A review of High Dynamic Range Imaging on Static Scenes

Berkin Abanoz, Meng Wang

Dec 12, 2008
Boston University
Department of Electrical and Computer Engineering
ECE Technical Report No. 2008-04

BOSTON UNIVERSITY

A REVIEW OF HIGH DYNAMIC RANGE IMAGING ON STATIC SCENES

Berkin Abanoz, Meng Wang

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

Dec 12, 2008
ECE Technical Report No. 2008-04

Summary

Dynamic Range, the ratio of maximum radiance to the minimum radiance, of a highly illuminated scene cannot be represented by conventional Low Dynamic Range (LDR) imaging systems. Therefore there is a need for a system which can capture all the information in these scenes and which can represent this information. High dynamic Range Imaging (HDRI) is a process which is composed of three parts, acquisition, recovery of radiance map, and displaying this High Dynamic Range (HDR) radiance map on LDR displays. The first part aims acquiring information about different parts of the scene, different details of the scene with different images. The second part aims fusing the details coming from different images into a single radiance map. The last part is a compression step where the HDR information of radiance map is compressed so that it can be represented on LDR displays.

This project mainly focuses on the second part, the recovery of radiance map. The purpose of the project is to review and to compare some of the methods in the literature for the recovery of radiance map. We also propose a new method which will combine different details for a scene in one single image without recovering the radiance map.

Contents

List of Figures

List of Tables

List of Images

1 Introduction

Dynamic Range (DR) , which can be referred as the ratio of maximum radiance to the minimum radiance, of a real life scene can be up to 500000:1, whereas DR for human eye is 10000:1 and for a LCD 1000:1. Therefore, High Dynamic Range (HDR) of a highly illuminated scene cannot be represented by conventional Low Dynamic Range (LDR) imaging systems. For example, in a sunny day it not possible to take an image which both captures the details in sunny areas and in dark shadowy areas. Either the sunny areas will be overexposed or the dark areas will be underexposed. Therefore there is a need for a system which can capture all the information in these scenes and which can represent this information. High dynamic Range Imaging (HDRI) is a process which is composed of three steps, acquisition, recovery of radiance map, and displaying this High Dynamic Range (HDR) radiance map on LDR displays.

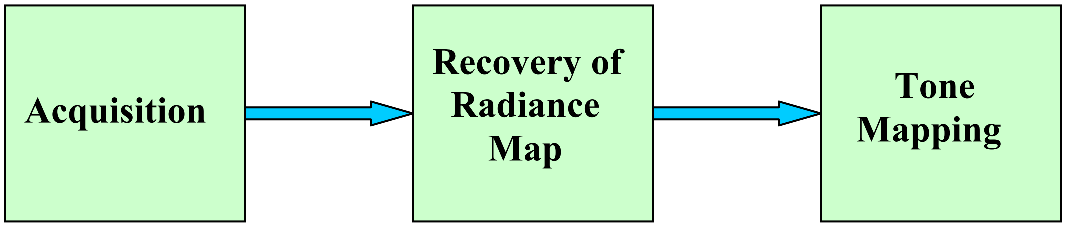


Fig. 1.1 HDRI Process

The acquisition process is taking making multiple images with different exposure times; therefore representing details about different parts of a scene. The second part aims fusing the details coming from multiple images with different exposure times into a single radiance map whose pixel values are proportional to true radiance values in the scene. Since the radiance map has a high dynamic range, we cannot display it efficiently on LDR displays. The last part, tone mapping, is a compression step where the HDR information of radiance map is compressed so that it can be represented on LDR displays.

2 Related Work

Scene radiance plays an important role in various kinds of computer vision problems since obtaining this information is required for explore the interaction of light with scene objects. Application such as shape from shading requires accurate scene radiance to determine the orientation of surface normal[13]; photometric stereo algorithms estimate the scene structure using the difference of radiance between various images[14]; color constancy algorithms separate the effect of illumination from the reflectance in the scene by study the change of radiance[15]; in the field of computer visualization, an accurate scene radiance permit realistic merging of computer rendered object and real scene objects[16].

Camera response function provides the necessary information we need to accurately reconstruct the scene radiance. For a specific camera, its camera response function indicates how will it response to the irradiance of the scene in terms of pixel value. Once the response function is known, we can recover the true irradiance of the scene [17]. A conceptually simple way to generate such a function would be camera calibration. Use camera to capture a set of known radiance values (I) and record the measured pixel values (Z). From these pairs (I, Z) of data, we can get a table that represents the function f. This is the basis of methods that are called chart-based, because they usually employ a calibration chart, such as the Macbeth chart [18], which include patches of known relative reflectance. Nevertheless, placing a chart in the scene can be inconvenient or difficult in the field. For instance, the response function will change when the camera is in another condition, say, aperture or temperature. Recording all the response function for any setting of a specific camera is infeasible unless we use simplified approximation of this function.

Methods that do not use calibration charts are called chartless. The chartless recovery of f from observed data has been extensively studied in recent years. Those techniques are more interested because of their flexibility and possibility to integrate calibration methods. Chartless methods resort to a collection of differently exposed images captured without any prior knowledge and restrictions of the radiances in the scene, and try to recover the function purely based on the pixel values that are recorded by the camera.

Main efforts of recovery of f have been focused on exploiting a sequence of images of the same scene taken at different exposures. By comparing corresponding intensity values between images in the sequence, Mann and Picard demonstrated constraints on f [19] which allowed them to determine the response at a sequence of points with the help of known exposure ratios between the images. Debevec and Malik also assumed the ratio of exposures is known, but their approach does not require a parametric form for the response [20]. Instead, they imposed a smoothness constraint on the response. Mitsunaga and Nayar assumed that the inverse response function can be closely approximated by a polynomial [21]. Then, they estimated the coefficients of the polynomial and the exposure ratios starting with a rough estimate using an iterative method which alternates between recovering the response and estimating the exposure ratios. Tsin et al. [23] and, separately, Mann [14] recovered the response and exposure ratios by combining the iterative approach from [21], with the nonparametric recovery in [20]. All chartless methods use the assumption that between a pair of images of a static scene, all irradiances change by the same exposure ratio base while recovering the response function [19], [20], [24], [21], [23]. Grossberg and Nayar use intensity mapping function, which describe how pixel value in one image maps to that in other images to find the response function. Their approach can decompose into two steps: 1. the recovery of the intensity mapping function from images and 2.the recovery of the combination of the response and exposure ratios using the intensity mapping function.

Another approach to make HDR image is directly combining images into high dynamic range synthesis image using fusion technique. This technique has been introduced to the problem of multi-exposure image fusion by Burt et al. since 1993 [9]. Various of fusion techniques such as spline fusion[8] and poisson blending [10] works visually very well for either image domain or gradient domain [11].

3 Overview

The project mainly focuses on the second step of HDRI process, the recovery of radiance map. This is basically modeling the process from scene radiance to LDR digital images, pixel intensities. The image acquisition pipeline for digital camera and and film camera is modeled by Figure 3.1 in [1]. The mapping from scene radiance to sensor irradiance is linear and if we consider short exposure time, then the integration in the shutter can be modeled by a multiplication,

$$X = E * \Delta t \tag{3.1}$$

where X is the sensor exposure and E is the sensor irradiance and Δt is the exposure time. Either you use a digital camera or a film camera, different nonlinear mappings can occur from sensor exposure to final digital values, pixel intensities. For example in film camera development process has an S shaped nonlinear response to light or scanning process can have a nonlinear response. In digital camera digitization, remapping, gamma correction can introduce nonlinear mapping. The goal is to model the overall nonlinearity by a single function, called camera response function, which is assumed to be monotonic or semi-monotonic. Therefore, by using inverse of this function, inverse camera response function, we can map pixel intensities to radiance map which is a scale factor of scene radiance.

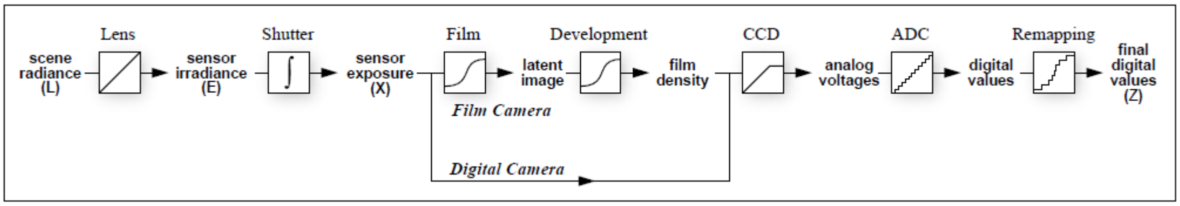


Fig. 3.1 Image Acquisition Pipeline

First of all, we will implement 3 different methods in the literature for determining the inverse camera response function and we will try t o compare these methods. Secondly, we will propose a third method to represent the all information in HDR scene without recovering the radiance map in a final single image.

4 Method 1

The first method is proposed by Debevec and Malik [1], it is a very popular method in HDR reconstruction and it is based on determining inverse camera response function using multiple pictures with different exposure time and their exposure time and with this function determining the radiance map.

The authors claim that the response of a film to variations in exposure can be summarized by the characteristic curve which is the graph of optical density D against the logarithm of exposure X. X is the product of irradiance E and the exposure time Δt. Let Z be the pixel intensity obtained after nonlinear operations such as development, scanning, digitization. Therefore, Z is a nonlinear function of the exposure X at the pixel.

Let $Z=f\ (\ X\ )$, then f is the camera response function and if we assume that f is monotonically increasing, its inverse is well defined, and $X = f^{-1}(\ Z\ ),$ where X is the sensor exposure and Z is the pixel intensity. The steps of the algorithm are first to compute logarithm of the inverse camera response function, and finally the sensor irradiance E, which is proportional to radiance in the scene L. For color photography, each color channel can be treated separately.

The algorithm assumes static scenes and it is based on pixel correspondences. In other words, the pixels with same position in different images correspondences to the same point in the scene. The inputs to the algorithm are a number of digitized photographs with different exposures and their exposure times Δt_j, and

$$Z_{ij} = f\,(\,E_i\,\Delta t_j\,) \tag{4.1}$$

where i is the spatial index and j is the exposure time index.

$$f^{-1}\,(\,Z_{ij}\,) = \ E_i\,\Delta t_j \tag{4.2}$$

$$ln\,f^{-1}\,(\,Z_{ij}\,) = ln\,E_i + ln\,\Delta t_j \tag{4.3}$$

Let $g = ln\,f^{-1}$ then

$$g\,(\,Z_{ij}\,) = ln\,E_i + ln\,\Delta t_j \tag{4.4}$$

The aim is to recover g and the irradiances E_i that best satisfy the set of equations arising from equation (4.4) in a least-squared error sense. Recovering g means recovering the finite set of values that $g(Z)$ can take, since Z can take finite number of values.

Let N be the number of pixel locations, P the number of photographs, then the problem becomes finding $(Z_{max} - Z_{min} + 1)$ values of $g(Z)$ and the N values of $ln\ E_i$ that minimizes the following quadratic objective function:

$$\theta = \sum_{i=1}^{N}\sum_{j=1}^{P}\left[g\left(Z_{ij}\right) - \ln E_i - \ln \Delta t_j\right]^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} g''(z)^2 \tag{4.5}$$

The first term in equation (4.5) ensures that the solution satisfies the set of equations arising from equation (4.4). The second term ensures the smoothness of g and g'' is given by,

$$g''(z) = g(z-1) - 2g(z) + g(z+1) \tag{4.6}$$

λ in equation (4.5) is chosen according to noise experimented in $Z_{ij.}$

Singular value decomposition is proposed to find the solution for the minimization.

In order to improve the solution, additional clarifications are proposed:

a) The solution for *g(z)* and E_i can be found up to a scale factor α, therefore they put an extra constraint, $g(Z_{mid})=0$, where $Z_{mid} = (Z_{min} + Z_{max})/2$.

b) In order to emphasize the smoothness and fitting terms towards the middle of the curve, a weighting function *w(z)* is proposed.

$$w(z) = \begin{cases} z - Z_{\min} & z \le Z_{mid} \\ Z_{\max} - z & z > Z_{mid} \end{cases} \tag{4.7}$$

This weighting function determines the confidence of pixel intensities. It gives zero weight to overexposed and underexposed pixels values.

Then the objective function becomes:

$$\theta = \sum_{i=1}^{N}\sum_{j=1}^{P}\left[w(Z_{ij})\left\{g\left(Z_{ij}\right) - \ln E_i - \ln \Delta t_j\right\}\right]^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1}\left[w(z)g''(z)\right]^2 \tag{4.8}$$

To ensure a sufficiently overdetermined system *N*(P-1)* should be greater than $(Z_{max} - Z_{min})$, hence to use every pixel location is impractical due to computational complexity. Pixels can be sampled from the regions of image with low intensity variance. The authors claim that the reason why they did not put a constraint for a monotonic *g* is that by their experience the smoothness constraint is enough to make *g* monotonic.

After recovering g, the irradiance values can be found in two ways, either by using a single image; $\ln E_i = g(Z_{ij}) - \ln \Delta t_j$, or by using all images and the weighting function;

$$\ln E_{ij} = \frac{\sum_{j=1}^{P} w(Z_{ij})\left(g(Z_{ij}) - \ln \Delta t_j\right)}{\sum_{j=1}^{P} w(Z_{ij})} \tag{4.9}$$

Using multiple exposures reduces the noise in the recovered irradiance values and reduces the effect of imaging artifacts. Additionally, since the weighting function ignores the saturated pixel intensities “blooming” artifacts have little impact.

