High Dynamic Range Video

Sing Bing Kang Matthew Uyttendaele Simon Winder Richard Szeliski

Interactive Visual Media Group, Microsoft Research, Redmond, WA

Figure 1: High dynamic range video of a driving scene. *Top row: Input video with alternating short and long exposures. Bottom row: High dynamic range video (tonemapped).*

Abstract

Typical video footage captured using an off-the-shelf camcorder suffers from limited dynamic range. This paper describes our approach to generate high dynamic range (HDR) video from an image sequence of a dynamic scene captured while rapidly varying the exposure of each frame. Our approach consists of three parts: automatic exposure control during capture, HDR stitching across neighboring frames, and tonemapping for viewing. HDR stitching requires accurately registering neighboring frames and choosing appropriate pixels for computing the radiance map. We show examples for a variety of dynamic scenes. We also show how we can compensate for scene and camera movement when creating an HDR still from a series of bracketed still photographs.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—display algorithms; I.4.1 [Image Processing and Computer Vision]: Enhancement—Digitization and image capture.

Keywords: Image processing, video processing, high dynamic range, tonemapping.

1 Introduction

The real world has a lot more brightness variation than can be captured by the sensors available in most cameras today. The radiance of a single scene may contain four orders of magnitude from shadows to fully lit regions. Typical CCD or CMOS sensors only capture about 256-1024 levels. (The non-linear allocation of levels in a gamma curve can improve this slightly.)

The limited dynamic range of cameras has inspired many solutions in recent years. One method of obtaining a full radiance map is to take multiple images at different exposures and to combine these to create a High Dynamic Range (HDR) map of the scene [Mann and Picard 1995; Debevec and Malik 1997; Mitsunaga and Nayar 1999; Tsin et al. 2001; Mann et al. 2002]. Because these techniques require multiple input images, there is potential for motion between the inputs due to either dynamic elements in the scene or a moving (i.e., handheld) camera. Mitsunaga and Nayar [1999] address this problem to a limited extent by fitting a global motion model to the inputs. Mann *et al.* [2002] register differently exposed frames using homographies, which can compensate for larger camera rotations. Bogoni [2000] uses affine motion followed by per-pixel flow to register different exposures, but details of frame registration are not given. In a related approach, Uyttendaele *et al.* [2001] use a block-based exposure adjustment technique to remove exposure artifacts in image mosaics. However, they do not compute HDR images.

The need to fuse images taken at different times can be removed using multiple image detectors, novel sensors (e.g., National LM9628 sensor, IMS Chips HDRC sensors, Silicon Vision Products, SMaL Camera, Pixim) or spatially varying pixel exposures [Mitsunaga and Nayar 2000]. However, our focus in this paper is on what can be achieved using widely available conventional (low dynamic range) image sensors and video cameras.

Once a high dynamic range image has been computed, it can then be rendered to a display. Since typical displays are only able

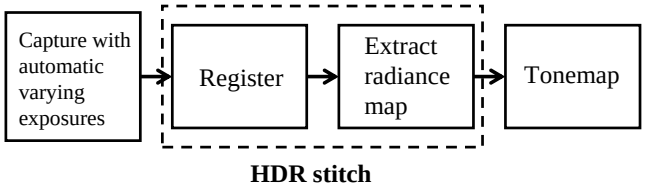

Figure 2: The processing stages involved in producing an HDR video.

to yield about two orders of magnitude, a contrast reduction must be performed on the HDR image. This tonemapping problem has recently been explored by a number of researchers [Durand and Dorsey 2002; Fattal et al. 2002; Reinhard et al. 2002].

Our work addresses the problem of generating HDR maps using multiple exposures in the presence of motion between the exposures. This enables us to generate HDR video sequences as well as HDR still images of moving scenes. The result of applying our approach to a driving video can be seen in Figure 1. Generating an HDR video consists of automatically determining the temporal exposure bracketing during capture, motion-compensating information between neighboring images, and tonemapping for viewing (Figure 2). This produces an HDR video that has the same frame rate as the variable exposure input sequence. To generate HDR still images, we use a similar HDR stitching technique.

Our video capture solution differs from previous efforts in that it employs a simple reprogramming of the auto-gain mechanism in a camera. This allows us to use the inexpensive and high resolution sensors available today, unlike novel sensor designs which are not yet widely available and may suffer from a lack of resolution. In Section 2, we present an auto-gain algorithm that intelligently varies the exposure from frame to frame in order to capture different parts of a scene's radiance map. When compared to the spatially varying pixel exposures approach [Mitsunaga and Nayar 2000], our approach can be viewed as sub-sampling along a temporal rather than a spatial dimension.

