UNIVERSIDADE FEDERAL DO PAMPA

Anderson dos Santos da Rosa

Restauração de Imagens Baseada em Múltiplos Quadros

Alegrete
2023

Anderson dos Santos da Rosa

Restauração de Imagens Baseada em Múltiplos Quadros

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

Alegrete
2023

Anderson dos Santos da Rosa

RESTAURAÇÃO DE IMAGENS BASEADA EM MÚLTIPLOS QUADROS

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 4 de dezembro de 2023.

Banca examinadora:

_____

Prof. Dr. Marcelo Resende Thielo

Orientador

Unipampa

_____

Profa. Dra. Raquel Mainardi Pillat Basso

Unipampa

_____

Prof. Dr. Bruno Boessio Vizzotto

Assinado eletronicamente por **MARCELO RESENDE THIELO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 04/12/2023, às 19:36, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **BRUNO BOESSIO VIZZOTTO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 05/12/2023, às 20:32, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **RAQUEL MAINARDI PILLAT BASSO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 11/12/2023, às 10:05, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1318059** e o código CRC **11BDCC65**.

RESUMO

A restauração de imagens com defeitos é predominantemente realizada por meio de redes generativas para inpainting. Em muitos casos, essas redes podem apenas reproduzir padrões dos pixels vizinhos ao defeito, resultando na perda de informação. Para imagens impressas em larga escala, como embalagens de produtos, capas de livros e discos, que possuem múltiplos exemplares, a restauração pode se beneficiar da abundância de exemplares para combinar informações e preencher lacunas causadas pela degradação da imagem. Partindo desse princípio, foram investigadas soluções de um domínio semelhante, conhecido como *Multi-Frame Super Resolution* (MFSR), que utiliza informações de várias imagens para realizar a super resolução. O objetivo é adaptar essa abordagem para combinar informações de múltiplas observações de uma imagem distribuída em grande escala, visando restaurar a imagem original. Considerando o pipeline do MFSR, composto por extração de features, alinhamento, fusão e reconstrução da imagem, foi proposta uma técnica de alinhamento de imagens que não requer o uso de redes neurais profundas. Isso se deve à natureza plana do objeto em questão, eliminando a necessidade de ferramentas tão poderosas para o alinhamento. Ainda está pendente a proposta de um método para realizar a fusão das imagens alinhadas e gerar a imagem restaurada.

Palavras-chave: Aprendizado de Máquina. Super Resolução. Visão Computacional.

ABSTRACT

The restoration of images with defects is currently carried out predominantly by generative networks for inpainting. In some cases, the image may end up merely replicating patterns from neighboring pixels around the defect, as there is no way to recover the missing information. In the case of images printed on a large scale, such as product packaging, book covers, and discs, where multiple copies of these packages still exist, the restoration of these images can benefit from this larger quantity of copies to combine the information contained in each copy to fill in the gaps caused by image degradation. Based on this principle, solutions from a similar domain that leverages information from multiple images for super-resolution, known as Multi-Frame Super Resolution (MFSR), have been investigated. The aim is to create or adapt a solution that combines information from multiple observations of a widely distributed printed image to restore the original image. Given the MFSR pipeline composed of feature extraction, alignment, fusion, and image reconstruction, a form of image alignment has been proposed without the need to use deep neural networks, as there is no need for such powerful tools to align images of a flat object. A method for merging the aligned images to generate the restored image is still to be proposed.

Key-words: Machine Learning. Super Resolution. Computer Vision.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Uma imagem digital pode ser representada por uma matriz de duas dimensões, onde cada elemento da matriz é um pixel. A resolução espacial de uma imagem está associada à densidade de pixels em uma área: quanto maior a densidade de pixels da imagem, maior é a sua resolução (PEDRINI; SCHWARTZ, 2007). Geralmente, quando se trata de aplicações de imagem, quase sempre é desejada uma resolução maior, seja para tornar a imagem mais fácil de ser interpretada por um humano ou por algum processo automatizado (MILANFAR, 2010). Além das limitações de resolução associadas à densidade do sensor, as imagens capturadas podem sofrer degradações durante os estágios de aquisição, transmissão e processamento. Essas degradações, conhecidas como ruído, são apenas uma faceta das possíveis imperfeições. Outras limitações de qualidade incluem o *blur* nas lentes devido ao tamanho da abertura do dispositivo, *blur* causado pelo movimento durante o tempo de abertura, difração no diafragma, efeitos de aberração cromática, entre outros desafios que podem impactar a fidelidade da imagem final.

Desenvolver dispositivos para aprimorar a aquisição de imagens é uma tarefa dispendiosa, e em muitos casos, a substituição de equipamentos mais antigos é impraticável, especialmente em contextos como câmeras de segurança e imagens de satélite. Além disso, a melhoria nos dispositivos não tem impacto nas imagens já existentes. Uma alternativa viável para enfrentar esse desafio é aprimorar as imagens por meio de software, como ocorre na tecnologia de Super Resolução (SR). A SR consiste em ferramentas ou conjuntos de processos que buscam reconstruir uma imagem de alta resolução (HR) a partir das informações contidas em uma ou várias imagens de baixa resolução (LR). Essas imagens de baixa resolução podem representar versões degradadas da imagem HR, contornando assim as limitações do hardware de aquisição, com um custo relativamente baixo. Além disso, essa abordagem pode ser aplicada a imagens já existentes (MAISELI; OGADA; GAO, 2015)(MILANFAR, 2010).

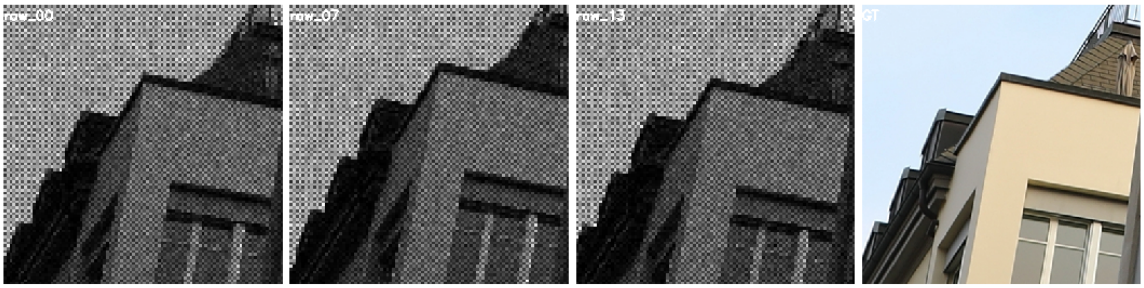
A Super Resolução de imagens encontra aplicação em diversos contextos, como imagens capturadas por sistemas de monitoramento, super resolução de vídeos, sistemas de sensoriamento remoto, imagens médicas e fotografia em modo *burst*. Os processos de Super Resolução (SR) podem ser categorizados em dois grupos: multi-frame super resolution (MFSR) e single-frame super resolution (SFSR). As abordagens SFSR buscam reconstruir uma imagem de alta resolução (HR) a partir de apenas uma imagem de baixa resolução (LR). No entanto, devido à quantidade limitada de informações contidas em uma única imagem LR, a qualidade da reconstrução pode ser comprometida. Dependendo da resolução desejada para a saída HR, podem surgir padrões que não estavam presentes na entrada LR (AREFIN et al., 2020). Já nas abordagens MFSR, a imagem HR é reconstruída a partir de múltiplas imagens LR, combinando as informações contidas em cada uma delas. Quando as entradas LR apresentam variações nos pixels devido ao movimento da câmera ou da cena capturada, diferentes informações são obtidas em cada

imagem LR. Essas variações podem ser exploradas para reconstruir uma imagem HR com maior qualidade (BHAT et al., 2021a).

Nas primeiras soluções de Super Resolução (SR) baseadas em deep learning, a abordagem Single-Frame Super Resolution (SFSR) recebeu mais atenção do que a Multi-Frame Super Resolution (MFSR), impulsionada pela popularidade dos modelos generativos. Alguns autores também associam essa preferência ao SFSR devido à maior facilidade de encontrar aplicações para essa abordagem.

Inicialmente, o MFSR era predominantemente aplicado a imagens de sensoriamento remoto e super resolução de vídeo. No entanto, mais recentemente, houve uma popularização do MFSR aplicado à fotografia em modo burst (burst-SR). Isso se deve, em grande parte, à introdução do desafio Burst Image Super-Resolution Challenge, apresentado em Bhat et al. (2021b), no evento NTIRE 2021, realizado em parceria com a conferência CVPR (Computer Vision and Pattern Recognition). Neste desafio, cada solução recebe um burst de 14 imagens LR no formato RAW, com a adição de ruído. Cada solução é encarregada de realizar a remoção de ruídos, a reconstrução de cores e o aumento de resolução. Um exemplo de um burst de imagens LR no formato RAW, acompanhado da imagem HR correspondente, pode ser visualizado na Figura 1.

Figura 1 – Burst de imagens Raw da solução descrita em Bhat et al. (2021a)



Fonte: do autor

1.1 Proposta

A Figura 2 ilustra o problema que este trabalho busca solucionar. Embora haja várias caixas deste autorama, lançado nos anos 80, espalhadas pelo país, a imagem original utilizada para estampar esta caixa possivelmente não está mais disponível em alta qualidade nos dias de hoje. Além disso, é provável que seu formato digital em alta resolução sequer tenha sido criado.

O objetivo deste trabalho é a reconstrução de uma imagem digital referente a imagens antigas que não possuem uma representação digital ou quando estas representações se perderam durante o passar dos anos. A reconstrução da imagem seria executada através das informações contidas em múltiplas observações desta imagem como fotos ou scans

Figura 2 – Caso de referência.

Fonte: do autor

da imagem que será reconstruída. Ao aproveitar as soluções desenvolvidas para a tarefa de MFSR, que efetivamente combinam informações provenientes de múltiplas imagens de baixa resolução (LR) para reconstruir uma imagem sem ruído e com uma resolução até 4 vezes maior que as imagens de entrada, nossa intenção é criar uma ferramenta baseada nesses métodos para resolver o problema anteriormente descrito. Partimos da hipótese de que ao utilizar métodos MFSR, além da remoção da degradação da imagem devido ao processo de aquisição de imagem, também seja possível abordar a degradação ocorrida ao longo do tempo, como desbotamento, ranhuras, manchas, entre outros.

Entre as diversas aplicações do Multi-Frame Super Resolution (MFSR), aquela que mais se assemelha ao problema de restauração de imagem é a **burst-SR**. No sensoriamento remoto, as imagens geralmente carecem de detalhes significativos, enquanto na super resolução de vídeo, o espaçamento entre cada frame é muito pequeno, resultando em uma diferença mínima no alinhamento das imagens em comparação com o problema que se pretende solucionar. Portanto, o ponto de partida mais adequado seria a investigação das tecnologias relacionadas ao *burst-SR*

Dado que a diferença entre cada frame enriquece o resultado do MFSR, é mais apropriado utilizar imagens provenientes de diferentes exemplares da imagem que precisa ser reconstruída. Dessa forma, os defeitos presentes em cada exemplar serão distintos, possibilitando a combinação das partes mais bem preservadas de cada exemplar para reconstruir a arte original. Em resumo, a abordagem ideal não consiste em usar múltiplas imagens de um único exemplar, uma vez que as regiões danificadas desse exemplar não poderiam ser reconstruídas devido à falta de informação sobre essas áreas específicas. Durante o desenvolvimento do trabalho, o escopo foi limitado à investigação do alinhamento de imagens, que é uma das etapas do processo de Multi-Frame Super Resolution (MFSR). Já que ao obtermos múltiplas imagens degradadas alinhadas em relação a uma mesma referência, será possível combinar efetivamente essas imagens para gerar, enfim, a imagem restaurada.

1.2 Organização do trabalho

Até este ponto, foram apresentadas uma breve contextualização sobre o processo de aquisição de imagens, uma introdução ao tema de Super Resolução (SR) e a exposição do problema que este trabalho busca resolver parcialmente. No Capítulo 2, a tecnologia Multi-Frame Super Resolution (MFSR) é detalhadamente apresentada, revisando diversas abordagens dessa técnica. No Capítulo 3, são listadas as ferramentas e métodos utilizados no desenvolvimento da solução proposta. No Capítulo 4, são avaliados os resultados obtidos por ferramentas de MFSR, descrevendo as diferenças entre os domínios de MFSR e restauração de imagens. O Capítulo 5 apresenta a solução de alinhamento de imagens proposta. Por fim, o Capítulo 6 encerra o trabalho, fornecendo perspectivas para trabalhos futuros.

2 MULTI-FRAME SUPER RESOLUTION

O MFSR foi explorado pela primeira vez por TSAI (1984). Desde então, diversas técnicas foram propostas para executar essa tarefa, sendo que a maioria delas tenta formalizar um modelo que simula o processo de aquisição de imagens. Dado que este processo não é perfeito devido a limitações físicas do hardware usado no processo, é comum observar diversos efeitos indesejados na imagem capturada, como serrilhado, borrões, baixa resolução e ruído (MILANFAR, 2010). As técnicas mais tradicionais de MFSR são modelos de geração de dados que simulam o processo de aquisição das imagens LR, incluindo os processos que geram esses efeitos indesejados. Os modelos de MFSR seguem um padrão semelhante ao descrito a seguir na equação 2.1.

Seja Y_i a i-ésima imagem LR de dimensões $m \times n$ gerada pelo processo de aquisição de imagem representada em forma vetorial, X a imagem HR de dimensões $pm \times pn$ que pretende-se reconstruir também representada em forma vetorial onde p é o fator de redução na resolução da imagem, D_i a decimação que ocorre na i-ésima imagem LR, B_i o *blur* que ocorre na i-ésima imagem LR, W_i a distorção que ocorre na i-ésima imagem LR, e N_i o ruído que incidiu na i-ésima imagem, temos:

$$Y_{i_{[mn\times 1]}} = D_{i_{[mn\times pmpn]}} B_{i_{[pmpn\times pmpn]}} W_{i_{[pmpn\times pmpn]}} X_{[pmpn\times 1]} + N_{i_{[mn\times 1]}} \tag{2.1}$$

Para realizar o processo de Super Resolução (SR), Peleg, Keren e Schweitzer (1987) introduziram uma abordagem em que o problema de SR é abordado de forma reversa. Inicialmente, uma imagem HR aproximada é gerada, passando k vezes pelo processo de aquisição modelado, resultando em um conjunto de k imagens LR. Em seguida, é calculado o erro entre as imagens LR obtidas pelo processo modelado e as imagens LR reais. A geração iterativa de imagens HR busca minimizar esse erro, possibilitando encontrar uma imagem HR estimada que teoricamente se assemelha à imagem HR (cena observada no momento da aquisição das imagens). Essa imagem HR estimada seria aquela que geraria as imagens LR reais quando submetida ao processo de aquisição de imagem modelado.

Além do erro observado entre as imagens LR geradas e as imagens LR reais (termo de fidelidade), assume-se a priori um modelo sobre a imagem (termo de regularização). Esse modelo é utilizado para avaliar as imagens geradas, penalizando certos padrões indesejados na imagem e evitando soluções que não pareçam plausíveis para um espectador humano (MILANFAR, 2010). A matriz resultante da multiplicação das matrizes D_i, B, N_item tamanho $mn \times pmpn$ (mn é número de pixels contidos na imagem LR e $pmpn$ é o número de pixels que formam a imagem HR) e os seus elementos são desconhecidos. Para reconstruir esta matriz é assumido um modelo de movimentação entre as imagens LR um modelo de ruído, e um modelo de blur (MILANFAR, 2010).

