# Bitmap Movement Detection: HDR for Dynamic Scenes

Fabrizio Pece*, Jan Kautz*

* Dept. of Computer Science
University College London
Malet Place - London WC1E 6BT, UK
email: f.pece@cs.ucl.ac.uk, j.kautz@cs.ucl.ac.uk
www: www.cs.ucl.ac.uk/staff/F.Pece/, www.cs.ucl.ac.uk/staff/J.Kautz/

## Abstract

Exposure Fusion and other HDR techniques generate well-exposed images from a bracketed image sequence while reproducing a large dynamic range that far exceeds the dynamic range of a single exposure. Common to all these techniques is the problem that the smallest movements in the captured images generate artefacts (ghosting) that dramatically affect the quality of the final images. This limits the use of HDR and Exposure Fusion techniques because common scenes of interest are usually dynamic. We present a method that adapts Exposure Fusion, as well as standard HDR techniques, to allow for dynamic scene without introducing artefacts. Our method detects clusters of moving pixels within a bracketed exposure sequence with simple binary operations. We show that the proposed technique is able to deal with a large amount of movement in the scene and different movement configurations. The result is a ghost-free and highly detailed exposure fused image at a low computational cost.

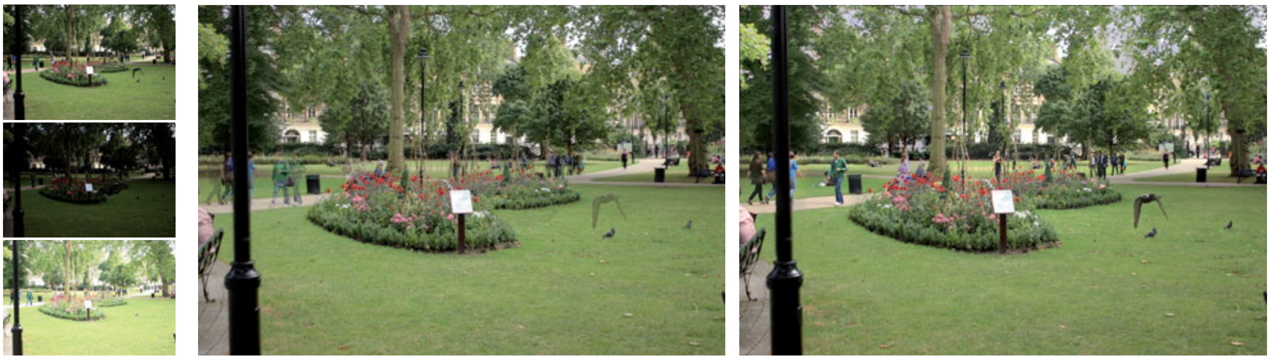
**Keywords:** HDR, Exposure Fusion, Motion Detection, Time Varying Photography, Motion Correction

*First presented at the European Conference on Visual Media Production 2010, extended and revised for JVRB*

## 1 Introduction

The real world spans a dynamic range that is larger than the limited one spanned by modern digital cameras. This poses a major problem when reproducing digital images: not all the details in a scene can be represented with conventional Low Dynamic Range (LDR) images. These problems typically manifest themselves in the presence of both overly dark and bright areas due to under- or over-exposure. High Dynamic Range (HDR) photography solves these problems by combining differently exposed pictures in order to enlarge the dynamic range captured in an image [RWPD05, DM97]. In a similar fashion, Exposure Fusion [MKVR07] solves these problems by directly fusing a set of LDR images into a single LDR exposure, dramatically simplifying the image generation process. However, for these techniques it is essential that the scene is completely static in order to obtain artefact-free results. In fact, any small change between exposures produces a particular kind of image artefact called ghosting. This limits the use of both HDR and Exposure Fusion imagery, as many common scenes contain dynamic elements.

Our goal is to adapt HDR techniques to dynamic scenes such that ghosting artefacts are detected and corrected, while maintaining Exposure Fusion's computational efficiency. To this end, we propose the Bitmap Movement Detection (BMD) algorithm. It detects clusters of moving pixels, which then guides the Exposure Fusion image generation. The best-exposed exposure is used to recover each area affected by movement. Hence, our technique produces fused images that keep only the best exposed part of the scene, see Figure 1. We show that the proposed method performs well even when the scene is affected by large



(a) Exposure stack. (b) Standard exposure fusion. (c) Our result.

Figure 1: An example of a dynamic scene. With standard techniques (Exposure Fusion), ghosting will occur. We propose a simple method to determine dynamic regions that allows us to prevent artefacts.

and substantial changes. Besides qualitative analysis, we also present a performance analysis, which shows that BMD can deal efficiently with large images. BMD is a simple, yet effective technique. The core of the algorithm relies on simple binary operations, and therefore its computation time is very fast. However, its speed does not sacrifice quality: our results are identical or superior to the ones obtained with other deghosting algorithms. For these reasons we believe that BMD and Exposure Fusion can be directly implemented on camera hardware to directly capture and generate fused images of dynamic scenes.

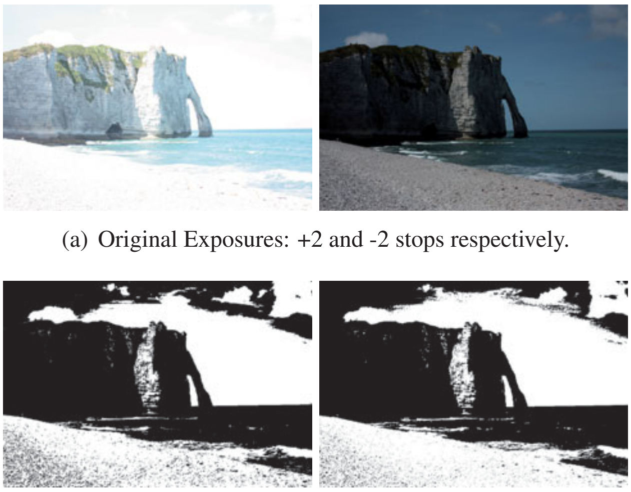
## 2 Related Work

### 2.1 Motion detection

Different approaches have been suggested to detect movement clusters in the LDR images and a large number of these take the illumination variance at each pixel into account. Unfortunately, since the exposures used in the sequence are taken with different exposure configurations, these methods are not directly applicable for the HDR or Exposure fusion case. Specific techniques for HDR images have been proposed as well and can be broadly divided into three groups: algorithms that use a single exposure to correct each affected area, ones that use more than one exposure per affected area, and techniques that prevent artefacts by directly changing the HDR weighting scheme.

