

# High Dynamic Range Image Display Using Level Set Framework

Ruifeng Xu  
University of Central Florida  
rxu@cs.ucf.edu

Sumanta N. Pattanaik  
University of Central Florida  
sumant@cs.ucf.edu

## ABSTRACT

This paper presents a novel solution to the High Dynamic Range (HDR) image compression problem using level set framework. Using level set framework, this method separates the HDR image into detail and profile. Then it uses a global tone mapping function to compress the profile. Finally it produces a low dynamic range version of the image for display by adding the details to the compressed profile. This method is capable of compressing the dynamic range while retaining the detail in the final image of various HDR images in a short time.

## Keywords

Level set method, sigmoid function, high dynamic range image display.

## 1. INTRODUCTION

High Dynamic range images have become more and more popular and important in computer graphics research and applications [Deb97a]. Debevec proposed a simple method to compose high dynamic range images from a series of common photographs with different exposures [Deb97a]. High dynamic range images are a natural outcome of various renders [War97a].

However, the high dynamic range image also brings challenges for display, because the existing presentation devices, including CRT monitors, printers, liquid crystal display, have low dynamic ranges. A simple linear tone mapping will cause loss of much detail. So, advanced range compression techniques are necessary to present high dynamic images on an ordinary low dynamic range display in a visual realistic way [Dic01a].

The problem of high dynamic range image compression can be formulated as a non-linear dynamic range compression problem satisfying the following conditions [Tum99a].

- Keep details undisturbed as much as possible
- Compress profile into a lower dynamic range

Using this formulation, the whole signal can be seen as composed of a profile and the detail. The profile contains low frequency information and is smoothly changing. But the detail contains high frequency information. An intuitive approach to compress the signal is to – compress the profile, but retain the

details. The first step to use this approach is to separate the profile and the detail from the signal. Tumblin et al [Tum99b] developed an LCIS based approach to carry out this separation. In this paper we propose a simpler approach based on level set method.

## 2. RELATED WORK

A lot of work has been done on this topic over the past years [Ash02a, Dur02a, Fat02a, Pat98a, Rei02a, Rah96a, Sch94a, Tum99b, War97a]. For a detailed review, refer to [Dic01a, Rei02a]. In this section we will restrict our discussion only to the work of [Tum99a] which is more relevant to our work.

### Tumblin's LCIS

Based on the observation that artists are able to create very realistic painting starting with the profile of the background and then addition detail to it, Tumblin et al [Tum99b] introduced the novel idea of separating the detail of an image from its profile, range compressing the profile to fit into the limited display range and finally adding the detail to generate realistic appearance. Their method makes use of anisotropic diffusion to extract details from the image at various spatial scales and makes use of a global non-linear compression operator for range compression of the profile. Only drawbacks of this method are: the method generates excessive detail at times; uses at least 8 parameters in the detail generation process which makes the method a little formidable for user control. This may be the main reason why the method has not received wide popularity.

### Range Compression

One of the key steps in High Dynamic Range image display is the range compression. Various functions, such as, logarithm, square root and S-shaped (Sigmoid-function) functions have been proposed to carry out this compression. While all of these

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Journal of WSCG, Vol.11, No.1., ISSN 1213-6972*  
*WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.*  
Copyright UNION Agency – Science Press

functions are approximations of the response of the visual system to the intensity of light, S-shaped function seem to be a more accurate [Hun95a] representation of the response of the visual system to an intensity range. Various researchers [Sch95a, Tum99b, Pat02a, Rei02a] have used such a function for intensity range compression. This function plateaus at higher and lower end of the intensity range, but gives linear response in about 2–3 log unit of intensity around the ambient intensity of the scene. Figure 1 shows a typical S-function. The major difficulty with an S-function is that it overly suppresses the detail in both high and low luminance areas of the image. As we see in Figure 1, the log luminance between  $[2, +\infty)$  and  $(-\infty, -2]$  are compressed into a small response region  $[0.88, 1)$  and  $(0, 0.12]$ , respectively. And the detail in regions where log luminance greater than 2 or less than -2 will also be suppressed into a small region of 0.12. The greater the luminance, the heavier the penalty of the associated detail. However, our method tries to keep the detail by separating the detail before range compression and adding back detail after the range compression.

