

# 6 Digitale Holografie

Volker Ahlers

Nichts spricht grundsätzlich dagegen, die Transmissionsverteilung eines Hologramms mittels der im Abschnitt über Holografie beschriebenen Formeln von einem Computerprogramm berechnen zu lassen. Rechengeschwindigkeit und Speicherplatz des verwendeten Computers setzen dem theoretisch Denkbaren jedoch schnell praktische Grenzen. So muß die Hologrammplatte in eine endliche Anzahl von Zellen eingeteilt werden, was die Auflösung allzu feiner Interferenzmuster verhindern kann; auch die Objektinformation muß in einer endlichen Anzahl von Speicherzellen abgelegt sein. Größere Probleme bereitet allerdings die benötigte Rechenzeit; eine genaue Simulation der Beleuchtung eines dreidimensionalen Objektes inklusive Streuung an der Oberfläche und Beugung an Kanten ist, zumindest mit heutigen seriellen Computern, unmöglich. Eine Beschränkung auf vergleichsweise einfache, transparente Objekte ist daher unumgänglich.

Nichtsdestotrotz findet die digitale Holografie einige Anwendungen, da Hologramme von Objekten erstellt werden können, die entweder real nicht existieren oder nicht in gewöhnlicher Form holografisch aufgenommen werden können. In den ersten Bereich fallen holografische Filter (s. Abschnitt über Fourieroptik), in den zweiten beispielsweise astronomische Objekte, die mittels digitaler Holografie dreidimensional dargestellt werden können (s. auch [40]). Zudem sind durch die weitere Entwicklung von Parallelrechnern gewaltige Rechenzeitverkürzungen absehbar.

## 6.1 Berechnung des digitalen Hologramms

Wir beginnen damit, unsere Hologrammplatte in die  $x$ - $y$ -Ebene zu setzen (s. Abb. 40). Da der Computer digital rechnet, muß diskretisiert werden, die Platte wird also fortan als Raster aus  $M \times N$  Punkten  $Q_{mn}$  mit den Koordinaten

$$(x_m, y_n, 0), \quad m = 0 \dots M-1, n = 0 \dots N-1$$

angesehen werden. Da numerisch stets mit dimensionslosen Größen gerechnet wird, Längen und Abstände also keine absolute Bedeutung haben, können wir die Rasterweiten in  $x$ - und  $y$ -Richtung getrost gleich eins setzen.

Das Objekt wird durch eine Ansammlung von  $L$  Punkten  $P_l$  mit den Koordinaten

$$(x_l, y_l, z_l), \quad l = 0 \dots L-1$$

repräsentiert, von denen jeweils Licht in Form einer Kugelwelle ausgesendet wird, die im Punkte  $Q_{mn}$  die Amplitude

$$E_{lmn} = \frac{A_l}{r_{lmn}} \exp(i(kr_{lmn} + \phi_l)) \quad (57)$$

besitzt, wobei

$$r_{lmn} = \sqrt{(x_l - x_m)^2 + (y_l - y_m)^2 + z_l^2} \quad (58)$$

der Abstand zwischen  $P_l$  und  $Q_{mn}$  ist und die Bedeutung der anderen Terme gleich erklärt werden wird.

Um den Rechenaufwand in Maßen zu halten, nehmen wir einige Vereinfachungen vor:

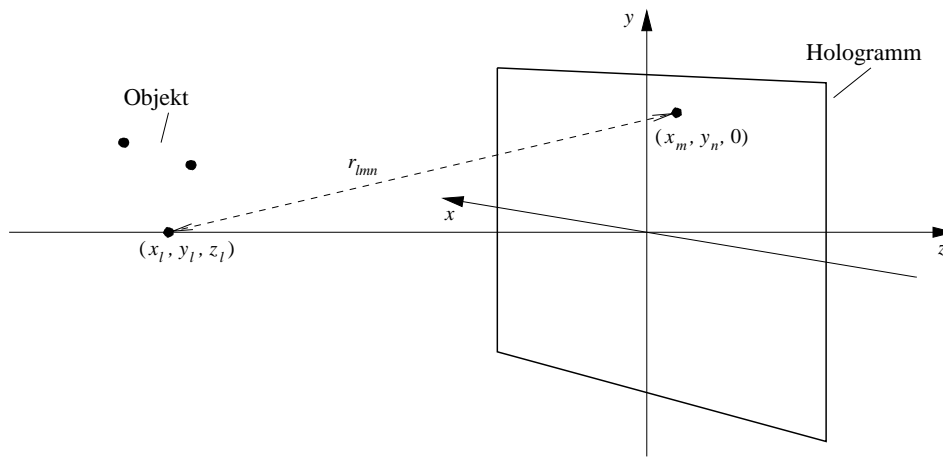


Abbildung 40: Zur Erklärung der verwendeten Koordinaten

- Alle Objektpunkte senden Lichtwellen der gleichen Amplitude aus:

$$A_l = A = \text{const} \quad \text{für alle } l \quad (59)$$

- Außerdem senden sie Lichtwellen gleicher Phase aus, die wir gleich null setzen:

$$\phi_l = 0 \quad \text{für alle } l \quad (60)$$

- Das Objekt befinde sich im Fernfeld, so daß wir setzen können:

$$\frac{A_l}{r_{lmn}} = \frac{A}{r_{lmn}} = \text{const}; \quad (61)$$

der Einfachheit halber setzen wir diese Konstante gleich eins:

$$\frac{A_l}{r_{lmn}} = 1 \quad \text{für alle } l, m, n \quad (62)$$

- Da mit dimensionslosen Größen gerechnet wird, können wir der Wellenlänge  $\lambda$  des Laserlichtes einen beliebigen — auch grob unrealistischen — Wert zuweisen. Um  $M \times N \times L$  weitere Multiplikationen einzusparen, setzen wir daher  $\lambda = 2\pi$  und erhalten

$$k = \frac{2\pi}{\lambda} = 1 \quad (63)$$

Schauen wir nun, was vom ursprünglichen Ausdruck noch übrig geblieben ist, so finden wir

$$E_{lmn} = \exp(ir_{lmn}). \quad (64)$$

Hier kann  $r_{lmn}$  trotz Fernfeldannahme natürlich nicht als konstant angenommen werden, da es gerade die erwünschte Modulation des Referenzlichtes bewirkt; während kleine relative Änderungen im Betrag von  $E_{lmn}$  keine großen Auswirkungen haben, hängt die komplexe Exponentialfunktion empfindlich von  $r_{lmn}$  ab.

Die Amplituden aller in  $Q_{mn}$  eintreffenden Kugelwellen addieren sich zu

$$E_{Smn} = \sum_{l=0}^{L-1} E_{lmn}; \quad (65)$$

das S steht dabei für *Signal*. Hinzu kommt noch das Referenzlicht, das der Einfachheit halber als ebene Welle mit einer Ausbreitungsrichtung senkrecht zur Hologrammplatte angenommen wird, so daß gilt:

$$E_{Rmn} = E_R = \text{const} \quad \text{für alle } m, n; \quad (66)$$

ferner sei  $E_R$  reell. Für die Intensität im Punkte  $Q_{mn}$  folgt also

$$\begin{aligned} I_{mn} &= |E_{Smn} + E_R|^2 \\ &= (E_{Smn} + E_R)(E_{Smn} + E_R)^* \\ &= |E_{Smn}|^2 + (E_{Smn} + E_{Smn}^*)E_R + E_R^2 \\ &= |E_{Smn}|^2 + 2\text{Re}(E_{Smn})E_R + E_R^2, \end{aligned} \quad (67)$$

wobei ausgenutzt wurde, daß  $E_R$  reell ist.

