ディジタルホログラフィにおける再生像シミュレータ

赤　堀　　寛*

Reconstruction Image Simulator in Digital Holography

Hiroshi AKAHORI

Abstract

A system simulating images reconstructed from the Fourier-type computer generated hologram is described. Simulated images are depicted on a display with 256 levels of monochrome tone. The effects of quantization and speckle noise on the images are incorporated into the simulator. A computer program of the simulator is outlined and several examples of simulated images are shown.

1. はじめに

通常の光ホログラフィにおいて，記録すべき対象物体の情報を含んだ物体波は，コヒーレント光を用いて物理的に作成される。これに対してディジタルホログラフィでは，コンピュータを使ってディジタル的に物体波データを算出する。それ故にこうして得られるホログラムは計算機ホログラムと呼ばれる。計算機ホログラムは，実在しない物体の像の表示を可能にするので，この特徴を活かした応用が数多く考えられている。

光ホログラフィと同様に，ディジタルホログラフィも作成過程と再生過程という2つの過程からなる。ホログラムから像を再生する過程は基本的には同じであるが，作成過程は両者で大きく異なる。ディジタルホログラフィの場合，ホログラムの作成過程はさらに，物体波データを算出するステップと物体波データを媒体に記録するステップとに分けられる。前者はコンピュータによる処理であり，この処理の中味によって計算機ホログラムの再生像の形式，混入する雑音の種類，エネルギー効率などが決められる。前者がソフトウェア的なステップであるのに対して，後者の物体波データ記録ステップはハードウェア的なステップといえよう。このステップの問題点のひとつは，物体波データの記録に，通常，相当に長い処理時間を必要とすることである。

ディジタルホログラフィの実用化を推進するためには，雑音が少なく，エネルギー効率の良い像を再生可能なホログラムの作成法を確立することが重要である。作成手法の探索にあたっては，物体波データ算出ステップでさまざまな工夫を取り入れ，その効果を再生像の状態から評価するという実験を繰り返すことが必要になろう。ところが，前述のハードウェア的ステップを含んだ作成過程を経たのでは，ひとつの再生像を得るのに長い時間を要し，こうした実験を能率的に進める上で大きな障害となる。これを避けるには，計算機ホログラムを実際に作成することなしに，作成手法の探索実験を行うことができればよい。計算機ホログラムの再生像の状態をソフトウェア的に模擬することがひとつの解決策である。このような理由により再生像シミュレータの作成を試みた。

本論文で述べる再生像シミュレータは，最も一般的なフーリエ変換形の計算機ホログラムからの再生像を模擬するものである。しかも，再生像の中の真の像（もとの物体像）のみを表示し，再生像に現われる雑音に関しては，ディジタルホログラフィに共通的な量子化

平成2年9月28日受理

* 情報工学科

雑音とスペックル雑音が考慮される。

2. ディジタルホログラフィのタイプと再生像

計算機ホログラムに記録しようとする物体データを$N\times N$の点開口の2次元配列であると考え，それを次のように表わす。

$$\{f(k,\ l);\ k=0,\ 1,\ \cdots,\ N-1,\ l=0,\ 1,\ \cdots,\ N-1\}$$

$f(k,\ l)$はk行l列における点開口から放射される光の複素振幅である。これは振幅成分$a(k,\ l)$と位相成分$\phi(k,\ l)$を用いて次のように表わされる。

$$f(k,\ l)=a(k,\ l)\exp[i\phi(k,\ l)] \qquad (1)$$

ところで，人間の眼に感じられたり，光検出器で検知できるのは光の強度$ff^*=a^2$である。したがって，物体データとして与えられるのは実質的には振幅成分$a(k,\ l)$であり，位相成分$\phi(k,\ l)$については任意の選択が許される。ただし，$\phi(k,\ l)$の選び方は，後述するように，再生像雑音に影響する。

フーリエ変換形のディジタルホログラフィでは，ホログラム面上の物体波情報は物体の複素振幅分布の離散的フーリエ変換(DFT)で与えられる。複素振幅$f(k,\ l)$のDFTを$F(m,\ n)$とすれば，これは次式で表わされる。

$$F(m,\ n)=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}f(k,\ l)\exp[-2\pi i(km+ln)/N] \qquad (2)$$

