High Dynamic Range Video with Ghost Removal

Stephen Mangiat and Jerry Gibson

University of California, Santa Barbara, CA, 93106

ABSTRACT

We propose a new method for ghost-free high dynamic range (HDR) video taken with a camera that captures alternating short and long exposures. These exposures may be combined using traditional HDR techniques, however motion in a dynamic scene will lead to ghosting artifacts. Due to occlusions and fast moving objects, a gradient-based optical flow motion compensation method will fail to eliminate all ghosting. As such, we perform simpler block-based motion estimation and refine the motion vectors in saturated regions using color similarity in the adjacent frames. The block-based search allows motion to be calculated directly between adjacent frames over a larger search range, yet at the cost of decreased motion fidelity. To address this, we investigate a new method to fix registration errors and block artifacts using a cross-bilateral filter to preserve the edges and structure of the original frame while retaining the HDR color information. Results show promising dynamic range expansion for videos with fast local motion.

Keywords: High Dynamic Range (HDR) Video, Block-based Motion Estimation, Bilateral Filter

1. INTRODUCTION

Most consumer cameras capture only a small fraction of the brightness variation that occurs in everyday life. A typical 24-bit color sensor can reproduce 256 levels per channel, whereas an outdoor sunlit scene may require on the order of 10,000 levels.[1] The result of this limited dynamic range is saturated pixels and poorly exposed images. Auto-exposure mechanisms try to minimize the number of saturated pixels or correctly expose a region of interest such as a person's face, yet they fail to correctly expose the entire frame. As shown in Fig. 1 (a) and (b), this leads to image degradations such as white skies or shadowy foregrounds.

High Dynamic Range Imaging (HDRI) uses hardware or software methods to produce images with expanded dynamic range, as shown in Fig. 1 (c). The most common method for HDR still photography combines multiple exposures of a scene. In this way, the bright regions will be captured in the shorter exposures while the dark regions are captured in the longer exposures. Using pixel values, shutter times, and the camera response function, one can estimate a scene-referred, high dynamic range radiance map. This HDR radiance map is mapped back into displayable range on low dynamic range media using "tone mapping".[1]
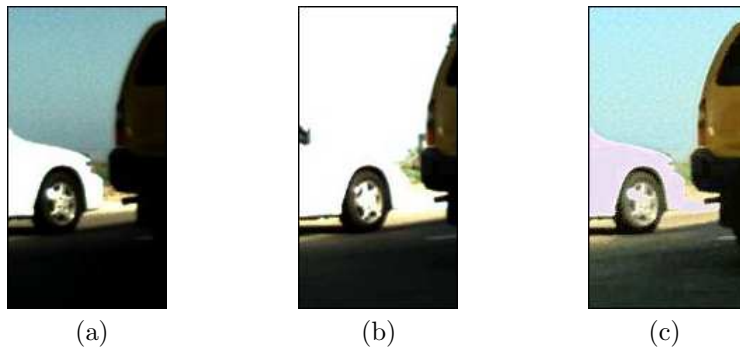

E-mail: {smangiat, gibson}@ece.ucsb.edu

Figure 1. (a) Low Dynamic Range, Short Exposure (b) Low Dynamic Range, Long Exposure (c) High Dynamic Range

The method described above works well for static scenes, but is not directly applicable to video because pixels must correspond exactly between the different exposures. Any motion between frames will lead to ghosting artifacts. Ref. 2 addressed this by alternating the exposure between consecutive frames and warping adjacent frames onto the current frame using a gradient-based optical flow technique. An alternating exposure technique is advantageous because it can be performed using cheap camera sensors, which often have very poor dynamic range. One important application that uses inexpensive hardware and will greatly benefit from HDR video is video conferencing (VC). Saturated pixels on a user's face severely degrade the quality of experience, and this problem will be worsened as mobile devices move conferences outdoors. A VC scenario also motivates a real-time software method.

