Puzzle Similarity: A Perceptually-guided Cross-Reference Metric for Artifact Detection in 3D Scene Reconstructions

Supplementary Material

Summary In this supplementary material, we include extra details on the implementation of *PuzzleSim*, with exemplary code in Pytorch, as well as extra details on our choice of model backbone for the metric and extended qualitative and quantitative results with comparisons with all tested metrics. Additionally, we include extra information on our recursive automatic inpainting application and an ablation study.

1. Puzzle Similarity Implementation Details

Recall from Section 3 that we can compute the similarity map based on an outer product.

$$
\begin{aligned}
&\hat{\mathcal{F}}_\ell(\mathcal{I}^{1:N}) \in \mathbb{R}^{N\times H_\ell\times W_\ell\times C_\ell}\\
&\tilde{\mathcal{F}}_\ell(\mathcal{I}^{1:N}) = \text{flatten}\left(\hat{\mathcal{F}}_\ell(\mathcal{I}^{1:N})\right) \in \mathbb{R}^{NH_\ell W_\ell\times C_\ell}\\
&\mathcal{S}_\ell(\mathcal{I}) = \overbrace{\text{rowmax}\ \underbrace{\tilde{\mathcal{F}}_\ell(\mathcal{I}^{1:N}_{\text{ref}})\otimes\tilde{\mathcal{F}}_\ell(\mathcal{I})}_{\in\mathbb{R}^{NH_\ell W_\ell\times H_\ell W_\ell}}}^{\in\mathbb{R}^{H_\ell W_\ell}}
\end{aligned}
\tag{1}
$$

Computing this outer product naïvely would require substantial amounts of memory, as the resulting matrix before taking the maximum over rows has a dimensionality of (NHW, HW), thus NH^2W^2 elements. $N = 100$ reference images of size 128×128 would result in $26,843,545,600$ elements, requiring ≈ 100GB memory for 32-bit floating point numbers. We observed that this problem is similar to flash attention [2] and derived from it a memory-efficient implementation.

We leverage the fact that we are taking the maximum along rows of size (NHW). Knowing this, we can compute partial results by looping over either N, H, or W and aggregating the current maximum for each element, which reduces the memory footprint. At the same time, we take advantage of the GPU's cache hierarchy by looping in blocks. This gives an additional speedup without loss of generality. With this approach, we can cut the biggest matrix to (NbW, HW) in case we choose to aggregate along the height dimension while running over b-sized blocks with $b << H$. The final algorithm is detailed below.

```python
def puzzle_similarity(F, img)
    """
    F: base model
    img: image to test
    """
    layer_similarities = []
    ref_feats = compute_normalized_features(F, refs)
    features = compute_normalized_features(F, img)
    for layer in layers:
        refs = ref_feats[layer]
        img_ = feats[layer].squeeze()
        N, C, H, W = refs.shape

        candidates = []
        # factor over h, the dimension that you max over
        block_size = 4
        for h in range(0, H, block_size):
            sim = torch.einsum(
                'cHW,nchw->nHWhw',
                img_, refs[:, :, h:h+block_size, :]
            )
            c_WH = (
                sim
                # what was rows in sim is now last dimension
                .reshape(N, H * W, -1)
                .max(dim=-1)  # distribute max over ref.W
                .values  # get max values instead of indices
                .max(dim=0)  # distribute max over ref.N
                .values  # get max values instead of indices
            )
            candidates.append(c_WH)

        sim_map = (
            torch.stack(candidates, dim=0)
            .max(dim=0)  # distribute max over ref.H
            .values
            .view(H, W)  # reshape to spatial map
        )
        sim_map = upsample(sim_map * w[layer], img.shape)
        layer_similarities.append(sim_map)

    return sum(layer_similarities)
```

In our implementation, we use a block size of 4, which reduces the matrix size to (N4W, HW), reducing memory load to only $\approx$ 3.5GB. This approach enables us to compute *PuzzleSim* efficiently even on high-resolution images, given that computing time is primarily dominated by memory fetches in our metric.

