

# Image Registration for High Dynamic Range Image Generation

Hao Xie, Botao Deng, Tiecheng Su

## Abstract

Merging multiple low dynamic range (LDR) images with different exposures to a High Dynamic Range (HDR) image is a popular method of enhancing both the highlight and shadow details in photography. However, it suffers from the misalignments of LDR images caused by motion of hand-held camera and objects in the imaged scene, which result in blurring effect and ghosting phenomenon respectively.

This project aims at automatically removing those artifacts via image registration. A global registration algorithm based on median threshold bitmap and two local registration algorithms, i.e., weight map based exposure fusion and histogram mapping based ghosting removal, are adopted to counteract the effect of moving camera and objects. Using multiple sets of LDR images, it is verified that the algorithms can effectively enhance the visual effect of final HDR images compared with commercial HDR software.

*Electrical and Computer Engineering, University of Rochester, New York, United State*

## Contents

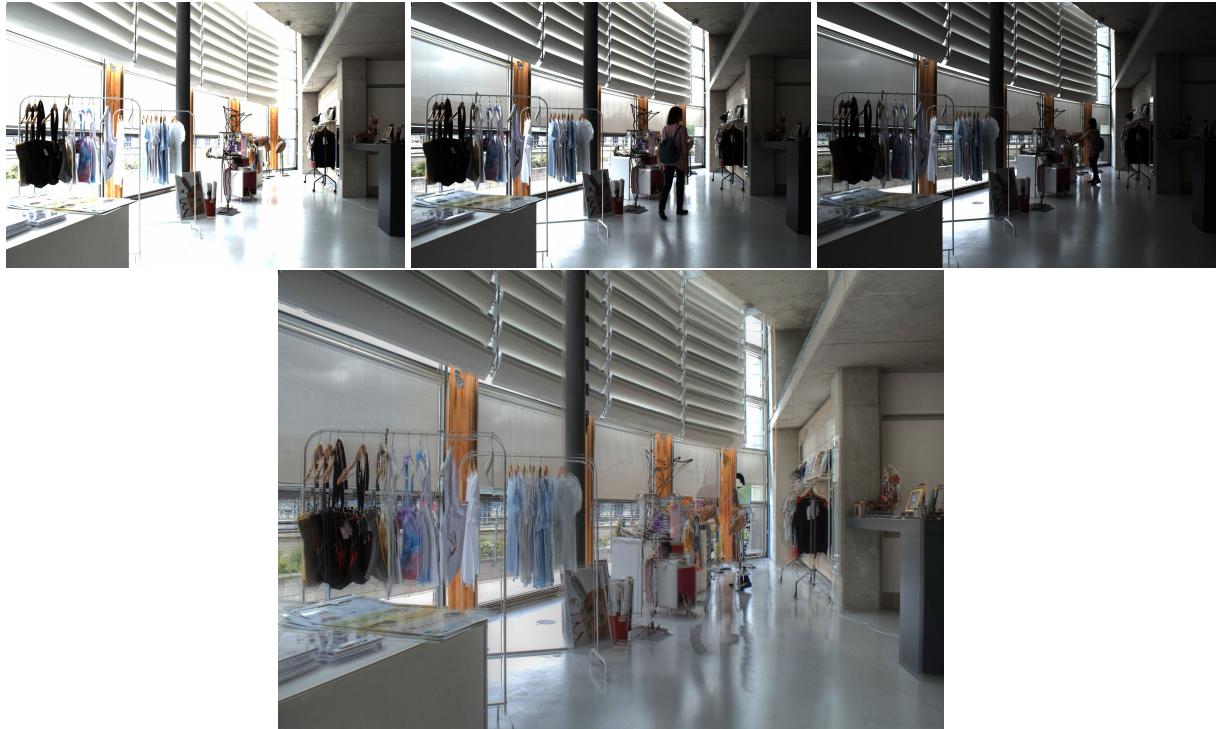
<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Global Image Registration</b>	<b>2</b>
<b>3</b>	<b>Weight Map Based Exposure Fusion</b>	<b>3</b>
3.1	Ghost Detection . . . . .	3
3.2	Weight Map Based Integration . . . . .	3
3.3	Multi-Resolution Image Blending . . . . .	4
<b>4</b>	<b>Histogram Mapping Based Ghost Removal</b>	<b>5</b>
<b>5</b>	<b>Conclusion</b>	<b>6</b>
	<b>References</b>	<b>6</b>

