G. Yourganov, W. Stuerzlinger

Tone-Mapping for High Dynamic Range Images

Technical Report

Dept. of Computer Science, York University

December 2001

1. Image composition

For the acquisition of high dynamic range image, we employed two different shutter speed settings: 1/60 sec. (the default shutter speed) and 1/500 sec. The images obtained with the 1/60 sec. and 1/500 shutter speed will be referred to as the “bright” image and the “dark” image, respectively. Then for both shutter speeds we estimated the radiance values for each pixel (this was done according to Paul Debevec’s paper):

$$\ln E_i = g(Z_{ij}) - \ln ?\, t_j \qquad (1)$$

Here, Z_{ij} is the pixel at position i and shutter speed j, $? t_j$ is the jth shutter speed (in seconds), g is the response function of the camera, and E_i is the estimated radiance.

Then, for each pixel we had to decide whether we should use the radiance value form the “bright” or from the “dark” image. It was observed that if the pixel is saturated, its intensity value is 230. So we can calculate the radiance of the scene point that causes the saturation in the “bright” image:

$$\ln E_{SAT} = g\,(230) - \ln ?\, t_{BRIGHT}$$

So, the points with the radiance of E_{SAT} and above saturate the “bright” image. Keeping this in mind, we can construct a high dynamic image from the two images. If the pixel is not saturated, we use the intensity in the “bright” image to calculate the radiance according to the formula (1). If the pixel is saturated, we use the intensity from the “dark” image. Using this method, there are no sharp transitions between the areas that were calculated from the “bright” and the “dark” image (actually these transitions are not noticeable at all).

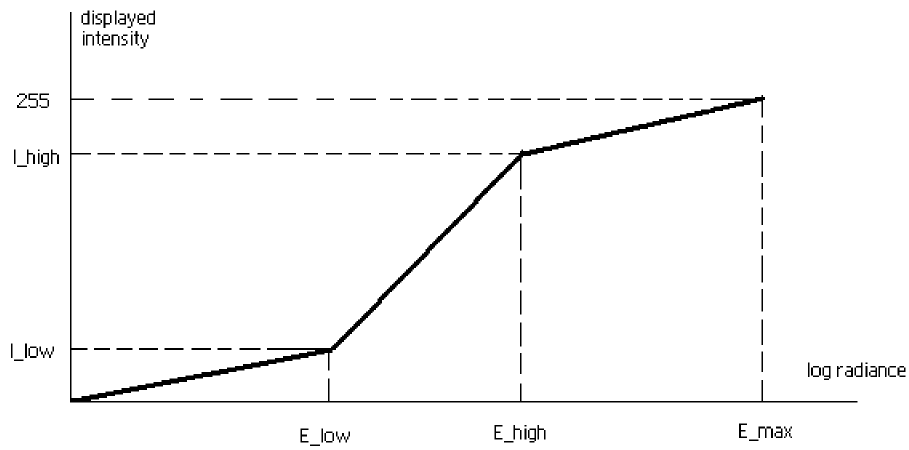
For details see: G. Yourganov, W. Stuerzlinger, Acquiring High Dynamic Range Images in Real Time, Technical Report, Dept. of Computer Science, York University, May 2001.

2. Tone mapping

The result of the image composition is the array of the same size as the image, and this array holds the radiance values for each pixel in the image. The radiance values are floating point numbers with quite large range (in our case,we observed the range [0..1200]; the range of log radiance was [1..7.1]). The next task is to map this radiance range to the displayable range, which is [0..255]. There are several ways we can do this:

a) “*linear mapping*”: brutally scale the radiances to the [0..255] range. This method produces awful results, because most of the information of the image is in the low radiance range, and it gets mapped to a rather small intensity range. Practically, on the resulting image you can only see the light sources well.
b) “*logarithmic mapping*”: take the log of the radiances and then linearly scale to [0...255]. This works *way* better and can actually be used to display the high contrast image (I think it has something to do with the fact that human visual perception is logarithmic, although I am not sure). Actually, Debevec’s formula (1) already gives us the value of log radiance, so in my program I used that.

c) “*sigmoid logarithmic mapping*”: instead of linearly scaling the log radiances, we scale them according to the following “sigmoid” function:



So the radiances below E_low and above E_high are, in a sense, “compressed”, and the radiance range [E_low … E_high] is “expanded”. We have to make sure that relatively little information in the image is above and below that range, because there will be some loss of details. At the end, the result appears “sharper” than with simple logarithmic mapping. In general, the radiance values in the range [0..E_low] are linearly mapped to [0..I_low], the range [E_low … E_high] is mapped to [I_low … I_high], and everything above is mapped to [I_high … 255]. With a reasonable choice of parameters (E_low, I_low) and (E_high, I_high), this method works quite well.

d) “*sigmoid linear mapping*”: for the sake of completeness, we can apply the same idea of sigmoid mapping to the linear mapping in (a). We have to set E_high to be relatively small, so the range of intensities above E_high is effectively compressed (so the user sees something other than light sources in the image). This method works quite better than straightforward linear mapping, but still worse than the logarithmic mappings.

