



**Micael Aléxis Pinto
Hilário Ramos**

Sistema de projeção holográfica digital de imagem e vídeo

Digital holographic image and video projection system



Universidade de Aveiro
2023

**Micael Aléxis Pinto
Hilário Ramos**

Sistema de projeção holográfica digital de imagem e vídeo

Digital holographic image and video projection system

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de eletrónica e telecomunicações, realizada sob a orientação científica do Doutor Rui Escadas, Professor associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

Prof. Doutor Telmo Reis Cunha
professor associado da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Paulo Sérgio de Brito André
professor catedrático da Universidade de Lisboa - Instituto Superior Técnico

Prof. Doutor Rui Manuel Escadas Ramos Martins
professor auxiliar da Universidade de Aveiro

**agradecimentos /
acknowledgements**

Agradeço toda a ajuda ao meu orientador professor Rui Escadas pelo seu acompanhamento a esse trabalho e ao professor Pedro Pombo por me ajudar na realização de alguns testes práticos no modulador espacial de luz e me dar uma melhor luz na compreensão relativamente a alguns dos conceitos base da holografia

Palavras Chave

programação, aplicações holográficas, desenho de hardware, manipulação de imagem e vídeo, multimédia.

Resumo

Este documento apresenta o trabalho desenvolvido na criação de um sistema holográfico de projeção de imagem e vídeo em tempo real. A grande motivação deste projeto reside na complexidade do mesmo, tendo sido abordados tanto a geração de vídeos holográficos a partir de imagens(ou sequência de imagens) 2D, bem como a sua reconstrução e projeção num setup experimental. Este trabalho envolveu inicialmente a pesquisa dos conceitos básicos sobre a holografia e posteriormente estudou-se alguns algoritmos de holografia computacional, que culminaram num sistema capaz de projeção holográfica em tempo real, a partir de um modulador espacial de luz. Apesar de funcional, o sistema criado apresenta algumas limitações para vídeos mais complexos e de viabilidade económica, devido ao elevado custo de um modulador espacial de luz, sendo então propostas sugestões de continuação deste trabalho para aprimoramento do sistema em si de forma a colmatar tais limitações, encerrando assim este documento.

Keywords

programming, holographic applications, hardware design, image and video manipulation, multimedia.

Abstract

This document presents the work developed in the creation of a holographic system for real-time image and video projection. The main motivation behind this project lies in its complexity, addressing both the generation of holographic videos from 2D images (or image sequences) and their reconstruction and projection in an experimental setup. The initial phase of this work involved researching the basic concepts of holography, followed by the study of various computational holography algorithms. These efforts culminated in a system capable of real-time holographic projection using a spatial light modulator. Despite its functionality, the created system has some limitations for more complex videos and economic feasibility due to the high cost of a spatial light modulator. Therefore, suggestions for future work to enhance the system and overcome these limitations are proposed, concluding this document.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	vii
Lista de Excertos de Código	ix
1 Introdução	1
1.1 Enquadramento e motivação	1
1.2 Objetivos	2
1.3 Metodologia de trabalho	2
2 Fundamentos básicos da Holografia	5
2.1 Luz	5
2.2 Interferência	7
2.3 Coerência da luz	9
2.4 Holografia	12
2.4.1 Definição	12
2.4.2 Gravação do holograma e reconstrução da imagem original	12
3 Holografia computacional	17
3.1 Geral	17
3.2 Tipos de hologramas	17
3.2.1 Holograma de Gabor	18
3.2.2 Holograma fora do eixo	20
3.2.3 Holograma de imagem	22
3.2.4 Holograma de Fourier	25
3.3 Holografia gerada por computador	28
3.3.1 Holograma de desvio de fase	29
3.3.2 Holograma Kinoform	33

3.3.3	Holograma binário de Fourier	35
3.4	Modulador espacial de luz	37
4	Implementação do projeto	39
4.1	Arquitetura do sistema	39
4.1.1	Geral	39
4.1.2	Hardware	40
4.1.3	Software	42
4.2	Testes remotos no raspberry	44
4.3	Simulação	54
4.3.1	Simulação da reconstrução de imagens individuais	55
4.3.2	Simulação da reconstrução de vídeos	64
4.4	Simulação em tempo real com o modulador espacial de luz	66
4.4.1	Configuração ótica da experiência	66
4.4.2	Resultados com imagens	67
4.4.3	Resultados com vídeos	70
4.5	Orçamento do projeto e alternativas futuras	76
5	Conclusão	79
5.1	Geral	79
5.2	Limitações do sistema	79
5.3	Futuros projetos	80
6	Bibliografia	81
7	Anexos	83

Lista de Figuras

1.1	Arquitetura do sistema.	2
2.1	Interferência construtiva.	8
2.2	Interferência destrutiva.	9
2.3	Interferômetro de Michelson.	10
2.4	Interferômetro de Young.	11
2.5	Gravação do holograma.	13
2.6	Reconstrução da imagem.	14
2.7	Exemplo de reconstrução da imagem.	14
3.1	Gravação do holograma através do método de Gabor.	18
3.2	Reconstrução da imagem do holograma de Gabor.	19
3.3	Gravação do holograma do tipo fora do eixo.	21
3.4	Reconstrução da imagem do holograma fora do eixo.	22
3.5	Gravação do holograma de imagem.	23
3.6	Reconstrução da imagem do holograma de imagem.	23
3.7	Objeto original (a) e a imagem reconstruída a partir da luz branca com as distâncias entre a imagem e o holograma de (b)1cm, (c)5cm e (d)15cm.	24
3.8	Gravação do holograma de Fourier.	25
3.9	Reconstrução da imagem do Holograma de Fourier.	26
3.10	Holograma de Fourier.	27
3.11	Reconstrução da imagem para o holograma de Fourier.	28
3.12	Reconstrução da imagem para o holograma de desvio de fase.	29
3.13	Descrição da célula em (a) como sendo o holograma complexo e (b) como o desvio de fase binária do holograma.	30
3.14	(a) Objeto original, (b) Holograma de desvio de fase, (c) Reconstrução da imagem, (d) Reconstrução da imagem ampliada com a seleção da ordem de difração de (c).	33
3.15	(a) Objeto original, (b) Máscara da fase aleatória com a fase entre $-\pi$ e π , (c) Espectro do objeto com máscara de fase aleatória, (d) Reconstrução da imagem do holograma kinoform.	35

3.16	(a) Holograma binário $B_r(x, y)$, (b) Holograma binário $B_i(x, y)$, (c) Reconstrução da imagem através de $B_r(x, y)$, (d) Reconstrução da imagem através de $B_r(x, y) + jB_i(x, y)$.	37
4.1	Arquitetura geral do sistema.	40
4.2	Raspberry pi 4 model B [www.raspberrypi.com].	40
4.3	LC 2002 Transmissive SLM [12].	42
4.4	Interface do programa de conversão de imagem para holograma.	43
4.5	Vídeo do holograma a ser construído.	44
4.6	Uso do ifconfig no terminal.	45
4.7	Criação de uma nova conexão.	46
4.8	Conectar ao Remote_Test	47
4.9	Credenciais de acesso.	48
4.10	Acesso remoto ao raspberry.	49
4.11	Acesso ao raspberry pelo winscp [19].	50
4.12	Envio de uma imagem aleatória para o raspberry.	51
4.13	Mensagem de envio.	51
4.14	Imagem enviada.	52
4.15	Git clone do repositório.	52
4.16	Repositório do código fonte.	53
4.17	Execução do programa.	54
4.18	Boneco stickman.	56
4.19	Simulação do Stickman para $z=-40$	57
4.20	Simulação do Stickman para $z=-50$	57
4.21	Simulação do Stickman para $z=-60$	58
4.22	(a) $z=-40$ mm, (b) $z=-50$ mm e (c) $z=-60$ mm.	59
4.23	Simulação com $\delta/2$	60
4.24	Delta igual ao delta da simulação.	61
4.25	Simulação com delta $2x$	61
4.26	Test Done com resolução de 512X512.	62
4.27	Test Done com resolução de 635X635.	62
4.28	Reconstrução da imagem do holograma com comprimento de onda no verde.	63
4.29	Reconstrução da imagem do holograma com comprimento de onda no azul.	63
4.30	Simulação do vídeo do discurso político	64
4.31	Simulação do vídeo do vídeo Globus-Wyatt Earth.	65
4.32	Simulação do vídeo do vídeo Globus-Wyatt Earth parte2.	65
4.33	Simulação do vídeo do trailer do filme quantumania.	65
4.34	Simulação da letra de uma música.	66
4.35	Configuração ótica do sistema.	67

4.36	Reconstrução da imagem de um cubo.	68
4.37	Reconstrução da imagem de um stickman.	68
4.38	Reconstrução da imagem da sigla DH.	69
4.39	Reconstrução da imagem de uma cara.	69
4.40	Reconstrução da imagem da sigla FCU.	70
4.41	Reconstrução da imagem da expressão Test Done.	70
4.42	Execução do programa.	71
4.43	Software de animação.	72
4.44	Animação com stickman no SLM.	73
4.45	Mensagem de fim da animação.	74
4.46	Reconstrução do número 1.	75
4.47	Reconstrução do número 10.	75
4.48	Reconstrução da palavra YES.	76
4.49	Representação de um DLPDLCR230NPEVM [11].	77

Lista de Tabelas

4.1	Propriedades do modulador espacial de luz	41
4.2	Tabela de preços	76

Lista de Excertos de Código

1	Código para simulação do holograma de imagem	84
2	Código de matlab para simulação do holograma de Fourier	85
3	Código de matlab para geração do holograma de desvio de fase.	86
4	Código de matlab para gerar o holograma kinoform	87
5	Código de matlab para gerar dois hologramas binários para sintetizar e exibir o holograma	88
6	Software com interface para converter imagens em hologramas	89
7	Software para conversão do vídeo numa sequência de hologramas	90
8	Software para simulação da imagem reconstruída	91
9	Software de reconstrução do vídeo holográfico	91
10	Software com interface para converter imagens em hologramas	92

Introdução

O tema desta dissertação foi proposta pelo próprio autor Micael Ramos, aluno do mestrado em engenharia de eletrónica e telecomunicações da universidade de Aveiro, no mês de Outubro de 2022. Esta dissertação consiste num trabalho de pesquisa e de projeto, que resulta num sistema de visualização holográfica de imagens e de vídeos, enviados remotamente a partir de um computador, como descrito na proposta de dissertação escrita pelo autor.

1.1 ENQUADRAMENTO E MOTIVAÇÃO

Dada à crescente evolução dos sistemas de realidade aumentada e também de realidade virtual, aplicável em diversas áreas da atividade humana, como por exemplo o entretenimento, foi proposto no âmbito deste projeto a implementação de um sistema, capaz de efetuar a projeção holográfica de uma imagem, ou de um vídeo remotamente a partir de um computador ou de um servidor. Pretende-se com este projeto conseguir a projeção de imagem 3D o mais realista possível, tendo como principal aplicação a área do entretenimento e do marketing.

A estrutura do sistema proposto para a realização deste projeto terá como sua composição interna um microcontrolador para manipulação de imagem e de vídeo, um módulo de comunicação para receção de dados, um módulo de som para a execução do som, caso se trate de um vídeo com som incluído, e por fim, a peça fundamental para o funcionamento deste projeto, um módulo de projetor holográfico, que, em termos práticos, projeta um holograma de uma imagem 2D manipulada pelo microcontrolador, dando origem a uma reconstrução dessa mesma imagem em 3D. A partir desse pensamento foi criado um modelo genérico de sistema de projeção holográfica de imagem e de vídeo, como o descrito na figura 1.1.

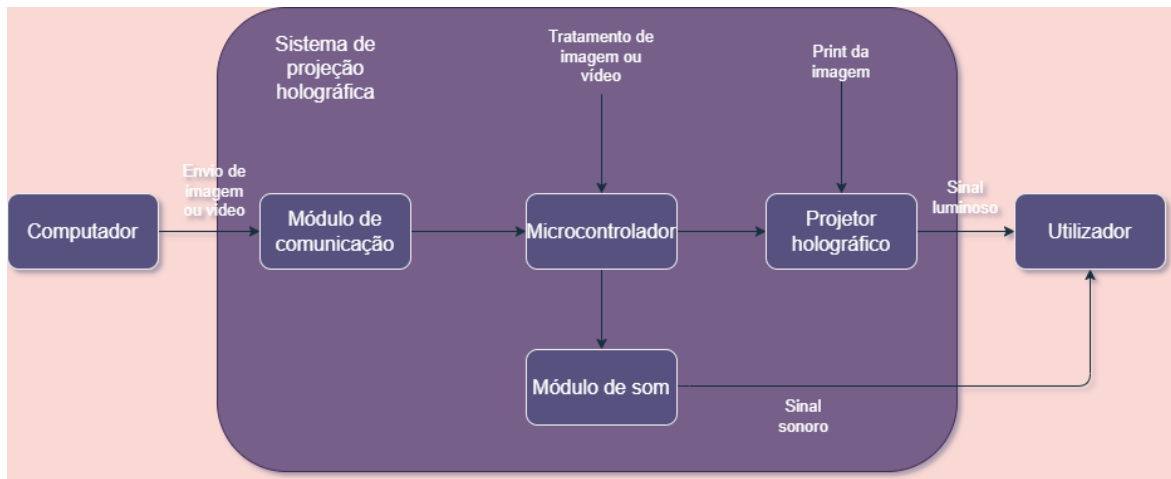


Figura 1.1: Arquitetura do sistema.

1.2 OBJETIVOS

Referenciando o que foi falado no enquadramento e motivação, que contém a base inicial da proposta desta dissertação, é também proposto um conjunto de objetivos base a alcançar no final da realização deste trabalho de pesquisa e de projeto.

Tais objetivos bases representam os requisitos necessários para o funcionamento esperado do sistema a ser implementado, ao qual é tido em consideração a sua adequação em relação ao nível de dificuldade pensado na proposta desta dissertação e ao facto de a mesma se basear numa área do conhecimento que vai um pouco além do conhecimento de um aluno de engenharia eletrónica e telecomunicações. Esses objetivos propostos para o funcionamento esperado do sistema final foram:

- Ter comunicação remota entre computador e dispositivo;
- Visualização de imagens holográficas minimamente nítidas, com altura, comprimento e profundidade;
- Deve incluir som, caso exista;
- Relativamente ao vídeo, a velocidade do mesmo deverá ser no mínimo de 10 FPS(Frames Por Segundo) e além disso deverá ter uma resolução mínima de 640x480;

1.3 METODOLOGIA DE TRABALHO

Para uma execução eficiente e em tempo útil da implementação do sistema proposto desta dissertação e no término da mesma, com uma margem considerável de sucesso, foi criada uma metodologia de trabalho baseada nos seguintes passos:

- 1) Pesquisa bibliográfica, através de livros ou de artigos científicos relacionados com a área da holografia;
- 2) Simulação de algoritmos de geração de hologramas encontrados nas pesquisas bibliográficas através do matlab;
- 3) Desenho e implementação do hardware e do firmware, conforme os requisitos estipulados para o funcionamento do sistema;

- 4) Realização de testes básicos, uso de imagens com letras, números ou outros símbolos;
- 5) Implementação da comunicação entre o computador e o dispositivo;
- 6) Realização de testes de envio de imagens e de vídeo para o dispositivo de forma remota;
- 7) Escrita do documento da dissertação;

Em termos cronológicos foi definido que até finais de Fevereiro, e inícios de Março estivesse como finalizada os passos 1) e 2). Os passos 3) e 4) seriam tratados no mês de Março até metade do mês de Abril, sendo os passos 5) e 6) efetuados até ao final de Abril e inícios de Maio. Por fim o passo 7), que foi a escrita da dissertação, sendo encerrado deste modo o ciclo de trabalho.

Este plano cronológico foi praticamente cumprido na maioria dos casos, havendo apenas alguns atrasos relativamente às deadlines no plano cronológico, devido à complexidade na concretização de tais passos, o que provocou um gasto substancial de tempo que foi colmatado através da realização de outras tarefas do trabalho em simultâneo, concluindo desta forma o trabalho em tempo útil.

O documento da dissertação contém a parte de pesquisa relativamente aos conceitos base da holografia e da holografia computacional nos capítulos 2 e 3, a parte da implementação do projeto, com os respetivos resultados, se encontra no capítulo 4 e a parte da conclusão da dissertação no capítulo 5. No final, existe ainda uma secção referida como anexos, onde todo o código usado para este trabalho se encontra disponível.

Fundamentos básicos da Holografia

Este capítulo em causa contém alguns tópicos básicos fundamentais que representam a base da holografia, que vai desde do fenómeno ondulatório da luz até a uma breve explicação relativamente ao conceito base da holografia, a definição dessa técnica e como se grava um holograma, de forma analógica, para depois visualizar a reconstrução da imagem do objeto que foi gravado.

2.1 LUZ

A luz que atinge os nossos olhos pode ser descrita como uma onda eletromagnética ou como uma corrente de partículas, denominadas de fotões. Tais descrições dependem do tipo de experiência em estudo, por exemplo, a interação da luz com uma estrutura atómica de matéria pode ser descrita pela ótica quântica, teoria da física que lida com a descrição da luz como um corrente de fotões.

Para fenómenos como a refração, difração e interferência, a luz é descrita através do modelo de uma onda, ao qual é baseado inteiramente na teoria clássica do eletromagnetismo. Os fenómenos de interferência e de difração são a base da holografia, sendo o modelo ondulatório da luz usado neste contexto. Desse modo a equação da onda da luz no vácuo e dada pela expressão

$$\nabla^2 \vec{E} - \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2} = 0, \quad (2.1)$$

onde \vec{E} é o campo elétrico e ∇^2 é representa o operador Laplaciano, que é definido pela expressão

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}, \quad (2.2)$$

a variável c da equação (2.1) representa a velocidade da luz, que no vácuo assume o valor aproximado de

$$c = 3 \times 10^8 \text{ m/s}. \quad (2.3)$$

O campo elétrico \vec{E} é uma quantidade vetorial, isso implica que a onda pode vibrar em qualquer direção ao qual é perpendicular à propagação da luz. No entanto é também possível, para o caso da nossa aplicação, considerar que se está sob presença de uma onda que apenas vibra num único plano, ou seja, é possível considerar, para o nosso caso, que a onda de luz se encontra linearmente polarizada. Nesse caso pode se escrever a equação da onda da forma

$$\frac{\partial^2 E}{\partial z^2} - \frac{1}{c^2} \frac{\partial^2 E}{\partial t^2} = 0, \quad (2.4)$$

onde nesse caso a luz propaga na direção de z .

A equação da onda descrita resulta numa onda planar harmônica e linearmente polarizada expressa como

$$E(x, y, z, t) = a \cos(\omega t - \vec{k} \cdot \vec{r} - \varphi_0). \quad (2.5)$$

Em que $E(x, y, z, t)$ representa o módulo do campo elétrico vetorial no ponto com o vetor $Vec(r) = (x, y, z)$ no tempo de t . A variável a representa a amplitude e o vetor de onda \vec{k} representa a direção de propagação da onda e é descrita como

$$\vec{k} = k\vec{n}, \quad (2.6)$$

o vetor \vec{n} representa a unidade vetorial da direção de propagação. Os pontos de fase igual estão localizados nos planos paralelos que são perpendiculares à propagação da onda. O módulo de \vec{k} , também chamado de número de onda, é dada pela expressão

$$|\vec{k}| = k = \frac{2\pi}{\lambda}. \quad (2.7)$$

A frequência angular ω corresponde à frequência f da onda da luz que é dada por

$$\omega = 2\pi f. \quad (2.8)$$

A frequência f também se encontra relacionada com o comprimento de onda λ através da velocidade de propagação da onda da luz c

$$c = \lambda f. \quad (2.9)$$

Por fim a fase é dada por

$$\varphi = -\vec{k} \cdot \vec{r} - \varphi_0, \quad (2.10)$$

em que a variável φ_0 representa a fase constante.

No vácuo, os comprimentos de onda da luz visível se encontram na faixa dos 400nm (cor violeta) até aos 780nm (cor vermelha), o que representa uma faixa de frequência entre os $7.5 \times 10^{14} Hz$ até aos $3.8 \times 10^{14} Hz$ respetivamente. Os sensores de luz, como um fotodíodo, filmes fotográficos, um CCD, ou até mesmo um olho humano não conseguem detetar essas altas frequências. O que é possível de detetar é a intensidade de radiação luminosa que atinge tais sensores de luz. Pode dizer que essa intensidade é proporcional ao valor médio do quadrado do campo elétrico. Assim a intensidade da radiação luminosa é expressa por

$$I = \epsilon_0 c \langle E^2 \rangle_t = \epsilon_0 c \lim_{t \rightarrow +\infty} \frac{1}{2T} \left(\int_{-T}^T E^2 dt \right). \quad (2.11)$$

O termo $\epsilon_0 c$ é uma constante que resulta da formalidade das equações de Maxwell. O ϵ_0 é a permissividade elétrica do vácuo. Para o modelo da onda plana descrita na Eq.(2.5), o resultado final da intensidade da luz é expressa como

$$I = \epsilon_0 c a^2 \langle \cos^2(\omega t - \vec{k}\vec{r} - \varphi_0) \rangle_t = \frac{1}{2} \epsilon_0 c a^2, \quad (2.12)$$

isso significa que de acordo com a Eq.(2.12) a intensidade da luz é proporcional ao quadrado da amplitude.

A expressão da Eq.(2.5) pode ser escrita na sua forma complexa.

$$E(x, y, z, t) = a \operatorname{Re}\{ \exp(i(\omega t - \vec{k}\vec{r} - \varphi_0)) \}. \quad (2.13)$$

O termo Re se refere à parte real da função complexa. O termo Re pode ser omitido em simulações computacionais.

Uma vantagem para a representação complexa da onda planar é o simples facto de ser possível a fatorização das partes espaciais e temporais dos exponenciais complexos.

$$E(x, y, z, t) = a \exp(i\varphi) \exp(i\omega t). \quad (2.14)$$

Em inúmeros cálculos óticos apenas a distribuição espacial da onda é a parte que interessa. Nesse caso específico então é possível ignorar a parte temporal e é considerada apenas a parte espacial do campo elétrico, à qual é denominada de amplitude complexa.

$$A(x, y, z) = a \exp(i\varphi). \quad (2.15)$$

Dessa forma é possível então escrever a intensidade da radiação I como

$$I = \frac{1}{2} \epsilon_0 c |A|^2 = \frac{1}{2} \epsilon_0 c A^* A = \frac{1}{2} \epsilon_0 c a^2, \quad (2.16)$$

onde o A^* é o complexo conjugado do A . Em grande parte dos cálculos práticos, o valor absoluto do I não é interessada, então o fator $\frac{1}{2} \epsilon_0 c$ pode simplesmente ser negligenciado, o que implica que a intensidade da radiação luminosa possa ser calculada como $I = |A|^2$.

2.2 INTERFERÊNCIA

O fenómeno da interferência resulta de uma superposição entre duas ou mais ondas no espaço. Se considerarmos que cada onda singular é descrita matematicamente por $\vec{E}_i(\vec{r}, t)$, a solução da equação da onda será descrita pelo princípio da superposição.

$$\vec{E}(\vec{r}, t) = \sum_i \vec{E}_i(\vec{r}, t), \quad i \in \mathbb{N}. \quad (2.17)$$

Tudo isso apenas é possível, pois a equação de onda é uma equação diferencial linear.

Considere agora duas ondas monocromáticas com frequências iguais e consequentemente comprimentos de onda iguais, polarizadas numa mesma direção. As amplitudes complexas dessas ondas são descritas como

$$A_1(x, y, z) = a_1 \exp(i\varphi_1), \quad (2.18)$$

$$A_2(x, y, z) = a_2 \exp(i\varphi_2). \quad (2.19)$$

Pelo princípio da superposição, a amplitude complexa resultante é calculada pela soma dessas duas ondas monocromático.

$$A = A_1 + A_2. \quad (2.20)$$

De acordo com a Eq.(2.16) a intensidade é expressa como

$$I = |A_1 + A_2|^2 = (A_1 + A_2)(A_1 + A_2)^* = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\varphi), \quad (2.21)$$

em que I_1 e I_2 representam intensidades individuais e $\Delta\varphi = \varphi_1 - \varphi_2$. A intensidade resultante representa a soma das intensidades individuais mais o termo $2\sqrt{I_1 I_2} \cos(\Delta\varphi)$, ao qual depende da diferença de fase entre as ondas. A intensidade alcança o seu máximo em todos os pontos ao qual se aplica a expressão

$$\Delta\varphi = 2n\pi, \quad n \in \mathbb{N}. \quad (2.22)$$

Isso resulta no fenómeno ao qual é chamado de interferência construtiva, que representa a soma das amplitudes de duas ou mais ondas que resultam numa onda maior. Para uma melhor exemplificação é possível visualizar a figura 2.1 que explica, numa forma mais visual, a interferência construtiva de duas ondas.

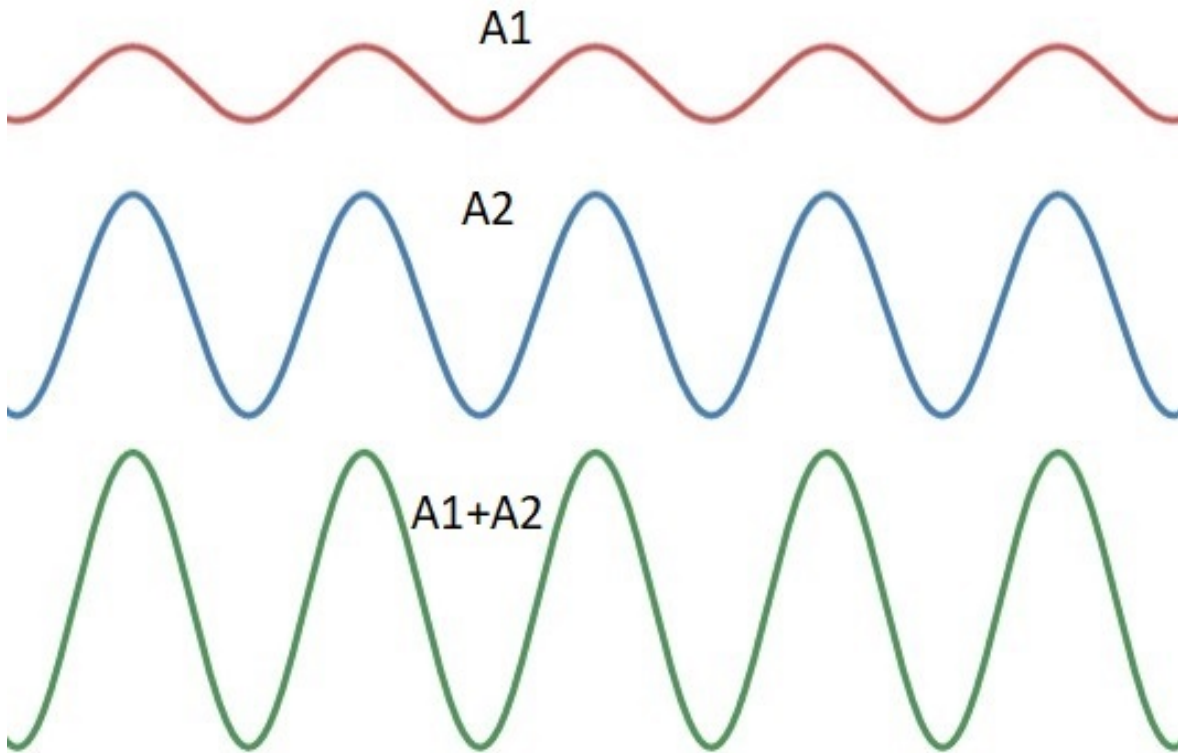


Figura 2.1: Interferência construtiva.

Em contrapartida a intensidade alcança o seu mínimo onde

$$\Delta\varphi = (2n + 1)\pi, \quad n \in \mathbb{N}. \quad (2.23)$$

Isso resulta no fenômeno denominado de interferência destrutiva, que representa a soma das amplitudes de duas ou mais ondas que resultam num onda menor. Tal acontece porque as ondas que são somadas tem sinais opostos, ou seja, é somado uma onda com sinal inverso relativamente à outra onda no qual se soma. É possível, através da visualização da figura 2.2, compreender melhor a explicação dada relativamente ao fenômeno da interferência destrutiva.

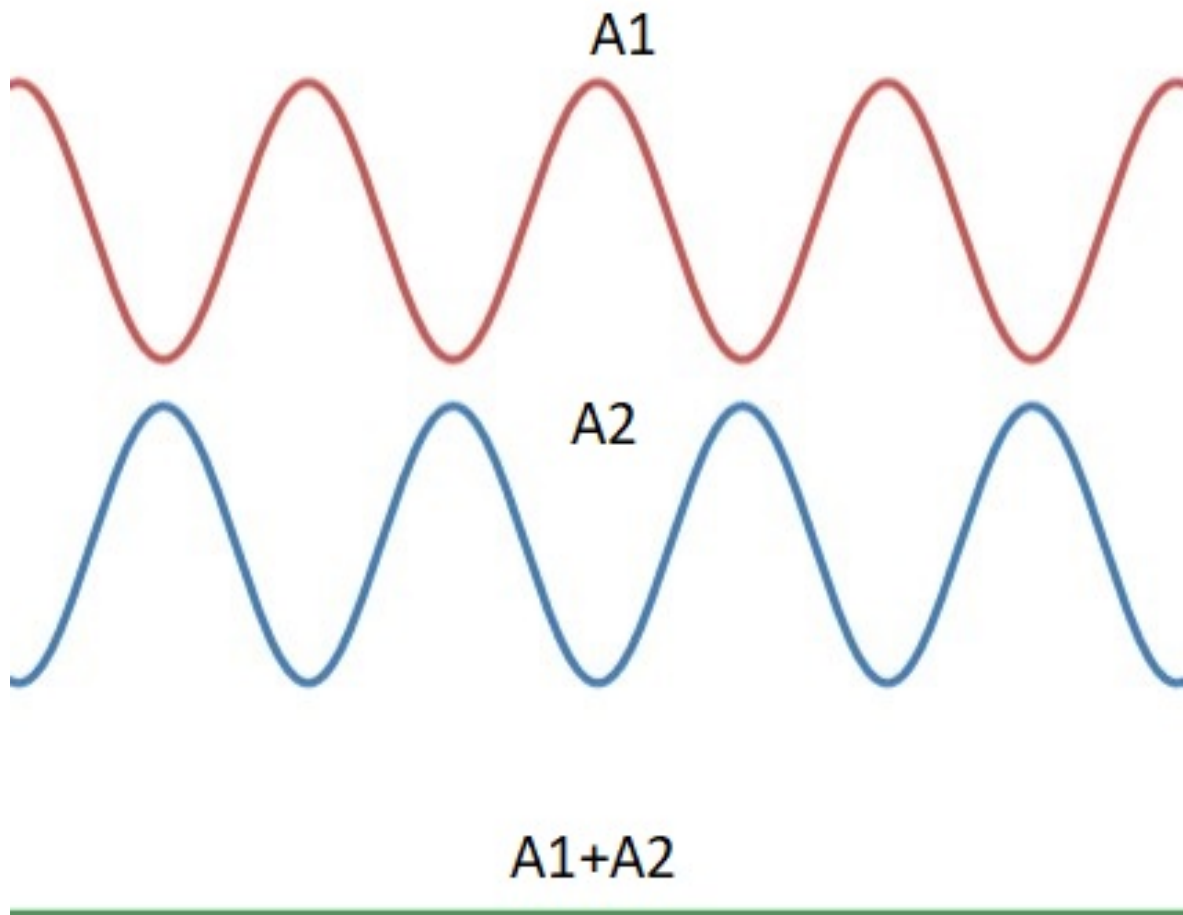


Figura 2.2: Interferência destrutiva.

2.3 COERÊNCIA DA LUZ

Nem todas as ondas de luz causam fenômenos de interferência entre si. É possível reparar que, numa experiência hipotética, duas fontes distintas de luz, sendo por exemplo duas lâmpadas ligadas direcionadas a uma tela, são simplesmente aditivas, ao qual apresentam apenas um brilho uniforme de acordo com a soma das suas intensidades individuais ao invés de apresentarem franjas escuras e claras como o esperado pela eq.(2.21).

De forma a gerar interferência entre duas ondas de luz, as fases de cada onda individual devem estar correlacionadas de uma forma especial. A esta propriedade de correlação é chamada de coerência. A coerência é a habilidade da luz em causar o fenômeno da interferência. Relativamente à coerência apenas existem dois tipos, a coerência temporal e a coerência espacial. A coerência temporal descreve a correlação da onda entre si em diferentes instantes. Um exemplo de coerência temporal pode ser encontrado num protótipo de dois feixes como o descrito no interferômetro de Michelson, ver na figura 2.3.

Num interferômetro de Michelson, a luz emitida pela fonte luminosa, por exemplo um laser, é dividida em duas ondas parciais através do divisor de feixe (BS). As ondas parciais viajam para os dois espelhos, M1 e M2, respectivamente e são refletidas de volta para o sentido da fonte luminosa. Depois de novamente passar pelo divisor de feixe, ambas as ondas parciais são sobrepostas numa tela. Normalmente as ondas parciais sobrepostas não são exatamente paralelas, no entanto se interferem com um ângulo pequeno. O resultado é um padrão de interferência bidimensional.

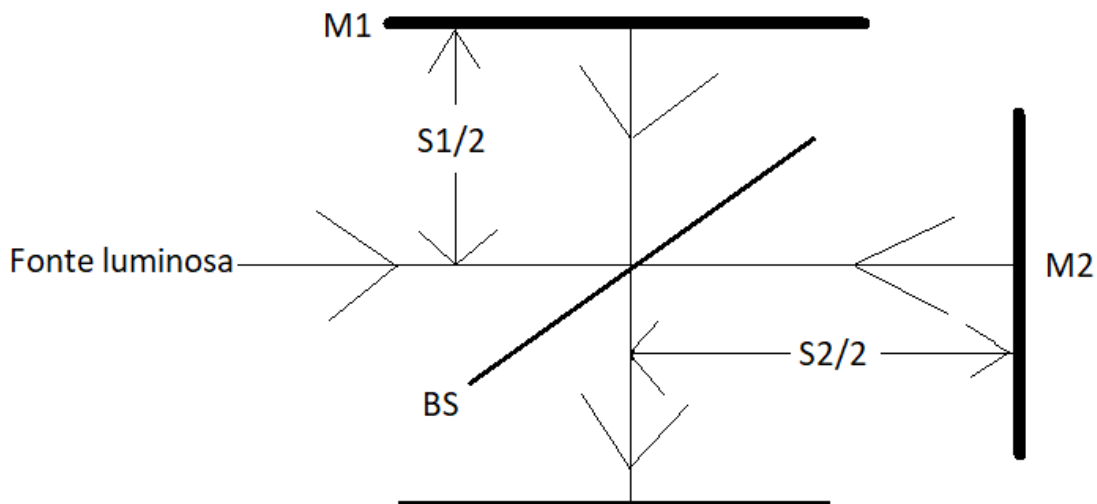


Figura 2.3: Interferômetro de Michelson.

