reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 1

HDR Image Capture

04

HDR images may be captured from real scenes or rendered using 3D computer graphics (CG) techniques such as radiosity and raytracing. A few modern graphics cards are even capable of generating HDR images directly. The larger topic of CG rendering is well covered in other textbooks [24,37,58,117,144]. In this chapter, the focus is on practical methods for capturing high-quality HDR images from real scenes using conventional camera equipment. In addition, commercial hardware designed to capture HDR images directly is beginning to enter the market, which is discussed toward the end of this chapter.

4.1 PHOTOGRAPHY AND LIGHT MEASUREMENT

A camera is essentially an imperfect device for measuring the radiance distribution of a scene, in that it cannot capture the full spectral content and dynamic range. (See Chapter 2 for definitions of color and radiance.) The film or image sensor in a conventional or digital camera is exposed to the color and dynamic range of a scene, as the lens is a passive element that merely refocuses the incoming light onto the image plane. All of the information is there, but limitations in sensor design prevent cameras from capturing all of it. Film cameras record a greater dynamic range than their digital counterparts, especially when they expose a negative emulsion.

Standard black-and-white film emulsions have an inverse response to light, as do color negative films. Figure 4.1 shows example response curves for two film emulsions, demonstrating a sensitive range of nearly 4 log units, or a 10,000:1

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 2

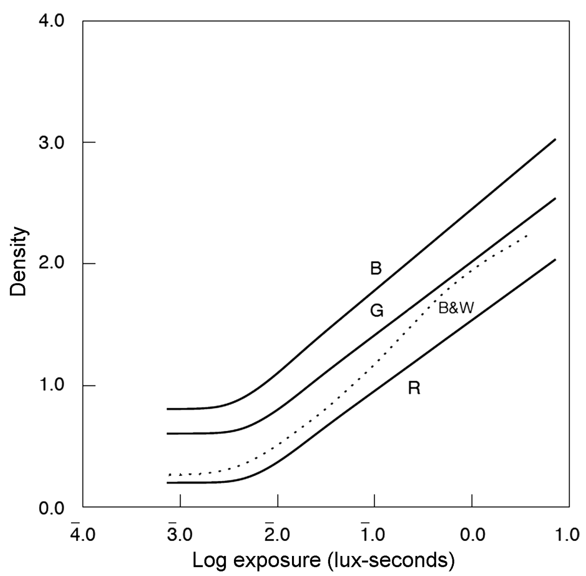

FIGURE 4.1 *Characteristic film curves showing response for color (R, G, and B) and black-and-white negative films over nearly four orders of magnitude.*

contrast ratio. Depending on the quality of the lens and the blackness of the camera's interior, some degree of "flare" may inhibit this range, particularly around a bright point such as the sun or its reflection. Although the capability of recording 4 log units of dynamic range is there, flare may reduce the effective dynamic range somewhat.

The film development process may also limit or enhance the information retrieved from the exposed emulsion, but the final constraining factor is of course the printing process. It is here where tone mapping takes place, in that the effective dynamic range of a black-and-white or color print is about 100:1 at best. Darkroom

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 3

techniques such as dodge-and-burn may be used to get the most out of a negative, although in an industrial film processing lab what usually happens is more akin to autoexposure after the fact.

To extract the full dynamic range from a negative, we need to digitize the developed negative or apply a "dry developing method" such as that developed by Applied Science Fiction and marketed in the Kodak Film Processing Station [25]. Assuming that a developed negative is available, a film scanner would be required that records the full log range of the negative in an HDR format. Unfortunately, no such device exists, although it is technically feasible. However, one can take a standard film scanner, which records either into a 12-bit linear or an 8-bit sRGB color space, and use multiple exposures to obtain a medium-dynamic-range result from a single negative. The process is identical to the idealized case for multiple-exposure HDR capture, which we describe in the following section. The same method may be used to obtain an HDR image from a sequence of exposures using a standard digital camera, or to enhance the dynamic range possible with a film camera.

4.2 HDR IMAGE CAPTURE FROM MULTIPLE EXPOSURES

Due to the limitations inherent in most digital image sensors, and to a lesser degree in film emulsions, it is not possible to capture the full dynamic range of an image in a single exposure. However, by recording multiple exposures a standard camera with the right software can create a single HDR image (i.e., a *radiance map*, as defined in Chapter 2). These exposures are usually captured by the camera itself, although in the case of recovering HDR information from a single negative the same technique may be applied during the film-scanning phase.

By taking multiple exposures, each image in the sequence will have different pixels properly exposed, and other pixels under- or overexposed. However, each pixel will be properly exposed in one or more images in the sequence. It is therefore possible and desirable to ignore very dark and very bright pixels in the subsequent computations.

Under the assumption that the capturing device is perfectly linear, each exposure may be brought into the same domain by dividing each pixel by the image's exposure time. From the recorded radiance values L_{e}, this effectively recovers irradiance values E_{e} by factoring out the exposure duration.[1]

Once each image is in the same unit of measurement, corresponding pixels may be averaged across exposures — excluding, of course, under- and overexposed pixels. The result is an HDR image.

In practice, cameras are not perfectly linear light measurement devices, objects frequently do not remain still between individual exposures, and the camera is rarely kept still. Thus, in practice this procedure needs to be refined to include camera response curves, image alignment techniques, and ghost and lens flare removal.

Extracting a medium-dynamic-range radiance map from a single negative is relatively straightforward because it does not require alignment of multiple frames, and does not suffer from object displacement that may occur during the capture of several exposures. It therefore serves as the basis for the techniques presented later in this chapter.

4.3 FILM SCANNING

In the ideal case for creating an HDR image from multiple LDR exposures, the scene or image should be completely static (e.g., an exposed and developed negative). We assume that the response curve of the film is known. In addition, the LDR capture device (such as an 8-bit/primary film scanner with known response curves) should provide some means of exactly controlling the exposure during multiple captures.

Creating an HDR image under these conditions starts by taking scans with multiple exposures. In addition, the system response is inverted to get back to a linear relation between scene radiances and pixel values. Each scanned image is multiplied by a calibration factor related to its exposure, and combined into an HDR result. The only question is what weighting function to use in averaging together the linear exposures. Of course, the lightest and darkest pixels at the limits of each exposure should be excluded from consideration because these pixels are under- or overexposed. But how should the pixels between be weighted?

1 The quantity captured by the camera is spectrally weighted radiance. As such, calling this quantity "radiance" is inappropriate. However, the spectral response curve is typically not the same as the CIE $V(\lambda)$ curve, and therefore this quantity also cannot be called "luminance" [18]. When the term *radiance* or *irradiance* is used, it should be understood that this refers to spectrally weighted radiance and irradiance.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 5

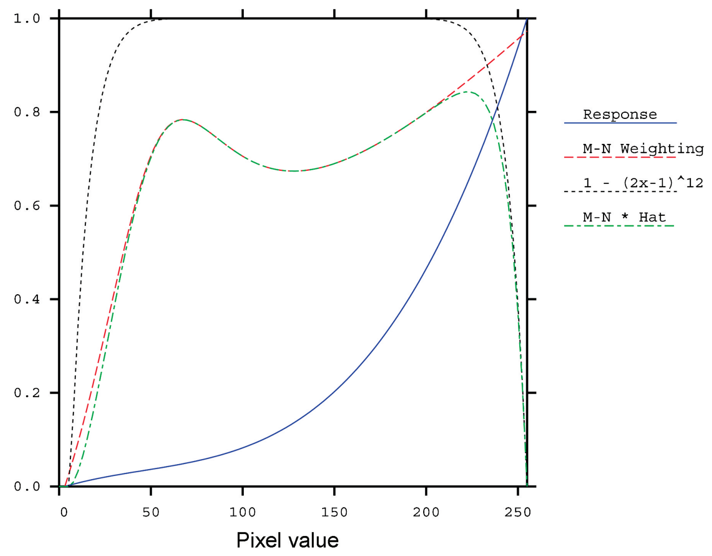

FIGURE 4.2 *The inverted system response function (solid line) and recommended Mitsunaga–Nayar weighting function, multiplied by an additional hat function* $1-(2x-1)^{12}$ *to devalue the extrema, which are often suspect.*

Mann and Picard proposed a certainty/weighting function equal to the derivative of the system response curve for each color channel, using the argument that greater response sensitivity corresponds to greater certainty [5]. Debevec and Malik used a simple hat function based on the assumption that mid-range pixels are more reliable [18]. Mitsunaga and Nayar used signal theory to argue for multiplying Mann and Picard's weight by the response output, in that larger values are less influenced by a constant noise floor [82]. Any of these methods will yield a satisfactory result, although the latter weighting function is better supported by signal theory. The Mitsunaga–Nayar weighting seems to work best when multiplied by a broad hat

FIGURE 4.3 *Our example exposure sequence. Each image is separated by two f-stops (equal to a factor of 4, or* $0.6 \log_{10}$ *units).*

function, as shown in Figure 4.2. This avoids dubious pixels near the extremes, where gamut limitations and clamping may affect the output values unpredictably.

Figure 4.3 shows a sequence of perfectly aligned exposures. Figure 4.4 (left) shows the weighting used for each of the three contributing exposures, where blue is used for the longest exposure, green for the middle exposure, and red for the shortest exposure. As this figure shows, most pixels are a mixture of multiple exposures, with some pixels relying solely on the extremes of the exposure range. Figure 4.4 (right) shows the combined result, tone mapped using a histogram adjustment operator [142].

