

Reversão anaglífica baseada em método rápido

Felipe Carneiro Machado

Rudinei Goularte*

felipecmachado05@usp.br

rudinei@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

São Carlos, São Paulo

ABSTRACT

3D formats are still relevant due to their ability to convey depth through stereoscopic techniques. Among these, the anaglyph method stands out for its simplicity and low cost, encoding stereo pairs into a single image. Despite its advantages in compression and transmission, anaglyph encoding discards significant information, making its reversion complex. Reversing anaglyphs enables legacy content recovery and broader compatibility with various stereoscopic visualization methods. This work proposes an efficient approach to anaglyph reversion by exploring *Block Matching* algorithm, commonly used in video coding. Unlike traditional methods relying on pixel descriptors and global optimizations—often computationally expensive—our technique aims to handle radiometric differences between stereo views while maintaining high image quality and improved performance. Experimental results demonstrate the potential of this method as an alternative to existing solutions.

KEYWORDS

Anaglyph reversion, Block matching, Stereo images

1 INTRODUÇÃO

O consumo de conteúdo multimídia é uma realidade no cotidiano humano. Nesse contexto destaca-se o conteúdo 3D pela sua capacidade única de proporcionar a sensação de profundidade. Formalmente, a geração e o consumo de conteúdo 3D se dá pela técnica da estereoscopia, a qual consiste na representação de uma cena através de duas imagens com pequenas disparidades entre si, chamadas pares estéreo, simulando as diferenças entre as visões esquerda e direita dos olhos. Recebidas as visões separadas, o cérebro realiza a fusão das imagens, com sensação de profundidade [3].

Para o consumo de conteúdo estereoscópico, diversos métodos foram desenvolvidos, permitindo que cada olho receba apenas a sua visão da cena. Como o mais simples e de menor custo, o método anaglífico se destaca pela fusão das visões em uma única imagem, formada por dois canais de cor de uma das visões do par estéreo e um canal da outra. Tal método apresenta vantagens claras para compressão, armazenamento e transmissão, visto que reduz pela metade o volume de dados do conteúdo [13].

Além da compressão de dados, conteúdo anaglífico pode ser revertido a uma aproximação do par estéreo original, propiciando independência de dispositivo de visualização. A independência permite que conteúdo anaglífico seja consumido em qualquer meio¹ de visualização estereoscópica disponível, ou até como imagem 2D.

Entretanto, a reversão anaglífica é uma tarefa desafiadora devido ao grande descarte de informação durante a codificação. Algoritmos de reversão anaglífica presentes na literatura (Seção 2) baseiam-se na similaridade entre as visões do par estéreo, assumindo que os canais de cores descartados de uma visão podem ser recuperados a partir dos pixels correspondentes nos canais preservados da outra visão. Para tal, algoritmos do estado da arte baseiam-se na otimização de funções globais, ou correspondência local baseada em descritores de pixels, utilizando técnicas de colorização nas regiões onde não é possível a recuperação direta. Estes possuem a desvantagem de serem complexos e computacionalmente custosos [1, 7].

Diferentemente, métodos rápidos baseiam-se em buscas em blocos de pixels, objetivando maior eficiência. Uma alternativa ainda pouco explorada é a utilização de algoritmos de *Block Matching* [6], comuns em codificação de vídeo, para realizar a busca por regiões-par entre as visões esquerda e direita. Neste contexto, existem dois principais desafios que impactam a qualidade das imagens recuperadas. O primeiro é encontrar uma representação intermediária das imagens que permita comparação direta. Para isso, deve ser independente de cor, devido às diferenças radiométricas entre os canais de cor de um anáglifo. O segundo é a dimensão de uma janela de busca. Caso o tamanho seja menor que o necessário, o algoritmo não pode encontrar correspondências corretas, e caso seja maior, aumenta a chance de falsos positivos.

Assim, este trabalho propõe um método de reversão anaglífica baseado em *Block Matching*, capaz de: a) tratar as diferenças radiométricas equiparando-se em qualidade às melhores técnicas da literatura; b) ser mais eficiente que tais técnicas. As demais seções deste artigo estão assim organizadas: a Seção 2 apresenta uma revisão de trabalhos relacionados. A Seção 3 discute o método proposto, destacando a representação de imagem e o cálculo de janela de busca para o *Block Matching*. A Seção 4 apresenta os resultados experimentais obtidos comparando-se com trabalhos relacionados. A Seção 5 apresenta as considerações finais e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Algumas das primeiras técnicas de reversão anaglífica desenvolvidas modificam o processo de codificação, armazenando informações

*Orientador.