O comprimento do caminho ótico do BS para M1 e de volta a BS é S_1 e do BS para o M2 e de volta a BS é S_2 . Para haver interferência diferença entre S_1 e S_2 não deverá exceder um determinado comprimento de L . Se a diferença entre S_1 e S_2 exceder esse comprimento de L , obtido experimentalmente, a interferência que permite a visualização das franjas desaparece e dá lugar a um brilho uniforme na tela.

A partir desse comprimento de caminho crítico é possível calcular o tempo de emissão correspondente para o trem da onda, ao qual é conotado de tempo de coerência.

$$\tau = \frac{L}{c} \quad (2.24)$$

A coerência espacial por sua vez descreve uma correlação mútua entre diferentes partes da mesma frente de onda. Essa propriedade pode ser medida através de um interferômetro

de Young, como descrito na figura 2.4. O interferômetro de Young é constituído por uma abertura de dois orifícios montado entre a fonte luminosa e a tela. Em determinadas condições, em que a distância entre os orifícios não excede o limite crítico a_k , é possível de verificar no interferômetro de Young padrões de interferência resultado de raios de luz que viajaram em caminhos diferentes até à tela.

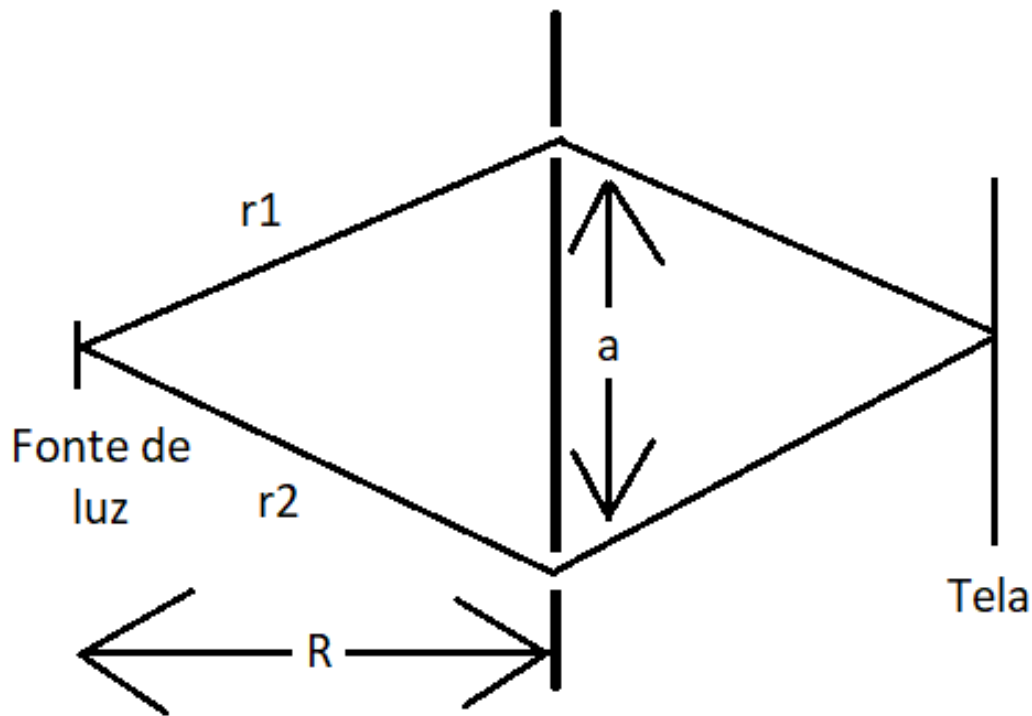


Figura 2.4: Interferômetro de Young.

O limite crítico a_k é denominado de coerência de distância e é dado pela expressão:

$$a_k = \frac{\lambda R}{h} \quad (2.25)$$

Em contraste com a coerência temporal, a coerência espacial não depende somente das propriedades da fonte luminosa como também depende da geometria do interferômetro. Se, por exemplo, a distância entre os orifícios aumentar ou a distância entre a fonte luminosa e a abertura diminuir de tal modo que $a > a_k$, o fenómeno da interferência simplesmente desaparece.

Caso seja da curiosidade do leitor, existem mais descrições relativamente ao interferômetro de Michelson e de Young que podem ser encontrados nas páginas de 11 a 15 na referência [2].

2.4 HOLOGRAFIA

2.4.1 Definição

A holografia é uma técnica inventada no ano de 1948 pelo físico e engenheiro eletrotécnico Gabor (pode ser encontrado na página 27 na referência [1]) que funciona como um método de gravação e reconstrução da amplitude e fase do campo eletromagnético. A holografia combina as duas palavras gregas, holos que significa "completo" e graphein que significa "escrever ou gravar". Dessa forma a técnica da holografia consiste na gravação completa da imagem, ou seja, da sua amplitude e fase, ao contrário da técnica da fotografia, no qual apenas é gravada a amplitude, o que implica perda de informação da fase que oferece a característica de tridimensionalidade à imagem.

Uma imagem armazenada holograficamente, também conhecida como um holograma, é a gravação de um padrão de interferências entre o campo de onda espalhado pelo objeto e o fundo coerente, denominado como onda de referência. Normalmente um holograma é gravado numa superfície plana, no entanto a mesma contém a informação sobre todo o campo de onda tridimensional. A informação do respetivo holograma se encontra codificado sob forma de tiras de interferências, que geralmente se encontram invisíveis aos olhos humanos devido às altas frequências espaciais. A imagem representada no holograma pode ser reconstruída através da iluminação do holograma com um feixe de luz.

2.4.2 Gravação do holograma e reconstrução da imagem original

Os hologramas costumam ser gravados a partir de uma configuração ótica constituída por uma fonte de luz coerente (laser), espelhos e lentes para guiar a luz e um meio de gravação, como por exemplo uma placa fotográfica, para a gravação do holograma. Um exemplo típico de uma configuração ótica para gravação do holograma é apresentada na figura 2.5, no qual apresenta um divisor de feixe que separa o feixe, que serve de iluminação do objeto, do feixe de referência, que serve de iluminação direta do meio de gravação. Ambos os feixes interferem entre si no meio da gravação e o padrão de interferência é guardado através por exemplo do desenvolvimento químico da placa fotográfica (sendo esta o meio da gravação), no caso de se tratar de um holograma analógico. Desta forma o holograma fica gravado no meio de gravação.

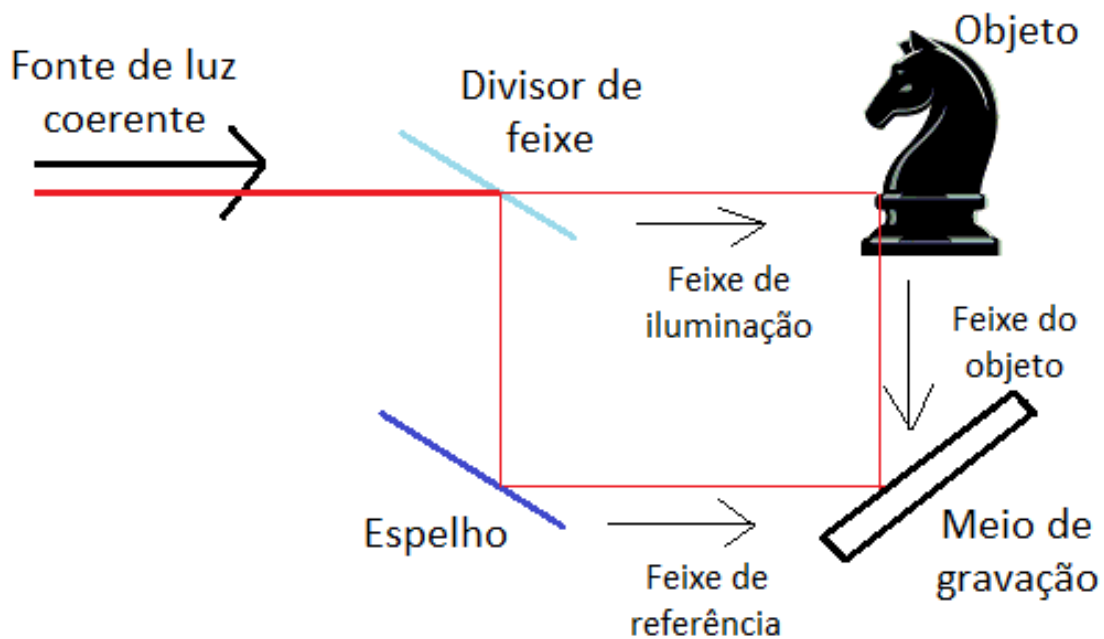


Figura 2.5: Gravação do holograma.

Após a gravação do holograma, para reconstruir a imagem do objeto original a partir do mesmo basta, a partir de uma fonte luminosa qualquer, iluminar o holograma gravado na placa e dessa forma é possível visualizar a imagem reconstruída, tal como o apresentado na figura 2.6. Nesse caso o observador observa a imagem virtual, que se encontra armazenado na placa e ao qual é indistinguível do objeto original. Dessa forma a imagem reconstruída exibe todos os efeitos de perspectiva e profundidade de foco. Um exemplo real da reconstrução da imagem de um holograma, gravado em seu respectivo meio de gravação, pode ser visualizado na figura 2.7, onde é possível verificar, através da luz de uma lâmpada, a existência das imagens virtuais que correspondem aos objetos reais usados para a gravação do holograma.

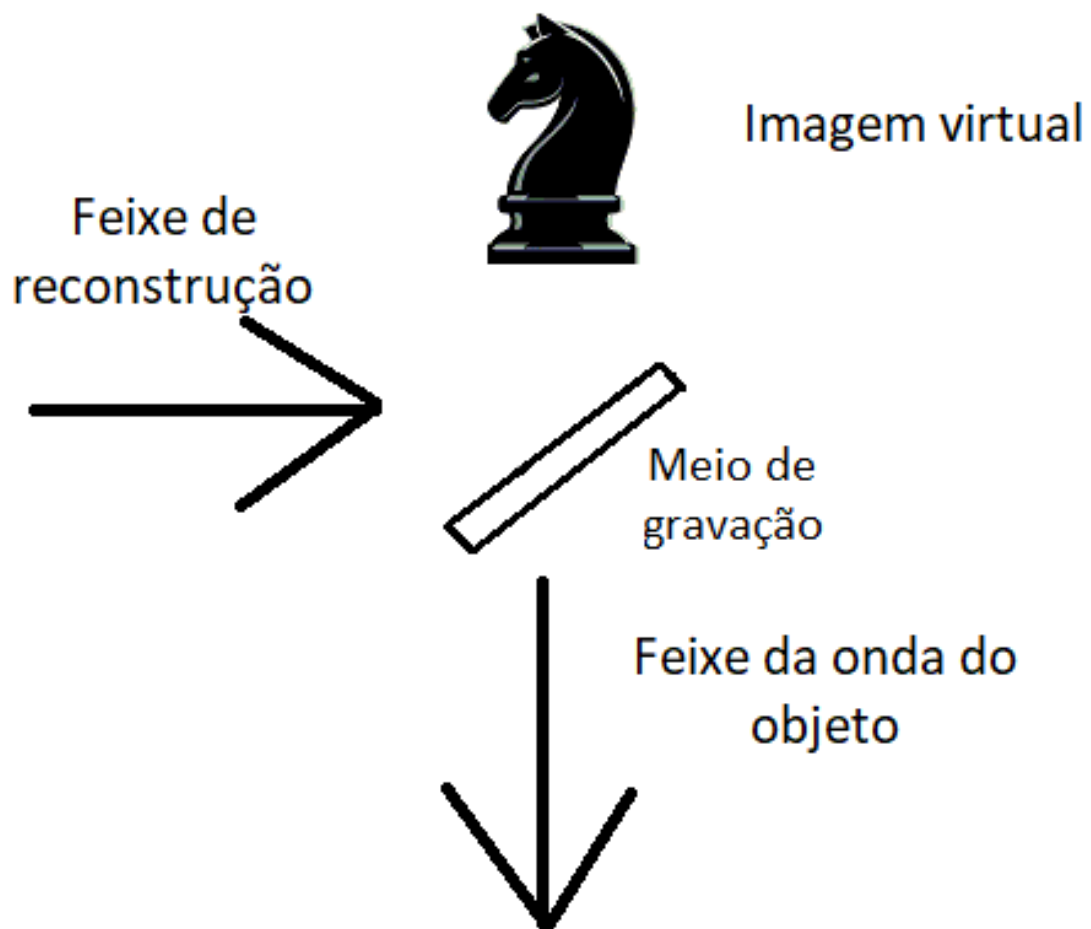


Figura 2.6: Reconstrução da imagem.



Figura 2.7: Exemplo de reconstrução da imagem.

O processo holográfico abordado acima pode ser descrito matematicamente através do formalismo adotado na secção 2.1 para a descrição da luz. A amplitude complexa da onda do

objeto é descrito como

$$E_O(x, y) = a_O(x, y) \exp(i\varphi_O(x, y)), \quad (2.26)$$

com a amplitude real a_o e fase φ_o .

Da mesma forma, a amplitude complexa da onda de referência equivale à amplitude complexa da onda do objeto com a diferença que a amplitude real é denotada por a_R e a fase é denotada por φ_R .

$$E_R(x, y) = a_R(x, y) \exp(i\varphi_R(x, y)). \quad (2.27)$$

Ambas as ondas se interferem na superfície do meio da gravação. A intensidade é então calculada por

$$I(x, y) = |E_O(x, y) + E_R(x, y)|^2. \quad (2.28)$$

A transmissão de amplitude $h(x, y)$ da chapa fotográfica revelada(ou de outro meio de gravação) é proporcional à intensidade luminosa $I(x, y)$. A expressão da transmissão de amplitude é dada por

$$h(x, y) = h_0 + \beta\tau I(x, y). \quad (2.29)$$

A constante β representa o rácio entre a transmitância de amplitude e a característica de exposição do material sensível à luz. Para o caso das emulsões fotográficas o β é negativo. O τ é o tempo de exposição e o h_0 representa a amplitude de transmissão da placa não exposta, podendo ser negligenciada com a substituição da placa por um CCD(Charge Coupled Device) para gravação do holograma, entrando assim nos domínios da holografia digital.

Para a reconstrução da imagem a partir do holograma a amplitude de transmissão deve ser multiplicado pela amplitude complexa do feixe de reconstrução(feixe de referência) como visualizado na equação abaixo.

$$E_R(x, y)h(x, y) = [h_0 + \beta\tau(a_R^2 + a_O^2)]E_R(x, y) + \beta\tau a_R^2 E_O(x, y) + \beta\tau E_R^2(x, y)E_O^*(x, y). \quad (2.30)$$

O primeiro termo do lado direito da equação é a referência do feixe multiplicado pelo fator. Isso representa as ondas não difratadas que passam pelo holograma(Ordem de difração 0). O segundo termo é a reconstrução da onda objeto, que forma a imagem virtual. O fator real $\beta\tau a_R^2$ apenas influencia no brilho da imagem. O terceiro termo gera uma distorção da imagem real do objeto. Para o caso da holografia fora do eixo a imagem virtual, a imagem real e a onda não difratada são separadas, como será presenciado no capítulo da holografia computacional.

A razão para a distorção da imagem real do objeto se resume ao fator de variação complexa espacial E_R^2 , ao qual modula a imagem que forma a onda do objeto conjugado E_O^* . Para que uma imagem real não distorcida seja formada no processo de reconstrução deve ser usado um feixe de referência conjugado E_R^* para a reconstrução da imagem.

Holografia computacional

3.1 GERAL

No capítulo anterior foram abordados princípios básicos da holografia, desde do fenômeno ondulatório da luz até ao processo de gravação e reconstrução da imagem original a partir do holograma gravado num meio de gravação. Nesse capítulo em concreto se fará uso dos princípios básicos da holografia para gerar hologramas e suas respectivas reconstruções das imagens originais através de meios computacionais com uso de alguns algoritmos e pelo conhecimento de alguns tipos de hologramas utilizados para simulação computacional de hologramas.

Esse capítulo incidirá somente na simulação computacional de vários algoritmos de geração de hologramas e suas respectivas reconstruções das imagens originais, assim como os variados tipos de hologramas existentes e suas propriedades, sendo assim, neste capítulo não será especificado qual o algoritmo usado para a realização do projeto como o todo. Trata-se assim de um capítulo que visa apenas uma explicação dos vários algoritmos e seus respectivos tipos de hologramas usados para simulação computacional e para geração de hologramas ao qual será na prática aplicado a um modulador espacial de luz (SLM) com o objetivo de projetar a imagem reconstruída do holograma em tempo real tal como pretendido na realização final do projeto.

Este capítulo foi baseado na pesquisa referente à fonte bibliográfica indicada como [1] nos capítulos **Types of holograms** e **Computer-generated holography**.

3.2 TIPOS DE HOLOGRAMAS

Nessa seção será abordado alguns tipos básicos de hologramas e seus princípios de gravação e reconstrução holográfica. Em alguns tipos de hologramas serão mostradas simulações em matlab da gravação do holograma e da reconstrução da respectiva imagem original.

3.2.1 Holograma de Gabor

O holograma de Gabor trata-se de um tipo de holograma no qual a onda de referência e a onda do objeto se encontram em paralelo no momento da gravação. O holograma em si é criado através da iluminação do objeto no qual se pretende gerar o holograma e atrás desse objeto se encontra um material fotossensível ou então um dispositivo como o CCD que é colocado como o meio de gravação do holograma como descrito na figura 3.1.

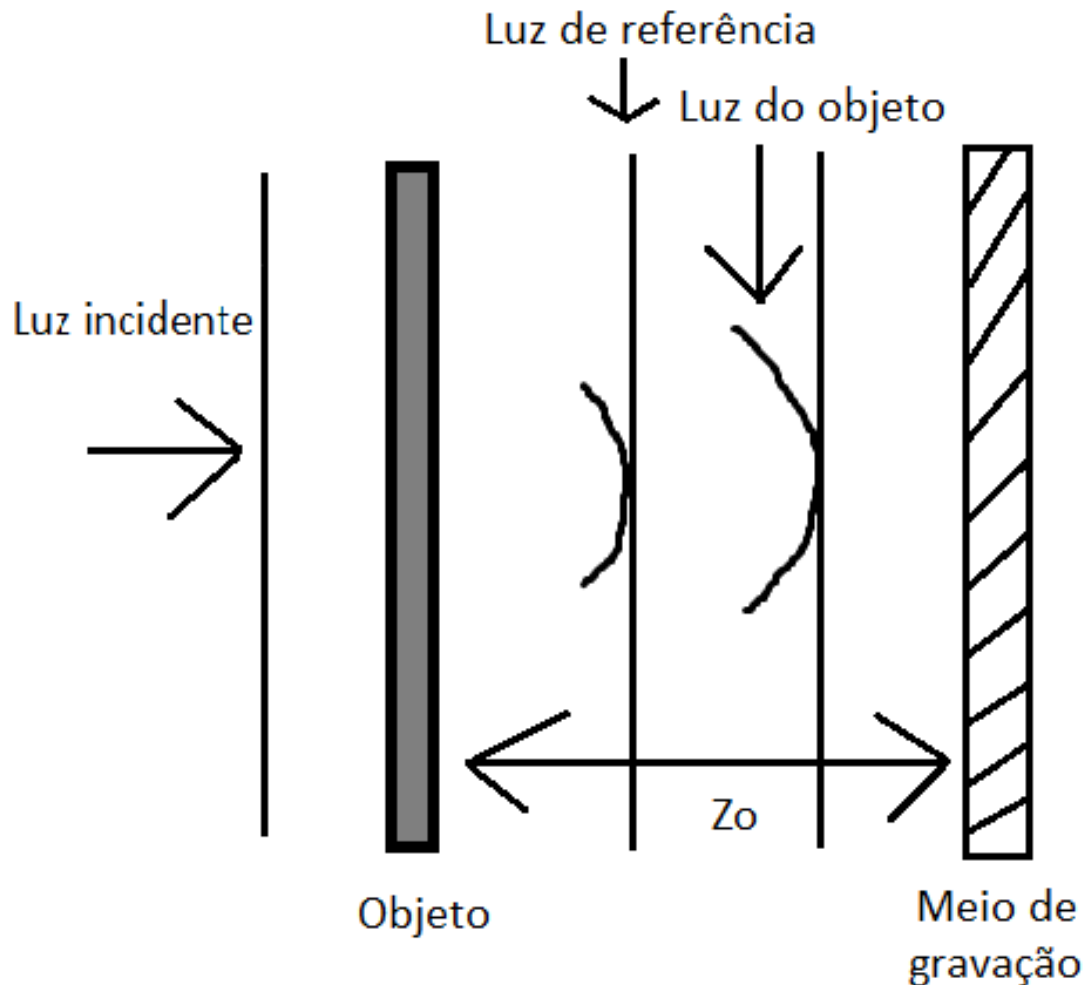


Figura 3.1: Gravação do holograma através do método de Gabor.

Para a reconstrução da imagem original do objeto no meio de gravação basta incidir o mesmo com luz, formando a imagem virtual atrás do meio de gravação a uma distância Z_0 do mesmo tal como o descrito na figura 3.2, que apresenta esquematicamente uma bola como sendo a imagem da reconstrução objeto. Tal como a imagem virtual, a imagem real, que se encontra no lado do observador, é formada à mesma distância Z_0 da imagem virtual.

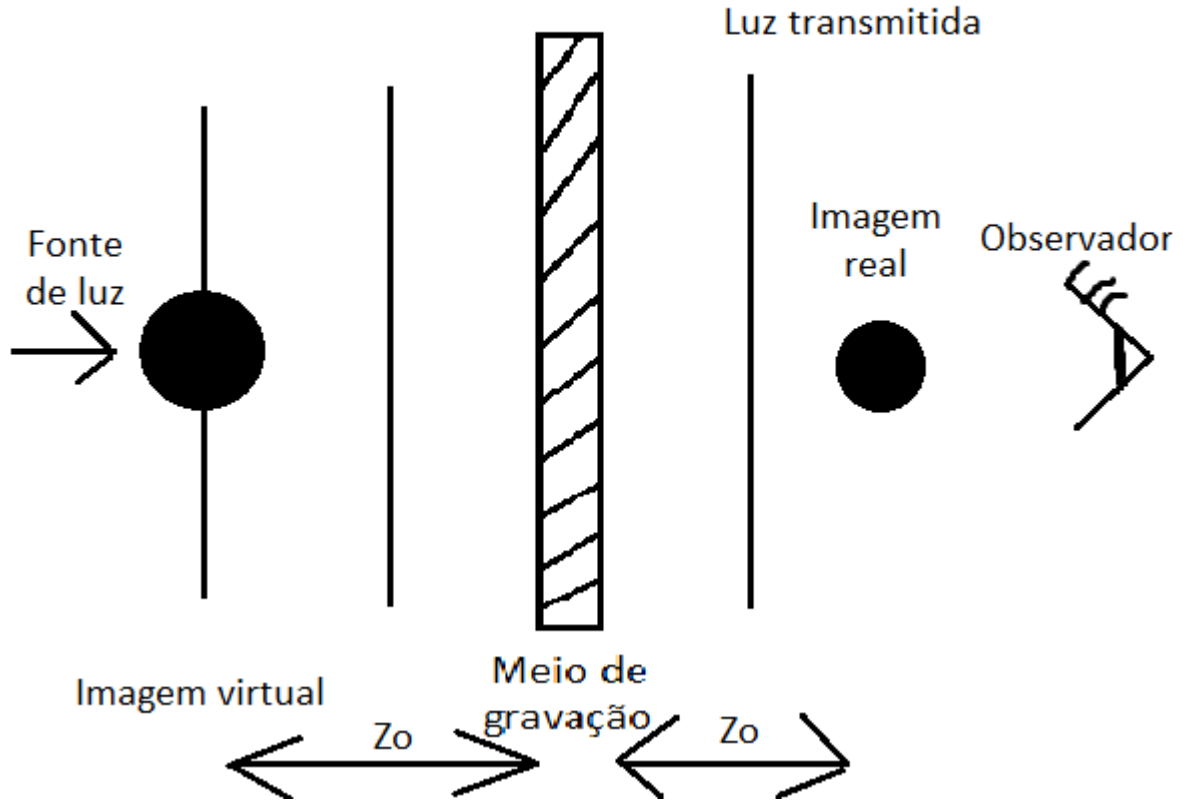


Figura 3.2: Reconstrução da imagem do holograma de Gabor.

O método de Gabor somente é efetivo quando o objeto é limitado por uma pequena flutuação $\Delta(x, y)$ da transparência uniforme σ_0 para dessa forma o objeto de onda espalhado pelo objeto não disturba a referência uniforme da onda excessivamente. Dado que a transparência total do objeto $\sigma(x, y)$ é dado pela expressão

$$\sigma(x, y) = \sigma_0 + \Delta(x, y), \quad (3.1)$$

com a condição de que $\Delta(x, y) \ll \sigma_0$. Dessa forma o campo complexo no plano do holograma, a uma distância Z_0 do objeto, pode ser expressa por

$$u(x, y) = A[\sigma_0 + \delta(x, y)] * h(x, y; Z_0) = \psi_c + \psi_0(x, y), \quad (3.2)$$

onda A é a amplitude da luz incidente e $h(x, y; Z_0)$ é a resposta impulsional no espaço. A variável ψ_c é o campo de transmissão uniforme, ao qual serve como luz de referência e o campo espalhado, ψ_0 , como a amplitude de flutuação considerada como o campo do objeto. Em resultado disso, a intensidade do plano do holograma ou o holograma com a amplitude de transmitância pode ser expressa como

$$t(x, y) = |\psi_c + \psi_0(x, y)|^2 = |\psi_c|^2 + |\psi_0|^2 + \psi_c \psi_0^*(x, y) + \psi_c^* \psi_0(x, y). \quad (3.3)$$

No processo de reconstrução, o holograma é iluminado pela onda plana da luz de reconstrução. O campo complexo que emerge do holograma é então proporcional a $t(x,y)$.

O termo $\psi_c\psi_0^*(x,y)$ é responsável pela reconstrução da imagem real, o complexo conjugado do objeto(imagem gémea) se encontra a uma distância de Z_0 atrás do holograma.

$$\psi_c\psi_0^*(x,y) * h(x,y) = \psi_c[A\Delta(x,y) * h(x,y;Z_0)]^* * h(x,y) \propto \Delta^*(x,y). \quad (3.4)$$

De modo semelhante, o termo $\psi_c^*\psi_0(x,y)$ é responsável pela reconstrução da imagem virtual da amplitude de flutuação $\Delta(x,y)$ na localização do objeto. Depois de retropropagar à distância de Z_0 , tem-se

$$\psi_c^*\psi_0(x,y) * h(x,y) = \psi_c^*[A\Delta(x,y) * h(x,y;Z_0)] * h(x,y) \propto \Delta^*(x,y). \quad (3.5)$$

Os primeiros dois termos da equação 3.3 correspondem à transmissão do feixe de ordem 0, que é dado por

$$[|\psi_c|^2 + |\psi_0(x,y)|^2] * h(x,y;z), \quad (3.6)$$

onde z é qualquer distância atrás do holograma.

A vantagem do método de Gabor se encontra na simplicidade da configuração ótica para a gravação do holograma e é possível produzir o holograma de Gabor através do uso de uma fonte de luz de baixa coerência. Por um outro lado, dado que os campos de luz se propagam juntos na mesma direção no estágio de reconstrução, eles são então observados simultaneamente e a imagem reconstruída é sempre borrada pela transmissão de ordem 0 da luz e da imagem gémea. Um outro problema no holograma de Gabor é o facto de que a amplitude de flutuação do objeto deve ser muito pequeno, $\Delta(x,y) \ll \sigma_0$, para tornar a técnica útil.

3.2.2 Holograma fora do eixo

Contrariamente ao método de Gabor, ao qual a luz de referência e a luz incidente se propagam na mesma direção, no método da holografia fora do eixo, também chamado de off-axis no inglês, a luz de referência e a luz incidente se encontram a percorrer em direções distintas, dado que a luz de referência vai ser usada como uma onda plana com um ângulo de offset, θ , em relação ao meio da gravação (ver figura 3.3).

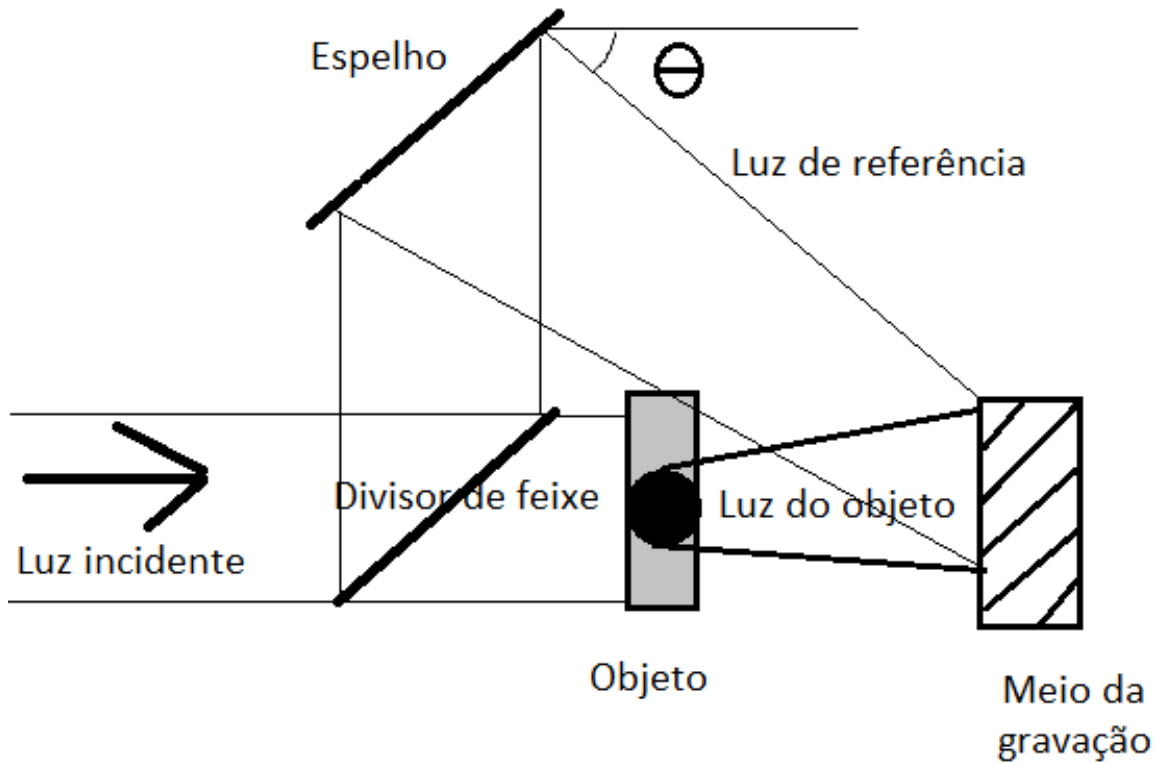


Figura 3.3: Gravação do holograma do tipo fora do eixo.

Assumindo que a propagação do vetor de referência da luz se encontra no plano x-z, o holograma do tipo fora do eixo pode ser expresso como

$$t(x, y) = |\psi_0(x, y) + \psi_r e^{jk_0 \sin(\theta)x}|^2, \quad (3.7)$$

ou também pode ser expressa de forma equivalente como

$$t(x, y) = |\psi_0(x, y)|^2 + |\psi_r|^2 + 2|\psi_0^*(x, y)\psi_r| \cos[2\pi f_x x + \phi(x, y)], \quad (3.8)$$

onde $\psi_0(x, y)$ e $\psi_r e^{jk_0 \sin(\theta)x}$ são amplitudes complexas da onda objeto e da onda de referência, respectivamente, no meio da gravação. O termo $\phi(x, y)$ representa o ângulo de fase do $\psi_0^*(x, y)\psi_r$ como $\psi_0^*(x, y)\psi_r = |\psi_0^*(x, y)\psi_r| e^{jk_0 \sin(\theta)x}$. Por fim, $f_x = \frac{\sin(\theta)}{\lambda}$ é chamado de frequência portadora espacial do holograma, o que relembra a terminologia usada em sistemas de comunicação, onde a onda portadora é usada para transportar a mensagem na teoria de modulação.

Na reconstrução, o holograma é iluminado por uma luz de reconstrução na mesma direção que a luz de referência, como demonstrada na 3.4.

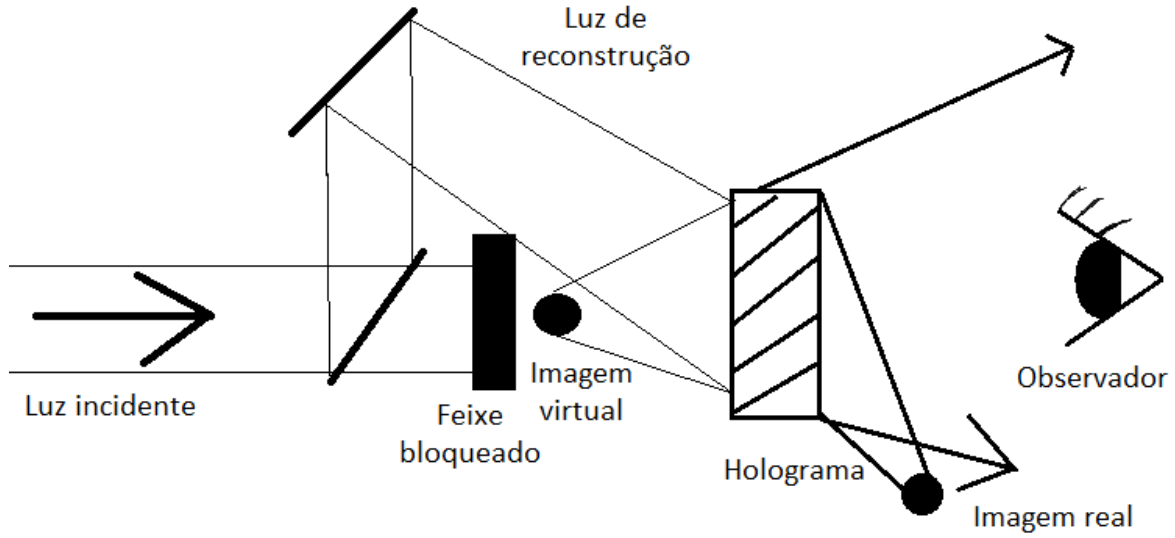


Figura 3.4: Reconstrução da imagem do holograma fora do eixo.

Assumindo que a amplitude da luz de reconstrução é a mesma que a luz de referência, o campo complexo atrás do holograma pode ser expressa como

$$\psi_r e^{jk_0 \sin(\theta x)} \times t(x, y) = |\psi_r|^2 \psi_r e^{jk_0 \sin(\theta x)} + |\psi_0(x, y)|^2 \psi_r e^{jk_0 \sin(\theta x)} + \psi_0(x, y) |\psi|^2 + \psi_0^*(x, y) \psi_r^2 e^{j2k_0 \sin(\theta x)}. \quad (3.9)$$

De forma idêntica à holografia de Gabor, os primeiros dois termos do membro direito da equação representam a luz transmitida. O terceiro termo contribui para a reconstrução virtual da imagem. Por fim, o último termo contribui para o conjugado real da imagem.

