

Ghost removal in HDRI acquisition

Mateusz Markowski*

Computer Graphics Group
West Pomeranian University of Technology, Szczecin

Abstract

This paper presents a fast, robust and completely automatic technique for creating high dynamic range images of non-static scenes. The technique is based on probability maps. These maps are calculated with comparison function from sequences of hand-held photographs. In practice, several basic problems occur when taking an image sequence. First, the camera is moving which causes images to misalign. This results in a blurry HDR image. Secondly, objects are in movement causing ghost artifacts. Finally, the dynamic range of a single image is decreased by sensor noise. Our solution is a GPU based application for compositing HDR images that can handle all aforementioned problems.

Keywords: ghost removal, dynamic scene HDRI acquisition, automatic image position alignment, GPU

1 Introduction

The radiance of real scenes around us varies dozen orders of magnitude. A typical LDR (Low Dynamic Range) camera is not able to capture full range of a scene luminance. HDR (high Dynamic Range) camera sensors with extended dynamic range exist but they are very not popular at consumers market [8, 9]. Moreover, the dynamic range of such sensors is also limited to 7-8 orders of magnitude [5]. The most popular method of the HDR image acquisition captures an HDR image based on a sequence of photographs taken with different exposure times [6]. The main drawback of this technique is the limitation to static scenes. Any movement in input photographs can cause the ghost artifacts (Figure 1). Moreover, in the case of hand-held photographs, another artifacts like object misalignments can appear due to camera shifts and rotations [13].

In the paper we proposed a new technique for acquisition of non-static scenes. Our application of this technique allows to create correct HDR image based on a simple sequence of three LDR photographs with overlapped ghost regions. Additionally, the application aligns photographs and provides image de-noising. All functionalities are fully automatic. The technique is robust and fast due to GPU-based implementation.

The paper is organized as follows. In section 2 previous

works are discussed. In section 3, the application of our HDR acquisition technique is presented in details. Section 4 shows and discusses achieved results. In the last section, the paper is concluded and possible future work is suggested.

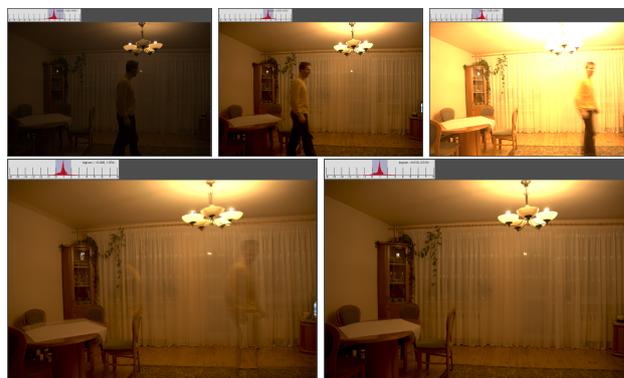


Figure 1: Problem: LDR sequence (top row), HDRI: conventional approach (bottom left), our approach (bottom right).

2 Previous work

There is a growing demand for HDR image of both static and dynamic scenes. That's why as far as hardware solutions of HDRI acquisition are not easily available, software solutions will be needed. A few approaches have been developed in order to remove ghosts artifacts during HDRI acquisition. The first technique is based on tracking non-static objects by matching their key points in a sequence of images [12]. The method fails for occluded objects or for patterns for which it is not possible to find correct matchings. Another approach replaces the whole regions, where ghost artifacts are likely to occur, with reference ones. The regions can be selected manually [7][10] or detected automatically [2][3]. Unfortunately, the technique works correctly only when the whole dynamic range of a region can be registered in a single image exposure. A different solution was presented in [4] where iterative propagation of ghost probability was used. The method requires a large number of images in LDR sequence and still background for moving objects. Moreover, it is time consuming and must be computed in many iterations.

*mmarkowski@wi.ps.pl

Image registration is another problem during acquisition of HDR image. Misalignments between photographs in a sequence can appear due to camera movement (in the case of hand-held photographs) or not careful usage of tripod. There are two basic techniques of image registration: matching key points and checking pixels difference. In the first case the same drawbacks as during ghost removal occurs (matching key-points problem). The solutions based on pixels difference generally give better results. In some software solution for alignment only horizontal and vertical shifts without rotation compensation are considered due to complexity of computations [13].

