Accelerating Modern Image Processing Operators with Bilateral Guided Upsampling

Elim Yi Lam Kwan (ylk25)

I. Introduction

Image processing algorithms face significant performance challenges when running on high-resolution images produced by modern cameras. Additionally, photos in recent time are taken and processed entirely on mobile devices, which have limited computational power. This has resulted in growing interest in research on edge persevering low complexity image processing techniques. A common approach seen in literature is to apply the operation on a downsampled version of the original image, then upsampled the result to obtain a high-resolution output which preserves quality while simultaneously minimising computation. Chen et al. (2016) presented the concept of Bilateral Guided Upsampling [1] (BGU) for modelling image processing operators: given low-resolution reference input/output pairs, the relationship between the input and output pixels' intensity can be approximated with local curves, and these curves are later fitted to the high-resolution image to generate the final output. BGU has proven to be quite useful and has been used as the base model for other research [2]. Moreover, the original paper explored the effectiveness of their framework on Local Laplacian, Style Transfer, Unsharp mark, Colourization, Portrait Style transfer, L_0 Smoothing, Matting and Dehazing operators. Despite this, it would be useful to provide more empirical results to showcase the transferability of BGU for other scale-invariant operators.

Therefore, we would like to present two new applications of BGU - Tone Mapping and Gradient Domain Enhancement, which are widely-use modern days operators. Our results suggested that the output of BGU is able to successfully resemble the effect of tested operators, with Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) of above 0.95 and 32.38 dB, as well as an only 4-6% decrease in Tone Mapped Image Quality Index (TMQI) relative to the ground truth.

Finally, we would also like to propose an alternative to enforcing smoothness. The current approach of adding constraints to the least-square problem has heavily penalised the timing performance. Our experiment results suggested that our simple approach is able to increase the SSIM by 2.94% relative to the bare-bone system with no smoothness term, which is much closer to the SSIM generated by the full system. We provided a step-by-step demonstration in Jupyter Notebook [1].

To summarise, our main contributions in this paper are:

- Demonstrated two new applications of BGU - Tone Mapping and Gradient Domain Enhancement.
- Proposed an alternative method for enforcing smoothness.
- Justified the choice of the interpolation schemes used.

[1]Link: https://github.com/elimkwan/Bilateral-Guided-Upsampling

II. Related work

Bilateral Filter aims to smooth out the image without blurring edges though it comes with a high computational cost. Given its wide applications, the area of real-time edge-preserving smoothing filter has been actively studied. Joint bilateral upsampling [3] made use of the additional information available from the original high-resolution input image; they first applied a spatial filter to the downsampled image, followed by a similar range filter on the high-resolution input. However, this only generates piece-wise smooth solutions. Alternatively, He et al. (2013) [4] proposed to fit a 2D arrays of affine functions on image patches to model the relationship between the input and the reference output. BGU combined both approaches while adding additional smoothness constraints and adopted a 3D array instead to overcome the piece-wise smooth limitations and to increase expressiveness. This allows the framework to represent a wider range of operators instead of just blurring operations. As aforementioned, we would like to complement the list and further verify BGU applicability on tone mapping and gradient domain enhancement filters.

With the increasing utility of High Dynamic Range (HDR) imaging [5, 6, 7, 8], there is a need to develop tools to display HDR images on everyday devices. Dynamic range refers to the difference between the darkest and lightest tones in an image. Typical displays are encoded in 8 bit, hence, there are only up to 255 levels of intensities, considerably less than human vision. Moreover, modern image sensors can capture a image with wide dynamic range and the process of translating those HDR images to the 255 range is referred to as tone mapping. Tone mapping filter was indeed one of the verified operators in the joint bilateral upsampling paper [3]. Hence, in this project, we would like to investigate the effectiveness of BGU on tone mapping. Our hypothesis is that BGU will be able to cope with the operator well due to their similar origins.

In addition, we would also like to experiment whether BGU will work well with operators that focus on transformation in the perceptual domain - such as image enhancement in the gradient domain. Such concepts have been widely studied in the context of image blending [9], image matting [10], color-to-gray mapping [11] and even tone mapping [7].

Moreover, in the full model, spatial smoothness was enforced by adding the first-order Taylor series of the operator to the least-square system, which imposed significant computational load. Hence, a fast approximate method [1] was developed. Rather than treating the whole system as a single least-squares problem, the fast approximation fits the affine model by solving a smaller subset of the least-squares problem at individual grid cell. To enforce smoothness, a smoothing filter was applied to the parameters stored in each grid cell. Our alternative methods also adopted blurring, but instead of applying it to the parameters, we applied it to the low-resolution reference output that we are trying to match our low-resolution input with. We will evaluate its effectiveness in Section IV-C.

