(z)$, containing both poles and zeros, can obviously be realized by a combination of the two structures (Figure 4.1 and Figure 4.2) merely using FIR-filters, as shown in Figure 4.3.

4.2.2 Stability

Due to the feedback loop the error-diffusion system can become unstable. The decision of the quantizer is then independent of the input signal applied to the system. In order to prevent such a situation we at least have to require that the

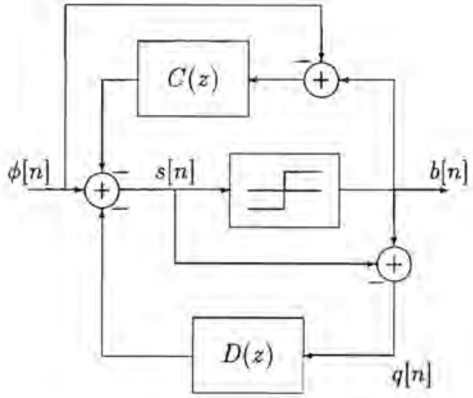


Figure 4.3: Block-diagram for one-dimensional (general) error diffusion.

input of the quantizer $s[n]$ can not grow without bound. To this end we consider the z-transform

$$S(z) = \sum_n s[n] z^{-n} \; . \tag{4.9}$$

In the same way the z-transforms $\Phi(z)$, $B(z)$ and $Q(z)$ are defined. We remark that for finite length sequences the z-transform is defined in the entire z-plane. Considering the block-diagram for general error-diffusion (Figure 4.3) we find for the input of the quantizer

$$S(z) = \frac{1 + C(z)}{1 - D(z)} \Phi(z) - \frac{C(z) + D(z)}{1 - D(z)} B(z) \; . \tag{4.10}$$

The input of the quantizer can not grow exponentially if the zeros of the denominator $1 - D(z)$ in (4.10) are within the unit circle, that is

$$1 - D(z) \neq 0 \quad \text{for} \quad |z| > 1 \; . \tag{4.11}$$

This condition guarantees that the linear part of the error-diffusion system is BIBO-stable (bounded-input bounded-output). However, when the zeros of $1 - D(z)$ are inside but close to the unit circle, the amplitude of $|s[n]|$ can become quite large, resulting in an input-independent behaviour. Condition (4.11) is thus a necessary but not a sufficient condition for an error-diffusion system to have desirable quantization properties.

Remark

Broja et al. (1986) have formulated a sufficient condition for a stable internal error-diffusion system ($C(z) = 0$):

$$\sum_{m=1}^{M} |d[m]| \leq 1 \ . \tag{4.12}$$

This condition is found by requiring that at some instant n the internal signal $s[n]$ is bounded by $|s[n]| \leq 1+c$ if all previous s satisfy the same bound. With $c \geq 1$ the quantizer error is bounded according to $|q| \leq c$ and therefore we have

$$\begin{aligned} |s[n]| &= |\phi[n] + \sum_{m=1}^{M} d[m]q[n-m]| \\ &\leq |\phi[n]| + \sum_{m=1}^{M} |d[m]||q[n-m]| \\ &\leq 1 + c\sum_{m=1}^{M} |d[m]| \ . \end{aligned} \tag{4.13}$$

Stability condition (4.12) is thus sufficient to keep the internal signal $s[n]$ bounded in amplitude.

Under this stability condition the roots of $1 - D(z)$ are within the unit circle. Outside the unit circle, that is for $|z| > 1$, we have

$$|D(z)| = |\sum_{m=1}^{M} d[m]z^{-m}| \leq \sum_{m=1}^{M} |d[m]||z|^{-m} < \sum_{m=1}^{M} |d[m]| \ . \tag{4.14}$$

Stability condition (4.12) implies that $|D(z)| < 1$ for $|z| > 1$ and therefore $1 - D(z) = 0$ does not have solutions outside the unit circle. The converse, however, is not true. Equation (4.12) is a very severe and most likely not necessary condition. For large filter-orders M the set of allowed diffusion coefficients in the coefficient space becomes very small, which reduces the freedom in the choice of the (internal) diffusion coefficients considerably.

According to the block-diagram in Figure 4.3 the input of the FIR filter $C(z)$ is bounded, resulting in a bounded contribution to the input of the quantizer. A constraint for the external diffusion coefficients $c[n]$ concerning the maximal contribution to $s[n]$ can be derived in a way similar to (4.13). A less severe constraint for the external diffusion coefficients is

$$1 + C(z) \neq 0 \quad \text{for} \quad |z| > 1 \ . \tag{4.15}$$

This condition is found by requiring BIBO-stability when the quantizer is replaced by a simple through connection. When the quantizer is placed back the feedback signal $b[n]$ is bounded. Since the quantizer increases the energy in the feedback signal only if $|s| < 1$, it is fair to assume that the quantizer input will not become substantially larger.

We remark that stability in the BIBO-sense does not imply that the error-diffusion system is free of limit cycles. Actually, the presence of limit cycles is exploited in the quantization of the input signal. With zero input the first-order (internal) error-diffusion system generates the output sequence $\ldots 1, -1, 1, -1 \ldots$. This way the average of the output signal equals the input signal.

4.2.3 A linear model for error diffusion

Due to the presence of the quantizer in the error-diffusion system the input-output relation is nonlinear. In order to understand the quantization effects, a linear model for error diffusion is introduced by regarding the effect of the quantizer as the addition of an external signal $q[n]$. In this way we obtain for the *general* error-diffusion system (consisting of internal *and* external error diffusion) the linear system shown in Figure 4.4. The input-output relation for the linear model reads

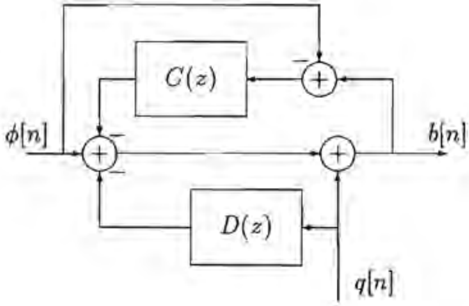


Figure 4.4: Linear model for error diffusion.

$$B(z) = \Phi(z) + \frac{1 - D(z)}{1 + C(z)} Q(z) \ . \tag{4.16}$$

Considering the z-transform $E(z)$ of the quantization error $e[n] = b[n] - \phi[n]$ we thus find that the quantization error $e[n]$ is related to the quantizer error $q[n]$ according to

$$E(z) = \frac{1 - D(z)}{1 + C(z)} Q(z) = H(z) Q(z) \ . \tag{4.17}$$

For computer-generated Fourier holograms we are particularly interested in the deviation between the Fourier transforms $B_d(\theta)$ and $\Phi_d(\theta)$. For z on the unit circle $|z| = 1$ the z-transform passes into the Fourier transform:

$$\Phi_d(\theta) = \Phi(e^{i2\pi\theta}) = \sum_{n \in (N)} \phi[n] e^{-i2\pi\theta n} \ , \tag{4.18}$$

with the summation over $n = -\frac{1}{2}N - 1, \ldots, \frac{1}{2}N$, denoted by $\langle N \rangle$. According to (4.16) the Fourier transform of the binary hologram thus consists of two parts: the original object $\Phi_d(\theta)$ and the Fourier transform of the quantizer error multiplied by the transfer function

$$H(e^{i2\pi\theta}) = \frac{1 - D(e^{i2\pi\theta})}{1 + C(e^{i2\pi\theta})} \tag{4.19}$$

The original object $\Phi_d(e^{i2\pi\theta})$ appears in the window $\mathbb{F}$ which, due to the assumed oversampling (cf. p. 10), constitutes a small part of the fundamental interval $-\frac{1}{2} < \theta \leq \frac{1}{2}$. This means that by choosing appropriate feedback filters we are able to shape the Fourier transform of the quantization error and lower the contribution in $\mathbb{F}$. This result is obtained at the expense of an increase of the noise contribution outside $\mathbb{F}$. For obvious reasons we call $H(z)$ the noise shaping transfer function.

Example

In the case of first-order internal error diffusion ($C = 0$) we find for the noise shaping transfer function

$$H(z) = 1 - D(z) = 1 - z^{-1} = \frac{z-1}{z} . \tag{4.20}$$

Due to the zero for $z = 1$ the quantization error will have a small contribution at low frequencies $\theta \approx 0$. This is shown in more detail by considering $z = e^{i2\pi\theta}$. The resulting amplitude transfer function

$$|H(e^{i2\pi\theta})|^2 = |1 - D(e^{i2\pi\theta})|^2 = 2 - 2\cos(2\pi\theta) , \tag{4.21}$$

is shown in Figure 4.5. When we take the object window $\mathbb{F}$ in the vicinity of the origin, we obtain with this error-diffusion system a (one-dimensional) binary hologram with quantization noise in the reconstruction that is substantially lower than for a hologram obtained without error diffusion ($D = 0$).

In order to design an appropriate filter we first have to consider the properties of the quantization error (cf. (4.17)). An overview on the analysis of quantization effects in digital filters is given by Butterweck et al. (1988). Modeling the action of a quantizer by the introduction of a noise source is a well-known technique in the study of digital systems (Oppenheim and Schafer, 1975). Often, the noise is assumed to be uncorrelated with the quantizer's input signal and to have a flat power spectrum (white noise). Such an assumption is justified if the amplitude range of the quantizer's input is covered by a large number of quantization levels. Moreover, the fluctuation in the input of the quantizer is supposed to be of such a nature that the difference in two consecutive signal samples is large compared to the step-size of the quantizer's characteristic.

In the case of quantizing Fourier holograms neither of the two conditions is satisfied. Not only has the quantizer merely two levels but also the input signal

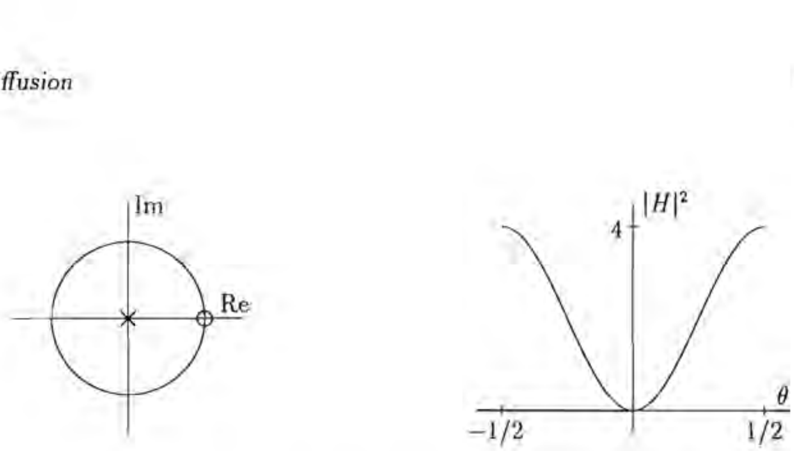


Figure 4.5: a. Zero-pole plot of first-order one-dimensional internal error diffusion. b. Squared modulus of the transfer function.

of the error-diffusion system (and therefore the input of the quantizer) is smooth due to the oversampling. Still, we adopt the above assumption about the quantizer error. Stated otherwise, we assume that the sequence $q[n]$ consists of N consecutive samples of a white random process with expectation

$$\mathrm{Exp}\,[q[n]] = 0 \tag{4.22}$$

and autocorrelation

$$r_{qq}[n,m] = \mathrm{Exp}\,[q[n]q[m]] = \sigma_q^2\delta[n-m] \ , \tag{4.23}$$

where σ_q^2 denotes the variance. The (squared) amplitude of the Fourier transform

$$Q(e^{i2\pi\theta}) = \sum_{n\in(N)} q[n]e^{-i2\pi\theta n} \tag{4.24}$$

is known as periodogram (Oppenheim, Schafer, 1975). Considering the expectation of the (squared) amplitude

$$\mathrm{Exp}\left[|Q(e^{i2\pi\theta})|^2\right] = \sum_{n\in(N)}\sum_{m\in(N)} \mathrm{Exp}\,[q[n]q[m]]\,e^{-i2\pi\theta(n-m)} = N\sigma_q^2 \ , \tag{4.25}$$

we find that the ensemble average is proportional to the power spectrum (the Fourier transform of the autocorrelation function) of the random process. In a single periodogram large fluctuations are superimposed on the expectation, and only after appropriate smoothing $|Q(e^{i2\pi\theta})|^2$ becomes $N\sigma_q^2$. So, under the assumption that the quantization error has a flat amplitude spectrum, the distribution of the quantization error in the reconstruction plane is (after appropriate smoothing) proportional to $|H(e^{i2\pi\theta})|^2$.

By means of an appropriate choice of the feedback filters $D(z)$ and $C(z)$ a desired distribution of the quantization error can be approximately realized. One approach to design the feedback filters is to minimize the (expected) total contribution of the quantization error in the object window $\mathbb{F}$

$$\mathrm{Exp}\,[P] = N\sigma_q^2 \int_{\mathbb{F}} \frac{|1 - D(e^{i2\pi\theta})|^2}{|1 + C(e^{i2\pi\theta})|^2} \mathrm{d}\theta \,. \tag{4.26}$$

In the special case of internal error diffusion only ($C(z) = 0$), determining the optimal diffusion coefficients turns out to be a linear problem. For internal error diffusion we minimize

$$\mathrm{Exp}\,[P] = N\sigma_q^2 \int_{\langle 1 \rangle} |A(\theta)|^2 |1 - D(e^{i2\pi\theta})|^2 \mathrm{d}\theta \,, \tag{4.27}$$

where $|A(\theta)|^2$ is defined on the fundamental interval $-\frac{1}{2} < \theta \leq \frac{1}{2}$ (denoted by $\langle 1 \rangle$) according to

$$|A(\theta)|^2 = \begin{cases} 1 & \theta \in \mathbb{F} \cup \mathbb{F}^* \\ a_s^2 & \text{elsewhere} \end{cases} \,. \tag{4.28}$$

The role of the small constant a_s, which equals 0 in (4.26) will be explained later. By means of Parseval's theorem we can write for (4.27) in the spatial domain

$$\mathrm{Exp}\,[P] = N\sigma_q^2 \sum_n (a[n] * (\delta[n] - d[n]))^2 =$$

$$N\sigma_q^2 \sum_n \left(a[n] - \sum_{m=1}^{M} d[m] a[n-m] \right)^2 \,. \tag{4.29}$$

The discrete signal $a[n]$ is found as the inverse Fourier transform of the $A(\theta)$. We remark that since the phase of $A(\theta)$ is undetermined (only $|A|^2$ is prescribed), the discrete signal $a[n]$ is not known. However, $a[n]$ serves merely as an intermediate signal in the derivation of an appropriate expression for the noise power. Apart from the constant $N\sigma_q^2$ the right-hand term of (4.29) is a quadratic form in the diffusion coefficients $d[m]$ and reads more specifically

$$\sum_{m=1}^{M} \sum_{l=1}^{M} d[m]d[l] \sum_n a[n-m]a[n-l] - 2 \sum_{m=1}^{M} d[m] \sum_n a[n]a[n-m] + \sum_n a^2[n] \,. \tag{4.30}$$

In this equation we recognize the (deterministic) autocorrelation $r[n] = a[n] \star a[n]$:

$$r[m] = \sum_n a[n]a[n-m] \,. \tag{4.31}$$

In this way the quadratic expression can be simplified according to

$$\sum_{m=1}^{M} \sum_{l=1}^{M} d[m]d[l]r[m-l] - 2 \sum_{m=1}^{M} d[m]r[m] + r[0] = \boldsymbol{d}^T R \boldsymbol{d} - 2\boldsymbol{r}^T \boldsymbol{d} + r[0] \,, \tag{4.32}$$

where we have introduced the coefficient vector $\boldsymbol{d} = (d[1], \ldots, d[M])^T$, the vector $\boldsymbol{r} = (r[1], \ldots, r[M])^T$ and the Toeplitz matrix R with elements $R_{ml} = r[m-l]$. Since the autocorrelation $r[n]$ is the inverse Fourier transform of $|A(\theta)|^2$ the coefficients are easily determined using

$$r[n] = \int_{\langle 1 \rangle} |A(\theta)|^2 e^{-i2\pi n\theta} d\theta = \int_{\langle 1 \rangle} |A(\theta)|^2 \cos(2\pi n\theta) d\theta \ . \tag{4.33}$$

The last equation follows from the fact that $|A(\theta)|^2 = |A(-\theta)|^2$. Rewriting (4.32) into

$$(\boldsymbol{d} - R^{-1}\boldsymbol{r})^T R(\boldsymbol{d} - R^{-1}\boldsymbol{r}) + r[0] - \boldsymbol{r}^T R^{-1}\boldsymbol{r} \ , \tag{4.34}$$

we find that the minimum of (4.27) occurs for

$$\boldsymbol{d} = R^{-1}\boldsymbol{r} \ . \tag{4.35}$$

The matrix R is positive definite (Haykin, 1987), hence the solution exists and is unique. Equation (4.35) is similar to the Wiener-Hopf equation (Haykin, 1987). We remark that this result is also derived in a different context by Laakso and Hartimo (1992). With the optimal set of diffusion coefficients the expected noise power equals

$$\mathrm{Exp}\,[P] = N\sigma_q^2 \left(r[0] - \boldsymbol{r}^T \boldsymbol{d}\right) \ . \tag{4.36}$$

Actually, the design of an optimal error-feedback filter is a constrained optimization problem. We have to minimize the quadratic object function (4.27) subject to the constraint that the zeros of $1 - D(z)$ are within the unit circle $|z| = 1$. However, designing an error-feedback filter by minimizing (4.27) has the nice property that the solution is automatically stable. This is due to the fact that an unstable solution for which $1 - D(z)$ vanishes for some z outside the unit circle can never minimize (4.27). To understand this we factorize $1 - D(z)$ in the form

$$1 - D(z) = \prod_{m=1}^{M} \left(1 - z_m z^{-1}\right) \ , \tag{4.37}$$

with $z_m, m = 1 \ldots M$, the zeros of $1 - D(z)$. In (4.27) we need

$$|1 - D(e^{i2\pi\theta})|^2 = \prod_{m=1}^{M} |1 - z_m e^{i2\pi\theta}|^2 \ . \tag{4.38}$$

Now imagine that some zero z_m is outside the unit circle. The corresponding factor in (4.38) then satisfies

$$\begin{aligned} |1 - z_m e^{i2\pi\theta}| = |z_m||z_m^{-1} - e^{i2\pi\theta}| = |z_m||e^{-i2\pi\theta} - z_m^{-1*}| = \\ |z_m||1 - z_m^{-1*} e^{i2\pi\theta}| > |1 - z_m^{-1*} e^{i2\pi\theta}| \ . \end{aligned} \tag{4.39}$$

This means that the 'mirrored' zero z_m^{-1*} inside the unit circle leads to a smaller factor and hence to a smaller transfer function in (4.38). Thus a noise shaper $1 - D(z)$ with zeros outside the unit circle assumes larger values on the unit circle than its counterpart with the zeros 'outside' replaced by mirrored zeros 'inside'.

Example

As an example we have determined the optimal diffusion coefficients for an internal error-diffusion system with a feedback filter of order $M = 20$. For the object window $\mathbb{F}$ we have taken $1/16 < \theta \leq 3/16$, the constant a_s equals 0.01. After having calculated the correlation coefficients, the optimal diffusion coefficients are determined using (4.35). According to Figure 4.6a, showing the roots of $1 - D(z)$, the linear part of the error-diffusion system is stable. With the obtained set of diffusion coefficients a test-signal has been quantized. The Fourier transform of the test-signal equals zero for frequencies outside $\mathbb{F}$. Inside the object window the amplitude is constant while the phase has a random distribution. In Figure 4.6b the Fourier transform of the quantization error $e[n]$ is shown, together with the expected distribution of the transformed quantization error. Obviously, the contribution of the quantization error is lowered within the object window at the expense of a large contribution outside the window. The noise shaping characteristic describes the distribution well, but large fluctuations occur in the periodogram.

