

Holografia Digital

Ana Silva, M15432

Resumo—Este relatório detalha o processo de reconstrução numérica e a geração de vistas a partir de hologramas a cores, disponíveis na base de dados B-Com. O objetivo principal foi simular a observação dos hologramas a partir de diferentes posições e profundidades. Neste trabalho foram utilizados os hologramas 'Dices' e 'Biplane' (4096×4096), cujos dados foram processados utilizando o algoritmo *Angular Spectrum Method* (ASM) implementado em *Python*. A metodologia envolveu a divisão do holograma numa grelha 4×4 para gerar 16 vistas e a variação da distância de propagação (z) para simular três planos de foco distintos. Os resultados demonstraram a eficácia do ASM na focagem seletiva de objetos em diferentes profundidades e a compilação das sequências de vistas em vídeos demonstrou a visualização dos hologramas a partir de diferentes posições.

Index Terms—Holograma, ASM, vistas.

I. INTRODUÇÃO

Este relatório documenta o processo de reconstrução numérica e geração de sequências de vistas a partir de hologramas a cores, provenientes da base de dados B-Com [1].

A tarefa principal trata-se da geração de vistas que simulam o movimento de observação do holograma a partir de diferentes ângulos e a utilização do algoritmo *Angular Spectrum Method* (ASM) para a reconstrução da imagem em três planos de foco distintos.

Este documento está estruturado em torno da descrição detalhada de cada fase do trabalho, abrangendo desde a seleção dos hologramas e das ferramentas utilizadas, a metodologia de reconstrução ASM e geração das vistas e vídeos, até à demonstração dos resultados obtidos e as conclusões relativas ao trabalho realizado.

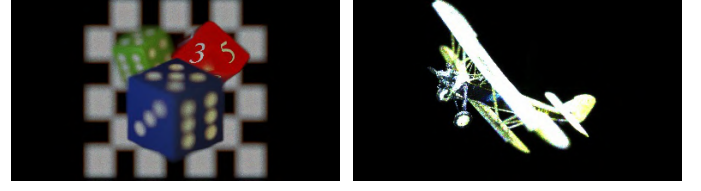
Todo o código utilizado para a geração das vistas e reconstrução do holograma está disponível em [2], juntamente com os resultados obtidos.

II. METODOLOGIA E IMPLEMENTAÇÃO

A. Hologramas Escolhidos

A primeira parte deste trabalho foi a escolha de dois hologramas a cores. Estes foram obtidos a partir da base de dados da B-Com [1], ambos disponíveis em [3]. Estes hologramas foram sintetizados através de um algoritmo desenvolvido pelo *Institute of Research & Technology* (IRT) b<>com.

O primeiro holograma escolhido foi o 'Dices' conforme apresentado na Figura 1a. A sua escolha justifica-se pela sua composição, apresentando três dados de cores e distâncias variadas, o que se mostrou interessante para testar a sua reconstrução em diferentes distâncias de foco. Este holograma apresenta os valores de $0.4 \mu\text{m}$ de *pixel pitch* e a localização da cena entre 0.164 e 0.328 cm. O segundo holograma escolhido foi o 'Biplane', ilustrado na Figura 1b, escolhido devido à melhor qualidade dos resultados obtidos em comparação



(a) Holograma 'Dices'.

(b) Holograma 'Biplane'.

Figura 1: Hologramas escolhidos.

às restantes opções. Este possui um $1 \mu\text{m}$ de *pixel pitch* e a localização da cena vai de 0.936 a 1.24 cm.

Para ambos os *datasets* escolhidos foi utilizada a resolução de 4096×4096 pixeis. Quanto à cor estes *datasets* têm os seguintes comprimentos de onda (λ) para os respetivos canais RGB:

- Vermelho: $\lambda_R = 640$ nm;
- Verde: $\lambda_G = 532$ nm;
- Azul: $\lambda_B = 473$ nm.

Os dados disponibilizados por estes hologramas consistiam em dois ficheiros *.exr*, que representam as partes real e imaginária da onda do objeto no plano do holograma. Estes ficheiros estão codificados em *floats* de 32 bits por canal.

B. Ferramentas

A execução da reconstrução dos hologramas 4096×4096 exige um ambiente de computação com recursos significativos. Devido às limitações de *hardware* e ao facto de ter tido problemas na execução local do código, foi tomada a decisão de utilizar o *Google Colab* para todo o trabalho.

Para a reconstrução numérica do holograma, utilizou-se o algoritmo ASM, cuja implementação foi realizada através da função *propASM* disponibilizada pela professora e visível em 1.

```
def propASM(u1, pitch, lbd, z):
    M, N = u1.shape
    dx = pitch
    dy = pitch
    fx = np.linspace(-1.0 / (2.0 * dx), 1.0 / (2.0 * dx), N, endpoint=False)
    fy = np.linspace(-1.0 / (2.0 * dy), 1.0 / (2.0 * dy), M, endpoint=False)
    FX, FY = np.meshgrid(fx, fy)
    w = np.sqrt(np.maximum(0.0, (1.0 / lbd)
        ** 2 - FX**2 - FY**2))
    H = np.exp(-2j * np.pi * w * z)
    U1 = np.fft.fftshift(np.fft.fft2(u1))
    U2 = U1 * H
    u2 = np.fft.ifft2(np.fft.ifftshift(U2))
    return u2
```

Listing 1: Implementação do ASM em *Python*.