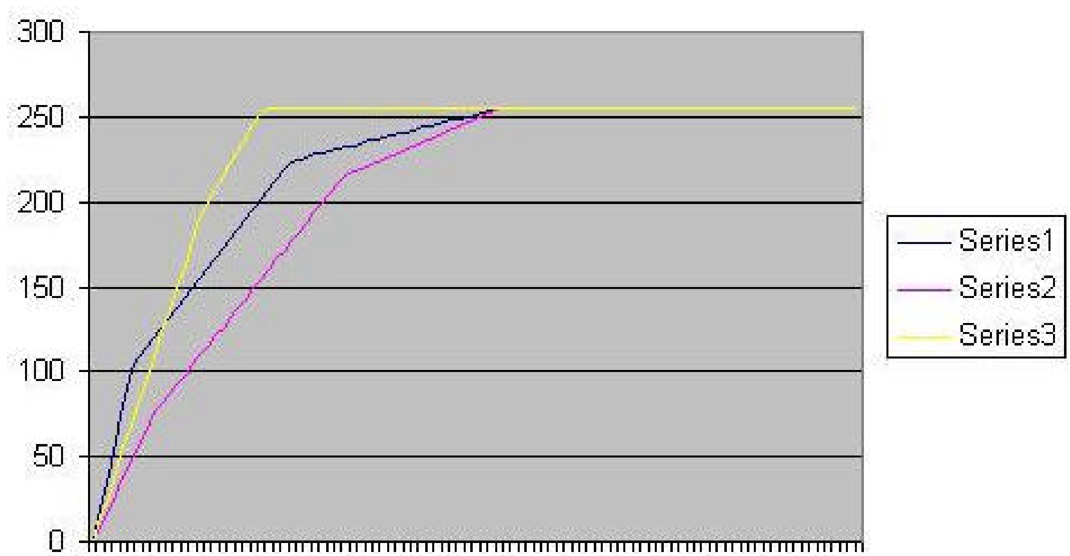
We conducted a series of user tests, where the users were to choose which tone mapping looks better, and most of the users consistently preferred sigmoid logarithmic tone mapping (although I must repeat that the parameters for the sigmoid curve must be carefully selected if we want good-looking results).

3. User Test 1: choosing the best way of tone mapping

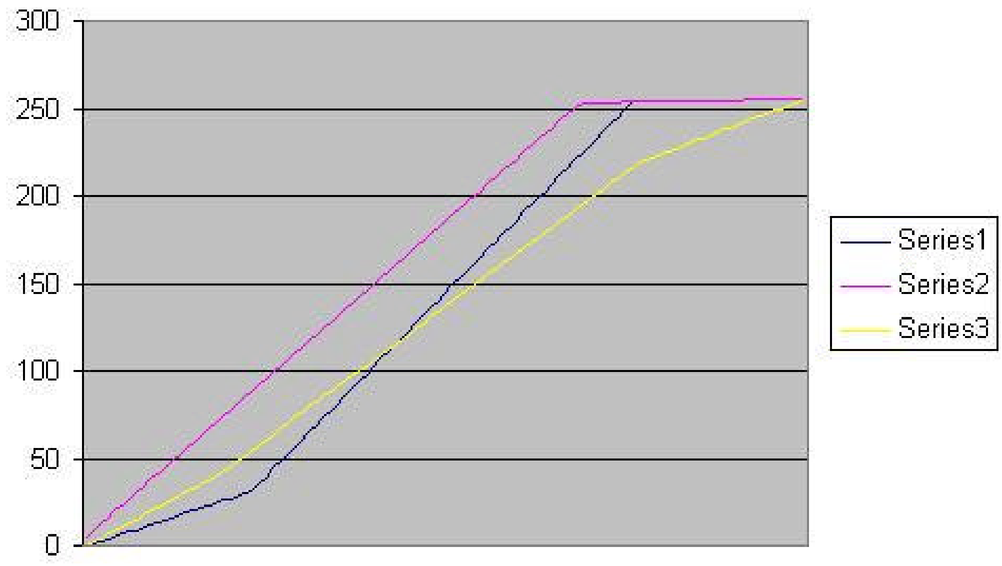
For the test, we have prepared a bunch of high dynamic range images. All images displayed the same scene, and the “bright” and “dark” images used to calculate HDR images were also the same, so the only things different about the images were the methods of tone mapping. We have used sigmoid logarithmic and sigmoid linear tone mappings with different parameters of the sigmoid curve; we also used a “bright” image of the scene, taken with the shutter speed 1/60 sec. (the default shutter speed of the camera); and to spice things up we also used histogram equalization method. For the last part, we took one straightforward linear and one straightforward logarithmic image, loaded them into the "xv" image viewer and used their "Histogram equalization" option (this method gives you the maximum contrast).

