Near constant-time optimal piecewise LDR to HDR inverse tone mapping

Qian Chen, Guan-Ming Su, Peng Yin

Qian Chen, Guan-Ming Su, Peng Yin, "Near constant-time optimal piecewise LDR to HDR inverse tone mapping," Proc. SPIE 9404, Digital Photography XI, 94040O (27 February 2015); doi: 10.1117/12.2080389

Event: SPIE/IS&T Electronic Imaging, 2015, San Francisco, California, United States

Near Constant-time Optimal Piecewise LDR to HDR Inverse Tone Mapping

Qian Chen, Guan-Ming Su, Peng Yin
Dolby Laboratories Inc., 432 Lakeside Drive, Sunnyvale, CA USA 94085

ABSTRACT

In a backward compatible HDR image/video compression, it is a general approach to reconstruct HDR from compressed LDR as a prediction to original HDR, which is referred to as inverse tone mapping. Experimental results show that 2-piecewise 2nd order polynomial has the best mapping accuracy than 1 piece high order or 2-piecewise linear, but it is also the most time-consuming method because to find the optimal pivot point to split LDR range to 2 pieces requires exhaustive search. In this paper, we propose a fast algorithm that completes optimal 2-piecewise 2nd order polynomial inverse tone mapping in near constant time without quality degradation. We observe that in least square solution, each entry in the intermediate matrix can be written as the sum of some basic terms, which can be pre-calculated into look-up tables. Since solving the matrix becomes looking up values in tables, computation time barely differs regardless of the number of points searched. Hence, we can carry out the most thorough pivot point search to find the optimal pivot that minimizes MSE in near constant time. Experiment shows that our proposed method achieves the same PSNR performance while saving 60 times computation time compared to the traditional exhaustive search in 2-piecewise 2nd order polynomial inverse tone mapping with continuous constraint.

Keywords: HDR, inverse tone mapping, piecewise polynomial prediction, pivot point, continuous constraint

1. INTRODUCTION

The discrepancy between the wide ranges of illumination that can be captured in high dynamic range (HDR) and the small ranges that can be reproduced by existing displays makes it difficult to display HDR source accurately [1]. Tone mapping is introduced for such purpose to compress HDR content to low dynamic range (LDR) for proper display. However, the reverse problem has an equal importance. In a backward compatible HDR image/video compression, it is a general approach to encode 8-bit LDR representation in base layer, and reconstruct HDR from the compressed LDR using an inverse mapping as a prediction to original HDR source [2]. The techniques to produce HDR image from its LDR version is referred to as inverse tone mapping. Simply taking an existing tone mapping operator and invert it for inverse tone mapping does not guarantee ideal result [3]. In reality, how a HDR image is mapped to LDR is often unclear or too complicated for inversion, forcing us to find simple alternatives for inverse tone mapping. Clearly, the shape of inverse tone-mapping curve is highly nonlinear. One observation is highlight and light sources need to be expanded more aggressively than remainder of the image [4], resulting in a piecewise shaped curve with at least one splitting point between each scaling function [1][4].

1.1 Piecewise polynomial inverse mapping

A simple non-linear inverse tone mapping is polynomial. Let s_i be the i^{th} pixel in an LDR image. Let v_i as the corresponding co-located HDR pixel. Let $\hat{v}_i$ as the corresponding co-located *predicted* HDR pixel. Suppose there are P pixels in the given picture. For the i^{th} pixel, the predicted value using n^{th} order polynomial can be expressed as

$$\hat{v}_i = m_0 + m_1 s_i + m_2 {s_i}^2 + \cdots + m_n {s_i}^n \tag{1}$$

To obtain a polynomial predictor from LDR to HDR, we may resort to least squared solution. We can arrange above equation into matrix form $\hat{\boldsymbol{v}} = \boldsymbol{Sm}$, or

Digital Photography XI, edited by Nitin Sampat, Radka Tezaur, Dietmar Wüller, Proc. of SPIE-IS&T Electronic Imaging
SPIE Vol. 9404, 94040O · © 2015 SPIE-IS&T · CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2080389

$$\begin{bmatrix} \hat{v}_0 \\ \hat{v}_1 \\ \vdots \\ \hat{v}_{P-1} \end{bmatrix} = \begin{bmatrix} 1 & s_0 & \dots & s_0{}^n \\ 1 & s_1 & \dots & s_1{}^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & s_{P-1} & \dots & s_{P-1}{}^n \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_n \end{bmatrix} \tag{2}$$

The least squared solution is obtained by

$$\boldsymbol{m} = (\boldsymbol{S}^T\boldsymbol{S})^{-\mathbf{1}}(\boldsymbol{S}^T\boldsymbol{v})$$

Where

$$\boldsymbol{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{P-1} \end{bmatrix}$$

is the original HDR pixel value.

