VFX project1:
High Dynamic Range Imaging

B01902037 楊孟遠 / B01902039 施楷文

Introduction

High dynamic range (HDR) images have much larger dynamic ranges than traditional images' 256 levels. In addition, they correspond linearly to physical irradiance values of the scene. Hence, they are useful for many graphics and vision applications.

To do HDR imaging, we first have to take photographs under different exposures. Using these photographs and the values of the exposure time, we reconstruct the response curve of each color channel and get the radiance map (save as .hdr file). After we got the radiance map, we do tone mapping on the radiance map to generate the LDR image.

We use C++ with OpenCV 3.1.0 and MATLAB_R2016a to implement the programs, and our programs were tested on Mac OS X 10.11.3.

Image alignment

For image alignment, we use the algorithm based on median(percentile) threshold bitmap. We briefly explain it as follow: first, convert the image into grayscale. Second, transform the grayscale image into a bitmap by thresholding the image using the median of intensities. Then, search for the optimal offset from all possible offset using image pyramid and bitwise xor(and) operations.

Based on the algorithm stated, we also propose a method to handle the hand-held rotation of the exposure sequence. We can choose a range of angles and rotate the grayscale image; then, using the above hierarchy procedure to align the horizontally and vertically.

Figure1 shows the difference bitmap of two different exposure images. We can see the rotation can be eliminated after our method.

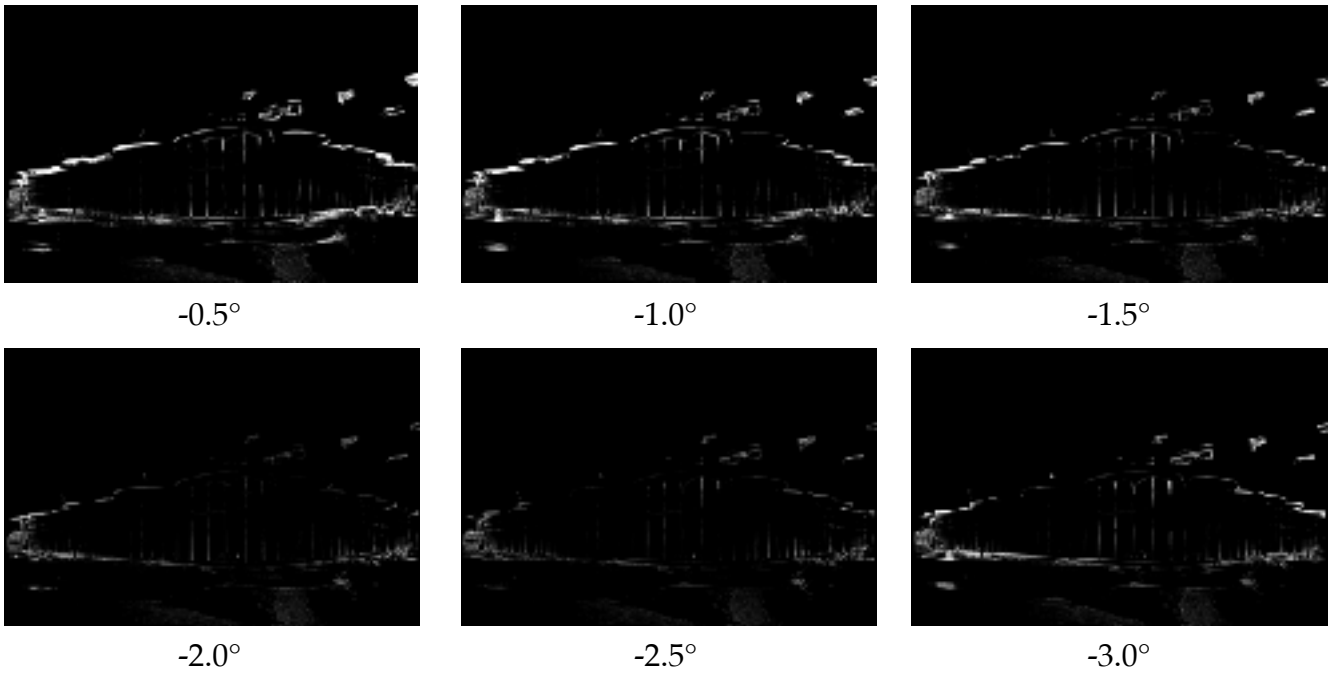


Figure 1. The difference bitmap between two different exposure library images over different degree of rotation.

Assembling HDR image

According to *Debevec*'s method. First, we define function $g = f^{-1}$ which map the final digital values (Z) to radiance and time. Solving the linear equation that minimize the objective function, we can get the function g with very small domain size. As we know the exposure time, we can get the radiance and reconstruct the High Dynamic Range image using its radiance value.

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

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

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

After deriving three colors' g function, we then applying the following formula to construct the radiance map of three color and then get the high dynamic range images

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

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

Tone Mapping

The tone mapping algorithm we use is Photographic Tone Production without implementing the local operator. First, we calculate the intensity of the HDR image. And then we find the Lm for the whole map. Using Lm, we got Ld. That's how we generate the global operator.