To see the impact of our blockwise implementation, we compare its runtime with the naïve implementation. In Tab. 1, we show the results for different image sizes and number of reference images. After five warm-up steps, we measured the computation time in milliseconds, averaging over 200 runs. The $\pm$ indicates half the distance between the 0.05 and 0.95 quantiles. We observe that the blockwise implementation appears more stable and scales better. The experiment was performed on an NVIDIA GeForce GTX 3090 with 24GB of memory.

2. On the Choice of Backbone Model

In Tab. 2, we summarize the differences we considered when choosing our backbone. While *VGG* models achieve higher performance on the ImageNet benchmark [6], model size and computational complexity are problematic in efficiency terms for our metric. Furthermore, added model capacity and improved classification performance did not

Table 1. Comparison of the blockwise and naive implementation across different numbers of references and image sizes in ms.

Image Size	# References	PuzzleSim (blockwise)	PuzzleSim (naïve)
(240, 131)	25	$3.8_{\pm 1.95}$	$2.2_{\pm 1.36}$
(240, 131)	50	$6.0_{\pm 0.58}$	$26.3_{\pm 0.13}$
(240, 131)	75	$14.7_{\pm 0.16}$	$41.4_{\pm 0.17}$
(240, 131)	100	$20.1_{\pm 0.05}$	$57.9_{\pm 0.07}$
(480, 262)	25	$5.6_{\pm 0.29}$	$4.5_{\pm 0.11}$
(480, 262)	50	$15.3_{\pm 0.08}$	$12.7_{\pm 0.09}$
(480, 262)	75	$185.7_{\pm 0.00}$	$331.8_{\pm 0.00}$
(480, 262)	100	$299.2_{\pm 0.00}$	$499.1_{\pm 0.00}$
(960, 524)	25	$56.8_{\pm 0.07}$	$57.2_{\pm 0.74}$
(960, 524)	50	$321.3_{\pm 0.00}$	$727.2_{\pm 0.00}$
(960, 524)	75	$2653.3_{\pm 0.00}$	$5421.8_{\pm 0.00}$
(960, 524)	100	$5799.1_{\pm 0.00}$	$15353.6_{\pm 0.00}$
(1920, 1048)	25	$754.4_{\pm 0.00}$	Out-of-memory
(1920, 1048)	50	$1516.7_{\pm 0.00}$	Out-of-memory
(1920, 1048)	75	$31104.0_{\pm 0.00}$	Out-of-memory
(1920, 1048)	100	$69807.4_{\pm 0.00}$	Out-of-memory

seem to substantially improve the alignment of our model with human perception; rather hindering it. *AlexNet* and *SqueezeNet* offer much greater speed and lower memory consumption while also producing slightly preferable maps in our empirical evaluations.

Model	**#Parms**	**Acc.**	**Efficiency**	**Mem.**
VGG-19	144M	High	Low	High
VGG-16	138M	High	Low	High
AlexNet	60M	Mid	Mid	Mid
SqueezeNet	1.2M	Mid	High	Low

Table 2. Pre-trained models considered for our backbone. Accuracy refers to their relative performance on the ImageNet benchmark [6].

3. Dataset Collection Experiment Details

The experiment was run on a DELL U2718Q monitor with a consistent display setting across all participants, keeping a constant viewing distance of around 70 cm under controlled lighting conditions. We recruited 22 participants (10 male, 11 female, 1 undisclosed) with a mean age of 24, all possessing normal or corrected to normal visual acuity. All test subjects were compensated for their time.

4. Extended Metric Validation Results

In this section, we will provide more visual comparisons and further experiments comparing our metric against full-reference (FR) metrics. Typically, a clean reference image is not accessible, but the way we collected our dataset, we made sure to have one, regardless, so we can rank cross-reference also against hypothetical FR metric performances.