C. Geração das Vistas

A geração das 16 vistas foi implementada com o objetivo de simular o que seria visto ao olhar para o holograma a partir de 16 posições ligeiramente diferentes.

O processo começa por dividir o plano do holograma original de 4096×4096 numa grelha 4×4 . Para cada uma destas 16 posições, é isolada a parte correspondente à vista e o resto da matriz é preenchida a zeros. As dimensões da vista são calculadas como a dimensão total dividida pelo número de linhas e colunas. Assim, para os hologramas escolhidos que são 4096×4096 a implementação foi feita com vistas pequenas de 1024×1024 , sem qualquer sobreposição.

Para experimentação e avaliação de potencial melhoria, foi realizada uma implementação adicional para o holograma 'Dices' utilizando vistas sobrepostas de 2048×2048 pixels, e sobreposição de 682 pixels. A única alteração necessária no código foi o ajuste do cálculo das coordenadas iniciais para a vista. No entanto, os resultados não mostram uma melhoria significativa na qualidade comparando aos resultados anteriores. Então, devido às restrições de tempo e recursos computacionais juntamente com a ausência de um ganho significativo de qualidade, não se alargou a implementação de vistas sobrepostas ao restante trabalho.

Para cada uma das 16 vistas, é aplicada a reconstrução numérica utilizando a função *propASM* separadamente para os três canais de cor. Após a propagação, os valores são todos normalizados. O parâmetro de distância de propagação, z , determina o plano de foco da vista gerada. Os valores de z escolhidos para cada holograma, baseados na localização da cena e de modo a explorar a profundidade, foram:

- Holograma 'Dices': $Z_1 = 0.0018$ m, $Z_2 = 0.0025$ m, e $Z_3 = 0.0030$ m;
- Holograma 'Biplane': $Z_1 = 0.0100$ m, $Z_2 = 0.0115$ m, e $Z_3 = 0.0122$ m.

Este processo é repetido para cada um dos três focos definidos, resultando num total de 48 imagens (16 vistas \times 3 focos).

D. Geração dos Vídeos

A etapa final deste trabalho consistiu na compilação das 6 sequências de vistas, três para cada holograma, em vídeos. Para este propósito utilizei a *framework FFMpeg* [4], uma ferramenta de linha de comandos de código aberto, já utilizada em trabalhos anteriores. Com esta *framework* foi possível gerar o vídeo a partir de um único comando. O comando utilizado foi o seguinte:

```
ffmpeg -framerate 2 -pattern_type glob -i
'*.png' -c:v libx265 -pix_fmt yuv420p
video.mp4
```

Neste comando, o argumento *-framerate 2* define que o vídeo deve ser reproduzido a 2 fps. A parte do *-pattern_type glob -i '*.png'* permite ao *FFmpeg* a ler todas as imagens PNG presentes na diretoria como *frames* de entrada. Finalmente, o *codec libx265* foi escolhido para a compressão do vídeo e o *yuv420p* para o formato dos píxeis, gerando o ficheiro de saída *video.mp4*.

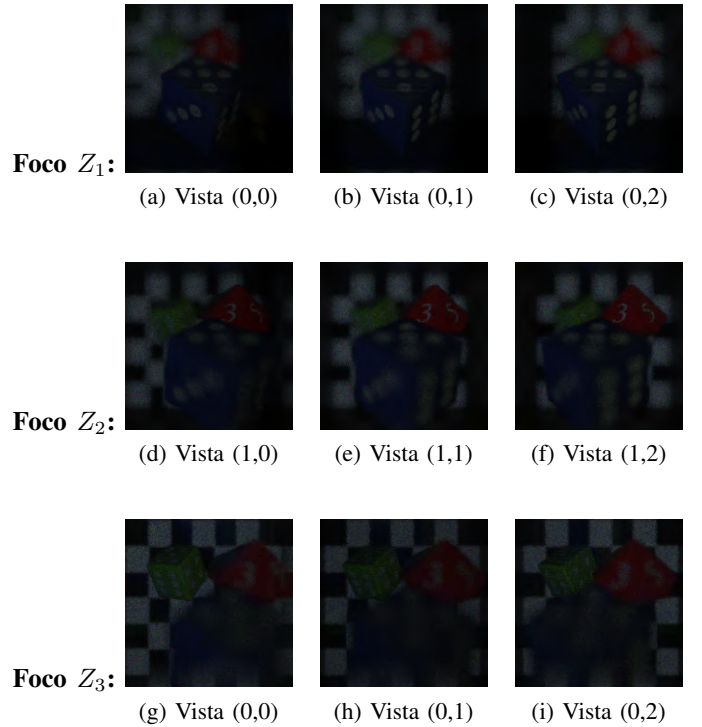
III. RESULTADOS

Nesta secção são apresentados os resultados obtidos na geração das vistas. Estes resultados são demonstrados para cada holograma exemplificados por três vistas para cada distância de foco z .