In: V Concurso de Trabalhos de Iniciação Científica (CTIC 2025). Anais Estendidos do XXXI Simpósio Brasileiro de Sistemas Multimídia e Web (CTIC'2025). Rio de Janeiro/RJ, Brasil. Porto Alegre: Brazilian Computer Society, 2025.
© 2025 SBC – Sociedade Brasileira de Computação.
ISSN 2596-1683

¹Métodos comuns utilizam um par estéreo como entrada: óculos passivos (polarizados) - técnica utilizada em cinemas, por óculos ativos (*shutter glasses*) e visualização autoestereoscópica (sem necessidade de óculos) [9].

adicionais com o fim de permitir a reversão, sendo úteis para fins de compressão, mas não sendo capazes de lidar com anáglifos legados [10, 15, 17]. Existem técnicas que utilizam apenas o anáglifo como entrada, sendo possível aplicá-las a qualquer conteúdo anaglífico, lidar com conteúdo legado e produzir anáglifos mais compactos, ao custo de resultados com qualidade de imagem inferior. As técnicas de interesse se encaixam no segundo grupo e, para o melhor do conhecimento adquirido, são descritas a seguir.

Joulin e Kang [5] propuseram uma variação do *SIFT-Flow*, para correspondência entre pixels. Para regiões oclusas, onde não é possível a recuperação direta, aplica-se uma extensão do algoritmo de colorização por otimização de Levin et al. [8]. Williem, Raskar e Park [14] modelaram o problema como uma minimização de energia global, também aplicando a colorização de Levin et al.. Ambas possuem resultados com boa qualidade de imagem, porém possuem complexidade computacional de, no mínimo, $O(n^3)$ e (n^4) , respectivamente [16].

Mais recentemente, foram propostas alternativas de menor complexidade computacional. Kunze et al. [7] explorou a aplicação de alinhamento de séries temporais ao problema, utilizando colorização baseada em *K-Nearest-Neighbours* (KNN), resultando no algoritmo SIRA com resultados de qualidade de imagem ligeiramente inferiores aos anteriores, porém com complexidade $O(n^2)$. Costa et al. [1] propôs a técnica *PixRadio*, que utiliza um descritor de pixel baseado em diferenças relativas, com o mesmo algoritmo de colorização do SIRA, atingindo os melhores resultados para qualidade de imagem, com complexidade linear.

Yugoshi [16] propôs o uso de *Block Matching* sobre representações de imagens baseadas em bordas, obtidas via Detector Canny, como uma opção de complexidade $O(n)$. A técnica, nomeada *ARBFLS*, utiliza uma janela de busca pequena e fixa, não tratando regiões oclusas através de colorização. Apesar de rápida, a qualidade dos resultados é inferior às técnicas anteriores.

Neste trabalho propõe-se o uso de *Block Matching* para realizar busca por correspondências, objetivando eficiência. Porém, diferente da técnica *ARBFLS*, utiliza uma representação de imagem que considera diferenças radiométricas e janela de busca variável, com o objetivo é alcançar melhorias em qualidade de imagem.

3 BMARBLE

Neste trabalho optou-se pelo uso do algoritmo *Full-Search* de *Block Matching* para a reversão anaglífica de imagens por sua simplicidade e eficiência (referido apenas como *Block Matching*). Algoritmos como os propostos por Joulin et al. (2013) e Williem et al. (2015), apesar de oferecerem maior qualidade, possuem também maior complexidade. Por isso, o foco está em algoritmos com complexidade linear, como o *PixRadio* e os algoritmos de *Block Matching*. Além disso, apesar de não explorado neste trabalho, *Block Matching* é um algoritmo paralelizável [2] que permite implementação em hardware VLSI [6].

A técnica *BMARBLE* (*Block Matching Anaglyphical Reversion Based on Laplacian of Gaussian Encoding*) proposta subdivide-se em: separação dos canais, representação intermediária, determinação da janela de busca e busca de correspondências, reciprocidade, transferência de cor e colorização, a serem detalhadas a seguir.

3.1 Separação e preparo de canais

Após a leitura do anáglifo de entrada ($An = R_E, G_D, B_D$), é realizada a separação dos canais esquerdo e direito, preenchendo os canais vazios com os valores existentes: imagem da visão esquerda, $IE = R_E, R_E, R_E$; imagem da visão direita, $ID = G_D, G_D, B_D$. A escolha de G_D para preencher o R_D faltante segue a proposta de Yugoshi et al. [16], devido ao canal verde possuir maior impacto na luminância.

