

USING IMAGE FUSION TO CAPTURE HIGH-DYNAMIC RANGE (HDR) SCENES

“High dynamic range” (HDR) refers to the ability to distinguish details in scenes containing both very bright and relatively dark areas. This can occur in scenes where incident light is present (e.g., imaging a light source and the surrounding area). This can also occur in situations with bright reflections or in high contrast indoor/outdoor scenes where one needs to capture details in both bright sunlight and dark shadows.

HDR image fusion combines two images of the same scene, taken with radically different exposures, into a single image spanning the broadest possible range of light intensities (see Figure 1 below).

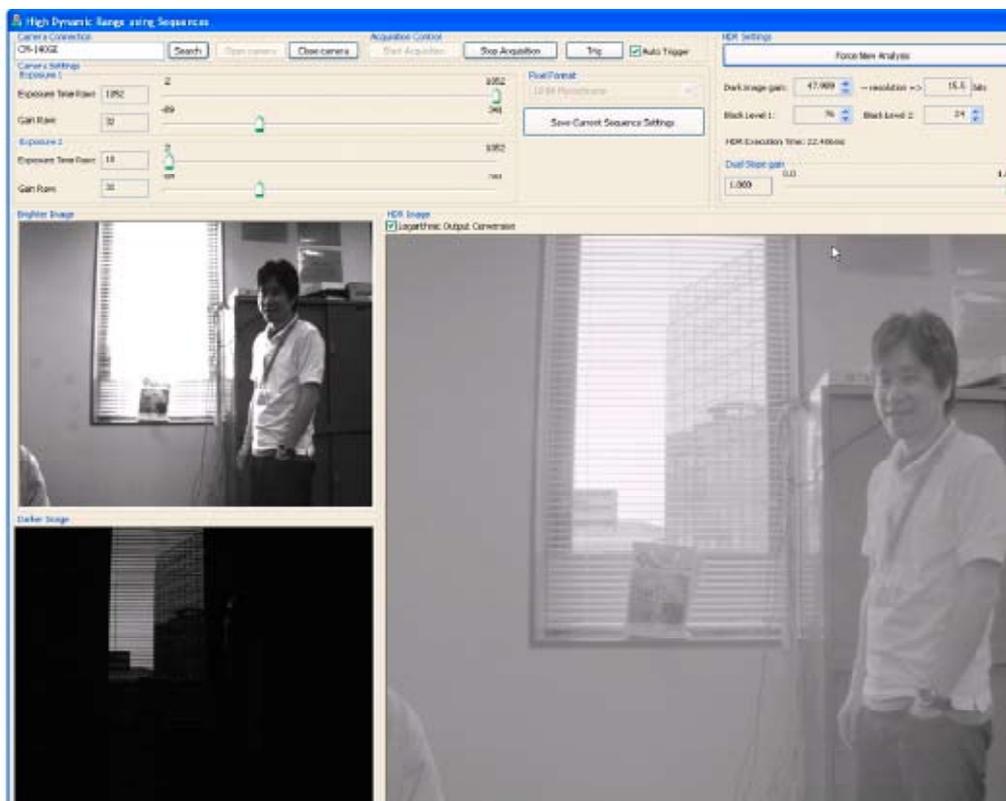


Figure 1 - HDR image fusion example

HDR image capture techniques

There are two basic techniques for capturing the images needed for HDR image fusion.

1) For best results, a 2CCD camera such as JAI's AD-081 is used. This camera has a prism-based design that enables both bright and dark images to be captured simultaneously along a common optical path for crisp HDR imaging of full-motion video (see Figure 2). Of course, because the camera contains two CCDs, and because it is more complex to assemble, 2CCD cameras are more expensive and may have limited speed and/or resolution options.



Figure 2 - 2CCD camera for HDR image fusion (AD-081 series)

2) A second alternative involves the use of a special “Sequence Trigger” function with a standard CCD camera. The Sequence Trigger - which is available on many of JAI’s GigE Vision cameras - enables the camera to be pre-programmed to automatically capture two closely-spaced images with dramatically different gain and/or shutter settings as trigger signals are received. For inspection applications where the object under inspection stops briefly, this approach can provide two perfectly registered images for HDR image fusion. Even in live action scenes, the fusion of Sequence Trigger images can produce remarkably good real-time HDR video in many cases.

