

# Sistema para Visualização Holográfica de Figuras Geradas por Computador

**Marcelo Diamand**

*Dissertação apresentada à  
Faculdade de Engenharia  
Elétrica como pré-requisito  
parcial para a obtenção do  
título de Mestre em  
Engenharia Elétrica.*

**Orientador :  
Prof. Dr. José Joaquín Lunazzi**

9510769  
Depto. de Engenharia de Computação e Automação Industrial  
Faculdade de Engenharia Elétrica  
Universidade Estadual de Campinas

Este exemplar corresponde à redação final da tese  
defendida por Marcelo Diamand

\_\_\_\_\_, aprovada pela Comissão

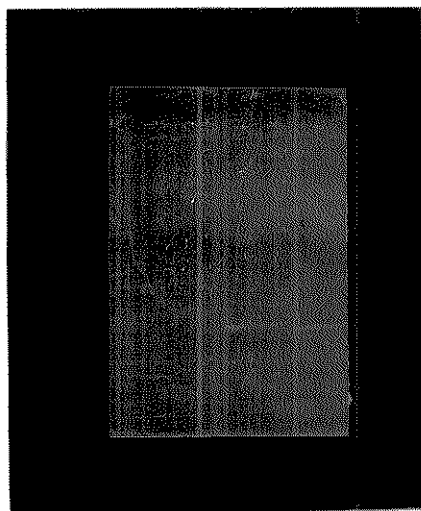
Julgadora em 10 / 12 / 94

Orientador

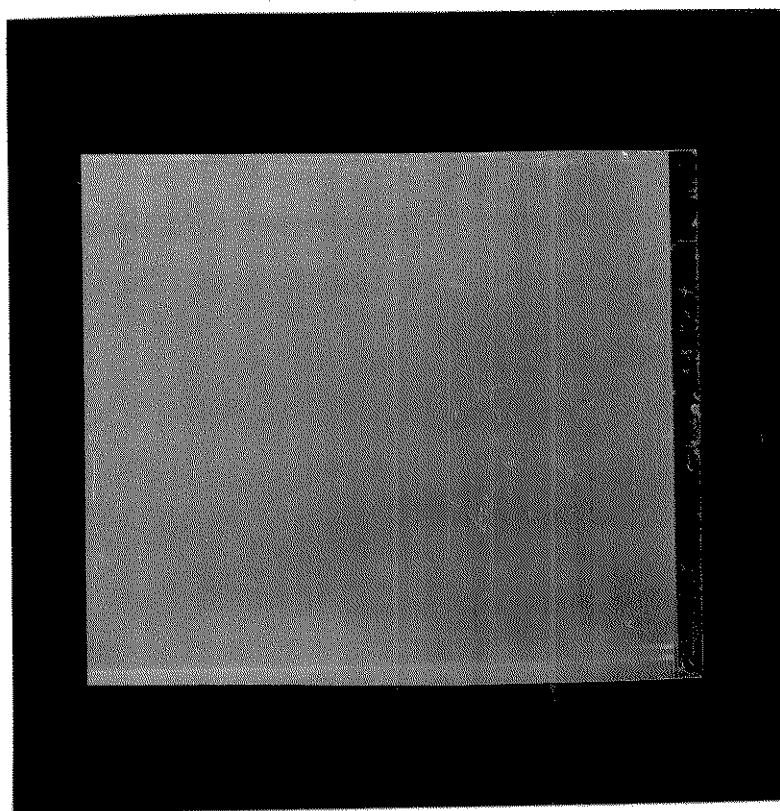


Campinas, 16 de dezembro de 1994

*Dedico este trabalho a meus queridos pais **Arthur** e **Eva**, a minha noiva **Simone** e a meus irmãos **Rosa** e **Andre**, pelo carinho e apoio que sempre me deram.*



**Amostra da rede difrativa de  $v=528$  linhas/mm.**



**Amostra de Tela Holográfica de  $\langle v \rangle = 1.200$  linhas/mm.  
Se iluminada pela direita, apresentará dispersão horizontal do espectro.**

## **RESUMO**

Este trabalho descreve o desenvolvimento de um sistema gerador de figuras tridimensionais que, por meio de tela holográfica, permite observação sem óculos especiais e oferece paralaxe horizontal contínua, como no caso de imagens holográficas convencionais. A geração de imagens é controlada por computador e utiliza luz branca e elementos ópticos holográficos para definir a profundidade (coordenada **Z**) de cada ponto da imagem. Trata-se de uma proposta inédita que apresenta como principal vantagem em relação aos sistemas similares a possibilidade de se alcançar grandes dimensões de exibição através de projeção.

## **ABSTRACT**

This work describes the development of a system generator of three-dimensional figures that, by means of a holographic screen, allows viewing without special goggles and offers continuous horizontal parallax, resembling the case of conventional holographic images. The image generation is controlled by a computer and employs white light and holographic optical elements to define the depth (**Z** coordinate) of every point of the image. This is an original proposal that presents as main advantage regarding similar systems the possibility to achieve large exhibition dimensions through projection.

## AGRADECIMENTOS

Ao Prof. José Joaquín Lunazzi, meus sinceros agradecimentos pela orientação, incentivo e principalmente pela amizade.

Ao Prof. Léo Pini Magalhães, cujo apoio foi importante para a realização deste trabalho.

À Cristina Lunazzi, pelo companheirismo e pelo constante suporte administrativo.

À Prof. Lucila Cescato, pelas valiosas discussões e pela amizade.

À Prof. Neidenêi Gomes Ferreira, pelas dicas importantes sobre a confecção desta dissertação.

A todos os amigos do Lab. de Óptica, especialmente ao Paulo Magno, Agnaldo, Eduardo, David, Bian, Carlos Raimundo, Ivan (1), Alexandra, Leandro, Bernardo e Ivan (2).

Ao pessoal de apoio técnico: João Petrúcio, Do Carmo, Mônica, Alcides, Claudemir (oficina de eletrônica) e Costa (lab. de ensino de Óptica).

A todos os integrantes do grupo de HoloImagens, com os quais tive o prazer de trabalhar e conviver: Juan, Paula, Edmundo, Juliana, Lorena, Marcelo Eduardo, Masao, Giovanni, Jorge e Regina.

A Marcelo Moraes, pelo entusiasmo e pelas valiosas informações.

A todo o pessoal do LCA, em particular a Alexandre Magalhães, pelo auxílio na elaboração do programa de exibição sincronizada de quadros.

Aos colegas e professores da Universidade Federal do Rio de Janeiro, que direta ou indiretamente contribuíram para a conclusão deste trabalho.

A FAPESP, pelo apoio financeiro e pela pontualidade.

# **Índice**

<b>1. INTRODUÇÃO</b>	<b>2</b>
<b>2. VISUALIZAÇÃO TRIDIMENSIONAL</b>	<b>3</b>
2.1 INDICADORES DE PROFUNDIDADE	3
2.2 TECNOLOGIAS EXISTENTES	4
2.3 HOLOPROJETOR : UMA NOVA PROPOSTA	7
<b>3. PRINCÍPIOS DA TELA HOLOGRÁFICA</b>	<b>9</b>
<b>4. TÉCNICAS DESENVOLVIDAS</b>	<b>15</b>
4.1 PROJEÇÃO COM PROFUNDIDADE FIXA	15
4.2 SELETOR DE PROFUNDIDADE	16
4.2.A MECANISMOS DE CODIFICAÇÃO	16
4.2.B PROGRAMA DE SIMULAÇÃO	20
<b>5. PROTÓTIPO A MOTOR DE PASSO</b>	<b>24</b>
5.1 DESCRIÇÃO GERAL	24
5.2 FONTE DE LUZ BRANCA	26
5.3 SISTEMA ÓPTICO	27
5.4 ACIONADOR (DRIVER)	31
5.5 DEFLETORES	33
5.6 PROGRAMA DE OPERAÇÃO, CALIBRAÇÃO E CONTROLE	34
5.7 ANÁLISE DA TAXA DE EXIBIÇÃO	35
5.8 CALIBRAÇÃO DO SISTEMA	36
5.9 EXEMPLO DE UTILIZAÇÃO	37
5.10 ANÁLISE DOS RESULTADOS	40
5.11 SUGESTÕES DE IMPLEMENTAÇÃO	40
5.11.A FILTRO CROMÁTICO DE OCLUSÃO	40
5.11.B PROTÓTIPO BASEADO EM PROJETOR DE TV	41
<b>6. CONCLUSÕES</b>	<b>44</b>

## **REFERÊNCIAS BIBLIOGRÁFICAS**

### **ANEXOS**

**I. Trabalho Aceito para Publicação na Revista APPLIED OPTICS**

**II. Listagem do Programa SIMHPROJ (simulação do sistema)**

**III. Listagem do Programa HPROJ (controle, operação e calibração do sistema)**

# 1. Introdução

A invenção do primeiro estereoscópio em 1838 por Wheatstone [1] marcou o início de uma era de conquistas na visualização natural e realista de imagens tridimensionais. A observação espacial de informações é particularmente útil em áreas como diagnóstico médico, modelamento molecular, CAD e controle de tráfego aéreo.

Os sistemas de exibição tridimensional [2], ou simplesmente displays 3D, têm por objetivo fornecer ao sistema visual do observador indicadores de profundidade [3], pelos quais o cérebro percebe os relacionamentos espaciais da cena. Após um longo período de domínio da estereoscopia novas tecnologias estão surgindo, tais como a holografia eletrônica e os sistemas de varredura volumétrica. A observação de cenas 3D aproxima-se cada vez mais do ideal sugerido pelos filmes de ficção científica, onde simulação e realidade são quase indistinguíveis.

Este trabalho apresenta uma alternativa inédita de display 3D chamado HoloProjeto, que dispensa o uso de óculos especiais e gera imagens volumétricas com paralaxe horizontal contínua a partir de comandos do computador.

No capítulo 2 conceitos básicos da visualização tridimensional são introduzidos, enfatizando-se os indicadores de profundidade utilizados pelo sistema visual humano na percepção da tridimensionalidade. Em seguida analisa-se os principais displays 3D desenvolvidos até hoje, e finalmente propõe-se uma alternativa inédita: o HoloProjeto, um sistema para visualização holográfica de figuras geradas por computador.

O capítulo 3 versa sobre os princípios da tela holográfica, elemento óptico desenvolvido pelo Prof. Lunazzi capaz de realizar projeções tridimensionais, e que tornou possível a elaboração do HoloProjeto.

Técnicas desenvolvidas preliminarmente à construção do protótipo encontram-se descritas no capítulo 4. Aborda-se primeiramente a realização de uma experiência de projeção com profundidade fixa, cujos resultados positivos demonstraram a viabilidade da proposta. Apresenta-se então a sequência de desenvolvimento do seletor de profundidade, que permitiu a geração de figuras volumétricas.

Já no capítulo 5 mostra-se todos os módulos que compõem o protótipo “HoloProjeto versão 1.0” baseado em motores de passo. Após uma descrição geral, são apresentados a fonte de luz branca, o sistema óptico, o acionador dos motores, os defletores e o programa de operação, calibração e controle. Realiza-se então uma análise da taxa de exibição do sistema, importante pela questão da cintilação das imagens. Ilustra-se o uso do sistema através de um exemplo completo, partindo da definição de uma figura (no caso, um cubo) até a obtenção da imagem 3D resultante. Após uma análise dos resultados, vislumbra-se sugestões para futura implementação, incluindo a habilitação para a realização de oclusão e a conversão do atual modo de desenho das figuras de vetorial para varredura volumétrica. Este último modo, baseado em projetor de TV a cristal líquido, permitiria a visualização holográfica de imagens volumétricas multiplanares, como as originárias de tomografia e ressonância magnética.

Finalmente as conclusões a respeito do trabalho desenvolvido encontram-se no capítulo 6.

## 2. Visualização Tridimensional

### 2.1 Indicadores de Profundidade

O sistema visual humano percebe e compreende o mundo em três dimensões, usando indicadores de profundidade que podem ser classificados em dois grupos, como descrito na Tabela 1 e na Tabela 2 [4].

Indicador (Grupo I)	Descrição
Perspectiva Linear	Objetos distantes parecem menores.
Tonalização e Sombreamento	Indicam posições relativas às fontes de luz.
Perspectiva Aérea	Objetos mais distantes parecem menos distintos, nebulosos e mais escuros.
Oclusão	Objetos mais próximos escondem objetos mais distantes.
Gradiente de Textura	Objetos mais distantes possuem a textura mais “borrada”.
Tamanho da Imagem na Retina	Comparação de objetos de dimensões conhecidas. (Ex.: Homem e elefante do mesmo tamanho - O elefante está mais distante.)
Paralaxe temporal	Efeito de profundidade pelo movimento de um objeto 3D.

Tabela 1 - Indicadores de profundidade do grupo I.

Indicador (Grupo II)	Descrição
Acomodação	Mudança na distância focal da lente do olho.
Convergência	Rotação interna dos olhos.
Disparidade binocular	Diferenças entre as imagens dos olhos direito e esquerdo.
Paralaxe de movimento	Mudanças na imagem devido ao movimento do observador.

Tabela 2 - Indicadores de profundidade do grupo II.

O **grupo I** contém os indicadores de profundidade representáveis em um monitor 2D convencional, e sua percepção baseia-se em experiências previamente adquiridas. Já os indicadores do **grupo II** não podem ser representados por um monitor 2D convencional, e fornecem parâmetros físicos pelos quais a noção de profundidade é mais precisa.

Pesquisadores da área de computação gráfica (e, mais recentemente, visualização volumétrica) criam imagens e cenas 3D ilusórias sobre telas 2D através do cálculo e exibição de indicadores do **grupo I**. Estas imagens carecem de indicadores importantes



fornecidos por uma cena 3D verdadeira, apresentam um único ponto de vista para ambos os olhos e necessitam de uma razoável quantidade de cálculo para produzir os indicadores de profundidade (calcular perspectiva; remover linhas e superfícies escondidas; adicionar tonalização, iluminação e sombras; etc.).

Em relação à percepção espacial o ser humano possui sensores com disparidade bilateral comuns à visão, audição e olfato. É claro portanto que o sistema de percepção de posição evoluiu para usar a informação de disparidade bilateral. A disparidade binocular, com a qual a profundidade é inferida pela interpretação do cérebro a partir de pequenas diferenças entre as imagens recebidas por cada olho (*stereopsis*) é a base da tecnologia das imagens ESTÉREO. Cada olho é artificialmente provido de uma vista ligeiramente diferente de uma mesma imagem, frequentemente (mas nem sempre) através de óculos especiais. Estas vistas em conjunto representam a realidade de maneira mais completa. No próximo item serão descritos brevemente estes e outros sistemas de imagens tridimensionais.

## 2.2 Tecnologias Existentes

Os sistemas tridimensionais podem ser classificados em quatro categorias: MONO, ESTÉREO, ESTÉREO-INTERATIVO e HOLO, como mostra a Tabela 3.

Sistema 3D	Características
MONO	<ul style="list-style-type: none"> <li>• Apenas indicadores do grupo I.</li> <li>• Mesma imagem para ambos os olhos.</li> <li>• Pouca precisão na representação da profundidade.</li> </ul>
ESTÉREO	<ul style="list-style-type: none"> <li>• Baseado na disparidade binocular.</li> <li>• Reduz a necessidade de oclusão para projetos em CAD.</li> <li>• Incompatibilidade entre convergência e acomodação.</li> <li>• Utiliza algum mecanismo de canalização das imagens para cada olho.</li> <li>• Ausência de paralaxe de movimento.</li> </ul>
ESTÉREO-INTERATIVO	<ul style="list-style-type: none"> <li>• Atualização da cena 3D por sensoriamento da posição do observador (paralaxe de movimento).</li> <li>• Larga zona de visão.</li> <li>• O usuário precisa “vestir” o equipamento de visualização.</li> <li>• Geralmente para um único observador.</li> </ul>
HOLO	<ul style="list-style-type: none"> <li>• Oferece todos os indicadores de profundidade do grupo II, por oferecer profundidade real.</li> <li>• Observação sem óculos especiais.</li> <li>• Paralaxe contínua.</li> </ul>

Tabela 3 - Classificação geral dos displays 3D

A categoria MONO, que inclui os sistemas de computação gráfica convencional, tem como característica a apresentação da mesma imagem para ambos os olhos, contando apenas com os indicadores de profundidade do grupo I.

Os sistemas estereoscópicos (categoria ESTÉREO) baseiam-se na disparidade binocular como principal indicador de profundidade. Possibilitam também a convergência, utilizada pelo observador na seleção de planos de profundidade, que no entanto é incompatível com a acomodação sobre a tela, prejudicando a naturalidade da observação e limitando a faixa de profundidade permitida. Dependendo da aplicação podem trazer economia de processamento, como por exemplo num sistema de CAD no qual pode-se dispensar a remoção de linhas ocultas, porque elas aparecem diferenciadas em profundidade. Em geral não possuem paralaxe de movimento, ou quando a possuem esta é descontínua. Utilizam diversas técnicas para canalizar as imagens para os olhos correspondentes, tais como óculos coloridos ou polarizados. Aos sistemas que dispensam o uso de óculos dá-se o nome de auto-estereoscópicos. Alguns auto-estereoscópicos utilizam a alteração da convergência, como os “estereogramas” [5], tão difundidos atualmente. Outros empregam folhas lenticulares [6] ou barreiras de paralaxe [7], mas apresentam como principais desvantagens a descontinuidade da zona de visão e posições de inversão de relevo.

A realidade virtual popularizou os sistemas ESTÉREO-INTERATIVOS, sendo aquele montado num capacete o seu exemplo mais conhecido. O protótipo do “capacete” foi apresentado em 1968 por Ivan Sutherland [8][9], para reforçar sua idéia de que a evolução dos displays resultaria na produção de imagens e outros estímulos sensoriais com tal fidelidade que seria impossível distinguir simulação de realidade. O sistema uniu as tecnologias de estereoscopia e sensoriamento de posição, o que permitiu disparidade binocular, paralaxe de movimento e grande zona de visão<sup>1</sup>. Os principais obstáculos ao seu desenvolvimento são o sensoriamento rápido e preciso da posição da cabeça e da mão, e o projeto de um sistema gráfico capaz de atualizar constantemente o par estéreo com a frequência necessária. Outra desvantagem é a intrusão física característica da tecnologia que prejudica a naturalidade da observação. Além do “capacete”, há sistemas interativos baseados em óculos de cristal líquido, nos quais o usuário observa um monitor fixo externo sincronizado com os óculos.

Os displays HOLO se diferenciam dos demais por oferecer todos os indicadores de profundidade do grupo 2, com a paralaxe de movimento contínua e dispensando o uso de óculos especiais. As suas imagens “ocupam espaço”, ou seja, os pontos luminosos que as compõem podem ser detectados óptica ou eletronicamente<sup>2</sup>. Objetos mais próximos de uma cena estão de fato mais próximos, como numa cena real, tal que os olhos do observador focalizam diferentemente em objetos diferentes, dependendo da proximidade de cada um<sup>3</sup>. A pesquisa por displays HOLO possui duas abordagens principais: holografia eletrônica e varredura volumétrica.

A holografia eletrônica teve como motivação a holografia convencional [10], baseada no registro do padrão de interferência entre dois feixes de laser, um proveniente

---

<sup>1</sup> Ângulo sólido definido pela área sobre a qual o observador pode mover-se e ainda ver a imagem.

<sup>2</sup> Esta detecção é possível pela convergência luminosa no plano horizontal.

<sup>3</sup> Compatibilidade natural entre acomodação (monocular) e convergência (binocular).

do objeto e outro servindo como referência. A holografia convencional conta com a formidável capacidade de dados que podem ser armazenados no filme holográfico, mas é capaz de produzir somente imagens estáticas<sup>4</sup>. O objetivo da holografia eletrônica é gerar imagens holográficas de forma dinâmica, tendo sido considerada inicialmente inviável pela quantidade de processamento envolvido [11]. O uso de alguns artifícios para reduzir o volume de informação, tais como a eliminação da paralaxe vertical, permitiu que no MIT se construísse o Vídeo Holográfico [12]. Neste sistema o holograma é simulado linha a linha, com cada porção elementar sendo calculada a partir da soma dos efeitos de todos os pontos do objeto presentes no plano horizontal corrente. Atualmente existem duas versões deste sistema. A de primeira geração, colorido, permite até duas atualizações por segundo de quadros holográficos ocupando 6 Mb cada. Seu volume de imagem é de 40mm (largura), 35mm (altura) e 50mm (profundidade), possuindo uma zona de visão horizontal de 15° e uma taxa de repetição de 37 Hz. Já o sistema de segunda geração é monocromático, com um volume de exibição de 140 x 90 x 150 mm<sup>3</sup>. Os hologramas calculados ocupam 36 Mb, sendo que cada atualização demanda pelo menos 10 segundos. A zona de visão horizontal é de 30°, enquanto que a taxa de repetição é de 30Hz. Apesar dos bons resultados decorrentes principalmente de esforços para otimizar os cálculos, o sistema ainda necessita de um super-computador de arquitetura paralela<sup>5</sup> para simular rapidamente o padrão interferométrico e depende da evolução de algumas tecnologias para poder aumentar as dimensões das imagens. Por estas razões vem sendo considerado pouco prático [13].

Os sistemas HOLO por varredura volumétrica são de vários tipos, sendo que sua principal vantagem a respeito da holografia eletrônica é o ativamento direto dos *voxels* no volume de exibição, o que reduz drasticamente a demanda computacional. Por outro lado estão limitados a apresentar figuras sem oclusão, e dificilmente podem oferecer os demais indicadores de profundidade do grupo I. Um sistema chamado “Espelho Varifocal” [14][15] foi um dos primeiros a usar a varredura volumétrica, e empregou um espelho oscilante para produzir imagens 3D. A idéia básica é utilizar um espelho flexível cuja distância focal possa ser rapidamente alterada, e posicioná-lo de forma a refletir um monitor convencional para o observador. A mudança periódica de distância focal faz com que a distância ao monitor pareça aumentar ou diminuir de alguns centímetros durante um ciclo de tipicamente 30 Hz. Pontos, linhas e outras primitivas simples podem ser exibidas. Foi um dos poucos sistemas HOLO a se tornar um dispositivo comercial. Espelhos chaveados eletronicamente [16] foram os elementos fundamentais para a elaboração de uma alternativa semelhante porém sem partes móveis, reportado em 1985 pela empresa *Tektronix*. A ausência de citações recentes, entretanto, impossibilitou o estudo de suas características e resultados. Outros sistemas utilizam telas móveis para ciclicamente varrerem um certo volume, e sobre estas telas são apresentadas<sup>6</sup> sequencialmente “fatias” da imagem 3D [4]. Costumam oferecer as maiores zonas de

---

<sup>4</sup> A não ser no caso do holograma “multiplex”, capaz de mostrar uma pequena sequência pré-gravada de imagens.

<sup>5</sup> O principal computador utilizado pelo Vídeo Holográfico é o *Connection Machine 2*, com 32000 microprocessadores (cada um com memória local de 64Kb) funcionando em paralelo.

<sup>6</sup> As telas móveis podem ser tanto passivas quanto ativas. As primeiras apenas difundem as imagens nelas projetadas, enquanto que as últimas são compostas por elementos emissores de luz tais como o LED.

visão horizontal (podendo chegar a 360°) mas seu campo de visão<sup>7</sup> é restrito por problemas mecânicos decorrentes da movimentação da tela (rotação ou translação). Além do ruído e da vibração, ainda há por questões de velocidade e até mesmo de segurança a necessidade de utilizar um gabinete transparente cobrindo as imagens, o que prejudica a naturalidade da observação.

Ainda há que se mencionar um outro sistema recente [17] também baseado em varredura volumétrica. Este sistema utiliza um meio fluorescente contido num cubo onde se consegue excitar exclusivamente a região de cruzamento de dois feixe laser. Os feixes laser são movimentados de forma que seu cruzamento realize uma varredura no cubo. Embora promissor, seu desenvolvimento está apenas começando.

Nosso trabalho apresenta uma nova alternativa de sistema HOLO, diferente das abordagens tradicionais de holografia eletrônica e de varredura volumétrica. O sistema, denominado HoloProjeto, é apresentado no próximo item.

### ***2.3 HoloProjeto : Uma Nova Proposta***

Analisando-se os displays tridimensionais apresentados no item anterior nota-se problemas importantes. Para citar alguns exemplos, no caso dos sistemas baseados em varredura volumétrica, o uso de grandes componentes móveis restringe (por razões mecânicas) as dimensões máximas do volume de trabalho, e impede que se formem figuras fora de um gabinete de proteção. Os sistemas estereoscópicos interativos, além de necessitarem de grande processamento para a atualização das perspectivas, exigem que o usuário “vista” equipamentos especiais, o que torna a percepção menos natural. O Vídeo Holográfico do MIT pode exibir apenas pequenas imagens, enquanto demanda grande capacidade computacional para simular o processo de obtenção holográfica.

O sistema para visualização holográfica de figuras geradas por computador, proposto neste trabalho, representa uma nova alternativa de display tridimensional. O processo de ativamente de *voxels* dentro do volume de trabalho é predominantemente óptico, e por dispensar cálculos complicados, pode ser controlado até por um microcomputador como o PC-XT. As figuras são formadas por um tipo especial de projeção, e portanto podem assumir grandes dimensões, dependendo apenas das distâncias utilizadas e da quantidade de luz fornecida pela fonte. Este tipo especial de projeção é chamado de “holoprojeção por codificação cromática” [18] e possibilita que as figuras, observáveis diretamente, se formem parte à frente (imagem real) e parte atrás (imagem virtual) de uma tela holográfica. Inicialmente concebido para funcionar como um sistema vetorial, suas únicas partes móveis são pequenos espelhos, semelhantes aos utilizados em shows de laser convencionais.

A Figura 1 mostra o esquema simplificado do sistema. A partir dos vértices de uma figura, o microcomputador envia comandos para o projetor tridimensional. Um feixe convergente, proveniente de uma fonte de luz branca, é desviado vertical e horizontalmente (atribuição de **X** e **Y**) e sofre dispersão cromática (codificação de **Z**) no interior do projetor. A tela holográfica decodifica a coordenada **Z** e produz um ponto

---

<sup>7</sup> Campo de visão: Maior tamanho de imagem que pode ser vista pelo display.

luminoso (real ou virtual). O observador interpreta a repetição cíclica da trajetória

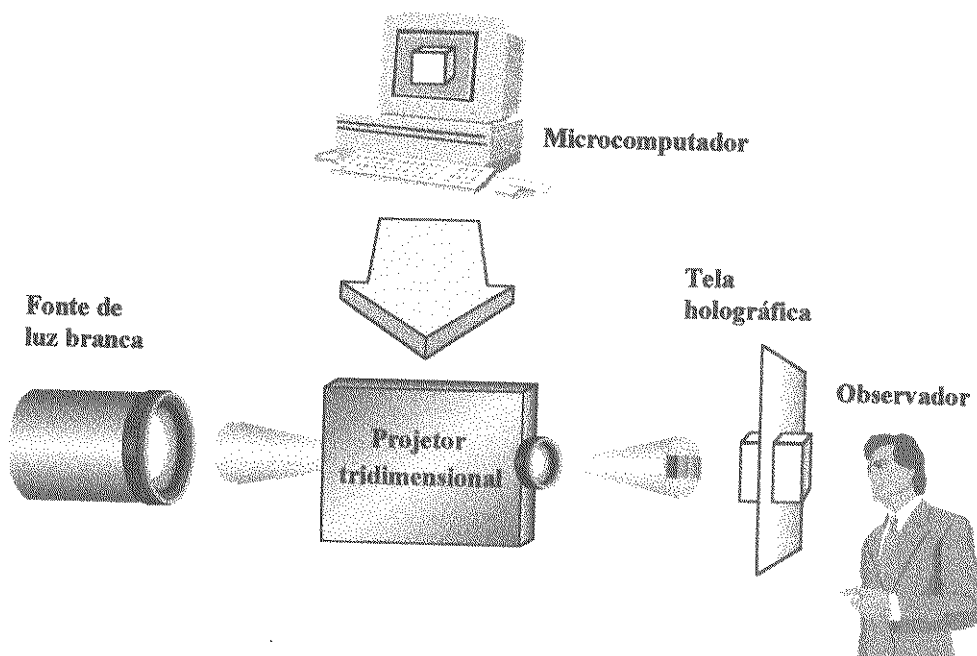


Figura 1 - Esquema do display tridimensional proposto

espacial deste ponto luminoso como a figura pretendida, graças à persistência da visão<sup>8</sup>.