Additionally, recent trends in accelerating image processing involve the use of machine learning. Instead of just treating the general image processing operators as a black box, machine learning techniques can be applied on specific parts of the pipeline as well, potentially resulting in better performance as we incorporate our domain knowledge into the solution. Hence, one of the most popular extensions of the BGU framework is Deep BGU [2] proposed by Gharbi et al. (2017). They developed a supervised learning approach to learn the parameters for the affine models. Given the time and GPU resources limitations of the project, we did not apply any machine learning frameworks on our model and verifying the effectiveness of deep BGU on the additional operators proposed is listed as one of the future work.

III. Methodology

In this section, the details of the BGU will be explained. Our proposed alternative optimisation techniques and operators' algorithms will also be introduced.

A. Bilateral Guided Upsampling and Proposed Optimisation Strategies

The work on BGU is based on the intuition that the I/O pixel intensities for any image operators over a sufficiently small (e.g., $1x1$) image patches can be represented by a line. Visualisation of it can be found from Figure 1 (e.g., the input vs output pixels' intensity of the Local Laplacian operator). However, as the patch size increases, it contains a larger range of intensities. As a result, the constant model no longer holds, but it can still be considered as piece-wise linear. This relationship can be approximated by considering the next term in the Taylor series, which forms our local affine model. However, the first-order Taylor expansion will be invalid when edges are present in the patch. Hence, affine models are also computed based on intensity ranges. A bilateral grid, with luminance as the z-axis, has been used to stores these local models. For well-behaved operators, these local curves should vary smoothly, since input pixels of similar intensity should have similar output intensity. Therefore, spatial smoothness can be enforced by minimising the partial derivatives of models' coefficients. The full optimisation problem can be formulated as:

$$\arg\min_{\gamma} \int (\beta - AS^T\gamma)^2 + (\lambda_x D_x \gamma)^2 + (\lambda_x D_x \gamma)^2 + (\lambda_x D_x \gamma)^2 dxdydz \qquad (1)$$

where β, A, S^T, γ representing the reference output images, input images, the slice matrix and the target parameters respectively.

However, the additional derivatives terms significantly increased the system's timing complexity. A simple alternative is to apply a Gaussian filter on the reference low-resolution image, i.e.,

$$\arg\min_{\gamma} \int (\text{Guassian}(\beta) - AS^T\gamma)^2 \qquad (2)$$

It is understood that the original paper also provides a fast approximation method, which divided the global optimisation objective into smaller overlapping least-square problems. Although their approach may generate better performance, our method requires minimal changes to the original pipeline, hence will be ideal for users who would like to have a system with a smaller footprint or can act as a simple add-on for existing implementations.

The affine models are computed based on the low-resolution I/O pairs and could be later applied on the full resolution input through interpolation. In the original publication, trilinear interpolation method was adopted, though we are also interested in the effectiveness of utilising other interpolation techniques of similar complexity. Hence, we have also tested this using nearest-neighbour interpolation.

B. Bilateral Guided Upsampling Implementation Details

A straightforward measure of applying bilateral grid on colour I/O images will be to adopt a 5D grid and store a 3×4 affine matrix for each (x, y, r, g, b) cell. However, this would result in a paucity of data for fitting individual affine matrix. Therefore, we have adopted a hybrid model similar to the original paper. Globally, the grid is 3D with axis $(x, y, luminance)$. Locally, each 3×4 affine model is responsible for modelling the colour transformation between $[A_r, A_g, A_b, 1]$ and $[\beta_r, \beta_g, \beta_b]$. The grid used in all the experiment is $20 \times 15 \times 10$.

Additionally, all the implementations have been vectorised, allowing it to be accelerated with parallel programming in the future. For the global least square problem, it is too complex to be solved by $sparse.linalg.lsqr$, instead we have adopted $SparseQr$ library, which is the equivalent of $x = A\backslash b$ in MATLAB with CUDA acceleration. This helps to put us on par with the original MATLAB implementation [12].

C. Tone-Mapping and Gradient Domain Operators

We have verified the 4 most common tone mapping algorithms which are available with the OpenCV toolbox: Drago [5], Reinhard [6], Durand [8], Mantiuk [7]. Global operators, such as Drago and Reinhard, compress contrasts based on globally derived quantities. Drago scaled images in a logarithmic domain with adaptive base depending on each pixel's radiance, while Reinhard utilised photoreceptor adaptation. For local operators, Durand decomposed images into base layer and detail layer with bilateral filter and compressed contrast only on the base layers. Lastly, Mantiuk utilised gradients of the layers in a Gaussian pyramid, and performed the transformations in the HSV domain. It will be intriguing to investigate how these tone mapping operators' features affect the BGU approximations.