$$m=0,\ 1,\ \ldots,\ N-1$$
$$n=0,\ 1,\ \ldots,\ N-1$$

ここで$F(m,\ n)$はフーリエ面上の複素振幅であるから，次のように表わすことができる。

$$F(m,\ n)=A(m,\ n)\exp[i\theta(m,\ n)] \qquad (3)$$

ただし，$A(m,\ n)$は振幅成分，$\theta(m,\ n)$は位相成分である。

フーリエ係数の2次元配列$\{F(m,\ n);\ m=0,\ 1,\ \ldots,\ N-1,\ n=0,\ 1,\ \ldots,\ N-1\}$が，ホログラムに記録すべきデータとなる。これをホログラムデータと呼ぶことにする。計算機ホログラムはホログラムデータを正方形セルの2次元アレイに記録したものである。フーリエ係数のデータをセルの中にどのような形式で記録させるかが問題になるが，記録形式に基づいて様々なタイプの計算機ホログラムが考えられている。最も代表的な方法はう回位相（detour phase）法と呼ばれるもので，ローマンタイプ[1,2]，リータイプ[3]，2位相タイプ[4]，などが知られている。この方法は，セルの中に透明な窓を開け，この窓の大きさと位置でフーリエ係数を表示することに特徴がある。この方法によるホログラムは，作成が比較的容易であるが，ホログラムを再生したとき，もとの物体の像だけではなく，強力なエネルギーをもった光軸上の輝点，共役像，高次の回折像などが現われるため，光のエネルギー効率が極めて悪い。これに対して，光のエネルギー効率が非常に高い，キノフォーム[5]と呼ばれる方法がある。これは，フーリエ係数の振幅成分を一定と見なし（振幅情報を棄却することになる），ホログラムデータを位相成分のみで表わすものである。ホログラム面上のセルの透過率を100%とし，位相成分をその光学的厚さで表わすことができるわけであるから，キノフォームの回折効率は原理的には100%である。しかも再生段階ではもとの物体の像のみが現われる。ただし，振幅成分の棄却に伴なう誤差や位相成分の正確な記録の難しさ，などの問題点を解決しなければならない。これらの他にも数多くの特徴的な方法が知られている[6~12]。

以上述べたように，ホログラムデータを物理的に記録する方法の違いによって，計算機ホログラムにはいろいろなタイプがある。再生過程で現われる像の様子や混入する雑音の種類もこのタイプに依存するところが大きい。そこで今回の再生像シミュレータでは，すべてのホログラフィタイプの再生過程で共通的に現われる部分について模擬することにした。

3. 再生像雑音

計算機ホログラムの再生像に現われる雑音は，①量子化誤差による雑音，② スペックル雑音，③ 表示誤差による雑音などが主要なものと考えられる。これらの中で，①と②はホログラムデータの中に混入され，しかも計算機ホログラムの各タイプに共通して現われるものである。これに対して③は，ホログラムデータを実際に物理的に記録するときに発生するものである。このため③の雑音の中味はホログラムデータの記録方法に依存し，計算機ホログラムの各タイプに固有なものになると考えられる。したがって，ここで述べる再生像シミュレータでは①および②の共通的な雑音が取り扱われる。

3.1　量子化誤差

計算機ホログラムはフーリエ係数 $F(m, n)$ を物理的に記録したものである。記録の際に，フーリエ係数の振幅や位相を表現できる値の数は，記録装置の仕様によって限られる。したがって振幅および位相の値を，記録装置によって決められる一定数のレベルにあらかじめ量子化しておかなければならない。

(1)　振幅の量子化

量子化レベルの数が M，振幅成分 $A(m, n)$ の最大値が A_{max} であるとき，量子化誤差の最大値 P は

$$P=A_{max}/(2M) \tag{4}$$

となり，振幅成分の最大値に比例する。振幅の量子化誤差を減少させるには A_{max} を小さくすること，すなわちフーリエスペクトルのダイナミックレンジを減少させることが必要である。

式 (1) において $\phi(k, l)$=一定とすると，式 (2) より

$$|F(0, 0)|=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}a(k, l) \tag{5}$$

となる。これは，原点 (0, 0) における振幅成分が $N\times N$ 個の実数値の和で表わされることを示す。このことはさらに，極めて大きな振幅成分が存在し，その結果振幅の量子化誤差が大きくなることを意味する。このような現象を避けるには，$\phi(k, l)$ に適当な値の位相を付加すればよい。区間 $[0, 2\pi]$ 上に一様分布するランダム位相を付加する方法が従来から行われているが，こうすると後述するようにスペックル雑音の発生という別の問題が生じる。