A qualidade das imagens HR geradas está diretamente ligada a uma boa escolha destes modelos. Por outro lado, as técnicas de Super Resolução (SR) baseadas em aprendizado podem aprender esses modelos a partir de imagens reais, eliminando a necessidade

de gerar um modelo matemático que descreva o processo de geração de imagem ou o termo de regularização (VALSESIA; MAGLI, 2021).

No prosseguimento deste capítulo, serão apresentados trabalhos relevantes que podem contribuir para o desenvolvimento da solução proposta. Além disso, serão discutidas abordagens utilizadas na realização desta tarefa, abrangendo soluções fundamentadas em *Maximum A Posteriori* (MAP), abordagens híbridas e aquelas fundamentadas em aprendizado profundo.

2.1 MAP

Os métodos de Multi-Frame Super Resolution (MFSR) mais tradicionais são baseados na otimização de uma função que descreve o processo de aquisição de imagem. Uma vantagem desses métodos é a dispensa de datasets de treinamento. Em seu trabalho, Liu et al. (2021) propõem um modelo de geração de imagem tradicional, conforme descrito na equação 2.2. A solução proposta consiste em um termo de regularização e um termo de fidelidade. O termo de fidelidade tem a responsabilidade de minimizar o erro entre as imagens LR e a imagem HR estimada, enquanto o termo de regularização visa tornar a solução mais robusta. Aqui, Y_k representa as imagens LR do conjunto de tamanho K $onde, k = 1, 2...K$, D é a decimação que ocorre em todas as imagens LR Y_k , X é a imagem HR latente, B_k é a matriz de blur da k-ésima imagem LR, M_k é a matriz de movimento da k-ésima imagem, n_k representa o ruído na k-ésima imagem.

$$Y_k = DB_kM_kX + n_k \tag{2.2}$$

O sistema pode ser reescrito como $W_k = DB_kM_k$, sendo que o erro observado entre cada k-ésima imagem LR e a imagem estimada pelo processo de geração de imagem é obtido por $r_k = |W_kX - Y_k|$. Liu et al. (2021) resumem o processo de MFSR na Equação 2.3. Trata-se basicamente de encontrar a imagem latente X que minimiza o resultado da Equação 2.3, onde $\hat{X}$ é a imagem HR estimada. γ é o termo de regularização da imagem estimada. O somatório representa a fidelidade da imagem reconstruída, λ é um termo de balanceamento entre o termo de regularização e o termo de fidelidade.

$$\hat{X} = \arg\min_X \left\{ \sum_{k=1}^{K} {r_k}_p^p + \lambda\gamma(X) \right\} \tag{2.3}$$

O processo de solução usado por (LIU et al., 2021) é baseado na abordagem MAP que aplica o teorema de Bayes descrito na Equação 2.4 que é útil na resolução de problemas de probabilidade condicional. A abordagem MAP busca encontrar a imagem latente X com a maior probabilidade de ser observada dado que foram observadas as imagens LR Y_k, $onde\ k = 1, 2, ...K$. Reescrevendo a Equação 2.4 em termos do problema de MFSR temos a equação 2.5. Assumindo que o objetivo não é encontrar a probabilidade exata de $p(X|Y_1, Y_2, ...Y_K)$ mas sim maximizar o valor encontrado. É possível desconsiderar

o denominador do teorema de Bayes e buscar imagens HR que maximizem o valor do numerador, e assumindo que o modelo de ruído é Gaussiano.

$$p(x|d) = \frac{p(d|x)p(x)}{p(d)} \tag{2.4}$$

$$p(X|Y_1, Y_2, ...Y_K) = \frac{p(Y_1, Y_2, ...Y_K|X)p(X)}{p(Y_1, Y_2, ...Y_K)} \tag{2.5}$$

A probabilidade condicional de observar o conjunto de imagens LR dado uma imagem HR X é dada pela equação 2.6 (MILANFAR, 2010; LIU et al., 2021). A função objetivo da abordagem MAP pode ser extraída ao aplicar a função logarítimica no numerador do teorema de Bayes como descrito na Equação 2.7. Considerando apenas os termos referentes à X obtém-se a Equação 2.8. Finalmente, gerando a função objetivo clássica da abordagem MAP para MFSR descrita na Equação 2.9, onde $p(X)$ é o modelo de imagem assumido a priori chamado por Liu et al. (2021) de termo de regularização, e o somatório pode ser relacionado ao termo de fidelidade como descrito na Equação 2.3.

$$p(Y_1, Y_2, ...Y_K|X) = \left(\frac{\beta}{2\pi}\right)^{\frac{kM}{2}} \exp\left\{-\frac{\beta}{2}\sum_{k=1}^{K} {r_{k2}}^2\right\} \tag{2.6}$$

$$\log(p(Y_1, Y_2, ...Y_K|X)p(X)) = \log(p(Y_1, Y_2, ...Y_K|X)) + \log(p(X)) \tag{2.7}$$

$$\log(p(Y_1, Y_2, ...Y_K|X)) = \log(e^{-\frac{\beta}{2}\sum_{k=1}^{K} {r_{k2}}^2}) + \log(p(X)) \tag{2.8}$$

$$\hat{X}_{MAP} = \frac{\beta}{2}\sum_{k=1}^{K} {r_{k2}}^2 - \log(p(X)) \tag{2.9}$$

O modelo usado por Liu et al. (2021) é uma variação do BTV (Bilateral Total Variation) que compara a imagem HR com outras versões dela mesma após deslocamentos. O termo proposto tem a adição de um tensor T para identificar bordas e é descrito na equação 2.10, onde S_x^ne S_y^m são operações de deslocamento na imagem X de n pixels na horizontal e m pixels na vertical, P é o tamanho máximo dos deslocamentos da imagem em alguma direção, ζ é um peso variável que está no intervalo $0 < \zeta < 1$, ω_T é um peso definido com base nos valores do tensor T.

$$\gamma(X) = \sum_{n=-P}^{P}\sum_{m=-P}^{P} \omega_T \zeta^{|m|+|n|} X - S_x^n S_y^m X_1 \tag{2.10}$$

Liu et al. (2021) substitui a norma ℓ^2 na Equação 2.9 pela função *half-quadratic* descrita na Equação 2.11, Gerando o termo de fidelidade que calcula o erro descrito na Equação 2.12. Onde I é o número de pixels na k-ésima imagem, r_{k_i} é o erro observado no i-ésimo pixel da k-ésima imagem, a_k é o parâmetro da função *half-quadratic* usado na k-ésima imagem que é definido por $a_k = \frac{max(r_k)}{r_k}$ onde $max(r_k)$ é o erro da imagem LR

com maior erro em relação à imagem estimada e r_k é o erro da k-ésima imagem em relação à imagem estimada, C é uma matriz de confiança com pesos atribuídos a cada imagem inversamente proporcional ao erro observado em cada imagem LR.(LIU et al., 2021)

$$f(x, a) = a\sqrt{a^2 + x^2} - a^2 \tag{2.11}$$

$$\hat{X} = \arg\min_X \left\{ \sum_{k=1}^{K} \sum_{i=1}^{I} C(a_k \sqrt{a_k^2 + r_{ki}} - a_k^2) \right\} \tag{2.12}$$

Substituindo os termos de fidelidade e regularização da Equação 2.3 pelas equações 2.12 e 2.10 Liu et al. (2021) obtém a equação 2.13 que deve ser otimizada para solucionar o problema de MFSR de maneira tradicional.

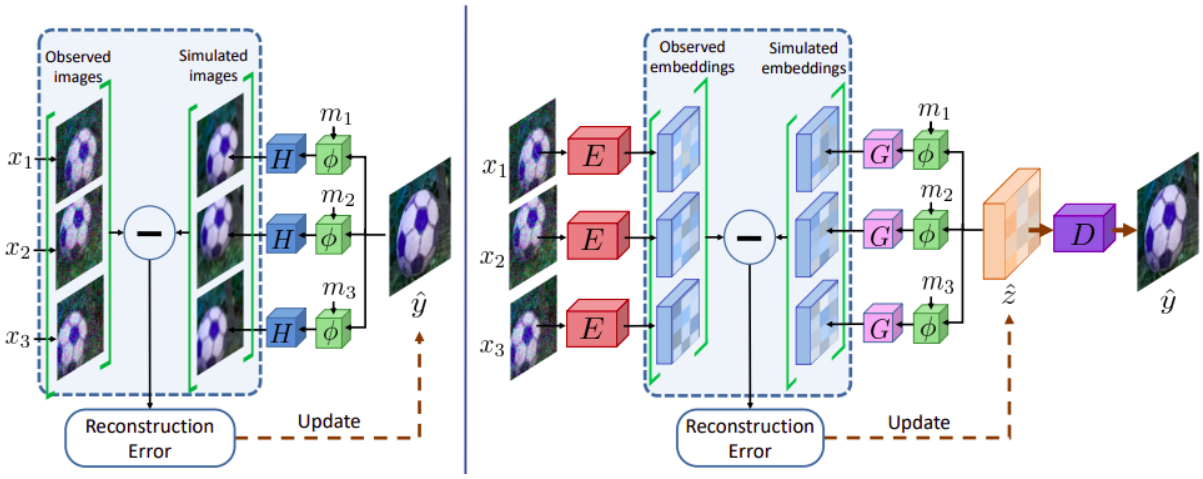
$$\hat{X} = \arg\min_X \left\{ \sum_{k=1}^{K} \sum_{i=1}^{I} C(a_k \sqrt{a_k^2 + r_{ki}} - a_k^2) + \lambda \sum_{n=-P}^{P} \sum_{m=-P}^{P} \omega_T \zeta^{|m|+|n|} X - S_x^n S_y^m X_1 \right\} \tag{2.13}$$

2.2 Híbrido

Além das abordagens mais tradicionais, há trabalhos que seguem um modelo híbrido, mesclando soluções baseadas em aprendizado com os modelos matemáticos das abordagens tradicionais. Um exemplo é a solução descrita por Lecouat, Ponce e Mairal (2021), que adota uma abordagem de resolução inversa do modelo de geração de imagem. Nesse método, todos os termos desconhecidos do modelo de geração de imagem são aprendidos de maneira supervisionada utilizando uma variação das *Convolutional Neural Network* (CNN). Durante o treinamento da CNN, as imagens são geradas sinteticamente aplicando um modelo de geração de imagem k vezes em cada imagem do conjunto de dados *Zurich RAW to RGB* proposto por Ignatov, Gool e Timofte (2020). Esses conjuntos de k imagens LR geradas são emparelhados com as respectivas imagens HR para formar o conjunto de dados de treinamento. Bhat et al. (2021c) adotam um modelo de rede neural semelhante ao utilizado por Lecouat, Ponce e Mairal (2021), que também combina abordagens mais tradicionais com soluções baseadas em aprendizado. A Figura 3 mostra uma comparação entre a solução mais tradicional, baseada em modelos de otimização, e a proposta por Bhat et al. (2021c).

No lado esquerdo da Figura 3, é apresentada a abordagem tradicional, na qual uma imagem $\hat{y}$ é gerada e passa por um modelo de degradação, onde ϕ_{m_i} representa a operação de movimento e rotação observada entre as imagens x_i, e H é a operação de degradação (decimação e desfoque) que ocorre em todas as imagens. Ao passar a imagem $\hat{y}$ pelas operações H e ϕ_{m_i}, são geradas imagens LR simuladas que são comparadas com as imagens LR reais, gerando um erro usado na reestimação da imagem $\hat{y}$ até que a imagem estimada seja aceitável.

Figura 3 – Comparação entre modelos



Fonte: Bhat et al. (2021c)

O lado direito da Figura 3 mostra a solução desenvolvida por Bhat et al. (2021c), na qual o operador H é substituído por uma composição $G = E \circ H \circ D$, onde E é um codificador que mapeia cada imagem x_i para um *embedding space*. O erro é calculado no *embedding space* e usado para reestimar a representação latente $\hat{z}$ da imagem $\hat{y}$, e D é um decodificador que gera a imagem $\hat{y}$ através da representação latente $\hat{z}$.

2.3 Aprendizado profundo

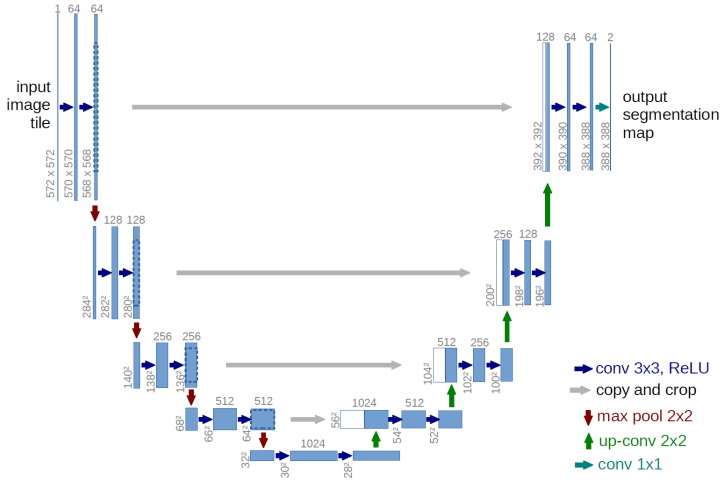
Há inúmeras soluções desenvolvidas baseadas em aprendizado profundo. De acordo com Luo et al. (2022), para resolver o problema de Multi-Frame Super Resolution (MFSR) aplicado à fotografia em modo *burst* (burstSR), as soluções geralmente apresentam as seguintes etapas: extração de *features*, alinhamento de *features*, fusão e reconstrução da imagem HR. As soluções de Super Resolução de Vídeo (videoSR) também seguem um modelo semelhante. A seguir, são descritas algumas abordagens para realizar cada etapa do MFSR baseado em redes neurais profundas.

2.3.1 Extração de Features e Reconstrução da Imagem

A primeira etapa no MFSR é a extração de *features*, assim como em diversos outros processos da visão computacional, como alinhamento de imagens, segmentação de imagem, reconhecimento facial, etc. No contexto do MFSR, a extração de *features* pode ser realizada por meio de uma *Convolutional Neural Network* (CNN), um *autoencoder* ou uma Residual Neural Network (ResNet), entre outras tecnologias. Geralmente, essas tecnologias apresentam uma arquitetura em forma de U, como na Figura 4. Cada quadrado azul na imagem é um *feature map* com múltiplos canais, e o número de canais está

escrito na parte superior do quadrado. A largura e altura do tensor estão descritas ao lado esquerdo do quadrado.

Figura 4 – Arquitetura de uma U-net para segmentação de imagens



Fonte: Ronneberger, P.Fischer e Brox (2015)

A figura apresenta uma extração de *features* comum, onde camadas de convolução são seguidas por camadas de *pooling*. No gargalo da rede, onde estão os *feature maps* de menor dimensão, é onde as *features* foram efetivamente extraídas e são processadas de acordo com o propósito da rede, que, no caso do exemplo, é a segmentação de imagens. No contexto do MFSR, as *features* ainda seriam alinhadas e combinadas para gerar o *feature map* da imagem de saída estimada. Após o gargalo da rede, operações de aumento de resolução dos *feature maps*, como convolução transposta, são realizadas para remontar uma imagem do tamanho original, ou no caso do MFSR, uma imagem maior que as originais.

2.3.2 Alinhamento

Uma forma de alinhamento de *features* adotado no MFSR é a *Deformable Convolution Network* (DCN), ou, em tradução para o português, rede de convolução deformável. Em uma convolução que usa um *kernel* $3x3$ para varrer o *feature map* x e gerar um novo *feature map* de saída y, normalmente tem-se um *kernel* com os seguintes deslocamentos $R = \{(-1,-1),(-1,0),...,(1,1)\}$ em relação à posição do centro do *kernel* p_0. Assim, cada posição em y é dada pela Equação 2.14, onde w são os pesos para cada posição do *kernel* (DAI et al., 2017).