Eine wirkliche Hologrammplatte erhält durch die Belichtung  $B(x, y) = t_B I(x, y)$  eine Transmissionsverteilung

$$T(x, y) = a - bB(x, y), \quad (68)$$

sofern im linearen Bereich der Kennlinie des Fotomaterials gearbeitet wird. Da der Drucker, auf dem das digitale Hologramm später ausgegeben werden soll, keine Graustufen, sondern nur schwarze oder weiße Punkte drucken kann, ist eine Übernahme dieser Formel nicht sinnvoll; eine Erzeugung von Graustufen durch Rasterung rief nämlich unerwünschte Beugungen hervor. Abhilfe schafft eine binäre Kennlinie, die je nach Intensität entscheidet, ob ein weißer ( $T(x, y) = 1$ ) oder schwarzer ( $T(x, y) = 0$ ) Punkt gesetzt wird (s. Abb. 41). Diese Vereinfachung ist nicht so unrealistisch wie sie auf den

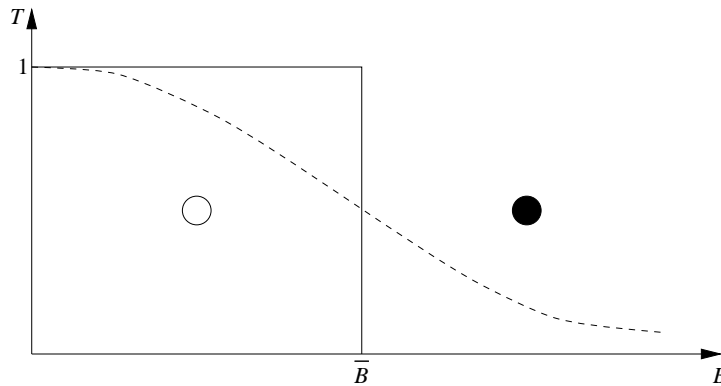


Abbildung 41: Transmissionskennlinien eines realen (gestrichelte Kurve) und eines digitalen (durchgezogene Kurve) Hologramms

ersten Blick wirken mag, da auch die Kennlinien einiger realer Fotomaterialien recht steil sind; tatsächlich bringt das Abfotografieren des berechneten Hologramms von einem Graustufenbildschirm keine wesentliche Verbesserung gegenüber der hier verwendeten binären Ausgabe auf einem Drucker. Zur Darstellung einer rechteckigen Kennlinie sind höhere Potenzen erforderlich,

$$T = a + bB + cB^2 + dB^3 + \dots, \quad (69)$$

die sich bei der Rekonstruktion durch höhere Beugungsordnungen bemerkbar machen; diese sind bei wenigen Objektpunkten deutlich erkennbar, verschwimmen jedoch bei größerem  $L$ .

Um die mittlere Belichtung  $\bar{B} = t_B \bar{I}$  zu bestimmen, muß nicht erst die ganze Intensitätsverteilung berechnet werden; nehmen wir nämlich an, daß die Amplitude der Referenzwelle sehr viel größer ist als die der Signalwelle,

$$E_R \gg E_{S_{mn}} \quad \text{für alle } m, n, \quad (70)$$

so können wir setzen

$$\bar{I} = |E_R|^2 = E_R^2. \quad (71)$$

Die Belichtungszeit hat keine Bedeutung, kann also gleich eins gesetzt werden, so daß sie verschwindet. Ein schwarzer Punkt wird gesetzt, falls gilt:

$$I_{mn} \geq \bar{I}, \quad (72)$$

Einsetzen von (67) und (71) liefert

$$|E_{S_{mn}}|^2 + 2\operatorname{Re}(E_{S_{mn}})E_R + E_R^2 \geq E_R^2 \quad (73)$$

$$\Leftrightarrow |E_{S_{mn}}|^2 + 2\operatorname{Re}(E_{S_{mn}})E_R \geq 0. \quad (74)$$

$\operatorname{Re}(E_{S_{mn}})$  ist in der Größenordnung von  $|E_{S_{mn}}|$ ; da  $E_R$  sehr viel größer ist als  $E_{S_{mn}}$ , spielt der Term  $|E_{S_{mn}}|^2$  für den Wert der linken Gleichungsseite keine große Rolle und kann vernachlässigt werden. Dann hängt das Vorzeichen der linken Gleichungsseite nur noch von  $\operatorname{Re}(E_{S_{mn}})$  ab, nehmen wir  $E_R$  beispielsweise als positiv an, so folgt als Bedingung für das Setzen eines schwarzen Punktes<sup>2</sup>