(2)　位相の量子化

位相成分 $\theta(m, n)$ は 0 と 2π の間の角度をとる。量子化レベルの数を L とすれば，量子化誤差の最大値 Q は

$$Q=\pi/L \tag{6}$$

となり，量子化レベル数 L のみに依存する。

3.2　スペックル

計算機ホログラムに記録される物体データは，記録しようとする実物体を $N\times N$ の標本点でサンプリングすることによって得られたものであると見なしてもよい。ただし実物体の振幅は実数値である。この物体データを記録したホログラムを再生するとき，標本点においてだけでなく，標本点と標本点の間の空間も含めた再生面上でもとの実物体が正確に再生されるのは，式 (1) の位相成分 $\phi(k, l)$ が一定のときである。言いかえれば，実物体の位相成分が一様であるときにのみ，標本化定理が満たされるものと考えられる。

これに対して，ランダム位相を付加することによって得られた物体データは，実物体をナイキスト周波数よりも低い周波数でサンプリングしたものであり，その結果このデータにはエイリアシングエラーが含まれると考えるべきである。こうした物体データを記録した計算機ホログラムを再生すると，標本点では正確な像が得られるが，標本点と標本点の間ではエイリアシングエラーに起因すると見られる斑点状のランダムパターン，いわゆるスペックル雑音が現われる。

標本点における再生像の複素振幅は，フーリエデータ $\{F(m, n)\}$ を離散的フーリエ逆変換することによって与えられる。

$$f(k, l)=\frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}F(m, n)\exp[2\pi i(km+ln)/N] \tag{7}$$

$$k=0, 1, \ldots, N-1$$
$$l=0, 1, \ldots, N-1$$

スペックルの状態を知るためには，標本点と標本点の間の点の複素振幅を求めることが必要である。そこで，再生面上の任意の点 (k', l') における複素振幅を求めよう。k' および l' を

$$\begin{aligned}k'&=k+\delta \quad |\delta|<1\\ l'&=l+\varepsilon \quad |\varepsilon|<1\end{aligned} \tag{8}$$

とおくと

$$\begin{aligned}f(k', l')&=\frac{1}{N_2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}F(m, n)\\ &\quad\times\exp\left[2\pi i\frac{(k+\delta)m+(l+\varepsilon)n}{N}\right]\\ &=\frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}F'(m, n)\exp[2\pi i(km+ln)/N]\end{aligned} \tag{9}$$

ただし，

$$F'(m, n)=F(m, n)\exp[2\pi i(\delta m+\varepsilon n)/N] \tag{10}$$

である。

この結果，式 (10) のように，フーリエ係数に位相項を付加したものに対してフーリエ逆変換を行えばよいことがわかる。なお，画像全体の傾向を知るためには (9) 式の計算を繰り返して，それらの結果を重ね合わ

せる必要がある。

スペックルの状態を求めるもうひとつの方法は，$N\times N$ のホログラムデータを用いて，$KN\times KN$ の表示点の振幅を一度に計算するというものである。ホログラムデータを次のように与えれば，隣接する標本点間に等間隔で配列された $(K-1)$ 個の点における像の振幅も計算することができる。

$$F'(m,\ n)=$$
$$\begin{cases} F(m,\ n);\ m=0,\ 1,\ \ldots,\ \dfrac{N}{2}-1,\ n=0,\ 1,\ \ldots,\ \dfrac{N}{2}-1 \\ F(m-KN+N,\ n);\ m=KN-\dfrac{N}{2},\ KN-\dfrac{N}{2}+1,\ \ldots,\ KN-1 \\ \qquad\qquad n=0,\ 1,\ \ldots,\ \dfrac{N}{2}-1 \\ F(m,\ n-KN+N);\ m=0,\ 1,\ \ldots,\ \dfrac{N}{2}-1 \\ \qquad\qquad n=KN-\dfrac{N}{2},\ KN-\dfrac{N}{2}+1,\ \ldots,\ KN-1 \\ F(m-KN+N,\ n-KN+N) \\ \qquad ;\ m=KN-\dfrac{N}{2},\ KN-\dfrac{N}{2}+1,\ \ldots,\ KN-1 \\ \qquad\quad n=KN-\dfrac{N}{2},\ KN-\dfrac{N}{2}+1,\ \ldots,\ KN-1 \\ 0\ ;\ \text{上記以外のとき} \end{cases} \tag{11}$$

たとえば $K=2$ の場合の再生像の複素振幅は，式(11)と式(7)から，次式のようになる。

$$\begin{aligned} r(k,\ l)=\frac{1}{4N^2}\Bigg\{ & \sum_{m=0}^{N/2-1}\sum_{n=0}^{N/2-1} F(m,\ n) \\ & \times\exp\left[\pi i(km+ln)/N\right] \\ & +(-1)^k\sum_{s=N/2}^{N-1}\sum_{n=0}^{N/2-1} F(s,\ n) \\ & \times\exp\left[\pi i(ks+ln)/N\right] \\ & +(-1)^l\sum_{m=0}^{N/2-1}\sum_{t=N/2}^{N-1} F(m,\ t) \\ & \times\exp\left[\pi i(km+lt)/N\right] \\ & +(-1)^{k+l}\sum_{s=N/2}^{N-1}\sum_{t=N/2}^{N-1} F(s,\ t) \\ & \times\exp\left[\pi i(ks+lt)/N\right]\Bigg\} \end{aligned} \tag{12}$$