Moreover, as utilised by some tone mapping operators, we would like to evaluate the effectiveness of BGU in approximating gradient domain transformation as well. Human visual system is more sensitive to pixel differences than absolute pixel values, performing image enhancement in the gradient domain and reconstruct the image from it often minimise the generation of artefacts. We have utilised the previous implementation from Practical 1 with piece-wise linear transformation equation. It induced a brightening effect and its mathematical form is:

$$y = \begin{cases} 1.458x \ , \ x < 0.425 \\ 0.661x + 0.339 \ , \ \text{else} \end{cases} \tag{3}$$

IV. Evaluation

Overall, we demonstrated that BGU is capable of accelerating tone mapping and gradient domain operators, with SSIM above 0.95 and PSNR ranging from 32.38 to 41.53 dB.

A. Performance Matrices

In all the experiments, we have verified the PSNR and SSIM of the full resolution BGU output. PSNR indicated the ratio between the power of the maximum signal and noise, while SSIM provides additional information for perceived similarity by taking texture into account. A good result should consist of both high PSNR and SSIM.

Moreover, to draw a fair comparison among BGU performance on various tone mapping algorithms, TMQI [13] has been adopted for quantifying the quality of Low Dynamic Range (LDR) images subjectively, which is a standard matrix used in tone mapping supervised learning problem [14]. S, N and Q in TMQI metric represents the Structural Fidelity, the Statistical Naturalness, and a combination of both in the reconstructed LDR. We first manually calibrate the parameters of the tone mapping operators to produce LDR images of similar TMQI (with $Q \approx 0.21$) using the same high resolution HDR. Then, the tone mapping operators are used to generate the low-resolution I/O image pairs. Finally, given the high-resolution HDR, and the low-resolution image pairs, BGU is performed. The TMQI of the resultant BGU LDR output is compared with the ground truth TMQI (i.e., the TMQI of the original high-resolution pairs). This experiment flow allows us to compare the capability of BGU in perceiving information in HDR images.

Furthermore, the timing performance was not used as performance metric because we have yet to accelerate our designs with parallel computing. As references, the original non accelerated implementation takes about 15 seconds, while their fast approximation recorded a 2 ms and 10 ms on desktop and mobile GPUs respectively. We believed our vectorised Python implementation with similar structure will record similar results and implementing it will be left as future work. Lastly, images used in the Local Laplacian and Tone Mapping Operators experiments are from the original research [1] and sIBL [15].

B. Performance of Bilateral Guided Upsampling on Tone Mapping and Gradient Enhancement Operators

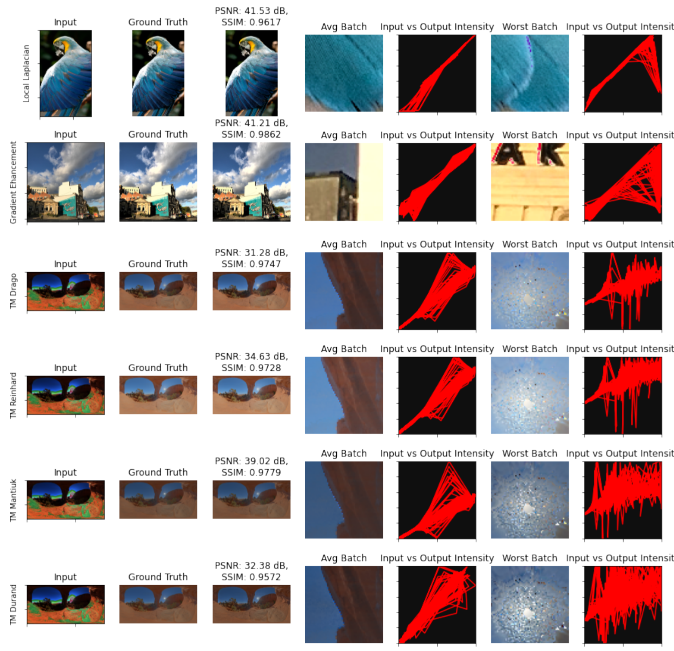

Fig. 1. Evaluations of Bilateral Guided Upsampleing (BGU) on Local Laplacian, Tone Mapping (TM) and Gradient Enhancement Operators. The input(x) vs output(y) intensity in the average-case batch and the worst-case batch of the output have been shown. Batch size is 64×64 pixels. Full resolution results can be found on the GitHub Repository.

