Convex Optimization Final Project: Recover Radiance Map from Photographs and Tone Mapping Methods for Display

Chia-Kai Liang*
Graduate Institute of Communication Engineering,
National Taiwan University

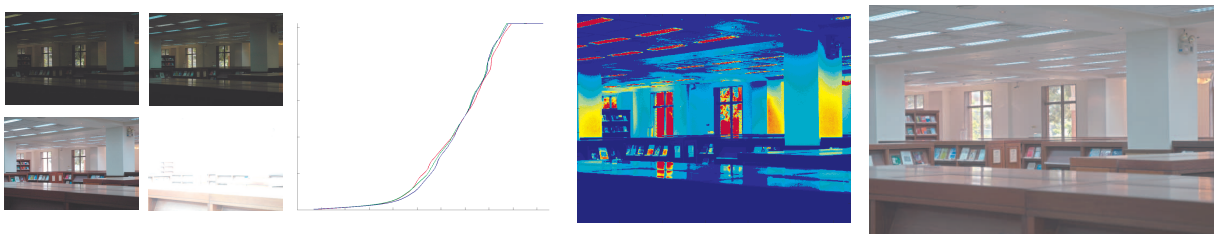


Figure 1: (From left to right) *Given a number of photos, we can recover the response function of the camera and the irradiance map of the scene. Then we can generate a more vivid picture using novel tone mapping algorithms.*

Abstract

This project gives a thorough survey on high dynamic range imaging, which is an emerging research area. The project can be segmented into three parts. First, a system which recovers the irradiance map from photographs with different exposure times is implemented and improved. Second, we implement two tone mapping algorithms. Besides implementation, we make perceptual and theoretic comparisons on them. Last but not least, we formulate the objective and constraint functions of the tone mapping system, and try to obtain an optimal tone mapping result using convex optimization. Many techniques learnt in the course are applied in this project.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—High Dynamic Range Imaging; I.4.0 [Image Processing and Computer Vision]: General—Tone Mapping;

Keywords: high-dynamic-range imaging, tone mapping, computational photography

1 Introduction

Human see the real world with a wide range of intensity and color. For example, the average luminance of the starlight is around 10^{-3} cd/m^2 and the scene under sunshine is 10^5 cd/m^2. However, the maximum luminance value of the current display devices is only 150 cd/m^2 and the contrast ratio is no more than two orders of magnitude. Also the traditional image capture devices are designed for these displays, and thus the dynamic range of the recorded images is much lower than that of the luminance in the real world.

Recovering the scene with full dynamic range is called *high dynamic range imaging*, and the technique to present high-dynamic-range images on the current display is called *tone mapping*. This final project targets on these two topics and the applications of linear programming and convex optimization.

This report is organized as follows. In Section 2 we briefly describe the image acquisition pipeline in the camera and then we show how to reverse this process. In sections 3 we give the implementation details and experimental results of two popular tone mapping algorithms. The tone mapping method we proposed is demonstrated in Section 4. The conclusions are drawn in Section 5.

2 Radiance Recovery

Figure 2 shows the general processes of a digital camera. We describe those processes one-by-one and then we present a efficient method to recovery the radiance map from the final digital values. Similar analysis and technique can be applied to traditional cameras.

2.1 Image Acquisition Pipeline

The front end of the camera is the optical systems composed of a number of lens. The optical system determines the focus range, depth of field, and the field of view of the picture. For the sensor element, either CCD or CMOS, the output voltage is proportional to the number of photons pass through the optics and hit on the sensor. However, its upper voltage is limited. Therefore, a shutter is placed before the sensor array to control the amount of irradiance. In other words, the setting of the exposure time determines the dynamic range we desire to present in the picture, which is usually smaller than the real dynamic range of the scene.