If the multiple exposures come not from multiple scans of a single negative but from multiple negatives or digital images, combining images may become problematic. First, the camera may shift slightly between exposures, which will result in some subtle (and possibly not-so-subtle) misalignments that will blur the re-

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 7

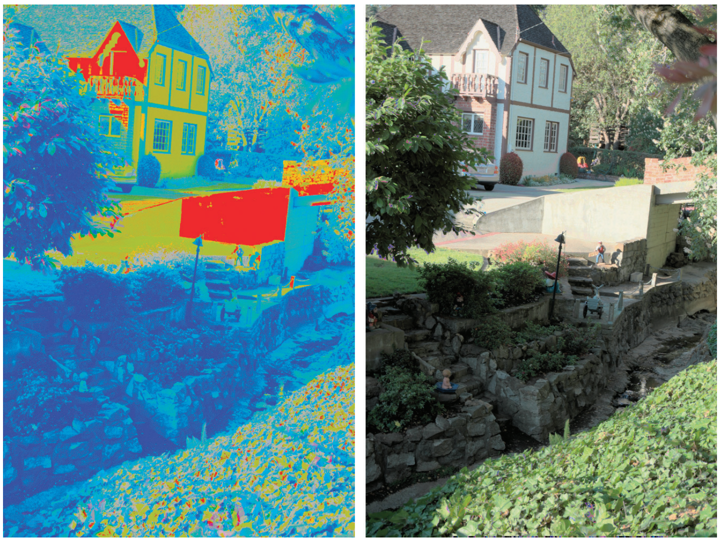

FIGURE 4.4 *The combined HDR result, tone mapped using the histogram adjustment operator (described in Section 7.2.8) in the right-hand image. The left-hand image shows contributing input image weights, where blue shows where the longer exposure dominates, green the middle, and red the shorter exposure. Most output pixels are an average of two or more input values, which reduces noise in the final result.*

sults. Second, if the actual system response function is unknown the images must be aligned before this function can be estimated from the given exposures. Third, objects in the scene may shift slightly between frames or even make large movements, such as people walking in the scene as the photos are taken. Finally, flare in the camera lens may fog areas surrounding particularly bright image regions, which may not be noticeable in a standard LDR image. We will address each of these problems in turn and present some workarounds in the following sections.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 8

4.4 IMAGE REGISTRATION AND ALIGNMENT

Although several techniques have been developed or suggested for image alignment and registration, most originating from the computer vision community, only two techniques to our knowledge address the specific problem of aligning differently exposed frames for the purpose of HDR image creation. The first technique, from Kang et al. [63], handles both camera movement and object movement in a scene, and is based on a variant of the Lucas and Kanade motion estimation technique [77]. In an off-line postprocessing step, for each pixel a motion vector is computed between successive frames. This motion vector is then refined with additional techniques, such as hierarchical homography (introduced by Kang et al.), to handle degenerate cases.

Once the motion of each pixel is determined, neighboring frames are warped and thus registered with one another. Then, the images are ready to be combined into an HDR radiance map. The advantage of this technique is that it compensates for fairly significant motion, and is suitable (for instance) for capturing HDR video by exposing successive frames by different amounts of time.

Although this method is suitable for significant motion, it relies on knowing the camera response function in advance. This presents a catch-22: alignment is needed to register samples to derive the camera response function, but the camera response function is needed to determine alignment. If the camera response is known or can be computed once and stored based on a set of perfectly aligned images, the catch-22 is solved.

A second alignment technique (described in following material) employs a *mean threshold bitmap* (MTB), which does not depend on the camera response function for proper alignment [141]. This technique is also about 10 times faster than the Kang et al. method, in that it performs its alignment operations on bitmaps rather than 8-bit grayscale images, and does not perform image warping or resampling. However, the MTB alignment algorithm does not address moving objects in the scene, and is not appropriate for arbitrary camera movements such as zooming and tilting. The method of Kang et al. may therefore be preferred in cases where arbitrary camera movement is expected. In the case of object motion, we recommend a simpler and more robust postprocessing technique in Section 4.7.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 9

4.5 THE MEAN THRESHOLD BITMAP ALIGNMENT TECHNIQUE

In this section, we describe a method for the automatic alignment of HDR exposures [141].[2] Input to this exposure algorithm is a series of N 8-bit grayscale images, which may be approximated using only the green channel, or derived as follows from 24-bit sRGB with integer arithmetic.[3]

$$Y = (54\ R + 183\ G + 19\ B)/256$$

One of the N images is arbitrarily selected as the reference image, and the output of the algorithm is a series of N-1 (x, y) integer offsets for each of the remaining images relative to this reference. These exposures may then be recombined efficiently into an HDR image using the camera response function, as described in Section 4.6.

The computation focuses on integer pixel offsets, because they can be used to quickly recombine the exposures without resampling. Empirical evidence suggests that handheld sequences do not require rotational alignment in about 90% of cases. Even in sequences in which there is some discernible rotation, the effect of a good translational alignment is to push the blurred pixels out to the edges, where they are less distracting to the viewer.

Conventional approaches to image alignment often fail when applied to images with large exposure variations. In particular, edge-detection filters are dependent on image exposure (as shown in the left side of Figure 4.5, where edges appear and disappear at different exposure levels). Edge-matching algorithms are therefore ill suited to the exposure alignment problem when the camera response is unknown. The MTB approach incorporates the following features.

- Alignment is done on bilevel images using fast bit-manipulation routines.
- The technique is insensitive to image exposure.
- For robustness, it includes noise filtering.

2 Reprinted by permission of A.K. Peters, Ltd., from Greg Ward, “Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures,” *Journal of Graphics Tools*, 8(2):17–30, 2003.

3 This is a close approximation of the computation of luminance as specified by ITU-R BT.709.

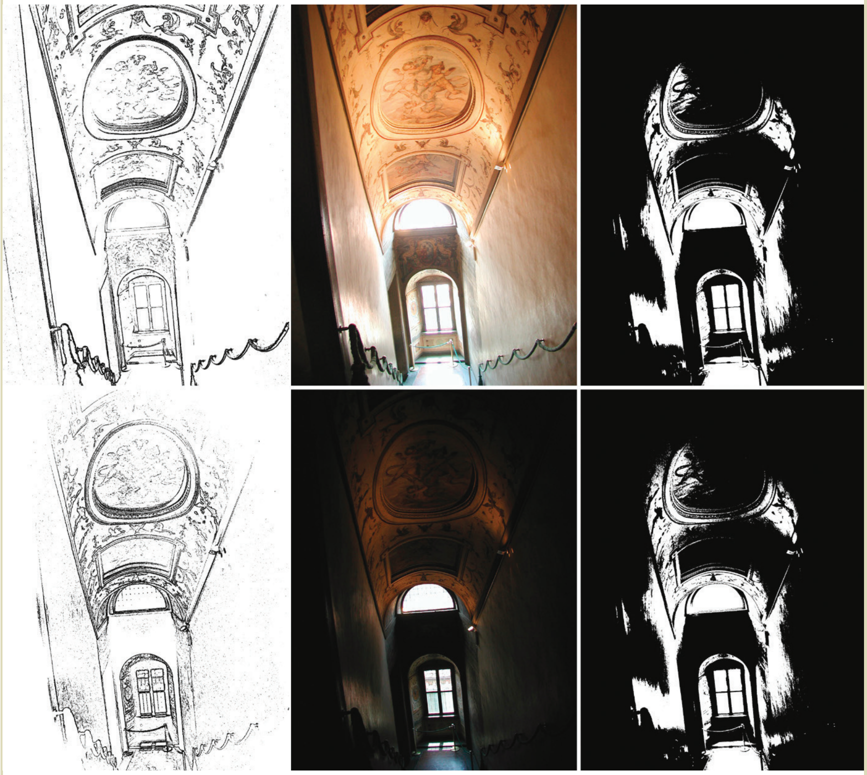

FIGURE 4.5 Two unaligned exposures (middle) and their corresponding edge bitmaps (left) and median threshold bitmaps (right). The edge bitmaps are not used precisely because of their tendency to shift dramatically from one exposure level to another. In contrast, the MTB is stable with respect to exposure.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 11

The results of a typical alignment are discussed in Section 4.5.4. If we are to rely on operations such as moving, multiplying, and subtracting pixels over an entire high-resolution image, the algorithm is bound to be computationally expensive, unless our operations are very fast. Bitmap images allow us to operate on 32 or 64 pixels at a time using bitwise integer operations, which are very fast compared to byte-wise arithmetic. We use a bitmap representation that facilitates image alignment independent of exposure level, the a forementioned *median threshold bitmap*. The MTB is defined as follows.

1. Determine the median 8-bit value from a low-resolution histogram over the grayscale image pixels.
2. Create a bitmap image with 0s where the input pixels are less than or equal to the median value, and 1s are where the pixels are greater.

Figure 4.5 shows two exposures of an Italian stairwell (middle), and their corresponding edge maps (left) and MTBs (right). In contrast to the edge maps, the MTBs are nearly identical for the two exposures. Taking the difference of these two bitmaps with an exclusive-or (XOR) operator shows where the two images are misaligned, and small adjustments in the x and y offsets yield predictable changes in this difference due to object coherence. However, this is not the case for the edge maps, which are noticeably different for the two exposures, even though we attempted to compensate for the camera's nonlinearity with an approximate response curve. Taking the difference of the two edge bitmaps would not give a good indication of where the edges are misaligned, and small changes in the x and y offsets yield unpredictable results, making gradient search problematic. More sophisticated methods of determining edge correspondence are necessary to use this information, and we can avoid these and their associated computational costs with the MTB-based technique.