4.1. Extended Qualitative Results

We include extensive qualitative results on all tested full-, cross- and no-reference metrics in Figs. 1 and 2 with more examples. The color coding is always relative and scaled between the minimum and maximum score of each individual map. To facilitate comparison, we inverted the color coding of distance metrics, such that red indicates poor quality and blue indicates good quality. Among the FR metrics, we observe that the reference allows them to predict fine-grained inaccuracies but overall fails to prioritize artifacts incongruent to human observers. Providing a fine map resolution helps with identifying problematic primitives. The coarser resolutions of PIQE and CrossScore could result in challenges in this regard. CNNIQA maps

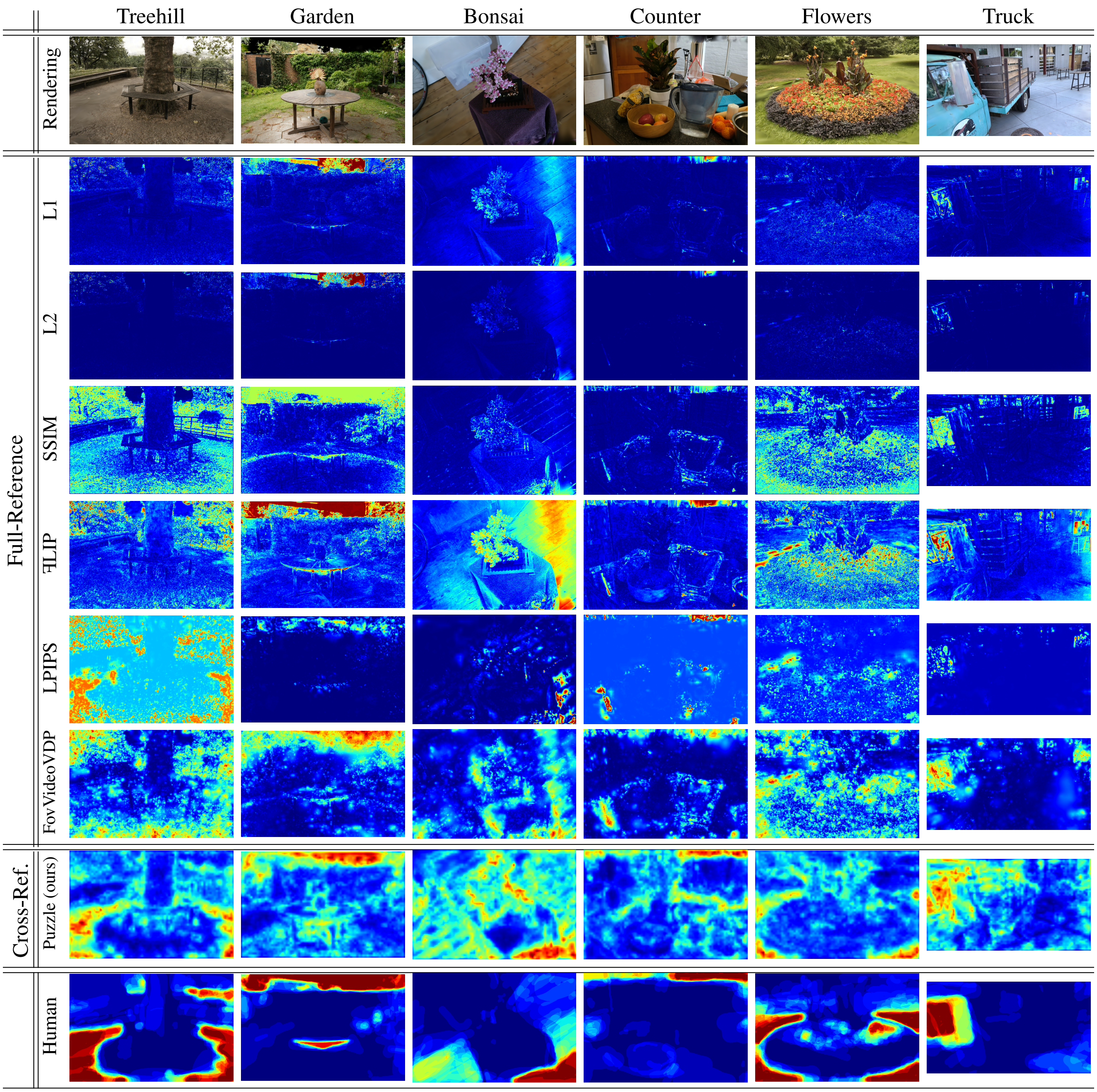

Figure 1. More elaborate comparison to all other FR metrics. For LPIPS we chose the VGG backbone as it is the most popular choice. Warm colors indicate artifacts or poor reconstruction quality.

deliver remarkable precision and resolution, however, they align poorly with human-perceived artifacts. PAL4VST seems to be an extremely conservative binary segmentation model, generally flagging the right areas but too cautiously.