O HoloProjeto pertence a categoria HOLO, de acordo com a Tabela 3. Possui a simplicidade de ativamente dos *voxels*, característica dos sistemas de varredura volumétrica, e não necessita de grandes telas móveis, assim como na holografia eletrônica. Como vantagem adicional as dimensões de seu volume de exibição dependem somente do arranjo óptico de ampliação.

<sup>8</sup> Esta característica do sistema visual também é explorada por shows de laser, televisão e cinema.

### 3. Princípios da Tela Holográfica

A tela holográfica foi desenvolvida pelo Prof. Lunazzi em 1987 e apresentada ao público em 1989 [19][20] em complemento à descoberta de que a fotografia colorida de alguns hologramas conservava a informação tridimensional. O uso da tela holográfica permitiu que fossem criadas diversas técnicas de geração de imagens tridimensionais com paralaxe (holoimagens), tais como:

- Visor Auto-Estéreo para TV [21];
- Ampliador por Conservação Direcional [22];
- Ampliador por Codificação Cromática [18];
- Cinema Holográfico [23].

A tela holográfica tem sido construída em material flexível no formato de até 0,75m x 1,14m, e é um elemento óptico difrativo obtido a partir da interferência de dois feixes luminosos monocromáticos, um divergente e outro convergente. O feixe divergente serve como feixe referência, enquanto que o feixe convergente serve como feixe objeto. A

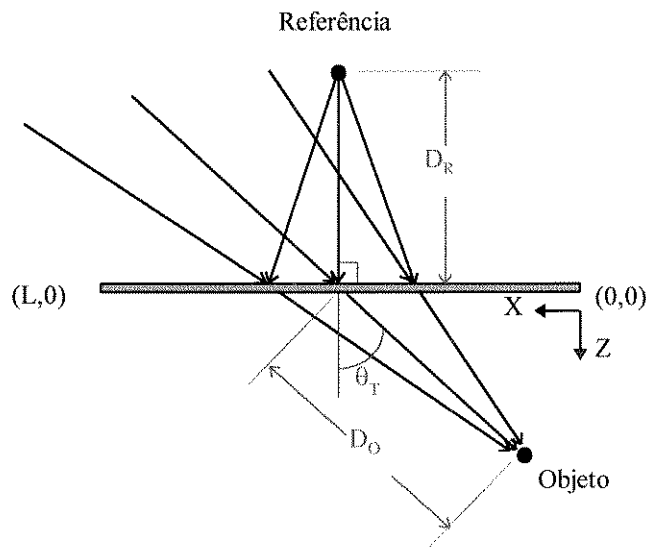


Figura 2 - Arranjo para obtenção da tela holográfica

Figura 2 ilustra o arranjo holográfico de obtenção.

Uma vez exposto aos feixes referência e objeto, o filme holográfico é revelado (de forma semelhante ao processo fotográfico) e o padrão de interferência gravado torna-se inalterável, dando origem a uma tela holográfica.

A tela holográfica pode ser caracterizada pela função  $v_{TH}(x)$  que descreve sua frequência espacial<sup>9</sup>. Esta função baseia-se no conceito da frequência espacial local [24],

<sup>9</sup> Normalmente expressa em linhas por milímetro.

segundo o qual cada pequena porção do elemento é considerada como uma rede de difração com frequência espacial dada pela equação:

$$v(x) = \frac{\sin \theta_r(x) - \sin \theta_o(x)}{\lambda_0} \quad (\text{Eq. 1})$$

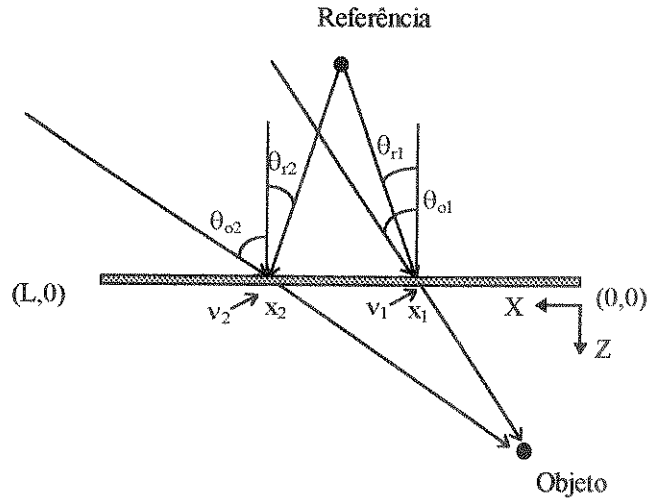
aonde

$x$  é a posição no elemento,

$v(x)$  é a frequência espacial em  $x$ ,

$\theta_r(x)$  e  $\theta_o(x)$  são os ângulos de incidência dos feixes referência e objeto em relação à normal do filme em  $x$  e

$\lambda_0$  é o comprimento de onda utilizado.



**Figura 3 - Variação da frequência espacial da tela holográfica, pela variação dos ângulos de incidência ao longo do eixo X no arranjo de obtenção**

A Figura 3 mostra a variação da frequência espacial da tela holográfica de acordo com a equação 1, pela variação dos ângulos de incidência ao longo do eixo X no arranjo de obtenção.

Partindo-se da equação 1 e dos parâmetros geométricos do arranjo de obtenção (Figura 2), a frequência espacial  $v_{TH}(x)$  da tela holográfica é dada por:

$$v_{TH}(x) = \frac{1}{\lambda_0} \left[ \frac{D-x}{\sqrt{(B^2 + (D-x)^2)}} - \frac{x-A}{\sqrt{(C^2 + (x-A)^2)}} \right] \quad (\text{Eq. 2})$$

aonde

$$A = L/2 - d_o \sin \theta_T;$$

$$B = d_R;$$

$$C = d_o \cos \theta_T;$$

$$D = L/2.$$

Como já mencionado, ao contrário de uma rede de difração, a frequência espacial  $v_{TH}(x)$  não é constante ao longo do elemento. Esta variação faz com que raios incidentes sejam difratados em ângulos diferentes<sup>10</sup>, segundo a equação da difração:

$$\sin \theta_d = M\lambda v + \sin \theta_i \quad (\text{eq. 3})$$

aonde

$\theta_d$  é o ângulo difratado,

$\theta_i$  é o ângulo de incidência,

$\lambda$  é o comprimento de onda do feixe incidente e

$M$  é a ordem de difração<sup>11</sup>.

e se interceptem a uma certa distância do elemento. Este comportamento é análogo ao de uma lente convencional refrativa [25], com a diferença de que a distância focal depende do comprimento de onda e de que o eixo óptico do objeto não é o mesmo que o eixo óptico da imagem. Esta analogia está ilustrada na Figura 4. A tela holográfica pode ser considerada então como uma lente difrativa, cujos “focos espectrais” podem ser calculados pela interseção dos raios difratados segundo as equações 2 e 3.

A Figura 5 mostra um exemplo de focalização da tela holográfica. Os raios verdes que passam por  $P_0$  são desviados para o foco  $F_{verde}$ , enquanto que os raios vermelhos que passam por um ponto próximo  $P_1$  são desviados para o foco  $F_{vermelho}$ . Os olhos direito e esquerdo de um observador foram colocados na figura para realçar que os olhos recebem raios de cores diferentes<sup>12</sup> em cada posição lateral, devido ao espalhamento angular horizontal dos focos espectrais.

---

<sup>10</sup> Dependendo da posição e do ângulo de incidência.

<sup>11</sup> Neste trabalho utiliza-se  $M=1$ , sendo que as demais ordens de difração são descartadas. Mesmo presentes, elas estariam fora da região de interesse.

<sup>12</sup> De forma similar ao que ocorre na estereoscopia com anaglifos, onde as cores observadas são filtradas pelos óculos.



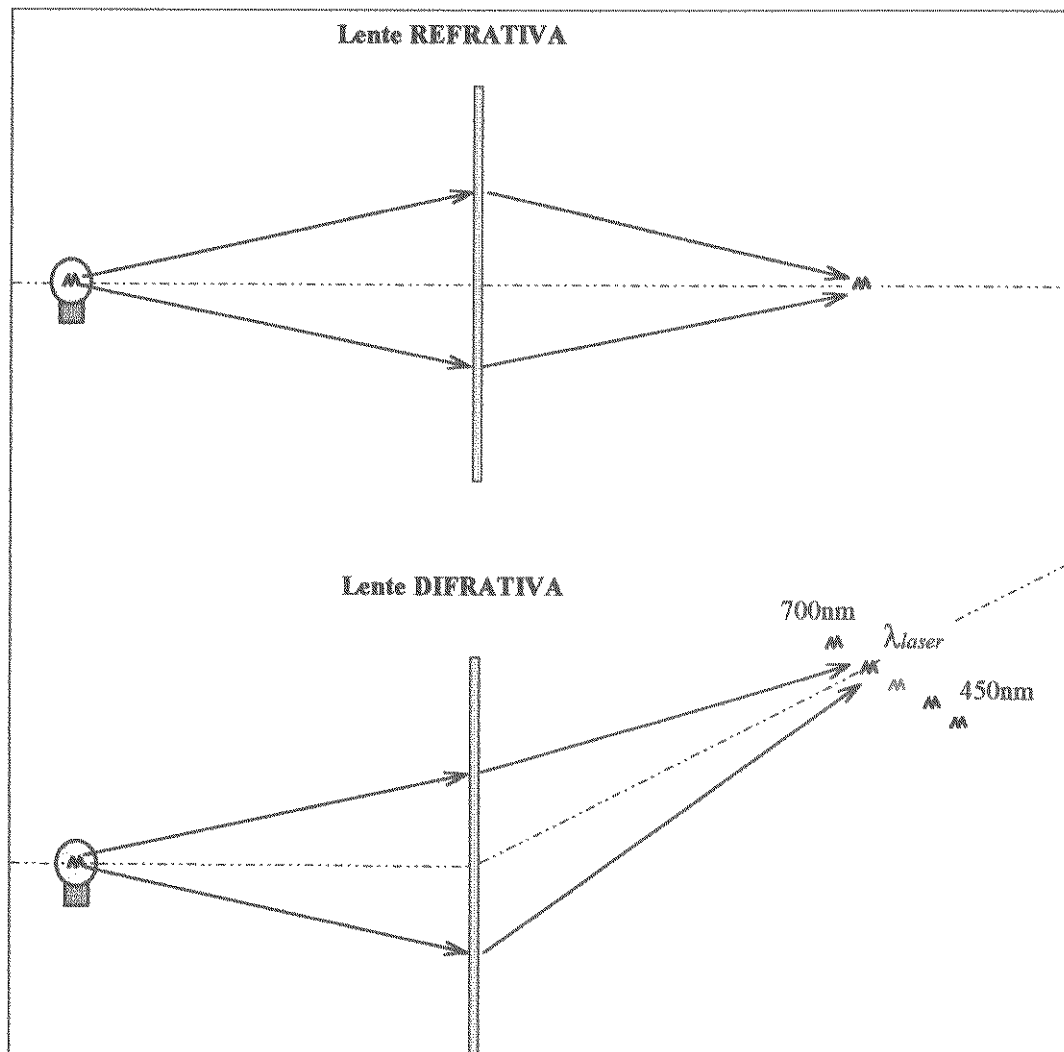


Figura 4 - Analogia gráfica entre lentes convergentes refrativa (convencional) e difrativa. Além da “quebra” do eixo óptico, a distância focal da lente difrativa depende do comprimento de onda  $\lambda$ .

O HoloProjeto emprega a tela holográfica como decodificadora espectral de profundidade, explorando seu poder de direcionamento. Cada *voxel* produzido pela tela holográfica é previamente codificado sob a forma de um borrão espectral horizontal nela projetado, cuja largura contém a informação de profundidade<sup>13</sup>. A Figura 6 ilustra as três possíveis situações desta decodificação, que são  $Z < 0$ ,  $Z = 0$  e  $Z > 0$ . Na figura manteve-se o observador fixo, correspondendo sempre a um mesmo par de cores.

<sup>13</sup> É interessante comentar que a largura entre as duas cores de uma anaglifo também é traduzida em profundidade, com a diferença fundamental de que são necessários óculos especiais e que só é possível a observação de um único ponto de vista estereoscópico.

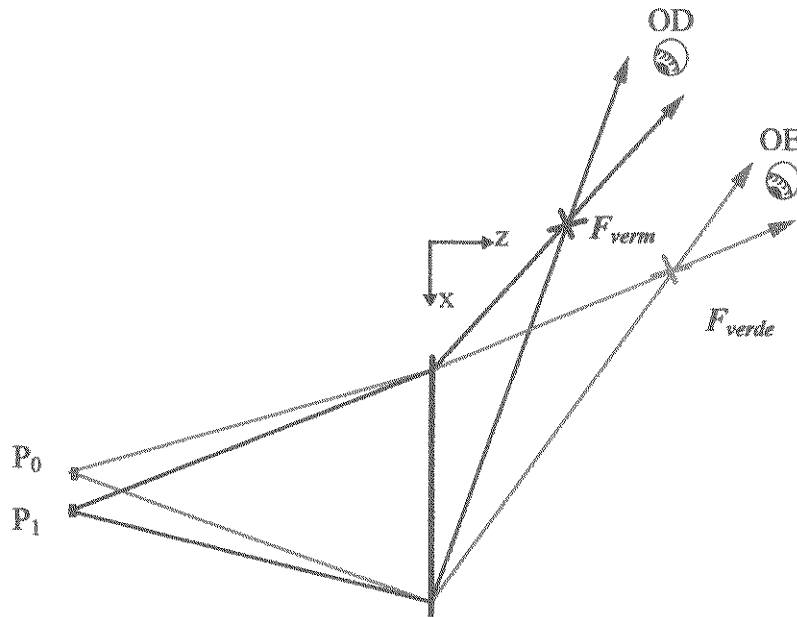


Figura 5 - Exemplo de focalização da tela holográfica. Por simplicidade, apenas dois comprimentos de onda foram representados.

Cada deslocamento lateral do observador seleciona duas novas cores para os olhos, que constituem um novo par estereoscópico (Figura 7). O resultado é a exibição do *voxel* com paralaxe horizontal, com cores que varrem o espectro de um extremo ao outro da zona de visão. Este *voxel* é de fato uma imagem, que quando real ( $Z > 0$ ) pode ser projetado num anteparo<sup>14</sup> na forma de uma linha vertical.

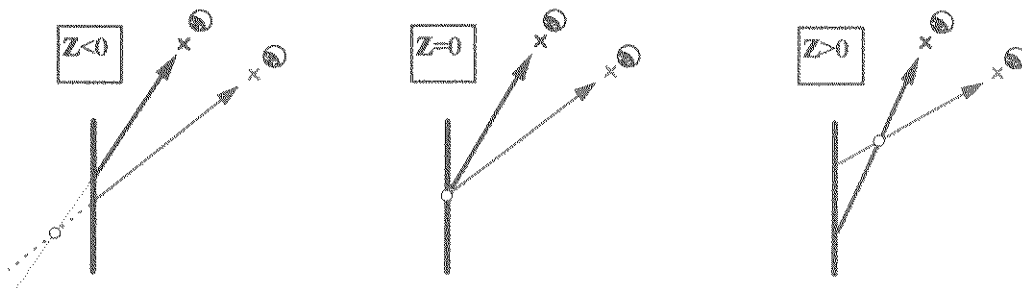
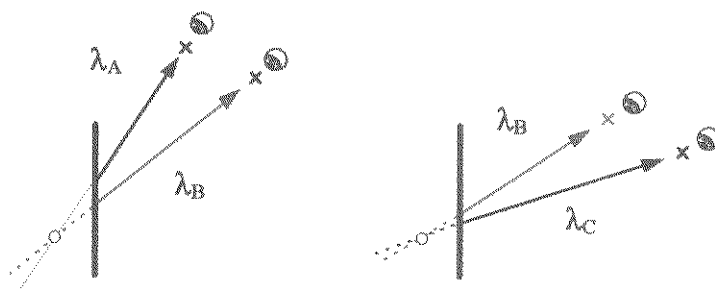


Figura 6 - Decodificação espectral de profundidade pela tela holográfica, a partir de uma posição fixa do observador. Note-se que quando  $Z$  muda de sinal ocorre a inversão do espectro projetado na tela

<sup>14</sup> Diferentemente de um par-estéreo convencional, onde o efeito de tridimensionalidade é obtido por “indução” visual. A imagem observada não está fisicamente mais à frente ou mais atrás, mas o cérebro a interpreta assim.

A tela holográfica é utilizada no HoloProjeto como o último estágio antes do observador. Por ser um elemento puramente óptico e fixo, a decodificação da profundidade e a consequente projeção holográfica dos *voxels* são instantâneas e automáticas, pois acontecem na velocidade da luz. Algumas características do sistema são diretamente influenciados pelos parâmetros da tela holográfica, como a zona angular e o



**Figura 7 - Exemplo de paralaxe da tela holográfica no caso de  $Z < 0$ , pelo deslocamento lateral do observador. O voxel fica sempre no mesmo lugar, mas as cores mudam ( $\lambda_A > \lambda_B$  e  $\lambda_B > \lambda_C$ ).**

campo de visão, a faixa do espectro observável e o brilho.

A tela holográfica ainda poderia ser utilizada como um complemento interessante aos sistemas de varredura volumétrica da categoria HOLO (discutidos no item 2.2) ampliando o tamanho de suas imagens, desde que formadas por luz branca.

## 4. Técnicas Desenvolvidas

### 4.1 *Projeção com Profundidade Fixa*

A viabilidade das idéias apresentadas no capítulo 2.3 foi inicialmente verificada através de uma experiência simples, cujo objetivo principal era gerar alguma figura a uma determinada distância fixa da tela holográfica. O resultado foi a geração de um anel luminoso observável com paralaxe horizontal contínua a frente (e posteriormente atrás) da

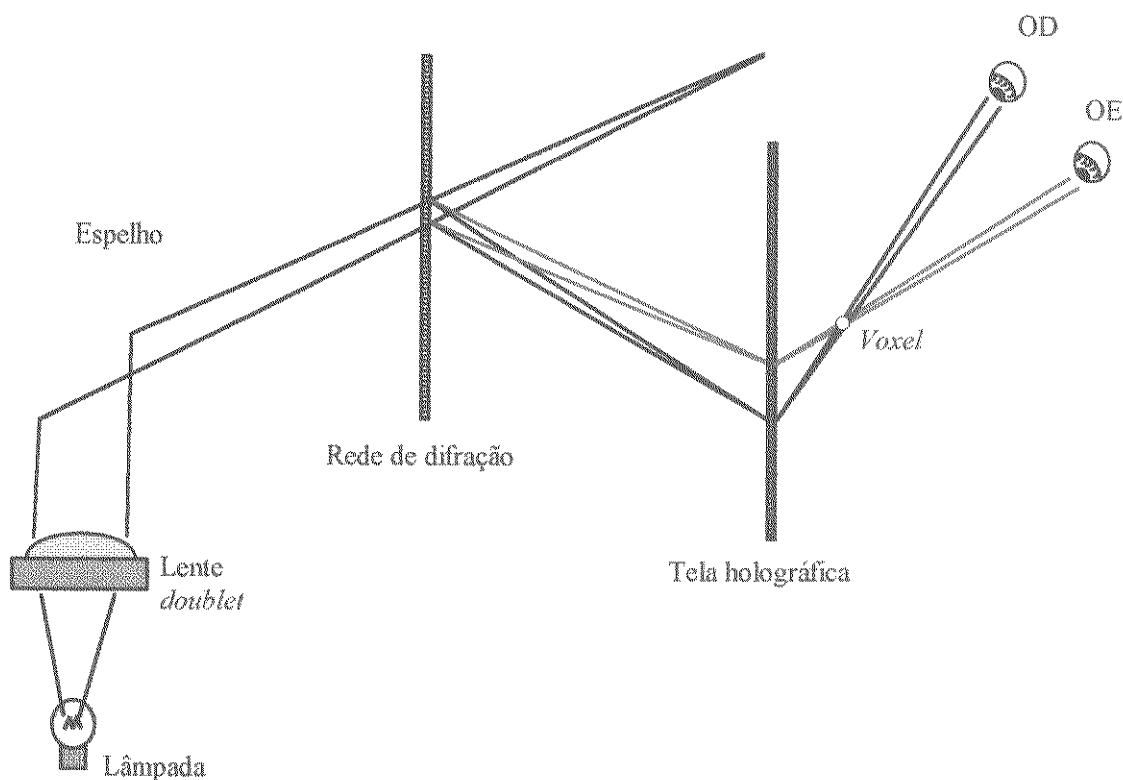


Figura 8- Arranjo para projeção com profundidade fixa.

tela holográfica [26], como descrito a seguir.

Como o sistema em questão baseia-se na dispersão cromática de um feixe de luz branca para conferir profundidade aos *voxels*, descartou-se a utilização de um emissor laser. Embora possa produzir um feixe intenso e concentrado, tal emissor é inerentemente monocromático.

Como esquematizado na Figura 8, um feixe convergente de luz branca foi produzido através de um *doublet* de 50cm de distância focal situado a frente de uma pequena lâmpada halógena de 2,8V/0,8A. Para defletir o feixe e gerar uma figura bidimensional fixou-se um pequeno espelho em forma de disco no eixo de um motor de

corrente contínua, com a normal do espelho a um ângulo de  $3^\circ$  em relação ao eixo. O círculo produzido pelo movimento giratório do espelho foi projetado numa rede de difração de 590 linhas/mm, responsável pela dispersão cromática e portanto pela profundidade dos *voxels*. Posicionou-se uma tela holográfica de  $15 \times 15 \text{ cm}^2$  (e frequência espacial média de 600 linhas/mm) a 60cm da rede e de forma a se projetar somente a ordem 1 da figura difratada, e observou-se um anel luminoso de 10cm de diâmetro “flutuando” a aproximadamente 5cm à frente da tela. Como experiência adicional deslocou-se a tela holográfica de maneira a se projetar a ordem -1, e observou-se o anel atrás da tela, uma vez que a dispersão cromática foi simétrica a anterior.

A zona angular de visão permitiu que o anel fosse observado de vários pontos de vista, como num holograma. Seu valor correspondeu ao movimento lateral do observador de 30cm, quando a uma distância de 100cm da tela.

Note-se que a rede mantém a convergência do feixe incidente [27], apenas provocando um desvio que depende do comprimento de onda e abrindo um “leque espectral” centrado na região de incidência luminosa. A abertura deste leque determina a largura do espectro projetado na tela holográfica, que finalmente a decodifica como profundidade do *voxel*, de acordo com a Figura 6.

## 4.2 *Seletor de Profundidade*

Sendo este o projeto de um display vetorial tridimensional, é preciso determinar a forma com que as coordenadas são atribuídas aos *voxels*. Na seleção de **X** e **Y** utiliza-se espelhos ortogonalmente dispostos para desviar horizontal e verticalmente o feixe de luz, similarmente a um show de laser convencional. Seria possível aplicar tal aparato na experiência descrita no item 4.1 e exibir figuras bidimensionais mais complexas à frente ou atrás da tela holográfica, mas o potencial de representação espacial com profundidade estaria sendo sub-explorado. Como já mencionado, a coordenada **Z** (profundidade) deve ser selecionada em duas etapas, uma de codificação e outra de decodificação, esta última realizada opticamente pela tela holográfica. A seguir define-se a técnica de codificação de **Z** que possibilitou a produção de figuras com volume.

### 4.2.a *Mecanismos de Codificação*

Antes que se chegasse ao mecanismo de codificação da coordenada **Z** utilizado no protótipo do sistema (capítulo 5), foram criadas e avaliadas diversas técnicas, descritas no quadro comparativo da Tabela 4.

O deslocamento longitudinal entre uma rede de difração e um objeto pontual produz imagens espectrais com diferentes dispersões cromáticas [28] [29]. Se estas imagens forem focalizadas na tela holográfica, serão produzidos *voxels* a diferentes profundidades. Este conceito é a base para as duas últimas técnicas listadas na Tabela 4, e está ilustrado na Figura 9.