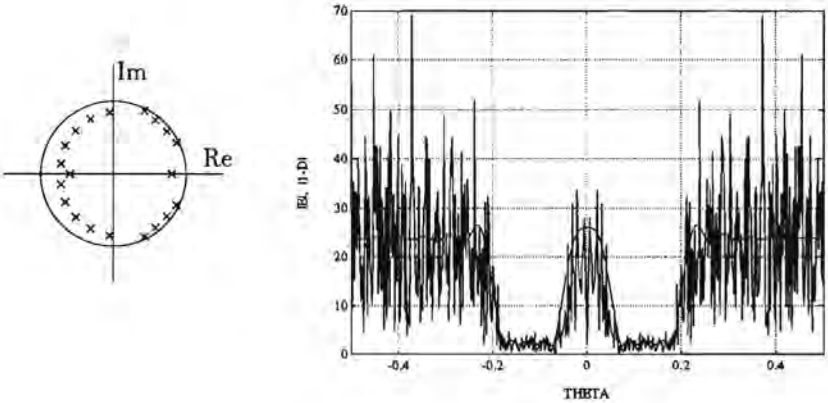


Figure 4.6: a. Roots of $1 - D(z)$ for an optimal set of diffusion coefficients. b. Expected distribution of transformed quantization error and computer simulation result.

With the error-diffusion system in the above example a desired noise-shaping was achieved. However, if the constant a_s is taken too small the roots of $1 - D(z)$ are very close to (but inside) the unit circle. In that case the linear part of the error-diffusion system is still stable, but the input of the quantizer becomes very large and the system will oscillate. In order to find a sufficiently large parameter a_s, a number of computer simulations have to be carried out.

Finding the optimal external diffusion coefficients $c[n]$ in (4.26) is a nonlinear problem and not as easy to solve as the linear problem for the internal diffusion coefficients.

4.3 Two-dimensional error diffusion

4.3.1 Two-dimensional recursive systems

In this section we discuss error diffusion for two-dimensional signals. The block-diagram for two-dimensional general error diffusion is shown in Figure 4.7. Although this diagram seems a straightforward generalization of its one-dimensional counterpart in Figure 4.3, there are some major differences between one-dimensional and two-dimensional error diffusion, which are considered in this section. The two-dimensional discrete input signal $\phi[n_1, n_2]$ (with $\max \phi[n_1, n_2] = 1$) is transformed into a binary output signal $b[n_1, n_2]$. Again, by means of the error-feedback filters 'previous' quantizer and quantization errors are taken into account in the decision of the quantizer [1]. This is described by the set of equations

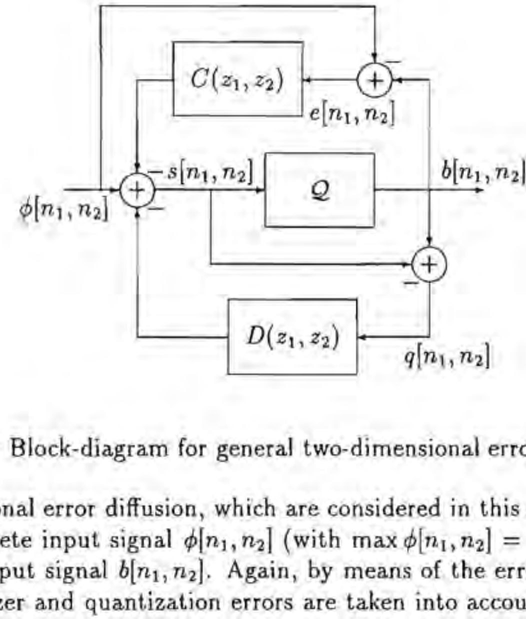


Figure 4.7: Block-diagram for general two-dimensional error diffusion.

$$\begin{aligned} s[n_1, n_2] &= \phi[n_1, n_2] - d[n_1, n_2] * q[n_1, n_2] - c[n_1, n_2] * e[n_1, n_2] \\ b[n_1, n_2] &= \mathcal{Q} s[n_1, n_2] \\ q[n_1, n_2] &= b[n_1, n_2] - s[n_1, n_2] \qquad (4.40) \end{aligned}$$

[1] The concepts of two-dimensional ordering as well as of past and future are discussed further down.

$$e[n_1, n_2] \;=\; b[n_1, n_2] - \phi[n_1, n_2] \,,$$

with $d[n_1, n_2]$ and $c[n_1, n_2]$ the impulse response of the internal and the external error feedback, respectively. In the one-dimensional case we required the error-feedback filters to be causal (with at least one elementary delay) in order to have a recursive computable system. In the two-dimensional case a closer look on recursive computability is appropriate.

From (4.40) we derive the *nonlinear* difference equation

$$s[n_1, n_2] = \phi[n_1, n_2] - d[n_1, n_2] * (Qs[n_1, n_2] - s[n_1, n_2]) \\ -c[n_1, n_2] * (Qs[n_1, n_2] - \phi[n_1, n_2]) \quad (4.41)$$

for the error-diffusion system. A general theory for *linear* shift-invariant two-dimensional systems governed by a difference equation is discussed by Lim (1990). The difference equation is regarded as a computational procedure and is said to be recursively computable when there exists a path we can follow in computing every output sample recursively, one sample at a time. With the *nonlinear* error-diffusion system we have a local nonlinear operation Q embedded in a linear shift-invariant two-dimensional system, which we require to be recursively computable. Due to the local character of the quantizer this requirement automatically gives rise to a recursive computable error-diffusion system.

We assume that both feedback filters have an impulse response whose nonzero values are in a particular region, called the support of the feedback filters. In order to make recursive computation possible, the feedback filters are restricted to have 'wedge support' (Lim, 1990). This means that the support is bounded by two lines emanating from the origin, where the angle between the lines is less than 180°. An example of a system with wedge support is shown in Figure 4.8. The support of the system is indicated by the solid dots (●) . We remark that in contrast with the original definition of wedge support the origin is not included in the support of the filter.

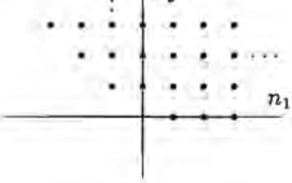


Figure 4.8: Example of an impulse response with wedge support.

In accordance with one-dimensional error diffusion we require that only a finite subset of the wedge support of the feedback filters is actually used (FIR). We have

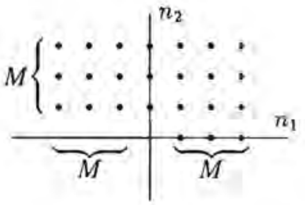


Figure 4.9: Feedback filter with asymmetrical half plane support of order M.

chosen for the support shown in Figure 4.9, also known as asymmetrical half plane support. In this way the support of the feedback filter is described by one parameter M, referred to as the filter-order. The total number of diffusion coefficients equals $2(M^2 + M)$.

Given the support of the feedback filters a number of paths can be followed in order to compute (4.40), all leading to the same result. A possible path (which we have used) is to process the signal samples in a row-by-row fashion, starting in the bottom-left corner. This situation is shown in Figure 4.10. For convenience we

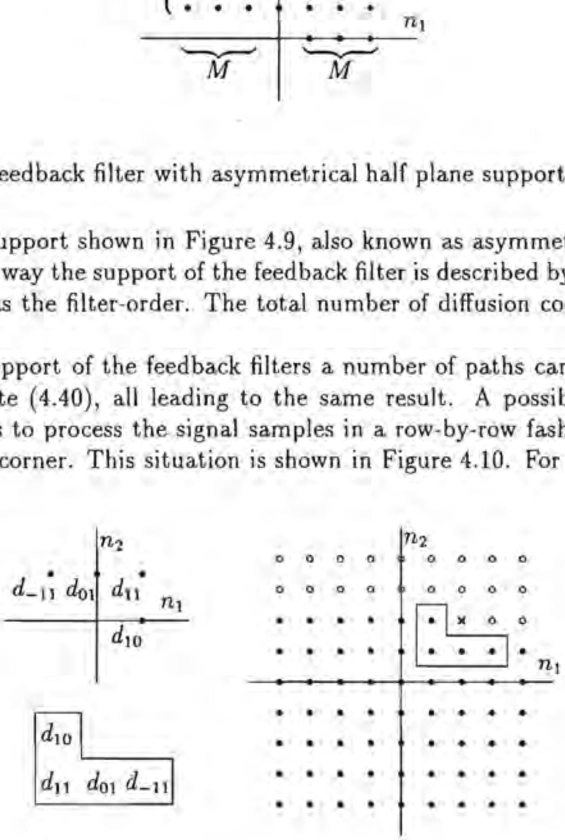


Figure 4.10: Internal error diffusion with an error-feedback filter with asymmetrical half plane support of order $M = 1$.

consider internal error diffusion ($C = 0$) only, with filter-order $M = 1$. The feedback mask is the mirrored version of the support mask $d[n_1, n_2]$. By means of this mask a weighted sum of quantizer errors of previous signal samples (•) is formed, which is taken into account in the decision of the quantizer for the present sample (×) . The samples denoted by open dots (○) have not yet been quantized.

This procedure can also be regarded in a different manner. After a sample has been quantized the quantizer error is propagated (or diffused) to neighbouring non-quantized samples, as shown in Figure 4.11. The quantizer error is weighted for the different directions as given by the coefficients of the impulse response $d[n_1, n_2]$. For this reason these coefficients are often called diffusion coefficients. The idea of

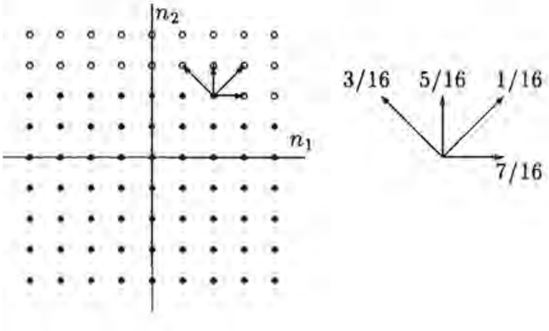


Figure 4.11: Error diffusion according to Floyd and Steinberg.

diffusing the quantizer error to non-quantized neighbouring samples was originally introduced by Floyd and Steinberg (1976) for the transformation of gray-tone images into binary images (halftoning). The set of diffusion coefficients as proposed by Floyd and Steinberg (Figure 4.11) sum up to 1. In this way the total quantizer error is diffused, and the (local) average level of the binary pattern equals the original gray-tone level. Later on we shall return to the implications of such a choice for the set of diffusion coefficients.

As stated before, the row-by-row processing of the samples in Figure 4.10 is just one of the possible paths to follow. An alternative path for example is to process the samples along diagonals in the $(-M-1,1)^T = (-2,1)^T$ direction. Note, that the samples on these diagonals can be quantized independently. This is of interest for parallel signal processing in a multiprocessor environment.

4.3.2 Stability

For one-dimensional error diffusion stability was analyzed in z-domain terms, particularly with regard to the various system functions. In the two-dimensional case the system function of the internal error-feedback filter is defined as the two-dimensional z-transform of the impulse response $d[n_1, n_2]$:

$$D(z_1, z_2) = \sum_{(n_1,n_2)} \sum_{\in R_d} d[n_1, n_2] z_1^{-n_1} z_2^{-n_2} . \tag{4.42}$$

The summation is over the support of the impulse response, denoted by R_d. In a similar way the transfer function $C(z_1, z_2)$ of the external feedback filter is defined, where the summation is over R_c, the support of the impulse response $c[n_1, n_2]$.

In accordance with the one-dimensional case stability implies that the input of the quantizer is not allowed to grow exponentially. A necessary condition is the requirement that the system with transfer function

$$G(z_1, z_2) = \frac{1}{1 - D(z_1, z_2)} \tag{4.43}$$

is stable. This is equivalent with the requirement that the impulse response $g[n_1, n_2]$ of this system is absolutely summable:

$$\sum_{n_1} \sum_{n_2} |g[n_1, n_2]| < \infty \, . \tag{4.44}$$

Weissbach (1992) describes an approach, where (4.44) is used to test stability. In order to determine the impulse response $g[n_1, n_2]$ the difference equation

$$g[n_1, n_2] = \delta[n_1, n_2] + d[n_1, n_2] * g[n_1, n_2] \tag{4.45}$$

is solved numerically. Next, the convergence of (4.44) is tested numerically by determining the sum over a (sufficiently) large region of the support of $g[n_1, n_2]$.

Here, the stability of the error-diffusion system is tested in the z-domain. The z-transform

$$G(z_1, z_2) = \sum_{n_1} \sum_{n_2} g[n_1, n_2] z_1^{-n_1} z_2^{-n_2} \tag{4.46}$$

converges uniformly for those values of (z_1, z_2) where

$$\sum_{n_1} \sum_{n_2} |g[n_1, n_2]| |z_1|^{-n_1} |z_2|^{-n_2} < \infty \, . \tag{4.47}$$

Equation (4.47) defines the region of convergence (ROC), which depends on $|z_1|$ and $|z_2|$ only. According to (4.44) and (4.47) a system is stable iff the 'unit surface' $(|z_1| = 1, |z_2| = 1)$ is in the ROC. As we have already seen, stability testing for one-dimensional error diffusion is rather straightforward. For a causal sequence we know that if z_o is in the ROC, every z with $|z| > |z_o|$ must also be in the ROC. Consequently, the system is stable iff the poles of $G(z)$ are within the unit circle $|z| = 1$. Unfortunately, the situation is far more complicated in two dimensions. There the polynomial $1 - D(z_1, z_2)$ has zero surfaces for which $D(z_1, z_2) = 1$ and which replace the isolated zero points in the one-dimensional case.

The first step in two-dimensional stability analysis is the transformation of wedge support sequences to first-quadrant support sequences. In our case the support R_g of the sequence $g[n_1, n_2]$ (cf. (4.45)) follows from the support R_d, shown in Figure 4.9. The resulting $g[n_1, n_2]$ has wedge support (similar to Figure 4.8, the origin included) with the two boundaries emanating from the origin in the respective directions $(1, 0)^T$ and $(-M, 1)^T$. By means of the linear mapping

$$\tilde{g}[n_1, n_2] = g[n_1 - Mn_2, n_2] \, , \tag{4.48}$$

the wedge support sequence $g[n_1, n_2]$ is transformed to the first-quadrant support sequence $\tilde{g}[n_1, n_2]$. Since

$$\sum_{n_1}\sum_{n_2} |\tilde{g}[n_1, n_2]| = \sum_{n_1}\sum_{n_2} |g[n_1, n_2]| \tag{4.49}$$

stability properties are invariant under this transformation, and we are allowed to analyse the stability of

$$\tilde{G}(z_1, z_2) = \frac{1}{1 - \tilde{D}(z_1, z_2)} \tag{4.50}$$

instead of $G(z_1, z_2)$. This is advantageous, since the stability analysis of first-quadrant support sequences is easier than the stability analysis of wedge support sequences. We remark that the first-quadrant support sequence $\tilde{d}[n_1, n_2]$ and the original wedge support sequence $d[n_1, n_2]$ are related through the linear mapping $\tilde{d}[n_1, n_2] = d[n_1 - Mn_2, n_2]$, similar to (4.48).

One of the first theorems concerning two-dimensional stability was developed by Shanks (Lim, 1990). Applied to our case, this theorem reads

$$\text{Stability} \;\Leftrightarrow\; \tilde{D}(z_1, z_2) \neq 1 \text{ for any } |z_1| \geq 1, |z_2| \geq 1 \;. \tag{4.51}$$

This can be seen as an extension of the causal one-dimensional case, where the solutions of $\tilde{D}(z) = 1$ are not allowed to lie outside the unit circle. In the two-dimensional case it is in general not possible to solve $\tilde{D}(z_1, z_2) = 1$, and therefore a 4-D search would be necessary in order to check (4.51). Due to the enormous amount of work (4.51) is not feasible in practice as a stability test. A more convenient stability test, which is equivalent with (4.51) is (Lim, 1990)

$$\begin{aligned} \text{Stability} \;\Leftrightarrow\; & \tilde{D}(z_1, z_2) \neq 1 \text{ for } |z_1| = 1, |z_2| \geq 1 \text{ and} \\ & \tilde{D}(z_1, z_2) \neq 1 \text{ for } |z_1| \geq 1, |z_2| = 1 \;. \end{aligned} \tag{4.52}$$

This stability test requires two 3-D searches, but can be carried out by many 1-D stability tests. First, we find the solution z_2 of $\tilde{D}(e^{i2\pi\theta_1}, z_2) = 1$ for $-\frac{1}{2} < \theta_1 \leq \frac{1}{2}$. The resulting rootmap is sketched in the z_2 plane. Next, we find the solution of z_1 of $\tilde{D}(z_1, e^{i2\pi\theta_2}) = 1$ for $-\frac{1}{2} < \theta_2 \leq \frac{1}{2}$ and sketch the resulting root map in the z_1 plane. If both root-maps stay within the respective unit circles the system is stable.

Example

In Figure 4.12 we have shown the root maps of a second-order two-dimensional recursive system with coefficients $d[n_1, n_2]$:

$$\begin{array}{ccccc} -0.2005 & 0.0000 & 0.1702 & 0.0000 & -0.1886 \\ 0.0000 & 0.7535 & 0.0000 & -0.4427 & 0.0000 \\ & & & 0.0000 & -0.2436 \end{array} \;. \tag{4.53}$$

Both root maps are within the unit-circle; hence the system is stable.

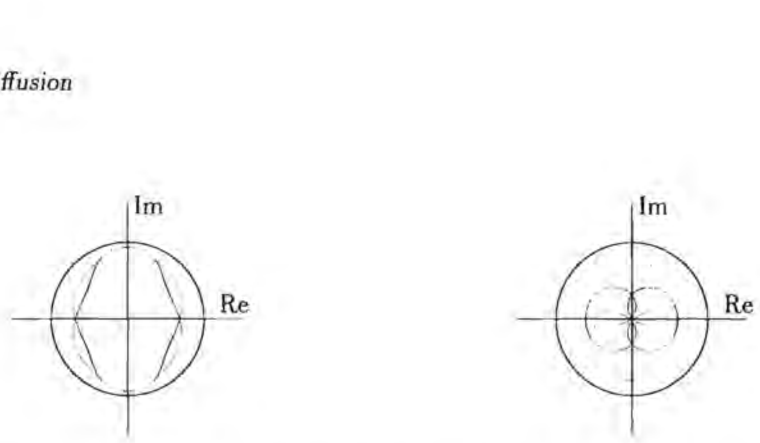


Figure 4.12: Root-maps of a second-order two-dimensional recursive system.

It is possible to reduce the amount of work further by the introduction of more sophisticated stability tests, see for example (Lim, 1990). By sketching the root-maps, however, we get an impression how far the root-maps are from the unit circle. For error diffusion, where linear stability is only a necessary condition, this information is important.