Because Sequence Triggering does not require any special optical design, it is a more affordable approach than 2CCD cameras, and can be applied to cameras with a wide range of speed and resolution options.

Sequence triggering explained

As noted, the Sequence Trigger function enables users to pre-program the camera to change its settings automatically after each image is captured (see Figure 3). With JAI’s Sequence Trigger, the settings that can be changed include shutter speed, gain level, and region-of-interest (ROI). Each time a new trigger signal is received, the camera captures a new image using the next group of settings in the sequence. A sequence can include up to 10 different combinations of settings, which are stepped through as each new trigger is received. When the end of the sequence is reached, it repeats again from the beginning.

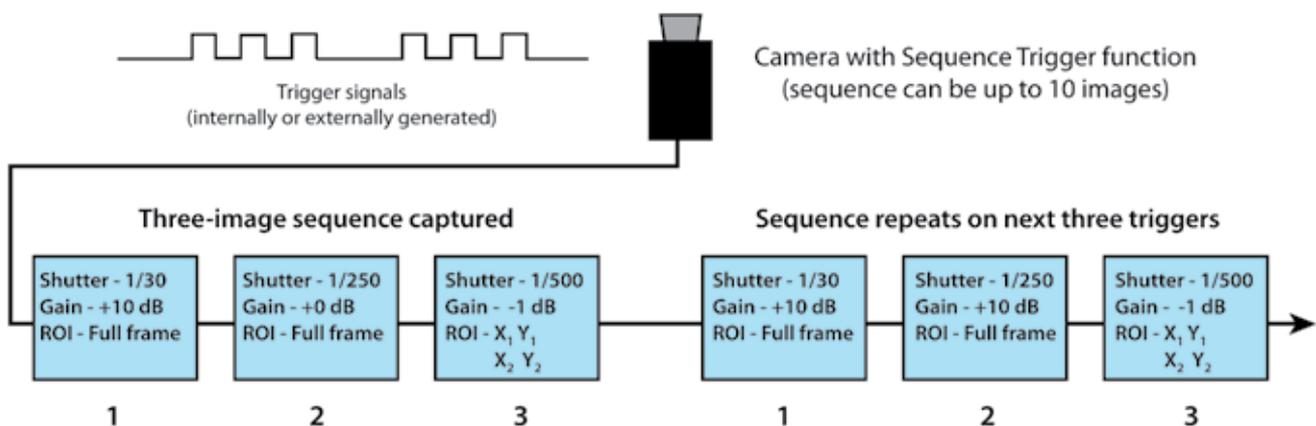


Figure 3 - Sequence Trigger operation

JAI's Sequence Trigger can be used to address situations where a single inspection station must look for multiple defects, each requiring a different gain and/or shutter setting to be properly rendered. Examples include flat panel inspection where the reflective qualities can require different settings to minimize glare or to look for defects below the surface, web inspection of metal rolls where different defects in the material become apparent at different light settings, and printed circuit board inspection where different areas of the board being inspected have significantly different contrasts and reflective properties.

Rather than forcing the user to find a sub-optimal "middle ground" for all the images, the Sequence Trigger mode lets users capture each image with the proper exposure for the area being inspected. Triggers can be generated in response to objects as they pass, or can be used in multi-step inspections where the camera moves over the object in a pre-determined route.

Sequence triggering for high dynamic range

While the previous examples describe cases where each exposure would be analyzed separately, sequence triggering can also be used for HDR image capture and fusion.

To accomplish this, users define a simple two-exposure sequence using the JAI Sequence Trigger. One exposure is defined with a relatively slow shutter speed in order to capture details in the more darkly lit areas of the scene, while the other uses a much faster shutter speed to capture details in the areas that are overexposed in the first image. The style of triggering depends on the specific imaging scenario.

If the object being examined can be made to pause briefly on the inspection line, then asynchronous external triggering can be used to capture the two-image sequence. As the item stops, two consecutive triggers are sent to the camera at an interval equal to or greater than the frame rate of the camera. For example, if the camera has a frame rate of 60 fps, two trigger pulses sent 1/60th of a second apart will cause the camera to capture and output a two-image sequence with different exposures as defined by the two shutter settings.