4.2. Extended Quantitative Results

Adapting pooled metrics to produce spatial quality maps We selected commonly used image quality metrics (L1, L2, SSIM [7], LPIPS [11]). We adapt L1, L2, and SSIM by simply removing the pooling step. For LPIPS, after computing the distance in the embedding space, the metric already upsamples each feature map back to the original image resolution. We can then pool across maps but not along 2D image dimensions, allowing us to retain the map. We compare LPIPS with three base models: `VGG`, `AlexNet`, and `SqueezeNet`. We further com-

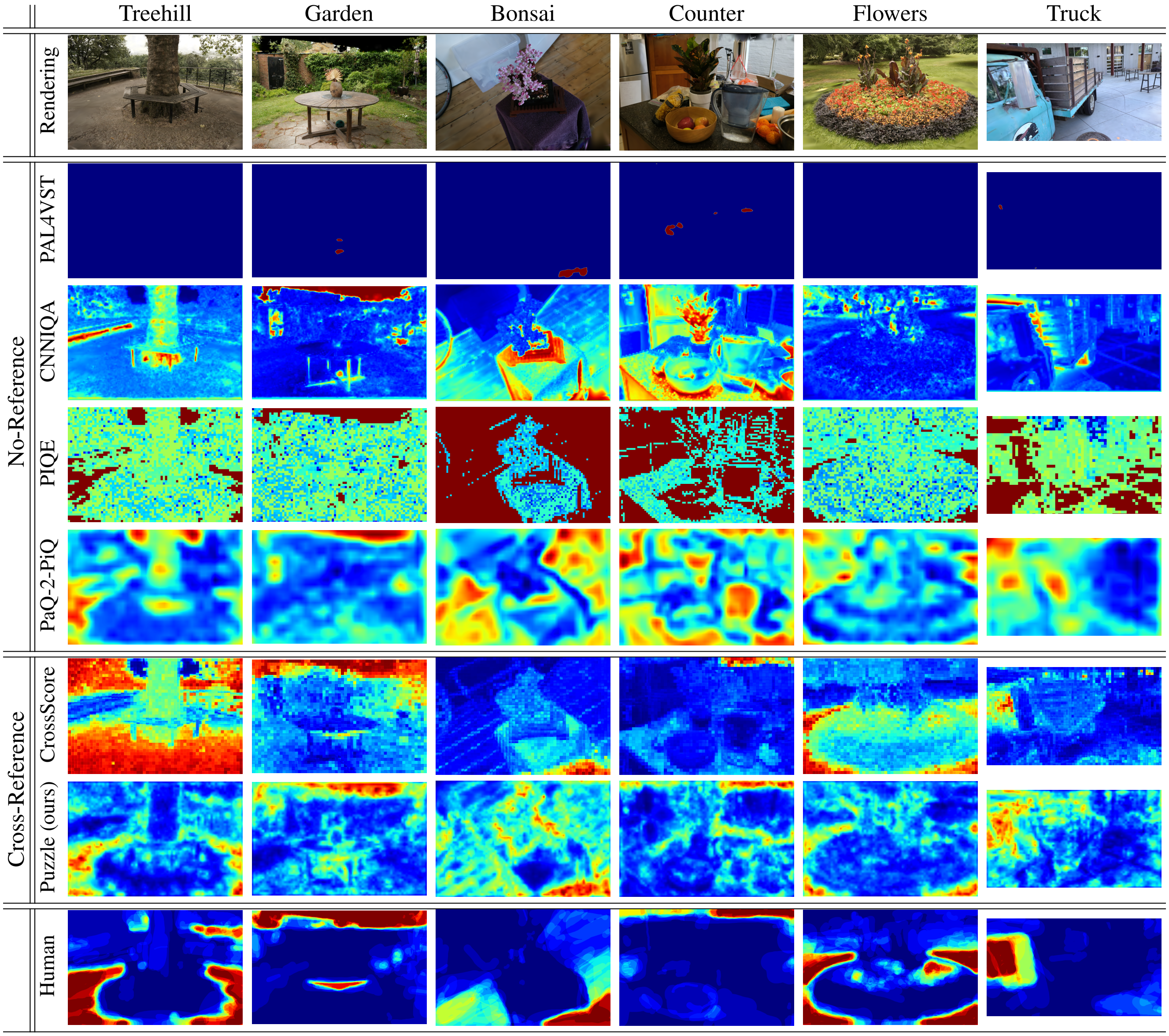

Figure 2. More elaborate comparison to all other no-reference and cross-reference metrics. Warm colors indicate artifacts or poor reconstruction quality.

pare with ꟻLIP [1], a commonly found metric in the field of physically-based rendering. It already produces spatial quality maps by default.

Comparison with Full-Reference Metrics Tab. 4 shows the average Pearson and Spearman correlation averaged across all datasets. On average, after SSIM, ꟻLIP and LPIPS achieve the highest correlation with human judgment within full-reference metrics when excluding VDPs that we cover separately in the next paragraph. Our method supersedes all full-reference metrics while maintaining a lower standard deviation compared to the best-performing metrics, showing consistently high performance over various types of artifacts. Tab. 3 depicts average correlation scores for each dataset. While our metric prevails on most datasets, ꟻLIP performs better on *garden*, which we attribute to the fact that most common artifacts are pitch-black regions (see Fig. 1) where the camera faces empty space. These kinds of artifacts are easy to spot, especially when a reference is available. Our hypothesis, as to why our metric outperforms FR metrics, is that they capture different quantities that, unlike our metric, do not necessarily align well with human judgment.