The constancy of an MTB with respect to exposure is a very desirable property for determining image alignment. For most HDR reconstruction algorithms, the alignment step must be completed before the camera response can be determined, in that the response function is derived from corresponding pixels in the different exposures. An HDR alignment algorithm that depends on the camera response function poses a catch-22 problem, as described earlier. By its nature, an MTB is the same for any exposure within the usable range of the camera, regardless of the

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 12

response curve. As long as the camera's response function is monotonic with respect to world radiance, the same scene will theoretically produce the same MTB at any exposure level. This is because the MTB partitions the pixels into two equal populations: one brighter and one darker than the scene's median value. Because the median value does not change in a static scene, the derived bitmaps likewise do not change with exposure level.[4]

There may be certain exposure pairs that are either too light or too dark to use the median value as a threshold without suffering from noise, and for these we choose either the 17th or 83rd percentile as the threshold, respectively. Although the offset results are all relative to a designated reference exposure, we actually compute offsets between adjacent exposures and thus the same threshold may be applied to both images. Choosing percentiles other than the 50th (median) results in fewer pixels to compare, and this makes the solution less stable and thus we may choose to limit the maximum offset in certain cases. The behavior of percentile threshold bitmaps is otherwise the same as the MTB, including stability over different exposures. In the remainder of this section, when we refer to the properties and operations of MTBs, the same applies for other percentile threshold bitmaps.

Once the threshold bitmaps corresponding to the two exposures have been computed, there are several ways to align them. One brute force approach is to test every offset within the allowed range, computing the XOR difference at each offset and taking the coordinate pair corresponding to the minimum difference. A more efficient approach might follow a gradient descent to a local minimum, computing only local bitmap differences between the starting offset (0,0) and the nearest minimum. We prefer a third method, based on an image pyramid that is as fast as gradient descent in most cases but more likely to find the global minimum within the allowed offset range.

Multiscale techniques are well known in the computer vision and image-processing communities, and image pyramids are frequently used for registration and alignment. (See, for example, [127].) This technique starts by computing an image pyramid for each grayscale image exposure, with $\log_2(\text{max\_offset})$ levels past the base resolution. The resulting MTBs are shown for two example exposures in Figure 4.6. For each smaller level in the pyramid, we take the previous grayscale

4 Technically, the median value could change with changing boundaries as the camera moves, but such small changes in the median are usually swamped by noise, which is removed by this algorithm.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 13

FIGURE 4.6 *A pyramid of MTBs is used to align adjacent exposures one bit at a time. The smallest (rightmost) image pair corresponds to the most significant bit in the final offset.*

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 14

image and filter it down by a factor of two in each dimension, computing the MTB from the grayscale result. The bitmaps themselves should *not* be subsampled, as the result will be subtly different and could potentially cause the algorithm to fail.

To compute the overall offset for alignment, we start with the lowest-resolution MTB pair and compute the minimum difference offset between them within a range of ± 1 pixel in each dimension. At the next resolution level, we multiply this offset by 2 (corresponding to the change in resolution) and compute the minimum difference offset within a ± 1 pixel range of this previous offset. This continues to the highest-resolution (original) MTB, where we get the final offset result. Thus, each level in the pyramid corresponds to a binary bit in the computed offset value.

At each level, we need to compare exactly nine candidate MTB offsets, and the cost of this comparison is proportional to the size of the bitmaps. The total time required for alignment is thus linear with respect to the original image resolution and independent of the maximum offset, in that the registration step is linear in the number of pixels, and the additional pixels in an image pyramid are determined by the size of the source image and the (fixed) height of the pyramid.

4.5.1 THRESHOLD NOISE

The algorithm just described works well in images that have a fairly bimodal brightness distribution, but can run into trouble for exposures that have a large number of pixels near the median value. In such cases, the noise in near-median pixels shows up as noise in the MTB, which destabilizes the difference computations.

The inset in Figure 4.7 shows a close-up of the pixels in the dark stairwell exposure MTB, which is representative of the type of noise seen in some images. Computing the XOR difference between exposures with large areas such as these yields noisy results that are unstable with respect to translation because the pixels themselves tend to move around in different exposures. Fortunately, there is a straightforward solution to this problem.

Because this problem involves pixels whose values are close to the threshold, these pixels can be excluded from our difference calculation with an *exclusion bitmap*. The exclusion bitmap consists of 0s wherever the grayscale value is within some specified distance of the threshold, and 1s elsewhere. The exclusion bitmap for the

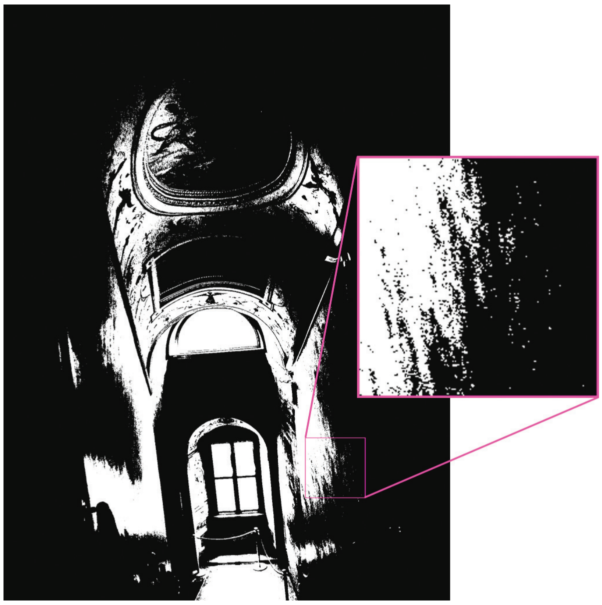

FIGURE 4.7 *Close-up detail of noisy area of MTB in dark stairwell exposure (full resolution).*

exposure in Figure 4.7 is shown in Figure 4.8, where all bits are zeroed for pixels within ±4 of the median value.

We compute an exclusion bitmap for each exposure at each resolution level in the MTB pyramid, and then take the XOR difference result for each candidate offset, ANDing it with both offset exclusion bitmaps to compute the final difference.[5] The effect is to disregard differences that are less than the noise tolerance in our images.

5 If we were to AND the exclusion bitmaps with the original MTBs before the XOR operation, we would inadvertently count disagreements about what was noise and what was not as actual pixel differences.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 16

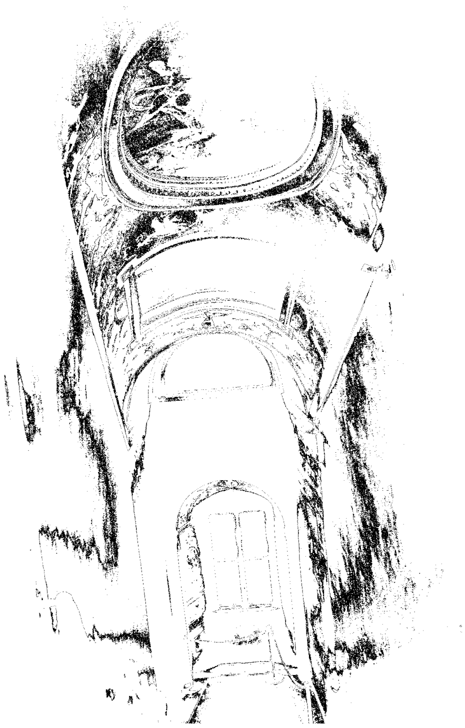

FIGURE 4.8 *An exclusion bitmap, with zeroes (black) wherever pixels in our original image are within the noise tolerance of the median value.*

This is illustrated in Figure 4.9, which shows the XOR difference of the unaligned exposures before and after applying the exclusion bitmaps. By removing those pixels that are close to the median, the least-reliable bit positions in the smooth gradients are cleared but the high-confidence pixels near strong boundaries (such as the edges

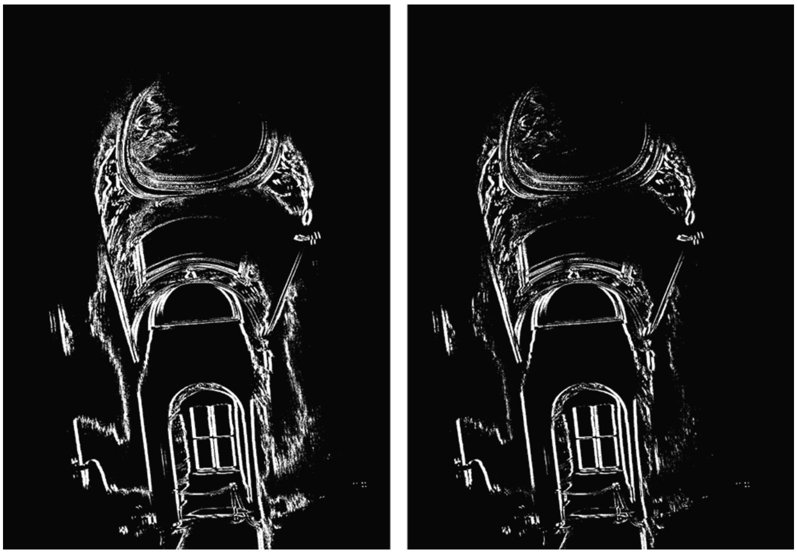

FIGURE 4.9 *The original XOR difference of the unaligned exposures (left), and with the two exclusion bitmaps ANDed into the result to reduce noise in the comparison (right).*

of the window and doorway) are preserved. Empirically, this optimization seems to be very effective in eliminating false minima in the offset search algorithm.

