# Mountable Dynamic Range Enhancer for Digital Cameras

Cheng-Han Wu[1], Po-Nan Li[2], and Gordon Wetzstein[*]
Department of Electrical Engineering, Stanford University
350 Serra Mall, Stanford, CA 94305, USA
[1]chw0208@stanford.edu; [2]liponan@stanford.edu
[*]Project adviser

## Abstract

*We designed and implemented an innovative dynamic range enhancing device that can be mounted on digital cameras and allows photographers to obtain a high dynamic range photo in single shot without post-processing. A liquid-crystal array is programmed to spatially modulate incident light and optically tone-map the camera module for dynamic range enhancement, preserving the high-light details that were originally over-exposed.*

## 1. Introduction

High dynamic scenes are one of the most challenging scenarios encountered by photographers. Originally developed in Taiwan, the black card technique is a post processing-free solution to partial exposure reduction in bright areas of the scene. By shaking a black card in front of the lens, photographers can reduce local exposure of the scene thus obtain a photo with balanced exposure within single shutter release; nevertheless, such technique would requires practice and several trials. Latest HDR features on mobile device cameras take multishots, which requires static imaging condition. Inspired by the black card technique, we have designed and implemented a programmable liquid crystal (LC) based filter for digital cameras. Through real-time modulation in front of lenses, highly localized exposure reduction can be realized with a computer-generated mask for a certain scene and create an optical tone mapping effect without using burst photography and post processing.

## 2. Related work

### 2.1. Liquid-crystal glass

Combined with polarizers, liquid-crystal (LC) glasses have been widely used to modulate color and light intensity for display applications. Previous usage of LC arrays as photography filters can be found in [1] where a programmable binary liquid crystal array (LCA) is used as a coded aperture. However, to our best knowledge there are currently no related application of using LCD as lens filters for cameras for exposure or color modulation.

### 2.2. Tone mapping

#### 2.2.1 Software solution

Popular High Dynamic Range (HDR) techniques such as [2] and [3] use post-processing algorithms to merge photos with different exposure together. Either global tone mapping or local tone mapping weight photos in the stack to extract maximum feature. For global tone mapping, an optimal function is determined for the scene and weights are assigned to individual photos for merging. On the other hand, local tone mapping uses a pixel-wise calculating algorithm to extract maximum details for each pixel, which can be formulated as

$$\hat{X} = \exp\left[\frac{\sum_i w^{(i)}(\log(I_{lin}^{(i)}) - \log(t^{(i)}))}{\sum_i w^{(i)}}\right], \tag{1}$$

where $\hat{X}$ is the estimated pixel value, $I_{lin}$ is the original value before gamma correction, $t$ is the exposure time, $w$ is the computed weight for each pixel, defined by

$$w^{(i)} = \exp\left[-4\frac{(I_{lin}^{(i)} - 0.5)^2}{0.5^2}\right]. \tag{2}$$

Such tone-mapping solutions can be implemented by using an inverse bit-mask. Meylan *et al*. demonstrate that an inverse bit-mask can be used to reduce glare and specular highlights. In this work, an inverse bit-mask is added to videos for HDR display application by using a low-passed luminescence image [6].

#### 2.2.2 Hardware solution

To achieve real-time HDR, Zhao *et al*. at MIT use a different sensor structure to achieve HDR effects, providing a

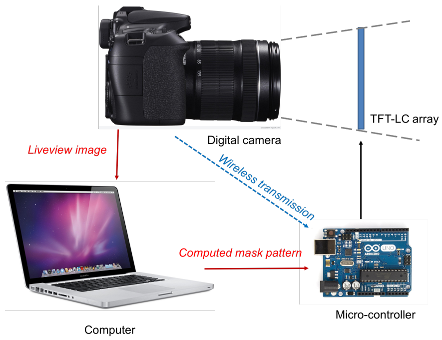



Figure 1: System overview of our post-processing free tone-mapping system.

hardware solution. After a pixel is saturated during exposure, it will be reset to zero for a new exposure instead of staying at saturation. By tone mapping images of a modulo camera with images of a intensity camera, an ultra high dynamic range (UHDR) photo can be produced [4]. Nevertheless such implementation requires ground-up modification or fabrication of the camera sensor, which may not be commercially available for existing platforms .