<b>TÉCNICA</b>	<b>VANTAGENS</b>	<b>DESVANTAGENS</b>
Variação do ângulo de incidência numa rede de difração	<ul style="list-style-type: none"> <li>• Implementação simplificada.</li> </ul>	<ul style="list-style-type: none"> <li>• Pouca variação da dispersão</li> <li>• Grande deslocamento lateral “parasita”</li> <li>• Luminosidade não uniforme.</li> <li>• Eficiência como função do ângulo de incidência</li> </ul>
Defletor acusto-óptico “programável”	<ul style="list-style-type: none"> <li>• Eliminação de partes móveis.</li> </ul>	<ul style="list-style-type: none"> <li>• Pouca variação da dispersão</li> <li>• Alta seletividade cromática</li> <li>• Alto custo (eletrônica de acionamento)</li> </ul>
Lente difrativa [31]	<ul style="list-style-type: none"> <li>• Possível redução do deslocamento lateral parasita.</li> </ul>	<ul style="list-style-type: none"> <li>• Necessidade de alta precisão no arranjo de obtenção.</li> <li>• Dispersão fora do plano horizontal.</li> </ul>
Rede de difração com movimento longitudinal	<ul style="list-style-type: none"> <li>• Facilidade de construção.</li> </ul>	<ul style="list-style-type: none"> <li>• Introdução de movimento longitudinal com perda de resposta mecânica.</li> <li>• Necessidade de uma lente objetiva.</li> </ul>
Seleção angular com rede fixa	<ul style="list-style-type: none"> <li>• Facilidade de construção.</li> <li>• Eliminação do movimento longitudinal da rede.</li> </ul>	<ul style="list-style-type: none"> <li>• Necessidade de uma lente objetiva com grande abertura angular.</li> <li>• Introdução de <math>\Delta Z</math> ao longo de um voxel.</li> </ul>

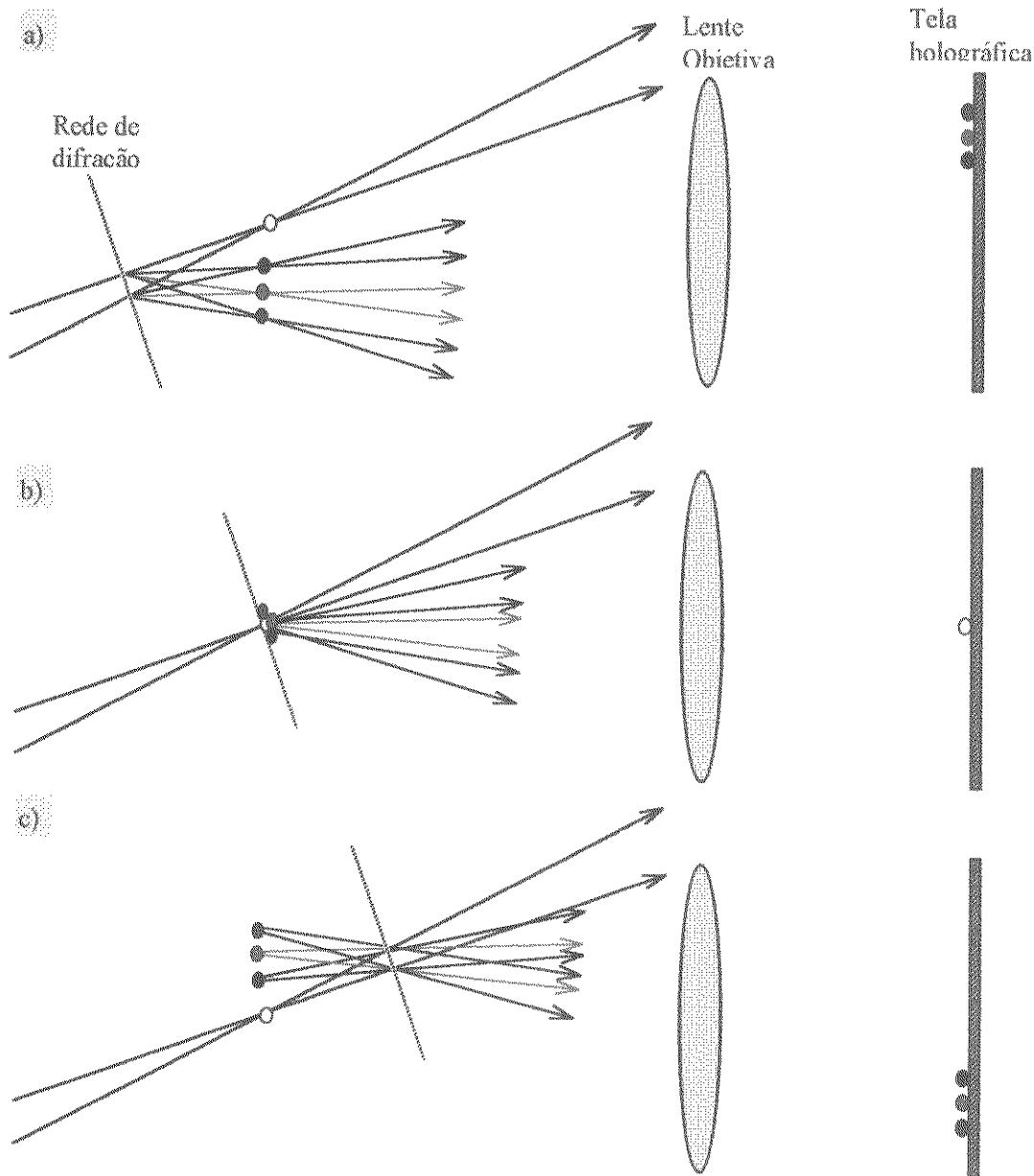
**Tabela 4- Quadro comparativo entre as técnicas avaliadas para codificação de profundidade.**

O foco de um feixe branco convergente atua como um objeto pontual apresentado à rede de difração. As posições relativas entre o “objeto”, lente objetiva e tela holográfica são as mesmas nas três situações da Figura 9. Apenas a rede se desloca com sua normal sempre alinhada com a direção do feixe incidente, estando por trás (a), coincidente (b) e à frente do “objeto” (c). A imagem espectral<sup>15</sup> [30] (representada aqui por apenas três comprimentos de onda) sofre inversão de (a) para (c) passando por largura nula em (b)<sup>16</sup>, conservando sua posição longitudinal em relação à lente objetiva. O fato de a imagem espectral permanecer no plano objeto da lente objetiva possibilita sua correta focalização na tela holográfica, e se deve à propriedade da rede de difração de reproduzir a convergência do feixe incidente nos feixes difratados [27]. Numa experiência preliminar obteve-se o deslocamento do anel luminoso numa excursão de 700mm atravessando a tela

<sup>15</sup> Entenda-se por imagem espectral aquela formada pela superposição das imagens difrativas de todas as cores do espectro.

<sup>16</sup> Esta situação sugere uma experiência didática interessante. Focalizando-se um diapositivo sobre uma rede de difração e coletando-se os raios difratados de uma das ordens por meio de uma lente, projeta-se uma imagem “branca” num anteparo situado a uma certa distância. A demonstração revela que a informação da imagem é conservada na difração, apesar da aparente desordem resultante da superposição das cores.

holográfica, movimentando-se a rede de apenas 20mm. O movimento longitudinal da rede determina a profundidade dos *voxels*, introduzindo um deslocamento lateral associado a ser compensado. Apesar deste resultado, o movimento longitudinal da rede introduz restrições quanto à velocidade de resposta mecânica do sistema, importante para

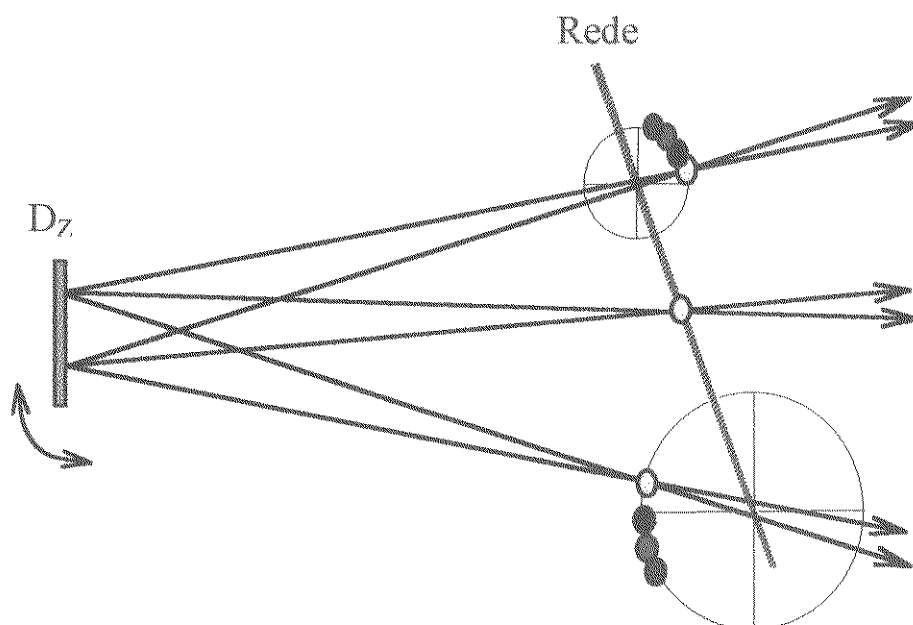


**Figura 9 - Técnica de codificação de profundidade baseada na projeção da imagem espectral de um ponto-objeto em relação ao qual desloca-se longitudinalmente uma rede de difração.**

a qualidade das figuras produzidas. Há porém uma alternativa que permite que o “objeto” atravessasse a rede sem a necessidade de movimento longitudinal, como descrito a seguir.

Como mostrado na Figura 10, um feixe convergente ao incidir num espelho que se movimenta angularmente na horizontal produzirá um foco se deslocando por uma região aproximadamente plana. Se uma rede for posicionada de forma a cruzar esta região, haverá variação da dispersão cromática, nula no ponto de interseção. Esta técnica de “seletor angular com rede fixa” corresponde à solução adotada e permitiu a substituição do movimento longitudinal da rede pelo movimento angular de um espelho, tal como é feito no controle das coordenadas X e Y. A velocidade de resposta mecânica, tão importante para a qualidade das figuras geradas, tornou-se potencialmente compatível com a de um sistema bidimensional (tipo show de laser), sendo um espelho defletor a única parte móvel necessária ao controle de Z.

A Figura 11 mostra três possíveis situações ( $z > 0$ ,  $z = 0$ ,  $z < 0$ ) do seletor de profundidade baseado na solução adotada. À esquerda da figura estão detalhes ampliados



**Figura 10 - Esquema do “seletor angular com rede fixa”, técnica de codificação adotada. A alteração da dispersão cromática é feita através de um espelho ( $D_z$ ) com movimento angular e uma rede de difração fixa e inclinada.**

dos pontos de incidência na rede.



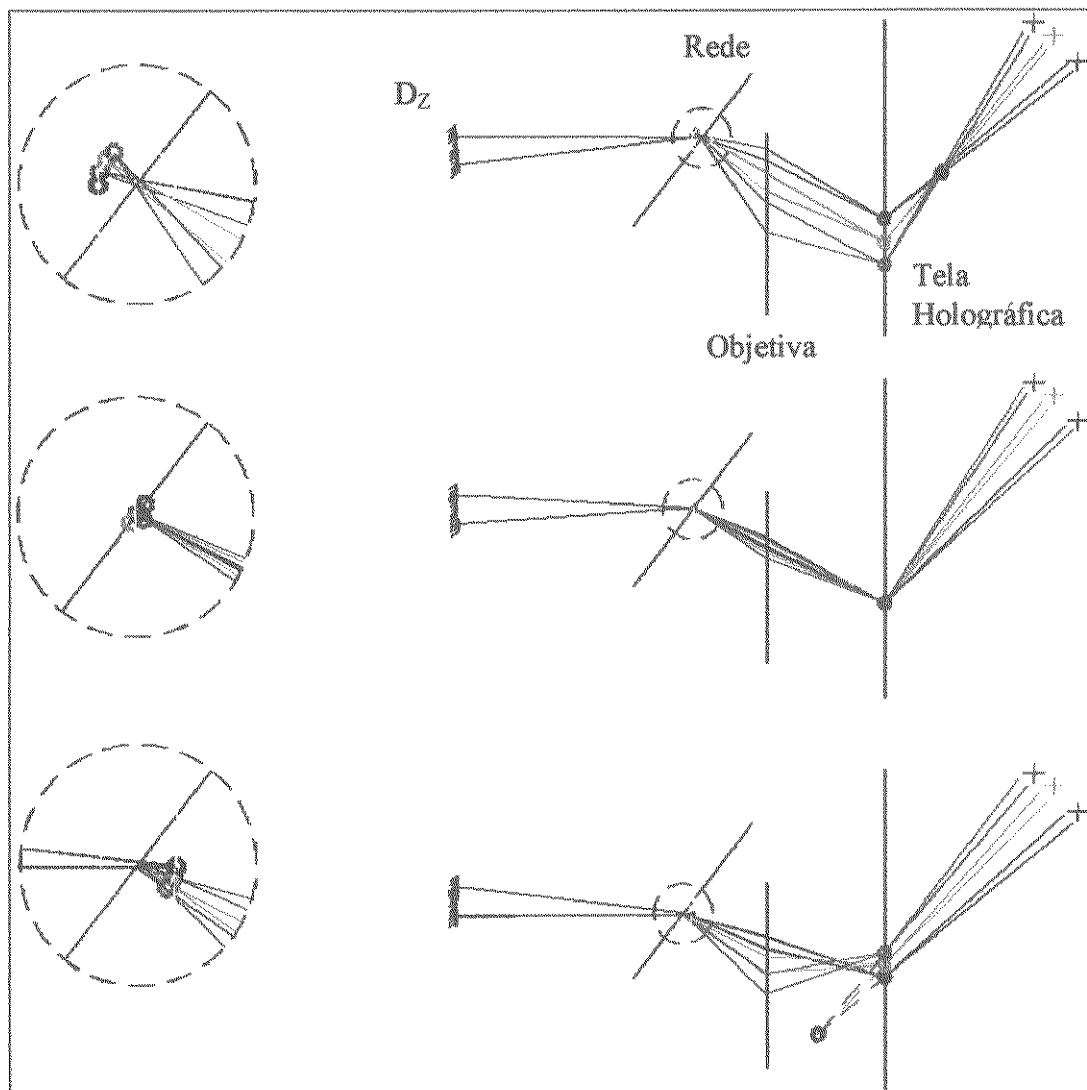


Figura 11 - Exemplo de três situações de seleção Z, a partir da solução adotada. À esquerda estão os detalhes ampliados dos pontos de incidência na rede.

#### 4.2.b Programa de Simulação

Para facilitar a implementação e melhorar a compreensão a respeito do “seletor Z angular com rede fixa” elaborou-se um programa em linguagem pascal. O programa, chamado de SIMHPROJ, visa simular o processo de codificação da coordenada Z através do modelamento difrativo e geométrico das partes envolvidas. Através de sua saída gráfica interativa ilustrada na Figura 12, é possível observar-se a trajetória da imagem espectral ampliada pela lente objetiva a medida em que o espelho  $D_z$  gira. A noção de posição deste arco espectral ampliado é importante para o posicionamento da tela holográfica, tal que a figura final seja a mais nítida possível. Vale ainda comentar que a posição de dispersão cromática nula corresponde a coordenada  $Z=0$ , ou seja, pontos da figura final exatamente sobre a tela holográfica.

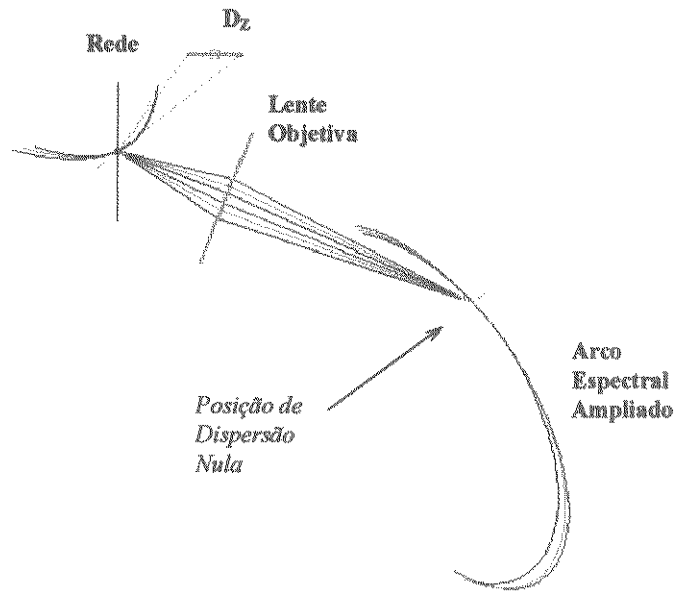


Figura 12 - Saída gráfica do programa de simulação SIMHPROJ.

Os elementos representados no modelo são a fonte de luz branca, o espelho  $D_z$ , a rede de difração (reflexiva), e a lente objetiva<sup>17</sup>. A partir de comandos do usuário para movimentar  $D_z$ , o programa atualiza o ângulo de reflexão dos raios marginais do feixe de luz branca proveniente da fonte. Estes valores são geometricamente convertidos em ângulos de incidência na rede, e então os ângulos de difração de ordem 1 (ou -1) são calculados segundo a eq. 3 (pag. 11) para os comprimentos de onda correspondentes ao vermelho, verde e azul. Na sequência, para cada uma destas cores são computadas as posições  $(z,x)$  de suas respectivas imagens difratadas, levando-se em consideração os pontos de incidência na rede. Para simular a ampliação destas imagens o programa modela a lente objetiva como uma lente delgada aplanática livre de aberração cromática, e emprega as equações:

$$A = \frac{i}{o} = -\frac{p'}{p} \quad (\text{eq. 4})$$

do aumento linear transversal, e:

$$\frac{1}{f} = \frac{1}{p} + \frac{1}{p'} \quad (\text{eq. 5})$$

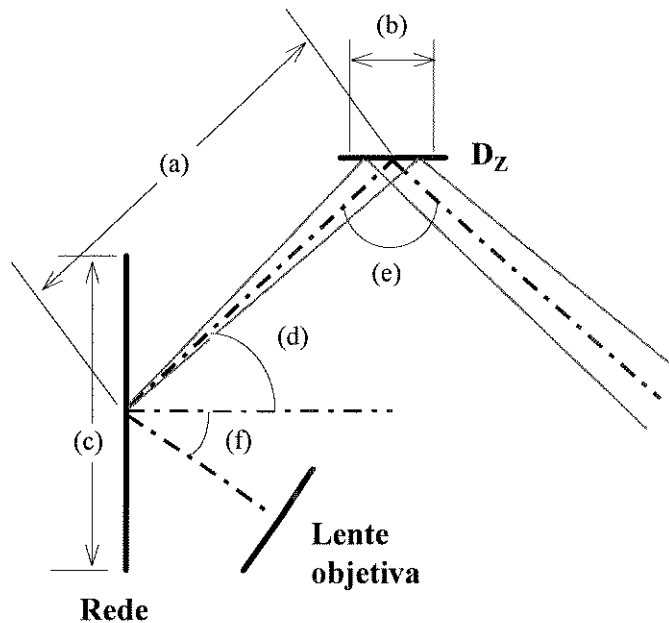
de gauss, aonde:

$f$  é a distância focal da lente;

$p$  a distância longitudinal do objeto à lente;

<sup>17</sup> O espelho  $D_y$  foi omitido por não fazer parte do seletor  $Z$ , cuja análise é bidimensional.

$p'$  a distância longitudinal da imagem à lente;  
 $o$  o tamanho do objeto (distância transversal em rel. ao eixo óptico) e  
 $i$  é o tamanho da imagem (distância transversal em rel. ao eixo óptico).



**Figura 13 - Representação gráfica de alguns dos parâmetros solicitados pelo programa de simulação SIMHPROJ.**

Antes de iniciar a simulação, o programa SIMHPROJ pede ao usuário que digite os parâmetros do sistema<sup>18</sup>, que são:

- a) Braço de rotação de  $D_Z$  (a distância em relação ao centro de  $D_Z$  a que converge o feixe de luz branca);
- b) Largura útil de  $D_Z$ ;
- c) Largura útil da rede de difração;
- d) Ângulo do braço de  $D_Z$  em relação à normal da rede na posição inicial (situação de focalização do feixe branco na rede);
- e) Ângulo de Incidência sobre  $D_Z$  (semi-ângulo entre a direção do feixe branco incidente em relação ao braço na posição inicial);
- f) Ângulo do eixo óptico da lente objetiva em relação à normal da rede;
- g) Ordem de difração considerada (1 ou -1);
- h) Frequência espacial da rede;
- i) Distância focal da lente objetiva;
- j) Distância total de ampliação (a partir da qual o programa encontra a única posição de ampliação da lente);
- k) Escala de exibição.

<sup>18</sup> No caso do usuário teclar <Enter> ao invés de responder, valores padrão (*default*) serão assumidos.

Alguns dos parâmetros solicitados pelo programa estão ilustrados na Figura 13.

A sessão de entrada encerra-se e aparece na tela o desenho em escala do seletor, com os elementos distribuídos de acordo com os parâmetros digitados, o traçado de raios da situação inicial (com  $z=0$ ) e os arcos espectrais ampliados. Estes arcos definem a trajetória das três imagens espectrais ampliadas cujos deslocamentos são provocados pela rotação de  $D_z$ . A provável posição do plano vertical da tela holográfica é definida pela tangente média dos arcos espectrais na posição de dispersão nula, para que a focalização sobre a tela seja a maior possível e haja a possibilidade de inversão de profundidade. O uso do programa tornou mais eficiente a busca de parâmetros ótimos para o sistema, como por exemplo para compatibilizar o ângulo entre a direção central dos raios vindo da lente objetiva e a provável direção da tela holográfica, com o ângulo de incidência de operação da tela.

O uso do programa mostra que a variação de dispersão cromática produz um deslocamento lateral indesejado, justamente ao longo do arco espectral. Na implementação prática faz-se necessário compensar este deslocamento através de um outro espelho. Como descrito no capítulo 5 utilizou-se no protótipo o próprio espelho  $M_X$ , que acumulou as funções de controle da coordenada  $X$  e compensação da coordenada  $Z$ .

As teclas  $\langle \uparrow \rangle$  e  $\langle \downarrow \rangle$  fazem girar  $D_z$ , e conseqüentemente produzem novo traçado de raios. Para “congelar” um feixe, ou seja, para que uma determinada configuração de traçado de raios não se apague quando do próximo movimento de  $D_z$ , o usuário conta com a tecla  $\langle C \rangle$ . Para salvar uma tela em disco tecla-se  $\langle S \rangle$ .

A listagem do programa SIMHPROJ encontra-se em anexo.

## 5. Protótipo a Motor de Passo

### 5.1 Descrição Geral

Um protótipo do sistema para visualização holográfica de figuras geradas por computador foi construído, baseado em motores de passo. A elaboração deste protótipo, denominado de “HoloProjeto versão 1.0”, permitiu que um programa de computador atuasse sobre o sistema óptico e gerasse figuras vetoriais com paralaxe contínua, comprovando a viabilidade do projeto. A principal desvantagem foi que o emprego de motores de passo reduziu a quantidade de *voxels* mostrados por segundo, em relação ao que se poderia atingir com galvanômetros. Figuras simples puderam ser exibidas a taxas de repetição suficientemente altas para a retenção visual.

O HoloProjeto é constituído basicamente de um microcomputador PC-XT, uma fonte “pontual” de luz branca, acionador (*driver*) de motores de passo, arranjo óptico e defletores de feixe. A partir de um arquivo contendo os vértices de uma figura, o microcomputador controla os defletores de feixe de forma que o arranjo óptico produza um ponto de luz que passe pelos vértices. Este ponto de luz denomina-se “holocursor” por oferecer paralaxe horizontal contínua. O processo é semelhante ao utilizado em shows de laser convencionais, com a diferença importante de que os vértices das figuras

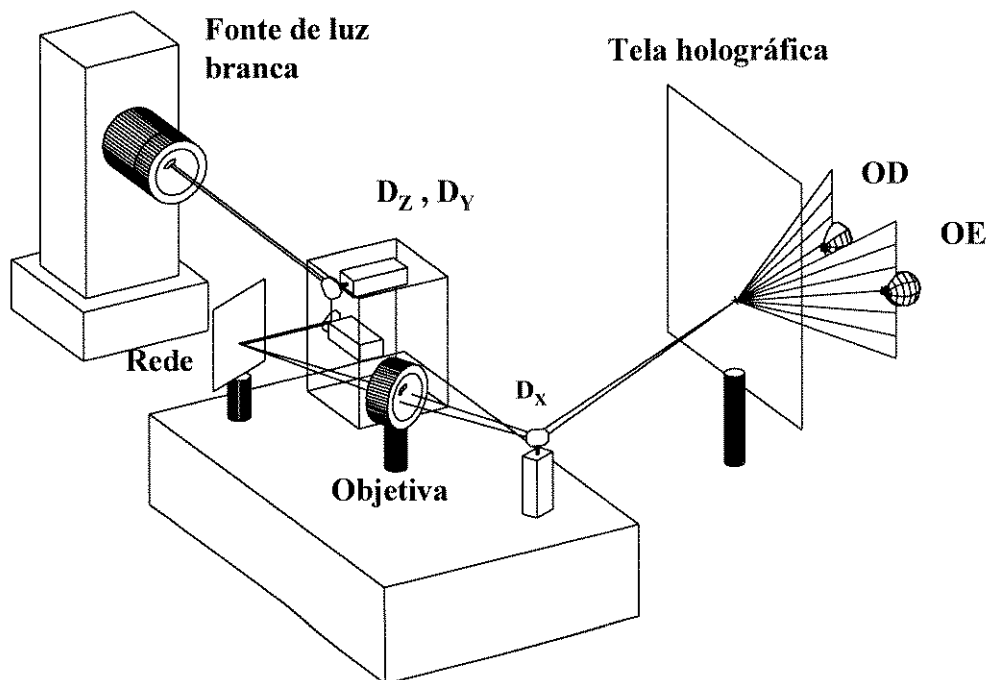
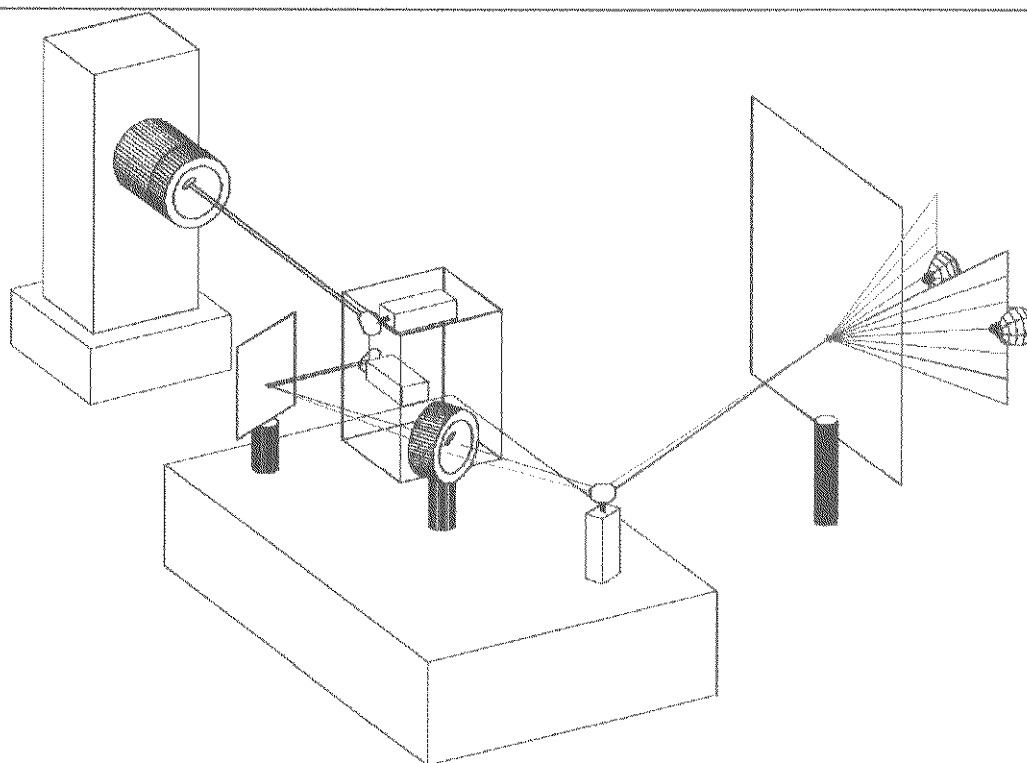
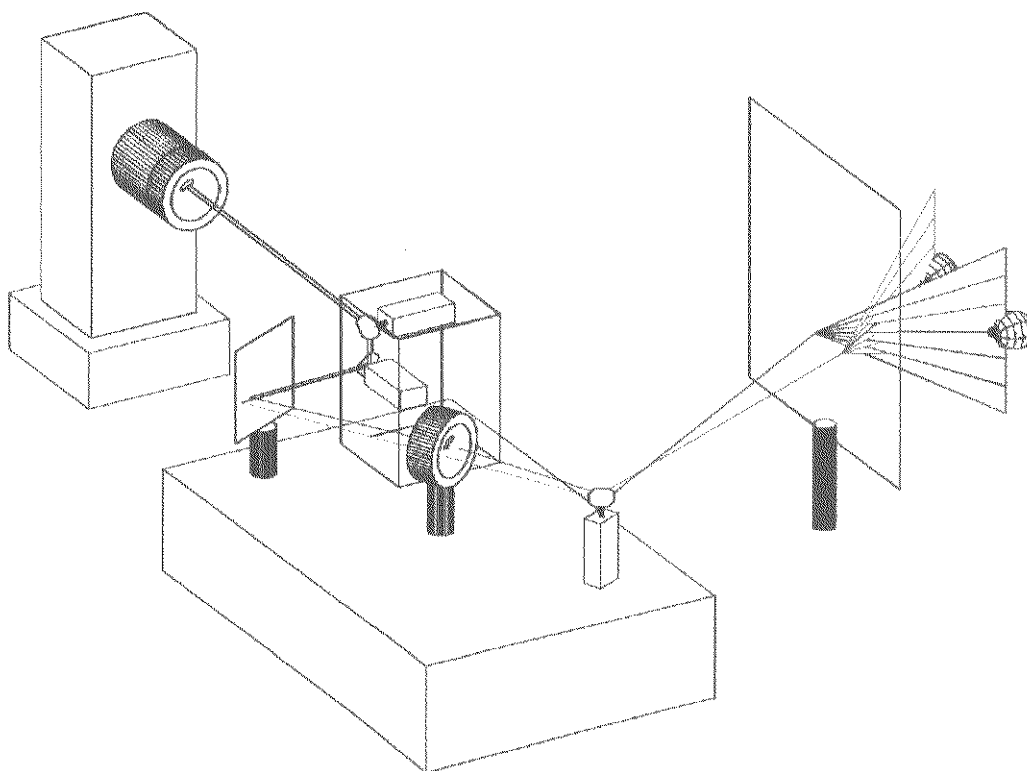


Figura 14 - Visão geral do HoloProjeto

apresentadas não estão limitados à superfície fixa de um anteparo.