4.3.3 A linear model for two-dimensional error diffusion

In analogy with the one-dimensional case we obtain a linear model for error diffusion by considering the quantizer error $q[n_1, n_2]$ as an external signal, as shown in Figure 4.13. The two-dimensional z-transform of the input signal $\phi[n_1, n_2]$ is defined

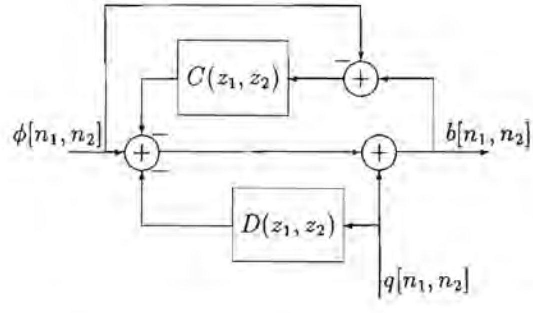


Figure 4.13: A linear model for two-dimensional error diffusion.

according to

$$\Phi(z_1, z_2) = \sum_{n_1 \in \langle N \rangle} \sum_{n_2 \in \langle N \rangle} \phi[n_1, n_2] z_1^{-n_1} z_2^{-n_2} \, . \tag{4.54}$$

In a similar way the z-transforms $B(z_1, z_2)$, $S(z_1, z_2)$ and $Q(z_1, z_2)$ are defined. Note that for signals with finite support the z-transform is defined in the entire $z_1 z_2$-plane. The input-output relation of the linear model in Figure 4.13 reads in terms of z-transforms

$$B(z_1, z_2) = \Phi(z_1, z_2) + \frac{1 - D(z_1, z_2)}{1 + C(z_1, z_2)} Q(z_1, z_2) \ . \tag{4.55}$$

With the application of Fourier holograms we are particularly interested in the deviation between the two-dimensional Fourier transforms of the discrete signals $b[n_1, n_2]$ and $\phi[n_1, n_2]$. To this end we evaluate the z-transform for values of z_1, z_2 on the bi-circle $|z_1| = 1, |z_2| = 1$. In that case, the two-dimensional z-transform passes into the two-dimensional Fourier transform for discrete signals:

$$\Phi_d(\theta_1, \theta_2) = \Phi(e^{i2\pi\theta_1}, e^{i2\pi\theta_2}) = \sum_{n_1 \in \langle N \rangle} \sum_{n_2 \in \langle N \rangle} \phi[n_1, n_2] e^{-i2\pi(\theta_1 n_1 + \theta_2 n_2)} \ , \tag{4.56}$$

with similar expressions for $B(z_1, z_2)$ and $Q(z_1, z_2)$. With $E(z_1, z_2) = B(z_1, z_2) - \Phi(z_1, z_2)$ we thus find for the quantization error in the Fourier plane

$$E(e^{i2\pi\theta_1}, e^{i2\pi\theta_2}) = \frac{1 - D(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})}{1 + C(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})} Q(e^{i2\pi\theta_1}, e^{i2\pi\theta_2}) \ . \tag{4.57}$$

According to (4.57) we observe in the hologram plane the Fourier transform of the quantizer error $q[n_1, n_2]$ filtered by

$$H(e^{i2\pi\theta_1}, e^{i2\pi\theta_2}) = \frac{1 - D(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})}{1 + C(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})} \ . \tag{4.58}$$

Example

The set of diffusion coefficients as proposed by Floyd and Steinberg (cf. Figure 4.11) has the special property that

$$\sum_{m_1, m_2} \sum_{\in R_d} d[m_1, m_2] = 1 \ . \tag{4.59}$$

Under this condition the noise shaping transfer function

$$H(z_1, z_2) = 1 - D(z_1, z_2) = 1 - \sum_{m_1, m_2} \sum_{\in R_d} d[m_1, m_2] z_1^{-m_1} z_2^{-m_2} \tag{4.60}$$

vanishes for $(z_1 = 1, z_2 = 1)$. As a result the contribution of the quantization error is small near $(\theta_1 = 0, \theta_2 = 0)$ in the Fourier plane. This effect is shown in Figure 4.14, where we have shown the modulus of the transformed quantization error which is the result of quantizing a digital image using the Floyd-Steinberg diffusion coefficients. When this image is viewed from a sufficiently large distance, the high-frequency quantization noise is attenuated by the low-pass visual system and the observed image resembles the original quite well.

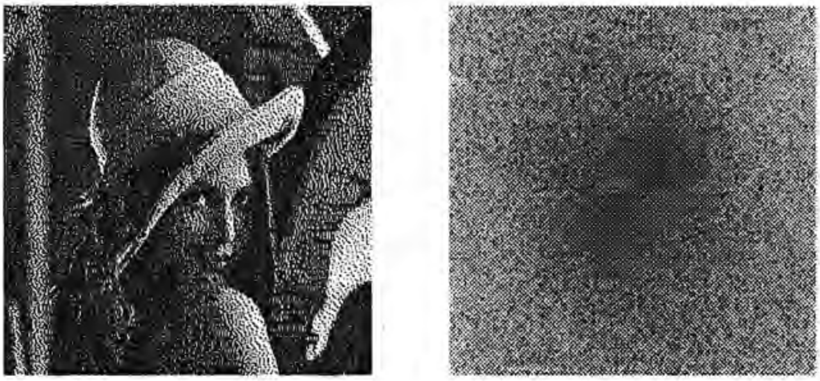


Figure 4.14: a. Digital image halftoned with error diffusion. b. Modulus of the quantization error in the Fourier plane.

Likewise it is possible to use feedback filters for the quantization of computer generated holograms in order to decrease the contribution of the transformed quantization error in the object window. In this thesis we restrict ourselves to internal feedback filters ($C = 0$). An attractive method based on separable two-dimensional filters is proposed by Weissbach (1992). There, the desired noise shaping effect is obtained by generating zero lines in the reconstruction plane. An example of such a feedback filter is discussed in Section 4.4. In the next subsection we discuss how the least-square design method is applied in two dimensions. We remark that finding the optimal external error-feedback filter is a nonlinear problem which has been considered by Kim and Kim (1986).

4.3.4 Least-square feedback filter design

In accordance with the one-dimensional case, the design of an optimal internal error-feedback filter is a linear problem. Considering (4.57) we have to find a set of diffusion coefficients in order to minimize

$$\iint_{\mathbb{F}} |1 - D(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})|^2 \mathrm{d}\theta_1 \mathrm{d}\theta_2 \,. \tag{4.61}$$

For stability reasons, however, we minimize

$$\iint |A(\theta_1, \theta_2)|^2 |1 - D(e^{i2\pi\theta_1}, e^{i2\pi\theta_2})|^2 \mathrm{d}\theta_1 \mathrm{d}\theta_2 \,, \tag{4.62}$$

where the weight-function $|A(\theta_1, \theta_2)|^2$ is chosen according to

$$|A(\theta_1, \theta_2)|^2 = \begin{cases} 1 & \theta \in \mathbb{F} \cup \mathbb{F}^* \\ a_s^2 & \text{elsewhere} \end{cases} \tag{4.63}$$

In the original formulation (4.61) we have $a_s = 0$. If the filter turns out to be unstable, a_s^2 has to be increased. Also some form of smoothing at the boundary of $\mathbb{F}$ can be taken into consideration. This possibility will, however, not be explored here.

Using Parseval's theorem for two-dimensional signals we can express the object function (4.62) in the spatial domain:

$$\sum_{n_1}\sum_{n_2}\left(a[n_1,n_2]*(\delta[n_1,n_2]-d[n_1,n_2])\right)^2 = \\ \sum_{n_1}\sum_{n_2}\left(a[n_1,n_2]-a[n_1,n_2]*d[n_1,n_2]\right)^2 \ . \quad (4.64)$$

The two-dimensional convolution gives rise to a finite sum over the support R_d of the internal feedback filter, according to

$$\sum_{n_1}\sum_{n_2}\left(a[n_1,n_2]-\sum_{m_1,m_2}\sum_{\in R_d} d[m_1,m_2]a[n_1-m_1,n_2-m_2]\right)^2 \ . \quad (4.65)$$

In a straightforward way (4.65) is written as

$$r[0,0]-2\sum_{m_1,m_2}\sum_{\in R_d} d[m_1,m_2]r[m_1,m_2]+ \\ \sum_{m_1,m_2}\sum_{\in R_d}\sum_{l_1,l_2}\sum_{\in R_d} d[m_1,m_2]d[l_1,l_2]r[m_1-l_1,m_2-l_2] \ , \quad (4.66)$$

where we have introduced the two-dimensional (deterministic) autocorrelation

$$r[n_1,n_2]=a[n_1,n_2]\star a[n_1,n_2]=\sum_{m_1}\sum_{m_2}a[m_1,m_2]a[m_1-n_1,m_2-n_2] \ . \quad (4.67)$$

In order to express (4.66) in a standard quadratic form equivalent to (4.32) we label the diffusion coefficients in the support R_d according to $n_1(m), n_2(m)$ with $m = 1\ldots 2(M^2+M)$, where the specific order of the coefficients is not important. Next, we introduce the diffusion coefficient vector $\boldsymbol{d}$ with elements $d_m = d[n_1(m), n_2(m)]$ and rewrite (4.66) into

$$r[0,0]-2\boldsymbol{r}^T\boldsymbol{d}+\boldsymbol{d}^T R\boldsymbol{d} \ , \quad (4.68)$$

where we have the correlation vector $\boldsymbol{r}$ with elements $r_m = r[n_1(m), n_2(m)]$ and the correlation matrix R with elements $R_{ml} = r[n_1(m)-n_1(l), n_2(m)-n_2(l)]$. Contrary to the one-dimensional case the matrix R is not a Toeplitz matrix. The two-dimensional autocorrelation coefficients are determined using

$$r[n_1,n_2]=\int_{(1)}\int_{(1)}|A(\theta_1,\theta_2)|^2\cos(2\pi(n_1\theta_1+n_2\theta_2))\mathrm{d}\theta_1\mathrm{d}\theta_2 \ . \quad (4.69)$$

The minimum of (4.68) occurs for

$$\boldsymbol{d}=R^{-1}\boldsymbol{r} \ . \quad (4.70)$$

In contrast with the one-dimensional case, minimizing (4.62) does not give automatically rise to a set of diffusion coefficients with zero surfaces of $1-\tilde{D}(z_1,z_2)$ within the unit bi-circle.

4.4 Quantizing hologram distributions with error diffusion

In this section we compare a number of (internal) error-feedback filters designed for the calculation of binary amplitude holograms. The original intensity-object and the accompanying hologram are shown in Figure 4.15. The object's intensity has been multiplied by a random phase distribution. The respective sizes of the object and the hologram are 32^2 and 128^2 samples, which means that we have used 4 times oversampling in the determination of the hologram samples. The (discrete) object is located in the Fourier plane within the object window

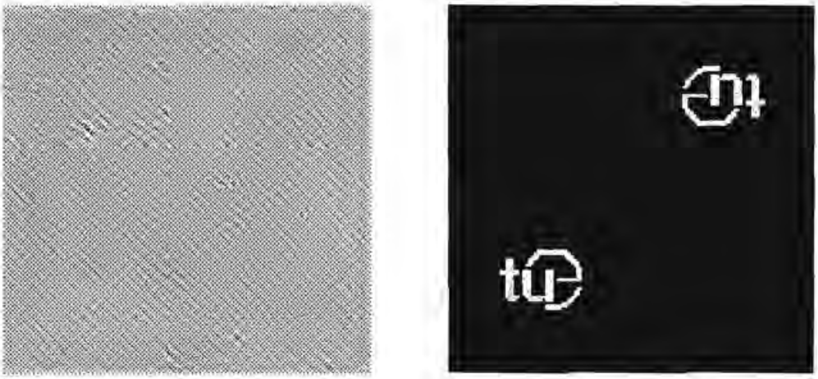


Figure 4.15: a. Original hologram distribution. b. Modulus of the Fourier transform of the hologram.

$$\mathbb{F} = \left\{(\theta_1, \theta_2) \mid \tfrac{1}{8} < \theta_1 \leq \tfrac{3}{8}, \tfrac{1}{8} < \theta_2 \leq \tfrac{3}{8}\right\} . \qquad (4.71)$$

In order to obtain an amplitude hologram, the bipolar output $b[n_1, n_2]$ of the error-diffusion system is transformed into a unipolar signal. In the results both the binary holograms and the modulus of its Fourier transform are shown. The large dc-peak in the reconstruction plane has been suppressed.

A comparison of the results of the computer simulations for the different feedback filters is made by calculation of the signal-to-noise ratio (2.30), where amplitude-errors are considered only. To this end we consider the numerical approximation:

$$\mathrm{SNR} = \frac{\sum\limits_{k_1,k_2} \sum\limits_{\in \mathbf{F}} |\Phi[k_1, k_2]|^2}{\sum\limits_{k_1,k_2} \sum\limits_{\in \mathbf{F}} \left(|B[k_1, k_2]| - |\Phi[k_1, k_2]|\right)^2} . \qquad (4.72)$$

For the calculation of the hologram efficiency $\eta = \eta_t \eta_d$ we have used:

$$\eta_t = \frac{1}{N^2} \sum_{n_1, n_2} \sum_{\in \mathbf{H}} b^2[n_1, n_2] \tag{4.73}$$

and

$$\eta_d = \frac{\sum\limits_{k_1, k_2} \sum\limits_{\in \mathbf{F}} |B[k_1, k_2]|^2}{\sum\limits_{k_1 \in \langle N \rangle} \sum\limits_{k_1 \in \langle N \rangle} |B[k_1, k_2]|^2}, \tag{4.74}$$

which follow from (2.34) and (2.35). Using Parseval's theorem we find for the total efficiency

$$\eta = \frac{1}{N^4} \sum_{k_1, k_2} \sum_{\in \mathbf{F}} |B[k_1, k_2]|^2 . \tag{4.75}$$

Next we discuss the results of three different filters: 'hardclipping', a filter with zero lines in the reconstruction plane, and a least-square optimal filter of order $M = 2$.

Hardclipping

With hardclipping of the hologram distribution no error feedback is applied at all ($d[n] = 0$). The resulting binary hologram and its Fourier transform are shown in Figure 4.16. A large contribution of the quantization error is observed in the object window in the reconstruction plane. As a result, the signal-to-noise ratio is small (SNR = 0.1). The efficiency of the binary amplitude hologram, on the other hand, is rather large: $\eta = 0.1$.

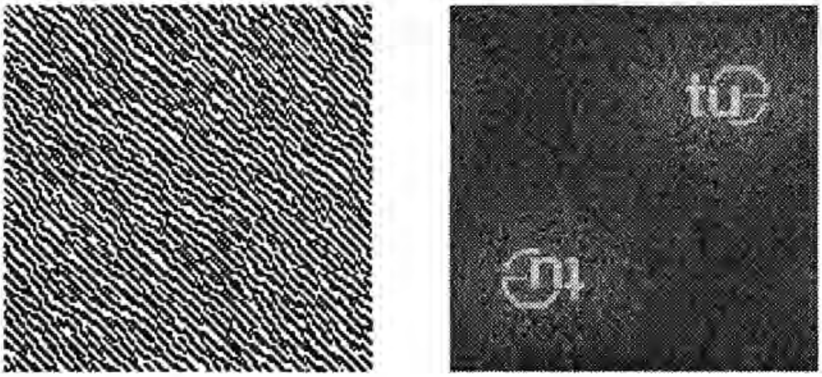


Figure 4.16: a. Binary hologram distribution when no error feedback is applied (hardclipping). b. Modulus of the Fourier transform of the binary hologram.

Zero Lines

In the one-dimensional case it is possible to lower the contribution of the transformed quantization error in the object window by placing a zero of the noise shaping transfer function at an appropriate position on the unit circle. A similar approach with separable noise shaping filters is possible in the two-dimensional case where the zeros become zero lines on the unit surface. We remark that in that case the error-diffusion system is marginally stable. For our choice of the object window we take the lines $\theta_1 - \theta_2 = 0$ and $\theta_1 + \theta_2 = \frac{1}{2}$ (Weissbach, 1992). In terms of z_1 and z_2 we have $z_1 z_2^{-1} = 1$ and $z_1^{-1} z_2^{-1} = -1$. For the noise shaping function we find

$$H(z_1, z_2) = (1 + z_1^{-1} z_2^{-1})(1 - z_1 z_2^{-1}) = 1 + z_1^{-1} z_2^{-1} - z_1 z_2^{-1} - z_2^{-2} \, . \qquad (4.76)$$

With $H(z_1, z_2) = 1 - D(z_1, z_2)$ the diffusion coefficients are found to be $d[1,1] = -1$, $d[-1,1] = 1$ and $d[0,2] = 1$. The quantized hologram and its Fourier transform are shown in Figure 4.17. Due to the zero lines, which can be observed clearly in the reconstruction plane, the contribution of the transformed quantization error is lowered in the object window. In this case we have SNR = 12, which is much larger than for the previous case. The error-feedback filter has the disadvantage that the transformed quantization error is also reduced in regions where it is not necessary. The gain in the signal-to-noise ratio is obtained at the expense of a lower efficiency, which is reduced to $\eta = 0.007$.

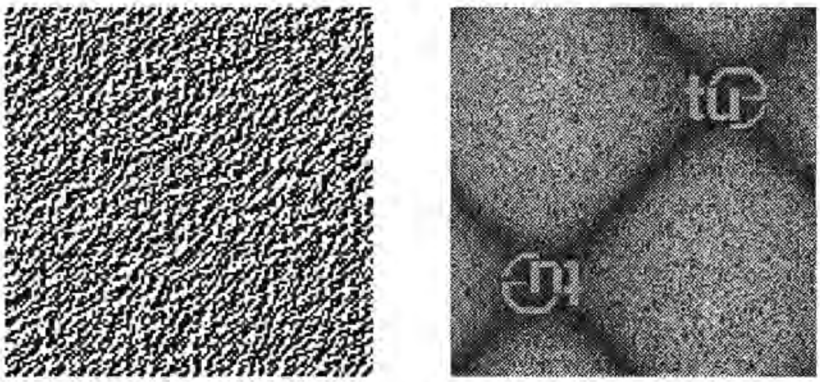


Figure 4.17: a. Binary hologram distribution obtained with the zero lines approach. b. Modulus of the Fourier transform of the binary hologram.

Least-square filter design

Using the least-square design method for a filter of order $M = 2$, a stable filter is found for $a_s^2 = 0.01$. The diffusion coefficients and the zero surfaces of this filter were given in the example on p. 74. Quantization of the hologram with the obtained diffusion coefficients results in the binary hologram shown in Figure 4.18. The signal-to-noise ratio of the reconstructed object (also shown in Figure 4.18) equals SNR = 25. Again, the efficiency equals $\eta = 0.007$. Note that the two-dimensional version of stability condition (4.12) does not hold for the obtained set of diffusion coefficients. Determining the diffusion coefficients under this constraint has been considered by Barnard (1988) for an error-feedback filter of order $M = 1$.

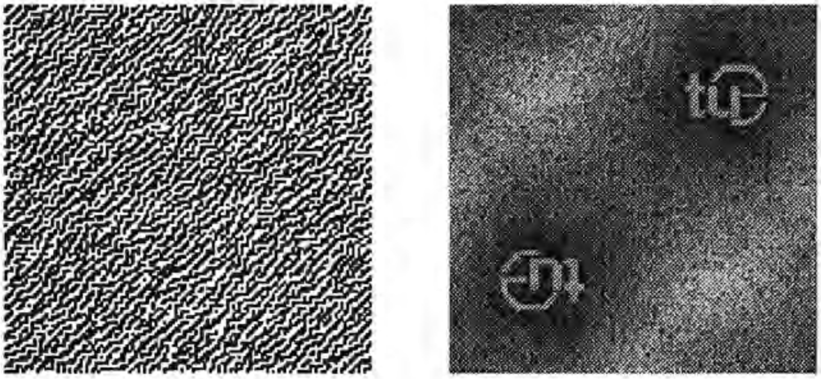


Figure 4.18: a. Binary hologram distribution obtained with a least-square optimal filter of order $M = 2$. b. Modulus of the Fourier transform of the binary hologram.

We remark that calculation of the scaling factor λ (2.33) using the approximation

$$\lambda = \frac{\sum\sum_{k_1,k_2 \in \mathbf{F}} |B[k_1,k_2]||\Phi[k_1,k_2]|}{\sum\sum_{k_1,k_2 \in \mathbf{F}} |\Phi[k_1,k_2]|^2}, \tag{4.77}$$

showed that in the last two cases $\lambda \approx 1$, as was required. For hardclipping, however, we found a rather large scaling factor ($\lambda \approx 3.5$), which is the underlying reason for the large efficiency. With λ taken into account in the signal-to-noise ratio for hardclipping we found SNR = 6.