After the sensor collects charges for a period of time (the exposure time), the output voltage is input to analog-to-digital converter (ADC). In commercial cameras the quantized bit number usually ranges from 12 to 16 bits, which still preserves a large dynamic

*Student ID: f93942031 e-mail: f93942031@ntu.edu.tw

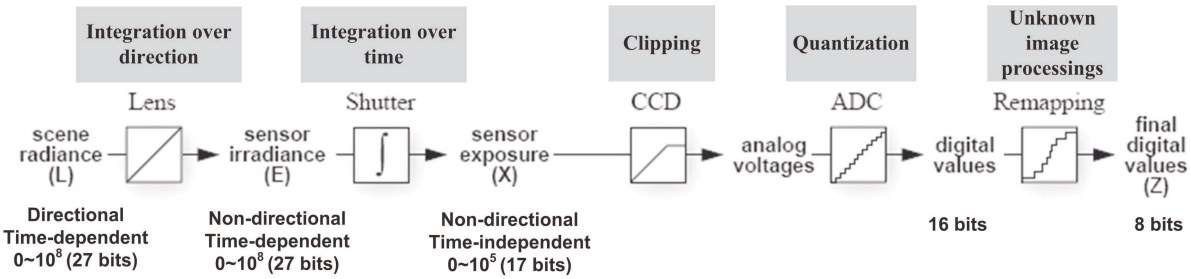


Figure 2: Image acquisition pipeline. The bottom line shows the required bit-length to represent full-ranged data.

range. However, *because cameras are designed to generate displayable and pleasurable pictures*, a number nonlinear functions are applied, such as gamma correction and white balance, and the final output is quantized to 8-bits. Because these functions are sequential and independent, we can model all of them as a remapping function. The nonlinearity of the remapping function breaks the additive property of the luminance. That is, if one pixel is twice larger than the other, it does not mean that the actual luminance is twice as well. This phenomenon destroys many assumptions in computer vision algorithms, such as shape from shading and motion estimation.

2.2 The Algorithm

Cameras transform the density of the photons to digital value through a number of processes, and the whole system can be modeled as a response function. Debevec and Malik proposed a simple but efficient method to estimate the response function from irradiance value to digital value, and further recover the irradiance map [1997]. Here we present the concept of the algorithm, and the implementation details are given in the next section.

The value X that CCD sensor measures is the integration of the irradiance E over the exposure time Δt. Assume that the scene is still, and thus the irradiance is constant over time, X becomes the product of E and Δt:

$$X = E \cdot \Delta t$$

As we mentioned before, we can model all processes of the camera as a response function of X:

$$Z = f(X) = f(E \cdot \Delta t)$$

Assume f is monotonic and invertible, we have

$$f^{-1}(Z) = E \cdot \Delta t$$

Taking the logarithm on both sides, and we denote g as $\ln f^{-1}$, we have the following equation:

$$g(Z) = \ln f^{-1}(Z) = \ln(E \cdot \Delta t) = \ln E + \ln \Delta t. \quad (1)$$

Since Z are the final pixel value and Δt is specified by the user, the problems we are interested in are estimating g and recovering E. In normal cases, this problem is unconstrained. For example, given an image with N pixels, we have N observed Z and N equalities like (1). However, we have N unknown E plus an unknown function g, so the observations are not enough.

We can overcome this problem in many aspects. First, although g is inherently a continuous function, the possible input values of Z are only integers and range from 0 to 255. As a result, we only need to recover a finite number of $g(z)$. Second, we can increase the number of observations by taking many photographs with different exposure times. Because we assume the irradiance E is constant, the unknown variables does not increase with the pictures we take. Denote the P as the number of photographs, we have:

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j, j = 1 \ldots P \quad (2)$$

where i is the index of pixels and j is the index of pictures. The problem is over-constrained as long as

$$N \cdot P > 256 + N. \quad (3)$$

Now we can formulate the problem explicitly. A good estimation of the response function g makes the following objective function as small as possible:

$$\sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2, \quad (4)$$

where the constraint is that g should be monotonic and smooth. Because this constraint is implicit and not always true, we relax it by introducing a *Lagrange multiplier*, and Equation (4) becomes

$$\sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=1}^{254} g''(z)^2. \quad (5)$$

After g is estimated, we can calculate the original irradiance value according to Equation (2):

$$\ln E_i = \text{avg}(g(Z_{ij}) - \ln \Delta t_j). \quad (6)$$

2.3 Implementation Details and Results

Although Equation (5) seems a reasonable objective function, we can refine it to take account of real scenarios. First, due to the presence of noise in the captured images, Equation (1) does not always hold. Second, we know that the upper output voltage is limited, so the assumption that $g(z)$ is monotonic and smooth does not hold when z is near 255. These two problems can be alleviated by introducing a regularization function:

$$w(z) = \begin{cases} z+1 & \text{if } z \leq 128 \\ 256 - z & \text{if } z > 128 \end{cases} \quad (7)$$

and Equation (5) becomes:

$$\sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=1}^{254} [w(Z_{ij}) g''(z)]^2. \quad (8)$$

