G. Yourganov, W. Stuerzlinger

Acquiring High Dynamic Range Video at Video Rates

Technical Report

Dept. of Computer Science, York University

May 2001

1. Introduction

In the real world, the scene may contain some well-illuminated areas as well as some areas that are in the darkness. The dynamic range of the scene, or the radiance ratio between the brightest and the darkest spots, can be quite big – it can span several orders of magnitude in some cases. This is a lot wider range than the range that our digital camera can adequately capture, and that our monitor can adequately display. The areas that are too bright for the dynamic range of our camera are going to be saturated on our image, and the areas that are too dark will be under-exposed. In both cases, all the information about these areas will be lost in the image.

A way to solve this problem is to combine different shutter speeds of our camera. Smaller shutter speeds can give us good information about the bright areas (because we won't have saturation), and longer shutter speeds effectively capture the dark areas. So by alternating the shutter speeds and by combining the two images into one in real-time, we can achieve higher dynamic range for digital video cameras.

2. Constructing high dynamic range image: Debevec's algorithm

In [1], Paul Debevec and Jitendra Malik have proposed a useful algorithm that can calculate the radiometric response of the camera and combine differently exposed images into one high dynamic range image. Radiometric response is a function that describes the relationship between the amount of light received by film grain or CCD, the exposure setting, and the final pixel brightness. The algorithm was designed for still photographic images, but it can be easily applied to digital video as well.

The main advantage of their approach is that they make very little assumptions about the response function: indeed, the only constraints are that the function is invertible and smooth. The proposed algorithm has proved to be quite robust and easy to use. And, as a bonus, the theory behind it is easy to understand.

Let us consider a pixel the position *i* in the image. Let E_i be the irradiance (total amount of incoming energy) for the corresponding photosensitive element of the camera, and let Δt_j be the exposure time. If $f(\cdot)$ is the response function of the camera, then the intensity of the image pixel is

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

We assume that the inverse of $f(\cdot)$ is well defined:

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

Now, let us define a function $g(\cdot)$ as $\ln f^{-1}(\cdot)$; then, we have

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \qquad (1)$$

Instead of finding the response function $f(\cdot)$, we can find the discrete values of $g(\cdot)$ for all possible pixel intensities; let *M* be the number of such distinct intensity levels. Normally, for a black and white image *M* is 256.

Now, let us consider a set of *P* images, each taken with a different exposure time. Each image depicts the same scene and consists of *N* pixels. If we write equation (1) for each pixel and each exposure time, we will have *N***P* equations, with *i* varying from 1 to *N*, and *j* varying from 1 to *P*. The knowns here are the exposure times and the pixel intensities; we have to find the irradiance values and the values that function *g* (z) takes over the range of pixel intensities. We have to find such values of E_i and g (z) that best satisfy our set of equations in the least square error sense. This is equivalent to minimizing the objective function O:

$$O = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + l \sum_{M} g''(z)^2 \qquad (2)$$

The second term is introduced in order to make $g(z)$ smooth; to do so, we have to minimize the second derivative of $g(z)$. The parameter l represents the “significance” of the smoothness relative to the significance of data fitting: if we want to make our function smoother, we must increase l. Second derivative is defined discretely as

$$g''(z) = g(z-1) - 2g(z) + g(z+1)$$

So, at the end we have $N*P$ equations for data fitting (in the form of (1)) and M-2 equations for smoothness. Number of unknowns is $N + M$; this overdetermined system of equations can be solved, for example, using singular-value decomposition method.

An additional constraint is enforced on g (z): let Z_{mid} be the pixel intensity in the middle of the intensity range (Z_{mid} = 125 for 256 intensity levels), and we set g (Z_{mid}) = 0.