The acquisition process is followed by an offline process that motion-compensates the captured video and estimates a full radiance map at each frame time (Section 3). This operation, which we call *HDR stitching*, establishes dense correspondences between images in order to combine pixels at different exposures. In combining pixels, we introduce a consistency check between the motion-compensated intermediate results. This makes the system more robust to errors in flow estimation.

Before we can view the HDR video, it must be tonemapped. Applying one of the existing algorithms on a frame-by-frame basis is not sufficient, as this can lead to visible temporal inconsistencies in the mapping. In order to compensate for this, we extend one of the existing tonemapping techniques to operate on HDR video using statistics from temporally neighboring frames in order to produce tonemapped images that vary smoothly in time (Section 4).

2 Real-time exposure control

The auto gain control (AGC) of a typical video camera measures the brightness of the scene and computes an appropriate exposure. Most scenes have a greater dynamic range than can be captured by the camera's 8-bit per pixel sensor. Because of this, some pixels will be saturated and some will be under-exposed. In order to capture a greater dynamic range, we have developed a system that varies exposure settings on a per-frame basis. The basic idea is to sequence the settings between different values that appropriately expose dark and bright regions of the scene in turn. A post-processing step (Section 3) then combines these differently exposed frames.

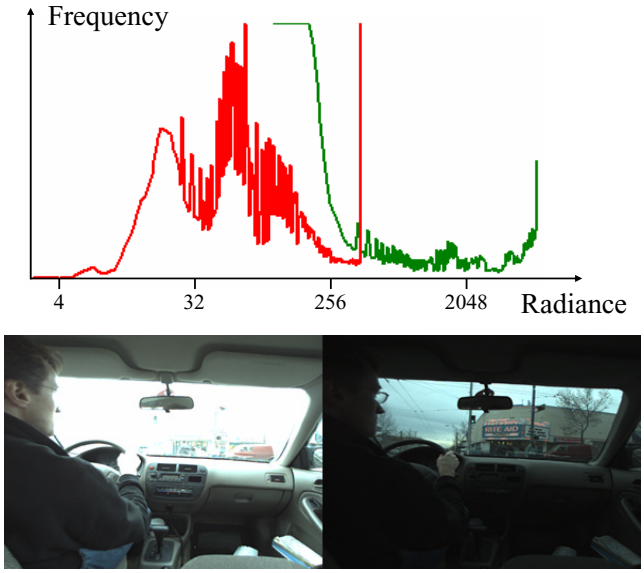

Figure 3: Two input exposures from the driving video. *The radiance histogram is shown on top. The red graph goes with the long exposure frame (bottom left), while the green graph goes with the short exposure frame (bottom right). Notice that the combination of these graphs spans a radiance range greater than a single exposure can capture.*

Our system is similar to auto-bracketing found in many still picture cameras today. When auto-bracketing a scene, the camera determines the correct exposure using the current metering mode, and then additionally captures two more exposures at fixed multiples of the original exposure, for example two f-stops up and down. We take a similar approach. However, instead of using a fixed multiple, we compute a more ideal set of exposures for the current scene. This allows for dynamic range expansion while keeping as many pixels as possible reasonably exposed in all images, thereby facilitating good motion analysis.

For our experiments, we use a 15 frame per second LadyBug Firewire camera from Pt. Grey Research which has a programmable control unit. The firmware was upgraded with a bank of four shutter (CCD integration time) and gain (ADC gain) registers. The camera does a round-robin through the bank using a different register set at each frame time. Additionally, the camera tags every frame with the current settings so that these can be used during the radiance map computation. A real-time AGC algorithm (running on a PC tethered to the camera) determines the next group of four settings.

In our current implementation, the exposure settings alternate between two different values and are continuously updated to reflect scene changes. The appropriate exposures are automatically determined from scene statistics computed on a sub-sampled frame. The ratio between exposures can vary from 1, if a single setting is adequate to capture the scene's intensities, to a user specified maximum. In practice, we have found that a maximum of 16 gives good results, since this gives a dynamic range expansion of 4 bits while still allowing for motion analysis to be done between the exposures. Figure 3 shows successive frames captured by the camera along with their corresponding histograms in radiance space. In this case, the exposure ratio is close to the maximum of 16.