4.5 Symmetrical error diffusion

So far, the error-feedback filters were required to have wedge support in order to make recursive computation possible. The finiteness of the hologram, however, admits other approaches for the determination of the binary hologram samples. Consequently, error-feedback filters with four quadrant support can be applied. The diffusion coefficients then possess certain symmetry properties due to the symmetry in the configuration in the Fourier plane. For this reason we speak of symmetrical error diffusion (Anastassiou, 1989).

For convenience we consider an error-diffusion system with external feedback only, which is described by

$$\begin{aligned} s[n_1,n_2] &= \phi[n_1,n_2] - c[n_1,n_2] * e[n_1,n_2] \\ b[n_1,n_2] &= \mathcal{Q}s[n_1,n_2] \\ e[n_1,n_2] &= b[n_1,n_2] - \phi[n_1,n_2] \ . \end{aligned} \tag{4.78}$$

By means of substitution we derive from the above set of equations

$$b[n_1,n_2] = \mathcal{Q}\left(-c[n_1,n_2] * b[n_1,n_2] + \phi[n_1,n_2] + c[n_1,n_2] * \phi[n_1,n_2]\right) . \tag{4.79}$$

With a wedge support external error-feedback filter ($c[n_1,n_2] = 0$) the binary hologram samples are determined recursively, one sample at a time. In the next chapter we consider how to solve (4.79) in the more general case where $c[n_1,n_2]$ has four quadrant support (again with $c[n_1,n_2] = 0$). We remark that symmetrical internal error-diffusion was introduced by Anastassiou (1989).

4.6 Discussion

In this chapter we have considered the quantization of computer-generated Fourier holograms by means of error diffusion. A linear model for error diffusion is obtained by modeling the quantization error as an additive noise term. In this linear approximation the quantization noise in the Fourier plane is determined by the diffusion coefficients through the transfer function of a noise shaping filter. The internal diffusion coefficients determine the numerator, while the external diffusion coefficients determine the denominator of the transfer function. We have restricted ourselves to the design of internal error-feedback filters, which can be formulated as a least-square optimization problem.

Due to its recursive nature, the error-diffusion system can become unstable. In order to avoid such a situation a stability criterion for the (internal) diffusion coefficients is formulated, which states that the zero surfaces of the noise shaping transfer function have to lie within the unit bi-circle. The accompanying root-maps show to which degree the stability boundary is approached for a calculated set of diffusion coefficients. In the design of the internal error-feedback filter one parameter has to be adjusted manually until a stable error-diffusion system is obtained. Formulating necessary and sufficient stability conditions for error-diffusion systems is still an unsolved problem.

Designing an external feedback filter is a nonlinear problem. A solution to this problem has been proposed by Kim and Kim (1986). In this approach a filter approximation error and a stability error for the feedback filter is determined. The approximation error is a measure for the deviation between the realized and the desired noise shaping transfer function. The stability error is a measure for the stability of the error diffusion system, and is determined by means of complex cepstrum techniques (Ekstrom et al., 1976; 1980). Using nonlinear optimization techniques both the approximation error and the stability error are minimized. In (Kant, 1993) this method is applied for both internal and external feedback filters. Computer simulations show that with internal error feedback in general a higher signal-to-noise ratio is obtained than with external error feedback. Comparable results are achieved with least-square optimal feedback filters, which are calculated within a fraction of the time needed for the nonlinear optimization.

The least-square design method for the (internal) feedback filters is based on the minimization of the signal-to-noise ratio. Consequently, the contribution of the quantization noise in the Fourier plane can become quite large outside the object window, resulting in a small diffraction efficiency. In order to optimize the diffraction efficiency as well, the desired noise shaping characteristic should also be specified outside the object window. In (Kant, 1993) feedback filters have been designed which give rise to a higher diffraction efficiency. This result is achieved at the expense of a lower signal-to-noise ratio. Applying the least-square filter design method for the optimization of both the signal-to-noise ratio and the diffraction efficiency is an interesting subject for further research.

Chapter 5

Hopfield's neural network

5.1 Introduction

In the previous chapter we have seen that recursive computability is an important aspect of error diffusion. We already noted that with a finite number of samples recursive computability is an unnecessary restriction. In this chapter we shall find that the sequential updating law of a discrete-time Hopfield neural network is a generalization of error diffusion. The question then arises whether it is possible to obtain better holograms by means of a Hopfield neural network. After an introduction to Hopfield's neural network in Section 5.2, we discuss in Section 5.3 how such a network is applied for the calculation of binary holograms. The combination of a Hopfield network with simulated annealing, known as the Boltzmann machine, is the subject of Section 5.4.

With Hopfield's neural network the hologram samples are calculated in an iterative way rather than a recursive way. A number of iterative methods for the calculation of holograms are known in literature, such as the iterative-Fourier transform algorithm (IFTA), the direct binary search method (DBS) and projections on convex sets (POCS). In this chapter we restrict ourselves to Hopfield's neural network and the Boltzmann machine, but some similarities and differences in relation with the other methods will be mentioned.

In the remainder of this introduction we reformulate the original problem definition (3.6) for discrete signals. Equation (3.6) is a measure for the deviation between the original and the binary hologram. The starting point for discrete signals is slightly different, and is discussed below. In order to avoid tedious notation we first consider one-dimensional signals; later on the results are generalized for two dimensions. Given the hologram samples $\phi[n]$ for $n \in \langle N \rangle$ our goal is to find the binary signal $b[n]$ which minimizes the 'distance'

$$d = \int_{(1)} |A(\theta)|^2 |B(e^{i2\pi\theta}) - \Phi(e^{i2\pi\theta})|^2 \mathrm{d}\theta \,, \tag{5.1}$$

where the frequency weighting function $|A(\theta)|^2$ is given by

$$|A(\theta)|^2 = \begin{cases} 1 & \theta \in \mathbb{F} \cup \mathbb{F}^* \\ 0 & \text{elsewhere} \end{cases} \tag{5.2}$$

We recall that the Fourier transform of $b[n]$ is defined by

$$B(e^{i2\pi\theta}) = \sum_{n\in\langle N\rangle} b[n]e^{-i2\pi\theta n} , \tag{5.3}$$

and a similar expression holds for $\Phi(e^{i2\pi\theta})$. Note that (5.1) is similar to (4.27); we can also write (5.1) as a quadratic form, but now in the binary samples $b[n]$. Here, we give a derivation in the frequency domain instead of the spatial domain. Considering the fact that

$$|B(e^{i2\pi\theta}) - \Phi(e^{i2\pi\theta})|^2 = \sum_{n\in\langle N\rangle} \sum_{m\in\langle N\rangle} (b[n] - \phi[n])\,(b[m] - \phi[m])\,e^{-i2\pi\theta(n-m)} \tag{5.4}$$

we can rewrite (5.1) into

$$d = \sum_{n\in\langle N\rangle} \sum_{m\in\langle N\rangle} (b[n] - \phi[n])\,(b[m] - \phi[m])\,R_{nm} \tag{5.5}$$

with the correlation coefficients R_{nm} given by

$$R_{nm} = \int_{\langle 1\rangle} |A(\theta)|^2 e^{-i2\pi(n-m)\theta} \mathrm{d}\theta . \tag{5.6}$$

With the introduction of the signal vector $\boldsymbol{\phi} = (\phi[-\frac{1}{2}N], \ldots, \phi[\frac{1}{2}N-1])^T$ and the binary signal vector $\boldsymbol{b} = (b[-\frac{1}{2}N], \ldots, b[\frac{1}{2}N-1])^T$ we find for (5.5)

$$d = d(\boldsymbol{b}, \boldsymbol{\phi}) = (\boldsymbol{b} - \boldsymbol{\phi})^T R (\boldsymbol{b} - \boldsymbol{\phi}) . \tag{5.7}$$

According to (5.6) the matrix R has a Toeplitz structure.

Equation (5.1) considers the difference between the Fourier transforms of the original hologram and the binary hologram for a continuous frequency region. Instead we could consider discrete frequencies $\theta = k/N$, with $k \in \langle N\rangle$. Our goal is then to minimize

$$d = \frac{1}{N} \sum_{k\in\langle N\rangle} |A_k|^2 |B[k] - \Phi[k]|^2 , \tag{5.8}$$

with $B[k] = B(e^{i2\pi k/N})$ and $\Phi[k] = \Phi(e^{i2\pi k/N})$. Introducing the Fourier transform vectors $\boldsymbol{B} = (B[-\frac{1}{2}N], \ldots, B[\frac{1}{2}N-1])^T$ and $\boldsymbol{\Phi} = (\Phi[-\frac{1}{2}N], \ldots, \Phi[\frac{1}{2}N-1])^T$ we rewrite (5.8) into

$$d = \frac{1}{N} \|A(\boldsymbol{B} - \boldsymbol{\Phi})\|_2^2 = \frac{1}{N} (\boldsymbol{B} - \boldsymbol{\Phi})^H A^H A (\boldsymbol{B} - \boldsymbol{\Phi}) , \tag{5.9}$$

where $A^H A$ is a diagonal matrix with elements $(A^H A)_{kk} = |A(\frac{2\pi k}{N})|^2$. In accordance with (5.2) we could take $(A^H A)_{kk} = 1$ for $k \in \mathbb{F}$ and $(A^H A)_{kk} = 0$ elsewhere. The signal vectors and the accompanying Fourier transform vectors are related according to

$$\begin{aligned} \boldsymbol{\Phi} &= F\boldsymbol{\phi} \\ \boldsymbol{B} &= F\boldsymbol{b}\,, \end{aligned} \tag{5.10}$$

where F is the Fourier transformation matrix, with elements $F_{kn} = e^{-i2\pi kn/N}$. The Fourier transformation matrix is symmetric ($F^T = F$) and has an inverse $F^{-1} = \frac{1}{N}F^*$, hence $F^H = F^{T*} = NF^{-1}$. Using the last property we find for (5.9)

$$d(\boldsymbol{b},\boldsymbol{\phi}) = (\boldsymbol{b}-\boldsymbol{\phi})^T F^{-1} A^H A F(\boldsymbol{b}-\boldsymbol{\phi})\,. \tag{5.11}$$

Now the correlation matrix reads $R = F^{-1}A^H AF$, and has a circulant structure. With the above choice of $(A^H A)_{kk}$ the correlation matrix R is positive semidefinite.

The discrete-time signal $\boldsymbol{\phi}$ can be considered as a point in an N-dimensional signal space. Since the amplitude of $\phi[n]$ is assumed to be bounded according to $\max|\phi[n]| = 1$, this point is somewhere in the N-dimensional hypercube $C^N = [-1,1]^N$. Each binary signal $\boldsymbol{b}$ is one of the 2^N vertices of the hypercube. Given the signal $\boldsymbol{\phi}$ we thus have to find the closest vertex, where the distance between two signals is defined according to (5.7) or (5.11). The set of equidistant signals $\boldsymbol{x}$ with respect to $\boldsymbol{\phi}$, according to $d(\boldsymbol{x},\boldsymbol{\phi}) = c^2$, forms an ellipsoid with center of gravity $\boldsymbol{\phi}$. (The directions and the lengths of the axes are determined by the eigenvectors and eigenvalues of R, many of which are zero.) Finding the closest vertex can thus be visualized in the following way. Starting with $c = 0$ (corresponding to $\boldsymbol{x} = \boldsymbol{\phi}$) the ellipsoide is expanded by increasing c until a vertex is reached. When more vertices are reached simultaneously, we select the vertex with the smallest index.

By repeating this process we assign to every point $\boldsymbol{x}$ in the hypercube the closest vertex. The signal space is thus divided in connected regions

$$C_i = \left\{\boldsymbol{x} \in [-1,1]^N \mid d(\boldsymbol{x},\boldsymbol{b}_i) \le d(\boldsymbol{x},\boldsymbol{b}_j); j \neq i\right\}\,. \tag{5.12}$$

All points in a certain region C_i are mapped onto the accompanying vertex $\boldsymbol{b}_i$. The regions are convex and form the so-called Voronoi or Dirichlet partitioning of the hypercube (Voronoi, 1907). The set of points $\boldsymbol{x}$ with equal distance to the pair of vertices $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ is defined according to

$$S_{ij} = \left\{\boldsymbol{x} \in [-1,1]^N \mid d(\boldsymbol{x},\boldsymbol{b}_i) = d(\boldsymbol{x},\boldsymbol{b}_j)\right\}\,. \tag{5.13}$$

In a straightforward way it can be shown that the set S_{ij} is the hyperplane

$$(\boldsymbol{b}_j - \boldsymbol{b}_i)^T R\left[\boldsymbol{x} - \tfrac{1}{2}(\boldsymbol{b}_i + \boldsymbol{b}_j)\right] = 0\,. \tag{5.14}$$

For each pair of vertices $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ the accompanying hyperplane goes through $\frac{1}{2}(\boldsymbol{b}_i + \boldsymbol{b}_j)$ and has a normal vector $R(\boldsymbol{b}_j - \boldsymbol{b}_i)$. Clearly, the hyperplanes are the basis for the partitioning of the hypercube in decision regions, in the sense that each boundary is part of such a hyperplane. As can be concluded from (5.14) the partitioning is completely determined by the correlation matrix R and hence by the weighting function $|A(\theta)|^2$.

Example
We consider a few examples of decision regions for a signal space of dimension $N = 2$, where we have the input signal $\boldsymbol{\phi} = (\phi_1, \phi_2)^T$ and the binary signal $\boldsymbol{b} = (b_1, b_2)^T$. We follow the discrete frequency treatment, where we need the Fourier transformation matrices

$$F = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } F^{-1} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} . \tag{5.15}$$

Without loss of generality we write for the correlation matrix

$$\begin{aligned} R &= F^{-1}A^H A F \\ &= \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & a \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \tfrac{1}{2}\begin{bmatrix} 1+a & 1-a \\ 1-a & 1+a \end{bmatrix}, \end{aligned} \tag{5.16}$$

with a real parameter $a \geq 0$.

For $a = 1$ both frequencies are equally weighted in the error measure (5.8). The resulting correlation matrix then reads

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} , \tag{5.17}$$

so that the distance d in (5.8) becomes the Euclidean distance. Points of equal distance from the origin are on circles

$$\boldsymbol{x}^T R \boldsymbol{x} = x_1^2 + x_2^2 = c^2. \tag{5.18}$$

Using (5.14) we find the decision regions as shown in Figure 5.1a. Given a vector $\boldsymbol{\phi}$ the closest vertex is determined by means of

$$\begin{aligned} b_1 &= \text{sign } (\phi_1) \\ b_2 &= \text{sign } (\phi_2) . \end{aligned} \tag{5.19}$$

Obviously, with Euclidean distance (when all frequencies are equally weighted) we quantize each sample independently (hardclipping).

When a is decreased to $a = \frac{1}{3}$ the correlation matrix turns into

$$R = \tfrac{2}{3}\begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}. \tag{5.20}$$

Points of equal distance from the origin are now on ellipses

$$x^T R x = x_1^2 + x_1 x_2 + x_2^2 = c^2. \tag{5.21}$$

The accompanying decision regions are shown in Figure 5.1b. It can now be shown that the closest vertex for a given signal can be determined by solving the set of nonlinear equations

$$\begin{aligned} b_1 &= \operatorname{sign}[\phi_1 - f(\phi_2)] \\ b_2 &= \operatorname{sign}[\phi_2 - f(\phi_1)]\,, \end{aligned} \tag{5.22}$$

where $f(\cdot)$ is a piece-wise linear function.

In the last example a is further decreased to $a = 0$. Now, only the dc-term is considered in the frequency domain, resulting in a correlation matrix

$$R = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{5.23}$$

Points of equal distance from the origin for this degenerate case are now on the lines

$$x^T R x = (x_1 + x_2)^2 = c^2 \tag{5.24}$$

The boundaries of the Voronoi cells are shown in Figure 5.1c. Note that input signals in the middle decision region don't have a unique binary representation.

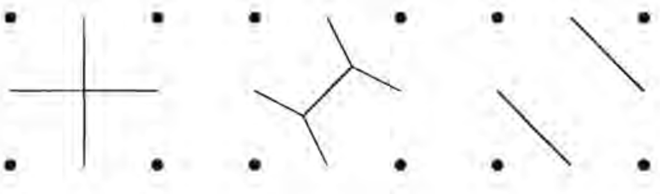


Figure 5.1: Voronoi cells in a two-dimensional signal space for $a = 1$ (a), $a = \frac{1}{3}$ (b) and $a = 0$ (c). The vertices $\{\pm 1, \pm 1\}$ are denoted by solid dots.

In the previous examples we have seen how the matrix R gives rise to decision regions in the signal space. In order to find the optimal binary representation $\boldsymbol{b}$ for a given signal $\boldsymbol{\phi}$ we have to determine in which decision region $\boldsymbol{\phi}$ is located. Unfortunately, in a high-dimensional signal space it is not clear which hyperplanes contribute to the boundary of each decision region. Checking all possible vertices, on the other hand, is not a feasible solution, as the number of vertices grows exponentially as a function of the dimension N.

For the optimal binary hologram, however, we only desire a good approximation within the object window in the reconstruction plane. Due to the resulting degrees of freedom (outside the object window), we can expect that a number of reasonably good solutions exist. For such kind of problems local search methods can provide an alternative. Given an initial binary hologram (vertex), the idea is to search a local neighbourhood (vertices) for a better solution. As soon as a better solution is found, the local neighbourhood of this binary hologram is searched. Repeating this process, we find in a finite number of steps a binary hologram which is locally optimal with respect to its neighbourhood. So, instead of trying to find the optimal hologram we content ourselves with a good hologram, which can be found with a reasonable amount of computation.

Considering (5.7) or (5.11) we thus try to find the binary vector $\boldsymbol{b}$ that minimizes the cost function

$$C(\boldsymbol{b}) = \boldsymbol{b}^T R \boldsymbol{b} - 2\boldsymbol{\phi}^T R \boldsymbol{b} \; . \tag{5.25}$$

The term $\boldsymbol{\phi}^T R \boldsymbol{\phi}$ does not depend on the choice of $\boldsymbol{b}$ and has therefore been omitted. Hopfield's neural network, which is discussed in the next section, provides a local search method that can be applied to solve this combinatorial, quadratic optimization problem.

We conclude this introduction with the remark that in the error measures (5.1) and (5.8) both amplitude and phase errors are considered. For amplitude-only problems (cf. (2.30)), the error measure becomes

$$d = \int_{\mathbb{F}} \left(|B(e^{i2\pi\theta})| - |\Phi(e^{i2\pi\theta})| \right)^2 \mathrm{d}\theta \; . \tag{5.26}$$

A similar expression holds for the discrete frequency case. In order to derive an expression for the distance in terms of the binary hologram samples it is convenient however to consider the deviation $(|B|^2 - |\Phi|^2)^2$ instead of $(|B| - |\Phi|)^2$ in (5.26). It is possible to show (van Gompel, 1993) that the distance can then be written as

$$d = \sum_m \sum_n \sum_o \sum_p \alpha_{mnop} b_m b_n b_o b_p + \sum_m \sum_n \beta_{mn} b_m b_n \; . \tag{5.27}$$

Higher-order neural networks (Guyon et al., 1988) can be applied to solve such higher-order optimization problems. Because of the complexity we shall not follow this approach here. An alternative solution to this problem is based on the idea that under the assumption that the phase of Φ equals the phase of B, the distance (5.1) passes into (5.26). Since the amplitude $|\Phi|$ is prescribed only, we are able to solve the amplitude-only optimization problem by solving the amplitude-phase optimization problem. In Subsection 5.3.3 we discuss this approach in more detail.