Building upon the work of Ref. 2, we illustrate a new method to produce HDR video using a camera that captures frames with alternating short and long exposures. Motion estimation is again used to map adjacent frames onto the current frame. However, due to noise, saturations, and fast moving objects, gradient-based optical flow will fail to eliminate all ghosting. Alternatively, we use simpler block-based motion estimation, and enhance these results using the color information in areas with good correspondence and the edge information of the current frame to find and fix poorly registered pixels. Brief overviews of our system and capture methods are provided in Sec. 2 and Sec. 3. This is followed by a detailed discussion of our motion estimation (Sec. 4), HDR reconstruction, and artifact removal modules (Sec. 5). Finally, Sec. 6 will show some promising HDR results for dynamic videos.

2. SYSTEM OVERVIEW

A system overview for our high dynamic range video method is illustrated in Fig. 2. The inputs into this process are the current, previous, and next frame, spanning different exposure times. In order to obtain multiple exposures at every time instant, all three frames are passed into a block-based motion estimation module. The current frame and its prediction are then used to reconstruct a high dynamic range radiance map, which is then tone mapped back into displayable range. Finally, block artifacts and other mis-registrations are fixed in the artifact removal module, creating the final HDR frame. A high dynamic range video is ultimately recovered at the input frame rate.
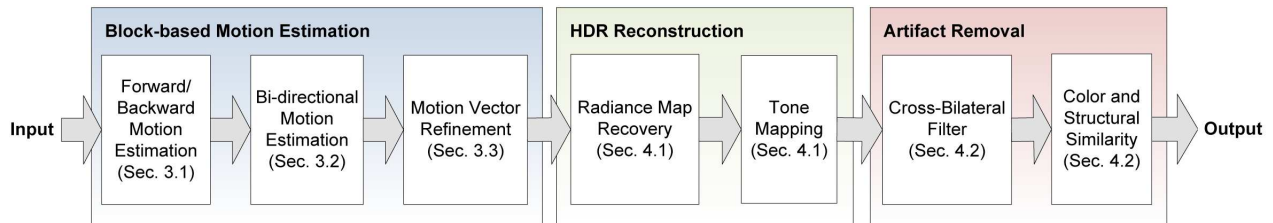

Figure 2. High Dynamic Range Video System Overview

3. CAMERA RESPONSE CURVE AND CAPTURE

Prior to video capture, we estimate the camera response curve for the red, green, and blue channels using a sequence of 12 known exposures as defined in Ref. 3. This function f relates exposure to pixel value, where exposure is the product of radiance E and exposure time Δt. Consequently, if we define the pixel values across the calibration images as $Z_{ij} = f(E_i \Delta t_j)$ and an inverse log function $g = \ln f^{-1}$, then g can be estimated using least squares optimization over the set of equations $g(Z_{ij}) = \ln E_i + \ln \Delta t_j$, where i is a spatial index and j ranges over all exposures.[3] An advantage of this method is that it does not make assumptions about the shape of the curve. Yet though it provides a simple way to convert pixel value Z_{ij} (0-255) to radiance E_i, a parametric model is a more practical means for the opposite conversion. Consequently, we fit g^{-1} with an exponential of the form $Ae^{BE_i} + C$, which is useful for "re-exposing" images as discussed in Sec. 4. Several camera parameters such as gain, brightness, and white balance are fixed, isolating the shutter speed. We found it useful to generate different camera response curves for both outdoor and indoor lighting conditions. Since the RGB responses are

estimated independently, we also estimate red-green and red-blue ratios to preserve chromaticity between the pixel measurements and radiance values.[4]

We use a Point Grey Research Firefly MV camera to capture video with alternating exposures at 30 fps[*]. It is important to select exposures that can expand the dynamic range, while maintaining a sufficient overlap for registration. For a detailed automatic dual-exposure algorithm, the reader is referred to Ref. 2. Here, we use the built-in auto-exposure for several frames to initialize a center shutter speed, and simply define the two shutter speeds with respect to this middle ground. These shutter speeds are then used for the duration of the video.