If, instead, our intent is to use the Sequence Trigger for HDR imaging of a live scene, we can use an internal trigger timed to match the camera's frame rate. By repeatedly generating trigger pulses, the camera can be made to output a continuous stream of image pairs at half the total frame rate of the camera. In other words, on a 0.4-megapixel camera running at 60 fps, a set of two images, ready for HDR image fusion, can be output by the sequence trigger at a rate of 30 pairs per second. Using the high performance image fusion functions included in the JAI SDK, image pairs can then be analyzed and blended into a single high dynamic range image in only a few milliseconds, producing an HDR video stream at nearly the full 30 fps rate.

Of course, as in the first scenario, the second image will be captured 1/60th of a second after the first image. If there is movement involved -- for example, a live video surveillance scenario, a traffic application, or other real-world scene -- the image fusion process must contend with the fact that some items in the second image will have shifted position slightly. In many cases, as it turns out, the Sequence Trigger approach can still provide excellent results, though not as precise as those achieved from the two simultaneous images produced by a 2CCD camera.

To begin with, JAI's HDR software functions are designed to perform image fusion by relying mostly on the pixels from only one of the two images (the "base" image - see the following sections). Only the oversaturated pixels have their values "fused" with the pixels from the second image. Thus, provided the shutter speed on the base image is fast enough to capture a crisp image, the effects of any motion will be limited only to the brightest pixels in the scene.

Furthermore, unless the brightest objects in the scene are moving extremely rapidly relative to the camera's optical axis, there's a good chance that they won't have shifted more than a few pixels between frames. This is especially true if the second image in the sequence is the one with the faster shutter speed and is therefore completely captured at the very start of the second frame. Thus, when a region of saturated pixels from the first image are fused with their counterparts from the darker second image, most of the details will still be displayed, with only a slight spatial shift and some darkness on the trailing edge.

For many applications, this is more than sufficient for HDR viewing or analysis, but in cases where absolute pixel precision is required, a 2CCD solution is still recommended.

HDR “fusion” functions and the JAI SDK

Once pairs of exposures are being produced - either by the Sequence Trigger or a 2CCD camera - the HDR image fusion process can be performed by special functions included in the JAI GigE Vision Software Development Kit (SDK). The simplest method is to use the sample application provided with the JAI SDK. Two versions are available - one for 2CCD cameras and the other for single CCD cameras using the Sequence Trigger mode. In addition, users desiring a more customized approach can create their own HDR image fusion application by accessing the underlying functions themselves. Documentation for the functions is included in the JAI SDK.

JAI’s sample HDR image fusion application enables users to define the exposure values for the light and dark images in order to best capture the full dynamic range of the scene. Depending on whether 8-bit or 10-bit output has been selected, HDR video with up to 20-bits of dynamic range (120 dB) can be generated by mathematically combining information from the two images as shown in Figure 4. The JAI sample application automatically analyzes the relationship between the two exposures to calculate the proper calibration factor to be used as it replaces oversaturated pixels in the base image with information from the darker exposure. For a more detailed discussion, see Appendix A.