Also, Debevec makes a somewhat *ad hoc* assumption that g (z) will fit the data more poorly for very low and very high values of z, since the function has more slope in these regions. So he introduces a weighting function *w(z)* go give more emphasis to the data points in the middle of intensity range. A hat function is used for that, so that Z_{mid} gets the maximum weight. Objective function (2) now takes the form

$$O = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij})\}[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + l \sum_{M} [w(z) g''(z)]^2$$

Once we have recovered the response function *g(z)*, we can reconstruct the high dynamic range image, i.e. for each pixel in our image we can estimate the corresponding irradiance value:

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

Debevec proposes to use pixel values from all exposures to estimate the irradiance. These pixels are weighted by the same weighting function *w(z)*:

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

The advantages of Debevec's method are numerous. No assumptions are made about the response function, other than its invertability. Most radiometric calibration methods assume the monotonicity of the response function; to achieve this, we can increase the smoothness coefficient l, so the function will be monotonic as well as smooth. Also, this method is quite robust, due to the simplicity of equations and stability of singular value decomposition method used to solve the equation system.

3. Constructing high dynamic range image: Mitsunaga's algorithm

In [2], Tomoo Mitsunaga and Shree Nayar developed a slightly different method of radiometric calibration for the case when we don't know exact exposure time intervals. They use a polynomial model for the response function, which makes their method somewhat less flexible than the one developed by Debevec and Malik.

Let us start by analyzing how the pixel brightness corresponds to the scene radiance. The image irradiance E (the amount of incoming energy) relates to the scene radiance L as follows:

$$E = L\frac{p}{4}(\frac{d}{h})^2 \cos^4 f$$

where h is the focal length of the camera lens, d is its aperture diameter, and f is the angle between the principal ray and the optical axis. In the idealized case when the camera has a linear response, the brightness of the corresponding pixel would be $I = Et$, where t is the time interval during which the camera is exposed to the scene. Quantity I, which is traditionally called "photographic exposure", can be expressed as the product

$$I = L\ k\ e$$

Here, $k = \cos^4 f / h^2$, and $e = (p\ d^2 / 4)\ t$, which will be referred to as the *exposure setting* of the camera, and can be controlled by adjusting aperture d or exposure time interval t.

If the camera has the response function f (typically non-linear), pixel brightness will be $Z = f(I)$. Let g be the inverse of response function f, and, therefore, $I = g(Z)$. Mitsunaga's approach uses a polynomial model for this function:

$$I = g(Z) = \sum_{k=0}^{N} c_k Z^k$$

so the task of the calibration becomes one of finding the polynomial coefficients and the maximum order N.

Also, this algorithm does not require the exact values of exposure time intervals (which is quite practical, because for many cameras these values are either not specified at all or approximate values are given). These intervals are also estimated in the process of calibration. Actually, the algorithm works not with the exposure time intervals, but rather with the ratios of consecutive intervals.

Let us take a look at two images of the same scene, taken with exposure settings e_q and e_{q+1}. Let $R_{q,q+1}$ be their ratio e_q / e_{q+1}. Then the ratio of quantities of light received at any CCD (let us say that pixel p corresponds to that CCD) is

$$\frac{I_{p,q}}{I_{p,q+1}} = \frac{L_p k_p e_q}{L_p k_p e_{q+1}} = R_{q,q+1}$$

If g is the inverse of response function f, then we have

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

And, since we represent g with a polynomial, this can be rewritten as

$$\frac{\sum_{k=0}^{N} c_k Z_{p,q}{}^k}{\sum_{k=0}^{N} c_k Z_{p,q+1}{}^k} = R_{q,q+1}$$

If we know the ratios of exposure settings and maximum order N, then the problem of finding the response function can be solved approximately by calculating the values of polynomial coefficients c_k that minimize the error function

$$e = \sum_q \sum_p \left[\sum_{k=0}^{N} c_k Z_{p,q}{}^k - R_{q,q+1} \sum_{k=0}^{N} c_k Z_{p,q+1}{}^k \right]^2$$

This can be obtained by setting to zero the derivatives of e with respect to the coefficients:

$$\frac{\partial e}{\partial c_k} = 0$$

and solving the resulting system of linear equations.