ただし，$s=m-N$，$t=n-N$ である。

もとの標本点では

$$k=2p\quad p=0,\ 1,\ \ldots,\ N-1$$
$$l=2q\quad q=0,\ 1,\ \ldots,\ N-1$$

が成立するから，これを式(12)に代入すれば

$$\begin{aligned} r(2p,\ 2q)=&\frac{1}{4N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,\ n) \\ &\times\exp\left[\pi i(2pm+2qn)/N\right] \\ =&\frac{1}{4}f(p,\ q) \end{aligned} \tag{13}$$

となる。これはもとの標本点における物体データに比例することがわかる。

この方法により，$KN\times KN$ のデータのフーリエ逆変換を1回実行することによって，スペックル雑音の様子を算出することができる。ここで述べる再生像シミュレータではこの方法が用いられる。

4. 再生像シミュレータの概要

再生像シミュレータは，フーリエ変換形の計算機ホログラムから再生される，もとの物体の像(真の像)の強度分布を模擬し，これをディスプレイに表示するためのシステムである。このシミュレータでは，量子化誤差による雑音とスペックル雑音の影響が取り扱われる。使用した計算機はSPARC Station 1 4/60C-8，表示装置は19インチカラーディスプレイである。これの解像度は1152(H)×900(V)画素(81 dpi)である。このディスプレイを256階調のモノクロ表示装置として使用する。コンピュータプログラムのほとんどの部分はフォートランで記述されるが，ディスプレイにパターンを表示するサブルーチンは，グラフィックスソフトウェアSun CGIを利用するためC言語で書かれる。プログラムは，主プログラムimage. fと離散的フーリエ変換サブルーチンfft. f，画像表示用サブルーチンdisp. f, init. c, rect. c, circ. c, term. cからなる。これらのプログラム群のオブジェクトモジュールがSun CGIのライブラリとリンクされて実行モジュールが作成される。なお，画像表示用サブルーチンのプログラムリストを付録に示す。

4.1 主プログラムの説明

主プログラムimage. fのフローチャートを図1に示す。これを参照して主プログラムの概要を説明する。

(1) Reconstruction condition（再生条件の設定）

像の再生条件を規定するパラメータの値を設定する。

n：ホログラムデータの縦および横方向の大き

さ。

k：フーリエ面の拡大倍率。k の値が大きいほど，スペックルの状態が詳細に表示される。

name：ホログラムデータを格納したファイル名(文字変数)。

mode：表示モードの値。

np：振幅の量子化レベル数。

nq：位相の量子化レベル数。

iht：画像のコントラストを改善するために用いる指数値。

n および k はパラメータ文の中に記述し，他はプログラムの実行時にキーボードから読込まれる。

(2)　Hologram data（ホログラムデータ入力）

指定されたファイルをオープンしてホログラムデータを読込む。ホログラムデータはあらかじめこのファイルに格納しておく。

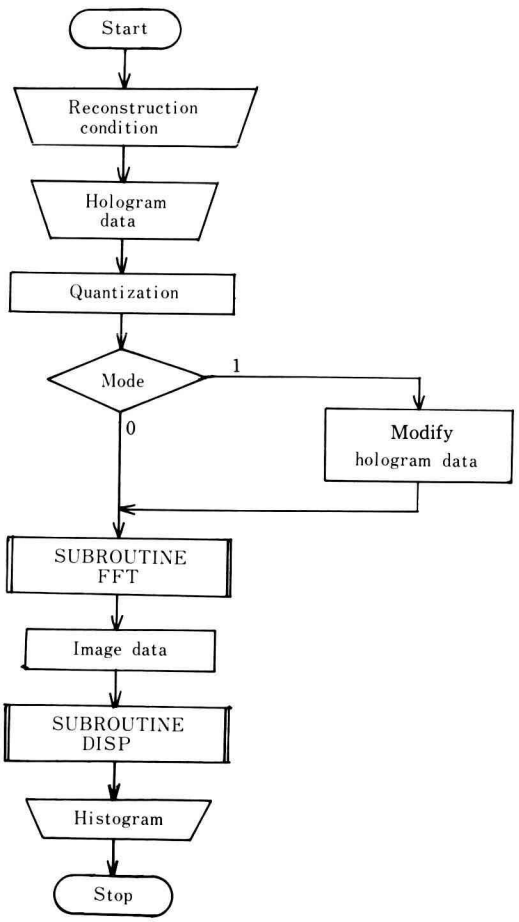


Fig. 1.　Flowchart of the main program, image.