4. MOTION ESTIMATION

The capture method described in Sec. 3 provides an input video with alternating short and long exposures. However in order to generate an HDR output, *both* exposures must be available at every time instant. This requires accurate motion estimation (ME) to determine pixel correspondences between frames. In addition, this ME problem has unique challenges due to the severe illumination change between frames and saturated pixels. The "HDR stitching" method in Ref. 2 used gradient-based optical flow, while the method presented here uses a block-based approach. In this section, we discuss the advantages and disadvantages of each method.

As in Ref. 2, information from both the previous and next frames is used to generate a motion-compensated frame. Taken as is, the change in exposure precludes motion estimation between adjacent frames because the brightness constancy assumption is violated. Therefore the first step is to boost the short exposure to match the long exposure. To do this we "re-expose" the pixel values in the short exposure Z_s to find pixel values in a simulated long exposure Z_l using

$$Z_l = g^{-1}(g(Z_s) - \ln \Delta t_s + \ln \Delta t_l), \tag{1}$$

where Δt_s and Δt_l are the short and long exposure times, and g^{-1} is modeled with an exponential curve as described in Sec. 3.

Even after boosting, a gradient-based optical flow technique is ill-suited for estimating the forward/backward flow fields directly using the current frame. The boosted short exposure will appear noisy and grainy, edges may be lost, and there still may be a slight variation in brightness due to inaccuracies in the camera response curve. A block-based approach improves robustness in these suboptimal conditions. As a consequence, Ref. 2 uses only bi-directional prediction for local motion estimation, meaning the displacements between frames used for ME (previous and next) are larger. Though there is work to address this, a limitation of gradient-based techniques is poor performance for fast moving objects with large displacements. This manifested itself in the results of Ref. 2 as ghosting and poor registration for fast moving objects such as hands. A block-based technique is better suited for displacements over a significantly large search range.

Block-based techniques do have a clear disadvantage when compared to gradient-based techniques, and that is the fidelity of the motion field. Only one motion vector is assigned to entire blocks, restricting the block to translations. Furthermore, moving features that are smaller than the block size may be incorrectly characterized. This limitation must be addressed for a block-based method to be feasible, and we will describe our efforts towards this in Sec. 4.3 and Sec. 5.2. Still, a block-based technique provides several other advantages. Color information can be easily incorporated into the matching metric. Importantly, block-based techniques can have low complexity and are typically used for video compression, making them suitable for real-time applications such as video conferencing.

4.1 Forward and Backward ME

Following frame boosting, the next step in our motion estimation approach is to calculate the forward and backward motion vectors for the current frame (I_t) with respect to the previous and next frames (I_{t-1} and I_{t+1}). We use the H.264 JM Reference software with Enhanced Predictive Zonal Search (EPZS), a 16×16 block size, and Sum of Absolute Differences (SAD) in both luma and chroma components.[5] This simplified motion search

---

[*]An unmodified Firefly MV camera cannot guarantee that changes to camera parameters such as exposure time will affect the next frame. We therefore capture at 60 fps, alternating the exposure every two frames, and downsample temporally to mitigate this limitation.

is fast and produces smooth motion fields. The two motion fields are then combined to form a single motion field by choosing the motion vector with minimum SAD over each block. Labels are stored to reference either the previous or next frame (or both if the forward and backward MVs are both zero). If the camera itself is moving (exhibiting global motion), an affine transform between each pair of frames can be estimated from block motion vectors using a least squares approach.

4.2 Bi-directional ME

Due to pixel saturation, it is unlikely that the forward/backward flow field will be valid for the entire frame. Nevertheless, valid regions are used to inform bi-directional motion estimation over all "saturated" blocks. A block is labeled as saturated when the number of pixels with luminance above (or below) a threshold is greater than 50% of the entire block. A motion search is then performed at each saturated block in the current frame by calculating the SAD between blocks in the previous and next frames. Given a block at location (x_i, y_i) in the current frame, $(x_1, y_1) = (x_i, y_i) + \mathrm{MV}_k$ and $(x_2, y_2) = (x_i, y_i) - \mathrm{MV}_k$ represent the locations of the candidate blocks in the previous and next frames. In this way, the hole filling problem is avoided.