Comparison with Visual Difference Predictors The explicit model of low-level human vision in VDPs usually

Table 3. Pearson Correlation between full-reference Image Metrics, our cross-reference Metric and Human Perception per Dataset

		bicycle	bonsai	counter	drjohnson	flowers	garden	kitchen	playroom	stump	train	treehill	truck
Pearson	L1	0.203	0.318	0.420	0.316	0.067	0.607	0.594	0.603	0.134	0.211	0.142	0.335
	L2	0.193	0.307	0.418	0.332	0.080	0.573	0.597	0.619	0.130	0.183	0.144	0.323
	SSIM [7]	0.174	0.304	0.379	0.367	0.252	0.525	0.601	0.549	0.247	0.522	0.398	0.503
	ꟻLIP [1]	0.267	0.405	0.451	0.366	0.192	0.719	0.681	0.649	0.238	0.181	0.139	0.400
	LPIPS (vgg) [11]	0.197	0.413	0.389	0.240	0.243	0.546	0.602	0.531	0.232	0.364	0.392	0.394
	LPIPS (alex) [11]	0.169	0.276	0.264	0.233	0.346	0.394	0.536	0.502	0.154	0.271	0.251	0.340
	LPIPS (squeeze) [11]	0.238	0.402	0.290	0.289	0.103	0.561	0.512	0.504	0.216	0.353	0.240	0.401
	FovVideoVDP [4]	0.369	0.457	0.562	0.467	0.306	0.661	0.824	0.753	0.370	0.583	0.337	0.593
	PuzzleSim (ours)	0.594	0.565	0.618	0.461	0.609	0.675	0.768	0.636	0.505	0.642	0.717	0.593
Spearman	L1	0.150	0.238	0.232	0.209	0.050	0.390	0.346	0.501	0.118	0.150	0.190	0.287
	L2	0.155	0.250	0.235	0.218	0.053	0.401	0.352	0.513	0.123	0.146	0.191	0.291
	SSIM [7]	0.168	0.120	0.253	0.415	0.248	0.540	0.248	0.496	0.279	0.471	0.402	0.433
	ꟻLIP [1]	0.204	0.336	0.252	0.272	0.201	0.466	0.437	0.565	0.210	0.114	0.214	0.363
	LPIPS (vgg) [11]	0.199	0.256	0.312	0.340	0.184	0.443	0.488	0.560	0.236	0.427	0.407	0.341
	LPIPS (alex) [11]	0.195	0.182	0.218	0.184	0.159	0.366	0.343	0.363	0.175	0.256	0.297	0.190
	LPIPS (squeeze) [11]	0.207	0.382	0.317	0.351	0.048	0.386	0.497	0.587	0.210	0.389	0.321	0.311
	FovVideoVDP [4]	0.339	0.312	0.455	0.421	0.291	0.534	0.646	0.735	0.381	0.554	0.362	0.473
	PuzzleSim (ours)	0.468	0.393	0.382	0.499	0.428	0.428	0.658	0.601	0.307	0.540	0.548	0.440