3.2.3 Holograma de imagem

Num holograma de imagem, a imagem real do objeto é formada num meio de gravação através do uso de uma lente como indicado na figura 3.5. A luz que emerge da imagem real serve como onda do objeto. Matematicamente, na fase da gravação do holograma, a imagem do holograma é dada pela expressão

$$t(x, y) = |\psi_i(x, y) + \psi_r e^{jk_0 \sin(\theta x)}|^2, \quad (3.10)$$

onde $\psi_i(x, y)$ representa o campo complexo da imagem real do holograma e $\psi_r e^{jk_0 \sin(\theta x)}$ é a referência fora do eixo da onda plana. Relativamente à reconstrução e assumindo que a amplitude da reconstrução da luz é a mesma que a referência da luz, como demonstrada na figura 3.6, o campo complexo atrás do holograma pode ser expresso por

$$t(x, y) \times \psi_r e^{jk_0 \sin(\theta x)} = [|\psi_i|^2 + |\psi_r|^2] e^{jk_0 \sin(\theta x)} + \psi_i(x, y) \psi_r^* + \psi_i^*(x, y) \psi_r e^{jk_0 \sin(\theta x)}. \quad (3.11)$$

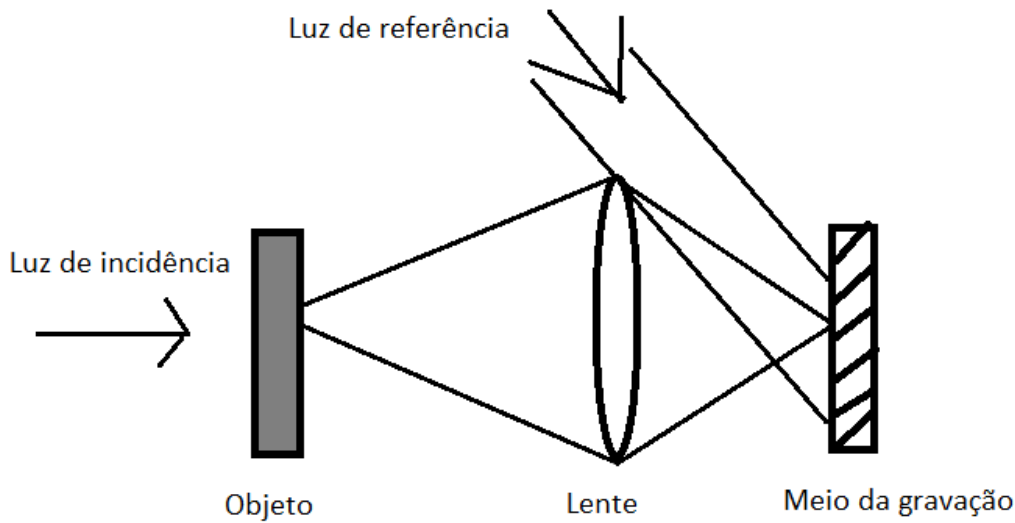


Figura 3.5: Gravação do holograma de imagem.

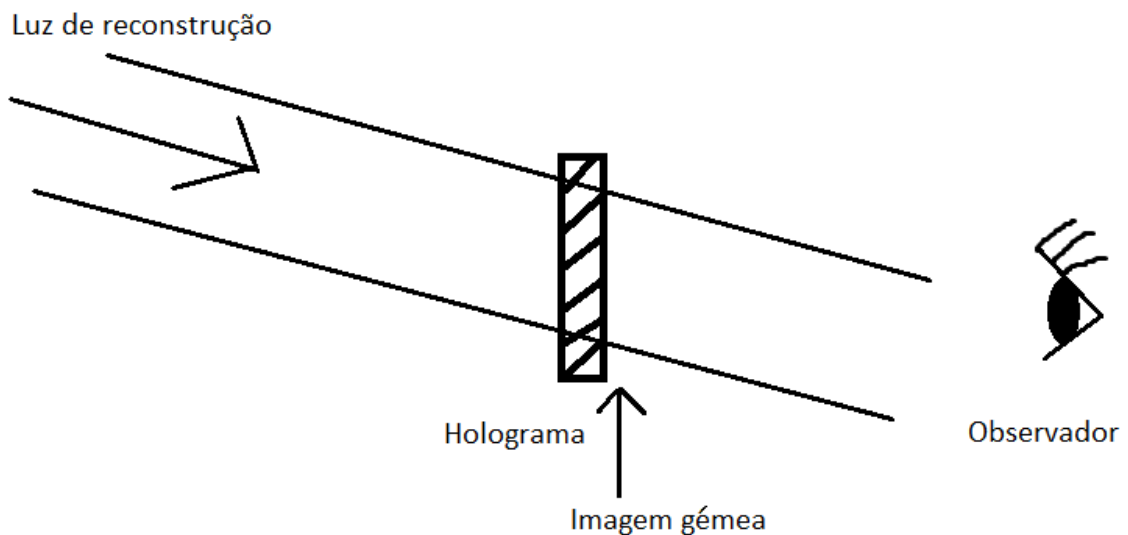


Figura 3.6: Reconstrução da imagem do holograma de imagem.

O primeiro termo do lado direito representa a propagação do feixe de ordem 0 na direção do feixe de reconstrução. O segundo termo representa imagem reconstruída e se situa no plano do holograma. O terceiro termo representa a imagem gémea, ao qual é também reconstruída no plano do holograma, no entanto este propaga na direção que é 2θ afastado do eixo de z , no entanto o observador vê a reconstrução da imagem ao longo de z .

Não é possível de observar a imagem reconstruída do holograma convencional fora do eixo(off-axis) através do uso de uma fonte policromática de luz, como uma lâmpada por exemplo, pois existe aberração cromática no holograma.

É possível verificar através da simulação em matlab que a aberração cromática é proporcional à distância entre o objeto e o plano do holograma.

Para demonstração da aberração cromática na reconstrução da imagem do holograma através da luz branca, apenas foi considerado a difração da imagem virtual de um holograma fora do eixo. A luz de ordem 0 e a imagem gêmea são ignoradas nessa simulação. A simulação será de um objeto que representa a letra D e H e contém 256×256 pixels, sendo que cada um desses pixels possui o tamanho de $50\mu m$ e o comprimento de onda é de $0.65\mu m$. O ângulo de offset é de 10° .

Nessa simulação vão ser usados vários comprimentos de onda no intervalo de $0.45\mu m$ até $0.65\mu m$, com a diferença de $0.005\mu m$ cada comprimento de onda, o que forma um total de 41 imagens reconstruídas. Quando sobrepostas, simula uma imagem reconstruída com a luz branca. Para verificar que a aberração cromática varia com a distância quando usado luz policromática, será usado como distância da imagem reconstruída ao holograma 1cm, 5cm e por fim 15cm e como tal será verificado que há medida que a distância aumenta maior será visível o efeito da aberração cromática na imagem, como o descrito na figura 3.7.

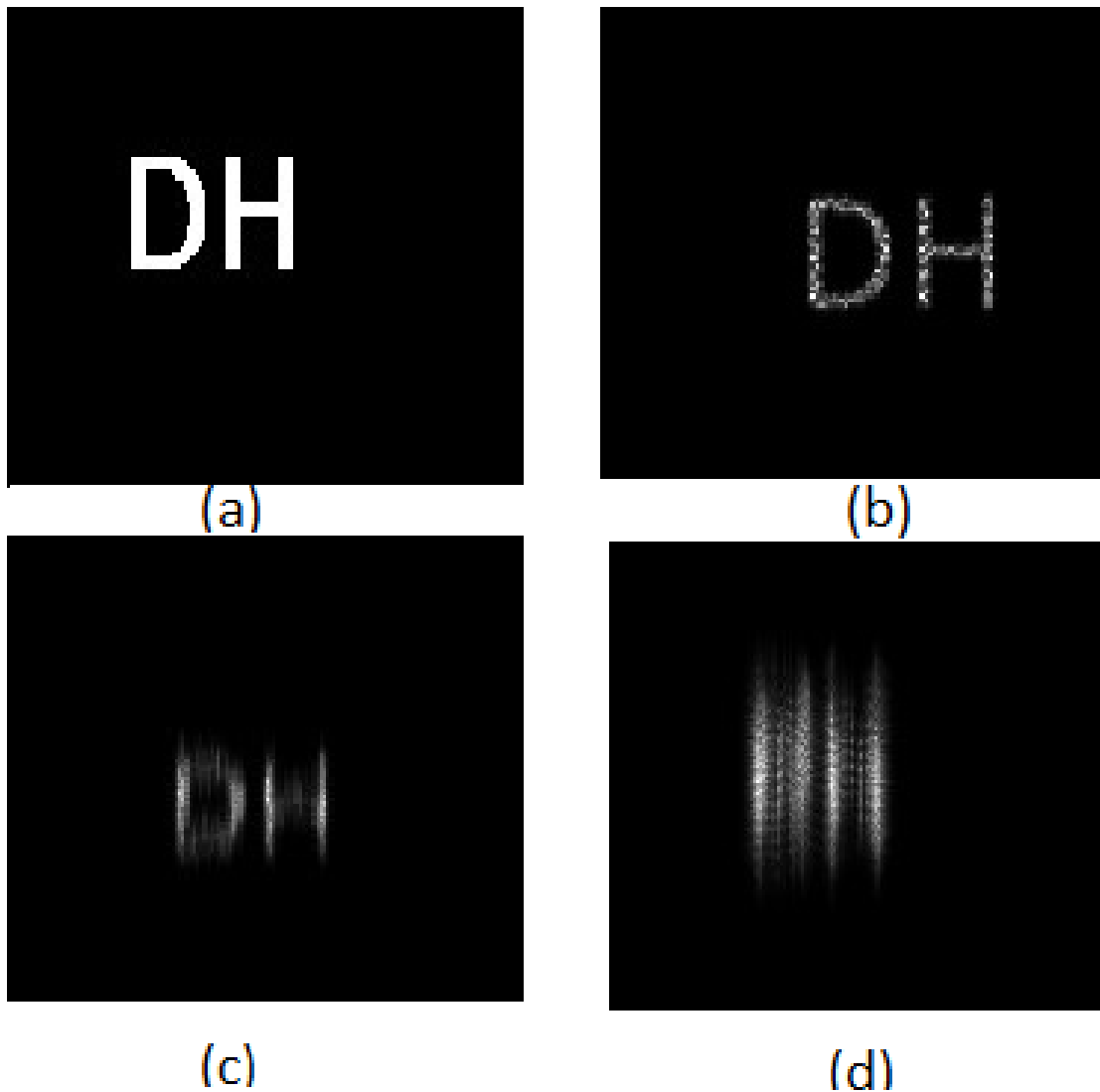


Figura 3.7: Objeto original (a) e a imagem reconstruída a partir da luz branca com as distâncias entre a imagem e o holograma de (b)1cm, (c)5cm e (d)15cm.

E dessa forma fica demonstrado que não só esse tipo de holograma apresenta aberrações cromáticas quando usado luz policromática como a mesma se torna mais visível à medida que a distância entre a imagem reconstruída e o holograma é maior, deixando a imagem ilegível. O algoritmo de matlab usado nesta simulação se encontra representada no trecho de código 1, nos anexos.

3.2.4 Holograma de Fourier

Um holograma pode ser chamado de holograma de Fourier, caso a onda do objeto no plano do holograma seja descrito pela transformada de Fourier do objeto. Os hologramas de Fourier podem ser gravadas a partir de uma configuração ótica como a indicada na figura 3.8. Nessa configuração o objeto é fixado na frente da lente 2, sendo a parte de trás do plano dessa lente a transformada de Fourier da luz do objeto. Entretanto, o ponto de luz focado ao lado do objeto, através do uso de lente 1 é usado como a luz de referência, criando uma onda plana inclinada no meio da gravação.

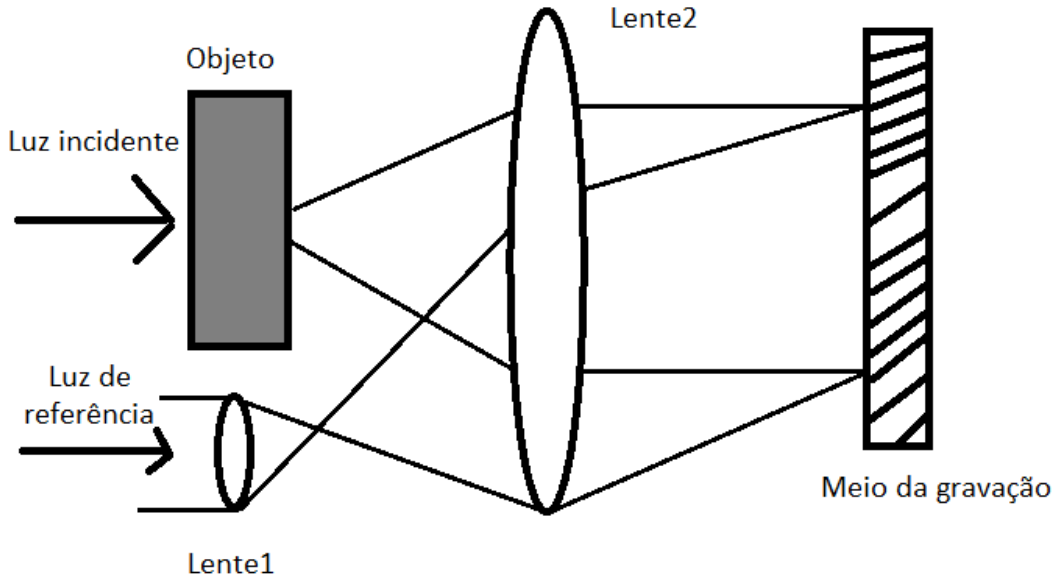


Figura 3.8: Gravação do holograma de Fourier.

O campo ótico total na parte de trás do plano focal da lente 2 pode ser expressa por

$$\psi_t(x, y) = \mathcal{F}\{\sigma_0(x, y) + A\delta(x - x_0, y)\} = \sum_0\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) + Ae^{\frac{jk_0x_0x}{f}}, \quad (3.12)$$

onde $\sigma_0(x, y)$ é a amplitude da transmitância do objeto e $\sum_0(kx, ky)$ é a transformada de Fourier do $\sigma_0(x, y)$. O termo $\delta(x - x_0, y)$ representa a referência da luz em $x = x_0, y=0$. A variável f é o comprimento focal da lente 2 e A é a amplitude da luz de referência. Por consequência o holograma de Fourier pode ser expresso como

$$t(x, y) = \left|\sum_0\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right)\right|^2 + |A|^2 + \sum_0\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) \times A^* \exp\left(\frac{-jk_0x_0x}{f}\right) + \sum^*\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) \times A \exp\left(\frac{jk_0x_0x}{f}\right). \quad (3.13)$$

No processo de reconstrução, o holograma colocado na frente do plano focal da lente é iluminado com uma onda plana de amplitude unitária como mostrada na figura 3.9.

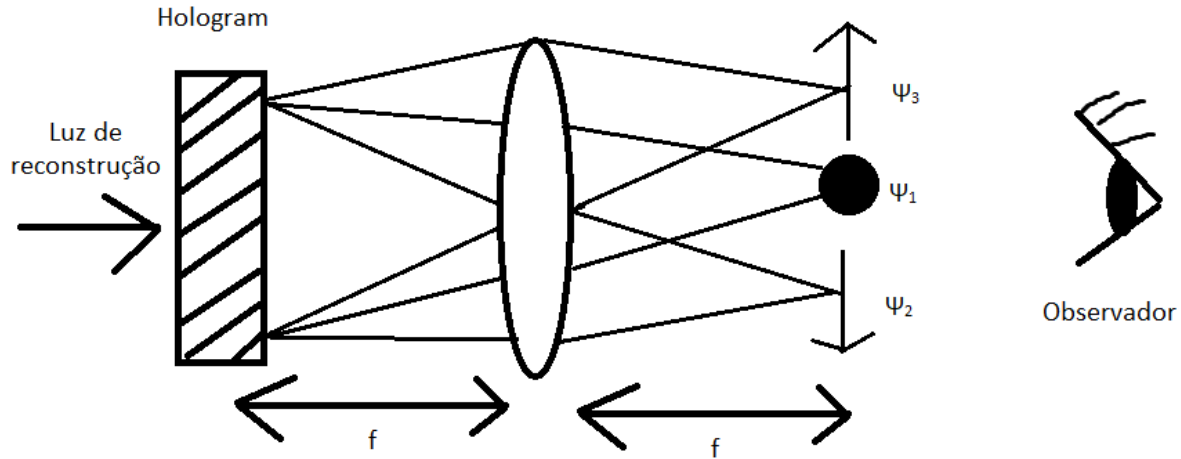


Figura 3.9: Reconstrução da imagem do Holograma de Fourier.

O campo complexo para trás do plano focal da lente contém termos. Os 3 termos são expressos por:

$$\psi_1(x, y) = \frac{f^4}{k_0^4} \sigma_0(-x, -y) \otimes \sigma_0(-x, -y) + |A|^2 \sigma(x, y), \quad (3.14)$$

$$\psi_2(x, y) = \frac{f^2}{k_0^2} A^* \sigma_0(-x + x_0, -y), \quad (3.15)$$

$$\psi_3(x, y) = \frac{f^2}{k_0^2} A \sigma_0^*(x + x_0, y). \quad (3.16)$$

O termo ψ_1 representa um feixe de ordem zero localizado em $x=0, y=0$ atrás do plano focal. O termo ψ_2 representa a reconstrução da imagem invertida em $x=x_0, y=0$ e o termo ψ_3 representa o conjugado da imagem em $x=-x_0, y=0$. Dessa forma, ψ_1, ψ_2 e ψ_3 podem ser separados um do outro.

Para demonstrar melhor o que acontece num holograma de Fourier decorreu-se à simulação da gravação do holograma de Fourier e reconstrução da respetiva imagem original. Primeiro o campo complexo do padrão do objeto no plano focal pode ser obtido através da realização da transformada de Fourier. Depois, a onda plana inclinada é adicionada ao campo complexo como sendo a luz de referência produzindo o holograma de Fourier como o descrito na figura 3.10. O holograma de Fourier pode ser reconstruído aplicando a transformada de Fourier. A imagem reconstruída é a demonstrada na figura 3.11. Na imagem reconstruída será possível ver ambas as imagens, a reconstruída e a imagem gémea no plano de reconstrução. A separação entre as 3 ordens de difração no plano de reconstrução depende do ângulo de offset da luz de referência, caso esse ângulo de offset seja muito pequeno isso eventualmente resultará numa diafonia(interferência indesejada) entre as ordens de difração adjacentes.

A demonstração foi realizada a partir da imagem com as letras "DH" e através do código em matlab para verificar por simulação computacional o que efetivamente acontece.

Fourier Hologram

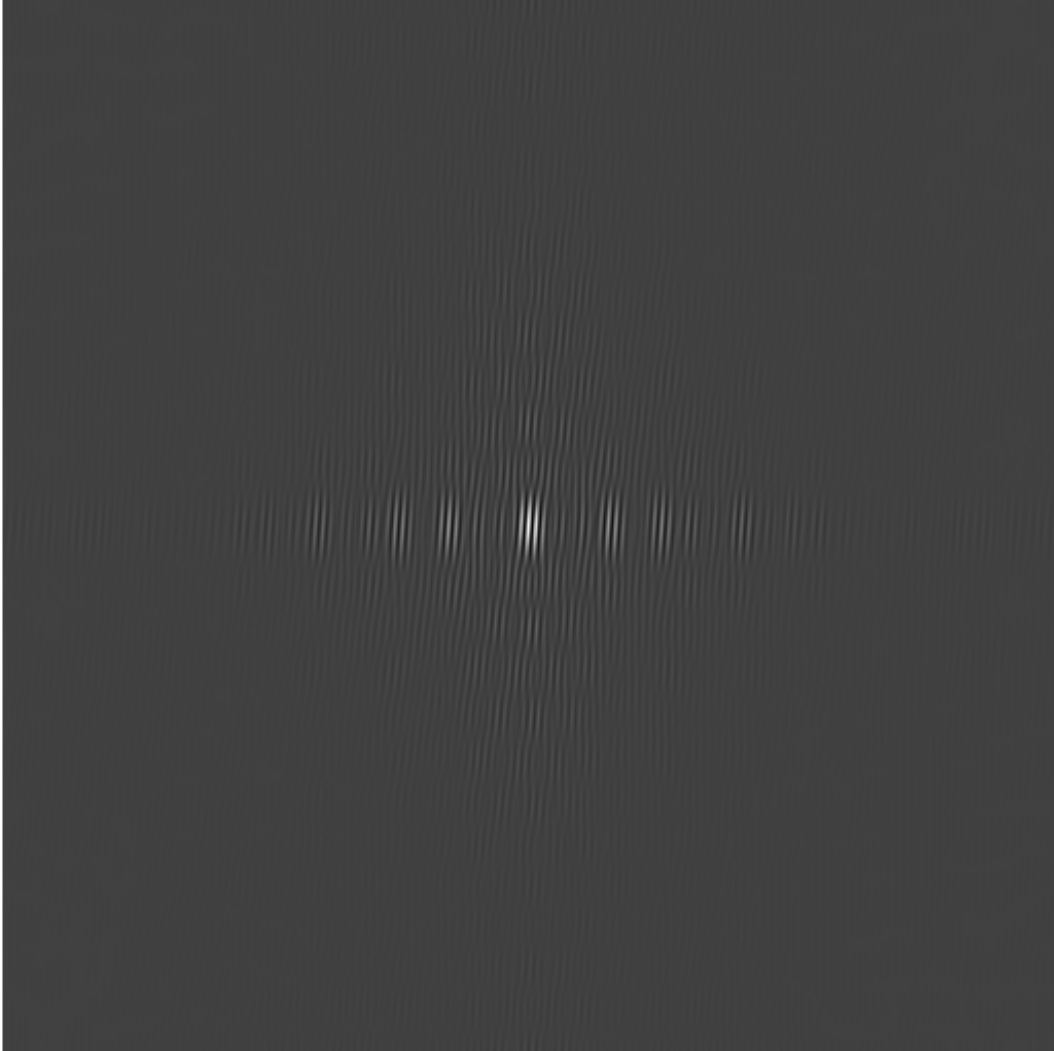


Figura 3.10: Holograma de Fourier.

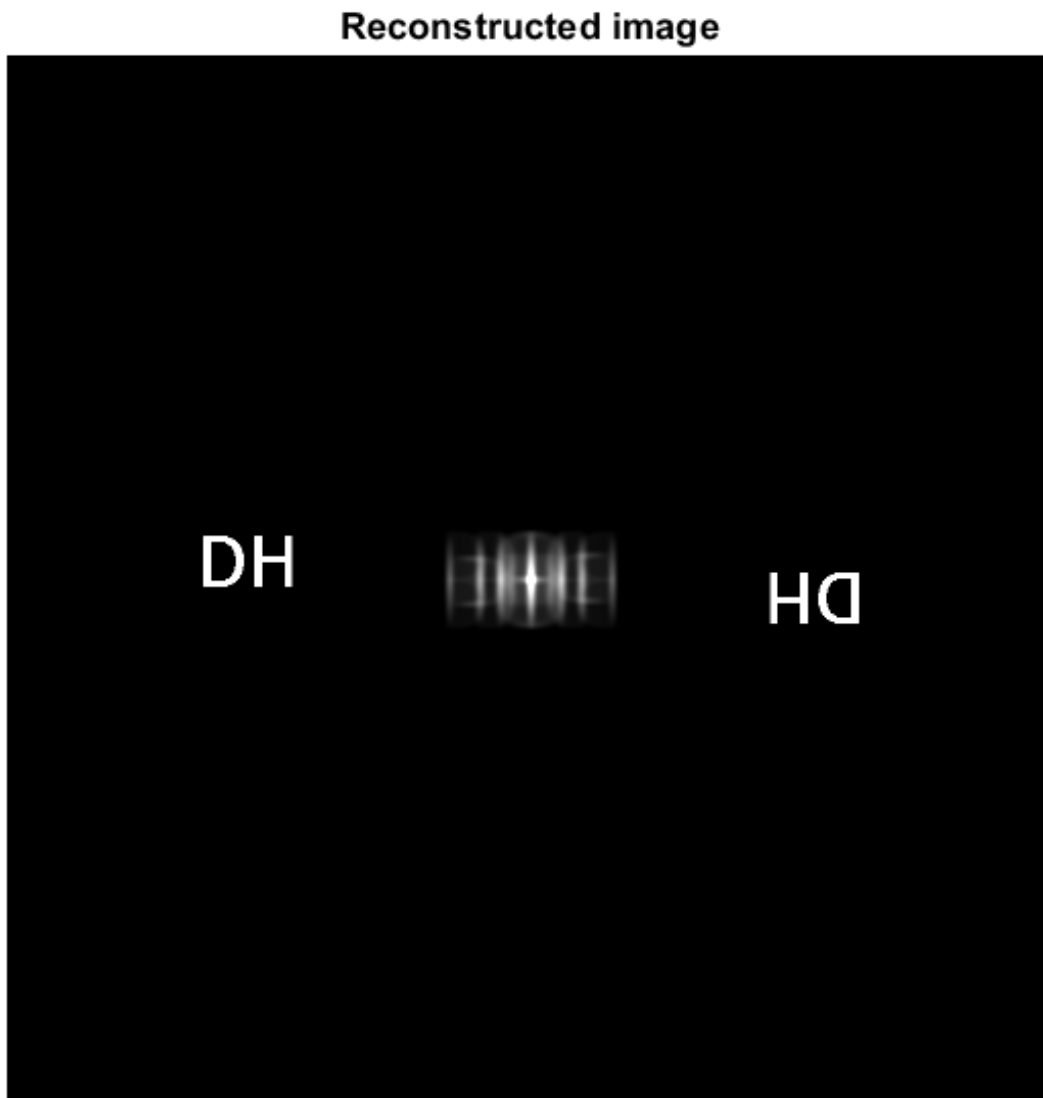


Figura 3.11: Reconstrução da imagem para o holograma de Fourier.

O código usado para a realização desta simulação se encontra no trecho de código 2, nos anexos.

3.3 HOLOGRAFIA GERADA POR COMPUTADOR

A holografia gerada por computador lida com métodos que são usados para geração de hologramas digitalmente. Um holograma pode ser impresso num filme ou então carregado num modulador espacial de luz(SLM) para reconstrução holográfica. A maior vantagem do uso da holografia gerada por computador(CGH) é o facto de que objetos tridimensionais não precisam necessariamente de existir no mundo real. Em outras palavras, pode-se dizer que tais objetos, ao qual se pretende fazer display, podem ser fictícios. Para gerar CGHs, diferentes métodos foram desenvolvidos para adequação do seu uso nos mais variados dispositivos de display e seus métodos de reconstrução.

3.3.1 Holograma de desvio de fase

O holograma de desvio de fase, também chamado de detour-phase hologram do inglês, trata-se de um holograma do tipo de Fourier e um dos primeiros métodos de geração de hologramas por computador. A figura 3.12 demonstra a configuração ótica para a reconstrução do holograma de Fourier através da transformada de Fourier da lente a uma distância focal de f .

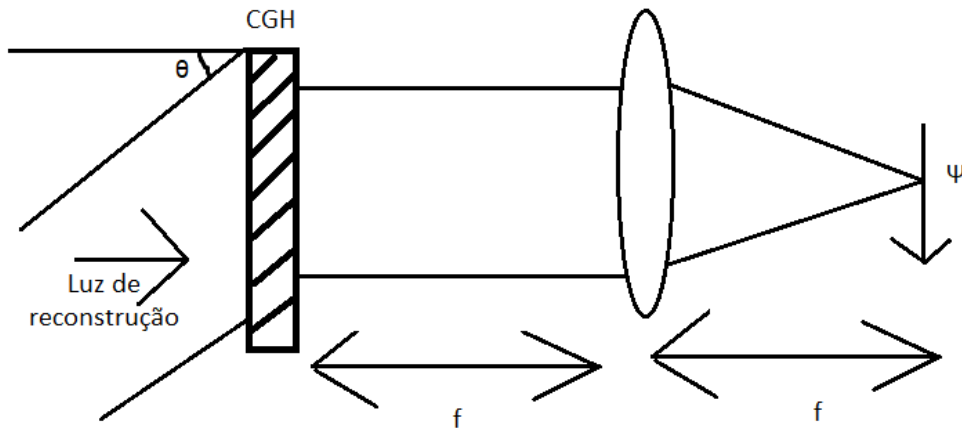


Figura 3.12: Reconstrução da imagem para o holograma de desvio de fase.

O holograma $H(x,y)$ e a amplitude complexa da imagem reconstruída $\psi(x,y)$ estão relacionadas da seguinte forma

$$\psi(x,y) = \mathcal{F}\{H(x,y)\psi_r(x,y)\}_{k_x=\frac{k_0x}{f}, k_y=\frac{k_0y}{f}}, \quad (3.17)$$

onde $\psi_r(x,y) = \exp(-jk_0 \sin(\theta)x)$ denota a amplitude complexa da onda plana da reconstrução da luz e θ é o ângulo da inclinação.

Geralmente, o holograma é uma função complexa representada por $H(x,y) = a(x,y)\exp(-j\phi(x,y))$, onde $a(x,y)$ e $\phi(x,y)$ são respetivamente o módulo e a fase do holograma. O objetivo é criar um padrão de amplitude binária $H_b(x,y)$ para aproximar do holograma complexo $H(x,y)$. É possível, considerando apenas o limite do tamanho da janela no plano de reconstrução encontrar $H_b(x,y)$ tal que

$$\mathcal{F}\{H_b(x,y)\psi_r(x,y)\} \approx \mathcal{F}\{H(x,y)\psi_r(x,y)\}, \quad (3.18)$$

com a janela. Primeiro, a área do holograma é dividida num array de células unitárias, e o tamanho de cada uma será $w \times w$, como o descrito na figura 3.13. Para o holograma complexo $H(x,y)$, pode-se considerar que tanto $a(x,y)$ e $\phi(x,y)$ variam muito devagar com qualquer célula. As células podem então ser substituídas por um array de pontos de origem, o que produz um holograma complexo amostrado através da expressão

$$H_s(x, y) = \sum_{m,n} a_m e^{-j\phi_{mn} x \sigma(x-x_m, y-y_n)}, \quad (3.19)$$

onde (m,n) indexa a célula centrada em $x_m = mw$, $y_n = nw$, $a_{mn} = a(x_m, y_n)$ e $\phi_{mn} = \phi(x_m, y_n)$. Sob iluminação da onda plana inclinada, cada ponto de origem num holograma amostrado produz uma onda plana no plano de reconstrução, sendo o campo complexo no plano de reconstrução como

$$\psi_s(x, y) = \sum_{m,n} a_{mn} e^{-j\phi_{mn}} e^{-jk_0 \sin(\theta) x_m} e^{j \frac{k_0}{f} (x_m x + y_n y)}. \quad (3.20)$$

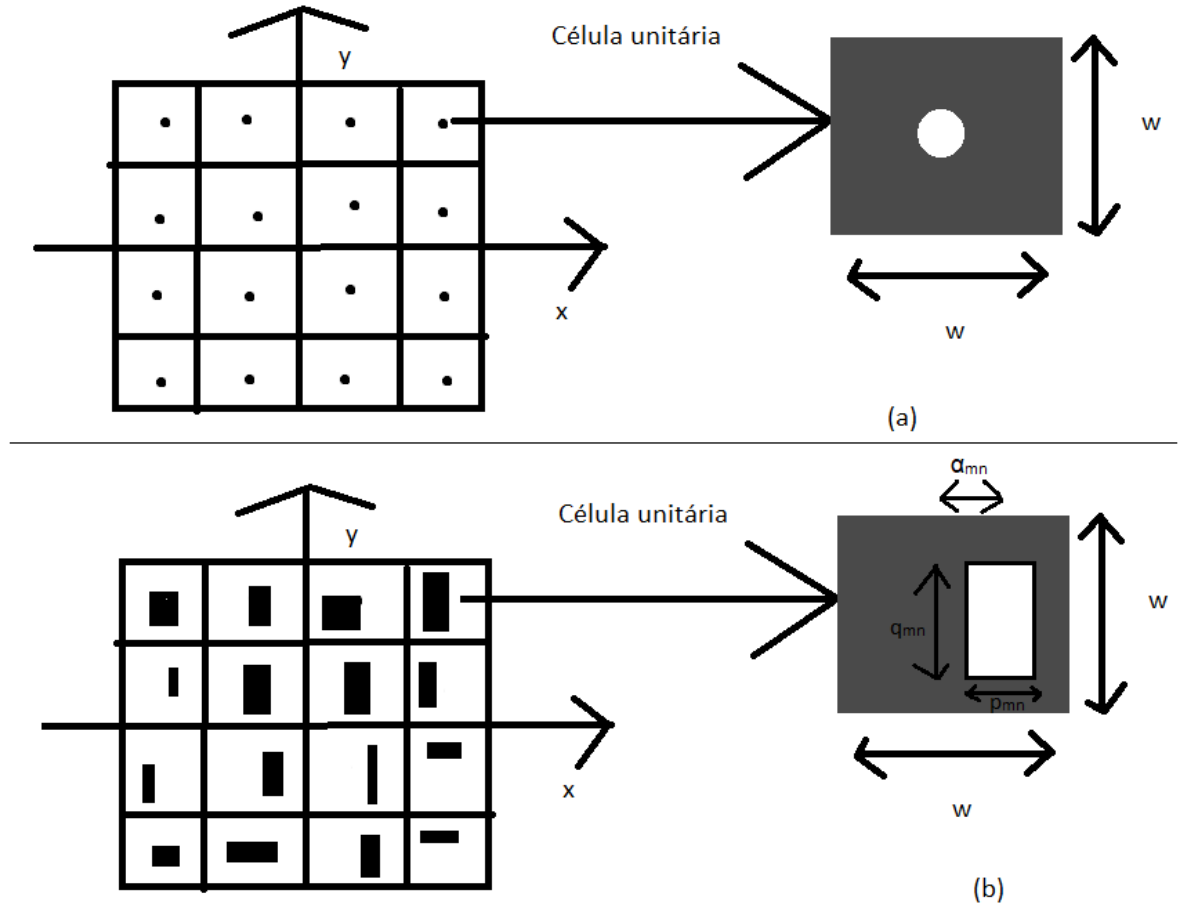


Figura 3.13: Descrição da célula em (a) como sendo o holograma complexo e (b) como o desvio de fase binária do holograma.

É possível eliminar o efeito indesejado da fase, o termo $\exp(-jk_0 \sin(\theta) x_m)$, através da projeção da largura da célula unitária de modo a que

$$w = \frac{2\pi l}{k_0 \sin(\theta)}, l \in 1, 2, 3, \dots, \quad (3.21)$$

como $\exp(-j2\pi lm) = 1$, isso implica

$$\psi_s(x, y) = \sum_{m,n} a_{mn} e^{-j\phi_{mn}} e^{j\frac{k_0}{f}(x_m x + y_n y)}, \quad (3.22)$$

isso representa a aproximação da transformada de Fourier do holograma complexo amostrado $a_{mn} e^{-j\phi_{mn}}$.

