Robust Patch-Based HDR Reconstruction of Dynamic Scenes

Pradeep Sen¹ Nima Khademi Kalantari¹ Dan B Goldman³ Maziar Yæsoubi² Soheil Darabi²

¹University of California, Santa Barbara ²UNM Advanced Graphics Lab ³Adobe

Robust Patch-Based HDR Reconstruction of Dynamic Scenes

1 Introduction

High-dynamic range (HDR) imaging has the potential to transform the world of photography. Unlike traditional low-dynamic range (LDR) images that measure only a small range of the total illumination of a scene, HDR images capture a much wider range and therefore more closely resemble what photographers see with their own eyes. However, despite their tremendous potential, existing approaches for high-quality HDR imaging have severe limitations. For example, specialized camera hardware has been proposed to capture HDR content directly (e.g., [Nayar and Mitsunaga 2000; Tocci et al. 2011]), but these devices are typically expensive and are currently unavailable to the general public.

To make high-quality HDR imaging widespread, we must focus on approaches that use standard digital cameras. The most common approach is to take sequential LDR images at different exposure levels (known as bracketed exposures) and then merge them into an HDR image [Mann and Picard 1995; Debevec and Malik 1997]. Although this technique can produce spectacular results (see, e.g., [Ratcliff 2012]), the original approaches work only for static scenes because they typically assume a constant radiance at each pixel over all exposures. When the scene has moving content (or the camera is hand-held), this method produces ghost-like artifacts from even small misalignments between exposures. This is a serious limitation, since real-world scenes often have moving objects and real-world cameras are not often mounted on tripods.

The problem of removing motion artifacts for sequential HDR imaging has been the subject of extensive research and has led to two major kinds of approaches. The first kind assume that the images are mostly static and that only small parts of the scene have...
Our approach for image hole filling [Wexler et al. 2007], image summarization to exploit self-similarities in images to reconstruct information missing during alignment that was occluded in an exposure of an over-lit scene content as well as camera motion) and can even fill in dynamic scenes captured with a set of broad-exposed, and in its poorly-exposed regions is locally similar to the other LDR sources, containing as much information from them as possible. By directly optimizing local similarity of the output to all sources using a patch-based approach, we effectively integrate the alignment and merging processes, in contrast to previous methods which often aligned the sources before merging them. This results in an algorithm that not only computes the desired HDR image directly, but also reconstructs “aligned” exposures as a by-product that can be merged with any standard technique. Our algorithm can handle sequential exposures with significant camera/scene motion and can produce HDR images superior to previous methods.

2 Previous Work

We begin by reviewing the previous work to remove the HDR ghosting artifacts of dynamic scenes captured with a set of bracketed exposures. A thorough review of HDR imaging is beyond the scope of this paper, so interested readers are directed to texts on the subject [Reinhard et al. 2010; Banterle et al. 2011]. We categorize the two general kinds of proposed algorithms to address the ghosting problem in the subsections that follow.

2.1 Algorithms that reject ghosting artifacts

These algorithms assume the images can be globally registered so that each pixel can be classified as either static or “ghosted” (containing movement across the different exposures). These techniques
try to identify ghosted pixels and only use information from a subset of exposures in these locations.

The key differences between these methods is how they detect the ghosting regions. Liu and El Gamal [2003] proposed a new sensor model that rejects information from ghosted regions. Grosch [2006] mapped pixels from one exposure to the other and used the difference between these values to compute an error map that accounts for motion. Jacobs et al. [2008] proposed approaches based on variance and entropy. Jinno and Okuda [2008] used Markov Random Fields to detect occluded and saturated regions and exclude them from the HDR result. Sidibe et al. [2009] used the fact that pixel values in static regions usually increase as the exposure increases to identify ghosting. Gallo et al. [2009] detected motion between two exposures by measuring the deviation of their pixel values from the expected exposure ratio. Min et al. [2009] proposed to compute multilevel threshold maps from the images and compare them to detect motion. Wu et al. [2010] used criteria such as consistency in the radiance and color across exposures. Pec et al. [2010] computed the median threshold bitmap for each exposure and labeled pixels that did not have the same value as movement. Raman and Chaudhuri [2011] used a segmentation algorithm based on super-pixel grouping to detect which regions have motion. Finally, Zhang and Cham [2012] detected motion by looking for changes in the gradient between exposures.