3 HDR stitching

Since frames are captured with temporally varying exposures, generating an HDR frame at any given time requires the transfer of pixel color information from neighboring frames. This, in turn, requires that the pixel correspondences across different frames be highly accurate. We refer to the process of transferring color information from neighboring frames and extracting the HDR image as HDR stitching.

The source video contains alternating long and short exposure frames. The first step in HDR stitching is to generate both a long and a short exposure frame at every instant so that a radiance map can be computed from the pair. This requires that we synthesize the missing exposures using a warping process (Section 3.1), which in turn requires highly accurate motion estimation (Sections 3.2 and 3.3). Once the interpolated frames have been synthesized, we selectively blend pixels from these images to compute a high dynamic range radiance image (Section 3.4).

3.1 Image warping for frame interpolation

Our HDR stitching process generates four intermediate warped frames: two frames unidirectionally warped from the previous and next frames (S_U^- and S_U^+ in Figure 4), and two bidirectionally warped frames S_B^- and S_B^+ computed by comparing the previous and next frames directly. The unidirectionally warped frames are used when there is a good correspondence between the current and previous/next frames, while the bidirectionally warped images are used in regions where the current frame is too dark or too saturated to reliably establish correspondences.

Before we find pixel correspondence between frames with different exposures, we first *boost* the intensity of the frame with the shorter exposure. To find the amount of intensity boosting required, we use the camera response function to convert the shorter exposed image to a radiance map, followed by conversion to a virtual image having the longer exposure using the inverse response. This virtual image should match the pixel values (modulo discretization and noise) of the longer exposed image with which it is being registered.

Let us assume that the current frame L is captured at a long exposure with adjacent frames S^- and S^+ captured at short exposures (Figure 4). We first use a unidirectional version of the motion estimation algorithm described in Section 3.2 to establish dense pixel correspondences between a boosted version of the previous/next image and the current frame (green portions of Figure 4). (In the converse LSL situation, the central frame is boosted first.) The computed flow fields are then applied directly to the previous and next frames to synthesize the unidirectionally warped previous and next images S_U^- and S_U^+, respectively.

The process for computing the bidirectionally interpolated frames is more complex (black part of Figure 4). First, if necessary, we boost the previous or next image in case they are differently exposed. Next, we match these two images using the bidirectional motion estimation algorithm described below (Section 3.2). The resulting blended (frame interpolated) image $\tilde{S}_B$ could be used to compute radiance values in saturated or dark regions of the current frame.

Unfortunately, because of camera jitter and acceleration in object movement, there may still be some mis-registration between $\tilde{S}_B$ and the current frame L. For this reason, we register a boosted version of $\tilde{S}_B$ (i.e., $\tilde{L}_B$) with the current frame using the hierarchical homography algorithm described in Section 3.3, which is more conservative in its motion estimates. The resulting flow field is then added to the original flow field to produce a refined flow to be applied to the neighboring frames. We call the resulting final outputs the bidirectionally warped (or interpolated) images S_B^- and S_B^+.

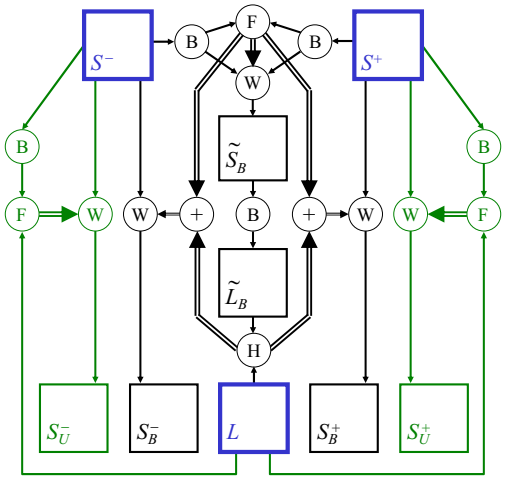

Figure 4: Image warping for HDR computation. *This diagram shows the case where the sequence is SLS, but the algorithm for LSL is similar (S=short exposure image, L=long exposure image). The dark double lines indicate flow (motion) estimates, while the lighter single lines are computed images. The B, F, H, and W circles correspond to boosting (intensity compensation), flow estimation, hierarchical homography, and image warping, respectively. The original (captured) images are shown in blue, unidirectionally interpolated images in green, and bidirectionally interpolated images in black. The meaning of the image labels is given in the text.*

3.2 Motion estimation

Our motion estimation algorithm consists of two stages. First, we globally register S^- and S^+ by estimating an affine transform that maps one onto the other. We then use gradient-based optical flow to compute a dense motion field that forms a local correction to the global transform.