Para compreender como o holograma binário pode simular o holograma complexo amostrado $H_s(x, y)$, considere que cada célula é aberta uma abertura com a área proporcional a a_{mn} . A fase da luz pode ser controlada através do deslocamento ligeiro do centro da abertura. Esse é o princípio do holograma de desvio de fase. Para verificar esse efeito do desvio de fase, é assumido que a abertura aberta de cada célula é retangular com o tamanho de $p_{mn} \times q_{mn}$, e é deslocada por α_{mn} ao longo da direção x, como descrito na figura 3.13. Por esse modo, o holograma binário pode ser expresso como

$$H_b(x, y) = \sum_{m,n} \text{rect}\left(\frac{x - x_m - \alpha_{mn}}{p_{mn}}, \frac{y - y_n}{q_{mn}}\right). \quad (3.23)$$

Sob iluminação da onda plana inclinada, o campo complexo no plano da reconstrução é dado por

$$\psi_b(x, y) = \mathcal{F}\{exp(-jk_0 \sin(\theta)x) H_b(x, y)\}, \quad (3.24)$$

onde o termo, $exp(-jk_0 \sin(\theta)x)$ se torna unitário através da aproximação anteriormente feita. Além disso através de mais algumas aproximações é possível dizer que

$$\psi_b(x, y) \approx \sum_{m,n} p_{mn} q_{mn} e^{-jk_0 \sin(\theta) \alpha_{mn}} e^{j\frac{k_0}{f}(x_m x + y_n y)}. \quad (3.25)$$

Através da comparação entre as equações (3.22) e (3.25) é possível definir que

$$p_{mn} q_{mn} = c a_{mn}, \quad (3.26)$$

e

$$K_0 \sin(\theta) \alpha_{mn} = \phi_{mn}, \quad (3.27)$$

onde c é uma constante. Dessa forma, na janela de observação, o holograma binário pode reconstruir a imagem como um holograma complexo. Na análise feita acima, a abertura na célula era retangular, no entanto a mesma pode ser de qualquer forma, como circular por exemplo, com a área proporcional a a_{mn} . É possível através das equações (3.27) e (3.21) determinar o desvio da abertura dentro de cada célula.

$$\alpha_{mn} = \frac{\phi_{mn}}{k_0 \sin(\theta)} = \frac{\phi_{mn} w}{2\pi l}. \quad (3.28)$$

O termo ϕ_{mn} se encontra entre π e $-\pi$, e o intervalo de α_{mn} está dentro de $\pm \frac{w}{2l}$.

Agora, com toda o conhecimento teórico relativo a esta CGH, será então gerado do holograma de desvio de fase e simulado a sua reconstrução. Nessa simulação usado 16×16

pixels para a construção da célula e o holograma consiste numa resolução de 64×64 células. Desse modo o total será 1024×1024 pixels no holograma. É de notar que é necessário anexar uma máscara de fase aleatória contra o objeto de modo ampliar seu espectro, caso contrário, grande parte da energia será concentrada no feixe de ordem zero e o alcance dinâmico efetivo do holograma produzido será reduzido. A fase aleatória do objeto é transformada para o domínio de Fourier também com resolução de 64×64 . O valor de cada pixel é um complexo que vai ser codificado para um padrão binário como uma célula. O código desta simulação se encontra escrito no matlab de acordo com o trecho de código 3, nos anexos.

O módulo e a fase, como o indicado no código do matlab estão quantizados nos níveis 16 e 8, respetivamente. Foi definido que $p_{mn} = 9$ pixels e $q_{mn} = 1 \sim 16$ pixels para representar os 16 níveis do módulo. Também se define $\alpha = -4 \sim 3$ pixels para representar 8 níveis de fase. Em outras palavras, se seleccionou $l=2$ na equação (3.21) e (3.28). Através dessa configuração a abertura do orifício sempre ficará dentro da célula. O holograma de desvio de fase produzido é o demonstrado na figura 3.14, em conjunto com a imagem original e a imagem reconstruída assim como a sua ampliação. É verificado que existem várias ordens de difração da luz, resultado da grelha de células, no entanto apenas ordens de difração específicas satisfazem as assunções da análise feita, o que resulta na qualidade das imagens. É de notar que no exemplo dessa simulação a eficiência de difração é muito baixa, porque foi usado uma codificação de padrão de baixa eficiência. Dando o exemplo que o p_{mn} não pode exceder 9 pixels quando a largura da célula é cerca de 16 pixels no exemplo dessa simulação. Talvez o uso de outras codificações de padrões nesse algoritmo podem melhorar a eficiência de difração.

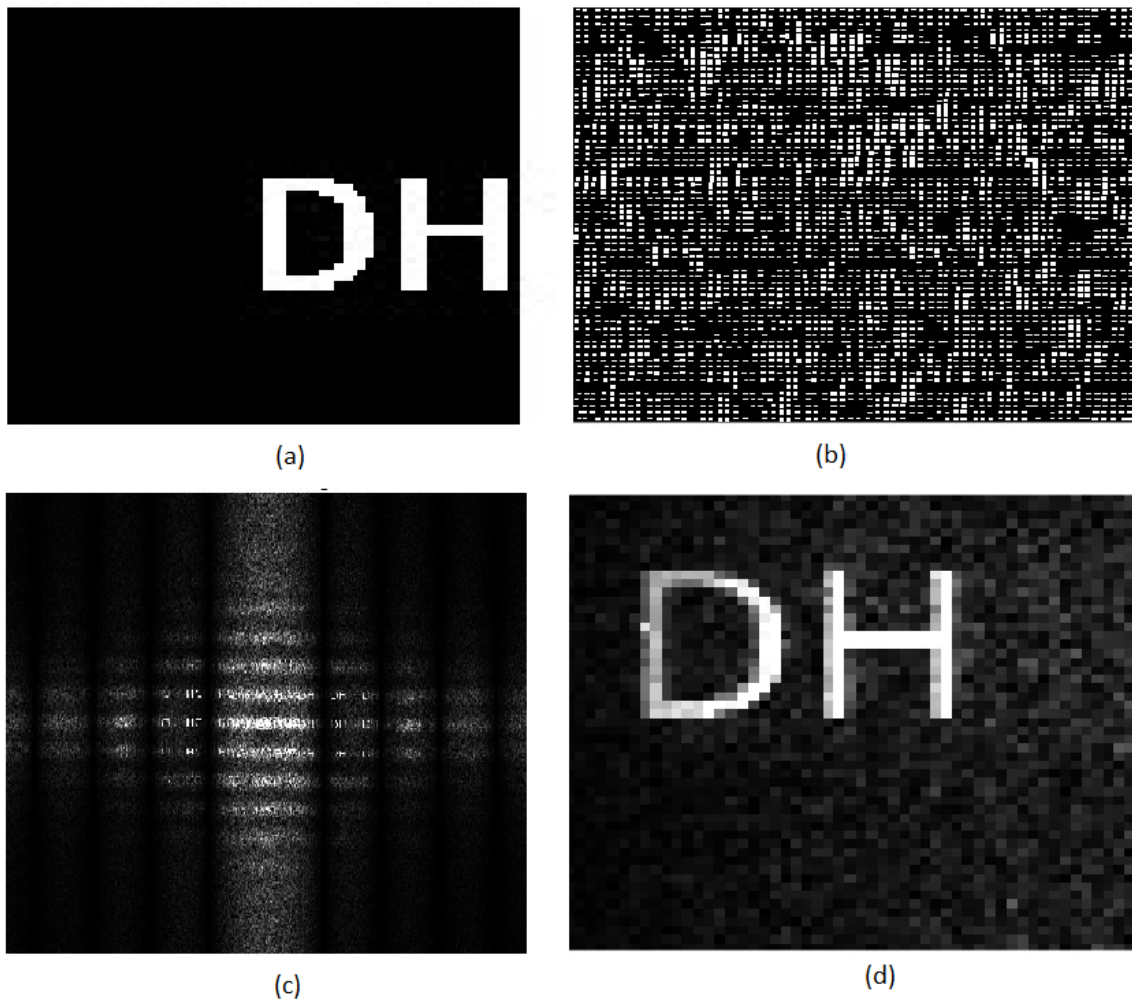


Figura 3.14: (a) Objeto original, (b) Holograma de desvio de fase, (c) Reconstrução da imagem, (d) Reconstrução da imagem ampliada com a seleção da ordem de difração de (c).

3.3.2 Holograma Kinoform

Apesar do algoritmo de desvio de fase poder representar a fase assim como o módulo do holograma complexo amostrado, a sua eficiência de difração é relativamente baixa. De modo a alcançar uma eficiência de difração mais elevada, o holograma kinoform foi proposto. O holograma kinoform é também baseado nos esquemas da holografia de Fourier. De forma geral, o holograma de Fourier é dado pela expressão $H(x, y) = a(x, y)exp[-j\phi(x, y)]$, sendo um holograma complexo. Como mencionado no exemplo da simulação anterior, uma máscara de fase aleatória é usualmente afixada contra o padrão do objeto para espalhar a energia através de um espectro amplo. Caso a energia espalhada seja uniforme, o módulo $a(x, y)$ perde a sua importância, podendo ser ignorada. Desse modo apenas é necessário a extração da sua fase $\phi(x, y)$ e produzir um padrão de tom cinza, pelo qual o nível de cinza é proporcional à fase. O padrão de tom cinza pode ser exibido num modulador espacial de luz que apenas module a fase ou impressa num filme fotográfico. Dessa forma o holograma de fase será dado pela expressão

$$H_p(x, y) = \exp[-j\phi(x, y)]. \quad (3.29)$$

O desenho de um holograma kinoform é muito simples. Idealmente, apenas existe uma ordem de difração a emitir vindo do holograma kinoform. Dessa forma o holograma kinoform pode ser entendido como um holograma do tipo dentro do eixo. Além desse pormenor, a eficiência de difração do holograma kinoform pode alcançar os 100%. Entretanto a fabricação de um holograma kinoform não é tão simples quanto o seu design. Primeiro, as franjas de tom cinza da amplitude devem ser traçadas de forma correta. Em segundo a amplitude da transmitância deve ser convertida linearmente para o atraso de fase. Por fim, o atraso de fase de ser precisa num intervalo entre 0 a 2π . Caso tal não aconteça, a eficiência de difração decai e a luz de ordem zero aparece. Em termos de holografia gerada por computador, os hologramas kinoform são fáceis de gerar, pois apenas é necessário de extrair a informação da fase vinda da campo complexo computada do objeto e exibir num modulador espacial de luz de fase. De modo a demonstrar o princípio do holograma kinoform, foi usado como padrão uma foto minha tirada no meu computador que se encontra representada na figura 3.15, sendo usada como a entrada e também se fixou uma máscara de fase aleatória contra o padrão. A máscara da fase aleatória é também mostrada na mesma figura no lado direito da imagem original representado pela letra b. A fase do espectro do objeto com a máscara da fase aleatória se encontra na figura 3.15(c). A energia se espalha pela fase aleatória no domínio espectral, o espectro é localmente não uniforme. No entanto, é assumido que o módulo uniforme é gerado no holograma kinoform, o que causará um notável ruído salpicado (conhecido também como speckle noise) na reconstrução da imagem, como descrita em 3.15(d). Tal acontece devido ao módulo não uniforme do espectro. É possível evitar o ruído através do uso específico de objetos padrão, ou alternativamente o uso de procedimentos de otimização como é o exemplo do algoritmo iterativo da transformada de Fourier.

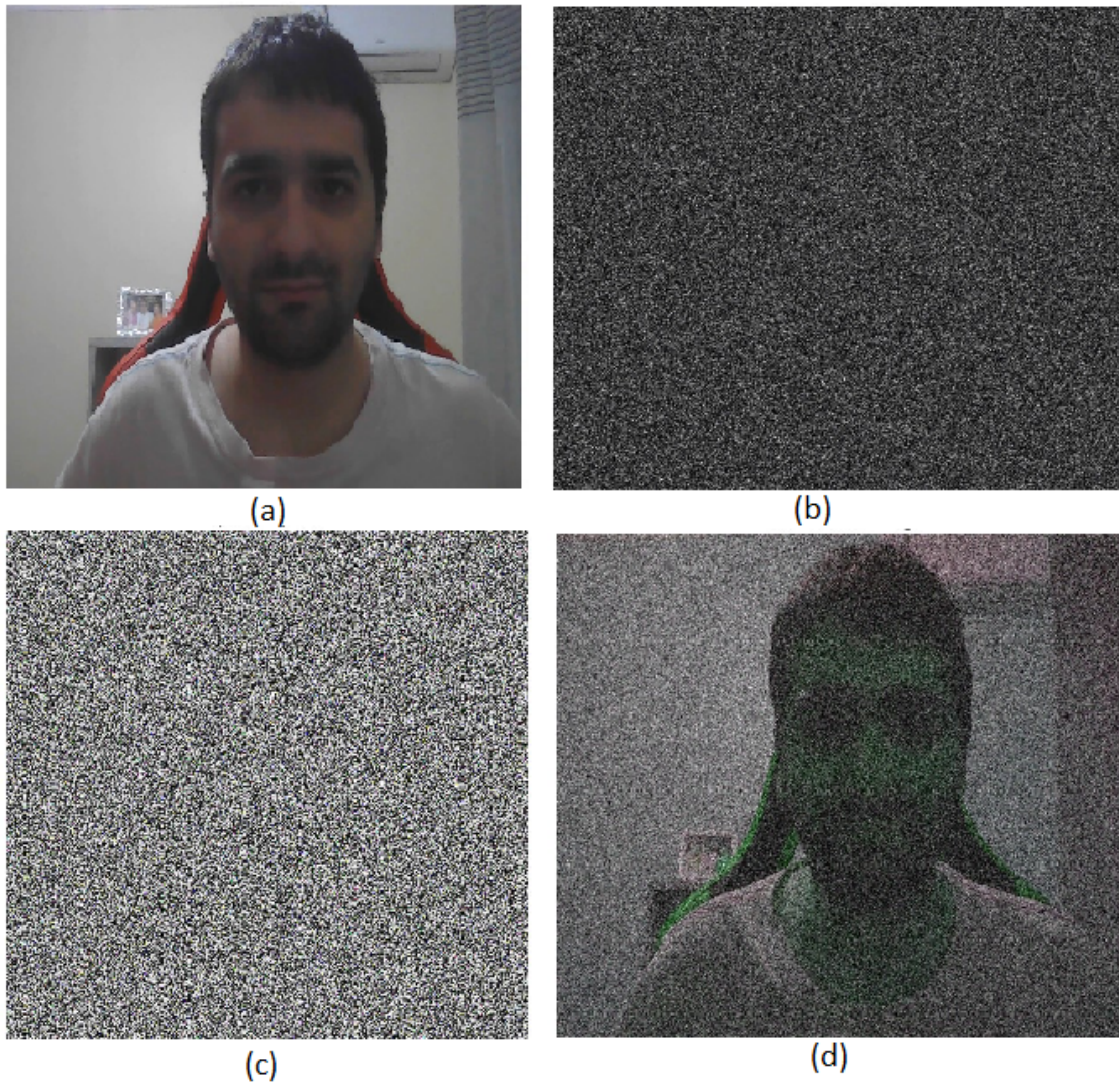


Figura 3.15: (a) Objeto original, (b) Máscara da fase aleatória com a fase entre $-\pi$ e π , (c) Espectro do objeto com máscara de fase aleatória, (d) Reconstrução da imagem do holograma kinoform.

O código usado no matlab para a execução desta simulação com o algoritmo do holograma kinoform é o que se encontra representado no trecho de código 4, nos anexos.

3.3.3 Holograma binário de Fourier

Considere duas posições deslocadas, $H_r(x-d, y_0)$ e $H_i(y+d, y_0)$ separados a uma distância de $2d$ são exibidas no plano $x_0 - y_0$, onde $H_r(x, y)$ e $H_i(x, y)$ representam respectivamente a parte real e imaginária do holograma complexo $Hc(x, y) = |Hc|e^{j\theta}$. A parte real do holograma complexo é dada por $H_r(x, y) = Re\{Hc\} + I_0$ e a parte imaginária é dada por $H_i(x, y) = Im\{Hc\} + I_0$. O termo I_0 representa uma constante positiva que influencia os valores do holograma a serem positivos, sendo que dessa forma possam ser exibidos no holograma de amplitude. A grade empregada é uma grade de amplitude dada por

$$g(x, y) = \frac{1}{2} + \frac{m}{2} \cos(2\pi \frac{x}{\Lambda}), \quad (3.30)$$

onde m é a profundidade da modulação da grade e Λ é o período da grade. Para assegurar um cadastro perfeito dos dois hologramas no plano de saída, $\Lambda = \lambda \frac{f}{d}$. Também, de forma a eliminar o termo I_0 de ser exibido no plano de saída, um filtro de densidade neutra com a placa de fase dada por $H_{dc} = mI_0 e^{j\frac{5\pi}{4}}$ no centro do plano de entrada é usado.

Para concluir o processo de binarização, primeiro se multiplica o objeto padrão, u , por uma fase aleatória e depois se calcula o campo de difração correspondida pela sua convolução com h , que representa a resposta impulsional espacial na ótica de Fourier, para alcançar o meio de gravação. Os hologramas binários são então obtidos pela binarização das partes reais e imaginárias do holograma complexo pela binarização de sinal. Ambos os dois hologramas binários são, portanto dadas por

$$B_r(x, y) = B^0\{Re\{(u \times e^{j\theta}) * h\}\}, \quad (3.31)$$

$$B_i(x, y) = B^0\{Im\{(u \times e^{j\theta}) * h\}\}, \quad (3.32)$$

onde θ representa a função de fase aleatória entre 0 e 2π e $B^0\{.\}$ representa o operador de binarização com um valor limite definido por zero. Se o valor de entrada for maior que zero, então o valor de saída é 1, caso contrário o valor de saída é zero. O propósito da fase aleatória é reduzir o efeito da borda causada pela binarização.

Na simulação seguinte para demonstração do conceito, será considerado uma imagem com o texto "FCU" como mostrado na reconstrução da imagem (ver figura 3.16). Serão gerados, a partir da imagem dois hologramas binários $B_r(x, y)$ e $B_i(x, y)$ e feita a reconstrução da imagem para $B_r(x, y)$ e para $B_r(x, y) + jB_i(x, y)$ que representa o holograma complexo sintetizado.

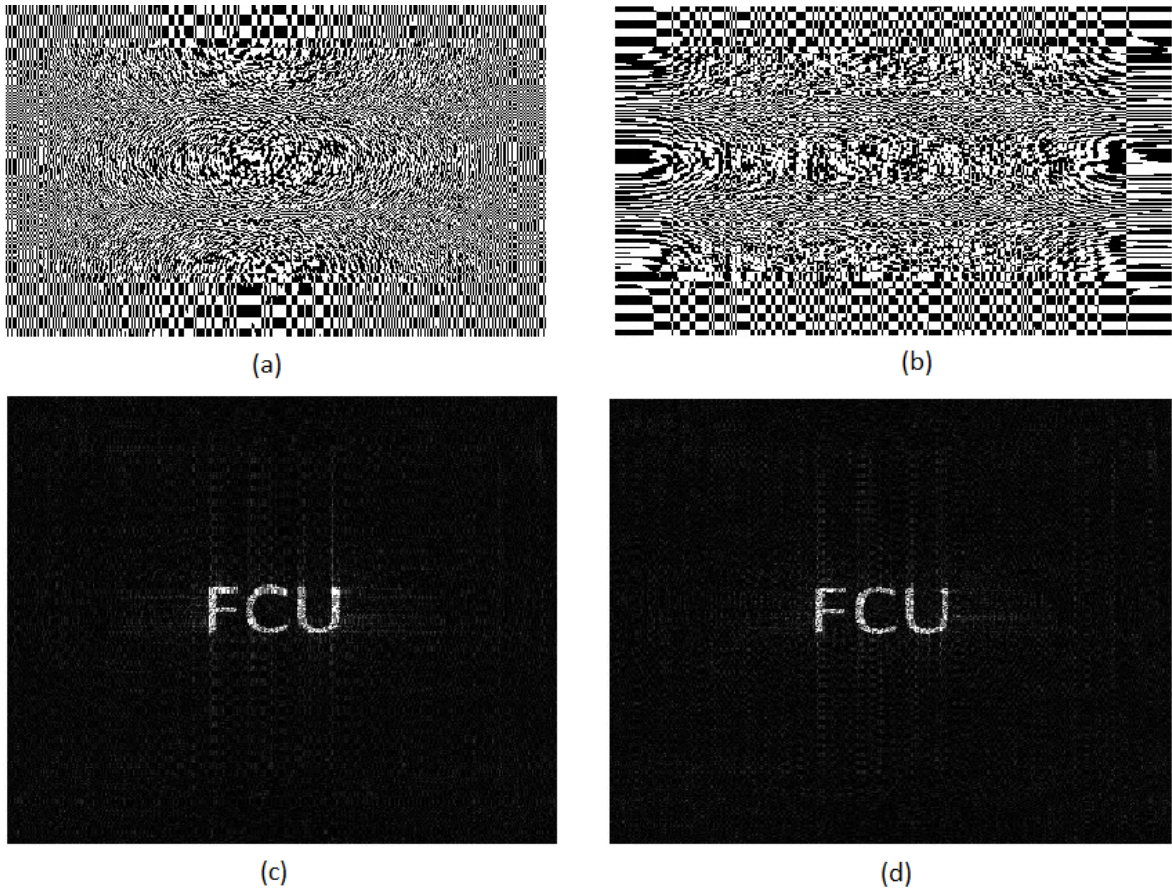


Figura 3.16: (a) Holograma binário $B_r(x, y)$, (b) Holograma binário $B_i(x, y)$, (c) Reconstrução da imagem através de $B_r(x, y)$, (d) Reconstrução da imagem através de $B_r(x, y) + jB_i(x, y)$.

Como é de reparar, as imagens reconstruídas se encontram nítidas, sendo a imagem reconstruída apenas com o $B_r(x, y)$ possui apenas um ruído ligeiramente maior a imagem reconstruída com o holograma complexo. O código do matlab usado para a realização desta simulação se encontra representada o trecho de código 5, nos anexos.

3.4 MODULADOR ESPACIAL DE LUZ

O modulador espacial de luz, muitas vezes também referido como SLM(Spatial Light Modulator), é um dispositivo eletronicamente programável que serve para manipulação das propriedades da luz, como a amplitude, fase ou polarização das ondas de luz no espaço e no tempo. Geralmente a tecnologia do modulador espacial de luz é semelhante à tecnologia das telas de LCD, baseando-se também no uso de cristais líquidos e polarizadores de luz nas suas telas.

Um dos propósitos do uso de um SLM é a visualização holográfica em tempo real de imagem e vídeos, dessa forma o SLM foi pensado para ser usado no âmbito desse projeto como um projetor holográfico para imagem e vídeo. No entanto um modulador espacial de luz moderno apenas permite modular ou a amplitude ou a fase e não ambos. Para a modulação tanto da amplitude como da fase, ou seja, para a modulação da função complexa seria necessário de

dois SLMs, um para a modulação da amplitude e um outro para a modulação da fase para a exibição do holograma complexo através do uso dos SLMs em cascata. Entretanto dado que apenas um SLM se encontra disponível para modulação da amplitude todo o software construído à frente, na fase da implementação, terá de ser adaptado para o uso apenas do holograma de amplitude.

Implementação do projeto

Os dois capítulos anteriores foram baseados nas pesquisas efetuadas pelo aluno ao longo do semestre, tendo como resultado um conjunto alargado de conhecimentos de diferentes técnicas e algoritmos usados na holografia, seja para simulação ou para projeção em tempo real com o uso do modulador espacial de luz.

Nesse capítulo é apresentada a implementação de toda uma ideia que originou a realização e escrita dessa dissertação a partir das bases acumuladas ao longo da pesquisa. O capítulo em si aborda toda a arquitetura do sistema no que diz respeito ao hardware e ao software do sistema, assim como simulações efetuadas ao nível do software. É abordada ainda a conectividade entre o computador e o sistema, detalhes técnicos do modulador espacial de luz usado como hardware para projeção holográfica de imagem e vídeo e até onde se conseguiu alcançar na implementação do projeto e o que faltou realizar para a finalização do projeto como um todo.

4.1 ARQUITETURA DO SISTEMA

4.1.1 Geral

A arquitetura geral do sistema elaborado para a implementação do projeto é semelhante ao modelo apresentado no capítulo da introdução. Contudo foi decidido, por questão de equipamento disponível e também pela facilidade da sua utilização, a colocação de um raspberry em substituição do microcontrolador e do módulo de comunicação. O raspberry é a parte do sistema que comunica com computador por wireless para transferência de imagem e vídeo e para controlo remoto por parte do utilizador. Neste componente é implementado todo o software necessário para a conversão de imagens ou vídeos em hologramas que depois serão usadas no projetor holográfico que representa o modulador espacial de luz. Todo o resto do sistema permanece inalterado. Dessa forma o sistema elaborado para sua implementação é o representado na figura 4.1.

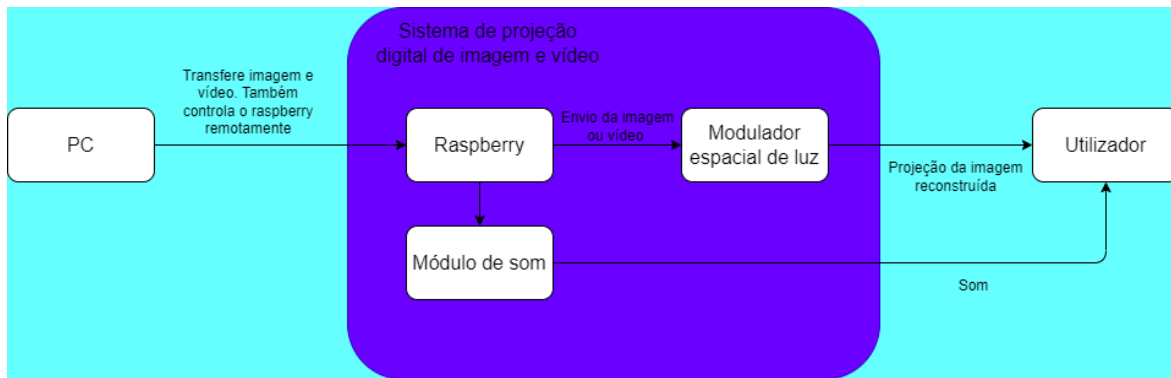


Figura 4.1: Arquitetura geral do sistema.

4.1.2 Hardware

Na parte do hardware foi utilizado um raspberry pi 4 (modelo B - ver figura 4.2) para tratamento de imagem e de vídeo e seu envio para o modulador espacial de luz. Este não possui uma porta HDMI, ao invés disso disponibiliza duas portas micro-HDMI. Desse modo para a comunicação entre o raspberry e o modulador espacial de luz, serão precisos dois adaptadores, um adaptador micro-HDMI para HDMI e um outro adaptador do HDMI para VGA, já que o modulador espacial de luz utilizado apenas permite como portas de entrada de informação as portas VGA e/ou ethernet.



Figura 4.2: Raspberry pi 4 model B [www.raspberrypi.com].

O modulador espacial de luz usado é um construído pela empresa Holoeye, tipo LC 2002

Transmissive SLM (que foi entretanto descontinuado) e as suas características, relativamente ao seu microdisplay, são as descritas na tabela 4.1, retirado do próprio website da Holoeye na seção "SPATIAL LIGHT MODULATORS -> DISCONTINUED DEVICES".

Display type	Transmissive LC
Resolution	800x600
Pixel Pitch	32 μ m
Fill Factor	55%
Active Area	21,0x26,0mm
Addressing	8 Bit(256 Gray Levels)
Signal Formats	VGA - SVGA Resolution
Frame Rate	60Hz

Tabela 4.1: Propriedades do modulador espacial de luz

A tabela descreve a resolução como sendo 800x600, acima do objetivo pensado, possui um endereçamento de 8 bits, implicando a necessidade de converter as imagens para grayscale até 256 níveis antes de as converter em hologramas. Também indica a necessidade do uso de um adaptador para VGA como já foi esclarecido anteriormente e o frame rate é à volta dos 60Hz, o que significa que em 1 segundo podem ser atualizadas até 60 imagens o que representa 6 vezes mais do que os 10 FPS propostos. Apesar de descontinuado, esta informação demonstra que o equipamento tem todas as características necessárias para a implementação do sistema pretendido. Na figura 4.3 pode-se observar o modulador espacial de luz utilizado.

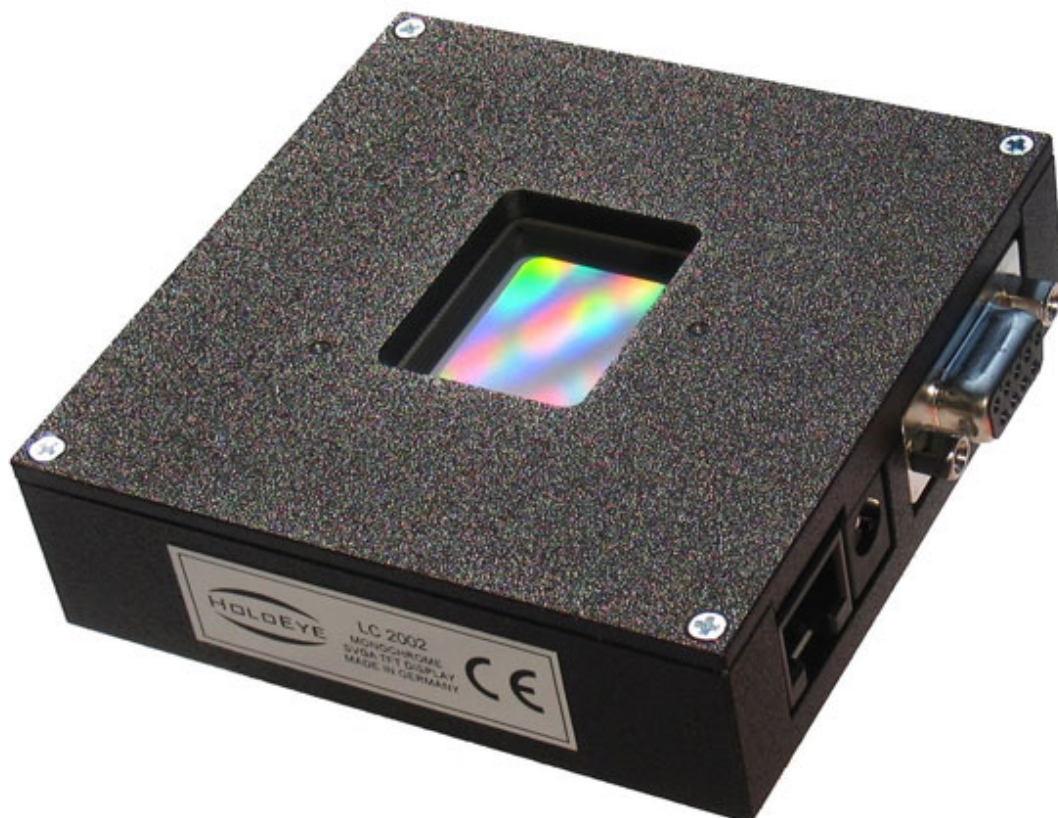


Figura 4.3: LC 2002 Transmissive SLM [12].

4.1.3 Software

Na parte do software foram desenvolvidos dois programas, um para conversão de imagens em hologramas e um outro para conversão de vídeos inteiros (com som incluído) em sequências de hologramas (também com som incluído). Ambos os programas foram desenvolvidos em python com recurso das bibliotecas `opencv` para manipulação de imagem, `numpy` para uso de cálculos matriciais e transformada de Fourier e por fim foi usado o `moviepy` para áudio. O algoritmo usado nestes dois programas foi o algoritmo do holograma binário de Fourier, o último apresentado no capítulo anterior, dada a sua facilidade de utilização, oferecer a possibilidade de se usar apenas o holograma real e por ser facilmente adaptado a qualquer resolução.

No primeiro programa apresentado, para conversão de imagens em hologramas, este possui uma interface básica, como o descrito na figura 4.4, que permite ao utilizador escrever o nome de um ficheiro de imagem nos formatos `.png`, `.bmp` ou `.jpg` e devolver o holograma correspondente, gravando a mesma automaticamente num ficheiro sob o nome `Holo_nome` do ficheiro, no formato `.bmp` ao clicar apenas num botão (no botão `ToHolo`, como se pode verificar na figura 4.4).

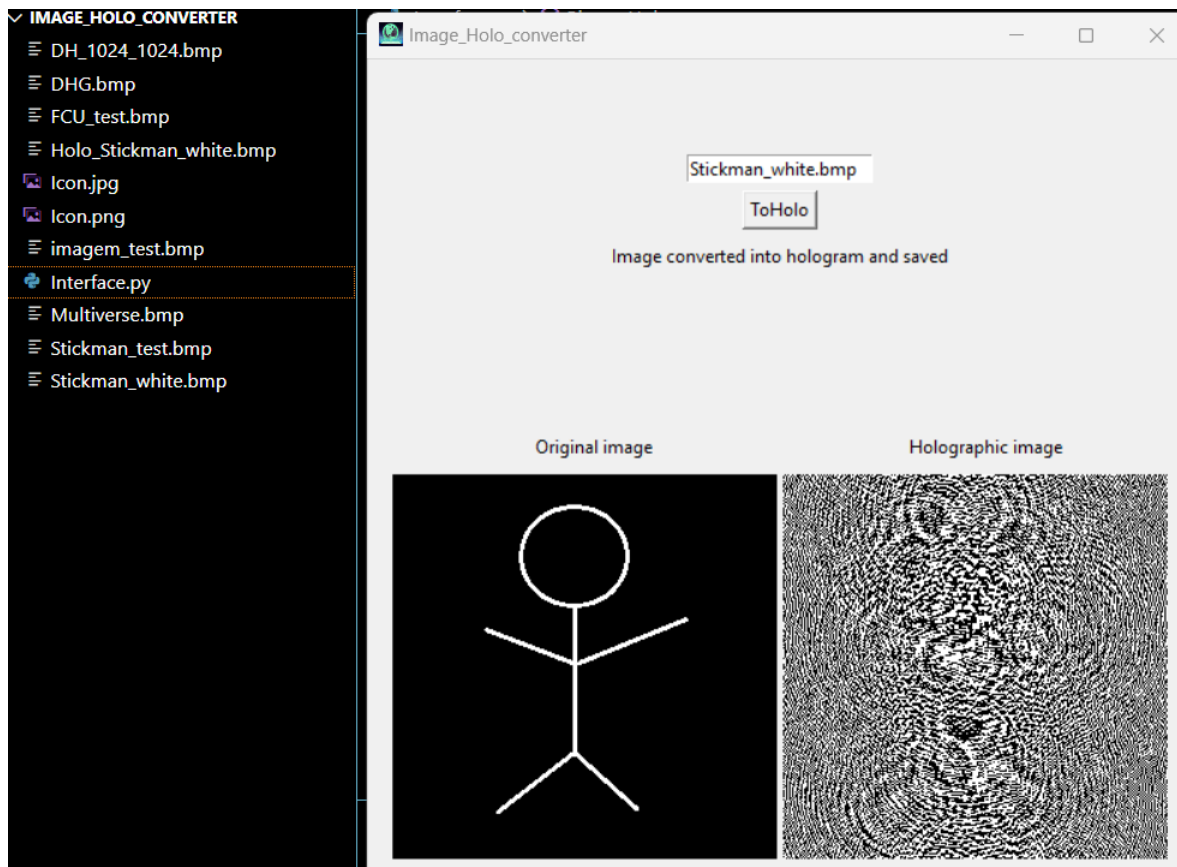


Figura 4.4: Interface do programa de conversão de imagem para holograma.

