EDGE-AWARE IMAGE PROCESSING WITH A LAPLACIAN PYRAMID BY USING CASCADE PIECEWISE LINEAR PROCESSING

[1] *Chien-Ming Lu* (*呂建明*), [1] *Sheng-Jie Yang* (*楊勝傑*), [1] *Chiou-Shann Fuh* (*傅楸善*)

Graduate Institute of Computer Science and Information Engineering,
National Taiwan University,Taipei, Taiwan
E-mail: {d01922027, d01922035, fuh}@csie.ntu.edu.tw

ABSTRACT

The Laplacian pyramid is popular for decomposing images into multiple scales and is widely used for image analysis [15]. However, because it is constructed with spatially invariant Gaussian kernels, the Laplacian pyramid is widely believed as being unable to represent edges well and as being ill-suited for edge-aware operations such as edge-preserving smoothing and tone mapping. Paris, Hasinoff, and Kautz successfully take Laplaician pyramid to process [15]. But they still take advantage of exponential function. We propose another linear approximation and use cascade piecewise linear for more correct approximation. We can see the exponential curve is easily implemented and has small error. In the result, we also show that it has better quality by using BRISQUE score [14].

Keywords *Local Laplacian Filter; Edge-aware Image Processing; Laplacian Pyramid; Cascade piecewise linear processing;*

1. INTRODUCTION

Laplacian pyramids have been applied to many applications such as harmonization [18], texture synthesis [11], compression [5]. But these techniques are commonly regarded as a poor choice for edge preserving. Smooth Gaussian kernels applied on pyramids are considered almost opposite to edge discontinuities. Further, the decimation of the levels is successive reduction by factor 2 of the resolution. It is often criticized for introducing aliasing artifacts, leading some researchers, e.g., [13], to recommend its omission. These arguments are often cited as a motivation for more sophisticated schemes such as edge-preserving optimization [3, 7], edge-aware wavelets [10], neighborhood filters [12, 20], and anisotropic diffusion [1, 16].

Laplacian pyramids can be implemented by easily resizing image and some other methods relying on more sophisticated techniques. For example, optimization-based methods [7, 8, 17], minimize a spatially inhomogeneous energy and build for each new image by decidated basis functions [9, 10, 19], and bilateral filters takes advantage of a spatially varying kernel [20].These additional levels of sophistication also often come with practical shortcomings. The neighborhood filters easily over-sharpen edges [4]. Optimization-based methods do not scale well since the solution has exponential complexity. Anisotropic diffusion is hard to set the parameters due to the iterative process. Some of these shortcomings can be improved by post-processing, for example, bilateral filter edge can be smoothed [2, 6, 12]. It takes additional computation and parameter setting.

Paris, Hasinoff, and Kautz offer one state-of-the-art edge-aware filters achieved with statndard Laplacian pyramids without post-processing following additional computation and parameter setting [15]. They use the i^{α} curve to control the texture part of the image. The schematic diagram is shown in red curve of Fig. 1. Because it is a nonlinear equation and not easily and directly implemented on the chip, we hope to easily implement on hardware in real time. Finally we propose the cascade piecewise linear method.

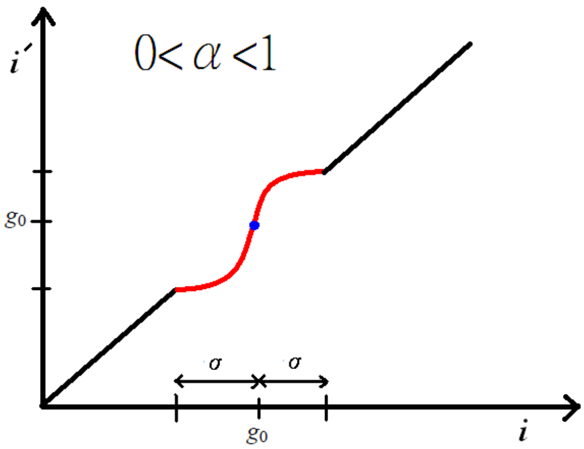


Fig. 1 The concept of local detail enhancement.