## 1. Introduction

The term “dynamic range” of an image is the ratio between the lightest and darkest pixel[1]. For most commercial cameras, the dynamic range is relatively low due to finite bit length such that there will be under-exposed or over-exposed area when the highlight and shadow coexist in the imaged scene. In fact, the real world can produce a large dynamic range that human visual system is able to adapt to but current digital imaging devices can not represent. To visually resemble the natural world, HDR generation is widely used in photography and computer graphics. The most developed approach like how the HDR mode of iPhone camera works is based on the merging of multiple LDR images with different exposures and thus the well-exposed area from different frames can be well utilized. Because each pixel of the final HDR image can be seen as the sum of weighted LDR images, the registration pre-processing of LDR inputs is required. Otherwise, blurring effect will arise even the camera

moves slightly during capture process. Even worse, there will be ghosting artifacts which look like haunting ghosts when the scene contains moving objects such as walking people and flag in the wind. One example with both camera and object motion is shown in **Figure 1**.

The registration can be divided into two stages – global registration to correct the camera motion and local registration to remove the ghosting. The first stage can be dealt with an algorithm called median threshold bitmap (MTB) method proposed by Ward [2]. It is fast and robust in most situations and have been recommended in most HDR textbooks and articles [1][3]. So, our project adopted this algorithm as our starting point for next ghosting removal stage, which will be discussed in **section 2**. To realize deghosting, many studies have been done in both image processing and computer vision communities. Recently, [3] presented a thorough review about the classification and evaluation of deghosting methods. It concluded that algorithms in [4][5] which fall into the categories of patch based registration and moving object selection have the best performance. However, the patch based and other optical-flow based [6] methods are time-consuming due to advanced computer vision algorithms. In our project, we mainly focused on the class of moving object selection which consists of motion detection and exposure selection. We adopted weight map based exposure fusion and histogram mapping based ghost removal as two parallel deghosting approaches with their strengths. The principles and results will be further described in **section 3** and **section 4** before we make a conclusion in **section 5**.



**Figure 1.** Merging images with different exposures without alignment and deghosting

## 2. Global Image Registration

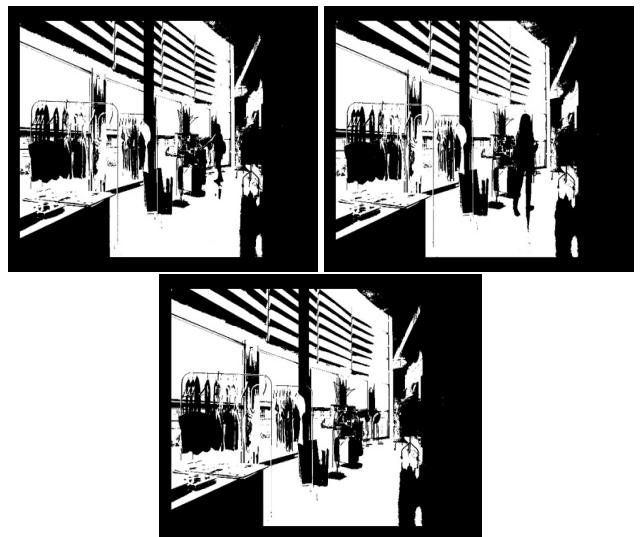
As mentioned in [section 1](#) the camera may move, i.e., translate or rotate, during acquisition of *LDR* images. There are some global registration methods listed below to realize image matching which can be a candidate for reducing the resulting blurring effect.

**Cross-correlation** This is a well developed technique for image matching, very simple to implement and can get sub pixel accuracy. However, it is computationally heavy, especially when the search range is large. To speed things up we can use sub-sampling to get rough alignment before doing alignment at full resolution.

**Edge Detection** The procedure is that applying edge detection technique to the images, then comparing the edge position. However, the *LDR* images with different exposures often contain under-exposed or over-exposed area, which is hard for edge detection due to low intensity differences. So this method is not what we are looking for.