We could further improve the prediction performance using piecewise polynomial prediction. To obtain the piecewise predictor, taking 2-piece predictor as an example, we need to find a pivot point which connects two different polynomial predictors. Assume the pivot point has value s_v in LDR, it will partition LDR range into two parts $[0, s_v)$ and $[s_v, 1)$. Assume there are L pixels in the lower partition and H pixels in the higher partition, where $L + H = P$. We will select different predictors for a pixel depending on which range it falls into:

if $0 \leq s_i < s_v$, $\hat{v}_i = m_0^L + m_1^L s_i + \cdots m_n^L s_i{}^n$, otherwise $\hat{v}_i = m_0^H + m_1^H s_i + \cdots m_n^H s_i{}^n$

where $\boldsymbol{m}^L = \begin{bmatrix} m_0^L \\ m_1^L \\ \vdots \\ m_n^L \end{bmatrix}$ and $\boldsymbol{m}^H = \begin{bmatrix} m_0^H \\ m_1^H \\ \vdots \\ m_n^H \end{bmatrix}$ are the predictor coefficients used in the lower partition and higher partition respectively. Given a pivot point, we can collect the predictor into matrix form as

$$\hat{\boldsymbol{v}}^L = \boldsymbol{S}^L\boldsymbol{m}^L$$

and

$$\hat{\boldsymbol{v}}^H = \boldsymbol{S}^H\boldsymbol{m}^H \tag{3}$$

The least square solutions to each partition is

$$\boldsymbol{m}^L = ((\boldsymbol{S}^L)^T\boldsymbol{S}^L)^{-1}((\boldsymbol{S}^L)^T\boldsymbol{v}^L)$$

and

$$\boldsymbol{m}^H = ((\boldsymbol{S}^H)^T\boldsymbol{S}^H)^{-1}((\boldsymbol{S}^H)^T\boldsymbol{v}^H) \tag{4}$$

For each given pivot point, we map the entire LDR to HDR using the derived predictors, and calculate the prediction error between the predicted and original HDR. The same procedure applies to all possible pivot points until we find the one with the minimal prediction error.

Fig. 1 illustrates LDR to HDR prediction using piecewise polynomial in R channels of HDR testing image AhwahneeGreatLounge [5]. It can be clearly seen that the dark and highlight parts of true LDR to HDR mapping (yellow scatter plot) have utterly distinctive characteristics. The dark part mapping is quite flat, while the highlight part demonstrates a sharp upturn, making the entire mapping an “L” shape, which cannot be well captured with one piece high order polynomial. For example, none of one piece 2nd, 3rd or 4th order is able to follow the upturn in highlight part. Instead, if we use 2-piece that searches the best pivot point to partition LDR range into two, and apply 2nd order polynomial to each partition respectively, the prediction accuracy can be greatly improved. For illustration purpose, we only searched 20 pivot points instead an exhaustive search of 255 points which is extremely time-consuming. The objective quality metric results for more testing HDR images of various piecewise polynomial predictions can be found in Table 1. 2-piece 2nd order polynomial prediction shows the best quality, which is consistent with our observation. Note that piecewise linear is a special case of piecewise polynomial with order 1. If the least square solution in high order turns to be singular, it will degrade to linear solution. Since piecewise polynomial always outperforms piecewise linear prediction unless HDR is a mere high bit-depth version of LDR, we will stay with 2-piece 2nd order polynomial prediction throughout this work.

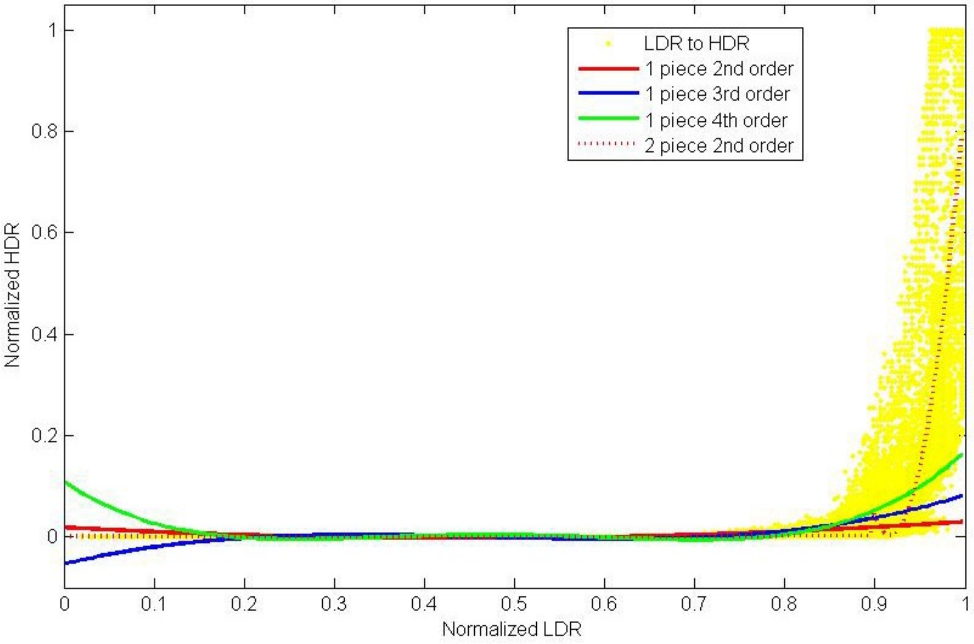

Figure 1 AhwahneeGreatLounge R Channel LDR to HDR Inverse Tone Mapping

1.2 Continuous piecewise polynomial inverse mapping

If we do not impose any constraint in the piecewise polynomial prediction, discontinuity issue arises near the pivot point, which may cause strange artifacts in a smooth area for those boundary pixels between pieces in reconstructed HDR image. For example, a clear discontinuity appears at the pivot point around 0.9 in 2-piece 2nd order polynomial prediction in Fig.2, which leads to unnatural artifact in smooth area of value 0.9.