Also the irradiance calculation can be refined into:

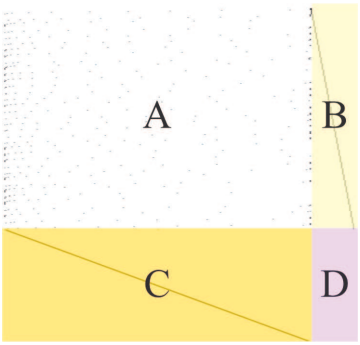


Figure 3: Matrix of our problem is composed of four matrices. Three of them are sparse (A, B, C) and one is zero (D).

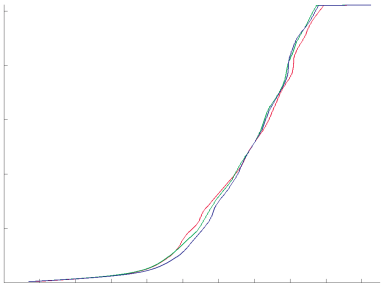


Figure 4: The response curve of Nikon D70. The x-axis is the logarithm of the input irradiance, and y-axis is the output pixel value. The color of each curve represents the color channel it corresponds.

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}. \quad (9)$$

Equation (7) works for the following reasons. We assume the noise level is spatially constant, so the observations of small z are less reliable. The regularization function can decrease the importance of these observations. Also the smoothness requirement of large z is weakened by the regularization function.

Since every term in Equation (8) is a 2-norm, the problem is a least-square-error minimization problem, and can be solved using Singular Value Decomposition (SVD) [Debevec and Malik 1997]. We implemented the SVD solver according to [Press et al. 1992]. It takes about 10 seconds to recover the response curve, which is too slow for interactive applications.

We find that matrix is very regular and sparse, as shown in Figure 3. Therefore we can speed up the solver by matrix decomposition. We apply a sparse LDL^T factorization and the running time is decreased by a factor of 5. We observe that the matrix can be segmented into four regions and they all have their own characteristics. Therefore block eliminating can be performed, but we do not observe any further acceleration.

The response curve of Nikon D70 is shown in Figure 4. The input data are 12 photographs with different exposure times. According to Equation (3), $N = 44$ pixels is quite enough to solve the problem. Here we pick these pixels randomly. The recovered irradiance map is shown in Figure 5.

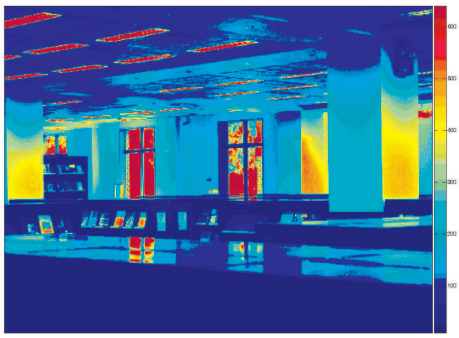


Figure 5: Irradiance map of a scene in NTU library. The maximum illumination is 741.85 cd/m^2, the minimum is 0.17 cd/m^2, and the contrast ratio is 4451:1.

3 Tone Mapping Algorithms

After the irradiance map and the response function are obtained, given a exposure Δt, we can generate a virtual photograph which follows the response function of the original camera. However, we can do much more than this. In this section, we present the basic principles of tone mapping, and then we describe two popular methods.

3.1 Principles

We need a transformation to display irradiance map on the normal display, just like the remapping function in the camera. Because the human do not sense the brightness in proportional to the irradiance, the transform function need to take human visual system (HVS) into account. This technique built in the camera or display is called tone mapping, and the transform function we deal with is a new category in it. The major difference to the traditional methods is that the input is the recovered irradiance map, while the old techniques are dealing with normal images. This research area is getting more and more attention in recent years. Many different approaches, either from digital signal processing, machine learning, or biological engineering, have been proposed [Ashikhmin 2002; Barladian 2003; Duan et al. 2004; Durand and Dorsey 2002; Fattal et al. 2002; Ledda et al. 2005; Li et al. 2005; Reinhard et al. 2002].