The framework of our method is similar to the framework presented used in [Tum99b]. We use the luminance contrast as the details. Using level set method, we separate the profile, which corresponds to relatively low frequency information, from the detail, which corresponds to relatively high frequency information. And compress only the profile within low dynamic range. Level set method is a simple, mature, and powerful tool. It has been successfully used in many fields, like image denoising and shape recovery [Osh88a, Mal96a, Set96a, Osh00a]. Some recent work, [Enr02a, Mus02a, Nyu02a] shows its potential usefulness in many areas of computer graphics. Our detail extraction using Level Set method is simple and fast. The amount of detail is controlled by a single parameter and hence the method is user friendly. We make use of a S-function function (discussed in the earlier paragraph) in our work. As our method separates the detail before range compression and adds back detail after the range compression we are able to keep most of the detail.

The method is described in detail in the next section.

### 3. HDR COMPRESSION IN LEVEL SET FORMULATION

In the Level Set framework an image can be viewed as a surface composed of profile and details. The level set method is used to compute the profile of this surface. Detail in the image can be computed from the original image and the profile. We will first discuss the general compression process, then

describe the level set method and finally give some experimental results.

#### General Compression process

There are five steps to do the HDR image compression in our method, and the whole compression process is illustrated in Figure 2.

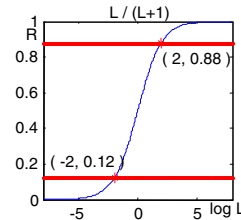


Figure 1. Sigmoid-function

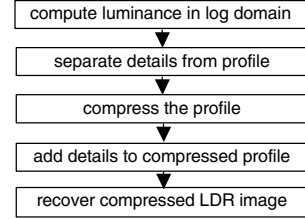


Figure 2. General HDR compression process

In our method, we compress only the luminance value, but keep the chrominance of each pixel. The original HDR image may take any format, like RGBE, TIFF, Loguv, etc. We will first extract the luminance from each pixel, then compress them, and recover pixel color from the compressed luminance at last. The method to compute luminance depends on the color space used to represent the original HDR image. For RGB color, we use the method by Greg Ward to compute the luminance, as shown in equation 1.

$$I(R, G, B) = 0.265 * R + 0.670 * G + 0.0648 * B \quad (1)$$

The log luminance values are processed by level set method for profile computation. The computation technique will be discussed in detail on next subsection.

After obtaining the profile from the luminance surface, we compute the details by subtracting profile from surface, as shown in equation 2. Note that the subtraction in logarithm domain is equivalent to quotient before taking logarithm.

$$I_{detail} = \frac{I_{image}}{I_{profile}} \quad (2)$$

The profile is compressed using the S-shaped function shown in equation 3 [Pat00a].

$$R(I) = R_{max} \frac{I^n}{I^n + \sigma^n} \quad (3)$$

where,  $I$  is pixel intensity or luminance,  $R$  is the compressed luminance ( $0 < R < R_{max}$ ),  $\sigma$  is the semi-saturation constant of  $I$  that causes the half-maximum response, and  $n$  is the sensitivity control parameter. Figure 1 is a plot of this equation with  $\sigma$ ,  $R_{max}$  and  $n$  as 1. The value of  $R_{max} \leq 1$  maps the luminance values to a displayable range of  $[0, 1]$ . The value of  $\sigma$  can be computed from ambient intensity values using equations proposed in [Pat00a]. For convenience, following [Rei02a], we

have taken the median of the luminance map divided by 0.18 as our  $\sigma$  value.

The compressed luminance map is recovered by adding the details to the compressed profile.

### Computing profile using Level Set method

The level set method was first formulated by Osher et al [Osh88a] as a general method to manipulate the implicit surface. Ever since, it is being extensively used and studied as a powerful tool in various areas like physics, fluid mechanics, material, computer vision, and computer graphics. We apply this method to extract the details from the luminance map.

The luminance map in logarithm domain is denoted as a super-surface  $I$  over 2-D domain. The motion of the surface is described by partial a differential equation (PDE) as shown in equation 4.

$$I_t + F(\kappa)|\nabla I| = 0 \quad (4)$$

and the initial surface is determined by equation as shown in equation 5

$$I = I(x, y, t = 0) = 0 \quad (5)$$

where,  $I$  is the luminance value at pixel  $(x, y)$ ,  $t$  is the time,  $I_t$  is the derivation over time,  $F(\kappa)$  is the moving speed,  $\kappa$  is surface curvature at point  $(x, y)$ , and  $\nabla I$  is the first order partial derivation over  $x$  and  $y$ .