Regarding the first group, Ward et al. [RWPD05] proposed a method to correct ghosting artefacts based on the variance of the weighted pixel-intensities; due to its simple implementation, this technique has been largely used in the standard HDR image generation framework as well as in Photosphere [Any] . Unfortunately, this method can easily fail in zones where the dynamic range is big or the motion is not wide, but it does work when the ghosts are easily segmentable. Jacobs et al. [JLW08] address the de-ghosting problem by using movement detection algorithm based on local pixel entropy. Entropy is used because this measure is not affected by intensity values and does not require camera calibration, but unfortunately it can easily fail in regions where the dynamic range is quite big.

The second groups of algorithms adopts an approach that takes into account a different number of exposures when recovering an affected zone. Gallo et al. [GGC$^+$09] propose a technique that tries to determine the correct number of exposures to use in affected areas for the HDR computation by evaluating, for each patch of the scene, the ghosting value, a measure of deviation of a certain exposure in a patch from the model predicted from another patch. The algorithm then builds the HDR image using different number of exposures on each defined patch, obtaining in this way ghost-free and consistent images. For the third group of algorithms, Khan et al. [KAR06] propose a technique that does not need object detection and movement estimation as it changes directly and iteratively the HDR weights to minimise the number of visible artefacts. This is done by evaluating the pixel membership probability to a non-parametric model used for the static part of the scenes. The main idea of the algorithm is that pixels that are part of the background, i.e. the static part of the scene, are more commonly present in an image than the ones that do not belong to it. This approach produces very good results, but is prohibitively expensive to compute. Motion detection based on optical flow has been proposed



(a) Original Exposures: +2 and -2 stops respectively.

(b) Bitmaps generated by MTB.

Figure 2: Bitmap similarity using MTB. MTB for two different exposures are shown. Note their similarity.

by Kang et al. [KUWS03]. They introduce a technique that uses optical flow to register pixels in adjacent video frames so that the images can be correctly combined. Unfortunately this method is very dependent on the quality of motion estimation, and thus it can easily fail. Also Bogoni uses motion estimation to tackle ghosts [Bog00]. After global registration, the author uses optical flow to perform per-pixel registration, allowing for locally correct exposure blending. Mann et al. [MMF02] register differently exposed frames through homographies, which allows them to estimate the camera response function and thus to produce an HDR image from a panning video. Mitsunaga and Nayar [NM00] introduce a technique that reduces the ghost artefacts by employing spatially varying pixel exposures.

### 2.2 Median Threshold Bitmap

The Median Threshold Bitmap (MTB) algorithm, introduced by Ward [War03] for the purpose of image alignment, is a technique that helps the comparison of images that are taken under different exposure settings by effectively removing most of the illumination differences between images. The algorithm computes a binary bitmap image by applying a threshold to the image based on its median pixels value ($mpv$). This bitmap, containing a partitioning into pixels brighter and darker than the $mpv$, has been shown to reveal image features while removing intensity differences between different exposures [War03]. Figure 2 shows two example bitmaps obtained with the MTB technique.

### 2.3 Exposure Fusion

Exposure Fusion [MKVR07] is a technique for directly fusing a bracketed exposure sequence of LDR images, which can be used as an alternative to the standard HDR image generation procedure. This technique is computationally efficient and does not require any tone mapping operator to compress the dynamic range, as the resulting image can be directly displayed on any common device. The technique does not require the camera's response curve, and instead relies on three simple per-pixel quality measures, contrast, saturation, and well-exposedness. A weighted average of these three measures is computed for each pixel, yielding a per-pixel weight map $W$ for each exposure in the sequence (weight maps are normalised to sum to one at each pixel). Conceptually, the exposures are then blended together using the per-pixel weights from the weight map. However, direct per-pixel blending produces artefacts, such as seams. The authors therefore use multi-scale blending to effectively prevent these.

## 3 Motion Detection

With the help of the MTB image descriptor, we propose a method to detect and isolate clusters of moving pixels within an exposure sequence. Figure 3 illustrates an overview of the proposed technique, which we call Bitmap Movement Detection (BMD). For each image in our exposure stack, we apply the MTB algorithm, yielding a stack of bitmaps $B_i$. In a static scene, we expect each pixel to preserve its bit value across all $B_i$. If the value changes in a pixel, we know that there was movement underneath it. So in order to detect movement pixels, we simply sum up all bitmaps $B_i$ yielding $M^*$. Any pixel in $M^*$ that is neither 0 nor $N$ (assuming $N$ exposures) is classified as a movement. $M^*$ may contain a certain amount of noise that could lead to incorrect movement detection (see Figure 4, left image). Hence, we refine $M^*$ using a sequence of morphological dilation and erosion in order to generate the final motion map $M$. The motion map $M$ for the sequence shown in Figure 1(a) is reported in Figure 4 (right image). Eroding and dilating $M^*$ are two essential steps that ensure that noise is removed from the map (i.e. erosion) while each correctly detected region is enlarged to include the entire motion area (i.e.

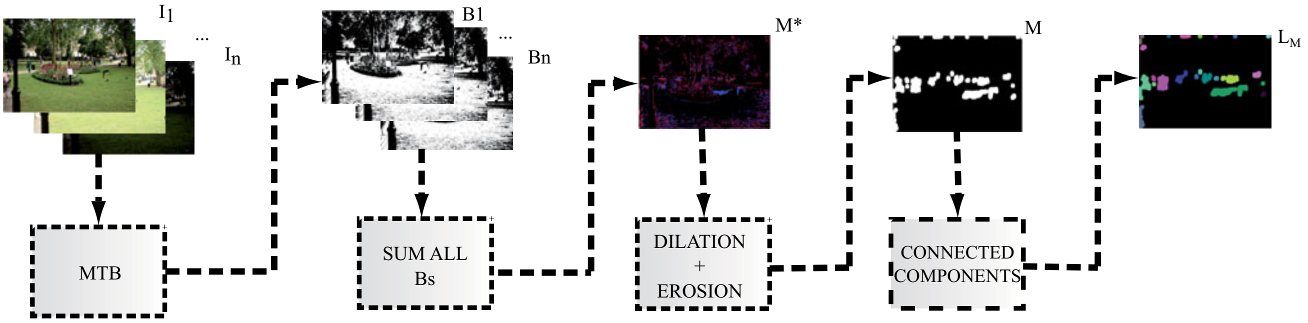


Figure 3: BMD algorithm overview.