3.2 Representação intermediária

Após a separação, os canais preenchidos possuem as formas corretas, mas com cores trocadas - diferenças radiométricas. Devido à impossibilidade da comparação direta entre os canais, é necessário gerar uma representação independente de cor. Para tal, é útil aplicar detecção de bordas, visto que as formas se mantêm dentro do par anaglífico, com pouco impacto das diferenças radiométricas.

Para evitar degradação de qualidade é desejável que a detecção seja resiliente a ruído, capte bordas em quaisquer direções e preserve o máximo de informação da imagem. Alternativas tradicionais, como o detector de bordas de Canny ou de Marr-Hildreth, realizam a limiarização e geram imagens binárias, descartando informação [4].

Assim, diferente de trabalhos relacionados, neste trabalho optou-se pelo uso de um filtro *LoG* (*Laplacian of Gaussian*), definido pela Equação 1, onde σ é o desvio padrão da gaussiana, o qual foi escolhido experimentalmente como 0,6. A parte gaussiana do filtro é responsável por suavizar as imagens, reduzindo ruído. A parte laplaciana detecta alterações bruscas de intensidade através da segunda derivada de forma isotrópica (invariante à rotação). Sua aplicação consiste em uma única convolução, tornando-o eficiente. A convolução do *LoG* com os canais convertidos para tons de cinza mantém mais informação do que aplicar filtros tradicionais, gerando a representação sobre a qual serão realizadas as comparações.

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (1)$$

3.3 Janela de busca e correspondências

A *BMARBLE*, utiliza *Block Matching* para encontrar pixels correspondentes entre os canais equivalentes das visões esquerda e direita, a fim de transferir as cores e recuperar o par estéreo.

Inicialmente, ambos os canais processados são divididos em blocos de 8x8 pixels. Cada bloco é comparado com todos os blocos dentro de uma **janela de busca** no canal correspondente da outra visão através do *SAD* (*Sum of Absolute Differences*). O *SAD* entre um bloco do canal esquerdo, cuja coordenada do canto superior esquerdo é dada pelo par (x_E, y_E) , com um bloco do canal direito de coordenadas (x_D, y_D) é dado pela Equação 2, na qual LoG_E e LoG_D correspondem aos canais processados. O bloco determinado como correspondente é aquele que minimiza o *SAD*.

$$SAD(x_E, y_E, x_D, y_D) = \sum_{i=0}^7 \sum_{j=0}^7 |LoG_E(x_E + i, y_E + j) - LoG_D(x_D + i, y_D + j)| \quad (2)$$

O tamanho da janela de busca, diferente das abordagens encontradas na literatura, é definida dinamicamente, de acordo com as disparidades ² presentes na imagem.

²Disparidade pode ser entendida como a distância, em pixels, do deslocamento entre dois pixels correspondentes.

Para definir a dimensão ótima para uma janela de busca, executa-se, inicialmente o *Block Matching* com uma janela de 60 pixels (definido empiricamente), gerando um histograma das disparidades obtidas. A partir do histograma, seleciona-se a maior disparidade com frequência maior que 0,01 (definida empiricamente), tomada como a dimensão máxima da janela. Todas as correspondências com disparidade maior que o valor selecionado são consideradas inválidas. Então, repete-se o *Block Matching*, com a janela atualizada, somente para os blocos com correspondências inválidas. Desse modo busca-se reduzir o número de falsos positivos resultantes da busca, melhorando a qualidade da imagem recuperada.

Também é possível melhor o desempenho da busca. Primeiramente, pares estéreos são capturados por câmeras de lentes pareadas não existe paralaxe negativa [13]. Isto implica que todas as disparidades se dão em apenas um sentido. Logo, a busca também. Adicionalmente, como as lentes de tais câmeras são alinhadas horizontalmente, implica que a captura possui disparidade vertical nula [1], permitindo que a busca seja feita apenas na horizontal.

3.4 Verificação de reciprocidade

A existência de regiões oclusas, de baixa textura e altas diferenças radiométricas implica em correspondências erradas para um dado pixel (x, y) . Assim, nesta etapa é realizada a verificação da validade das correspondências através da reciprocidade.

Um pixel (x, y) é considerado válido se a diferença entre sua disparidade a partir da visão esquerda e sua disparidade a partir da visão direita estiver dentro de um limiar $\delta = 1$ [1].