$$\operatorname{Re}(E_{S_{mn}}) \geq 0; \quad (75)$$

bei negativem  $E_R$  wird aus dem  $\geq$  ein  $\leq$ , was zu einer Invertierung des Ergebnisses führt. Da der Realteil einer endlichen Summe die Summe der Realteile ihrer Summanden ist, folgt durch Einsetzen von (64) und (65) unter Benutzung der Eulerschen Formel als Bedingung

$$\sum_{l=0}^{L-1} \cos(r_{lmn}) \geq 0. \quad (76)$$

Bemerkenswert an diesem Ergebnis ist, daß keine komplexen Zahlen mehr auftauchen, was für eine Überführung des Algorithmus in ein Computerprogramm ein großer Vorteil ist; außerdem ist die Amplitude der Referenzwelle ganz aus den Gleichungen verschwunden.

Eine weitere Vereinfachung der Rechenoperationen kann erreicht werden, wenn mit Rechteckwellen statt der hier verwendeten Sinuswellen gearbeitet wird (für Details s. [1], S. 128f); die Ergebnisse zeigen, daß Holografie nicht an sinusförmige Lichtwellen gebunden ist.

---

<sup>2</sup>Während des Praktikums wurde hier eine eher anschauliche Argumentation gegeben, s. [1], S. 123f.

## 6.2 Das Computerprogramm

Das folgende C-Programm ist im wesentlichen identisch mit dem von W. Lauterborn, T. Kurz und M. Wiesenfeldt gegebenen Beispiel (s. [1], S. 269). Die Hologrammplatte wird so plaziert, daß die optische Achse durch ihre Mitte geht. Um dies zu erreichen, wird schon beim Einlesen der Daten zu den  $x$ - und  $y$ -Koordinaten ein Offset von  $M/2$  beziehungsweise  $N/2$  addiert, außerdem werden die  $z$ -Koordinaten sofort quadriert; beides spart später Rechenoperationen. Die berechnete Transmissionsverteilung wird in ein `pgm`-File geschrieben. Dieses besteht aus einem ASCII-Header, der unter anderem die Größe des Bildes enthält, und einem Datenteil, in dem für jeden Pixel ein Byte die Graustufe angibt; benötigt werden nur die beiden extremen Graustufen 0 (weiß) und 255 (schwarz). Die Konstruktion `cos(fmod(..., ZWEI_PI))` ist ratsam, da der Kosinus üblicherweise durch Reihenentwicklung berechnet wird, so daß Argumente  $\gg 2\pi$  erhebliche Fehler hervorrufen können.

```
#include <stdio.h>
#include <math.h>

#define ZWEI_PI      6.2831853071796
#define MAX_L       200                /* Max. Anz. Objektpunkte */
#define HOLO_M      6000               /* Hologrammgroesse M*N   */
#define HOLO_N      4000

void lies_objekt(char *,int *,int *,int *,int *);

int main(int argc,char **argv)        /* Filenamen aus Kommandozeile */
{
    int obj_l = 0;                      /* Anz. Objektpunkte L     */
    int x_off[MAX_L],y_off[MAX_L];      /* x_l+M/2, y_l+N/2       */
    int z_qu[MAX_L];                    /* z_l^2                   */
    int dy_qu[MAX_L];                   /* (y_off-n)^2 fuer Zeile n */
    int dx,dy;                          /* x_off-m, y_off-n       */
    double summe;                       /* sum{cos{r_lmn}}        */
    int l,m,n;
    FILE *out;

    lies_objekt(argv[1],&obj_l,x_off,y_off,z_qu);
    out = fopen(argv[2],"wb");
    fprintf(out,"P5\n%d %d 255\n",HOLO_M,HOLO_N); /* Header fuer pgm */
    for(n=0;n<HOLO_N;n++) {              /* Schleife ueber Zeilen */
        for(l=0;l<obj_l;l++) {
            dy = y_off[l]-n;              /* dy^2 zwischenspeichern, da... */
            dy_qu[l] = dy*dy;             /* ...innerhalb einer Zeile konstant */
        }
        for(m=0;m<HOLO_M;m++) {          /* Schleife ueber Spalten */
            summe = 0.;
            for(l=0;l<obj_l;l++) {       /* Schleife ueber Obj.punkte */
                dx = x_off[l]-m;
                summe += cos(fmod(sqrt(dx*dx + dy_qu[l] + z_qu[l]),ZWEI_PI));
            }
            if(summe >= 0.)               /* Plot-Entscheidung: */
                fputc(255,out);          /* schwarzer Pixel */
            else
                fputc(0,out);             /* weisser Pixel */
        }
    }
    fclose(out);
    return 0;
}                                          /* end of main */
```

```

void lies_objekt(char *filename,int *obj_l,int *x_off,int *y_off,int *z_qu)
{
    int x,y,z;                                /* x_l, y_l, z_l */
    FILE *in;

    in = fopen(filename,"r");
    while(fscanf(in,"%d%d%d",&x,&y,&z) == 3) {
        x_off[*obj_l] = x+HOLO_M/2;          /* addiere Offset zu x_l, y_l */
        y_off[*obj_l] = y+HOLO_N/2;
        z_qu[*obj_l] = z*z;                  /* quadriere z_l          */
        (*obj_l)++;                          /* erhoehe L              */
    }
    fclose(in);
}                                             /* end of lies_objekt    */