2. PROPOSED METHOD

The nonlinear power equation can be approximately by piecewise linear point. If we want to approximate more correctly at lower normalized σ_r, it needs many points to approximate. If we just want to use fewer points to achieve the approximation, we write the equation as follows.

Piecewise linear point $A=[r(a_1), r(a_2), ... r(a_N)]$ where $r(i)$ is remapping value; $a_1, a_2, ... a_N$ are the normalized σ_r distributed evenly from 0 to 1

$$r(i)=r(a_{n-1})*r(a_1)+\frac{r(a_n)-r(a_{n-1})}{size(A)}$$

$$\text{if}\,(r(a_{n-1})<i<r(a_n))$$

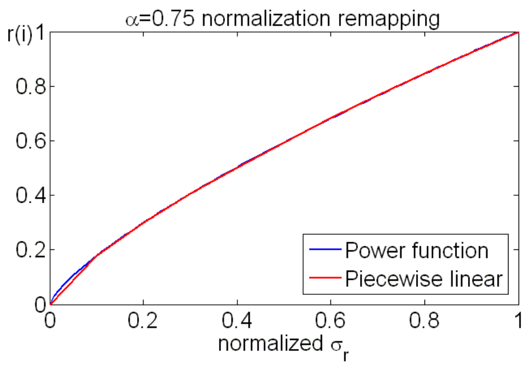
The simulation results are in the Fig. 2. They show the effect when α equals 0.75, 0.5, and 0.25. In Fig. 2(c), we can see that large difference exists from 0 to 0.1. Besides Fig 2(c), Fig. 2(b) also exists significant difference. Thus we propose the cascade piecewise linear to approach the nonlinear equation.

We use 10 piecewise linear points for the fist level, but it is not enough in Fig.2. We add the same 10 points piecewise linear to cascade in the range bwtween 0 to 0.1. The simulation result of the approximation is shown in Fig. 3.

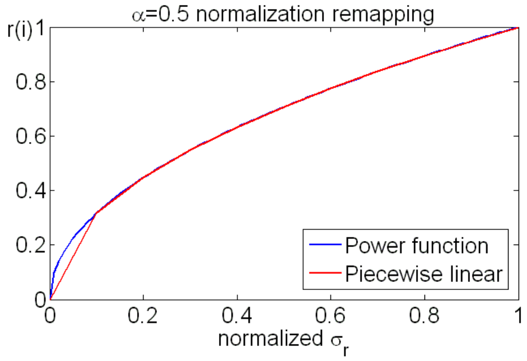
We can see the 10 times points in the range from 0 to 0.1. They almost has the same value on the curve of nonlinear equation i^{α}.

We only prepare 10 piecewise linear points for every α parameter. If we want to have 20 different α parameters for selecting, the hardware system just only stores 10x20 coefficients and saves memory space and can create 20 different nonlinear equations.

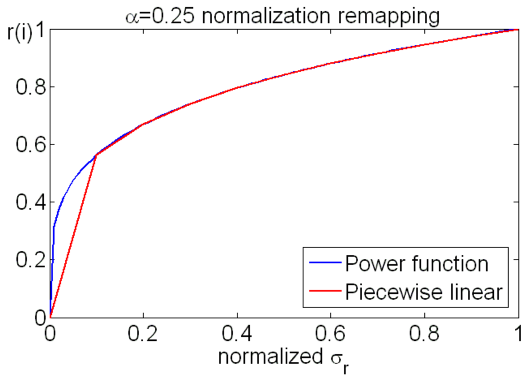
We want to know how large the error is after applying our proposed method between 0 to 0.1 of σ_r. We analyze the error distributed between 0 to 0.1 of σ_r by showing Figs. 4 to 6. It is easy to see the error in Part (a) of Figs. 4 to 6. After the comparison, our proposed method shows almost 0 in Part (a) of Figs. 4 to 6.
When we use small scale to see the detailed error distributed, it is really very small at about -17 order.