To avoid this issue, we impose a continuous constraint (CC) by enforcing the starting point of the 2nd piece to connect to the ending point of the 1st piece prediction. And the way to obtain the predictor needs some change [6]. First, we need to find the ending point of the 1st piece. Note the connecting point is always pivot point in LDR, i.e. s_v, and the predicted value of s_v using the first predictor coefficients is

$$\hat{v}_v = m_0^L + m_1^L s_v + m_2^L (s_v)^2 \tag{5}$$

In order for the 2nd piece to cross the ending point of the 1st piece $(s_v, \hat{v}_v)$, we can offset the origin to coordinate $(s_v, \hat{v}_v)$ and enforce the 2nd piece prediction to cross the origin. The predictor of the 2nd piece becomes

$$(\hat{v}_i - \hat{v}_v) = mt_1^H (s_i - s_v) + mt_2^H \left(s_j - s_v\right)^2 \tag{6}$$

We re-arrange the matrix

$$\boldsymbol{S}^H = \begin{bmatrix} s_0 - s_v & (s_0 - s_v)^2 \\ s_1 - s_v & (s_1 - s_v)^2 \\ s_2 - s_v & (s_2 - s_v)^2 \\ \vdots & \vdots \\ s_{H-1} - s_v & (s_{H-1} - s_v)^2 \end{bmatrix}, \quad \boldsymbol{v}^H = \begin{bmatrix} v_0 - \hat{v}_v \\ v_1 - \hat{v}_v \\ v_2 - \hat{v}_v \\ \vdots \\ v_{H-1} - \hat{v}_v \end{bmatrix} \tag{7}$$

And the least square solution is

$$\begin{bmatrix} mt_1^H \\ mt_2^H \end{bmatrix} = \boldsymbol{mt}^H = ((\boldsymbol{S}^H)^T \boldsymbol{S}^H)^{-1}((\boldsymbol{S}^H)^T \boldsymbol{v}^H) \tag{8}$$

Note the obtained coefficients are based on offset coordinate in (6), we have to shift back to the original coordinate as

$$m_0^H = (mt_2^H \cdot {s_v}^2 - mt_1^H \cdot s_v + \hat{v}_v)$$

$$m_1^H = mt_1^H - 2mt_2^H \cdot s_v$$

$$m_2^H = mt_2^H \tag{9}$$

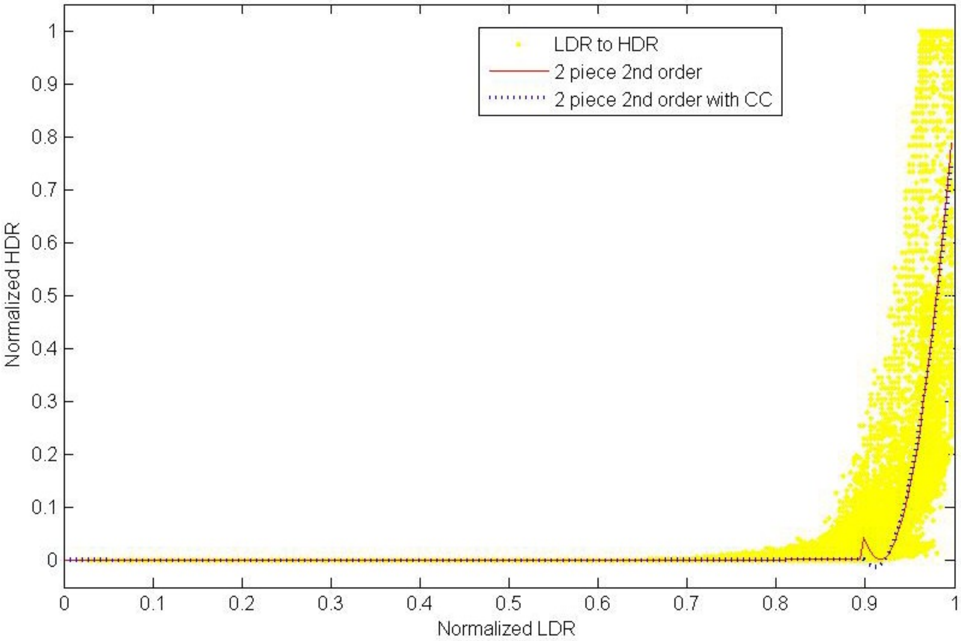

Figure 2 Piecewise Polynomial Prediction with vs. without Continuous Constraint in AhwahneeGreatLounge R Channel

Fig.2 shows 2-piece 2^{nd} order polynomial with and without CC in R channels of AhwahneeGreatLounge [5]. Apparently, the prediction transition of the two pieces at pivot point with CC is much smoother, and the pivot point also changes position with CC, which will lose some optimality compared to solution without CC. As shown in Fig. 3, for all searched pivot point, the prediction error using 2-piece 2^{nd} order polynomial with CC is no less or bigger than that without CC. Here we use sum of squared error (SSE) to measure the prediction error. This can be further confirmed by objective quality results in Table 1, where piecewise polynomial prediction without CC always performs slightly better in PSNR than when continuous constraint is imposed. However, it still has significantly better PSNR than single piece high order polynomial, and most importantly, the predicted image with CC looks smoother in visual quality. Therefore, we choose 2-piece 2^{nd} order polynomial prediction with CC and propose a fast implementation algorithm in the rest of the paper.