Since we want to compress the original logarithm luminance, we define our speed function as equation 6.

$$F(\kappa) = -e * \kappa$$

$$e = \exp\left(\frac{-sen *}{\sqrt{|D_x^+ - D_x^-|^2 + |D_y^+ - D_y^-|^2}}\right) \quad (6)$$

where  $sen$  is the sensitivity to the edge,  $D^{-x}, D^{+x}, D^{-y}, D^{+y}$  are first order forward and backward finite differences along the  $x$  and  $y$  axes respectively. Figure 3 shows the plot of the measure of “edgeness” as a function of the gradient change around pixels. “edgeness” here is used to describe the probability of finding an edge around pixels.

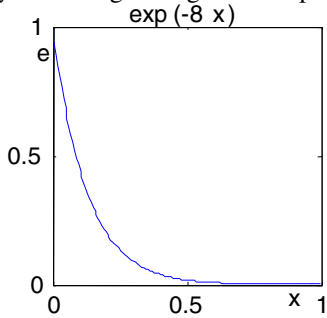


Figure 3. A function used for “edgeness”

Curvature faces towards the “protrusion”, so the minus speed moves the surface inside, so as to suppress the “protrusion”.  $e$  is used here as a measure of probability of an edge. The bigger of  $sen$ , the more sensitive the speed is to the edge. In our experiment, we use 8.0 for  $sen$ , and it works fine in all the test cases.

The numerical solution to equation 4 is shown in equation 7 [Osh88a].

$$I_{xy}^{n+1} = I_{xy}^n - \Delta t [\max(F_{xy}, 0) \nabla^+ + \min(F_{xy}, 0) \nabla^-] \quad (7)$$

where,  $I_{xy}^n, I_{xy}^{n+1}$  are the luminance at point  $(x, y)$  at step  $n, n+1$  respectively,  $\Delta t$  is the time step.

$$\nabla^+ = [\max(A, 0)^2 + \min(B, 0)^2 + \max(C, 0)^2 + \min(D, 0)^2]^{1/2} \quad (8)$$

$$\nabla^- = [\max(B, 0)^2 + \min(A, 0)^2 + \max(D, 0)^2 + \min(C, 0)^2]^{1/2}$$

$$A = D_{xy}^{-x} + \frac{\Delta x}{2} m(D_{xy}^{-x-x}, D_{xy}^{+x-x})$$

$$B = D_{xy}^{+x} - \frac{\Delta x}{2} m(D_{xy}^{+x+x}, D_{xy}^{+x-x}) \quad (9)$$

$$C = D_{xy}^{-y} + \frac{\Delta y}{2} m(D_{xy}^{-y-y}, D_{xy}^{+y-y})$$

$$D = D_{xy}^{+y} - \frac{\Delta y}{2} m(D_{xy}^{+y+y}, D_{xy}^{+y-y})$$

and,

$$m(x, y) = \begin{cases} \begin{cases} x, & |x| < |y| \\ y, & |x| > |y| \end{cases}, & xy \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

In equation 9,  $D_{xy}^{-x}, D_{xy}^{+x}, D_{xy}^{-y}, D_{xy}^{+y}$  are first order backward and forward finite difference along the  $x$  and  $y$  axes respectively.  $D_{xy}^{-x-x}, D_{xy}^{+x-x}, D_{xy}^{+x+x}, D_{xy}^{-y-y}, D_{xy}^{+y-y}, D_{xy}^{+y+y}$  are second order backward, central and forward finite difference along  $x$  and  $y$  axis's respectively.

There are two problems in this numerical approach. First, equation 7 can be solved iteratively, but how to determine when the iteration should be stopped? Second, how big is the  $\Delta t$  appropriate? If  $\Delta t$  is too small, more iterations are needed to get the same profile; vice versa, if too big, the computation will generate vibration and introduces noise. According to our experiments, it is found when the original luminance in logarithm domain ranges about 10, it is appropriate to use a  $\Delta t$  takes a value between 0.1 and 0.2. We map the original logarithm luminance into range  $[0, 10]$  before level set computation, and recover its original range after the computation. So,