3.5 Transferência direta de cor

A partir do mapa de disparidades válidas obtido com as etapas anteriores, realiza-se a cópia direta de cor entre os canais - cores faltantes em cada pixel válido de cada canal recebem os valores do pixel correspondente, recuperando parcialmente o par estéreo.

3.6 Colorização

Pixels em regiões de oclusão não possuem correspondentes, não podem ser recuperados com a técnica descrita e necessitam ser colorizados. Optou-se pelo algoritmo de colorização utilizado por Costa et al. [1] devido a seu menor custo computacional quando comparado aos de trabalhos relacionados ([5, 14]).

Para cada pixel inválido, busca-se pixels similares em uma janela quadrada a seu redor. A similaridade é calculada pela diferença absoluta de intensidades nos canais de cores originais. É realizada a seleção dos pixels cuja diferença seja inferior a um limiar γ , e atribui-se a média dos valores dos pixels similares e válidos como valor de colorização. A colorização é realizada de forma incremental, processando apenas pixels vizinhos a ao menos um pixel válido, minimizando o surgimento de cores espúrias.

3.7 Análise de custo

3.7.1 Representação intermediária. A aplicação do filtro Laplaciano da Gaussiana consiste em uma convolução entre a imagem e o kernel LoG, cuja complexidade é $O(N \cdot t_{kernel}^2)$, onde N representa o número de pixels e t_{kernel} , o tamanho do kernel de convolução, determinado a partir do desvio padrão da gaussiana, devendo ser o menor inteiro ímpar maior que 6σ [4]. Para o caso do *BMARBLE*,

escolheu-se empiricamente o valor $\sigma = 0.6$, portanto $t_{kernel} = 5$ e a etapa de pré-processamento tem complexidade $O(2 \cdot N \cdot 25)$.

3.7.2 Busca por correspondências e determinação da janela de busca. O algoritmo de *Full-Search* compara cada bloco com todos os outros presentes dentro da janela de busca através do *SAD*. Dessa forma, são realizadas $t_{bloco}^2 \cdot t_{janela}$ operações para cada um dos N/t_{bloco}^2 blocos, para os canais esquerdo e direito. A subetapa de determinação da janela realiza o *Block Matching* para todos os blocos da imagem, resultando em uma complexidade de $O(2 \cdot N \cdot 60)$, assumindo uma janela de 60 pixels.

Determinada a janela de busca, apenas é necessário repetir o *Block Matching* para os blocos inválidos. Como o método de determinação da janela descarta correspondências com frequência inferior a 1%, é seguro assumir que cerca de 1% das correspondências estarão inválidas, resultando em uma complexidade de $O(\frac{2}{100} \cdot N \cdot t_{janela})$. Somando as duas subetapas, temos $O(2n(60 + \frac{1}{100} t_{janela}))$

3.7.3 Colorização. De acordo com Costa (2021) [1], o custo do algoritmo de colorização utilizado é proporcional ao número de pixels inválidos, o que implica em complexidade $O(N)$, assim como a verificação de reciprocidade. Desse modo, a etapa de colorização possui complexidade temporal $O(N)$.

3.7.4 Comparação com PixRadio. Embora PixRadio e *BMARBLE* pertençam à mesma classe de complexidade temporal, *BMARBLE* se destaca por ter constantes ocultas menores, conforme demonstrado abaixo, baseando-se nas equações e parâmetros fornecidos por Costa et al. (2021), o que o torna intrinsecamente mais rápido na prática. Como ambas as técnicas utilizam colorização idêntica, é suficiente comparar apenas as etapas prévias, representação intermediária ($O(2 \cdot N \cdot 25)$) e busca por correspondências com determinação de janela ($O(2n(60 + \frac{1}{100} t_{janela}))$) para *BMARBLE*, cuja soma resulta na equação 3; e cálculo de representações ($2 \cdot N \cdot 312$), custos ($N \cdot 312 \cdot t_{janela}$), agregação ($2 \cdot N \cdot 50 \cdot t_{janela}$) e obtenção de disparidades ($2 \cdot N \cdot t_{janela}$), para *PixRadio*, cuja soma resulta na equação 4.

Nota-se que as constantes 312 e 50 nas equações do *PixRadio* advêm, respectivamente, do tamanho do vetor de representações da técnica e do tamanho do filtro gaussiano (considerando separabilidade) utilizado na etapa de agregação. Combinando as etapas, chega-se à constante 207 [1].

$$Custo_{BMARBLE} = 2 \cdot N \cdot (85 + \frac{t_{janela}}{100}) \quad (3)$$

$$Custo_{Pixradio} = 2 \cdot N \cdot (312 + 207 \cdot t_{janela}) \quad (4)$$

Comparando-se as equações 3 e 4 percebe-se que a *PixRadio* possui uma constante multiplicativa para o tamanho da janela muito maior que a da *BMARBLE*, confirmando um ganho de velocidade significativo.