Table 4. Aggregated correlation between all Image Metrics and Human Perception with mean and standard deviation across all datasets.

	Metric	Pearson ↑	Spearman ↑
Full-Reference	L1	$0.329_{\pm 0.226}$	$0.239_{\pm 0.176}$
	L2	$0.325_{\pm 0.221}$	$0.244_{\pm 0.178}$
	SSIM [7]	$0.402_{\pm 0.164}$	$0.339_{\pm 0.172}$
	ꟻLIP [1]	$0.391_{\pm 0.236}$	$0.303_{\pm 0.180}$
	LPIPS (vgg) [11]	$0.378_{\pm 0.154}$	$0.349_{\pm 0.146}$
	LPIPS (alex) [11]	$0.311_{\pm 0.145}$	$0.244_{\pm 0.111}$
	LPIPS (squeeze) [11]	$0.342_{\pm 0.156}$	$0.334_{\pm 0.168}$
	FovVideoVDP [4]	$0.523_{\pm 0.192}$	$0.459_{\pm 0.166}$
NR	PAL4VST [10]	$0.078_{\pm 0.112}$	$0.062_{\pm 0.085}$
	CNNIQA [3]	$0.144_{\pm 0.247}$	$0.130_{\pm 0.253}$
	PIQE [5]	$0.292_{\pm 0.222}$	$0.268_{\pm 0.221}$
	PaQ-2-PiQ [9]	$0.402_{\pm 0.178}$	$0.349_{\pm 0.225}$
CR	CrossScore [8]	$0.510_{\pm 0.204}$	$0.378_{\pm 0.209}$
	PuzzleSim (ours)	$0.615_{\pm 0.120}$	$0.474_{\pm 0.137}$

produces strong correlations between the metric and human assessment. We compare against the state-of-the-art FovVideoVDP [4]. While the metric can take into account higher-order perceptual cues like motion and eccentricity, we disabled them as we are 1) analyzing single static frames and 2) are not assuming specific viewing conditions such as display size or distance to the screen. Puzzle Similarity not only matches but often surpasses FovVideoVDP, particularly in texture-rich scenes like *treehill*, *stump*, and *flowers*. This is remarkable since FovVideoVDP is an FR metric, while we do not require a direct reference.

5. Automatic Recursive Inpainting

In this section, we will give more details about the inpainting application formulation and present an ablation study to showcase that our metric is the best candidate for our inpainting application.

5.1. Formulation Details

In the following, we provide extended details on the mathematical framework of the inpainting method. Upon sampling threshold candidates $\tau_{1:N}$ we threshold the initial similarity map $\mathcal{S}$ with each candidate to obtain the binary mask M_i as shown in Eq. 7. We inpaint the current image with each candidate mask and assess their quality with our *PuzzleSim* metric:

$$\begin{aligned} \hat{\mathcal{I}}_i &= \text{Inpaint}(\mathcal{I}, M_i) \\ \hat{\mathcal{S}}_i &= \text{PuzzleSim}(\hat{\mathcal{I}}_i) \end{aligned} \tag{2}$$

Table 5. Quantitative results of our inpainting application with different metrics producing the inpainting masks. The inpainted image is compared to the same image inpainted using human-generated masks. The dashed line separates no-reference (top) and cross-reference (bottom) metrics.

