https://doi.org/10.2352/lim.2024.5.1.26

Even Simpler Tone Curves

James Bennett and Graham Finlayson. School of Computing Sciences, University of East Anglia, Norwich, UK.

Abstract

Abstractly, a tone curve can be thought of as an increasing function of input brightness which, when applied to an image, results in a rendered output that is ready for display and is preferred. However, the shape of the tone curve is not arbitrary. Curves that are too steep or too shallow (which concomitantly result in too much or too little contrast) are not preferred. Thus, tone curve generation algorithms often constrain the shape of the tone curves they generate. Recently, it was argued that tone curves should - as well as being limited in their slopes - only have one or zero inflexion points.

In this paper, we propose that this inflexion-point requirement should be strengthened further. Indeed, the single inflexion-point-only constraint still admits curves with sharp changes in slope (which are sometimes the culprits of banding artefacts in images). Thus, we develop a novel optimisation framework which additionally ensures sharp changes in the tone curves are smoothed out (technically, mollified). Our even simpler tone curves are shown to render most real images to be visually similar to those rendered without the constraints. Experiments validate our method.

Introduction

Tone curves are a powerful tool for manipulating tones in images and occur at multiple stages in the process of taking a raw image to a final rendered output. Tone curves that map from real-world scene radiance to pixel values - thereby compressing the dynamic range - are sometimes called camera response functions [1]. Their inverses are used in radiometric calibration [2]. Tone curves are also used later in the camera processing pipeline [3, 4] where a power law curve is used to gamma encode the image suitable for display. In image (often contrast) enhancement, the rendered images are tone-mapped within the same dynamic range to create more pleasing renditions [5].

Tone curves must be strictly increasing (thus one-to-one) functions which avoids problems like tone inversions. Other constraints are required if we wish tone mapping to produce preferred images. Contrast Limited Histogram Equalisation casts the tone mapping problem as optimally increasing the contrast of an image (in the sense it should have a more uniform histogram of brightness) but constraining the curves to have a bounded slope [6]. While bounding the slope certainly makes tone curves better behaved, they can still be quite wiggly in nature.

Consequently, in [7] it was additionally proposed - to remove this wigglyness - that tone curves should be *simple*. A simple tone curve is defined to have zero or one inflexion point. Importantly, it was shown, on a very large dataset of user tone-mapped images, that the tone curves could be approximated with simple counterparts. The original versus simple tone-mapped images were almost visually indistinguishable, demonstrating that the latitude to make wiggly tone curves was not needed.

In this paper, we propose that tone curves should be even simpler. In Figure 1 a tone mapping function, T, is shown. It is piecewise linear comprising two ramps that meet at the point $(0.2, 0.8)$. The curve's slope is bounded in the interval $[\frac{1}{4}, 4]$.

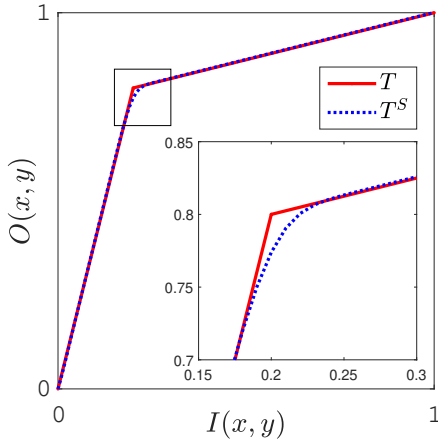

Figure 1. *The target curve, T, shown in red has a corner at $(0.2, 0.8)$ where the gradient is discontinuous. The blue dotted curve, T^S, found by our method with $\sigma = 2$, is optimally close to the target curve but smooth. The boxed region around the corner is shown in the detail view.*