5.2 Hopfield's neural network

In this section we give a short introduction to Hopfield's neural network. We consider a discrete, non-deterministic model (Hopfield, 1982) and a continuous, deterministic model (Hopfield, 1984).

5.2.1 A discrete, non-deterministic model

In 1982 J. Hopfield introduced a mathematical model in order to study the properties of physical systems built of a large number of interacting elements. In this model we have a number of processing elements (or neurons), which we label according to $n = 1, \ldots, N$. Each of the N neurons has two states $v_n = \{-1, +1\}$. By means of interconnections between the neurons the state of a neuron is passed on to other neurons. With the interconnection from neuron m to neuron n an interconnection weight W_{nm} is associated.

Each neuron samples its input at random times and adjusts its state according to the updating law

$$v_n^{\text{new}} = \begin{cases} +1 & \text{if } \sum_{m=1}^{N} W_{nm} v_m^{\text{old}} > t_n \\ v_n^{\text{old}} & \text{if } \sum_{m=1}^{N} W_{nm} v_m^{\text{old}} = t_n \\ -1 & \text{if } \sum_{m=1}^{N} W_{nm} v_m^{\text{old}} < t_n \end{cases} , \tag{5.28}$$

where t_n is the threshold value of neuron n. The interrogation by each neuron is of stochastic nature, independent of the updating of other neurons. The average updating rate is assumed to be equal for all neurons.

A network with symmetric interconnection ($W_{nm} = W_{mn}$) and no direct coupling from a neuron output to its input ($W_{nn} = 0$) always converges to a stable state in a finite number of updatings (Goles et al., 1985). To show this we associate with the total state $\boldsymbol{v} = (v_1, \ldots, v_N)^T$ of the network the energy function

$$H(\boldsymbol{v}) = -\sum_{n=1}^{N} \sum_{m=1}^{N} W_{nm} v_n v_m + 2 \sum_{m=1}^{N} t_m v_m = -\boldsymbol{v}^T W \boldsymbol{v} + 2\boldsymbol{t}^T \boldsymbol{v} . \tag{5.29}$$

The interconnections and the thresholds are specified by means of the matrix $W = [W_{nm}]$ and the vector $\boldsymbol{t} = (t_1, \ldots, t_N)^T$. Next we consider the change in energy

$$\Delta H = H(\boldsymbol{v}^{\text{new}}) - H(\boldsymbol{v}^{\text{old}}) . \tag{5.30}$$

Under the assumption that (only) the state of neuron k is updated (5.30) simplifies to

$$\Delta H = -2(v_k^{\text{new}} - v_k^{\text{old}}) \left[\sum_{m=1}^{N} W_{km} v_m^{\text{old}} - t_k \right] , \tag{5.31}$$

where we have used the fact that $W_{nm} = W_{mn}$ and $W_{nn} = 0$. When we combine (5.31) with the updating law (5.28) we find that during each update the energy is non-increasing, i.e.

$$\Delta H \leq 0 \ , \tag{5.32}$$

and decreasing if a state transition occurs. Since the energy has an absolute minimum, the system must converge to a stable state within a finite number of updates. This state has either local minimal energy or global minimal energy.

In order to simulate the above model on a digital computer we introduce discrete time-steps and select by means of a selection rule one neuron every time-step. (The selection rule should be of such a form that the neurons are selected with equal probability.) Only the selected neuron is allowed to change its state, using (5.28). For this reason we refer to this algorithm as sequential updating. Using the above energy argument we can conclude that this algorithm reaches a stable solution within a finite number of steps.

We remark that if $\sum_{m=1}^{N} W_{nm} v_m = t_n$ the state of the neuron remains unchanged. Since such an event hardly ever occurs we can reformulate the updating law as

$$v_n := \text{sign} \ \left[\sum_{m=1}^{N} W_{nm} v_m - t_n \right] \ , \tag{5.33}$$

with the sign-function defined according to (4.1). The notation := for the transition of v_n^{old} to v_n^{new} is also used in the remainder of this chapter.

5.2.2 A continuous, deterministic model

In the previous model the neurons have two states and the updating of the neurons is instantaneous at random times. Biological neurons and operational amplifiers (in a hardware realization of the mathematical model) have a continuous input-output relation. Moreover, due to non-avoidable integrative time-delays the dynamic behaviour of the system is described by a differential equation, rather than an updating law without memory. In this context, a continuous, deterministic model based on the above properties has been proposed by Hopfield (1984).

The input-output relation of the neurons is now modeled by a sigmoid function, shown in Figure 5.2. This continuous function is monotonically increasing and bounded. In accordance with the previous model we take for the asymptotical values ±1.

An electrical model for the continuous, deterministic system is shown in Figure 5.3. The sigmoid function represents the input-output characteristic $v_n = f(u_n)$ of nonlinear amplifiers with negligible response time and negligible input current. Output v_m is fed back to input u_n by means of a resistor R_{nm}. The differential equation which describes the dynamic behaviour of the system is found by formulating

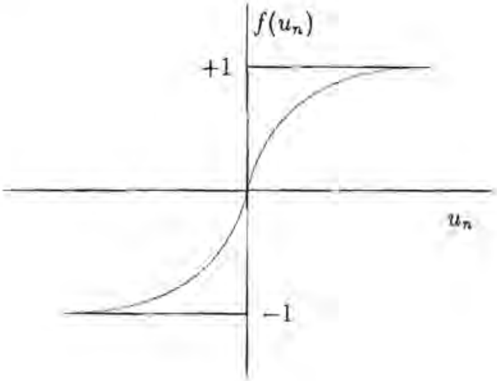


Figure 5.2: The sigmoid function as a continuous input-output relation of a neuron.

Kirchhoff's current relation for the input of the amplifiers:

$$\sum_{m=1}^{N} \frac{u_n - v_m}{R_{nm}} + \frac{u_n}{R_n} + C_n \frac{\mathrm{d}u_n}{\mathrm{d}t} + j_n = 0 \,, \tag{5.34}$$

where R_{nm}, R_n and C_n are assumed to be positive. Equation (5.34) is simplified by means of the introduction of the total conductance to ground

$$\frac{1}{\rho_n} = \frac{1}{R_n} + \sum_{m=1}^{N} \frac{1}{R_{nm}} \,, \tag{5.35}$$

for each neuron. This leads to

$$\rho_n C_n \frac{\mathrm{d}u_n}{\mathrm{d}t} + u_n = \sum_{m=1}^{N} \frac{\rho_n}{R_{nm}} v_m - \rho_n j_n \,. \tag{5.36}$$

Furthermore, under the assumption that the neurons have identical time constants $\tau = \rho_n C_n$ we finally find the set of N nonlinear differential equations

$$\tau \frac{\mathrm{d}u_n}{\mathrm{d}t} + u_n = \sum_{m=1}^{N} W_{nm} f(u_m) - t_n \,. \tag{5.37}$$

The interconnection weights are given by $W_{nm} = \rho_n / R_{nm}$, while the thresholds are chosen according to $t_n = \rho_n j_n$. With passive resistors only positive weight factors can be obtained. Negative weight factors are obtained by means of additional inverting amplifiers and 'negative signal wires'. This is not shown in Figure 5.3.

In order to show that the continuous model also converges to a stable state, Hopfield (1984) introduces the energy function

$$H = -\sum_{n=1}^{N} \sum_{m=1}^{N} W_{nm} v_n v_m + 2 \sum_{n=1}^{N} t_n v_n + 2 \sum_{n=1}^{N} \int_0^{v_n} f^{-1}(s) \mathrm{d}s \,. \tag{5.38}$$

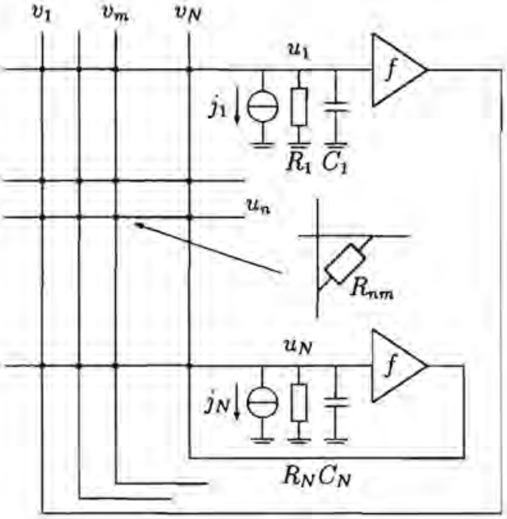


Figure 5.3: Electrical model for Hopfield's neural network

For a symmetric W we find for the derivative of H:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = -2\sum_{n=1}^{N} \frac{\mathrm{d}v_n}{\mathrm{d}t} \left(\sum_{m=1}^{N} W_{nm} v_m - u_n - i_n \right) . \tag{5.39}$$

Combining this result with (5.37) we have

$$\frac{\mathrm{d}H}{\mathrm{d}t} = -2\tau \sum_{n=1}^{N} \frac{\mathrm{d}f}{\mathrm{d}u_n} \left(\frac{\mathrm{d}u_n}{\mathrm{d}t} \right)^2 . \tag{5.40}$$

Since $f(\cdot)$ is a monotonically increasing function, the energy must decrease in time. Moreover $\mathrm{d}H/\mathrm{d}t = 0$ implies that the system is in equilibrium: $\mathrm{d}u_n/\mathrm{d}t = 0$ for $n = 1, \ldots, N$. In combination with the boundedness of the energy, we can conclude that the system always converges to a stable state with local minimal energy.

Apart from the last term, the energy function (5.38) of the continuous model is similar to the energy function (5.29) of the previous model. Note, however, that the vector $\boldsymbol{v}$ is binary-valued in the previous model and continuous in the present model. When we replace the sigmoid function $f(u)$ in Figure 5.2 with $f(\beta u)$ the last term in (5.38) turns into

$$\frac{2}{\beta} \sum_{n=1}^{N} \int_0^{v_n} f^{-1}(s) \mathrm{d}s . \tag{5.41}$$

With the constant β the gain of the sigmoid function can be adjusted; in the high-gain limit $\beta \rightarrow \infty$ the sigmoid function becomes the sign-function and $\boldsymbol{v}$ is binary-valued. As the last term in (5.38) vanishes for $\beta \rightarrow \infty$ we find that for the stable states of the continuous model the outputs of the neurons are binary-valued. Moreover, if we take the diagonal elements of W equal to zero, the stable states of the continuous model coincide with those of the discrete model. In contrast with the previous model such an assumption was not necessary to prove that the continuous model converges to a stable state.

For convenience we rewrite (5.37) as

$$\tau \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{u} = W f(\boldsymbol{u}) - \boldsymbol{t} \ , \tag{5.42}$$

where we have introduced the state vector $\boldsymbol{u} = (u_1, \ldots, u_N)^T$ and the threshold vector $\boldsymbol{t} = (t_1, \ldots, t_N)^T$. The sigmoid function is applied component-wise. The right-hand term in (5.42) depends nonlinearly on $\boldsymbol{u}$. However, in the high-gain limit the right-hand term changes only if the sign of (at least) one of the elements of $\boldsymbol{u}$ changes. Between such events, the right-hand term remains constant, resulting in a linear differential equation which can be solved analytically (Klijn, 1991). Given the state $\boldsymbol{u}$, we determine the moment when one of the neurons changes its output first. Next, the right-hand term is updated and the process is repeated. In this way exact simulation of the system (5.42) is possible.

An alternative way to solve (5.42) numerically, which is not restricted to the high-gain limit, is to introduce discrete time-steps $t = j\Delta t$ and approximate the derivative in (5.42) by a finite difference. Using a first-order approximation this leads to

$$\tau \frac{\boldsymbol{u}(j) - \boldsymbol{u}(j-1)}{\Delta t} + \boldsymbol{u}(j-1) = W f(\boldsymbol{u}(j-1)) - \boldsymbol{t} \ , \tag{5.43}$$

with $\boldsymbol{u}(j) = \boldsymbol{u}(j\Delta t)$. With the introduction of the constant $\alpha = \Delta t / \tau$ we find that the new state $\boldsymbol{u}(j)$ is determined by the old state according to

$$\boldsymbol{u}(j) = \alpha \left[W f(\boldsymbol{u}(j-1)) - \boldsymbol{t} \right] + (1-\alpha) \boldsymbol{u}(j-1) \ . \tag{5.44}$$

The step-size Δt should be small compared to the time constant τ of the neurons, that is, we should take $\alpha \ll 1$.

In general, a number of neurons change their binary output when the sign-function is applied in the iteration (5.44). This is in contrast with the exact simulation of (5.42), where only one neuron changes its binary output at a time. This method, which we call parallel updating, is thus an approximate simulation of the dynamic behaviour of the continuous model.

In equilibrium we have

$$\boldsymbol{v} = \operatorname{sign} \left[W\boldsymbol{v} - \boldsymbol{t} \right] \ , \tag{5.45}$$

where we have used $\boldsymbol{v} = \text{sign}(\boldsymbol{u})$. With sequential updating (5.33) we try to satisfy (5.45) for one element of $\boldsymbol{v}$ at a time. The same set of equations is derived from the parallel updating law (5.44) when $\alpha = 1$. In that case we apply the sequential updating law for all neurons in parallel. It is possible to show (Bruck and Goodman, 1988) that this results in an oscillatory behaviour (a cycle of length 2) of the state of the network.

5.3 Finding a binary hologram with a Hopfield neural network

5.3.1 Introduction

In the previous section we have discussed the two models proposed by Hopfield: a discrete, non-deterministic model and a continuous, deterministic model. Both systems converge to a stable state with local minimal energy

$$H(\boldsymbol{v}) = -\boldsymbol{v}^T W \boldsymbol{v} + 2\boldsymbol{t}^T \boldsymbol{v} \ . \tag{5.46}$$

(For the continuous model we have assumed the high-gain limit of the sigmoid function f). This property suggests applying the Hopfield network in combinatorial optimization problems where the cost function can be formulated as an energy function (Hopfield and Tank, 1985). This is the case with binary holograms, where the cost function

$$C(\boldsymbol{b}) = \boldsymbol{b}^T R \boldsymbol{b} - 2\boldsymbol{\phi}^T R \boldsymbol{b} \tag{5.47}$$

is of the same form as the energy function (5.46). This approach has been suggested by Anastassiou (1989) for the related problem of digital image quantization, and was first applied for computer-generated holograms by Just and Ling (1991).

Equating (5.47) and (5.46) we shall find that the weights and the thresholds of the network are given by

$$\begin{aligned} W &= -R + rI \\ \boldsymbol{t} &= -R^T \boldsymbol{\phi} = -\boldsymbol{\phi} \ . \end{aligned} \tag{5.48}$$

In the discrete model, the matrix W is assumed to be symmetric with zero elements on its main diagonal. In Section 5.1 we have seen that the matrix R is symmetric, but has a nonzero main diagonal with equal elements $R_{nn} = r$. For this reason, the additional term rI (with I the identity matrix) has been introduced in (5.48). Since the object is assumed to vanish outside the object window, the filtering by the matrix R has no effect and we have $\boldsymbol{t} = -\boldsymbol{\phi}$. Given the weights and the thresholds the neural network is completely determined. Starting with an initial configuration we let the network converge to a stable state. The binary samples of the (local) optimal hologram are then given as the neuron outputs.

The simulation of the discrete model on a digital computer results in an algorithm where only one neuron is updated every time-step (sequential updating). The simulation of the continuous model results in an algorithm where all neurons are allowed to change their state simultaneously (parallel updating). In the next subsections both approaches are discussed.

5.3.2 Sequential updating

In this section we elaborate on sequential updating. Following the previous sections, we consider one-dimensional signals. Later, the results are extended to two-dimensional signals. Furthermore, the relation between sequential updating and the direct binary search method (Seldowitz et al., 1987) is discussed.

When we substitute the weights and the thresholds (5.48) in the sequential updating law

$$v_n := \text{sign } \left[\sum_{m=1}^{N} W_{nm} v_m - t_m \right] , \tag{5.49}$$

we find

$$b_n := \text{sign } \left[- \sum_{m=1}^{N} R_{nm} b_m + \phi_n + r b_n \right] . \tag{5.50}$$

For convenience we derive an explicit expression for the diagonal elements r of the matrix R. If a continuous frequency region $\mathbb{F}$ is considered in the optimization problem we find

$$r = R_{nn} = \int_{\langle 1 \rangle} |A(\theta)|^2 \mathrm{d}\theta . \tag{5.51}$$

This follows directly from (5.6). When discrete frequencies are considered, we can write for the correlation matrix

$$R_{nm} = (F^{-1} A^H A F)_{nm} = \sum_{k=1}^{N} \sum_{l=1}^{N} F_{nk}^{-1} (A^H A)_{kl} F_{lm} . \tag{5.52}$$

Since $(A^H A)$ is a diagonal matrix, we have

$$R_{nm} = \sum_{k=1}^{N} F_{nk}^{-1} (A^H A)_{kk} F_{km} . \tag{5.53}$$

Using the properties $F_{nk}^{-1} = F_{kn}^{*}/N$ and $|F_{kn}|^2 = 1$ of the Fourier transform matrix, the diagonal elements are given by

$$r = R_{nn} = \sum_{k=1}^{N} F_{nk}^{-1} (A^H A)_{kk} F_{kn} = \frac{1}{N} \sum_{k=1}^{N} (A^H A)_{kk} = \frac{1}{N} \text{Trace } A^H A . \tag{5.54}$$

We now show that the sequential updating law of a discrete-time Hopfield neural network can be seen as a generalization of error diffusion. To this end we write for the correlation matrix in (5.48) $R = I + C$, with $C_{nn} = 0$. Scaling the diagonal elements of R to unity can be done without loss of generality. Since the Toeplitz matrix C acts as a convolution-operator, we find for the updating law (5.49)

$$b[n] := \text{sign}\ (-c[n] * b[n] + \phi[n] + c[n] * \phi[n])\ . \tag{5.55}$$

Comparing this result with (4.79) we conclude that with the sequential updating law we are able to find a solution of the symmetrical error-diffusion problem.

With sequential updating we start with an initial configuration $\boldsymbol{b}$, where each sample b_n has been assigned a value ±1 with equal probability. Next, the neurons are selected in random order (with equal frequency). To this end a sequence of integers is generated using a (pseudo) random number generator with a uniform distribution on $\{1, 2, \ldots, N\}$. For one neuron at a time (5.50) is applied. This algorithm requires $O(N)$ multiplications per neuron update. Due to the Toeplitz structure of R we only need $O(N)$ storage.

The matrix-vector product $R\boldsymbol{b}$ in (5.50) describes a one-dimensional convolution. A generalization for two-dimensional signals thus implies replacement by a two-dimensional convolution, where the filter coefficients are determined by the two-dimensional object window $\mathbb{F}$. Following the vector-representation of one-dimensional signals, the most natural way to represent a two-dimensional signal is by a matrix:

$$B = [\boldsymbol{b}_1 \ldots \boldsymbol{b}_N]\ . \tag{5.56}$$

The two-dimensional signal $b[n_1, n_2]$ is considered as a number of one-dimensional signals $\boldsymbol{b}_{n_2}$, which form the columns of the matrix [1]. In the same way we introduce the matrix Φ for $\phi[n_1, n_2]$. The formulation of a two-dimensional convolution in matrix-notation is in general not convenient. In order to adopt the one-dimensional formulation we concatenate the columns of B and construct a $N^2 \times 1$ vector $\boldsymbol{b}$ according to

$$\boldsymbol{b} = [\boldsymbol{b}_1^T \ldots \boldsymbol{b}_N^T]^T \tag{5.57}$$

The accompanying matrix R in (5.50) is a $N^2 \times N^2$ matrix with a highly redundant structure (R has only N^2 freedoms). In general, we have to calculate $O(N^2)$ multiplications per neuron update, and we have to store $O(N^2)$ coefficients.