(a)

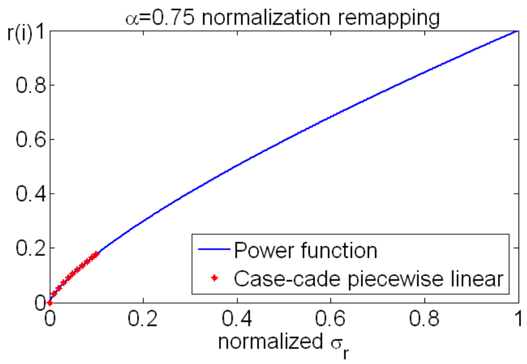


(b)

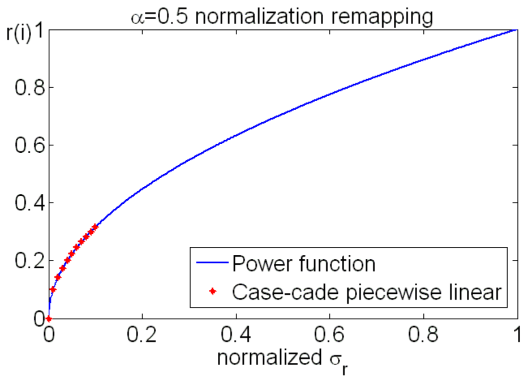


(c)

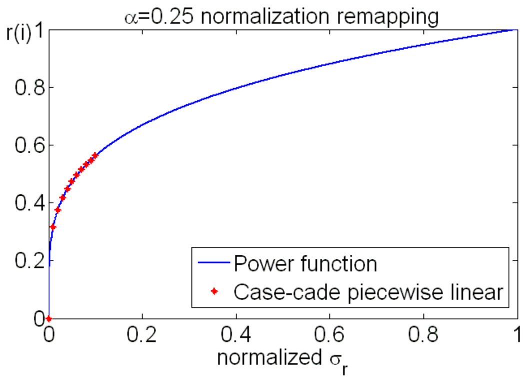
Fig. 2 Normalization remapping comparisons between power function and piecewise linear using different parameters. (a) α=0.75, (b) α=0.5, (c) α=0.25.



(a)

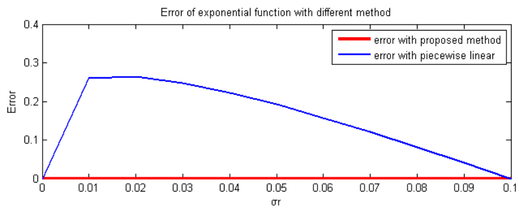


(b)

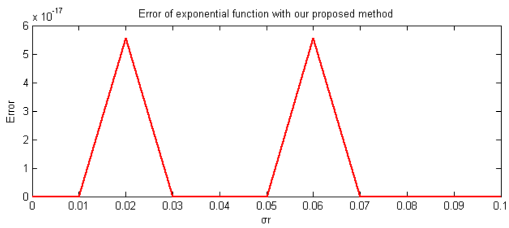


(c)

Fig. 3 Normalization remapping comparisons between power function and cascade piecewise linear using different parameters. (a) α=0.75, (b) α=0.5, (c) α=0.25

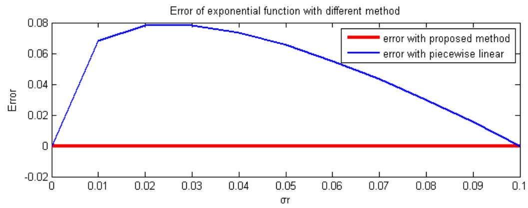


(a)

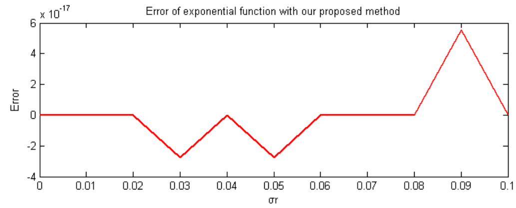


(b)

Fig. 4: (a) Error with piecewise linear, α= 0.25
(b) Error with proposed method, α= 0.25

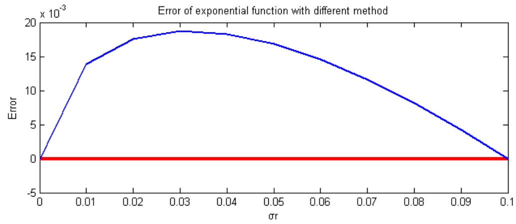


(a)