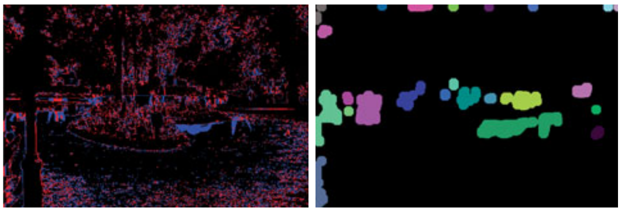


Figure 4: Motion map generated from Figure 1(a). Left image shows $M^*$; please note that non-black pixels are the ones marked for motion detection refinement. Right image shows $M$ after the application of the morphological operations; each cluster is coded with a different index, which in the figure is represented by a colour.

dilation). Thus, to correctly refine $M^*$, a good balance between the dilation kernel size, $s_d$, and the erosion kernel size, $s_e$, is required. In Section 4.1 we discuss how to choose working kernel sizes.

After erosion and dilation are performed, $M$ is converted into a "cluster map" where each identified cluster has a different label, which we compute using Connected Component labelling [HS92]. This yields the labelled motion map $L_M$ with labelled cluster areas $\Omega_i$ that contain the moving pixels which cause ghosting artefacts (see colour-coded labels in Figure 4 right).

## 3.1 HDR Integration

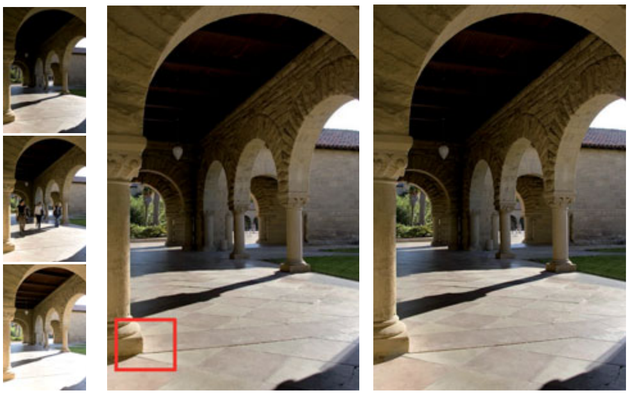
Now that we have found the regions where motion appears, we can easily integrate this into HDR imaging. We will show how to incorporate our proposed motion detection technique into Exposure Fusion, but a similar integration is possible into the HDR assembly stage.

To integrate Exposure Fusion with our motion detection technique we use the labelled motion map $L_M$ as a guide for the final blending. In fact, for each affected area $\Omega_i$ in $L_M$, we fill in the corresponding pixels in the final image with the best available exposure for that particular area (using Exposure Fusion's multiscale blending). The measure used to define the best available image is the well-exposedness quality measure already employed by Exposure Fusion. Given a cluster $\Omega_i$, we average all the well-exposed weights for each exposure $I_k$ of the stack associated to the $\Omega_i$ location. We then use the exposure $I_{k=\max_i}$ that has the maximum average to fill in $\Omega_i$. As a result, each moving cluster will contain values from a single exposure only, which has to be self consistent and ghost-free since the cluster is recovered from a single image rather than a combination. In practice, we change the weight map $W$ of Exposure Fusion in order to select the appropriate exposure for each affected area $\Omega_i$. I.e. we set the weights to 1 within $\Omega_i$ for the exposure $k = \max_i$ and to 0 for all other $k$'s. After the weights are corrected, Exposure Fusion generates the final image by collapsing the stack using its original weighted blending.

### 3.1.1 Multiple Exposures HDR Integration

The choice of using only the best exposed exposure for each affected area is motivated by the fact that the use of exactly one exposure ensures consistency of the final result with respect to the motion. However, this choice may sometimes reduce the information available, especially when more than one exposure can be used to enrich the dynamic range of a particular scene area (e.g. when a region with motion contains a large dynamic range). For this reason we developed an alternative solution that finds the subset of exposures that are considered ghost free for each motion blob in the scene.

Our proposed solution firstly computes a logical XOR between all the pair-wise combinations of MTB



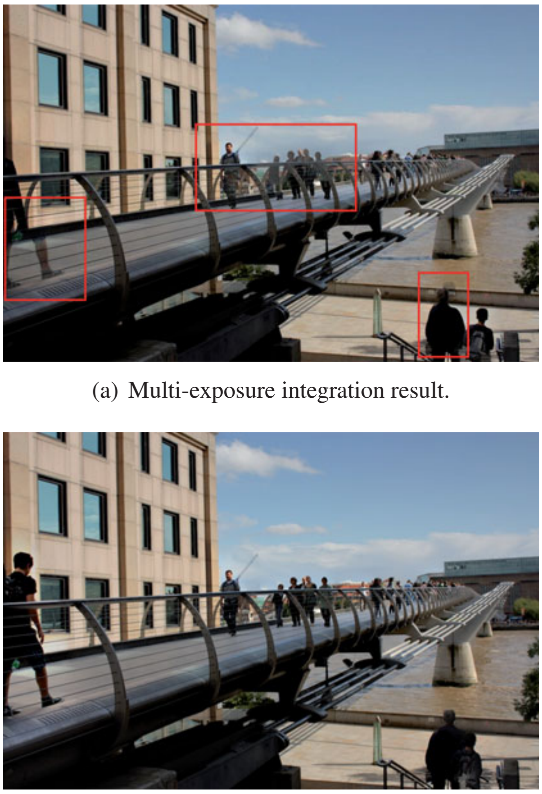
(a) Input stack. (b) Multi-exposures integration result. (c) Best exposure selection result.

Figure 5: Example of success of the multi-exposure selection method ($P = 15\%$). Improvements can be seen in the red rectangles.

s to isolate the exact exposures where movement happens (separately for each blob $b_i$ in the scene). Two exposures of a pair-wise combination are considered motion free (for a particular region) if less than $P\%$ of pixels change. This results in a modified labelled motion map $L^*_M$ where each affected area $\Omega_i$ is assigned to a motion-free sub-set of the original exposures set. Similarly to the single-exposure HDR integration, the modified motion map $L^*_M$ is used as a guide for the final HDR blending. However, rather than using a single, best exposed image for each motion blob, the blending process takes into account the exposure sub-set annotated in $L^*_M$, and blends it using Exposure Fusion's weighting scheme. In practice this means setting to zero the weights of the motion-affection exposures, and re-normalizing the weights of the remaining exposures so that they sum up to 1. Figure 5 shows result of using the multiple exposures integration (we used $P = 15\%$ for this case).