In order to maintain smoothness w.r.t. the motion vectors of non-saturated regions, an additional cost term is added to the SAD of candidate motion vectors so that the total cost minimized is

$$\mathrm{cost} = \mathrm{SAD}(\mathrm{I}_{t-1}(x_1, y_1), \mathrm{I}_{t+1}(x_2, y_2)) + \lambda |\mathrm{MV}_{pred} \pm \mathrm{MV}_k|, \qquad (2)$$

where MV_{pred} is a predicted motion vector, and λ is a constant. To find MV_{pred}, we first determine which adjacent frame is most valid in the current 5×5 neighborhood of blocks by tallying the labels of non-saturated blocks. The predicted MV is then the median of "non-saturated" MVs for this reference frame, and the $\pm$ in Eq. (2) is determined by whether the reference frame is forward or backward. Finally, the missing block is filled using *only* the block from this reference frame. Before moving onto the next block, the filled block is classified as non-saturated so that motion vectors of blocks filled correctly during the forward/backward ME stage can potentially propagate inward to fill large saturated regions.

4.3 Motion Vector Refinement in Saturated Regions

As discussed earlier in Sec. 4, a disadvantage of block-based motion estimation is the appearance of artifacts such as discontinuities at block boundaries. If there is corresponding color information in the current frame, we can mitigate these errors using methods discussed in Sec. 5.2. In saturated regions, we can employ a video completion method similar to "motion inpainting", as described in Ref. 6. This method is a *pixel-level* refinement, which treats error-prone pixels as holes and assigns motion vectors under the assumption that pixels with similar motion have similar color. In Ref. 6, local motion is estimated using a gradient-based technique and candidate pixels are taken from a window of neighboring pixels. Since our method assigns a single motion vector to entire blocks, the candidate pixel set must also include the nearest pixels in neighboring *blocks*.

The first step here is to locate pixels in the saturated region that are likely erroneous. Using the camera response curve described in Sec. 3, it is possible to determine the minimum brightness in a short exposure that will saturate in the long exposure ($Z_{s*} = g^{-1}(g(Z_{max}) - \ln \Delta t_l + \ln \Delta t_s)$, as well as the maximum brightness in the long exposure that will saturate in the short exposure ($Z_{l*} = g^{-1}(g(Z_{min}) - \ln \Delta t_s + \ln \Delta t_l)$. If the current frame is a short exposure, we can then locate pixels in the saturated regions whose predicted values are less than the threshold Z_{s*} (vice versa for a long exposure).

Once these pixels are located, they are treated as missing and subsequently filled inward starting at the hole boundaries. The set of 16 candidate pixels $\mathbf{q}_k$ for a given missing pixel $\mathbf{p}$ is illustrated in Fig. 3. Each candidate pixel that is not out of the frame or also in a hole has an associated motion vector MV_k. We choose the MV_k that maximizes the weighting function

$$w(\mathbf{p}, \mathbf{q}_k) = r(\mathbf{p}, \mathbf{q}_k) c(\mathbf{p}, \mathbf{q}_k) = \frac{1}{||\mathbf{p} - \mathbf{q}_k||} \frac{1}{||\mathrm{I}_{t'}(\mathbf{p} + \mathrm{MV}_k) - \mathrm{I}_{t'}(\mathbf{q}_k + \mathrm{MV}_k)||}, \qquad (3)$$

where $r(\mathbf{p}, \mathbf{q}_k)$ is inversely proportional to the geometric distance between $\mathbf{p}$ and $\mathbf{q}_k$, and $c(\mathbf{p}, \mathbf{q}_k)$ is inversely proportional to the color pseudosimilarity. We use a color difference metric that perceptually weights the error
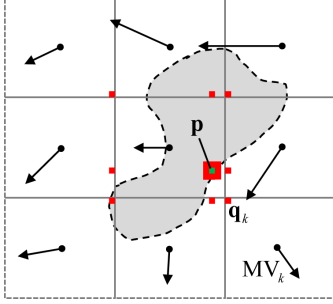
Figure 3. MV Refinement in Saturated Regions: The MV at pixel $\mathbf{p}$ is assigned as one of 16 candidate MVs for pixels $\mathbf{q}_k$ that maximizes a weighting function determined by geometric distance and color similarity.