In the paper we propose modified pixel-based approach for removing ghosts and misalignments. Deghosting is based on the ghost maps. The ghost maps are calculated using probability of belonging pixels to background. They depict regions where ghost artifacts are likely to occur or under- and overexposed pixels. The technique is fast due to GPU based implementation of de-ghosting and alignment modules.

3 Algorithm

We developed the GPU based application for HDRI acquisition based on taking a sequence of hand-held photographs. The algorithm used in the application has four successive stages: image de-noising, position alignment, ghost removal and HDRI composition. Three of them can be used optionally (Figure 2), however, it is easier to align denoised images or remove ghost from aligned photographs so each successive stage works better if previous stages were included into the acquisition pipeline.

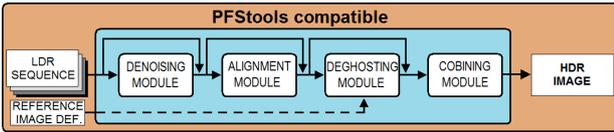


Figure 2: Image acquisition pipeline.

3.1 Image denoising module

The image denoising module is based on the wavelet coefficient thresholding method presented in [11]. We optimized the existing implementation of the algorithm. The new implementation based on GPU was developed.

3.2 Image alignment module

The images alignment module compensates camera shifts or rotations. In the first step, an image with middle exposure is chosen as a reference one (Figure 3). Other images in a sequence are aligned to this image. Three alignment parameters are iteratively optimized: dx , dy and α .

The transformations are calculated with sub-pixel precision (floating values are used) (Figure 4).



Figure 3: Reference image – middle exposure.

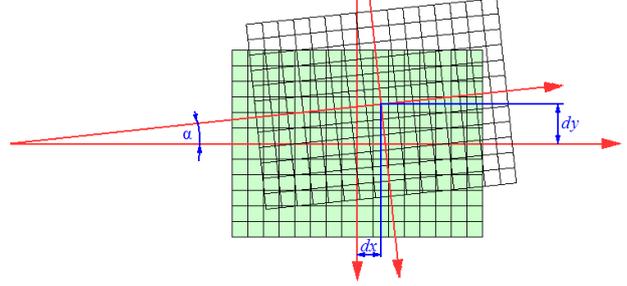


Figure 4: Image alignment parameters.

Sampling points of images are determined according to equation:

$$\begin{aligned} x' &= x \cos(\alpha) - y \sin(\alpha) + dx, \\ y' &= x \sin(\alpha) + y \cos(\alpha) + dy, \end{aligned} \quad (1)$$

where: x', y' - non-reference image pixel coordinates,
 x, y - reference image pixel coordinates,
 α - angle of non-reference image rotation,
 dx, dy - horizontal and vertical shifts.

Bilinear interpolation is used to calculate sampled value. If sampling point is outside image boundary, black value is returned. Each sampled value in non-reference and reference images is normalized by exposure:

$$E_i = \frac{S \pi d^2}{f^2} t_i, \quad (2)$$

where: d - aperture,
 S - ISO speed,
 f - focal length,
 t_i - exposure time of image i .

$$P_i(x, y)_c = \frac{S_i(x, y)_c}{E_i}, \quad (3)$$

where: $P_i(x, y)_c$ - exposure normalized component c of image i at xy position, $c \in \{R, G, B\}$,
 $S_i(x, y)_c$ - component c of input LDR image i at xy ,
 E_i - exposure value of image i .

Then normalized values are used in objective function K to minimize misalignments. Output tells us if compared values match each other in all RGB channels or not. Comparison function should have a margin for small registration errors. Output value should also be generally constant when values difference is equal or greater than some

threshold. That's why standard cross-correlation shouldn't be used. We propose bidirectional s-shape comparison function. Function parameters were tuned by trial and error.

$$K(P_{ref}(x,y), P_i(x',y')) = \sum_{c=R,G,B} \frac{1}{1 + \exp\left(-\left(\max\left(\frac{P_{ref}(x,y)_c}{P_i(x',y')_c}, \frac{P_i(x',y')_c}{P_{ref}(x,y)_c}\right) - 1.25\right) \cdot 25\right)}, \quad (4)$$

where: K - alignment comparison function,
 $P_{ref}(x,y)_c$ - component c of reference image at xy ,
 $P_i(x',y')_c$ - component c of non-ref. image at xy ,
 $c \in \{R, G, B\}$.