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_{x,y}\log(\delta + L_w(x,y))\right)$$

$$L_m(x,y) = \frac{a}{\bar{L}_w} L_w(x,y)$$

$$L_d(x,y) = \frac{L_m(x,y)}{1 + L_m(x,y)}$$

Dealing with local operators, we use Gaussian filter to blur the image and find the largest surrounding area from every pixel. However, we have some problems that make the computing complexity becomes very big, so we don't have remarkable results so far.

$$L_s^{blur}(x,y) = L_m(x,y) \otimes G_s(x,y)$$

$$V_s(x,y) = \frac{L_s^{blur}(x,y) - L_{s-1}^{blur}(x,y)}{2^{\phi}\, a/s^2 + L_s^{blur}}$$

$$s_{max} : \left|V_{s_{max}}(x,y)\right| < \varepsilon$$

An "artistic" HDR image without alignment

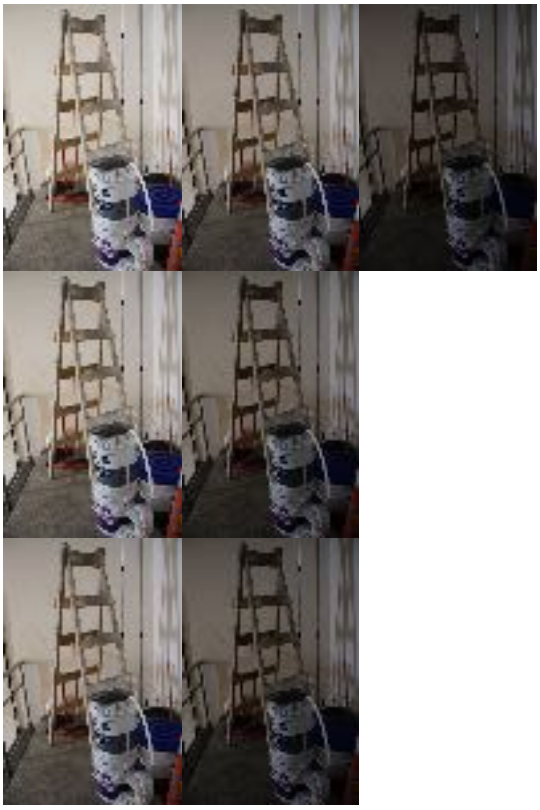
Results

1.

2.

3.

4.



Some Other Results

Reviews and Finding

1. Image Alignment

   • Even if we have the auto-alignment, the images photoed with tripod still have better performance than hand-held ones. The reason may because the bitmap technique cannot handle the different zooming and flipping. Figure 2 show the effect of auto-alignment; however, there are still some small blur in the image.

Figure 2. image without/with auto alignment.

- Threshold bitmap with different percentile threshold may sometimes affect the alignment. In most case, the different threshold result in the same alignment, but this is not the case when the distribution of pixel values are too close to dark or bright. As you may see in Figure 3, the dark side of the house is big, thus result in wrong alignment if we didn't choose the threshold properly.