4 RESULTADOS

A *BMARBLE* e a *ARBFLS* (como descrito em Yugoshi et al. [16]) foram implementados em linguagem *Python* com as bibliotecas *Numpy* e *OpenCV*, sem paralelismo ou otimizações. O código está disponível como software aberto no *GitHub*³. A comparação com a *PixRadio* segue os resultados reportados em [1].

³<https://github.com/FelipeCarneiroMachado/BMARBLE>

Utilizou-se os *datasets* de 2001 e 2003 da base *Middlebury* [11, 12]. Os *datasets* são compostos por, respectivamente, 7 e 2 imagens, capturadas sobre 10 diferentes disparidades, de modo a compor pares estéreo. As disparidades adotadas correspondem às utilizadas por Costa [1], de modo a realizar comparações justas entre técnicas.

Como métricas de qualidade, foram escolhidas a PSNR (*Peak Signal to Noise Ratio*), amplamente utilizada na literatura, e a SSIM (*Structural Similarity Index Measure*), considerada mais correlacionada com a percepção visual humana, porém ainda pouco utilizada na área. Para o caso da *PixRadio*, os valores de PSNR foram extraídos diretamente de Costa et al. [1], porém, para o SSIM, foi utilizado o código disponibilizado pelo autor.

A Tabela 1 apresenta os resultados de qualidade das imagens recuperadas. Em termos de PSNR a *BMARBLE* supera largamente a *ARBFLS*, porém, é superada pela *PixRadio* por menos de 3%. Em termos de SSIM, a *BMARBLE* supera as técnicas comparadas.

5 CONCLUSÕES

A técnica *BMARBLE* atingiu os objetivos propostos. Ao utilizar *Block Matching* para calcular correspondências aliado a um modo enxuto de calcular a janela de busca a *BMARBLE* consegue ser mais eficiente que as técnicas em estado da arte, conforme demonstrado na Seção 3.7. Ainda, esse ganho em eficiência vem sem perdas significativas em qualidade de imagem recuperada: 1) o modo dinâmico de determinar a janela de busca limita encontrar falsos positivos (Seção 3.3) pois é baseado na maior disparidade presente da imagem; 2) a representação intermediária utilizada nas buscas contém mais informações das imagens originais do que métodos correlatos (Seção 3.2). Esses fatos ficaram comprovados na análise experimental (Seção 4), onde a *BMARBLE* fica muito próxima da melhor técnica conhecida (*PixRadio*) em termos de PSNR (menos de 3% de diferença) e supera as duas técnicas comparadas em termos de SSIM, conforme apresentado na Tabela 1.

Como trabalhos futuros, pode-se: estender a técnica para vídeo anaglífico; desenvolver novas técnicas de colorização, de modo a reduzir o tempo de processamento bem como melhorar a qualidade visual das imagens; paralelizar a implementação do *Block Matching* da *BMARBLE* para obter maior eficiência.

6 AGRADECIMENTOS E CONTRIBUIÇÕES DOS AUTORES

Este trabalho foi financiado pelo Programa Unificado de Bolsas da Universidade de São Paulo. Felipe Carneiro Machado contribuiu ao

trabalho com conceitualização, análise, desenvolvimento do software, validação, experimentação e escrita do texto. Rudinei Goularte contribuiu ao trabalho com conceitualização, supervisão, análise, metodologia, gerenciamento do projeto, revisão e edição do texto.

