Predição Intra-Quadro Baseada em Aprendizado Profundo para Light Fields Densos

Italo Machado
Bruno Zatt
Daniel Palomino
idmachado@inf.ufpel.edu.br
zatt@inf.ufpel.edu.br
palomino@inf.ufpel.edu.br
Universidade Federal de Pelotas
Pelotas, Rio Grande do Sul

Attilio Fiandrotti
Universidade de Turin
Turin, Itália
attilio.fiandrotti@unito.it

ABSTRACT

This study proposes a new strategy for intra prediction of dense light fields by reinterpreting the problem as image inpaiting and using convolutional neural networks. Multiple architectures and training techniques were evaluated in order to identify the most efficient configuration for performing intra prediction in a video encoder for this type of data. Separate networks were trained for each of the 3 block sizes of the encoder and their performance evaluated separately and together. The results showed that the use of convolutional neural networks as intra predictors significantly improves coding efficiency in the EVC encoder, achieving an average BD-rate reduction of -30.53%.

KEYWORDS

Light Fields, Predição Intra, Redes Neurais Convolucionais, Codificação, Aprendizado Profundo

1 INTRODUÇÃO

Light Fields (LFs) fornecem uma representação abrangente de uma cena ao capturar informações de múltiplos pontos de vista, tornando-os altamente úteis para aplicações como exibições 3D e realidade virtual. Os dados contidos nos LFs melhoram a estimativa de profundidade e possibilitam operações avançadas de imagem, como o refoco após a captura da imagem [1, 14].

Ainda, os LFs podem ser visualizados em vários formatos, incluindo super multiview, imagens de plano epipolar e Lenslet (Figura 1), onde as vistas são rearranjadas em *Micro Images* (MIs). Cada um destes formatos preserva as correlações entre as vistas e oferecem benefícios distintos para análise e compressão. Além destes formatos as diferentes vistas do LF podem ser reorganizadas como *frames* sequenciais de um pseudo-vídeo (PVS) [7], contudo, esta representação desacopla a estrutura 4D das vistas e suas correlações.

As informações angulares adicionais agregadas por um LF causam um aumento considerável no tamanho dos dados necessários para representar os LFs, gerando desafios significativos para armazenamento e transmissão. Para tratar estes obstáculos alguns estudos utilizam codificadores de vídeo para comprimir imagens LF como PVS [6, 7, 13]. Ainda, baseando-se nesse método, algumas abordagens codificam um grupo selecionado de vistas-chave e, posteriormente, sintetizam as restantes [5]. Outros métodos empregam redes neurais convolucionais (RNCs) com mecanismos de atenção para alcançar resultados semelhantes [8]. No entanto, apesar de imitar os quadros de um vídeo para aproveitar as semelhanças angulares como se fossem temporais, ambas possuem naturezas distintas, o que impossibilita a devida exploração das informações angulares.

Explorando mais a fundo o processo de codificação, os codecs fragmentam as imagens em blocos e comprimem cada bloco de forma independente. Ferramentas de codificação, como predições intra-quadro e inter-quadros, aproveitam a redundância de informações tanto dentro dos quadros de vídeo quanto entre eles. A predição inter identifica redundâncias em diferentes quadros, enquanto a predição intra se foca no mesmo quadro. No entanto, essas ferramentas foram projetadas para vídeos convencionais e não se adaptam bem à estrutura 4D dos LFs. Por exemplo, a predição intra em codificadores de vídeo utiliza blocos vizinhos previamente decodificados para predição por interpolação, o que geralmente não se ajusta adequadamente ao comportamento de gradiente dos LFs no formato Lenslet. Conforme ilustrado na Figura 1, devido aos pequenos deslocamentos horizontais e verticais entre as vistas do LF, os *micro images* adjacentes exibem um padrão de gradiente consistente, tanto dentro de cada *micro image* quanto entre eles, diferindo significativamente do comportamento dos pixels em vídeos ou imagens naturais.