(a)



(b)

**Figura 15 - Na situação (a) o feixe branco incide no centro da rede, produzindo  $Z=0$  (voxel sobre a tela). Na situação (b) a rede sofre incidência em outra posição, resultando em  $Z>0$  (voxel entre a tela e o observador).**

A Figura 14 apresenta uma visão geral do sistema, obtida a partir do projeto tridimensional feito em AutoCAD. Os olhos direito (OD) e esquerdo (OE) de um observador recebem raios de cores diferentes (por exemplo verde e laranja) que o cérebro funde numa cor intermediária durante a percepção visual.

Para efeito de esclarecimento quanto à nomenclatura, vale comentar que as figuras produzidas são formadas por *voxels* dentro de um certo volume, devido à movimentação tridimensional de um *holocursor*.

O movimento em **X** e **Y** é obtido diretamente pela movimentação dos respectivos defletores. Já o movimento em **Z** é possível graças a um processo de codificação e decodificação espectral, realizada mecanicamente por **D<sub>Z</sub>** e opticamente pela rede de difração e a tela holográfica. A Figura 15 ilustra duas possíveis situações do sistema. Em (a) forma-se um voxel sobre a tela holográfica (**Z=0**), pois o feixe branco incide no centro da rede. Já em (b) há um deslocamento lateral do feixe branco incidente tal que outra posição da rede é iluminada, produzindo-se assim um *voxel* entre a tela holográfica e o observador (**Z>0**).

## 5.2 Fonte de Luz Branca

Assim como em sistemas de show de laser, o HoloProjeto forma suas figuras pelo movimento de um ponto luminoso<sup>19</sup>. A fonte de luz utilizada no caso do HoloProjeto é “branca”, ou seja, ocupa toda a faixa visível do espectro (450 a 700nm). Esta característica se deve ao mecanismo utilizado na codificação e decodificação da coordenada **Z**, baseado na variação da dispersão cromática do feixe. A substituição do laser por luz branca representa um avanço no esforço de se eliminar a exigência de

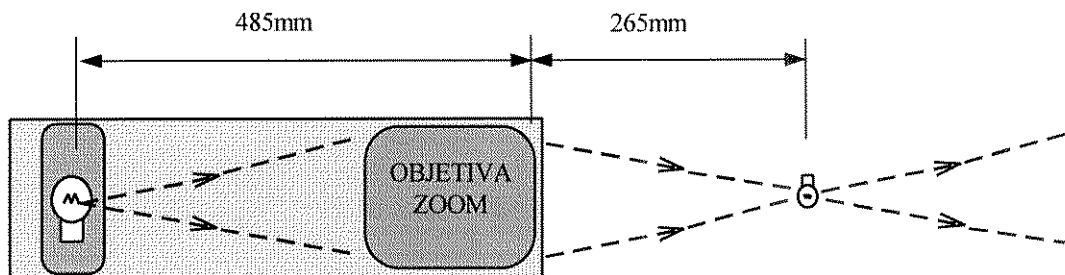


Figura 16 - Esquema da fonte de luz branca, com valores típicos de alinhamento.

monocromaticidade, comum aos processos de geração de imagens holográficas.

<sup>19</sup> A comparação se refere ao show de laser que realiza desenhos sobre um anteparo, a partir do movimento de um ponto luminoso projetado. Há também no entanto efeitos sobre o próprio feixe laser, que se torna visível ao atravessar um meio difusor (fumaça, por exemplo). Estes efeitos, embora preencham superfícies ou até volumes, estão limitados a superfícies definidas por retas que partem de um único ponto.

A fonte de luz branca, ilustrada na Figura 16, foi construída a partir de uma lâmpada halógena HaloStar OSRAM de 10W-6V, uma objetiva zoom f:4.5 com faixa focal de 80 a 205mm e um tubo de PVC para facilitar o alinhamento e evitar o vazamento de luz. Obteve-se assim um feixe convergente formando uma imagem bastante pontual (inferior a  $1\text{mm}^2$ ) do filamento da lâmpada. É importante que esta primeira imagem seja pequena, pois ainda sofre uma ampliação ao ser projetada na tela holográfica, como será descrito ainda neste capítulo.

A distância a que se forma e as dimensões da imagem do filamento são ajustáveis tanto pela posição longitudinal da lâmpada dentro do tubo quanto pela variação da distância focal da objetiva-zoom.

### 5.3 Sistema Óptico

Os elementos que compõe o sistema óptico são:

- Fonte de luz branca;
- Obturador;
- Defletores de feixe;
- Rede de difração reflexiva ( $1.300\text{ ln/mm}$ );
- Lente objetiva (6 X 7 cm, f:2.8; dist. focal de 90mm);
- Tela holográfica ( $63 \times 35\text{cm}^2$ ,  $1.200\text{ ln/mm}$  de freq. espacial média);
- Sistema visual do observador.

O caminho percorrido pela luz no HoloProjetor está ilustrado na Figura 17, partindo da fonte de luz branca até os olhos do observador. A fonte de luz branca (Figura 18) produz um feixe convergente, que após ser desviado pelos defletores  $\mathbf{D}_Y$  e  $\mathbf{D}_Z$  (Figura 19) atinge a rede de difração. Sobre a rede forma-se  $\mathbf{I}_1$ , a imagem reduzida do filamento da lâmpada da fonte. A primeira ordem da rede dá origem a  $\mathbf{I}_2$ , que é a imagem espectral de  $\mathbf{I}_1$ , enquanto que a ordem 0 é simplesmente descartada. A lente objetiva (Figura 20) é alinhada de forma a coletar os raios difratados e a projetá-los na tela holográfica sob a forma de  $\mathbf{I}_3$ , que corresponde à imagem espectral ampliada. Porém, antes de chegar à tela holográfica estes raios são desviados horizontalmente por  $\mathbf{D}_X$ . A tela holográfica então gera uma sequência de pontos de vista captados aos pares pelos olhos do observador, que finalmente observa um *voxel* com paralaxe horizontal contínua. Os *voxels* podem aparecer à frente, atrás ou na própria tela holográfica, dependendo da profundidade (coordenada  $\mathbf{Z}$ ) pretendida.

Há trechos da excursão do holocursor que devem ser invisíveis, como alguns retraços. Assim utiliza-se um obturador situado na saída da fonte de luz e movido por um motor de passo também controlado pelo acionador.

O sistema visual do observador é parte integrante do sistema óptico. A retenção visual permite a percepção de figuras aonde de fato só ocorre a sucessão de pontos luminosos. A tridimensionalidade das figuras só pode ser notada devido a visão



estereoscópica do observador, que também funde as cores dos diferentes pontos de vista em cores intermediárias.

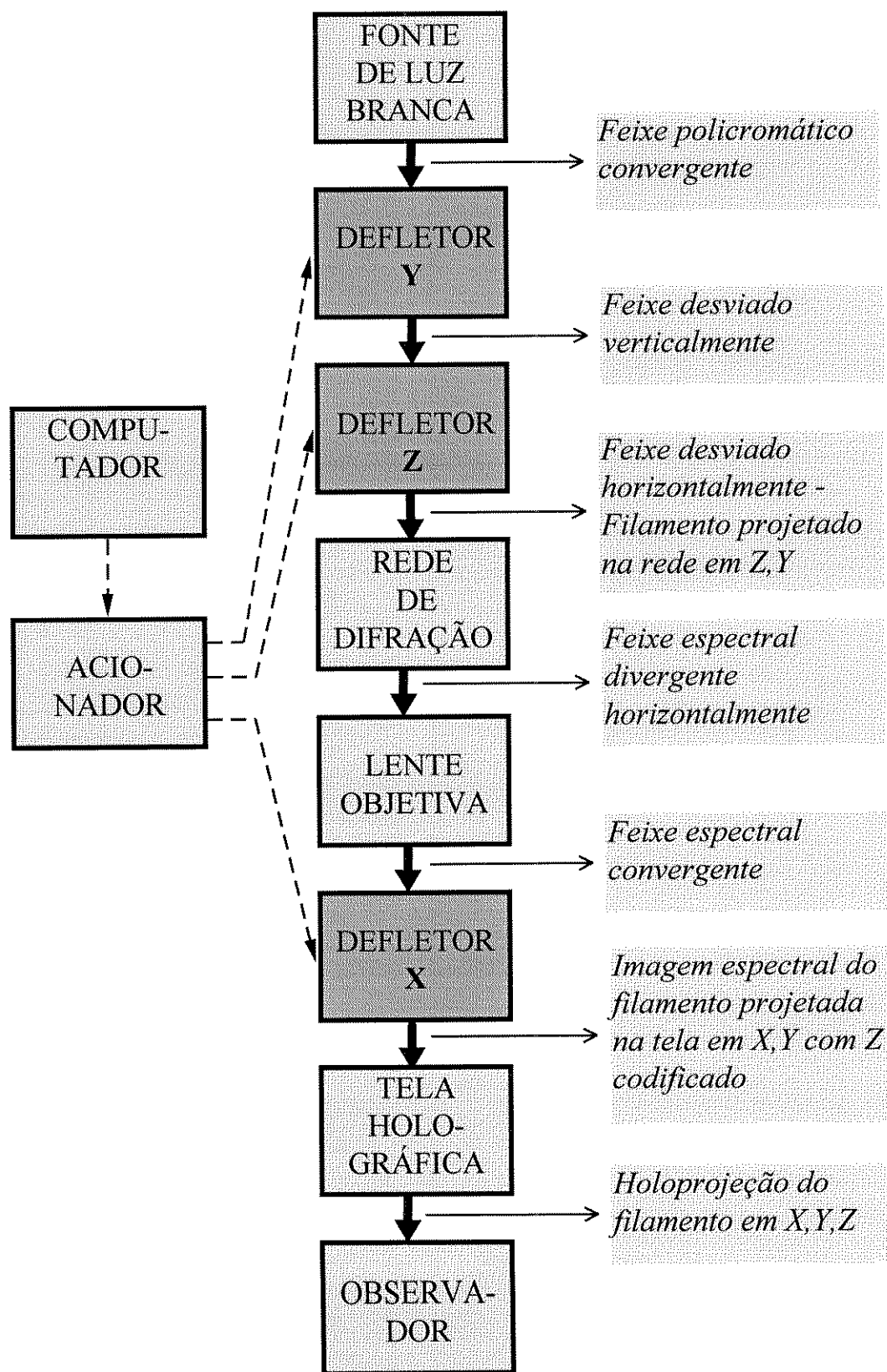
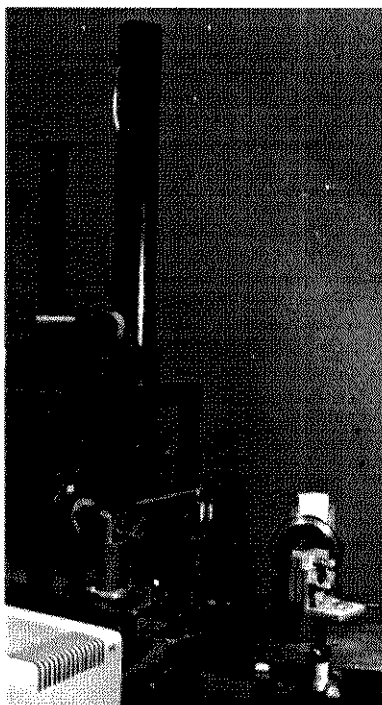
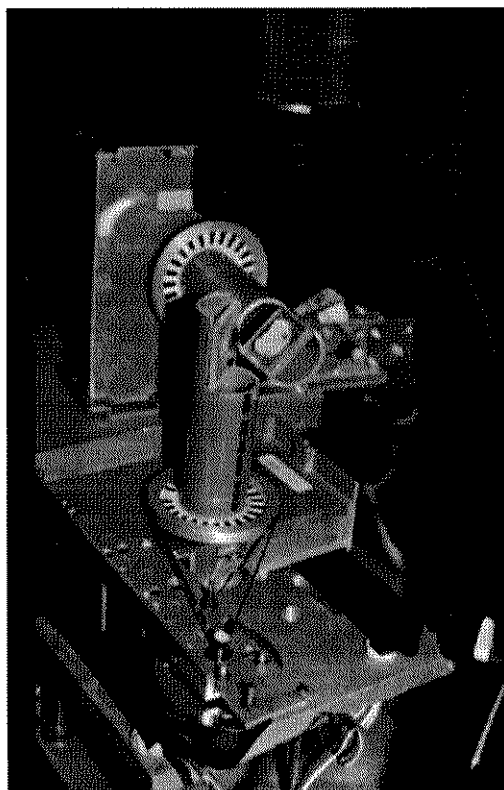


Figura 17 - Diagrama do HoloProjektor, segundo o caminho percorrido pela luz



**Figura 18 - Visão geral do arranjo óptico, sem a tela holográfica. A fonte de luz branca é o tubo vertical à esquerda da foto.**



**Figura 19 - Detalhe dos defletores  $D_y$  e  $D_z$ , situados logo abaixo da fonte de luz branca.**



Figura 20 - Detalhe do seletor de profundidade. Da esquerda para a direita: Defletor  $D_x$ , Lente objetiva e Rede de difração reflexiva.

#### 5.4 Acionador (driver)

A interface entre o microcomputador e os motores (dos defletores e do obturador) é um acionador de potência (*driver*). Este acionador foi encomendado com o objetivo de traduzir comandos digitais do micro em mudanças de tensão (chaveamento) que os motores de passo precisam para girar, de acordo com a sequência da Tabela 5.

	<i>Fase 1</i>	<i>Fase 2</i>	<i>Fase 3</i>	<i>Fase 4</i>	<i>Fase 1</i>
<b>Pino 1</b>	on	off	off	on	on
<b>Pino 2</b>	on	on	off	off	on
<b>Pino 3</b>	off	on	on	off	off
<b>Pino 4</b>	off	off	on	on	off

Tabela 5 - Sequência de chaveamento usual para motores de passo de 4 fases, executada pelo acionador a cada transição positiva do clock

As principais especificações do acionador são as seguintes:

- Interface com a porta paralela do microcomputador (evitando uma placa de saídas digitais);
- Comandos “inteligentes” compostos apenas por bits de *clock* e direção;
- Para motores de 4 fases de 12V/fase (fonte própria) ou de outras tensões;
- Controle simultâneo e independente de até 4 motores;
- Corrente total máxima fornecida de 5,5A.

Como indica o nome, o bit de direção indica o sentido de rotação do motor (sentido horário quando em 1 e anti-horário quando em 0). Para evitar “tropeços” do motor, o bit de direção só deve ser alterado quando o bit de *clock* estiver em 1.

O bit de *clock* determina a velocidade de rotação do motor, provocando um passo a cada transição positiva (0⇒1). Entenda-se por passo uma mudança de fase segundo a Tabela 5.

Como os motores são controlados simultaneamente o comando escrito na porta paralela deve ser previamente “montado” pelo programa de controle. Os 8 bits de cada comando representam a sequência  $D_4, C_4, D_3, C_3, D_2, C_2, D_1, C_1$ , aonde D e C significam direção e *clock* e os números indicam as saídas do acionador. Para que um motor ligado à saída  $n$  ande um passo, é preciso que  $C_n$  sofra uma transição positiva. Para exemplificar, o trecho de um programa em pascal para mover de um passo os motores ligados às saídas

```

procedure exemplo (var ultimo_comando: byte);
{para mover os motores 1 e 3 sem alterar as direções originais}

const
  LPT1=$3bc;
  PAUSA_ESCRITA=2;           {em ms}
  PAUSA_MECANICA=2;          {em ms}

begin
  comando:=ultimo_comando and $ee;      {1110 1110}
  port[LPT1]:=comando;
  delay(PAUSA_ESCRITA);
  comando:=comando or $11;              {0001 0001}
  {os motores andam na proxima instrução}
  port[LPT1]:=comando;
  delay(PAUSA_ESCRITA);
  delay(PAUSA_MECANICA);
  ultimo_comando:=comando;
end;

```

3 e 1 (conservando as direções) seria assim:

Durante o processo precisam ser respeitadas duas pausas diferentes. A “pausa\_escrita” visa fornecer o tempo necessário para a escrita do comando na porta paralela e para a leitura dos novos valores pelo acionador. A “pausa\_mecanica” permite

que nenhum comando seja dado enquanto os motores estiverem se acomodando na nova posição.

## 5.5 Defletores

Os defletores  $D_X$ ,  $D_Y$  e  $D_Z$  são partes móveis do sistema óptico sobre as quais atua o acionador, comandado pelo microcomputador.  $D_X$  e  $D_Z$  produzem desvios horizontais, enquanto que  $D_Y$  desvia verticalmente o feixe incidente. Cada defletor é composto de um motor de passo que, acoplado a uma redução mecânica, faz girar um espelho.

O tipo de motor de passo escolhido foi o de 4 fases e 12V/fase, por questões de disponibilidade. A resolução dos motores é de 100 passos/volta, i.e.  $3,6^\circ$  por passo. Experiências preliminares indicaram que este passo seria muito grande pois a faixa angular de trabalho de  $D_Y$  e  $D_Z$  é de  $3^\circ$  aproximadamente.

Inicialmente estipulou-se uma redução de 12:1 para todos os defletores, diminuindo-se assim o passo para  $0,3^\circ$  e permitindo uma resolução máxima de 10 passos por defletor. Esta redução mecânica foi obtida pelo uso de uma polia e uma fita de borracha passando diretamente sobre o eixo do motor, como mostra a Figura 21.

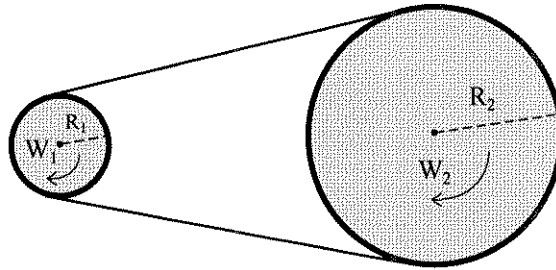


Figura 21 - Esquema de redução por polia utilizado nos defletores

A redução mecânica **RM**, que corresponde à proporção entre as velocidades angulares radiais, é calculada segundo a equação abaixo:

$$RM = \frac{R_1}{R_2} = \frac{W_2}{W_1} \quad (\text{Eq. 6})$$

A seleção da coordenada **Z** por  $D_Z$  introduz um deslocamento lateral do holocursor que precisa ser compensado por  $D_X$ . Tornou-se então interessante que os movimentos horizontais produzidos pelos dois defletores fossem aproximadamente iguais, tal que a compensação lateral pudesse ser realizada a cada passo em **Z**. A redução de  $D_X$  foi então adaptada para 6:1, dobrando-se o valor de  $R_1$ .

## 5.6 Programa de Operação, Calibração e Controle

O usuário interage com o sistema através de um programa chamado HPROJ, que reúne as funções de operação, calibração e controle.

Cada figura apresentada é recebida pelo programa através de um arquivo texto preparado pelo usuário. Uma lista de vértices absolutos define a figura, de acordo com o seguinte formato:

$X_1$	$Y_1$	$Z_1$	
$X_2$	$Y_2$	$Z_2$	<i>off</i>
$X_3$	$Y_3$	$Z_3$	
.	.	.	.
.	.	.	<i>off</i>
.	.	.	.
$X_{k-1}$	$Y_{k-1}$	$Z_{k-1}$	
$X_k$	$Y_k$	$Z_k$	

As coordenadas de cada vértice devem estar separadas por um ou mais espaços entre si e cada linha de texto só pode conter um vértice. Ao final de cada linha pode-se incluir o parâmetro *off*, que fará com que o segmento de reta definido pelo vértice anterior e o corrente não apareça pelo fechamento do obturador.

Muitas vezes as variações ( $\Delta_X$ ,  $\Delta_Y$  e  $\Delta_Z$ ) entre as coordenadas de um par de vértices são diferentes, o que demanda um tratamento especial para o deslocamento do holocursor. Para que o caminho percorrido seja aproximadamente uma reta, os valores finais de **X**, **Y** e **Z** devem ser atingidos simultaneamente. Portanto o programa normaliza as velocidades dos defletores através do cálculo dos incrementos aplicados a cada coordenada, de acordo com o seguinte procedimento:

- Atribuir a  $X_{POS}$ ,  $Y_{POS}$ ,  $Z_{POS}$  os valores do vértice anterior ( $X_{n-1}$ ,  $Y_{n-1}$ ,  $Z_{n-1}$ );
- Calcular as variações  $\Delta_X$ ,  $\Delta_Y$  e  $\Delta_Z$  entre os vértices anterior e corrente;
- Encontrar a maior variação  $\Delta_M$ ;
- Calcular os incrementos  $\delta_X$ ,  $\delta_Y$  e  $\delta_Z$ , segundo a fórmula  $\delta = \Delta / \Delta_M$ ;
- Incrementar  $\Delta_{Máx}$  vezes  $X_{POS}$ ,  $Y_{POS}$  e  $Z_{POS}$  por  $\delta_X$ ,  $\delta_Y$  e  $\delta_Z$ .

A posição angular dos defletores é determinada por ordens de avanço (ou retrocesso) de um passo de cada vez enviadas pelo acionador. Assim, embora  $X_{POS}$ ,  $Y_{POS}$  e  $Z_{POS}$  possam assumir valores fracionários, apenas o seu arredondamento é

considerado. A solução adotada corresponde a um algoritmo incremental, normalmente utilizado na implementação simplificada da rasterização<sup>20</sup> de figuras [32].

O programa HPROJ também possui a função de calibração da posição dos motores de passo. A calibração é realizada a cada vez que o acionador é ligado, pois quando isto ocorre os motores são posicionados aleatoriamente. O usuário pode avançar ou retroceder cada um dos motores através de algumas teclas, segundo a Tabela 6.

TECLA	FUNÇÃO
Seta para cima	Incrementa Y
Seta para baixo	Decrementa Y
Seta para a direita	Incrementa Z
Seta para a esquerda	Decrementa Z
[PgDn]	Incrementa X
[End]	Decrementa X
[F1]	Gira obturador no sentido horário
[F2]	Gira obturador no sentido anti-horário

**Tabela 6 - Teclas de calibração**

O procedimento de calibração está descrito no item 5.5.8.

## 5.7 Análise da Taxa de Exibição

O uso de motores de passo no lugar de galvanômetros a princípio inviabilizaria a geração de figuras holográficas a taxas de exibição suficientes para a retenção visual. Entretanto resultados preliminares revelaram que, ao menos para figuras mais simples, haveria observação direta sem muita cintilação. Isto foi possível devido à velocidade e repetibilidade dos defletores.

As figuras apresentadas pelo sistema são formadas por *voxels*, que correspondem às posições por onde passa o holocursor. Estas posições compreendem não somente os vértices, mas os pontos que os interligam. A quantidade de pontos entre cada par de vértices é dada pela máxima variação das três coordenadas ( $\Delta_M$ ).

O tempo necessário  $T_F$  para o holocursor percorrer todos os vértices da figura, em *ms*, é dado pela fórmula:

$$T_F = T_p \times \sum_{i=1}^{k-1} \Delta_{M_i} \quad (\text{Eq. 7})$$

Aonde

<sup>20</sup> Conversão de imagens em uma sequência de pontos, de acordo com a varredura de exibição.



- $T_p$  é o tempo médio em *ms* de um passo de holocursor (em qualquer direção) e
- $k$  é o número de vértices da figura.

A taxa  $R_F$  de repetição da figura em *Hz* é dada por:

$$R_F = 1000 / T_F \quad (\text{Eq. 8})$$

Para que se possa observar a figura diretamente é interessante que a taxa  $R_F$  seja pelo menos superior a 10, ou seja, que  $T_F$  seja menor que 100 *ms*. Sem obturador o valor típico de  $T_p$  é de 4 *ms*, o que resulta num número máximo de 25 *voxels* por figura.

O obturador foi introduzido com duas finalidades principais. Eliminar segmentos de reta da figura (permitindo por exemplo que a figura não começasse e terminasse na origem das coordenadas) e omitir imperfeições produzidas pelo movimento dos defletores. Sua utilização, porém, aumentou o valor típico de  $T_p$  para 50 *ms*, impossibilitando a geração de figuras observáveis diretamente. Cabe ressaltar que esta não representa uma limitação definitiva, bastando um obturador mais veloz para solucioná-la.

## 5.8 Calibração do Sistema

A calibração do sistema é dividida em duas fases: o **alinhamento do arranjo óptico** e o **ajuste eletrônico da posição dos defletores**.

O **alinhamento do arranjo óptico** só precisa ser executado na montagem do sistema ou no caso de mudanças de parâmetros. Partindo da fonte de luz branca posicionada na vertical e a uma altura arbitrária, os espelhos dos defletores  $D_Y$  e  $D_Z$  são dispostos ortogonalmente sob a fonte de maneira a captar a totalidade da seção transversal do feixe. Eles devem ser colocados o mais próximo um do outro possível (para que maiores desvios de  $D_Y$  não causem perda de luz) sem no entanto haver risco de colisão durante seu movimento.

Escolhida uma posição inicial para os defletores (correspondendo a  $Y=0$  e  $Z=0$ ) o feixe de luz de saída deve estar na horizontal. Regula-se a altura e a posição longitudinal da rede de difração de forma que a imagem do filamento proveniente da fonte seja focalizada no centro da rede. Quanto menor for esta imagem menores serão os *voxels* que formam as figuras finais, e para reduzi-la regula-se a fonte e atualiza-se a posição da rede.

A rede é girada horizontalmente até uma posição angular tal que o espectro de primeira ordem (+1 ou -1) apresente cores variando do vermelho ao azul, não ocorrendo perda por seletividade cromática devido ao aumento do ângulo de incidência [33]. Posiciona-se em seguida a lente objetiva com seu eixo óptico coincidindo com o centro da rede e na direção do feixe espectral. Neste ponto a pupila de entrada e o campo da lente objetiva [34] provocam a perda das componentes extremas do espectro, e conseqüentemente o estreitamento da zona horizontal de visualização holográfica. O

defletor  $D_x$  é posto a frente da lente objetiva numa certa posição inicial ( $X=0$ ) de forma a desviar horizontalmente o feixe espectral convergente que sai da lente objetiva para o centro da tela holográfica. Uma nova imagem do filamento, desta vez maior, é projetada na tela holográfica, incidindo a aproximadamente  $45^\circ$  (condição de funcionamento da tela utilizada). A focalização, realizada através do ajuste da distância entre a lente objetiva e a tela holográfica, tem por finalidade formar uma imagem branca. A observação da tela holográfica deve confirmar a presença de um holocursor exatamente sobre a tela.