(3)　Quantization（量子化）

ホログラムデータの振幅成分と位相成分を，それぞれ np, nq の量子化レベル数で量子化する。ただし，量子化を行わない場合を指定することもできる。

(4)　mode（表示モードの判定）

表示モードの値が 0 か 1 かを判定する。2 種類の表示モードの意味は次のようである。

mode＝0

もとの標本点における像強度のみを表示させ，スペックルを表示させないモードである。同一のホログラムデータを多数個，空間的に並べたものを記録させた計算機ホログラムからの再生像を模擬することになる。光信号を分配したり記憶したりするデバイスとして利用するような場合の像再生は，このモードに対応する。

mode＝1

スペックルの状態を表示させるためのモードである。単一のホログラムデータを記録させた計算機ホログラムからの再生像を模擬するもので，再生像は空間的に連続な画像として表示される。この場合には，ホログラムデータを式 (11) に基づいて修正する。すなわち，フーリエ面は縦および横方向のそれぞれが倍率 k で拡大される。

(5)　SUBROUTINE FFT（離散的フーリエ逆変換）

サブルーチン FFT を使ってホログラムデータをフーリエ逆変換し，再生像の複素振幅分布を求める。

(6)　Image data（表示画像の作成）

再生像の複素振幅分布から強度分布を求める。これがディスプレイへの表示画像となる。このときパラメータ iht が画像のコントラスト改善を指示する値に設定されていれば，iht によって決まる強度 I_t より小さい強度に対して表示画像への線形変換を行う。そして I_t より大きな強度については，これらをすべて最大レベルに量子化する。またここで強度のヒストグラムを算出する。

(7)　SUBROUTINE DISP（グラフィックス表示）

後述のサブルーチン disp (a, m, mode) を用いて，再生像の強度パターンをディスプレイにモノクロ(256階調）表示する。

(8)　Histogram（ヒストグラムデータ出力）

再生像強度のヒストグラムデータを出力させる。4.3 で説明するように，このデータは画像のコントラスト

を改善するために利用される。

4.2　グラフィックス表示サブルーチンの説明

再生像の強度パターンを標準のグラフィックスソフトウェア Sun CGI を用いて表示させるためのサブルーチン disp（a, m, mode）のフローチャートを図2に示す。これを参照してサブルーチンの概要を説明する。

（1）　サブルーチンの引数

サブルーチン disp は次の3つの引数をもつ。

a：再生像の強度データの2次元配列

m：配列 a の縦および横方向の大きさ

mode：表示モードの値

（2）　Geometric condition（幾何学的条件の設定）

画像の幾何学的位置がディスプレイ面の中心付近にある 800×800 画素の部分となるように，隣接する表示点の間の距離 nw および画像の左上端座標（nx, ny）を設定する。また，Sun CGI では画面の左下端が原点となるが，ここでは画面の左上端が原点となるように修正する。

（3）　SUBROUTINE INIT（CGI の初期設定）

サブルーチン init. c を用いて Sun CGI の初期設定を行う。すなわち，Sun CGI を初期化する，ビュー面を指定してこれをオープンする，VDC 空間の表示範囲を定義する，カラーテーブルを定義する，などの処理を行う。このサブルーチンではカラーテーブルの定義の際，同一のインデックスに対して RGB の3つの配列の値を同一にする。これにより 256 レベルの明るさをもった画素をモノクロ表示させることができる。

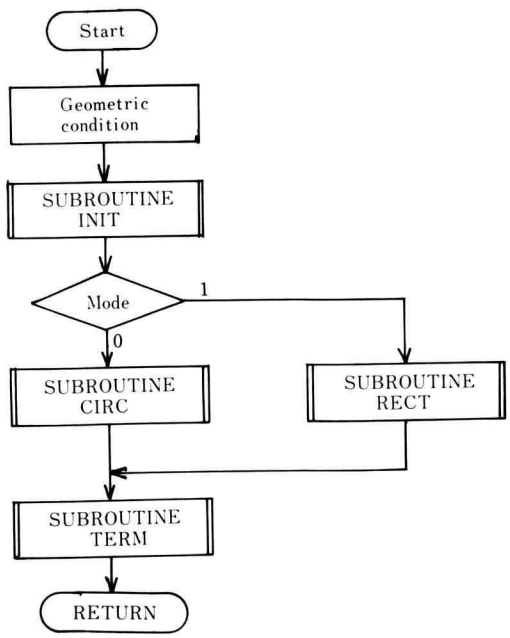


Fig. 2.　Flowchart of the subprogram, disp（a,m, mode）.