Para resolver essa questão, [15] propõe o uso de seis redes neurais convolucionais (RNCs) para aproveitar as correlações angulares nos LFs no formato lenslet durante a predição intra no codificador EVC. Duas redes são treinadas para cada tamanho de bloco (32, 16, 8), uma para áreas focadas e outra para áreas desfocadas. Áreas focadas, com informações angulares mais densas, usam blocos vizinhos padrão do HEVC, enquanto áreas desfocadas, com semelhanças mais esparsas, usam amostras de referência de MPs distantes. Modificações no decodificador são necessárias para acessar esses vizinhos distantes, introduzindo sobrecargas computacionais e de armazenamento.

Neste estudo, é proposta uma nova estratégia, abordando o desafio de predição intra de *light fields* como um problema de preenchimento. Trabalhos anteriores [12] demonstraram sucesso ao interpretar a predição intra para imagens naturais como um problema de preenchimento, utilizando blocos vizinhos como entrada para uma

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2024). Juiz de Fora, Brazil. Porto Alegre: Brazilian Computer Society, 2024.

ISSN 2966-2753

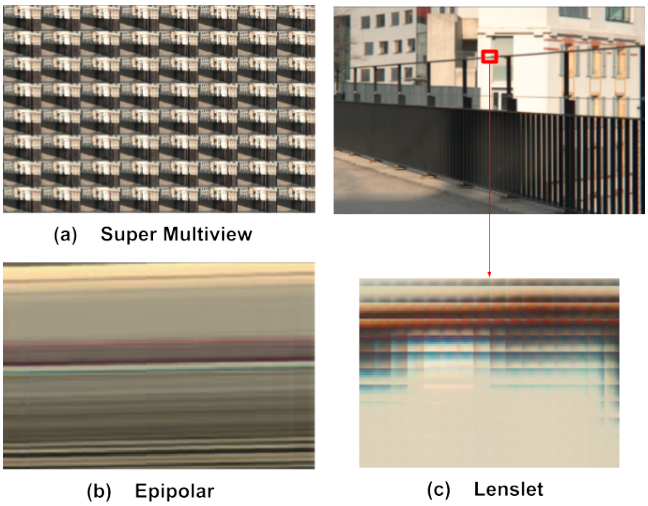

Figura 1: Formas de representar um *Light Field*

CNN que reproduz a distribuição de pixels do bloco alvo. Defendemos que essa abordagem pode ser aplicada de forma eficaz para preencher os comportamentos de gradiente dos LFs no formato lenslet, explorando as redundâncias angulares de maneira eficiente. Para isso, nossa proposta emprega RNCs como preditores intra para todos os tamanhos de bloco do codificador de vídeo EVC [2, 11], uma técnica que pode ser generalizada para outros codificadores de vídeo com estruturas de bloco semelhantes. Apresentamos uma análise das diferentes arquiteturas, técnicas e hiperparâmetros avaliados até chegarmos na melhor solução.

2 ARQUITETURA PROPOSTA

No decorrer da predição intra, o codificador EVC emprega um contexto composto por três blocos vizinhos para estimar o bloco atual a ser predito. Este trabalho visa realizar esta estimação através da técnica de preenchimento de imagens [1], na qual um *autoencoder* é utilizado para gerar pixels ausentes com base nos pixels dos blocos vizinhos. Portanto, conforme ilustrado na Figura 2, a arquitetura proposta recebe como entrada um contexto de tamanho HxW, contendo três blocos vizinhos juntamente com um bloco alvo vazio, e gera um contexto de saída com as mesmas dimensões. Considerando que o EVC divide a imagem em blocos de tamanho 32x32, 16x16 e 8x8, as dimensões HxW podem variar entre 64x64, 32x32 e 16x16. Note que uma arquitetura foi treinada separadamente para cada tamanho de bloco tendo em vista que as distribuições entre os blocos vizinhos e alvo mudam consideravelmente de acordo com o tamanho de bloco.