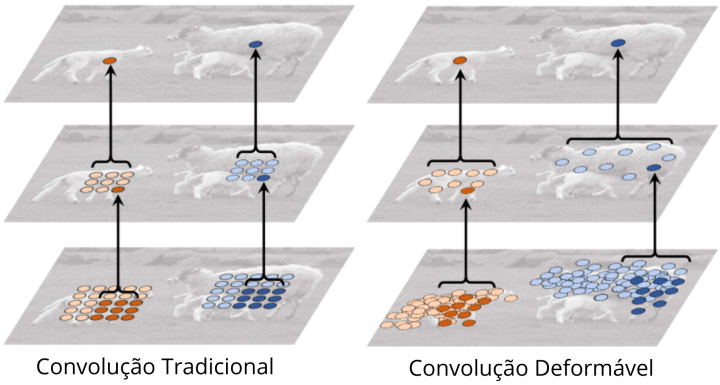
$$y(p_0) = \sum_{p_n \in R} w(p_n) x(p_0 + p_n) \tag{2.14}$$

A convolução deformável é uma evolução da convolução tradicional onde os deslocamentos realizados pelo *kernel* no *feature map* x não são fixados, desta forma os desloca-

mentos do *kernel* $3x3$ devem ser aprendidos pela rede. A definição de cada posição em y é dada conforme a Equação 2.15, na equação Δp_n é o deslocamento aprendido para cada posição de x (DAI et al., 2017). Além disso, a diferença entre a convolução tradicional e convolução deformável é ilustrada na Figura 5.

$$y(p_0) = \sum_{p_n \in R} w(p_n) x(p_0 + p_n + \Delta p_n) \tag{2.15}$$

Figura 5 – Comparação entre convolução tradicional e convolução deformável

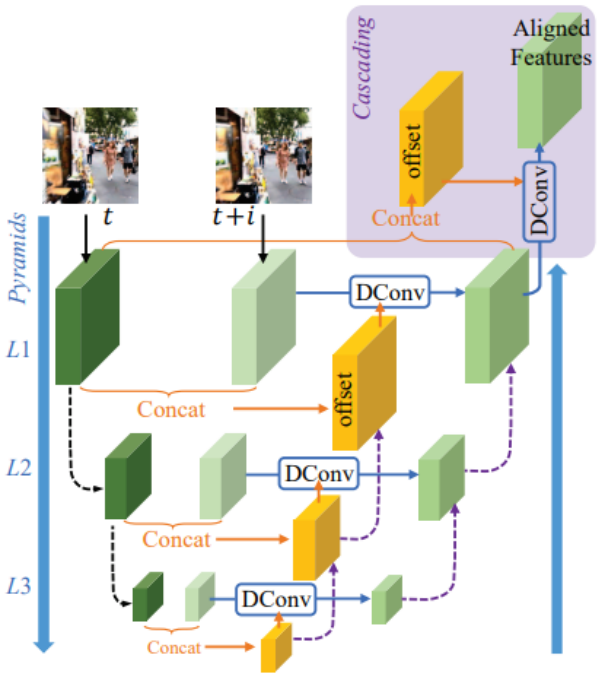


Fonte: adaptado de Dai et al. (2017)

Na solução descrita por Wang et al. (2019), baseada em *Pyramid, Cascading and Deformable convolution* (PCD), que apresenta uma estrutura em forma de pirâmide, onde em cada nível seguinte da pirâmide, as dimensões dos *feature maps* têm a metade da dimensão do nível anterior. No nível mais alto da pirâmide, onde as dimensões são menores, é mais fácil calcular o deslocamento entre as imagens. Esse deslocamento é usado para realizar a convolução deformável, alinhando as representações das imagens no nível da pirâmide. Os resultados obtidos são usados no nível anterior de maior resolução para calcular o deslocamento entre as representações e realizar o alinhamento. Essa abordagem, onde o resultado do nível com menos detalhes é usado para resolver o nível mais detalhado, é conhecida como *coarse-to-fine*. A estrutura do PCD é ilustrada na Figura 6.

Outra abordagem adotada para o alinhamento entre as imagens durante o MFSR é o *optical flow*. Este método permite quantificar o movimento entre duas imagens com base nas variações de brilho. O *optical flow* entre duas imagens, ambas com dimensões $W \times H$, resulta em um grid composto por $W \times H$ *optical flows*. Esses *optical flows* são vetores (u, v), representando a velocidade e a direção do movimento do objeto entre as imagens, conforme a Equação 2.16, onde δx é o deslocamento do pixel no eixo x, δy é o deslocamento no eixo y, e δt é o tempo decorrido entre a aquisição das imagens (NAYAR, 2022).

Figura 6 – PCD coarse to fine



Fonte: Wang et al. (2019)

$$(u, v) = (\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t}) \tag{2.16}$$

A estimação de *optical flow* pressupõe que o brilho de um ponto na imagem de referência permanece constante na segunda imagem, como expresso na equação 2.17. Isso pode ser observado em imagens adquiridas rapidamente, como em frames de um vídeo ou em um burst de imagens. Nestes casos, δt é muito pequeno. Outra suposição é que, além de δt, δx e δy também são muito pequenos. Considerando o brilho de um pixel na posição (x, y) em um momento t definido pela função $I(x, y, t)$, a aproximação de primeira ordem de Taylor, dada por $f(x + \delta x) = f(x) + \frac{df}{dx}\delta x$, leva à equação 2.18 (NAYAR, 2022) (HORN; SCHUNCK, 1981).

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \tag{2.17}$$

$$I(x + \delta x, y + \delta y, t + \delta t) \approx I(x, y, t) + \frac{dI}{dx}\delta x + \frac{dI}{dy}\delta y + \frac{dI}{dt}\delta t \tag{2.18}$$

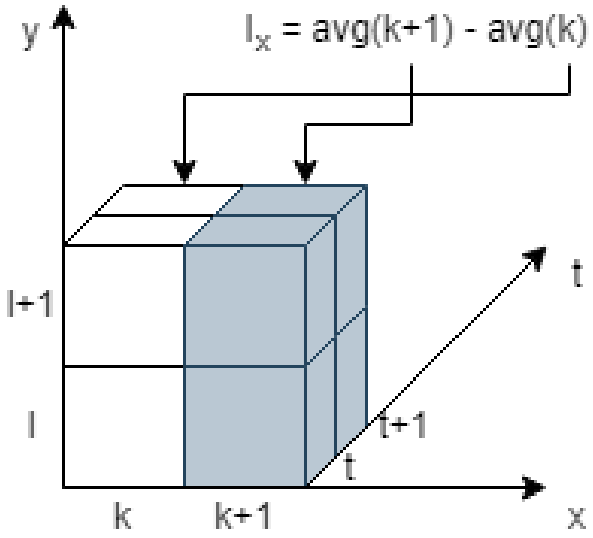
Ao subtrair a equação 2.17 da equação 2.18, dividindo por δt obtem-se a equação 2.19 que é equação de restrição do *optical flow*. Podemos substituir os termos descritos na equação 2.16 e renomeando as derivadas parciais de I por I_x, I_y, I_t para facilitar a escrita obtem-se a equação 2.20. Uma das formas de estimar os valores das derivadas parciais I_x, I_y, I_t em um determinado pixel da imagem é calculando usando um ponto no centro de um cubo 2x2x2 composto por pixels adjacentes no espaço e tempo ilustrado na figura 7,

onde a diferença da média dos *pixels* em cada face do cubo em relação ao eixo analisado resulta na estimativa da derivada parcial conforme as equações 2.21, 2.22 e 2.23 (NAYAR, 2022) (HORN; SCHUNCK, 1981).

$$\frac{dI}{dx}\frac{\delta x}{\delta t} + \frac{dI}{dy}\frac{\delta y}{\delta t} + \frac{dI}{dt} = 0 \tag{2.19}$$

$$I_x u + I_y v + I_t = 0 \tag{2.20}$$

Figura 7 – estimação da derivada parcial em relação a x



Fonte: do autor

$$\begin{aligned} I_x(k,l,t) = &\frac{1}{4}[I(k+1,l,t) + I(k+1,l,t+1) + I(k+1,l+1,t) + I(k+1,l+1,t+1)] \\ &- \frac{1}{4}[I(k,l,t) + I(k,l,t+1) + I(k,l+1,t) + I(k,l+1,t+1)] \end{aligned} \tag{2.21}$$

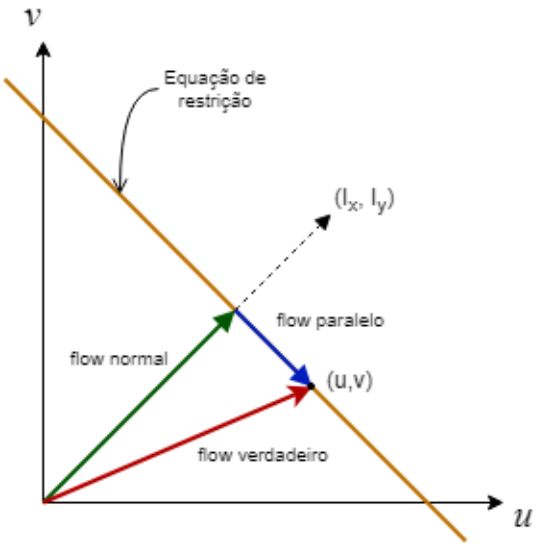
$$\begin{aligned} I_y(k,l,t) = &\frac{1}{4}[I(k,l+1,t) + I(k+1,l+1,t) + I(k,l+1,t+1) + I(k+1,l+1,t+1)] \\ &- \frac{1}{4}[I(k,l,t) + I(k+1,l,t) + I(k,l,t+1) + I(k+1,l,t+1)] \end{aligned} \tag{2.22}$$

$$\begin{aligned} I_t(k,l,t) = &\frac{1}{4}[I(k,l,t=1) + I(k+1,l,t+1) + I(k,l+1,t+1) + I(k+1,l+1,t+1)] \\ &- \frac{1}{4}[I(k,l,t) + I(k+1,l,t) + I(k,l+1,t) + I(k+1,l+1,t)] \end{aligned} \tag{2.23}$$

Após a estimativa dos valores das derivadas parciais, os valores de (u, v) representam um ponto dentro da reta formada pela equação de restrição do *optical flow*, como ilustrado na figura 8. A única informação conhecida é a reta formada pela equação de restrição. O *flow* normal é dado pela equação 2.24. Para encontrar o (u, v), que pode estar em qualquer ponto da reta de restrição, é necessário estimar o *flow* paralelo à reta estabelecendo outras restrições, assim, o *flow* verdadeiro pode ser obtido ao somar os *flows* normal e paralelo.

$$flow\ normal = \frac{|I_t|}{(I_x^2 + I_y^2)}(I_x, I_y) \tag{2.24}$$

Figura 8 – Equação de restrição do *optical flow*



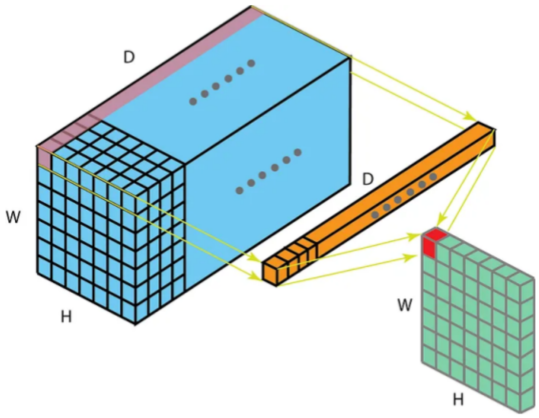
Fonte: do autor

Para aplicar o *optical flow* em imagens com deslocamento significativo, é comum utilizar uma abordagem conhecida como *coarse to fine*. Nessa abordagem, tanto a imagem de referência quanto a imagem que será alinhada passam por vários níveis de uma pirâmide. Em cada nível, a resolução das imagens é reduzida pela metade, resultando em uma diminuição da distância entre os pixels. No último nível da pirâmide, o *optical flow* é estimado. Em seguida, a imagem é transformada de acordo com o *optical flow* calculado. Após essa transformação, a resolução da imagem é novamente ajustada usando algum método de interpolação para igualar o tamanho da imagem no nível anterior. A imagem resultante é então usada no nível anterior para auxiliar no alinhamento daquele nível específico. Essa abordagem *coarse to fine* permite lidar melhor com deslocamentos maiores, pois a estimativa inicial é realizada em imagens de baixa resolução, onde os deslocamentos podem ser mais facilmente identificados. Em seguida, esses resultados são refinados progressivamente em níveis de resolução mais altos, proporcionando uma abordagem mais robusta para lidar com variações de movimento.

2.3.3 Fusão

Uma das técnicas empregadas para realizar a fusão de imagens durante o MFSR é a convolução 1x1. Nas etapas iniciais do MFSR, vários *feature maps* são extraídos a partir de múltiplas imagens de entrada e combinados de diversas maneiras para formar uma representação abrangente dessas imagens, com dimensões HxWxD. A convolução 1x1 utiliza *kernels* de tamanho 1x1xD, sendo que o resultado de cada convolução é uma representação com dimensões WxHx1, conforme ilustrado na Figura 9. Ao realizar N convoluções 1x1xD, obtém-se, ao final, uma representação HxWxN. Se N for menor que D, as informações contidas nos D canais originais foram comprimidas em N canais. Essa abordagem de convolução 1x1 é eficaz para reduzir a dimensionalidade da representação sem perder informações essenciais, contribuindo assim para uma fusão eficiente das características extraídas das diferentes imagens durante o processo de MFSR.

Figura 9 – Convolução 1x1



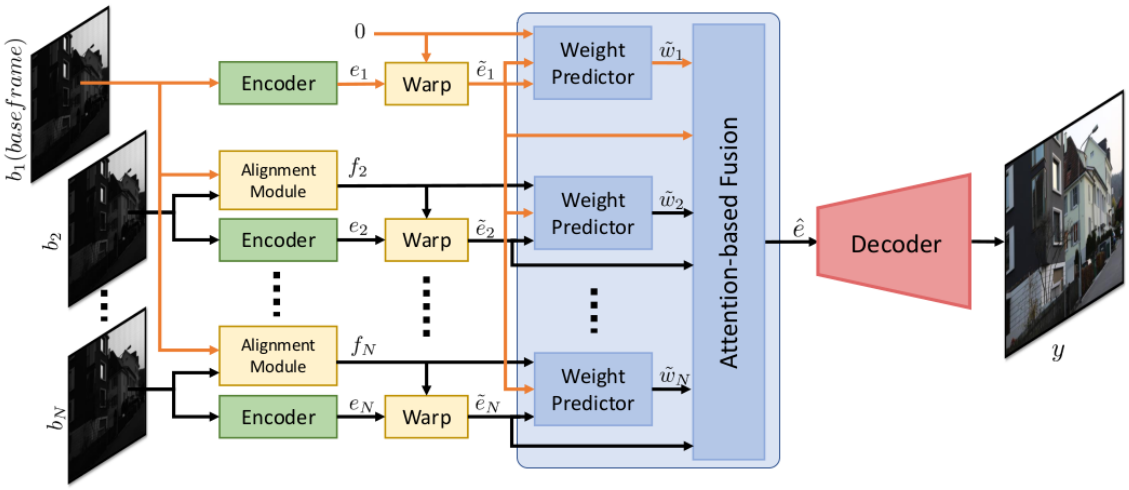
Fonte: Bai (2019)

Outro método utilizado para a fusão de *features* é baseado em mecanismos de atenção, como o empregado por (BHAT et al., 2021a) na rede *Deep Burst*, conforme ilustrado na Figura 10. No processo de fusão baseado em atenção, inicialmente, os pesos de cada um dos *feature maps* são definidos por meio de um preditor de pesos treinado para essa tarefa. O preditor de pesos recebe os *feature maps* alinhados, projetados em uma representação com metade da altura e largura originais, visando um processamento mais rápido. Além disso, o *optical flow* calculado em relação à imagem de referência é considerado no processo. Essa abordagem atencional permite que a rede atribua diferentes níveis de importância aos *features* extraídos, focalizando mais atentamente nas regiões relevantes e adaptando-se dinamicamente às características presentes nas diferentes imagens de entrada.