## 3.2 Discussion

Even though the multiple exposures integration can improve the information recovered for each moving area, it is difficult to select the right percentage P%. When the movement appears only in a small subset of the input stack, and thus it is totally absent from the rest of the image set, it is often possible to choose the right parameter, and the per-exposure XOR computation is able to effectively isolate the ghost free exposures, improving the final results. Commonly though, no single parameter $P$ will work for all affected regions. If $P$ is too large, no region will be classified as in motion (despite containing movement), which creates obvious artefacts. If $P$ is too small, every region will be classified as containing motion, mostly due to noise in the MTB, which consequently will not improve the results over the basic algorithm from the previous section. This problem is illustrated in Figure 6, which shows a motion configuration that let the per-exposure XOR technique fail (please refer to Figure 11(g) for the whole exposure stack employed for the image generation). Any large enough $P$ that improves some area, fails in others. Unfortunately, this behaviour is rather common in many scenes, and thus we decided to adopt only the best exposed selection in the recovery of the affected zone to prevent potential artefacts in the final results.



(a) Multi-exposure integration result.

(b) Best exposure selection result.

Figure 6: Example of failure of the multi-exposure integration technique. Red rectangles shows the artefacts in the final image.

A potential solution to this problem might be found in dynamically computing the value of $P$ for each motion blob and only for the HDR region. We note that usually not all the affected regions in a scene contain high dynamic range lighting; for those regions, using a single exposure would still lead to visually pleasant results. For all the other regions (i.e. the HDR, motion affected regions), one could dynamically apply the multi-exposure integration using individually optimised $P$ values. This could make the multi-exposure integration more robust, but it will most certainly introduce additional overhead in the final computation, since the HDR regions need to be localised before to apply the motion correction. We reserve further investigation on this for future work.

# 4 Results

We have tested our algorithm on a variety of dynamic scenes to evaluate its performance under different movement configurations. For all the results discussed in this section, we employed the single-exposure HDR integration technique described in Section 3.1. Figure 1 shows scene with a large number of small movements that blend in with the background. Our method generates a flawless image with no artefacts or inconsistent areas: this is particularly important because it shows that our method deals well with small and compact movements, a class of motion notoriously hard to detect. Moreover, the result also presents smooth transition between fused zones. The preliminary motion map $M^*$, together with the final motion map $M$, are reported in Figure 4. Results obtained on a similar movement configuration are reported in Figure 12(i) and Figure 12(l), while Figure 12(c), Figure 13(f) and Figure 13(i) show that our method can successfully correct small, ghost-affected areas with a high level of detail.

Figure 8 presents the standard fused image and our result generated from a stack of 9 exposures of a highly dynamic scene. This scene includes a large amount of motion, introduced by the moving crowd, and thus this movement configuration can be classified as “wide”. The figure shows that our method considerably improves the final result by erasing all the artefacts and selecting the appropriate replacement for the corrected clusters.

Figure 11(c), Figure 12(f) and Figure 13(c) show the result obtained from a scene with large horizontal motions, while in Figure 11(f) and 11(i) the objects are moving towards the camera. Objects moving towards the camera are particularly hard to detect because the area they span is very narrow. However our method is able to handle with this configuration, as well as with the horizontal motion, with very small errors.

| **w × h × N** | **EF** | **BMD** |
|---|---|---|
| 550 × 820 × 3 | 4.22 sec | 0.627 sec |
| 683 × 1024 × 3 | 5.97 sec | 0.980 sec |
| 1366 × 2048 × 3 | 23.08 sec | 3.03 sec |
| 550 × 820 × 6 | 7.15 sec | 1.01 sec |
| 550 × 820 × 9 | 7.15 sec | 1.42 sec |
| 683 × 1024 × 6 | 10.97 sec | 1.63 sec |
| 683 × 1024 × 9 | 10.97 sec | 1.96 sec |

Table 1: Computational times of the original Exposure Fusion (EF) and BMD on a 2.4 GHz Intel Core 2 Duo.

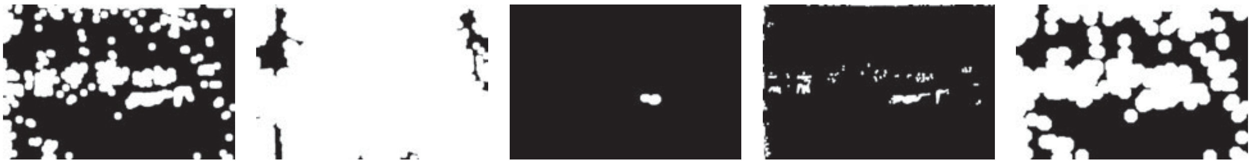
Further, we have compared our results with the techniques described in [GGC$^+$09, JLW08, RWPD05] and with the results obtained with the commercial tool *Photomatix [HDR]*. Figure 10 shows results generated from a set of three, five and four exposures respectively. The methods in [JLW08, RWPD05, HDR] did not correctly remove the ghosts present in the scene. Even the method by Gallo et al. [GGC$^+$09] yields small artefacts in one case (Figure 10(f)), probably introduced by the use of a gradient-domain tone-mapping algorithm. Our method identifies and removes all artefacts present in the scenes, while being more efficient than other methods.

Finally, Table 1 lists the performance obtained by BMD for different image resolutions (with and without Exposure Fusion) to generate a fused image. BMD efficiently performs motion detection and it yields very good performance even when applied to large resolution images or to large sequences. Moreover, its integration in Exposure Fusion does not substantially increase the total computational time.

## 4.1 Discussion and Limitations

The kernel sizes used for the dilation and erosion of the motion mask affect the final results and a good balance between the dilation kernel size, $s_d$, and the erosion kernel size, $s_e$, is required. For all our results, we have set $s_e = 3$ and $s_d = 17$ and always yielded good results. As already explained in Section 3, $s_e$ sets the sensibility of the algorithm to isolate and eliminate the outliers from the moving pixels (noisy clusters) and $s_d$ is directly responsible for the enlarging of the moving clusters when moving pixels are missed. Figure 7 shows the impact of different values of $s_d$ and $s_e$ on an unrefined motion map $M^*$.

BMD produces very consistent results, but there are cases where it fails to detect movement clusters. For instance, when the input exposure sequence does not provide enough information to distinguish between



(a) Best $M$. $s_e = 3$, $s_d = 17$. (b) $s_e = 1, s_d = 17$. (c) $s_e = 10, s_d = 17$. (d) $s_e = 3, s_d = 5$. (e) $s_e = 3, s_d = 30$.