（4）　mode（表示モードの判定）

表示モードの値を判定し，mode 値に基づいた表示方法を選択する。

（5）　SUBROUTINE CIRC（円形表示）

mode＝0 のときには，サブルーチン circ(cenx, ceny, nrad, ic）を用いて，もとの標本点における点像のみをその強度に応じた明るさの円で表わす。ここで，cenx および ceny は円の中心の x および y 座標，nrad は円の半径である。円を点像に類似させるため nrad＝$\frac{1}{8}$nw と定める。ic はカラーテーブルを参照するときのカラーインデックスである。これは表示点の強度を256 レベルで量子化したときのレベル値である。実際に円を描くには CGI コマンド circle(& cen, rad）を用いる。

（6）　SUBROUTINE RECT（正方形表示）

mode＝1 のときには，サブルーチン rect(llx, lly, urx, ury, ic）を用いて，ひとつの表示点の像をその強度に応じた明るさの正方形で表わす。ここで，llx および lly は正方形の左下端（lower left）にある頂点の x および y 座標，urx および ury は右上端(upper right）の頂点の x および y 座標である。ic はカラーインデックスである。再生像を空間的に連続な画像として表示させるために，正方形の一辺の長さを nw に固定し，画像の表示面を正方形で完全に被覆する。実際に正方形を描くには CGI コマンド rectangle（&ll, &ur）を利用する。このコマンドは一般には短形を描くためのものである。

（7）　SUBROUTINE TERM（CGI の終了）

サブルーチン term. c を用いてビュー面をクローズし，Sun CGI を終了させる。終了の処理は，コンソールの任意のキーを押すことによって起動される。

4.3　表示画像のコントラストの改善

ディスプレイに表示される画像の明るさは通常，再生像の強度データを線形変換することによって得られる。ところが少数の表示点での強度が非常に大きい場

合，強度の全範囲にわたって線形変換をしたのでは，表示画像のコントラストが低下する。これを避けるためには，部分的線形変換を取り入れることが必要になる。言いかえれば，表示画像が図3の強度ヒストグラムをもつとき，頻度の大部分が集中する強度範囲$[I_0, I_t]$に対して線形変換を行い，I_t 以上の強度を I_t に等しいと見なせばよい。ディスプレイが256レベルの階調表示を可能にするので，$[I_0, I_t]$の強度範囲を255レベルに量子化し，I_t 以上の強度に最大の量子化レベルを割り当てるようにする。

再生像シミュレータを用いてコントラストを改善した画像を表示するには，最初のプログラム実行で得られた強度ヒストグラムデータから線形変換すべき範囲$[I_0, I_t]$を求め，この情報を利用してプログラムを再度実行させる。具体的には，主プログラムの再生条件の設定時に，強度 I_t に対応するインデックス iht の値を入力する。

計算機ホログラムの作成条件によっては，非常に大きな強度の表示点が光軸上に現われる。こうした場合，コントラストを改善しないと再生像の状態を適確に把握することは難しい。また通常の室内光の下で画像を見るようなときには，コントラストの改善は特に効果的である。

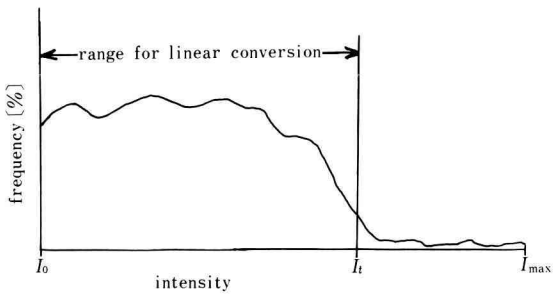


Fig. 3. An intensity histogram of the image with low contrast.

5. 表　示　例

図4に示す16×16の2値パターンに次の3種類の位相を付加したものを物体データとした。

(a) 一定位相

(b) 区間〔0, 2π〕上の一様ランダム位相

(c) 区間〔0, $\pi/2$〕上の一様ランダム位相

これら3種類の物体データを記録したホログラムから再生されるべき像を再生像シミュレータを用いて模擬した。

図5は mode＝0, np＝8 の場合である。すなわち量子化レベル数8で振幅のみの量子化を行い，もとの標本点での強度を表示したものである。図6は mode＝1, k＝8 の場合である。すなわち量子化は行わずにスペックル状態を表示したものである。ただし，図6(b)および(c)はコントラスト改善後の画像である。