Rather than computing forward or backward flow fields between the previous and next frames separately, we compute one bidirectional field at the current frame time. This allows us to avoid the hole-filling problems of forward-warping when generating each interpolated frame. At each pixel in the output frame ($\tilde{S}_B$), we obtain one composite vector that points into both previous and next frames. These vectors are each the sum of affine and local components. The affine component is derived from the global warping parameters, rescaled to warp either from S^- to the output frame or from S^+ to the output. (Note that the resulting two motion fields are not necessarily symmetric.) The local component is generated by our symmetrical optical flow algorithm.

For local motion estimation, we use a variant of the Lucas and Kanade [1981] technique in a Laplacian pyramid framework [Bergen et al. 1992]. (Note that Bergen et al. [1992] has a nice introduction to the topic of optical flow.) We add to this a number of enhancements to handle degenerate flow cases. Rather than simply warping one source image progressively towards the other one at each iteration, we warp both source images towards the intermediate frame and estimate the residual flow vectors between these two warped images. As the residuals are accumulated down the pyramid, they give rise to a symmetric flow field centered at the intermediate frame. We augment this technique by including the global affine flow during the warping so the accumulated residuals are always represented in terms of a local correction to the asymmetric global (affine) flow.

For the unidirectional case, we use a modification of the above motion estimation procedure (including the affine stage) to generate the single-warped frames S_U^- and S_U^+ as described in Section 3.1. In this case, the procedure directly matches the current frame L to its

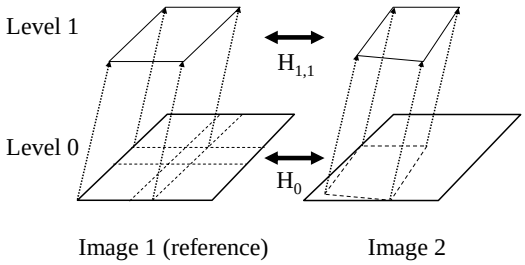

Figure 5: Hierarchical homography computation. *Note that only the first two levels and one of the quadrants at Level 1 are shown.*

boosted neighbor. The resulting flow field is unidirectional because only the neighboring frame is re-warped at each iteration step.

3.3 Hierarchical homography

Because of camera jitter and possible accelerations in object motion, the bidirectionally interpolated image may not be perfectly registered with the central (current) image. We therefore use an alternative image registration method to refine the registration between the boosted interpolated frame ($\tilde{L}_B$) and the current frame (L). Constraining the flow is desirable at this point as it reduces the possibility of erroneous mapping in unreliable regions of saturated and low-contrast pixels. (While computing flow with image masks that single out saturated or dark pixels is a possibility, we would then have to set thresholds on what we deem to be saturated or dark.)

The idea of hierarchical homography is shown in Figure 5, which is simplified to illustrate two levels and one quadrant only. At the highest resolution, full frame registration is performed to find the best 2D perspective transform (i.e., homography) between the two input images, producing homography H_0. The current image (Image 1) is then broken up into overlapping quadrants shown partially in dotted lines. The global motion for each quadrant is inherited from its parent. If there is insufficient intensity variation within the quadrant (we set this threshold at 10 gray levels), it is left alone. Otherwise its global motion is refined by performing full image registration between that quadrant from the current image and the appropriately sampled counterpart from the second image. The boundary of the sub-image from the second image is computed based on H_0 in this illustration. In the example shown in Figure 5, this refined transform between the sub-image pair is $H_{1,1}$.

This operation is repeated for all the levels (two in our case) and all the quadrants. The resulting full image flow is then computed using the local homographies. At and near the boundaries of each quadrant, their flows are weight-averaged to reduce flow discontinuities. This flow is then used to correct the bidirectional flow as described in Section 3.1.

3.4 Radiance map recovery

In this section, we describe the process of combining the input images with their warped neighbors to produce a radiance map. Several techniques have been proposed to do this [Mann and Picard 1995; Debevec and Malik 1997; Mitsunaga and Nayar 1999; Tsin et al. 2001]. In each of these techniques, the input images are converted to radiance images using the known exposure value and a computed response function. The final radiance value at a pixel is then computed as the weighted average of the corresponding pixels in these radiance images. In our system, we compute the response function f_R of our camera (and the corresponding weighting function f_W) using the technique of Mitsunaga and Nayar [1999].