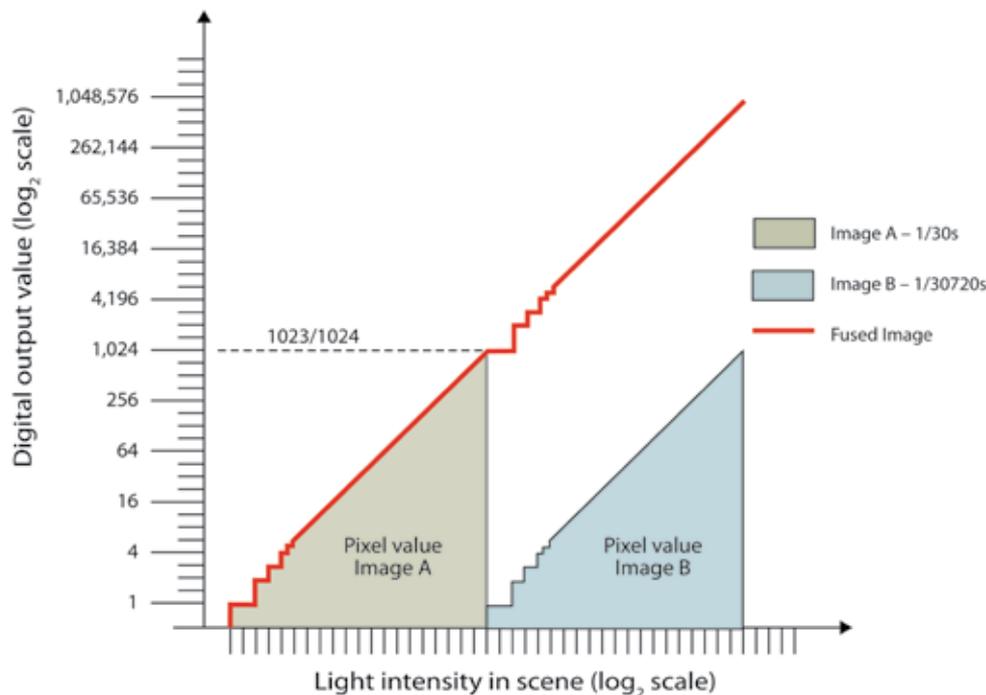


Figure 4 - Image fusion, maximum dynamic range

For an HDR image without any “gaps” in the intensity information, the maximum ratio between the two exposures is 2^{10} (1,024x) in the case of 10-bit output and 2^8 (256x) in the case of 8-bit output. This is illustrated by the red “fused image” line in Figure 4. When a less than maximum ratio is used, the JAI sample application automatically “overlaps” the image information from the two exposures, again using only the relevant information from the second image to fill in details in the oversaturated pixels of the base image (see Figure 5).

In both cases, the image fusion algorithm calculates a complete set of 16-bit values for every pixel in the image. This “linear” data can be saved and used for accurate computer-based analysis of the HDR information in the image.