A mudança da posição angular da rede serve para alterar o  $\Delta Z$  do volume que contém os *voxels*, mas precisa ser feita com cuidado. A rede representa o plano objeto da lente objetiva e pode prejudicar a focalização sobre a tela holográfica, que deve coincidir com o plano imagem.

O **ajuste eletrônico da posição dos defletores** deve ser realizado a cada vez que o sistema é ligado, pois o acionador inicialmente posiciona os defletores de forma aleatória. O programa HPROJ permite que ajuste-se os defletores pelo uso de algumas teclas, como descrito na Tabela 6 (pág. 35). O usuário faz girar  $D_y$  e  $D_z$  para centralizar a primeira imagem na rede de difração, e em seguida comanda  $D_x$  para centralizar a segunda imagem na tela holográfica.

## 5.9 Exemplo de Utilização

Este item mostra o procedimento para obter-se do sistema uma figura tridimensional, e ao final apresenta fotografias do resultado.

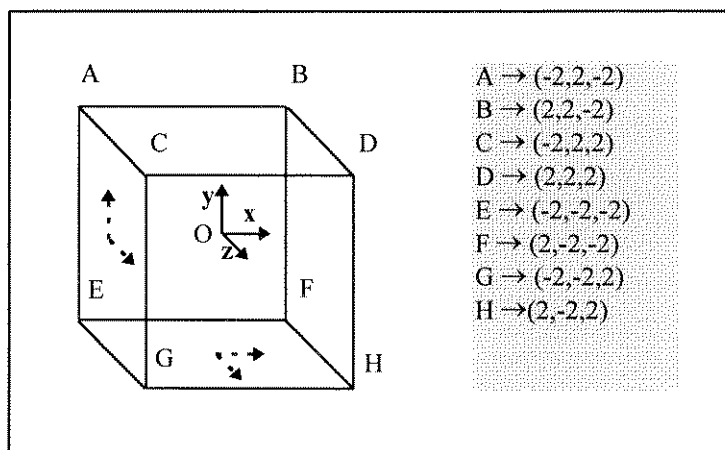


Figura 22 - Vértices de um cubo centrado na origem.

A figura escolhida foi a de um cubo  $4 \times 4 \times 4$ . Os vértices do cubo estão descritos na Figura 22.

Uma vez listados os vértices, definiu-se a trajetória do holocursor para que completasse seu percurso no menor tempo possível. Para facilitar esta otimização utilizou-se uma representação bidimensional da trajetória, como mostrado na Figura 23.

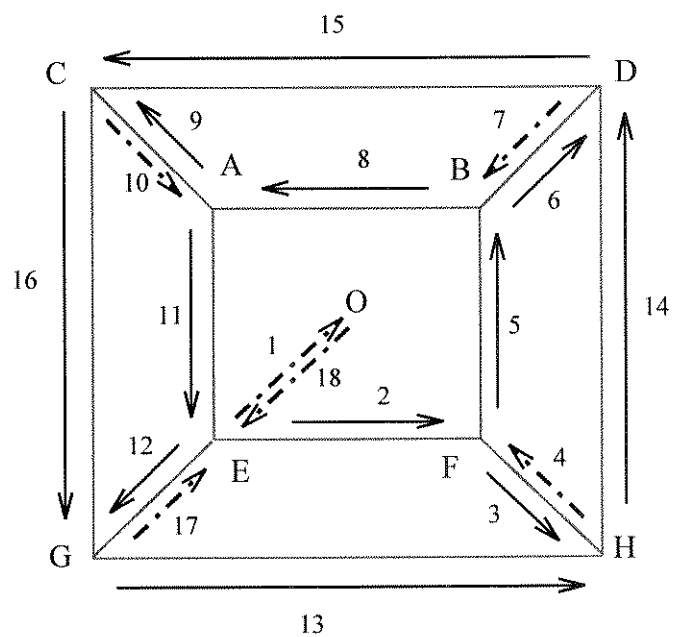


Figura 23 - Representação bidimensional da trajetória do holocursor

SEGMENTO	ORIGEM	DESTINO	APAGADO
1	O	E	✓
2	E	F	
3	F	H	
4	H	F	✓
5	F	B	
6	B	D	
7	D	B	✓
8	B	A	
9	A	C	
10	C	A	✓
11	A	E	
12	E	G	
13	G	H	
14	H	D	
15	D	C	
16	C	G	
17	G	E	✓
18	E	O	✓

Tabela 7 - Descrição vetorial da figura

As setas tracejadas indicam invisibilidade (obturador fechado), necessária para alguns segmentos de reta que não fazem parte da figura ou são repetidos. A natureza vetorial do sistema exige uma trajetória contínua começando e terminando no mesmo ponto **O**, mesmo que como neste exemplo não faça parte da figura.

Da Figura 23 extraiu-se a Tabela 7, que corresponde à descrição vetorial do cubo. Preparou-se então o arquivo texto a ser utilizado pelo programa HPROJ, segundo o formato já apresentado neste capítulo.

0	0	0	
-2	-2	-2	<i>off</i>
2	-2	-2	
2	-2	2	
2	-2	-2	<i>off</i>
2	2	-2	
2	2	2	
2	2	-2	<i>off</i>
-2	2	-2	
-2	2	2	
-2	2	-2	<i>off</i>
-2	-2	-2	
-2	-2	2	
2	-2	2	
2	2	2	
-2	2	2	
-2	-2	2	
-2	-2	-2	<i>off</i>
0	0	0	<i>off</i>

O arquivo ao lado foi digitado e gravado com o nome de “cubo.”, e o programa HPROJ executado. Para registrar o resultado foram tiradas duas fotos a partir de pontos de vista diferentes, formando assim um par estéreo (Figura 24). Como o tempo necessário para o holocursor percorrer esta figura foi da ordem de segundos, as tomadas foram de longa exposição (1,5 segundos por ponto). A cor da vista esquerda foi de um comprimento de onda maior (laranja) do que a cor da vista direita (verde), o que resultou numa coloração intermediária.

Optou-se por obter o feixe entre um vértice e outro, de forma a gerar-se uma figura formada apenas por *voxels*. Isto porque cada passo não produz uma reta, mas uma curva sinuosa devido a acomodações mecânicas dos defletores.

O cubo gerado foi de  $15 \times 15 \times 15 \text{ cm}^3$ , segundo medidas feitas por inspeção visual indireta. Neste tipo de medida um objeto luminoso (uma pequena lanterna, por ex.) é usado como referência de profundidade. Quando dois objetos encontram-se a uma mesma profundidade, sua separação lateral se mantém quando o observador muda de ponto de vista. Assim o objeto luminoso foi deslocado longitudinalmente até que sua

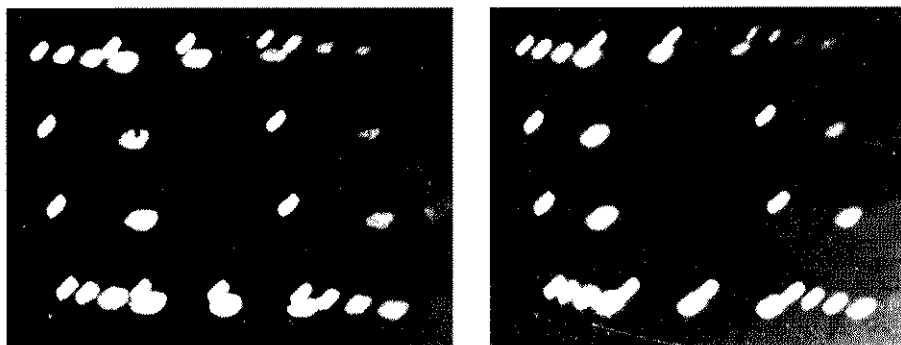


Figura 24 - Vistas esquerda e direita do resultado

profundidade coincidissem com cada *voxel* medido.

É interessante notar a diferença entre as duas fotografias, especialmente quanto à largura da face esquerda do cubo. A terceira dimensão também pode ser percebida olhando-se na direção da folha e relaxando-se a visão, até que as duas vistas se fundam.

## 5.10 Análise dos Resultados

O emprego de motores de passo tornou possível a exibição de figuras tridimensionais simples tipo “esqueleto” (*wire-frame*) a taxas de exibição de até 10 vezes por segundo, permitindo a retenção visual. A resolução atingida foi de  $6 \times 6 \times 6$  (216 voxels), ocupando um volume de trabalho de  $8.000 \text{ cm}^3$  (8 litros). A distância típica de observação foi de 1m, na qual o observador pode mover seus olhos dentro de uma zona de visão de  $9 \times 30 \text{ cm}^2$ .

Além do cubo ilustrado no item 5.5.9 testou-se o sistema com outras figuras e animações. Dentre elas destacam-se a de um ponteiro girando e cruzando a tela e um quadrado girando em torno de um eixo vertical no plano da tela.

O cubo apresentou distorção geométrica longitudinal (face anterior menor que a face posterior), que pode ser futuramente compensada pelo próprio programa. De uma forma geral os resultados corresponderam com o esperado.

O trabalho com o protótipo a motor de passo comprovou a viabilidade da proposta de um novo display tridimensional, e forneceu experiência para a construção da próxima versão, que será baseada em galvanômetros.

## 5.11 Sugestões de Implementação

A substituição dos motores de passo por galvanômetros tornará o sistema capaz de gerar figuras com a mesma qualidade do que as geradas por sistemas de show de laser convencionais. Os galvanômetros utilizados nestes sistemas possuem uma resposta em frequência muitas vezes maior que a do motor de passo, além de apresentar uma resolução de posicionamento que só depende da qualidade do conversor digital-analógico que os controla.

O uso de fontes de luz branca alternativas podem aumentar o brilho das figuras até que não haja necessidade de escuridão no ambiente de observação. A característica modular do sistema permitiria a fácil reposição da fonte atual por outra mais intensa e pontual, como uma lâmpada de arco de xenônio. Também promissora seria a utilização de um laser de corante sintonizável, com o qual se varreria rapidamente uma faixa do espectro visível na apresentação de cada *voxel*. Entretanto a fonte ideal seria o “laser branco” [35], cujo feixe é resultado da emissão de pulsos ultra-curtos (da ordem de femtosegundos) com grande alargamento espectral.

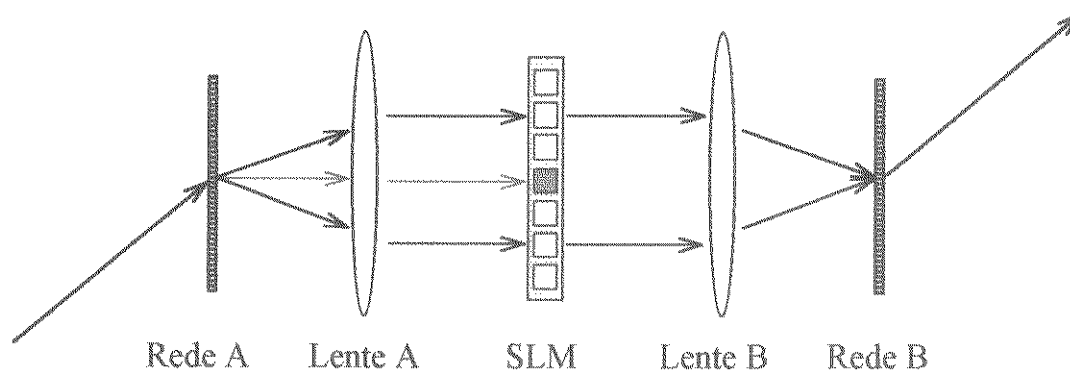
### 5.11.a Filtro Cromático de Oclusão

A oclusão, importante indicador de profundidade do grupo I, representa um obstáculo de difícil solução para os sistemas HOLO baseados em varredura volumétrica. Isto se deve ao fato de que seria necessário alterar a oclusão de acordo com a posição do observador, assim como ocorre em uma cena 3D real. Uma grande vantagem do HoloProjeto é a possibilidade de implementação da oclusão, baseada na propriedade da

onda  $\lambda$  (por se encontrar por trás de outro *voxel*). Para realizar esta oclusão, bastaria eliminar  $\lambda$  do espectro do feixe de luz no momento em que o *voxel* fosse produzido. Isto poderia ser implementado utilizando-se um modulador espacial de luz (SLM), como ilustra a Figura 25.

O dispositivo proposto deve ser acoplado logo na saída da fonte de luz branca, de modo a fornecer ao sistema um feixe devidamente subtraído dos comprimentos de onda correspondentes aos ângulos de oclusão de cada *voxel*. Os componentes espectrais do feixe de luz branca original são espalhados pela Rede A, que fica na distância focal da Lente A. Deste modo os raios saem colimados da lente e atravessam um SLM, controlado computador em sincronismo com a geração dos *voxels*. As cores não bloqueadas pelo SLM seguem para a Lente B, que as faz convergir sobre a Rede B, que tem a mesma frequência espacial que a Rede A. Assim, por simetria, a difração sobre a Rede B produz um feixe já sem as cores relativas aos ângulos de oclusão, num ângulo de saída coincidente com o de entrada.

Além da oclusão, efeitos como transparência e reflexão especular [36] ainda



**Figura 25 - Esquema do seletor de oclusão proposto. Apenas três cores foram representadas, mas participam do processo de filtragem todas as componentes espectrais do feixe original.**

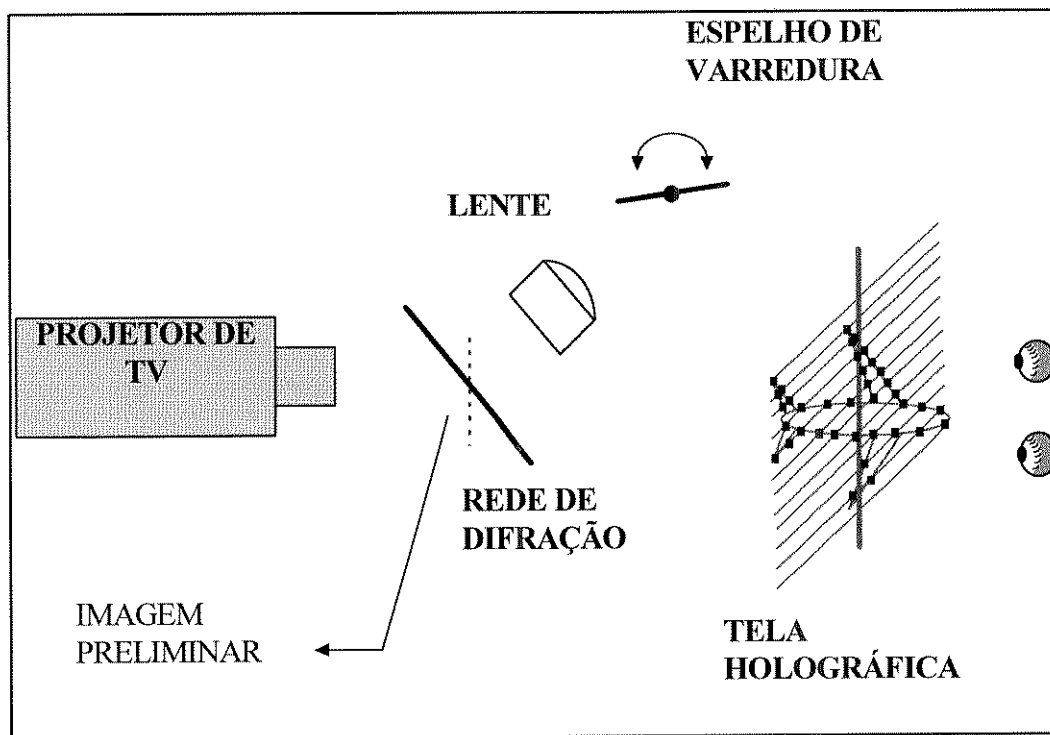
poderão ser obtidos caso o SLM permita bloqueios graduáveis.

### 5.11.b Protótipo Baseado em Projetor de TV

Experiências realizadas durante o projeto do sistema inspiraram a idéia de um HoloProjetor por varredura volumétrica, que poderia formar imagens 3D a partir de “fatias” como as produzidas em tomografia ou ressonância magnética. A holoprojeção continuaria sendo a base do processo, o que manteria a flexibilidade das dimensões do volume de exibição.

Nesta idéia, um projetor de TV de cristal líquido elimina a necessidade da seleção vetorial para X e Y. A imagem do cristal líquido é holoprojetada tal como já feito com diapositivos [37], de forma a cruzar a tela holográfica. A única parte móvel do sistema é um espelho de varredura cuja função é deslocar transversalmente esta holo-imagem formando assim um volume de exibição trapezoidal. Esta varredura pode ser facilmente implementada, bastando para isso uma estrutura poligonal espelhada fixa ao eixo de um

um espelho de varredura cuja função é deslocar transversalmente esta holo-imagem formando assim um volume de exibição trapezoidal. Esta varredura pode ser facilmente implementada, bastando para isso uma estrutura poligonal espelhada fixa ao eixo de um motor de corrente contínua girando a velocidade constante. O sincronismo entre a varredura e a exibição das “fatias” seria obtido pela fotodetecção de um feixe de laser



**Figura 26 - Proposta do HoloProjeto por varredura volumétrica. Além de imagens geradas por computador, poderá exibir imagens volumétricas multiplanares como as obtidas em tomografia.**

auxiliar incidindo sobre a estrutura espelhada.

Uma vez que a taxa de exibição (*refresh*) do sistema seria dada pela taxa do monitor dividida pelo número de fatias pretendidos para cada “holo-quadro”, tornou-se interessante comprovar a possibilidade de se exibir de forma sincronizada na tela do computador o mesmo número de quadros diferentes quanto fosse a taxa de repetição do monitor (neste caso 72Hz, somando-se os campos par e ímpar). A partir de uma configuração constituída de um 386DX, monitor *SyncMaster3 Samsung* e placa de vídeo *Trident* de 1Mb, e optando-se pelo modo texto (25 linhas x 80 colunas) construiu-se um programa em C para exibir ciclicamente quatro telas diferentes (gerando uma animação simples). Com base no *clock* do próprio computador mediu-se o intervalo de tempo necessário à exibição de 72 destas telas sincronizadamente à varredura vertical do monitor e o resultado foi de 1 segundo. Isto significa que, ao menos para o modo texto, é possível explorar ao máximo a taxa de repetição.

A experiência mostrou que já seria possível, sem alteração da configuração, construir-se um sistema capaz de exibir “holo-quadros” com 12 fatias a 6Hz, ou de 6

fatias a 12Hz. Uma possibilidade para se aumentar este produto (72) consiste do aproveitamento das outras duas telas de cristal líquido normalmente presentes num LCTV, tirando-se os respectivos filtros de cor empregados na tricromia. Cada uma das telas ficaria então com um conjunto distinto de fatias, tornando-se viável então exibir por exemplo holo-quadros com 18 fatias a 12Hz.



## 6. Conclusões

A implementação do protótipo do Holoprojetor permitiu:

- Geração de figuras volumétricas por computador, inclusive com algumas animações simples;
- Observação com paralaxe horizontal contínua e sem óculos especiais;
- Comprovação prática da viabilidade do sistema proposto;
- Discussão sobre possíveis aprimoramentos;
- Concepção de novas alternativas.

O levantamento das tecnologias de geração tridimensional de imagens possibilitou o enquadramento do Holoprojetor no contexto atual.

A análise original da Tela Holográfica holográfica, elemento óptico importante para várias experiências de projeção 3D, contribuirá para a sua divulgação e utilização em futuros projetos.

A sugestão de implementação do Holoprojetor por varredura volumétrica parece ser o melhor caminho a seguir. A curto prazo, porém, a simples substituição dos motores de passo por galvanômetros produzirá um aumento significativo da resolução e da taxa de exibição do sistema.

Este foi o primeiro passo para o desenvolvimento de uma nova tecnologia baseada na Tela Holográfica, que oferece como principais vantagens:

- Imagens exibidas a grandes dimensões;
- Baixa demanda computacional;
- Paralaxe horizontal contínua;
- Observação direta (sem óculos especiais);
- Imagens dinâmicas e atravessando a tela;
- Possibilidades de implementação de oclusão e de reflexão especular.

Cada vez mais se faz necessário o uso de uma mídia para representação espacial de informação, e a tecnologia nacional pode dar sua contribuição através da continuidade desta pesquisa.

## **Referências Bibliográficas**

- [1] - C. Wheatstone, "Contributions to the Philosophy of Vision: Part 1, On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision", *Roy. Soc. Lond. Phil. Trans.*, vol. 128, pp. 371-394 (1838).
- [2] - T. Okoshi, "Three-Dimensional Imaging Techniques", Academic Press, NY (1976).
- [3] - H. R. Schiffman, "Sensation and Perception", Wiley, NY (1990).
- [4] - T. E. Clifton III, F. Wefer, "Direct Volume Display Devices", *IEEE Computer Graphics & Applications* (July), pp. 57-65 (1993).
- [5] - Seiji Horibuchi, "Stereogram", Cadence Books, San Francisco, LA, (1994).
- [6] - T. Okoshi, "Three-Dimensional Displays", *proc. IEEE*, vol. 68, no. 5, pp. 548-564 (1980).
- [7] - G. B. K. Meacham, "Autostereoscopic Displays - Past and Future", *SPIE vol. 624 - Advances in Display Technology VI*, pp. 90-101 (1986).
- [8] - Sutherland, I. E., "The Ultimate Display", *anais do 1965 IFIP Congress*, 2, pp. 506-608 (1965).
- [9] - Sutherland, I. E., "A Head-Mounted Three Dimensional Display", *FJCC*, Thompson Books, Washington, DC, pp. 757-764 (1968).
- [10] - R. J. Collier, C. B. Buckhardt, L. H. Lin, "Optical Holography", Academic Press, London UK, pp. 1-34 (1971).
- [11] - E. Leith, J. Upatnieks, K. Haines, "Requirements for a Wavefront Reconstruction Television Facsimile System", *J. S. M. P. T. E.*, vol. 74, pp. 893-896 (1965).
- [12] - S. A. Benton, P. St.-Hilaire, M. Lucente, J. D. Sutter, W. J. Plesniak, "Real-Time Computer-Generated 3D Holograms", *SPIE vol. 1983*, pp. 536-543 (1993).
- [13] - Jeff Hecht, "Holographic Visions Prove Hard to Reach", *Laser Focus World* (setembro), pp. 57-59 (1994).
- [14] - Traub, A. C. "Stereoscopic Display Using Rapid Varifocal Mirror Oscillations", *Applied Optics* (junho), 6(6), pp. 1085-1087 (1967).
- [15] - Fuchs, H., S. M. Pizer, E. R. Heinz, S. H. Bloomberg, L. Tsai, "Design of and Image Editing with a Space-Filling Three-Dimensional Display based on a Standard Raster Graphics System", *SPIE vol. 367*, agosto, pp. 117-127 (1982).
- [16] - Buzak, T.S., "A Field Sequential, Discrete-Depth Plane, Three-Dimensional Display", *Society for Information Display*, Orlando, Florida (1985).
- [17] - E. A. Downing, L. Hesselink, R. M. Macfarlane, "Solid-State Three-Dimensional Volumetric Display", *proc. of the 14th International Display Research Conference - International Workshops on Active Matrix LCDs & Display Materials* (October), California (1994).
- [18] - J. J. Lunazzi, "Holoprojetor por Codificação Cromática", *anais do IV Simpósio Estadual de Lasers e Aplicações*, S. José dos Campos - SP, outubro, pp. 78-82 (1990).
- [19] - J. J. Lunazzi, A. A. Tagliaferri, "Holofotografia", *livro de resumos do XVII ENFMC*, Caxambú - MG, maio (1989).
- [20] - "UNICAMP dá início ao cinema holográfico", *matéria publicada no jornal FOLHA DE SÃO PAULO*, página G-6 inteira, 12 de maio (1989).

- [21] - J. J. Lunazzi, "CAD com Visualização Tridimensional por Tela Holográfica", anais do *XXIV Congresso Nac. de Informática SUCESU 91*, setembro, São Paulo - SP, pp. 423-426 (1991).
- [22] - Patente Francesa No. F. R. 89.07241/010689 (1989).
- [23] - J. J. Lunazzi, P. M. Boone, "One Step Technique for Enlarging Straddling Holographic Images by White-Light Projection onto a Diffractive Screen", *Optics Letters*, novembro (1994).
- [24] - R.J. Collier, C.B. Buckhardt, L.H. Lin, "Optical Holography", Academic Press, London UK, pp. 63-66 (1971).
- [25] - Stephen A. Benton, "The Mathematical Optics Of White Light Transmission Holograms", Anais do *International Symposium on Display Holography*, IL, pp. 5-14 (1982).
- [26] - J. J. Lunazzi, M. Diamand, J. M. Jaramillo, "HoloCAD, A New Possibility", Anais do *V Simpósio Estadual de Lasers e Aplicações*, São Paulo - SP, outubro, pp. 118-119.(1992).
- [27] - J. J. Lunazzi, M. Diamand, "3d Display System Based on Holographic Screen and Microcomputer Driven Galvanometers", aceito para publicação na revista *Applied Optics* (1994).
- [28] - J. J. Lunazzi, "Holophotography with a Diffraction Grating", *Optical Engineering*, janeiro, vol. 29-1, pp. 15-18 (1990).
- [29] - J. J. Lunazzi, J. M. Jaramillo, "Calculations on Diffracted Images of 3D Objects", anais do XVII Encontro de Física da Matéria Condensada, Caxambú, MG (1994).
- [30] - J. J. Lunazzi, J. M. J. Ocampo, "Binocular Disparity and Stereopsis Obtained under White Light by Means of a Diffraction Grating", aceito para publicação no *Journal of Modern Optics*, dezembro (1994).
- [31] - J. J. Lunazzi, M. Diamand, J. M. Jaramillo, "HoloCAD - Diffractive Imaging Analysis and Development of a Depth Selector"
- [32] - J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, "Computer Graphics - Principle and Practice", 2a ed., Editora Addison-Wesley, pp. 72-78 (1990).
- [33] - R. Petit, "Eletromagnetic Theory of Gratings", Springer-Verlag, Berlim (1980).
- [34] - F. A. Jenkins, H. E. White, "Fundamentals of Optics", 4a edição, McGraw-Hill.
- [35] - T. B. Norris, *Optics Letters* 17, 1009 (1992). Vide prospecto da COHERENT - "RegA<sup>TM</sup> 9000 High Rep-Rate Ti:Sapphire Regenerative Amplifier".
- [36] - J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, "Computer Graphics - Principle and Practice", 2a edição, Editora Addison-Wesley, pp. 728-731 (1990).
- [37] - J. J. Lunazzi, P. M. Boone, "3D Image Projection Using Large Diffraction Screens", anais do *International Symposium on Display Holography*, Lake Forest, IL - U.S.A., July, pp. 17-20 (1994).

# ANEXO I

*Trabalho aceito para publicação na revista  
APPLIED OPTICS (Optical Technology)*

# Three-dimensional display system based on a holographic screen and microcomputer-driven galvanometers

José J. Lunazzi, and M. Diamand

A system based on a 65 cm × 35 cm holographic screen is described as capable of the display of three-dimensional figures with continuous horizontal parallax. This display is accomplished through the use of a microcomputer and three galvanometric mirrors.