O codificador da arquitetura é constituído por 5 camadas convolucionais com *stride* 1 intercaladas por 1 uma convolução *stride* 2. Todas as camadas do codificador utilizam kernels de tamanho 3x3 e são seguidas por uma função de ativação PreLU. Note que a última camada de codificação não é seguida por uma camada convolucional com *stride* 2, pois isso reduziria um contexto de entrada de 8x8 a zero e, consequentemente, necessitaria de uma rede diferente para blocos de tamanho 8x8.

No lado do decodificador, a arquitetura consiste em quatro camadas. Com exceção da camada final, todas as camadas incluem uma operação de *up-sample* com um fator de 2, seguida por uma camada convolucional com passo 1, refletindo o lado do codificador. A última camada é uma convolução transposta com kernels de 4x4 e um passo de 2, culminando em uma função Sigmóide que produz valores de luminância entre 0 e 1. É importante mencionar que a rede gera um contexto de saída com o mesmo tamanho da entrada, e a região de interesse é posteriormente cortada, pois experimentos demonstraram que essa estratégia resulta em uma eficiência aprimorada [1]. A rede possui um total de 3 milhões de parâmetros ajustáveis e um tamanho estimado de 18,37 MB. Além disso, ao contrário de abordagens similares, como [1], que empregam convoluções mascaradas, optamos por utilizar convoluções regulares mais simples.

2.1 Preparação do Dataset

O conjunto de dados EPFL [10] foi dividido em dois grupos de 105 imagens para treinamento e 12 imagens para teste, seguindo a metodologia proposta em [15]. Essas imagens foram extraídas de arquivos brutos usando o pacote *plenopticam* [4], resultando em LFs com resolução de 622x432x13x13 onde as vistas escurecidas foram descartadas.

Posteriormente, para alinhar o tamanho dos *micro images* aos tamanhos de bloco, as 5 vistas mais externas foram descartadas, gerando um LF com 8x8 vistas. Por fim, os LFs foram reorganizados no formato de lenslet para melhor explorar as redundâncias angulares das vistas.

Para avaliar a melhor forma de treinar a rede e evitar *overfitting*, foram conduzidos experimentos para treinar as redes usando diferentes estratégias para selecionar os exemplos de treino e técnicas de transformação. Estas incluíram a seleção sequencial de blocos e a escolha aleatória de blocos de várias áreas das LFs. Além disso, treinamos a rede sem aplicar transformações e também aplicando rotações em 90, 180, 270 ou 360 graus e espelhamento horizontal a cada bloco com uma probabilidade de 50%, resultando em todas as possíveis variações formando blocos quadrados.

Os experimentos indicaram que a abordagem mais eficaz para evitar *overfitting* foi selecionar blocos aleatoriamente e aplicar transformações durante o treinamento. Especificamente, 25% do número total possível de blocos foram selecionados aleatoriamente para cada LF, com sobreposição permitida entre blocos. Aumentar o número de blocos de amostra por LF levou a tempos de treinamento mais longos sem melhorias nas curvas de aprendizado ou eficiências na predição. Outro cuidado importante no recorte dos blocos foi de sempre mantê-los alinhados com o início e fim dos macro pixels, o que ocorrerá naturalmente durante o processo de codificação em vista que estes possuem tamanhos múltiplos dos tamanhos de bloco do codificador.

As redes neurais foram então treinadas usando os dados gerados por 100 épocas, o otimizador Adam foi utilizado com uma taxa de aprendizado de 1^{-4} e uma taxa de decaimento exponencial de 0.2. O otimizador SGD (*Stocastic Gradient Descent*) e *learning rates* de 1^{-5}, 2^{-5}, 2^{-3}, 1^{-3} também foram avaliados, contudo, estes não atingiram melhoras nas convergências.