5 Method 2

The second method is proposed by Mitsunaga and Nayar [2], as the first method this is also pixel based. They approximate the inverse camera response function by higher order of polynomial of degree *N*.

Let *E* be the sensor irradiance and *L* the scene radiance, then they can be related by,

$$E = L\frac{\pi}{4}\left(\frac{d}{h}\right)^2 \cos^4 \phi \tag{5.1}$$

where *h* is focal length, *d* diameter of aperture and ϕ angle subtended by the principal ray from the optical axis.

The linear radiometric response of the system can be expressed by,

$$X = L\,k\,e \tag{5.2}$$

where *X* is sensor exposure, $k = cos^4 \phi \;/ h^2$, $e = (\pi d^2/4)\Delta t$.

Many stages of image acquisition introduce nonlinearities. The pixel intensity can be related to the sensor exposure by,

$$Z = f(X) \tag{5.3}$$

Then *f* is the camera response function.

The unknowns are the scene exposures, hence the inverse camera response function $g = f^{-1}$ which will give $X = g\ (Z)$ should ne determined. The authors claim that any response function can be modeled by a higher order polynomial.

$$X = g(Z) = \sum_{n=0}^{N} c_n Z^n \tag{5.4}$$

To determine *g* means to determine the coefficients and the order *N*.

Assume there are two images taken with different exposures Δt_q, Δt_{q+1}, then the ratio of image exposures become

$$R_{q,q+1} = \frac{\Delta t_q}{\Delta t_{q+1}} \tag{5.5}$$

By using equation (5.11), the ratio of scaled radiance at any given pixel *p* becomes

$$\frac{X_{p,q}}{X_{p,q+1}} = \frac{L_p k_p \Delta t_q}{L_p k_p \Delta t_{q+1}} = \frac{\Delta t_q}{\Delta t_{q+1}} = R_{q,q+1} \tag{5.6}$$

The inverse response function can be related to exposure ratios by,

$$\frac{g(Z_{p,q})}{g(Z_{p,q+1})} = R_{q,q+1} \tag{5.7}$$

If the images are ordered such that $\Delta t_q < \Delta t_{q+1}$, then $0 < R_{q,q+1} < 1$, and the polynomial model for the inverse response function is substituted in equation (5.7), the following equation is obtained:

$$\frac{\sum_{n=0}^{N} c_n Z_{p,q}{}^{n}}{\sum_{n=0}^{N} c_n Z_{p,q+1}{}^{n}} = R_{q,q+1} \tag{5.8}$$

Note that if we use the function (5.8) to recover g and R jointly, from equation (5.7), we also have *$(g(Z_{p,q})/g(Z_{p,q+1}))^u$ = $(R_{q,q+1})^u$*. therefore, we can recover *g, R* pairs up to an exponent. If all measurements are normalized such that $0 \le M \le 1$, the indeterminable scale can be fixed using *f (1)=1*, and this can be used as an additional constraint to the algorithm.

$$c_N = 1 - \sum_{n=0}^{N-1} c_n \tag{5.9}$$

The algorithm will first assume that there is an initial good estimate for *R* and then makes the estimation of *R* part of the algorithm with an iterative process.

By using equation (5.8) then the objective function to minimize is

$$\varepsilon = \sum_{q=1}^{Q-1} \sum_{p=1}^{P} \left[\sum_{n=0}^{N} c_n M_{p,q}{}^{n} - R_{q,q+1} \sum_{n=0}^{N} c_n M_{p,q+1}{}^{n} \right]^2 \tag{5.10}$$

where *Q* is the total number of images and *P* is the number of pixel locations.

The coefficients can be found by solving the system of linear equations that result from $\frac{\partial \varepsilon}{\partial c_n} = 0$. Then, the coefficients can be used to update the ratios *$R_{q,q+1}$*,

$$R_{q,q+1}{}^{(k+1)} = \frac{1}{P} \sum_{p=1}^{P} \frac{\sum_{n=0}^{N} c_n{}^{(k)} M_{p,q}{}^{n}}{\sum_{n=0}^{N} c_n{}^{(k)} M_{p,q+1}{}^{n}} \tag{5.11}$$

The convergence condition for the algorithm is

$$\left| f^{(k)}(M) - f^{(k-1)}(M) \right| < eps, \forall M \tag{5.12}$$

To determine the order of polynomial an upper bound is determined and then the order which gives the minimum error ε can be chosen.

6 Method 3

The third method proposed by Grossberg and Nayar [3] is a two stage, parametric, histogram based method. In their paper, they define the intensity mapping function as the function that correlates the measured brightness values of two differently exposed images given by,

$$\tau(d) = H_2^{-1}(H_1(d))^3 \tag{6.1}$$

6.1 Intensity mapping

Grossberg and Nayar developed the intensity mapping theorem, saying that the histogram *h₁(u)* of one image, the histogram *h₂(u)* of a second image (of the same scene) is necessary and sufficient to determine the intensity mapping function τ." [3]

They develop an algorithm to recover the camera response function based on exposure ratios and the intensity mapping function. The recovery of the response decomposes into two parts: the recovery of the intensity mapping function from images, and the recovery of the combination of the response and exposure ratios from the intensity mapping function.

The authors state the following theorem: "Given the histogram of one image, the histogram of the second image is necessary and sufficient to determine the intensity mapping function." They prove this in three steps:

Normalize all images to have unit area. Compute the cumulative histogram for each image:

$$H(B) = \int_0^B h(u)du \tag{6.2}$$

Let τ be the intensity mapping function which maps each intensity B_2 in the second image to intensity B_1 in the first image: B_1 = τ (B_2). The set of image points in the first image with intensity less than B_1, must be the same as the set in the second image with intensity less than B_2 since they correspond to the same set of scene points. Therefore, $H_1(\tau(B_2)) = H_2(B_2)$

This means that the intensity mapping function and the first histogram determine the second histogram. If one replaces B_1 with u and solves for τ,

$$\tau(u) = H_2^{-1}(H_1(u)) \tag{6.3}$$

This completes the proof for the theorem. The algorithm can be summarized in the following steps: Compute the cumulative histograms for all images. Invert the cumulative histograms using linear interpolation. For each pair of consecutive images, compute intensity mapping function (by histogram specification).

6.2 Recovery of inverse response function

For recovering the inverse response function, the Grossberg and Nayar assume that the inverse response is a polynomial function [2]. The pairs (n/255, τ (n/255)) for $0 \leq n \leq 255$ are combined with the pairs (τ-1(n/255), n/255). Each pair (B1, B2) gave a constraint from

$$g\ (B_2) = k\ g(B_1) \tag{6.4}$$

where g is the inverse camera response function and k is the exposure ratio. The certainty about the estimated intensity mapping function depends on the amount of pixels in the images with these intensities. Hence, the pair *(B_1, τ (B_1))* with the number of pixels with value equal to B_1, in other words with $C = h1\ (B_1)$. To weight the least squares problem, the constraint

$$g\ (\tau\ (B)) = k\ g(B) \tag{6.5}$$

is multiplied by squared root of C. Similarly, the constraints for pairs *(τ-1(B2), B2)* with

$$W = \sqrt{h_2(B_2)} \tag{6.6}$$

g is assumed to be a sixth order polynomial and with all constraints together, a linear system for the coefficients for g is solved.

7 Method 4

For the specific purpose of HDR image synthesis, delicate calculation of irradiance of the scene is over complete for this task. Other ways to directly get image from input LDR images are proposed by many researchers since original work of Burt and Kolczynski [102]. Their approach targeted on combining different images and rendering an seamless image, regardless the true content and irradiance in the scene.

Here we propose a new way to accomplish this using independent component analysis. By carefully scrutiny of the different LDR images and camera response function found using previous methods, we find that in the middle exposed region, i.e. properly exposed areas, the content of the image change little, while the over exposed and under exposed areas have some scene objects visible in some LDR image and disappear (washed out) in other LDR image, we regard each LDR image is a mixture of different contents, some of which appear in one or more LDR images. Our goal is separate the different contents contained in LDR images, then combine together to synthesis HDR images. This view point of HDRI bring us a more principled way to analysis the underlying objects, which are under different lighting conditions.

Though ICA method is out of the focus of this paper, a basic idea of ICA is needed to understand why we choose this tool.

ICA is a computational method for separating a multivariate signal into additive subcomponents supposing the mutual statistical independence of the non-Gaussian source signals [104].The data is represented by the random vector

$$x = (x_1, \ldots, x_m) \tag{7.1}$$

and the components as the random vector

$$s = (s_1, \ldots, s_n) \tag{7.2}$$

The task is to transform the observed data x, using a linear static transformation W as

$$s = Wx \tag{7.3}$$

In the context of our problem, x_i represent individual LDR images and s_i the different exposure component. After we separate the exposure components, we use conventional fusion techniques to synthesis the final HDR images. Because in each exposure component, there are a lot of uniform area which is separated by ICA method, a proper threshold should be chosen to eliminate those uniform areas. Then we just sum those exposure components with a weight to render the HDR image. The weight is calculated according to the value on the rims of each exposure area.

8 Experimental Results

We used 4 different sets of images for testing the methods. The first set is the dataset including a series of images of the Stanford Memorial Church. This is the set which was used in [1] and it can be downloaded from [5]. The .png images were taken from PhotoCD scans of film pictures taken on Kodak Gold 100 ASA film and then the scans were decoded to 512x768 pixel resolution. The pictures were manually registered using feature points and homographies using the Facade photogrammetric modeling system. The pure blue color indicates shifted image border areas. The exposure times for the images are given in Table. 8.1.

Image Name	**1/exposure time (1/sec.)**
memorial0061.ppm	0.03125
memorial0062.ppm	0.06250
memorial0063.ppm	0.12500
memorial0064.ppm	0.25000
memorial0065.ppm	0.50000
memorial0066.ppm	1.00000
memorial0067.ppm	2.00000
memorial0068.ppm	4.00000
memorial0069.ppm	8.00000
memorial0070.ppm	16.00000
memorial0071.ppm	32.00000
memorial0072.ppm	64.00000
memorial0073.ppm	128.00000
memorial0074.ppm	256.00000
memorial0075.ppm	512.00000
memorial0076.ppm	1024.00000

Table 8.1 Exposure times for memorial image set

The second set is from [6]. This set also is composed of multiple images of the same scene taken with different exposure times. Their resolution is 1024x768 pixels. The exposure times are listed on Table 8.2.

Image Name	**exposure time (sec.)**
İmg01.jpg	13.00000
İmg02.jpg	10.00000
İmg03.jpg	4.00000
İmg04.jpg	3.20000
İmg05.jpg	1.00000
İmg06.jpg	0.80000
İmg07.jpg	1/3
İmg08.jpg	1/4
İmg09.jpg	1/60
İmg10.jpg	1/80

Table 8.2 Exposure times for second image set

The third image set is taken in the 9th floor of Photonics Building and the exposure times are given in Table 8.3. The resolution for this set is 2272x1704 pixels.

Image Name	**exposure time (sec.)**
P1010015.jpg	1/13
P1010016.jpg	1/8
P1010017.jpg	1/2

Table 8.3 Exposure times for third image set

The last set is also a series of images that we took. The resolution for this set is 640x480. The exposure times for this set is given in Table 8.4.

Image Name	**exposure time (sec.)**
P1010007.jpg	1/15
P1010008.jpg	1/8
P1010009.jpg	1/5
P1010010.jpg	1/3
P1010011.jpg	1

Table 8.4 Exposure times for fourth image set

The images in these four data sets are given in Appendix.

2.1 Method 1

We tested the first method on the memorial image set. Since we could not use all pixels in the images, we needed to eliminate some of them. Therefore, we first down sampled the images by 4 and then discarded the edges detected by sobel edge detection of matlab. Then, we calculated the g function for each channel using the memorial images, “memorial0063.ppm, memorial0064.ppm, memorial0065.ppm, memorial0066.ppm” with lambda value equal to 1000. The g function calculated for each channel is shown in figure 8.1. Then, we used 9 images to calculate the radiance map, “memorial0062.ppm, memorial0063.ppm, memorial0064.ppm, memorial0065.ppm, memorial0066.ppm, memorial0067.ppm, memorial0071.ppm, memorial0072.ppm, memorial0073.ppm”.