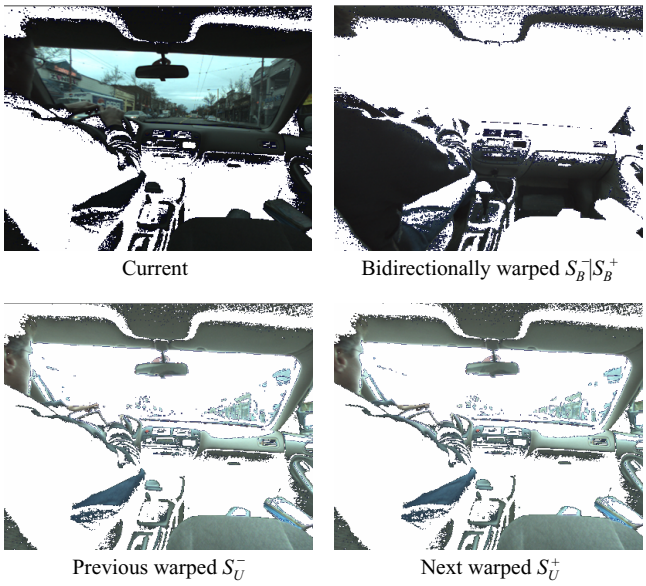

Figure 6: Radiance map computation. *Contributing regions from the current and warped frames are shown in color. Pixels that fail the consistency check (and hence whose modulation weight $f_M = 0$) are shown in white.*

Existing approaches to radiance map computation assume perfectly registered input images. Due to the possibility of misregistrations in the first step of HDR stitching, we relax this requirement, which makes our system more tolerant to errors in pixel registration. The following steps are taken for the case where the input image is a long exposure and the adjacent frames are short exposures.

1. Convert L, S_U^-, S_U^+, S_B^-, and S_B^+ to radiance images using the response function and their respective exposure values. These radiance images are denoted by $\hat{L}$, $\hat{S}_U^-$, $\hat{S}_U^+$, $\hat{S}_B^-$, and $\hat{S}_B^+$, respectively.

2. Identify pixels in the current image L that are above a maximum value as being saturated. These pixels are assumed to produce poor registration with adjacent frames. As a result, the final radiance map is filled in with bidirectionally interpolated pixels from $\hat{S}_B^-$ and $\hat{S}_B^+$. In order to avoid possible blurring, we use pixels from only the previous frame $\hat{S}_B^-$, unless these are inconsistent with the current frame (i.e., they do not map to the saturated range), in which case we use pixels from $\hat{S}_B^+$.

3. In other regions, compute the radiance map using a weighted blend

$$R = \frac{f_{WM}(p,p^-)p^- + f_{WM}(p,p^+)p^+ + f_W(p)p}{f_{WM}(p,p^-) + f_{WM}(p,p^+) + f_W(p)}, \quad (1)$$

where the pixels p, p^-, and p^+ come from the current, previous, and next radiance images $\hat{L}$, $\hat{S}_U^-$, and $\hat{S}_U^+$, respectively.

The weighting function $f_{WM}(p,q) = f_M(|p-q|)f_W(p_w)$ is the intensity-based weight function f_W [Mitsunaga and Nayar 1999] modulated by a plausibility map f_M, which is simply a Hermite cubic,

$$f_M(\delta) = \begin{cases} 2\left(\frac{\delta}{\delta_{max}}\right)^3 - 3\left(\frac{\delta}{\delta_{max}}\right)^2 + 1 & \text{if } \delta < \delta_{max} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Figure 7: Representative stills from two HDR video examples: *Left: Fish market scene, Right: Harbor scene. For each scene, the top left quadrant is a short exposure frame, and the top right quadrant is a long exposure frame. The bottom left quadrant shows what the frame would look like for an exposure equal to the geometric mean of the short and long exposures.*

that downplays warped radiance values that are too different from the corresponding current radiance value. δ_{max} is a user-specified parameter, which we set to a radiance value equivalent to 16 intensity levels in the longest exposed input image. While currently f_M was chosen heuristically, a more principled approach would be to use the noise statistics of the camera.

Computing the radiance values for the case where the current image is a short exposure follows the same reasoning except for step 2. In this step, dark pixels are discarded instead of saturated ones.

Figure 6 illustrates our radiance map recovery algorithm. The current frame is taken with a short exposure. We show only the pixels that contribute to the radiance map.