We can expand this algorithm to the case when exact exposure setting ratios are not known. The user must provide the initial guesses for the ratios, let us call them $R_{q,q+1}^{(0)}$. Then, the iterative scheme is used: with the help of these ratios, we estimate the set of coefficients $c_k^{(1)}$, which, in turn, is used to compute the next guess for ratios:

$$R_{q,q+1}^{(1)} = \sum_p \frac{\sum_{k=0}^{N} c_k^{(1)} Z_{p,q}^{\ k}}{\sum_{k=0}^{N} c_k^{(1)} Z_{p,q+1}^{\ k}}$$

and so forth. We stop when the convergence criterion

$$| f^{(n)}(Z) - f^{(n-1)}(Z) | < \boldsymbol{d}$$

is satisfied for all values of Z, where $\boldsymbol{d}$ is a small number provided by the user.

N, maximum order of the polynomial, can be found in the following fashion: the user places the upper bound on N, and the algorithm is applied repeatedly to find N that minimizes $\boldsymbol{e}$.

The main advantage of this method is the fact that it doesn’t require the user to know exact shutter speed values; also, the polynomial model used for the response function is compact and easy to use. However, in practice I have observed that the method is not very robust.

4. Acquisition of high dynamic range image in digital video

My implementation of the high dynamic range video can be broken up into the following steps:

- Radiometric calibration of the video camera. This step gives us the radiometric response function for a particular camera: for each pixel intensity value Z and exposure time Δt (used to obtain the image containing this pixel), the function assigns the irradiance value E, which is a measure of the light incoming to the corresponding CCD.
- Construction of the high dynamic range image. We acquire two images with two different exposure settings, calculate the corresponding irradiance values for each pixel using the response function, and use these values to make the high dynamic range image.
- Display of the high dynamic range image. In a general case, where the range of the resulting image is wider than the displayable range, we have to perform tone mapping on the image to make it displayable.

4.2 Radiometric Calibration

Two algorithms of radiometric calibration were tested: one as described by Paul Debevec and the other as described by Mitsunaga.

In both cases, we have to provide a set of differently exposed pictures. The scene has to be carefully chosen such that it contains bright objects, such as a lamp or a window on a sunny day, and large areas that are illuminated relatively poorly and at the same time contain some important scene features (the indoors of a room in the shadow). Shutter speeds have to be chosen so that they could meet the following criteria:

a) a substantial subset of the range of shutter speeds available for this camera was covered;

b) the details of the brightest spots of the scene could be revealed on a snapshot only if the fastest shutter speed setting was used;

c) at the same time, the only way to get most details of the darkest regions of the scene was to set the camera to the slowest shutter speed.

Debevec's algorithm requires a set of differently exposed images of the same scene and the exposure times specified for each image. The result is the response function represented as a set of discrete values specified for each pixel intensity value in the range of [0; 255] (which covers the range of intensities for 8-bit pixels).

Mitsunaga's approach does not assume that we know the exact values of the shutter speed; it tries to estimate them. So, together with differently exposed images, we need to provide a set of initial guesses for the shutter speeds. This approach to calibration gives us a response function, represented as a set of polynomial coefficients, and the estimated shutter speed values.

The calibration software can be downloaded from the authors' websites:

- Paul Debevec's: http://www.debevec.org
- Tomoo Mitsunaga's: http://www1.cs.columbia.edu/CAVE/tomoo/RRHomePage/rrhome.html

4.2 High Dynamic Range image acquisition

Acquisition of an image with high dynamic range is performed in real time using the response function of the camera and two images taken at different shutter speeds. The main idea of the process is to estimate, for each pixel, the level of irradiance that of the corresponding CCD element of the camera. Note that this irradiance is calculated in relative units, rather than in W/m^2.

The two shutter speeds must be carefully selected. The lower shutter speed of the two (which is the one that gives the darker image) must be selected so that no pixels are saturated, because corresponding pixels in the brighter image will be definitely saturated, and it would be impossible to recover any information from these pixels. In general, for the recovery of the information about the dark areas of the scene, the image taken with higher shutter speed will be more significant (because of the low level of illumination, these areas don't register too well in the other image); and, the image taken with the lower shutter speed will be more responsible for the brighter areas (because they might saturate the CCD at the higher shutter speed). Also, these two speeds should be quite far apart, to obtain a higher range of irradiances and therefore a higher dynamic range of the resulting image.