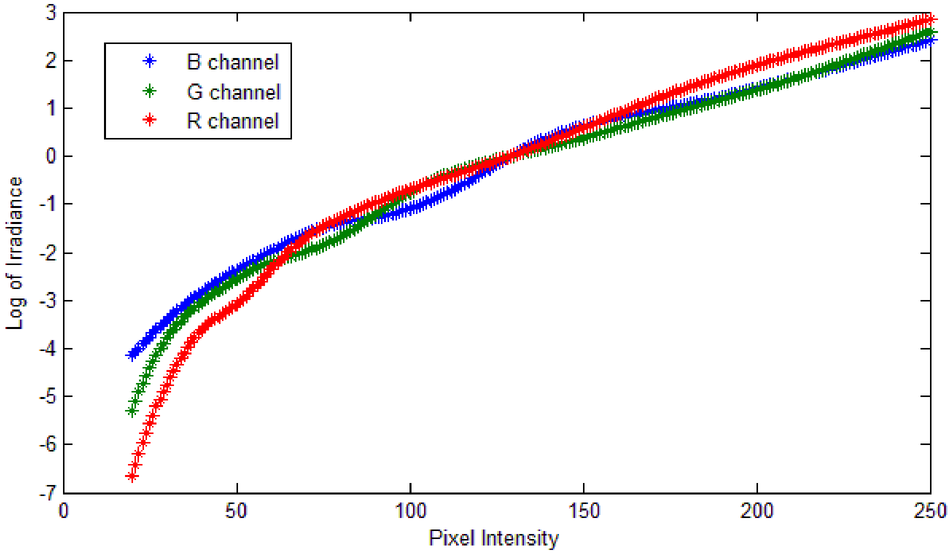


Figure 8.1 G function of method 1 for each channel for memorial set

Since this method can use multiple images to calculate the g function and we could use 9 images to calculate radiance map, the dynamic range of the radiance map was very high. Therefore, we used two different tone mapping methods to display the radiance map. The first one linear tone mapping and the second one is histogram adjusted tone mapping [7] .

Linear tone mapping uses the equation:

$$d = \frac{\left(w - w_{\min}\right)}{R_w} R_d + d_{\min} \tag{8.1}$$

where $R_w = \left(w_{\max} - w_{\min}\right)$ and $R_d = \left(d_{\max} - d_{\min}\right)$, R representing the radiance map values, d representing the pixel intensities that we are mapping radiance map to.

Histogram adjusted tone mapping uses the equation:

$$d = P(w)R_d + d_{\min} \tag{8.2}$$

where $P(w) = \frac{H(w)}{H(w_{\max})}$, *H(w)* is the cumulative histogram.

These mappings are applied to all channels together, not separately.

The resulting images, for linear tone mapping and for histogram adjusted tone mapping are image 8.1 and image 8.2, respectively.

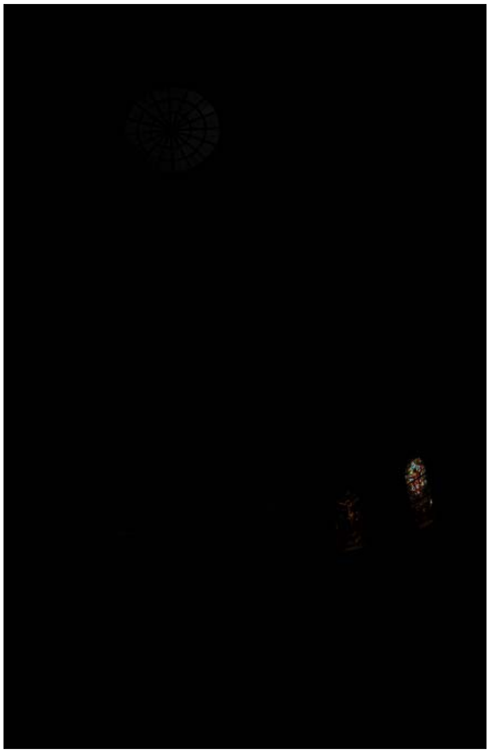


Image 8.1 Linear tone mapped radiance map of method 1 for memorial set

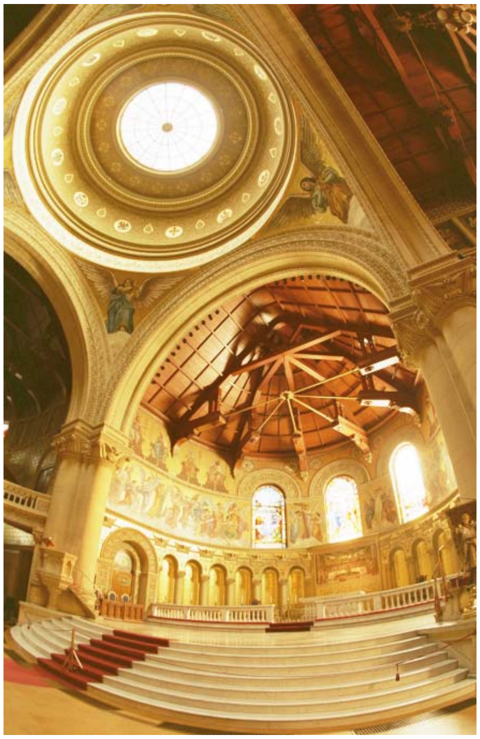


Image 8.2 Histogram adjusted tone mapped radiance map of method 1 for memorial set

We also tested method 1 on images that we took; third data set and fourth data set. The g function for each channel for third data set is given in Figure 8.2 and for the fourth data set in Figure 8.3. The resulting images from linear tone mapping and from histogram tone mapping are image 8.3 and image 8.4 respectively for third data set, image 8.5 and image 8.6 for the fourth data set.

