Feature Refinement to Improve High Resolution Image Inpainting

Prakhar Kulshreshtha
Geomagical Labs, Inc.
prakhar@geomagical.com

Brian Pugh
Geomagical Labs, Inc.
bpugh@geomagical.com

Salma Jiddi
Geomagical Labs, Inc.
salma@geomagical.com

Abstract

In this paper, we address the problem of degradation in inpainting quality of neural networks operating at high resolutions. Inpainting networks are often unable to generate globally coherent structures at resolutions higher than their training set. This is partially attributed to the receptive field remaining static, despite an increase in image resolution. Although downscaling the image prior to inpainting produces coherent structure, it inherently lacks detail present at higher resolutions. To get the best of both worlds, we optimize the intermediate featuremaps of a network by minimizing a multiscale consistency loss at inference. This runtime optimization improves the inpainting results and establishes a new state-of-the-art for high resolution inpainting. Code is available at: https://github.com/geomagical/lama-with-refiner/tree/refinement.

1. Introduction

Image inpainting is the task of filling missing pixels or regions in an image [19]. This task finds application in image restoration, image editing, Augmented Reality, and Diminished Reality [12] [4]. Several methods have been proposed to solve this problem. [6, 17] inpaint missing regions using gradient guided diffusion of colors from neighboring pixels. [5, 7] sample patches from unmasked areas of the image that satisfy well-defined similarity criteria. Patch-based solutions are widely adopted in image editing tools like Gimp [2] and Photoshop [1].

Existing approaches often struggle with global consistency when the masked-region is large enough to encompass multiple texture or semantic regions [20, 23]. Conditional Generative Adversarial Networks (cGAN) have been developed to address this issue via an intermediate global representation [11, 14, 19, 20, 25]. Even with cGAN, a large receptive field is critical for high inpainting performance [16]. Various techniques have been proposed to increase the effective receptive field, such as Fourier convolutions [16], diffusion models [15], contextual transformations [21], and transformers [8, 10].

In this work, we focus on improving the inpainting quality of existing networks at high resolutions. Increasing the operating image-size proportionally decreases the available local context to the network when inpainting a region, which causes incoherent structures and blurry textures [21]. To solve this problem, we propose a novel coarse-to-fine iterative refinement approach that optimizes featuremaps via a multiscale loss. By using lower resolution predictions as guidance, the refinement process produces detailed high resolution inpainting results while maintaining the color and structure from low resolution predictions (Fig. 1). No additional training of the inpainting network is required; only featuremaps are refined during inference [13, 18].

Figure 1. Results from our multiscale refinement. Left: input image. Center: inpainting with Big-LaMa [16]. Right: inpainting with Big-LaMa + our refinement.

2. Multiscale Feature Refinement

Our multiscale feature refinement follows a coarse-to-fine approach to iteratively add more detail to an inpainting
prediction (Fig. 2).

Figure 2. Iterative refinement of the inpaint prediction at a single scale by optimizing the latent featuremap $z$.

An image-pyramid of the input RGB image and inpainting mask is constructed to be used as network inputs at multiple inference resolutions. The smallest scale is approximately equal to the network’s training resolution. We assume that the network will perform best at training resolution and use it as the basis for all inpainting structure guidance.

The model is split into “front” and “rear” sections, similar to [18]. Typically, these correspond to the encoder and decoder portions of the network, respectively. At the lowest resolution, we perform a single forward pass through the entire inpainting model to get an initial inpainting prediction. For each subsequent scale, we run a single forward pass through the “front” to generate an initial featuremap $z$. Multiple featuremaps (e.g. from skip-connections) can be jointly optimized, but are not investigated in this paper.

The rear part of the network processes $z$ to produce an inpaint prediction. The prediction is then downscaled to match the resolution of the previous scale’s result. Downscaling involves applying a Gaussian filter, followed by bilinear interpolation. The Gaussian filter removes high frequency components and prevents aliasing during downscaling. An L1 loss is computed between the masked inpainted regions and is minimized by updating $z$ via backpropagation. This will optimize $z$ to produce a higher resolution prediction that has similar characteristics to the previous scale.

Figure 3 shows an example of how refinement improves the quality of predictions. At high resolutions (1024px), our method has significant improvements regarding structure-completion when compared to Big-LaMa [16]. Our refinement also contains more details compared to the upscaled low resolution prediction (512px).

Figure 3. From left to right, first row: (i) input image, (ii) inpainting at size 512, (iii) inpainting at size 1024 (iv) inpainting at 1024 with refinement. Second row: zoomed-in corresponding inpainted areas.

Python pseudocode for our multiscale refinement is described in Algorithm 1. The `multiscale_inpaint` function generates the image pyramid and iterates over the multiple scales, while `predict_and_refine` produces a per-scale refined prediction.

```python
def predict_and_refine(image, mask, inpainted_low_res, model, lr=0.001, n_iters=15):
    z = model.front.forward(image, mask)
    optimizer = Adam([z], lr)
    for _ in range(n_iters):
        optimizer.zero_grad()
        inpainted = model.rear.forward(z)
        inpainted_downscaled = downscale(inpainted)
        loss = l1_over_masked_region(inpainted_downscaled, inpainted_low_res, mask)
        loss.backward()
        optimizer.step()  # Updates z
    inpainted = f_rear.forward(z)
    return inpainted

def multiscale_inpaint(image, mask, model, smallest_scale=512):
    images, masks = build_pyramid(image, mask, smallest_scale)
    n_scales = len(images)
    inpainted = model.forward(images[0], masks[0])
    for i in range(1, n_scales):
        image, mask = images[i], masks[i]
        inpainted_low_res = inpainted
        inpainted = predict_and_refine(image, mask, inpainted_low_res, model)
    return inpainted
```