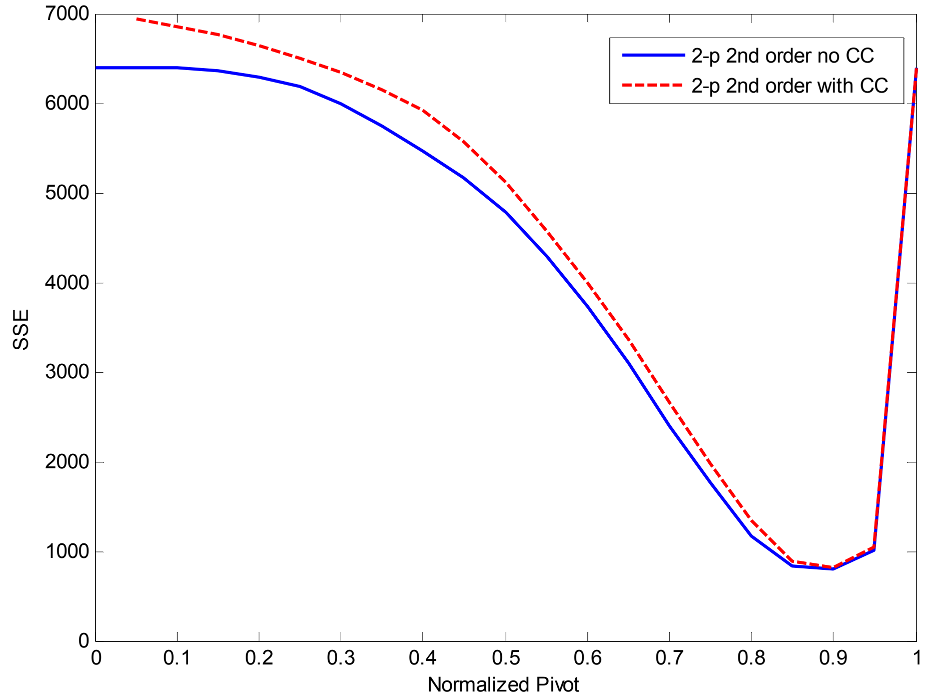

Figure 3 SSE vs. Pivot in 2-piece 2nd Order Polynomial with and without Continuous Constraint

Table 1 PSNR(dB) and Computation Time of LDR to HDR Inverse Tone Mapping (20 Pivot Search Points for 2-piecewise polynomial)

HDR Image	Prediction	R	G	B	Time Elapsed(second)
AwahneeGreat Lounge	1-p 2nd order	34.1481	35.4010	36.8270	2.8075
	1-p 3rd order	34.5133	35.7131	37.1368	3.8006
	1-p 4th order	35.0362	36.1884	37.7815	7.4106
	2-p 2nd order no CC	37.2620	43.2714	43.1063	81.1446
	2-p 2nd order CC	37.2004	42.4520	42.9768	85.6984
GoldenGate	1-p 2nd order	37.1975	41.1037	40.8966	2.4736
	1-p 3rd order	38.4583	41.7455	41.1907	3.5800
	1-p 4th order	40.2971	43.1700	41.2145	5.2667
	2-p 2nd order no CC	46.8538	50.2905	42.9151	71.2735
	2-p 2nd order CC	46.1710	50.1942	42.9115	87.1294
BarHarborSunr ise	1-p 2nd order	32.0904	34.7869	39.0455	2.2208
	1-p 3rd order	33.5155	35.7894	39.5181	3.0281
	1-p 4th order	35.2333	37.4025	41.0009	4.4390
	2-p 2nd order no CC	41.1181	52.3395	51.4861	60.3272
	2-p 2nd order CC	41.0742	51.0419	50.5915	72.6932
LuxoDoubleC hecker	1-p 2nd order	33.7147	34.2756	36.6575	1.0940
	1-p 3rd order	34.9388	35.5495	37.7546	1.6657
	1-p 4th order	36.1107	36.9185	38.8789	2.5023
	2-p 2nd order no CC	39.0998	41.1201	42.6677	32.1277
	2-p 2nd order CC	38.8283	40.5031	42.3371	36.2127
NiagaraFalls	1-p 2nd order	24.3342	23.0953	21.3680	2.0840
	1-p 3rd order	24.5452	23.1936	21.4163	3.0382
	1-p 4th order	24.5634	23.2195	21.4167	4.3457
	2-p 2nd order no CC	24.6107	23.2958	21.4831	60.3855
	2-p 2nd order CC	24.6090	23.2952	21.4827	75.9378

It is clear that 2-piecewise 2nd order polynomial gives the best prediction, followed by 2-piecewise 2nd order polynomial with continuous constraint. However, it is also the most time-consuming method because to find the optimal pivot point to split LDR luminance range to 2 pieces requires exhaustive search.