The reason why traditional methods cannot work is that they usually apply a global function on the whole image, which is only a rough approximation of HVS. The parameters of the global function are based on some statistics of the whole image, such as minimum, maximum, and average irradiance values. Although they work well in cameras and displays, they usually fail when applying to high-dynamic-range irradiance maps.

In fact, the dynamic range that a retina cell can sense is slightly larger than that of a sensor, but usually we can perceive full dynamic range of the scene, from grains in the shadow to reflections of the sunlight on metal. Therefore, to know more details of how our eyes work shall guide us how to design a good tone mapping algorithm.

Beside the global transform function, which is about a logarithm of the irradiance, human can adjust the response gain of each nervous adaptively. In dark regions, the received signal is boosted while it is compressed in bright regions. However, things are much more complicated than that.

What hard to define is *the regions of the scene*. Although human naturally have this ability to some extent, we keep *learning* how to identify the regions since childhood. For example, if there is a variation of illumination between two points, it does not imply they belong to different regions. It can be the perturbation from

texture, or a gradual transition of reflectance. In these cases, human do not think they belong to the different regions, and response gain of these two points are the same. Therefore, the variation of brightness between them is preserved. On the contrary, when the ground is shadowed by a passenger, although we know the shadowed and non-shadowed grounds are the same object, the brain automatically treat them as different regions, and increase the response gain of the shadowed region. Therefore we can see the grains in both regions.

To sum up, if we want to have a tone mapping algorithm that mimic our eyes, we need to achieve these goals:

Global Contrast Reduction The dynamic range should be no larger than that of the display. However, if the reduction ratio is too high and thus only a partial dynamic range is utilized, the image would look pale.

Local Details Preservation. While the global contrast is reduced, the local perturbation among pixels within the same region must be preserved. One useful judgment is *Just Noticeable Difference* (JND). That is, if the difference of brightness between two pixels is larger than 0.1 times either of them, human would definitely distinguish these two pixels. Therefore tone mapping algorithms should preserve any pixel difference larger than JND.

In this project, we focus on two popular tone mapping algorithms, gradient domain compression and bilateral filtering [Fattal et al. 2002; Durand and Dorsey 2002]. There are many reasons we choose to implement them. First, they are mathematically well-formulated in their own aspects, while many algorithms rely on many heuristic principles or empirical parameters, such as [Ashikhmin 2002] and [Reinhard et al. 2002]. Second, one step in gradient domain compression is solving a Poisson equation, which is a least-square problem, and thus we can apply what we have learnt in the course. These two methods may not be the best ones up to date, but this statement depends on the way we judge the tone mapping method [Ledda et al. 2005]. While many tone mapping algorithms target on generating a image as close as what we see the real world, these two methods usually enhance the visual quality. We believe in most cases the latter judgment is more useful, especially for aesthetic applications.

3.2 Gradient Domain Compression

From the above analysis we conclude that human is more sensitive to the local changes of irradiance rather than its absolute value, since it is modified by the response gain. Therefore we should manipulate the first-order derivatives of the signal, and reconstruct the result.

For 2-D images, first-order derivatives become a gradient field:

$$\begin{aligned}\nabla I(x,y) &= \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \\ &\approx \left(\frac{I(x+1,y)-I(x-1,y)}{2}, \frac{I(x,y+1)-I(x,y-1)}{2}\right),\end{aligned} \tag{10}$$

where the second line is the central difference approximation the continuous gradient field.

Our goal is to reduce large change in the image while preserving the details. This is achieved by designing a attenuating map Φ to the gradient field:

$$G(x,y) = \Phi(x,y)\nabla I(x,y). \tag{11}$$

In [Fattal et al. 2002], Φ is a function of $\nabla I(x,y)$:

$$\Phi(x,y) = \frac{\alpha}{\|\nabla I(x,y)\|}\left(\frac{\|\nabla I(x,y)\|}{\alpha}\right)^{\beta}. \tag{12}$$

The gradients larger than α are attenuated ($\beta < 1$) and the gradients smaller than α are boosted. We follow the description in [Fattal et al. 2002] and set α to 0.1 times the average gradient magnitude and β to 0.85.