Here is the set of the images (the actual images are included in the appendix):
#1: the original "bright" image of the scene, taken with the 1/60 sec. shutter speed (the default shutter speed of the camera);
#2, #3, #4: sigmoid linear mapping with different parameters of the sigmoid curve;
#5: linear mapping with histogram equalization;
#6, #7, #8: sigmoid logarithmic mapping, again with different mapping parameters;
#9: logarithmic mapping with histogram equalization.

Our preliminary tests showed that straightforward linear tone mapping looked really bad compared to other ways, so we didn't use it for the test. For the images #2, #3, and #4, the sigmoid curves looked like this:



Here, the blue line shows mapping used for image #2, the pink line – for #3, and the yellow line – for #4. And here are the sigmoid curves for images #6, 7 and 8:



Here, the blue line shows the mapping used in image #6, the pink line – for #7, and the yellow line – for #8. So image #8 used mapping that was close to a straight line (which means linear scaling), image #6 used a somewhat more contrast-enhancing mapping, and image #7 compressed the high radiance range without compressing the low radiance range.

Fifteen users were tested. The users were asked to rate the image on the scale of 1 (worst) to 9 (best). To get complete results and statistical analysis, please refer to Joe Elek's report. I can only say that overall the sigmoid logarithmic mapping was the winner. Image #6, the contrast-enhancing sigmoid logarithmic mapping, was picked as the best by all but one user (user #13). The users weren't as unanimous with picking the second-best image, but on average image #8 had the second best rating (7.0). Images #7 and #1 got average ratings of 5.87 and 5.93. Image #1 represents the snapshot with the camera at default shutter speed with no image composition or tone mapping, so we can say that all images with sigmoid logarithmic tone mapping were at least as good (#7) or significantly better (#8 and #6) than the images directly fro the camera. To me personally, that means that the whole thing is worth doing.

Now, about the "losers". The sigmoid linear mappings performed really bad, all users (except for one, #13) have picked image #3 as the worst, and all users (again, except for #13) have picked #4 as second-worst. Image #2 got the third lowest average grade, although the users weren't as unanimous.

Now, for the histogram equalization. Both #5 and #9 received average rating somewhere in the middle (5.07 and 5.33 respectively). Most users gave them a moderate rating, although some users rated them quite high – two users picked #5 as second best, and one user picked #9 as second best. As for the notorious user #13, the histogram equalization images were the favourite: image #9 was picked as the best, #6 – as second best, and #5 – as third best.

There seemed to be no correlation whatsoever to the users' gender or background (we have tested psychology and computer science students), although 15 users weren't enough to make a conclusion that no correlation exists.

I could note another thing. If you make your tone-mapping close to a straight line (like image #8), you get a good preservation of details, although the contrast of the image is not the best. If your tone mapping curve is more sharp (like image #6), your contrast is enhanced, but some of the details might be lost (because of compressing the low and the high radiance ranges). Finally, histogram equalization enhances the contrast to the maximum, at the expense of the detail preserving. In our tests (this test, the test #2 and the preliminary tests), we seemed to observe two groups of users: some people like the preservation of details, and the other people value more the "sharpness" of images; this last group is apparently smaller (but it includes me). Among the users of our tests, three users (#4, #13 and #15) preferred the contrast-enhancing tone mappings; the users #2, #7 and #12 gave the logarithmic mappings the highest grade, and in our case logarithmic mapping is the best for detail preservation. But again, fifteen tests are not nearly enough to make any far-reaching conclusions.

4. User Test 2: Selecting parameters for tone mapping

We expected that sigmoid logarithm tone mapping would outperform other kinds of tone mapping, and picking the parameters for the sigmoid curve is a very important issue; se we have conducted another test, where the users were asked to select the mapping curve parameters themselves.

The users were asked to pick the parameters E_low, E_high, I_low and I_high. They were given some initial mapping curve, and they used the mouse to move the points (E_low, I_low) and (E_high, I_high) until they liked the quality of the image (the image was re-mapped each time they changed the parameters).