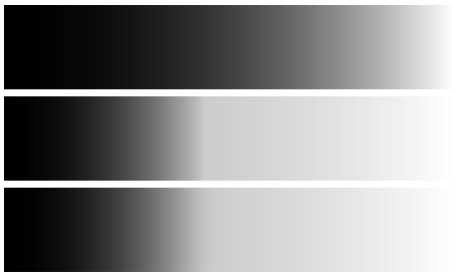

Figure 2. *Top is $I(x,y) = x^2$, $0 \le x \le 1$. Middle is $T(I(x,y))$ from Figure 1 which exhibits a Mach band. Bottom is $T^S(I(x,y))$ and has no artefact.*

Moreover, according to the definition given in [7], the curve is simple. Indeed, it has no inflexion point: the gradient is constant or decreasing across the entire domain. Yet, the sharp change in gradient at $(0.2, 0.8)$ rankles. Approached from the left of $(0.2, 0.8)$, the gradient of the curve is 4 and approached from the right, it is $\frac{1}{4}$. There is a step change in the gradient at this point. We call large discontinuities in gradients *corners*. In this paper, we propose that *even simpler* tone curves are corner-free and, concomitantly, smooth in some sense (which we define later).

A smoother version of the tone curve, T^S, is also plotted in Figure 1. The curve looks almost identical except at the gradient discontinuity where the function now smoothly transitions from a gradient of 4 to a gradient of $\frac{1}{4}$.

Why should we wish to make simple curves even simpler? Our concern is not simply aesthetic in terms of tone curves and gradients; it is also of practical importance. In the top of Figure 2 we show an input image $I(x,y) = x^2$, $0 \le x \le 1$. Because it is quite dark, we will apply the tone curve shown in Figure 1 to brighten it. If $I(x,y)$ denotes an image brightness at image location (x,y) then an output image, $O(x,y)$ is calculated by

$$O(x,y) = T(I(x,y)) \tag{1}$$

The result of applying the tone curve T to the input image is shown in the middle of Figure 2. Observe the visible discontinuity just left of the centre (corresponding to an input brightness of 0.2). The step change in gradient at $(0.2, 0.8)$ is visible in the output image. The corner 'appears' as a band that is brighter than its surroundings despite the tones being strictly increasing in tonal value (brightness). These regions of perceived (though not actual) tonal inversion are known as Mach bands [8, 9]. Clearly, we do not wish to make Mach bands appear in any processed images. Let us apply the smoothed tone function T^S to the same input from the top of Figure 2, generating the output image at the bottom. Here, no Mach band is apparent. Elsewhere, (away from the corner) the rendered tones are the same.

A non-smooth function can be made smooth by convolution with a class of smooth functions known as mollifiers [10]. Amongst these, is the Gaussian function which is uniquely suitable for our purposes [11]. Novelly, we show how to incorporate Gaussian smoothing into a tone curve optimisation. We show how to find the even simpler tone curve - a mollified version of a simple tone curve - that is closest to a target curve. That is, we are not using Gaussian convolution to post-process tone curves but rather incorporating convolution into the optimisation.

The background to this paper is presented next, followed by our even simpler tone curve method. Experiments on the MIT-Adobe FiveK (FiveK) dataset [12] of 25000 tone-adjusted image pairs show that even simpler curves can well-describe a user's tone adjustment. The paper finishes with a short conclusion.

Background

Simple Tone Curves: It is useful to understand the *meaning* of a tone curve represented as a vector. Suppose we have n uniformly spaced input tonal values in the domain [0,1], written $\boldsymbol{b} = [0, 1/(n-1), 2/(n-1), ..., 1]^\top$. Evaluating the tone curve T from Equation (1) at these input tonal values, results in the n-vector $\boldsymbol{t} = [T(0), T(1/(n-1)), T(2/(n-1)), ..., T(1)]^\top$. Here and throughout the superscript $^\top$ denotes the transpose operator. We use t_i to denote the i-th component of $\boldsymbol{t}$. With an appropriate interpolation scheme, the vector pair $\boldsymbol{b}$, $\boldsymbol{t}$ defines the tone curve.