2. NEAR-CONSTANT TIME PIECEWISE POLYNOMIAL INVERSE TONE MAPPING

In this section, we propose a fast algorithm that can complete optimal 2-piece 2nd order polynomial inverse tone mapping from LDR to HDR in near constant time.

2.1 Motivations

Suppose we select pivot point s_v. Recall in 2-piece 2nd order prediction, the solution to the two partitions segmented s_v is

$$\boldsymbol{m}^L = ((\boldsymbol{S}^L)^T\boldsymbol{S}^L)^{-1}((\boldsymbol{S}^L)^T\boldsymbol{v}^L)$$

and

$$\boldsymbol{m}^H = ((\boldsymbol{S}^H)^T\boldsymbol{S}^H)^{-1}((\boldsymbol{S}^H)^T\boldsymbol{v}^H) \tag{10}$$

To facilitate our discussion, we use the following notations

$$\boldsymbol{B} = \boldsymbol{S}^T\boldsymbol{S},\ \boldsymbol{a} = \boldsymbol{S}^T\boldsymbol{v} \tag{11}$$

For the lower partition

$$\boldsymbol{B}^L = \begin{bmatrix} b_{00}^L & b_{01}^L & b_{02}^L \\ b_{10}^L & b_{11}^L & b_{12}^L \\ b_{20}^L & b_{21}^L & b_{22}^L \end{bmatrix} = (\boldsymbol{S}^L)^T\boldsymbol{S}^L = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ s_0 & s_1 & s_2 & \dots & s_{L-1} \\ {s_0}^2 & {s_1}^2 & {s_2}^2 & \dots & {s_{L-1}}^2 \end{bmatrix} \begin{bmatrix} 1 & s_0 & {s_0}^2 \\ 1 & s_1 & {s_1}^2 \\ 1 & s_2 & {s_2}^2 \\ \vdots & \vdots & \vdots \\ 1 & s_{L-1} & {s_{L-1}}^2 \end{bmatrix} \tag{12}$$

Each entry of the matrix can be expressed as

$$\begin{gathered} b_{00}^L = L \\ b_{01}^L = b_{10}^L = \sum_{i=0}^{L-1} s_i \\ b_{02}^L = b_{11}^L = b_{20}^L = \sum_{i=0}^{L-1} {s_i}^2 \\ b_{21}^L = b_{12}^L = \sum_{i=0}^{L-1} {s_i}^3 \\ b_{22}^L \sum_{i=0}^{L-1} {s_i}^4 \end{gathered} \tag{13}$$

Likewise,

$$\boldsymbol{a}^L = \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \end{bmatrix} = (\boldsymbol{S}^L)^T\boldsymbol{v}^L = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ s_0 & s_1 & s_2 & \dots & s_{L-1} \\ {s_0}^2 & {s_1}^2 & {s_2}^2 & \dots & {s_{L-1}}^2 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{L-1} \end{bmatrix} \tag{14}$$

Each entry can be expressed as

$$a_0^L = \sum_{i=0}^{L-1} v_i$$
$$a_1^L = \sum_{i=0}^{L-1} s_i v_i$$
$$a_2^L = \sum_{i=0}^{L-1} {s_i}^2 v_i \tag{15}$$

The higher partition can be organized into matrix in the same way, with each entry having different summation range $[L, \quad P]$, where P is total number of pixels.

$$\boldsymbol{B}^H = \begin{bmatrix} b_{00}^H & b_{01}^H & b_{02}^H \\ b_{10}^H & b_{11}^H & b_{12}^H \\ b_{20}^H & b_{21}^H & b_{22}^H \end{bmatrix} = (\boldsymbol{S}^H)^T \boldsymbol{S}^H = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ s_L & s_{L+1} & s_{L+2} & \cdots & s_{P-1} \\ {s_L}^2 & {s_{L+1}}^2 & {s_{L+2}}^2 & \cdots & {s_{P-1}}^2 \end{bmatrix} \begin{bmatrix} 1 & s_L & {s_L}^2 \\ 1 & s_{L+1} & {s_{L+1}}^2 \\ 1 & s_{L+2} & {s_{L+2}}^2 \\ \vdots & \vdots & \vdots \\ 1 & s_{P-1} & {s_{P-1}}^2 \end{bmatrix} \tag{16}$$

$$\boldsymbol{a}^L = \begin{bmatrix} a_0^H \\ a_1^H \\ a_2^H \end{bmatrix} = (\boldsymbol{S}^L)^T \boldsymbol{v}^L = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ s_L & s_{L+1} & s_{L+2} & \cdots & s_{P-1} \\ {s_L}^2 & {s_{L+1}}^2 & {s_{L+2}}^2 & \cdots & {s_{P-1}}^2 \end{bmatrix} \begin{bmatrix} v_L \\ v_{L+1} \\ v_{L+2} \\ \vdots \\ v_{P-1} \end{bmatrix} \tag{17}$$

2.2 Observations

Note that there is limited number of possible values in LDR image. Suppose the bit depth is 8, LDR value s_i has 256 possibilities. Keeping this in mind, we have the following observations regarding the entry in $\boldsymbol{B}$.