4 Temporal tonemapping

Tone mapping is used to convert floating-point radiance maps into an 8-bit representation suitable for rendering. This process must reduce the dynamic range of each frame while also maintaining a good contrast level for both brightly and darkly illuminated regions. In addition, the transforms between adjacent frames must be consistent to prevent temporal artifacts such as flickering. (Note that Pattanaik *et al.* [2000] go even further by modeling the temporal response of the human visual system in their (global) temporal tonemapping algorithm.)

Since we do not expect a rapid change in scene intensity, we apply a much simpler technique. We make use of the tonemapper presented by Reinhard et al. [2002], which is based on the photographic technique of dodging and burning. In our first stage, the radiance image is converted to CIE space and the chromaticity coordinates are recovered. The luminance image is then processed to compress the dynamic range. Finally, the chrominance is re-inserted to give the final byte-range RGB image.

Our temporal tonemapper consists of global and local stages:

For the global mapping, we compute the average and maximum luminances, which control the transfer function that provides a good initial luminance mapping. The log-average luminance is given by

$$F_w = \exp\left(\frac{1}{N}\sum_{x,y,i}\log(\epsilon + F_i(x,y))\right), \qquad (3)$$

where ϵ is a small value (10^{-6}), N is the total number of pixels, and F_i is the causal temporal neighborhood consisting of frames at times $k-1$ and k. Using a set of frames to control the global mapping helps to prevent flicker in the tonemapped sequence.

The tonemapper also contains a local normalization, which is computed using a scale-space-based edge-preserving filter. This is described in detail in [Reinhard et al. 2002]. We did not apply temporal coherence on the local normalization process.

5 Results

In this section, we show results for three different dynamic scenes: a fish market, a harbor, and a drive along a busy street. We also describe an example involving static images of a sunrise scene taken with a handheld moving camera. The processing time for each HDR video frame (768×1024 pixels) is 10 seconds, with registration and radiance mapping taking 8 seconds and tonemapping taking 2 seconds on a 2GHz Pentium 4 machine.

Figure 7 shows representative stills from the fish market and harbor scenes. For each scene, the top left quadrant is a short exposure frame, and the top right quadrant is a long exposure frame. The bottom left quadrant shows what the frame would look like for an exposure equal to the geometric mean of the short and long exposures. The image in the bottom right quadrant is generated using our method.

Fish market scene. A snapshot from the fish market scene can be seen on the left of Figure 7. While the single exposure version looks reasonable, there is some saturation (especially in the middle and upper right corner) as well as low contrast areas. In the frame generated using our HDR approach, good details can be seen almost everywhere (except for the base of the counter, where even the long exposure frame shows little detail).

Harbor scene. This video was captured inside an office overlooking a harbor. In the video, the ferry can be seen moving outside the window while some human activity can be observed inside the office. As shown in Figure 7, the single exposure has both significantly large saturated and low contrast regions. On the other hand, in the frame generated using our approach, the ferry and water can be clearly seen. More details can also be seen inside the office.

Driving scene. The results for the driving scene can be seen in Figure 1. In this example, the driver drives through a busy street at about 25 mph. This was a particularly difficult scene because occasionally there is large frame to frame displacement due to the fast motion of the driver's hand. Our optical flow algorithm sometimes fails for such large motions, but this problem could be alleviated using a higher frame-rate camera. As mentioned earlier, the capture rate of our camera is currently 15 frames per second and similar exposures are therefore sampled at 7.5 frames per second.

6 HDR image from image stills

The ideas used in creating high dynamic range video can also be applied to image stills, especially when there is camera or scene motion. Figure 8 shows the steps involved in transforming such a set of image stills. The input image sequence is first sorted according to exposure, and the image with the most number of "valid" pixels is chosen as the reference ("current") frame. A pixel is considered "valid" if it is not saturated or of low contrast; in our implementation, each "valid" RGB value must be between (and excluding) 16 and 255.

Each neighboring pair is then registered in the direction of the reference. Neighboring pairs are chosen because they exhibit less visual change, which results in more robust registration. As before, the shorter exposure image is boosted to match the longer exposure neighbor prior to registration, and registration involves global motion estimation followed by local per-pixel flow.

Once the flow fields have been computed, they are appropriately concatenated to permit every frame to be registered (i.e., stabilized) with the reference image. The stabilized images are then used to recover the radiance map in a manner similar to that described in Section 3.4. The difference is that each stabilized image is now treated as a unidirectionally-warped image (i.e., either S_U^- or S_U^+).