4.5.2 OVERALL ALGORITHM

The full algorithm with the exclusion operator is given in the recursive C function (`GetExpShift`), shown in Figure 4.10. This function takes two exposure images, and determines how much to move the second exposure (`img2`) in x and y to align it with the first exposure (`img1`). The maximum number of bits in the

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 18


```
GetExpShift (const Image *img1, const Image *img2,
             int shift_bits, int shift_ret[2])
{
  int    min_err;
  int    cur_shift[2];
  Bitmap tb1, tb2;
  Bitmap eb1, eb2;
  int    i, j;
  if (shift_bits > 0) {
    Image sml_img1, sml_img2;
    ImageShrink2(img1, &sml_img1);
    ImageShrink2(img2, &sml_img2);
    GetExpShift(&sml_img1, &sml_img2, shift_bits-1, cur_shift);
    ImageFree(&sml_img1);
    ImageFree(&sml_img2);
    cur_shift[0] *= 2;
    cur_shift[1] *= 2;
  } else
    cur_shift[0] = cur_shift[1] = 0;
  ComputeBitmaps(img1, &tb1, &eb1);
  ComputeBitmaps(img2, &tb2, &eb2);
  min_err = img1->xres * img1->yres;
  for (i = -1; i <= 1; i++)
    for (j = -1; j <= 1; j++) {
      int     xs = cur_shift[0] + i;
      int     ys = cur_shift[1] + j;
      Bitmap  shifted_tb2;
      Bitmap  shifted_eb2;
      Bitmap  diff_b;
      int     err;
      BitmapNew(img1->xres, img1->yres, &shifted_tb2);
      BitmapNew(img1->xres, img1->yres, &shifted_eb2);
      BitmapNew(img1->xres, img1->yres, &diff_b);
      BitmapShift(&tb2, xs, ys, &shifted_tb2);
      BitmapShift(&eb2, xs, ys, &shifted_eb2);
```

FIGURE 4.10 The `GetExpShift` algorithm.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 19


```
      BitmapXOR(&tb1, &shifted_tb2, &diff_b);
      BitmapAND(&diff_b, &eb1, &diff_b);
      BitmapAND(&diff_b, &shifted_eb2, &diff_b);
      err = BitmapTotal(&diff_b);
      if (err < min_err) {
        shift_ret[0] = xs;
        shift_ret[1] = ys;
        min_err = err;
      }
      BitmapFree(&shifted_tb2);
      BitmapFree(&shifted_eb2);
    }
    BitmapFree(&tb1); BitmapFree(&eb1);
    BitmapFree(&tb2); BitmapFree(&eb2);
}
```

FIGURE 4.10 (*Continued.*)

final offsets is determined by the `shift_bits` parameter. The more important functions called by `GetExpShift` are as follows.

`ImageShrink2 (const Image *img, Image *img_ret)`: Subsample the image `img` by a factor of two in each dimension and put the result into a newly allocated image `img_ret`.

`ComputeBitmaps (const Image *img, Bitmap *tb, Bitmap *eb)`: Allocate and compute the threshold bitmap `tb` and the exclusion bitmap `eb` for the image `img`. (The threshold and tolerance to use are included in the `Image struct`.)

`BitmapShift (const Bitmap *bm, int xo, int yo, Bitmap *bm_ret)`: Shift a bitmap by (`xo`,`yo`) and put the result into the preallocated bitmap `bm_ret`, clearing exposed border areas to zero.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 20

`BitmapXOR (const Bitmap *bm1, const Bitmap *bm2, Bitmap *bm_ret)`: Compute the XOR of `bm1` and `bm2` and put the result into `bm_ret`.

`BitmapTotal (const Bitmap *bm)`: Compute the sum of all 1 bits in the bitmap.

Computing the alignment offset between two adjacent exposures is simply a matter of calling the `GetExpShift` routine with the two image structs (`img1` and `img2`), which contain their respective threshold and tolerance values. (The threshold values must correspond to the same population percentiles in the two exposures.) We also specify the maximum number of bits allowed in the returned offset, `shift_bits`. The shift results computed and returned in `shift_ret` will thus be restricted to a range of $\pm 2^{\texttt{shift\_bits}}$.

There is only one subtle point in this algorithm, which is what happens at the image boundaries. Unless proper care is taken, non-zero bits may inadvertently be shifted into the candidate image. These would then be counted as differences in the two exposures, which would be a mistake. It is therefore crucial that the `BitmapShift` function shifts 0s into the new image areas, so that applying the shifted exclusion bitmap to the XOR difference will clear these exposed edge pixels as well. This also explains why the maximum shift offset needs to be limited. In the case of an unbounded maximum shift offset, the lowest-difference solution will also have the least pixels in common between the two exposures (one exposure will end up shifted completely off the other). In practice, we have found a `shift_bits` limit of 6 (±64 pixels) to work fairly well most of the time.

4.5.3 EFFICIENCY CONSIDERATIONS

Clearly, the efficiency of the MTB alignment algorithm depends on the efficiency of the bitmap operations, as nine shift tests with six whole-image bitmap operations apiece are performed. The `BitmapXOR` and `BitmapAND` operations are easy enough to implement, as we simply apply bitwise operations on 32-bit or 64-bit words, but the `BitmapShift` and `BitmapTotal` operators may not be as obvious.

For the `BitmapShift` operator, any 2D shift in a bitmap image can be reduced to a 1D shift in the underlying bits, accompanied by a clear operation on one or two

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 21

edges for the exposed borders. Implementing a 1D shift of a bit array requires at most a left or right shift of B bits per word, with a reassignment of the underlying word positions. Clearing the borders then requires clearing words where sequences of 32 or 64 bits are contiguous, and partial clears of the remaining words. The overall cost of this operator, although greater than the XOR or AND operators, is still modest. This `BitmapShift` implementation includes an additional Boolean parameter that turns off border clearing. This optimizes the shifting of the threshold bitmaps, which have their borders cleared later by the exclusion bitmap, and thus the `BitmapShift` operator does not need to clear them.

For the `BitmapTotal` operator, a table of 256 integers is computed corresponding to the number of 1 bits in the binary values from 0 to 255 (i.e., 0, 1, 1, 2, 1, 2, 2, 3, 1, . . . , 8). Each word of the bitmap can then be broken into chunks (measured in bytes), and used to look up the corresponding bit counts from the precomputed table. The bit counts are then summed to yield the correct total. This results in a speedup of at least 8 times over counting individual bits, and may be further accelerated by special-case checking for zero words, which occur frequently in this application.

4.5.4 RESULTS

Figure 4.11 shows the results of applying the MTB image alignment algorithm to all five exposures of the Italian stairwell, with detailed close-ups showing before and after alignment. The misalignment shown is typical of a handheld exposure sequence, requiring translation of several pixels on average to bring the exposures back atop each other. We have found that even tripod exposures sometimes need minor adjustments of a few pixels for optimal results.

After applying this translational alignment algorithm to over 100 handheld exposure sequences, a success rate of about 84% was found, with 10% giving unsatisfactory results due to image rotation. About 3% failed due to excessive scene motion — usually waves or ripples on water that happened to be near the threshold value and moved between frames — and another 3% had too much high-frequency content, which made the MTB correspondences unstable. Most of the rotation failures were mild, leaving at least a portion of the HDR image well aligned. Other failures were more dramatic, throwing alignment off to the point where it was better not to apply any translation at all.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 22

FIGURE 4.11 *An HDR image composited from unaligned exposures (left) and detail (top center). Exposures aligned with the MTB algorithm yield a superior composite (right) with clear details (bottom center).*

4.6 DERIVING THE CAMERA RESPONSE FUNCTION

Combining LDR exposures into an HDR image requires knowledge of the camera response function to linearize the data. In general, the response function is not provided by camera makers, who consider it part of their proprietary product differentiation. Assuming an sRGB response curve (as described in Chapter 2) is unwise, because most makers boost image contrast beyond the standard sRGB gamma to produce a livelier image. There is often some modification as well at the ends of the curves, to provide softer highlights and reduce noise visibility in shadows. However, as long as the response is not altered by the camera from one exposure to the next, it is possible to deduce this function given a proper image sequence.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 23

4.6.1 DEBEVEC AND MALIK TECHNIQUE

Debevec and Malik [18] demonstrated a simple and robust technique for deriving the camera response function from a series of aligned exposures, extending earlier work by Mann and Picard [5]. The essential idea is that by capturing different exposures of a static scene one is effectively sampling the camera response function at each pixel. This is best demonstrated graphically.