```

Der weite Graustufenbereich des pgm-Formates wird natürlich nicht ausgenutzt; eleganter wäre es, ein Bitmapformat zu verwenden, in dem pro Pixel nur ein Bit angibt, ob ein schwarzer oder weißer Punkt gesetzt werden soll. Eine enorme Verkürzung der Rechenzeit könnte der Einsatz eines Parallelcomputers bringen. Da jeder Hologrammpunkt  $Q_{mn}$  separat berechnet und dabei von den übrigen Punkten nicht beeinflusst wird, könnten verschiedene Prozessoren gleichzeitig verschiedene Hologrammpunkte bearbeiten, eine Aufgabe, die wie geschaffen für den Einsatz eines SIMD-Rechners (Same Instruction, Multiple Data) ist.

### 6.3 Beispiele einiger berechneter Hologramme

Das Hologramm eines einzelnen, eine Kugelwelle aussendenden Punktes ist eine Fresnelsche Zonenplatte (s. Abb. 42); zwei Punkte erzeugen entsprechend eine Struktur, in der zwei sich überlagernde Zonenplatten erahnt werden können (s. Abb. 43).

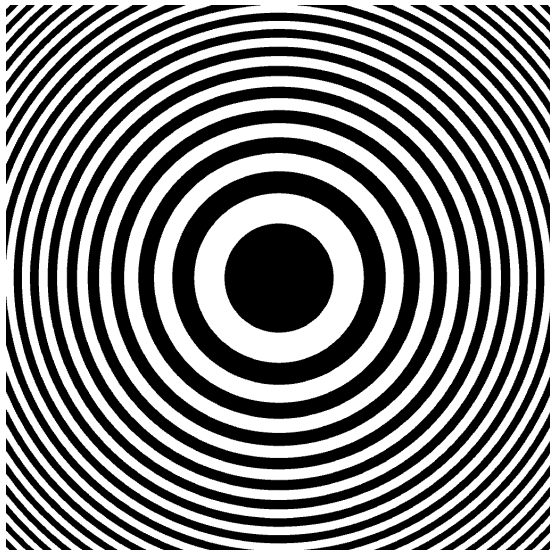


Abbildung 42: Digitale Hologramme eines Punktes bei  $(0, 0, -9000)$ , ...