Figure 7: Higher values of $s_d$ or lower values of $s_e$ (Figure 7(e) and 7(b)) lead to an extreme movement detection; higher values of $s_e$ or lower values of $s_d$ (Figure 7(c) and 7(d)) lead to incomplete movement detection.

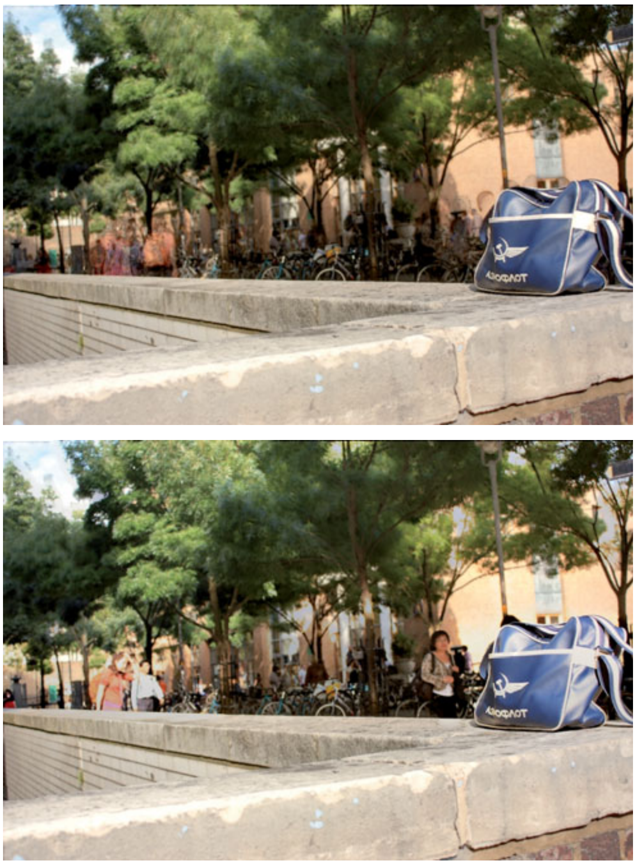
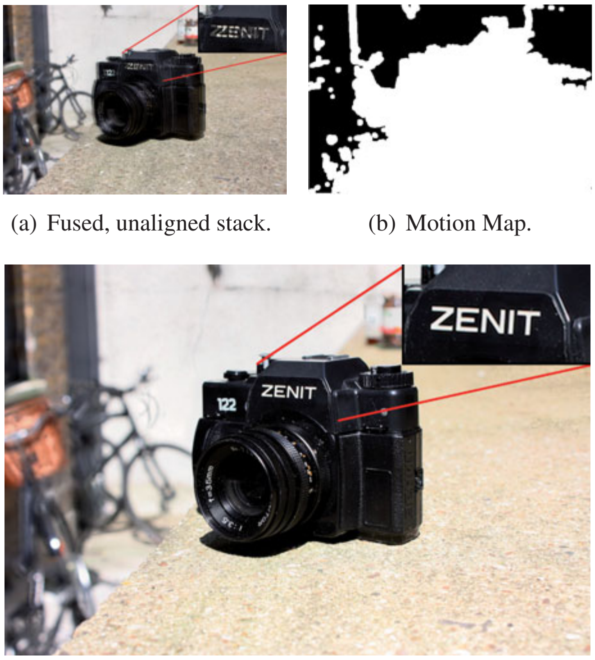


Figure 8: Example of standard fused image (top) and our result (bottom) for a highly dynamic scene.



(a) Fused, unaligned stack. (b) Motion Map.

(c) Fused, unaligned stack using BMD.

Figure 9: BMD result on an unaligned input stack.

still and moving objects, BMD cannot completely identify the motion. This can happen when the scene (or part of it) is over or under exposed for the whole sequence, or when the intensity difference between the moving object and the background is too small, preventing BMD to segment the motion. This is the case of Figure 11(l), where BMD fails in the portion that is always over-exposed (red area). Adding another correctly exposed exposure would prevent the problem.

Further, BMD assumes that the input stack contains only aligned input images. When this is not the case, the algorithm fails in detecting motion regions as non-aligned areas are wrongly classified as dynamic. For instance, Figure 9(c) shows the result of employing BMD motion detection on a stack that contains subtle camera movements (the stack was acquired with an hand-held camera). Even though the scene is completely static, the algorithm identifies large dynamic areas, due to subtle camera movements (Figure 9(b)). This results in a ghost-free fused image (in contrast to the ghosting-affected image generated by directly fusing the stack, Figure 9(a)), which, however, largely corresponds to a single exposure of the stack, as BMD erroneously classifies a large percentage of the image as dynamic. This is expected behaviour for unaligned images. However, we do not consider this as a limitation of our technique, as HDR generation methods commonly require the input stack to be perfectly aligned.

Finally, as already discussed in Section 3.2, the proposed single-exposure HDR integration can dramatically reduce the lighting information used in the final results for HDR, moving areas. We proposed an alternative solutions for these cases, noticing however that such solution might introduce unwanted artefacts due to its dependency from a single threshold value $P$. Dynamically computing such value for each region may improve the proposed approach, but it is not clear whether this might introduce an additional overhead in the final computation. We reserve such investigation for future work.

## 5 Conclusion

We have presented a technique that extends standard HDR imaging techniques to handle dynamic scenes by detecting and correcting ghosting artefacts introduced by moving objects. We have shown that our algorithm works well on a large variety of movement configurations and that it yields fast computation times. The technique is successful even when the motion affects a substantial part of the scene or when the movements are located on the background and are very compact. The results are similar or better than the ones obtained by other techniques. Nonetheless, our motion detection method is much faster, and the combination with Exposure Fusion makes it a highly efficient technique. Our motion detection relies only on simple binary operations, and thus it can be easily implemented directly on camera hardware. Moreover, we believe fused images could be generated almost in real time when implemented on GPUs.

**Citation**

Fabrizio Pece and Jan Kautz, *Bitmap Movement Detection: HDR for Dynamic Scenes*, Journal of Virtual Reality and Broadcasting, 10(2013), no. 2, August 2013, urn:nbn:de:0009-6-36506, ISSN 1860-2037.

## References

[Any] Anyhere Software, *Photosphere*, `http://www.anyhere.com`, Last visited: August 17th, 2012.

[Bog00] Luca Bogoni, *Extending Dynamic Range of Monochrome and Color Images through Fusion*, IEEE CVPR, vol. 3, 2000, pp. 7–12.