However, the modified gradient field G in not always integrable. In fact we can reconstruct an image from a gradient field if and only if its *curl* is zero:

$$\frac{G_x(x,y)}{\partial y} - \frac{G_y(x,y)}{\partial x} = 0. \tag{13}$$

Since the reconstruction using integration is not feasible, we change our target: *find an image R such that its gradient field is most similar to G* [Fattal et al. 2002]:

$$\min \sum\sum F(\nabla R(x,y), G) = \min \sum\sum \|\nabla R(x,y) - G(x,y)\|^2. \tag{14}$$

If R minimizes (14), it must satisfy Euler-Lagrange equation:

$$\frac{\partial F}{\partial R} - \frac{d}{dx}\frac{\partial F}{\partial R_x} - \frac{d}{dy}\frac{\partial F}{\partial R_y}. \tag{15}$$

Rearrange terms we obtain a Poisson equation:

$$\nabla^2 R = \mathrm{div} G, \tag{16}$$

where ∇^2 is the Laplacian operator $\nabla^2 R = \frac{\partial^2 R}{\partial x^2} + \frac{\partial^2 R}{\partial y^2}$ and div$G$ is the divergence of G. The discrete approximations of these two operators are:

$$\nabla^2 R \approx R(x+1,y)+R(x-1,y)+R(x,y+1)+R(x,y-1)-4R(x,y), \tag{17}$$

$$\mathrm{div} G \approx G_x(x,y) - G_x(x-1,y) + G_y(x,y) - G_y(x,y-1). \tag{18}$$

Therefore the problem become a formal least-square problem $Ax = b$. x is R in vector form, each element of b is a divergence (18), and each row of A contains five nonzero components (17). Because R is a displayable image, each value must be equal or larger than zero. Boundary conditions are applied the signals around the boundary. Either Neumann or Dirichlet conditions are proper.

The intuitive methods to solve this problem are using Singular Value Decomposition, or we can treat it as a feasibility problem and apply simplex method. However, the computational costs of these two methods are unaffordable because of the size of x. An image of normal size has $640 \times 480 = 307,200$ pixels, and resolution of modern cameras is much higher than that. In this case, A is a matrix of size $307200 \times 307200 \cong 94.37$G, which is far beyond the capability of desktops. Even for an 160×120 image, it requires several minutes to reconstruct the image.

We can see that the structure of the matrix A is very regular. Besides rows corresponding to boundary conditions, the diagonals are all -4, and the other four 1s are spread in diagonal. To be efficient, we can split A as:

$$A = L + D + U, \tag{19}$$

where D is the diagonal part of A, L is the lower triangle of A and U is the upper triangle. Since the diagonal part dominates the system, we can use descent method and simply take the descent vector as

$$x^t = -D^{-1}(L+U)x^{t-1} + D^{-1}b. \tag{20}$$

Because D is a diagonal matrix, the evaluation of D^{-1} is very fast.

To further reduce the computational cost, *multi-grid method* can be applied. It is a numerical method, and we only explain the concept here. When a gradient field is sampled on regular grid, we can down-sample it and solve a simpler Poisson equation. Then the results and residual are interpolated to the original size, and then we have a better start point for iteration. Using infinite levels, the