If any of pixel values is under- or over-exposed the value 1 is returned (instead of calculation of K). In GPU implementation, the images are represented by textures. Dimensions of the textures are reduced to power of two (see Figure 5). For fast summation, they are downsampled by factor of two until one pixel dimension is reached. The smaller values in that texture the better alignment. In each iteration, texture values are compared with values computed in the previous iteration until the best alignment for each images is reached. Up to four images can be aligned in parallel in GPU based implementation due to four channels RGBA architecture.

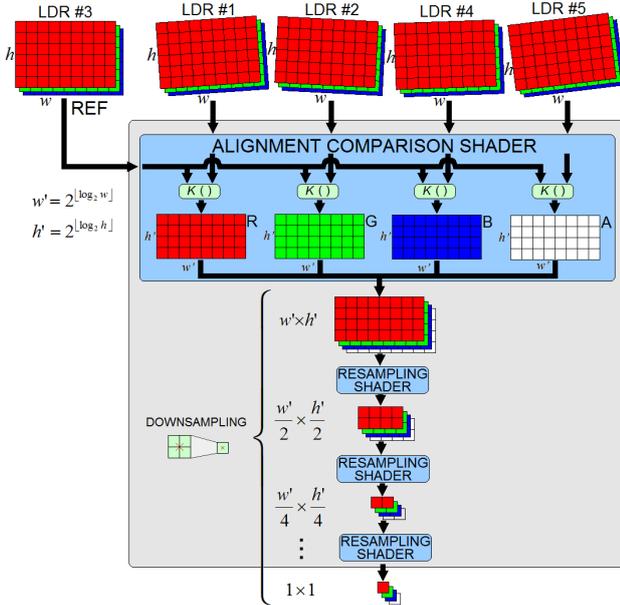


Figure 5: Alignment module hardware implementation.

3.3 Ghost removal module

In de-ghosting stage the ghost map is generated for each LDR image. It estimates a probability that a pixel belongs to a still background or to moving object (ghost). The ghost maps are computed in three stages: initial map generation, normalization and final smoothing.

The initial ghost map is computed based on equation:

$$G_i(x,y) = \sum_{j=1, j \neq i}^k F(P_i(x,y), P_j(x,y)), \quad (5)$$

where: G_i - ghost map of i -th image,
 F - deghosting comparison function,
 P_i, P_j - i -th and j -th normalized LDR images,
 k - number of images.

The de-ghosting comparison function F is based on sigmoid function (equation 6). At development stage function was designed and tuned with registration error distribution graphs. Even for static scenes captured values aren't linearly dependent due to registration errors. That's why we have used distribution graphs to determine acceptable and unacceptable limits in color difference.

$$F(P_i(x,y), P_j(x,y)) = \sum_{c=R,G,B} \frac{1}{1 + \exp\left(-\left(\max\left(\frac{P_i(x,y)_c}{P_j(x,y)_c}, \frac{P_j(x,y)_c}{P_i(x,y)_c}\right) - 1.35\right) \cdot 20\right)}, \quad (6)$$

where: F - deghosting comparison function,
 $P_i(x,y)_c$ - component c of normalized image i at xy ,
 $P_j(x,y)_c$ - component c of normalized image j at xy .

The initial ghost maps must be normalized. The normalization is based on equation:

$$G_i^n(x,y) = \left(\frac{G_i(x,y) - g_{min}}{g_{max} - g_{min}}\right)^3. \quad (7)$$

where: g_{min}, g_{max} - minimum and maximum value in all ghost maps ($G_i(x,y)$),
 $G_i(x,y)$ - pixel of a ghost map i at xy ,
 $G_i^n(x,y)$ - pixel of normalized ghost map i at xy ,
 k - number of images.

Additionally, the zero value is assigned to over-exposed pixels.

A special mode of normalization, that gives better results, was designed for 3-images sequence (such sequence is often captured using automatic exposure bracketing mode (AEB) in DSLR cameras). It takes into consideration various special conditions: under- and over-exposed pixels, color difference greater than a threshold. For such special pixels, the modified normalization equation is used:

$$G_i^n(x,y) = \left(\frac{G_i(x,y) - g_{min}}{g_{max} - g_{min}}\right)^4. \quad (8)$$

Normalized ghost maps are prone to errors. They look noisy due to independent computation of each pixel in ghost maps (see Figure 6). In the next stage the ghost map smoothing is proceeded. It integrates map values with neighbor pixels.