The prior art optimisation to find the *simple* curve $\hat{\boldsymbol{t}}$ that best approximates a target curve $\boldsymbol{t}$ is solved for using the constrained optimisation summarised in Equations (2).

$$\underset{\hat{\boldsymbol{t}},\, c \in \{1,2,3,4\}}{\arg\min} \quad \|\hat{\boldsymbol{t}} - \boldsymbol{t}\| \tag{2a}$$

$$\text{s.t.} \begin{cases} \hat{t}_1 = t_1 & \text{(2b)} \\ \hat{t}_n = t_n & \text{(2c)} \\ \boldsymbol{D}\hat{\boldsymbol{t}} \geq \boldsymbol{0} & \text{(2d)} \\ \boldsymbol{A}^c \hat{\boldsymbol{t}} \geq \boldsymbol{0} & \text{(2e)} \end{cases}$$

This least-squares optimisation, subject to linear equality and inequality constraints, is readily solved using quadratic programming [13, 14]. The objective function (2a) indicates we wish to find the closest simple curve $\hat{\boldsymbol{t}}$ to $\boldsymbol{t}$ where simple is defined by the constraints (2b) to (2e) and depend on the 'case' parameter c. The minimisation is over four cases $c \in \{1,2,3,4\}$ with each case indexing an inflexion point condition. The equality constraints in (2b) and (2c) ensure the curve maps tones to the same output range as the target curve $\boldsymbol{t}$.

The $n \times n$ matrix $\boldsymbol{D}$ calculates the first derivative of a vector. In terms of the optimisation at hand, $\boldsymbol{D}\hat{\boldsymbol{t}}$ is

$$[\boldsymbol{D}\hat{\boldsymbol{t}}]_i = \frac{\hat{t}_i - \hat{t}_{i-1}}{h}, \quad \text{for } i = 2, 3, ..., n\,, \tag{3}$$

where the input domain step size is $h = \frac{t_n - t_1}{n-1}$ and $[\boldsymbol{D}\hat{\boldsymbol{t}}]_1 = [\boldsymbol{D}\hat{\boldsymbol{t}}]_2$ (we adopt homogeneous Neumann boundary conditions). Here and later we use a single subscript on a matrix to denote the i-th row of the matrix. Equation (2d) can now be seen as a constraint that the tone curve is an increasing function.

Finally, Equation (2e) constrains the curve to have zero or one inflexion point. An inflexion point is defined by a change in sign of the second derivative, hence a second derivative operator $\boldsymbol{D}^2$ is defined in Equation (4) as the second-order central finite difference approximation where we do not calculate 2nd derivatives at the boundary.

$$[\boldsymbol{D}^2\hat{\boldsymbol{t}}]_i = \frac{\hat{t}_{i-1} - 2\hat{t}_i + \hat{t}_{i+1}}{h^2}, \quad \text{for } i = 2, 3, ..., n-1 \tag{4}$$

Recall that the minimisation is defined over the four cases, therefore there are four matrices $\boldsymbol{A}^c$ that are indexed by the case number $c \in \{1,2,3,4\}$. Let us consider each in turn.
Case 1: zero inflexion point, gradient increasing,

$$\boldsymbol{A}^1 = \boldsymbol{D}^2 \tag{5}$$

Case 2: zero inflexion point, gradient decreasing,

$$\boldsymbol{A}^2 = -\boldsymbol{D}^2 \tag{6}$$

Clearly, $-\boldsymbol{D}^2\hat{\boldsymbol{t}} \geq \boldsymbol{0} \implies \boldsymbol{D}^2\hat{\boldsymbol{t}} \leq \boldsymbol{0}$. That is, the second derivative is negative and the gradient is decreasing - as required.
Case 3: one inflexion point, gradient increasing then decreasing,

$$\boldsymbol{A}^3 = \begin{bmatrix} \boldsymbol{D}^2_{[1..f]} \\ -\boldsymbol{D}^2_{[(f+1)..n]} \end{bmatrix} \quad \text{for } f = 2, 3, ..., n-1 \tag{7}$$

where $[a..b]$ defines an interval of integers in the range a to b inclusive and block notation is used to represent the vertical concatenation of the matrices. Here the subscript $_{[a..b]}$ indexes all the rows of a matrix in this range.
Case 4: one inflexion point, gradient decreasing then increasing,

$$\boldsymbol{A}^4 = \begin{bmatrix} -\boldsymbol{D}^2_{[1..f]} \\ \boldsymbol{D}^2_{[(f+1)..n]} \end{bmatrix} \quad \text{for } f = 2, 3, ..., n-1 \tag{8}$$