## 3. Project overview

We proposed to design and implement a device that can help cameras capture single shot HDR photo without post-processing. Mounted in front of the lens, this device uses a programmable transparent LC panel to project a mask fitted to the scene, achieving real time tone-mapping. Real time "live view" from a camera is captured for imaging processing on a computing unit, e.g. a computer or a micro-controller. The computed mask is then transmitted via the controller (Arduino) and projected onto the LC panel, as shown in our system schematic (Fig. 1). This mask can locally and dynamically modulate incident light intensity thus enhance the photo's dynamic range in real-time.

## 4. Implementation

### 4.1. Experimental Setup

To realize our design, we have built three generations of prototypes, which are described as follows.

#### 4.1.1 1st generation: color LCD

Our first attempt was to build a mountable device for digital single-lens reflex (DSLR) cameras. To enable continuous tone-mapping, we used a 5-inch $800 \times 400$ color TFT-LCD purchased from Adafruit Industries[1]. We removed the LCD panel's backlight module and linear polarizers on each side of the glass to obtain a "transparent" LC glass (Fig. 2). Nevertheless, we found it unfeasible to employ this LCD glass because of the tiny cell structures of TFTs (thin film transistors). Fig. 5 (a) shows the micrograph of the color LCD used. Each pixel, of size $135 \times 135$ $\mu$m, consists of three color filter cells. Those tiny structures, unfortunately, impose diffraction and severely distort the image. An attempt of de-blurring the image taken with the LC glass was made by using deconvolution operation, but was unsuccessful (Fig. 3).

[1]https://www.adafruit.com/products/1680

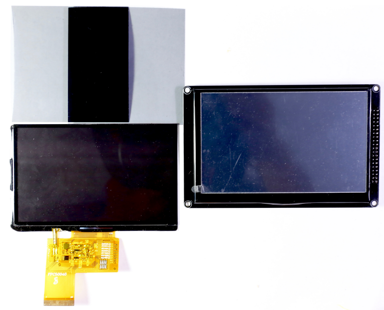


Figure 2: We disassembled a commercial LCD panel (left) and sandwiched the transparent glass with two linear polarizers (right).

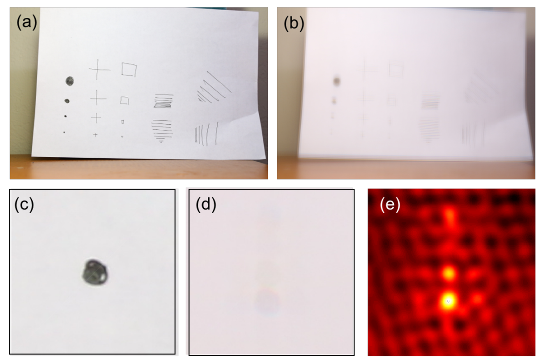
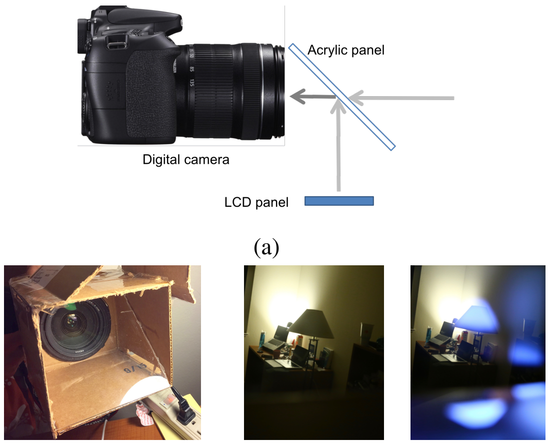



Figure 3: Sample target taken by DSLR without (a) and with (b) mounting LCD glass. (c,d) Zoom-in of the solid circle at the bottom-left of (a) and (b), respectively. (e) Blur-kernel computed from (c) and (d).