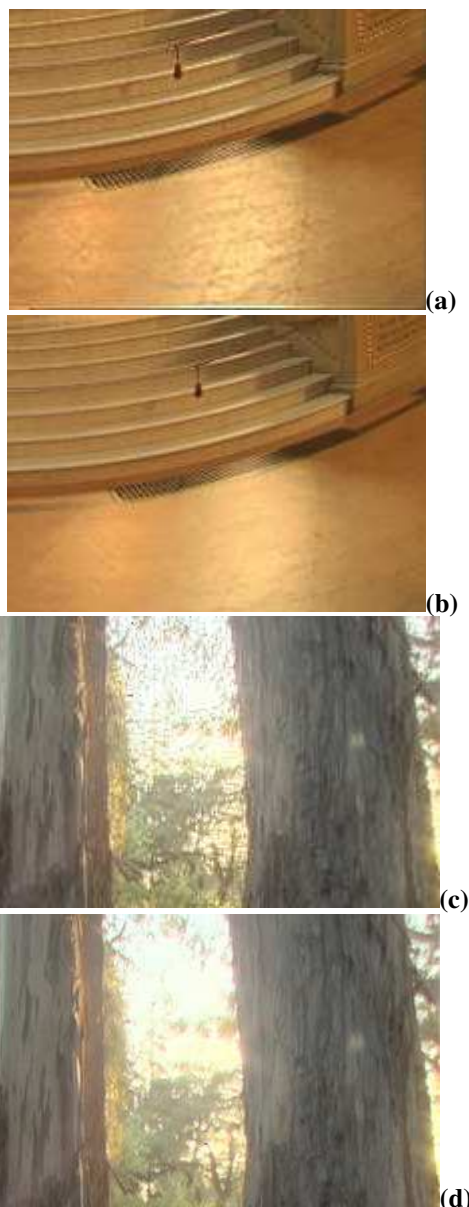
we can always use a value between 0.1 and 0.2 for  $\Delta t$ . The results shown in this paper has been generated taking a value of 0.12 for  $\Delta t$ . As to the first problem, the more iteration, the more detail will be generated. We provide it as a parameter to the user to control the amount of detail. This is the only parameter of our approach, and images are shown to illustrate it in the next section. *imgmanf* in our implementation is used to compute the profile using level set method. The Matlab6.0 implementation of *imgmanf* is available in Appendix A, and the whole algorithm for high dynamic range display are available in Appendix B.

#### 4. EXPERIMENTAL RESULTS

Some experimental results are shown in this section to verify our approach. Our method takes less than one minute for most of our test images on a 1.7G Pentium processor with 256M memory. The “memorial” image shown in Figure 5a, which contains 768×512 pixels, took only 23 seconds on our platform. The results show more desirable detail in the final images, like highlights and texture. The images (a) and (c) of Figure 4 are the result from our algorithm, and the images (b) and (d) are results from applying the same S-shaped function to the whole image. The images (b) and (d) of Figure 4 are what one would get by using Reinhard’s method [Rei02a] without local adaptation. For Figure 4(a) and (b) (part of Memorial image), the floor texture is apparent in the image on Figure 4(a), but not apparent in the image on Figure 4(b). For Figure 4(c) and (d), you can also find many leaves between trunks in the image on Figure 4(c), which is lost in the image on Figure 4(d).

Figure 5 shows the comparison between results using LCIS and our method. Note that our method does not have the problem of excessive detail as can be seen in the image generated using LCIS method. More results using our method are available in Appendix C.