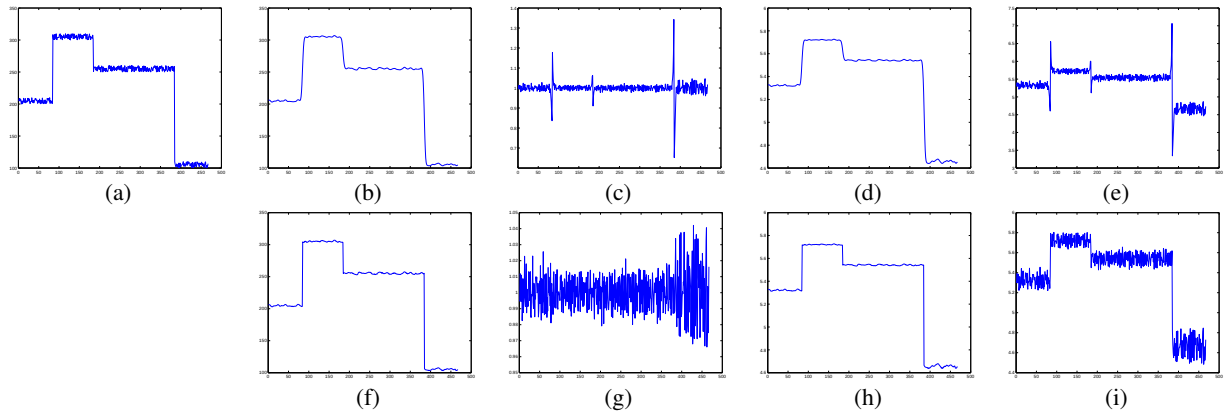


Figure 6: Example of bilateral filtering. (a) Original signal. The dynamic range is 300. Details are random noise with amplitude 10. (b) Base layer using Gaussian filter. The transitions between regions are blurred. (c) Detail layer with (b). (d) Compression of (b). The dynamic range is 5.7. (e) Combined signal, the signals between regions are erroneous. (f) Base layer using bilateral filter. (g) Detail layer with (f). (h) Compression of (f). (i) Combined signal. The dynamic range is successfully reduced and sharp transitions are preserved.

theoretical complexity will approach $O(N)$, where N is the number of samples.

Another fast Poisson solver is to convert the signal and operators to frequency domain, perform de-convolution, and transform the resulted signal back to spatial domain. Due to the efficiency of Fast Fourier Transform (FFT), the complexity of this method is about $O(N\log N)$. On our platform it takes about 8 seconds to solve a Poisson equation of size 640×480. Although it is theoretically slower than multi-grid method, the actual processing times are similar. We think it is because Matlab highly optimizes FFT.

3.3 Bilateral Filtering

Bilateral filtering is a signal processing technique and has been applied to many problems. We implemented Durand and Dorsey's work [2002] just for comparison with gradient domain compression. This work is not directly related to the course, and thus only general concepts will be explained.

As we mentioned, the tone mapping algorithm should process the irradiance transition of region boundaries and details in different ways. Therefore, we wish to decompose the image into two signals, where one of them contains the average irradiance value of each region, and the other contains details only. We can reduce the dynamic range of the formal signal, and leave the latter unchanged. These two signals are then combined and the dynamic range reduction is achieved while local details are preserved.

The basic idea is that because transitions between regions are usually much larger than details. Therefore we can remove details by a low-pass filter:

$$I_{base}(\mathbf{i}) = \sum_{\mathbf{j}\in N} w_s(\mathbf{j}-\mathbf{i})I(\mathbf{j}) / \sum_{\mathbf{j}\in N} w_s(\mathbf{j}-\mathbf{i}), \tag{21}$$

where the coefficient w_s is a function of the distance between $\mathbf{i}$ and $\mathbf{j}$, and N denotes the filter size. In our implementation, a Gaussian filter is applied. The mean is zero and its variance is adjusted manually.

The filtered signal is the *base layer* of the original image, and we can also obtain the *detail layer* by dividing the original signal by the base layer. A 1-D example is shown in Figure 6(b) and (c).

However, applying a low-pass filter will smooth everything, even the sharp transition between regions. Therefore, besides real details, the detail layer would contain the high frequency components of the base layer, and they are preserved while low frequency components are attenuated. The result is that the transitions between regions will be reversed, which is called *halo effect*, as shown in Figure 6(e).

This problem occurs because the filter coefficients in (21) only depend on spatial distance between signals, and the intensity distance is ignored. Bilateral filtering solves this problem by modifying the filter coefficients:

$$I_{base}(\mathbf{i}) = \sum_{\mathbf{j}\in N} w_s(\mathbf{j}-\mathbf{i})w_l(I(\mathbf{j})-I(\mathbf{i}))I(\mathbf{j}) / \sum_{\mathbf{j}\in N} w_s(\mathbf{j}-\mathbf{i})w_l(I(\mathbf{j})-I(\mathbf{i})), \tag{22}$$