**Key words:** Three-dimensional display, holographic optical elements, diffractive optics, CAD, computer-generated holography.

Direct-volume display devices<sup>1</sup> need massive moving components<sup>2,3</sup> and cannot generate straddling figures because such figures cannot be shown outside of the protective housing. Real-time computer-generated holography<sup>4</sup> requires an enormous computing capability and generates small images.

The authors<sup>5,6</sup> have described the possibility of displaying, under white light, computer-generated three-dimensional figures with continuous horizontal parallax without making a hologram. The  $(x, y)$  position of each voxel (i.e., volume element) of the figures is determined as it would be in a laser show, i.e., by deflection of a light beam through two galvanometric mirrors. The  $z$  coordinate is determined in two steps: A  $z$  controller first encodes it by means of a variable chromatic dispersion applied to a white-light beam. A holographic screen then decodes it and finally places the voxel in its three-dimensional position.

In this note we describe a system that is able to generate volume figures under white light on the basis of diffraction encoding and decoding. The number of moving components can be reduced to three galvanometric mirrors. As shown in Fig. 1, lens L1 produces a small real image of a lamp filament. One then rapidly positions the image by tilting mirrors Mz

and My. This movement of the filament image occurs approximately within a plane that crosses diffraction grating DG. Object points presented to the diffraction grating are images, in fact, so permitting the  $x$  coordinate to change easily from positive to negative thus generates the final straddling images. The positions of the images diffracted by grating DI constitute a spectral sequence.<sup>7</sup> This sequence is equidistant from the illuminated region on the grating, as follows: Consider that the waves that compose the white-light beam are parabolic. When they are incident at an angle  $\theta$  and diverge at a distance  $x = z_0$  from the grating, their expression is

$$\Psi_i(x) = \exp(ikx \sin \theta) \exp(-ikx^2/2z_0), \quad (1)$$

where a unitary amplitude is assumed. The spatial frequency of the grating is  $\nu$  along the spatial coordinate  $x$ , which yields a transmittance that is expressed as

$$t(x) = 1 + a \cos k\nu x. \quad (2)$$

When the light traverses the grating, the diffracted images are obtained through multiplication of the incident beam by the term that corresponds to diffraction along the lens direction. The result is

$$\Psi_d(x) = \frac{a}{2} \exp[ik\nu x(\lambda_R - \lambda)] \exp(-ikx^2/2z_0), \quad (3)$$

which corresponds to a beam diverging from a distance  $z_0$  and chromatically spread according to the classical diffraction equation of a grating. (This reasoning is also valid for converging incident waves.) The further the image is from the grating, the more

The authors are with the Campinas State University, C.P. 6165, 13083-970 Campinas-SP, Brazil. J. J. Lunazzi is with the Physics Institute; and M. Diamand is with the Faculty of Electrical Engineering.

Received 23 June 1994; revised manuscript received 1 November 1994.

0003-6935/95/000001-03\$06.00/0.

© 1995 Optical Society of America.

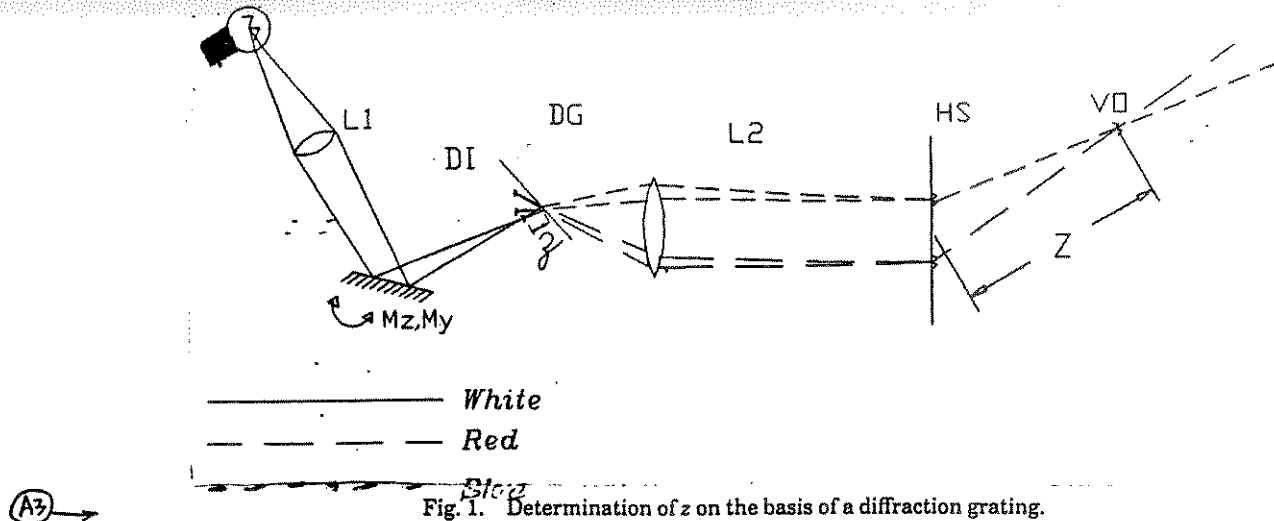


Fig. 1. Determination of  $z$  on the basis of a diffraction grating.

the spectral dispersion increases. Lens L2 projects an image of DI onto the holographic screen (HS). The holographic screen then decodes the spectral-image dispersion into the  $z$  coordinate of voxel VO.

The holographic screen is a special diffractive lens<sup>8</sup> that receives white light from a projecting lens, as described in Fig. 2. It was built in such a way that, for example, red rays passing through A also pass through A' (the focus for red wavelengths), whereas blue rays passing through nearby point B also pass through B' (the focus for blue wavelengths).

This two-point passage occurs because of  $v_s(x)$ , which is the spatial-frequency function of the holographic screen. It is described by the equation

$$v_s(x) = \frac{1}{\lambda_0} \left\{ \frac{D-x}{[B^2 + (D-x)^2]^{1/2}} - \frac{x+A}{[C^2 + (x+A)^2]^{1/2}} \right\}, \quad (4)$$

where  $A$ ,  $B$ ,  $C$ , and  $D$  are constants geometrically determined in the setup and  $\lambda_0$  is the wavelength employed during the exposure.

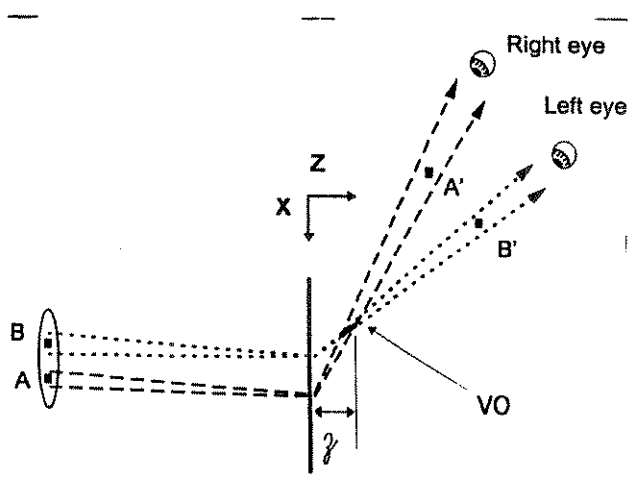


Fig. 2. Spectral decoding of depth ( $z$ ). For simplicity, only two wavelengths were drawn.

In a manner similar to the observation of a real point, the observer's eyes receive different stereopairs when looking in the vicinity of the activated voxel VO. The intersection of the beams takes place further away when the width of the projected spectrum is larger, and it is zero when the width is null. The voxel's  $z$  coordinate is thus defined from the spectrum width that impinges on HS. In this scheme the position of mirror Mz determines the  $x$  coordinate but also generates a lateral displacement that must be compensated. The compensation can be performed with an additional galvanometric mirror that is located immediately after lens L2. This mirror can also introduce the appropriate  $x$  value of the voxel and for simplicity was not introduced in our experiment.

Compared with a conventional two-dimensional display, we can say that the only additional moving part is mirror Mz. This fact allows the system to generate three-dimensional images at speeds that are compatible with conventional two-dimensional vector displays.

We conducted our experiment by imaging the filament of a halogen flash lamp through a zoom objective lens that has a 52-mm aperture, an  $f$  number of 4.5, and a focal-length range of 80–205 mm. The lens was located 485 mm from the lamp. A rotating mirror was used to produce a luminous ring. The mirror was fixed to the axis of a small motor that rotated at 50 Hz and made an angle of  $3^\circ$  with the rotation axis. This mirror was placed at a distance of 205 mm from the first lens. The ring reached a 1300-line/mm photoresist holographic diffraction grating, which was placed 265 mm from the mirror and had a diameter of approximately 10 mm. A second zoom objective lens with a 45-mm aperture, an  $f$  number of 2.8, and a 135-mm focal length was positioned at a distance of 200 mm after the grating. The optical axis of this objective was aligned to be parallel to the diffracted beam, at approximately  $45^\circ$  in relation to the normal of the grating. A holographic screen of dimensions 63 cm  $\times$  35 cm was placed 380 mm from the second objective.

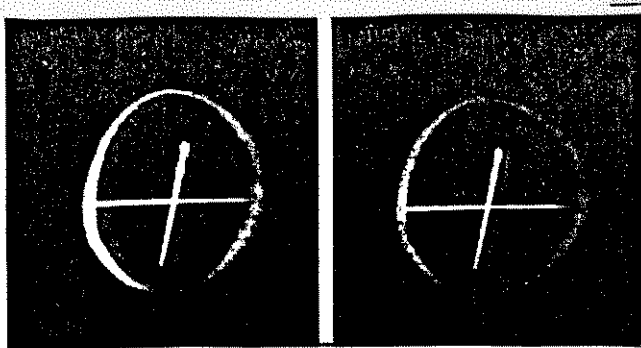


Fig. 3. Stereo photographic pair showing the generated figure.

The angular viewing zone permitted the figure to be observed sideways, as in a hologram. It extended over 22 cm laterally, at distance of 1 m from the screen. As expected, the final image was a luminous tilted ring that was suspended in air and that straddled the holographic screen, as shown in Fig. 3. This stereopair was obtained from arbitrary positions within the viewing zone that were separated laterally by 7 cm. An X-shaped luminous mark was projected onto the screen to be used as a reference for parallel-eye viewing. The result encouraged us to improve the experimental conditions, and further developments on the display of wire-frame figures in a volume of 36 cm  $\times$  65 cm  $\times$  90 cm through the use of three microcomputer-driven galvanometric mirrors will be published elsewhere.

It is possible to use a spatial light modulator instead of galvanometers to control the  $(x, y)$  coordinates. The association of this technique to  $z$  scanning will permit the generation of surfaces and solids.

We acknowledge financial contributions from the following institutions: the Fundacao de Amparo a Pesquisa do Estado de Sao Paulo, the Coordenadoria de Aperfeiçoamento de Pessoal de Nivel Superior, and the Conselho Nacional de Desenvolvimento Científico e Tecnológico. We also acknowledge L. H. Cescato, who made the diffraction grating.

#### References

1. T. E. Clifton III and F. L. Wefer, "Direct volume display devices," *IEEE Comput. Graphics Appl.* July, 57-65 (1993). (AI)
2. B. G. Blundell, A. J. Schwarz, and D. K. Horrell, "The cathode ray sphere: a prototype system to display volumetric three-dimensional images," *Opt. Eng.* 33, 180-184 (1993).
3. D. Solomon, "Volumetric imaging launches graphics into a 3-D world," *Photon. Spectra* June, 129-135 (1993). (AI)
4. P. St.-Hilaire, S. A. Benton, M. Lucente, and P. M. Hubel, "Color images with the MIT holographic video display," in *Practical Holography VI*, S. A. Benton, ed., Proc. Soc. Photo-Opt. Instrum. Eng. 1667, 73-83 (1992).
5. J. J. Lunazzi, M. Diamand, and J. M. Jaramillo, "HoloCAD, a new possibility," in *Proceedings of the Fifth Simpósio Estadual de Lasers e Aplicações* (publisher, city, country, 1992), pp. 118-119. (AI)
6. J. J. Lunazzi, M. Diamand, and J. M. Jaramillo, "HoloCAD—diffractive imaging analysis and development of a depth selector," in *Proceedings of the Sixteenth Encontro de Física da Matéria Condensada* (Publisher, city, country, 1993), pp. 98-101. (AI)
7. J. J. Lunazzi, "Holophotography with a diffraction grating," *Opt. Eng.* 29, 15-18 (1990).
8. J. J. Lunazzi, "New possibilities in the utilization of holographic screens," in *Practical Holography VI*, S. A. Benton, ed., Proc. Soc. Photo-Opt. Instrum. Eng. 1667, 289-293 (1992).

# ANEXO II

***LISTAGEM DO PROGRAMA  
“SIMHPROJ” DE SIMULAÇÃO DA  
CODIFICAÇÃO DE PROFUNDIDADE***



```
program SIMHPROJ;
```

```
{Modelamento do seletor-z, baseado num espelho Mz giratorio que redireciona}
{o foco branco que converge a partir da fonte de luz branca. O foco branco des-}
{creve um arco que cruza um rede de difracao reflexiva. Cada posicao do }
{foco branco ira corresponder a uma imagem difratada formada por uma }
{saio das cores do espectro. A posicao media destas imagens difratadas de-}
{vera formar um 'arco difrativo', em posicao semelhante ao arco branco. }
{ O importante e que este arco difrativo seja alinhado o melhor possivel em}
{relacao ao 'plano objeto' da objetiva L2, que projeta-o na tela holografica. }
```

```
uses crt, graph;
```

```
const
```

```

RADS      = 180/PI;      {Converte em graus}
GRAUS     = PI/180;      {Converte em radianos}
PASSO_INICIAL = 0.5*GRAUS;
B_        = 50;          {mm}
NL_       = 595;         {ln/mm}
ALFA_     = 45*GRAUS;    {radianos}
BETA_     = 45*GRAUS;    {radianos}
LZ_       = 20;          {mm}
LR_       = 50;          {mm}
T_        = 50;          {mm}
J_        = -1;
LAMBDA_RED      = 650e-6; {mm}
LAMBDA_BLUE    = 450e-6; {mm}
LAMBDA_GREEN    = 550e-6; {mm}
ESCALA_X        = 1;      {proporcao horizontal mm/pixel}
ESCALA_Y        = 1;      {proporcao vertical mm/pixel}
FOCOSIZE        = 2;      {Tamanho do foco branco na rede, em mm.}
GAMA_          = -45*GRAUS; {radianos}
FOCAL_         = 50;      {mm}
DIST_A         = 2000;    {mm}
COR_LUZ        = 14;
COR_ARCO       = COR_LUZ;
COR_REDE       = 13;
COR_ESPELHO    = 7;
CORLENTE       = COR_ESPELHO;
VERMELHO       = 12; {atributos para exibicao}
VERDE          = 10;
BRANCO         = 15;
AZUL           = 9;
```

```

var
b,
{ * Distancia em mm: entre o eixo de rotacao de Mz e o }
{foco do feixe branco, que coincide com o centro da }
{rede. }
ni,
{ * Frequencia espacial em ln/mm. }
alfa,
{ * Angulo de b (braco) em relacao a normal da rede. Se }
{alfa for zero o arco branco esta focalizado ao maximo }
{na rede. }
beta,
{ * Angulo do feixe branco que vem de L1 em relacao a }
{normal do espelho, na posicao em que o foco ocorre no }
{meio da rede. Tambem pode ser interpretado como o }
{semi-angulo entre o feixe branco incidente em Mz e o }
{braco que une o centro da rede ao centro de Mz. }
lz,
{ * Largura do espelho Mz - determina a base do cone de }
{luz branca. }
lr,
{ * Largura da rede de difracao }
t,
{ * Distancia do anteparo de teste. }
ro,
{ * Posicao angular do espelho Mz em rad. }
{ * Valor anterior de ro. }
passo_rot,
{ * Passo de rotacao de Mz em radianos. }
lambda,
{ * Comprimento de onda do raio difratado. }
x1,x1_ant,
x2,x2_ant,
xw,xw_ant,
zw,zw_ant,
teta1,
{ * Posicao do focos brancos, atual e anterior. }
{ * Angulos de incidencia do feixe branco na rede. }
teta2,
{ * Angulos de difracao. }
{ * Coordenadas da imagem difratada (vermelho) }
zdr_ant,xdr_ant, { * Posicao anterior da imagem difratada. }
zdg,xdg, { * Coordenadas da imagem difratada (verde) }
zdg_ant,xdg_ant, { * Posicao anterior da imagem difratada. }
zdb,xdb, { * Coordenadas da imagem difratada (azul) }
zdb_ant,xdb_ant, { * Posicao anterior da imagem difratada. }
zar,xar, { * Coordenadas da imagem ampliada (vermelho) }
zar_ant,xar_ant, { * Posicao anterior da imagem ampliada. }
zag,xag, { * Coordenadas da imagem ampliada (verde) }
zag_ant,xag_ant, { * Posicao anterior da imagem ampliada. }

```

```

zab,xab, { * Coordenadas da imagem ampliada (azul) }
zab_ant,xab_ant, { * Posicao anterior da imagem ampliada. }
xm1,zm1,
xm2,zm2, { * Coordenadas-mundo dos extremos de Mz. }
xm1_ant,zm1_ant,
xm2_ant,zm2_ant, { * Valores anteriores dos extremos de Mz. }
xm0,zm0, { * Coordenadas-mundo do centro de Mz. }
xm0_ant,zm0_ant, { * Valores anteriores do centro de Mz. }
escala, { * Escala aplicada na exibicao. }
diam_lente, { * Diametro de L2. }
gama, { * Angulo da normal de L2 em relacao a normal da rede. }
focal, { * Distancia focal de L2. }
dist_amp, { * Distancia total de ampliacao (dist_obj+dist_imag). }
dist_objm, { * Distancia objeto media, entre centro da rede e cen-
{ tro de L2. }
aumento, { * Aumento linear transversal. }
aumr,aumg,aumb, { * A.L.T de cada cor. }
aumr_ant,aumg_ant,
aumb_ant, { * A.L.T anterior de cada cor. }
zl0,xl0, { * Posicao do centro de L2. }
zl1,xl1, { * Posicao do extremo direito de L2. }
zl2,xl2, { * Posicao do extremo esquerdo de L2. }
d_o, { * Distancia entre a maior posicao no eixo optico de
{L2 por uma imagem difratada e o centro de L2. }
max,
max_r,max_g,max_b, { * Maior distancia no eixo optico e em relacao ao cen-
{tro da rede por uma imagem difratada. }
zl1r,xl1r,
zl2r,xl2r, { * Pontos de incidencia vermelha na lente. }
zl1g,xl1g,
zl2g,xl2g, { * Pontos de incidencia verde na lente. }
zl1b,xl1b,
zl2b,xl2b, { * Pontos de incidencia azul na lente. }
zl1r_ant,xl1r_ant,
zl2r_ant,xl2r_ant,
zl1g_ant,xl1g_ant,
zl2g_ant,xl2g_ant,
zl1b_ant,xl1b_ant,
zl2b_ant,xl2b_ant { * Valore anteriores. }
: real;
j, { * Ordem de difracao considerada : 1 ou -1. }
tx0, { * Origem de Z em coordenadas-tela. }
ty0, { * Origem de X em coordenadas-tela. }

```

```

{ * O meio da rede e colocado em tx0,ty0. }
tx_min,tx_max, { * Limites horizontais em coordenadas tela. }
tx_delta, { * tx_delta = tx_max - tx_min }
ty_min,ty_max, { * Limites verticais em coordenadas tela. }
ty_delta, { * ty_delta = ty_max - ty_min }
nota, { * Frequencia do som para o beep. }
duracao, { * Duracao do beep. }
code : integer; { * Variavel necessaria ao comando Val. }
resposta : string; { * Variavel de entrada usada para input com default. }
tecla : char; { * Variavel de entrada para tecladas. }
frozen,nextfrozen, { * Indica que os raios e focos correntes nao podem
{ mais ser apagados. }
erro,errodif,erroamp,
erroamp_ant, { * Variaveis auxiliares para retorno de erro }
errodif_ant,
fim_prog,
fim : boolean; { * Variavel auxiliar para termino de loops. }
arq : text;
PaletteArray: array [0..255] of record
R,G,B: byte;
end;

```

```

{[PROCEDURES E FUNCTIONS PARA MANIPULACAO DA PALETTE E GRAVACAO DA
TELA]}

```

```

{-----}
procedure SaveTGAScreen(Name:string);
{-----}

var
TGAFFile: text;
X,Y,C: integer;

begin
{Create TGA file and saves header.}
assign (TGAFFile,Name+'.TGA');
rewrite(TGAFFile);
write(TGAFFile,chr(0));
write(TGAFFile,chr(0));
write(TGAFFile,chr(2));
for X:=0 to 8 do
write(TGAFFile,chr(0));

```

```

X:=GetMaxX;
Y:=GetMaxY;
write(TGAFFile,chr(X mod 256));
write(TGAFFile,chr(X div 256));
write(TGAFFile,chr(Y mod 256));
write(TGAFFile,chr(Y div 256));
write(TGAFFile,chr(24));
write(TGAFFile,chr(32));

setcolor(BRANCO);
for Y:= 0 to GetMaxY-1 do begin
  for X:=0 to GetMaxX-1 do begin
    C:=GetPixel(X, Y);
    write(TGAFFile,chr(PaletteArray[C].B*4));
    write(TGAFFile,chr(PaletteArray[C].G*4));
    write(TGAFFile,chr(PaletteArray[C].R*4));
  end;
  rectangle(GetMaxX-1,Y,0,Y-1);
end;
rectangle(GetMaxX-1,GetMaxY-1,0,GetMaxY-1);
close(TGAFFile);
end;

{-----}
function Pal16(N:integer):integer;
{-----}

begin
  if N<8 then
    if N<>6 then
      Pal16:=N
    else
      Pal16:=20
    else if N<16 then
      Pal16:=N+48;
    end;
  end;

{-----}
procedure SetRGB16(N,R,G,B:integer);
{-----}

begin
  PaletteArray[N].R:=R;
  PaletteArray[N].G:=G;
  PaletteArray[N].B:=B;

```

```

  SetRGBPalette(Pal16(N),R,G,B);
end;

{-----}
procedure NewSetRGB(N,R,G,B:integer);
{-----}

begin
  PaletteArray[N].R:=R;
  PaletteArray[N].G:=G;
  PaletteArray[N].B:=B;
  SetRGBPalette(N,R,G,B);
end;

{||||||||||||||||||||||||||||||||||||||||}

{-----}
function tan(alfa:real):real;
{-----}

begin
  if abs(alfa)=pi/2 then
    tan:=maxlongint*(alfa/abs(alfa))
  else
    tan:=sin(alfa)/cos(alfa);
end;

{-----}
function arcsin(a:real):real;
{-----}

const
  seguranca=0.01;

begin
  if (abs(1-abs(a))<=seguranca) or (abs(a)>1) then
    arcsin:=pi/2*(a/abs(a))
  else
    begin
      arcsin:=arctan(a/sqrt(1-sqr(a)));
    end;
end;

```

```
{-----}
function tx(z:real):integer;
{-----}
```

```
begin
tx:=round(tx0+(z*escala));
end;
```

```
{-----}
function ty(x:real):integer;
{-----}
```

```
begin
ty:=round(ty0+(x*escala));
end;
```

```
{-----}
function dist(a1,b1,a2,b2 : real) : real;
{-----}
```

```
var
radicando : real;
```

```
begin
radicando:=abs((sqrt(a1-a2)+(sqrt(b1-b2))));
if radicando=0 then
dist:=0
else
dist:=sqrt(radicando);
end;
```

```
{-----}
procedure intersecao(x1,teta1,x2,teta2 : real;
var z,x : real;
var erro : boolean);
{-----}
```

```
begin
erro:=false;
if teta1=teta2 then
erro:=true
```

```
else
begin
z:=(x1-x2)/(tan(teta1)-tan(teta2));
x:=-z*tan(teta1)+x1;
end;
end;

{-----}
function dobj_amp(dist_amp,focal : real):real;
{-----}
{A partir de uma distancia total de ampliacao DIST_AMP e da distancia}
{focal FOCAL, calcula a unica possivel distancia objeto para ampliacao.}

begin
dobj_amp:=(dist_amp-sqrt(dist_amp*(dist_amp-4*focal)))/2;
end;

{-----}
function aum_maximo(dist_amp,focal : real) : real;
{-----}
{A partir de uma distancia total de ampliacao DIST_AMP e da distancia}
{focal FOCAL, calcula o unico possivel aumento maior que um.}

begin
aum_maximo:=focal/(dobj_amp(dist_amp,focal)-focal);
end;

{-----}
procedure calc_lente(dist,diametro,gama : real;
var
z10,x10,z11,x11,z12,x12 : real);
{-----}

begin
z10:=dist*cos(gama);
x10:=dist*sin(gama);
z11:=z10+(diametro/2)*cos(PI/2-gama);
x11:=x10-(diametro/2)*sin(PI/2-gama);
z12:=z10-(diametro/2)*cos(PI/2-gama);
x12:=x10+(diametro/2)*sin(PI/2-gama);
```

```

end;

{-----}
procedure plota_lente(zl0,xl0,zl1,xl1,zl2,xl2,real,cor:integer);
{-----}

var
tam_circle : integer;
begin
tam_circle:=round(escala*(dist(zl1,xl1,zl2,xl2)/30));
setcolor(cor);
SetLineStyle(SolidL,n,0,ThickWidth);
{line(tx(0),ty(0),tx(zl0),ty(xl0));}
line(tx(zl1),ty(xl1),tx(zl2),ty(xl2));
SetLineStyle(SolidL,n,0,NormWidth);
circle(tx(zl0),ty(xl0),tam_circle);
end;

{-----}
procedure calc_amp(zd,x,zl0,xl0,focal,gama : real;
var
za,xa,aumento : real);
{-----}

var
fi,
p,p_linha,
o,i,
z_aux,x_aux : real;

begin
fi:=arctan((xl0-xd)/(zl0-zd))-gama;
p:=dist(zd,x,zl0,xl0)*cos(fi);
o:=dist(zd,x,zl0,xl0)*sin(fi);
p_linha:=abs(focal*p/(p-focal));
aumento:=p_linha/p;
i:=aumento*o;
z_aux:=zl0+(p_linha*cos(gama));
x_aux:=xl0+(p_linha*sin(gama));
za:=z_aux+i*cos(PI/2-gama);
xa:=x_aux-i*sin(PI/2-gama);

```

```

{*teste*
setcolor(VERMELHO);
line (tx(z_aux),ty(x_aux),tx(za),ty(xa));
*****}
end;

{-----}
procedure plota_foco(z,x,real,cor:integer);
{-----}

begin
setcolor(cor);
rectangle(tx(z-FOCOSIZE/2),ty(x-FOCOSIZE/2),
tx(z+FOCOSIZE/2),ty(x+FOCOSIZE/2));
end;

{-----}
procedure foco_amp(z,x,aumento:real;cor:integer);
{-----}

var
tamanho : real;

begin
tamanho:=FOCOSIZE*aumento;
setcolor(cor);
rectangle(tx(z-tamanho/2),ty(x-tamanho/2),
tx(z+tamanho/2),ty(x+tamanho/2));
end;

{-----}
function dif_angle(teta,ni,lambda:real;ordem:integer):real;
{-----}

var
sin_dif : real;

begin
sin_dif:=(ordem*lambda*ni)*sin(teta);
dif_angle:=arcsin(sin_dif);