In general, users tended to like a "smooth" tone mapping, where the points were lying more or less close to the straight line. In that case, we can see more details in the image. Other users set the parameters further from the straight line, so the image appears "sharper", but the preservation of the details is not as good.

14 users were tested. The results were *very* scattered, so I wouldn't say that there is a certain way that all users like.

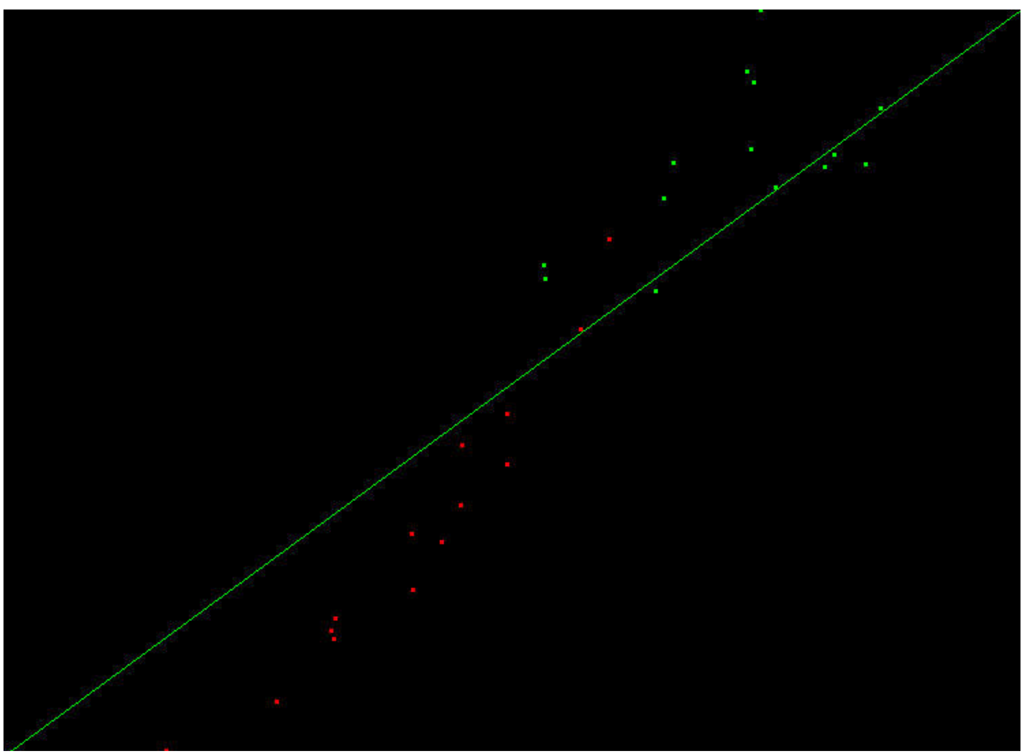
E_low	E_high	I_low	I_high
2.870774	5.416194	56.56364	193.3364
3.531052	5.760687	116.3727	200.2909
3.531052	4.5741	99.21819	158.1
4.248746	4.631516	175.7182	189.6273
3.205698	5.243947	85.30909	206.3182
3.071728	4.6985	72.79091	201.6818
2.325327	6.153026	46.82727	220.2273
1.913849	5.21524	18.54546	232.7455

2.315758	5.827671	39.87273	204.4636
3.215267	3.798991	105.7091	162.2727
4.047791	6.047764	145.1182	201.2182
1.13874	5.310932	1.854545	253.6091
2.861205	3.789422	75.57272	166.9091
2.296619	5.263086	42.65455	229.0364

Here is the statistical data:

	E_low	E_high	I_low	I_high
Average	2.898115	5.123648	77.29481	201.4169
Median	2.971251	5.253517	74.18182	201.45
Std. Deviation	0.84434	0.746163	48.27077	27.26021

Here is the plotted data. Red dots correspond to points (E_low, I_low), and green dots – to points (E_high, I_high). The straight line represents linear scaling (where the radiances are linearly mapped to [0 … 255]).



The user preferences are too scattered to make any conclusions, except for that there is no universal way that all users like. Again, some users like the “sharpness” of images, whereas the others value the preservation of details more.

Acknowledgment

Thanks to Joe Elek for his help with the user test

Appendix. The images for Test 1

The short description is below each image.

Image #1 (the “bright” image form the camera)

Image #2 (sigmoid linear mapping)

Image #3 (sigmoid linear mapping)

Image #4 (sigmoid linear mapping)

Image #5 (linear mapping with histogram equalization)

Image #6 (sigmoid logarithmic mapping)

Image #7 (sigmoid logarithmic mapping)

Image #8 (sigmoid logarithmic mapping)

Image #9 (logarithmic mapping with histogram equalization)