Onde a função `Holo_save` grava o holograma criado a partir da imagem, a função `Binary_Hologram` é a própria versão do algoritmo em python, a função `Holo_converter` é a função responsável pela conversão da imagem no seu respectivo holograma, pela sua normalização na escala de cinza com 256 níveis e por guardar o holograma num ficheiro com formato `.bmp`. As funções de `print` servem para exibir a imagem e o seu respectivo holograma na interface. Por fim a função `File_search` serve para procura do ficheiro correspondente ao escrito pelo utilizador que, caso não exista, o programa é impedido de avançar para as funções seguintes.

O código fonte referente a este programa é dado pelo trecho de código 6, nos anexos.

Para o segundo programa, este já não possui interface, tendo o utilizador que inicializar o programa a partir da linha de comandos. Caso o ficheiro não exista, o programa simplesmente dá erro e encerra. Nesse programa também só é possível converter ficheiros em formato `.mp4`, sendo depois guardados no formato `.avi`. Em relação ao áudio, este é separado do vídeo, havendo um ficheiro no formato `.mp3` e um ficheiro no formato `.avi`, sem som, guardados e no final o programa junta esses dois ficheiros e forma um terceiro, também em formato `.avi`, no entanto já com o áudio incluído. Na execução do programa, com a escolha de um ficheiro válido, aparece um padrão de interferências que representa uma sequência de hologramas a ser exibidos no tempo como o indicado na figura 4.5, que corresponde a um frame desta sequência.

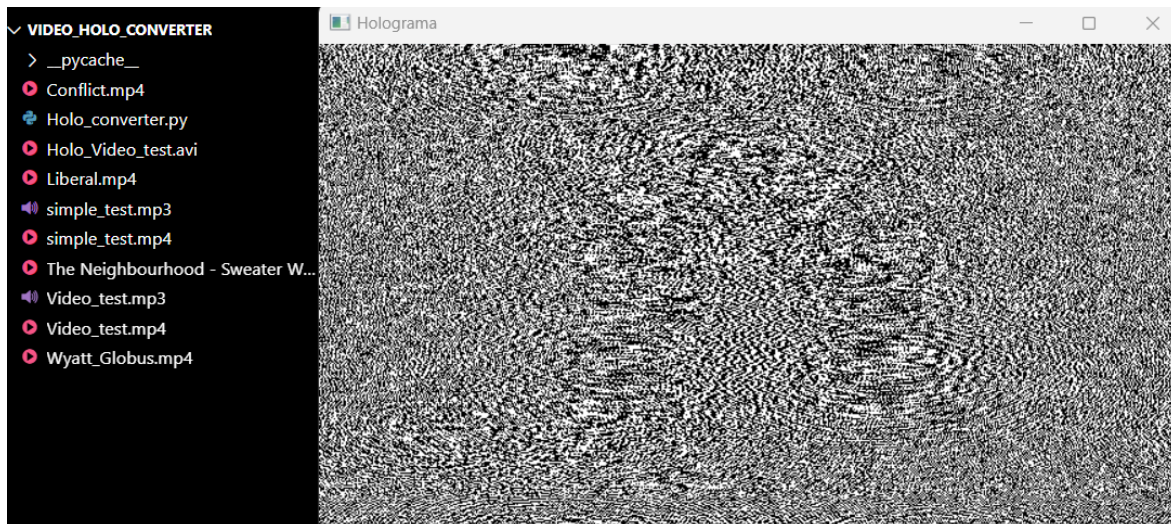


Figura 4.5: Vídeo do holograma a ser construído.

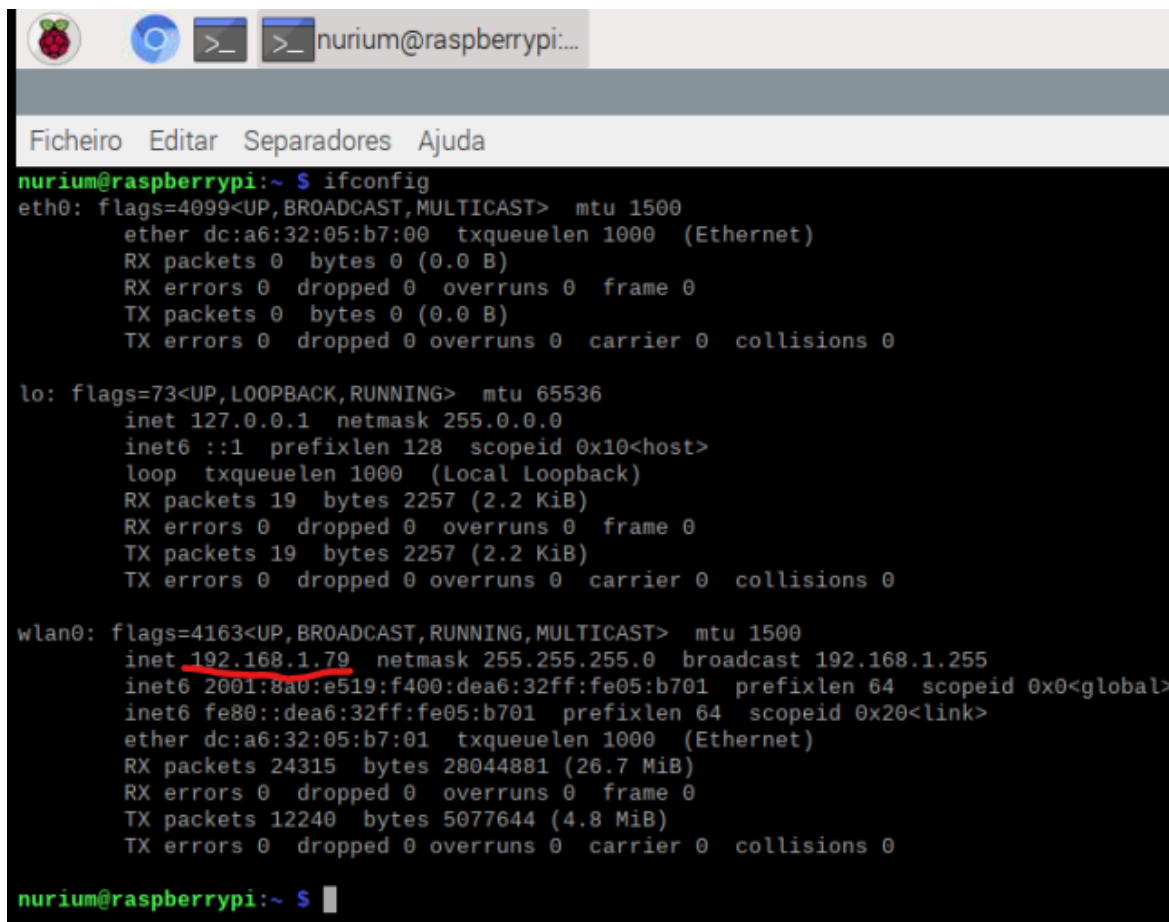
O código fonte referente a esse programa é dado no trecho de código 7, nos anexos.

4.2 TESTES REMOTOS NO RASPBERRY

No raspberry foram realizados vários testes remotos para averiguar formas de ter completo controlo do raspberry por via remota e também possuir a capacidade de poder enviar ficheiros de imagem e vídeo para o mesmo de modo a executar a sua conversão para hologramas com o software, que foi abordado anteriormente, no próprio dispositivo.

No lado do computador pessoal, foi instalado o **VNC viewer**[18] para comunicação remota para com o raspberry, que permite partilha de desktop e o **winscp**[19], plataforma da microsoft windows, para transferência de arquivos e de ficheiros do computador pessoal para o raspberry. Já no lado do raspberry foram configuradas as permissões de modo a permitir o uso SSH e VNC no dispositivo.

Dessa forma foi aberto um terminal do linux e usou-se o comando `ifconfig` para descobrir o endereço ip do raspberry, como descrito na figura 4.6.



```
nurium@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether dc:a6:32:05:b7:00 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 19 bytes 2257 (2.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 2257 (2.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.79 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2001:8a0:e519:f400:dea6:32ff:fe05:b701 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::dea6:32ff:fe05:b701 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:05:b7:01 txqueuelen 1000 (Ethernet)
    RX packets 24315 bytes 28044881 (26.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12240 bytes 5077644 (4.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

nurium@raspberrypi:~ $
```

Figura 4.6: Uso do ifconfig no terminal.

O endereço ip do dispositivo, neste caso em particular, é 192.168.1.79 como o indicado no terminal representado na figura 4.6. Com essa informação foi aberto o VNC e foi criada uma nova conexão, no qual foi inserido o endereço ip do servidor e um nome que identifique essa conexão, por exemplo Remote_Test, como o descrito na figura seguinte.

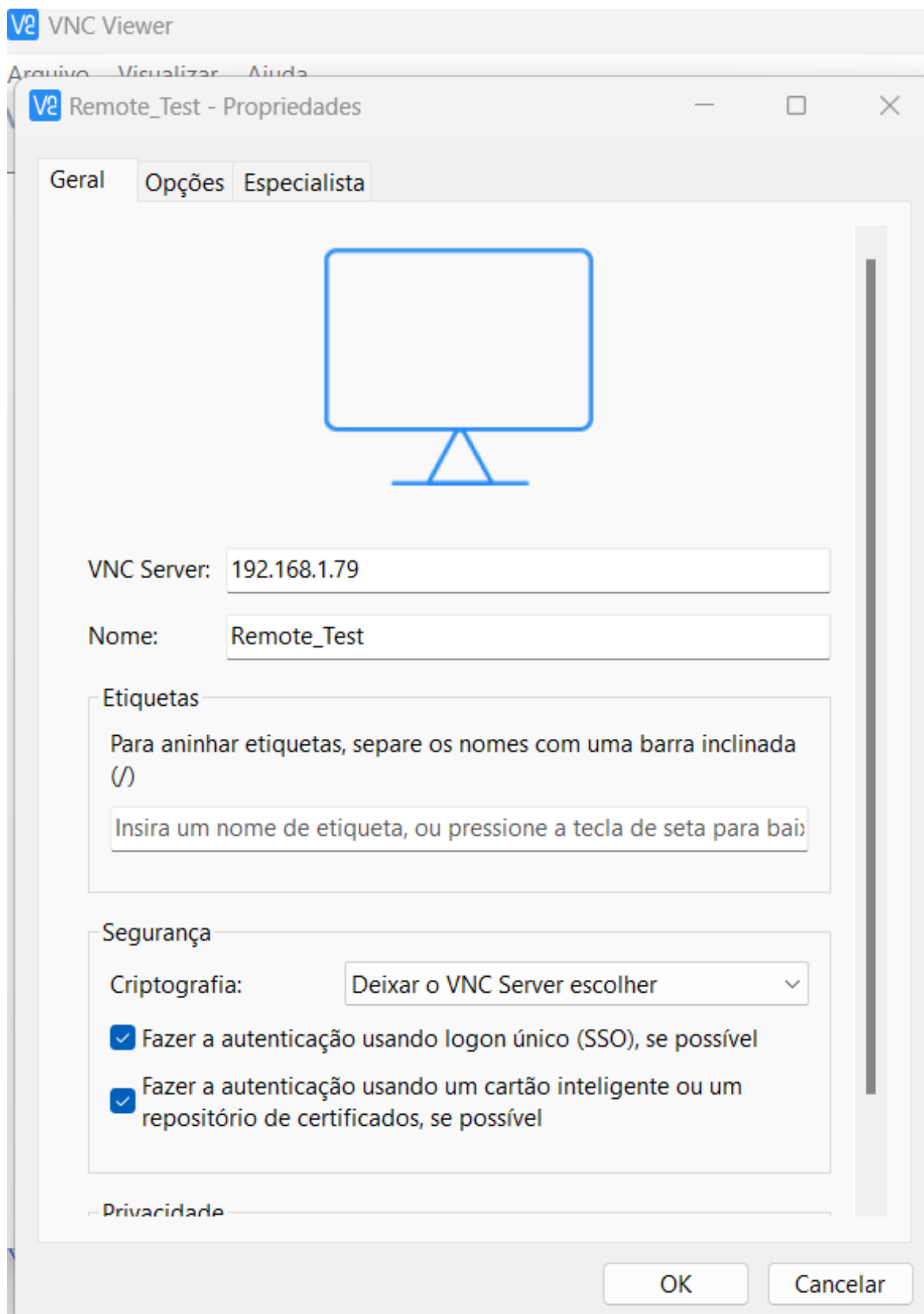


Figura 4.7: Criação de uma nova conexão.

Depois da criação de uma nova conexão a mesma aparece na interface do programa como

o indicado na figura 4.8. Para se conectar ao raspberry basta clicar na conexão criada na plataforma em causa.

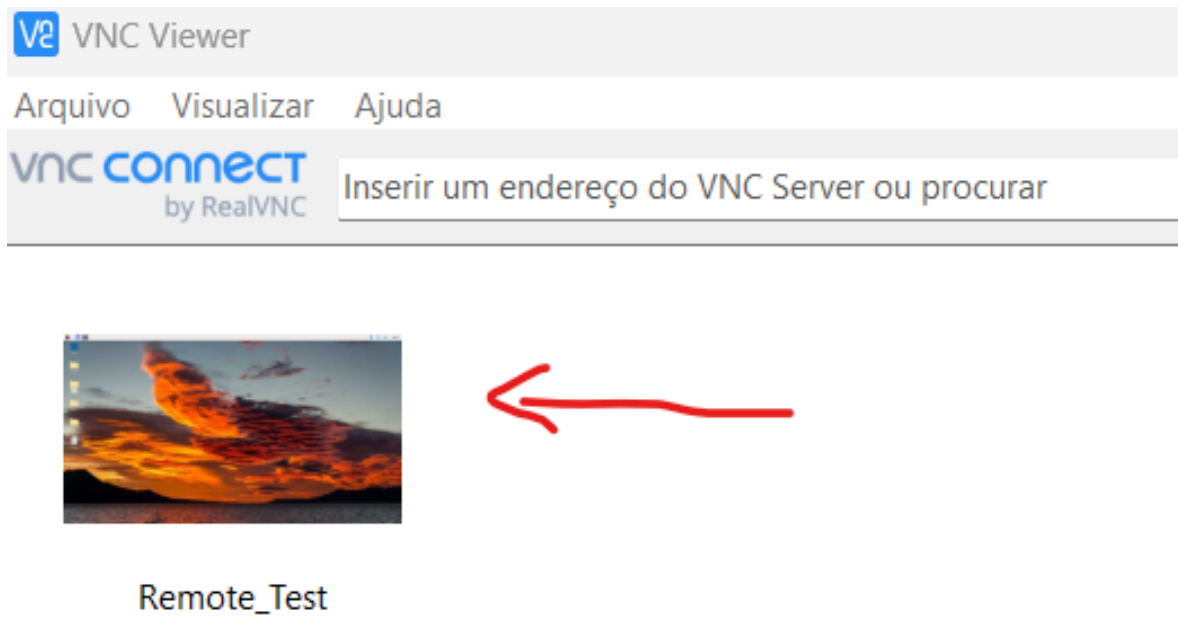


Figura 4.8: Conectar ao Remote_Test

Ao clicar no Remote_Test será pedido as credenciais, nome e senha do raspberry para assumir controlo remoto no mesmo, tal como o representado na figura 4.9.

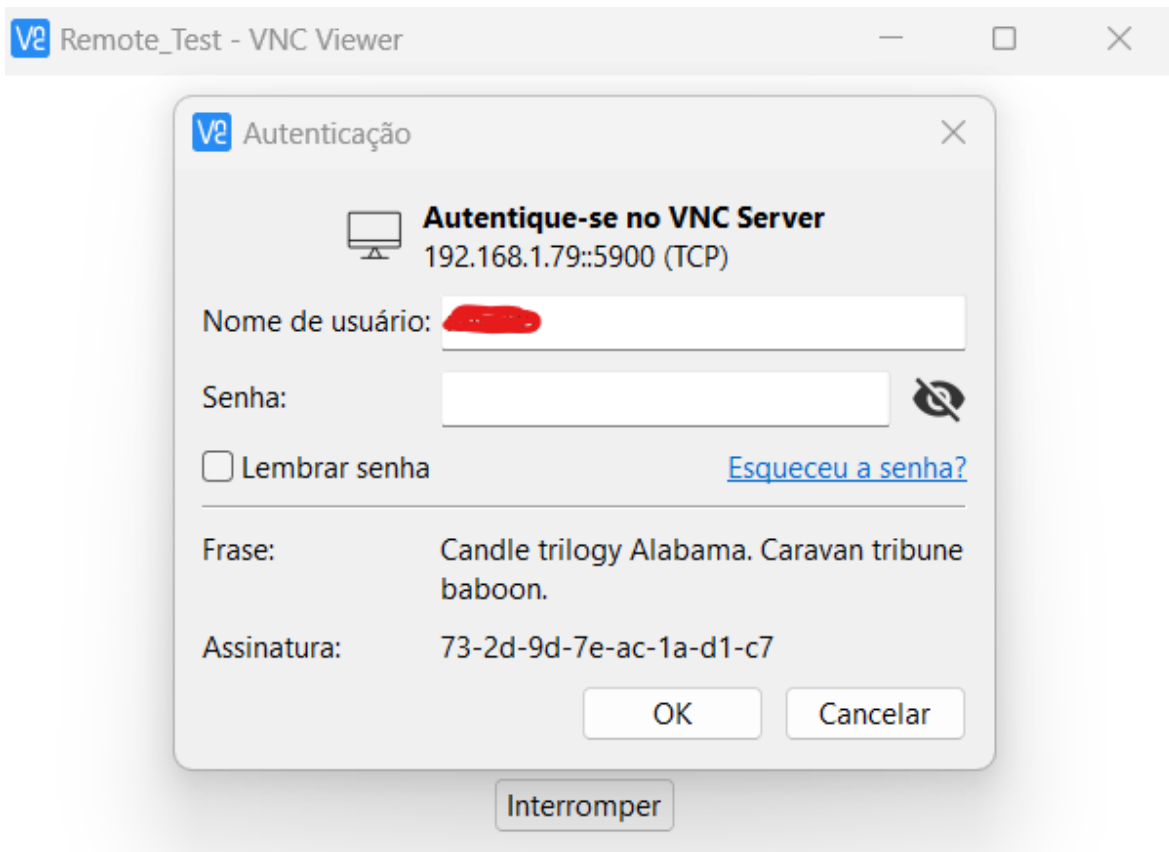


Figura 4.9: Credenciais de acesso.

Depois de dadas as credenciais ao sistema, nome e senha, e clicado no botão "OK", vai aparecer o ambiente do sistema operativo do linux do raspberry, indicando que a partir desse momento já é possível ter controlo remoto do dispositivo, tal como se encontra descrito na figura 4.10.

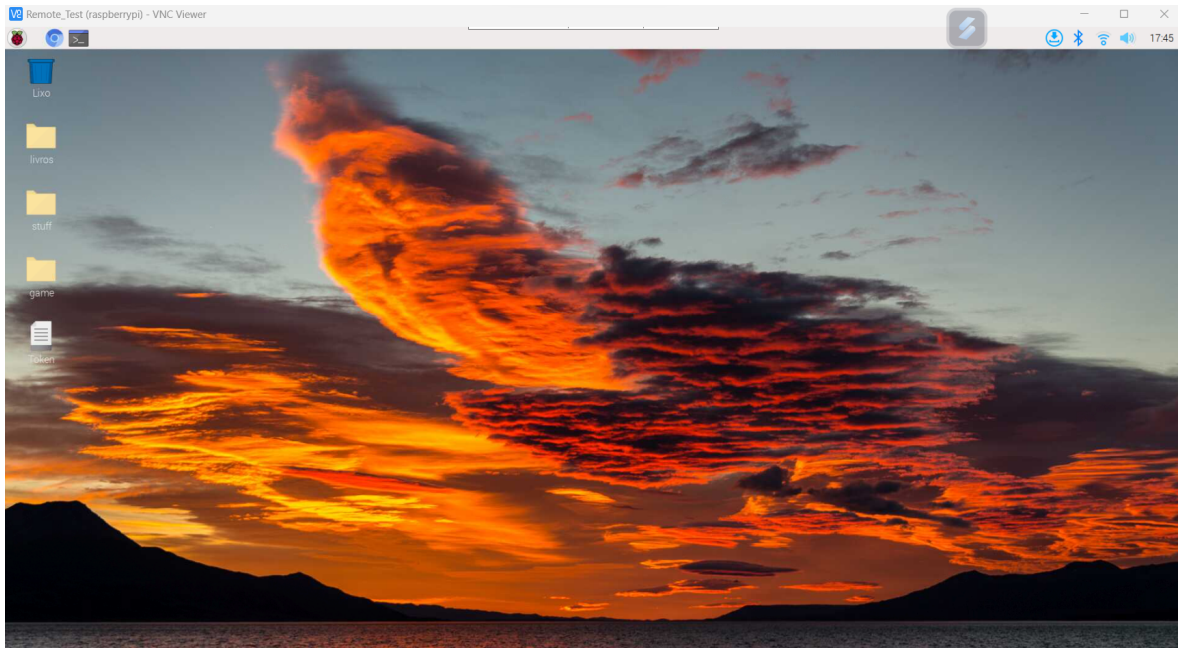


Figura 4.10: Acesso remoto ao raspberry.

É possível ter acesso ao raspberry com o **VNC viewer**[18], no entanto para envio de ficheiros e arquivos do computador para o raspberry é preciso de ter conta "premium" de modo a conseguir fazer isso com o **VNC viewer**[18], ou seja, é preciso de ter uma licença profissional ou enterprise, ao qual se paga todos os meses.

Para a transferência de ficheiros e arquivos do computador para o raspberry sem a necessidade de ter uma licença de **VNC viewer**[18] paga, foi instalado o software **winscp**[19] apenas para envio de ficheiro e de arquivos remotamente. Primeiro se abriu o **winscp**[19] e apareceu uma interface de acesso ao raspberry no qual pedia o endereço ip, o nome do utilizador e a senha, como o representado na figura 4.11.

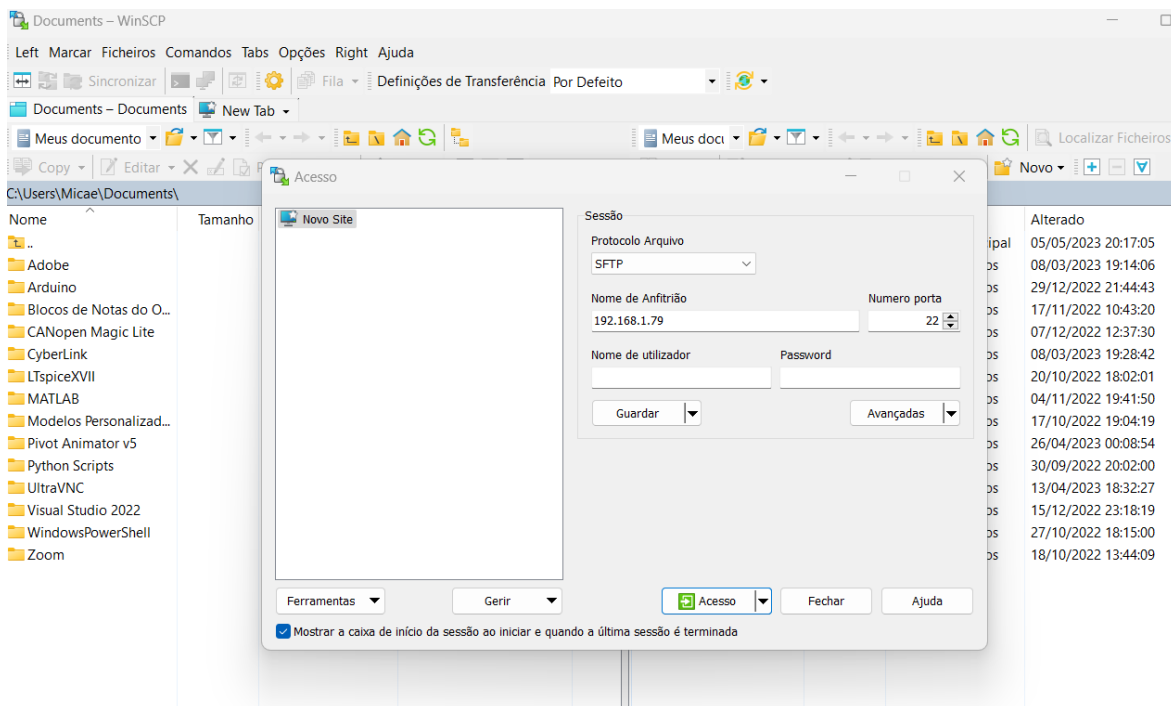


Figura 4.11: Acesso ao raspberry pelo **winscp**[19].

Ao ter acesso ao raspberry pelo **winscp**[19] é possível verificar que no lado esquerdo se encontra o sistema de ficheiros do windows do computador pessoal no qual me encontro a usar e no lado direito se encontra o sistema de ficheiros do raspberry. Depois disso, navegou-se até ao diretório onde se encontram várias imagens, na parte do computador como indicado na figura 4.12, algumas delas já convertidas em hologramas e foi selecionada um ficheiro aleatório para o lado direito da figura, que representa o sistema de ficheiros do linux. Na parte do raspberry colocou-se o diretório de ficheiro localizado no ambiente de trabalho, assim caso o ficheiro seja efetivamente enviado tal será visualizado de forma imediata no desktop do sistema operativo raspbian do raspberry. Quando arrastar o ficheiro do diretório onde se encontra o ficheiro da imagem no windows para a desktop do raspberry e largar, uma mensagem aparece tal como o indicado na figura 4.13.

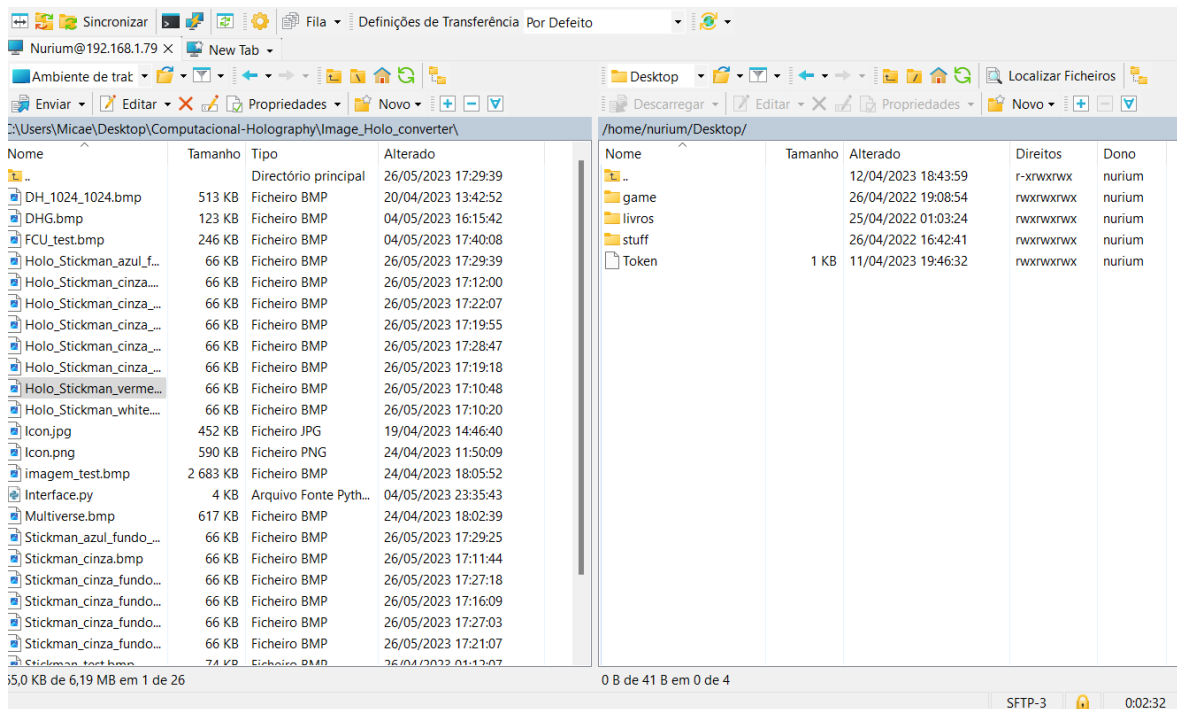


Figura 4.12: Envio de uma imagem aleatória para o raspberry.

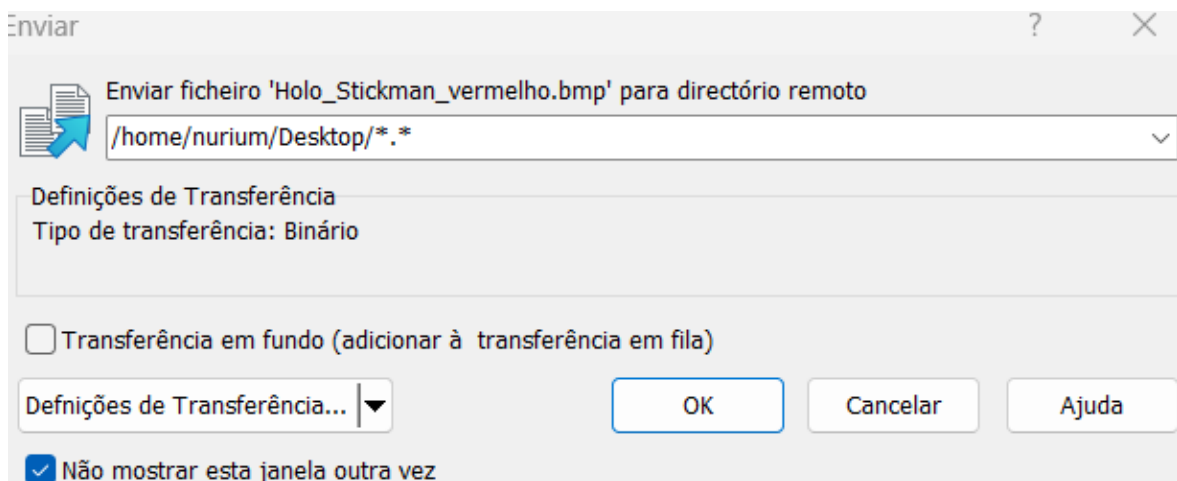


Figura 4.13: Mensagem de envio.

Basta clicar no ok, ou então no ENTER e o ficheiro da imagem aparece na desktop do raspberry, demonstrando que efetivamente funciona e que é possível a transferência remota de arquivos e ficheiros com facilidade, como o indicado na figura 4.14.

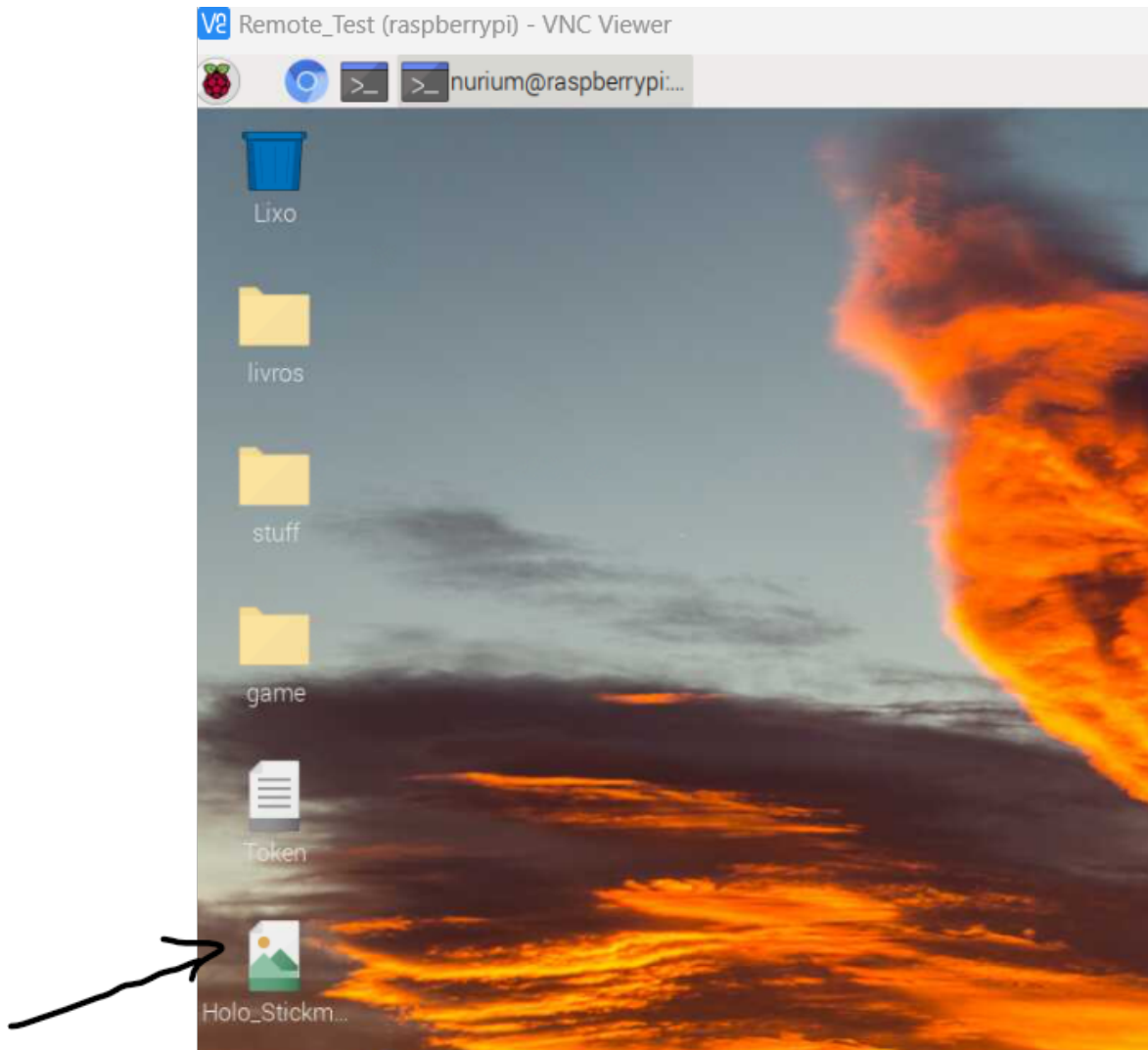


Figura 4.14: Imagem enviada.