Some algorithms do not require the explicit identification of ghosted pixels at all. Khan et al. [2006] modified the weights of the HDR merging function based on the probability that a pixel is static. Eden et al. [2006] used the distance of an exposure’s radiance to that of a reference to select a single exposure for each pixel. Heo et al. [2010] computed the joint probability density function between exposures to map values from one exposure to another, and then used the Gaussian-weighted distance to a reference value to weight each exposure during merging.

However, none of these deghosting algorithms can produce accurate results when there is moving HDR content since they all assume that a pixel’s radiance can be computed from the same pixel (or block around it) in all exposures. Instead, a moving HDR object would have properly-exposed pieces in different parts of the image in each frame. For this reason, these papers all show results using only largely static scenes with small moving objects – none are like that of Fig. 1 with a large moving subject. However, these techniques tend to produce fewer artifacts than the optical-flow-based alignment methods we will discuss next, and so commercial HDR software (e.g., [Photomatix 2012]) typically uses deghosting approaches like these.

### 2.2 Algorithms that align the different exposures

These approaches try to align the different LDR exposures before merging them into the final HDR image. Although the alignment of images has long been studied in the image processing and vision communities (see, e.g., [Brown 1992; Zitová and Flusser 2003]), its application to HDR imaging has special considerations. Here, the input images are not of equal exposure so the color constancy assumption of many algorithms is violated. Even if we map images to the same radiance space using the camera response curve [Debevec and Malik 1997; Mitsunaga and Nayar 1999], they will have regions that are too dark/light and therefore invalid during alignment. This makes standard image registration techniques unsuitable for this application.

The simpler approaches to align the LDR sources solve for a transformation that accounts for camera motion between exposures. Ward [2003] solved for a translation factor while Tomaszewski and Mantiu [2007] used SIFT feature points to compute a homography to align the images. Akyüz [2011] used a simple correlation kernel assuming only translation. Yao [2011] used phase cross-correlation to perform global motion estimation. These approaches all assume that the scene is rigid and on a plane, which is not the case for scenes such as the one in Fig. 1.

More sophisticated alignment methods are based on optical flow (OF) algorithms [Lucas and Kanade 1981; Baker et al. 2011]. Bogoni [2000] used local unconstrained motion estimation using optical flow to warp the images into alignment. Kang et al. [2003] significantly improved optical-flow approaches by introducing two key steps: a hierarchical homography to constrain the flow in regions where the reference was too light/dark to make it converge better, and an HDR merging process that rejects the aligned image wherever it is too far from the reference, similar to those used in deghosting approaches. Mangiat and Gibson [2010] proposed a block-based bidirectional optical flow method using color information to find better correspondences.

The current state-of-the-art method in LDR alignment for HDR applications is the work of Zimmer et al. [2011]. They used an optical flow based method to minimize their proposed energy function consisting of a gradient term and a smoothness term to ensure smooth reconstruction of the regions where matching fails due to occlusion or saturation. Based on the displacement map obtained from the previous stage and with another energy function, they reconstruct the HDR image which has also been super-resolved.

In summary, however, the quality of the HDR images produced by these techniques is fundamentally limited by the accuracy of the alignment. Even the state-of-the-art optical flow algorithms are brittle in cases with complex motion and occlusions, which is why many use special HDR merging steps to reject misaligned images (as in deghosting) and cannot use standard merging techniques. Furthermore, optical flow cannot typically synthesize new content and thus cannot handle disoccluded content that could be made visible when aligning one image to another (see, e.g., Fig. 6).

### 3 A new optimization for HDR reconstruction

#### 3.1 Framework

Given a set of $N$ LDR sources taken with different exposures and at different times $(L_1, \ldots, L_N)$, our primary goal is to reconstruct an HDR image $H$ that is aligned to one of them (the reference, called $L_{ref}$), but contains HDR information from all $N$ exposures. To pose the problem as an energy minimization, we begin by asking the question: what are the desired properties of $H$?

---

**Table 1:** Notation used in this paper. Here $l^k(H)$ is the approximate inverse of $h(L_k)$, but not exact because of the clipping process that occurs when capturing an LDR image. $g^k(L_q)$ is simply the composition of $h()$ and $l()$.