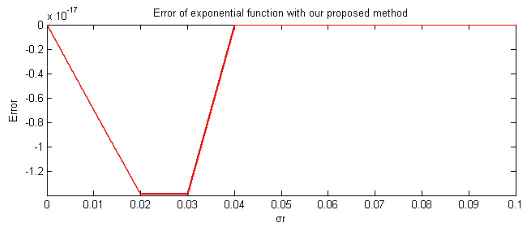


(b)

Fig. 5: (a) Error with piecewise linear, α= 0.5
(b) Error with proposed method, α= 0.5



(a)



(c)

Fig. 6: (a) Error with piecewise linear, α= 0.75
(b) Error with proposed method, α= 0.75

3. EXPERIMENT RESULT

After using our proposed method to replace nonlinear equation and use only two pyramid levels, the result images part (b) of Figs. 7 to 10. are almost the same with the result of Paris et al.'s work [15], whose result image are the part (a) of Figs. 7 to 10. We apply one objective score which is BRISQUE score [14]. It does not need another image for reference. It can give the quality result according to the image self quality. When we run 2 different σ and α values, the BRISQUE score shows the four results are very similar with Paris et al.s's work [15]. Thus it achieves our goal.
Besides, the cascade piecewise linear has better quality than the nonlinear equation because BRISQUE score means better when it is lower. We can see the four

results in Table 1. First, our proposed method gets the BRISQUE score value -2.157870 when σ is 0.25 and α is 0.25 , it is less than -1.9524 the output of paper [15]. Second, our proposed method gets the BRISQUE score value -1.86854 when σ is 0.25 andd α is 0.5, it is also less than -1.8523 the output of paper [15]. Third, our proposed method gets the BRISQUE score value 7.767620 when σ is 0.5 and α is 0.25, it is again less than 7.917620 the output of paper [15]. But finally, the output of paper [15] is better than our proposed when σ is 0.5 and α is 0.5. The score are 0.207623 and 0.248562 respectively for local Laplacian filter and our proposed method. Among four results, our three results are better than the output of paper [15].

If we use pure piecewise linear method with σ_r divided by the same distance method, it needs 100 point for achieving the same quality on one exponential function. If we want to use two segment piecewise linear method and choose σ_r equals 0.1 for demarcation point. It needs 10 point between 0 to 0.1 and 9 ponts between 0.1 to 1.0. This method needs totally 19 points. We propose cascade piecewise linear method and it just only need 10 points to achieve almost the same quality with exponential function. The memory requirement of the three methods are made in Table 2.

4. CONCLUSION

We propose a cascade piecewise linear method for local Laplacian filter suitable on real-time hardware system and save memory space than pure piecewise linear method and two segment piecewise linear method in Table 2. If we want to implement the local Laplacian filter, we need about 20 different exponential functions for user different settings. The comparison of memory size of different methods are shown in Table 3. Besides the less memory is needed and our proposed can have a little better quality than the nonlinear equation according to BRISQUE score. BRISQUE score is developed according to large image data base. When the image adds some suitable noise inside, it will get worse score. The difference of our proposed method and nonlinear equation is at 0~0.01. In this regions nonlinear equation rises faster than our method. Thus the smooth region by using nonlinear equation where σ_r<0.01 will become more noisy than our proposed method. In this paper, our proposed method can offer less memory requirement and keep good quality for detail enhancement.

(a)

(b)

Fig. 7: (a) Local laplacian filter σ = 0.2, α= 0.25
(b) Proposed method σ = 0.2, α= 0.25

(a)

(b)

Fig. 8: (a) Local laplacian filter σ = 0.2, α= 0.5
(b) Proposed method σ = 0.2, α= 0.5

(a)

(b)

Fig. 9: (a) Local laplacian filter σ = 0.5, α= 0.25
(b) Proposed method σ = 0.5, α= 0.25

(a)

(b)

Fig. 10: (a) Local laplacian filter σ = 0.5, α= 0.5
(b) Proposed method σ = 0.5, α= 0.5

Table 1: Brisque score.