We use the sunrise scene (Figure 9a) as an illustration. Here there is both camera motion and cloud motion relative to the ground. If we were to just perform global registration (2D perspective or homography), we would obtain the result shown in Figure 9b,c, with (c) being the magnified version of the middle right part of (b). If we apply global registration followed by local registration, we obtain a significantly better result as shown in Figure 9d,e. Notice the crisper appearance of the tree branches. We used the standard metadata information (called EXIF tags) stored in each of the stills to automate our radiance map computation. For the sunrise example (five stills, each still being 1024×768), registration (global and local) and radiance mapping took 30 seconds while tonemapping took 2 seconds.

7 Discussion

Looking at the results, you can see that our technique produces videos with increased dynamic range while handling reasonably large amounts of visual motion. For very fast motions, however, given our current low sampling rate of 15 frames per seconds, our technique sometimes produces artifacts. Using a camera with a faster capture rate would certainly help, as would improvements in the image registration algorithms. In particular, the ability to deal with occlusions (perhaps by extracting object boundaries and constructing a layered motion model) would be useful.

In addition to scenes with rapid motion and significant occlusions, scenes with significant amounts of non-rigid effects are also very difficult for our technique to handle. These non-rigid effects include inter-reflections and specularities on non-Lambertian surfaces, as well as complicated translucent objects. Such effects may be explained using multiple motion models (e.g., [Tsin et al. 2003]), which we do not currently use in our technique. The easiest scenes for our technique are those with mostly Lambertian, highly textured surfaces with little occlusion and small motion.

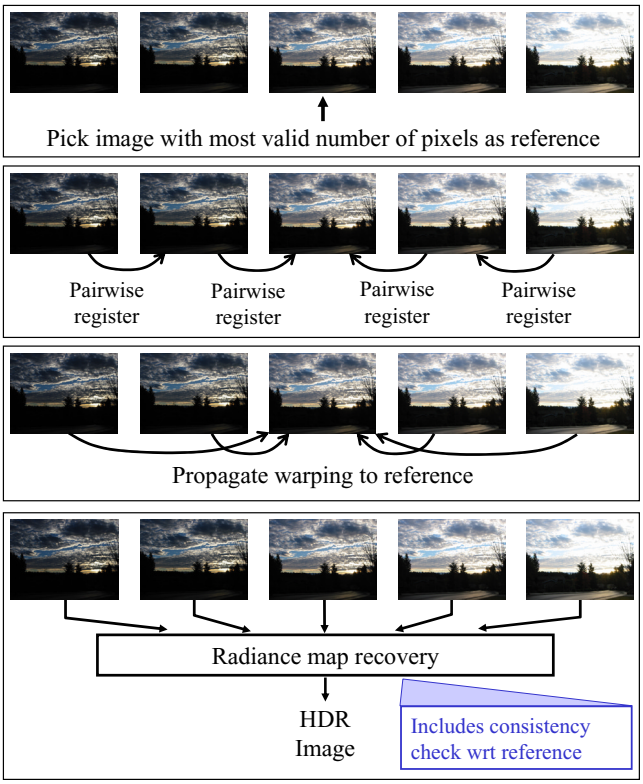

Figure 8: Steps in creating HDR image from image stills. *From top to bottom: select reference image, pairwise register (global followed by local registration), propagate flow fields, and radiance map recovery with consistency check.*

For scenes with very large brightness ranges, such as a dark indoor scene looking out on a bright day, using just two exposures may not be adequate. Increasing the exposure gap between successive frames will capture the dynamic range better, but will make the image registration step more brittle and will lead to increased image noise in the mid-tones. Using more than two exposures is another option we considered, but similar exposures (where registration has the best chance of success) are then temporally farther apart, again leading to potential registration and interpolation problems. Having a faster capture rate would allow a larger set of consecutive different exposures to be taken and merged.

Currently, our weighting and modulation functions are based on the work of [Mitsunaga and Nayar 1999; Tsin et al. 2001]. We plan to use an integrating sphere (whose interior surface is totally Lambertian) to more accurately characterize both the response curve and the noise characteristics of the camera.

8 Conclusions

In this paper, we have presented a technique for creating high dynamic range video from a sequence of alternating light and dark exposures. The first part of our system is a novel gain control algorithm that selects the best pair of exposures as a function of the pixel brightness distribution. The central component of our approach is the HDR stitching process, which includes global and local registration steps to compensate for pixel motion, as well as an algorithm to select the most trustworthy pixels for radiance map computation.