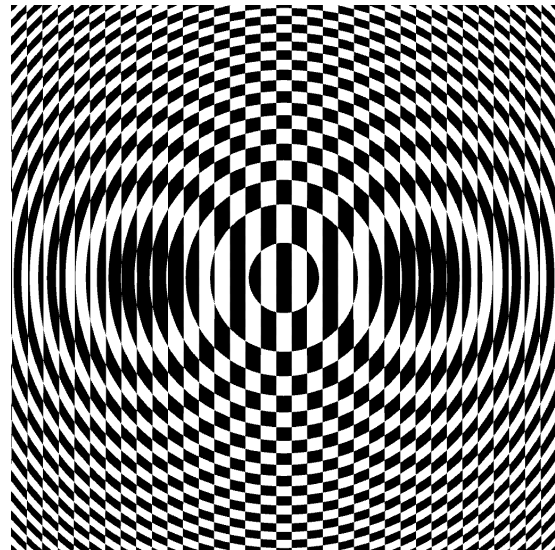


Abbildung 43: ... zweier nebeneinander liegender Punkte bei  $(\pm 500, 0, -9000)$  ...

Als etwas interessanteres Objekt wurde der Schriftzug „HOLO 30“ aus 78 Punkten erzeugt, deren  $x$ - beziehungsweise  $y$ -Koordinaten Werte zwischen  $-1000$  und  $+1100$  beziehungsweise  $+3000$  und  $+4300$  haben. Das Wort „HOLO“ und die Zahl „30“ liegen in zwei

verschiedenen Ebenen parallel zur Hologrammplatte, bei  $z = -10000$  beziehungsweise  $z = -9800$ . Abb. 44 zeigt das berechnete Hologramm.