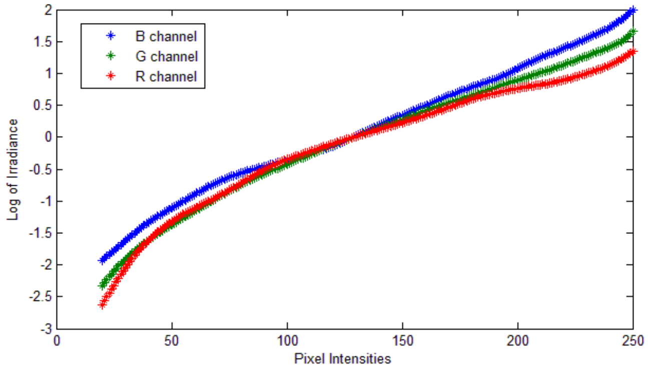


Figure 8.2 G function of method 1 for each channel for third set

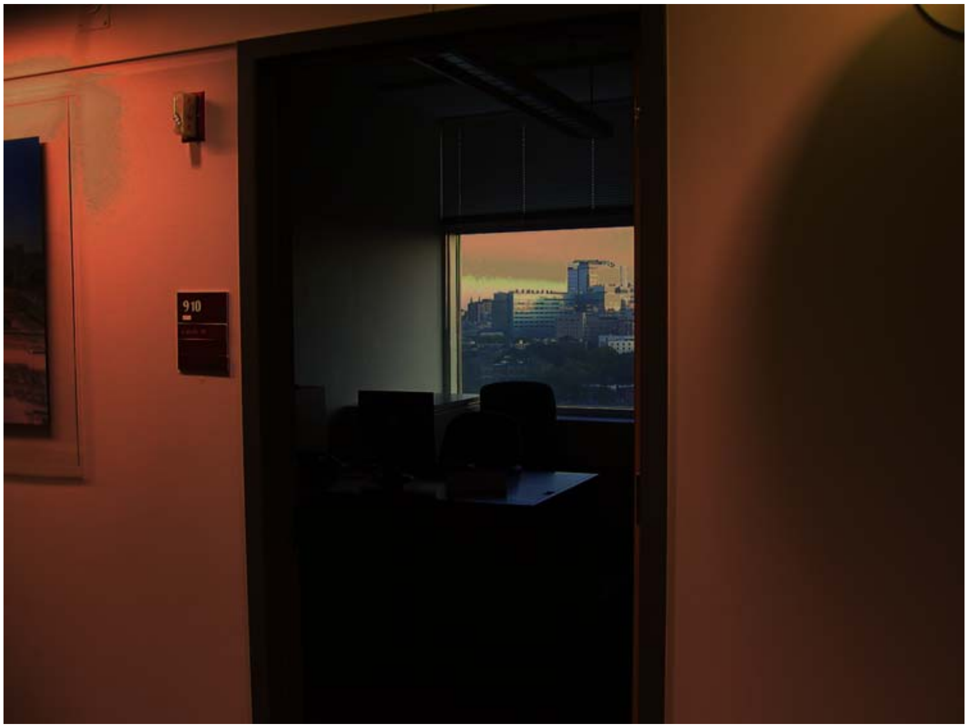


Image 8.3 Linear tone mapped radiance map of method 1 for third set

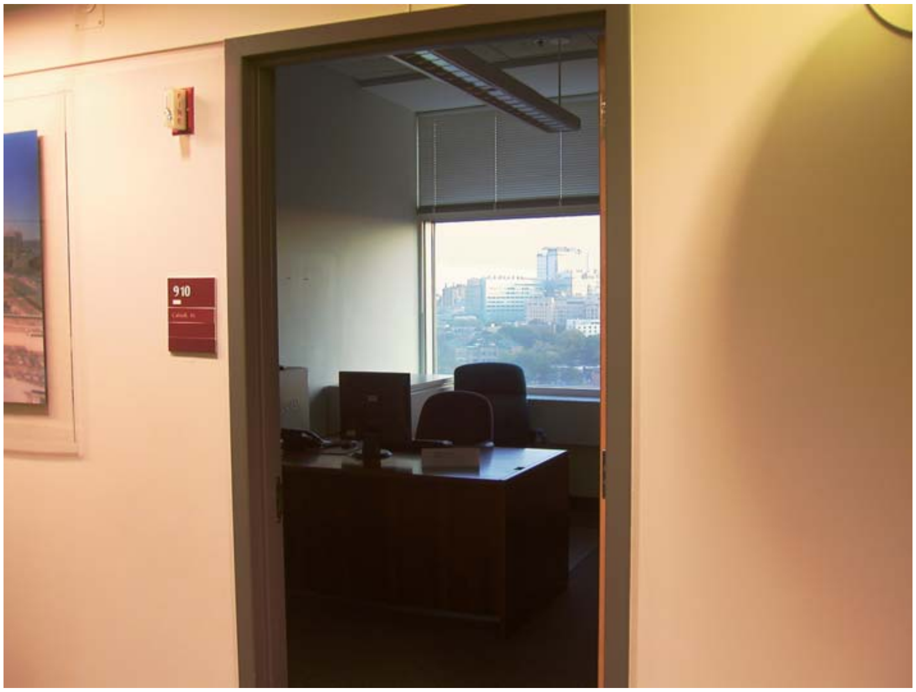


Image 8.4 Histogram adjusted tone mapped radiance map of method 1 for third set

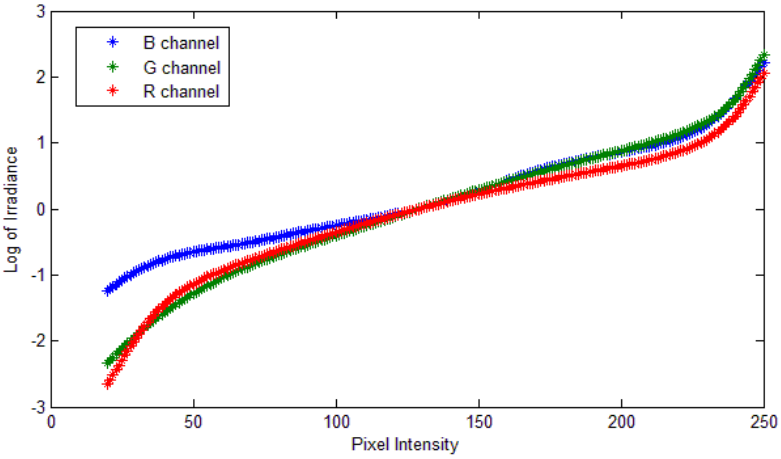


Figure 8.3 G function of method 1 for each channel for fourth set

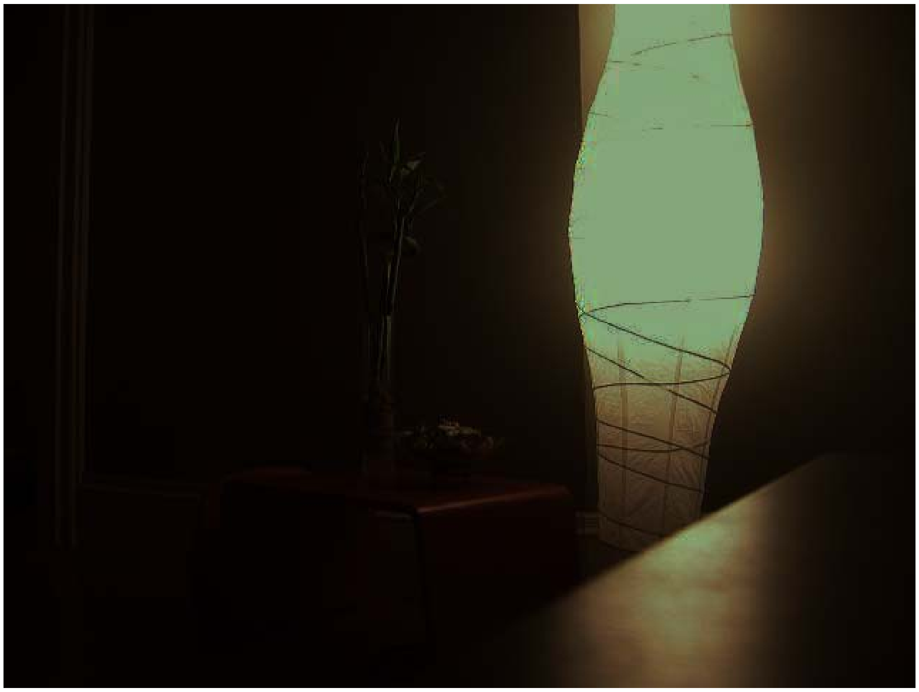


Image 8.5 Linear tone mapped radiance map of method 1 for fourth set

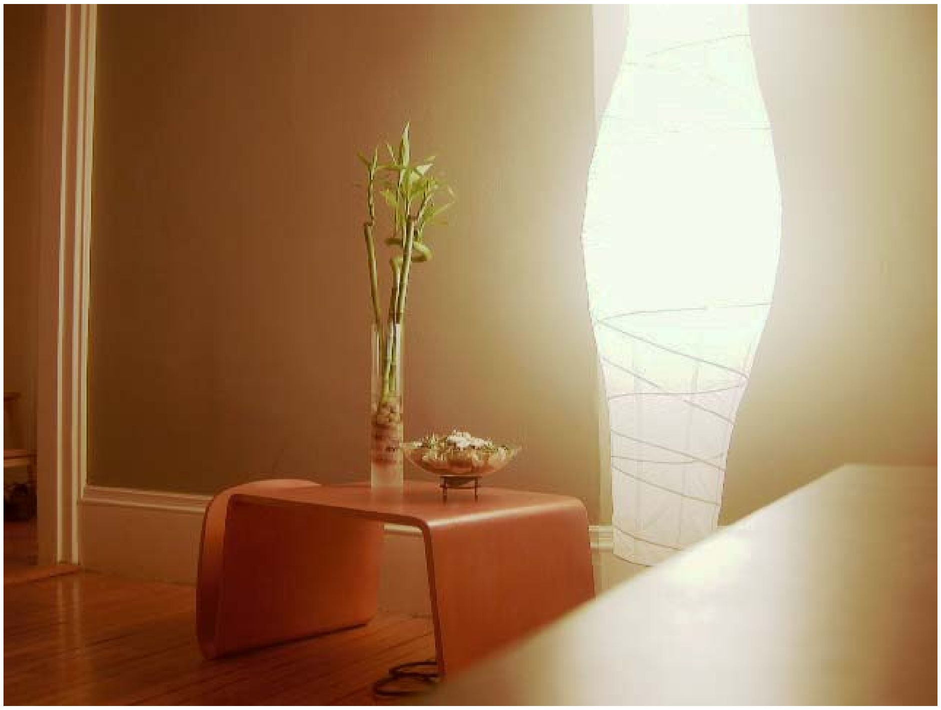


Image 8.6 Histogram adjusted tone mapped radiance map of method 1 for fourth set

The matlab codes are given in Appendix.

8.2 Method 2

We tested method 2 with two sets of images, memorial data set and second data set. We tested the algorithm with two images as input and with more than two images as input. Whereas the algorithm converged when it takes two images as input, it did not converge with more than two images. Additionally, we needed to discard some of the pixels. Firstly, we discarded the edges as we did in the first method. Secondly, since the algorithm does not apply any weighting according to confidence value of pixels, we discarded some pixels with thresholding the ones below 0.05 and above 0.97.

For the memorial data set when the convergence check value was 0.02, polynomial order was 6 for each channel, and initial R value vas 0.4 with images "memorial0063.ppm, memorial0065.ppm", we obtained the inverse camera response function in Figure 8.4 and the linear tone mapped radiance map in image 8.7.

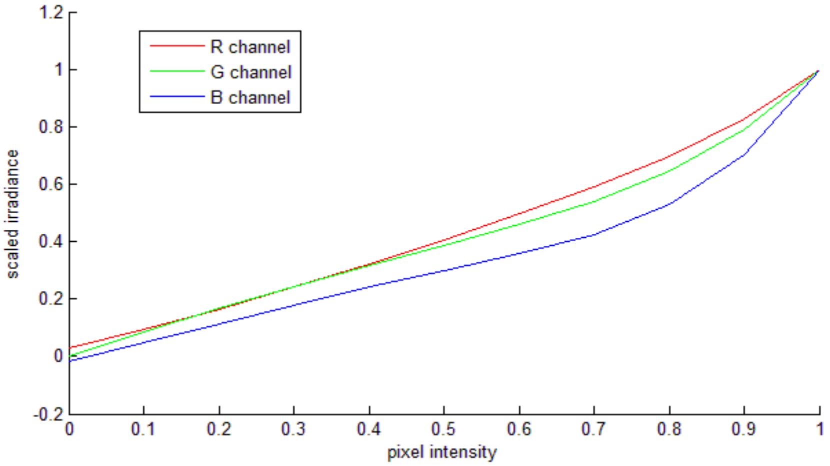


Figure 8.4 Inverse Response Function for each channel for memorial set with method 2

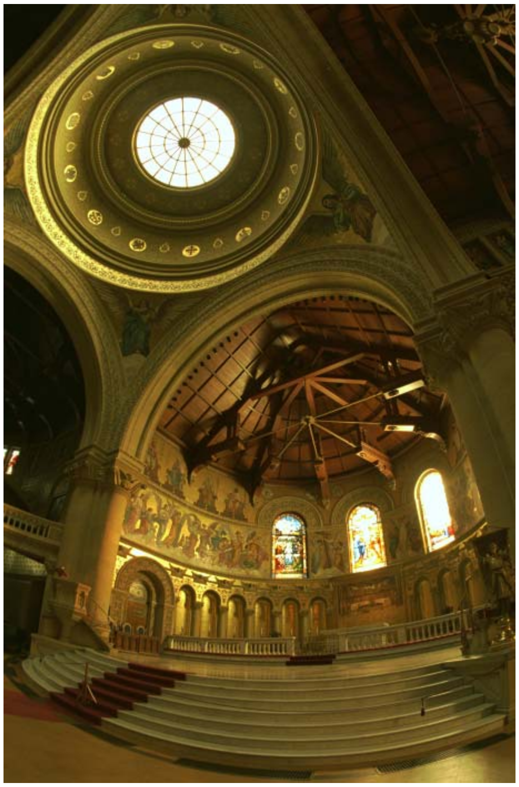


Image 8.7 Linear tone mapped radiance map for memorial set for method 2

While testing the method on the second data set we used the images “img07.jpg, img04.jpg”, and initial R value 0.45, convergence check value 0.01 and polynomial order 7, for all channels, respectively. The resulting inverse camera response function is given in Figure 8.5, linear tone mapped radiance map in image 8.8, histogram adjusted tone mapped radiance map in image 8.9.

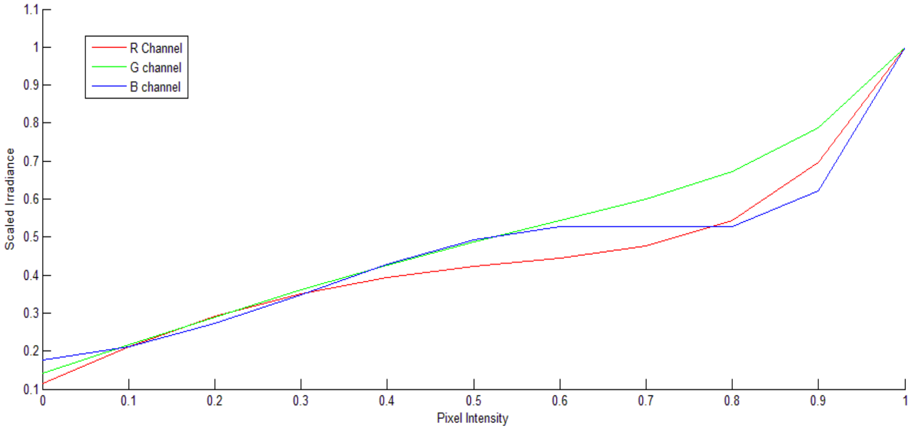


Figure 8.5 Inverse Camera Response Function for each channel for second data set for method 2

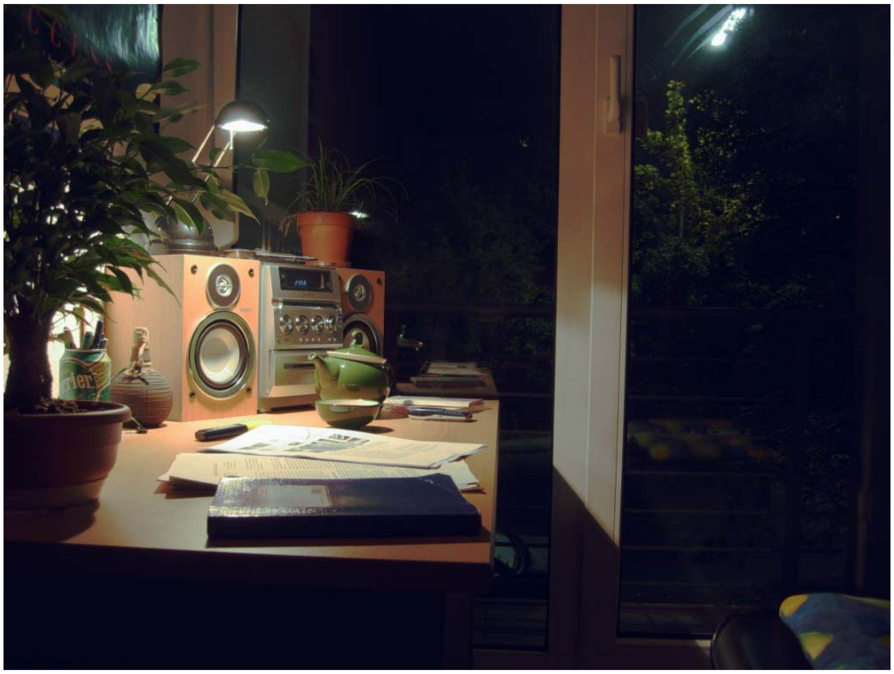


Image 8.8 Linear tone mapped radiance map for second data set for method 2

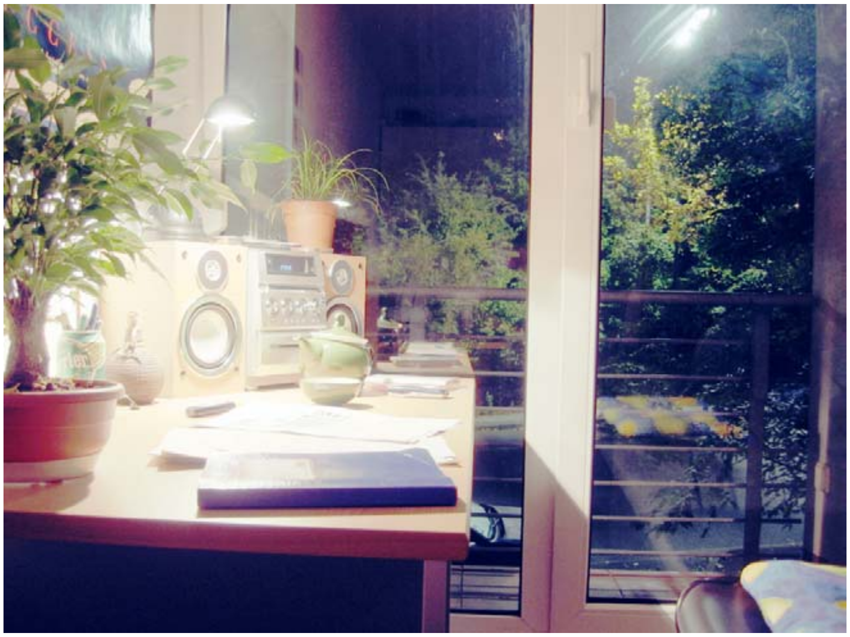


Image 8.9 Histogram adjusted tone mapped radiance map for second data set for method 2

8.3 Method 3

For the third method, firstly we use two images from ‘memorials’ image sequence and recover the *f* function by resort to the zigzag path on the two accumulated histogram graph in 6. Each point is on this camera response function. If we calculate all the points using all the images for 3 color channels, we can get many points on the response function. A polynomial regression is used to fit all the point to one curve.