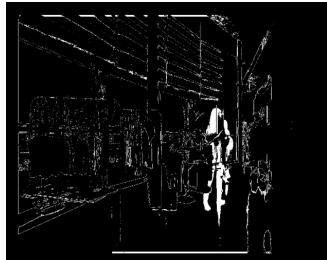
**Median Threshold Bitmap (MTB)** Unlike edge detection, it is found that the results of binarizing the *LDR* images using each median gray level are independent of exposure ([Figure 2](#)). Note that the results keep constant except the area of walking people. We find that *MTB* method is robust enough to be unaffected by small moving area. Then, we can compute the difference between two bitmaps with an *XOR* operator to find where the two images are misaligned ([Figure 3](#)). Because we use three exposures as the input *LDRs*, we select

the mid-exposed image as the reference, and the others as the testing images.



**Figure 2.** Median threshold bitmaps of different exposures

To speed up the computation of overall offset, we adopt a pyramid multi-scale technique. We start with the lowest resolution *MTB* pair which is a down-sampled version of original *MTB*, and compute the minimum difference offset between them within a range of  $-1$  to  $1$  pixel in each dimension. At the next resolution level, we multiply this offset by  $2$  (corresponding to the change in resolution) and compute the minimum



**Figure 3.** XOR difference bitmap of two exposures

difference offset within  $-1$  to  $1$  pixel range of this previous offset. The iterative process continues to the highest (original) resolution  $MTB$  to get overall offset result. After we find the offset, we can apply the translation to the original testing image. What we should pay attention to is how to crop the aligned image. Because we zero-padded the images before comparison, we can simply find out which row or column of the images are all-zero, and then take the common part of all images.

The results show that  $MTB$  method can effectively find the shift to make the testing images aligned so that the blurring effect can be reduced. In our cases, we assumed that all the camera motions belong to translation, which is valid since there are little rotation in most situations. In theory, we can generalize  $MTB$  method to all possible transformations of image which can be expressed as affine transformation.

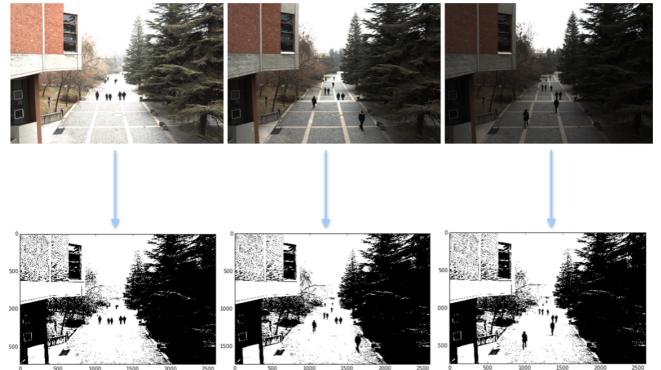
### 3. Weight Map Based Exposure Fusion

After global registration, our goal is to avoid the potential ghosting in the merged image. The deghosting process basically consists of two parts, ghost detection and image integration.

#### 3.1 Ghost Detection

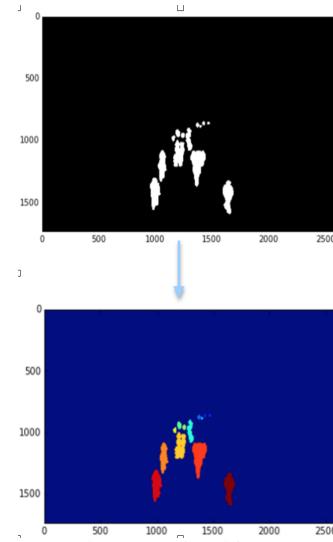
The ghost detection algorithm we apply is called *median threshold bitmap algorithm*[7]. As mentioned above, MTB is stable with respect to exposures. So the first step during ghost detection is to apply median threshold filter to each image  $L_k$ , creating different Bitmaps  $B_k$ , which are shown in **Figure 4**. In a static scene, the pixels' value across different bitmaps are expected to be the same. And if a pixel have different values in different Bitmaps, it would be consider as movement. Based on this assumption, we produce ghost map  $M^*$  based on bitmaps. Noise or unsatisfied detection might occur in  $M^*$ , so we use morphological image processing to avoid those effects. If there is too much noise, image erosion with larger size kernel would be applied. If the ghosting area is too small to be recognizable, we apply image dilation. In most cases, both operators are used.