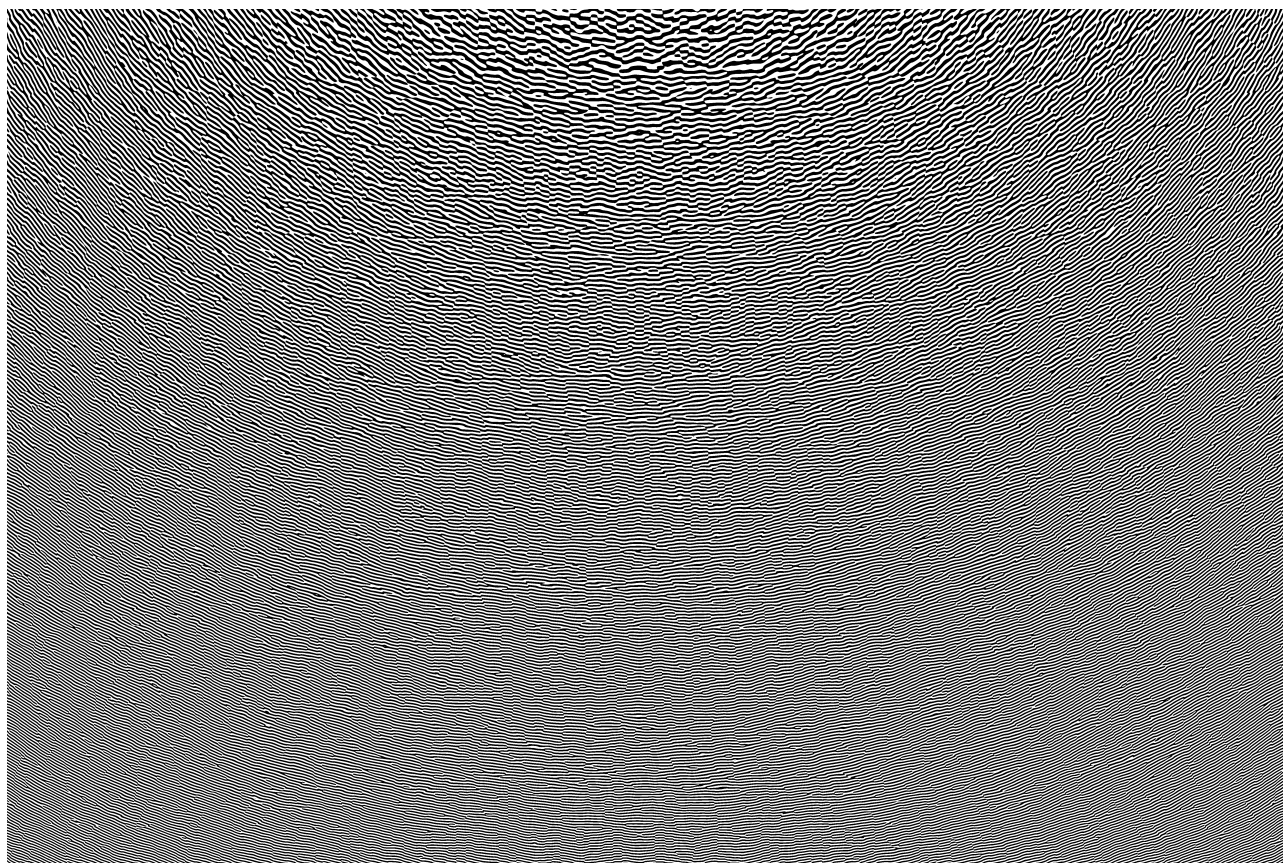


Abbildung 44: ... und des aus 78 Punkten bestehenden Schriftzuges „HOLO 30“

Auf einer institutseigenen Sun Ultra-Sparc („Hopper“) wurden folgende Rechenzeiten gemessen:

Objekt	Hologrammgröße	ungefähre Rechenzeit
ein Punkt	$2000 \times 2000$	25 s
zwei Punkte	$2000 \times 2000$	50 s
„HOLO 30“ (78 Punkte)	$6000 \times 4000$	3 h

## 6.4 Rekonstruktion

Das pgm-File muß, um auf einem Drucker ausgegeben werden zu können, in das Postscriptformat umgewandelt werden; dazu wurde xv verwendet; es wurde darauf geachtet, daß das Bild in Originalgröße („Normal Size“) und ohne Graustufen („B/W Dithered“) gespeichert wurde. Außerdem wurde beim Speichern die Auflösung des Bildes an die des verwendeten Druckers angeglichen, der in unserem Fall ein HP LaserJet 4+ mit 600 dpi (dots per inch) war. Das auf Papier gedruckte Hologramm wurde anschließend auf hochauflösenden Positivfilm (Dokumentenfilm) abfotografiert.

Sei  $a$  die Breite des entstandenen Dias; da das Hologramm mit einer Längeneinheit von 1 berechnet wurde, ergibt sich ein Verkleinerungsfaktor von

$$\gamma = \frac{a}{M}, \quad [\gamma] = 1 \text{ m.} \quad (77)$$

Die für die „Aufnahme“ benutzte Wellenlänge errechnet sich damit zu

$$\lambda' = \gamma \cdot 2\pi, \quad (78)$$

der Punkt  $P'_l$  bekommt die Koordinaten

$$(x'_l, y'_l, z'_l) = (\gamma x, \gamma y, \gamma z). \quad (79)$$

Wird das Hologramm mit Laserlicht der Wellenlänge  $\tilde{\lambda}$  rekonstruiert, so ergibt sich — bei senkrechter Beleuchtung — eine weitere Skalierung in  $z$ -Richtung (s. [1], S. 111f), so daß sich als Koordinaten für den rekonstruierten Punkt  $\tilde{P}_l$  ergeben

$$(\tilde{x}_l, \tilde{y}_l, \tilde{z}_l) = \left( \gamma x, \gamma y, \frac{2\pi\gamma^2}{\tilde{\lambda}} z \right). \quad (80)$$

Wegen des senkrecht einfallenden Referenzlichtes kann bei der Rekonstruktion das virtuelle direkte Bild nicht durch das Hologramm hindurch betrachtet werden. Wünschenswert sind daher Abstände  $\tilde{z}_l$  von bis zu einem Meter, damit das reelle, konjugierte Bild auf einem Tisch aufgefangen werden kann; mit  $M = 6000$ ,  $a = 36 \text{ mm}$  (Kleinbildfilm) und  $\tilde{\lambda} = 633 \text{ nm}$  (He-Ne-Laser) ergeben sich  $z_l$ -Koordinaten von etwa  $-3000$ , die jedoch nicht unbedingt realisiert werden können, da das Interferenzmuster des Hologramms um so feiner wird, je näher das Objekt an die Hologrammplatte heranrückt, und daher bei zu kleinem  $z_l$  nicht mehr aufgelöst werden kann. Deshalb ist es gegebenenfalls besser,  $z_l$  groß zu wählen und bei der Rekonstruktion Linsen zu verwenden, um das Bild auf einem Tisch von endlicher Länge in vernünftiger Größe betrachten zu können. Den bei der Rekonstruktion der berechneten digitalen Hologramme verwendeten Strahlengang zeigt Abb. 45. Linse L1 und die Blende wurden durch ein Raumfrequenzfilter *ohne* eingelegtes