Metric	PSNR ↑	SSIM ↑	LPIPS ↓
PAL4VST [10]	15.03	0.436	0.580
CNNIQA [3]	23.07	0.850	0.151
PIQE [5]	21.85	0.742	0.250
PaQ-2-PiQ [9]	24.49	0.879	0.120
CrossScore [8]	24.58	0.863	0.118
PuzzleSim (ours)	25.16	0.902	0.102

To determine the candidate quality, we compute the average change in the similarity δ_i of the inpainted regions:

$$|M_i| = \sum_{h=1}^{H_{\mathcal{I}}} \sum_{w=1}^{W_{\mathcal{I}}} M_i^{(h,w)}$$

$$\delta_i = \frac{1}{\lambda_i |M_i|} \underbrace{\sum_{h=1}^{H_{\mathcal{I}}} \sum_{w=1}^{W_{\mathcal{I}}} (\hat{\mathcal{S}}_i^{(h,w)} - \mathcal{S}^{(h,w)}) M_i^{(h,w)}}_{\text{Similiarity improvement of the inpainted area}} \tag{3}$$

$$\lambda_i = |M_i|^{\frac{1}{p}}$$

where $|M_i|$ indicates the number of inpainted pixels and λ_i is a regularization term penalizing bigger masks with strength p that we empirically chose to be 4. Once the initial threshold τ_* is found, we iteratively find a new threshold in the interval $\tau_* \pm \alpha^{-1}$ std$(\hat{\mathcal{S}}_*)$, where α is a hyperparameter that we set to 10 for all examples. By including the standard deviation, we dynamically adapt to the distribution of similarity scores. As the scores become uniform, the interval becomes narrower, facilitating convergence.

5.2. Ablation Study

To showcase that our metric is the best choice for our inpainting application, we run an ablation study by testing all other cross-reference and no-reference metrics as a drop-in replacement for our metric. We leverage our human masks to produce an upper bound for inpainting results by thresholding the human annotations to obtain an optimal mask for inpainting. The optimal inpaintings are compared to all the automatically generated results from our application through PSNR, SSIM, and LPIPS. We repeated this procedure for our entire dataset, and the averaged results obtained are shown in Tab. 5, where our metric outperforms all competitors on all criteria. With CrossScore following up second, we observe that cross-reference metrics have the upper hand on this task. This was expected since cross-reference metrics leverage more information compared to no-reference metrics that have to judge images at face value.

References

[1] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. FLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2):1–23, 2020. 4, 5

[2] Tri Dao, View Profile, Daniel Y. Fu, View Profile, Stefano Ermon, View Profile, Atri Rudra, View Profile, Christopher Ré, and View Profile. Flashattention. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 16344–16359, 2022. 1

[3] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional Neural Networks for No-Reference Image Quality Assessment. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, Columbus, OH, USA, 2014. IEEE. 5, 6

[4] Rafał K. Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. FovVideoVDP: a visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics*, 40(4):1–19, 2021. 5

[5] Venkatanath N, Praneeth D, Maruthi Chandrasekhar Bh, Sumohana S. Channappayya, and Swarup S. Medasani. Blind image quality evaluation using perception based features. In *2015 Twenty First National Conference on Communications (NCC)*, pages 1–6, 2015. 5, 6

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2015. arXiv:1409.0575 [cs]. 1, 2

[7] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. Conference Name: IEEE Transactions on Image Processing. 3, 5

[8] Zirui Wang, Wenjing Bian, and Victor Adrian Prisacariu. CrossScore: Towards Multi-View Image Evaluation and Scoring. In *Computer Vision – ECCV 2024*, pages 492–510, Cham, 2025. Springer Nature Switzerland. 5, 6

[9] Zhenqiang Ying, Haoran Niu, Praful Gupta, Dhruv Mahajan, Deepti Ghadiyaram, and Alan Bovik. From Patches to Pictures (PaQ-2-PiQ): Mapping the Perceptual Space of Picture Quality. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3572–3582, Seattle, WA, USA, 2020. IEEE. 5, 6

[10] Lingzhi Zhang, Zhengjie Xu, Connelly Barnes, Yuqian Zhou, Qing Liu, He Zhang, Sohrab Amirghodsi, Zhe Lin, Eli Shechtman, and Jianbo Shi. Perceptual Artifacts Localization for Image Synthesis Tasks. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7545–7556, Paris, France, 2023. IEEE. 5, 6

[11] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. pages 586–595, 2018. 3, 5