| $L_{ref}$ | reference input LDR source |
| $L_1, \ldots, L_N$ | input LDR sources |
| $H$ | HDR image result which should look like $L_{ref}$ but contain information from $L_1, \ldots, L_N$ |
| $l^k(H)$ | produces an LDR image at exposure $k$ from HDR image $H$: $l^k(H) = \text{clip}((H/\text{exposure}(k))^{1/\gamma})$ |
| $h(L_k)$ | maps LDR image $L_k$ to the linear HDR radiance domain: $h(L_k) = (L_k)^\gamma \times \text{exposure}(k)$ |
| $g^k(L_q)$ | maps the $g^k$ LDR source to the $k^{th}$ LDR exposure: $g^k(L_q) = l^k(h(L_q))$ |
| exposure($k$) | indicates the exposure ratio between the $k^{th}$ exposure and the reference, assuming the reference exposure has unit radiances |
| $\alpha_{\text{ref}}$ | trapezoid function indicating how well a pixel of $L_{\text{ref}}$ is exposed |
| $\lambda()$ | triangle weighting function used in traditional HDR merging, defined in Eq. 4 of [Debevec and Malik 1997] |
| $L_{k_{clip}}, L_{(\gamma)}$ | the $p^{th}$ pixel in LDR source $L_k$ and HDR image $H$, respectively |
One desirable property is that if our ideal $H$ is “exposed” with function $h^k(L)$ that maps the radiance values of $H$ to the exposure range of the reference source (see Table 1), it should be very close to $L_{ref}$. Likewise, if the LDR reference $L_{ref}$ is mapped to the linear radiance domain $h(L_{ref})$, it should be similar to $H$ for the pixels where it is properly exposed. This ensures that $H$ looks like $L_{ref}$ so that it appears to be taken by a real camera and does not have unrealistic artifacts. Also, this helps to preserve as much information as possible from the well-exposed pixels of $L_{ref}$.

To include information from all other exposures in places where $L_{ref}$ is poorly exposed, the HDR image $H$ in these parts should be “similar” to any input source $L_k$ mapped through the response curve of the $k^{th}$ exposure: $l^k(H)$. Since the scene and camera are moving, however, $l^k(H)$ need not match $L_k$ exactly, because $H$ might not be aligned to both $L_{ref}$ and $L_k$ simultaneously. Instead, we propose a metric based on bidirectional similarity (BDS) [Simakov et al. 2008] to measure this similarity concretely. Minimizing BDS implies that for every patch of pixels in $l^k(H)$ there should be a comparable patch in $L_k$ (which Simakov et al. called “coherence”), and for every patch in $L_k$ there is a comparable patch in $l^k(H)$ (called “completeness”), across multiple scales.

Combining these two properties together results in an energy equation with two basic terms:

$$E(H) = \sum_{p \in \text{pixels}} [\alpha_{ref}(p) \cdot (h(L_{ref})_p - H(p))^2 + (1 - \alpha_{ref}(p)) \cdot E_{MBDS}(H | L_1, \ldots, L_N)] \quad (1)$$

Here, the first term ensures that $H$ is similar to $h(L_{ref})$ in an $L_2$ sense for pixels that are well-exposed. The second term constrains the remaining poorly exposed pixels to match the other exposures, using a modified form of BDS ($E_{MBDS}()$). The balance between these two terms is controlled by a per-pixel weighting $\alpha_{ref}$, which indicates how well each reference pixel is exposed (see Sec. 4.3). Instead of the traditional BDS measure defined by Simakov et al. [2008], the bidirectional metric was extended to utilize all sources, which worked more generally for our application'. This new multisource bidirectional similarity measure (MBDS) is described in more detail in the Appendix. MBDS is applied to all $N$ source images in our input stack by defining an energy function that tries to keep each exposure $H$ of the HDR image $H$ as similar as possible to all input sources adjusted to that exposure:

$$E_{MBDS}(H | L_1, \ldots, L_N) = \sum_{k=1}^{N} \text{MBDS}(l^k(H) | g^k(L_1), \ldots, g^k(L_N)), \quad (2)$$

where $g^k(L_n)$ is a function that maps the $k^{th}$ LDR source to the $k^{th}$ LDR exposure (see Table 1). This function ensures that every exposure of the HDR image $l^k(H)$ is “similar” to the exposure-adjusted versions of all $N$ input images in the regions that their pixels are properly exposed, which will help to bring information from these other exposures to produce the final HDR image.

We call Eq. 1 the HDR image synthesis equation. The first term ensures that the algorithm uses information from $L_{ref}$ only in the regions where its pixels are properly exposed. In these regions, the resulting HDR image $H$ should be a close match to the reference input. In the parts of the image where it is over/under-exposed, the second term adds information from the other exposures through a bidirectional similarity energy term. In these poorly-exposed regions, every patch in the final HDR image $H$ at a given exposure

\[1\] However, we found experimentally that standard BDS still worked fine in the majority of cases. See discussion in Sec. 4.5.

