**AE-HDR: an Automatic Exposure Framework for High Dynamic Range Content**

Francesco Banterle*
University of Warwick

Paolo Banterle†
University of Siena

(a) Tone Mapped HDR image (b) Exposure Map $E_{map}$

(c) The density function $p(i,j)$

(d) The final result

---

**Abstract**

Automatic exposure (AE) is a common method implemented in consumer cameras to calculate the best exposure that tries to minimize under- and overexposed pixels for a scene. We present a general framework for an AE algorithm for high dynamic range (HDR) imaging inspired by tone mapping operators (TMO) and pointing devices. Compared to other methods such as [Neumann et al. 1998] we propose a more general framework and an approach that enables exploring images and videos in different areas of the image/frame.

**Keywords:** High Dynamic Range Imaging, Tone Mapping

1 **The Framework**

Our framework can be divided in three steps: exposure map calculation, exposure calculation and motion stabilization. During exposure calculation the objective is to calculate the correct exposure for each pixel. Firstly the HDR luminance $L_w$ is tone mapped to $L_d$ using a TMO, Photographic operator [Reinhard et al. 2005] with automatically estimated parameters. Secondly, the exposure map is calculated as $E_{map} = L_w / L_d$. In the second part, we calculated the final exposure that will be used for scaling the HDR image. We decided to calculate the final exposure as a weighted average of a region of interest (ROI), with an arbitrary shape, that needs to be analyzed. The use of only the exposure in the center of the ROI could cause flickering if $C$ takes a high luminance value during video playback. We defined a density function $p(i,j) \in (0,1]$ in the ROI and $p(i,j) = 0$ outside it, which represents weights for the average. The value of $p(i,j)$ is normalized ($\int p(i,j) d(i,j) = 1$). So the final exposure $e$ is defined as $e = \sum_{i,j} p(i,j) E_{map}(i,j)$. In our implementation we used a circle as the ROI with position $(c_x, c_y)$ and radius $r = \min(w,h)/10$ where $w$ and $h$ are respectively the width and height of the image. Finally the density function was modeled with a Gaussian function $p(i,j) = \frac{1}{\sigma^2} \exp\left(-\frac{(i - c_x)^2}{\sigma^2} + \frac{(j - c_y)^2}{2\sigma^2}\right)$ where $\sigma = r/3$. When the ROI is moved fast between a very dark region and a very bright region, flickering can occur. To avoid this problem we introduced motion stabilization as the last step of our framework. We modeled it using a simple average of $n$ exposures calculated in the last 1/4s, a value that we found to give good results from our experiments. The new exposure is calculated as follows $e' = \sum_{i=0}^{n} e_i \exp\left(-\frac{(i - c_x)^2}{\sigma^2}\right)$, where $e_0$ is the current exposure, and $\sigma^2 = 0.25$ is the temporal variance. The main application of our framework is an easy tool for exploring HDR images or videos. Moving the ROI with a pointing device enables users to visualize regions of interest without the problem of changing the exposure for an area that is needed to explore. Furthermore the application of an eye-tracker would improve the experience allowing automatic pointing and simulating real world experience. The advantages are mainly two: reduction of color shifting caused by TMOs and speed because only a linear slice of information is needed for each ROI.

We implemented our framework on the GPU using shader model 3.0, and it performed more than 200fps for images and videos at 1080p on an Intel Pentium 4 HT 2.8GHz, 1Gb of Ram, and GeForce 7800 with 256 Mb of ram. A second possible application is the visualization of HDR images or video on an HDR display. As shown in [Akýüz et al. 2007] a LDR image or video can provide an HDR experience using inverse tone mapping [Banterle et al. 2006]. Therefore, an AE for HDR images/videos can be a proper solution. Furthermore, our framework can be seen as a compression tool, indeed HDR content can be processed using our framework and stored using common LDR standards. In particular for HDR videos the ROI can be positioned in a way that follows the application of an eye-tracker would improve the experience allowing automatic pointing and simulating real world experience. The advantages are mainly two: reduction of color shifting caused by TMOs and speed because only a linear slice of information is needed for each ROI.

---

*e-mail: f.banterle@warwick.ac.uk
†e-mail: banterle@unisi.it

---

**References**