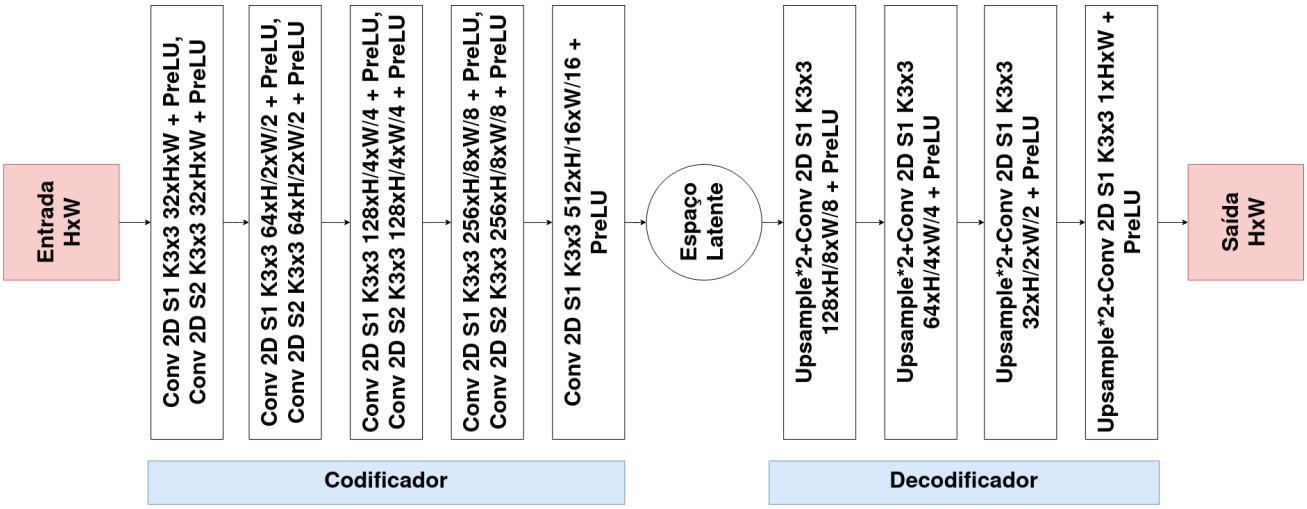

Figura 2: Diagrama de Blocos da Arquitetura Proposta

2.2 Integração com o Codificador

O modelo treinado reside em um servidor com memória GPU. Ele recebe blocos de entrada por meio de conexões UDP e transmite de volta os tamanhos de bloco preditos. O servidor pode lidar com tamanhos de contexto de 64x64, 32x32 ou 16x16, oferecendo flexibilidade para vários codecs. Este projeto permite a experimentação perfeita com diferentes arquiteturas de rede em codecs que compartilham estruturas de blocos semelhantes, exigindo poucas ou nenhuma modificação de código.

Para aproveitar o preditor do lado do servidor, o codec alvo sofre uma alteração na configuração. Um arquivo de configuração personalizado substitui um dos modos intra existentes pelo preditor proposto. Durante o processo de codificação, quando o codec encontra o modo designado, ele transmite os três blocos vizinhos ao redor do bloco de predição atual para o servidor via UDP. O codec então pausa sua predição regular e aguarda a resposta do servidor contendo o bloco predito.

A seleção do modo intra para substituição é crucial. Em vista de que o codificador EVC usa códigos de comprimento variável para sinalizar os modos de predição, é importante associar o modo mais provável de ser selecionado ao código de menor comprimento [3]. Portanto, os preditores propostos para os 3 tamanhos de bloco foram inseridos no lugar do modo DC (modo 0).

Esta abordagem de substituição oferece várias vantagens. Em primeiro lugar, gera um bitstream totalmente decodificável. Nenhuma informação adicional é necessária além dos dados codificados, supondo que o decodificador tenha acesso ao mesmo modelo de preditor que o codificador. Em segundo lugar, embora o preditor pudesse ser introduzido como um modo intra completamente novo, essa abordagem requer modificações nos esquemas de sinalização do codificador e do decodificador. O método de substituição evita essa complexidade adicional.

3 RESULTADOS

Está seção está dividida em duas subseções, experimentos preliminares e resultados finais. Na primeira abordaremos os experimentos realizados para encontrarmos a melhor configuração de treino e arquitetura da rede. Já na segunda discutiremos os resultados obtidos pela arquitetura final proposta.