		Ground Truth TMQI	BGU Output TMQI	Percentage Change
TM Drago	Q	0.2117	0.2019	-4.63%
	S	0.0127	0.0108	-14.41%
	N (10^{-13})	4.29	6.67	55.46%
TM Reinhard	Q	0.2111	0.2022	-4.24
	S	0.0125	0.0109	-13.27
	N (10^{-13})	6.86	9.83	43.29
TM Mantiuk	Q	0.2105	0.1992	-5.37
	S	0.0124	0.0104	-16.57
	N (10^{-13})	1.02	1.58	54.72
TM Durand	Q	0.2104	0.1978	-5.98
	S	0.0124	0.0101	-18.33
	N (10^{-13})	4.16	6.03	45.19

Fig. 2. Compare the Tone Mapped Image Quality Index (TMQI) of the low dynamic range outputs from Bilateral Guided Upsampling(BGU) to the ground truth. ($Q = overall$; $S = StructuralFidelity$; $N = StatisticalNaturalness$)

Overall, the BGU output has shown promising outcome in modelling all the tested operators, with an SSIM of above 0.95 and PSNR ranging from 32.38 to 41.53 dB. This performance gain can be explained as the operators matches with our affine model hypothesis. In Figure 1, for average batches with edges, the input and output pixel intensity shows a linear relationship. For the worst-case batch, similar observations can be drawn although they have been severely impacted by noise.

BGU approximation works well with Local Laplacian and Gradient Enhancement operators, as their input-output relationship is approximately a curve. Although a more fine-grain intensity range may be required to eliminate all the artefacts as shown in the worst-case batch because those failures occur when some pixels with identical input intensities were being mapped to a wide range of output intensities.

BGU also generates reasonable result when performing as a tone mapping operator, with only $4-6\%$ decrease in TMQI relative to the ground truth image pairs. The outputs show quite a lot of artefacts around the brightest spot, which may be resulted from the colour transfer problem. Since colours often fall outside the colour gamut when contrast is compressed, well-designed tone mapping operators have taken care of it, while our BGU approach as a general model was not designed with these features in mind.

Thirdly, from Figure 2, we observed that BGU provides a better approximation for global tone mapping operators Drago and Reinhard. This may be due to the fact that quantity derived locally is more complex to be approximated with a coarse model. Besides, BGU significantly improved the Statistical Naturalness compared to the ground truth pairs, in which we credited it to the additional smoothness terms.

Finally, in terms of limitations, as mentioned in the original publication [1], the framework does not scale well with resolution-dependent operators, as we require it to produce useful information about the filter when applied at a reduced resolution images. Unfortunately, we did not manage to alleviate this inherent property.

C. Performance of Proposed Alternative in Enforcing Smoothness

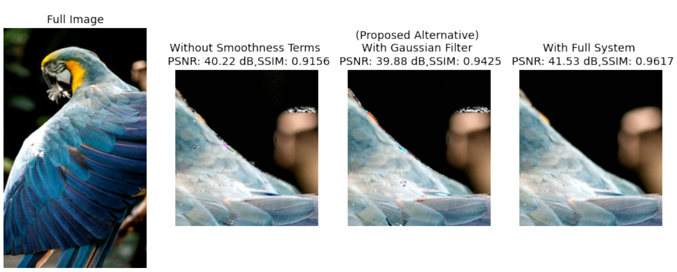

Fig. 3. Impact of Smoothness Terms

As shown in Figure 3, our proposed Gaussian Blur method brings a 2.94% improvements in SSIM relative to the bare-bone system that did not enforce smoothness (i.e. $\arg\min_\gamma \int \beta - AS^T\gamma)^2$), which brings us much closer to the SSIM achieved by the full system.

We acknowledge that the fast approximation method proposed by the original paper has achieved better output. Their methods only induced a 1.99 percentage decrease in PSNR relative to the full system, while ours caused a 3.97 percentage drop. However, our method is much simpler in design and required minimal changes to the original pipeline, hence it is useful given the smaller footprint and as a quick improvement to existing implementations.

D. Performance Change for Adopting other Interpolation Schemes

	Linear	**Nearest-neighbour**
PSNR (dB)	41.53	32.72
SSIM	0.9617	0.8746

Fig. 4. Impact of Interpolation Schemes used in upsampling

Figure 4 shows that the original trilinear interpolation scheme provides much better result compared to nearest-neighbour interpolation. This is unsurprising considering nearest-neighbour interpolation on continuous data often generates blocky results. Through this experiment, we have verified that using trilinear interpolation yield better result.