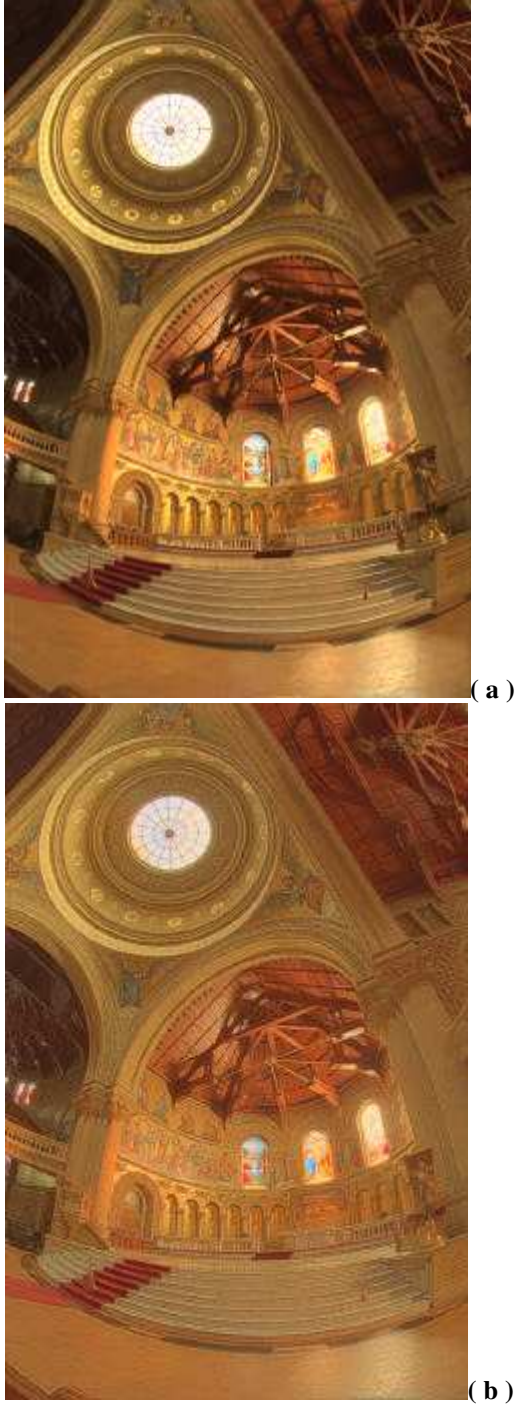
Our method has only one parameter, *nstep*, which controls how much detail in the final image. This parameter generally takes value between 5 and 50. The larger the parameter value, more the detail added to the final image. In our experiments, 15 was sufficient for most of the test images. Figure 6 shows the result as a function of the value of this parameter. The image in Figure 6(a) contains the least detail, while image Figure 6(d) contains the most detail among the images in the figures. The four detail maps in Figure 7 correspond to the compression results in Figure 6 respectively. The size of detail map in Figure 7 is 480×720, and the time to compute them on our platform (1.7G Pentium processor) are 0 secs, 4 secs, 7secs, and 13 secs, respectively.



**Figure 4. comparison of images resulting from our method (a and c) vs. resulting from a method using S-compression alone (b and d).**

#### 5. DISCUSSION AND CONCLUSION

We presented a High Dynamic Range image display method based on extraction of detail using level set method and the range compression function used in [Pat00a, Rei02a]. The results are better than the application of a single range compression function on the whole image. The range compression function compresses the detail in the plateau area. However, our method avoids this problem by compressing the range in the profile image and restoring the previously extracted detail. Our method is faster than the local adaptation based approach proposed [for example: Rei02a]. Our approach can be applied to most HDR images and obtain desirable results in one or two minutes.



**Figure 5: comparison of results (a) using our method (b) using LCIS method.**

The fundamental approach used in our method is very similar to the approach proposed in [Tum99b]. Profile extraction using Level Set method is simple, easier to understand, and more computationally efficient than anisotropic diffusion used in [Tum99b]. During our work, we also find some further questions. We use one parameter to control the amount of details. But, how much details will make the results look best? Too much detail or too little detail both spoils the final results. We think the

solution is closely related to the perception of human vision. We also find noise sometimes exists with the details, and so one has to be careful not to accentuate the influence of the noise.



**(a) nstep = 0**



**(b) nstep = 3**

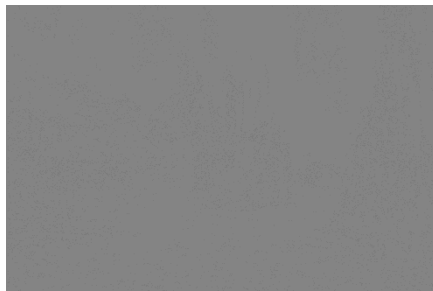


**(c) nstep = 7**

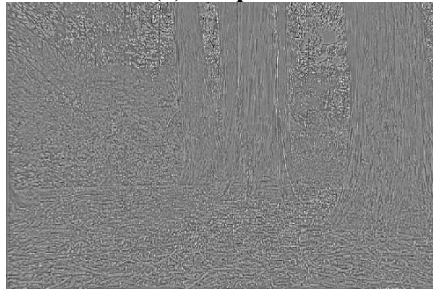


**(d) nstep = 14**

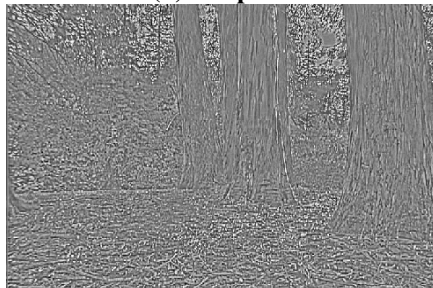
**Figure 6. Images taken with different parameter *nstep* Images (a) to (d) have *nstep* as 0, 3, 7, 14. Image (a) contains the least detail, and image (d) contains the most detail.**