Algorithm 1. PyTorch pseudocode of multiscale refinement.
| Method                      | Thin Brush | Medium Brush | Thick Brush | Time per Image |
|-----------------------------|------------|--------------|-------------|----------------|
|                            | FID↓ | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ | LPIPS↓ | Seconds |
| AOTGAN [21]                 | 17.387 0.133 | 34.667 0.144 | 54.015 0.184 | 0.43 |
| LatentDiffusion [15]       | 18.505 0.141 | 31.445 0.149 | 38.743 0.172 | 31.56 |
| MAT [10]                   | 16.284 0.137 | 27.829 0.135 | 38.120 0.157 | 0.56 |
| ZITS [8]                   | 15.696 0.125 | 23.500 0.121 | 31.777 0.140 | 4.14 |
| LaMa-Fourier [16]          | 14.780 0.124 | 22.584 0.120 | 29.351 0.140 | 0.16 |
| Big-LaMa [16]              | **13.143 0.114** | **21.169 0.116** | **29.022 0.140** | **0.26** |
| Big-LaMa+refinement (ours) | 13.193 **0.112** | **19.864 0.115** | **26.401 0.135** | **4.56** |

Table 1. Performance comparison against recent inpainting approaches on 1k 1024x1024 size images sampled from [3]. Inference time per image was calculated on a single NVIDIA RTX A5000 GPU.

3. Experiments

In our experiments, we apply iterative multiscale refinement to Big-LaMa [16]. A downscaling factor of 2 is used to build the image pyramid. The output featuremap from the downscaler portion of Big-LaMa (see $f_{\text{front}}$ in Fig. 2) will be optimized. This featuremap was chosen based on the observation that featuremaps farther from the prediction layer have a larger receptive field and are able to influence more of the output [18].

At each scale, we perform 15 refinement iterations using Adam optimizer with a learning rate of 0.002. To prevent the network from optimizing against low resolution infill in thin regions where the network is already performing well, we erode the mask with a 15 pixel circular kernel prior to applying L1 loss to the inpainted regions.

4. Results

Inpainting networks are typically benchmarked on Places2 [24]. However, this dataset does not have high resolution images for evaluation purposes. Instead, we will use images from the Unsplash-Lite Dataset, which contains 25k high resolution nature-themed photos [3]. We randomly sampled 1000 images to evaluate on (linked here).

Each image is resized and cropped to 1024x1024, and a set of masks is generated with thin, medium, and thick brush strokes, using the methodology described in [16]. These different mask-types are evaluated separately to observe the effect the width of the mask has on image inpaint quality. In accordance with recent works [8, 16], performance is evaluated using FID scores [9] and LPIPS [22].

We compare against other methods in Table 1 and Figures 4, 5. Our method outperforms reported state-of-the-art inpainting networks for medium and thick masks, while performing similarly to Big-LaMa [16] for thin masks. The thin-mask performance is similar because there is sufficient surrounding context to complete the structure.

Although our refinement produces higher scoring results, it also takes significantly longer to process an image. For each image, multiple forward and backward passes are required. This increase inference-time proportionally with number of scales and optimization steps. Refinement also increases memory usage because gradients are required at runtime, consequently reducing the maximum resolution that can fit in GPU memory. Our approach produces infills with stronger global consistency and sharper textures. Additional results are available via this linked video.

5. Conclusion

We proposed a multiscale refinement technique to improve the inpainting performance of neural networks on images at resolution higher than the native training resolution. This refinement is network agnostic, and requires no additional model retraining. Our results indicate that this technique significantly outperforms other state-of-the-art approaches at high resolution inpainting.
Figure 5. Comparison on more examples of size 1024x1024 from Unsplash dataset [3]. In each row, from left to right, we show: input image, output using AOTGAN [21], Latent-Diffusion(LD) [15], MAT [10], ZITS [8], LaMa-Fourier(LaMa-F) [16], Big-LaMa(LaMa-B) [16], Big-LaMa with refinement(LaMa-BR - ours). Last image in each row is the ground-truth (GT).

References

[1] Adobe photoshop. 1
[2] Gimp, gnu image manipulation program. 1
[3] Unsplash dataset. 2020. 3, 4
[4] Joe Bardi. What is diminished reality? r&d engineer ken
moser, phd, explains, Aug 2016. 1

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 1

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000. 1

[7] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. Proceedings., volume 2, pages II–II. IEEE, 2003. 1

[8] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3, 4

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3

[10] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3, 4

[11] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 1

[12] Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects., June 2017. 1

[13] Calvin Murdock, MingFang Chang, and Simon Lucey. Deep component analysis via alternating direction neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 820–836, 2018. 1

[14] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 1

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2021. 1, 3, 4

[16] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 1, 2, 3, 4

[17] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 1

[18] Tsun-Hsuan Wang, Fu-En Wang, Juan-Ting Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Plug-and-play: Improve depth estimation via sparse data propagation. *arXiv preprint arXiv:1812.08350*, 2018. 1, 2, 3

[19] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 1

[20] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 1

[21] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 1, 3, 4

[22] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 3

[23] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 1

[24] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 3

[25] Manyu Zhu, Dongliang He, Xin Li, Chao Li, Fu Li, Xiao Liu, Errui Ding, and Zhaoxiang Zhang. Image inpainting by end-to-end cascaded refinement with mask awareness. *IEEE Transactions on Image Processing*, 30:4855–4866, 2021. 1