[DM97] Paul E. Debevec and Jitendra Malik, *Recovering high dynamic range radiance maps from photographs*, Proceedings of the 24th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 369–378, ISBN 0-89791-896-7.

[GGC$^+$09] Orazio Gallo, Natasha Gelfand, Wei-Chao Chen, Marius Tico, and Kari Pulli, *Artifact-free High Dynamic Range Imaging*, IEEE International Conference on Computational Photography (2009), 1–7.

[HDR] HDRsoft Sarl, *Photomatix*, `http://www.hdrsoft.com`, Last visited August 17th, 2012.

[HS92] Robert M. Haralick and Linda G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, Boston, MA, 1992, ISBN 0201569434.

[JLW08] Katrien Jacobs, Celine Loscos, and Greg Ward, *Automatic High-Dynamic Range Image Generation for Dynamic Scenes*, IEEE Computer Graphics and Applications **28** (2008), no. 2, 84–93, ISSN 0272-1716.

[KAR06] Erum Arif Khan, Ahmet Oguz Akyüz, and Erik Reinhard, *Ghost Removal in High Dynamic Range Images*, IEEE International Conference on Image Processing, 2006, pp. 2005–2008.

[KUWS03] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, *High dynamic range video*, ACM Trans. Graph. **22** (2003), no. 3, 319–325.

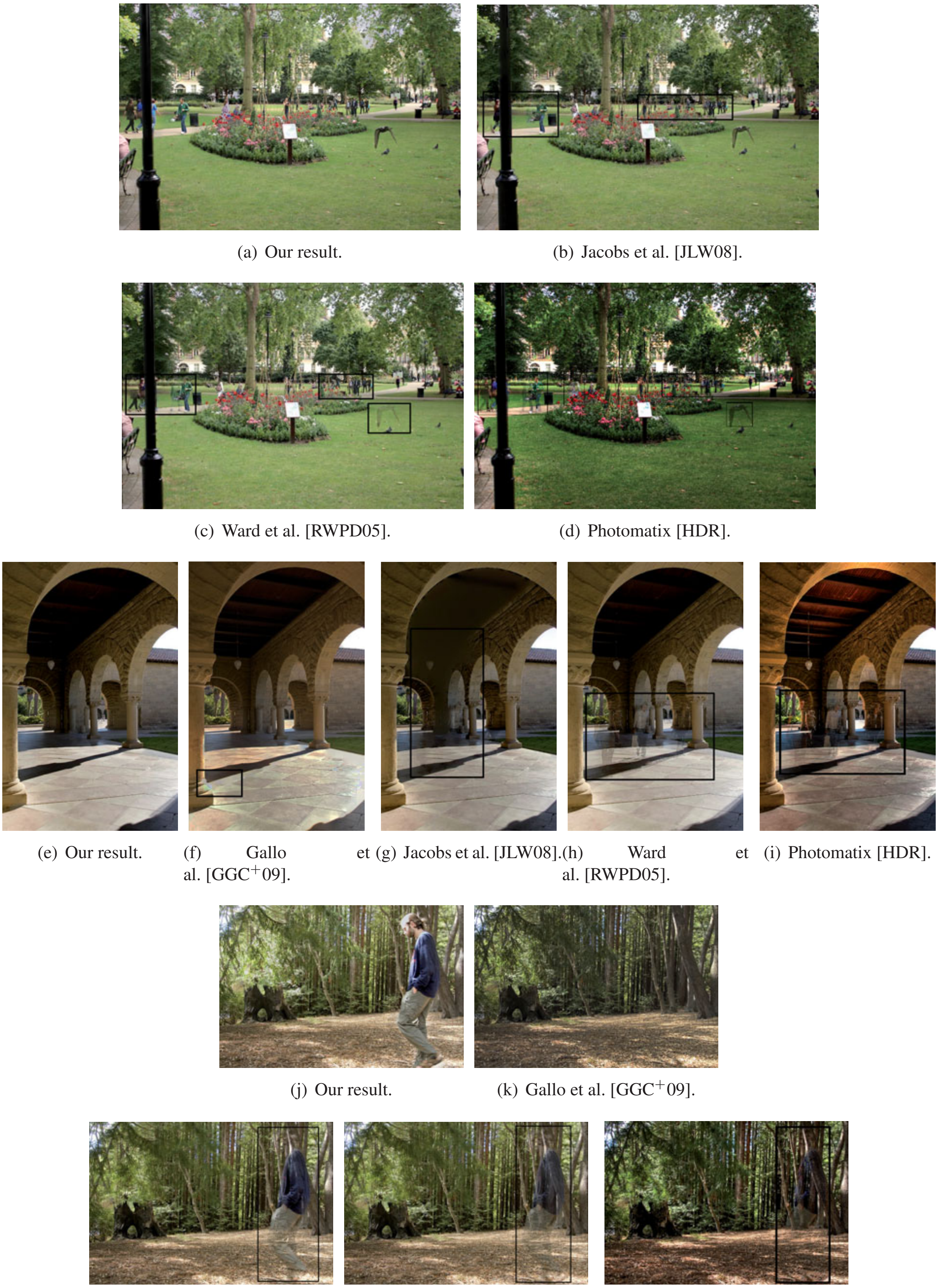
[MKVR07] Tom Mertens, Jan Kautz, and Frank Van Reeth, *Exposure Fusion*, Pacific Graphics, 2007, pp. 382–390.

[MMF02] Steve Mann, Corey Manders, and James Fung, *Painting with looks: photographic images from video using quantimetric processing*, Proceedings of ACM Multimedia, 2002, pp. 117–126, ISBN 1-58113-620-X.

[NM00] Shree K. Nayar and Tomoo Mitsunaga, *High dynamic range imaging: Spatially varying pixel exposures*, Proc. IEEE CVPR, vol. 1, 2000, pp. 472–479.

[RWPD05] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2005, ISBN 0-12-585263-0.

[War03] Greg Ward, *Fast, Robust Image Registration for Compositing High-Dynamic Range Photographs from Handheld Exposures*, Journal of Graphics Tools **8** (2003), no. 2, 17–30, ISSN 1086-7651.



(a) Our result.

(b) Jacobs et al. [JLW08].

(c) Ward et al. [RWPD05].

(d) Photomatix [HDR].

(e) Our result.

(f) Gallo et al. [GGC+09].

(g) Jacobs et al. [JLW08].

(h) Ward et al. [RWPD05].

(i) Photomatix [HDR].

(j) Our result.