(a)  $nstep = 0$



(b)  $nstep = 3$



(c)  $nstep = 7$



(d)  $nstep = 14$

**Figure 7.** Details extracted using our method from “grove” image are shown here. (a), (b), (c), and (d) show the details for  $nstep=0, 3, 7$  and  $14$ , respectively.

## 6. ACKNOWLEDGEMENT

This work was partially funded by ARMY-STRICOM.

## 7. REFERENCES

- [Ash02a] Ashikhmin, M. A Tone Mapping Algorithm for High Contrast Images. 13<sup>th</sup> Eurographics Workshop on Rendering (2002), pp.1-11,2002.
- [Deb97a] Debevec, P.E., and Malik, J. Recovering High Dynamic Range Radiance Maps from Photographs. Proceedings of SIGGRAPH97, pp.369-378,1997.

- [Dic01a] DiCarlo, J. M., and Wandell, B. A. Rendering high dynamic range images. Proceedings of the SPIE: Image sensors, 3965, pp.392-401,2001.
- [Dur02a] Durand, F., and Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images. Proceedings of SIGGRAPH02, pp.257-266,2002
- [Eis47a] Eisenhart, L. P. An introduction to differential geometry with use of the tensor calculus. Princeton University Press,1947.
- [Enr02a] Enright, D. P., Marschner, S. R., and Fedkiw, R. P. Animation and Rendering of Complex Water Surfaces. Proceedings of SIGGRAPH02, pp.736-744,2002.
- [Fat02a] Fattal, R., Lischinski, D., and Werman, M. Gradient Domain High Dynamic Range Compression. Proceedings of SIGGRAPH02, pp.249-256,2002.
- [Hun95a] Hunt, R. W. G. The Reproduction of Color. Fountain Press, England,1995.
- [Mal96a] Malladi, R., and Sethian, J. A. A unified Approach to Noise Removal, Image Enhancement, and Shape Recovery. IEEE Transaction on Image Processing, 5(11), pp.1554-1568,1996.
- [Mus02a] Museth, K., Breen, D. E., Whitaker, R. T., and Barr, A. H. Level Set Surface Editing Operator. Proceedings of SIGGRAPH02, pp.330-338,2002.
- [Ngu02a] Nguyen, D. Q., Fedkiw, R. P., and Jensen, K. W. Physically Based Modeling and Animation of Fire. Proceedings of SIGGRAPH02, pp.721-728,2002.
- [Osh88a] Osher, S., and Sethian, J. A. Fronts propagating with Curvature-Dependent Speed: Algorithm Based on Hamilton-Jacobi Formulations. Journal of Computational Physics 79, pp.12-49,1988.
- [Osh00a] Osher, S., and Fedkiw, R. P. Level Set Methods: An Overview and Some Recent Results. Technical Report,2000.
- [Pat98a] Pattanaik, S. N., Ferwerda, J. A., Fairchild, M.D., and Greenberg, D. P. A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display. Proceedings of SIGGRAPH98, pp.287-298,1998.
- [Pat00a] Pattanaik, S. N., Tumblin, J.E., Yee, H., and Greenberg, D.P. Time-Dependent Visual Adaptation for Realistic Real-Time Image Display. Proceedings of SIGGRAPH00, pp.47-53,2000.
- [Pat02a] Pattanaik, S. N., and Yee, H. Adaptive Gain Control for High Dynamic Range Image Display. Proceedings of Spring Conference in Computer Graphics (SCCG2002), Budmerice, Slovak Republic, April 24-27, 2002.
- [Rei02a] Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. Photographic Tone Reproduction for Digital Images. Proceedings of SIGGRAPH02, pp.267-276,2002.
- [Rah96a] Rahman, Z., Jobson, D. J., and Woodell, G. A. A Multiscale Retinex for Color Rendition and Dynamic Range Compression. SPIE International Symposium on Optical Science, Engineering, and Instrumentation, Conference on Signal and Image Processing XIX, 2847,1996.
- [Sch94a] Schlick, C. Quantization techniques for visualization of high dynamic range pictures. Proceedings of 5<sup>th</sup> Eurographics Workshop, June 1994, pp.7-20,1994.
- [Set96a] Sethian, J. A. Level Set Methods: evolving interfaces in geometry, fluid mechanics, computer