1. Denote the bin count for each possible LDR value b ($b \in [0,255]$) as h_b: $h_b = \sum_{i=0}^{P-1}(s_i == b)$
2. The value of s_i, ${s_i}^2$, ${s_i}^3$ and ${s_i}^4$ can be obtained through pre-calculated look-up-tables (LUT), so that we do not need to calculate it every time. There are total 256 entries in each LUT as s_i can only be integers between 0 and 255. Denote the pre-calculated value as

$$t_b = \frac{b}{2^8}$$
$${t_b}^2 = \left(\frac{b}{2^8}\right)^2$$
$${t_b}^3 = \left(\frac{b}{2^8}\right)^3$$
$${t_b}^4 = \left(\frac{b}{2^8}\right)^4 \tag{18}$$

Then the entries in $\boldsymbol{B}^L$ can be rewritten as

$$b_{00}^L = \sum_{b=0}^{s_v-1} h_b$$
$$b_{01}^L = b_{10}^L = \sum_{b=0}^{s_v-1} h_b t_b$$

$$b_{02}^{L}=b_{11}^{L}=b_{20}^{L}=\sum_{b=0}^{s_v-1}h_b{t_b}^2$$
$$b_{21}^{L}=b_{12}^{L}=\sum_{b=0}^{s_v-1}h_b\,{t_b}^3$$
$$b_{22}^{L}=\sum_{b=0}^{s_v-1}h_b\,{t_b}^4$$

(19)

3. We can also obtain the summation of HDR values to include bin count of value b as

$$w_b=\sum_{i=0}^{P-1}(s_i==b)v_i$$
$${w_b}^2=\sum_{i=0}^{P-1}(s_i==b){v_i}^2$$

(20)

Hence, the entry in $\boldsymbol{a}^L$ becomes

$$a_0^L=\sum_{b=0}^{s_v-1}w_b$$
$$a_1^L=\sum_{b=0}^{s_v-1}w_b\,t_b$$
$$a_2^L=\sum_{b=0}^{s_v-1}w_b\,{t_b}^2$$

(21)

The entries in $\boldsymbol{B}^H$ and $\boldsymbol{a}^H$ can be derived in the same way. From 2 and 3, the entries of $\boldsymbol{B}$ and $\boldsymbol{a}$ can be obtained by a few multiplications and summations for every searched pivot, much simpler than traditional method, where we have to scan the entire image to segment all pixels to two partitions and collect them into $\boldsymbol{S}^L$, $\boldsymbol{v}^L$ and $\boldsymbol{S}^H$, $\boldsymbol{v}^H$ respectively, which is very time-consuming.

4. For every searched pivot, we need to calculate prediction error and pick the one with minimal distortion. Denote the predicted HDR value of v_i as $\hat{v}_i$. SSE for lower partition is calculated as

$$D^L(s_v)=\sum_{i=0}^{L-1}(v_i-\hat{v}_i)^2=\sum_{i=0}^{L-1}{v_i}^2-2\sum_{i=0}^{L-1}v_i\hat{v}_i+\sum_{i=0}^{L-1}{\hat{v}_i}^2 \quad (22)$$

Each term can be further simplified as

$$\sum_{i=0}^{L-1}{v_i}^2=\sum_{b=0}^{s_v-1}{w_b}^2$$
$$\sum_{i=0}^{L-1}{\hat{v}_i}^2=\sum_{b=0}^{s_v-1}h_b{\hat{v}_b}^2$$

$$\sum_{i=0}^{L-1} v_i \hat{v}_i = \sum_{b=0}^{s_v - 1} w_b \hat{v}_b$$

(23)

Therefore,

$$D^L(s_v) = \sum_{b=0}^{s_v - 1} \left({w_b}^2 - 2 w_b \hat{v}_b + h_b {\hat{v}_b}^2 \right) \tag{24}$$

Similarly, for higher partition,

$$D^H(s_v) = \sum_{b=s_v}^{2^8 - 1} \left({w_b}^2 - 2 w_b \hat{v}_b + h_b {\hat{v}_b}^2 \right) \tag{25}$$

And the overall prediction distortion is

$$D(s_v) = D^L(s_v) + D^H(s_v) \tag{26}$$

The optimal pivot point is selected to be

$$s_v^{opt} = \arg min\ D(s_v) \tag{27}$$

In conclusion, if we pre-calculated LDR histogram h_b, LUT $t_b, {t_b}^2, {t_b}^3, {t_b}^4$, and HDR statistics w_b and ${w_b}^2$, the solution to the predictors for each pivot point becomes very simple. Since the majority of calculation time is taken by pre-calculation, to increase the number of searched pivot does not have significant impact on total computation. Hence, the computation time of the proposed fast algorithm is close to a constant given a LDR to HDR mapping problem. Hence, we can carry out the most thorough pivot point search from 0 to 255 to find the optimal pivot that minimizes SSE in near constant time.