REFERENCES

- [1] A. L. D. Costa. 2021. *Reversão anaglífica - um novo método baseado em cálculo de correspondências robusto a diferenças radiométricas*. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional). ICMC/USP, São Carlos – SP.
- [2] M. Dasygenis and P. Michailidis. 2014. Evaluating modern parallelization techniques on block matching algorithms. In *Proc. of the 18th Panhellenic Conference on Informatics*. <https://doi.org/10.1145/2645791.2645839>
- [3] B. Goldstein. 2013. *Sensation and Perception* (8 ed.). Wadsworth Cengage Learning, Belmont, CA. 1–490 pages.
- [4] R. C. Gonzalez and R. E. Woods. 2006. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA, Chapter 10.
- [5] A. Joulin and S. B. Kang. 2013. Recovering Stereo Pairs from Anaglyphs. In *2013 IEEE Conf. on Computer Vision and Pattern Recognition*. 289–296. <https://doi.org/10.1109/CVPR.2013.44>
- [6] J. Konrad. 2005. 3.10 - Motion Detection and Estimation. In *Handbook of Image and Video Processing (Second Edition)* (second edition ed.), AL BOVIK (Ed.). Academic Press, Burlington, 253–274. <https://doi.org/10.1016/B978-012119792-6/50079-6>
- [7] L. F. Kunze, R. Goularte, and E. P. M. Sousa. 2020. SIRA - An efficient method for retrieving stereo images from anaglyphs. *Signal Process. Image Commun.* 85, 115866 (July 2020), 115866.
- [8] A. Levin, D. Lischinski, and Y. Weiss. 2004. Colorization using Optimization. *ACM Transact. on Graphics* 23 (06 2004). <https://doi.org/10.1145/1015706.1015780>
- [9] Bernard Mendiburu. 2009. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press.
- [10] F. M. Rodrigues, J. K. Yugoshi, and R. Goularte. 2016. HaaRGlyph: A New Method for Anaglyphic Reversion in Stereoscopic Videos. In *Proc. of the 22nd Brazilian Symposium on Multimedia and the Web*. <https://doi.org/10.1145/2976796.2976864>
- [11] D. Scharstein and R. Szeliski. 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Intern. Journal of Computer Vision* 47 (04 2002), 7–42. <https://doi.org/10.1023/A:1014573219977>
- [12] D. Scharstein and R. Szeliski. 2003. High-accuracy stereo depth maps using structured light. *Comput. Vision Pattern Recognit.* 1, 1–195. <https://doi.org/10.1109/CVPR.2003.1211354>
- [13] StereoGraphics Corporation. 1997. *Stereographics Developers Handbook: Background on Creating Images for CrystalEyes and SimulEyes*.
- [14] W. Williem, R. Raskar, and I. K. Park. 2015. Depth Map Estimation and Colorization of Anaglyph Images Using Local Color Prior and Reverse Intensity Distribution. In *2015 IEEE Intern. Conf. on Computer Vision (ICCV)*. 3460–3468. <https://doi.org/10.1109/ICCV.2015.395>
- [15] Peter Wimmer. [n. d.]. *DeAnaglyph*. Retrieved July 03, 2025 from https://www.3dtrv.at/knowhow/DeAnaglyph_en.aspx
- [16] J. K. Yugoshi. 2018. *Reversão anaglífica baseada em busca local rápida*. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional). ICMC/USP, São Carlos – SP.
- [17] M. R. U. Zingarelli, L. A. Andrade, and R. Goularte. 2011. Reversing Anaglyph Videos Into Stereo Pairs. In *Proceed. of the 17th Brazilian Symposium on Multimedia and the Web*. <https://sol.sbc.org.br/index.php/webmedia/article/view/5621>

Table 1: Resultados

Imagem	ARBFLS				BMARBLE				PixRadio			
	Esquerda		Direita		Esquerda		Direita		Esquerda		Direita	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Teddy	14.47	0.50	18.17	0.84	22.31	0.84	25.99	0.96	26.81	0.89	30.08	0.96
Barn1	20.57	0.64	21.03	0.86	30.31	0.94	30.46	0.97	30.59	0.91	32.62	0.96
Poster	17.10	0.52	18.89	0.83	26.75	0.92	28.05	0.95	29.06	0.90	30.07	0.96
Cones	14.72	0.39	17.09	0.79	19.51	0.81	22.58	0.98	22.74	0.82	25.78	0.95
Venus	19.90	0.63	21.06	0.88	30.21	0.94	30.55	0.98	32.30	0.93	34.86	0.96
Bull	22.05	0.71	24.44	0.90	30.94	0.94	31.32	0.98	33.54	0.92	37.22	0.96
Barn2	22.54	0.77	23.65	0.91	31.45	0.95	32.49	0.98	32.43	0.93	36.17	0.97
Sawtooth	18.00	0.58	21.05	0.87	29.05	0.94	32.38	0.98	29.70	0.92	33.15	0.97
Tsukuba	19.18	0.65	20.29	0.87	30.65	0.95	30.74	0.98	30.30	0.93	31.35	0.98
Média	18.73	0.60	20.63	0.86	27.91	0.92	29.40	0.97	29.72	0.90	32.37	0.96