After morphological image processing is performed,  $M^*$  turns out to be a “cluster map”. To avoid images with different exposures contributing to the same cluster, we label each



**Figure 4.** Bitmaps obtained by applying median threshold to images with different exposures

cluster with a unique index. This way we guarantee that the motion area in the final image would only come from image of a certain exposure, and the motion area would not be splitted into different parts with different exposures. This yields the ghost map  $L_M$  (**Figure 5**), and we will use this map to help us integrate the images.



**Figure 5.** Ghost map by summing bitmaps and the ghost map with label

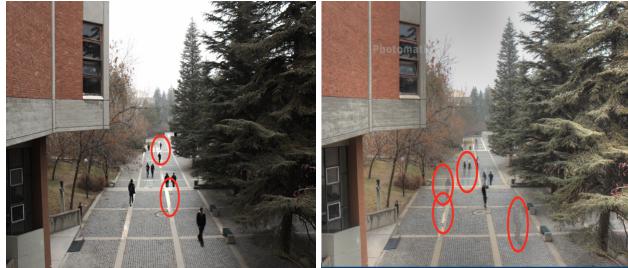
#### 3.2 Weight Map Based Integration

The method we use to integrate images is called *weight map based exposure fusion*[8]. For each image  $L_k$ , we produce a weight map  $W_k$  based on three parameters: contrast, saturation and exposedness. The area with high values in weight map indicates that area in the corresponding image  $L_k$  will contribute more to the final image. And before integration, the weight map is normalized so that they sum up to one in each pixel. Then we incorporate the motion map  $L_M$  we got

in ghost detection with the weight map. In no motion area:

$$R_{ij,k} = \sum_{k=1}^N W_{ij,k} L_{ij,k} \quad (1)$$

The subscript  $i, j, k$  refers to pixel  $(i, j)$  in the  $k^{th}$  image,  $N$  represents how many low exposure images we get overall. But in the motion area, we average all the well-exposed weight parameters for each exposure  $L_k$  in the motion area, and set the weight of the highest average as one. Thus, the motion area in the final image would only come from one image, and the ghosting is avoided. The result is shown in **Figure 6**



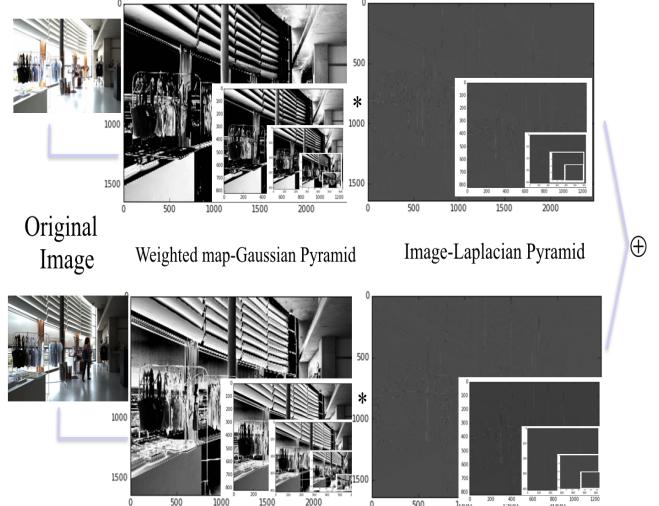
**Figure 6.** Left: *HDR* image using proposed algorithm. Right: *HDR* image using Photomatix® without deghosting