Dessa forma, o preditor pode atribuir pesos pequenos para regiões desalinhadas, pesos uniformes para áreas com poucos detalhes, ou pesos que promovem uma fusão

sensível às bordas, evitando suavização excessiva nas bordas. Os pesos calculados pelo preditor possuem o formato $\frac{W}{2} \times \frac{H}{2} \times D$, onde H, W, D representam a altura, largura e profundidade do *feature map*, respectivamente. O *feature map* fundido é obtido pela soma ponderada dos *feature maps*, de acordo com os pesos definidos pelo preditor. Essa abordagem permite que a fusão seja adaptativa, destacando características relevantes e preservando detalhes importantes durante o processo de MFSR.

Figura 10 – Rede Deep Burst



Fonte: Bhat et al. (2021a)

3 MATERIAIS E MÉTODOS

Neste capítulo, são apresentados os processos e ferramentas propostos ou descritos por terceiros que foram utilizados no desenvolvimento deste trabalho. Essas ferramentas foram empregadas tanto para a realização de testes quanto para a comparação de resultados.

3.1 Geração de Dataset

Devido à necessidade de realizar testes e treinamento de ferramentas, torna-se necessário o uso de um conjunto de dados (*dataset*). Como não foi encontrado um *dataset* de múltiplas imagens degradadas de uma mesma ilustração, faz-se necessária a composição de um novo *dataset*, seja por meio da coleta de imagens reais ou pela geração sintética de imagens que simulem os defeitos encontrados em imagens reais de maneira convincente. O processo de sintetização de imagens com defeitos adotado neste trabalho foi inspirado no método descrito por Wan et al. (2020). A Figura 11 ilustra o processo de sintetização de imagens degradadas.

Figura 11 – Sintetização de imagem degradada

Fonte: do autor

Se a imagem for menor que $W \times H$, ela é aumentada mantendo a proporção (*aspect ratio*) e, em seguida, recortada para ter o tamanho $W \times H$. Após a redimensionalização ou recorte da imagem, ela é misturada com uma textura de papel, dependendo dos valores de uma máscara binária. Nessa máscara, onde o valor é 255, o resultado recebe o mesmo valor da textura de papel; onde o valor da máscara é 0, ele recebe o valor contido na imagem

recortada. Isso simula degradação estrutural na imagem, como arranhões e buracos na impressão. A imagem recortada também é mesclada com a textura de papel usando uma soma ponderada com um peso aleatório para a imagem recortada, o que resulta em um efeito de desbotamento no resultado. Por fim, existe uma probabilidade de ocorrer um ruído na imagem final, que pode ser do tipo *Speckle* ou *Gaussian*. Há também a probabilidade de simular o efeito de *blur* ou de baixa resolução. Para a geração de múltiplas amostras da mesma imagem, podem ocorrer deslocamentos em uma direção aleatória e distorção na perspectiva da imagem.

3.2 Detecção de Defeitos estruturais

Para realizar a detecção de defeitos estruturais, foi utilizada a rede desenvolvida por Wan et al. (2020). A arquitetura da rede é em forma de U, e ela recebe como entrada uma imagem com resolução de 256x256, gerando uma máscara binária com a mesma resolução como saída. Nessa máscara, os valores 255 representam áreas da imagem com defeitos, enquanto os valores 0 representam áreas sem defeitos. A Figura 12 ilustra a entrada e saída da rede de detecção de defeitos.

Figura 12 – Detecção de defeitos estruturais

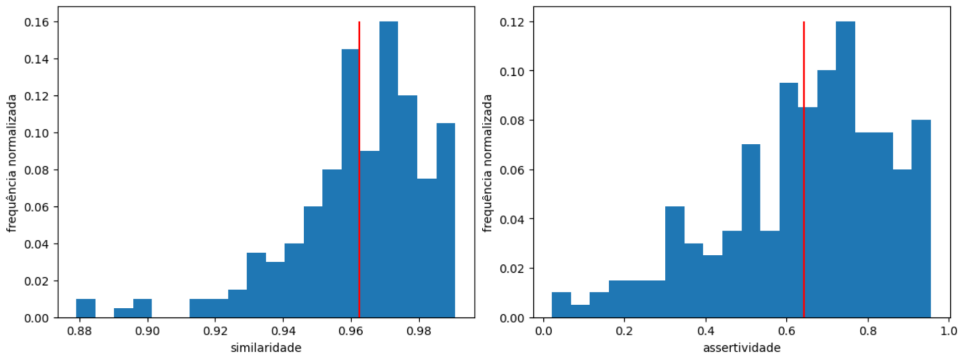
Fonte: do autor

A fim de avaliar os resultados da rede de detecção de defeitos, foi gerado um *dataset* de 200 imagens, conforme descrito na Seção 3.1. Os resultados obtidos pela rede foram comparados com as máscaras usadas para gerar o *dataset*. Como ambas são máscaras binárias, foi realizada apenas uma subtração elemento a elemento. O número de zeros contidos no resultado da subtração foi contado e dividido pelo número total de pixels para obter a proporção de acertos e medir a similaridade entre as máscaras. Os resultados obtidos podem ser visualizados no gráfico à esquerda na Figura 13. A média obtida foi

de 0,9626, e o desvio padrão foi de 0,0203. A linha vermelha na figura representa a média obtida nos resultados.

A primeira impressão sugere que a rede apresenta resultados impressionantes. No entanto, o valor elevado de similaridade foi observado devido à predominância de áreas sem defeito nas imagens. Dessa forma, mesmo uma máscara composta apenas por valores 0 apresentaria uma alta similaridade com a máscara esperada. Assim, outra forma de avaliar a ferramenta foi definida. Desta vez, consideraram-se apenas as regiões da imagem realmente com defeito na máscara esperada, possibilitando quantificar a assertividade da rede nas regiões com defeito. O gráfico à direita na Figura 13 mostra os resultados obtidos neste teste. Neste caso, a média obtida foi de 0,6433, e o desvio padrão foi de 0,2052.

Figura 13 – Resultado dos testes da detecção de defeitos



Fonte: do autor

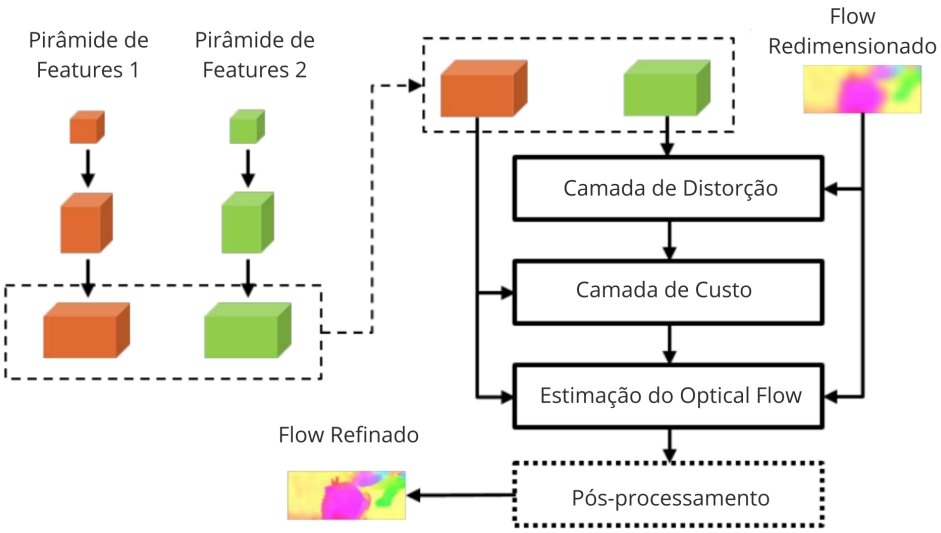
Embora o resultado da rede no teste de assertividade não seja tão impressionante quanto a primeira análise indicava, é importante destacar que a rede desenvolvida por Wan et al. (2020) separa o processo de detecção de defeitos do processo de restauração das imagens. Isso permite que a saída da detecção possa ser usada posteriormente para realizar outras etapas do processo de restauração de imagens com defeitos baseado em múltiplos frames, como alinhamento e fusão de imagens sensível a defeitos estruturais.

3.3 Estimação do optical flow com PWC-net

A rede PWC-net, conforme descrita por Sun et al. (2018), estima o *optical flow* entre duas imagens I_1 e I_2 usando uma rede neural baseada na abordagem clássica de pirâmide de imagens. No entanto, a estimação ocorre entre os *feature maps* aprendidos de cada imagem. Adicionalmente, existe uma camada que estima o custo para alinhar os pixels das imagens. Esse custo alimenta uma CNN que calcula o *optical flow*, passando por alguns pós-processamentos que levam em consideração informações contextuais para refinar a saída. A Figura 14 ilustra a PWC-Net.

Para uma pirâmide de extração de *features* que contém L níveis de representações de uma imagem I_t, onde c_t^0 é o nível zero da pirâmide, representando a própria imagem

Figura 14 – PWC-Net



Fonte: adaptado de Sun et al. (2018)

I_t. Para gerar a representação das *features* no nível c_t^l, a representação no nível c_t^{l-1} passa por filtros de convolução que reduzem a resolução pela metade e aumentam o número de canais. Para distorcer as representações da imagem I_2 no l-ésimo nível c_2^l e gerar a representação distorcida c_w^l, é utilizado o *optical flow* w^{l+1} calculado no nível $l+1$, redimensionado usando interpolação bilinear representada pela função $up_2(w^{l+1})$, conforme a equação 3.1. Nessa equação, x representa o índice do pixel e, no último nível $(L-1)$, $up_2(w^{l+1})$ é definido como 0.

$$c_w^l(x) = c_2^l(x + c_{up_2}(w^{l+1})(x)) \tag{3.1}$$

A representação distorcida c_w^l é utilizada no cálculo do custo cv^l, que representa o custo associado a um pixel na representação de I_1 no l-ésimo nível, denotada por c_1^l, e o pixel correspondente na representação distorcida c_w^l. Para calcular o *optical flow* w^l no l-ésimo nível, cv^l e $up_2(w^{l+1})$ são utilizados como entrada para uma CNN que estima w^l. Em cada nível da pirâmide, são utilizados parâmetros distintos para a CNN.

Foi empregada uma implementação não oficial da *PWC-Net* usando PyTorch, desenvolvida por Niklaus (2018), que também é utilizada nos seguintes trabalhos Luo et al. (2021), Bhat et al. (2021a), Luo et al. (2022), sendo esta implementação considerada o estado da arte no contexto de *burst super resolution*.

3.4 Detecção de Features com ORB

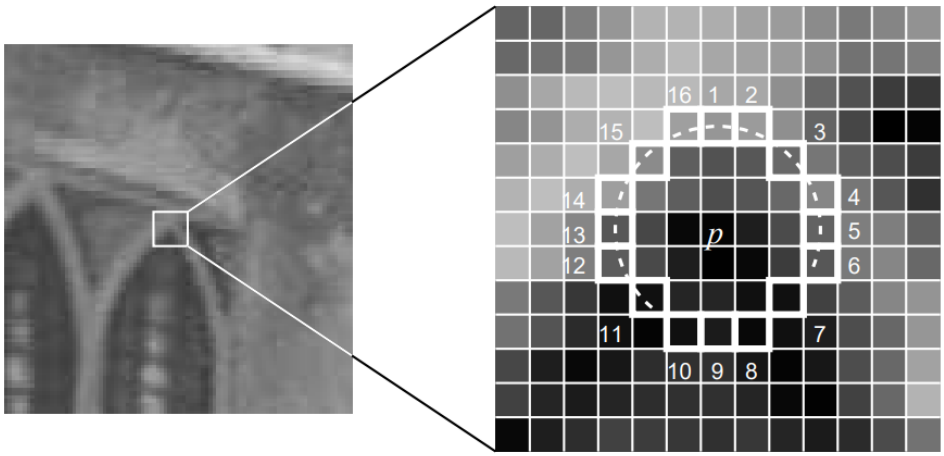
O método de detecção de *features* ORB (Oriented FAST and Rotated BRIEF) foi proposto por Rublee et al. (2011). Este método fundamenta-se na detecção de *features* através de uma variação do método FAST (Features from Accelerated Segment Test),

conforme descrito em Rosten e Drummond (2006) e revisado em Rosten, Porter e Drummond (2010). Além disso, o ORB faz uso dos descritores *features* BRIEF (Binary Robust Independent Elementary Features) modificados, conforme detalhado em Calonder et al. (2010).

O método FAST identifica cantos em imagens utilizando um círculo de 16 pixels ao redor de cada pixel p em uma imagem em escala de cinza. Para que p seja considerado um canto, deve haver pelo menos n pixels mais claros que a intensidade de p somado a um *threshold* t $(I_p + t)$, ou pelo menos n pixels mais escuros que $I_p - t$. Na Figura 15, n é definido como 12. Inicialmente, apenas os pixels 1, 5, 9 e 13 ao redor de p são testados. Para que p seja considerado um canto, pelo menos 3 desses pixels devem ser mais claros que $I_p + t$ ou mais escuros que $I_p - t$.

O ORB utiliza o método oFAST baseado no FAST com $n = 9$ para detectar cantos nas imagens que podem ser usados como *keypoints*, que são os pontos de interesse de um imagem, o FAST não implementa uma detecção de *features* capaz de identificar features independente de escala e orientação. Para contornar esta limitação Rublee et al. (2011) identificam múltiplas escalas de features usando uma piramide de escala da imagem e encontrando as *features* em cada nível da pirâmide. A detecção da orientação das features é realizada usando momentos da imagem definidos na equação 3.2 onde $I(x, y)$ representa a intensidade dada pelo valor de cinza no pixel (x, y). A orientação da *feature* é dada pelo ângulo do vetor que vai centro da feature até o centróide definido pela equação 3.3. No método oFAST são considerados apenas os pixels dentro de um *patch* circular para cálculo dos momentos m_{10}, m_{01}, m_{00} .

Figura 15 – FAST detecção de *keypoints*



Fonte: Rosten e Drummond (2006)

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{3.2}$$

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{3.3}$$

As *features* encontradas pelo oFAST são descritas por meio de descritores rBRIEF, que são baseados no BRIEF com modificações para lidar com mudanças de orientação nas *features*. Os descritores BRIEF são strings de 256 bits, em que cada bit representa o resultado de um teste t realizado entre duas localizações de um *patch*, conforme a equação 3.4, em que $p(x)$ é a intensidade na localização x. As localizações são definidas pelo método utilizado.

Para incorporar informações sobre a orientação da *feature*, os autores do rBRIEF definem uma matriz de rotação para o ângulo θ da orientação da *feature* e aplicam rotação a todas as localizações onde os testes serão realizados, conforme a matriz de rotação. Os valores são discretizados em incrementos de 12 graus. No método rBRIEF, cada teste corresponde a um par de janelas 5x5 dentro de um *patch* de tamanho 31x31.