2.3 Fast algorithm with continuous constraint

To impose continuous constraint on the least square solution, slight change occurs to higher partition while the lower partition remains the same. Note $\boldsymbol{S}^H$ is without constant term under continuous constraint. Hence we have

$$\boldsymbol{B}^H = \begin{bmatrix} b_{00}^H & b_{01}^H \\ b_{10}^H & b_{11}^H \end{bmatrix} = (\boldsymbol{S}^H)^T \boldsymbol{S}^H$$

$$= \begin{bmatrix} s_L - s_v & s_{L+1} - s_v & s_{L+2} - s_v & \cdots & s_{P-1} - s_v \\ (s_L - s_v)^2 & (s_{L+1} - s_v)^2 & (s_{L+2} - s_v)^3 & \cdots & (s_{P-1} - s_v)^2 \end{bmatrix} \begin{bmatrix} s_L - s_v & (s_L - s_v)^2 \\ s_{L+1} - s_v & (s_{L+1} - s_v)^2 \\ s_{L+2} - s_v & (s_{L+2} - s_v)^3 \\ \vdots & \vdots \\ s_{P-1} - s_v & (s_{P-1} - s_v)^2 \end{bmatrix} \tag{28}$$

And each entry in $\boldsymbol{B}^H$ can be expressed by shifting indices of ${t_b}^2$, ${t_b}^3$ and ${t_b}^4$

$$b_{00}^H = \sum_{b=s_v}^{2^8 - 1} h_b {t_{(b-s_v)}}^2$$

$$b_{01}^H = b_{10}^H = \sum_{b=s_v}^{2^8 - 1} h_b {t_{(b-s_v)}}^3$$

$$b_{11}^H = \sum_{b=s_v}^{2^8 - 1} h_b {t_{(b-s_v)}}^4$$

(29)

Similarly,

$$\boldsymbol{a}^H = \begin{bmatrix} a_0^H \\ a_1^H \end{bmatrix} = (\boldsymbol{S}^H)^T \boldsymbol{v}^H = \begin{bmatrix} s_L - s_v & s_{L+1} - s_v & s_{L+2} - s_v & \dots & s_{P-1} - s_v \\ (s_L - s_v)^2 & (s_{L+1} - s_v)^2 & (s_{L+2} - s_v)^3 & \dots & (s_{P-1} - s_v)^2 \end{bmatrix} \begin{bmatrix} v_L - \hat{v}_v \\ v_{L+1} - \hat{v}_v \\ v_{L+2} - \hat{v}_v \\ \vdots \\ v_{P-1} - \hat{v}_v \end{bmatrix} \tag{30}$$

And each entry $\boldsymbol{a}^H$ can be written as

$$a_0^H = \sum_{i=L}^{P-1} (s_i - s_v)(v_i - \hat{v}_v) = \sum_{b=s_v}^{2^8-1} \sum_{i=L}^{P-1} (s_i == b)(s_i - s_v)(v_i - \hat{v}_v) = \sum_{b=s_v}^{2^8-1} t_{(b-s_v)}(w_b - h_b \hat{v}_v)$$

$$a_1^H = \sum_{i=L}^{P-1} (s_i - s_v)^2(v_i - \hat{v}_v) = \sum_{b=s_v}^{2^8-1} \sum_{i=L}^{P-1} (s_i == b)(s_i - s_v)^2(v_i - \hat{v}_v) = \sum_{b=s_v}^{2^8-1} {t_{(b-s_v)}}^2(w_b - h_b \hat{v}_v)$$

(31)

Finally, after deriving predictors from (8), the actual predictors will be converted by (9).

3. SIMULATION RESULTS

To illustrate the performance of the proposed fast piecewise polynomial prediction of LDR to HDR inverse tone mapping, we apply the method to the same testing images in Section 1. Table 1 concludes that though 2-piece 2nd order polynomial achieves the best prediction in PSNR, followed by 2-piece 2nd order polynomial with CC, it is also more time consuming than any one piece polynomial alternative.

Table 2 shows both PSNR and computation time of the exhaustive and proposed fast algorithm for 2-piece 2nd order polynomial with CC. To evaluate how much time the fast method can save compared to the exhaustive search method, an exhaustive search of all possible 255 pivot points ($s_v \in [0,254)$, $s_v \in \mathbf{I}$) has been carried out in our experiments for the two methods, as $s_v = 0$ and $s_v = 255$ both correspond to one piece partition and eventually the same case. Naturally, we got the identical PSNR performance for all RGB channels because both methods can find the pivot point of the optimal objective quality in an exhaustive search. Timing-wise, because most computation happens before pivot point search to construct histogram and LUT, which can be conveniently utilized to obtain least square solution and prediction error for every searched pivot with minor cost, the search time is close to constant regardless of the number of pivot point searched in the fast method, as is illustrated in Fig. 4. We observe the proposed algorithm saves a dramatic amount of more than 60 times computation than the exhaustive search method for all 5 testing images consistently.