### 3.3 Multi-Resolution Image Blending

The algorithm we apply above can basically produce *HDR* image. But imperfections still exist, intensity discontinuity continuously shows up between the background and the ghosting area. The reason is that the absolute intensity of the original images is quite different. Even after balancing with weight map, intensity discontinuities still exist between neighbouring pixels. So we improve the integration process by using *Muti-Resolution Image Blending* algorithm[9]. One of the great properties of this method is that it blends images based on their features instead of intensity. It first produces a Laplacian pyramid  $\mathbf{L}\{L_k\}$  for original images, and a Gaussian pyramid  $\mathbf{G}\{W_k\}$  for the weight maps. And instead of implementing weight maps on original image  $L_k$ , the weight maps are used against the Laplacian pyramid. Let the  $l^{th}$  level in a pyramid be defined as  $\mathbf{L}\{L_k\}^l$ . And the images are blended as follow:

$$\mathbf{L}\{R\}_{ij}^l = \sum_{k=1}^N \mathbf{G}\{W\}_{ij,k}^l \mathbf{L}\{L\}_{ij,k}^l \quad (2)$$

Multiplying the each Laplacian pyramid with normalized weight map pyramid level by level would produce weighted Laplacian pyramid. And then adding them together on each level produces a fused Laplacian pyramid. This process is shown in (**Figure 7**). And we repeat this process for R, G, B channel separately, and eventually get three fused pyramids for R, G, B channel respectively. At last, we rebuild the final image by collapsing the fused pyramids.



**Figure 7.** The process of multiplying gaussian pyramid with laplacian pyramid

The results are shown in **Figure 8**, before we go further we need to classify the testing images first. The testing images we use consist of two different ghosting phenomenon[3]: *scene with “deformable” body motion (flag)*<sup>1</sup> and *scene with spatial displacement* (walking people, car, cafe). The word “deformable” indicates that the moving object may overlap with itself in different exposures, which increases the difficulty for us to detect the exact motion area of the object. And for the ghosting of displacement, we further develop it into *object with strong lit background* and *object with light lit background*.



**Figure 8.** Left: *HDR* using weight map based exposure fusion. Right: *HDR* using multiresolution fusion

We can infer from the testing result that both image integration algorithms don't work very well on ghosting with deformable body motion (flag). That is due to the disadvantage of the *bitmap based ghosting detection* algorithm. The overlapped area between the moving objects in different images would not be consider “motion area” because the pixels’ value won’t change in those area. And in the second case,

<sup>1</sup> Sample images including flag, walking people, car, cafe and shop are given in this [link](#)

the *weight map based exposure fusion* algorithm shows acceptable results when intensity difference between object and background can be neglected (cafe, car), but fails in other cases (shop, walking people). While *multi-resolution image blending* algorithm works well for both cases by blurring the edges between motion area and background. But no matter which algorithm we use, the final image would appear a litter bit darker than the middle exposure image, and it needs us to refine the weight map generation process to raise the contribution of over-exposure image.

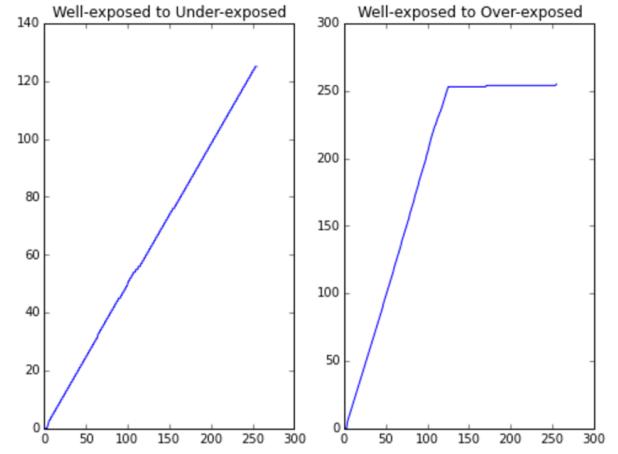
#### 4. Histogram Mapping Based Ghost Removal

In this section, we introduce another algorithm based on histogram mapping to realize deghosting followed by the results of global registration. Inspired by the constant *MTB* in **section 2**, which is expected due to quasi-linear relation between pixel intensity and exposure time, we can further generalize the intensity mapping relation between two exposures. In [5], Grosch derived the camera response function that describes the relation between irradiance and pixel intensity. Then, the irradiance of one point in the scene can be calculated from a pixel and the corresponding pixel response in another exposure can be obtained given exposure time. Thus, the inconsistent pixels can be detected as motion points. However, deriving the camera response function is not handy due to the process of curve fitting [10]. Instead, we choose to take the advantage of intensity mapping to predict the pixel intensity in another exposure. From [10], we learn that intensity mapping function  $\tau(\cdot)$  from one image to another can be estimated by their cumulative histograms. The estimation process is similar to histogram specification. Based on  $\tau(\cdot)$  and its inverse function  $\tau^{-1}(\cdot)$  (**Figure 9**), we can predict the pixel intensity of testing image as it should be given the corresponding pixel intensity of reference image. Here, we introduce a new similarity index  $S(I_1, I_2)$  to quantitatively judge the inconsistency [11]:

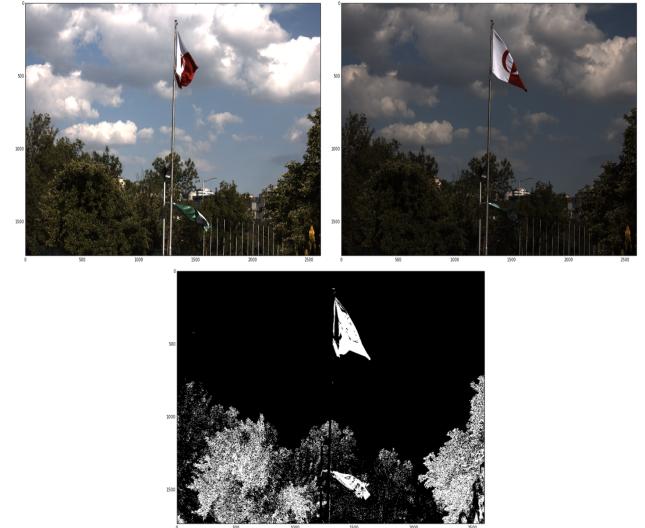
$$S(I_1, I_2) = \begin{cases} \frac{2\tau_{12}(I_1)I_2+1}{\tau_{12}^2(I_1)+I_2^2+1} & \text{if } I_1 \text{ is more reliable} \\ \frac{2\tau_{21}(I_2)I_1+1}{\tau_{21}^2(I_2)+I_1^2+1} & \text{otherwise} \end{cases} \quad (3)$$

where  $I_1, I_2$  are the pixel intensity of reference image and testing image,  $\tau_{12}, \tau_{21}$  are the intensity mapping function from reference image to testing image and vice versa. The statement that  $I_1$  is more reliable means that the pixel in reference image is not under-exposed or over-exposed. In this way, a lower similarity implies that motion happens at that point and an ideal mapping will make the similarity index equal to 1. We can threshold the similarity map to decide which pixel to keep or not. For the color image with three channels, we empirically set the threshold sum of three similarity indexes as 2.9 to detect the motion. An example with moving flag and

leaves in the wind is shown in **Figure 10**. The area occupied by moving objects has been highlighted in the bitmap.



**Figure 9.** Estimated intensity mapping function between two exposures



**Figure 10.** Two exposures and their similarity bitmap

Once we detect the motion, we can choose to drop it or replace it with keeping the right pixels. But before that, we should do some morphological operations to reduce the noise as well as fill the small holes in the bitmap. This step can guarantee that we manipulate the motion area as a cluster rather than isolated parts so that the final HDR image can be visually consistent. Some algorithms like what we did in **section 3** only take one exposure of motion area to contribute to the final generation. It is simple but at the sacrifice of dynamic range of moving objects. Because we have estimated the intensity mapping function, we can predict the intensity as it should be which can replace the pixel detected as motion. So we can synthesize updated testing images which have the same content like same position of people with consistent exposure. The intermediate result is shown in **Figure 11**.