A seleção dos 256 testes que compõem o rBRIEF seguiu o seguinte processo: os autores executaram todos os testes possíveis em um *dataset* de 300.000 *keypoints*. Os testes foram, então, ordenados em um vetor T, composto por todos os testes, de acordo com a sua distância em relação à média. Os testes foram selecionados em ordem e descartados caso a sua correlação com os testes já selecionados fosse maior que um *threshold*, até que houvessem 256 testes selecionados para compor o descritor do rBRIEF.

$$t(x_1, x_2) = \begin{cases} 1, & se\ p(x_1) < p(x_2) \\ 0, & se\ p(x_1) \geq p(x_2) \end{cases} \tag{3.4}$$

3.5 Métricas de avaliação

Para avaliar os resultados obtidos em diversas tarefas de visão computacional de forma objetiva, é necessário definir métricas para comparar os resultados obtidos por cada solução. No caso do MFSR, as métricas mais comumente utilizadas para avaliar a similaridade entre a saída esperada e a saída obtida são: LPIPS, PSNR e SSIM.

O PSNR (peak signal-to-noise ratio) é utilizado para comparar duas imagens f e g de um mesmo tamanho. A fórmula para uma imagem em escala de cinza é dada pela equação 3.5, onde MSE é o erro médio ao quadrado entre as imagens, conforme definido na equação 3.6. Quanto menor o erro médio, maior é o valor do PSNR (HORé; ZIOU, 2010).

$$PSNR(f, g) = 10log_{10}(\frac{255^2}{MSE(f, g)}) \tag{3.5}$$

$$MSE(f, g) = \frac{1}{wh} \sum_{i=1}^{w} \sum_{j=1}^{h} (f_{ij} - g_{ij})^2 \tag{3.6}$$

O SSIM (Structural Similarity Index Measure) é uma métrica composta por três termos de comparação: iluminação, contraste e similaridade estrutural. A equação 3.7 descreve o SSIM, onde $I(f,g)$ é a comparação entre a iluminação média das imagens, $c(f,g)$ é a comparação do contraste entre o desvio padrão das imagens, e $s(f,g)$ é a comparação de similaridade estrutural medida entre a correlação das imagens. Os resultados de todos os termos estão no intervalo [0,1], onde 1 representa que as imagens são idênticas (HORé; ZIOU, 2010).

$$SSIM(f,g) = I(f,g)c(f,g)s(f,g) \tag{3.7}$$

O LPIPS (Learned Perceptual Image Patch Similarity), conforme descrito em Zhang et al. (2018), compara a ativação de uma rede para duas imagens de entrada. Para cada patch x, x_0 extraído das imagens, é calculada a distância entre as *features* extraídas e normalizadas, denotadas por $\hat{y}, \hat{y}_0$, com dimensões $H \times W \times C$. A equação 3.8 define a distância entre dois patches, onde w_l é um peso utilizado para escalar cada canal do *patch*. O resultado médio obtido para cada *layer* passa por uma rede para prever o LPIPS. O valor do LPIPS representa o erro entre a ativação da rede para as duas imagens; portanto, quanto menor o valor, mais semelhantes são os *patches*.

$$d(x,x_0) = \sum_l \frac{1}{H_l W_l} \sum_i^{H_l} \sum_j^{W_l} \sqrt{w_l \odot (\hat{y}^{ijl} - \hat{y}_0^{ijl})^2} \tag{3.8}$$

Outra métrica usada foi o *end-point-error* (EPE), uma medida comumente utilizada em benchmarks de *optical flow*, que mede quantas vezes a distância calculada entre dois *optical flows* foi maior que um determinado limite. Por exemplo, o EPE3 é a porcentagem de vezes que o erro entre os vetores foi maior que 3. As equações 3.9 e 3.10 demonstram o cálculo do EPE3, onde $F(x,y,u)$ e $F(x,y,v)$ são os valores das coordenadas u e v do vetor na posição x, y.

$$E_3(x,y) = \begin{cases} 1, & \text{se } \sqrt{(F(x,y,u) - F_d(x,y,u))^2 + (F(x,y,v) - F_d(x,y,v))^2} \geq 3 \\ 0, & \text{se } \sqrt{(F(x,y,u) - F_d(x,y,u))^2 + (F(x,y,v) - F_d(x,y,v))^2} < 3 \end{cases} \tag{3.9}$$

$$epe3 = \frac{1}{wh} \sum_x^w \sum_y^h E_3(x,y) \tag{3.10}$$

4 RESULTADOS E DISCUSSÃO

O MFSR é frequentemente utilizado para resolver problemas nos quais os dados de entrada, neste caso as imagens de baixa resolução (LR), exibem uma alta uniformidade, pois são capturados pelo mesmo dispositivo em um curto intervalo de tempo, resultando em resoluções idênticas e passando pelo mesmo processo de degradação. No entanto, conforme mencionado anteriormente, na restauração de imagens baseada em múltiplos quadros, as imagens LR provavelmente serão capturadas por dispositivos distintos. Cada exemplar de uma determinada ilustração está em posse de pessoas diferentes, que podem estar distantes umas das outras e utilizam dispositivos de captura variados. Portanto, a solução deve ser capaz de alinhar e realizar a fusão das características extraídas de imagens com diferentes resoluções.

Para que os defeitos presentes nas imagem não sejam interpretados como uma *feature* valida da imagem durante as etapas do processo restauração deve haver uma forma de detectar estes defeitos e atribuir um peso para cada *feature* ou pixel de acordo com a detecção de defeito empregada, para assim tornar a solução sensível a defeitos. No restante deste capítulo são avaliadas algumas soluções de MFSR de é discutido o alinhamento de imagens já que é uma etapa importante do processo de MFSR e proposta uma abordagem para alinhamento das imagens sem a necessidade de usar redes neurais profundas.

4.1 Avaliação de Soluções MFSR

Entre todos os trabalhos revisados, algumas soluções tornaram públicas as implementações de suas abordagens. Essas implementações foram avaliadas para auxiliar na tomada de decisão sobre qual abordagem seguir na resolução do problema. A Tabela 1 resume as aplicações analisadas e as classifica de acordo com o tipo de abordagem adotada para cada etapa do processo de MFSR.

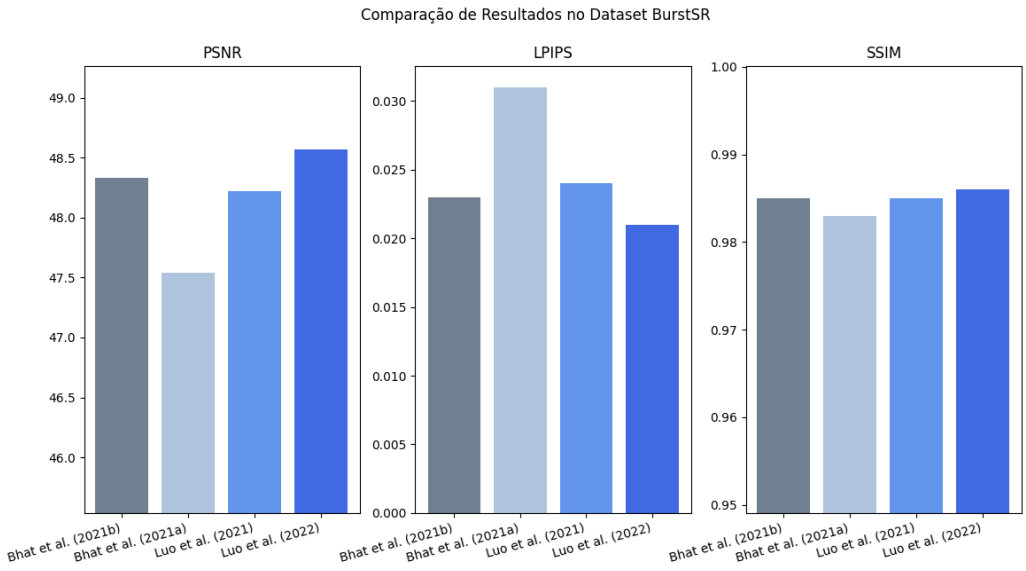
No entanto, é importante observar que as aplicações de Burst SR recebem como entrada imagens no formato RAW, que ainda não passaram pelo processo de ISP (Image Signal Processing). Como descrito na apresentação do problema, as imagens que pretendemos alimentar no MFSR já existem em outro formato. Portanto, será necessário contornar esse problema, seja gerando imagens RAW a partir das imagens existentes ou adaptando essas soluções existentes para que possam receber entradas em outro formato.

Para avaliar objetivamente os resultados obtidos por cada uma das aplicações, foram utilizadas as métricas PSNR, LPIPS e SSIM. Como o dataset BurstSR desenvolvido por Bhat et al. (2021a) é empregado nas quatro aplicações, é possível comparar os resultados obtidos por cada uma delas, conforme ilustrado na Figura 16. Em todas as métricas de desempenho, a solução descrita em (LUO et al., 2022) apresenta os melhores resultados. Destaca-se que, no caso do LPIPS, quanto menor o resultado obtido, melhor é o desempenho.

Tabela 1 – Descrição das Aplicações Avaliadas

Artigo	Extração	Alinhamento	Fusão	Reconstrução
Bhat et al. (2021c)	Encoder	MAP		Decoder
Bhat et al. (2021a)	Encoder	Optical Flow	Attention	Decoder
Luo et al. (2021)	FEPCD		CNLF	LRCN
Luo et al. (2022)	Swin Transformer	Flow + DCN	1x1 Conv	Swin Transformer

Figura 16 – Comparação de resultados



Fonte: do autor

Todas as aplicações analisadas disponibilizam, de alguma forma, funcionalidades interessantes, tais como inversão do *pipeline* de aquisição (conversão de RGB para RAW), algumas funções de pós-processamento para remoção de ruídos e geração de burst de imagens RAW a partir de uma imagem sRGB de entrada. Além disso, elas oferecem as redes pré-treinadas para avaliação. Vale destacar que Bhat et al. (2021a) e Bhat et al. (2021c) apresentam documentação um pouco mais detalhada em relação às outras duas.

Entretanto, as soluções propostas por Luo et al. (2021) e Luo et al. (2022) parecem ter uma execução mais direta. Diversos parâmetros de execução podem ser passados como argumentos no momento da execução, sem a necessidade de criar arquivos de configuração, como ocorre em Bhat et al. (2021a) e Bhat et al. (2021c). Para cada uma das aplicações, foi desenvolvido um notebook no Google Colab, o que facilita o compartilhamento e permite o uso de recursos na nuvem, como acesso a aceleração por GPU.

4.2 Alinhamento de imagens

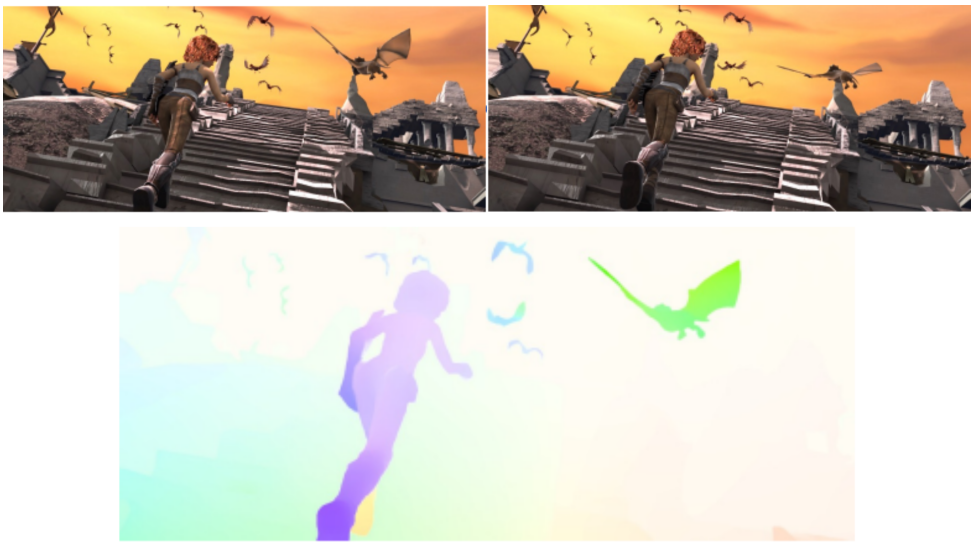
Nesta seção, são discutidas as diferenças encontradas entre os domínios dos problemas de *video-SR* e *burst-SR* em relação à restauração de imagens baseada em múltiplos quadros com foco no alinhamento de imagens. Além disso, é avaliado o impacto que os defeitos na imagem geram na estimação do *optical flow*.

4.2.1 Diferença entre os domínios

No MFSR, as imagens de entrada correspondem a uma cena observada onde podem existir múltiplos objetos de interesse. No caso do *Burst SR* e do *Video SR*, durante o intervalo entre os quadros, cada objeto presente na cena pode estar em movimento em diferentes direções. Enquanto os objetos se movem, o fundo da cena pode parecer imóvel ou se movendo em uma direção devido à movimentação da câmera. Neste caso, para alinhar as imagens, a ferramenta usada deve ser capaz de lidar com movimento que não é linear para toda a imagem, justificando assim o uso do *optical flow*, que é frequentemente encontrado nas ferramentas de MFSR.

A figura 17 ilustra a diferença de movimento entre quadros de múltiplos objetos em uma cena. Cada pixel na representação gráfica do *optical flow* (terceira imagem na figura 17) é definido pela magnitude e orientação do vetor do *optical flow* naquela posição. A orientação define a cor do pixel, e a magnitude define a intensidade da cor. Na figura 17, pode-se observar que a orientação de vetores vizinhos pode mudar drasticamente caso esteja localizado na borda de um objeto.

Figura 17 – Optical flow



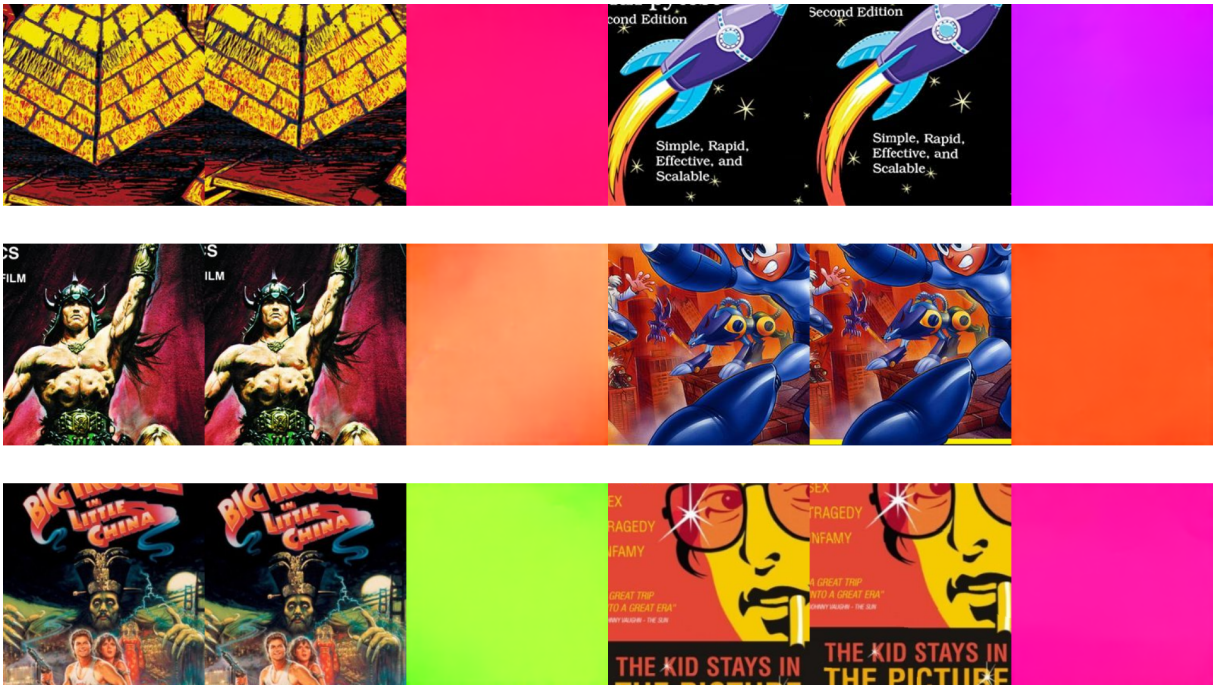
Fonte: adaptado Zhao et al. (2022)

Na restauração de imagens, há apenas um objeto de interesse, que é a ilustração impressa em algum objeto plano ou que se assemelha a um plano. A movimentação entre

as imagens ocorre apenas por diferença na posição da câmera e no ângulo da câmera em relação ao objeto. Assim, é esperado que o deslocamento entre as imagens seja mais linear do que em imagens onde existam múltiplos objetos.