The Dataset: The FiveK dataset [12] is comprised of 5,000 images that have been retouched by five experts. Each expert has made adjustments according to their preference resulting in renditions ranging from similar to distinctly different.

The FiveK dataset has been used in designing automatic image enhancement algorithms [15, 16] that are constrained to be global enhancements in [17] and further to be human interpretable curves in [18]. Although each expert can make many different kinds of adjustments, in summary, their individual edits can be well-approximated by a single global tone curve. Following the notation from the prior art [7], let $\underline{I}(x,y)$, $\underline{P}(x,y)$ and $\underline{P}^G(x,y)$ denote respectively an input image, an expert adjusted output and an approximation thereof. The relationship between these three images is summarised as:

$$\begin{aligned} \underline{I} &= [L_I^* \; a_I^* \; b_I^*]^\top \\ \underline{P} &= [L_P^* \; a_P^* \; b_P^*]^\top \\ \underline{P}^G &= [T(L_I^*) \; a_P^* \; b_P^*]^\top \end{aligned} \tag{9}$$

where dropping the spatial dependence on (x,y), for a single given pixel, $\underline{I}$, $\underline{P}$ and $\underline{P}^G$ are 3-vectors. In this work, the CIELAB colour space [19] is used. It follows that the function T in Equation (1) denotes a tone curve that approximately maps L_I^* to L_P^*. From [7], we have the 25,000 tone curves that map the input images to their corresponding outputs.

Method: Even Simpler Tone Curves

A limitation of the existing simple tone curve method is that it can produce curves that are not continuous in gradient which can lead to artefacts in the images, including Mach bands (see Figures 1 and 2 and related text). We therefore want to augment the method that produces simple curves, to additionally ensure the curves are smooth.

A simple solution here would be to convolve a simple tone curve with a Gaussian kernel to smooth out any step changes in gradient. However, this would not result in the *optimal* smooth curve in the sense that it is closest to the target curve. In our new *even simpler tone curves* method we embed Gaussian smoothing into the optimisation. To do so, let us define $\hat{\boldsymbol{t}}$,

$$\hat{\boldsymbol{t}} = \boldsymbol{G}\boldsymbol{z}, \tag{10}$$

where $\boldsymbol{G}$ is an $n \times n$ Gaussian smoothing matrix (to be defined). The vector $\boldsymbol{z}$ represents a curve that, when smoothed, will be close to the target curve $\boldsymbol{t}$. An advantage of this decomposition is that some aspects of the optimisation will be expressed in terms of the curve $\boldsymbol{z}$ and others in terms of $\hat{\boldsymbol{t}}$.

Let's denote a 1D Gaussian kernel as the w-component vector $\boldsymbol{g}$ where the standard deviation is σ and the window size is $w = 2\lceil 3\sigma \rceil + 1$. Note from this definition that the kernel has an odd number of components, thereby ensuring a central element at $\omega = \lceil 3\sigma \rceil + 1$. Thus, $\boldsymbol{g}$ is defined in Equation (11) which samples the components g_i from the Gaussian function. We further normalise $\boldsymbol{g}$ such that it sums to 1.

$$g_i = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(i-\omega)^2}{2\sigma^2}\right) \quad \text{for } i = 1,2,...,w \tag{11}$$