3.1 Experimentos preliminares

A função das camadas de *up-sample* na rede servem apenas para aumentar o tamanho dos mapas de características, o que também pode ser feito com uma deconvolução com um tamanho de passo de 2. Avaliamos a utilização desta estratégia, bem como a substituição da última camada deconvolucional pela *up-sample*+conv como as suas camadas anteriores. Os resultados mostraram que ambas estratégias incorrem em perdas de BD-rate de aproximadamente 1%.

Entre as diferentes arquiteturas experimentadas, avaliamos a utilização de *highway connections* e uma arquitetura U-net com *skip connections* na tentativa de manter melhor a informação estrutural entre as camadas da rede. Esta abordagem produziu uma perda de cerca de 1,5%, apesar de ter a sobrecarga de complexidade das *skip connections*. Além disso, as ligações de saltos impossibilitam a descodificação do espaço latente numa compressão end to end, uma vez que as *skip connections* não podem ser transmitidas para o lado do descodificador. Desta maneira, determinamos que não era uma estratégia que merecesse uma investigação mais aprofundada. Por último, a utilização de *highway connections* proporcionou ganhos de menos de 1% para algumas sequências, mantendo a mesma eficiência em média, o que não justifica a sobrecarga de ligações.

Por fim, avaliamos a utilização de núcleos dilatados nas convoluções para explorar melhor as redundâncias angulares entre as vistas. No entanto, concluímos que o contexto é demasiado pequeno para ser proveitoso e as nossas experiências produziram uma perda de BD-rate de aproximadamente 4%.

3.2 Resultados Finais

Depois de todos os experimentos, a rede neural mais eficiente foi treinada separadamente para os 3 tamanhos de bloco presentes no EVC: 32x32, 16x16 e 8x8. O impacto de adicionar os preditores para

Tabela 1: BD-rates de todos preditores treinados individualmente e em conjunto.

Preditor / Sequência	32x32	16x16	8x8	32x32,16x16,8x8
	BD-rate	BD-rate	BD-rate	BD-rate
Ankylosaurus-&-Diplodocus-1	-22.17	-28.17	-24.98	-32.7
Bikes	-27.64	-36.97	-33.94	-41.88
Black-Fence	-6.59	-12.62	-10.32	-10.42
Ceiling-Light	-2.78	-13.15	-19.69	-21.89
Danger-de-Mort	-18.32	-30.38	-31.86	-33.67
Friends-1	-14.48	-22.4	-16.62	-28.13
Houses-&-Lake	-16.63	-22.3	-15.35	-22.15
Reeds	-10.04	-14.24	-11.36	-14.66
Rusty-Fence	-19.19	-30.81	-31.19	-36.06
Slab-&-Lake	-31.52	-40.52	-38.78	-40.16
Swans-2	-35.99	-39.43	-35.01	-46.34
Vespa	-30.47	-34.46	-31.31	-38.33
Média	**-19.65**	**-27.12**	**-25.03**	**-30.53**

cada um dos tamanhos de bloco e todos em conjunto podem ser observados na Tabela 1.

Ao inserir apenas o preditor para blocos de tamanho 32x32 se obteve um ganho médio na eficiência de codificação de -19.65%, já os preditores de tamanho 16x16 e 8x8 obtiveram eficiências de -27.12% e -25.03% respectivamente. Em vista que um bloco de tamanho 32x32 é constituído por 16 MIs (4x4 MIs de 8x8 pixels), conclui-se que o padrão entre as diferentes MIs do bloco podem divergir o suficientemente para que as redes não consigam prevêlas bem em todos os casos. Já ao utilizarmos tamanhos de blocos menores de 16x16 que contém 4MIs, obtivemos uma boa capacidade de predição e generalização. Já o preditor 8x8, embora possa parecer mais fácil predizer apenas 1 MI a partir de outras 3 MIs por ser um espaço de busca menor, a pequena vizinhança pode não apresentar informações o suficiente para a rede poder detectar corretamente o padrão do próximo bloco.