The smoothing is based on dilatation, erosion and convolution. Firstly the (3×3) dilatation is applied (equation



Figure 6: Example LDR images sequence (left column: $\frac{1}{640}s$, $\frac{1}{160}s$, $\frac{1}{40}s$) and their normalized ghost maps (right column).

9). It removes one or two pixels regions recognized as ghost from the ghost maps. After dilation, the (15×15) erosion is applied (equation 10). It fills holes in ghost regions recognized as valid pixels. These regions are additionally extended during dilatation. Finally ghost maps are smoothed based on convolution with (5×5) window kernel (equation 11). The example results after each step are presented in (Figure 7)

$$G_i^d(x, y) = \max_{dx=-1}^1 \left(\max_{dy=-1}^1 (G_i^n(x + dx, y + dy)) \right), \quad (9)$$

$$G_i^e(x, y) = \min_{dx=-7}^7 \left(\min_{dy=-7}^7 (G_i^d(x + dx, y + dy)) \right), \quad (10)$$

$$G_i^c(x, y) = \sum_{dx=-2}^2 \frac{10 - |dx|}{50} \sum_{dy=-2}^2 \frac{10 - |dy|}{50} G_i^e(x + dx, y + dy), \quad (11)$$

where: $G_i(x, y)$ - pixel of ghost map i at xy position,
 G_i^n - ghostmap after normalization,
 G_i^d - ghostmap after dilatation,
 G_i^e - ghostmap after erosion,
 G_i^c - ghostmap after convolution,
 dx, dy - horizontal and vertical offsets in windowed kernel.

The ghost maps generation module was implemented in GPU (see Figure 8 for details). It creates maps in a single rendering pass. For more than four images in an image sequence, multi-texture hardware extension (MRT) is used. To improve performance, horizontal and vertical kernels are applied separately in each operation.

3.4 HDRI composition module

The final HDRI composition is based on the method presented in [1]. The main improvement is integration of the ghost map with the final equation:

$$H(x, y) = \frac{\sum_{i=1}^k \frac{S_i(x, y)_c}{E_i} W_i(x, y) G_i^c(x, y)}{\sum_{i=1}^k W_i(x, y) G_i^c(x, y)}, \quad (12)$$

where: $H(x, y)$ - pixel of output HDR image at xy position,
 $S_i(x, y)$ - pixel of input LDR image i at xy ,
 E_i - exposure value of image i .
 $G_i^c(x, y)$ - grayscale pixel of ghost map i at xy ,
 $W_i(x, y)$ - pixel weight of image i at xy .

Moreover, the weight function is modified:

$$W_i(x, y) = \sum_{c=r,g,b} \min \left(\frac{1}{1 + \exp(5 - 100S_i(x, y)_c)}, \frac{1}{1 + \exp(30S_i(x, y)_c - 24)} \right), \quad (13)$$

where: $W_i(x, y)$ - pixel weight of image i at xy position,
 $S_i(x, y)_c$ - pixel c component of LDR image i at xy .

It has been tuned in trial and error process in witch ghost maps task was taken into consideration. A chart of the weight function for a single color component is shown in (Figure 9).

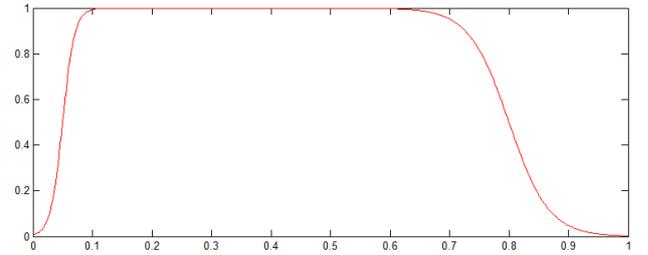


Figure 9: Weight function graph for single color component.

The HDR image color is computed based on equation 12 in a single rendering pass. The chart of shader used in this task is presented in Figure 10. The shader has scalable number of input textures.

4 Results

A few example image sequences were used to test quality and performance of the HDRI acquisition application. Both hand-held and tripod sequences with varying number of images and exposure range were assessed. The best results were achieved for exposure difference less or equal



Figure 7: Ghostmap $i=1$ results after: dilatation – G_i^d (left), erosion – G_i^e (middle), convolution – G_i^c (right).