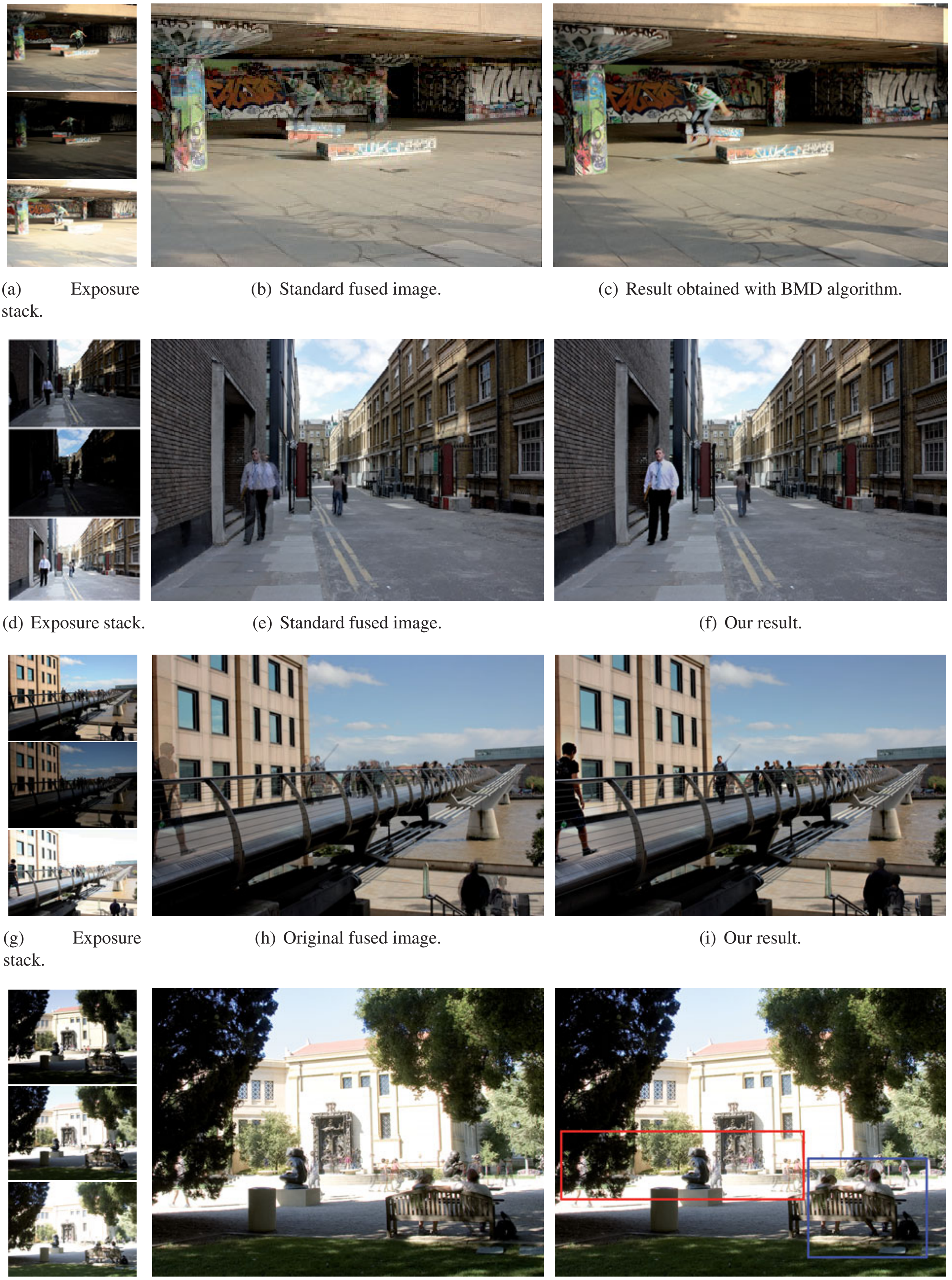
(k) Gallo et al. [GGC+09].

(l) Jacobs et al. [JLW08].

(m) Ward et al. [RWPD05].

(n) Photomatix [HDR].

Figure 10: Variety of comparisons. The exposure stacks used to generate the images in the second and third example are courtesy of Gallo et al. [GGC+09]



(a) Exposure stack.

(b) Standard fused image.

(c) Result obtained with BMD algorithm.

(d) Exposure stack.

(e) Standard fused image.

(f) Our result.

(g) Exposure stack.

(h) Original fused image.

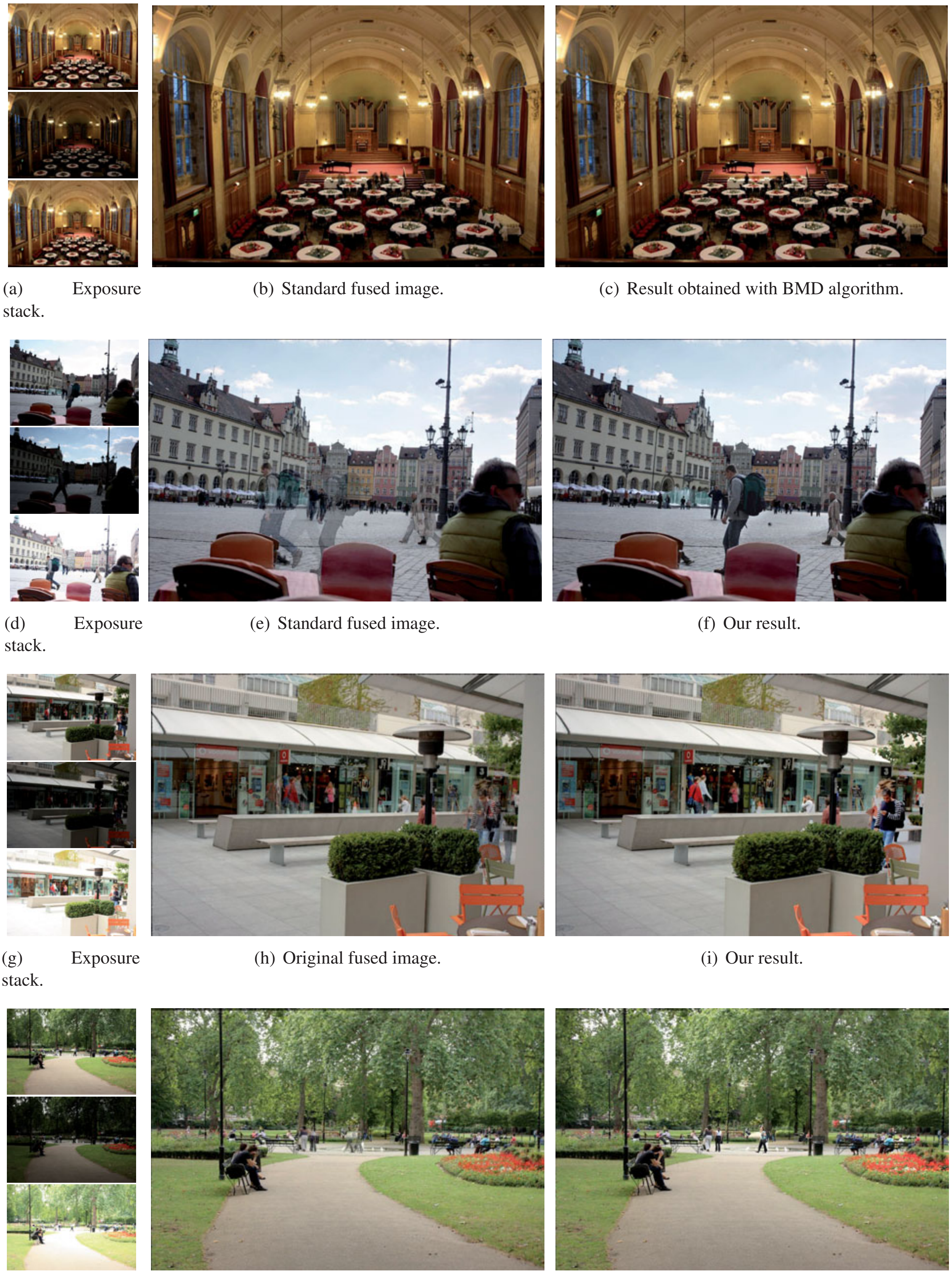
(i) Our result.