	Local Laplacian Filter	*Proposed Method*
Flower σ=0.2 α=0.25	-1.9524	-2.157870
Flower σ=0.2 α=0.5	-1.8523	-1.86854
Flower σ=0.5 α=0.25	7.917620	7.767620
Flower σ=0.5 α=0.5	0.207623	0.248562

Table 2: Memory score for different exponential functions.

Method	*Pure Piecewise Linear Method*	*Two segment piecewise linear*	*Proposed Method*
Memory size	100 words coefficient	19 words coefficient	10 words coefficient

Table 2: Memory score for 20 different exponential functions.

Method	*Pure Piecewise Linear Method*	*Two segment piecewise linear*	*Proposed Method*
Memory size	2000 words coefficient	380 words coefficient	200 words coefficient

ACKNOWLEDGEMENT

This research was supported by the National Science Council of Taiwan, R.O.C., under Grants NSC 98-2221-E-002 -150 -MY3 and NSC 101-2221-E-002 -194, by Himax Technology, Winstar Technology, Test Research, and Lite-on.

REFERENCES

[1] G. Aubert and P. Kornprobst, "Mathematical Problems in Image Processing: Partial Differential Equations and The Calculus of Variations," *Applied Mathematical Sciences,* Springer, Munich, Vol. 147, 2002.

[2] S. Bae, S. Paris, and F. Durand, "Two-Scale Tone Management for Photographic Look," *ACM Transactionsion Graphics (Proc. SIGGRAPH),* Vol. 25, Issue 3, pp. 637–645. 2006.

[3] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A Gradient-Domain Optimization Framework for Image and Video Filtering," *ACM Transactions on Graphics,* Vol. 29, Issue 2, Article 10, 2010.

[4] A. Buades, B. Coll, and J.-M. MOREL, "The Staircasing Effect in Neighborhood Filters and Its Solution," *IEEE Transactions on Image Processing,* Vol. 15, Issue 6, pp. 1499–1505, 2006.

[5] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communication,* Vol. 31, Issue 4, pp. 532–540, 1983.

[6] F. Durand and J. Dorsey, "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 21, Issue 3, pp. 257-266, 2002

[7] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 27, Issue 3, Article 67, 2008.

[8] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 21, Issue 3, pp. 249-256, 2002.

[9] R. Fattal, "Edge-Avoiding Wavelets and Their Applications," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 28, Issue 3, Article 22, 2009.

[10] R. Fattal, R. Carroll, and M. Agrawala, "Edgebased Image Coarsening," *ACM Transactions on Graphics,* Vol. 29, Issue 1, Article 6, 2009.

[11] D. J. Heeger and J. R. Bergen, "Pyramid-Based Texture Analysis/Synthesis," *Proceedings of the ACM*

SIGGRAPH Conference, New York, NY, pp. 229-238, 1995.

[12] M. Kass and J. Solomon, "Smoothed Local Histogram Filters," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 29, Issue 3, Article 100, 2010.

[13] Y. Li, L. Sharan, and E. H. Adelson, "Compressing and Companding High Dynamic Range Images with Subband Architectures," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 24, Issue 3, pp. 836-844, 2005.

[14] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," *IEEE Transactions on Image Processing*, Vol. 21, Issue 12, pp. 4695–4708, 2012.

[15] S. Paris, S. W. Hasinoff, and J. Kautz, "Local Laplacian Filters: Edge-Aware Image Processing with a Laplacian Pyramid," *ACM Transactions on Graphics* (*Proceedings of SIGGRAPH,* Vol. 30, Issue 4, Article 68, 2011.

[16] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 12, Issue 7, pp. 629–639, 1990.

[17] K. Subr, C. Soler, and F. Durand, "Edge-Preserving Multiscale Image Decomposition Based on Local Extrema," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia),* Vol. 28, Issue 5, Article 147, 2009.

[18] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-Scale Image Harmonization," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 29, No. 4, Article 125, 2010.

[19] R. Szeliski, "Locally Adapted Hierarchical Basis Preconditioning," *ACM Transactions on Graphics (Proc. SIGGRAPH),* Vol. 25, Issue 3, pp. 1135-1143, 2006.

[20] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proceedings of IEEE International Conference on Computer Vision,* , Bombay, pp. 839–846. 1998.