![Figure 3: This is the inner core of the algorithm that runs at a single scale to find a solution to the HDR image synthesis equation. Only three exposure levels are shown here, although our algorithm runs on all $N$ exposures and is repeated at multiple scales.](image-url)
which forms the core of our HDR image reconstruction algorithm (see Fig. 3). To do this, the energy is minimized in two stages:

**Stage 1:** In the first stage, the algorithm minimizes for the $I_1, \ldots, I_N$ that appear in the second and third terms of Eq. 4. It first uses a search and vote similar to Simakov et al. [2008] (Sec. 4.2), which solves for the MBDS term by enforcing both the completeness and coherency terms. It then blends in $I^H_p$ to each $I_k$ using the previous $H$ in order to encourage the solution to be close to the exposed value from $H$, which optimizes for $I_k$’s in the third term.

**Stage 2:** Here the algorithm optimizes for the $H$ variable in the first and last terms of Eq. 4. First, it merges images $I_1, \ldots, I_N$ that were computed in the first stage into an intermediate HDR result $\tilde{H}$ using the standard HDR merge process [Debevec and Malik 1997]:

$$H_{(p)} \leftarrow \frac{\sum_{k=1}^{N} A(I_{(k)}(p)) h(I_{(k)}(p))}{\sum_{k=1}^{N} A(I_{(k)}(p))}.$$  

(5)

This $\tilde{H}$ contains information from all the other exposures which optimizes for $H$ in the first term of Eq. 4. However, $H$ still needs to be set to match the reference $L_{\text{ref}}$ in parts where $L_{\text{ref}}$ is well-exposed to minimize the first term of Eq. 4. To do this, our algorithm always injects the input image directly into $H$ using the appropriate alpha blending weights from Eq. 4:

$$H_{(p)} \leftarrow \alpha_{\text{ref}}_{(p)} \cdot h(L_{\text{ref}}(p)) + (1 - \alpha_{\text{ref}}_{(p)}) \cdot \tilde{H}_{(p)}.$$  

(6)

Once the new $H$ has been computed, it is used to extract the new image targets for our next iteration and our algorithm goes back to stage 1. These two stages are performed at every iteration of the algorithm until it converges. Furthermore, as is common for patch-based methods like this (e.g., Simakov et al. [2008]), this core algorithm is performed at multiple scales, starting at the coarsest resolution and working to the finest (Sec. 4.4). After the algorithm has converged, it has solved for both the desired HDR image $H$ as well as the “aligned” images at each exposure $I_1, \ldots, I_N$. An overview of our method is listed in Algorithm 1.

4 Implementation

This section provides many of the implementation details needed to reproduce our results and accelerate our algorithm. Readers interested in our implementation can find source code and data sets at http://www.ece.ucsb.edu/~psen/hdr/.

4.1 Image pre-processing

If the LDR sources are in JPEG or some other non-linear format, we first convert them into a linear space (range 0 to 1) using the appropriate camera response curve [Debevec and Malik 1997] which is assumed to be known or can be estimated using established techniques (e.g., [Mitsunaga and Nayar 1999; Lin et al. 2004; Lin and Zhang 2005; Lin and Chang 2009]). A gamma curve with $\gamma = 2.2$ is then applied to the linear raw data to get the input sources $L_1, \ldots, L_N$ for our algorithm. This is done because our algorithm computes differences between patches during matching, and doing this in a linear space does not adequately reflect the way people see differences perceptually. We found that by performing the MBDS process in the gamma domain, the final reconstructions look better in the dark parts of the image. All image operations are done in floating point (with exception of the search step discussed next, which is uint8) and the range of the reference exposure is defined to be of unit radiance.

4.2 Search and vote

To begin our matching process, our algorithm needs an initial guess for the $I_k$’s in the first iteration. To do this, the reference image is simply exposure-corrected to come up with the target for each exposure: $I_k \leftarrow g^h(I_{\text{ref}})$. Note that the initial target of the optimization does not affect the final result much, since this only impacts the first iteration at the coarsest scale. The two stages of our algorithm ensure that after the first iteration, information from all sources is propagated to all other exposure levels.

To implement the MBDS metric, we used the publicly-available implementation of Barnes et al. [2009] for the search/vote portion of the first stage accelerated by the PatchMatch algorithm, with modifications to handle multiple sources for MBDS. For each target exposure level $k$, our method runs a dense search step a repeated number of times on all adjusted source exposures $g^h(I_{\text{ref}})$ using the current image at that level $I_k$ as the MBDS target input. The bidirectional search produces two nearest neighbor fields (NNF’s) for each source exposure $q$: one for coherence and one for completeness. Note that the completeness search is masked, meaning that the search is only conducted in the well-exposed parts of each source exposure $g^h(I_q)$. This effectively implements the $w_\gamma(P)$ term in Eq. 9 with a hard mask. For every pixel in the final coherence NNF, the algorithm chooses the one in the stack of NNF’s that results in the smallest $L_2$ distance, which handles the min term over all the sources in Eq. 9. This results in $N$ NNF’s for the completeness term and one NNF (with an additional component to identify the source) for the coherence term for every exposure level $q$.