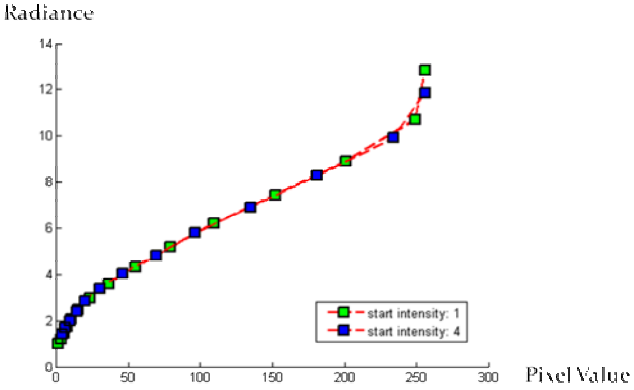


Figure 8.6 Radiance and pixel value points

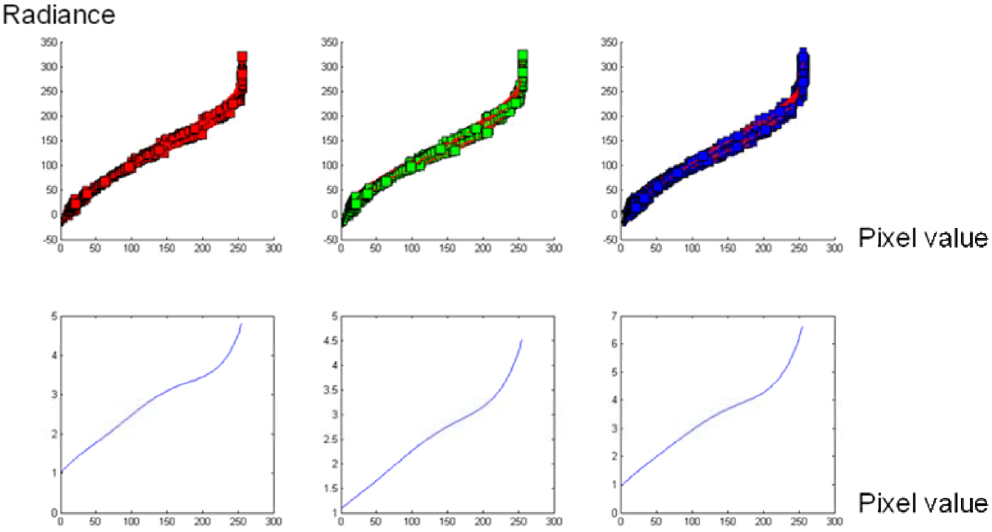


Figure 8.7 Polynomial fitting for 3 channels

After we calculate the camera response function, we use two images to calculate the HDR image. The following image is the result of this synthesis. To display this HDR image, linear tone mapping is used. It does contain both details in both over and under exposed areas in individual LDR images. Both we also notice that there are some hue changes in the image, this is due the fact that we calculate the camera response function for each color channel separately, if one or more channel have an inaccurate response, that color component will mapped to a wrong radiance the hue will change.

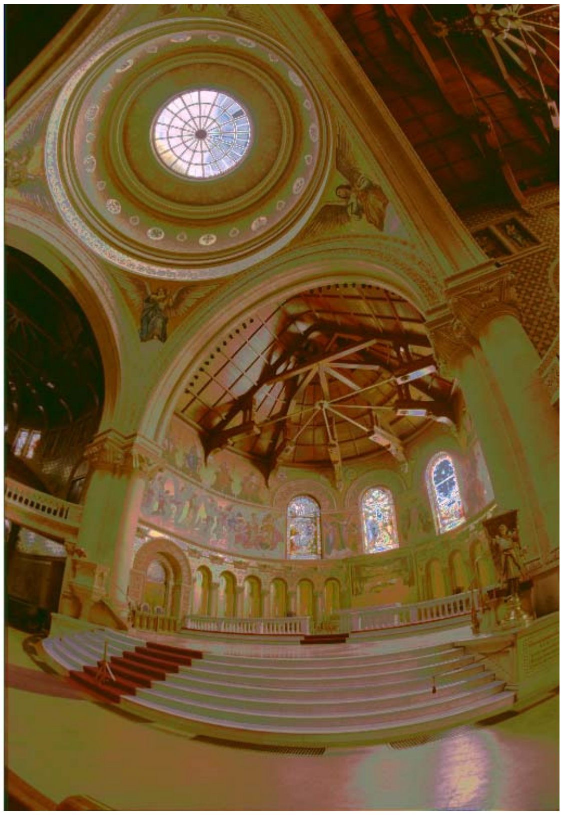
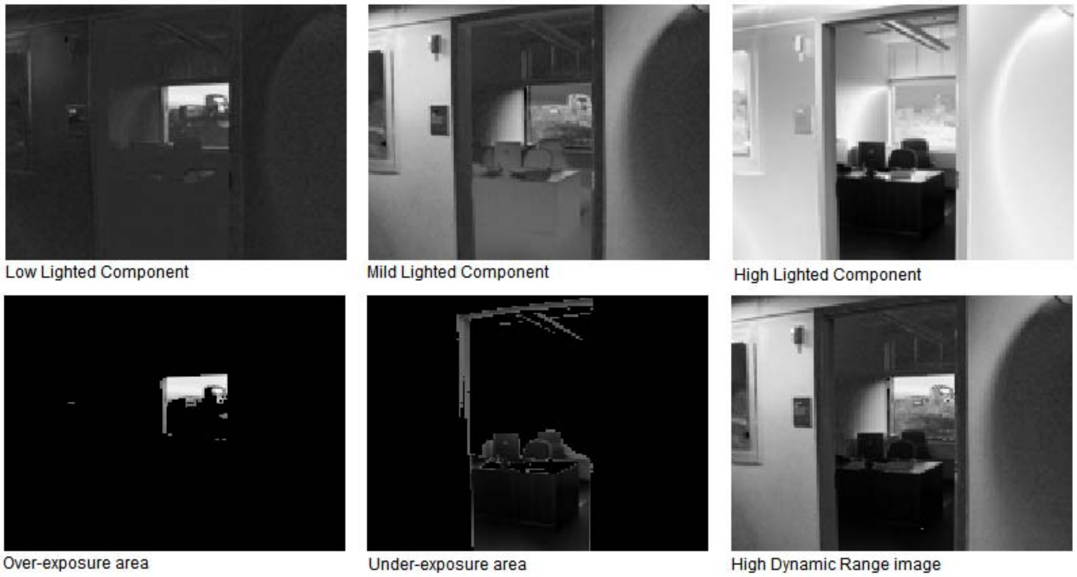


Image 8.10 Linear tone mapped radiance map for method 3 for memorial set

8.4 Method 4

For ICA method, we try this method on ‘memorial’ image set and the images we took at the 9th floor of Photonics Center.



Low Lighted Component

Mild Lighted Component

High Lighted Component

Over-exposure area

Under-exposure area

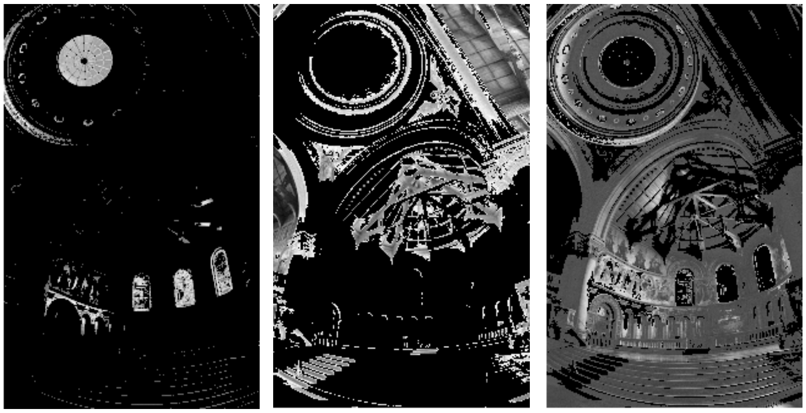
High Dynamic Range image

Image 8.11 Three lighting components, exposure components and HDR image for ICA

For the office image set, we use 3 images as the input for ICA. The first row of the image above shows the ICA yield 3 lighting components: low mild and high. The second row shows the exposure components after we eliminate the uniform area in those images (which is due to the ICA separated the contents in those areas).

The last image is the synthesized HDR image by summation of over-exposure area and under-exposure area together with the mild lighted component.

We also test this method on ‘memorial’ images. In the figure below, the left three images are the three exposure area from ICA and threshold, and the right image is the summation of the three images with weights.



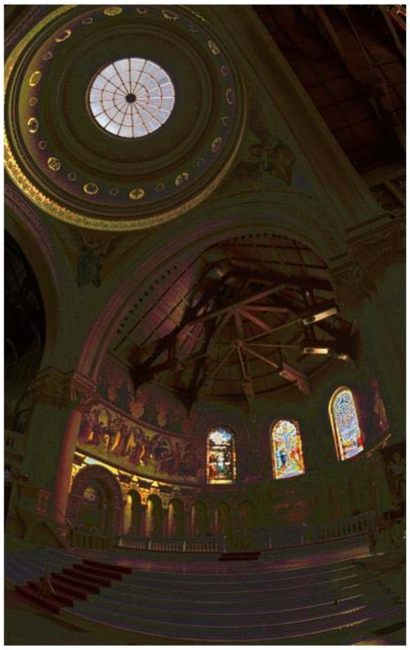


Image 8.12 Image result of ICA method

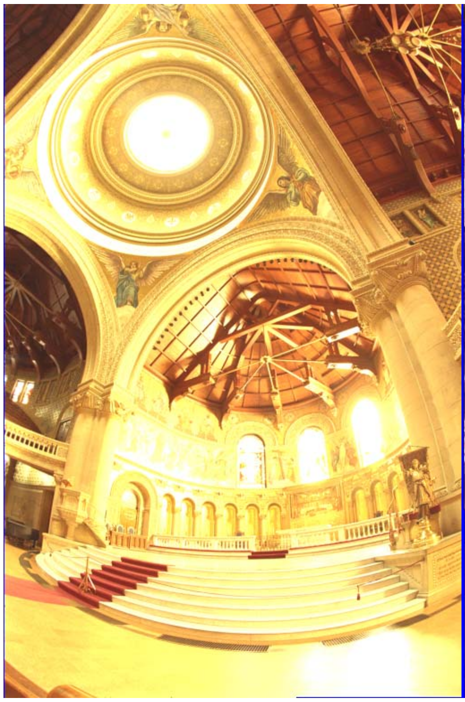
9 Conclusion

We have implemented three different methods to recover the inverse camera response function and the radiance map, and two different tone mapping methods to show the radiance map on LDR displays. We also proposed a new ICA based HDR method which can represent the information on HDR without recovering the radiance map. All these four methods van accurately find the camera response function using image sets composed of static scenes.

We found that method 1 is less sensitive to input images since it applies some weighting according to confidence of pixel values and some regularization which controls the smoothness of inverse camera response function. With method 1 and method 3 we managed to use multiple images in calculating the inverse camera response function, but with method 2 the algorithm converged only with 2 images.

We also found out that histogram based tone mapping gives more preferable results than linear tone mapping.