V. Conclusion

In conclusions, we confirmed that BGU is capable of recreating tone mapping and gradient enhancement effects on full resolution input faithfully with a pair of low-resolution guided images. Our implementation achieved an SSIM and PSNR of above 0.95 and 32.38dB and maintained a reasonable TMQI (only 4 to 6% percentage drop) when using BGU to approximate tone-mapped LDR images. We also gained useful insights that BGU is more suited for global tone mapping operators instead of local ones and justified the use of trilinear interpolation over nearest-neighbours.

Also, our proposed alternatives for enforcing smoothness shows that applying Gaussian filter on the low resolution guided output image results in smoothing effect as well, which helps improve the SSIM by 2.94% relative to a system that did not enforce smoothness. Although it did not outperform the fast approximate methods proposed in the original paper, it shows adequate performance and can be used as a simple add-on for existing implementations due to its simplicity.

In the future, we would like to verify the timing performance of the system by accelerating the implementation with custom hardware or GPU. Moreover, adopting machine learning techniques for a different part of the model pipeline will also be one of the future directions, for instance, one can approximate the affine model with supervised learning approach [2].

References

[1] Jiawen Chen et al. "Bilateral guided upsampling". In: *ACM Transactions on Graphics* 35 (Nov. 2016), pp. 1–8. DOI: 10.1145/2980179.2982423.

[2] Michaël Gharbi et al. "Deep Bilateral Learning for Real-Time Image Enhancement". In: *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.3073592. URL: https://doi.org/10.1145/3072959.3073592.

[3] Johannes Kopf et al. "Joint Bilateral Upsampling". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)* 26.3 (2007), to appear.

[4] Kaiming He, Jian Sun, and Xiaoou Tang. "Guided Image Filtering". In: *IEEE transactions on pattern analysis and machine intelligence* 35 (June 2013), pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213.

[5] Frederic Drago et al. “Adaptive Logarithmic Mapping For Displaying High Contrast Scenes”. In: *EUROGRAPHICS 2003 (EUROGRAPHICS-03) : the European Association for Computer Graphics, 24th Annual Conference, Blackwell, 419-426 (2003)* 22 (Sept. 2003). DOI: 10.1111/1467-8659.00689.

[6] Erik Reinhard and Kate Devlin. “Dynamic Range Reduction Inspired by Photoreceptor Physiology”. In: *IEEE transactions on visualization and computer graphics* 11 (Jan. 2005), pp. 13–24. DOI: 10.1109/TVCG.2005.9.

[7] Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. “A Perceptual Framework for Contrast Processing of High Dynamic Range Images”. In: *ACM Trans. Appl. Percept.* 3.3 (July 2006), pp. 286–308. ISSN: 1544-3558. DOI: 10.1145/1166087.1166095. URL: https://doi.org/10.1145/1166087.1166095.

[8] Frédo Durand and Julie Dorsey. “Fast Bilateral Filtering for the Display of High - dynamic - range Images”. In: *ACM Trans. Graph.* 21 (July 2002), pp. 257–266. DOI: 10.1145/566654.566574.

[9] Patrick Pérez, Michel Gangnet, and Andrew Blake. “Poisson Image Editing”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 313–318. ISSN: 0730-0301. DOI: 10.1145/882262.882269. URL: https://doi.org/10.1145/882262.882269.

[10] Jian Sun et al. “Poisson matting”. In: *ACM Trans. Graph.* 23 (Aug. 2004), pp. 315–321. DOI: 10.1145/1015706.1015721.

[11] Amy Gooch et al. “Color2Gray: Salience-preserving color removal”. In: *ACM Trans. Graph.* 24 (July 2005), pp. 634–639. DOI: 10.1145/1186822.1073241.

[12] Jiawen Chen. *Bilateral Guided Upsampling*. URL: https://github.com/google/bgu (visited on 01/16/2020).

[13] H. Yeganeh and Z. Wang. “Objective Quality Assessment of Tone-Mapped Images”. In: *IEEE Transactions on Image Processing* 22.2 (2013), pp. 657–667. DOI: 10.1109/TIP.2012.2221725.

[14] Aakanksha Rana et al. “Deep Tone Mapping Operator for High Dynamic Range Images”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 1285–1298. ISSN: 1941-0042. DOI: 10.1109/tip.2019.2936649. URL: http://dx.doi.org/10.1109/TIP.2019.2936649.

[15] Unknown. *sIBL Archive HDRI Datset (Creative Commons Attribution-Noncommercial-Share Alike 3.0 License))*. URL: http://www.hdrlabs.com/sibl/archive.html (visited on 01/14/2021).