Figure 4.12 shows three separate image positions sampled at five different exposures (Figure 4.13). The relative exposure ratios at each of the three positions are

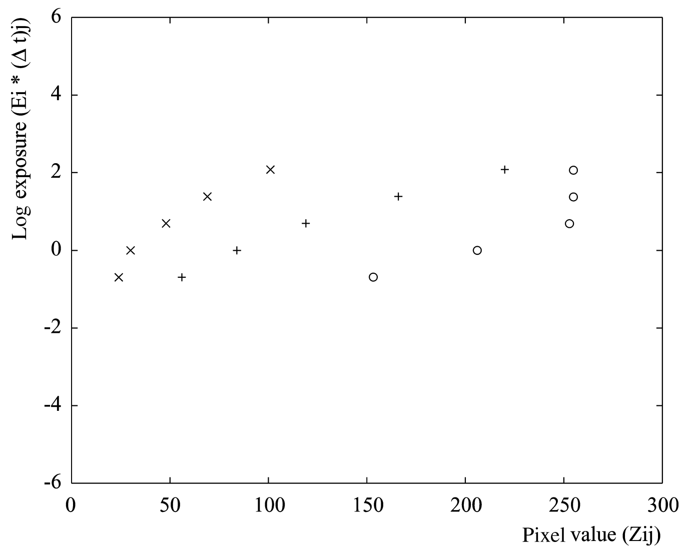

FIGURE 4.12 *Plot of $g(Z_{ij})$ from three pixels observed in five images, assuming unit radiance at each pixel. The images are shown in Figure 4.13.*

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 24

FIGURE 4.13 *Three sample positions over five exposures shown in Figure 4.12.*

given by the speed settings on the camera, and thus we know the shape of the response function at three different parts of the curve. However, we do not know how these three curve fragments fit together. Debevec and Malik resolved this problem using linear optimization to find a smooth curve that minimizes the mean-squared error over the derived response function. The objective function they use to derive the logarithmic response function $g(Z_{ij})$ is as follows.

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \left\{ w(Z_{ij}) \left[g(Z_{ij}) - \ln E_i - \ln \Delta t_j \right] \right\}^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} \left[w(z)\, g''(z) \right]^2$$

There, Δt_j is the exposure time for exposure j, E_i is the film irradiance value at image position i, and Z_{ij} is the recorded pixel at position i and exposure j. The weighting function $w(Z_{ij})$ is a simple hat function, as follows.

$$w(z) = \begin{cases} z - Z_{\min} & \text{for } z \le \frac{1}{2}(Z_{\min} + Z_{\max}) \\ Z_{\max} - z & \text{for } z > \frac{1}{2}(Z_{\min} + Z_{\max}) \end{cases}$$

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 25

This equation is solved using singular-value decomposition to obtain an optimal value for $g(Z)$ for every possible value of Z, or 0 to 255 for an 8-bit image. Each of the RGB channels is treated separately, yielding three independent response functions. This assumes that interactions between the channels can be neglected. Although this assumption is difficult to defend from what we know about camera color transformations, it seems to work fairly well in practice.

Figure 4.14 shows the result of aligning the three curves from Figure 4.12, and by applying the minimization technique to many image samples it is possible to obtain a smooth response function for each channel.

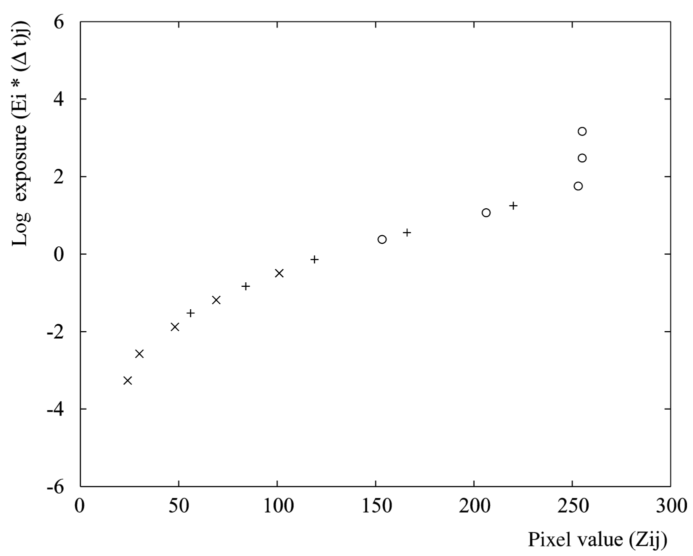

FIGURE 4.14 Normalized plot of $g(Z_{ij})$ after determining pixel exposures.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 26

4.6.2 MITSUNAGA AND NAYAR TECHNIQUE

Mitsunaga and Nayar presented a similar approach, in which they derive a polynomial approximation to the response function [82] rather than the enumerated table of Debevec and Malik. The chief advantage they cite in their technique is the ability to resolve the exact exposure ratios in addition to the camera response function. This proves important for lower-cost consumer equipment whose aperture and shutter speed may not be known exactly. Mitsunaga and Nayar define the following N-dimensional polynomial for their response function:

$$f(M) = \sum_{n=0}^{N} c_n M^n$$

For consistency with Debevec and Malik, we suggest the following variable replacements:

$$\begin{aligned} N &\to K \\ n &\to k \\ M &\to Z \\ Q &\to P \\ q &\to j \end{aligned}$$

The final response function is thus defined by the $N + 1$ coefficients of this polynomial, $\{c_0, \ldots c_N\}$. To determine these coefficients, they minimize the following error function for a given candidate exposure ratio, $R_{q,q+1}$ (the scale ratio between exposure q and $q + 1$):

$$\varepsilon = \sum_{q=1}^{Q-1} \sum_{p=1}^{P} \left[\sum_{n=0}^{N} c_n M_{p,q}^n - R_{q,q+1} \sum_{n=0}^{N} c_n M_{p,q+1}^n \right]^2$$

The minimum is found by determining where the partial derivatives with respect to the polynomial coefficients are all zero (i.e., solving the following system of

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 27

$N+1$ linear equations).

$$\frac{\partial \varepsilon}{\partial c_n} = 0$$

As in previous methods, they only solve for the response up to some arbitrary scaling. By defining $f(1) = 1$, they reduce the dimensionality of their linear system by one coefficient, substituting

$$c_N = 1 - \sum_{n=0}^{N-1} c_n.$$

The final $N \times N$ system can be written as follows.

$$\begin{bmatrix} \sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,0}(d_{p,q,0} - d_{p,q,N}) & \dots & \sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,0}(d_{p,q,N-1} - d_{p,q,N}) \\ \dots & \dots & \dots \\ \sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,N-1}(d_{p,q,0} - d_{p,q,N}) & \dots & \sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,N-1}(d_{p,q,N-1} - d_{p,q,N}) \end{bmatrix}$$
$$\times \begin{bmatrix} c_0 \\ \dots \\ c_{N-1} \end{bmatrix} = \begin{bmatrix} -\sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,0} d_{p,q,N} \\ \dots \\ -\sum_{q=1}^{Q-1} \sum_{p=1}^{P} d_{p,q,N-1} d_{p,q,N} \end{bmatrix}$$

Here,

$$d_{p,q,n} = M_{p,q}^n - R_{q,q+1} M_{p,q+1}^n.$$

The original Mitsunaga and Nayar formulation only considers adjacent exposures. In practice, the system is more stable if all exposure combinations are considered. The error function becomes a triple sum by including a sum over $q' \neq q$ instead of just comparing q to $q+1$. This then gets repeated in the sums of the combined system of equations, where $d_{p,q,n}$ is replaced by

$$d_{p,q,q',n} = M_{p,q}^n - R_{q,q'} M_{p,q'}^n.$$

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 28

To compute the actual exposure ratios between images, Mitsunaga and Nayar apply an interactive technique, where the previous system of equations is solved repeatedly, and between each solution the exposure ratios are updated using the following.

$$R_{q,q+1}^{(k)} = \sum_{p=1}^{P} \frac{\displaystyle\sum_{n=0}^{N} c_n^{(k)} M_{p,q}^n}{\displaystyle\sum_{n=0}^{N} c_n^{(k)} M_{p,q+1}^n}$$

Iteration is complete when the polynomial is no longer changing significantly, as follows.

$$\left| f^{(k)}(M) - f^{(k-1)}(M) \right| < \varepsilon, \quad \forall M$$

This leaves just one final problem: What is the polynomial degree N? The authors recommend solving for every degree polynomial up to some maximum exponent (e.g., 10), accepting the solution with the smallest error, ε. Fortunately, the solution process proceeds quickly and this is not much of a burden. It is a good idea to ensure that the same degree is selected for all color channels, and thus a combined ε function is preferable for this final test.

4.6.3 CHOOSING IMAGE SAMPLES FOR RESPONSE RECOVERY

Each of the techniques described for camera response recovery requires a set of intelligently selected samples from the exposure sequence. In principle, one could use every pixel from every image, but this would only add to the computation time while actually reducing stability in the solution due to misaligned and noisy data. Once the exposures have been aligned, the following procedure for selecting sample patches is recommended.

1. Sort the exposures from lightest to darkest.
2. Select an appropriate sample patch size and an optimal number of patches, and initialize (clear) the patch list.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 29

3. Determine how many patches from the previous exposure are still valid for this one.
4. Compute how many more patches are needed for this exposure. If none, go to the next exposure (Step 3).
5. Search for valid patches using randomized rejection sampling. A valid patch is brighter than any of the previous exposure's patches, does not overlap any other patch, and possesses a low internal variance. It is also within the valid range for this exposure.
6. Once we have found enough patches or given up due to an excess of rejected samples, we continue to the next exposure (Step 3).

A target of 50 12-by-12 pixel patches per exposure seems to work well. In cases where the darker exposures do not use their full range, it becomes difficult to find new patches that are brighter than the previous exposure. In practice, this does not affect the result significantly, but it is important for this reason to place a limit on the rejection sampling process in Step 5, lest we go into an infinite loop.

Figure 4.15 shows an exposure sequence and the corresponding patch locations. Adjacent exposures have nearly the same patch samples, but no patch sample survives in all exposures. This is due to the range restriction applied in Step 5 to avoid unreliable pixel values. Figure 4.16 shows a close-up of the middle exposure with the patches shown as boxes, demonstrating the low variance in the selected regions. By rejecting high-contrast areas, errors due to exposure misalignment and sensor noise are minimized.