- vision, and materials science. Cambridge University Press,1996.
- [Smi90a] Smith, J.A. 2-D signal and image processing, Stanford,1990.
- [Soc00a] Socolinsky, D. A. Dynamic Range Constraints in Image Fusion and Visualization. In Proceedings of Signal and Image Processing 2000, Las Vegas, November 2000.
- [Tum93a] Tumblin, J., and Rushmeier, H. Tone reproduction for realistic images. IEEE Computer Graphics and Application, 13(6), pp.42-48,1993.
- [Tum99a] Tumblin, J., Hodgins, J. K., and Guenter, B. K. Two methods for display of high contrast images. ACM Transactions on Graphics 18(1), pp.56-94,1999.
- [Tum99b] Tumblin, J., and Turk, G. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In Proc. Proceedings of SIGGRAPH99, pp.83-90,1999.
- [War97a] Ward, G. L., Rushmeier, H., and Piatko, C. A visibility matching tone reproduction operator for high dynamic range scenes. IEEE Transactions on Visualization and Computer Graphics, 3(4), pp.291-306,1997.

## Appendix A

The Matlab6.0 code of level set method for computing the profile is given below.

```

%%%this function implements Eqn 7.
function [rev]=imgmanf(im,delt,nstep)
%%%intialize vars
xdim = size(im,1); %%size of image
ydim = size(im,2);
ll = 1; rr = xdim; %%boundaries of image
bb = 1; tt = ydim;
phi = zeros(rr,tt); %%local variables
nphi= zeros(rr,tt);
Npoint = (rr+tt)/2;
delt = 1/Npoint; dely = 1/Npoint;
eps = 1.0;
%%%Initializing original front
phi = init(im,ll,rr,tt,bb);
while nstep>0
    %%%compute new front
    nphi= phi_1(phi,ll,rr,tt,bb,delt,
        delx,dely,eps);
    phi = nphi;
    nstep = nstep - 1;
end
%%%return profile
rev = phi(ll:rr,bb:tt);
%%% initialize original level curve
function [phi]=init(im,ll,rr,tt,bb)
    for xx = ll:rr
        for yy =tt:bb
            phi(xx,yy) = im(xx,yy);
        end
    end
end
%%compute curvature using central difference
function [rev]=curv(phi,j,k,ll,rr,tt,bb)
    phixx = (phi(j+1,k)- 2.0*phi(j,k) + phi(j-1,k) );
    phiyy = (phi(j ,k+1) - 2.0*phi(j,k) + phi(j ,k-1));
    phixy = (phi(j+1,k+1) + phi(j-1,k-1) - phi(j-1,k+1) -
        phi(j+1,k-1))/4.0;
    phix = (phi(j+1,k ) - phi(j-1,k ))/2.0;
    phiy = (phi(j ,k+1) - phi(j ,k-1))/2.0;
%%%mean curvature
rev = (-phixx*(1.0+phiyy*phiy)+2.0*phiy*phix*phixy-
    phiyy*(1.0+phix*phix))/((1.0+phix^2+phiy^2)^1.5);
%%% solve PDE for one time step
function [rev]=phi_1(phi,ll,rr,tt,bb,delt,delx,dely,eps)