A. Holograma 'Dices'

A Figura 2 ilustra algumas das vistas geradas para o holograma 'Dices', demonstrando o efeito da focagem seletiva em três planos de focos distintos. O holograma 'Dices' foi ideal para este exemplo, uma vez que contém objetos posicionados a diferentes distâncias. Assim, através da variação da distância de propagação, z , no algoritmo ASM, foi possível simular a focagem em profundidade. No Foco Z_1 (vistas 2a a 2c), a focagem é feita no plano do dado azul, enquanto os dados mais distantes continuam desfocados. Ao propagar para o Foco Z_2 (vistas 2d a 2f), o dado vermelho é focado. Finalmente, no Foco Z_3 (vistas 2g a 2i), o foco é feito para o plano mais distante, onde o fundo e o dado verde demonstram maior nitidez. Assim, demonstra-se a capacidade do ASM em simular o efeito de profundidade através da reconstrução numérica. Adicionalmente, as diferentes vistas permitem visualizar através de ângulos diferentes toda a cena.

Figura 2: Demonstração das vistas geradas para os três planos de foco (Z_1, Z_2, Z_3) para o holograma 'Dices'.

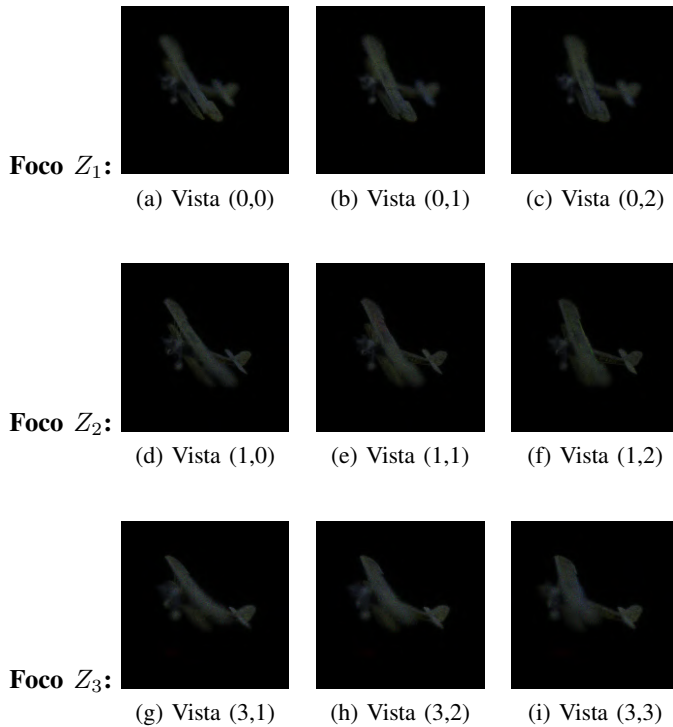


B. Holograma 'Biplane'

A Figura 3 apresenta algumas das vistas geradas para o holograma 'Biplane', que, embora consista num único objeto, ainda permite demonstrar a focagem seletiva em diferentes partes do avião. O Foco Z_1 (vistas 3a a 3c) permitiu focar

na asa mais próxima, enquanto o Foco Z_2 (vistas 3d a 3f) mostra com maior nitidez o corpo e cauda do avião e, por fim, o Foco Z_3 (vistas 3g a 3i) focou na asa mais distante. Além dos diferentes planos de foco a visualização através das 16 vistas demonstra claramente a mudança de ângulo com que se observa o objeto.

Figura 3: Demonstração das vistas geradas para os três planos de foco (Z_1, Z_2, Z_3) para o holograma 'Biplane'.



IV. CONCLUSÃO

Neste trabalho foi alcançado o objetivo principal de reconstrução de hologramas a cores da base de dados B-Com. A etapa essencial foi a criação de 16 de vistas no plano do holograma e a variação da distância de propagação para os três focos escolhidos. A utilização do ASM permitiu simular o efeito de profundidade, focando seletivamente em diferentes elementos da cena. Este efeito é particularmente notório no *dataset 'Dices'* que contem dados em diferentes planos de profundidade, conforme mostrado na Figura 2. Para melhor demonstração dos resultados obtidos pelas diferentes vistas e focos foram gerados vídeos a partir das sequências de vistas. Assim, os resultados demonstraram a eficácia do ASM na focagem seletiva de objetos em diferentes profundidades e a compilação das sequências de vistas em vídeos permitiu a visualização dos hologramas a partir de diferentes posições.

REFERÊNCIAS

- [1] A. Gilles, P. Gioia, R. Cozot e L. Morin, «Computer generated hologram from multiview-plus-depth data considering specular reflections», em *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2016, pp. 1–6.
- [2] A. Silva, *Holografia Digital*, Repositório GitHub, 2025. URL: https://github.com/ana1f35/Holo_CM.
- [3] I. of Research & Technology (IRT) b-com, *Holographic Images*, 2016. URL: <https://hologram-repository.labs.b-com.com/#/holographic-images>.
- [4] FFmpeg Developers, *FFmpeg*, 2025. URL: <https://ffmpeg.org/>.