Finally, Figure 4.17 shows the recovered response function for this sequence fitted with a third-order polynomial using Mitsunaga and Nayar's method, and compares it to the standard sRGB response function. The camera produces an artificially exaggerated contrast with deeper blacks on its LDR exposures. This type of response manipulation is fairly standard for consumer-grade cameras, and many professional SLRs as well.

4.6.4 CAVEATS AND CALIBRATION

To apply these techniques successfully, it helps to follow some additional guidelines, as follows.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 30

FIGURE 4.15 *Red squares indicate size and location of patch samples in each exposure.*

- Use aperture priority or manual exposure mode, so that only the exposure time is allowed to vary. This reduces problems associated with vignetting (light falloff toward the edge of the image).
- Fix the camera's white balance on a specific setting for the entire sequence, preferably daylight (i.e., D_{65}).
- If the camera offers an "optimized color and contrast" mode, switch it off. The more settings you can fix manually, the less likely the camera will be altering the response function between exposures. This applies particularly to automatic ISO/ASA and programmed exposure modes.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 31

FIGURE 4.16 *Close-up of central exposure showing selected patch regions.*

- Use a tripod if possible, and control your camera via a tether to a laptop computer if this option is available. The less touching of the camera during a sequence the fewer alignment problems you will experience.

In general, it works best to calibrate your camera's response one time, and then reuse this calibration for later exposure sequences. In this way, the scene and exposure sequence may be optimized for camera response recovery. For such a sequence, perform the following.

- Set the camera on a tripod and use a tether if available. Alignment may still be necessary between exposures if the camera is touched during the sequence, but the method described in the previous section will work far better with

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 32

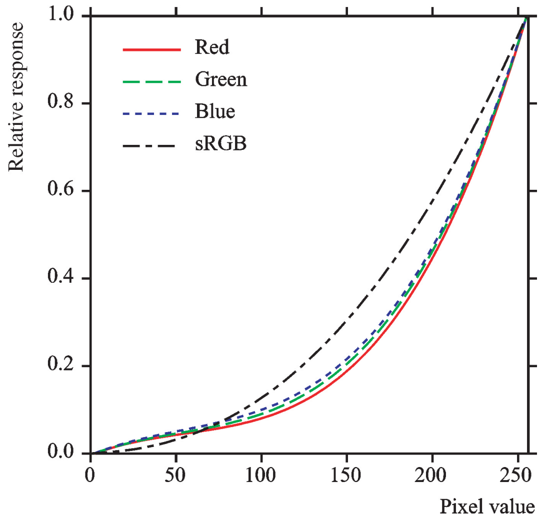

FIGURE 4.17 *Recovered red, green, and blue response functions for the image sequence shown in Figure 4.15.*

a tripod than a handheld sequence in which the exposure is being changed manually.

- Choose a scene with large gray or white surfaces that provide continuous gradients for sampling. The closer your scene is to a neutral color the less likely color transforms in the camera will undermine the response recovery process.
- Choose a scene with very bright and very dark areas, and then take a long sequence of exposures separated by 1 EV (a factor of two in exposure time). The darkest exposure should have no RGB values greater than 200 or so, and the lightest exposure should have no RGB values less than 20 or so. Do not

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 33

include an excess of exposures beyond this range, as it will do nothing to help with response recovery and may hurt.

- If you have access to a luminance meter, take a reading on a gray card or uniform area in your scene to provide absolute response calibration.

Once a camera has been characterized in this way, it is possible to combine handheld bracketed sequences that are too short to reliably recover the response function.

4.7 GHOST REMOVAL

Once the exposures are aligned with each other and the camera's response curve is determined, we may safely combine the images (as described in Section 4.3). However, if some person or object was moving during the image sequence acquisition they may appear as "ghosts" in the combined result, due to their multiple locations. The technique described by Kang et al. [63] attempts to address this problem during alignment by warping pixels according to local content, but even if this can be done correctly in the presence of people who change posture as well as position it still leaves the problem of filling in holes that were obstructed in some views but not in others.

A simpler approach is based on the observation that each exposure in the sequence is self-consistent, which means that we can simply choose one exposure or another in specific regions to obtain a ghost-free result. The HDR capacity may be lost within these selected regions, but as long as the ghosts are local and compact, the overall image will still capture the full range of light.

Figure 4.18 shows an HDR image captured from a bracketed sequence of five exposures, excerpted in the left-hand side of the figure. People walking in and out of the temple result in a trail of ghosts appearing in the combined result.

Fortunately, it is relatively easy to detect motion of this type in exposure sequences. As the images are combined using the weighted average (described in Section 4.3), the weighted variance can be computed simultaneously at each pixel, shown in Figure 4.19. The weighted variance is defined as the weighted sum of squares at each pixel over the square of the weighted average, the quantity minus 1. (We compute these quantities separately for red, green, and blue channels and then take the maximum at each pixel.) In addition to the moving people, some variance

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 34

FIGURE 4.18 *Five exposures combined into a single HDR image, where people moving through the scene have caused ghosting in the result.*

is detected at high-contrast edges due to imperfections in the lens and sensor. These regions will usually be rejected by a minimum area constraint, but may cause false positives on small stationary objects.

At this point, the variance image could simply be thresholded and a single exposure selected to substitute for all high-variance pixels. However, this would incur significant artifacts. First, parts of moving objects whose pixels happened to correspond to the background locally would break apart. Second, and more seriously, choosing a single exposure for all high-variance pixels would result in excessive information loss, as problem pixels may be found in very different brightness regions.

The algorithm could be modified to pick the best exposure for each problem pixel, but this would create an even more serious breakup problem because different parts of the same object will be better exposed in different frames, where

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 35

FIGURE 4.19 *Variance computed at each pixel over our exposure sequence, showing where information is changing unexpectedly due to movement.*

the object's position is also different. It is therefore important to isolate separable, high-variance regions and choose the best exposure for each. Such a segmentation is shown in Figure 4.20. This segmentation is computed as follows.

1. Reduce the variance image by a factor of 10 in each dimension to save computation.
2. Compute the threshold bitmap where local variance is greater than 0.18.
3. Smear the threshold bitmap around a radius of 3 pixels to cover edges and join adjacent ghost regions.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 36

FIGURE 4.20 *Segmented regions corresponding to isolated ghosts in our exposure sequence. Segment colors were randomly selected.*

4 Compute a "background" bitmap segment from a union of contiguous (flood-filled) low-variance regions that cover at least 0.1% of the image.
5 Identify "ghost" bitmap segments as disjoint flood-filled regions within the background segment, with each ghost also covering at least 0.1% of the image.

In Figure 4.20, the tops of the two silhouetted poles were inadvertently picked, due to the high variance at their edges. In addition, a large region in which people where passing each other on the walk ended up as one segment (violet). Fortunately, this segment is contained in a similarly lighted region, and thus a single exposure is adequate to capture its dynamic range. The uneven edges of some regions indicate

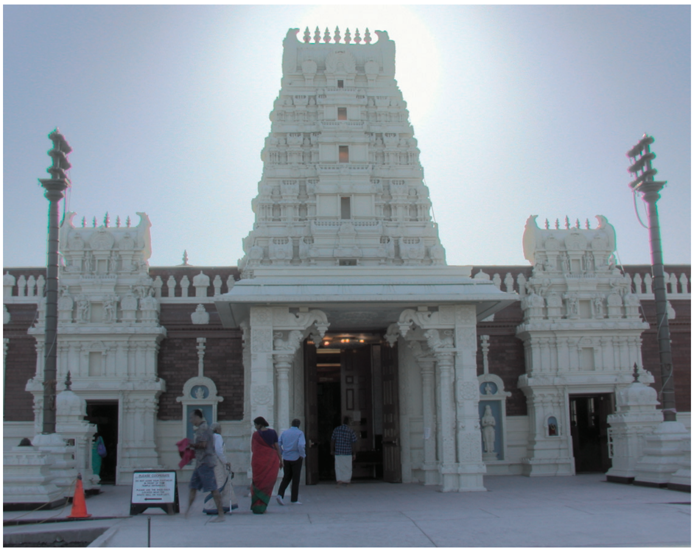

FIGURE 4.21 *The combined HDR result with ghosts removed.*

low-variance pixels, where we limit our ghost removal using linear interpolation as explained below.

To choose which exposure to use in which region, a histogram is generated from the floating-point values for each ghost segment. We then consider the largest value after ignoring the top 2% as outliers and choose the longest exposure that includes this 2% maximum within its valid range. We apply the corresponding exposure multiplier for each ghost segment then linearly interpolate between this exposure and the original HDR result using each pixel's variant as our mixing coefficient. This ensures that extremely low-variance pixels within an identified ghost segment are left unaltered. Figure 4.21 shows the combined result with ghosts removed. The final image is not perfect, and one man's bare foot has been summarily amputated, but the overall result is an improvement, and this technique is quick and relatively straightforward.

4.8 LENS FLARE REMOVAL

After eliminating motion between exposures, there may still be artifacts present due to the camera's optics. Most digital cameras are equipped with optics that are consistent with the inherent limitations of 24-bit digital images. In other words, manufacturers generally do not expect more than two orders of magnitude to be captured in the final image, and thus certain parameters may be relaxed in the lens and sensor design relative to a 35-mm-film camera, for example. For an HDR capture process, however, the limitations of the system's optics are more apparent, even in a well-made digital camera. Small issues such as the thickness and finish of the aperture vanes can make a big difference in the distribution of light on the sensor. The quality of coatings on the lenses and the darkness and geometry of the interior surrounding the sensor also come into play. Overall, there are many components that affect the scattering of light in an image, and it is difficult or impossible to arrive at a single set of measurements that characterize the system and its dependencies. Therefore, we prefer a dynamic solution to the lens flare problem, based only on the captured image.