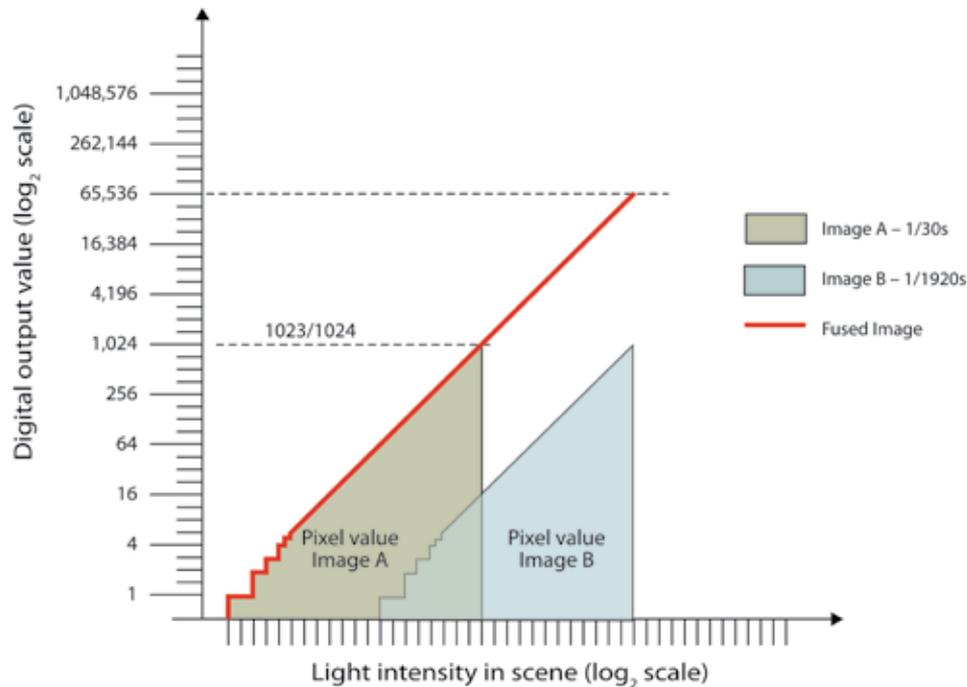


Figure 5 - Image fusion with overlap

This is in contrast to the typical situation with specialized CMOS sensors used for high dynamic range imaging. These sensors often boast the ability to handle situations with dynamic ranges of 16-bits or higher, but they do so on chips that may only support 10-bits or 12-bits of information. They achieve this by using specialized algorithms that convert from linear pixel values to logarithmic calculations as pixel values near saturation. While this enables the sensor to effectively “compress” the brightest pixel information into a smaller total number of pixel values, unless the exact algorithm is known by the user it can make it very difficult to “reverse engineer” the actual pixel values for accurate linear comparisons or analysis (see Figure 6).

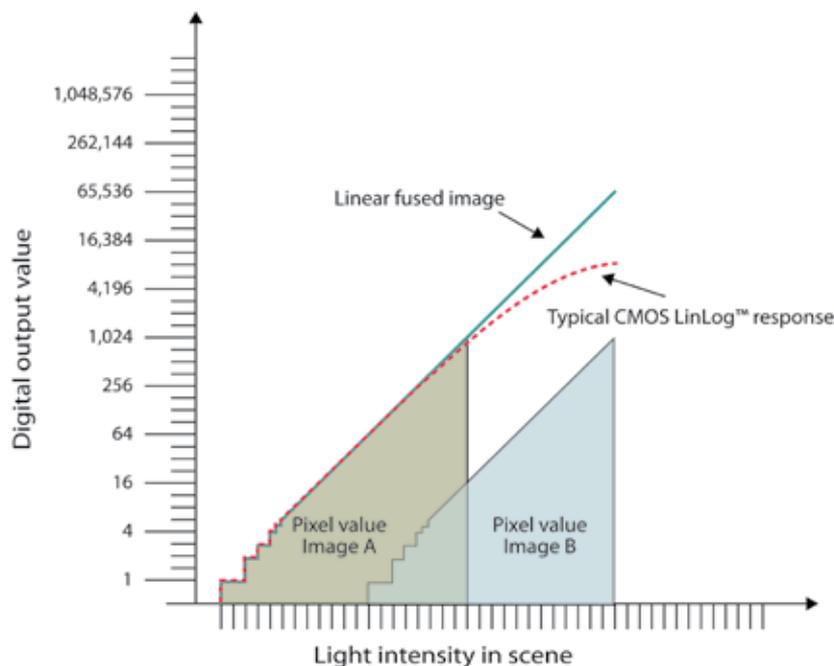


Figure 6 - Linear HDR image fusion vs. linear/logarithmic compression

Displaying the HDR image

One issue with any high dynamic range approach is that it is hard to display such an image on a standard monitor. While the underlying 16- or 20-bit linear pixel values can be used for computer analysis of an HDR scene, they cannot be displayed on a standard monitor without “compressing” the information to fit the bit depth of the monitor and display application. Standard monitors still only support 8-bit images, and even though newer monitors may have contrast ratios capable of supporting up to 12-bits of dynamic range, the actual display application may only support 8-bit image data.

Simple scaling of the HDR data into 8-bit data for display typically over-darkens the image due to the extreme gap between the brightest and darkest pixels. JAI’s sample HDR image fusion application utilizes a sophisticated two-step process whereby raw values are first converted into their base 2 logarithms, then are scaled to fit the depth of the display (see Appendix B for a more detailed discussion).

This approach preserves the raw values for high precision in machine vision processing, while reducing the amount of compression applied to the “lowlights” in the image. The result, as shown in Figure 1, tends to be a better visual approximation of the high dynamic range data for most applications. However, depending on the light intensities that are of greatest interest, users can develop their own mapping routines to produce different results.

Color HDR images

The preceding sections have focused on monochrome image fusion, however it is equally possible to produce HDR color images using the same methods as described here. The new HDR functions and sample application provided with the JAI SDK can automatically perform image fusion on two raw Bayer images produced using the Sequence Trigger method. Since these are simply monochrome images prior to interpolation, the same HDR image fusion technique is used to compensate for oversaturated pixels in the base image, regardless of whether those pixels contain red, green, or blue information in the Bayer mosaic. Once interpolation is performed, the result is a color HDR image (Figure 7).