Para entendermos melhor o porque deste comportamento cabe analisarmos os casos em que cada preditor atua melhor. Inicialmente, o preditor de tamanho 32x32 embora já atinja ganhos significativos, supera apenas o preditor 8x8 nas sequências *Swans* 2 e *Houses & Lake*. Isso se deve ao fato de que ambas sequências possuem vastas áreas homogêneas que se beneficiam de tamanhos de blocos maiores. Em vista que blocos maiores providenciam bons ganhos de codificação e a exploração de áreas homogêneas é de grande importância no contexto de compressão de vídeos e imagens, ainda assume-se como importante a utilização de um preditor para este tamanho de bloco mesmo que com menor performance que os demais. Ainda, observe que as sequências *Black Fence* e *Ceiling Light* possuem melhoras de apenas -6.59% e -2.78% respectivamente. A natureza destas imagens possuem uma grande diferença de profundidade entre objetos, o que causa uma distância maior entre os pixels de uma MI e, consequentemente, as tornam mais difíceis de predizer e menos adequadas para um tamanho de bloco maior.

Desta maneira, ao avaliarmos a eficiência do preditor para blocos de tamanho 16x16 para estas sequências, podemos observar que a eficiência de codificação foi quase duplicada para a sequência *Black Fence* e aumentada em aproximadamente 5 vezes para a *Ceiling Light*. Por fim, ao utilizarmos um preditor para apenas os tamanhos de bloco 8x8 observou-se uma eficiência superior aos demais preditores nas sequências *Danger de Mort* e *Rusty Fence*, que por sua vez são LFs de grades com texturas complexas e constantes alternações entre objetos próximos (a grade) e distantes (plano de fundo). Este comportamento torna MIs próximas muito similares mas ao mesmo tempo MIs mais distantes se tornam muito distintas, fazendo com que a utilização destas para a predição não seja muito frutífera.

Para obtermos os benefícios de cada um dos 3 tamanhos de bloco concomitantemente, estes foram inseridos em conjunto no codificador. Vale ressaltar que, ao estarem sendo utilizados ao mesmo tempo, os preditores competem entre si, logo, seus ganhos não são somados, providenciando um aumento de BD-rate de -5,5% e permitindo que nossa proposta atinja um BD-rate total de -30.53%. Observe também que, assim como esperado, ao complementarem uns aos outros, os três preditores quando usados em conjunto possuem melhores eficiências de codificação para todas as sequências.

4 CONCLUSÃO

Este artigo propôs reinterpretar a predição de imagens *Light Field* densas como um problema de preenchimento de imagens e solucioná-lo utilizando redes neurais convolucionais. Para atingir este objetivo foram realizados diversos experimentos avaliando diferentes arquiteturas e estratégias de treino.

Os experimentos mostraram que utilizar camadas de *up-sample* seguida de convoluções regulares no decoder ao invés de convoluções transpostas com *stride* 2 se mostraram em torno de 1% mais eficientes. Ao analisarmos diferentes arquiteturas, a arquitetura U-net se mostrou em torno de 1.5% menos eficiente embora tenha um adicional de complexidade das *skip connections*. Já a arquitetura em *Highway* não apresentou ganhos o suficientes para justificar o custo extra das suas conexões. Por fim, selecionar recortes aleatórios do LF com rotações e um *learning rate* de $1 * 10^{-4}$ se mostrou a configuração mais eficiente para atingir altas eficiências de compressão e evitar o *overfit* das redes.

A rede proposta foi treinada e testada em três instâncias, uma para cada um dos 3 tamanhos de blocos do codificador EVC. Este codificador foi então utilizado para comprimir as sequências de teste utilizando como preditor intra as redes separadamente e em conjunto. O preditor para o tamanho de bloco de 16x16 se mostrou o mais eficiente atingindo um BD-rate de -27.12%. Já ao utilizar os preditores propostos para os 3 tamanhos de bloco atingiu-se um ganho de -30.53% de BD-rate.