#### 4.1.2 2nd generation: augmented photography

Due to the inaccessibility of using the color LCD panel as a filer for it producing too much artifacts, we came up with another idea of using augmented reality (AR) technique (Fig. 4) to project a bright color pattern onto the scene by using a transparent acrylic panel as a beam splitter. The same color LCD was employed again, but as a reflective display device this time, rather than as a tranmissive one. However, such projection is just adding extra brightness onto




(a)

(b) (c) (d)

Figure 4: (a) An AR-inspired approach of projecting color mask onto the scene. (b) Camera setup using Acrylic panel. (c) Original scene. (c) Augmented scene.

the scene instead of reducing bright areas. The other challenge we encountered is that the projected scene should be focused at infinity, which is practically challenging without additional optical components. Finally, this configuration does not post significant effects in bright lighting conditions, which may be an issue since most circumstances where HDR is used have saturating lighting condition.

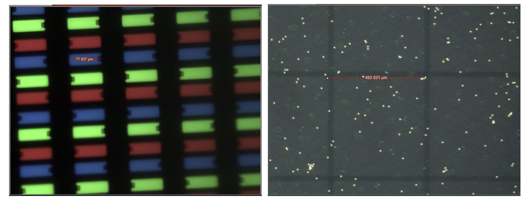
#### 4.1.3 3rd generation: monochromatic LCD

Finally, we go back to the design of our first generation prototype (Sec. 4.1.1), but this time we employ a monochromatic LCD[2] instead of a color one. Since the monochromatic LCD has larger pixel size, as shown in Fig. 5 (b), this prototype is image distortion free. Here we also change the camera to be working with this device from a DSLR to a mobile phone as we found a large size monochromatic LCD might not be commercially available. With this new set-up, we have been able to provide a proof of concept. For example, our first HDR image shown in Fig. 7 demonstrates that the LC filter can locally attenuate higt-lights and the details can thus be preserved. Schematic and working principles will be detailed in following subsections.

### 4.2. LCD control

In all of three generations of prototypes, an Arduino Uno R3 micro-controller is used to drive the LCD. A C++ "server" program is uploaded to and running on the Arduino to standby for external instructions, while a Matlab program running on a computer keeps communicate with the Arduino via the standard serial interface (Fig. 8 (a)).

[2]https://www.adafruit.com/products/250



(a) (b)

Figure 5: (a) Micrograph of a color TFT-LCD used. Each pixel consists of three color filter cells, each with size $28 \times 78$ $\mu$m. (b) Micrograph of a monochromatic TFT-LCD used. Each pixel cell is of size $500 \times 500$ $\mu$m.

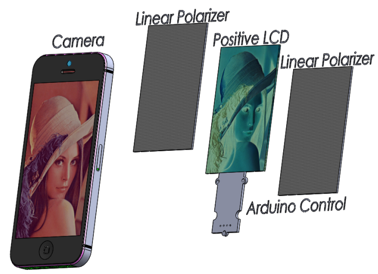



Figure 6: Schematic of LCD control: a naturally white (positive) LCD glass obtained by disassemble a display module. Two linear polarizers are attached to each side of the LCD glass to enable light intensity modulation.

Particularly, to support the $800 \times 480$ color LCD employed in our 1st and 2nd prototypes, an additional driver board, also from Adafruit Industries[3], was utilized for decoding and caching. Fig. 6 exhibits how we use a LCD glass to modulate light intensity. The transparent TFT liquid-crystal glass was obtained by disassembling a display module, and is sandwiched with two plastic linear polarizers[4] on each side.

### 4.3. Real-time image acquisition

To acquire the camera's live view on a mobile phone for further image processing, We use *QuickTime Player* on a Mac to stream an iPhone's screen in real time (Fig. 8 (b)). The iPhone's screen displayed on the Mac is then captured by using an open-source Matlab function *getscreen*[5].

### 4.4. Image processing

Originally we planned to use the tone-mapping algorithm to compute a gray-scale mask [3] so the device can modulate the light intensity accordingly [5]. Later, as we

[3]https://www.adafruit.com/product/1590
[4]Available on Amazon
[5]http://www.mathworks.com/matlabcentral/fileexchange/22031-getscreen