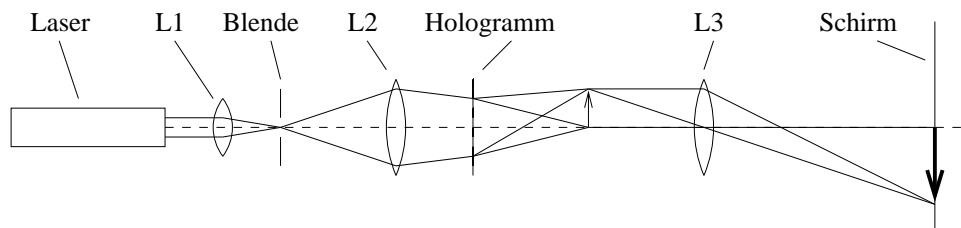


Abbildung 45: Strahlengang zur Rekonstruktion



Pinhole gebildet, die Linsen L2 und L3 besaßen Brennweiten von 80 mm beziehungsweise 50 mm; L3 diente zum Vergrößern des reellen konjugierten Bildes.

Die Rekonstruktionen der Hologramme eines beziehungsweise zweier Punkte waren stark verrauscht, Abb. 46 zeigt das (noch etwas nachbehandelte) Bild zweier Punkte. Der

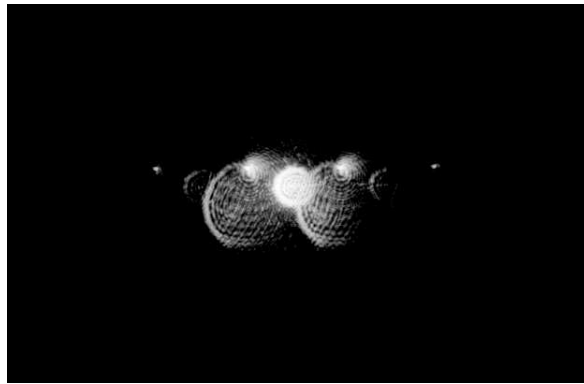


Abbildung 46: Rekonstruktionen der Hologramme zweier Punkte ...

mittlere, helle Kreis ist die verbreiterte nullte Ordnung, direkt links und rechts darüber sind die beiden rekonstruierten Punkte zu erkennen. Die großen Kreise unterhalb dieser beiden Punkte sind unerwünschte Beugungsmuster. Weiter außen sind links und rechts noch zwei kleinere Punkte zu erkennen; dies sind die durch die rechteckige Kennlinie hervorgerufenen höheren Beugungsordnungen (s. Abschnitt 6.1).

Die zwei Ebenen, in denen die Bestandteile des Schriftzuges „HOLO 30“ liegen, waren bei der Rekonstruktion nur mit viel Wohlwollen zu unterscheiden; ein größerer Unterschied zwischen den beiden  $z$ -Koordinaten wäre von Vorteil gewesen. Die dreidimensionale Natur des rekonstruierten Bildes ließ sich jedoch erkennen, wenn unterschiedliche Regionen des Hologramms beleuchtet wurden. Die beiden Bestandteile „HOLO“ und „30“



Abbildung 47: ... und des Schriftzuges „HOLO 30“ ...



Abbildung 48: ... in zwei verschiedenen Ansichten

verschoben sich gegeneinander, so als würde man mal von vorn, mal von der Seite, mal von unten auf das Bild schauen. Abb. 47 und 48 zeigen zwei solche Ansichten. Scharf abgebildet ist jeweils das konjugierte Bild, der helle Kreis in der Mitte stellt wieder die verbreiterte nullte Ordnung dar, der Bereich unterhalb davon das direkte Bild, welches durch Verschieben von L3 scharf abgebildet werden konnte.