For voting, the patches for the coherence NNF are summed in the standard way [Simakov et al. 2008] using the patches from the appropriate exposure at each pixel. For the completeness NNF’s, on the other hand, our algorithm uses each NNF to sum the respective patches from each adjusted exposure and then averages them together. The final result can then be generated by summing these two terms together and then dividing by the appropriate weight, which gives us our new $I_k$. This process is repeated for all $N$ sources.

4.3 HDR merge

In order to accelerate the convergence of our algorithm, it should avoid blending pixels from other sources with the reference exposure in Eq. 6 during the merging process if they have been clearly misaligned. To implement a simple consistency check, the calculation of $\tilde{H}$ in Eq. 5 is split into two parts: $\tilde{H}^+$ which is a merge of the images that are lower than the reference (by computing Eq. 5 from $k = 1$ to ref $-1$), and $\tilde{H}^-$ which is a merge of the images that
are higher than the reference (by computing Eq. 5 from \(k = \text{ref} + 1\) to \(N\)). We then approximate Eq. 6 as:

\[
H(p) = (1 - \alpha_{\text{ref}}(p)) (\alpha^+(p) \tilde{H}^+(p) + \alpha^-(p) \tilde{H}^-(p)) + \alpha_{\text{ref}}(p) \cdot g(L_{\text{ref}}(p)),
\]

where the \(\alpha\) blend values are based on the pixel values of the reference source \(L_{\text{ref}}\), as shown by the plots in the figure to the right. In our implementation, we used values of 0.1 and 0.9 for the minimum and maximum valid values \(v_{\text{min}}\) and \(v_{\text{max}}\). This equation can be understood better by examining the process at the finest iteration scale, where \(\alpha^+\) and \(\alpha^-\) cannot both be 1 at the same time (this is not true at coarser scales, where the reference image has been downsampled using an anti-aliasing filter). The \(\alpha^+\) term focuses on the lower values of the reference (where the higher exposures will provide detail), while \(\alpha^-\) focuses on the higher values (where the lower exposures will do this). Because of the triangle functions \(A\) used to weight the exposures, the exposures lower than the reference would not contribute much to the region covered by the \(\alpha^+\) and vice-versa. So \((1 - \alpha_{\text{ref}}(p)) (\alpha^+(p) \tilde{H}^+(p) + \alpha^-(p) \tilde{H}^-(p)) \approx (1 - \alpha_{\text{ref}}(p)) \tilde{H}\).

This separation of \(\tilde{H}\) into two terms now allows us to do a simple consistency check using Eq. 7 directly. In parts of the image where the reference is under-exposed (\(L_{\text{ref}}(p) < v_{\text{min}}\)), our algorithm only blends values of \(\tilde{H}^+\) with Eq. 7 if \(L_{\text{ref}}(\tilde{H}^+) < v_{\text{min}}\). Likewise, wherever the reference is over-saturated (\(L_{\text{ref}}(\tilde{H}^-) > v_{\text{max}}\)), values of \(\tilde{H}^-\) are only blended if \(L_{\text{ref}}(\tilde{H}^-) > v_{\text{max}}\).

Unlike many optical flow-based algorithms, the aligned images \(I_1, \ldots, I_N\) after our algorithm has converged do not require any consistency check and can be merged using any standard procedure. Furthermore, unlike deghosting algorithms where consistency checks are used in one pass to cull information, ours is part of the optimization to help convergence. Removing this check produces comparable images with similar HDR content, but the algorithm takes longer to converge.

The second stage concludes by merging the images to form intermediate HDR image \(H\). Function \(\tilde{I}(H)\) then extracts the correct exposures to create targets for the first stage in the next iteration. These are used by the matching/voting step, along with the NNF’s from the previous iteration described in Sec. 4.2.