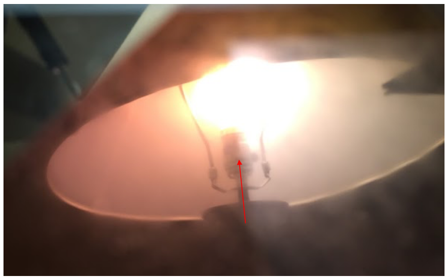
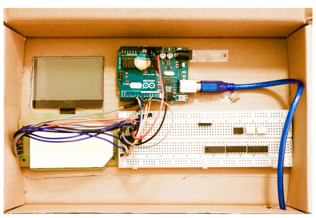
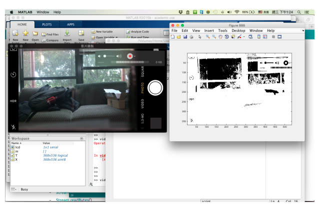


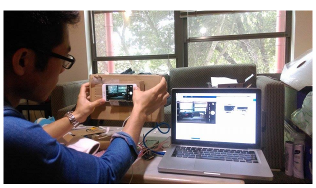
Figure 7: First HDR image, ever, taken with our 3rd generation prototype. A black rectangular filtering mask was applied to left part of the scene. The red arrow annotation indicates the boundary of the mask: the details near the light bulb in the left part is enhanced by our device. This photo was taken with an iPhone 6.



(a)



(b)

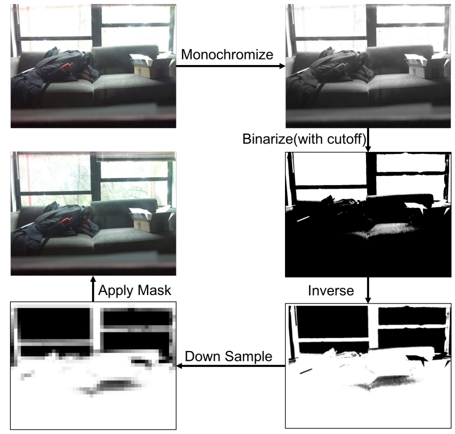


(c)

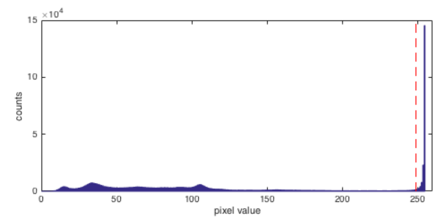
(d)

Figure 8: (a) Mountable mask generating system is packaged in a cardboard box containing an Arduino microcontroller, driver circuit for LCD and USB cable for serial communication. (b) Screen shot of real-time monitoring of image capturing and mask generation on a laptop. (c,d) The 3rd generation prototype is handy for mobile phone users.

replaced the color LCD with a monochromatic one, by which the gray-scale mask is no longer available, the tone-mapping process is reduced to a cut-off determination process, in which the mask pattern is determined by a given cutoff (Fig. 8 (b)). In our implementation, the live view image is converted to gray-scale and its intensity histogram is calculated. The cutoff is set at the pixel value with count number smaller than $1\%$ count number of maximum pixel value, 255 for example for a 8-bit image (originally 24-bit for a color image). Fig. 9 (b) shows the histogram computed from the photo shown in Fig. 10 (a). In this example, the value 255 has $1.5 \times 10^5$ counts, so the cutoff is set at 249,




(a)




(b)

Figure 9: (a) Flowchart for mask application. (b) Pixel value histogram of the picture shown in Fig. 10. Red dashed line indicates the mask-defining cutoff.

which has a count number around $1.4 \times 10^3$, and the mask is therefore defined by positions with pixel value equal or larger than 249.

However in real utilization, the cutoff is fixed to an arbitrary number, say 200, for simplicity to reduce real time computation latency. Moreover, because the LCD we are using only has $128 \times 64$ pixels, and the flash memory available in Arduino is limited (only 2,048 bytes), the computed mask is then down sampled to a low resolution version, and encoded to bit steam so it can be transmitted to Arduino via the serial interface (Fig. 9 (a)).