10 Appendix



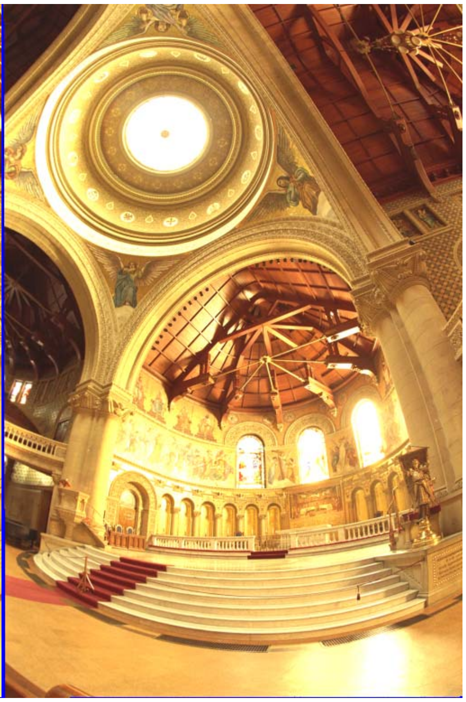


Image 1 memorial0061.ppm Image 2 memorial0062.ppm

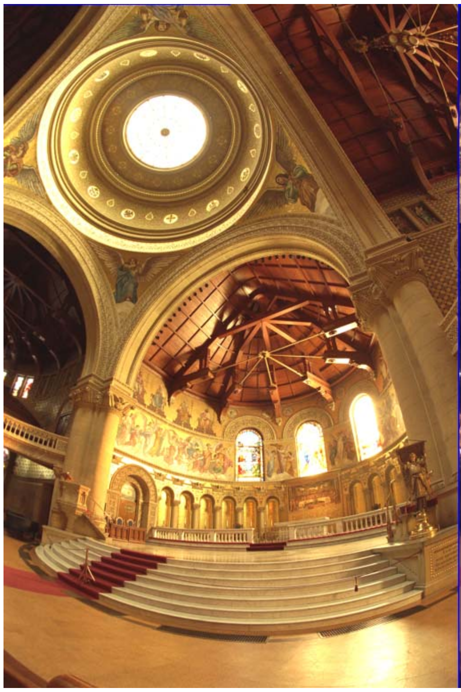


Image 3 memorial0063.ppm Image 4 memorial0064.ppm

Image 5 memorial0065.ppm Image 6 memorial0066.ppm

Image 7 memorial0067.ppm Image 8 memorial0068.ppm

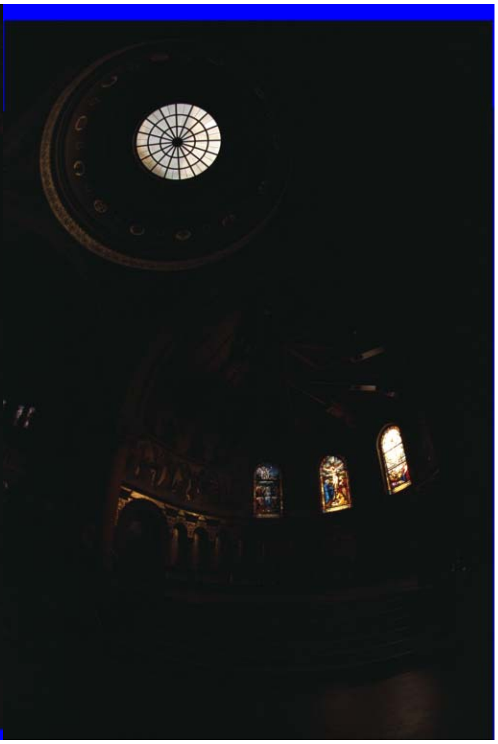


Image 9 memorial0069.ppm Image 10 memorial0070.ppm

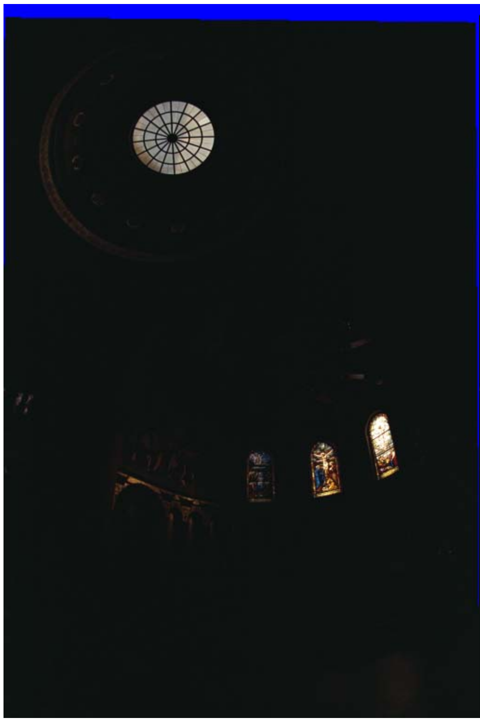
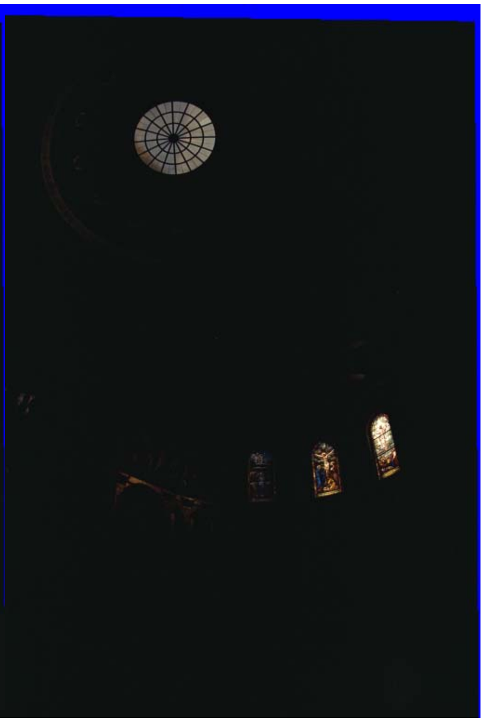



Image 11 memorial0071.ppm Image 12 memorial0072.ppm

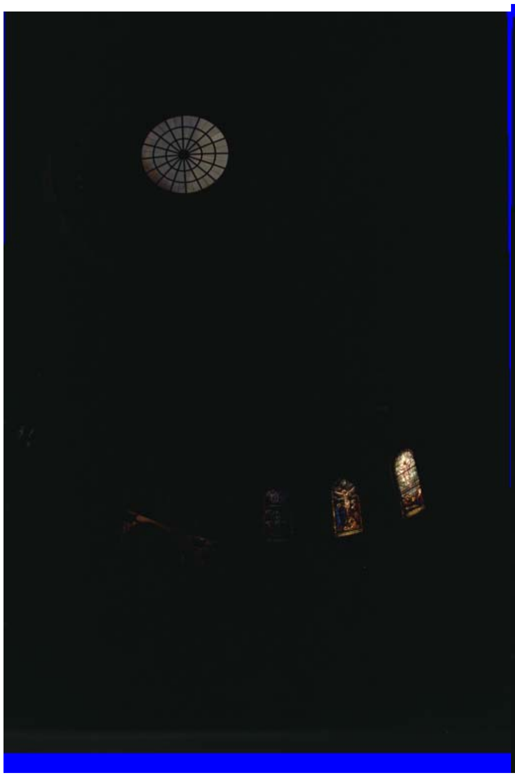
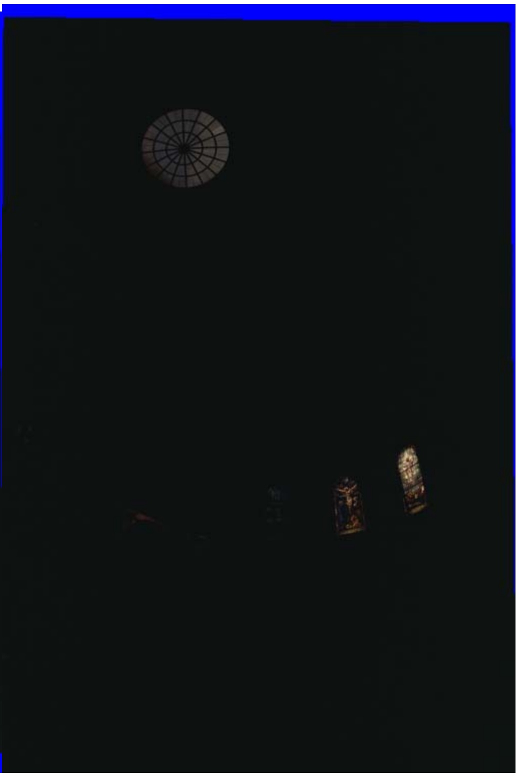