Because it is important to keep all of the optical properties of the camera consistent during HDR capture, normally only the shutter speed should be manipulated between exposures. Thus, the actual distribution of light on the sensor plane never varies, only the length of time the sensor is exposed to it. Therefore, any flare effects present in one exposure are present to the same degree in all exposures and will sum consistently into our HDR result. For this reason, there is no need to work on individual exposures, as it would only serve to increase our computational burden. The camera's *point spread function* (PSF) is a physical measure of the system optics, and it may be characterized directly from the recorded radiances in an HDR image.

4.8.1 THE POINT SPREAD FUNCTION

The PSF as it is defined here is an idealized radially symmetric characterization of the light falloff surrounding a point of light in a perfectly dark surrounding. It

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 39

FIGURE 4.22 *An isolated spot of light in a darkened environment for the purpose of measuring the point spread function of a camera.*

could be measured by making a pinhole in a piece of aluminum foil in front of a lightbulb in a box and photographing it in a completely dark environment, as shown in Figure 4.22. The edge of the hole ought to be perfectly sharp, but it generally is not. The spread of light around the hole corresponds to light scattered within the lens of the digital camera.

This photograph of the pinhole could then be used to correct the combined HDR result for other photographs made with precisely the same lens settings — zoom and aperture. However, this procedure is a lot to expect of even the most meticulous photographer, and because lens flare also depends strongly on dust and oils that come and go over time it is not practical to maintain a set of calibrated PSFs for any but the most critical applications.

However, there is a technique whereby the PSF may be approximated based on image content, as we will demonstrate with the HDR capture shown in Figure 4.23.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 40

FIGURE 4.23 *Our input test image for estimating lens flare using the same aperture and zoom as in Figure 4.22 (tone mapped with the histogram equalization operator, described in Section 7.2.8).*

Despite the fact that this image contains no localized bright spots, and hence no easily measured PSF, a reasonable estimate of lens flare may still be obtained.

It is assumed that in the image there exist some dark pixels near very bright (or "hot") pixels. To the extent this is true, it will be possible to estimate the PSF

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 41

for the camera.[6] It is also assumed that the lens flare is radially symmetric. This is admittedly a crude approximation, but required by the estimation procedure. Thus, the goal is to find and remove the radially symmetric component of flare. Streaks and other asymmetrical artifacts generated by the camera optics will remain. The automatic flare removal consists of the following steps.

1. Compute two reduced-resolution HDR images: one in color and one in grayscale. Call these I_{CR} and I_{GR}, respectively.
2. Identify "hot" pixels in I_{GR}, which are over some threshold.
3. Draw annuli around each hot pixel to compute a least squares approximation to the PSF using the method described in the following section.
4. Apply the PSF to remove flare from the final HDR image.

Reducing the working resolution of our HDR image achieves a major speedup without significantly impacting the quality of the results, in that flare tends to be a distributed phenomenon. A reduced image size of at most 128 pixels horizontally or vertically is sufficient. The threshold setting for Step 2 is not particularly important, but we have found a value of 1,000 times the minimum (reduced) pixel value to work well for most images. Of course, a different threshold is advisable if the minimum is zero. Steps 3 and 4 require some explanation, which we give in the following subsections.

4.8.2 ESTIMATING THE PSF

The PSF defines how light falls off around bright points in the image.[7] To estimate the PSF, the minimum pixel values around all "hot" pixels in the image are measured, thus arriving at a conservative estimate of the PSF. To do this, the potential contributions of all hot pixels at a certain distance from the darker (non-hot) pixels are summed to build up an estimate of the PSF from the corresponding minima, radius by radius. For example, Figure 4.24 shows an image with exactly three of

6 If this assumption is false, and there are no dark pixels near sources, lens flare will probably go unnoticed and there is no need to remove it.

7 In fact, it defines how light falls off around any point in the image, but only the bright points matter because the falloff is so dramatic.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 42

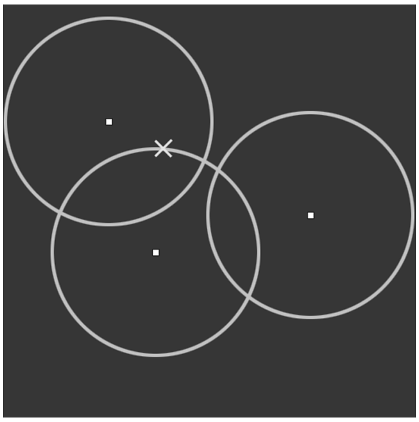

FIGURE 4.24 *An example image with exactly three bright pixels, surrounded by circles showing where their PSF influences overlap. Each PSF radius will contain exactly one minimum, marked with an X in this example.*

these super-bright pixels. The same radius is drawn around all three pixels, creating three overlapping circles. If the PSF were known a priori, we could compute the contribution of these hot pixels at this distance by multiplying the PSF, which is a function of radius, by each hot pixel value. Conversely, dividing the darker pixels around each circle by the circle's center gives an upper bound of the PSF.

Furthermore, the PSF at this distance cannot be greater than the minimum of all darker-pixel/hot-pixel ratios. Where the circles overlap, the sum of hot pixel contributions should be considered. In fact, a convenient approach is to sum all three hot pixel values around each circle in another grayscale image. (This example shows three distinct circles, but in general there are many hot pixels adjacent to each other, which create a great deal of overlap in the contributing annuli.) The PSF upper bound at that radius will then equal the minimum ratio of the darker pixels in the circles to their hot pixel sums (the point marked with an X in the example). This technique extends directly to any number of hot pixels. The estimation procedure is as follow.

1. For each radius we wish to consider:
   a. Sum the hot pixel values into a radius range (annulus) in a separate grayscale image.
   b. Find the minimum ratio of darker-pixel/hot-pixel sum for all annuli.
2. If the minimum ratio is not less than the previous (smaller) radius, discard it (because we assume the PSF is monotonically decreasing).
3. For each minimum ratio pixel, identified for each sample radius, consider all flare contributions to this pixel over the entire image as described below.

Once we have an estimate of the upper limit of the PSF at each radius, these minimum pixels can be used to fit a third-degree polynomial, $p(x)$, using the reciprocal of the input radius for x.[8] For each identified minimum pixel position with value P_i, we can write the following equation.

$$P_i = \sum_j P_j \left(C_0 + \frac{C_1}{r_{ij}} + \frac{C_2}{r_{ij}^2} + \frac{C_3}{r_{ij}^3} \right)$$

Here, the P_js are the contributing pixel values over the rest of the image, and the r_{ij}s are the distances between the minimum pixel P_i and each contributing pixel position. This equation can be rewritten as follows.

$$P_i = C_0 \sum_j P_j + C_1 \sum_j \frac{P_j}{r_{ij}} + C_2 \sum_j \frac{P_j}{r_{ij}^2} + C_3 \sum_j \frac{P_j}{r_{ij}^3}$$

The sums in this equation then become coefficients in a linear system in which the four fitting parameters (C_0 through C_3) are the unknowns. As long as there are more than four minimum pixel values, P_i, it should be possible to solve this as an overdetermined system using standard least squares minimization. Heuristically, a better solution may be obtained if we assign minimum and maximum permitted

8 The use of a third-degree polynomial and fitting to the reciprocal of distance are heuristic choices we have found to produce good results at an economical cost.

values for the distance between pixels, r_{ij}. Anytime the actual distance is less than the minimum radius (3 pixels in the reduced image), we use a distance of 3, instead. Similarly, the distance is clamped to a maximum of half the image width. This avoids stability problems and sensitivity to local features in our image. It also avoids the possibly incorrect removal of flare too close to light sources. This is generally impossible anyway, in that flare from the lens can be so great that the underlying information is washed out. In such cases, no recovery can be made. This often happens at bright source boundaries.

Figure 4.25 compares the point spread function measured in Figure 4.22 to our estimate derived solely from the input image in Figure 4.23. Other than the artificial plateau we imposed by constraining the minimum r_{ij} to 3 pixels, the two curves

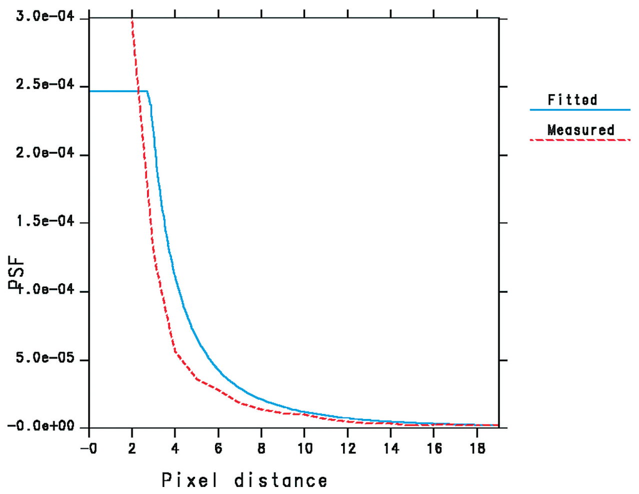

FIGURE 4.25 *Comparison between directly measured PSF from Figure 4.22 and function fitted using image in Figure 4.23.*

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 45

are a reasonable match. The fitted function shows a slightly greater flare than the measured one, but this is explained by the fact that the measurement was based on a spot near the center of the image. Optical flare becomes more pronounced as one moves farther toward the edges of an image, especially in a wide-angle lens. Since the fitting function was applied over the entire image, we would expect the globally estimated PSF to be slightly greater than a PSF measured at the center.