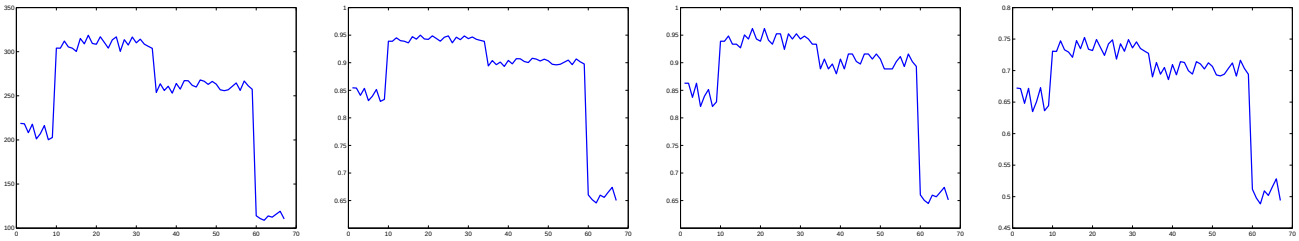
where w_l is a function of the intensity difference. Again a Gaussian filter with zero mean is applied. The 1-D example is given in the second row of Figure 6. It is simple to extend this concept to 2-D images, and the dynamic range reduction of the base layer is simply a global transform function.

3.4 Results

Two implemented methods are applied to many photographs, and some results are shown in Figure 7 - 10. The original dynamic ranges of these images are all above four degrees of magnitude. There are some parameters in both methods which can be adjusted to optimize the result, but we fixed them for fair comparison. Basically both methods successfully reduce the dynamic range, and the details are more visible than using a fixed exposure time (Small images in Figure 9 and 10).

Both methods have their own defects. Bilateral filtering may lose some details, like the patterns around the skylight in the image Memorial Church. Also halo effect still arises. For example, in Figure 10, the boundaries of the windows are brightened.

As for gradient domain compression, one defect is that although large transitions are attenuated, small transitions are boosted. Therefore some regions with small illumination variation become ragged, such as the stair steps in Memorial Church and clouds in Vine Sunset. Also because the image is reconstructed from the gradient field in a least-square-error sense, reconstruction errors occur everywhere. For example, in a flat region, small errors may still arise and perturb this region, like the right wall in Figure 10(a).

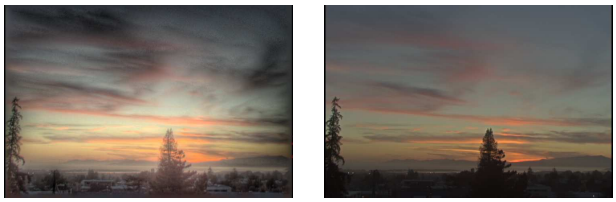


(a) (b) (c) (d)

Figure 11: One result of the tone mapping using convex optimization. (a) Original signal. The dynamic range is 300. Details are random noise with amplitude 10. (b) Result using global operator in [Reinhard et al. 2002]. The details are compressed to below just noticeable difference. (c) Result of the proposed method. The details are enhanced. (d) Result of bilateral filtering. The dynamic range of (b)-(d) are 1.

(a) (b)

Figure 7: Tone mapping result of Memorial Church. (a) Gradient domain compression. (b) Bilateral filtering. Image source: Paul Debevec.



(a) (b)

Figure 8: Tone mapping result of Vine Sunset. (a) Gradient domain compression. (b) Bilateral filtering. Image source: Paul Debevec.

(a) (b)

Figure 9: Tone mapping result of NTU Library. (a) Gradient domain compression. (b) Bilateral filtering.

4 Tone Mapping using Convex Solver

The methods described in the previous section do not consider tone mapping as an optimization problem. They design their systems based on the knowledge of HVS, and when the performance is not good enough, users have to adjust the parameters manually.

We can solve this problem from another point of view. We want to preserve the visibility of the details in the image, while the dynamic range of the display is limited. If we can formulate these two constraints in convex form, and we also have a good object function, we can solve the problem to achieve tone mapping. Duan et al. proposed a similar concept [2004], where they formulate the tone mapping as a vector quantization problem, and apply machine learning method to obtain the optimal quantization table.

We think it would take huge computational power to obtain a optimal solution for 2-D signals, therefore in this project we simplify the problem to 1-D signals. Similar to HVS, we want to obtain a gain map $G(x)$ such that:

$$I'(x) = G(x)I(x). \tag{23}$$

I' is the tone mapped signal, which is limited by the display (assume the range is normalized to [0 1]):

$$I'(x) > 0, I'(x) < 1.0. \tag{24}$$

Now we add the visibility constraint based on just noticeable difference. Within a region, if the difference between two pixels is larger than 1% of their average value, the tone mapped signal should preserve this difference. The constraint is:

$$\begin{aligned}&(I(x)-I(y)) > 0.01(I(x)+I(y))/2\\ \Rightarrow\ &(I'(x)-I'(y)) > 0.01(I'(x)+I'(y))/2,\end{aligned} \tag{25}$$

and after rearrangement, it becomes

$$199 \cdot I(x)G(x) - 201 \cdot I(y)G(y) > 0. \tag{26}$$

Also we want to avoid the halo effect, so the gradient direction of the signal should be preserved:

$$\nabla I'_x / \nabla I_x = \frac{I(x+1)G(x+1) - I(x)G(x)}{I(x+1) - I(x)} \geq 0. \tag{27}$$