We now construct a 2D matrix $\boldsymbol{G}$ that, when multiplied by a vector (in this case $\boldsymbol{z}$), yields the same result as convolving the vector $\boldsymbol{z}$ by the 1D Gaussian kernel $\boldsymbol{g}$. The Gaussian kernel is placed on each row with its central element on the diagonal. Clearly, this will result in some elements of $\boldsymbol{g}$ falling outside the boundaries of $\boldsymbol{G}$. Approaches to dealing with the boundaries of filtering include zero-padding, constant-padding or mirroring the signal [20]. The issue is that these methods might introduce new inflexion points, causing the previously expressed simple constraints to be violated. Consider the same curve as Figure 1 illustrated in the top left plot of Figure 3 but extrapolated with constant values at z_1 and z_n. Convolving with the Gaussian kernel shown in red, produces the curve shown top right which has a new inflexion point circled in blue. For this reason, we extend the curve linearly at the boundaries. The middle row of Figure 3 shows this with no new inflexion point in the output.

Let us consider the convolution at the boundary (at the beginning of the tone curve) in more detail. Remembering that z_1 denotes the first point on the tone curve (the lower boundary) and noting z_{-1} is the first extrapolated point, the discrete gradient at this position is $z_2 - z_1$. It follows that $z_{-j} = z_1 - j(z_2 - z_1) = (1+j)z_1 - jz_2$ where, for example, $j = 0,1,...,u$ and the Gaussian kernel has u components that fall beyond the boundary. During convolution, these terms will be multiplied by a Gaussian weight e.g. $g_i z_{-j}$. Observe that z_{-j} depends only on a known linear combination of z_1 and z_2. It follows that we can update the rows of matrix $\boldsymbol{G}$ near the boundary so that the dot product of these rows with the curve $\boldsymbol{z}$ correctly computes the convolution assuming a straight line extrapolation of the tone curve at the upper and lower boundaries. The bottom row of Figure 3 illustrates this. The curve - that has not been extended - is multiplied by the matrix $\boldsymbol{G}$ to produce the same curve as the convolution of $\boldsymbol{g}$ and

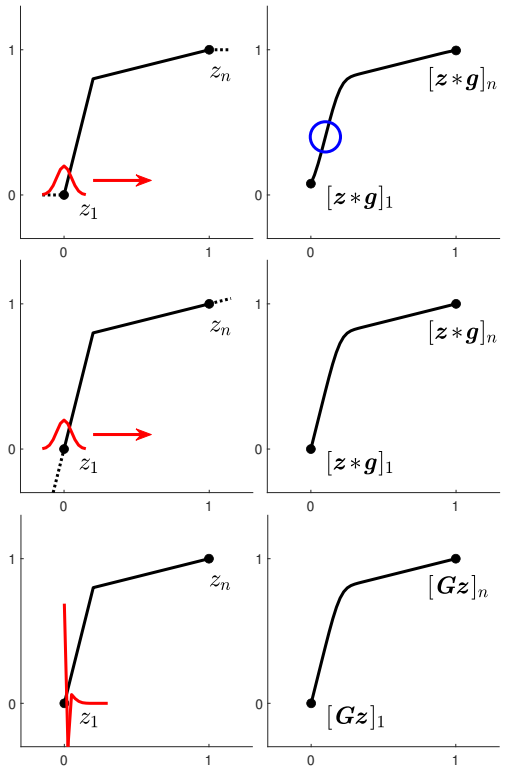

Figure 3. *Top: a tone curve $\boldsymbol{z}$ padded with constant values can introduce inflexion points (e.g. blue circle) when convolved with a Gaussian (red). Middle: extrapolating the curve linearly avoids this issue. Bottom: instead of extending $\boldsymbol{z}$, $\boldsymbol{G}$ can be defined to yield the same result. NB: each row of $\boldsymbol{G}$ is different near the boundary so only the first row is shown (red).*

(linearly extrapolated) $\boldsymbol{z}$ as per the middle plots. Note that only the first row of $\boldsymbol{G}$ is drawn (in red) because each row is different until (and after) the central portion of $\boldsymbol{G}$ where the Gaussian kernel did not fall beyond the boundary. We will given more details on the construction of $\boldsymbol{G}$ elsewhere.