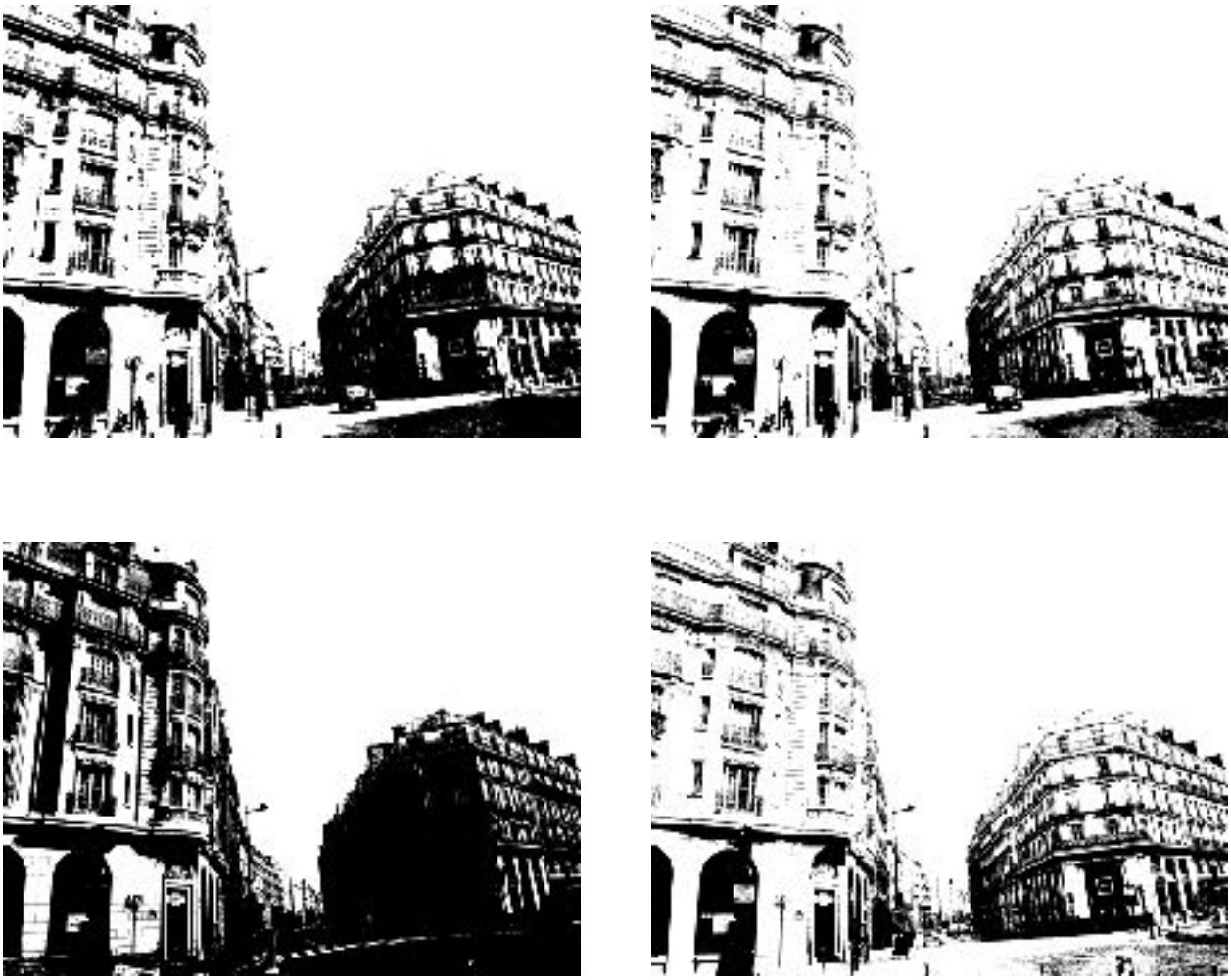
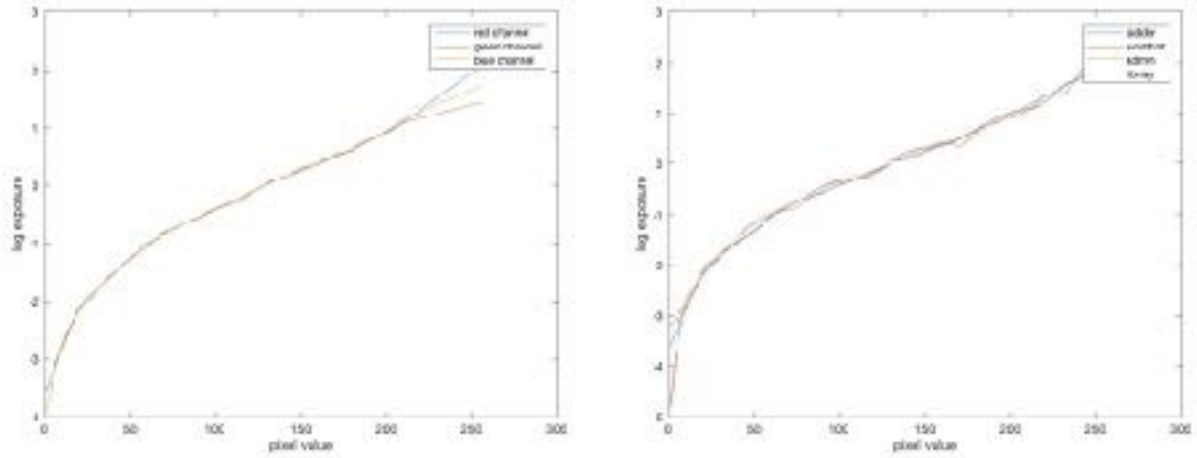


Figure 3. The threshold bit map of two unaligned exposures with 50% threshold(left) and 25% threshold(right) .

2. HDR assembling

• The recovered response curves of one camera are almost the same in spite of different color channels or exposure sequences. Figure 4 show the finding, the curve are very similar with three colour channels and four different exposure sequences. The small different may because of the random sampling of pixels in the exposure sequences.



3. Tone Mapping:

• When we were implementing the Photographic Tone Production, we found that the complexity of generating local operators could be very high if we just brute-force implement it. It would take about an hour to process a small .hdr image (about 800*600). So we should do some smart optimization on it or it would be really time-consuming.

• When doing the Photographic Tone Production, after we get the Ld map, we should recover the LDR image by exponential way instead of just linear recovering it. ex: LDR = Ld map* pow((radiance map/intensity map, gamma)), 0 < gamma < 1 instead of just doing: LDR = Ld map* radiance map/intensity map

• When there's a visible color deviation, we could manually change the gamma value for specific color channel, for example, if the covered LDR image is too red, we can manually reduce the gamma value for red channel slightly.