図5および図6から，(a) 一定位相の場合にはスペックル雑音は生じないが振幅の量子化雑音が大き

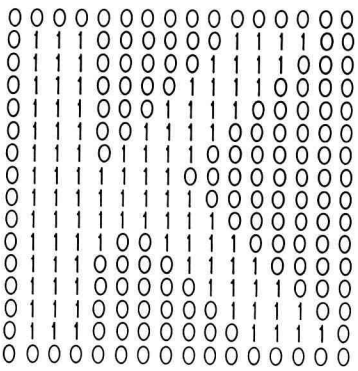
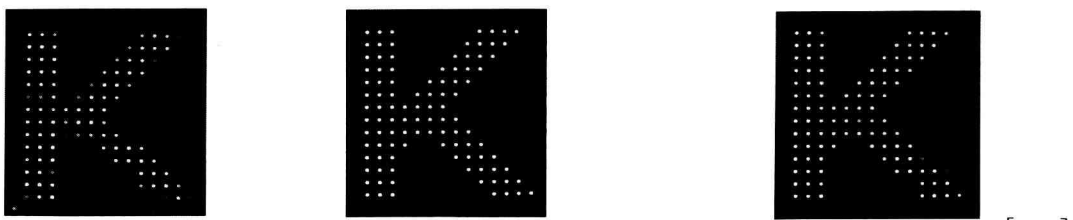
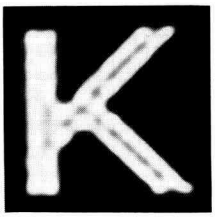


Fig. 4. An original object of 16×16 binary numbers

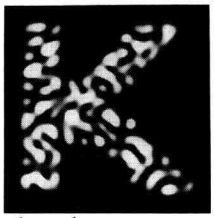


(a) constant phase　(b) random phase on a range $[0, 2\pi]$　(c) random phase on a range $\left[0, \frac{\pi}{2}\right]$

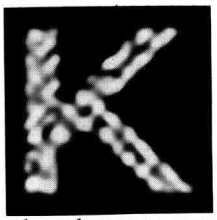
Fig. 5. Examples of simulated image in case of mode＝0.



(a) constant phase



(b) random phase on a range $[0, 2\pi]$



(c) random phase on a range $\left[0, \frac{\pi}{2}\right]$

Fig. 6. Examples of simulated image in case of mode=1.

く，(b) $[0, 2\pi]$ 上のランダム位相の場合にはスペックル雑音が大きく振幅の量子化雑音が小さいことがわかる。また(c)の場合の雑音は(a)と(b)の場合の中間の大きさである。これらのことは理論的に予測される結果と一致する。このことは再生像シミュレータの有用性を示すものといえよう。

6. あ と が き

ディジタルホログラフィの研究にとって有用な再生像シミュレータを作成した。これはフーリエ変換形計算機ホログラムの再生像を模擬してディスプレイに表示させるものである。本論文で述べたものは，第1段階として，計算機ホログラムの各タイプに共通した部分を模擬の対象とした。このシミュレータによって得られる画像は，画質の良否を十分に判別できるものであり，これからの研究を進める上で役立つものと思われる。

再生像シミュレータの改善点は，計算機ホログラムの各タイプに固有の問題，すなわち再生像の形式やホログラムデータの表示誤差などを考慮していくことである。たとえば，ローマンタイプホログラムの場合には共役像や高次回折像を，またキノフォームの場合には位相不整合の影響を取り込めるようなものへと拡張していく必要がある。

文 献

1) A.W. Lohmann and D.P. Paris, "Binary Fraunhofer Holograms, Generated by Computer", Appl. Opt. **6**, 1739 (1967).
2) B.R. Brown and A.W. Lohmann, "Computer-generated Binary Holograms", IBM J. Res. Develop. **13**, 160 (1969).
3) W.H. Lee, "Sampled Fourier Transform Hologram Generated by Computer", Appl. Opt. **9**, 639 (1970).
4) C.K. Hsueh and A.A. Sawchuck, "Computer Generated Double Phase Holograms", Appl. Opt. **17**, 3874 (1978).
5) L.B. Lesem, P.M. Hirsch, J.A. Jordan, Jr., "The Kinoform : A New Wavefront Reconstruction Device," IBM J. Res. Develop. **13**, 150 (1969).
6) D.C. Chu, J.R. Fienup and J.W. Goodman, "Multiemulsion On-Axis Computer Generated Holograms," Appl. Opt. **12**, 1386 (1973).
7) R. Hauck and O. Bryngdahl, "Computer-Generated Holograms with Pulse-Density Modulation," J. Opt. Soc. Am. A**1**, 5 (1984).
8) H. Akahori, "Spectrum Leveling by an Iterative Algorithm with a Dummy Area for Synthesizing the Kinoform," Appl. Opt. **25**, 802 (1986).
9) M.A. Seldowitz, J.P. Allebach, and D.W. Sweeney, "Synthesis of Digital Holograms by Direct Binary Search", Appl. Opt. **26**, 2788 (1987).
10) B.K. Jennison and J.P. Allebach and D.W. Sweeney, "Iterative Approaches to Computer-Generated Holography", Opt. Eng. **28**, 629 (1989).
11) F. Wyrowsky, "Iterative Quantization of Digital Amplitude Holograms," Appl. Opt. **28**, 3864 (1989).
12) F. Wyrowsky and O. Bryngdahl, "Speckle-free Reconstruction in Digital Holography," J. Opt. Soc. Am. A**6**, 1171 (1989).