In reformulating the optimisation, we first restate the objective function in Equation (12a), that the smoothed curve $\hat{\boldsymbol{t}} = \boldsymbol{G}\boldsymbol{z}$ will be close to the target curve, $\boldsymbol{t}$.

$$\underset{\boldsymbol{z},\, c \in \{1,2,3,4\}}{\arg\min} \quad \|\boldsymbol{G}\boldsymbol{z} - \boldsymbol{t}\| \tag{12a}$$

$$\text{s.t.} \begin{cases} [\boldsymbol{G}\boldsymbol{z}]_1 = t_1 & \text{(12b)} \\ [\boldsymbol{G}\boldsymbol{z}]_n = t_n & \text{(12c)} \\ \boldsymbol{D}\boldsymbol{z} \geq \boldsymbol{0} & \text{(12d)} \\ \boldsymbol{A}^c\boldsymbol{z} \geq \boldsymbol{0} & \text{(12e)} \end{cases}$$

Constraints (12b) and (12c) ensure the solved for curve has the same end-points as $\boldsymbol{t}$. Whilst, ultimately, we require the curve $\hat{\boldsymbol{t}}$ to be increasing and have limited inflexion points, constraining it would leave $\boldsymbol{z}$ free to overfit, introducing lots of wiggles and local extrema which, when multiplied by $\boldsymbol{G}$, would not guarantee the smoothness of $\hat{\boldsymbol{t}}$. The increasing gradient constraint (12d) and inflexion constraint (12e) are therefore placed on the curve $\boldsymbol{z}$.

However, the question must be asked whether the curve will still be simple after convolution. Crucially, convolution with a Gaussian kernel will not create local extrema (or zero-crossing) [21, 22]. This property applies to higher-order spatial derivatives, meaning no additional zero-crossings are introduced in the second derivative, hence no additional inflexion points. Therefore, a simple curve (no more than one zero-crossing of the second