However, in the special case where the object window is separable we can lower the number of multiplications per neuron update to $O(N)$ and the number of storage units to $O(N)$. A separable object window can be written as $\mathbb{F} = \mathbb{F}_1 \times \mathbb{F}_2$, with the one-dimensional object windows $\mathbb{F}_1$ and $\mathbb{F}_2$ for the respective coordinates θ_1 and

[1] Note the difference between the spectrum vector $\boldsymbol{B}$, defined in the Fourier domain, and the matrix B, defined in the spatial domain.

θ_2. The accompanying correlation matrices are denoted by R^c and R^r. The two-dimensional convolution ($R\boldsymbol{b}$) can now be formulated in a matrix notation, where we have a column operator R^c followed by a row operator R^r, according to

$$(R^r(R^cB)^T)^T = R^cBR^{r^T} \ . \tag{5.58}$$

Since the row operator R^r is a symmetrical matrix, the transpose operation T is omitted in the remainder of this section. The $N^2 \times N^2$ matrix R is related to R^c and R^r through the Kronecker tensor product (Barnett, 1990)

$$R = R^r \otimes R^c = \begin{bmatrix} R^r_{11}R^c & \dots & R^r_{1N}R^c \\ \vdots & & \vdots \\ R^r_{N1}R^c & \dots & R^r_{NN}R^c \end{bmatrix} . \tag{5.59}$$

It is possible to show that the term diag $(R)\boldsymbol{b}$ in (5.50) can be written as

$$\text{diag}\,(R^c)B\text{diag}\,(R^r) \ . \tag{5.60}$$

The (constant) diagonal terms $r^c = R^c_{nn}$ and $r^r = R^r_{nn}$ are determined by (5.51) or (5.54). The sequential updating law for two-dimensional signals thus becomes

$$B_{n_1n_2} := \text{sign}\ \left[-R^c_{n_1:}BR^r_{:n_2} + r^cr^rB_{n_1n_2} + \Phi_{n_1n_2}\right] \ . \tag{5.61}$$

The index n_1 : stands for row n_1 of the matrix in question, while : n_2 stands for column n_2. Equation (5.61) still requires $O(N^2)$ multiplications per neuron update, the storage requirement for the matrices R^c and R^r is only $O(N)$ memory units.

The number of multiplications can be lowered by selecting the neurons in a specific order. First, we select a column. Next, every neuron in this column is selected once. The process is then repeated for the next selected column. As soon as a column, say with index n_2, has been selected we calculate the intermediate vector

$$\boldsymbol{h} = BR^r_{:n_2} \ . \tag{5.62}$$

The updating law for the neurons in column n_2 reads

$$B_{n_1n_2} := \text{sign}\ \left[-R^c_{n_1:}\boldsymbol{h} + r^cr^rB_{n_1n_2} + \Phi_{n_1n_2}\right] \tag{5.63}$$

When a transition $B'_{n_1n_2} = -B_{n_1n_2}$ occurs, we have to correct the intermediate vector, according to

$$h_{n_2} := h_{n_2} + (B'_{n_1n_2} - B_{n_1n_2})R^r_{n_2n_2} \ . \tag{5.64}$$

For the updating of N neurons (in the same column) we have to determine the vector $\boldsymbol{h}$ once, which costs $O(N^2)$ multiplications. With $O(N)$ multiplications due to the corrections we find that the complexity per neuron update is lowered to $O(N)$

multiplications. The columns (and the neurons within each column) are selected using

$$n_2 := 1 + [-1 + n_2 + p] \bmod N \ , \tag{5.65}$$

with the constant $p \in \{1, 2, \ldots, N\}$. When N and p are taken coprime, each column is selected exactly once in N successive updates.

In the computer simulations of sequential updating we have placed the original object in a separable object window, chosen according to

$$\mathbb{F} = \left\{(\theta_1, \theta_2) \mid \tfrac{1}{8} < \theta_1 \le \tfrac{3}{8}, -\tfrac{1}{8} < \theta_2 \le \tfrac{1}{8}\right\} \ . \tag{5.66}$$

Again, the (discrete) object is multiplied by a random phase factor. The respective sizes of the hologram and the object are 128^2 and 32^2 samples. In a first experiment we have calculated a binary hologram using (5.61), where the neurons are selected in random order. The initial binary hologram is a random binary pattern, generated by setting the output of each neuron to ±1 with equal probability. After about $10N^2$ updatings the Hopfield neural network converged to the (suboptimal) binary pattern shown in Figure 5.4a. The modulus of the Fourier transform of the binary hologram is shown in Figure 5.4b. Outside the object windows the quantization noise is uniformly distributed over the reconstruction plane. The signal-to-noise ratio (4.72) of the reconstructed object equals $\mathrm{SNR} = 24$, while the efficiency (4.75) equals $\eta = 0.007$. When the calculation of the hologram was repeated with different

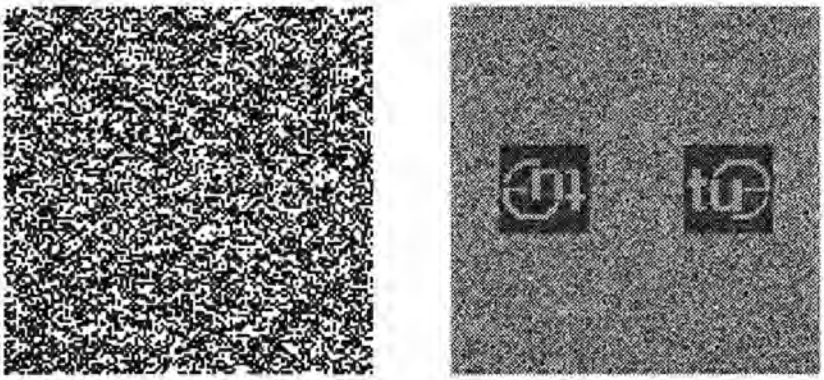


Figure 5.4: a. Binary hologram distribution obtained by means of a Hopfield neural network with sequential updating. b. Modulus of the Fourier transform of the binary hologram.

(random) initial binary patterns, we obtained holograms with about the same signal-to-noise ratio and efficiency. Moreover, also the required number of updatings does not seem to depend on the initial (random) binary hologram.

In a second experiment we have applied the accelerated version of sequential updating (5.63), where the neurons are selected per column. In about 10 iterations a (sub)optimal binary hologram with SNR $= 23$ and $\eta = 0.007$ was obtained. The binary hologram and its reconstruction resemble those shown in Figure 5.4. The results did not significantly change when the order of selection of the columns (and the neurons within each column) was altered. Apparently, selecting the neurons in a special order does not have a major influence on the required number of updatings nor on the quality of the binary hologram.

In order to make a comparison with recursive error diffusion possible we have designed an (internal) error feedback filter of order 2 using the least-square approach of Chapter 4. Applying the error diffusion algorithm resulted in a binary hologram with a signal-to-noise ratio of 30 and an efficiency $\eta = 0.007$. In spite of the extra computation, a smaller signal-to-noise ratio is obtained with Hopfield's neural network.

Still, applying a Hopfield neural network can be profitable. We have taken the binary hologram obtained with error diffusion as the initial neuron outputs. During 8 iterations the Hopfield network can improve the binary hologram, resulting in the binary pattern shown in Figure 5.5a. In the binary hologram we can still observe a structure due to the initial binary hologram. As a result the quantization noise is not uniformly distributed outside the object windows (Figure 5.5b); the noise shaping

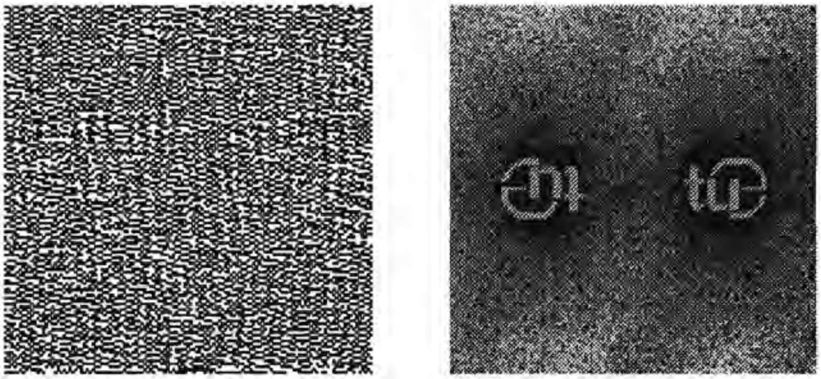


Figure 5.5: a. Binary hologram distribution obtained by means of a Hopfield neural network with sequential updating. The initial binary hologram is determined with error diffusion. b. Modulus of the Fourier transform of the binary hologram.

characteristic of the error diffusion system is still present. The signal-to-noise ratio is improved to SNR = 37, while the efficiency remains $\eta = 0.007$.

We conclude this section with a short discussion of the direct binary search algorithm (Seldowitz et al., 1987). This algorithm starts with a random binary configuration and searches for a better hologram in a local neighbourhood. To this end the effect of the transition of one binary sample on the approximation of the original object is considered. The local neighbourhood of a given binary hologram is thus defined as the set of binary holograms which differ exactly in one pixel (Hamming distance 1). If the approximation has improved we accept the transition, otherwise the old value is restored. This process is repeated until a local optimal solution is obtained.

Although the direct binary search (DBS) algorithm and sequential updating are based on the same concept, there are important differences. In contrast to the quadratic energy function of Hopfield's neural network, the cost function of the DBS method is less restricted. While the neural network in its present form minimizes both amplitude and phase errors, a DBS algorithm can be easily constructed for the minimization of amplitude errors only. Since the DBS method is rather slow, an accelerated version has been proposed (Jennison et al., 1991). However, only for amplitude-phase optimization problems acceptable calculation times are obtained. The authors suggest to solve amplitude-only optimization problems as an amplitude-phase optimization problem, with an arbitrarily chosen phase (see also p. 12). With Hopfield's neural network the complexity can be further decreased by means of parallel updating, which is discussed in the next subsection.

5.3.3 Parallel updating

In the discussion of parallel updating we first consider one-dimensional signals. With the weights and thresholds given by (5.48), the parallel updating law (5.44) turns into

$$\begin{aligned} \boldsymbol{u}(j) &= \alpha\left[-R\boldsymbol{b}(j-1) + r\boldsymbol{b}(j-1) + \boldsymbol{\phi}\right] + (1-\alpha)\boldsymbol{u}(j-1) \\ \boldsymbol{b}(j) &= \operatorname{sign}\left[\boldsymbol{u}(j)\right] . \end{aligned} \tag{5.67}$$

Although the continuous model does not require that the diagonal elements of W are equal to zero, such a choice is appropriate in the numerical approximation of the continuous model. With $O(N^2)$ multiplications for the updating of N neurons, we find that the complexity per neuron update is the same as with sequential updating. However, with the discrete frequency approximation, we can make use of the fast Fourier transformation and lower the complexity. Such an approach did not make sense for sequential updating, where one neuron was updated at a time. With the discrete frequency approximation we have $R = F^{-1}AA^HF$, and (5.67) turns into

$$\begin{aligned} \boldsymbol{u}(j) &= \alpha\left[-F^{-1}A^HAF\boldsymbol{b}(j-1) + r\boldsymbol{b}(j-1) + \boldsymbol{\phi}\right] + (1-\alpha)\boldsymbol{u}(j-1) \\ \boldsymbol{b}(j) &= \operatorname{sign}\left[\boldsymbol{u}(j)\right] . \end{aligned} \tag{5.68}$$

When we use the fast Fourier transformation for the operation $F^{-1}A^HAF$ we need $O(N \log N)$ multiplications per iteration or only $O(\log N)$ multiplications per neuron update, which leads to considerable computational savings.

The parallel updating law is easily generalized for two-dimensional signals, where $b[n_1, n_2]$, $\phi[n_1, n_2]$ and $u[n_1, n_2]$ are denoted by the respective matrices B, Φ and U. For the discrete Fourier transform of B we can write $FBF^T = FBF$ since the discrete Fourier transformation is a separable operation (cf. (5.58)). In this way we find for the two-dimensional updating law

$$\begin{aligned} U(j) &= \alpha \left[-F^{-1}(A^HA \circ (FB(j-1)F))F^{-1} + rB(j-1) + \Phi \right] + \\ &\qquad\qquad\qquad\qquad\qquad\qquad (1-\alpha)U(j-1) \\ B(j) &= \text{sign } [U(j)] \ . \end{aligned} \tag{5.69}$$

The filtering in the frequency domain with the mask A^HA is denoted by means of the Hadamard product $\circ$, which stands for component-wise multiplication (Barnett, 1990), that is

$$\begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \\ x_{N1} & \cdots & x_{NN} \end{bmatrix} \circ \begin{bmatrix} y_{11} & \cdots & y_{1N} \\ \vdots & & \vdots \\ y_{N1} & \cdots & y_{NN} \end{bmatrix} = \begin{bmatrix} x_{11}y_{11} & \cdots & x_{1N}y_{1N} \\ \vdots & & \vdots \\ x_{N1}y_{N1} & \cdots & x_{NN}y_{NN} \end{bmatrix} . \tag{5.70}$$

According to (5.69) the fast Fourier transformation is applied separately on the rows and the columns of B, resulting in $O(N^2 \log N)$ multiplications per iteration. Consequently, we have $O(\log N)$ multiplications per neuron update.

In order to determine the constant r for the 'diagonal terms', we have to calculate the contribution of an (arbitrary) sample B_{nm} to the product

$$\begin{aligned} &\left[F^{-1}((A^HA) \circ (FBF))F^{-1} \right]_{nm} = \\ &\qquad\qquad \sum_{k=1}^{N} \sum_{l=1}^{N} F_{nk}^{-1}(A^HA)_{kl} \left[\sum_{p=1}^{N} \sum_{q=0}^{N} F_{kp}B_{pq}F_{ql} \right] F_{lm}^{-1} \ . \end{aligned} \tag{5.71}$$

This contribution reads

$$r = \sum_{k=1}^{N} \sum_{l=1}^{N} F_{nk}^{-1}(A^HA)_{kl}F_{kn}F_{ml}F_{lm}^{-1} \ , \tag{5.72}$$

and can be simplified to

$$r = \frac{1}{N^2} \sum_{k=1}^{N} \sum_{l=1}^{N} (A^HA)_{kl} \ , \tag{5.73}$$

since $F_{kn}F_{nk}^{-1} = \frac{1}{N}$ and $F_{ml}F_{lm}^{-1} = \frac{1}{N}$.

The choice of the step-size α in the parallel updating law is a compromise between speed and quality. For a small value of α only a small number of neurons will change their output in each iteration step. As a result, a large number of iteration steps are required before a reasonable signal-to-noise ratio is obtained. For larger values of α the convergence speed is higher, but due to the increased inaccuracy in the numerical approximation (5.67) a larger number of neurons will make a wrong decision in the updating. As a result, the signal-to-noise ratio is smaller. The binary output of the network shows an oscillatory behaviour when the the step-size is increased further ($\alpha \rightarrow 1$). In Figure 5.6 the evolution of signal-to-noise ratio for different choices of α is shown. Starting with a rather large step-size which is gradually decreased during the process, one obtains a 'good' signal-to-noise ratio in a minimal number of iteration steps. In our experiments, however, we have taken a constant step-size $\alpha = 0.2$, which gives a good compromise between convergence speed and quality. This implies that the network does not converge to a stable state; the process is stopped as soon as the signal-to-noise ratio does not change significantly. After 30 iterations the signal-to-noise ratio equals 22, which is about the same as for sequential updating. Also the diffraction efficiency remains $\eta = 0.007$. The initial value of the matrix U is chosen at random, with $|U_{n_1 n_2}| \leq 1$. In general, the required number of iterations is larger for parallel updating than for sequential updating (about 10). Still, the reduction in complexity (per iteration) for parallel updating

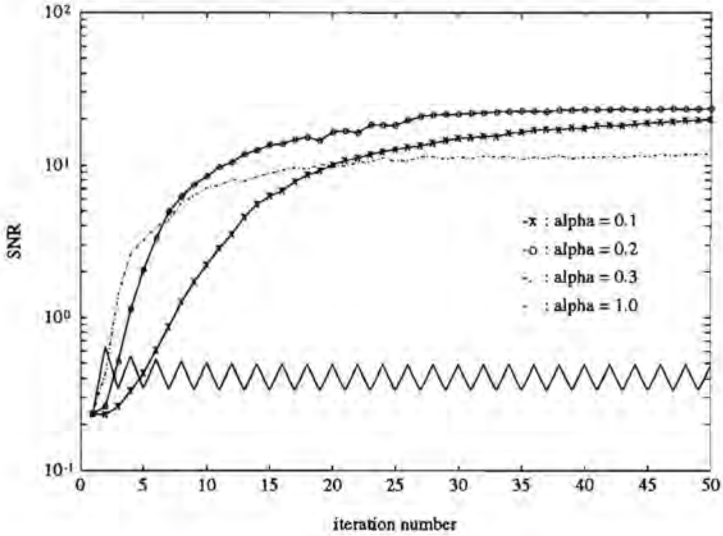


Figure 5.6: Evolution of the signal-to-noise ratio for different choices of the step-size α.

causes a decrease in computation time. Compared to error diffusion, however, the performance of the Hopfield neural network remains rather disappointing.

So far we tried to generate the (discrete) object $\Psi[k_1, k_2]$ within the object window, where the accompanying $\phi[n_1, n_2]$ is scaled according to $\max \phi = 1$. Instead we could try to generate $\mu\Psi$ within the object window, with the real factor $\mu > 1$. Provided that the noise power does not increase, this will result in a larger signal-to-noise ratio. Moreover, the efficiency will increase due to the larger diffraction efficiency. The binary hologram obtained with $\mu = 2$ is shown in Figure 5.7. The signal-to-noise ratio equals 80, while the efficiency is increased to 0.03. In the binary hologram we recognize the structure of the original hologram. According to (5.45) we have in equilibrium (for one-dimensional signals)

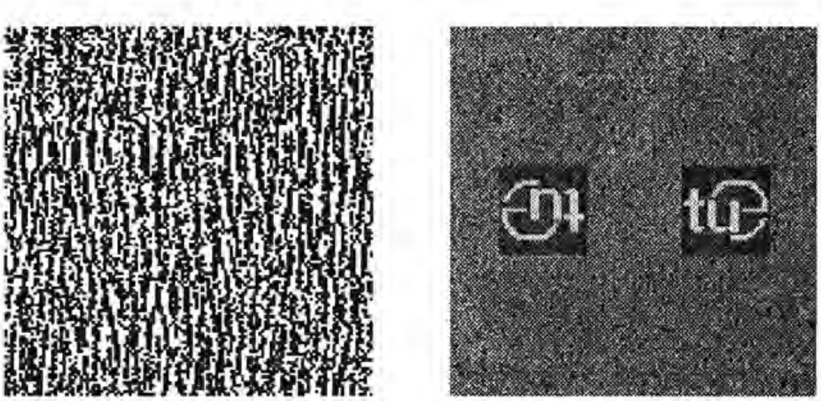


Figure 5.7: a. Binary hologram distribution obtained by means of a Hopfield neural network with parallel updating and $\mu = 2$. b. Modulus of the Fourier transform of the binary hologram.

$$\boldsymbol{b} = \text{sign } [-\boldsymbol{R}\boldsymbol{b} + r\boldsymbol{b} + \mu\boldsymbol{\phi}] \ . \tag{5.74}$$

For a large factor μ the stationary state goes toward $\boldsymbol{b} = \text{sign } (\boldsymbol{\phi})$, which explains the structure of binary hologram. This means that when μ is taken too large, the signal-to-noise ratio decreases. With $\mu = 2$ a scaling factor $\lambda \approx 1$ is obtained.