```

```

end;
{-----}
procedure calc_diff(x1,teta1,x2,teta2,ni,lambda:real;ordem:integer;
var teta1_d,teta2_d,z_d,x_d : real;
var erro : boolean);
{-----}

const
SEGURANCA=0.001;

begin
erro:=false;
teta1_d:=dif_angle(teta1,ni,lambda,ordem);
teta2_d:=dif_angle(teta2,ni,lambda,ordem);
if not(abs(teta1_d)-PI/2)<SEGURANCA and
not(abs(teta2_d)-PI/2)<SEGURANCA) then
begin
intersecao(x1,teta1_d,x2,teta2_d,z_d,x_d,erro);
z_d:=z_d;
end
else
erro:=true;
if erro then
begin
z_d:=0;
x_d:=1/2*((teta1_d)/abs(teta1_d));
end;
end;

{-----}
procedure plota_diff(x1,x2,z_d,x_d : real; cor : integer);
{-----}

begin
setcolor(cor);
plota_foco(z_d,x_d,cor);
{
if z_d>0 then
{Imagem difratada real e a direita da rede
begin

```

```

line (tx(0),ty(x1),tx(z_d),ty(x_d));
line (tx(0),ty(x2),tx(z_d),ty(x_d));

end
else
{Imagem difratada virtual e a esquerda da rede
begin
SetLineStyle(DottedLn,0,NormWidth);
line(tx(z_d),ty(x_d),tx(0),ty(x1));
line(tx(z_d),ty(x_d),tx(0),ty(x2));
SetLineStyle(SolidLn,0,NormWidth);
end;
}
end;
{-----}
procedure inicia_grafico;
{-----}

var
modo,driver : integer;

begin
DetectGraph(driver,modo);
InitGraph(driver,modo,'d:\language\tp7\bgf');
{InitGraph(driver,modo,'c:\linguas\pascal');}
setcolor(15);
setwriteMode(XORput);

{Define os limites da janela e a posicao da origem das coordenadas-mundo}
tx_min:=0;
tx_max:=GetMaxX;
tx_delta:=tx_max-tx_min;
ty_min:=0;
ty_max:=GetMaxY;
ty_delta:=ty_max-ty_min;
tx0:=round((tx_min+tx_delta)/4);
ty0:=round((ty_min+ty_delta)/2);

{Desenha a moldura da janela}
line(tx_min,ty_min,tx_max,ty_min);
line(tx_min,ty_max,tx_max,ty_max);
line(tx_min,ty_min,tx_min,ty_max);

```

```

line(tx_max,ty_min,tx_max,ty_max);
moveto(tx_min,ty_min);
end;
{-----}
procedure plota_rede(tamanho:real;cor:integer);
{-----}

begin
setcolor(cor);
line(tx0,ty(tamanho/2),tx0,ty(-tamanho/2));
end;
{-----}
procedure calc_espelho(braco,alfa,tamanho,ro:real;
var zm0,xm0,zm1,xm1,zm2,xm2:real);
{-----}

begin
xm0:=braco*sin(alfa);
zm0:=+braco*cos(alfa);
xm1:=xm0-(tamanho/2)*(sin(pi/2-alfa-ro));
zm1:=zm0-(tamanho/2)*(cos(pi/2-alfa-ro));
xm2:=xm0+(tamanho/2)*(sin(pi/2-alfa-ro));
zm2:=zm0+(tamanho/2)*(cos(pi/2-alfa-ro));
end;
{-----}
procedure plota_espelho(zm0,xm0,zm1,xm1,zm2,xm2:real;cor:integer);
{-----}

var
tam_circle:integer;

begin
tam_circle:=round(escala*(dist(zm1,xm1,zm2,xm2)/10));
setcolor(cor);
SetLineStyle(SolidLn,0,2);

```

```

line(tx(zm1),ty(xm1),tx(zm2),ty(xm2));
SetLineStyle(SolidLn,0,NormWidth);
circle(tx(zm0),ty(xm0),tam_circle);
end;
{-----}
procedure plota_feixe(b,beta,zm,xm:real;cor:integer);
{-----}
{Serve para mostrar o raio central do feixe convergente vindo de L1.}
{para verificar se a reflexao em Mz esta correta.}
begin
Setcolor(cor);
line(tx(zm0),ty(xm0),tx((b*sin(2*beta))/abs(sin(P1-alfa-2*beta))),ty(0));
end;
{-----}
procedure beep(frequencia,duracao:integer);
{-----}

begin
sound(frequencia);
delay(duracao);
nosound;
end;
{-----}
procedure gira(var ro:real; delta:real);
{-----}

begin
ro:=ro+delta;
if ro=beta then
beep(600,15);
end;
{-----}
procedure calc_incidencia(b,alfa,beta,ro,lz,zm,xm,zm1,xm1,zm2,xm2:real;
var zw,xw,teta1,teta2,x1,x2:real;
var caiu_fora:boolean);
{-----}

```

```

begin
{Posicao do foco branco 'w'}
zw:=zm-b*cos(2*beta-alfa-2*ro);
xw:=xm-b*sin(2*beta-alfa-2*ro);

{Angulos de incidencia}
teta1:=arctan((xw-xm1)/(zm1-zw));
teta2:=arctan((xw-xm2)/(zm2-zw));

{Pontos de incidencia na rede}
x1:=xm1+zm1*tan(teta1);
x2:=xm2+zm2*tan(teta2);

{Verificando se o feixe esta incidindo todo na rede}
if (abs(x1)>(lr/2)) or (abs(x2)>(lr/2)) then
caiu_fora:=true
else
caiu_fora:=false;
end;

{-----}
procedure plota_filamento(z,x,alfa,ro:real);
{-----}

const
METADE = FOCOSIZE/2;
QUARTO = FOCOSIZE/4;

begin
line(tx(z-METADE),ty(x),tx(z-QUARTO),ty(x-METADE));
line(tx(z-QUARTO),ty(x-METADE),tx(z),ty(x));
line(tx(z),ty(x),tx(z+QUARTO),ty(x-METADE));
line(tx(z+QUARTO),ty(x-METADE),tx(z+METADE),ty(x));
end;

{-----}
procedure plota_incidencia(b,alfa,tamanho,zw,xw,x1,x2,zm0,xm0,
zm1,xm1,zm2,xm2:real;cor:integer);
{-----}
{

```

---

```

var
xm0,zm0:real;      {Posicao do centro de rotacao do espelho}
}

begin
setcolor(cor);
plota_foco(-zw,xw,COR_LUZ);

{Linha do centro do espelho ate o foco branco, para dar iseia de angulo}
{line(tx(zw),ty(xw),tx(zm0),ty(xm0));;}

if zw>0 then
{Foco antes da rede}
begin
{raios ate o foco}

line(tx(zm1),ty(xm1),tx(zw),ty(xw));
line(tx(zm2),ty(xm2),tx(zw),ty(xw));

{raios do foco ate a rede}
line(tx(zw),ty(xw),tx(0),ty(x1));
line(tx(zw),ty(xw),tx(0),ty(x2));

{raios do foco refletido ate a rede}
SetLineStyle(DottedLn,0,NormWidth);
line(tx(-zw),ty(xw),tx(0),ty(x1));
line(tx(-zw),ty(xw),tx(0),ty(x2));
SetLineStyle(SolidLn,0,NormWidth);

end
else
{Foco depois da rede}
begin
{raios ate a rede}

line(tx(zm1),ty(xm1),tx(0),ty(x1));
line(tx(zm2),ty(xm2),tx(0),ty(x2));

{raios virtuais da rede ate o foco}
SetLineStyle(DottedLn,0,NormWidth);
line(tx(0),ty(x1),tx(zw),ty(xw));
line(tx(0),ty(x2),tx(zw),ty(xw));
SetLineStyle(SolidLn,0,NormWidth);

```



```

{raios para mostrar a posicao do foco refletido}
line(tx(0),ty(x1),tx(-zw),ty(xw));
line(tx(0),ty(x2),tx(-zw),ty(xw));

end;
end;

{-----}
procedure arco_branco(b,alfa,beta,lz,zm0,xm0,passo_rot,cor:integer);
{-----}

var
    passo,
    ro,
    zw,
    xw,
    zm1,xm1,
    zm2,xm2,
    tet1,
    tet2,
    x1,
    x2 : real;
    erro : boolean;

begin
    setwritemode(NormalPut);
    setcolor(cor);
    passo:=passo_rot;
    ro:=beta;
    calc_espeelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
    calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
        zw,xw,tet1,tet2,x1,x2,erro);
    moveto(tx(-zw),ty(xw));
    erro:=false;
    while not erro do
        begin
            calc_espeelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
            calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
                zw,xw,tet1,tet2,x1,x2,erro);
            if not erro then
                begin

```

```

            lineto(tx(-zw),ty(xw));
            ro:=ro+passo;
        end;
    end;
    ro:=beta;
    calc_espeelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
    calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
        zw,xw,tet1,tet2,x1,x2,erro);
    moveto(tx(-zw),ty(xw));
    erro:=false;
    while not erro do
        begin
            calc_espeelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
            calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
                zw,xw,tet1,tet2,x1,x2,erro);
            if not erro then
                begin
                    lineto(tx(-zw),ty(xw));
                    ro:=ro+passo;
                end;
            end;
            setwritemode(XORPut);
        end;
    {-----}
    procedure arco_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,lambda:real;
        j,cor:integer);
    {-----}

    var
        passo,
        ro,
        zw,
        xw,
        zm1,xm1,
        zm2,xm2,
        tet1,
        tet2,
        tet1_d,
        tet2_d,
        x1,
        x2,

```

```

zd,      : real;
xd      : real;
erro    : boolean;

begin
    setwritemode(NormalPut);
    setcolor(cor);
    passo:=passo_rot/5;
    ro:=beta;

    calc_espelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
    calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
        zw,xw,teta1,teta2,x1,x2,erro);
    calc_dif(x1,teta1,x2,teta2,ni,lambdaj,
        teta1_d,teta2_d,zd,xd,errodif);
    moveto(tx(zd),ty(xd));

    erro:=false;
    while not erro do
        begin
            calc_espelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
            calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
                zw,xw,teta1,teta2,x1,x2,erro);
            calc_dif(x1,teta1,x2,teta2,ni,lambdaj,
                teta1_d,teta2_d,zd,xd,errodif);
            moveto(tx(zd),ty(xd));

            erro:=false;
            while not erro do

```

```

begin
    calc_espelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
    calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
        zw,xw,teta1,teta2,x1,x2,erro);
    if not erro then
        begin
            calc_dif(x1,teta1,x2,teta2,ni,lambdaj,
                teta1_d,teta2_d,zd,xd,errodif);
            lineto(tx(zd),ty(xd));
            ro:=ro+passo;
        end;
    end;

    setwritemode(XORPut);
end;

{-----}
procedure arco_ampliado(zl0,xl0,focal,gama,b,alfa,beta,lz,zm0,xm0,
    passo_rot,lambdaj,real;
    j,cor:integer);
{-----}

var
    passo,
    ro,
    zw,
    xw,
    zm1,xm1,
    zm2,xm2,
    za,
    xa,
    aumento,
    teta1,
    teta2,
    teta1_d,
    teta2_d,
    x1,
    x2,
    zd,
    xd      : real;
    erro    : boolean;

```



```

begin
    passo:=passo_rot;
    ro:=beta;
    erro:=false;
    while not erro do
        begin
            calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,xm1,xm2,zm2,
                zw,xw,teta1,teta2,x1,x2,erro);
            if not erro then
                begin
                    calc_diff(x1,teta1,x2,teta2,ni,lambdaj,
                        teta1_d,teta2_d,zd,xd,errodif);
                    calc_amp(zd,xd,zl0,xl0,focal,gama,za,xa,aumento);
                    write(arq,ro*RADS:7:2);
                    write(arq,zd:7:2);
                    write(arq,xd:7:2);
                    write(arq,za:7:2);
                    write(arq,xa:7:2);
                    write(arq,aumento:7:2);
                    ro:=ro+passo;
                    end;
                end;
            end;
        end;
    ro:=beta;
    erro:=false;
    while not erro do
        begin
            calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,xm1,xm2,zm2,
                zw,xw,teta1,teta2,x1,x2,erro);
            if not erro then
                begin
                    calc_diff(x1,teta1,x2,teta2,ni,lambdaj,
                        teta1_d,teta2_d,zd,xd,errodif);
                    calc_amp(zd,xd,zl0,xl0,focal,gama,za,xa,aumento);
                    write(arq,ro*RADS:7:2);
                    write(arq,zd:7:2);
                    write(arq,xd:7:2);
                    write(arq,za:7:2);
                    write(arq,xa:7:2);
                    write(arq,aumento:7:2);
                    ro:=ro+passo;
                    end;
                end;
            end;
        end;
    end;
end;

end;

{-----}
function pos_eixo(z,x,gama : real) : real;
{-----}

begin
    pos_eixo:=(z-x*tan(-gama))*cos(-gama);
end;

{-----}
function posob_max(b,alfa,beta,gama,lz,zm0,xm0,passo_rot,lambdaj:real;
    j:integer) : real;
{-----}
{Encontra, para todos os pontos difratados, o que se encontra mais }
{afastado da origem quando projetado no eixo optico de L2. }

var
    passo,
    z_max,x_max,
    maximo,
    posicao,
    ro,
    zw,
    xw,
    teta1,
    teta2,
    teta1_d,
    teta2_d,
    x1,
    x2,
    zd,
    xd : real;
    erro : boolean;

begin
    maximo:=0;
    passo:=passo_rot/5;
    ro:=beta;
    erro:=false;

```

```

while not erro do
begin
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
if not erro then
begin
ro:=ro+passo;
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);
posicao:=pos_eixo(zd,xd,gama);
if posicao > maximo then
begin
maximo:=posicao;
z_max:=zd;
x_max:=xd;
end;
end;
end;

ro:=beta;
erro:=false;
while not erro do
begin
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
if not erro then
begin
ro:=ro+passo;
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);
posicao:=pos_eixo(zd,xd,gama);
if posicao > maximo then
begin
maximo:=posicao;
z_max:=zd;
x_max:=xd;
end;
end;
end;

posob_max:=maximo;
end;

{-----}
procedure tabl_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,lambda:real;

```

```

j:integer);
{-----}

var
passo,
ro,
zw,
xw,
teta1,
teta2,
teta1_d,
teta2_d,
x1,
x2,
zd,
xd : real;
erro : boolean;

begin
passo:=passo_rot;
ro:=beta;
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);
erro:=false;
while not erro do
begin
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
if not erro then
begin
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);
write(arq,ro*RADS:7:2);
write(arq,zw:7:2);
write(arq,xw:7:2);
write(arq,x1:7:2);
write(arq,teta1*RADS:7:2);
write(arq,x2:7:2);
write(arq,teta2*RADS:7:2);
write(arq,teta1_d*RADS:7:2);

```

```

write(arq,teta2_d*RADS:7:2);
if not errodif then
begin
write(arq,zd:7:2);
write(arq,xd:7:2);
end;
ro:=ro+passo;

writeln(arq);
end;
end;

writeln(arq,'->');
ro:=beta;
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);
erro:=false;
while not erro do
begin
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
if not erro then
begin
calc_diff(x1,teta1,x2,teta2,ni,lambda,j,
teta1_d,teta2_d,zd,xd,errodif);

write(arq,ro*RADS:7:2);
write(arq,zw:7:2);
write(arq,xw:7:2);
write(arq,x1:7:2);
write(arq,teta1*RADS:7:2);
write(arq,x2:7:2);
write(arq,teta2*RADS:7:2);
write(arq,teta1_d*RADS:7:2);
write(arq,teta2_d*RADS:7:2);
if not errodif then
begin
write(arq,zd:7:2);
write(arq,xd:7:2);
end;

ro:=ro+passo;
writeln(arq);
end;
end;

end;
end;

{-----}
procedure inicializa_vars;
{-----}

begin
passo_rot:=PASSO_INICIAL;
ro:=beta;
erro:=false;
errodif:=false;
frozen:=false;
nextfrozen:=false;

setrgb16(0,0,0,0);
setrgb16(1,0,0,31);
setrgb16(2,0,31,0);
setrgb16(3,0,31,31);
setrgb16(4,31,0,0);
setrgb16(5,31,0,31);
setrgb16(6,31,31,0);
setrgb16(7,31,31,31);
setrgb16(8,20,20,20);
setrgb16(9,0,0,63);
setrgb16(10,0,63,0);
setrgb16(11,0,63,63);
setrgb16(12,63,0,0);
setrgb16(13,63,0,63);
setrgb16(14,63,63,0);
setrgb16(15,63,63,63);
end;

{-----}
procedure atribui_defaults;
{-----}

begin
b:=B;
lz:=LZ;
lr:=LR;

```

```

j:=J;
ni:=NI;
alfa:=ALFA;
beta:=BETA;
escala:=ESCALA_X;
focal:=FOCAL;
dist_amp:=DIST_A;
diam_lente:=300;
end;

procedure espelho;
begin
  calc_espelho(b,alfa,lz,ro,zm0,xm0,zm1,xm1,zm2,xm2);
  plota_espelho(zm0,xm0,zm1,xm1,zm2,xm2,COR_ESPELHO);
end;

procedure espelho_ant;
begin
  calc_espelho(b,alfa,lz,ro_ant,zm0_ant,xm0_ant,
    zm1_ant,xm1_ant,zm2_ant,xm2_ant);
  plota_espelho(zm0_ant,xm0_ant,zm1_ant,xm1_ant,
    zm2_ant,xm2_ant,COR_ESPELHO);
end;

procedure lente;
begin
  calc_lente(d_o,diam_lente,gama,zl0,xl0,zl1,xl1,zl2,xl2);
  plota_lente(zl0,xl0,zl1,xl1,zl2,xl2,COR_LENTE);
end;

procedure incidencia;
begin
  calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
    zw,xw,teta1,teta2,xl1,xl2,erro);
  plota_incidencia(b,alfa,lz,zw,xw,xl1,xl2,zm0,xm0,zm1,xm1,zm2,xm2,COR_LUZ);
end;

procedure incidencia_ant;
begin
  plota_incidencia(b,alfa,lz,zw_ant,xw_ant,xl1_ant,
    x2_ant,zm0,xm0,zm1_ant,xm1_ant,
    zm2_ant,xm2_ant,COR_LUZ);
end;

```

```

procedure ampliacao;
begin
  if abs(xdr)<lr/2 then
    {difracao vermelha nao foi evanescente}
  begin
    calc_amp(zdr,xdr,zl0,xl0,focal,gama,zar,xar,aumr);
    foco_amp(zar,xar,aumr,VERMELHO);
  end;
  if abs(xdg)<lr/2 then
    {difracao verde nao foi evanescente}
  begin
    calc_amp(zdg,xdg,zl0,xl0,focal,gama,zag,xag,aumg);
    foco_amp(zag,xag,aumg,VERDE);
  end;
  if abs(xdb)<lr/2 then
    {difracao azul nao foi evanescente}
  begin
    calc_amp(zdb,xdb,zl0,xl0,focal,gama,zab,xab,aumb);
    foco_amp(zab,xab,aumb,AZUL);
  end;
end;

procedure ampliacao_ant;
begin
  if abs(xdr_ant)<lr/2 then
    {difracao vermelha anterior nao foi evanescente}
  begin
    foco_amp(zar_ant,xar_ant,aumr_ant,VERMELHO);
  end;
  if abs(xdg_ant)<lr/2 then
    {difracao verde anterior nao foi evanescente}
  begin
    foco_amp(zag_ant,xag_ant,aumg_ant,VERDE);
  end;
  if abs(xdb_ant)<lr/2 then
    {difracao azul anterior nao foi evanescente}
  begin
    foco_amp(zab_ant,xab_ant,aumb_ant,AZUL);
  end;
end;

procedure difracao;
begin
  calc_diff(xl,teta1,x2,teta2,ni,LAMBDA_RED,j,
    teta1_dr,teta2_dr,zdr,xdr,errodif);
  plota_diff(xl,x2,zdr,xdr,VERMELHO);
  calc_diff(xl,teta1,x2,teta2,ni,LAMBDA_GREEN,j,
    teta1_dg,teta2_dg,zdg,xdg,errodif);
  plota_diff(xl,x2,zdg,xdg,VERDE);
  calc_diff(xl,teta1,x2,teta2,ni,LAMBDA_BLUE,j,
    teta1_db,teta2_db,zdb,xdb,errodif);
end;

```

```

plota_dif(x1,x2,zdb,xdb,AZUL);
end;

procedure dif_ant;
begin
  plota_dif(x1_ant,x2_ant,zdr_ant,xdr_ant,VERMELHO);
  plota_dif(x1_ant,x2_ant,zdg_ant,xdg_ant,VERDE);
  plota_dif(x1_ant,x2_ant,zdb_ant,xdb_ant,AZUL);
end;

procedure arcos;
begin
  arco_branco(b,alfa,beta,lz,zm0,xm0,passo_rot,COR_ARCO);
  arco_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_RED,j,VERMELHO);
  arco_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_GREEN,j,VERDE);
  arco_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_BLUE,j,AZUL);
  arco_ampliado (z10,x10,focal,gama,
    b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_RED,j,VERMELHO);
  arco_ampliado (z10,x10,focal,gama,
    b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_GREEN,j,VERDE);
  arco_ampliado (z10,x10,focal,gama,
    b,alfa,beta,lz,zm0,xm0,passo_rot,
    LAMBDA_BLUE,j,AZUL);
end;

{-----}
procedure t_difracao;
{-----}
begin
  assign(arq,'c:\users\diamond\holocad\zcontrol\difracao.txt');
  rewrite(arq);
  writeln(arq,'HOLOCAD - SIMULACAO COM REDE REFLEXIVA');
  writeln(arq,'*****');
  writeln(arq,'Tabela de difracao');
  writeln(arq,'Braco de rotacao de Mz : 'b:4:0,' mm');
  writeln(arq,'Largura do espelho Mz : 'lz:4:0,' mm');
  writeln(arq,'Ordem de difracao considerada : 'j:2);
  writeln(arq,'Frequencia espacial da rede : 'ni:4:0,' ln/mm');
  writeln(arq,'Angulo do BRACO em rel. a normal da rede : ');
end;

writeln(arq,'Frequencia espacial da rede : 'ni:4:0,' ln/mm');;
writeln(arq,'Angulo do BRACO em rel. a normal da rede : ');
  (alfa*RADS)>6:1,' graus');
  (beta*RADS)>6:1,' graus');

writeln(arq);
writeln(arq,'VERMELHO ('(LAMBDA_RED*1e6):4:0,' nm)');
writeln(arq,'RO ZW XW X1 TETA1 X2 TETA2');
writeln(arq,'T1_D T2_D ZD XD');
writeln(arq);
tabl_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_RED,j);
writeln(arq);
writeln(arq,'VERDE ('(LAMBDA_GREEN*1e6):4:0,' nm)');
writeln(arq,'RO ZW XW X1 TETA1 X2 TETA2');
writeln(arq,'T1_D T2_D ZD XD');
writeln(arq);
tabl_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_GREEN,j);
writeln(arq);
writeln(arq,'AZUL ('(LAMBDA_BLUE*1e6):4:0,' nm)');
writeln(arq,'RO ZW XW X1 TETA1 X2 TETA2');
writeln(arq,'T1_D T2_D ZD XD');
writeln(arq);
tabl_espectral(b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_BLUE,j);
close(arq);
end;

{-----}
procedure t_ampliacao;
{-----}
begin
  assign(arq,'c:\users\diamond\holocad\zcontrol\amplia.txt');
  rewrite(arq);
  writeln(arq,'HOLOCAD - SIMULACAO COM REDE REFLEXIVA');
  writeln(arq,'*****');
  writeln(arq,'Tabela de Ampliacao');
  writeln(arq);
  writeln(arq,'Braco de rotacao de Mz : 'b:4:0,' mm');
  writeln(arq,'Largura do espelho Mz : 'lz:4:0,' mm');
  writeln(arq,'Ordem de difracao considerada : 'j:2);
  writeln(arq,'Frequencia espacial da rede : 'ni:4:0,' ln/mm');
  writeln(arq,'Angulo do BRACO em rel. a normal da rede : ');
end;

```



```

(alfa*RADS)/6:1,' graus');
writeln(arq,'Semi-ângulo do feixe branco em rel. ao BRACO : ');
(beta*RADS)/6:1,' graus');
writeln(arq,'Centro da lente L2 : Z=:zl0:4:1;', X=:xl0:4:1,');
writeln(arq,'Distancia focal de L2 : ,focal:4:1);
writeln(arq,'Angulo da normal de L2 em rel. a normal da rede : ',
(gama*RADS)/6:1);
writeln(arq);
writeln(arq,'VERMELHO (',(LAMBDA_RED*1e6)/4:0,' nm)');
writeln(arq,'-----');
writeln(arq,' RO ZD XD ZA XA A');
writeln(arq);
tabl_ampliacao(zl0,xl0,focal,gama,
b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_RED,j);
writeln(arq);
writeln(arq,'VERDE (',(LAMBDA_GREEN*1e6)/4:0,' nm)');
writeln(arq,'-----');
writeln(arq,' RO ZD XD ZA XA A');
writeln(arq);
tabl_ampliacao(zl0,xl0,focal,gama,
b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_GREEN,j);
writeln(arq);
writeln(arq,'AZUL (',(LAMBDA_BLUE*1e6)/4:0,' nm)');
writeln(arq,'-----');
writeln(arq,' RO ZD XD ZA XA A');
writeln(arq);
tabl_ampliacao(zl0,xl0,focal,gama,
b,alfa,beta,lz,zm0,xm0,passo_rot,LAMBDA_BLUE,j);
writeln(arq);
close(arq);
end;

{-----}
procedure salva_status;
{-----}
begin
ro:=ro_ant;
xw_ant:=xw;
zw_ant:=zw;
xl_ant:=xl;
x2_ant:=x2;
zm0_ant:=zm0;
xm0_ant:=xm0;
zm1_ant:=zm1;
xm1_ant:=xm1;

```

```

zm2_ant:=zm2;
xm2_ant:=xm2;
zdr_ant:=zdr;
xdr_ant:=xdr;
zdg_ant:=zdg;
xdg_ant:=xdg;
zdb_ant:=zdb;
xdb_ant:=xdb;
zar_ant:=zar;
xar_ant:=xar;
zag_ant:=zag;
xag_ant:=xag;
zab_ant:=zab;
xab_ant:=xab;
aumr_ant:=aumr;
aumg_ant:=aumg;
aumb_ant:=aumb;
zli1r_ant:=zli1r;
xli1r_ant:=xli1r;
zli2r_ant:=zli2r;
xli2r_ant:=xli2r;
zli1g_ant:=zli1g;
xli1g_ant:=xli1g;
zli2g_ant:=zli2g;
xli2g_ant:=xli2g;
zli1b_ant:=zli1b;
xli1b_ant:=xli1b;
zli2b_ant:=zli2b;
xli2b_ant:=xli2b;
erroamp_ant:=erroamp;
end;

{-----}
procedure recupera_status;
{-----}
begin
ro:=ro_ant;
zw:=zw_ant;
xw:=xw_ant;
xl:=xl_ant;
x2:=x2_ant;
zm0:=zm0_ant;
xm0:=xm0_ant;
zm1:=zm1_ant;