4.4 Extending our algorithm to multiple scales

Our optimization is a multiscale algorithm that performs the iterations shown in Fig. 3 over multiple scales (see, e.g., [Simakov et al. 2008]). In other words, it first matches the global structure at the coarse scales and then matches the local detail at the fine scales. As a preprocess for acceleration, our algorithm first generates an image pyramid for all input sources by downsampling them using a Lanczos filter. After it completes the set of iterations for Fig. 3 (lines 3–14 in Algorithm 1), it moves to the next scale. In our implementation, the lowest-resolution scale has \(35\) pixels in the smaller dimension and we have a total of \(10\) scales, so the algorithm must upscale the images by a ratio of \(\sqrt{\pi/35}\) in each dimension (\(\pi\) is the minimum dimension of the final image) when moving up a scale. The number of iterations is also adjusted at each scale, starting with \(50\) at the lowest scale and linearly decreasing this to \(5\) iterations at the finest scale.

When a scale is completely converged, the regular merging step is not performed. Rather, the final reconstructed LDR images are upscaled up to the next scale using a Lanczos filter. These upsampled images are then merged with the reference image from the input image pyramid using the same merging algorithm described earlier. This process injects the extra detail now available in the higher-resolution reference image into our iteration process. The algorithm also up scales all of the NNF’s computed in the previous iteration, and then proceeds with the next scale’s iterations.
We also experimented with varying the number of sources (the one that matched that particular exposure, tested, the algorithm produced artifact-free results using only one available to MBDS instead of using all \( N \)). For acceleration, the algorithm successfully with inputs in both JPEG and RAW formats, without any alignment preprocess except for the adjustments to JPEG’s described in Sec. 4.1. The HDR results were then manually tonemapped using Photomatix [2012] for presentation in the paper using exaggerated settings to enhance the HDR detail.

To judge the quality of our reconstructed images, we compare against several state-of-the-art approaches for HDR image alignment and deghosting. We compare our results to four optical flow (OF) algorithms: (1) the motion detail preserving optical flow (MDP OF) algorithm of Xu et al. [2010], (2) the large displacement optical flow (LD OF) of Brox and Malik [2011], (3) the optical flow implementation of Liu (Liu OF) [2009] based on the work of Brox et al. [2004] and Bruhn et al. [2005] to enable them to handle large motion, and (4) the algorithm of Zimmer et al. [2011], which is perhaps the state-of-the-art in preprocess alignment methods.

For the first three OF methods, Kang et al.’s [2003] hierarchical homography was used to constrain the flow where the reference image was unreliable, but it only improved the results for a few scenes. These methods often did equally well (or sometimes better) without it (we show the best results obtained either way). We also used Kang et al.’s merging approach, which improved the quality of the OF results considerably by filtering out misalignments. Therefore, the OF results presented here are at least comparable to Kang et

### 5 Results

We implemented our patch-based HDR image reconstruction algorithm in MATLAB which was sufficient for our purposes and tested it on a variety of real images, as shown throughout the paper. In every case, we selected an exposure around the middle of the stack as the reference. Although in theory any image could have been the reference, the middle exposure is usually well exposed, which gives us more well-exposed pixels to work with. We tested

![Figure 6: Our patch-based optimization can hole-fill information when visibility inconsistencies occur, which is not possible with any of the previous approaches. In this example, we have two input images (high and low, separated by 4 stops), and we are registering to the high exposure. However, the desired detail in the background of the low image is occluded by the subject, so the algorithm must reconstruct this missing information when aligning the images. Clearly optical flow methods and deghosting methods cannot handle this situation. Our algorithm, on the other hand, uses the information surrounding the hole to fill it in a plausible manner.](image)

![Figure 7: Optical flow methods cannot maintain the continuity of the scene outside the window, while Photomatix’s ghost-removal algorithm uses only one exposure in the moving regions, resulting in saturated halos around the head and branches outside. Tonemapping (TM) the reference shows that it is missing information outside the window, while our method produces acceptable results.](image)

![Figure 8: The complex movement in this scene causes problems for OF algorithms. Our algorithm matches the color quality of the ghosted HDR image the best, but without motion artifacts.](image)
Figure 10: Comparison of our aligned reconstructions and HDR images with those of Zimmer et al. [2011]. Zimmer was given the input images to run their code on them directly. In this case, their method is not able to align the moving objects (e.g., the man’s reflection on the piano) which introduces ghosting in the final HDR image. Our method, on the other hand, produces better results.

Figure 9: Our algorithm is able to faithfully reconstruct this complex scene. The optical flow methods, however, have artifacts, e.g., in the reflection of the hands on the piano.