### 4.5. Dynamic mask generation

Based on the techniques presented in subsections 4.2, 4.3 and 4.4, a filtering mask generation system is realized. In a real scenario, it is desired that the filter mask can change accordingly as the photographer moves the camera. To allow real-time update, i.e. to dynamically change the mask with the scene, a 3 sec cycle is implemented. In the first 0.5 sec, the LC panel is completely transparent, i.e. no mask pattern displayed on the glass, and the un-filtered scene is captured for analysis. Once the mask is computed, the mask will be projected on the LC panel for the following 2.5 sec. Limited by computation and communication latency, as well as

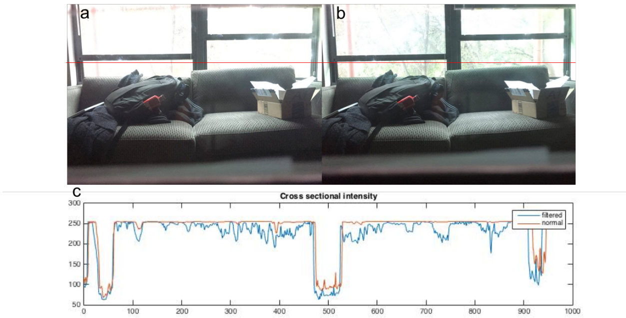



Figure 10: Intensity comparison before and after filtering. (a) Before applying LCD filter. (b) After applying coded inverse bit mask. (c) Pixel values along the red lines marked in (a) and (b) shows detail enhancement after filtering.

the LCD's update rate, this 3 sec updating configuration is known to be a most optimized one. Please refer to the Youtube video on the footnote for a more illustrative explanation of this work flow.

## 5. Results

To demonstrate HDR photography with our final prototype, we have produced a video to explain the working principle and present the results [6]. Below are some selected HDR photos taken with our device, and the result of a calibration test.

### 5.1. HDR images

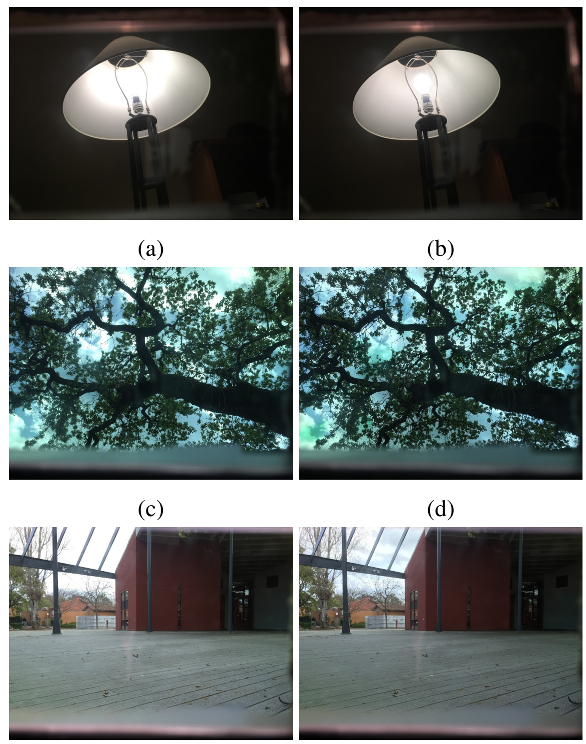
For the starter, we used a fixed rectangular mask to block the bright part in the scene. Fig. 10 (a) shows a photo taken with an iPhone 6, which has much area saturated, and this over-exposure can be fixed by using our device and the said rectangular mask, as shown in Fig. 10 (b).

We then employ a dynamic mask generation system, so the mask can be adaptively generated upon scene is captured. For example, as shown in Fig. 11, an apple-shaped mask is generated because the Apple logo in the scene is identified as an over-exposed area. Here, we intentionally make the mask misaligned, as the arrow in Fig. 11 marked, so the mask can be visible. In real application, the mask can be almost invisible yet it still works amazingly. Fig. 12 exhibits photos taken without or with the use of the adaptive mask for comparison.

### 5.2. Calibrations

To understand how the LC panel changes the scene captured by the camera, we used a setting shown in Fig. 13 (a), where a hand-drawn and a constant lighting were utilized. A freely-available iOS app *VSCO* that allows completely

Figure 11: An apple-shaped mask, which is intentionally misaligned for illustration, is generated in real-time.