in each color channel, rather than the l^2-norm.[7] Furthermore, as described in Ref. 6, $c(\mathbf{p}, \mathbf{q}_k)$ is referred to as *pseudo*similarity because it is calculated between pixels in $\mathrm{I}_{t'}$, which represents either the previous or next frame. It is possible that multiple candidates will have the same motion vector, so for these cases the weights are summed. Using this method, we can eliminate any pixels less than Z_{s*} or greater than Z_{l*}.

5. HDR RECONSTRUCTION

Following motion estimation, the current frame I_t and predicted frame $\widetilde{\mathrm{I}}_t$ provide a short and long exposure at each time instant. Therefore, given pixel values Z_{ij} and shutter times Δt_j, one can recover a high dynamic range *radiance map* using

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}, \tag{4}$$

where $w(Z_{ij})$ is a weighting function, i is the spatial index, and j is the frame index.[3] The choice of weighting function, discussed in detail in Sec. 5.1, has significant effects on the quality of the radiance map. Once the radiance map is calculated, it must be tone mapped back into displayable range. We use the method described in Ref. 8, which has global and local normalization and uses a dodging and burning technique to minimize halo effects.

5.1 Radiance Map Recovery

Despite the best efforts of an optical flow algorithm, there will always be registration errors in the predicted frame due to occlusions, etc. In Ref. 2, a tolerance for these errors is built into the weighting function during radiance map recovery. For saturated pixels in the current frame, they simply use the radiance calculated from the predicted frame. Yet for non-saturated pixels, they weight the predicted frame based upon the absolute difference between the radiances predicted by each. As such, if there is a large disparity between these radiances, the predicted frame will be weighted less. This weighting function, as well as a maximum disparity δ_{max} were determined empirically.

A main advantage of the method used in Ref. 2 is low complexity, so it is useful for possible real-time applications. One disadvantage is the difficulty in determining δ_{max} for a given frame. If the current frame is a short exposure, differences in the two radiances may be contributed to noise rather than mis-registration. Therefore setting δ_{max} too low could result in a noisy output, while setting it too high could allow registration errors to pass through. Furthermore, because the weighting function is not solely dependent on pixel values and often a single frame's radiance value is used, it does not guarantee that radiances will be temporally consistent. This may lead to flickering between frames, which was addressed by smoothing global luminance across a neighborhood of frames.[2]

We investigate a new radiance map recovery method to reduce mis-registration artifacts and flickering. We use a simple hat-function as defined in Ref. 3, which gives more weight to mid-range pixel values. This weighting function is only dependent on pixel value and ignores the difference between the two predicted radiances. Though

it is more lenient about passing poorly registered pixels or block artifacts onto the tone mapping stage, there are still two cases where *only* the radiance from the current frame is used: (1) the difference between the current frame radiance and predicted frame radiance is extremely large, and (2) the pixels disobey the monotonicity assumption, such as when a pixel is darker in the long exposure than it is in the short exposure. Any additional artifacts are then addressed *after* tone mapping.

5.2 Artifact Removal

Given our simple radiance weighting function, block artifacts and other poorly registered pixels will be passed onto the tone mapping process. Still, color information in the current frame can be utilized to locate and correct these errors. In Ref. 9, radial basis functions were used to learn a photometric mapping between a current frame and a tone mapped HDR composite created using a stereo camera running at different exposures. Registration errors were located using color similarity and the stereo matching cost. The effectiveness of this global mapping is limited because it does not take local spatial information into account.