In amplitude-only optimization we have the freedom to choose the phase of Ψ (Wyrowski, 1990). So far, the phase freedom has been exploited by the introduction of a random phase distribution for the original object. A solution for the amplitude-only optimization problem (2.30) was found by minimizing the amplitude-phase measure (2.29). However, since this approach implies the approximation of the (randomly chosen) phase of the object, a suboptimal result is obtained. In Section 5.1 we have already noted that the amplitude-phase optimization problem and the amplitude-only optimization problem are equivalent if the phase of Φ is taken equal to the phase of B. Of course, in order to know the phase of B the hologram samples are needed, and vice versa. To settle this problem, we try to determine the phase of Φ in an iterative way, according to $\Phi(j) = |\Phi| e^{i \arg B(j-1)}$. For the Hopfield neural network this means that the thresholds are adapted during updating. In our computer simulations we always found that the thresholds converged to stable configuration. Consequently, the network converges to a stable state. A proper explanation, however, of this surprising property has not been found yet. Sequential updating requires adaptation of all thresholds for every single neuron transition and is therefore not considered.

Using the phase freedom allows us to increase the factor μ further and improve both the efficiency and the signal-to-noise ratio. For $\mu = 3$ a binary hologram with an improved signal-to-noise ratio of 120 and an efficiency $\eta = 0.05$ is obtained. We remark that the scaling of the input signal can also be applied with error diffusion in order to obtain a better signal-to-noise ratio and efficiency. Exploiting the phase freedom is, due to its iterative character, not possible in the error diffusion process.

We end this section with a discussion of the similarities of parallel updating with the iterative Fourier-transform algorithm. The parallel updating law (5.68) describes an iteration between the hologram plane and the reconstruction plane, as shown in Figure 5.8 (for one-dimensional signals). The sign-operation (or otherwise the sigmoide-operation) is applied in the hologram plane, while in the reconstruction plane the Fourier transform of the state $\boldsymbol{U}$ is adjusted. The structure of the parallel updating algorithm is also found with the iterative Fourier-transform algorithm. This algorithm is applied in order to generate a signal subject to constraints in both the hologram and the reconstruction plane. For each constraint there is a

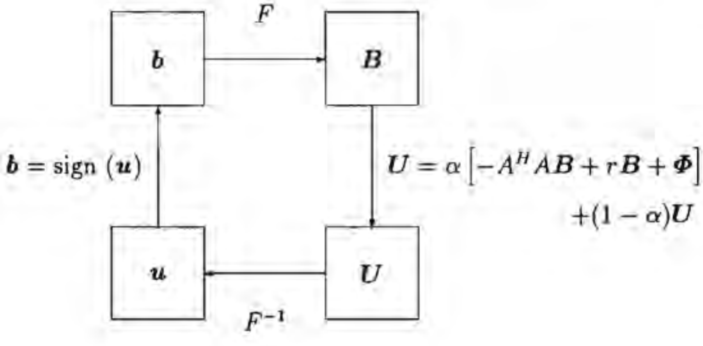

F
$\boldsymbol{b}$
$\boldsymbol{B}$
$\boldsymbol{b} = \text{sign}\ (\boldsymbol{u})$
$\boldsymbol{U} = \alpha\left[-A^H A\boldsymbol{B} + r\boldsymbol{B} + \boldsymbol{\Phi}\right]$
$+(1-\alpha)\boldsymbol{U}$
$\boldsymbol{u}$
$\boldsymbol{U}$
F^{-1}

Figure 5.8: Iteration between hologram plane and reconstruction plane.

set consisting of signals which satisfy this constraint. Provided that the iterative process converges, a member of the intersection of the sets is found.

In the special case where applying a constraint implies a projection on a closed convex set, the iterative process is guaranteed to converge (the intersection of the sets is assumed non-empty). This method is known as 'projections on convex sets' (Stark, 1992). For example, the properties $-1 \leq b_n \leq 1$ and $B_k = \Phi_k$ for $k \in \mathbb{F}$ both define convex sets $\boldsymbol{b}$. In order to find a signal with both properties, the projections

$$b_n = \begin{cases} 1 & u_n > 1 \\ u_n & \text{if } |u_n| \leq 1 \\ -1 & u_n < -1 \end{cases} \tag{5.75}$$

and

$$U_k = \begin{cases} \Phi_k & \text{if } k \in \mathbb{F} \cup \mathbb{F}^* \\ B_k & \text{elsewhere} \end{cases} \tag{5.76}$$

are applied in the hologram and the reconstruction plane, respectively. With the quantization of Fourier holograms, however, we have the sign-operator, which is not a projection on a convex set. Apparently, when the above operation $\boldsymbol{U} = -A^H A\boldsymbol{B} + \boldsymbol{B} + \boldsymbol{\Phi}$ is adjusted to the operation in Figure 5.8, the process still converges. With the iterative Fourier-transform algorithm more sophisticated constraints are imposed on the signals in both planes. This has been successfully applied in the quantization of Fourier holograms by Wyrowski (1989).

5.4 The Boltzmann machine

5.4.1 Simulated annealing

The discrete-time Hopfield neural network with sequential updating accepts only a transition to a new state if this results in a decrease in energy. It is due to this condition that the discrete-time Hopfield network is guaranteed to converge to a stable state in a finite number of transitions. Such a strict downhill search, however, has a major disadvantage. The energy of a stable state is (by definition) minimal in a local sense. Since the Hopfield network is unable to escape from such states, it is thus possible that the network converges to a state with an energy much higher than the global minimal energy. By means of a stochastic technique called simulated annealing it is, however, possible to circumvent such an undesirable situation.

In metallurgical annealing a metallic body is heated in a heat bath to a temperature at which the body (nearly) melts. At melting temperature the particles in the solid arrange themselves randomly, and dislocations in the lattice are eliminated. Next, the temperature of the heat bath is decreased very slowly, which prevents the formation of new dislocations. In this way one obtains a highly structured lattice at room temperature. The energy function of the metal is minimal for this state. When the temperature of the heat bath is lowered too quickly, the metal will be frozen into a meta-stable state with a higher energy due to lattice dislocations. The extreme case in which the temperature is lowered instantaneously, is known as quenching.

In 1953 Metropolis et al. introduced an algorithm, based on Monte Carlo techniques, for the simulation of a solid at thermal equilibrium. Given the state of the system, a new state is proposed by means of the introduction of a small change in (a part of) the system. If this change gives rise to a state with lower system energy, this new state is accepted. In case the change results in an increase of the system energy one accepts the new state with probability

$$P = e^{-\frac{E^{\text{new}} - E^{\text{old}}}{k_B T}} , \tag{5.77}$$

where E^{new} denotes the system energy of the proposed state and E^{old} denotes the system energy of the current state. The constant k_B is known as the Boltzmann constant, T is the temperature of the system. When a large number of state transitions is considered (at a constant temperature), the solid reaches thermal equilibrium which is characterized by the Boltzmann distribution (Aarts and Korst, 1989). This algorithm is known as the Metropolis algorithm. The simulation of metallurgical annealing was introduced by Kirkpatrick et al. (1982, 1983) and independently by Černy (1985). With simulated annealing the Metropolis algorithm is applied with a slowly decreasing temperature. This way a state with an energy equal (or very close) to the global minimum of the system energy is obtained. Due to this property simulated annealing is applied as a technique to solve combinatorial optimization problems. The solutions of the optimization problem are the states of the physical system, while the cost function of each solution is represented by the energy of the accompanying state.

Combining simulated annealing with the updating law of a Hopfield neural network makes escaping from local minima possible. The resulting neural network, which is known as the Boltzmann machine was introduced by Hinton et al. (1984). In the next subsection the Boltzmann machine is considered in more detail.

5.4.2 Sequential updating

Given a discrete-time Hopfield neural network in a certain state, we propose a new state by means of the inversion of one neuron output: $v_k^{\text{new}} = -v_k^{\text{old}}$. According to (5.31) the change in energy due to this transition reads

$$\Delta H = 4 v_k^{\text{old}} u_k^{\text{new}} , \tag{5.78}$$

where we have introduced the neuron input

$$u_k^{\text{new}} = \sum_{m=1}^{M} W_{km} v_m^{\text{old}} - t_k . \tag{5.79}$$

With the original Hopfield neural network the proposed transition is accepted only if this results in a decrease in energy. For a Boltzmann machine the updating law (5.33) is modified according to (Hecht-Nielsen, 1990)

$$v_k^{\text{new}} = \begin{cases} -\ v_k^{\text{old}} & \text{if } \Delta H < 0 \\ -\ v_k^{\text{old}} & \text{if } \Delta H \geq 0 \text{ and } \xi < e^{-\Delta H/c} \\ \ v_k^{\text{old}} & \text{otherwise} \end{cases} \tag{5.80}$$

In contrast with the original discrete-time Hopfield neural network, transitions which lead to an increase in energy are accepted with a certain probability. To this end the random number ξ is chosen on $[0, 1)$ with a uniform probability density function. The control parameter c plays the role of a temperature. The initial value of c is chosen such that almost all transitions are accepted. During the process the 'temperature' is decreased slowly. As the temperature goes towards zero only energy-decreasing transitions are accepted. The Boltzmann machine then functions as a true discrete-time Hopfield network, and thus must reach a stable state.

5.4.3 Parallel updating

In Subsection 5.3.3 we have seen that the complexity of a discrete-time Hopfield network with (fully) parallel updating is much smaller than for a Hopfield network with sequential updating. From this point of view we would like to construct a Boltzmann machine with parallel updating. In principle, a new state can be generated by inverting the output of a selected number of neurons. If this leads to a decrease in energy the new state is accepted; for an energy-increasing transition the Metropolis criterion (5.77) is applied. This way, however, a decision is made for a group of neurons rather than for individual neurons. This is in contrast to the main idea behind parallel computation with neural networks: each neuron is able to make its own decision. Moreover, this approach is not suitable for fully parallel updating. Here, we discuss a parallel Boltzmann machine which is based on the parallel updating law (5.44) for the Hopfield network. In analogy with (5.79) for sequential updating, we first determine

$$\boldsymbol{u}(j) = \alpha\left[W\boldsymbol{v}(j-1) - \boldsymbol{t}\right] + (1-\alpha)\boldsymbol{u}(j-1)\ . \tag{5.81}$$

Next, the acceptance-criterion

$$v_k(j) = \begin{cases} -\ v_k(j-1) & \text{if } \Delta\tilde{H}_k < 0 \\ -\ v_k(j-1) & \text{if } \Delta\tilde{H}_k \geq 0 \text{ and } \xi < e^{-\Delta\tilde{H}_k/c} \\ \ v_k(j-1) & \text{otherwise} \end{cases} \tag{5.82}$$

is applied for all neurons. For the 'change in energy per neuron' $\Delta\tilde{H}_k$ we introduce

$$\Delta\tilde{H}_k = 4v_k(j-1)u_k(j)\ , \tag{5.83}$$

similar to (5.78) for sequential updating. We admit that the term 'change in energy per neuron' is misleading. Each neuron is able to decide whether a transition is appropriate or not, dependent on the value of its partial energy. The sum of all the partial energy-changes $\Delta \tilde{H}_k$, however, is not equal to the change in the total energy ΔH. When the temperature goes to zero, only energy-decreasing transitions are accepted. The parallel Boltzmann machine then functions as the original Hopfield network with parallel updating. Under the assumption that α is small, the Boltzmann machine thus reaches a stable state.

In the last computer experiment we have simulated a Boltzmann machine with adaptive thresholds. The step-size α is taken 0.1, while the scaling factor equals $\mu = 3$. With a starting value of the control parameter $c = 50$ all neuron transitions are accepted. During the process c is slowly decreased, according to

$$c(j) = \gamma c(j-1) \ . \tag{5.84}$$

The evolution of the signal-to-noise ratio is shown in Figure 5.9 for the respective γ-values 0.95 and 0.85. In addition, the signal-to-noise ratio for a Hopfield network with parallel updating ($c = 0$) is shown as a reference. When the cooling is

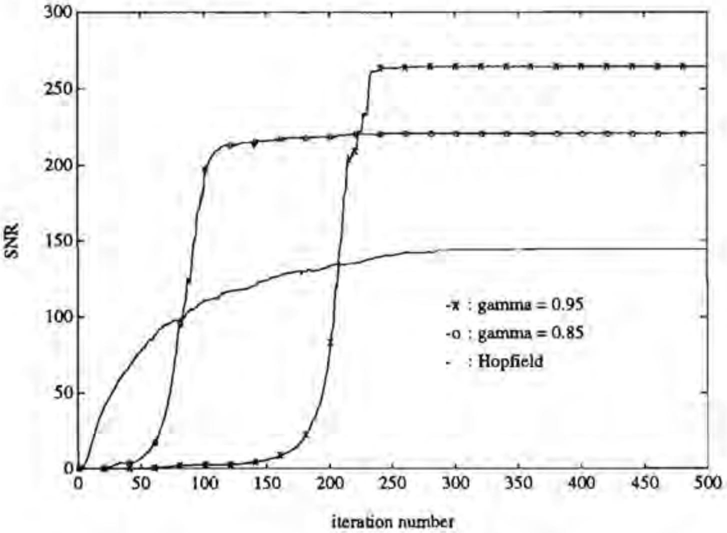


Figure 5.9: Evolution of the signal-to-noise ratio for a Hopfield neural network and for a parallel Boltzmann machine for various values of γ.

sufficiently slow we obtain a binary hologram with a signal-to-noise ratio of about 260 which is significantly larger than the result obtained with a Hopfield network. The efficiency equals 0.06. A major disadvantage of the Boltzmann machine is the large number of required iterations. Still, with a less slow cooling procedure a significantly gain in the signal-to-noise ratio can be achieved, while the number of required iterations remains acceptable.

5.5 Conclusion

In this chapter we have reformulated the hologram quantization problem as a combinatorial optimization problem. A Hopfield network, which is the generalization of error diffusion, can be applied in order to solve such a problem. Given the object and the object window in the reconstruction plane, the weights and the thresholds of the network are determined. Starting with an initial solution, the Hopfield network adjusts its state until a stable state is reached. The local optimal binary hologram is then given by the neuron outputs.

For the simulation of a Hopfield network on a digital computer, we can apply sequential updating or parallel updating. With sequential updating the state of one neuron is updated at a time. Selecting the neurons in a specific order makes a reduction in complexity from $O(N^2)$ multiplications per neuron update to $O(N)$ possible, while the total number of updatings remains the same. The signal-to-noise ratio of the holograms obtained with a Hopfield neural network is, compared to recursive error-diffusion, rather disappointing. The local search process of the Hopfield neural network stops at a local optimal solution, while better solutions exist. The Boltzmann machine, which combines the sequential updating law with simulated annealing, is able to escape from local optimal solutions. Due to its complexity, the 'sequential' Boltzmann machine has not been implemented.

With parallel updating all neurons are allowed to change their state simultaneously. Due to the special structure of the weight matrix, the fast Fourier transformation can be applied in order to reduce the complexity per neuron update further to $O(\log N)$. Although parallel updating requires more neuron updates than sequential updating an overall savings in computation is obtained for large holograms. The holograms obtained with parallel updating, however, have in general a somewhat smaller signal-to-noise ratio than those obtained with sequential updating. The proposed Boltzmann machine with parallel updating finds holograms with an improved signal-to-noise ratio within a reasonable amount of time.

Chapter 6

Discussion

In this thesis we have investigated how pulse-density modulation can be applied in order to transform a continuous hologram transmittance function into a binary function. Both continuous and discrete pulse-density modulation are considered. With continuous pulse-density modulation the positions of the pulses contained in the binary signal can be chosen continuously in space. This is in contrast with discrete modulation, where the pulses can be placed at fixed raster points only. Given the continuous hologram transmittance function a pulse distribution of the binary hologram has to be determined which forms a good approximation of the original hologram. For Fourier holograms this means that the deviation between the Fourier spectra of the original and the binary hologram should be a small as possible within a given frequency band, called the object window.

Finding the optimal continuous pulse distribution is a multi-dimensional optimization problem with the pulse positions as free parameters. However, solving this problem with standard techniques is intractable for a large number of pulses. A suboptimal solution can easily be found using a simple approach. By means of a running integration a one-dimensional transmittance function can be divided in partial functions with equal 'area'. When each partial signal is replaced by a binary pulse with the same area, a pulse-density signal with a local pulse density proportional to the amplitude of the positive transmittance function is obtained. Equating the (local) area of the transmittance function and the (local) area of the pulse-density signal has the effect that the spectral approximation error vanishes in the vicinity of the origin. With this approach the object window should thus be chosen around or near the origin.

In variations on this concept the position of small groups of pulses is determined for adjacent partial functions simultaneously. Theoretically, a smaller approximation error is obtained for the higher-order methods. Computer simulations, however, have shown the practical limitation of most of the higher-order methods. For the lower-order methods (Gauss-1, Gauss-2, moment-1, moment-2 and Cheby-3) the results of the computer simulations are in agreement with the theoretical expectation. This does not hold true for the other methods, which is probably caused by the inaccurate type of interpolation used in the determination of the pulse positions. Of course, this demonstrates the limitation of the higher-order methods. An interesting experiment to be carried out is to apply the integration concept to a periodic transmittance function belonging to a discrete object.

Unfortunately the one-dimensional integration concept is not directly applicable to continuous two-dimensional signals. Reformulating the integration concept in a differential form makes an extension to two dimensions possible. The two-dimensional pulse density signal is then found by solving the resulting set of nonlinear differential equations numerically. Although the use of the alternating direction implicit method reduces the complexity considerably, our present implementation still requires too much computation for a large number of pulses. A further reduction in computation time could be achieved by gradually increasing the number of pulses. Besides, we mention that an investigation of other types of differential equations which are easier to solve, e.g. the eikonal equation, could lead to interesting possibilities for two-dimensional pulse-density modulation.

When the hologram transmittance function is regarded as a probability density function, the continuous pulse-density modulation problem bears a resemblance to the vector quantization and the vector clustering problem. In this context the Linde-Buzo-Gray (LBG) algorithm and the Kohonen neural network have been discussed shortly. The LBG algorithm is designed to minimize a distortion function. Although currently used distortion functions give rise to an adaptation of the pulse density to the probability density function it is not obvious what kind of effect this has in the Fourier domain. On the other hand, formulating an appropriate distortion function in order to obtain a desirable effect in the Fourier domain is not trivial. The Kohonen network is able to generate a set of equiprobable (in pulse-density modulation terms: equal-area) pulses. However, this network is known to converge slowly. Further research has to be carried out to establish whether these approaches are applicable for pulse-density modulation.

When the positions of the pulses are restricted to fixed raster points we speak of discrete pulse-density modulation. The problem is now reduced to deciding for each raster point whether a pulse is placed or not. Under this condition the integration concept leads to first-order error diffusion. Due to its discrete character one-dimensional error diffusion can be easily extended to two-dimensional signals. To this end an appropriate processing order of the points in the two-dimensional raster is introduced. A threshold device or quantizer makes the decision for the consecutive raster points, introducing an error in each decision. The basic idea of error diffusion is to take the errors introduced at previous raster points into account in the decision of the raster point under consideration. To this end a weighted sum of the previous errors is formed by means of an error feedback filter. A linear model for error diffusion shows how the diffusion coefficients contained in this sum can be applied in order to obtain a desired shaping of the spectrum of the noise introduced by the quantizer. In this thesis we have concentrated on so-called internal error diffusion, for which the design of the feedback filter turns out to be a linear problem. Due to the recursive character, however, the error-diffusion system can become unstable. In order to avoid such a situation the choice of the diffusion coefficients is restricted by a stability condition. With the proposed design method stable sets of (internal) diffusion coefficients with desirable noise shaping properties are obtained. The formulation of necessary and sufficient stability conditions for the diffusion coefficients is still an unsolved problem.