Como trabalhos futuros pretende-se avaliar a eficiência dos preditores quando inseridos em codificadores de vídeo mais complexos como HEVC e VVC. Ainda pode-se aplicar técnicas de convoluções parciais como em [9, 12] ou técnicas de poda nas redes para diminuir seus tamanhos e aperfeiçoar suas performances. Outras métricas de Loss especializadas para codificação de vídeo também podem ser propostas e avaliadas como, por exemplo, SATD (Soma Absoluta das Transformadas das Diferenças). Por fim, ainda é possível propor um esquema de codificação E2E *(End to End*) para LFs densos.

ACKNOWLEDGMENTS

The authors of this work would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) Finance Code 001, CNPq, FAPERGS, Serrapilheira for funding this research.

REFERENCES

[1] Chih-Chieh Chen, Yi-Chang Lu, and Ming-Shing Su. 2010. Light field based digital refocusing using a DSLR camera with a pinhole array mask. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 754–757.

[2] Kiho Choi, Jianle Chen, Dmytro Rusanovskyy, Kwang-Pyo Choi, and Euee S Jang. 2020. An overview of the MPEG-5 essential video coding standard [standards in a nutshell]. *IEEE Signal Processing Magazine* 37, 3 (2020), 160–167.

[3] Thierry Dumas, Aline Roumy, and Christine Guillemot. 2019. Context-adaptive neural network-based prediction for image compression. *IEEE Transactions on Image Processing* 29 (2019), 679–693.

[4] Christopher Hahne and Amar Aggoun. 2021. PlenoptiCam v1.0: A Light-Field Imaging Framework. *IEEE Transactions on Image Processing* 30 (2021), 6757–6771. https://doi.org/10.1109/TIP.2021.3095671

[5] Junhui Hou, Jie Chen, and Lap-Pui Chau. 2018. Light field image compression based on bi-level view compensation with rate-distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 2 (2018), 517–530.

[6] Li Li, Zhu Li, Bin Li, Dong Liu, and Houqiang Li. 2017. Pseudo-sequence-based 2-D hierarchical coding structure for light-field image compression. *IEEE Journal of Selected Topics in Signal Processing* 11, 7 (2017), 1107–1119.

[7] Dong Liu, Lizhi Wang, Li Li, Zhiwei Xiong, Feng Wu, and Wenjun Zeng. 2016. Pseudo-sequence-based light field image compression. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–4.

[8] Faguo Liu, Qian Zhang, Tao Yan, Bin Wang, Ying Gao, Jiaqi Hou, and Feiniu Yuan. 2024. Light field image coding using a residual channel attention network–based view synthesis. *Data Technologies and Applications* (2024).

[9] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*. 85–100.

[10] Martin Rerabek and Touradj Ebrahimi. 2016. New light field image dataset. In *8th International Conference on Quality of Multimedia Experience (QoMEX)*.

[11] Jonatan Samuelsson, Kiho Choi, Jianle Chen, and Dmytro Rusanovskyy. 2019. Mpeg-5 evc. In *SMPTE 2019*. SMPTE, 1–11.

[12] Gabriele Spadaro, Roberto Iacoviello, Alessandra Mosca, Giuseppe Valenzise, and Attilio Fiandrotti. 2023. A Learnable EVC Intra Predictor Using Masked Convolutions. In *International Conference on Image Analysis and Processing*. Springer, 537–549.

[13] Soheib Takhtardeshir, Roger Olsson, Christine Guillemot, and Mårten Sjöström. 2024. A Deep Learning based Light Field Image Compression as Pseudo Video Sequences with Additional in-loop Filtering. *Electronic Imaging* 36 (2024), 1–6.

[14] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. 2013. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*. 673–680.

[15] Tingting Zhong, Xin Jin, Lingjun Li, and Qionghai Dai. 2019. Light field image compression using depth-based CNN in intra prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8564–8567.