Além disso, para pares de imagens com pouca diferença na rotação, os vetores serão muito parecidos em orientação e magnitude. Para ilustrar essa afirmação, foi estimado o *optical flow* usando a rede PWC-Net. As imagens usadas como entrada foram geradas usando a metodologia descrita na seção 3.1 com rotações aleatórias entre -2,5° e 2,5° nos eixos x e y, conforme a figura 18, onde podem ser vistas as imagens de entrada I_1 e I_2 e a representação gráfica do *optical flow* estimado entre as imagens, denotado aqui por $f(I_1, I_2)$.

Figura 18 – PWC-Net aplicada a deslocamento gerado por pequenas diferenças na perspectiva



Fonte: do autor

4.2.2 Impacto dos defeitos no Optical flow

Para avaliar se há algum impacto nos resultados obtidos pela PWC-Net quando aplicada a imagens com defeitos, foram geradas a partir das imagens I_1 e I_2, imagens com defeito sintético I_1^d e I_2^d. Em seguida, foi estimado o *optical flow* entre as imagens com defeito $f(I_1^d, I_2^d)$ para verificar o impacto dos defeitos sintéticos no *optical flow* estimado. Este processo foi realizado para 270 pares de imagens.

A figura 19 ilustra I_1, I_2, $f(I_1, I_2)$ na parte superior da figura e I_1^d, I_2^d, $f(I_1^d, I_2^d)$ na região inferior da mesma figura. As métricas usadas para mensurar a diferença entre

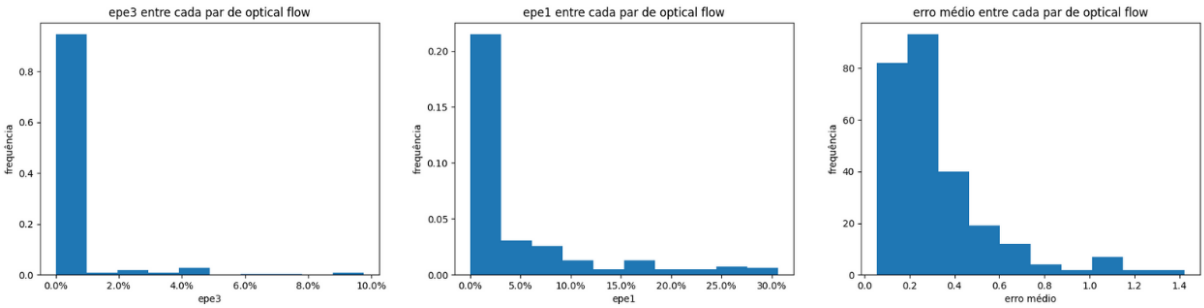
$F = f(I_1, I_2)$ e $F_d = f(I_1^d, I_2^d)$ foram a distância euclidiana média entre cada vetor presente nos *optical flows* estimados F, F_d e o epe3 e epe1.

Figura 19 – Exemplo de imagens geradas para avaliar a PWC-Net

Fonte: do autor

Como ocorre a mesma distorção entre as imagens I_1^d, I_2^d e as imagens I_1, I_2, os *optical flow* obtidos F, F_d deveriam ser iguais se os resultados da PWC-Net fossem invariantes para imagens com defeito. Isso é o que pretende-se verificar com os testes realizados. Os resultados obtidos para as métricas epe1 e epe3 e o erro médio para cada um dos pares F, F_d podem ser vistos na figura 20. Os resultados mostram uma porcentagem de erros baixa. Para o teste epe1, média $= 6,875$, desvio padrão $= 12,242$, para o teste epe3, média $= 1,195$, desvio padrão $= 4,9$, e para os erros médios, média $= 0,405$, desvio padrão $= 0,593$.

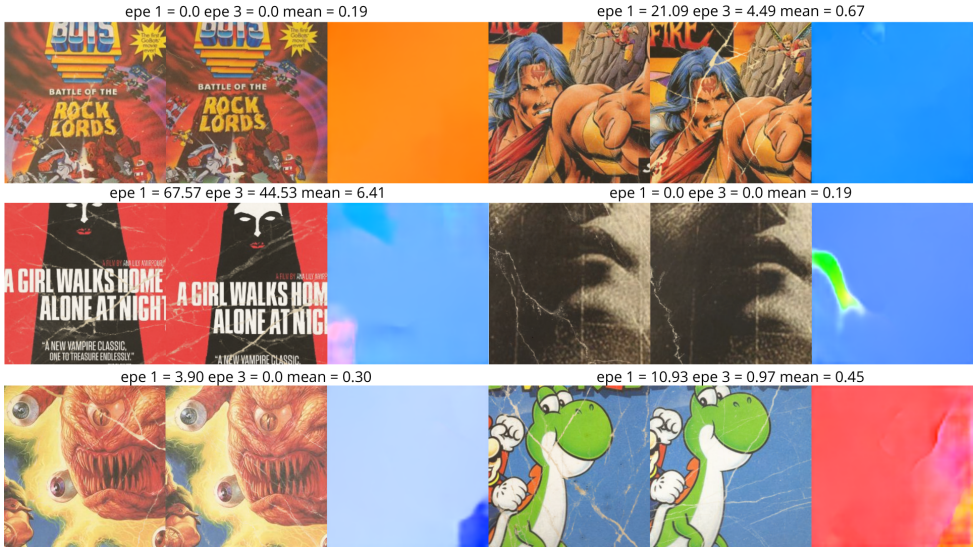
Figura 20 – Resultados obtidos para epe3 e epe1 e erro médio



Fonte: do autor

Embora os os resultados dos teste sejam bons nenhuma das métricas parece ser um bom indicador de quando houve uma mudança brusca na orientação ou magnitude dos vetores devido aos defeitos inseridos nas imagens. A figura 21 mostra alguns exemplos dosF_d e os valores obtidos para cada uma das métricas usadas. Como as imagens se deslocam em uma mesma direção realizar o *warping* usando um *optical flow* que apresenta diferentes direções pode comprometer o resultado do alinhamento entre as imagens.

Figura 21 – Impacto dos Defeitos na PWC-Net



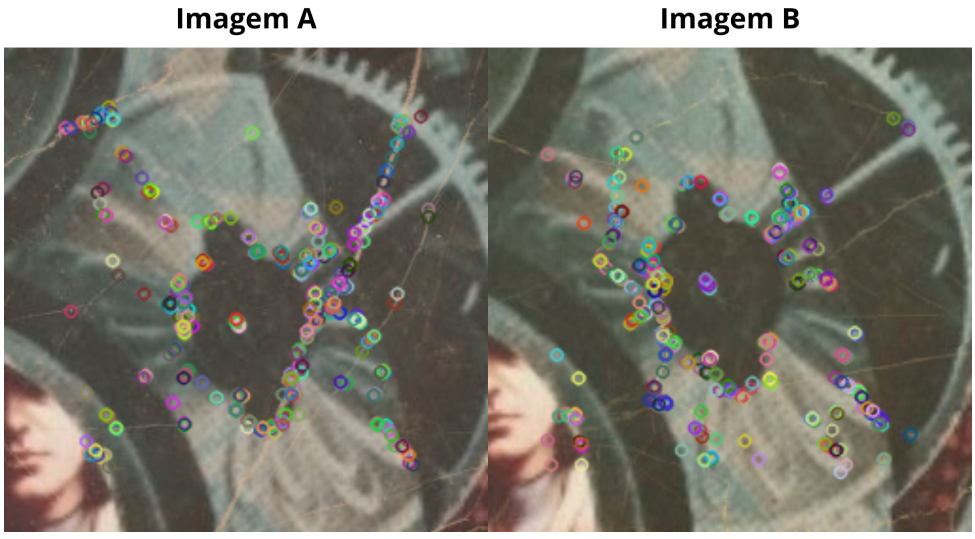
Fonte: do autor

5 ALINHAMENTO DE IMAGENS USANDO HOMOGRAFIA

De acordo com Dubrofsky (2009) e Agarwal, Jawahar e Narayanan (2005), homografias são transformações lineares de projeção ou uma relação entre pontos em planos diferentes. Usando homografias, é possível transferir pontos de uma imagem A para uma outra imagem B como se ambas as imagens estivessem em um mesmo plano, desde que as imagens tenham capturado uma mesma cena de um ponto de vista fixo apenas rotacionando a câmera, ou que tenham capturado um plano mesmo que de pontos de vista diferentes. Para computar o sistema que obtém esta matriz de transformação, são necessários pelo menos quatro pontos equivalentes nas imagens A e B, mas mais pontos devem ser usados para obter uma matriz mais precisa.

Para obter os pontos nas imagens de forma automatizada, poderia ser usado o *optical flow* computado entre as imagens, removendo os vetores que apresentam orientação ou magnitude muito diferentes dos demais (*outliers*). Desta forma, a falta de linearidade do *optical flow* seria contornada. Entretanto, para computar a matriz de transformação, foi usado o detector e descritor de *features* ORB, conforme descrito na seção 3.4, para encontrar pontos equivalentes entre as imagens A e B. Nesse processo, são detectadas as *features* em ambas as imagens, como ilustrado na figura 22. O ORB encontrou, em média, 423,066 *features* por imagem, com desvio padrão igual a 61,069. Cada círculo nas imagens corresponde a um *patch* considerado um ponto de interesse ou *keypoint* pelo detector de *features* do ORB, que detecta cantos. Às vezes, os termos *keypoints* e *features* são usados de forma intercambiável, mas aqui o termo *keypoint* será usado para se referir ao pixel no centro da *feature*.

Figura 22 – Detecção de features usando ORB

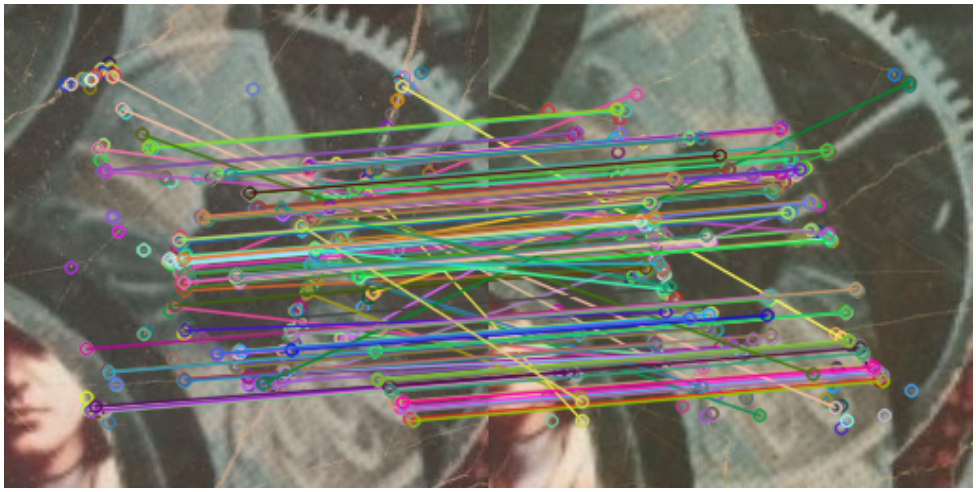


Fonte: do autor

Depois de detectar as *features*, os descritores dessas *features* são comparados usando a distância de Hamming, que é usada para comparar vetores binários. Todos

os descritores de A são comparados com todos os descritores de B. Os descritores com as menores distâncias são considerados como uma *feature* possivelmente equivalente. A equivalência só será confirmada após o descritor de B ser comparado com todos os descritores de A, e não houver nenhum outro descritor de A com uma distância menor. Mesmo usando essa validação cruzada, ainda podem haver descritores que foram relacionados erroneamente e não correspondem à mesma *feature*, como mostrado na figura 23.

Figura 23 – Descritores relacionados

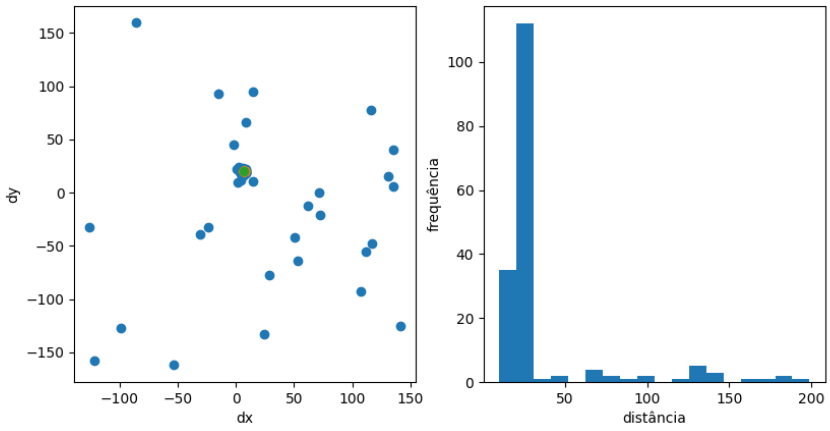


Fonte: do autor

Para identificar e remover os *outliers*, foi calculada a distância euclidiana entre as coordenadas do *keypoint* de A e as coordenadas do seu *keypoint* relacionado em B. Como o deslocamento entre as imagens é linear, essa distância deve permanecer a mesma ou muito próxima para os *inliers*. A figura 24 mostra a distância entre os *keypoints* relacionados e a distribuição da distância entre os *keypoints*. O gráfico da direita mostra que a maioria dos *keypoints* está em uma distância parecida, como esperado. O gráfico à esquerda mostra a seleção dos $k = 50$ vizinhos mais próximos de um centróide. O centróide é o *keypoint* que tem a menor distância média em relação aos seus k vizinhos mais próximos. A figura 25 ilustra os *keypoints* relacionados após a aplicação deste método para computar a homografia entre as imagens, os *keypoints* restantes, representados pelos pontos azuis no gráfico à esquerda, são descartados.