```

```

xm1:=xm1_ant;
zm2:=zm2_ant;
xm2:=xm2_ant;
zdr:=zdr_ant;
xdr:=xdr_ant;
zdg:=zdg_ant;
xdg:=xdg_ant;
zdb:=zdb_ant;
xdb:=xdb_ant;
zar:=zar_ant;
xar:=xar_ant;
zag:=zag_ant;
xag:=xag_ant;
zab:=zab_ant;
xab:=xab_ant;
aumr:=aumr_ant;
aumg:=aumg_ant;
aumb:=aumb_ant;
zli1r:=zli1r_ant;
xli1r:=xli1r_ant;
zli1g:=zli1g_ant;
xli1g:=xli1g_ant;
zli1b:=zli1b_ant;
xli1b:=xli1b_ant;
zli2r:=zli2r_ant;
xli2r:=xli2r_ant;
zli2g:=zli2g_ant;
xli2g:=xli2g_ant;
zli2b:=zli2b_ant;
xli2b:=xli2b_ant;
erroamp:=erroamp_ant;

end;

{-----}
procedure salvar_teta;
{-----}
begin
  with palettearray[0] do
    begin
      R:=63;
      G:=63;
      B:=63;
    end;
    savetgascreen('c:\users\diamand\holocad\zcontrol\zcentrf');
  end;
end;

```

```

with palettearray[0] do
begin
  R:=0;
  G:=0;
  B:=0;
end;
end;

{-----}
procedure posiciona_lente;
{-----}
begin
  {Posiciona a lente L2 no angulo de difracao verde (central no
  espectro) de um raio incidindo em 0,0.}
  gama:=dif_angle(alfa.ni,LAMBDA_GREEN,j);
  if abs(gama)>89*GRAUS then
  {se a posicao central ja for evanescente...}
  {atribuir um valor arbitrario para gama. }
  gama:=45*GRAUS*(gama/abs(gama));
  max:=0;
  max_r:=posob_max(b,alfa,beta,gama,lz,zm0,xm0,passo_rot,LAMBDA_RED,j);
  max_g:=posob_max(b,alfa,beta,gama,lz,zm0,xm0,passo_rot,LAMBDA_GREEN,j);
  max_b:=posob_max(b,alfa,beta,gama,lz,zm0,xm0,passo_rot,LAMBDA_BLUE,j);
  if max_r>max then max:=max_r;
  if max_g>max then max:=max_g;
  if max_b>max then max:=max_b;
  d_o:=max+dobj_amp(dist_amp,focal);
end;

{-----}
procedure calc_inlente (x1,teta1_d,x2,teta2_d,zl0,
xl0,gama,diam_lente,d_o : real;
var zli1,xli1,zli2,xli2 : real;
var erroamp : boolean);
{-----}
var
  b0 : real;
  muito_longe : boolean;
begin
  muito_longe:=false;
  b0:=d_o/sin(gama);
  intersecao(x1,-teta1_d,b0,-(90*GRAUS+gama),zli1,xli1,muito_longe);
  intersecao(x2,-teta2_d,b0,-(90*GRAUS+gama),zli2,xli2,muito_longe);
end;

```

```

if (muito longe) or
  (abs(dist(zl0,xl0,zli1,xli1)) > (diam_lente/2)) or
  (abs(dist(zl0,xl0,zli2,xli2)) > (diam_lente/2)) then
  erroamp:=true
else
  erroamp:=false;
end;

{-----}
procedure plota_inlente(x1,x2,zli1,xli1,zli2,xli2 : real;
  cor : integer);
{-----}

begin
  setcolor(cor);
  line(tx(0),ty(x1),tx(zli1),ty(xli1));
  line(tx(0),ty(x2),tx(zli2),ty(xli2));
end;

{-----}
procedure inlente;
{-----}

var
  erro_r,erro_g,erro_b : boolean;

begin
  if abs(xdr)<lr/2 then
    {difracao vermelha nao foi evanescente}
    begin
      calc_inlente(x1,teta1_dr,x2,teta2_dr,zl0,xl0,gama,diam_lente,d_o,
        zli1r,xli1r,zli2r,xli2r,erro_r);
      end;
    if abs(xdg)<lr/2 then
      begin
        calc_inlente(x1,teta1_dg,x2,teta2_dg,zl0,xl0,gama,diam_lente,d_o,
          zli1g,xli1g,zli2g,xli2g,erro_g);
        end;
    if abs(xdb)<lr/2 then
      begin
        calc_inlente(x1,teta1_db,x2,teta2_db,zl0,xl0,gama,diam_lente,d_o,
          zli1b,xli1b,zli2b,xli2b,erro_b);
        end;
    if (abs(xdr)<lr/2) and
      (abs(xdg)<lr/2) and

```

```

(abs(xdb)<lr/2) and
  (not(erro_r)) and
  (not(erro_g)) and
  (not(erro_b)) then
  begin
    erroamp:=false;
    plota_inlente(x1,x2,zli1r,xli1r,zli2r,xli2r,VERMELHO);
    plota_inlente(x1,x2,zli1g,xli1g,zli2g,xli2g,VERDE);
    plota_inlente(x1,x2,zli1b,xli1b,zli2b,xli2b,AZUL);
  end
else
  erroamp:=true;
end;

{-----}
procedure inlente_ant;
{-----}

begin
  if not erroamp_ant then
    begin
      plota_inlente(x1_ant,x2_ant,
        zli1r_ant,xli1r_ant,zli2r_ant,xli2r_ant,VERMELHO);
      plota_inlente(x1_ant,x2_ant,
        zli1g_ant,xli1g_ant,zli2g_ant,xli2g_ant,VERDE);
      plota_inlente(x1_ant,x2_ant,
        zli1b_ant,xli1b_ant,zli2b_ant,xli2b_ant,AZUL);
    end;
  end;

{-----}
procedure plota_outlente(zl,xl,za,xa : real ; cor : integer);
{-----}

begin
  setcolor(cor);
  line(tx(zl),ty(xl),tx(za),ty(xa));
end;

{-----}
procedure outlente;
{-----}

begin

```

```

if not erroamp then
begin
plota_outlente(zli1r,xli1r,zar,xar,VERMELHO);
plota_outlente(zli1g,xli1g,zag,xag,VERDE);
plota_outlente(zli1b,xli1b,zab,xab,AZUL);
plota_outlente(zli2r,xli2r,zar,xar,VERMELHO);
plota_outlente(zli2g,xli2g,zag,xag,VERDE);
plota_outlente(zli2b,xli2b,zab,xab,AZUL);
end;
end;

{-----}
procedure outlente_ant;
{-----}

begin
if not erroamp_ant then
begin
plota_outlente(zli1r_ant,xli1r_ant,zar_ant,xar_ant,VERMELHO);
plota_outlente(zli1g_ant,xli1g_ant,zag_ant,xag_ant,VERDE);
plota_outlente(zli1b_ant,xli1b_ant,zab_ant,xab_ant,AZUL);
plota_outlente(zli2r_ant,xli2r_ant,zar_ant,xar_ant,VERMELHO);
plota_outlente(zli2g_ant,xli2g_ant,zag_ant,xag_ant,VERDE);
plota_outlente(zli2b_ant,xli2b_ant,zab_ant,xab_ant,AZUL);
end;
end;

{-----}
procedure plot_a_dispo;
{-----}

var
tamanho : real;

begin
setcolor(COR_LUZ);
tamanho:=aumt*FOCOSIZE;
line(tx(zar+tamanho),ty(xar+tamanho),
tx(zar+tamanho),ty(xar+tamanho));
line(tx(zar+tamanho),ty(xar+tamanho),
tx(zar+tamanho),ty(xar+tamanho));
end;

{-----}
procedure ask_parametros (var b,ni,alfa,beta,lz,lr,t,escala,
focal,dist_ampa : real;
var j : integer;
var fim_prog : boolean);
{-----}

begin
restorectmode;
clrscr;
fim_prog:=false;

{Pergunta : b(mm)}
write('Braco de rotacao de Mz [',b:4:0,' mm] : ');
readln(resposta);
if resposta<>" then
if (resposta='FIM') or (resposta='fim') then
begin
fim_prog:=true;
exit;
end
else
val(resposta,b,code);

{Pergunta : lz(mm)}
write('Largura do espelho Mz [',lz:4:0,' mm] : ');
readln(resposta);
if resposta<>" then
val(resposta,lz,code);

{Pergunta : lr(mm)}
write('Largura da rede [',lr:4:0,' mm] : ');
readln(resposta);
if resposta<>" then
val(resposta,lr,code);

{Pergunta : j(1 ou -1)}
write('Ordem de difracao considerada [',j:2,' : ');
readln(resposta);
if resposta<>" then
val(resposta,j,code);

{Pergunta : ni(ln/mm)}
write('Frequencia espacial da rede [',ni:4:0,' ln/mm] : ');

```

```

readln(resposta);
if resposta<>" then
  val(resposta,ni,code);

{Pergunta : alfa(graus), convertido depois em rads}
write('Angulo do BRACO em rel. a normal da rede [',
      (alfa*RADS)/6.1,' graus] : ');

readln(resposta);
if resposta<>" then
  begin
    val(resposta,alfa,code);
    {convertendo em radianos}
    alfa:=alfa*GRAUS;
  end;

write('Semi-angulo do feixe branco em rel. ao BRACO [',
      (beta*RADS)/6.1,' graus] : ');
readln(resposta);
if resposta<>" then
  begin
    val(resposta,beta,code);
    {convertendo em radianos}
    beta:=beta*GRAUS;
  end;

{
  {Pergunta : gama
  write('Angulo da normal de L2 em rel. a normal da rede [',
        (gama*RADS)/6.1,' graus] : ');

  readln(resposta);
  if resposta<>" then
    begin
      val(resposta,gama,code);
      {convertendo em radianos}
      gama:=gama*GRAUS;
    end;
  }

  {Pergunta : focal}
  write('Distancia focal de L2 [',focal/4.0,' mm] : ');
  readln(resposta);
  if resposta<>" then
    val(resposta,focal,code);
  {Pergunta : dist_amp}
}

write('Distancia total de ampliacao [',dist_amp/4.0,' mm] : ');
readln(resposta);
if resposta<>" then
  val(resposta,dist_amp,code);

{Resposta para o usuario : Aumento maximo e distancia objeto}
writeln;
writeln('Aumento maximo : ',
        aum_maximo(dist_amp,focal)/3.1);
write ('Dist. entre a imagem difratada mais proxima de L2');
writeln(' e o centro de L2 : ',dobj_amp(dist_amp,focal)/3.1,'mm');
writeln;

{Pergunta : escala}
write('Escala de exibicao [',escala/4.1,'] : ');
readln(resposta);
if resposta<>" then
  val(resposta,escala,code);
end;

{*****}
{ PROGRAMA PRINCIPAL }
{*****}

begin
  atribui_defaults;
  while true do
    begin
      ask_parametros(b,ni,alfa,beta,lz,lr,l,escala,
                     focal,dist_amp,j,fim_prog);
      {Ponto de SAIDA do programa}
      if fim_prog then
        begin
          restorecrtmode;
          clrscr;
          exit;
        end;

        inicia_grafico;
        inicializa_vars;
        espelho;

```

```

begin
  {Apagando quadro anterior}
  espelho_ant;
  incidencia_ant;
  dif_ant;
  if not frozen then
    begin
      ampliacao_ant;
      inlente_ant;
      outlente_ant;
    end
  else
    frozen:=false;
  {Pintando novo quadro}
  espelho;
  incidencia;
  difracao;
  ampliacao;
  inlente;
  outlente;
  if nextfrozen then
    begin
      nextfrozen:=false;
      frozen:=true;
    end;
  end;
  end;
  restorecrtmode;
  clrscr;
end;
end.

```

```

posiciona_lente;
plota_rede(lt,COR_REDE);
lente;
{t_difracao;}
{t_ampliacao;}
incidencia;
difracao;
arcos;
ampliacao;
plota_disp0;
inlente;
outlente;
tecla:= '';
while ord(tecla)<>27 do
  begin
    tecla:=readkey;
    salva_status;
    case uppercase(tecla) of
      #0 : {Seta pressionada}
        begin
          tecla:=readkey;
          case tecla of
            'H' : {Seta para cima}
              gira(ro,-passo_rot);
            'P' : {Seta para baixo}
              gira(ro,+passo_rot);
          end;
        end;
      'S' : {Salvar tela}
        begin
          salvar_tela;
        end;
      'C' : {Congelar feixes}
        begin
          nextfrozen:=true;
        end;
    end;
  end;
end;
calc_incidencia(b,alfa,beta,ro,lz,zm0,xm0,zm1,xm1,zm2,xm2,
zw,xw,teta1,teta2,x1,x2,erro);
if erro then
  begin
    {beep(300,100);}
    recupera_status;
  end
else

```

# ANEXO III

***LISTAGEM DO PROGRAMA “HPROJ”  
DE OPERAÇÃO, CALIBRAÇÃO, E  
CONTROLE DO SISTEMA.***

```

program HPROJ.PAS;
{Programa de operacao, calibracao e controle do HoloProjektor v1.0}
{por Marcelo Diamand, setembro de 1994}

```

```

{$i-}
uses crt,graph;

```

```

const
  LPT1=$3bc;
  PAUSA=2;
  ACOMODA=2;
  PAUSA_OBT=20;
  ACOMODA_OBT=20;
  ESCALA=20;
  REPETICAO=3;
  K_OBTURA=2;
  FOTO=FALSE;
  PAUSA_FOTO=500;
  FATOR=1;

```

```

type
  voxel = Record
    x,y,z : integer;
    invisivel : boolean;
  end;
  liga_pontos = array[1..50] of voxel;
  quadro = Record
    npontos : integer;
    desenho : liga_pontos;
  end;
  sequencia = array [1..20] of quadro;

```

```

var
  figura : liga_pontos;
  animacao : sequencia;
  x0,y0,
  npontos,
  nquadros,
  ind,j : integer;
  comando : byte;
  fim.apagado : boolean;
  tecla : char;
  nomearq : string;
  {-----}

```

```

procedure beep(nota,duracao : integer);
{-----}

begin
  sound(nota);
  delay(duracao);
  nosound;
end;

{-----}
procedure avisa_erro (erro : integer);
{-----}

var
  msg_aux,
  msg : string;
  tecla : char;

begin
  case erro of
    2 : msg:='Arquivo nao encontrado.';
    3 : msg:='Diretorio nao encontrado.';
    5 : msg:='Acesso nao permitido.';
  101: msg:='Erro de escrita no disco.';
  152: msg:='O drive nao esta pronto.';
  158: msg:='Setor nao encontrado.';
  162: msg:='Erro de hardware.';
  else msg:='Algum erro ocorreu.';
  end;
  if erro<>0 then
    begin
      msg_aux:=' (Tecla algo...)';
      msg:=concat(msg,msg_aux);
      write(msg);
      beep(800,500);
      tecla:=readkey;
      clrscr;
      end;
    end;

  {-----}
  procedure acende;
  {-----}
  var
    fase1,fase2 : byte;
    passos : integer;

```



```

begin
if apagado then
begin
fase1:=comando;
fase1:=fase1 or $80; {1000 0000}
port[LPT1]:=fase1;
delay(PAUSA_OBT);
for passos:=1 to K_OBTURA do
begin
fase1:=fase1 and $bf; {1011 1111}
port[LPT1]:=fase1;
delay(PAUSA_OBT);
fase2:=fase1 or $40; {0100 0000}
port[LPT1]:=fase2;
delay(ACOMODA_OBT);
end;
comando:=fase2;
apagado:=false;
end;
end;

{-----}
procedure apaga;
{-----}
var
fase1,fase2 : byte;
passos : integer;

begin
if not(apagado) then
begin
fase1:=comando;
fase1:=fase1 and $7f; {0111 1111}
port[LPT1]:=fase1;
delay(PAUSA_OBT);
for passos:=1 to K_OBTURA do
begin
fase1:=fase1 and $bf; {1011 1111}
port[LPT1]:=fase1;
delay(PAUSA_OBT);
fase2:=fase1 or $40; {0100 0000}
port[LPT1]:=fase2;
delay(ACOMODA_OBT);
end;
end;
end;

comando:=fase2;
apagado:=true;
end;

{-----}
procedure le_sequencia(var seq : sequencia;
var nquadtros : integer);
{-----}

var
arq : text;
x,y,z, : integer;
npontos : integer;
fim_quadro : boolean;
nome, : string;
visibilidade : string;
erro : integer;

begin
erro:=1;
while erro<>0 do
begin
clrscr;
write('Entre com o nome do arquivo : ');
readln(nome);
if nome="" then
begin
clrscr;
halt;
end
else
begin
assign(arq,'c:\users\diamand\cone\'+nome);
reset(arq);
erro:=ioresult;
avisa_erro(erro);
if erro=0 then
begin
nquadtros:=0;
while not SeekEof(arq) do
begin
{Leitura de um quadro}
nquadtros:=nquadtros+1;
fim_quadro:=false;

```

```

npontos:=0;
x:=0;
while (not SeekEof(arq)) and (x<>999) do
begin
if SeekEofn(arq) then
readln(arq) {Go to next line}
else
begin
npontos:=npontos+1;
read(arq.x);
read(arq.y);
read(arq.z);
if not SeekEofn(arq) then readln(arq,visibilidade)
else
begin
readln(arq);
visibilidade:='a';
end;
if (visibilidade='off') or (visibilidade='OFF') then
seq[nquadrros].desenho[npontos].invisible:=true
else
seq[nquadrros].desenho[npontos].invisible:=false;
seq[nquadrros].desenho[npontos].x:=x;
seq[nquadrros].desenho[npontos].y:=y;
seq[nquadrros].desenho[npontos].z:=z;
end;
end;
if (x=999) then npontos:=npontos-1;
if (seq[nquadrros].desenho[npontos].x <>
(seq[nquadrros].desenho[1].x) or
(seq[nquadrros].desenho[npontos].y <>
(seq[nquadrros].desenho[1].y) or
(seq[nquadrros].desenho[npontos].z <>
(seq[nquadrros].desenho[1].z) then
{A figura nao e ciclica}
{Faz voltar ao primeiro ponto, invisivelmente}
begin
npontos:=npontos+1;
seq[nquadrros].desenho[npontos]:=seq[nquadrros].desenho[1];
seq[nquadrros].desenho[npontos].invisible:=true;
end;
seq[nquadrros].nptos:=npontos;
end;
close(arq);
end;

```

```

end;
end;
end;
{-----}
function i(x : integer) : integer;
{-----}
begin
i:=x0+(x*ESCALA);
end;
{-----}
function j(y : integer) : integer;
{-----}
begin
j:=y0+(y*ESCALA);
end;
{-----}
procedure plota_quadro (figura : quadro);
{-----}
var
npontos,l,
x1,y1,z1,
x2,y2,z2 : integer;
begin
npontos:=figura.nptos;
for l:=1 to npontos-1 do
begin
x1:=figura.desenho[l].x;
y1:=figura.desenho[l].y;
{z1:=figura.desenho[l].z;}
x2:=figura.desenho[l+1].x;
y2:=figura.desenho[l+1].y;
{z2:=figura.desenho[l+1].z;}
line((x1),j(y1),i(x2),j(y2));
delay(100);
end;
end;
{-----}

```

```

procedure hline (xi,yi,zi,xf,yf,zf: integer; invisivel : boolean);
{-----}
{ A partir dos pontos (xi,yi,zi) e (xf,yf,zf) envia }
{ comandos para o driver de motor de passo de }
{ modo a realizar uma reta. As saidas utilizadas }
{ sao : }
{ 1 (X), 2 (Y), 3 (Z) }
{-----}

var
  x_teorico,
  y_teorico,
  z_teorico,
  passo_x,
  passo_y,
  passo_z : real;
  delta_x,
  delta_y,
  delta_z,
  delta_max,
  passo,
  x_motor,
  y_motor,
  z_motor : integer;
  fase1,
  fase2 : byte;

begin
  if invisivel then apaga;
  fase1:=comando;
  delta_x:=(xf-xi);
  delta_y:=(yf-yi);
  delta_z:=(zf-zi);
  delta_max:=abs(delta_x);
  if abs(delta_y)>delta_max then delta_max:=abs(delta_y);
  if abs(delta_z)>delta_max then delta_max:=abs(delta_z);
  if delta_max>0 then
    begin
      passo_x:=delta_x/delta_max;
      passo_y:=delta_y/delta_max;
      passo_z:=delta_z/delta_max;
      if passo_x<0 then
        fase1:=fase1 or $02 {0000 0010}
      else
        fase1:=fase1 and $fd; {1111 1101}
    end;
  end;

```

```

    if passo_y<0 then
      fase1:=fase1 or $08 {0000 1000}
    else
      fase1:=fase1 and $f7; {1111 0111}
    if passo_z<0 then
      fase1:=fase1 or $20 {0010 0000}
    else
      fase1:=fase1 and $df; {1101 1111}
  {Havendo ou nao trocas de direcao, atualiza a entrada do driver}
  port[LPT1]:=fase1;
  comando:=fase1;
  delay(PAUSA);
  x_teorico:=xi;
  y_teorico:=yi;
  z_teorico:=zi;
  x_motor:=round(x_teorico);
  y_motor:=round(y_teorico);
  z_motor:=round(z_teorico);
  for passo:= 1 to abs(delta_max) do
    begin
      if FOTO and not invisivel then apaga;
      fase1:=comando;
      fase2:=fase1;
      x_teorico:=x_teorico+passo_x;
      if round(x_teorico) <> x_motor then
        begin
          x_motor:=round(x_teorico);
          fase1:=fase1 and $fe; {1111 1110}
          fase2:=fase2 or $01; {0000 0001}
        end;
      y_teorico:=y_teorico+passo_y;
      if round(y_teorico) <> y_motor then
        begin
          y_motor:=round(y_teorico);
          fase1:=fase1 and $fb; {1111 1011}
          fase2:=fase2 or $04; {0000 0100}
        end;
      z_teorico:=z_teorico+passo_z;
      if round(z_teorico) <> z_motor then
        begin
          z_motor:=round(z_teorico);
          fase1:=fase1 and $ef; {1110 1111}
          fase2:=fase2 or $10; {0001 0000}
        end;
      {O movimento acontece aqui}
    end;
  end;

```

```

port[LPT1]:=fase1;
delay(PAUSA);
port[LPT1]:=fase2;
delay(ACOMODA);
{-----}
if FOTO and not invisivel then
begin
  acende;
  delay(PAUSA_FOTO);
end
else
  comando:=fase2;
  fase1:=comando;
end;
end;
if invisivel then acende;
end;
{-----}
procedure inicia_driver;
{-----}
begin
  comando:=$FF; {11111111}
  port[LPT1]:=comando;
  delay(pausa);
end;
{-----}
procedure inicia_grafico;
{-----}
var
  modo,driver : integer;
begin
  DetectGraph(driver,modo);
  {Independente do Graph detectado, coloca em CGA HI}
  driver:=CGA;
  modo:=CGAhi;
  {InitGraph(driver,modo,'c:\language\tp7\bgi');}
  InitGraph(driver,modo,'c:\linguas\pascal');
  setcolor(15);
end;

```

```

{-----}
procedure calibra1(dir_m,fase1_m,fase2_m : byte);
{-----}
var
  fase1,fase2 : byte;
begin
  sound(700);
  delay(10);
  nosound;
  fase1:=comando;
  fase1:=fase1 or dir_m;
  port[LPT1]:=fase1;
  delay(PAUSA);
  fase1:=fase1 and fase1_m;
  port[LPT1]:=fase1;
  delay(PAUSA);
  fase2:=fase1 or fase2_m;
  port[LPT1]:=fase2;
  delay(ACOMODA);
end;
{-----}
procedure calibra2(dir_m,fase1_m,fase2_m : byte);
{-----}
var
  fase1,fase2 : byte;
begin
  sound(300);
  delay(10);
  nosound;
  fase1:=comando;
  fase1:=fase1 and dir_m;
  port[LPT1]:=fase1;
  delay(PAUSA);
  fase1:=fase1 and fase1_m;
  port[LPT1]:=fase1;
  delay(PAUSA);
  fase2:=fase1 or fase2_m;
  port[LPT1]:=fase2;
  delay(ACOMODA);
end;

```

```

{-----}
procedure calibra_obt1(dir_m,fase1_m,fase2_m : byte);
{-----}
var
  fase1,fase2 : byte;

begin
  sound(700);
  delay(10);
  nosound;
  fase1:=comando;
  fase1:=fase1 or dir_m;
  port[LPT1]:=fase1;
  delay(PAUSA_OBT);
  fase1:=fase1 and fase1_m;
  port[LPT1]:=fase1;
  delay(PAUSA_OBT);
  fase2:=fase1 or fase2_m;
  port[LPT1]:=fase2;
  delay(ACOMODA_OBT);
end;

{-----}
procedure calibra_obt2(dir_m,fase1_m,fase2_m : byte);
{-----}
var
  fase1,fase2 : byte;

begin
  sound(300);
  delay(10);
  nosound;
  fase1:=comando;
  fase1:=fase1 and dir_m;
  port[LPT1]:=fase1;
  delay(PAUSA_OBT);
  fase1:=fase1 and fase1_m;
  port[LPT1]:=fase1;
  delay(PAUSA_OBT);
  fase2:=fase1 or fase2_m;
  port[LPT1]:=fase2;
  delay(ACOMODA_OBT);
end;

```

```

{-----}
procedure show_quadro (figura : quadro);
{-----}
var
  npontos,l,
  x1,y1,z1,
  x2,y2,z2 : integer;
  visibilidade : boolean;

begin
  npontos:=figura.nptos;
  for l:=1 to npontos-1 do
    begin
      x1:=figura.desenho[l].x+(FATOR*figura.desenho[l].z);
      y1:=figura.desenho[l].y;
      z1:=figura.desenho[l].z;
      x2:=figura.desenho[l+1].x+(FATOR*figura.desenho[l+1].z);
      y2:=figura.desenho[l+1].y;
      z2:=figura.desenho[l+1].z;
      visibilidade:=figura.desenho[l+1].invisivel;
      hline(x1,y1,z1,x2,y2,z2,visibilidade);
    end;
  end;
  if FOTO then
    begin
      beep(300,200);
      beep(500,200);
      beep(800,200);
    end;
  end;

  { ***** }
  { PROGRAMA PRINCIPAL }
  { ***** }

begin
  le_sequencia(animacao,nquadros);
  clrscr;
  teste(animacao,1);
  readln;
  inicia_grafico;
  x0:=round(GetMaxX/2);
  y0:=round(GetMaxY/2);
  inicia_driver;
  apagado:=false;

```

```

end;
end;
end;
RestoreCtrlMode;
clscr;
end.

```

```

{plota_quadro(animacao[1]);}
fim:=false;
while not fim do
begin
for ind:=1 to quadros do
begin
for l:=1 to REPETICAO do
show_quadro(animacao[ind]);
if keypressed then
begin
tecla:=readkey;
case ord(tecla) of
0 : {tecla de comando especial pressionada}
begin
tecla:=readkey;
case ord(tecla) of
72 : {Seta para cima}
{Diminui y}
calibra1($08,$b,$04);
80 : {Seta para baixo}
{Aumenta y}
calibra2($f7,$b,$04);
77 : {Seta para a direita}
{Diminui Z}
calibra1($20,$ef,$10);
75 : {Seta para a esquerda}
{Aumenta Z}
calibra2($df,$ef,$10);
81 : {PgDn}
{Aumenta X}
calibra2($fd,$fe,$01);
79 : {End}
{Diminui X}
calibra1($02,$fe,$01);
59 : {F1}
{Gira obturador no sentido HORARIO}
calibra_obt1($80,$bf,$40);
60 : {F2}
{Gira obturador no sentido ANTI-HORARIO}
calibra_obt2($7f,$bf,$40);
end;
end;
27 : {Esc}
fim:=true;
end;

```