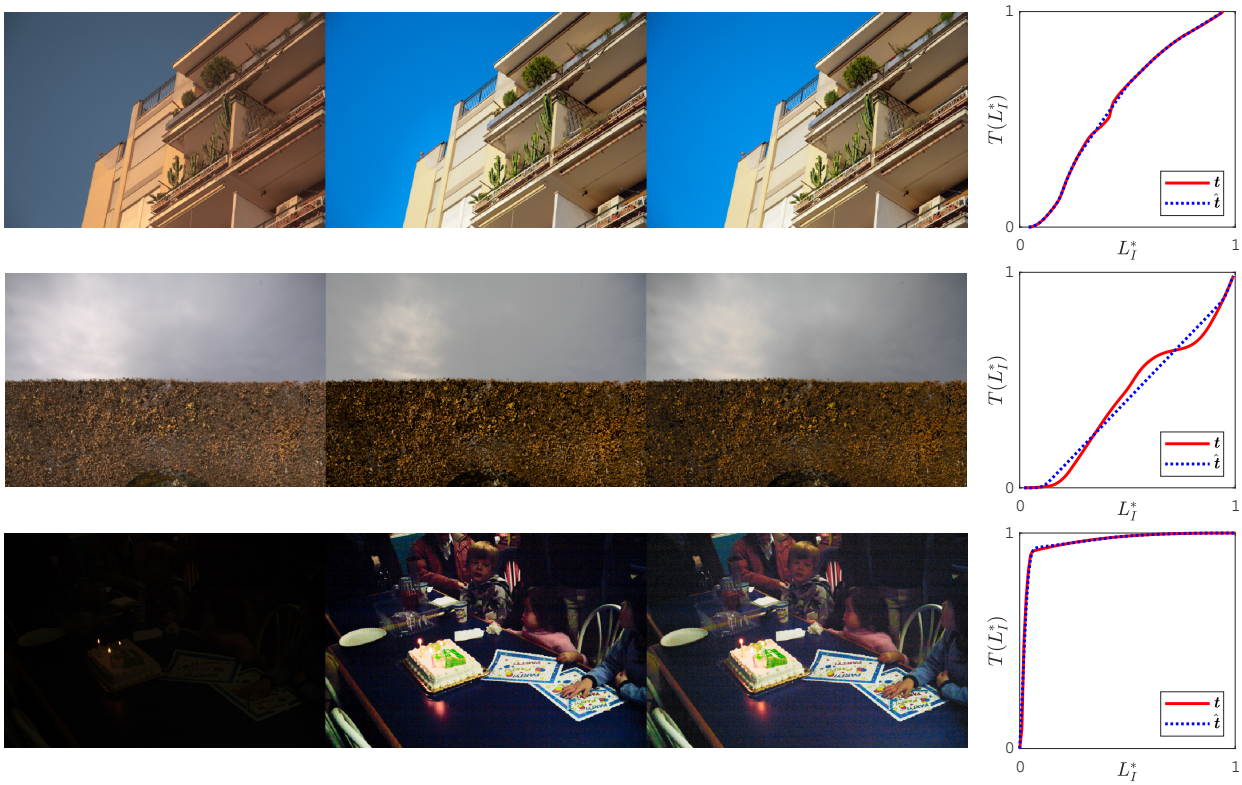

Figure 4. *Example visual results. First column shows the input image $\underline{I}(x,y)$, next is the ground truth image $\underline{P}^G(x,y)$, then the approximated simply enhanced image $\underline{\hat{P}}(x,y)$ and right shows the two tone curves that gave these enhancements. Top row: Image C4340, Middle: A326, Bottom: C3583.*

derivative) remains simple after convolution with a Gaussian. In other words, if $\boldsymbol{z}$ is simple, then so will $\hat{\boldsymbol{t}}$. Thus it is important to apply the inflexion point constraint to $\boldsymbol{z}$, Equation (12e).

Results and Discussion

The even simpler curve $\hat{\boldsymbol{t}}$ is obtained from the solution of the optimisation with Equations (10) and (12). The curve is applied to every pixel of an input image $\underline{I}(x,y)$, yielding $\underline{\hat{P}}(x,y)$ where

$$\underline{\hat{P}} = [\hat{T}(L_I^*)\ a_P^*\ b_P^*]^\top \tag{13}$$

Our method is applied to the 25,000 image pairs from the FiveK dataset [12], comparing $\underline{\hat{P}}(x,y)$ to $\underline{P}^G(x,y)$ by computing the mean ΔE colour difference [19]. Figure 4 shows results of the images at the 0.99 quantile, 3rd worst image and the worst image with their error statistics given in Table 1.

Table 1: Error statistics of the images shown in Figure 4

Image	Mean ΔE	Mean ΔE Rank
C4340	1.16	24751
A326	3.99	24998
C3583	7.67	25000

The 0.99 quantile image has $\Delta E = 1.16$ indicating it is almost impossible to observe any difference between the images as seen in Figure 4. That is, at least 99% of the images have no observable difference. In the middle row, with $\Delta E = 3.99$, image A326 has minimal observable difference which is not readily noticeable [23, 24]. In the worst case image with $\Delta E = 7.67$ the difference is noticeable but it is a very challenging enhancement problem given 10% of the input range is mapped to over 90% of the output range and almost no detail can be seen in $\underline{I}(x,y)$. A theme between several images with high ΔE is poor fitting in the dark tones when the curve is very steep. The sampled tone curve points are spaced relatively closely (in terms of distance along the curve) when the gradient is small and relatively far apart when the gradient is steep. This makes the averaging effect of the Gaussian smoothing more aggressive in the regions where the curve is steep. In future, we would like to investigate a reparameterisation of the curve to avoid this bias.

For the results thus far, $\sigma = 2$ has been used since the banding artefacts of Figure 2 disappeared at that level. For completeness, we report results conducted at different standard deviations, σ of the Gaussian filter in Table 2.