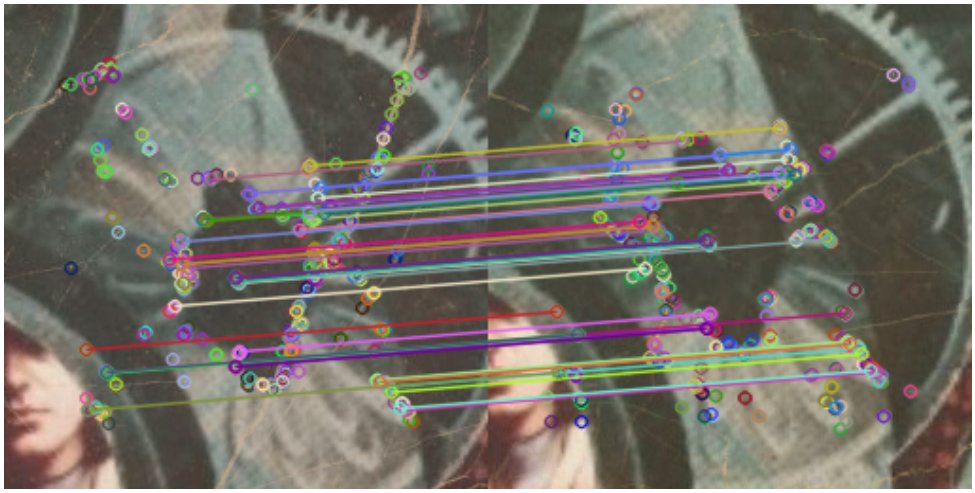
Após computar a homografia que transforma pontos no plano da imagem A para pontos na imagem B, para imagens com dimenões WxH primeiramente os pontos que definem os quatro cantos da imagem A {(0,0), (0, H-1), (W-1,0), (W-1, H-1)} são transfomados para obter o tamanho que a nova imagem deve ter para compreender o *warping* da imagem A já que após a transformação podem haver pontos fora dos limites definidos pelas dimensões original da imagem. Após realizar o *warping* usando a matriz de homografia a imagem B também é reescrita na imagem de saída para verificar visualmente o resultado do alinhamento conforme figura 26.

Figura 24 – Remoção de keypoints outliers



Fonte: do autor

Figura 25 – Keypoints relacionados após seleção dos pontos mais próximos



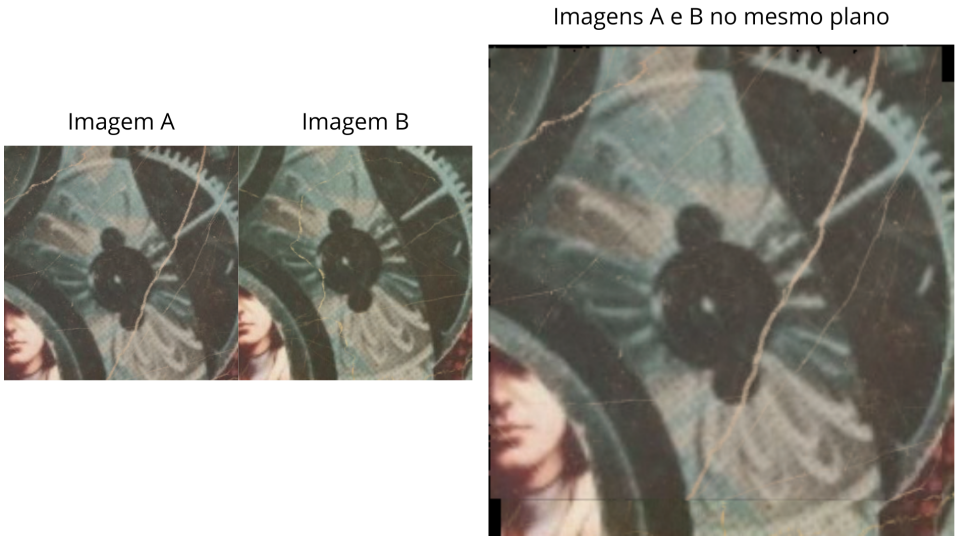
Fonte: do autor

5.1 Avaliação dos Resultados obtidos

Para avaliar os resultados obtidos durante o alinhamento usando PSNR e SSIM as imagens A e B devem ter o mesmo tamanho, então o alinhamento descrito anteriormente foi realizado sem redimensionar a imagen de saída. Também foi usada a máscara gerada pela detecção de defeitos descrita na seção 3.2 para remover os defeitos da imagen A após a realização do alinhamento. A figura 27 mostra esta processo, a imagem A transformada pela homografia tem a sua máscara obtida por um *threshold* para identificar as bordas da imagem combinada com a máscara detectada que é transformada pela mesma homografia, a imagem A é escrita subrepondo a imagem B apenas onde o valor da máscara é 1.

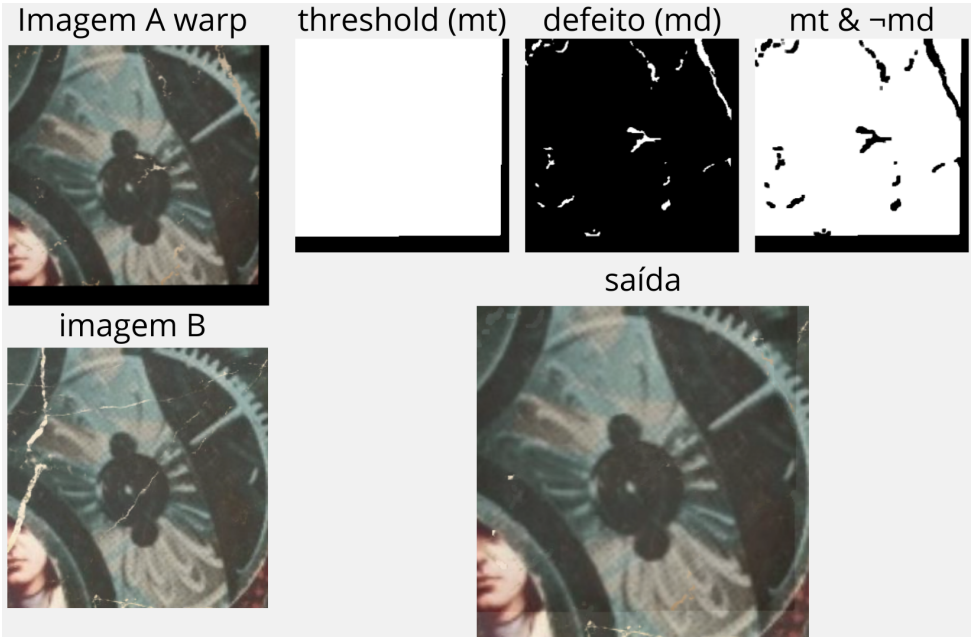
O processo de alinhamento de imagens com homografia foi realizado para os mesmos 270 pares de imagens usados na seção 4.2.2 usando a máscara de defeitos e também

Figura 26 – Alinhamento usando homografia



Fonte: do autor

Figura 27 – Aplicação da máscara de defeitos na imagem alinhada

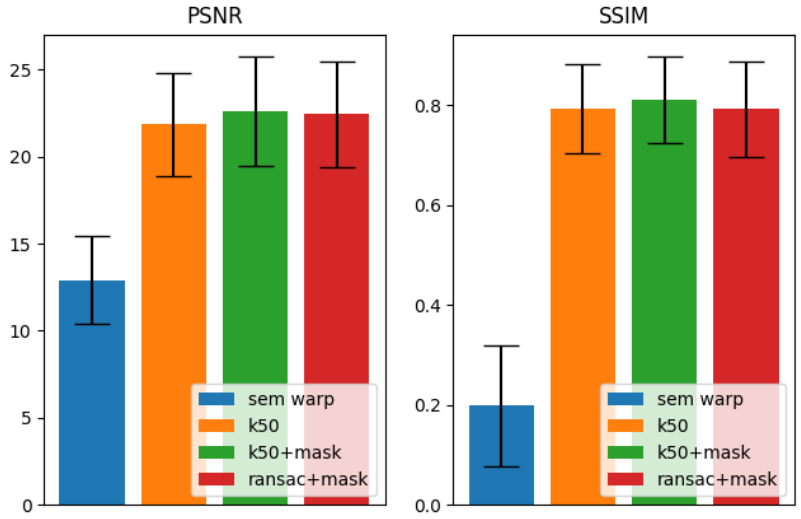


Fonte: do autor

sem usar as máscaras da detecção de defeitos a fim de quantificar a diferença entre usar a máscara e não usar a máscara, os testes também foram realizados usando RANSAC descrito em Fischler e Bolles (1981) com máscara de defeito, foram calculados o PSNR e o SSIM de cada execução e também das imagens antes de realizar o alinhamento para ter uma referência de quanto o alinhamento alterou o resultado obtido. A média e desvio padrão para as 270 imagens estão ilustradas na figura 28.

É possível verificar que o uso da máscara de defeitos teve um pequeno impacto

Figura 28 – Métricas calculadas

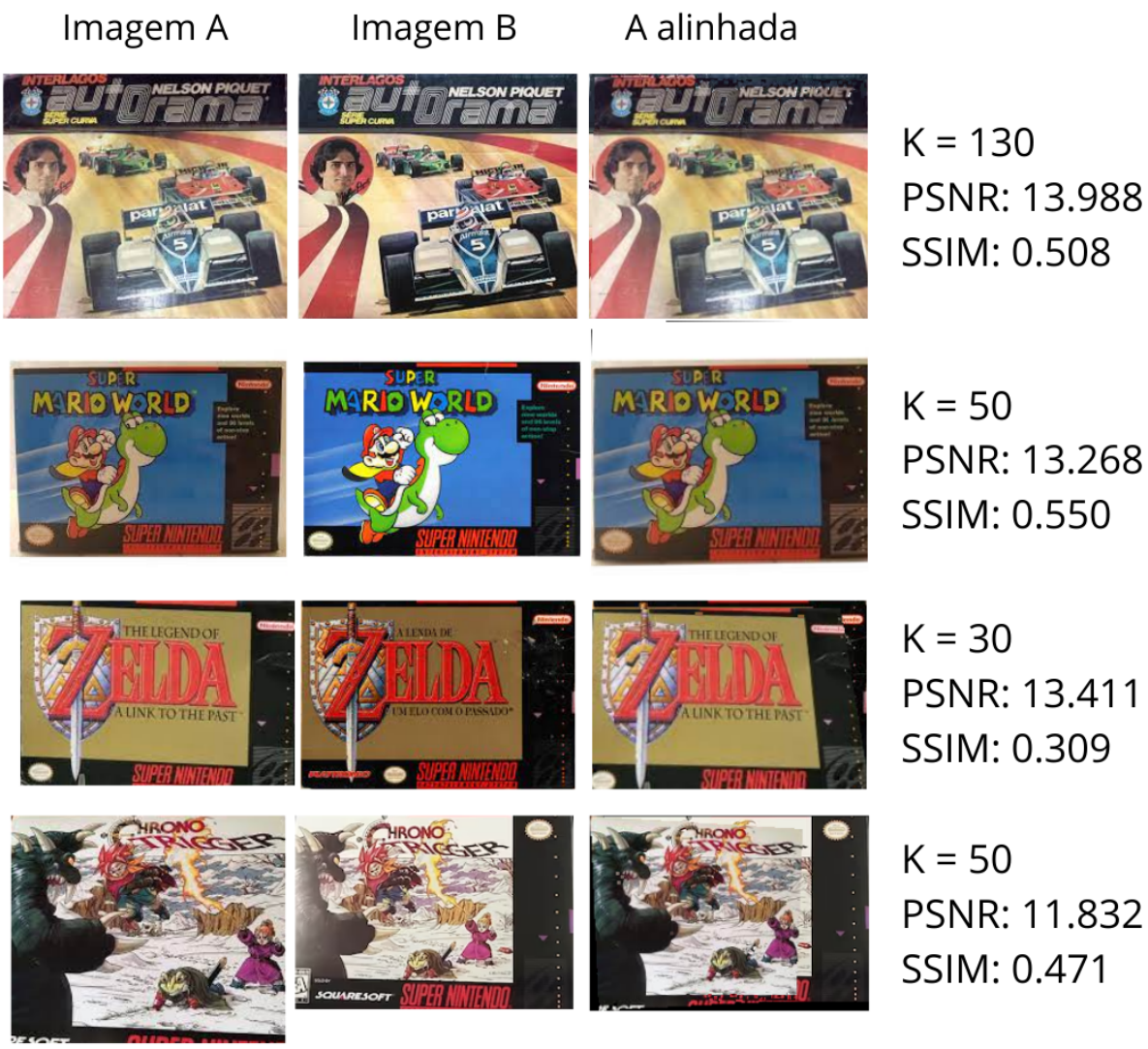


Fonte: do autor

positivo nos resultados de acordo com as duas métricas calculadas. A maior média para ambas as métricas foi do alinhamento descrito neste trabalho com média 22.624 para PSNR e 0.810 para o SSIM, usando RANSAC para detectar outliers e utilizando as máscaras de defeito o resultado foi um pouco inferior com médias 22,445 para o PSNR e 0,792 para o SSIM. O desvio padrão do RANSAC foi menor do que o desvio padrão da solução desenvolvida neste trabalho na métrica PSNR (3,037 3,124) e na métrica SSIM o RANSAC teve um desvio padrão mais alto (0,095 e 0,086).

Para verificar a aplicabilidade deste método de alinhamento a casos com imagens reais, foram reunidas algumas imagens de caixas de jogos e a imagem do caso de referência, para cada par de imagem uma imagem foi escolhida arbitrariamente como destino referência e a outra imagem foi alinhada seguindo o processo descrito anteriormente com o valor de k que teve o melhor resultado como ilustrado na figura 29. Os valores obtidos para o PSNR e SSIM foram muito menores que os obtidos no dataset sintético de avaliação, deve ser considerado qua as imagens de A e B reais não tinham exatamente o mesmo tamanho e eram menores que as imagens com defeito sintético, enquanto as imagens geradas correspondem a recortes de 256x256 de uma imagem maior as imagens encontradas geralmente tinham dimensões perto de 225x225 que correspondem à ilustração toda, quando não há bordas, então os detalhes da imagem vão ter detalhes com muito menos resolução que o dataset gerado, isto pode interferir no resultado obtido.

Figura 29 – Avaliação de resultados com imagens reais



Fonte: do autor

6 CONCLUSÃO

Durante o desenvolvimento deste trabalho, foram investigadas várias soluções de MFSR, com foco especial na etapa de alinhamento. Uma abordagem baseada em homografias, utilizando o método ORB para detectar pontos de interesse e relacioná-los entre as imagens, foi explorada como uma alternativa eficiente para o alinhamento de imagens de restauração baseada em múltiplos quadros. Ao contrário de cenários mais complexos, como os encontrados em visão computacional tradicional ou em problemas de *Burst SR* e *Video SR*, as imagens no domínio da restauração são, na maioria das vezes, planos com ilustrações. Isso permitiu explorar métodos mais simples e eficazes, evitando a necessidade de redes neurais profundas.

Uma contribuição deste trabalho foi a proposta de um método específico para detecção de *outliers* durante o processo de alinhamento, superando levemente o método RANSAC implementado pelo OpenCV, quando aplicado ao conjunto de dados sintético. Vale ressaltar que o RANSAC é um método robusto e amplamente utilizado, eficaz para uma variedade de problemas, enquanto o método desenvolvido neste trabalho pode ser usado em cenários nos quais existem pequenas diferenças lineares entre as imagens. No entanto, é importante reconhecer que a eficácia desse método pode depender das características específicas do conjunto de dados testado.

Durante a busca por imagens reais foi revelada uma dificuldade significativa em encontrar múltiplas representações de uma mesma ilustração original antiga, especialmente no contexto de jogos, música e filmes. Essa dificuldade é agravada pelo fato de que muitas embalagens e pôsteres foram refeitos por fãs. Além disso, quando imagens originais são localizadas, elas podem estar disponíveis apenas em baixa resolução ou capturadas a partir de ângulos muito diferentes, o que inviabiliza a aplicação do processo de restauração nos moldes descritos neste trabalho.