Figure 11: Comparison with Zimmer et al.’s method [2011] on their failure case. Our method can reconstruct the people and the vehicles despite their motion. Furthermore, our method brings in extra HDR detail in the clouds. Image courtesy of Henning Zimmer.

images produced by the different methods in Fig. 5. We see that our algorithm produces results closer to the ground truth image. In terms of MSE, our aligned reconstructions were one to two orders of magnitude better than the OF approaches.

The next test scene, Fig. 6, demonstrates the ability of our algorithm to fill in a visibility hole with complex information, which is difficult for OF algorithms. These OF methods have artifacts even after Kang et al.’s plausibility map rejects misalignments. Degaosting also fails for this scene, since the motion is in an HDR region and the algorithm has to choose which image to draw the radiance values from. In this case, it draws from the reference image (the high exposure), but the pixels are saturated which causes the radiance to be clamped in this region, producing a dark halo when tonemapped. Our algorithm, on the other hand, is able to reconstruct the detail in the occluded region using the information from neighboring patches that are visible, since our HDR image synthesis equation produces a final image that has content that exists somewhere in all input images.

Finally, Figs. 7 – 11 show the results of our algorithm applied to natural scenes and compared to several of the previous approaches. Specifically, Figs. 10 and 11 are comparisons with Zimmer et al.’s optical flow method [2011], the state-of-the-art approach for HDR image alignment. For Fig. 10, Zimmer was given our input sources to run their algorithm on, while in Fig. 11 our algorithm was run on one of their failure cases. In these examples, their algorithm is unable to align the sources to the reference in places where there is complex motion, while ours produces aligned images that can be merged into an artifact-free HDR result. We also refer readers to the additional images and results available on our website.

In terms of timing, our accelerated code runs on 7 input images in less than 3 minutes at 1350 × 900 resolution on an Intel dual quad-
This figure shows how our algorithm can sharpen an input source to match the depth of field of the reference. For this scene (our HDR result shown on the left), 10 stops of dynamic range were captured by changing both the aperture and the shutter time. This was the only way to take this picture since the camera was hand-held. On the right we show one of the original input frames, as well as our reconstruction. We see that the out-of-focus region on the bench has been made sharper to match the reference.

Figure 13: This scene from [Gallo et al. 2009] has moving people that sometimes appear in only one frame. We show the results of the deghosting methods of Gallo et al. (left) and Pece and Kautz [2010] (middle) using images provided by the respective authors. The former has block artifacts because of their per-block motion detection, and the latter still has some ghosting. Our method (top and right) can reconstruct the moving people, but it has artifacts as well. These appear as “washed out” regions where our algorithm only had information from one LDR image because the people in the reference disappeared in the other frames. Images courtesy of Orazio Gallo and Jan Kautz.

6 Discussion

Most photographers take bracketed exposures for HDR by changing only the shutter time between exposures, which maintains the same depth of field in each image and facilitates alignment/merging. Because our algorithm automatically aligns the reconstructed images while solving for the HDR result, we can produce good results even when the aperture changes between exposures, something that happened for most of the scenes acquired for this paper. Fig. 12 shows how our algorithm “sharpens” an input image to match the depth of field of the reference. This effect is stronger when using the “pairwise” accelerated version of our algorithm described in Sec. 4.5.

This capability gives photographers an extra parameter for adjusting their exposures in bracketed photography. For example, achieving 10 stops of additional dynamic range by changing the shutter time alone requires the longest exposure to be 1024× the shortest, which is impractical with hand-held cameras. Our approach gives photographers the flexibility to change the aperture as well when taking bracketed exposures, thereby allowing them to capture a larger dynamic range than could otherwise be captured.

However, our approach to HDR imaging has limitations. Because our algorithm preserves information from the reference wherever it is well-exposed, it can propagate artifacts from the reference (e.g., noise) in these regions to the HDR output. Also the voting process used to reconstruct the aligned images can sometimes introduce slight blurring which can be visible in the final output, although this is mitigated at higher resolutions. Finally, unlike specialized HDR cameras that capture all exposures simultaneously, our algorithm cannot reconstruct the HDR content if it is occluded during the exposures that would measure it properly. These regions would contain only LDR content when reconstructed, since our algorithm does not have valid information from the other exposures to draw from. An example of this is shown in Fig. 13. Addressing these issues is a subject of future work.

One advantage of our technique over HDR camera hardware is that we can adjust the exposure separation between images based on scene content. Different scenes have different dynamic ranges, and this flexibility ensures that we are “sampling” the dynamic range efficiently. HDR camera hardware cannot typically do this because the separation between exposures is usually fixed by the hardware.

We hope that this algorithm takes a step towards making high-quality HDR imaging more available to the general public. In the future, it is possible that camera manufacturers provide firmware to automatically take a series of bracketed set exposures for every scene to produce images like those we show in this paper.