Here, we instead use a cross-bilateral filter to filter the tone mapped HDR image using the edge information of the current frame. A bilateral filter preserves edges by combining pixels based on their geometric proximity as well as their photometric similarity.[10] Using a *cross*-bilateral filter effectively combines the color information of the HDR image with the edge information of the current frame. After filtering, we locate likely errors in registration and blocking artifacts by calculating the perceptual color difference and structural similarity (SSIM)[11] between the HDR image and its filtered counterpart. If the color difference is greater than a threshold or the structural similarity is less than a determined threshold (determined empirically), the pixel in the HDR frame is replaced using the corresponding pixel in the filtered image. In this way, features in the HDR image will be preserved if they are not completely lost in the filtered image. Again, this method is only used in non-saturated regions, as saturated regions will appear as flat colors in the filtered output.

6. RESULTS AND DISCUSSION

To test our method, we captured and processed several dynamic videos in varying environments[†]. First, Fig. 4 shows a portion of an output frame without and with motion vector refinement in saturated regions, as described in Sec. 4.3. Following bi-directional ME in the saturated blocks, small artifacts appear just above the top of the car due to the car's motion. Using the camera response curve, it is determined that these pixels are too dark to saturate in the long exposure. These pixels are then treated as holes, and their motion vectors are filled using the candidate MVs of neighboring blocks. This eliminates these artifacts as shown in Fig. 4 (c). This method works well to correct pixels that we can predict are erroneous (too bright or too dark), however future work needs to be done to make it effective for general block artifact removal in the saturated regions. We investigated using refinement over all pixels near block boundaries in saturated regions and obtained mixed results.

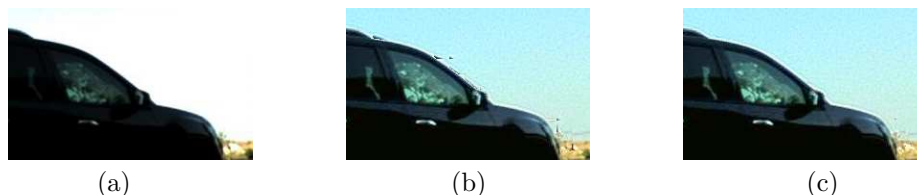| (a) | (b) | (c) |

Figure 4. (a) Current frame (long exposure) (b) HDR output without MV refinement in saturated regions (c) HDR output after MV refinement

Fig. 5 illustrates our artifact removal method in non-saturated regions utilizing a cross-bilateral filter. In a video conferencing scenario, fast movements such as blinking may not be captured smoothly using block motion vectors, resulting in the blockiness of Fig. 5 (a). Fig. 5 (b) shows the output of the cross-bilateral filter, which filters the color information of the HDR image using the edge information of the current frame. The block artifacts are successfully removed, yet much of the high frequency information is lost as well. By thresholding color and

---

[†]For sample videos, please visit http://vivonets.ece.ucsb.edu/HDR.html.

structural similarity between Fig. 5 (a) and (b), Fig. 5 (c) provides an output that removes mis-registration artifacts while retaining detail.

(a) (b) (c)

Figure 5. (a) HDR image before artifact removal (note the blockiness over the eyes) (b) The HDR image filtered using a cross-bilateral filter and the edges of the current frame (c) HDR image after artifact removal

We found that the cross-bilateral filtering method works very well for medium and larger sized objects in the frame. Results for small objects are shown in Fig. 6. Here, the left image represents the input to the filtering stage and the right image represents the output. Block artifacts are again successfully eliminated, however there is some loss of detail and color fading due to spatial smoothing. This can be mitigated by adjusting the size of the spatial filter, however it must be large enough to remove any mis-registrations. One way to address this is by only using the filtered output in areas with motion. In this way, the high frequency detail of static objects and backgrounds is maintained.