Image 13 memorial0073.ppm Image 14 memorial0074.ppm

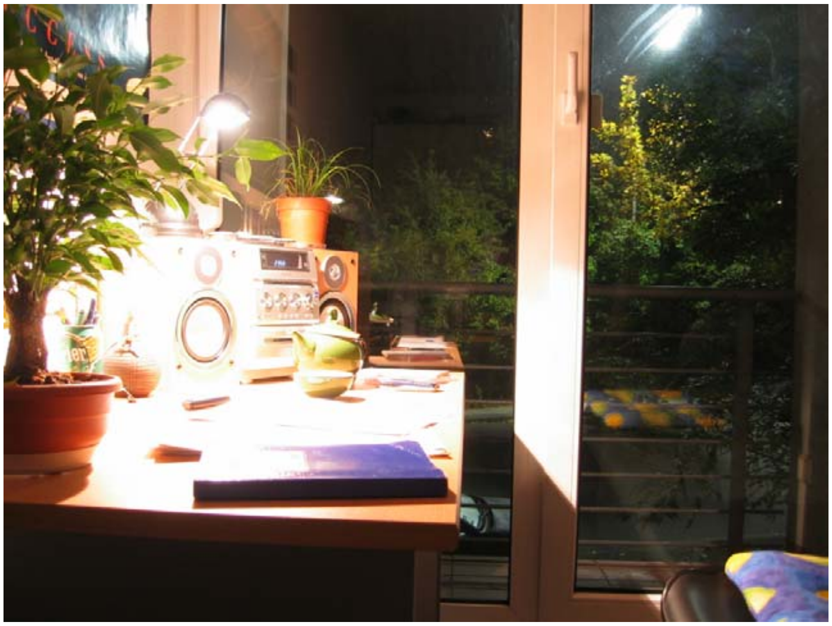


Image 15 img04.jpg

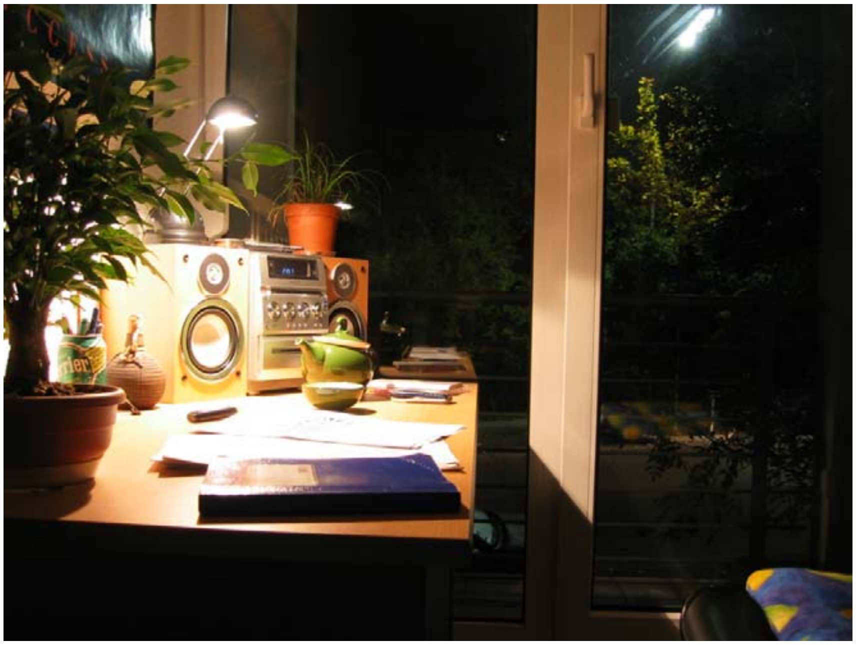


Image 16 img05.jpg

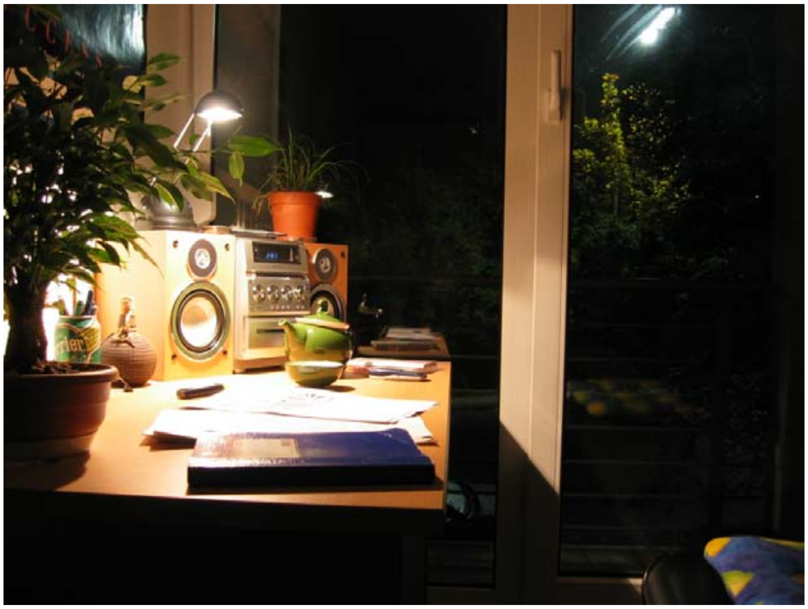


Image 17 img06.jpg

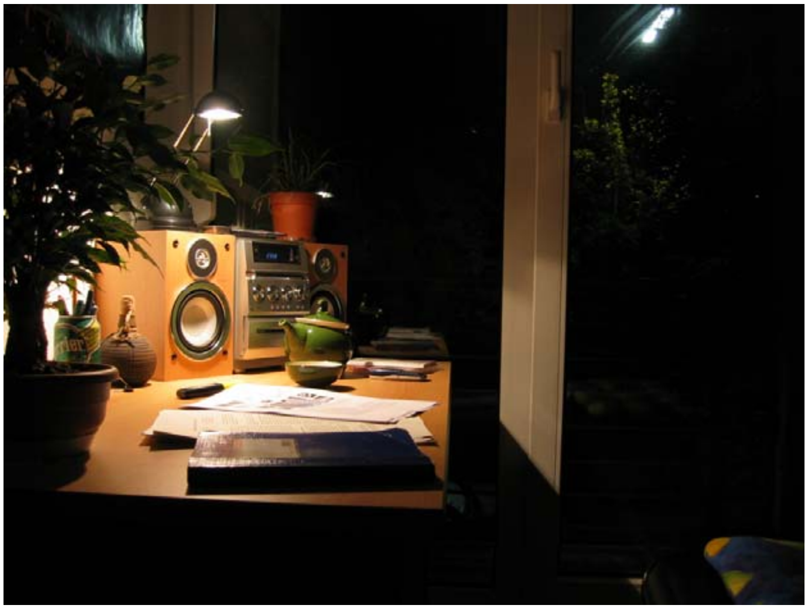


Image 18 img07.jpg

Image 19 P1010015.jpg

Image 20 P1010016.jpg

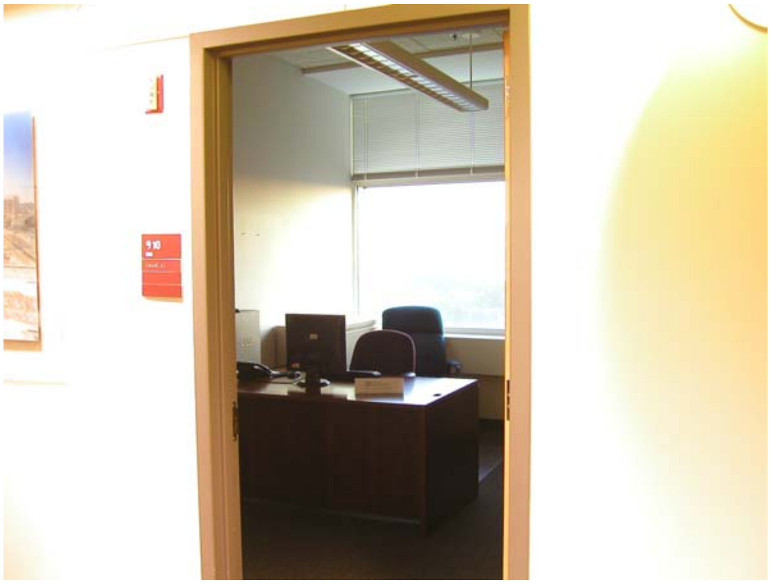


Image 21 P1010017.jpg

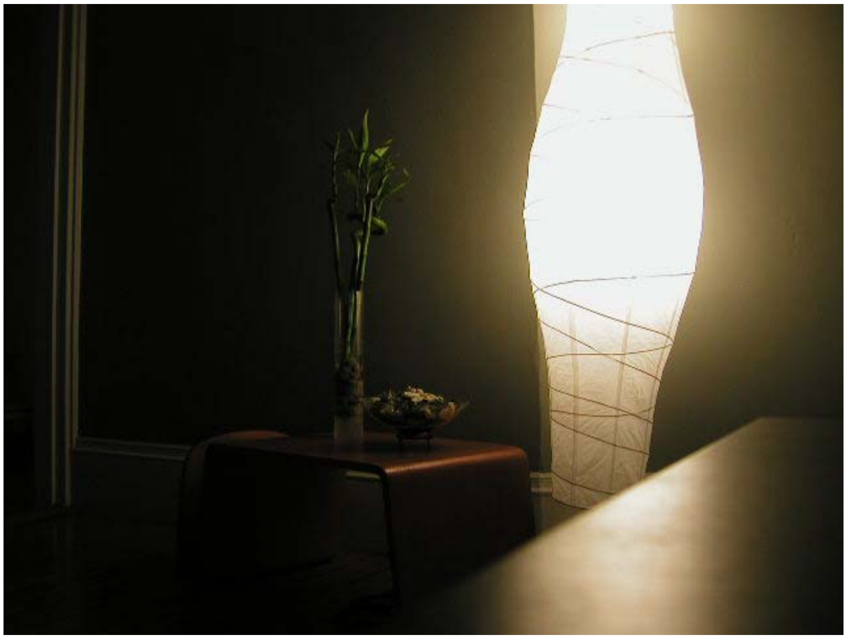


Image 22 P1010007.jpg

Image 23 P1010008.jpg

Image 24 P1010009.jpg

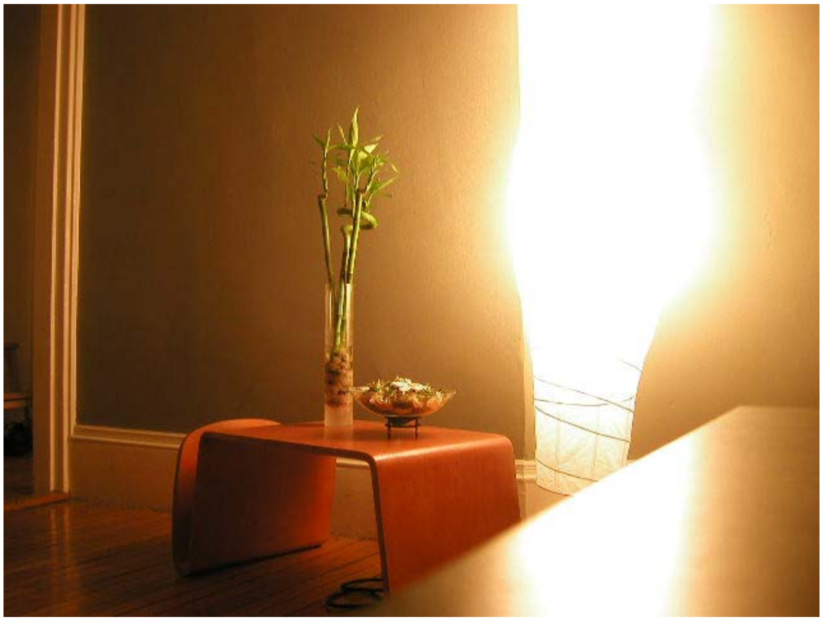


Image 25 P1010010.jpg

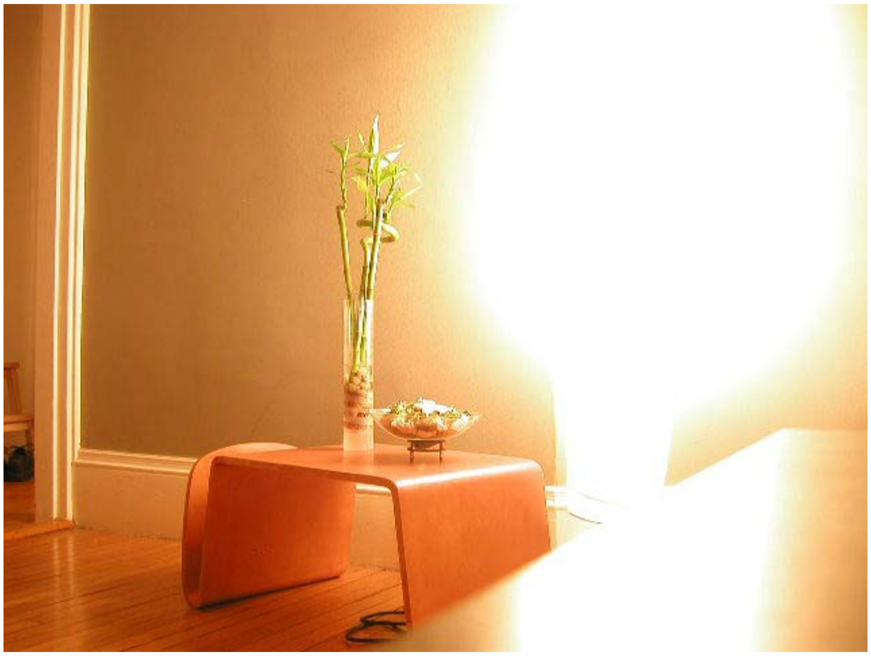


Image 26 P1010011.jpg

Method 1:

The main function:

```
clear all;
```

```
N=4;%number of input images
l=200*N; %lambda
%exposure times
memorial0061=0.03125;
memorial0062=0.0625;
memorial0063= 0.125;
memorial0064=0.25;
memorial0065=0.5;
memorial0066=1;
memorial0067=2;
memorial0068=4;
memorial0069=8;
memorial0070=16;
memorial0071=32;
memorial0072=64;
memorial0073=128;
memorial0074=256;
memorial0075=512;
memorial0076=1024;

%logarithm of expososure times
 R=[ log(1/memorial0063);...
     log(1/memorial0064);...
     log(1/memorial0065);...
     log(1/memorial0066)];

 %read the images
M1(:,:,:,1)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
3.png');
M1(:,:,:,2)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
4.png');
M1(:,:,:,3)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
5.png');
M1(:,:,:,4)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
6.png');
%
M1(:,:,:,5)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
6.png');
%
M1(:,:,:,6)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial006
7.png');
%
M1(:,:,:,7)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial007
1.png');
%
M1(:,:,:,8)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial007
2.png');
%
M1(:,:,:,9)=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial007
3.png');

%crop the blue parts
M=M1(30:710,30:470,:,:);

% calculate g function for each channel
for i=1:3
```

```
    [g(:,i)]=HDR2_multi(M,4,i,R,l);
    i
end


figure(1)
plot([20:250],g(20:250,:),'*')

w=[[0:127], [127:-1:0]];
w=w.*w;

%initialize radiance map
output=zeros(size(M(:,:,3,:),1),size(M(:,:,3,:),2),3);
for col=1:3
for i=1:size(M(:,:,col,:),2),

    for j=1:size(M(:,:,col,:),1)
        sumdiv=0;
        for imn=1:N

output(j,i,col)=output(j,i,col)+(w(M(j,i,col,imn)+1))*(g(M(j,i,col,imn)+1,col)
-R(imn));
            sumdiv=sumdiv+w(M(j,i,col,imn)+1);
        end
        output(j,i,col)=output(j,i,col)/sumdiv;
    end

end
end

%radiance map
figure(2)
imshow(exp(output))

%log of radiance map
figure(3)
imshow(output)

%linear tone mapping
figure(4)
Itmp=255*(exp(output)-min(min(min(exp(output)))))/(max(max(max(exp(output))))-
min(min(min(exp(output)))));
imshow(uint8(Itmp))

%histogram adjusted tone mapping
    nbins=1000000;
   channel=exp(output(:));
    map=cumsum(hist(channel,nbins+1));
    minc=min((channel));
    maxc=max((channel));
    rangec=maxc-minc;

for col=1:3
    for i=1:size(output,1)
        for j=1:size(output,2)
```

```
recons(i,j,col)=map(round(((exp(output(i,j,col))-minc)/rangec)*nbins+1))/map(n
bins+1)*255;
        end
    end
end
figure(5)
imshow(uint8(recons))
```

HDR2_multi function:

```
function [g]=HDR2_multi(Iin,N,color,B,l)

I=[];
I=Iin;

[h,w]= size(I(1:4:end,1:4:end,1));

mask=zeros(h,w);
tmp=zeros(h,w,N);

I=I(:,:,color,:);
for i=1:N

    tmp(:,:,i)=I(1:4:end,1:4:end,i);
    mask=or(mask,abs(gradient(tmp(:,:,i)))>1.5);
end
I=tmp;
mask=not(mask);

totalnumber=sum(sum(mask));


Imask=zeros(totalnumber,N);
Iout=zeros(totalnumber,N);
    good=0;
    for i=1:w
        for j=1:h
            if (mask(j,i))
                good=good+1;
                for imn=1:N
                    Imask(good,imn)=I(j,i,imn);
                end
            end
        end
    end

w=[[0:127], [127:-1:0]];
[g,lE]=gsolve(uint8(Imask),B,l,w);
```

gsolve.m given in [1]:

```
%
% gsolve.m ? Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system¡¯s response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
%
% Zmin = 0
% Zmax = 255
%
% Arguments:
%
% Z(i,j) is the pixel values of pixel location number i in image j
% B(j) is the log delta t, or log shutter speed, for image j
% l is lamdba, the constant that determines the amount of smoothness
% w(z) is the weighting function value for pixel value z
%
% Returns:
%
% g(z) is the log exposure corresponding to pixel value z
% lE(i) is the log film irradiance at pixel location i
%
function [g,lE]=gsolve(Z,B,l,w)
n = 256;
A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);
%% Include the data?fitting equations
k = 1;
for i=1:size(Z,1)
for j=1:size(Z,2)
wij = w(Z(i,j)+1);
A(k,Z(i,j)+1) = wij; A(k,n+i) = -wij; b(k,1) = wij * B(j);
k=k+1;
end
end
%% Fix the curve by setting its middle value to 0
A(k,129) = 1;
k=k+1;
%% Include the smoothness equations
for i=1:n-2
A(k,i)=l*w(i+1); A(k,i+1)=-2*l*w(i+1); A(k,i+2)=l*w(i+1);
k=k+1;
end
%% Solve the system using SVD
x = A\b;
g = x(1:n);
lE = x(n+1:size(x,1));
```

Method 2:

Main function:

```
%% constance setting
clear all;
R=0.45; %choose a suitable value for R (~0.5)
%% first read in two image
%
I3in1=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial0065.png'
);
%
I4in1=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial0063.png'
);
%
%I2in=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial0065.png'
);
%
%I1in=imread('C:\Users\TOSHIBA\Desktop\Memorial_SourceImages\memorial0068.png'
);
% I3in=I3in1(20:720,20:480,:);
% I4in=I4in1(20:720,20:480,:);
I3in=imread('C:\Users\TOSHIBA\Desktop\exposures\img07.jpg');
I4in=imread('C:\Users\TOSHIBA\Desktop\exposures\img04.jpg');

%polynomial  order
P=[7, 7, 7];

%find the coefficients of the polynomial for each channel
for i=1:3
   [c(1:P(i)+1,i),Rnew(i)]=HDRfunc(I3in,I4in,i,R,P(i));
   i
end

%% we have the inverse cam. resp. now, calculate the radiance value for each pixel
in the scene.

[h,w]= size(I3in(:,:,3));
Iout=zeros(h,w,3);
Iout1=zeros(h,w,3);
Iout2=zeros(h,w,3);
Iout3=zeros(h,w,3);
Iout4=zeros(h,w,3);
Ioutder1=zeros(h,w,3);
Ioutder2=zeros(h,w,3);
Ioutder3=zeros(h,w,3);
Ioutder4=zeros(h,w,3);
Iout1tot=zeros(h,w,3);
Iout2tot=zeros(h,w,3);
Iout3tot=zeros(h,w,3);
Iout4tot=zeros(h,w,3);
weight1=zeros(h,w,3);
weight2=zeros(h,w,3);
weight3=zeros(h,w,3);
weight4=zeros(h,w,3);
```

```
for col=1:3

   for i=1:P(col)+1
      %
Iout1(:,:,col)=Iout1(:,:,col)+c(i,col)*im2double(I1in(:,:,col)).^(i-1);
      %
Iout2(:,:,col)=Iout2(:,:,col)+c(i,col)*im2double(I2in(:,:,col)).^(i-1);
      Iout3(:,:,col)=Iout3(:,:,col)+c(i,col)*im2double(I3in(:,:,col)).^(i-1);
      Iout4(:,:,col)=Iout4(:,:,col)+c(i,col)*im2double(I4in(:,:,col)).^(i-1);
   end



end
for col=1:3

   for i=1:P(col)+1;
      k(i,col)=c(P(col)+2-i,col);
   end
   fder(:,col)=polyder(k(:,col));
   %fder(:,col)=polyder(k(1:P(col)+1,col));
end


for col=1:3

   for i=1:P(col)


Ioutder3(:,:,col)=Ioutder3(:,:,col)+fder(P(col)-i+1,col)*im2double(I3in(:,:,co
l)).^(i-1);

Ioutder4(:,:,col)=Ioutder4(:,:,col)+fder(P(col)-i+1,col)*im2double(I4in(:,:,co
l)).^(i-1);
   end

   Iout3tot(:,:,col)=Iout3(:,:,col).*Iout3(:,:,col)./Ioutder3(:,:,col);

Iout4tot(:,:,col)=Rnew(col)*Iout4(:,:,col).*Iout4(:,:,col)./Ioutder4(:,:,col);


   weight1(:,:,col)=Iout1(:,:,col)./Ioutder1(:,:,col);
   weight2(:,:,col)=Iout2(:,:,col)./Ioutder2(:,:,col);
   weight3(:,:,col)=Iout3(:,:,col)./Ioutder3(:,:,col);
   weight4(:,:,col)=Iout4(:,:,col)./Ioutder4(:,:,col);

Iout(:,:,col)=(Iout3tot(:,:,col)+Iout4tot(:,:,col))./(weight3(:,:,col)+weight4
(:,:,col));

end
```

```
figure(1);
imshow(log(Iout));

%display inv. cam. resp. func.
figure(2)

x=[0:0.1:1];

hold on
plot(x,polyval(k(:,1),x),'r');

plot(x,polyval(k(:,2),x),'g');

plot(x,polyval(k(:,3),x),'b');
hold off

%linear tone mapping
figure(3)
Itmp=255*(Iout-min(min(min(Iout))))/(max(max(max(Iout)))-min(min(min(Iout))));
imshow(uint8(Itmp))

%hist. adjusted tone mapping
nbins=1000000;
channel=Iout(:);
map=cumsum(hist(channel,nbins+1));
minc=min((channel));
maxc=max((channel));
rangec=maxc-minc;
for col=1:3
   for i=1:size(Iout,1)
      for j=1:size(Iout,2)

recons(i,j,col)=map(round((((Iout(i,j,col))-minc)/rangec)*nbins+1))/map(nbins+
1)*255;
      end
   end
end
figure(5)
imshow(uint8(recons))
```

Calculation of polynomial coefficients:

```
function [c,R]=HDRfunc(I1in,I2in,color,R,P)


c=zeros(P+1,1);%polynomial coefficient array
newR=-4; %update exposure ratio, intitial value 0;
eps=1e-2; %convergence critieria
A=zeros(P,P); %
b=zeros(P,1);

I11=double(I1in(:,:,color));
```

```
I21=double(I2in(:,:,color));

I1=I11(1:2:end,1:2:end);
I2=I21(1:2:end,1:2:end);
% I1=I11;
% I2=I21;


[h,w]= size(I1);

%% normalize images to 0-1
I1=I1/max(max(I1));
I2=I2/max(max(I2));

%% caculate the mask
%the edges, the area have high gradient values.
mask=or(edge(I1,'sobel'),edge(I2,'sobel'));
mask=bwmorph(mask,'thicken',3);
mask = or(or(I1>0.97,I1<0.05),mask);
mask=or(mask,or(I2>0.97,I2<0.05));

mask=not(mask);
totalnumber=sum(sum(mask));
%I1mask=I1.*mask;
%I2mask=I2.*mask;

I1mask=zeros(totalnumber,1);
I2mask=zeros(totalnumber,1);
good=0;
for i=1:w
   for j=1:h
      if (mask(j,i))
         good=good+1;
         I1mask(good)=I1(j,i);
         I2mask(good)=I2(j,i);
      end
   end
end

%% minimazation of error function

x=[0:0.05:1];
f=ones(size(x));
newf=zeros(size(x));
err_ind=1;
while    (max(abs(newf-f)))>eps  %check convergence



   err(err_ind)=(max(abs(newf-f)))
   err_ind=err_ind+1;
   for k=1:P
```

```
b(k)=-sum((I1mask.^(k-1)-I1mask.^(P)-R*(I2mask.^(k-1)-I2mask.^(P))).*(I1mask.^
P-R*I2mask.^P));

      for l=1:P

A(k,l)=sum((I1mask.^(k-1)-I1mask.^(P)-R*(I2mask.^(k-1)-I2mask.^(P))).*(I1mask.
^(l-1)-I1mask.^P-R*(I2mask.^(l-1)-I2mask.^P)));

      end
   end

   c(1:P)=lsqr(A,b);
   %  c(1:P)=A\b;
   c(P+1)=1-sum(c(1:P));


   Iout1=zeros(totalnumber,1);
   Iout2=zeros(totalnumber,1);
   for i=1:P+1
      Iout1=Iout1+c(i)*I1mask.^(i-1);
      Iout2=Iout2+c(i)*I2mask.^(i-1);
   end

   oldR=R;
   R=(sum(Iout1./Iout2))/totalnumber;





   f=newf;
   newf=zeros(size(x));
   for i=1:P+1;

      newf=newf+c(i)*x.^(i-1);
   end;
   plot(x,f,x,newf)
   R
end
```

Method 3:

```
%% constance setting
N=5;%number of input images
B=[log(1/0.0625);log(1/0.125);log(1/0.5); log(1/2); log(1/32)]; %choose a
suitable value for R (~0.5)
l=2*N;
ResizeRatio=0.2;

I=[];
%% first read in two image
I(:,:,:,1)=imread('E:\My Picture\Recognition\HDR\Memorial\memorial0062.png');
```

```
I(:,:,:,2)=imread('E:\My Picture\Recognition\HDR\Memorial\memorial0063.png');
I(:,:,:,3)=imread('E:\My Picture\Recognition\HDR\Memorial\memorial0065.png');
I(:,:,:,4)=imread('E:\My Picture\Recognition\HDR\Memorial\memorial0067.png');
I(:,:,:,5)=imread('E:\My Picture\Recognition\HDR\Memorial\memorial0071.png');

I=I(50:700,50:470,:,:);

[h,w]= size(I(:,:,1));
h=floor(h*ResizeRatio)+1;
w=floor(w*ResizeRatio)+1;
mask=zeros(h,w);
tmp=zeros(h,w,N);
I=I(:,:,3,:);% average over color
for i=1:N
    tmp(:,:,i)=imresize(I(:,:,i),ResizeRatio);
    mask=or(mask,abs(gradient(tmp(:,:,i)))>1.5);
end
I=tmp;
mask=not(mask);

totalnumber=sum(sum(mask));

%% minimazation of error function
Imask=zeros(totalnumber,N);
Iout=zeros(totalnumber,N);
    good=0;
    for i=1:w
        for j=1:h
            if (mask(j,i))
                good=good+1;
                for imn=1:N
                    Imask(good,imn)=I(j,i,imn);
                end
            end
        end
    end

w=[[0:127], [127:-1:0]];
[g,lE]=gsolve(uint8(Imask),B,l,w);

subplot(121);
plot(g);

tmp=uint8(tmp);
output=zeros(size(tmp,1),size(tmp,2));
for i=1:size(tmp,2)
    for j=1:size(tmp,1)
        sumdiv=0;
        for imn=1:N
            output(j,i)=output(j,i)+w(tmp(j,i,imn)+1)*g(tmp(j,i,imn)+1);
            sumdiv=sumdiv+w(tmp(j,i,imn)+1);
        end
        output(j,i)=output(j,i)/sumdiv;
    end
end
```

```
subplot(122);
imshow(output-min(min(output)),[]);
```

Method 4:

```
 I1r = imresize(mean(I1,3),[height width]);
 I2r = imresize(mean(I2,3),[height width]);
 I3r = imresize(mean(I3,3),[height width]);
 I4r = imresize(mean(I4,3),[height width]);
 XX = [I1r(:)';I2r(:)';I3r(:)';I4r(:)']';

SS=W*XX';

I_low=reshape(SS(1,:,:),height,width);
I_mid=reshape(SS(2,:,:),height,width);
I_high=reshape(-SS(3,:,:)/4,height,width);

TL1=0.4;
TL2=1.0;

TM1=0.5;
TM2=0.5;

TH1=0.0;
TH2=0.4;


I_lown=(I_low-min(min(I_low)))/(max(max(I_low))-min(min(I_low)));
I_midn=(I_mid-min(min(I_mid)))/(max(max(I_mid))-min(min(I_mid)))/3;
I_highn=(I_high-min(min(I_high)))/(max(max(I_high))-min(min(I_high)));

Th=ones(height,width);

maskl=(I_lown>(Th*TL1)).*(I_lown<(Th*TL2));
maskh=(I_highn>(Th*TH1)).*(I_highn<(Th*TH2));
I_lt=I_lown.*maskl;
I_mt=I_midn-I_midn.*or(maskl,maskh);
I_ht=I_highn.*maskh;

subplot (2,3,1);
imshow(I_lown);
subplot (2,3,2);
imshow(I_midn);
subplot (2,3,3);
imshow(I_highn);

subplot (2,3,4);
imshow(I_lt);
 subplot (2,3,5);
imshow(I_ht);
 subplot (2,3,6);
 imshow(I_mt+I_ht/2+I_lt);
```

11 References

[1] P. Debevec and J. Malik, “Recovering High Dynamic Range Radiance Maps From Photographs,” *Proc. ACM SIGGRAPH ’97* (1997), pp. 369–378.

[2] T. Mitsunaga and S. Nayar, “Radiometric Self Calibration,” *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (June 1999), Vol. 1, pp. 374–380.

[3] M. Grossberg and S. Nayar, “Determining the Camera Response From Images: What is Knowable? *IEEE Trans. PAMI 25*, 11 (November 2003), 1455–1467.

[4] A. Troccoli, S.B. Kang and S. Seitz, ”Multi-View Multi-Exposure Stereo,” *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06),* pp. 861-868, 2006.

[5] http://www.debevec.org/Research/HDR/

[6] http://www.mpi-inf.mpg.de/resources/hdr/calibration/pfs.html

[7] A. M. Sa, P. C. Carvalho, L. Velho, “High Dynamic Range Image Reconstruction,” Morgan Claypool Publishers, 2007.

[8] H.S Le, A.Anani, H. Li “High Dynamic Range Imaging Through Multi-Resolution Spline”

[9] P.J. Burt and R.J. Kolczynski, “Enhanced image capture through fusion,” in Proc. of ICCV, 1993, pp. 173–182.

[10] R. Szeliski, M. Uyttendaele, D. Steedly “Fast Poisson Blending using Multi-Splines”

[11] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” in Proc. of SIGGRAPH, 2002, pp. 249–256.

[12]Pierre Comon (1994) “Independent Component Analysis: a new concept”, Signal Processing, Elsevier, 36(3):287--314

[13] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah, “Shape from Shading:A Survey,” IEEE Trans. Pattern Analysis and Machine Intelligence,vol. 21, no. 8, pp. 690-706, Aug. 1999.

[14] S.K. Nayar, K. Ikeuchi, and T. Kanade, “Shape from Interreflections,” Int’l J. Computer Vision, vol. 6, no. 3, pp. 173-195, Aug. 1991

[15]L.T. Maloney and B.A. Wandell, “Color Constancy: A Method for Recovering Surface Spectral Reflectance,” J. Optical Society of Am. A, vol. 3, pp. 29-33, 1986.

[16] P.E. Debevec, “Rendering Synthetic Objects into Real Scenes,” Proc. ACM SIGGRAPH, pp. 189-198, 1998.

[17] N. Asada, A. Amano, and M. Baba, “Photometric Calibration of Zoom Lens Systems,” Proc. Int’l Conf. Pattern Recognition, vol. A, pp. 189-190, 1996.

[18] S. Mann and R.W. Picard, “Video Orbits of the Projective Group:A Simple Approach to Featureless Estimation of Parameters,” Proc. Int’l Conf. Image Processing, vol. 6, no. 9, pp. 1281-1295, Sept.1997.

[19] S. Mann and R. Picard, “Being ’Undigital’ with Digital Cameras:Extending Dynamic Range by Combining Differently Exposed
Pictures,” Proc. 46th Ann. IS&T Conf., pp. 422-428, 1995.

[20] P.E. Debevec and J. Malik, “Recovering High Dynamic Range Radiance Maps from Photographs,” Proc. ACM SIGGRAPH, pp. 369-378, 1997.

[21] T. Mitsunaga and S.K. Nayar, “Radiometric Self Calibration,” Proc CS Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 374- 380, June 1999

[22] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah, “Shape from Shading: A Survey,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 8, pp. 690-706, Aug. 1999.

[23] Y. Tsin, V. Ramesh, and T. Kanade, “Statistical Calibration of the CCD Imaging Process,” Proc. Int’l Conf. Computer Vision, vol. 1, pp. 480-487, 2001.

[24] S. Mann, “Comparametric Imaging: Estimating Both the Unknown Response and the Unknown Set of Exposures in a Plurality of Differently Exposed Images,” Proc. CS Conf. Computer Vision and Pattern Recognition, Dec. 2001.