```

```

newphi = zeros(rr, tt);
for yy = bb:tt
    for xx = ll:rr
        cu = curv(phi,xx,yy,ll,rr,tt,bb);
        spd = - eps*cu;
        if spd > 0.0 tg = g_HJ_plus(phi,xx,
            yy,delx,dely,ll,rr,tt,bb);
        else tg = g_HJ_minus(phi,xx,yy,
            delx,dely,ll,rr,tt,bb);
        end
        newphi(xx,yy) = phi(xx,yy) - spd*delt*tg;
    end
end
rev = newphi;
%%% compute numerical flux function
function [rev]=g_HJ_plus(phi,j,k,delx,dely,ll,rr,tt,bb)
    tA = A(phi,j,k,delx,dely); tB = B(phi,j,k,delx,dely);
    tC = C(phi,j,k,delx,dely); tD = D(phi,j,k,delx,dely);
    rev = -sqrt(1.0 + min(tA,0)^2+max(tB,0)^2
        + min(tC,0)^2+max(tD,0)^2);
%%% compute numerical flux function
function [rev]=g_HJ_minus(phi,j,k,delx,dely,ll,rr,tt,bb)
    tA = A(phi,j,k,delx,dely); tB = B(phi,j,k,delx,dely);
    tC = C(phi,j,k,delx,dely); tD = D(phi,j,k,delx,dely);
    rev = -sqrt(1.0 + min(tB,0)^2+max(tA,0)^2
        + min(tD,0)^2+max(tC,0)^2);
%%% compute derivative A of x at j,k
function [rev]=A(phi,j,k,delx,dely)
    DmDpPhi = phi(j+1,k) - 2.0*phi(j ,k) + phi(j-1,k);
    DmDmPhi = phi(j ,k) - 2.0*phi(j-1,k) + phi(j-2,k);
    rev = phi(j,k)-phi(j-1,k)
        + delx/2.0*m(DmDmPhi,DmDpPhi);
%%% compute derivative B of x at j,k
function [rev]=B(phi,j,k,delx,dely)
    DpDpPhi = phi(j+2,k) - 2.0*phi(j+1,k) + phi(j, k);
    DpDmPhi = phi(j+1,k) - 2.0*phi(j ,k) + phi(j-1,k);
    rev = phi(j+1,k)-phi(j,k)
        - delx/2.0*m(DpDpPhi,DpDmPhi);
%%% compute derivative C of y at j,k
function [rev]=C(phi,j,k,delx,dely)
    DmDpPhi = phi(j,k+1) - 2.0*phi(j,k) + phi(j,k-1);
    DmDmPhi = phi(j,k) - 2.0*phi(j,k-1)+ phi(j,k-2);
    rev = phi(j,k)-phi(j,k-1)
        + dely/2.0*m(DmDmPhi,DmDpPhi);
%%% compute derivative D of y at j,k
function [rev]=D(phi,j,k,delx,dely)
    DpDpPhi = phi(j,k+2)-2.0*phi(j,k+1)+phi(j,k) ;
    DpDmPhi = phi(j,k+1)-2.0*phi(j,k) +phi(j,k-1);
    rev = phi(j,k+1)-phi(j,k)
        - dely/2.0*m(DpDpPhi,DpDmPhi);
%%% compute m for second order
function [rev]=m(x,y)
    if x*y>=0.0
        if abs(x)<=abs(y) rev = x; else rev = y; end;
    else rev = 0.0;
end
%%%end

```

## Appendix B

The Matlab6.0 code of High Dynamic Range image display using Level Set method proposed in this paper is given below.

```

function [ ] =lsHDRC(filename, nsteps)
%%%set up parameters
delt = 0.15; % Δt
nsteps = 15; recommended
sensitivity = 8.0; %sen, for edge sensitivity
vision = 0.73; % for vision, the smaller, the more
% accurate of the vision
CIE_rf=.265074126; CIE_gr=.670114631;
CIE_br=.064811243; MIN_LUM = 1.0e-3;

```

```

%%Read in the luminance image and store in r, g, b
[r g b]=rgbereadrgb(rgbefilename);
%%Get luminance value
im = CIE_rf*r + CIE_gr*g + CIE_br*b;
im(find(im==0))=MIN_LUM;
%% transform luminance into logarithm domain.
loglum = log(im);
%% extract profile using LS =>prologlum
prologlum = imgmanf(loglum,delt,nsteps,nnsteps,sensitivity);
meanlum = exp(median(loglum(:)));
n = vision;
factor = exp(prologlum).^(n-1) ./(exp(prologlum).^n
+ (meanlum/0.18)^n);
%%recover low dynamic range image =>r,g,b
r = r .*factor; g = g .*factor; b = b .*factor;
%%show compressed image
tiffwrite(r,g,b,lhrfilename);
%%OK

```

### Appendix C. More examples using our method



(b)



(a)



(c)

The above images are compressed using our method. And the high dynamic range images (a),(b), can be found in <http://www.cs.ucf.edu/~reinhard/cdrom/hdr.html>  
 And (c) was kindly supplied to us by SpheronVR.