Desse modo, consegue-se ter controlo remoto do raspberry e enviar ficheiros e arquivos do computador para o raspberry. Para experimentar o software feito no computador, foi criado um repositório no github com o nome de **Computational-Holography**, onde foram colocados os programas desenvolvidos no repositório. No raspberry basta fazer o git clone desse repositório, como descrito na figura 4.15.

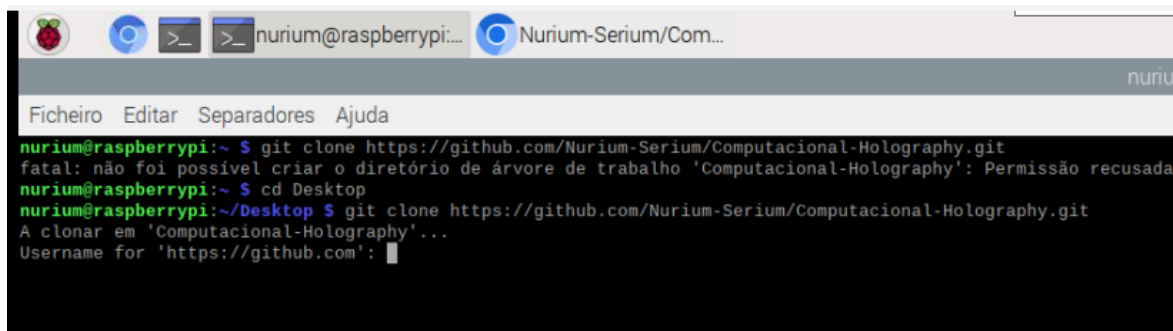


Figura 4.15: Git clone do repositório.

O repositório sob o nome de **Computational-Holography** aparece no desktop do raspberry como o indicado na figura 4.16, onde todo o código fonte se encontra.

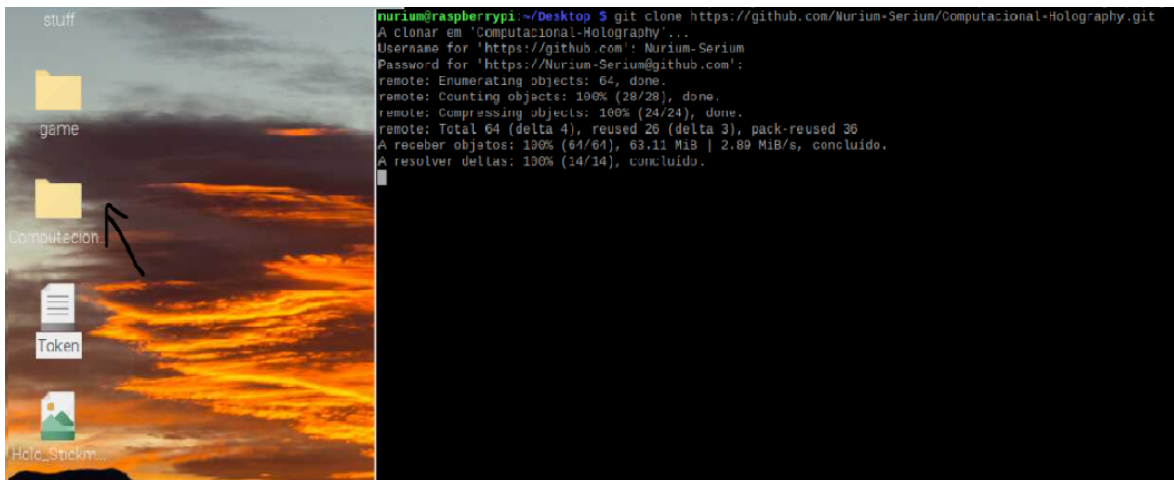


Figura 4.16: Repositório do código fonte.

Com o vscode instalado no raspberry, foi usado o primeiro programa indicado na subsecção do software na secção da arquitetura do sistema e assumindo que todas as bibliotecas estão instaladas, o programa em si aparece no seguinte modo.

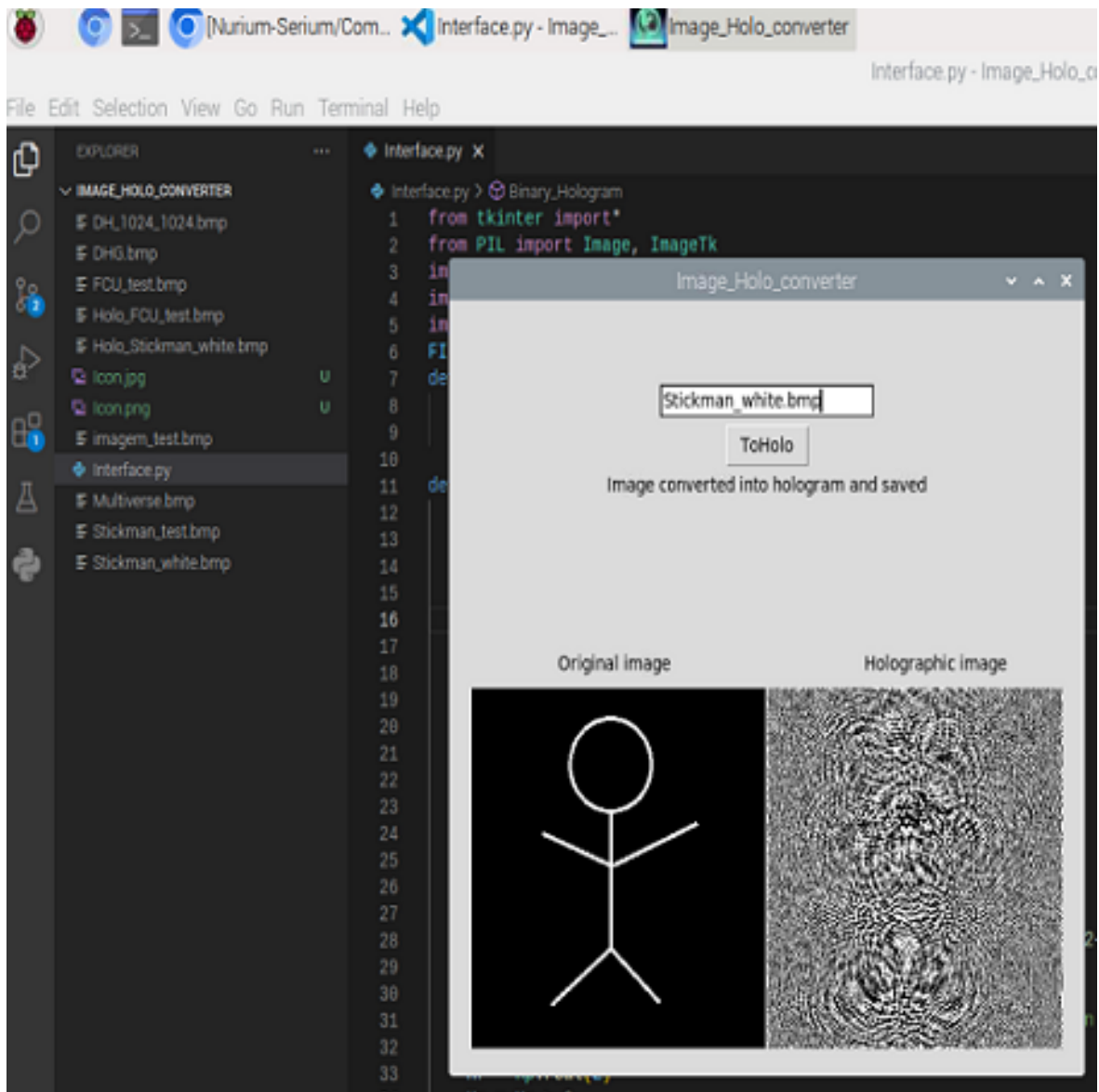


Figura 4.17: Execução do programa.

Após essas etapas o programa deve funcionar no raspberry. No caso das bibliotecas do python ainda não existirem, devem ser instaladas. Para tal deve-se abrir um terminal e escrever no mesmo os comandos `pip install numpy` para a instalação da biblioteca numpy, `pip install opencv-python` para a instalação da biblioteca e por fim, para o caso do áudio nos vídeos, fazer `pip install moviepy`.

4.3 SIMULAÇÃO

Para a realização de simulações relativamente à reconstrução das imagens originais dos respectivos hologramas gerados a partir do algoritmo usado, sem recurso ao modulador espacial de luz, foram desenvolvidos dois programas também em python, um usado para reconstrução da imagem individual e um outro usado para reconstrução do vídeo como um todo, frame por frame, com o áudio incluído.

4.3.1 Simulação da reconstrução de imagens individuais

O código fonte usado para a simulação da reconstrução de imagens individuais de um holograma tem a seguinte estrutura, como a indicada no trecho de código 8, nos anexos.

O que o programa faz é exibir primeiro o holograma e depois, ao clicar na letra q, exibe a imagem reconstruída num ficheiro que o utilizador escolher, caso exista. Também se faz questão de dar a entender o leitor que nesta simulação os parâmetros delta, lambda1 e z neste programa vão permanecer intocáveis, e que a única mudança será feita nesses mesmos parâmetros no código fonte da geração do holograma, como o indicado na subsecção do software, secção correspondente à arquitetura do sistema. A simulação será feita com os parâmetros do algoritmo que dão origem ao holograma diferentes relativamente aos parâmetros descritos no programa da simulação, como forma de demonstrar o seu efeito na exibição da reconstrução da imagem original. Parâmetros como distância entre os pixels (delta), distância focal (z), tamanho da imagem(M,N) e comprimentos de onda serão tidas em conta.

Para exemplificar a simulação, considera-se a imagem original de um boneco stickman com resolução 256x256 e diversas distâncias focais para -40, -50 e -60 mm, mantendo os restantes parâmetros iguais aos da simulação. O boneco stickman tem a forma idêntica ao da figura 4.18.

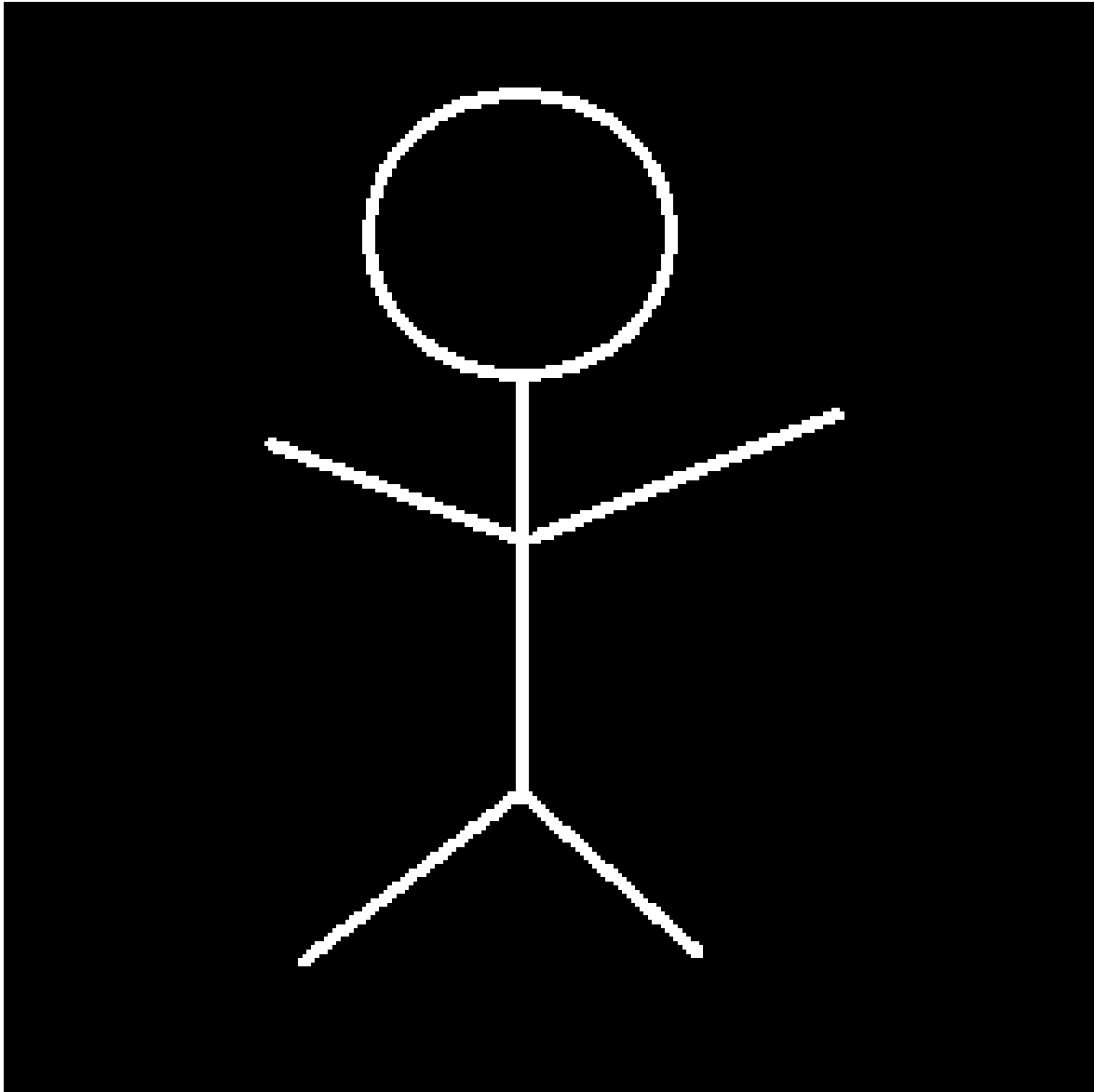


Figura 4.18: Boneco stickman.

Começa-se por simular para $z=-40\text{mm}$ e o resultado é uma reconstrução da imagem original não muito nítida, apesar de ainda se notar o boneco stickman, como o observado na figura 4.19.

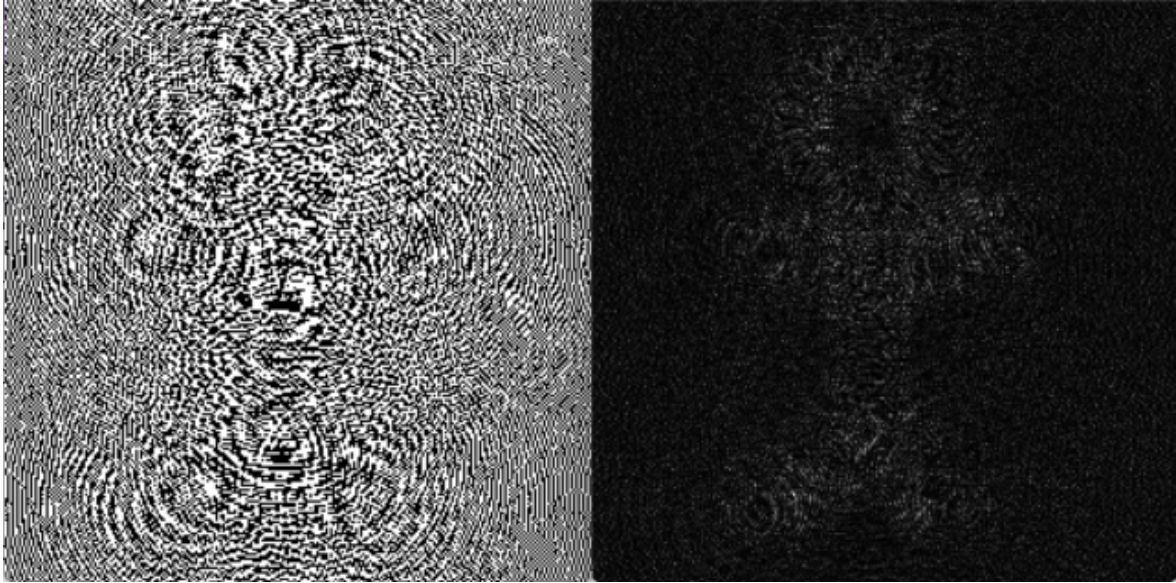


Figura 4.19: Simulação do Stickman para $z=-40$.

Para $z=-50$ mm, praticamente igual ao parâmetro da simulação, é possível de verificar a reconstrução da imagem original mais nítida, como o indicado na figura 4.20.

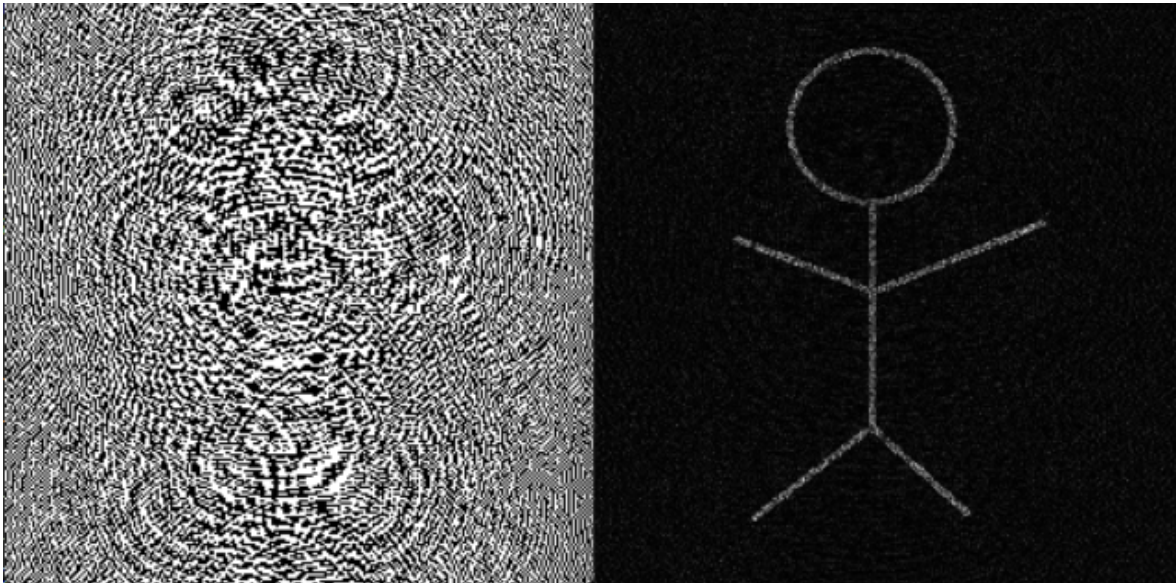


Figura 4.20: Simulação do Stickman para $z=-50$.

Para $z=-60$ mm o resultado da reconstrução da imagem é idêntico relativamente a $z=-40$ mm, ver figura 4.21.

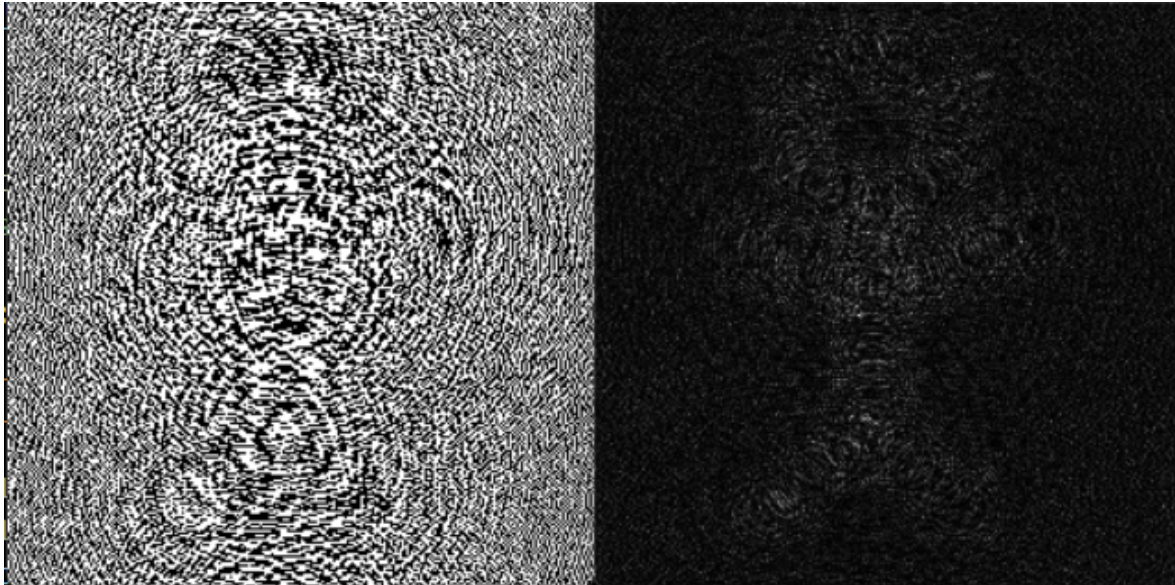
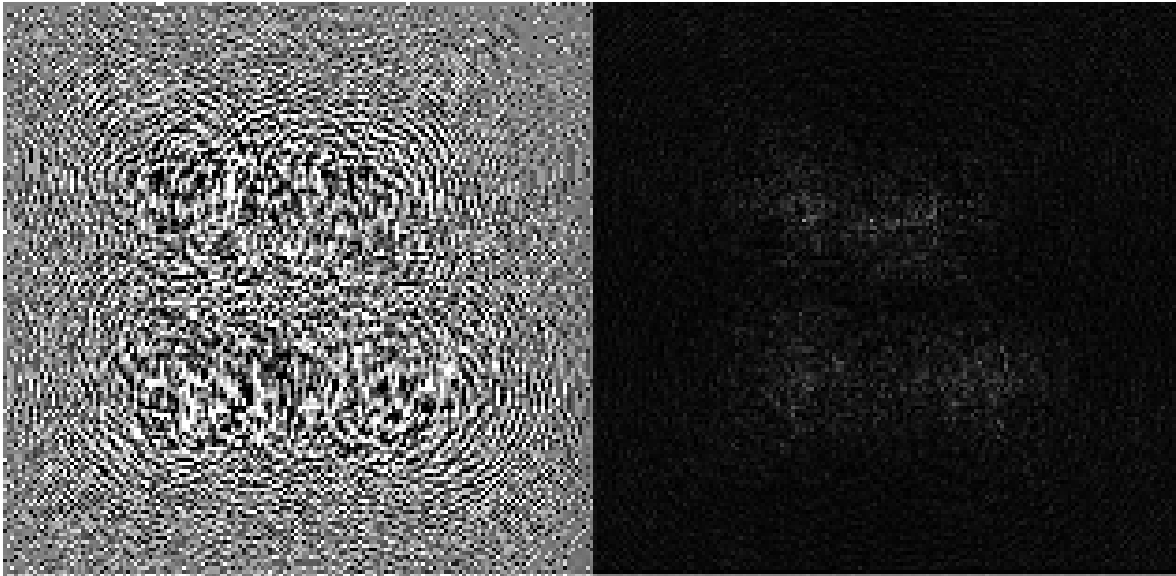
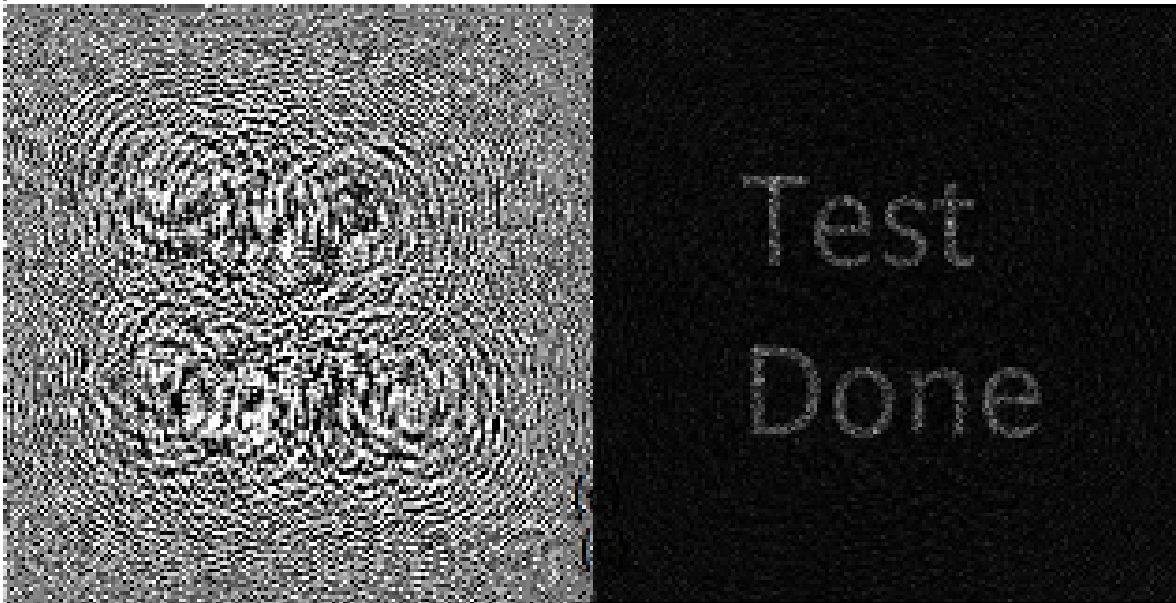


Figura 4.21: Simulação do Stickman para $z=-60$.

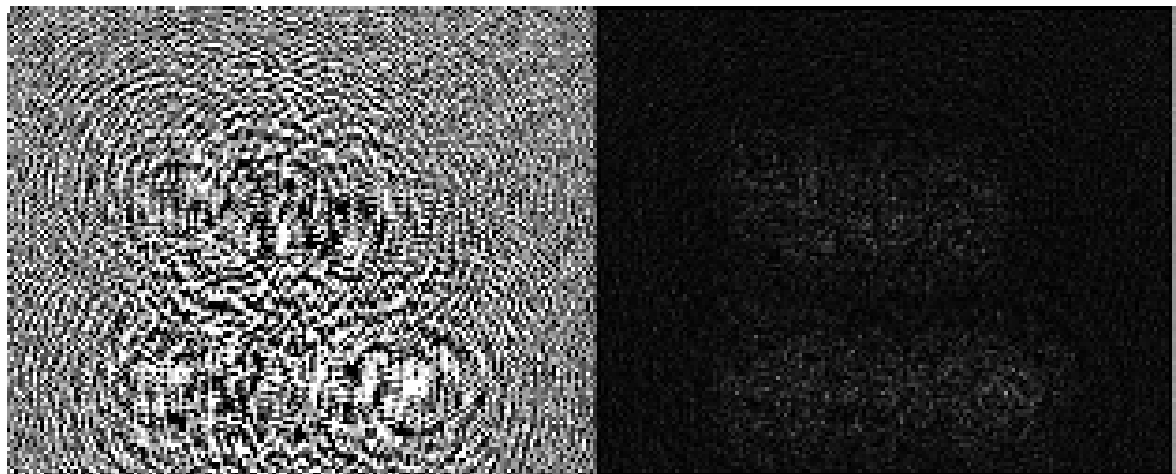
Isso demonstra que o parâmetro z influencia a reconstrução da imagem e é um parâmetro a ter em conta quando estiver a ser utilizado o modulador espacial de luz. O mesmo teste foi simulado com uma imagem mais complexa como por exemplo um texto a dizer "Test Done", e os resultados são equivalentes à simulação realizada com o boneco do stickman como aponta a figura 4.22.



(a)



(b)



(c)

Figura 4.22: (a) $z=-40\text{mm}$, (b) $z=-50\text{mm}$ e (c) $z=-60\text{mm}$.

Considere-se agora as variações do parâmetro delta, mantendo os restantes parâmetros iguais ao da simulação. Simulando a reconstrução da imagem original para um delta com valores de $12.66/2\mu m$, $12.66\mu m$ e $12.66\times 2\mu m$, a partir do uso da mesma imagem que na simulação anterior, é demonstrado através da simulação, que um holograma com um delta muito diferente do delta usado na reconstrução da imagem original, pelo menos com a diferença de $2x$, pode estragar a reconstrução da imagem original, deixando-a desfocada. É também de reparar que o holograma com a metade do delta exibe um padrão de interferência completamente ilegível, impossível de ser usado num modulador espacial de luz. Já o holograma com $2x$ o delta exibe um padrão de interferências que representa a própria imagem do boneco stickman, o que também não pode acontecer. Na prática esses dois hologramas seriam incapazes de produzir uma imagem num modulador espacial de luz, dada à má escolha da distância de amostragem (ver as figuras seguintes).

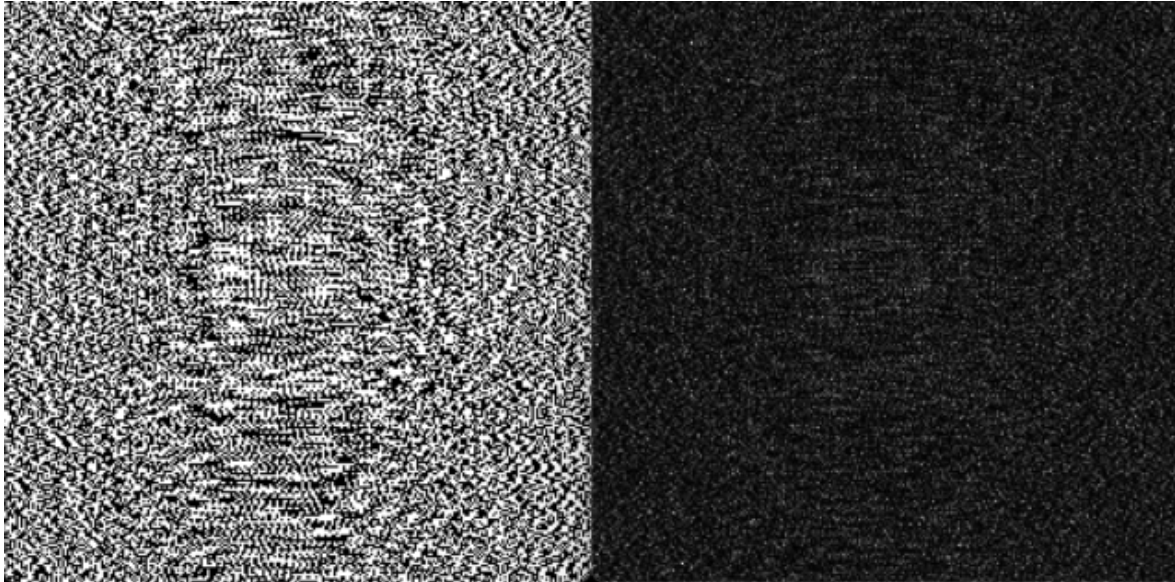


Figura 4.23: Simulação com $\delta/2$.

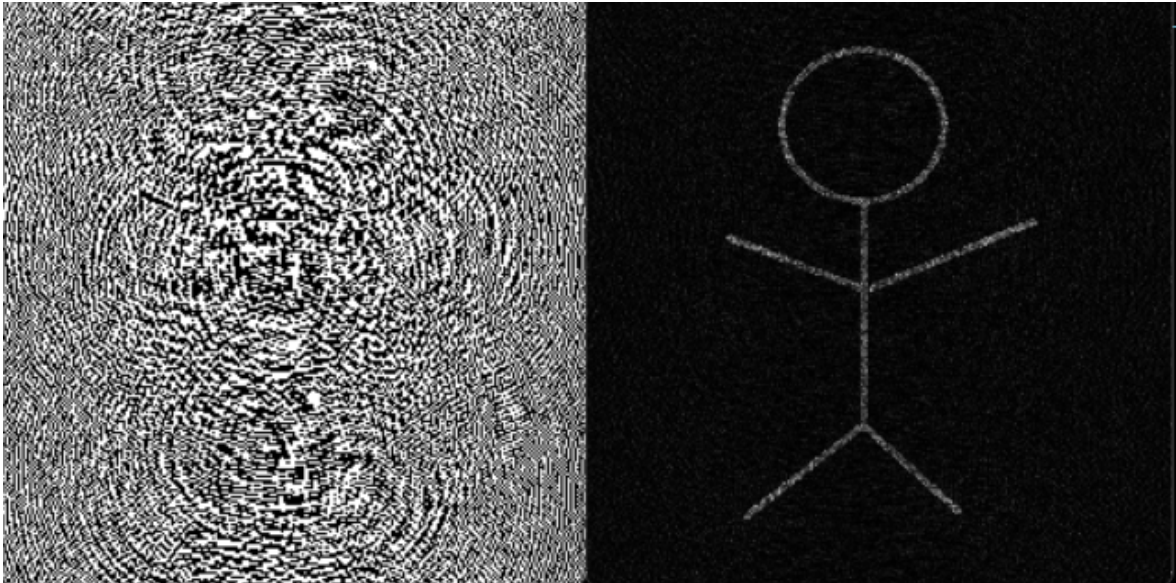


Figura 4.24: Delta igual ao delta da simulação.

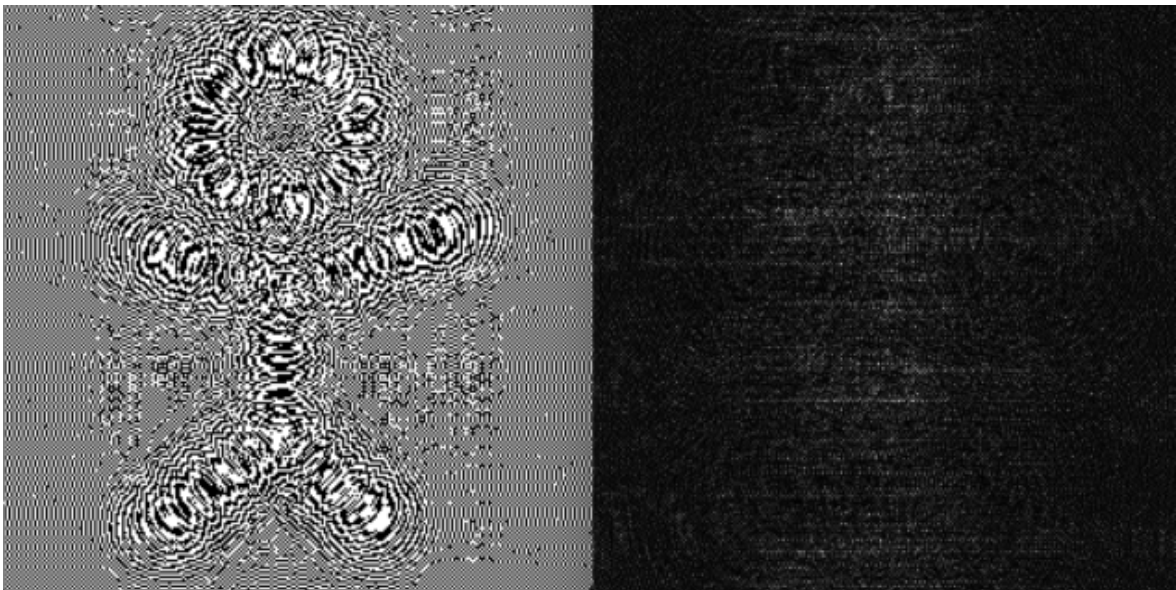


Figura 4.25: Simulação com delta 2x.