Since I used two methods of radiometric calibration (and these two methods lead to different ways of construction of high dynamic range image), two separate programs were written.

4.2.1 Using Debevec's approach

The inverse of response function, noted as *g*, can be used to calculate the irradiance at the camera element that corresponds to a pixel. For the exposure time setting Δt_j, and for pixel *i* with brightness Z_{ij}, the logarithm of relative irradiance is

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

In Debevec's paper, the authors recommend calculating the relative irradiance as follows: using the pixel brightness values at the same location of differently exposed images, calculate relative irradiances for each image and then calculate their weighted sum. I have chosen a different, simpler, approach.

First, we have to know what is the pixel brightness value when the camera CCD is saturated; this is quite easy to find out by selecting a long exposure time and then pointing the camera towards some bright area. Let us call the obtained pixel brightness as the *saturation threshold*.

Then, we obtain two images at two different shutter speeds. We go through the brighter image, pixel-by-pixel. We assume that the information at the pixel is useful, unless the pixel is saturated. In that case, we use the information at the same location in the darker image.

So, the logarithm of relative irradiance value is:

$$\ln E_i = \begin{cases} g(Z_{i,BRIGHT}) - \ln \Delta t_{BRIGHT} & \text{if } Z_{i,BRIGHT} < \text{saturation threshold} \\ g(Z_{i,DARK}) - \ln \Delta t_{DARK} & \text{otherwise} \end{cases}$$

After one walk through the brighter image, the high dynamic range image is constructed. Every pixel location of this image holds the value of relative irradiance.

To speed up the process, we can use look-up tables that hold the irradiance values for a given pixel value. In this case, we need to have two look-up tables, one for brighter image and one for lighter image; each one contains the values of

$$\ln E = g(Z) - \ln \Delta t$$

with the corresponding values of Δt_{BRIGHT} and Δt_{DARK}.

With the help of this look-up table, construction of the final image is easy and quick. First, we obtain the two images with our two different shutter speed settings. This is done consecutively: set the camera to the faster speed, take the darker image, than set it to the slower speed, take the brighter image. Now, we are ready to process the images. We traverse the brighter image pixel by pixel and compare the pixel brightness Z_{BRIGHT} to the saturated threshold. If it is below, we look at the look-up table computed for Δt_{BRIGHT} and obtain the irradiance value for Z_{BRIGHT}. If not, we look at the pixel brightness Z_{DARK} located at the same position in the darker image, and obtain the corresponding irradiance value from the look-up table computed for Δt_{DARK}. The obtained value goes to our high dynamic range image which will be displayed on the screen.

4.2.2 Using Mitsunaga's approach

Mitsunaga's radiometric calibration algorithm gives us two things: the response function (in the form of polynomial coefficients) and the estimates of time exposure intervals (or, rather, their ratios). From these estimates, we compute the "scaled exposures" e_q; the absolute values are not really necessary, so we calculate the scaled exposures so that their arithmetic mean is 1.

With the help of the reverse of response function (which we note as $g(Z)$), we can compute the photographic exposure for pixel at position p at image q:

$$I_{p,q} = g(Z_{p,q}) = \sum_{k=0}^{N} c_k Z_{p,q}{}^k$$

From this, we can get the relative irradiance value for pixel position p:

$$E_q = I_{p,q} / e_q$$

In the process of high dynamic range image construction, we go, pixel by pixel, through both images; using the response function and scaled exposure settings calculated for the shutter speeds we use, we compute two relative irradiance values. Then the final relative irradiance value is calculated using weighted average of the two irradiance values.