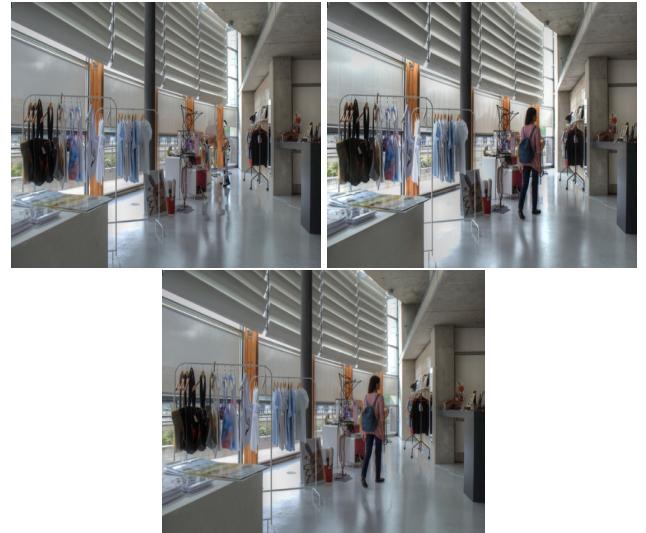
**Figure 11.** Registered images using intensity mapping method

To evaluate our algorithm, we utilize the commercial HDR software Photomatix® to realize HDR merging and display (tone-mapped version). Also, Photomatix® has its built-in alignment and deghosting which can be a reference for our evaluation. The **Figure 12** shows the final HDR result based on our proposed algorithm and two outputs of Photomatix® with automatic deghosting module or not. It can be seen that our algorithm can remove the ghosting effect. Note that we keep the over-exposed details near the windows compared with the result of Photomatix® automatic deghosting. We use the benchmark dataset offered by [3] to further test our algorithm's robustness. The results show that our algorithm can be effectively applied to 9 out of 10 image sets to remove the ghosting.<sup>2</sup> The one failure is the result of invalid estimation of intensity mapping function because the reference image is too dark such that there is no corresponding intensity in the under-exposed image to map.

## 5. Conclusion

Both global registration and local registration have been realized and tested in our project. For the global registration based on *MTB*, it is fast and robust, and its result can be helpful for the following deghosting. Then, we adopted two different algorithms to remove the ghosting. The first, weight map based exposure fusion, can reduce the artifacts and generate final *HDR* meanwhile. The other one, histogram mapping based method, will synthesize registered images of consistent content with different exposures to be merged. The final result shows that our proposed algorithm has a good balance between ghosting removal and loss of dynamic range compared with Photomatix®.

<sup>2</sup>For more testing images results based on histogram mapping, please hit [here](#)



**Figure 12.** Upper left: *HDR* image without deghosting. Upper right: *HDR* image with automatic deghosting of Photomatix®. Bottom: *HDR* image using proposed algorithm

The future works of our project will be

- How to realize the global registration for complex camera motion like rotation and tilting;
- How to optimize the exposure fusion to reduce the discontinuities between motion area and background;
- How to acquire higher accuracy of intensity mapping function estimation to improve the robustness.

Also, to realize real-time interactive image editing, the computational cost should be taken into considerations.

## References

- [1] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.
- [2] Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. *Journal of graphics tools*, 8(2):17–30, 2003.
- [3] Okan Tarhan Tursun, Ahmet Oguz Akyuz, Aykut Erdem, and Erkut Erdem. The state of the art in hdr deghosting: A survey and evaluation. In *Computer Graphics Forum*, volume 32, pages 348–362, 2015.
- [4] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaeoubi, Soheil Darabi, Dan B Goldman, and Eli Shechtman. Robust patch-based hdr reconstruction of dynamic scenes. *ACM Trans. Graph.*, 31(6):203, 2012.
- [5] Thorsten Grosch. Fast and robust high dynamic range image generation with camera and object movement. *Vision, Modeling and Visualization, RWTH Aachen*, pages 277–284, 2006.

- [6] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. *ACM Transactions on Graphics (TOG)*, 22(3):319–325, 2003.
- [7] Fabrizio Pece and Jan Kautz. Bitmap movement detection: Hdr for dynamic scenes. In *Visual Media Production (CVMP), 2010 Conference on*, pages 1–8. IEEE, 2010.
- [8] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pages 382–390. IEEE, 2007.
- [9] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532–540, 1983.
- [10] Michael D Grossberg and Shree K Nayar. Determining the camera response from images: What is knowable? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11):1455–1467, 2003.
- [11] Zhengguo Li, Susanto Rahardja, Zijian Zhu, Shoulie Xie, and Shiqian Wu. Movement detection for the synthesis of high dynamic range images. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3133–3136. IEEE, 2010.