O teste seguinte considera a resolução da imagem. Será usado como imagem de teste um texto com "Test Done" como feito anteriormente, desta vez uma com resolução de 512×512 e uma outra com 635×635 . O resultado da simulação, ver figura 4.26 e 4.27, indica que a resolução da imagem não influencia muito na geração do holograma, contrariamente aos parâmetros anteriormente simulados.

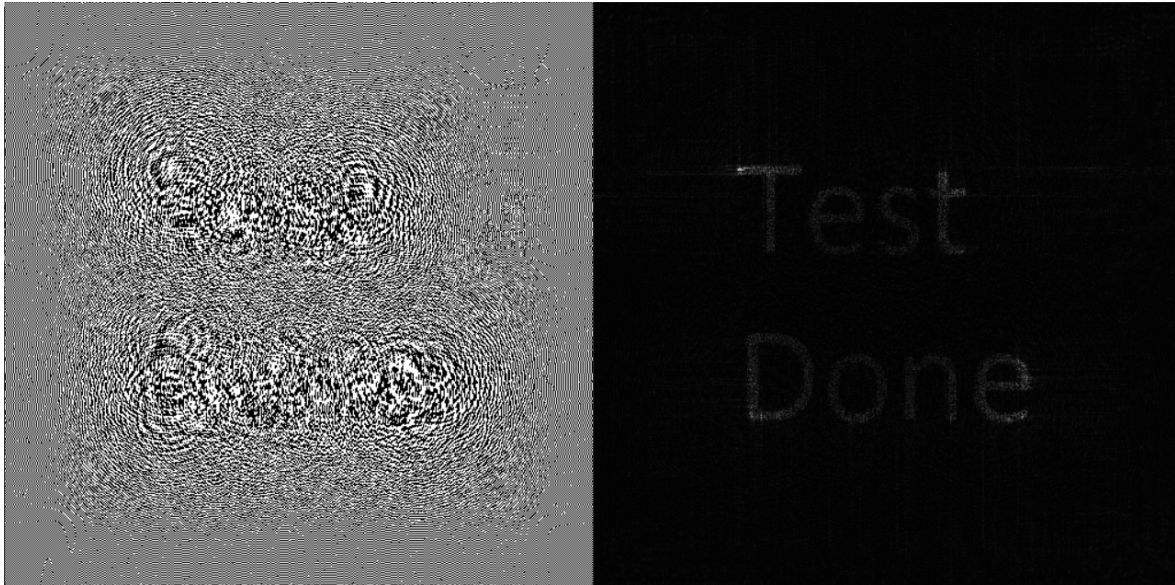


Figura 4.26: Test Done com resolução de 512X512.

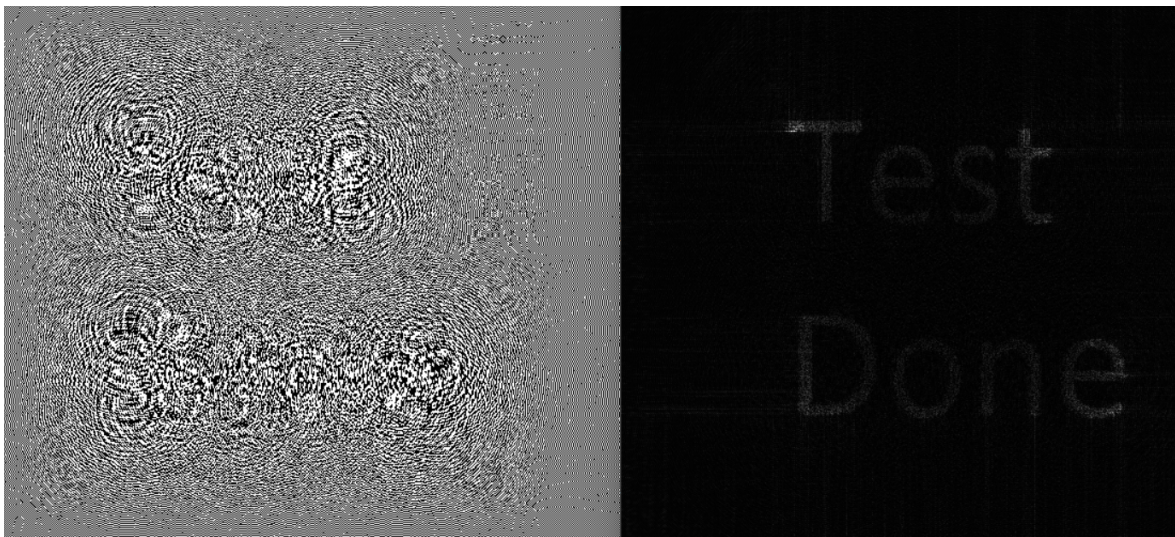


Figura 4.27: Test Done com resolução de 635X635.

O facto dessas imagens reconstruídas não parecerem tão nítidas quanto à imagem na figura 4.22(b) é devido à resolução das mesmas. Quanto maior a resolução menos nítida se torna a imagem, no entanto continua a ser possível usar imagens com resoluções por volta dos 640×360 , e até imagens com uma resolução ainda maior, o que pode ser relevante quando forem executadas simulações em vídeo.

O comprimento de onda usado nas simulações anteriores se situa no vermelho. A questão pertinente é, o que aconteceria caso se tentasse usar um holograma gerado no comprimento de luz situado no verde ou no azul no modulador espacial de luz e disparar um feixe de laser vermelho no padrão de interferências gerado no modulador espacial de luz? Essa é a seguinte simulação, no qual se trocou o comprimento de onda $633nm$ (vermelho) por um comprimento de onda, por exemplo, de $550nm$ (verde) ou de $460nm$ (azul). Começando pelo comprimento

de onda no verde ($550nm$), é possível ainda notar o formato do boneco stickman, no entanto verifica-se que a imagem reconstruída não é tão perfeita como a imagem reconstruída pelo holograma de comprimento de onda no vermelho, como o descrito na figura 4.28. Além disso, no comprimento da luz no azul é possível de notar a perda total de nitidez da imagem, indicando que neste caso a luz do laser vermelho simplesmente não difrata, como é verificado pela figura 4.29.

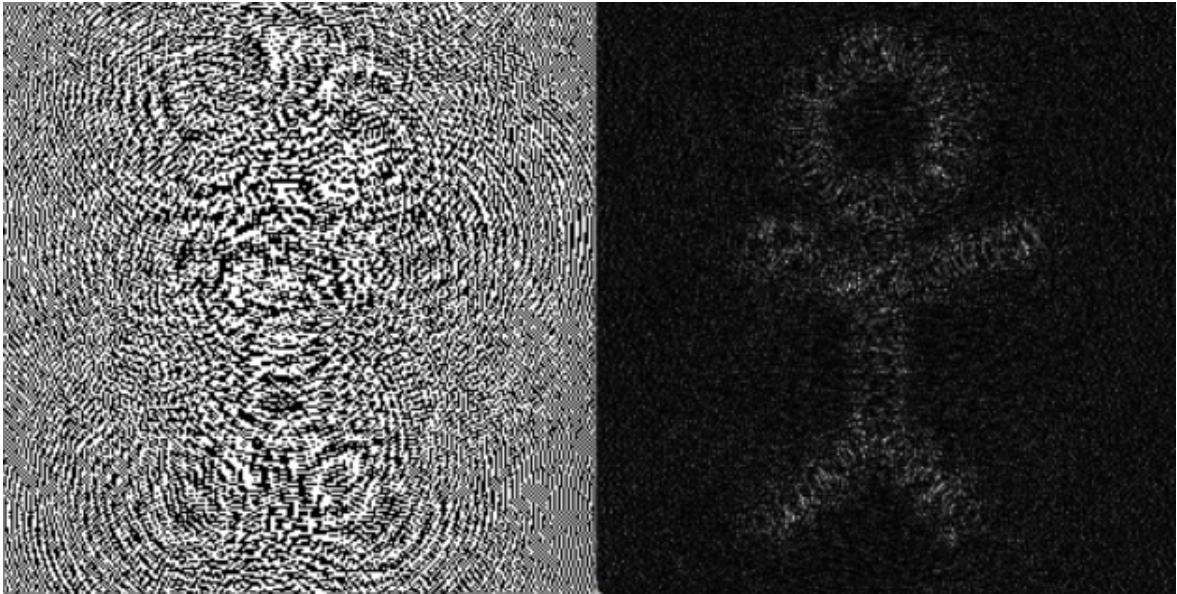


Figura 4.28: Reconstrução da imagem do holograma com comprimento de onda no verde.

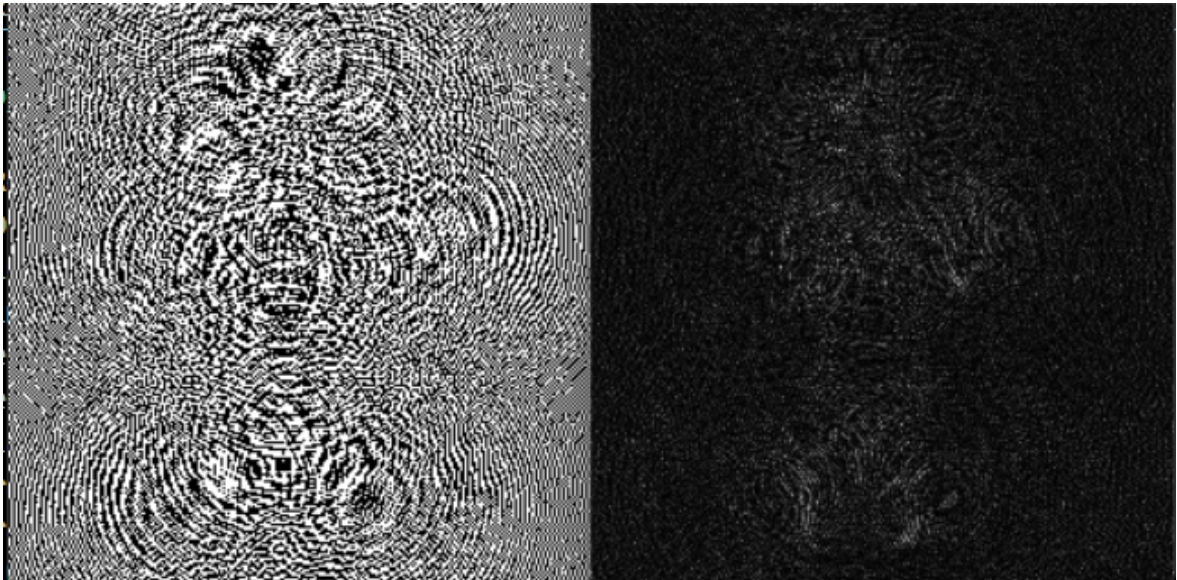


Figura 4.29: Reconstrução da imagem do holograma com comprimento de onda no azul.

Estes resultados mostram a importância do ajuste dos diversos parâmetros de acordo com as características do modulador espacial de luz e do comprimento de onda do laser a utilizar na configuração experimental.

4.3.2 Simulação da reconstrução de vídeos

Antes de se passar para os testes no modulador espacial de luz, considerou-se também fazer simulações em diversos vídeos de modo a verificar a qualidade com o qual as imagens de vídeo são reconstruídas. Primeiro de tudo, convém dizer que os vídeos simulados, a cores ou não, serão exibidos todos eles à escala de cinza, apresentando apenas tons de preto e branco. Indica-se ao leitor que os vídeos usados nas simulações podem facilmente ser achados em plataformas de redes sociais, como por exemplo o youtube.

O código fonte usado para simulação da reconstrução de frame a frame de um vídeo holográfico é dado pelo trecho de código 9, nos anexos.

Na prática o que o código faz é, através de um ficheiro de vídeo holográfico escolhido pelo utilizador, a reconstrução das frames originais do vídeo original, com o detalhe de apresentar esses frames a preto e branco e com áudio incluído. Dessa forma segue-se uma simulação de vídeo de um discurso político a ocorrer numa conferência de um partido político, como o representado na figura 4.30.

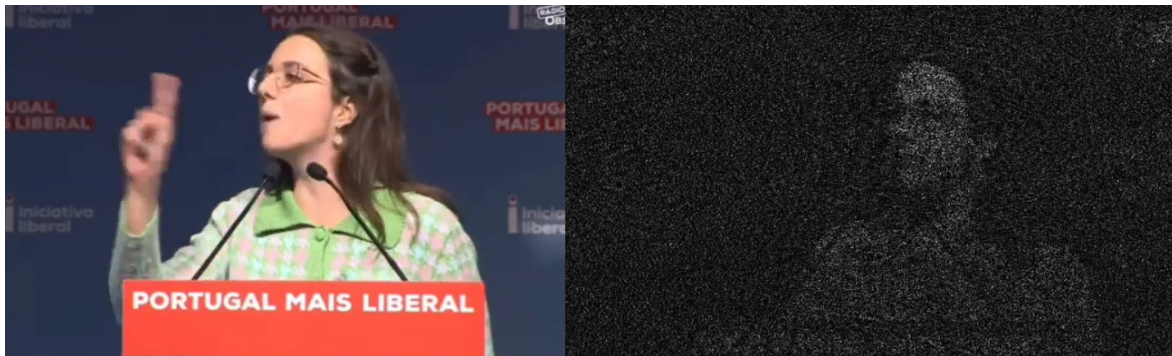


Figura 4.30: Simulação do vídeo do discurso político

É possível ver a reconstrução da imagem original com alguma nitidez relativamente à imagem original propriamente dita. Contudo essa nitidez possui limitações, dado que nem tudo o que está no vídeo de reconstrução é possível visualizar. A resolução desse vídeo foi de 544X360.

A segunda simulação efetuada foi relativamente a um vídeo chamado "Globus-Wyatt Earth". Esse vídeo possui uma resolução ligeiramente superior do que a anterior(640X360). No começo do vídeo, a imagem reconstruída é bem mais nítida que ao vídeo anterior como se mostra no frame apresentado na figura 4.31.

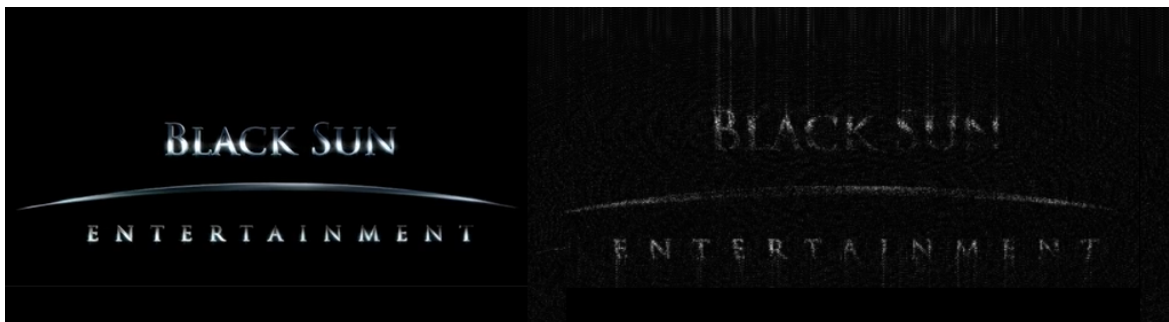


Figura 4.31: Simulação do vídeo do vídeo Globus-Wyatt Earth.

No entanto, é possível reparar que em certas partes do vídeo essa nitidez da imagem é menor, como no frame da figura 4.32.

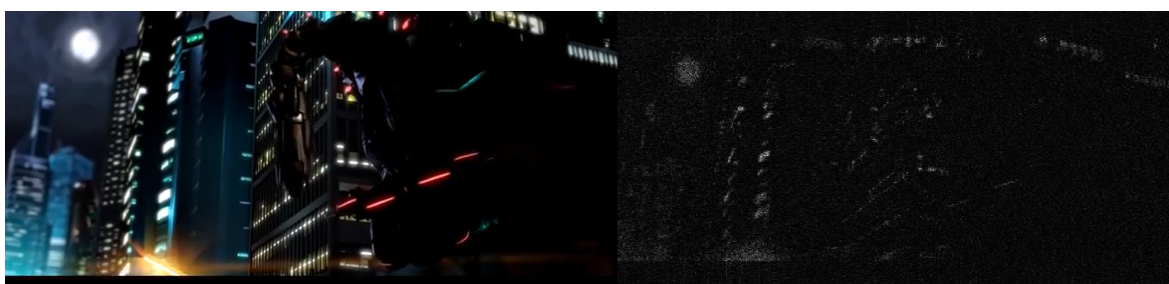


Figura 4.32: Simulação do vídeo do vídeo Globus-Wyatt Earth parte2.

É possível de notar a lua e os edifícios com alguma qualidade, no entanto é complicado de notar a presença de uma máquina, em forma de escorpião, agarrada ao edifício.

A terceira simulação efetuada foi relativamente ao trailer do filme "Quantumania", cujo resultado de uma frame se encontra ilustrado na figura 4.33.



Figura 4.33: Simulação do vídeo do trailer do filme quantumania.

É possível notar o Kang, uma personagem do trailer referido, na imagem reconstruída, no entanto não parece tão nítida como as imagens anteriores, apesar de possuir exatamente a mesma resolução que o vídeo da simulação anterior.

Por fim uma última simulação de vídeo, desta vez relativamente a uma letra de uma música chamada "Sweater Weather" criada pela banda "The Neighbourhood". Tal é descrito na figura 4.34.

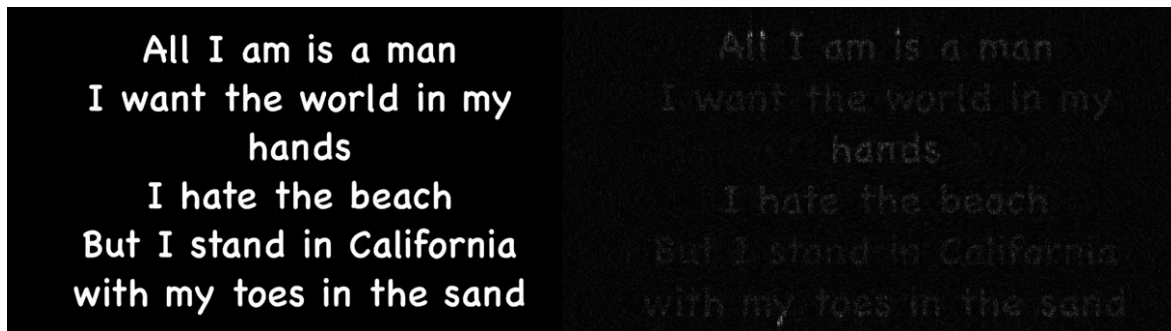


Figura 4.34: Simulação da letra de uma música.

A reconstrução da imagem desta simulação parece mais nítida que das restantes simulações, à exceção daquele vídeo com a imagem a dizer "BLACK SUN" no início. É evidente que a resolução do vídeo não causa tamanha diferença, já que este vídeo possui a mesma resolução relativamente aos dois últimos vídeos simulados anteriormente. É bem possível que haja detalhes na conversão de vídeos e imagens que podem comprometer a qualidade e nitidez da reconstrução das imagens originais. A cor e a intensidade do brilho dos fundos da imagem, a complexidade da imagem, a geometria dos objetos e outros possíveis detalhes, parecem influenciar os resultados do algoritmo utilizado para converter de imagens em hologramas. Em futuros trabalhos seria importante que fosse estudado meios e até mesmo outros algoritmos melhores para otimização da imagem reconstruída. Outro detalhe observado é que o som e o vídeo não estão sincronizados e por vezes o som acaba primeiro que o vídeo, sendo isso uma mera questão de código por resolver. Além disso, nas simulações realizadas é possível verificar que um vídeo de cerca de 7 minutos pode ocupar mais de 1.5GB de memória, o que é muito para um vídeo de 7 minutos. O que se pode tentar fazer é gerar vídeos holográficos para formato .mp4 ao invés do formato .avi.

4.4 SIMULAÇÃO EM TEMPO REAL COM O MODULADOR ESPACIAL DE LUZ

4.4.1 Configuração ótica da experiência

Para a realização da experiência, na qual será reconstruída a imagem original do holograma em tempo real, com recurso a um modulador espacial de luz, ela foi configurada óticamente de modo a que o utilizador tenha total controlo na manipulação a luz, sendo assim possível a visualização em tempo real da reconstrução da imagem original. A configuração do sistema para a projecção da imagem reconstruída foi montada numa mesa ótica como o indicado na figura 4.35.

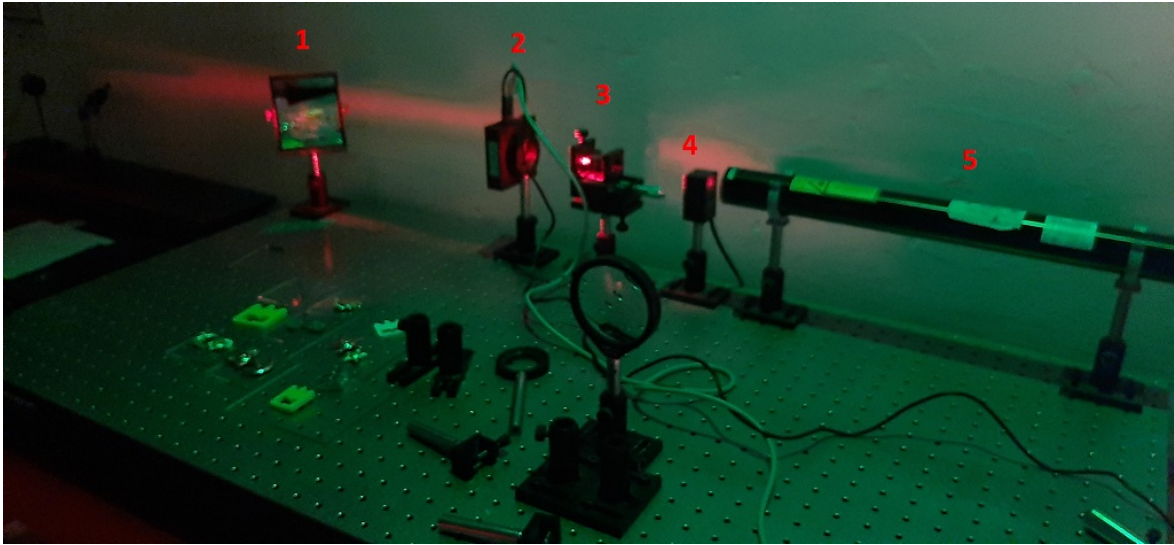


Figura 4.35: Configuração ótica do sistema.

Na configuração ótica representada pela figura 4.35 temos uma fonte de luz monocromática no comprimento de onda no vermelho, laser do tipo He-Ne com o comprimento de onda de 633nm, que se encontra representada pelo número 5. Um obturador ótico que impede a passagem de luz quando se encontra desligada (4). Um filtro espacial para limpar (filtrar) e expandir a luz (3). Um modulador espacial de luz, que, através do envio de padrões de interferência entre um dispositivo qualquer para o próprio modulador espacial de luz forma o respetivo padrão no ecrã do dispositivo (2). Por fim um espelho de primeira superfície para direcionar a luz para onde o utilizador pretende que a imagem projetada seja visualizada (1).

4.4.2 Resultados com imagens

A partir da configuração ótica usada para a experiência em causa e a partir do algoritmo usado para geração de hologramas por computador, foram realizados vários testes de modo a obter a reconstrução de imagens a partir dos hologramas gerados por computador que em comunicação com o modulador espacial de luz são exibidos dentro do ecrã do mesmo sob forma de um padrão de interferências. Desse modo, e considerando que nas características do modulador espacial de luz, como indicado na tabela relativamente às propriedades do mesmo, o pixel pitch ronda os $32\mu m$, sendo assim os parâmetros pelos quais experimentalmente apresenta os melhores resultados para uma imagem com resolução de 500×500 , encontrado através de vários testes, são $\Delta = 32\mu m$, $\lambda = 633\mu m$ e $z = -250mm$. É com isso possível mostrar algumas reconstruções de imagens com alguma nitidez. Alguns exemplos de reconstrução de imagens realizadas são apresentadas nas seguintes figuras (da 4.36 até 4.41).

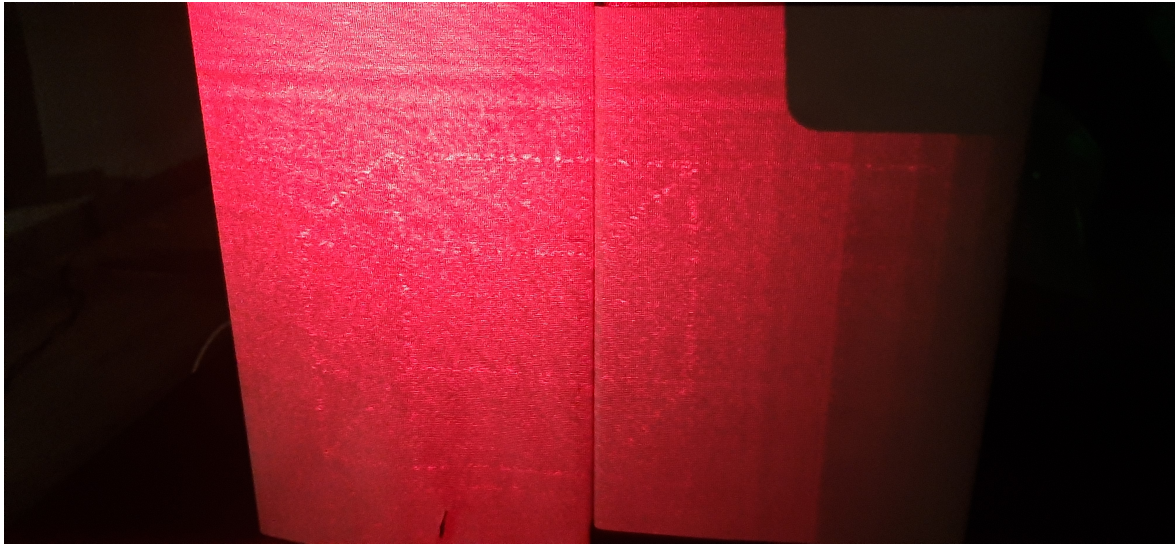


Figura 4.36: Reconstrução da imagem de um cubo.

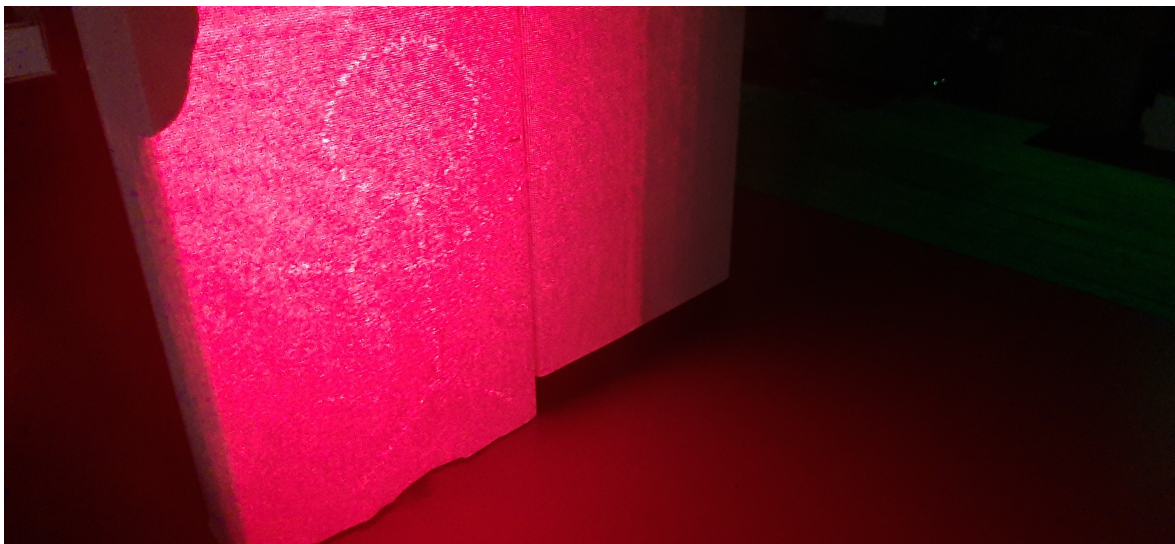


Figura 4.37: Reconstrução da imagem de um stickman.

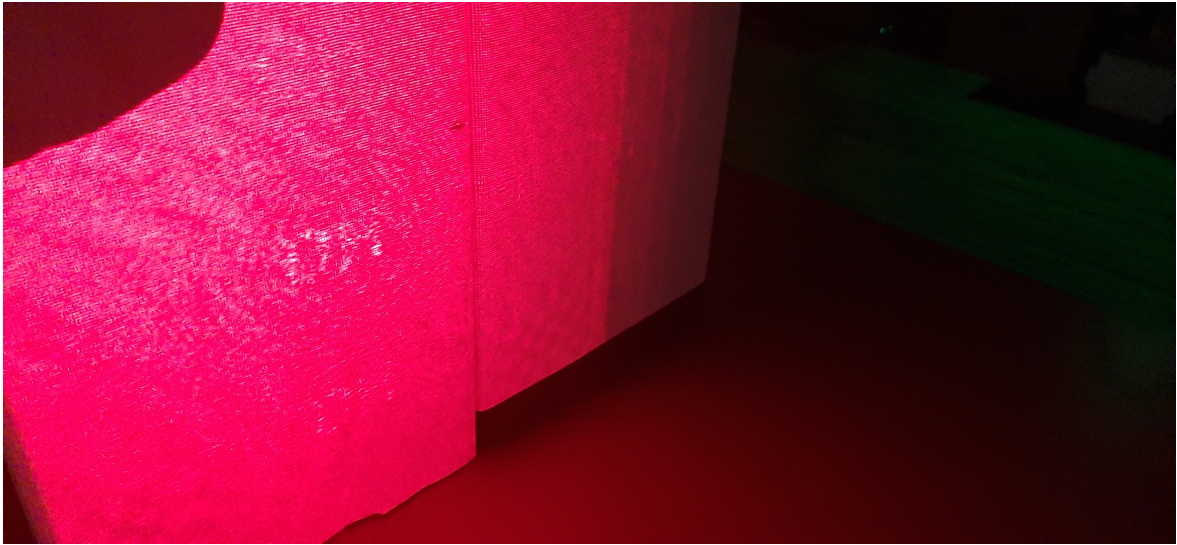


Figura 4.38: Reconstrução da imagem da sigla DH.

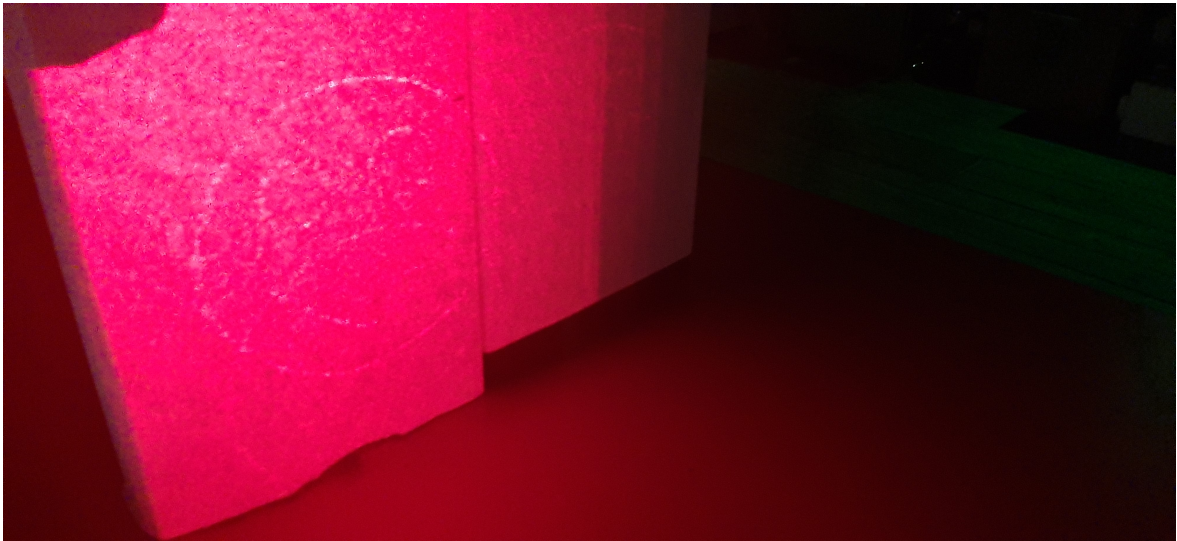


Figura 4.39: Reconstrução da imagem de uma cara.

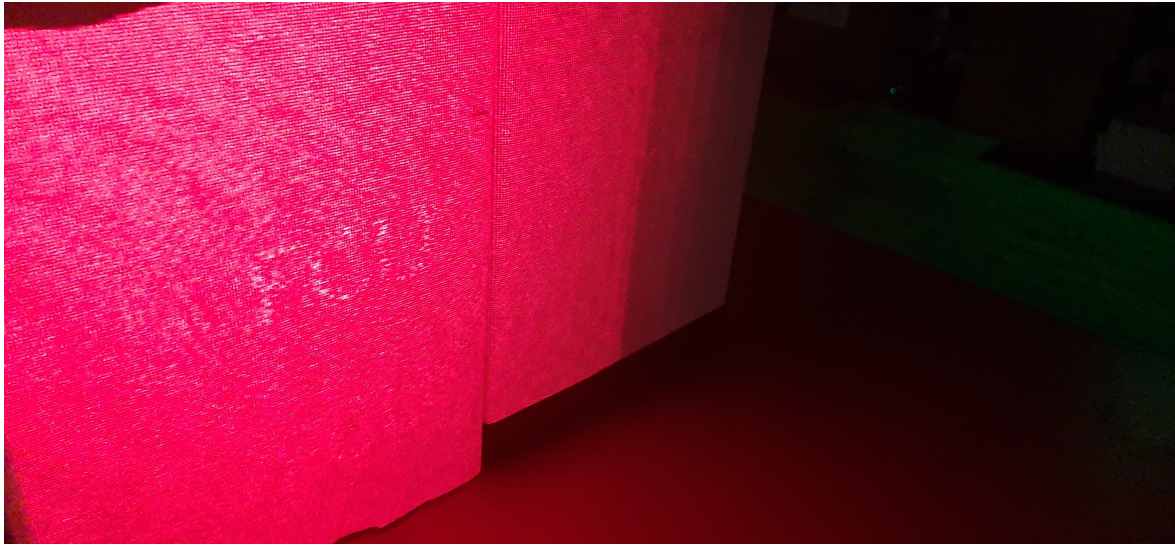


Figura 4.40: Reconstrução da imagem da sigla FCU.