Os resultados obtidos em imagens reais foram inferiores em relação ao conjunto de dados sintético. Esse desafio pode ser atribuído às diferenças nas características e resolução das imagens reais usadas. Para executar o processo de restauração de imagens baseado em multiplos quadros de maneira eficaz, é necessário reunir um conjunto diversificado de imagens ou scans, preferencialmente com alta resolução e com pouca diferença na perspectiva. Portanto, a restauração baseada em múltiplos quadros não é apenas um desafio técnico, mas também demanda esforços consideráveis na obtenção de dados. Esse empenho é necessário para superar as limitações associadas à variabilidade nas perspectivas e resoluções das imagens, possibilitando a criação de restaurações que capturem com fidelidade a essência da obra original.

A transformação de N imagens para um mesmo plano de referência, onde as imagens são representadas como um bloco de dimensões $H \times W \times N$, abre possibilidades interessantes para a realização de diversas operações na terceira dimensão, que representa as diferentes imagens. Essa estrutura tridimensional permite realizar fusões, alinhamen-

tos mais refinados, correção de defeitos combinando as dimensões em uma representação unica das N imagens de entrada.

REFERÊNCIAS

AGARWAL, A.; JAWAHAR, C.; NARAYANAN, P. A survey of planar homography estimation techniques. **Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12**, International Institute of Information Technology Hyderabad, India, 2005. Citado na página 45.

AREFIN, M. R. et al. Multi-image super-resolution for remote sensing using deep recurrent networks. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2020. Citado na página 15.

BAI, K. **A Comprehensive Introduction to Different Types of Convolutions in Deep Learning**. 2019. Disponível em: <https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>. Citado na página 29.

BHAT, G. et al. Deep burst super-resolution. In: **2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Los Alamitos, CA, USA: IEEE Computer Society, 2021. p. 9205–9214. Citado 7 vezes nas páginas 9, 16, 29, 30, 34, 39 e 40.

BHAT, G. et al. Ntire 2021 challenge on burst super-resolution: Methods and results. In: **2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2021. p. 613–626. Citado na página 16.

BHAT, G. et al. Deep reparametrization of multi-frame super-resolution and denoising. In: **2021 IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 2440–2450. Citado 4 vezes nas páginas 22, 23, 40 e 57.

CALONDER, M. et al. Brief: Binary robust independent elementary features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). **Computer Vision – ECCV 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1. Citado na página 35.

DAI, J. et al. Deformable convolutional networks. In: **2017 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. p. 764–773. Citado 2 vezes nas páginas 24 e 25.

DUBROFSKY, E. Homography estimation. **Diplomová práce. Vancouver: Univerzita Britské Kolumbie**, v. 5, 2009. Citado na página 45.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 24, n. 6, p. 381–395, jun 1981. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/358669.358692>. Citado na página 48.

HORN, B. K.; SCHUNCK, B. G. Determining optical flow. **Artificial intelligence**, Elsevier, v. 17, n. 1-3, p. 185–203, 1981. Citado 2 vezes nas páginas 26 e 27.

HORé, A.; ZIOU, D. Image quality metrics: Psnr vs. ssim. In: **2010 20th International Conference on Pattern Recognition**. [S.l.: s.n.], 2010. p. 2366–2369. Citado 2 vezes nas páginas 36 e 37.

IGNATOV, A.; GOOL, L.; TIMOFTE, R. Replacing mobile camera isp with a single deep learning model. In: **2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2020. p. 2275–2285. Citado na página 22.

LECOUAT, B.; PONCE, J.; MAIRAL, J. Lucas-Kanade Reloaded: End-to-End Super-Resolution from Raw Image Bursts. In: **ICCV 2021 - International Conference on Computer Vision**. Virtual, France: [s.n.], 2021. p. 1–16. Disponível em: <https://hal.inria.fr/hal-03323885>. Citado na página 22.

LIU, S. et al. Robust multi-frame super-resolution based on adaptive half-quadratic function and local structure tensor weighted btv. **Sensors**, v. 21, n. 16, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/16/5533>. Citado 3 vezes nas páginas 20, 21 e 22.

LUO, Z. et al. Bsrt: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 998–1008. Citado 5 vezes nas páginas 23, 34, 39, 40 e 57.

LUO, Z. et al. Ebsr: Feature enhanced burst super-resolution with deformable alignment. In: **2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2021. p. 471–478. Citado 3 vezes nas páginas 34, 40 e 57.

MAISELI, B.; OGADA, E. A.; GAO, H. A multi-frame super-resolution method based on the variable-exponent nonlinear diffusion regularizer. **EURASIP Journal on Image and Video Processing**, v. 2015, 07 2015. Citado na página 15.

MILANFAR, P. **Super-resolution imaging**. [S.l.]: CRC press, 2010. Citado 3 vezes nas páginas 15, 19 e 21.

NAYAR, S. K. **First Principles of Computer Vision**. 2022. Disponível em: <https://fpcv.cs.columbia.edu/>. Citado 3 vezes nas páginas 25, 26 e 27.

NIKLAUS, S. **A Reimplementation of PWC-Net Using PyTorch**. 2018. <https://github.com/sniklaus/pytorch-pwc>. Citado 2 vezes nas páginas 34 e 57.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais : princípios, algoritmos e aplicações**. [S.l.]: Cengage Learning Brasil, 2007. Citado na página 15.

PELEG, S.; KEREN, D.; SCHWEITZER, L. Improving image resolution using subpixel motion. **Pattern Recognition Letters**, v. 5, p. 223–226, 03 1987. Citado na página 19.

RONNEBERGER, O.; P.FISCHER; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: **Medical Image Computing and Computer-Assisted Intervention (MICCAI)**. Springer, 2015. (LNCS, v. 9351), p. 234–241. (available on arXiv:1505.04597 [cs.CV]). Disponível em: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>. Citado na página 24.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). **Computer Vision – ECCV 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 430–443. ISBN 978-3-540-33833-8. Citado na página 35.

ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 1, p. 105–119, 2010. Citado na página 35.

RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: **2011 International Conference on Computer Vision**. [S.l.: s.n.], 2011. p. 2564–2571. Citado 2 vezes nas páginas 34 e 35.

SUN, D. et al. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: . [S.l.: s.n.], 2018. Citado 3 vezes nas páginas 33, 34 e 57.

TSAI, R. Multiframe image restoration and registration. **Advance Computer Visual and Image Processing**, v. 1, p. 317–339, 1984. Citado na página 19.

VALSESIA, D.; MAGLI, E. Permutation invariance and uncertainty in multitemporal image super-resolution. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, 2021. Citado na página 20.

WAN, Z. et al. Bringing old photos back to life. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2020. Citado 4 vezes nas páginas 31, 32, 33 e 57.

WANG, L. et al. Learning a single network for scale-arbitrary super-resolution. In: **2021 IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 4781–4790. Citado na página 57.

WANG, X. et al. Edvr: Video restoration with enhanced deformable convolutional networks. In: **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2019. p. 1954–1963. Citado 2 vezes nas páginas 25 e 26.

ZHANG, R. et al. The unreasonable effectiveness of deep features as a perceptual metric. In: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 586–595. Citado na página 37.

ZHAO, S. et al. Global matching with overlapping attention for optical flow estimation. In: **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2022. p. 17571–17580. Citado na página 41.

ZHU, X. et al. Deformable convnets v2: More deformable, better results. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citado na página 57.

APÊNDICE A – IMPLEMENTAÇÕES DE TERCEIROS

Neste apêndice são listados todas os códigos de autoria de terceiros usados de alguma forma, seja como material para consulta ou uso, durante o desenvolvimento da solução proposta neste trabalho.

Repositório da rede BSRT desenvolvido e descrito por Luo et al. (2022) disponível em https://github.com/Algolzw/BSRT.

Modelo da rede EBSR descrita em Luo et al. (2021) disponível no seguinte repositório https://github.com/Algolzw/EBSR

Implementação da rede Deep Burst Super-Resolution descrita por **??**), disponível no seguinte repositório https://github.com/goutamgmb/deep-burst-sr.

Código da implementação oficial em python da solução descrita em Bhat et al. (2021c) disponível em https://github.com/goutamgmb/deep-rep.

Código referente a implementação da rede ArbSR descrita por Wang et al. (2021) disponível em https://github.com/The-Learning-And-Vision-Atelier-LAVA/ArbSR.

Implementação extra oficial do modelo descrito em Zhu et al. (2019) disponível em: https://github.com/lucasjinreal/DCNv2_latest.

Implementação não oficial da rede PWC-net descrito originalmente em Sun et al. (2018), a versão usada foi desenvolvida por Niklaus (2018) e está disponível em https://github.com/sniklaus/pytorch-pwc.

Implementação da rede descrita em Wan et al. (2020) disponível no seguinte repositório https://github.com/microsoft/Bringing-Old-Photos-Back-to-Life.

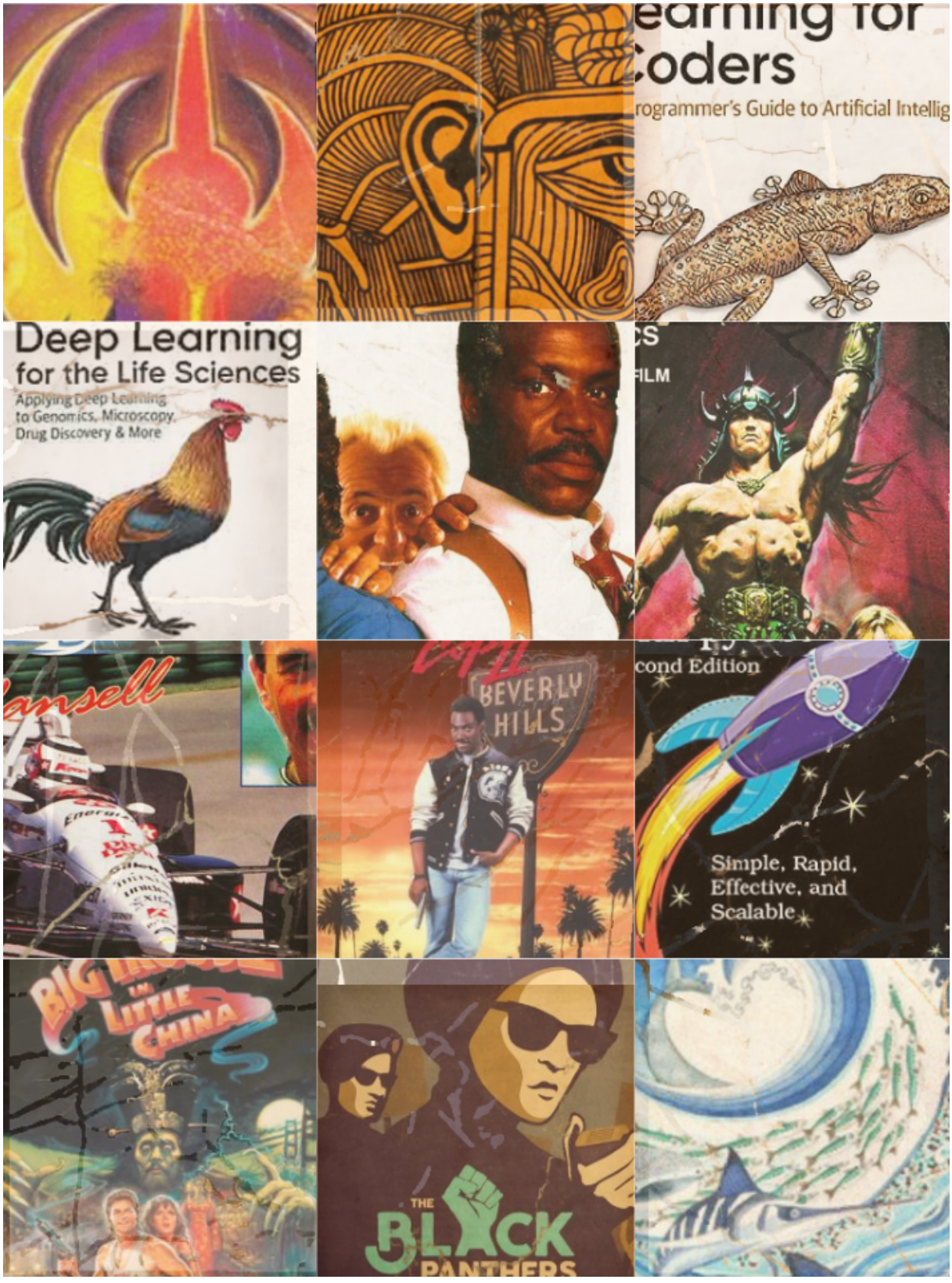
ANEXO A – EXEMPLOS DE IMAGENS GERADAS

Neste anexo são exemplificados algumas das imagens geradas durante os processos descritos neste trabalho. A figura 30 ilustra as imagens com defeito sintéticas usadas para testar o alinhamento com homografia e o impacto dos defeitos na rede PWC-Net.A figura 31 ilustra exemplos de imagens após realização do alinhamento usando homografia combinado com a detecção de defeitos.

Figura 30 – Exemplo de imagens com defeito sintético

Fonte: do autor

Figura 31 – Exemplo de imagens alinhadas com homografia

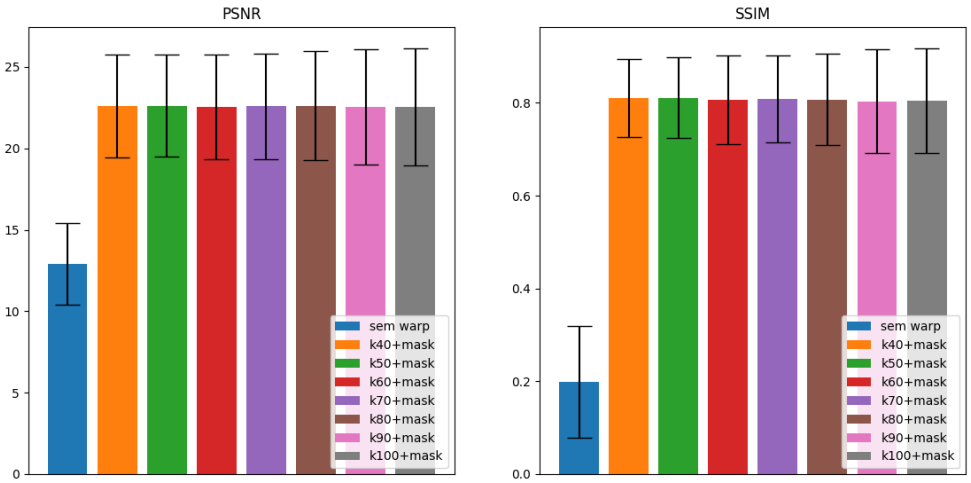


Fonte: do autor

ANEXO B – DESEMPENHO DO ALINHAMENTO PARA VALORES DIFERENTES DE K

O dataset de teste, composto por imagens com defeito sintético, foi executado para diferentes valores de k durante a seleção dos melhores *keypoints* para computar a homografia. A figura 32 mostra os resultados obtidos para as métricas de avaliação em função dos diferentes valores de k. Conforme o valor de k aumenta, a média das métricas muda muito pouco, enquanto o desvio padrão aumenta devido à inserção de mais pontos na computação da homografia. Pelo menos para o *dataset* de teste, o aumento de k não afeta positivamente o resultado.

Figura 32 – Resultados do alinhamento para diferentes valores de k



Fonte: do autor