The choice for weighting function is determined when we take noise in the image into account. The reliability of the measurement depends on its signal-to-noise ratio (SNR), which is, for our case, estimated as

$$I \frac{dZ}{dI} \frac{1}{s\,(Z)} = \frac{g(Z)}{s\,(Z) g'(Z)}$$

where $s(Z)$ is the standard deviation of measurement noise. We assume that the noise in the image is independent of the pixel brightness *Z*, so it is not significant when operating with relative values. Higher SNR value accounts for more reliable measurements, so we use the weighting function

$$w(Z) = \frac{g(Z)}{g'(Z)}$$

Since $g(Z)$ is a polynomial, its derivative is well defined (and easy to calculate). So, at the end, our high dynamic range image will contain, for every pixel position *p*, the value

$$E_p = w(Z_{p,DARK}) \frac{g(Z_{p,DARK})}{e_{DARK}} + w(Z_{p,BRIGHT}) \frac{g(Z_{p,BRIGHT})}{e_{BRIGHT}}$$

As in Debevec’s case, the algorithm was implemented using a look-up table, but in a slightly different way, because here we use the values from both pixels to get the final irradiance value. In our case the look-up table should map two values of pixel brightness (corresponding to the dark and the bright pixels) into one irradiance value. So in our look-up table construction we go through each possible pair of pixels from the [0…255] range and calculate the irradiance value for that pair.

With the aid of this table, the high dynamic range image construction is even more trivial than for Debevec’s algorithm. We sequentially acquire two different images using two different shutter speeds. Then, we go through the images pixel-by-pixel (we process both images at the same time). For each pixel position, we look up the value of mapped irradiance that corresponds to the pair of pixel intensities encountered at this position in the brighter and in the darker image. This mapped irradiance value goes into the mapped high dynamic range image, which will be displayed on the screen.

5. High Dynamic Range Display and User Interaction

The program has been implemented in OpenGL, which gives a good opportunity for low-level display control. The high dynamic range image was displayed using OpenGL function *glDrawPixels*, which sends the pixel array to the framebuffer.

The program works in real time, and, while it's running, the user can control some parameters of the video camera and some displaying options. These are: the image which is displayed (we can toggle between the darker image, the brighter image and the resulting HDR image), the low and high shutter speed settings of the camera, and the colour/black-and-white display mode.

If the colour mode is enabled, the algorithm works as follows:

- the images from the camera are aquired in RGB;
- the images are translated to YCrCb colour space. The Y component represents luminance, and the two other components represent chromaticity;

- the high dynamic range image is constructed, using the Y component. Then, the chromaticity component of the brighter image is added to each corresponding pixel;
- the high dynamic range image is converted from YCrCb to RGB and sent to the framebuffer.

6. Performance

The program was compiled and tested on a computer with Athlon 1.2 GHz processor and a NVIDEA GeForce2 video card. Microsoft Visual Studio was used to compile the program.

I have measured the frame rates using the OpenGL timer. Not surprisingly, frame rates for black-and-white and colour modes were markedly different, because, in order for the program to work in colour mode, it has to perform conversions between RGB and YCrCb colour spaces for each pixel. This slows down the execution by approximately a factor of two: in black-and-white mode, the average frame rate was 30.6 frames per second, and in colour mode – 13.4 frames per second.

Performance can be somewhat improved if we use the optimization option of the compiler. The code was compiled on Microsoft Visual Studio C/C++ compiler, which can optimize the compilation in order to maximize speed. If this option is used, frame rate of the colour mode improves to 15.9 frames per second. Performance in black-and-white mode is not affected noticeably.

This measurement of frame rate consists of measuring the time intervals required for acquisition of frames, processing (high dynamic range image constructio) and consequent display on the screen. I have also made further measurements of time interval required for the processing part alone. Results can be summarized as following:

Processing method	Processing time, in milliseconds
Debevec's algorithm	
b/w, without optimization	19.0
colour, without optimization	64.9
b/w, with optimization	10.9
colour, with optimization	54.3
Mitsunaga's algorithm	
b/w, without optimization	19.5
colour, without optimization	66.8
b/w, with optimization	6.0
colour, with optimization	54.1

Bibliography

[1] P.E. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs”, in *SIGGRAPH 97 Conference Proceedings, Computer Graphics Annual Conference Series, 1997,* Aug. 3-8 1997, pp. 369-378.

[2] T. Mitsunaga and S. K. Nayar, "Radiometric Self Calibration", in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, June, 1999.