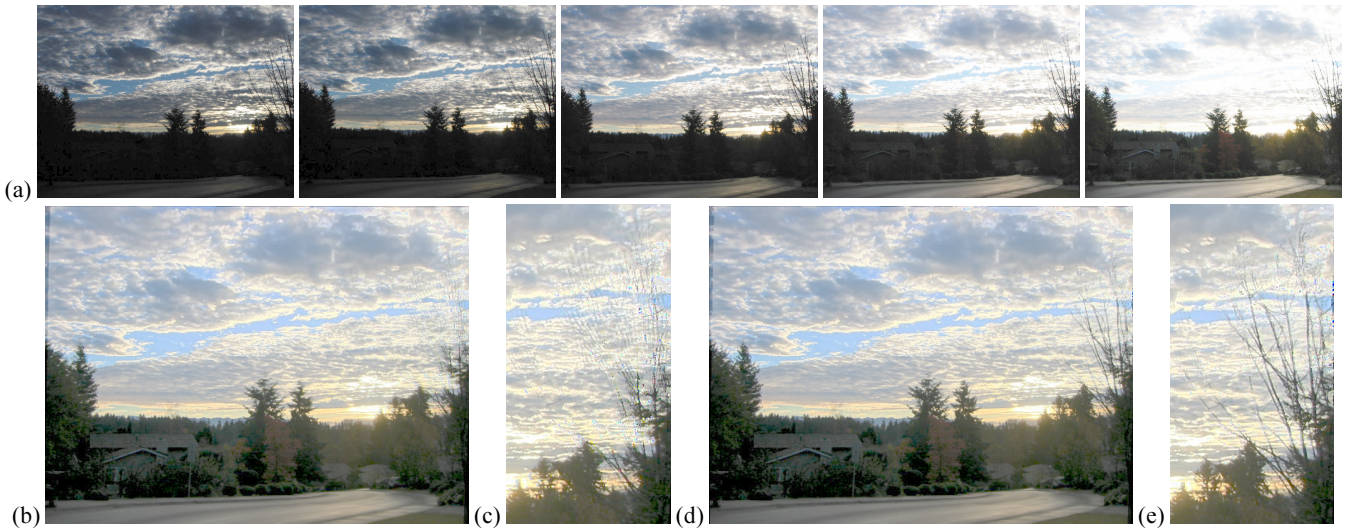

Figure 9: HDR image from stills: *sunrise example. (a) The five input images (each* 1024×768*), (b,c) Result of using only global (2D perspective) registration, (d,e) Result of using both global and local registration.*

The third part is a tone-mapping algorithm adapted to produce temporally coherent results. The resulting system can be used to produce both high dynamic range videos and still images taken with handheld cameras and scene motion.

References

Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision (ECCV'92)*, 237–252.

Bogoni, L. 2000. Extending dynamic range of monochrome and color images through fusion. In *International Conference on Pattern Recognition*, vol. 3, 7–12.

Debevec, P., and Malik, J. 1997. Recovering high dynamic range radiance maps from photographs. *Proceedings of SIGGRAPH 97* (August), 369–378.

Durand, F., and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (TOG) 21*, 3, 257–266.

Fattal, R., Lischinski, D., and Werman, M. 2002. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (TOG) 21*, 3, 249–256.

Lucas, B. D., and Kanade, T. 1981. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, 674–679.

Mann, S., and Picard, R. W. 1995. On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *IS&T's 48th Annual Conference*, Society for Imaging Science and Technology, Washington, D. C., 422–428.

Mann, S., Manders, C., and Fung, J. 2002. Painting with looks: Photographic images from video using quantimetric processing. In *ACM Multimedia*, 117–126.

Mitsunaga, T., and Nayar, S. K. 1999. Radiometric self calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, vol. 1, 374–380.

Mitsunaga, T., and Nayar, S. K. 2000. High dynamic range imaging: Spatially varying pixel exposures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, vol. 1, 472–479.

Pattanaik, S. N., Tumblin, J. E., Yee, H., and Greenberg, D. P. 2000. Time-dependent visual adaptation for realistic image display. *Proceedings of SIGGRAPH 2000* (July), 47–54.

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

Tsin, Y., Ramesh, V., and Kanade, T. 2001. Statistical calibration of CCD imaging process. In *Eighth International Conference on Computer Vision (ICCV 2001)*, vol. I, 480–487.

Tsin, Y., Kang, S. B., and Szeliski, R. 2003. Stereo matching with reflections and translucency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)* (to appear).

Uyttendaele, M., Eden, A., and Szeliski, R. 2001. Eliminating ghosting and exposure artifacts in image mosaics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. II, 509–516.