(j) Exposure stack.

(k) Original fused image.

(l) Our result. Please note the detection failure in the red box and the correct detection in the blue box.

Figure 11: Variety of results. The images in Figure 11(j) are courtesy of Gallo et al. [GGC$^+$09].



(a) Exposure stack.

(b) Standard fused image.

(c) Result obtained with BMD algorithm.

(d) Exposure stack.

(e) Standard fused image.

(f) Our result.

(g) Exposure stack.

(h) Original fused image.

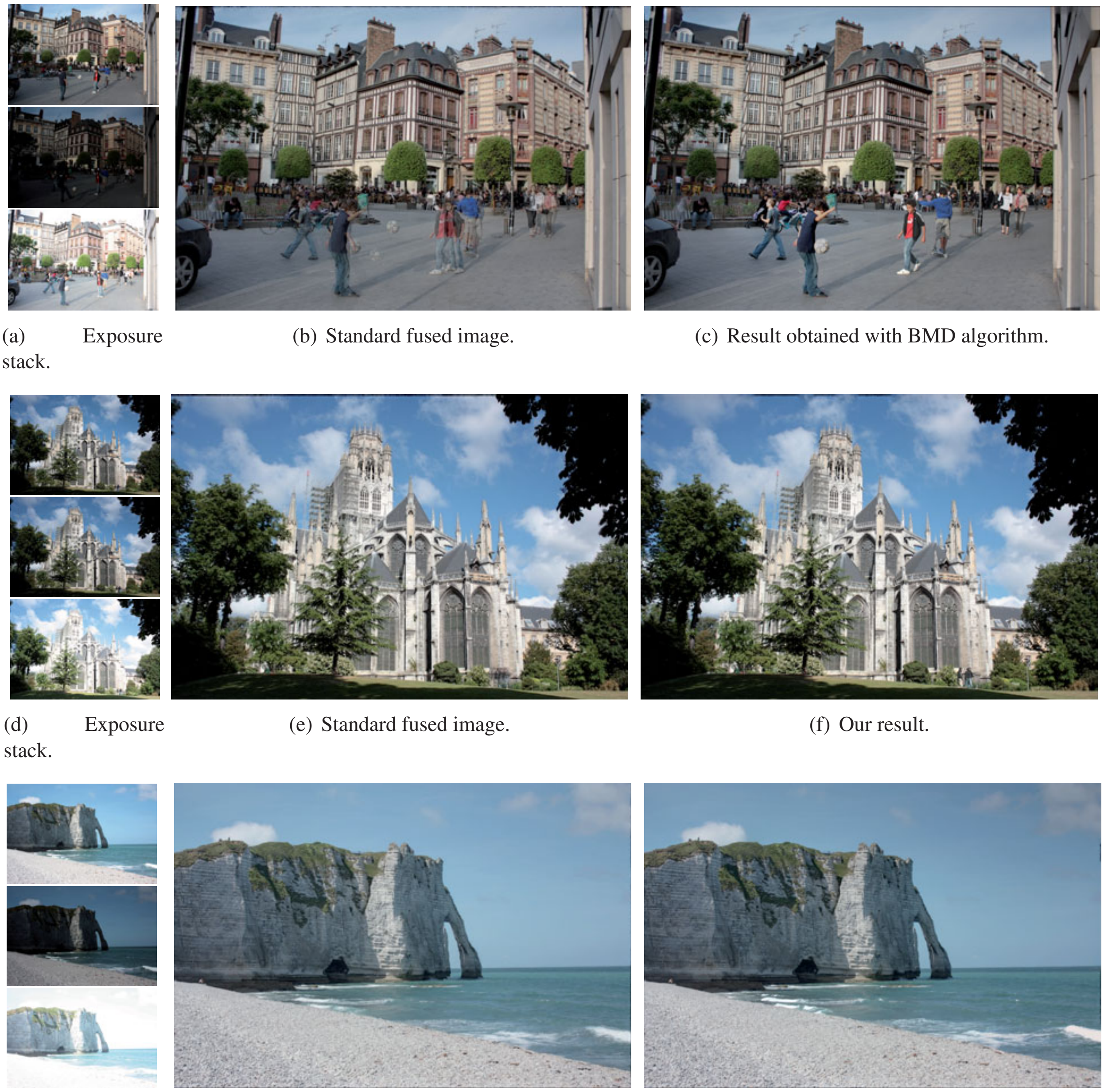
(i) Our result.

(j) Exposure stack.

(k) Original fused image.

(l) Our result.

Figure 12: Variety of results.



(a) Exposure stack.

(b) Standard fused image.

(c) Result obtained with BMD algorithm.

(d) Exposure stack.

(e) Standard fused image.

(f) Our result.

(g) Exposure stack.

(h) Original fused image.

(i) Our result.

Figure 13: Variety of results.