The proposed design method for the error-feedback filter is merely based on the minimization of the contribution of the error within the object window. Consequently, the contribution of the error outside the object window can become quite large, resulting in small efficiency of the hologram. A larger efficiency can be obtained at the expense of a decrease in the signal-to-noise ratio by prescribing the noise shaping characteristic outside the object window as well.

With error diffusion a decision is made for one raster point at a time. Similar to the higher-order integration methods for continuous pulse-density modulation small groups of raster points could be considered simultaneously, leading to vector quantization with error feedback. It can be safely expected that with such an approach an improvement in the signal-to-noise ratio can be achieved.

In order to make a recursive order of processing of the raster points possible the error-feedback filter is required to have wedge support. If a four-quadrant support is desired, the resulting nonlinear difference equation has to be solved iteratively. This leads to the sequential updating rule of a discrete-time Hopfield neural network, where the weights and the thresholds of the network are determined by the original hologram and the object window. With a separable configuration of the object window and its twin window in the reconstruction plane, a reduction in complexity is achieved. To this end the neurons are selected in a special order. In spite of the extra amount of computation, compared to error diffusion, the performance of Hopfield's network is rather disappointing. This can be explained by the fact that during updating the network performs a local search process which halts at a local optimal solution. With a Boltzmann machine, which combines the sequential updating law with simulated annealing, escape from local optimal solutions is made possible. Due to its complexity, the sequential Boltzmann machine has not been implemented.

In principle, parallel updating allows all neurons to change their state simultaneously. Actually, the number of simultaneous transitions is large in the initial stage and decreases during the iteration process. Due to the special structure of the interconnections weights, the fast Fourier transformation can be applied to parallel updating in order to lower the complexity per neuron update. Although the required number of iterations is larger for parallel updating than for sequential updating an overall reduction in computational effort is obtained. The presented parallel Boltzmann machine combines the faster parallel updating algorithm with simulated annealing. In applications where only the intensity of the object is of interest, the phase-freedom leads to a Boltzmann machine with thresholds that are adapted during updating. Especially for scaled intensity objects holograms with a significantly improved signal-to-noise ratio are obtained.

Both sequential and parallel updating are closely related to other iterative methods used for the calculation of binary holograms (direct binary search, the iterative Fourier-transform algorithm, projections on convex sets). A comparative study concerning the merits of the different iterative approaches seems therefore desirable.

Appendix A

In this appendix we derive (2.14). In order to avoid cumbersome notation we consider the one-dimensional case. In a straightforward way the derivation can be carried out in two dimensions. For the continuous signal $\psi(x)$ we have the Fourier transform pair

$$\begin{aligned} \Psi(u) &= \int \psi(x) e^{-i2\pi ux} \mathrm{d}x & \text{(A.1)} \\ \psi(x) &= \int \Psi(u) e^{i2\pi xu} \mathrm{d}u \ , & \text{(A.2)} \end{aligned}$$

while the Fourier transform pair for the discrete signal $\psi[n]$ reads

$$\begin{aligned} \Psi_d(\theta) &= \sum_n \psi[n] e^{-i2\pi\theta n} & \text{(A.3)} \\ \psi[n] &= \int_{\langle 1 \rangle} \Psi_d(\theta) e^{i2\pi\theta n} \mathrm{d}\theta \ . & \text{(A.4)} \end{aligned}$$

Next, we derive the relation between the spectra $\Psi(u)$ and $\Psi_d(\theta)$ in case the discrete signal $\psi[n]$ is the result of a sampling of the continuous signal $\psi(x)$ at sampling points $x = (n + \frac{1}{2})X$. Evaluation of (A.2) at these sampling points gives

$$\psi(nX + \tfrac{1}{2}X) = \int \Psi(u) e^{i\pi Xu} e^{i2\pi nXu} \mathrm{d}u \ . \tag{A.5}$$

Under our assumption, (A.5) must equal (A.4). To this end we introduce $\theta = Xu$ and divide the θ-domain in intervals of unit length. In this way we derive

$$\frac{1}{X} \sum_m \int_{m-\frac{1}{2}}^{m+\frac{1}{2}} \Psi(\frac{\theta}{X}) e^{i\pi\theta} e^{i2\pi\theta n} \mathrm{d}\theta = \int_{\langle 1 \rangle} \frac{1}{X} e^{i\pi\theta} \sum_m (-1)^m \Psi(\frac{\theta + m}{X}) e^{i2\pi\theta n} \mathrm{d}\theta \ , \tag{A.6}$$

where we have used $e^{i\pi m} = (-1)^m$ and $e^{i2\pi nm} = 1$. Consequently, we find the relation

$$\Psi_d(\theta) = \frac{1}{X} e^{i\pi\theta} \sum_m (-1)^m \Psi(\frac{\theta + m}{X}) \tag{A.7}$$

for the spectra $\Psi_d(\theta)$ and $\Psi(u)$.

References

Aarts E.H.L., Korst J.H.M. (1989) Simulated Annealing and Boltzmann Machines. New York: Wiley.

Abramowitz M., Stegun I.A. (1970) Handbook of Mathematical Functions. New York: Dover Publications.

Anastassiou D. (1989) Error diffusion coding for A/D conversion. IEEE Trans. Circuits Syst. **36**, 1175-1186.

Barnard E. (1988) Optimal error diffusion for computer-generated holograms. J. Opt. Soc. Am. A **5**, 1803-1817.

Barnett S. (1990) Matrices: Methods and Applications. Oxford: Clarendon Press.

Bracewell R.N. (1978) The Fourier Transform and its Applications. New York: McGraw-Hill.

Broja M., Eschbach R., Bryngdahl O. (1986) Stability of active binarization processes. Opt. Commun. **60**, 353-358.

Brown B.R., Lohmann A.W. (1966) Complex spatial filtering with binary masks. Appl. Opt. **5**, 967-969.

Bruck J., Goodman J.W. (1988) A generalized convergence theorem for neural networks. IEEE Trans. Inf. Theory **34**, 1089-1092.

Bryngdahl O., Wyrowski F. (1990) Digital holography – computer-generated holograms. In: Wolf E. (Ed.): Progress in Optics, Vol. 28. Amsterdam: North-Holland.

Burch J.J. (1967) A computer algorithm for the synthesis of spatial frequency filters. Proc. IEEE **55**, 599-601.

Butterweck H.J., Ritzerfeld J.H.F., Werter M.J. (1988) Finite wordlength effects in digital filters: a review. EUT Report 88-E-205, Eindhoven University of Technology.

Candy J.C., Temes G.C. (1992) Oversampling Delta-Sigma Data Converters: Theory, Design and Simulation. New York: IEEE Press.

Černy V. (1985) Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. J. Opt. Theory Appl. **45**, 41-45.

Christov C.I. (1982) Orthogonal coordinate meshes with a manageable Jacobian. In: Thompson J.F. (Ed.): Numerical Grid Generation. Amsterdam: North-Holland.

Dallas W.J. (1980) Computer-generated holograms. In: Frieden B.R. (Ed.): Topics in Applied Physics, Vol. 41. Berlin: Springler.

Davis P.J., Rabinowitz P. (1980) Methods of Numerical Integration. London: Acadamic Press.

Ekstrom M.P., Woods J.W. (1976) Two-dimensional spectral factorization with applications in recursive digital filtering. IEEE Trans. Acoust. Speech Signal Process. **24**, 115-128.

Ekstrom M.P., Twogood R.E., Woods J.W. (1980) Two-dimensional recursive filter design – A spectral factorization approach. IEEE Trans. Acoust. Speech Signal Process. **28**, 16-25.

Eschbach R., Hauck R. (1987) Binarization using a two-dimensional pulse-density modulation. J. Opt. Soc. Am. A **4**, 1873-1878.

Eschbach R. (1990) Pulse-density modulation on rastered media: combing pulse-density modulation and error diffusion. J. Opt. Soc. Am. A **7**, 708-716.

Floyd R.W., Steinberg L. (1976) An adaptive algorithm for spatial greyscale. Proc. Soc. Inf. Disp. **17**, 75-77.

Gabor D. (1948) A new microscopic principle. Nature London **161**, 777-778.

Goles E., Fogelman F., Pellegrin D. (1985) Decreasing energy functions as a tool for studying threshold networks. Discrete Appl. Math. **12**, 261-277.

Gompel T. van (1993) Synthesis of binary holograms by means of a neural network. Graduation thesis ESP-2-93 (in Dutch), Eindhoven University of Technology.

Goodman J.W. (1968) Introduction to Fourier Optics. New York: McGraw-Hill.

Guyon I., Personnaz L., Nadal J.P., Dreyfus G. (1988) High-order neural networks for efficient associative memory design. Neural Information Processing Systems, 233-241.

Hauck R., Bryngdahl O. (1984) Computer-generated holograms with pulse-density modulation. J. Opt. Soc. Am. A **1**, 5-10.

Haykin S. (1987) Adaptive Filter Theory. Englewood Cliffs: Prentice-Hall.

Hecht-Nielsen R. (1989) Neurocomputing. Amsterdam: Addison-Wesley.

Hinton G.F., Sejnowski T.J., Ackley D.H. (1984) Boltzmann machines: Constraint satisfaction networks that learn. Carnegie Mellon University Technical Report CMU-CS-84-119, Carnegie Mellon University, Pittsburg PA.

Hopfield J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **97**, 2554-2558.

Hopfield J.J. (1984) Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. USA **81**, 3088-3092.

Hopfield J.J., Tank D.W. (1985) 'Neural' computation of decisions in optimization problems. Biol. Cybern. **52**, 141-152.

Jagerman D. (1966) Investigation of a modified mid-point quadrature formula. Math. Comput. **20**, 79-89.

Jenkins B.K., Chavel P., Forchheimer R., Sawchuk A.A., Strand T.C. (1984) Architectural implications of a digital optical processor. Appl. Opt. **23**, 3465-3474.

Jennison B.K., Allebach J.P., Sweeney D.W. (1991) Efficient design of direct-binary-search computer-generated holograms. J. Opt. Soc. Am. A **8**, 652-660.

Just D., Ling D.T. (1991) Neural networks for binarizing computer-generated holograms. Opt. Commun. **81**, 1-5.

Kant G.W. (1993) Design of two-dimensional error-diffusion feedback filters. A spectral factorization approach. Graduation thesis ESP-12-93 (in Dutch), Eindhoven University of Technology.

Keller P.E., Gmitro A.F. (1993) Computer-generated holograms for optical neural networks: on-axis versus off-axis geometry. Appl. Opt. **32**, 1304-1310.

Kim J.G., Kim G. (1986) Design of optimal filters for error-feedback quantization of monochrome pictures. Inf. Sci. **39**, 285-298.

Kirkpatrick S., Gelatt C.D., Vecchi M.P. (1982) Optimization by simulated annealing. IBM Research Report RC 9355.

Kirkpatrick S., Gelatt C.D., Vecchi M.P. (1983) Optimization by simulated annealing. Science **220**, 671-680.

Klijn H. (1991) Analysis of the Hopfield network with respect to sample-controlled sequential pattern recognition. Graduation thesis 080-DV-9113, University of Twente.

Kohonen T. (1984) Self-Organization and Associative Memory. Berlin: Springer.

Koppelaar A.G.C. (1992) Two-dimensional continuous pulse-density modulation for optical applications. Graduation thesis ET-2-92 (in Dutch), Eindhoven University of Technology.

Kreyszig E. (1983) Advanced Engineering Mathematics. New York: Wiley.

Laakso T.I., Hartimo I. (1992) Noise reduction in recursive digital filters using high-order error feedback. IEEE Trans. Signal Process. **40**, 1096-1107.

Lee W.H. (1978) Computer-generated holograms: techniques and applications. In: Wolf E. (Ed.): Progress in Optics, Vol. 16. Amsterdam: North-Holland.

Lesem L.B., Hirsch P.M., Jordan J.A. (1968) Computer synthesis of holograms for 3-D display, Commun. ACM **11**, 661-674.

Lim J.S. (1990) Two-dimensional Signal and Image Processing. Englewood Cliffs: Prentice-Hall.

Linde Y., Buzo A., Gray R.M. (1982) An algorithm for vector quantizer design. IEEE Trans. Commun. **28**, 84-95.

Max J. (1960) Quantizing for minimum distortion. IEEE Trans. Inf. Theory **6**, 7-12.

Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E. (1953) Equation of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087-1092.

Oppenheim A.V., Schafer R.W. (1975) Digital Signal Processing. Englewood Cliffs: Prentice-Hall.

Papoulis A. (1965) Probability, Random Variables and Stochastic Processes. London: McGraw-Hill.

Seldowitz M.A., Allebach J.P., Sweeney D.W. (1987) Synthesis of digital holograms by direct binary search. Appl. Opt. **26**, 2788-2798.

Stark H. (1992) Mathematical projection methods for solving problems in imaging. Short Course Notes 113, Annual Meeting '92 OSA.

Voronoi G. (1907) Nouvelles applications des paramètres continus à la théorie des formes quadratiques. J. Reine Angew. Math. **133**, 97-178.

Weissbach S., Wyrowski F., Bryngdahl O. (1989) Digital phase holograms: coding and quantization with an error diffusion concept. Opt. Commun. **72**, 37-41.

Weissbach S. (1992) Das Fehlersdiffusionsverfahren: Theorie und Anwendung in der optischen Informationsverarbeitung. Ph.D. thesis, University Essen.

Wyrowski F., Bryngdahl O. (1988) Iterative Fourier-transform algorithm applied to computer holography. J. Opt. Soc. Am. A **5**, 1058-1065.

Wyrowski F. (1989) Iterative quantization of digital amplitude holograms. Appl. Opt. **28**, 3864-3870.

Wyrowski F. (1990) Diffraction efficiency of analog and quantized digital amplitude holograms: analysis and manipulation. J. Opt. Soc. Am. A **7**, 383-393.

Samenvatting

Voor de realisatie van computer-gegenereerde hologrammen wordt vaak gebruik gemaakt van apparatuur waarmee alleen binaire uitvoer gegenereerd kan worden. Een omzetting van de berekende transparantie-functie naar een binair hologram is dan noodzakelijk. Indien hiervoor uitgegaan wordt van pulsdichtheidsmodulatie bestaat de binaire transparantie-functie uit identieke binaire pulsen. De gewenste optische eigenschappen worden bij benadering verkregen door een lokale variatie in de pulsdichtheid. Voor Fourier-hologrammen eisen we dat de Fourier-spectra van het originele en het binaire hologram in een bepaalde frequentieband zo goed mogelijk overeenkomen.

Bij continue pulsdichtheidsmodulatie wordt geen beperking opgelegd aan de posities van de pulsen. Voor ééndimensionale signalen wordt uitgegaan van een integratie-concept, waarbij de pulsposities achtereenvolgens voor individuele pulsen of voor groepjes pulsen worden bepaald. Voor deze methodes is een schatting gegeven voor de benaderingsfout als functie van de pulsdichtheid. De resultaten van computer-simulaties zijn voor de lagere-orde methodes in overeenstemming met de theorie. Door het integratie-concept in differentiaalvorm te formuleren is een uitbreiding naar twee dimensies mogelijk. Het gevonden stelsel niet-lineaire differentiaalvergelijkingen is voor een relatief klein aantal pulsen numeriek opgelost. Voor een groot aantal pulsen vereist de methode in de huidige vorm echter te veel rekenwerk. Ter afsluiting is de relatie tussen continue pulsdichtheidsmodulatie en een aantal clustering-technieken besproken.

Bij discrete pulsdichtheidsmodulatie is de plaatsing van de pulsen beperkt tot vaste posities. Onder deze voorwaarde gaat het integratie-concept over in 'error diffusion', waarbij de toegelaten posities één voor één afgewerkt worden. Error diffusion kan eenvoudig op een tweedimensionaal raster toegepast worden door een tijdvolgorde in dit raster aan te geven. Bij de beslissing om op een gegeven rasterpunt al dan niet een puls te zetten, worden de fouten van voorgaande rasterpunten via een gewogen som in aanmerking genomen. Een lineair model beschrijft de invloed van de in deze som voorkomende diffusie-coëfficiënten op de afwijking tussen de spectra. Om instabiliteit van dit recursieve systeem te voorkomen, zijn er voorwaarden gesteld aan de keuze van de diffusie-coëfficiënten. Voor de berekening van geschikte, stabiele diffusie-coëffiënten behorende bij een gegeven frequentieband is een methode ontwikkeld. Met de hiermee gevonden coëfficiënten zijn hologrammen

berekend die een goede benadering van het origineel in de gegeven frequentieband opleveren.

De uitbreiding op error diffusion, waarvoor de pulsverdeling niet meer recursief te bepalen is, komt overeen met een Hopfield-netwerk. Het te benaderen spectrum en de frequentieband leggen de drempelwaarden en de gewichten van de verbindingen in dit neuraal netwerk vast. Startend vanuit een willekeurige pulsverdeling convergeren de uitgangswaarden van de neuronen naar een optimale pulsverdeling. Ondanks de extra hoeveelheid rekenwerk blijkt het Hopfield-netwerk geen verbetering ten opzichte van error diffusion op te leveren. Aangezien het netwerk een lokaal zoekproces uitvoert, eindigt het convergentietraject in een lokaal optimale pulsverdeling. Met een Boltzmann machine, die ontstaat door een Hopfield netwerk te combineren met 'simulated annealing', wordt ontsnapping uit een lokaal optimum mogelijk gemaakt en kunnen betere oplossingen gevonden worden. Door de speciale struktuur van de verbindingen tussen de neuronen, kan het Hopfield netwerk op een efficiënte manier geïmplementeerd worden. Deze eigenschap wordt ook bij de voorgestelde Boltzmann machine met parallel updating benut. Op deze manier is het mogelijk gebleken om binnen redelijke tijd een aanmerkelijk betere pulsverdeling te berekenen.

Stellingen

1. De diffusie-coëfficiënten in het artikel 'Optimal error diffusion for computer-generated holograms' worden bepaald onder een stabiliteitsvoorwaarde die voldoende maar niet nodig is. Dit leidt weliswaar tot een stabiele maar geenszins tot een optimale oplossing.

   E. Barnard, Optimal error diffusion for computer-generated holograms, J. Opt. Soc. Am. A 5, 1803-1817, 1988.

2. De conclusie in het artikel 'Neural networks for binarizing computer-generated holograms' dat een Hopfield netwerk in het algemeen betere hologrammen oplevert dan error diffusion is door de eenvoudige keuze van de diffusie-coëfficiënten voorbarig.

   D. Just, D.T. Ling, Neural networks for binarizing computer-generated holograms, Opt. Commun. 51, 1-8, 1991.

3. De recent ontwikkelde algoritmes voor computer-genereerde hologrammen leveren betere resultaten op dan de oorspronkelijke door A.W. Lohmann geïntroduceerde methode. Om onduidelijke redenen wordt toch teruggegrepen naar de laatstgenoemde methode of varianten hierop.

4. Het jarenlange onafhankelijke bestaan van de benamingen 'error diffusion' en '$\Sigma\Delta$-modulation' voor dezelfde techniek toont aan dat er een informatieprobleem in de wetenschappelijke communicatie bestaat.

5. Het belang van de rol die de optica kan spelen voor het aanreiken van nieuwe technologieën voor de ontwikkeling van hardware wordt door het bedrijfsleven onderschat.

6. Het aantal proefschriften op het gebied van computer-gegenereerde hologrammen weerspiegelt niet de praktische betekenis van dit vak. Het aantal toepassingen is gering en hierin zal waarschijnlijk geen verandering komen.

   G. Saxby, Practical Holography, Prentice Hall, 1988.

7. Het verschil tussen een Nederlander en een Vlaming is dat de Vlaming zich minder afvraagt wat dit verschil is.

Antwerpen, 18 april 1994
P. van den Bulck