As with monochrome images, movement in the scene will cause some slight imaging issues in areas with oversaturated pixels. Again, by using relatively fast frame rates, these issues can be virtually eliminated producing live-motion color images with 90-120 dB of dynamic range and clarity equal to or beyond that of traditional video output. Or, in the case of stop-action inspections, the result is an HDR color image with pixel perfect precision.

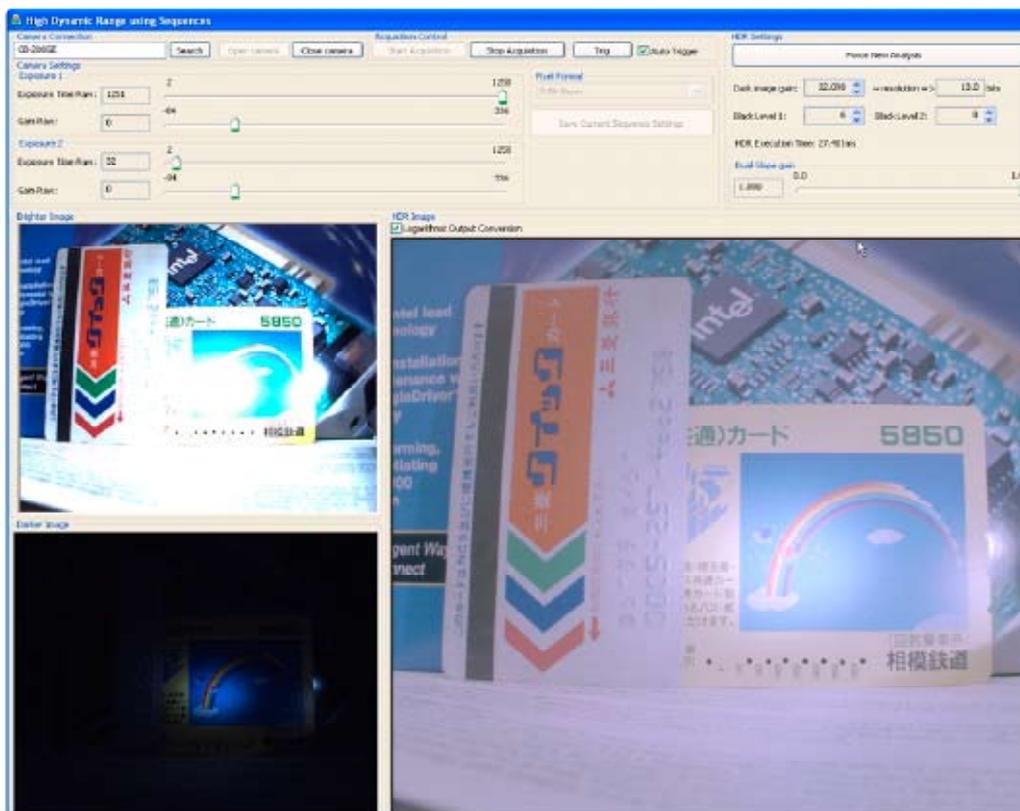


Figure 7 - Color HDR image fusion

As with any color output, white balancing is recommended to achieve the best color rendition. In this case, the white balancing is performed on the HDR video stream, after image fusion and color interpolation has been performed. Standard white balancing techniques can be used on the HDR output.

For more information about high dynamic range imaging, the JAI SDK, or Sequence Trigger mode, please contact JAI.

NOTE: HDR functions are included with JAI GigE Vision SDK and Control Tool v1.2.5 and above.

APPENDIX A - IMAGE FUSION ALGORITHMS

Although the JAI GigE Vision SDK contains predefined functions for image fusion, some users may want to experiment with their own image fusion routines. The routines built by JAI are based on understanding the ratio between the shutter speeds used to capture the image pairs.