4.8.3 REMOVING THE PSF

Given an estimate of the PSF, flare removal is straightforward. For each hot pixel in the image, we subtract the PSF times this pixel value from its surroundings. Because the neighborhood under consideration may extend all the way to the edge of the image, this can be an expensive operation. Once again, working with a reduced image lowers the computational cost to manageable levels. The steps for removing the PSF are as follows.

1. Create a reduced-resolution flare image, $\mathbf{F}_{\mathrm{CR}}$, and initialize it to black.
2. For each hot pixel in the reduced image $\mathbf{I}_{\mathrm{CR}}$, multiply by the PSF and add the product into $\mathbf{F}_{\mathrm{CR}}$.
3. If the value of any pixel in $\mathbf{F}_{\mathrm{CR}}$ is larger than its corresponding pixel in $\mathbf{I}_{\mathrm{CR}}$, reduce the magnitude of $\mathbf{F}_{\mathrm{CR}}$ uniformly to compensate.
4. Upsample $\mathbf{F}_{\mathrm{CR}}$ using linear interpolation and subtract from the original HDR image.

Step 3 ensures that no negative pixels are generated in the output and is necessary because the fitting method does not guarantee the most conservative PSF. Dependent on the interpolation and the local variance of the original pixels, we may still end up with negative values during Step 4 and should truncate these where they occur. An example result of automatic flare removal is shown in Figure 4.26, along with the reduced resolution flare image generated during Step 2.

4.9 DIRECT CAPTURE OF HDR IMAGERY

With the possible exception of lens flare removal, the techniques explained in the last section might be unnecessary if we had a digital sensor that could record the

FIGURE 4.26 *An image of a rosette window, before flare removal (left) and after (center). The right-hand image shows the estimated PSF applied to hot pixels in the image.*

full dynamic range of a scene in a single shot. In fact, such sensors are being actively developed, and some are even being marketed, but only a few integrated solutions are commercially available: the Autobrite cameras from SMaL Camera Technologies, the SpheroCam HDR panoramic camera from SpheronVR, and the Ladybug spherical camera from Point Grey Research. We will describe each of these systems briefly in this section.

4.9.1 VIPER FILMSTREAM

Grass Valley, a division of Thomson, introduced the Viper FilmStream camera for digital cinematography in the Fall of 2002 (*www.thomsongrassvalley.com/products/cameras/viper/*). This is currently the top-end performer for digital capture, and it produces an enormous amount of data (up to 444 Mbytes/sec!). The camera contains three HDTV 1080i (1,920 × 1,080-resolution) CCD sensors (one each for red, green,

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 47

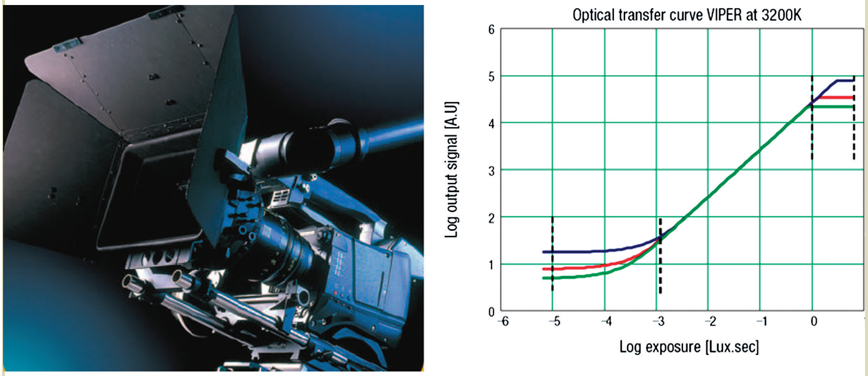

FIGURE 4.27 *The Viper FilmStream camera and its response curve.*

and blue) and records directly into a 10-bit/channel log format. The camera and its response functions are shown in Figure 4.27.

This chart shows that the Viper captures about three orders of magnitude, which is at least 10 times that of a standard digital video camera, and begins to rival film. The equipment is currently available for lease from Thomson.

4.9.2 SMaL

SMaL Camera Technologies of Cambridge, Massachusetts (*www.smalcamera.com*), markets a low-cost VGA-resolution CMOS sensor (the IM-001 Series) which is capable of recording extended-range images at twice video rates (60 fps). Through its unique design, individual pixel sensitivities are adjusted so that the chip captures about twice the dynamic range (in log units) of a standard CCD or CMOS sensor, or about four orders of magnitude. They currently offer two products that incorporate their Autobrite (TM) technology, a credit-card-size still camera, and a video surveillance camera (a prototype is shown in Figure 1.9). They also market a “digital

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 48

FIGURE 4.28 *An HDR image of one of the authors, captured using a SMaL Ultra-Pocket digital camera directly to RGBE format. As we can see in false color, the dynamic range captured in this image is about three orders of magnitude.*

imaging kit" to OEMs and system integrators who wish to incorporate the SMaL sensor in their products.

Figure 4.28 shows an image captured using the SMaL Ultra-Pocket camera. Due to its limited resolution (482 × 642), the SMaL sensor is not well suited to serious photographic applications, but this may change with the introduction of larger Autobrite arrays. Other aspects of chip performance, such as signal-to-noise ratio at each pixel and fill factor, may also affect the applicability of this technology.

4.9.3 PIXIM

Pixim of Mountain View, California (*www.pixim.com*), offers two 720 × 480 CMOS image sensors that boast a 10-bit digital video output with a 95-dB signal-to-noise ratio, corresponding to roughly four orders of magnitude. These sensors grew out of the Programmable Digital Camera Project headed by Abbas El Gamal and Brian Wandell at Stanford University, and employ "multisampling" on picture elements to minimize noise and saturation. Pixels are grouped with independent analog-to-digital converters (ADCs), and sampled multiple times during each video frame.

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 49

Conversion at each sensor group stops when either the frame time is up or the value nears saturation. In effect, each pixel group has its own electronic shutter and dynamic exposure system. For additional processing, the sensor chip is paired with a custom digital image processor, which handles video conversion and control. Pixim currently markets the sensors and development kits to OEMs, and it has been picked up by a few security camera makers. Smartvue (*www.smartvue.com*) has based its S2 line of wireless surveillance cameras on the Pixim chip set, and Baxall (*www.baxall.com*) recently introduced its Hyper-D camera.

4.9.4 SPHERONVR

SpheronVR of Kaiserslautern, Germany (*www.spheron.com*), has what is undeniably the highest-resolution and highest-performance HDR camera in existence; the SpheroCam HDR. This device boasts the ability to capture full spherical panoramas at a resolution of up to $13{,}000 \times 5{,}300$ pixels, covering nearly eight orders of magnitude in dynamic range (a 10^8:1 contrast ratio). However, because they use a line-scan CCD for their capture the process takes from 15 to 30 minutes to complete a full 360-degree scan at this resolution. Lower resolutions and dynamic ranges will scan faster, but one can never achieve a single-shot capture with a line-scan camera because the device must mechanically pan over the scene for its exposure. Nevertheless, this is the system to beat for panoramic capture and critical image-based lighting applications, and their deluxe package comes with an advanced software suite as well. Figure 4.29 shows a SpheroCam HDR image captured in Napa Valley, California, at a resolution of about $3{,}000 \times 2{,}100$. The dynamic range is 5.5 orders of magnitude.

4.9.5 POINT GREY RESEARCH

Point Grey Research, of Vancouver, Canada (*www.ptgrey.com*), recently came out with an upgrade of the SDK for their LadyBug spherical video camera, which enables it to capture six perspective HDR images in a single shot. Five 1/3-inch SVGA sensors ($1{,}024 \times 768$) look in a circle of horizontal directions to obtain a panorama, and a sixth sensor looks straight up, yielding a final image that covers 75% of the full sphere. Data are delivered in real time via a firewire cable to a tethered host computer. Figure 4.30 shows an example HDR result with a resolution of $3{,}600 \times 1{,}500$

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 50

FIGURE 4.29 *An HDR panorama captured by Spheron's SpheroCam HDR line-scan camera, tone mapped using a histogram adjustment operator.*

and a dynamic range in excess of four orders of magnitude. Notably, there is no evidence of ghosting or smearing problems, which one would see if the image were multiply exposed.

4.10 CONCLUSIONS

In the not-too-distant future digital still and motion picture photography may become exclusively HDR. After all, traditional film photography has provided medium-dynamic-range capture for nearly a century, and professionals expect and

reinhard v.2005/03/22 Prn:13/06/2005; 16:03 F:reinhard04.tex; VTEX/JOL p. 51

FIGURE 4.30 *An HDR panorama captured in a single shot using the six CCD sensors of Point Grey Research's LadyBug camera system.*

require this latitude during postproduction (i.e., printing). The current trend toward mixed reality in special effects is also driving the movie industry, which is increasingly digital, toward HDR. Advances in dynamic range will hit the professional markets first and slowly trickle into the semiprofessional price range over a period of years.

Unfortunately, consumers will continue to be limited to LDR digital cameras in the short term, as HDR equipment will be priced out of reach for some years to come. During this interim period, software algorithms such as those described in this chapter will be the most affordable way of obtaining and experimenting with HDR imagery, and applications will hopefully push the market forward.