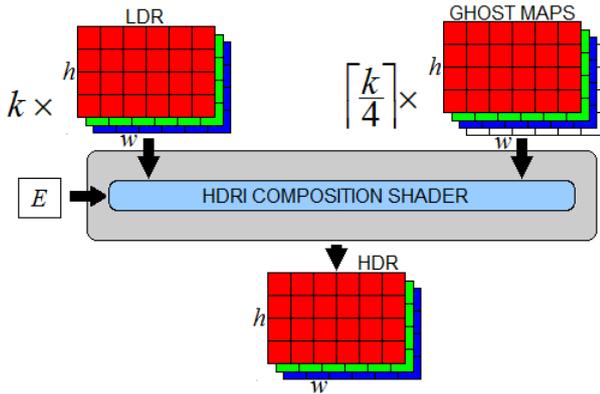


Figure 10: HDR composition shader.

to two F-stop. It is compatible with exposure bracketing functionality in typical DSLR cameras which allows for such exposure change. In Figure 11 we demonstrate results of image alignment. Even for exaggerated camera movement, the application correctly aligns all photographs. Another example (see Figure 12) presents ghosts removal results. In this sequence, moving objects (a woman playing with dog) causes ghosts artifacts. Our ghost removal module detects moving objects and removes ghost from a final HDR image. We compared our technique with existing application for acquisition of a dynamic HDRI scenes: Dynamic Photo HDR, Qtpfsgui and Photomatix. The results of comparison is shown in Figure 13. Our application seems to produce the best images with correctly removed ghosts. Our application requires 11 seconds to align, de-ghost and create a final HDR image based on a sequence of three LDR images of resolution 3039x2014 pixels (GPU textures upload/download 4.55 s, alignment 6.11 s, ghost detection and HDR composition 0.47 s). The high performance of algorithms computation was achieved due to careful GPU implementation. More examples is presented on project's website (<http://mmarkowski.strony.wi.ps.pl/grhdri.html>).

5 Conclusions and Future work

In this paper a fully automated tool for HDRI acquisition was presented. It allows to create HDRI images of static and dynamic scenes from hand-held photographs. In future work we plan to improve the ghost map generation module. We noticed that some ghost removal errors can occur for images with many high dynamic range ghost regions like reflections on a waving water. Performance of alignment module could be also improved, because currently it is 12-times slower module than de-ghosting and final composition modules.

References

- [1] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, pages 369–378, 1997.
- [2] Thorsten Grosch. Fast and robust high dynamic range image generation with camera and object movement. In *Vision, Modeling and Visualization, RWTH Aachen*, pages 277–284, 2006.
- [3] HDRsoft. *Photomatix Pro*, 2003. <http://www.hdrsoft.com>.
- [4] Erum Arif Khan, Ahmet Oğuz Akyüz, and Erik Reinhard. Ghost removal in high dynamic range images. In *IEEE International Conference on Image Processing*, 2006.
- [5] Grzegorz Krawczyk. *High Dynamic Range Capture and Tone Mapping*. MPI Informatik, Saarbrücken, Germany, June 2006.
- [6] Steve Mann. Compositing multiple pictures of the same scene. In *IS&T's 46th Annual Conference*, Cambridge, Massachusetts, May 1993.
- [7] Mediachance. *Dynamic Photo HDR*, 2008. <http://www.mediachance.com>.
- [8] Srinivasa G. Narasimhan and Shree K. Nayar. Enhancing resolution along multiple imaging dimensions using assorted pixels. *IEEE TRANSACTIONS*