Table 2: Quantiles of the mean ΔE per σ.

	Mean ΔE by quantile				
σ	**0.50**	**0.90**	**0.95**	**0.99**	**1.00**
0	0.0120	0.140	0.259	0.790	3.98
1	0.0235	0.181	0.327	0.892	4.16
2	0.0479	0.296	0.490	1.16	7.67
3	0.0831	0.445	0.688	1.45	11.9

Conclusion

Tone curves enhance images by mapping input to output tones and are used widely in image processing, including in the image processing pipelines in everyone's smartphones. Prior art proposed that tone curves should be simple - not wiggly - meaning they have one or zero inflexion points. In this paper, we have argued that tone curves should be even simpler and we - through a novel optimisation - smooth abrupt changes in the gradient of a tone curve. This paper presented a computational method to find the even simpler curve which was tested on a large dataset of 25,000 tone adjustments. We find artefacts are not introduced with this method and, save for a single challenging image, even simpler tone curves account for all user tone adjustments.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council, grants EP/S023917 and EP/S028730. The experiments were carried out on the High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia.

References

[1] M. Grossberg and S. Nayar, "Determining the camera response from images: What is knowable?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1455–1467, 2003.

[2] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum, "Radiometric calibration from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 938–945.

[3] W. C. Kao, S. H. Wang, L. Y. Chen, and S. Y. Lin, "Design considerations of color image processing pipeline for digital cameras," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, pp. 1144–1152, 2006.

[4] G. Sharma, *Digital Color Imaging Handbook*. CRC press, 2017.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. New York: Pearson, 2008.

[6] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 355–368, 1987.

[7] J. Bennett and G. Finlayson, "Simplifying tone curves for image enhancement," in *Color and Imaging Conference*, 2023, pp. 108–114.

[8] F. Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina*. Holden-Day, 1965.

[9] J. P. Thomas, "Threshold measurements of mach bands∗," *Journal of the Optical Society of America*, vol. 55, no. 5, pp. 521–524, 1965.

[10] K. O. Friedrichs, "The identity of weak and strong extensions of differential operators," *Transactions of the American Mathematical Society*, vol. 55, no. 1, pp. 132–151, 1944.

[11] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 26–33, 1986.

[12] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 97–104.

[13] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press Inc., 1999.

[14] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Prentice-Hall, 1974.

[15] H. U. Kim, Y. J. Koh, and C. S. Kim, "PieNet: Personalized image enhancement," in *European Conference on Computer Vision*, 2020, pp. 374–390.

[16] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[17] Y. Liu, J. He, X. Chen, Z. Zhang, H. Zhao, C. Dong, and Y. Qiao, "Very lightweight photo retouching network with conditional sequential modulation," *IEEE Transactions on Multimedia*, vol. 24, 2022.

[18] S. Moran, S. McDonagh, and G. Slabaugh, "CURL: Neural curve layers for global image enhancement," in *IEEE International Conference on Pattern Recognition*, 2021, pp. 9796–9803.

[19] "Colorimetry — Part 4: CIE 1976 L*a*b* colour space," CIE International Commission on Illumination, Standard, Jun. 2019.

[20] G. Karlsson and M. Vetterli, "Extension of finite length signals for sub-band coding," *Signal Processing*, vol. 17, no. 2, pp. 161–168, 1989.

[21] T. Lindeberg, "On the axiomatic foundations of linear scale-space," in *Gaussian scale-space theory*. Springer, 1997, pp. 75–97.

[22] A. L. Yuille and T. Poggio, "Fingerprints theorems for zero crossings," *Journal of the Optical Society of America A*, vol. 2, no. 5, pp. 683–692, 1985.

[23] M. Stokes, M. D. Fairchild, and R. S. Berns, "Precision requirements for digital color reproduction," *ACM Transactions on Graphics*, vol. 11, pp. 406–422, 1992.

[24] M. K. E. Mahy, L. V. Eycken, and A. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV," *Color Research & Application*, vol. 19, no. 2, 1994.