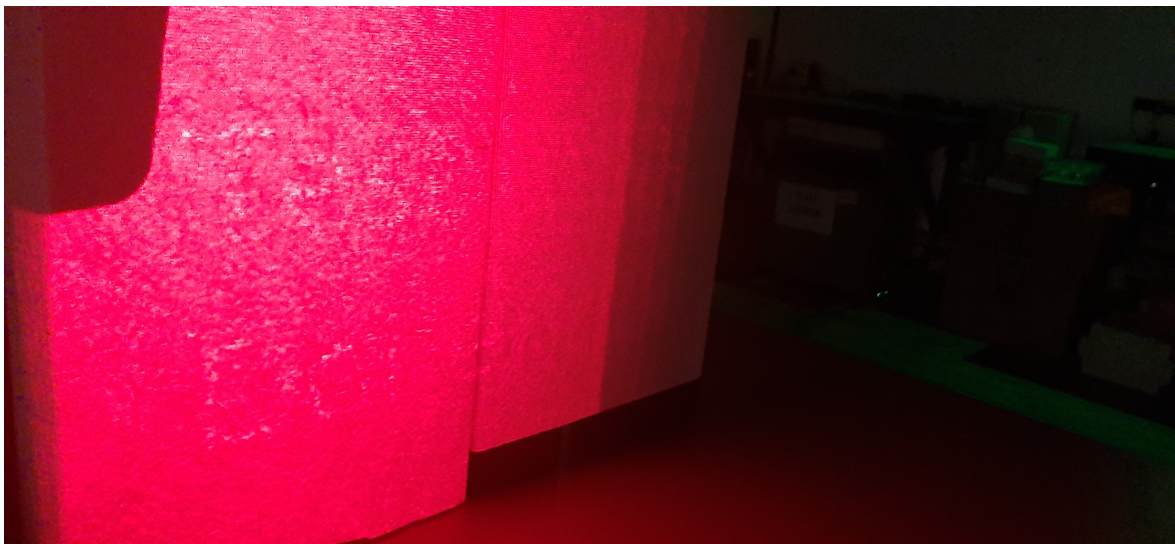


Figura 4.41: Reconstrução da imagem da expressão Test Done.

Através da experiência realizada com imagens isoladas verifica-se que, para uma fase aleatória, as letras tendem a ser menos nítidas que figuras ou sólidos geométricos. Além disso quanto maior a palavra ou a expressão menor será a sua nitidez.

4.4.3 Resultados com vídeos

Relativamente aos testes efetuados por vídeo e, tendo em consideração dos parâmetros obtidos experimentalmente a partir do teste de várias imagens com resolução a 500×500 se consegue visualizar imagens nítidas, também se usou vídeos com a resolução 500×500 através da conversão de todo e qualquer vídeo, usado em simulações anteriores, para essa resolução. No entanto, tais vídeos não tiveram a nitidez desejada na visualização da reconstrução de frame a frame de um vídeo inteiro. Desse modo, pensou-se numa abordagem mais simples onde se usaram vídeos com menor complexidade de figuras e formas para cada frame de

forma a reduzir o tamanho da animação e foi criado um novo programa, que através do uso de uma sequência de imagens forme um vídeo em loop, repetindo o vídeo a cada momento que o mesmo chegue ao fim. Seguindo este raciocínio, foi criado um algoritmo que pode ser encontrado no trecho de código 10, nos anexos.

O resultado do uso deste programa é representado na figura 4.42. É exibido a representação do holograma e da imagem reconstruída numericamente em movimento através do seguimento de várias imagens que se repetem em loop como o sugerido.

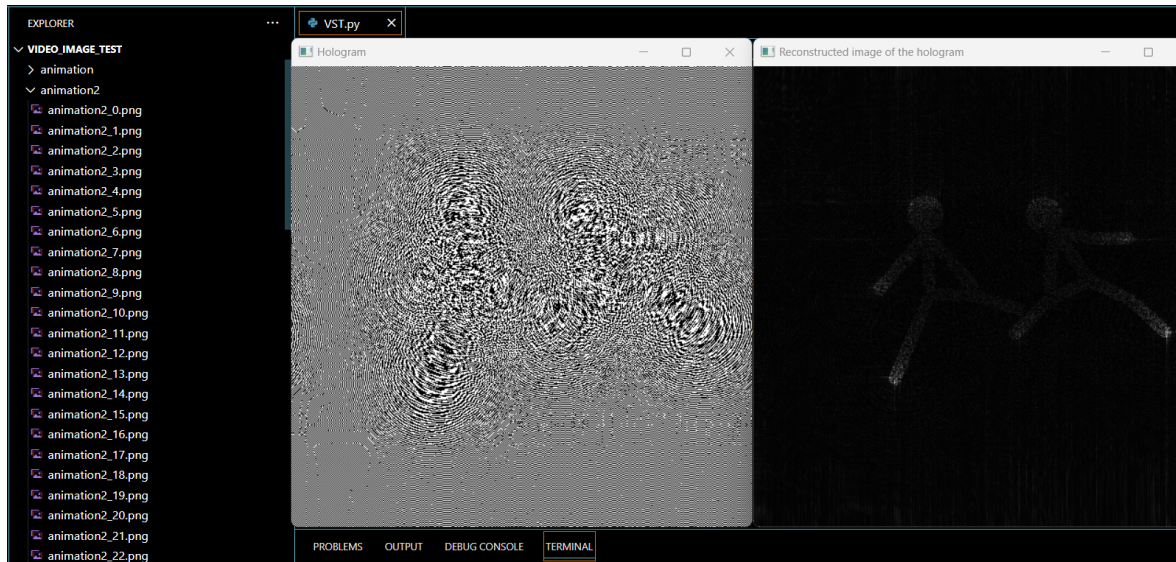


Figura 4.42: Execução do programa.

Os ficheiros das imagens estão dentro de uma pasta cujo nome da mesma representa o nome de todos os ficheiros com a concatenação da string "_número", em que o número vai de 0 até à imagem com o número n-1, sendo n o número total de ficheiros dentro da pasta.

Além disso, foram criadas animações de raiz com recurso ao software **Pivot Animator**[16], pelo qual é possível de criar animações com figuras e formas simples como o stickman e gravar numa pasta com imagens separadas, como o indicado na figura 4.43.

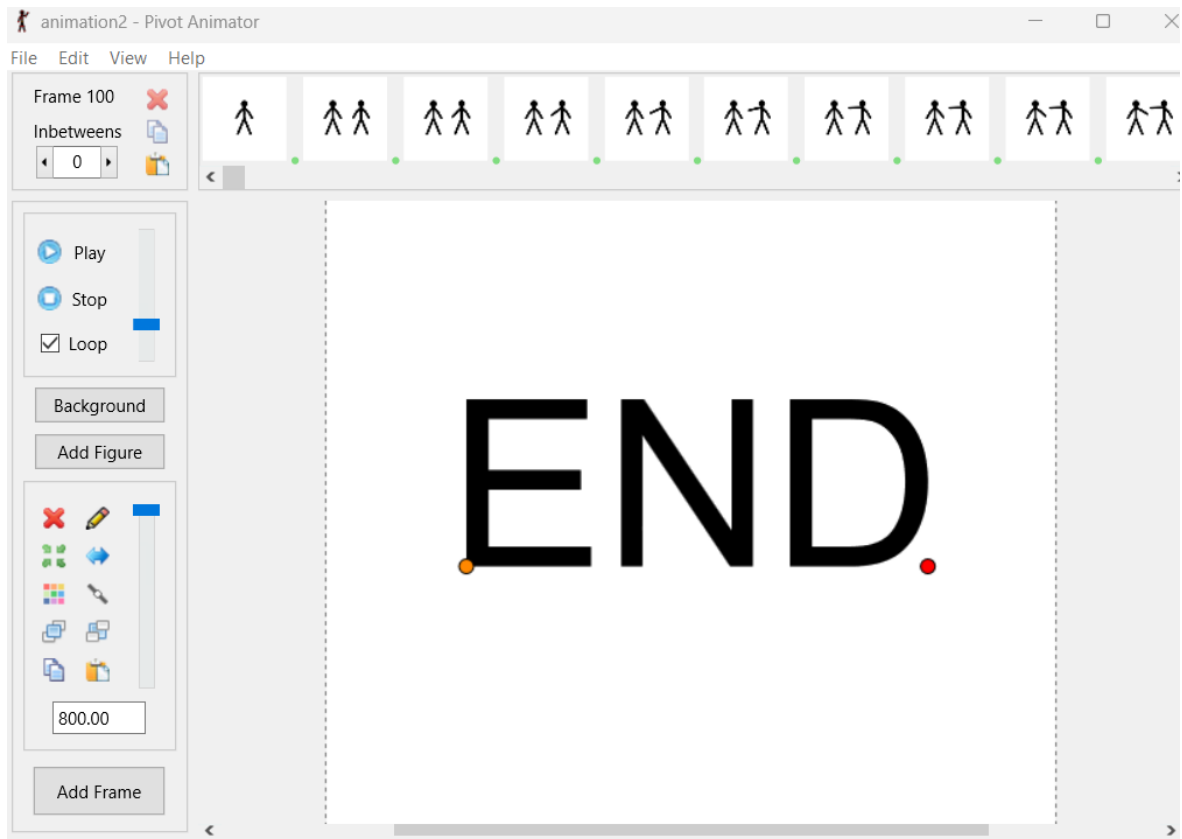


Figura 4.43: Software de animação.

Na simulação da animação em tempo real com recurso ao modulador espacial de luz o resultado foi o movimento do stickman a partir do seguimento de cada frame (ver figura 4.44).



Figura 4.44: Animação com stickman no SLM.

Nota-se com alguma nitidez dois stickmen a reagir um ao outro e a simular uma luta até que um vença e o vídeo acaba com a mensagem "END" e, como o programa se encontra em loop, o vídeo volta ao início. A mensagem "END" (ver figura 4.45) apesar de ser menos nítida que as figuras do stickman é ainda possível de ser notada.



Figura 4.45: Mensagem de fim da animação.

Um outro exemplo de animação construído de raiz, é de uma animação a fazer contagem de 1 até 10 e após a contagem até o número 10 aparece uma mensagem a dizer "YES". Os resultados demonstram, com alguma nitidez as imagens reconstruídas, que é possível de ler e identificar os símbolos, como se compara pelas figuras 4.46 até 4.48.

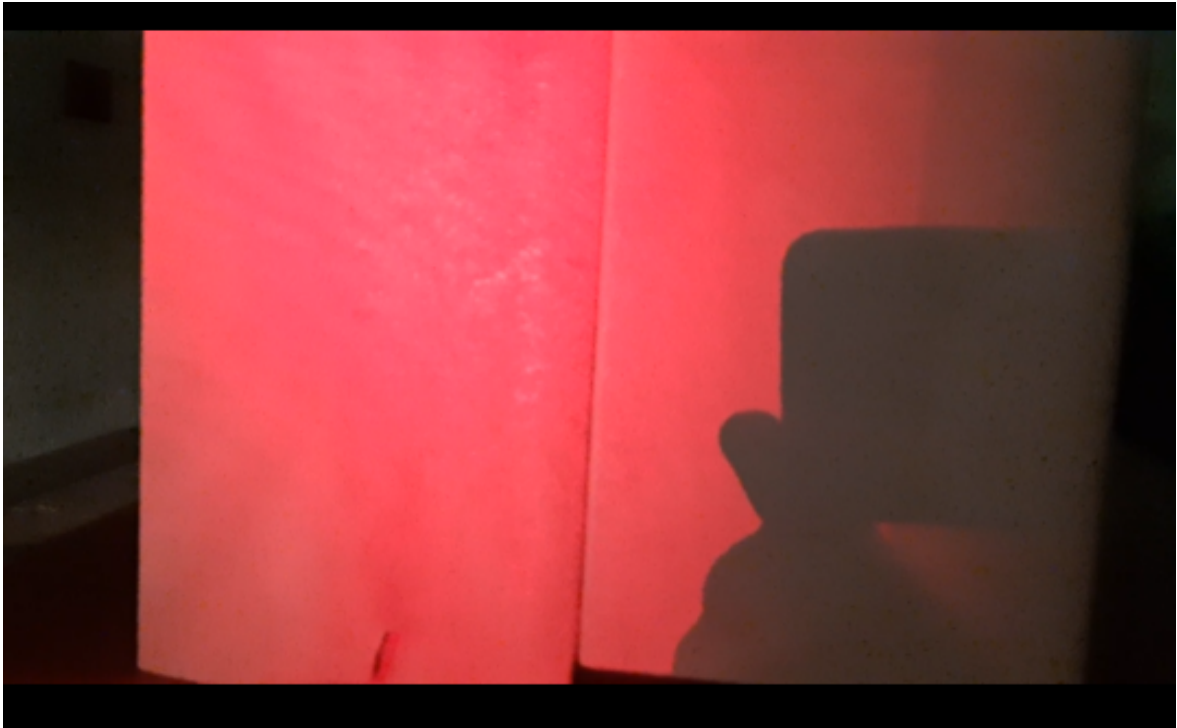


Figura 4.46: Reconstrução do número 1.



Figura 4.47: Reconstrução do número 10.



Figura 4.48: Reconstrução da palavra YES.

Desta forma se pode concluir, através dos resultados obtidos, que é possível, apesar de algumas limitações, produzir alguns vídeos básicos construídos de raiz com recurso a um modulador espacial de luz, considerando ainda que, nas experiências realizadas, a fase usada é aleatória.

4.5 ORÇAMENTO DO PROJETO E ALTERNATIVAS FUTURAS

No projeto inicialmente idealizado, foi realizado uma estimativa sobre os gastos necessários caso o material necessário à realização do projeto não estivesse disponível. Desta forma criou-se uma tabela de preços de cada material e o custo total do projeto.

Equipamento	Custo
Raspberry pi 4	64€
SLM	6000€
Coluna de som	35€
Total	6099€

Tabela 4.2: Tabela de preços

Segundo o que se encontra apresentado nessa tabela, o preço de um modulador espacial de luz, com suporte e laser incluídos, custa praticamente 600x mais do que todo o resto do equipamento proposto no projeto, indicando por consequência a inviabilidade do projeto para comercialização em massa. Desse modo consideram-se algumas alternativas futuras para tornar o projeto mais viável economicamente. Uma das possíveis propostas seria substituir o

modulador espacial de luz por um dispositivo com tecnologia de micro-espelhos, o qual possui o nome comercial de DLP (Digital Light Processor), mais especificamente à utilização do modelo DLPDLCR230NPEVM [11]. Esse modelo possui internamente um FPGA programado e é compatível com o raspberry para além de possuir interfaces RGB de 18bits paralelo, o que também pode ser útil para resolução do problema da imagem sem cor. Tudo isso por um custo de aproximadamente 280€, o que representa um custo significativamente menor do que o custo de um modulador espacial de luz convencional. O modelo proposto é idêntico ao descrito na figura 4.49 e pode ser encontrado no site da Texas Instruments [11].

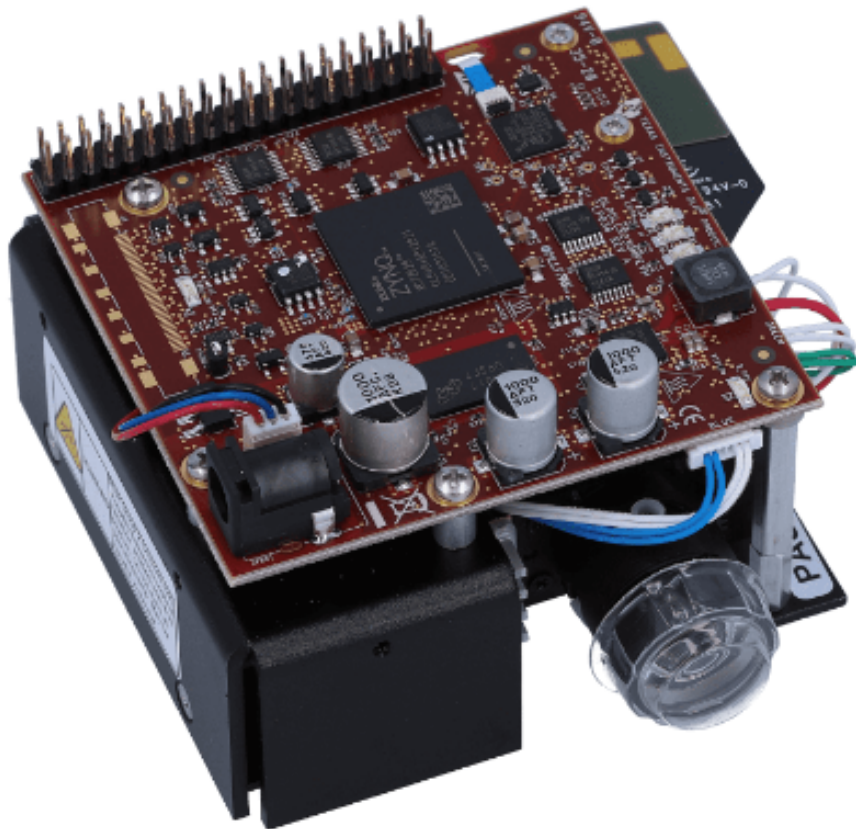


Figura 4.49: Representação de um DLPDLCR230NPEVM [11].

Uma outra alternativa pode ser através do uso de um LCD para a criação de um modulador espacial de luz de baixo custo, com a inclusão de uma board prototipada para manipulação de imagem e vídeo.

Conclusão

5.1 GERAL

Diante das simulações efetuadas em imagem e vídeo, verifica-se que os resultados das simulações são, apesar de algumas limitações na qualidade e nitidez da imagem, bastante satisfatórias e abrem a possibilidade para a sua execução de vídeos em tempo real a partir do modulador espacial de luz. Entretanto com algumas realizações de testes no modulador espacial de luz é possível verificar que ainda existem algumas limitações relativamente à nitidez e ao facto de alguns hologramas, mesmo com os parâmetros usados no qual a reconstrução geral das imagens apresentam melhores resultados, gerarem imagens reconstruídas de pouca qualidade. Isso geralmente resulta com palavras ou expressões grandes e com imagens de maior complexidade, como os frames dos vídeos encontrados em redes sociais como o youtube, apresentando demasiado ruído luminoso o que impede a visualização da reconstrução da imagem com alguma nitidez. O facto da fase ser aleatória no algoritmo apresentado parece ser uma possibilidade que causa esse ruído que impede a visualização correta da reconstrução da imagem. E quando se passa da simulação para a reconstrução da imagem em tempo real é possível de notar essa dificuldade em se gerar uma imagem nítida, ainda maior que numa simulação numérica por computador devido à aleatoriedade da fase que causa ruído luminoso com significativo em tempo real através do modulador espacial de luz.

5.2 LIMITAÇÕES DO SISTEMA

Apesar de faltar realizar mais testes em tempo real com recurso ao modulador espacial de luz, é no entanto possível de verificar algumas limitações existentes no sistema apresentado. No sistema desenvolvido, por exemplo, tem-se o problema da imagem reconstruída não ser a cores, limitando-se apenas à cor do laser utilizado. Este problema pode ser resolvido com recurso a um sistema de RGB de 3 lasers mais as alterações no algoritmo de forma a adicionar a cor azul e a cor verde na geração do holograma, para dessa forma ser possível refratar os comprimentos de onda no vermelho, azul e verde e gerar uma imagem a cores. Um outro

problema é o da dimensionalidade, mesmo que a informação tridimensional esteja guardada no holograma, a projeção da imagem reconstruída continuará a ser bidimensional, ou seja, o holograma não é capaz de produzir profundidade completa na imagem. Tal ocorrência pode ser resultado da imprecisão da fase devido à sua aleatoriedade e também ao facto de se usar matrizes 2D no algoritmo, sendo necessário a utilização de pixels especiais chamados de Voxels. A partir dessa perspectiva, provavelmente são necessários de algoritmos mais sofisticados e, além disso, como já foi referido anteriormente os moduladores espaciais de luz não conseguem projetar hologramas complexos por si só, necessitando do uso de dois moduladores espaciais de luz, um de amplitude e um outro de fase. Uma outra limitação se encontra no custo de uma modulador espacial de luz, normalmente esse dispositivo custa à volta de milhares de euros, tornando esse sistema inviável para ser comercializado. No entanto como os LCDs possuem uma tecnologia idêntica ao SLM pode ser possível tentar construir um modulador espacial de luz de baixo custo com recurso a um LCD de algumas dezenas de euros.

5.3 FUTUROS PROJETOS

Se este projeto em causa obtiver algum interesse por parte da comunidade universitária no que consta a projetos relacionados com holografia e caso pretendam dar uma continuação a esse projeto existe uma variedade de exemplos de possíveis projetos a serem realizados de forma a permitir a otimização e enriquecimento desse projeto. Alguns dos exemplos são:

- Implementação de um software com interface de conversão holográfica de imagem e vídeo;
- Implementação de um sistema eletrónico audiovisual com recurso a dispositivo de micro espelhos(DMD) ou a um LCD;
- Desenvolvimento de um sistema RGB com recurso a lasers para o modulador espacial de luz;
- Otimização da conversão de imagens em hologramas com algoritmos mais sofisticados;

Bibliografia

- [1] Ting-Chung Poon e Jung-Ping Liu(2014): Introduction to Modern Digital Holography.
- [2] U.Schanrs e W. Jueptner: Digital Holography, Digital Hologram Recording, Numerical Reconstruction, and Related Techniques.
- [3] Marcos R. R. Gesualdi, Algoritmos simplificados de hologramas computacionais: I – para geração de imagens 3D, Dezembro de 2022.
- [4] Derek Huang, Adam Roberts, Niranjana Shivaram, Arvinder S. Sandhu and Henry Timmers, A low-cost spatial light modulator for use in undergraduate and graduate optics labs, Março de 2022.
- [5] ERDEM SAHIN, ELENA STOYKOVA, ANI MÄKINEN and ATANAS GOTCHEV, Computer-Generated Holograms for 3D Imaging: A Survey, Março de 2022.
- [6] D.Gabor, A new microscopic principle, Nature 161, 777-778(1948).
- [7] Optical reconstruction of computer-generated holograms 3D scenes via spatial light modulators.
- [8] https://zulko.github.io/moviepy/getting_started/quick_presentation.html, pesquisado em Março de 2023.
- [9] https://matham.github.io/ffpyplayer/getting_started.html, pesquisado em Março de 2023.
- [10] https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html, pesquisado em Março 2023.
- [11] <https://www.ti.com/tool/DLPDLCR230NPEVM>, pesquisado em Março de 2023.
- [12] <https://holoeye.com/spatial-light-modulators/discontinued-devices>, pesquisado em Abril de 2023.
- [13] Computer-Generated Holography Using a Digital Signal Processor, Youchao Wang, Daoming Dong, Andrew Kadis, Peter J. Christopher, and Timothy D. Wilkinson Centre for Molecular Materials, Photonics and Electronics, University of Cambridge.
- [14] Complex-Amplitude Holographic Projection With a Digital Micromirror Device (DMD) and Error Diffusion Algorithm, Shuming Jiao , Dongfang Zhang, Chonglei Zhang, Yang Gao,

Ting Lei and Xiaocong Yuan.

[15] Computer simulated holography and computer generated holograms, Seymour Trester.

[16] <https://pivotanimator.net>, pesquisado em Maio de 2023

[17] Holography without photography, Thad G. Walker

[18] www.realvnc.com, pesquisado em Junho de 2023

[19] winscp.net, pesquisado em Junho de 2023

CAPÍTULO 7

Anexos

```

clear all;
close all;
Ii=imread('DH_256_256.bmp');
figure;
imshow(double(Ii));
title('Object pattern');
axis off

Ii=double(Ii);
PH=rand([256,256]);
Ii=Ii.*exp(2i*pi*PH);
M=512;
I=zeros(512);
I(128:383,128:383)=Ii;

z=15; % (cm, distance) Tested for 1, 5, 15
w=6500*10^-8; % (cm, wavelength)
delta=0.005; % cm, pixel size 50um
r=1:M;
c=1:M;
[C,R]=meshgrid(c,r);

% Forward propagation (650nm)
p=exp(-1i*2*pi*z.*((1/w)^2-(1/M/delta)^2.*(C-M/2-1).^2-(1/M/delta)^2.*(R-M/2-1).^2).^0.5);
A0=fftshift(iff2(fftshift(I)));
Az=A0.*p;
E=fftshift(fft2(fftshift(Az))); % 1st order of the hologram

% Reconstruction (650nm)
p=exp(-1i*2*pi*(-z).*((1/w)^2-(1/M/delta)^2.*(C-M/2-1).^2-(1/M/delta)^2.*(R-M/2-1).^2).^0.5);
A1=fftshift(iff2(fftshift(E)))
Az1=A1.*p;
R1=fftshift(fft2(fftshift(Az1)))
R1=(abs(R1)).^2;
figure;
imshow(R1/max(max(R1)));
title('Reconstructed image(650nm)');
axis off

% Reconstruction(450nm-650nm)
dw=50;
IMA=zeros(512,512);
for g=0:40;
    w2=(6500-dw*g)*10^-8; % reconstruction wavelength
    E2=E.*exp(2i*pi*sind(10)*(w-w2)/w/w2.*R*delta);
    % phase mismatch due to the wavelength shift
    p=exp(-1i*2*pi*(-z).*((1/w2)^2-(1/M/delta)^2.*(C-M/2-1).^2-(1/M/delta)^2.*(R-M/2-1).^2).^0.5);
    Az2=iff2(fftshift(E2)).*(fftshift(p));
    R2=fftshift(fft2(Az2));
    R2=(abs(R2)).^2; % summation of all wavelengths
    IMA=IMA+R2;
end
IMA=IMA/max(max(IMA));
figure;
imshow(IMA);
title('Reconstructed image(white light)');

```

Código 1: Código para simulação do holograma de imagem

```

clear all;
close all;
Ii=imread("DH_256_256.bmp");
Ii=double(Ii);

M=512;
I=zeros(M);
I(128:383,128:383)=Ii; % Zero-padding

figure;
imshow(mat2gray(abs(I)));
title('Object pattern');
axis off;

% Produce the Fourier Hologram
r=1:M;
c=1:M;
[C,R]=meshgrid(c,r);
O=fftshift(iff2(fftshift(I)));
R=ones(M,M);
R=R*max(max(max(abs(O))));
R=R.*exp(2i*pi.*C/4); % Tilted reference light
H=(abs(O+R)).^2 % Fourier Hologram
figure;
imshow(mat2gray(abs(H)));
title('Fourier Hologram');
axis off;

% Reconstruction
U=fftshift(iff2(fftshift(H)));
figure;
imshow(900.*mat2gray(abs(U)));
title('Reconstructed image');
axis off;

```

Código 2: Código de matlab para simulação do holograma de Fourier

```

clear all; close all;
I=imread('DH_64_64_right.bmp', 'bmp');
I=double(I);
figure; imshow(abs(I));
title('Original object')
PH=rand([64,64]);
I=I.*exp(2i*pi*PH);
FTS=fftshift(iff2(fftshift(I)));
A=abs(FTS);
figure;
imshow(mat2gray(A));
title('Object spectrum');

A=A./max(max(A))*15;
A=round(A);
B=angle(conj(FTS));
B=B-(min(min(B)));
B=B./max(max(B))*7;
B=round(B);
H=zeros(1024);
for m=1:64
    for n=1:64
        P=zeros(16);
        a=A(m,n);
        b=B(m,n);
        c=fix(a/2);
        d=rem(a,2);
        P(9-c:8+c+d,(1+b):(9+b))=1;
        H(16*(m-1)+1:16*(m-1)+16,16*(n-1)+1:16*(n-1)+16)=P;
    end
end
figure;
imshow(H);
title('Detour-phase CGH');
imwrite(H,'1AA.jpg','jpg');
% Reconstruction (FFT)
R=fftshift(iff2(fftshift(H)));
figure;
imshow(100.*mat2gray(abs(R)));
title('Reconstructed image');

```

Código 3: Código de matlab para geração do holograma de desvio de fase.

```

clear all;
close all;
I=imread('Myself.jpg', 'jpg');
I=double(I);
I=I./max(max(I));
figure;
imshow(mat2gray(I));
title('Original object');
PH=rand([480,640]);
I=I.*exp(2i*pi*PH);
FTS=fftshift(iff2(fftshift(I)));
A=abs(FTS);
figure;
imshow(mat2gray(A));
title('Spectrum modulus');
B=angle(FTS);
figure;
imshow(mat2gray(B));
title('Spectrum phase');
R=fftshift(iff2(fftshift(exp(-1j*B))));
figure;
imshow(mat2gray(abs(R)));
title('Reconstructed image');

```

Código 4: Código de matlab para gerar o holograma kinoform

```

clear all;
close all;
I1=imread('FCU_test.bmp','bmp');

I1=double(I1);
delta=0.01266; % sampling distance 12.66um
lambda=0.000633; % wavelength 0.633um

% add a random phase to the object
Rr=rand(500);
Rr=exp(1i*2*pi.*Rr);
IO=I1.*Rr;
figure;
imshow(mat2gray(abs(IO)));
title('Original object')
axis off;

[M N]=size(IO);
z=-80+lambda/4; %(nm, distance)
r=1:M;
c=1:N;
[C,R]=meshgrid(c,r);
p=exp(-1i*2*pi*z.*((1/lambda)^2-(1/M/delta)^2.*(C-N/2-1).^2-(1/N/delta)^2.*(R-M/2-1).^2).^0.5);
AO=fftshift(iff2(fftshift(IO)));
Az=A0.*p;
E=fftshift(fft2(fftshift(Az))); % propagation

% binary hologram
Hr=real(E);
Hr=(Hr>0);
Hi=imag(E);
Hi=(Hi>0);

% reconstruction of the binary hologram Hr
Ir=fftshift(fft2(fftshift(conj(p)).*iff2(fftshift(Hr))));
Ir=Ir.*conj(Ir);
Ir=Ir/(max(max(Ir)));

% reconstruction of the synthetic hologram
Hc=Hr+1i*Hi;
I=fftshift(fft2(fftshift(conj(p)).*iff2(fftshift(Hc))));
I=I.*conj(I);
I=I./max(max(I));
figure;
imshow(mat2gray(Hr));
title('Binary hologram B_r');
axis off;
figure;
imshow(mat2gray(Hi));
title('Binary hologram B_i');
axis off;
figure;
imshow(3*Ir);
title('Reconstructed image of B_r');
axis off;
figure;
imshow(3*I);
title('Reconstructed image of the synthetic hologram');
axis off;

```

Código 5: Código de matlab para gerar dois hologramas binários para sintetizar e exibir o holograma

```

from tkinter import*
from PIL import Image, ImageTk
import numpy as np
import cv2 as cv
import sys
FILE=''
def Holo_save(img):
    word=FILE.split('.')
    cv.imwrite('Holo_'+word[0]+'.bmp',img)
def Binary_Hologram(img):
    #Algoritmo de conversão da imagem para o holograma binário
def Holo_converter(img):
    #Algoritmo para geração, normalização, print e gravação do holograma
def print_original_image(img):
    # Print da imagem original na interface
def print_Holo_image(img):
    # Print do holograma na interface
def File_search(name):
    # Faz pesquisa do ficheiro, caso não exista ocorre um erro
if __name__=='__main__':
    #Função principal, todo o programa é executado a partir dessa função.
}

```

Código 6: Software com interface para converter imagens em hologramas

```

import numpy as np
import cv2 as cv
from moviepy.video.io import ffmpeg_tools
import moviepy.editor as mp
import sys
def Binary_Hologram(img):
    #Algoritmo de conversão da imagem para o holograma binário
FILE=input("Indique nome do ficheiro de video: ")
try:
    cap = cv.VideoCapture(FILE)
    if cap is None or not cap.isOpened():
        raise ConnectionError
except ConnectionError:
    print("[ErrorType]: {}".format(ConnectionError))
    sys.exit(0)
name=FILE.split('.')
new_name='Holo_'+name[0]+'.avi'
fps = cap.get(cv.CAP_PROP_FPS)
w=int(cap.get(3))
h=int(cap.get(4))
print('w:',w)
print('h:',h)
# Define the codec and create VideoWriter object
fourcc = cv.VideoWriter_fourcc(*'MJPG')
out = cv.VideoWriter(new_name, fourcc, fps, (w,h),0)
audio=mp.AudioFileClip(FILE)
audio.write_audiofile(name[0]+'.mp3')
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    # write the flipped frame
    gray=cv.cvtColor(frame,cv.COLOR_BGR2GRAY)
    Holo=Binary_Hologram(gray)
    Holo=abs(Holo)
    Holo=cv.normalize(Holo, None, 0, 255, cv.NORM_MINMAX, cv.CV_8U)
    out.write(Holo)
    cv.imshow('Holograma', Holo)
    if cv.waitKey(1) == ord('q'):
        break
cap.release()
out.release()
cv.destroyAllWindows()
txt=new_name.split('.')
ffmpeg_tools.ffmpeg_merge_video_audio(new_name,name[0]+'.mp3',txt[0]+'_audio.'+txt[1])

```

Código 7: Software para conversão do vídeo numa sequência de hologramas


```

import cv2 as cv
import numpy as np
import sys
def image_Reconstruction(img):
    #Algoritmo de reconstrução da imagem original
if __name__=="__main__":
    FILE=input("Indique nome do ficheiro da imagem: ")
    try:
        img = cv.imread(FILE)
        if img is None:
            raise ConnectionError
    except ConnectionError:
        print("[ErrorType]: {}".format(ConnectionError))
        sys.exit(0)
    cv.imshow("Display window", img)
    gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
    k = cv.waitKey(0)
    res=image_Reconstruction(gray)
    cv.imshow("Reconstructed image of the synthetic hologram",res)
    k=cv.waitKey(0)

```

Código 8: Software para simulação da imagem reconstruída

```

import numpy as np
import cv2 as cv
import sys
from ffpyplayer.player import MediaPlayer
def image_Reconstruction(img):
    # Algoritmo de reconstrução da imagem
FILE=input("Indique nome do ficheiro de video: ")
try:
    cap = cv.VideoCapture(FILE)
    if cap is None or not cap.isOpened():
        raise ConnectionError
except ConnectionError:
    print("[ErrorType]: {}".format(ConnectionError))
    sys.exit(0)
player=MediaPlayer(FILE)
while cap.isOpened():
    ret, frame = cap.read()
    audio_frame,val=player.get_frame()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    cv.imshow("frame", gray)
    image_Reconstruction(gray)
    if cv.waitKey(1) == ord('q'):
        break
cap.release()
cv.destroyAllWindows()

```

Código 9: Software de reconstrução do vídeo holográfico

```

import numpy as np
import cv2 as cv
def image_Reconstruction(img):
    #Reconstrução da imagem original a partir do holograma
def Binary_Hologram(img):
    #Algoritmo de conversão da imagem num holograma
if __name__=='__main__':
    count=0
    while True:
        FILE="animation2/animation2_"+str(count)+".png"
        img = cv.imread(FILE)
        gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
        gray=cv.bitwise_not(gray)
        Holo=Binary_Hologram(gray)
        Holo=cv.normalize(Holo, None, 0, 255, cv.NORM_MINMAX, cv.CV_8U)
        cv.imshow('Hologram',Holo)
        res=image_Reconstruction(Holo)
        cv.imshow('Reconstructed image of the hologram',res)
        if cv.waitKey(100) == ord('q'):
            break
        count=count+1
    if count==101:
        count=0

```

Código 10: Software com interface para converter imagens em hologramas