付　　録

サブルーチンプログラムのリスト

```
c     *** disp.f ***
c
      subroutine disp(a,m,mode)
      dimension a(m,m)
      integer cenx,ceny,llx,lly,urx,ury
c
      nw=800/m
      if(nw.eq.0) go to 3000
c
      nx=(1152-nw*m)/2
      ny=(900-nw*m)/2
c
      amax=0.0
      do 10 i=1,m
      do 10 j=1,m
   10 if(a(i,j).gt.amax) amax=a(i,j)
      p=amax/256.0
c
      call init
c
      if(mode.eq.1) go to 1000
c
      if(nw.lt.4) go to 3000
      nwh=nw/2
      nrad=nw/8
      if(nrad.eq.0) nrad=1
c
      ceny=ny+nwh
      do 20 i=1,m
      ceny=ceny+nw
      cenx=nx+nwh
      do 20 j=1,m
      cenx=cenx+nw
      ic=int((a(i,j)/p)+0.5)
      if(ic.ge.256) ic=255
   20 call circ(cenx,ceny,nrad,ic)
      go to 2000
c
 1000 lly=ny
      ury=ny+nw-1
      do 40 i=1,m
      llx=nx
      urx=nx+nw-1
      do 30 j=1,m
      ic=int((a(i,j)/p)+0.5)
      if(ic.ge.256) ic=255
      call rect(llx,lly,urx,ury,ic)
      llx=llx+nw
   30 urx=urx+nw
      lly=lly+nw
   40 ury=ury+nw
c
 2000 call term
      go to 4000
c
 3000 write(6,100)
  100 format('error')
 4000 return
      end
```

```
/* SUBCGI.C */

/* INIT.C */
#include <cgidefs.h>
#define NCOLOR 256
Cint    name;
Cvwsurf device;

init_()
{
        Ccoor   ll,
                ur;

        static Ccoor    vpll = {0,0};
        static Ccoor    vpur = {1152,900};

        Ccentry clist;
        static unsigned char    red[NCOLOR];
        static unsigned char    grn[NCOLOR];
        static unsigned char    blu[NCOLOR];

        int     i;
        for(i = 0; i < NCOLOR; i++)
        {
                red[i] = i;
                grn[i] = i;
                blu[i] = i;
        }
```

```
        open_cgi();

        NORMAL_VWSURF(device, CG2DD);
        device.cmapname[0] = 'm';
        device.cmapsize = NCOLOR;
        open_vws(&name, &device);

        clist.ra = red;
        clist.ga = grn;
        clist.ba = blu;
        clist.n = NCOLOR;
        color_table(0, &clist);
        vdc_extent(&vpll, &vpur);
}

/* RECT.C */

rect_(llx,lly,urx,ury,ic)
        int     *llx,*lly,*urx,*ury;
        int     *ic;
{
        Ccoor   ll,
                ur;
        int     llyl,
                uryl;

        ll.x = *llx;
        ur.x = *urx;
        llyl = *lly;
        uryl = *ury;
        ll.y = 900 - uryl;
        ur.y = 900 - llyl;
        interior_style(SOLIDI,OFF);
        fill_color(*ic);
        rectangle(&ll, &ur);
}

/* CIRC.C */

circ_(cenx,ceny,rad,ic)
        int     *cenx,*ceny,*rad;
        int     *ic;
{
        Ccoor   cen;
        int     radl;

        cen.x = *cenx;
        cen.y = *ceny;
        cen.y = 900-cen.y;
        radl = *rad;
        interior_style(SOLIDI,OFF);
        fill_color(*ic);
        circle(&cen,radl);
}

/* TERM.C */

term_()
{
        getchar();
        close_vws(name);
        close_cgi();
}
```