Table 2 PSNR(dB) and Computation Time of 2-piece 2nd Order Polynomial with CC

HDR Image	Prediction	R	G	B	Time Elapsed(second)
AwahneeGreatLounge	Exhaustive 2-p 2nd order CC	37.2837	45.1115	44.0152	2066.6505
	Fast 2-p 2nd order CC	37.2837	45.1115	44.0152	35.5377
GoldenGate	Exhaustive 2-p 2nd order CC	46.7661	50.4559	42.9141	2064.4720
	Fast 2-p 2nd order CC	46.7631	50.4559	42.9141	31.8371
BarHarborSunrise	Exhaustive 2-p 2nd order CC	41.0861	51.0328	52.7227	1692.3669
	Fast 2-p 2nd order CC	41.0861	51.0328	52.7227	26.1760
LuxoDoubleChecker	Exhaustive 2-p 2nd order CC	39.2810	40.6045	43.0883	863.3986
	Fast 2-p 2nd order CC	39.2810	40.6045	43.0883	12.9137
NiagaraFalls	Exhaustive 2-p 2nd order CC	24.6099	23.2971	21.4839	1757.6618
	Fast 2-p 2nd order CC	24.6099	23.2970	21.4839	26.3432

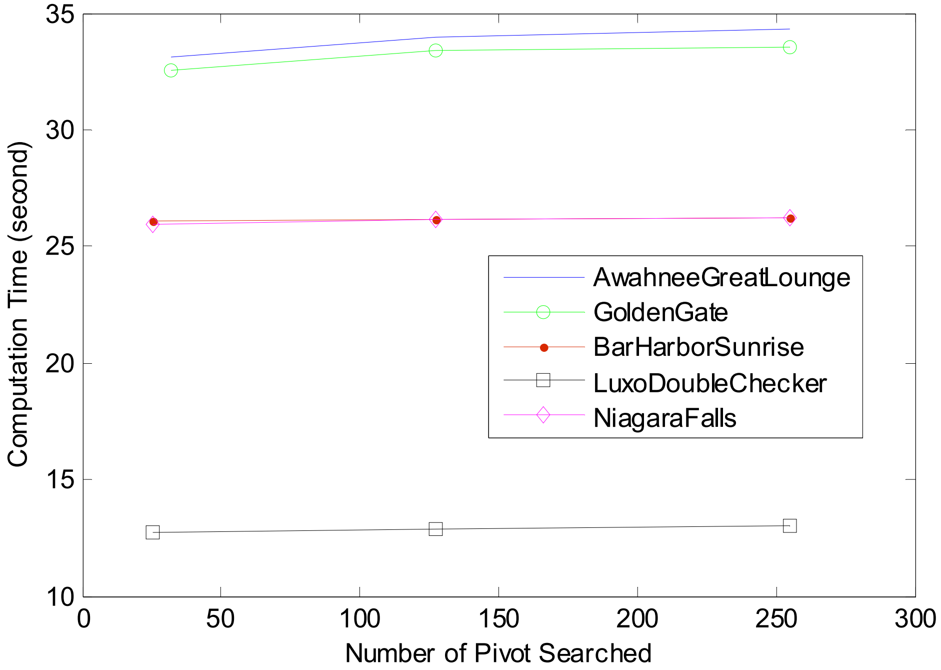

Figure 4 Near Constant Computation Time vs. Number of Pivot Searched

In our discussion and experiments, we focus on 2-piece 2nd order polynomial prediction between LDR and HDR inverse mapping of image. However, the proposed fast algorithm can be easily extended to more complicated scenarios. For example, in backward compatible HDR video compression, the time saving performance of the proposed near constant-time method can be more evident. This is because all LUT $t_b, {t_b}^2, {t_b}^3, {t_b}^4$ only need to be calculated once and cached to be used for every frame in the entire sequence. Moreover, the proposed fast algorithm can be integrated to multi-piece polynomial prediction which has more than 2 partitions. The pre-calculation before pivot search concept still applies, and one only needs to be cautious about the range in calculation of entries of $\boldsymbol{B}$ and $\boldsymbol{a}$ for each piece.

REFERENCES

[1] Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., Myszkowski, K., [High Dynamic Range Imaging – Acuisition, Display, and Image-Based Lighting], second edition, Morgan Kaufmann (2010).

[2] Mai, Z., Mansour, H., Mantiuk, R., Nasiopoulos, P., Heidrich, W., "Visually Favorable Tone-Mapping With High Compression Performance in Bit-Depth Scalable Video Coding", Multimedia, IEEE Transactions on, 15(7), 1503-1518 (2013).

[3] Banterle, P., Ledda, P., Debattista, L., and Chalmers, A., "Inverse Tone Mapping," 18th Eurographics Symposium on Rendering, 321-326 (2007).

[4] Meylan, L., Daly, S., and Susstrunk, S., "The Reproduction of Specular Highlights on High Dynamic Range Displays," Proceedings of the 14th Color Imaging Conference (2006).

[5] The HDR Photographic Survey: Thumbnails. www.cis.rit.edu/fairchild/HDRPS/HDRthumbs.html

[6] "Forcing a Least Squares Polynomial Through a Fixed Point," http://www.physicsforums.com/showthread.php?t=523360