We think although we adjust the gain locally, it should not make the result very different to the one using global operator. This is

(a) (b)

Figure 10: Tone mapping result of NTU Library Front Gate. (a) Gradient domain compression. (b) Bilateral filtering.

achieved implicitly in previous methods. Here we make this assumption as our objective function. We choose the global operator applied in [Reinhard et al. 2002]:

$$I_d(x) = \frac{I_t(x)}{1+I_t(x)} \frac{1}{\max(I_t(x)/1+I_t(x))}, \tag{28}$$

where $I'(x)$ is

$$I_t(x) = aI(x)/I_w, \tag{29}$$

and I_w is

$$I_w = \exp\big(\sum_x \frac{1}{N} \log I(x)\big), \tag{30}$$

where a is set as 0.36 in our experiment.

We want the signal $I' = gI$ is as close as I_d while the constraints still hold. The distance is defined in 2-norm and all constraints are linear. Therefore, our problem is a *quadratic programming problem*.

We use YALMIP to solve this problem [Lofberg 2004]. It is obvious the number of constraints is large, and thus in the experiment the number of variables is limited to 100. It takes about 1 minute to set all constraints and 4 seconds to solve the problem. One result is illustrated in Figure 11. We can see the details lost by global operator are recovered. The result is very similar to bilateral filtering, but we do not need to adjust any parameter. We do not compare our method to gradient domain compression here since the results of gradient domain compression are usually quite different to the original signals.

Although the results look promising, the computational cost is an obstacle. We increase the problem size to 720 and YALMIP takes about 8 minutes to solve it. If the problem size is increased to 320×240, it will exceed the maximum iteration number of YALMIP and fail.

5 Conclusion

Given a series of photographs with different exposure times, we have recovered the response function of the camera, and the irradiance map of the scene. Matrix decomposition is applied to speed up this process.

Two tone mapping algorithms have been implemented. In gradient domain compression, a Poisson solver is implemented which greatly reduces the processing time. We also implemented bilateral filtering for comparison.

Finally, we have formulated the tone mapping problem as a convex problem. The objective function and the constraints are formulated based on HVS. We have evaluated this idea on 1-D high-dynamic-range signals. Although the contrast reduction is achieved, it is not readily practical due to complexity. Nevertheless, we show the possibility to simulate human visual system mathematically.

References

ASHIKHMIN, M. 2002. A tone mapping algorithm for high contrast images. *The proceedings of 13th Eurographics Workshop on Rendering*, 145–155.

BARLADIAN, B. 2003. Robust parameter estimation for tone mapping operator. *13-th International Conference on Computer Graphics and Vision*, 106–108.

DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 369–378.

DUAN, J., QIU, G., AND FINLAYSON, G. 2004. Learning to display high dynamic range images. *Second European Conference on Color in Graphics, Imaging and Vision* (April), 542–547.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. 21*, 3, 257–266.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Trans. Graph. 21*, 3, 249–256.

LEDDA, P., CHALMERS, A., TROSCIANKO, T., AND SEETZEN, H. 2005. Evaluation of tone mapping operators using a high dynamic range display. *ACM Trans. Graph. 24*, 3, 640–648.

LI, Y., SHARAN, L., AND ADELSON, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph. 24*, 3, 836–844.

LOFBERG, J. 2004. Yalmip : a toolbox for modeling and optimization in matlab. *2004 IEEE International Symposium on Computer Aided Control Systems Design* (Sept), 284–289.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY., B. P. 1992. *Numerical Recipes in C*, 2nd ed. Cambridge University Press.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph. 21*, 3, 267–276.

SMITH, K., KRAWCZYK, G., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2006. Beyond tone mapping: enhanced depiction of tone mapped hdr images. *Eurographics 2006* (Sept.).