For example, if 10-bit monochrome output is being used, an image with up to 20-bits of dynamic range (~120 dB) can be created by setting the shutter speed of Image B to be 2^{10} (1,024) times the shutter speed of Image A. In other words, if Image A is set with a shutter speed of 1/30 sec., Image B would need to be set as close as possible to 1/30720 sec. using the camera's pre-set shutter or programmable exposure control. This would result in 1 count of output on Image B being roughly equivalent to what would be 1024 counts on Image A, had it not saturated at 1023 counts.

Our fused HDR image is created by applying a post processing routine that uses output from Image A when it is below saturation and from Image B when Image A is saturated. A simplified representation of this routine could be the Boolean expression:

```
if (pixel B < 1){
    pixel_out = pixel A
}else{
    pixel_out = pixel B * 1024
}
```

This approach uses Image B to add 10 more bits of dynamic range to the image as shown in Figure 4. If 8-bit output is used, the calibration factor between the two shutter speeds becomes 2^8 or 256. The maximum dynamic range in this case is 16 bits.

While the previous example produces the maximum linear dynamic range, it may also result in some issues around the 1023/1024 count transition that cause problems in the fused image. This is because of the fast shutter speed being used on Sensor B and the relatively low output precision (i.e., 1 count = 1024 while 2 counts = 2048). This means that the inherent noise in Sensor B has a much more noticeable effect, causing some pixels that are very close in actual light intensity to be output with dramatically different values. While this type of impact is expected in the darkest portions of an image, its effect on luminance values around the transition point between our two images can result in some very noticeable artifacts.

For many high dynamic range scenes, a better approach is to use shutter speeds that don't stretch the dynamic range to the maximum. By setting the shutter speeds so that the two images overlap by 2-4 bits, the total dynamic range is reduced, but so is the amplification of noise at the transition point to provide a better overall image throughout the full range.

For example, to produce a cleaner transition with 10-bit output, set the shutter on Image B to be 64 times faster than Image A. Now the 4 MSB of Image A will overlap with the 4 LSB of Image B (see Figure 5) and, when mathematically fused, will create a total linear dynamic range of 16 bits.

Now, our post processing routine could be handled as follows:

```
if (pixel B < 16){
    pixel_out = pixel A
}else{
    pixel_out = pixel B * 64
}
```

By overlapping the two images, our 16-bit HDR image utilizes the full precision of the lower 10-bits while reducing the effect of noise at the transition point and greatly increasing the precision (or smoothness) of the upper 6-bits.

APPENDIX B - MAPPING TO 8-BIT DISPLAYS

A simple way to map the raw pixel data into an 8-bit display is to multiply all the pixel values by a scaling factor equal to 255 divided by the maximum pixel value. In our 20-bit example, this means each value is multiplied by a factor of 255/1,048,575. Unfortunately, because of the “log scale” nature of the pixel values, this approach causes all the Image A values to be compressed into the lowest 4 bits of the display, causing a significant “darkening” of the image details that virtually eliminates the expected visual appearance of the high dynamic range image.

To compensate for this, one can convert the raw pixel values into their base 2 logarithms (floating point values) before calculating the scaling factor. Thus, in our 20-bit example, the floating point values from Image A would fall between 0.0 and 10.0 (i.e., 2^{10}), while the values from Image B would fall between 10.0 and 20.0 (2^{20}). These could then be mapped into 8-bit integer display values using a scaling factor of 255/20. Pseudocode for this might look like:

For all raw pixel values{

```
    pixel_display = Math.Log(pixel_out, 2.0) //Convert to log-2 values
}
ScaleFactor = 255 / 20 //Set a scale factor based on the maximum log-2 value
For all pixel display values{
    pixel_display = (pixel_display * ScaleFactor)
}
```

This approach reduces the compression on the Image A data to preserve most of the details of the “low-lights” in the image. Highlight information from Image B is then added only in the upper values of the 8-bit image. The result, as shown in Figure 1, tends to be a better visual approximation of the high dynamic range data for most applications, however, depending on the light intensities that are of greatest interest, different mapping routines may produce better results.

JAI has added several functions to the JAI SDK software to simplify the process of developing and customizing an HDR application using either Sequence Triggering or a 2CCD camera. In addition, a sample application is provided with the JAI SDK offering a turn-key HDR application. Consult the software documentation for more details regarding how to use the sample application or how to utilize the HDR functions in implementing your own.