(a) (b)
(c) (d)
(e) (f)

Figure 12: Photos taken, (a,c,e) without or (b,d,f) using, an adaptive mask.

manual ISO, shutter speed and white balance was used for taking photos for calibration. Fig. 13 (b), (c) and (d) show the photo taken with the *VSCO* app on an iPhone 6 without putting LC glass, with LC glass but all pixels off, and all pixels on, respectively. We report that the use of the monochromatic LC panel doesn't impair the image quality, as Fig. 13 (e) shows. Though the LC glass introduces a color change, which can only be noticed when the camera's white balance setting is locked, it can be automatically compensated by the camera, and thus would not make any

[6]https://www.youtube.com/watch?v=gOJSiDiE7hM

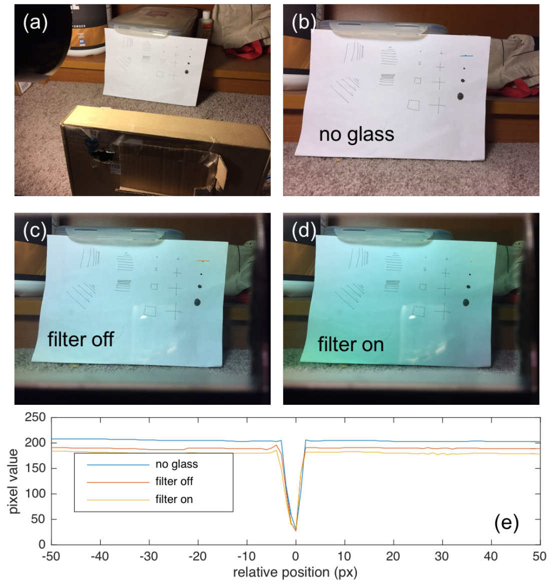



Figure 13: (a) set-up for calibration test, where three photos were taken (b) without LCD glass, (c) with LCD but filter off, and (d) with LCD and filter on, respectively. (e) Line plot of pixel values along lines indicated in panels (b,c,d).

troubles.

## 6. Conclusion

We have demonstrated that optically modifying the scene when taking pictures is able to realize one-shot and post processing-free dynamic range enhancement. Different from traditional HDR techniques, by optically tone mapping local areas in the photo, we are able to extract more detail from saturated area while maintaining overall image quality. Three different experimental setup have been realized to address the problem.

In our first generation system design, diffraction patterns imposed by the color LCD cause heavy distortion that deconvolution can not recover and resulted in poor image quality. On the other hand, in for the second design in which an AR-inspired technique was utilized, we found infinite projection design is hard to achieve and low performance in bright environment is a serious problem.

In the final design, a monochromatic LCD with larger physical pixel size rescues the distortion problem. It works well, and we have shown the adaptive mask projected on the LC panel helps preserve image detail. We further implement a control that keeps monitoring the scene and update the mask, achieving a dynamic system,

Future development based on this design can be made with use of gradient modification where exposure control can be tailored according to different exposure levels. Hardware simplification, such using a Bluetooth wireless module, can also be performed. Moreover, for the presented prototype, the mask generation has to go through three stages, and the Arduino micro-controller as hardware interface induces latency of about 1-2 seconds. By embedding the LC filter in front of the camera and migrating all computations onto a single electronic device, the product can be more compact and the real-time performance can be further enhanced.

## References

[1] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H. Chen, “Programmable Aperture Photography: Multiplexed Light Field Acquisition,” Proceeding ACM Siggraph 2008, 27, no. 55 (2008).

[2] Paul E. Debevec and Jitendra Malik, “Recovering High Dynamic Range Radiance Maps from Photographs,” Proceeding ACM Siggraph (1997).

[3] Frèdo Durand and Julie Dorsey, “Fast Bilateral Filtering for the Display of High-Dynamic-Range Images,” Proceeding ACM Siggraph (2002).

[4] Hang Zhao, Boxin Shi, Christy Fernandez-Cull, Sai-Kit Yeung, and Ramesh Raskar, “Unbounded High Dynamic Range Photography using a Modulo Camera,” ICCP 2015, Houston, TX.

[5] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers, “Inverse Tone Mapping,” Proceeding GRAPHITE ’06, 349 (2006).

[6] Laurence Meylan, Scott Daly, and Sabine Susstrunk, “Tone Mapping For High Dynamic Range Displays,” Proceeding SPIE-IS&T **6492**, 649210 (2007).