7 Conclusion

In this paper, we have presented a novel framework for HDR reconstruction based on a new energy-minimization equation called the HDR image synthesis equation. This equation codifies the objective of many HDR imaging approaches: to produce an HDR image that coherently uses all the content in the input exposures while being aligned to one of them. Our solution effectively integrates the LDR image alignment and HDR merging processes into one optimization, which is robust to complex motion and can successfully handle a wide range of natural scenes.

Appendix

Finally, we examine the similarity metric that might be used for our application. One option is the bidirectional similarity metric from Simakov et al. [2008]:

$$
BDS(T | S) = \frac{1}{|S|} \sum_{P \in S, Q \in T} \min d(P, Q) + \frac{1}{|T|} \sum_{Q \in T, P \in S} \min d(Q, P). \quad (8)
$$
tion between the exposures. In some cases, content that should be completeness) and vice versa (second term, coherence). However, ensures that all of the patches in \( S \). This function takes a pair of images (source \( L_{\text{ref}-3} \) to \( L_{\text{ref}} \). The arrow on the reference indicates a region that is saturated but is also occluded in the \( L_{\text{ref}-1} \) image. If we only use one source in the MBDS function, we do not have access to the correct, well-exposed information and therefore get an incorrect result as can be seen in the image in the lower right. By using all \( N \) sources simultaneously, we have access to \( L_{\text{ref}-3} \) and \( L_{\text{ref}-3} \) which provide the missing information to get a better HDR result.

This function takes a pair of images (source \( S \) and target \( T \)) and ensures that all of the patches in \( S \) can be found in \( T \) (first term, completeness) and vice versa (second term, coherence). However, this might not work in our application when there is significant motion between the exposures. In some cases, content that should be visible in the \( n^\text{th} \) exposure when “aligned” with the reference exposure might be occluded in \( L_{\text{ref}} \) (see, e.g., Fig. 14), so bidirectional similarity would not introduce the missing information. Instead, we extend the BDS function to use information from all other exposures as well, because the missing content might be visible (and well-exposed) in one of these other images. This leads to a new multisource bidirectional similarity (or rather dissimilarity) measure for our application:

\[
\text{MBDS}(T \mid S_1, \ldots, S_N) = \frac{1}{N} \sum_{k=1}^{N} \sum_{P \in S_{1}, \ldots, S_N} w_k(P) \min_{Q \in T} d(P, Q) + \\frac{1}{|T|} \sum_{Q \in T} \min_{P \in S_{1}, \ldots, S_N} d(Q, P),
\]

where \(|T|\) is the number of patches in target \( T \), \( P \) and \( Q \) are patches in \( S \) and \( T \), respectively, and \( d() \) is an \( L_2 \) distance metric similar to the one used in standard BDS. Note the addition of the \( w_k(P) \) term which weighs the source patches when calculating completeness based on how well-exposed they are. This helps us ignore over/under-exposed patches and give priority to well-exposed source patches when multiple sources map to the same target location. These weights are normalized to sum to 1.

The bidirectionality of our similarity metric is quite important in our application. Using only the coherence term without completeness could lead to repetitions and other implausible artifacts like those sometimes seen in hole filling (which typically uses only coherence), as in the hole-filling result in Fig. 2 where the woman’s face is duplicated. In these cases, the algorithm is relying solely on the boundary conditions around the edge of the poorly-exposed region to infer its contents. If there is ambiguity at the boundary, the algorithm without the completeness term could coherently copy content from the wrong part of the image to fill this area.

The completeness term helps ensure that appropriate content is drawn from the input images in these regions by suppressing unwanted duplication. Of course, in the case that an object appears only in a non-reference input source, completeness might incorrectly encourage such objects to appear in the final result. However, as with other patch-based algorithms that use completeness (e.g., Simakov et al.’s retargeting application), stray objects tend to be suppressed by averaging during the voting process and do not show up in the final result.

Acknowledgments

We thank Xu, Liu, and Brox for making their optical flow implementations available online for our comparisons. We also used Mertens’s exposure fusion code in our implementation of Zhang’s algorithm, and Henning Zimmer graciously ran his algorithm on our test cases for Fig. 10. Zimmer, Gallo, and Kautz also provided comparison images for Figs. 11 and 13. Finally, we acknowledge the helpful comments by the anonymous reviewers that greatly improved the final version of the paper. This work was partially funded by a National Science Foundation CAREER grant (IIS-0845396) and by Adobe.

References