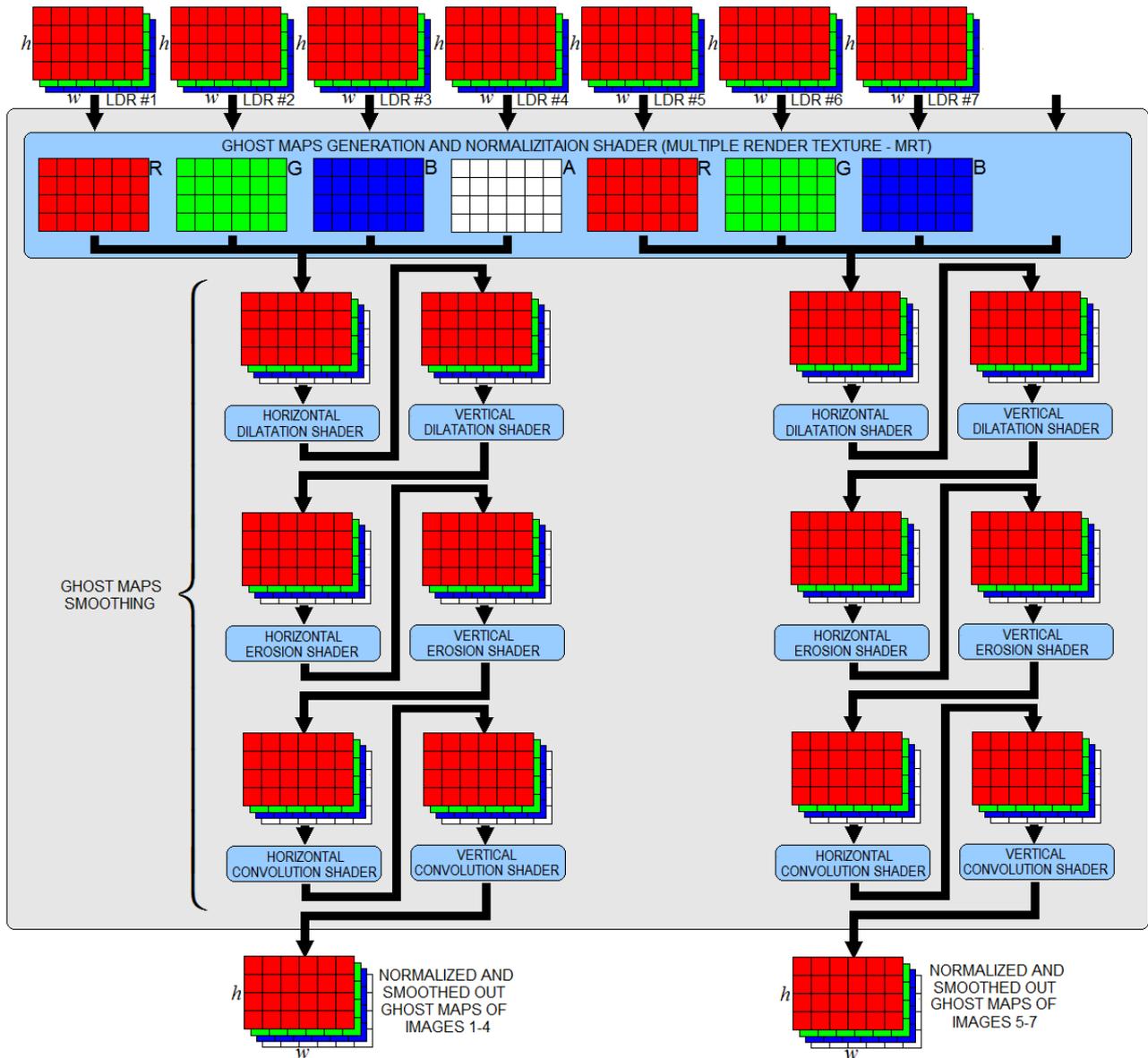


Figure 8: Ghost maps generation module.

ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, April 2005.

- [9] OMRON Automotive Electronics GmbH. *HDRCTechnology*, 2005.
- [10] Giuseppe Rota. *QtPfsGui - HDR Imaging Workflow Application*, 2007. <http://qtpfsgui.sourceforge.net>.
- [11] Ivan Selesnick and Crystal Wagner. *Double-Density Wavelet Software*. Polytechnic University's Brooklyn, 2004.
- [12] Anna Tomaszewska and Radosław Mantiuk. Image registration for multi-exposure high dynamic range image acquisition. In *WSCG*. Politechnika Szczecińska, 2007.
- [13] Gregory Ward Larson. *Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures*. Exponent - Failure Analysis Assoc., Menlo Park, CA, 2003.



Figure 11: Tonemapped HDRI without aligning(left) with aligning(right).



Figure 12: Deghosting example: LDR sequence (top row), conventional acquisition (bottom left), our approach (bottom right).



Figure 13: Deghosting example: LDR sequence (top row), conventional acquisition (bottom left), our approach acquisition (bottom right).