Figure 6. These sets of images show the effect of the cross-bilateral filter on very small objects. Blocking is removed, however there is some color fading and detail lost. (Left) Unfiltered HDR image (Center) cross-bilateral filter output (right) HDR image after artifact removal

Sample frames from two test videos are shown in Fig. 7. In the first scene, the camera sits in a shadow as two cars move through a roundabout. The yellow car is also in the shadow, while the white car is in direct sunlight. Using traditional auto-exposure, it is impossible to correctly expose both cars simultaneously. Due to pixel saturation, this scene presents several motion estimation challenges. The current frame, shown in the top-left corner, is a short exposure. It is clear that the entire bottom half of the yellow car has saturated to black. As such, this region cannot be successfully matched using an adjacent frame, so bi-directional ME as described in Sec. 4.2 is employed. This task is further complicated because a significant portion of the corresponding region in the adjacent frames is also saturated. However, as shown in the HDR output frame in the bottom-right corner, our method is able to correctly predict the motion of the saturated region using the motion vectors of the top of the car. This illustrates the importance of high quality motion vectors in non-saturated regions obtained during the forward/backward ME stage. The bottom-left image shows the current frame re-exposed to an exposure that is the geometric mean of the short and long exposures, simulating what an auto-exposure algorithm would capture. Both the trees and the yellow car are brighter and more vibrant in the HDR output, while maintaining a blue sky.

The second set of images in Fig. 7 shows an outdoor video conferencing scenario. As video conferencing for mobile devices becomes more widespread, high dynamic range video will become a crucial requirement. Saturated pixels on the user's face are analogous to audio clipping in a phone conversation, severely degrading the quality of experience. As shown in the simulated auto-exposure in the bottom-left, sunlight has caused portions of the right side of the user's face and much of the background to saturate. The color information in these regions is retained in the HDR output (bottom-right), while maintaining adequate brightness over the user's entire face.

Figure 7. Sample frames: (Top-Left) Short Exposure (Top-Right) Long Exposure (Bottom-Left) Simulated auto-exposure output using the geometric mean of the short and long exposures (Bottom-Right) High Dynamic Range Output

7. CONCLUSIONS

We have outlined a new method for the recovery of high dynamic range video from a sequence of alternating short and long exposures. Block-based motion estimation is used to obtain forward/backward motion estimates with respect to the current frame. A large search range and smaller displacements between adjacent frames help to accurately capture fast moving objects. In addition, our novel motion estimation scheme uses bi-directional block-based ME and pixel-level refinement over saturated regions. Following radiance map recovery and tone mapping, the HDR output is compared to a filtered version through the use of a cross-bilateral filter. In this way, mis-registrations and block artifacts are removed while retaining high frequency detail in the output image. This method shows promising results for dynamic scenes in various environments, promoting the creation of high dynamic range video with inexpensive cameras.

REFERENCES

[1] Reinhard, E., Ward, G., Pattanaik, S., and Debevec, P., [*High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*], Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005).

[2] Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R., "High dynamic range video," in [*ACM SIG-GRAPH*], 319–325 (2003).

[3] Debevec, P. E. and Malik, J., "Recovering high dynamic range radiance maps from photographs," in [*SIG-GRAPH '97*], 369–378, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1997).

[4] Mitsunaga, T. and Nayar, S. K., "Radiometric self calibration," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1**, 1374 (1999).

[5] "H.264/AVC JM reference software." http://iphome.hhi.de/suehring/tml/ (2008).

[6] Matsushita, Y., Ofek, E., Ge, W., Tang, X., and Shum, H.-Y., "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(7), 1150–1163 (2006).

[7] Riemersma, T., "Colour metric." http://www.compuphase.com/ cmetric.htm (Dec. 2008).

[8] Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J., "Photographic tone reproduction for digital images," *ACM Transactions on Graphics* **21**(3), 267–276 (2002).

[9] Mangiat, S. and Gibson, J., "Automatic scene relighting for video conferencing," in [*IEEE International Conference on Image Processing (ICIP)*], 2781 –2784 (nov. 2009).

[10] Paris, S. and Durand, F., "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vision* **81**(1), 24–52 (2009).

[11] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**, 600 –612 (april 2004).