# Multispectral, high dynamic range, time domain continuous imaging

**Henry Dietz, Paul Eberhart, Clark Demaree;**
**Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky**

## Abstract

*Time domain continuous imaging (TDCI) models scene appearance as a set of continuous waveforms, each recording how the value of an individual pixel changes over time. When a set of timestamped still images is converted into a TDCI stream, pixel value change records are created based on when the pixel value becomes more different from the previous value than the value error model classifies as noise. Virtual exposures may then be rendered from the TDCI stream data for arbitrary time intervals by integrating the area under the pixel value waveforms. Using conventional cameras, multispectral and high dynamic range imaging both involve combining multiple exposures; the needed variations in exposure and/or spectral filtering generally skew the time periods represented by the component exposures or compromise capture quality in other ways. This paper describes a simple approach in which converting the image data to a TDCI representation is used to support generation of a higher-quality fusion of the separate captures.*

## Introduction

The concept of capturing an image has long been entangled with the idea of dividing perceived reality into temporal intervals. A conventional photographic image represents the scene appearance averaged over the exposure time interval. The shutter is opened, light energy is collected, the shutter is closed, and the charge collected is processed to create an image. Even video is modeled as a sequence of images.

### Traditional HDR Capture

In that context, consider the tonal range faithfully recorded in an image captured during a single conventional exposure. The tonal range represented is not really determined by the scene, but by the range of brightnesses (luminances) that can be properly recorded by the sensor during the exposure time interval and preserved through processing. Processing parameters such as the analog and digital gains applied to implement various ISO sensitivities can alter that range, but the sensor's ability to record widely varying brightness is fundamentally bounded on the high end by well capacity and on the low end by quantum efficiency and noise. The classic ways to capture significantly extended tonality – a high dynamic range (HDR) image – involve combining multiple images captured with significantly different exposure parameters[3].

It is possible to construct a camera using multiple conventional sensors to capture HDR during a single exposure interval. For example, to avoid parallax, one or more beam splitters could



**Figure 1.** *Four cameras, one viewpoint via obscura in FourSee*

allow a single lens to light multiple sensors with the same view. Alternatively, the beam splitters could be replaced by a camera obscura arrangement in which multiple secondary lenses image the view projected by a single master lens. For example, Figure 1 shows FourSee, an example of the obscura approach in which four Canon PowerShot N cameras capture the image projected on a screen by the central large-format lens. However, these arrangements are awkward.

There is also the issue of how to give different portions of the scene's tonal range to the different sensors. Varying ISO tends to corrupt the tonal properties[2]. Varying aperture alters the captured depth-of-field, giving structures within the scene inconsistent appearances across captures (conceptually nearly the same problem as dealing with parallax). Perhaps the best method would be to impose a different neutral-density filter on each sensor, but that only allows extending the capturable tonal range into highlights – it cannot make more detail visible in the darkest shadow areas. Varying shutter speed would be most flexible, but then the images being combined for HDR do not represent the same time period.

### Traditional Multispectral Capture

Although the goal in multispectral imaging is quite different from that in HDR capture, the problems are similar. While HDR seeks to increase tonal resolution, multispectral aims to more precisely distinguish different wavelengths of light. Con-
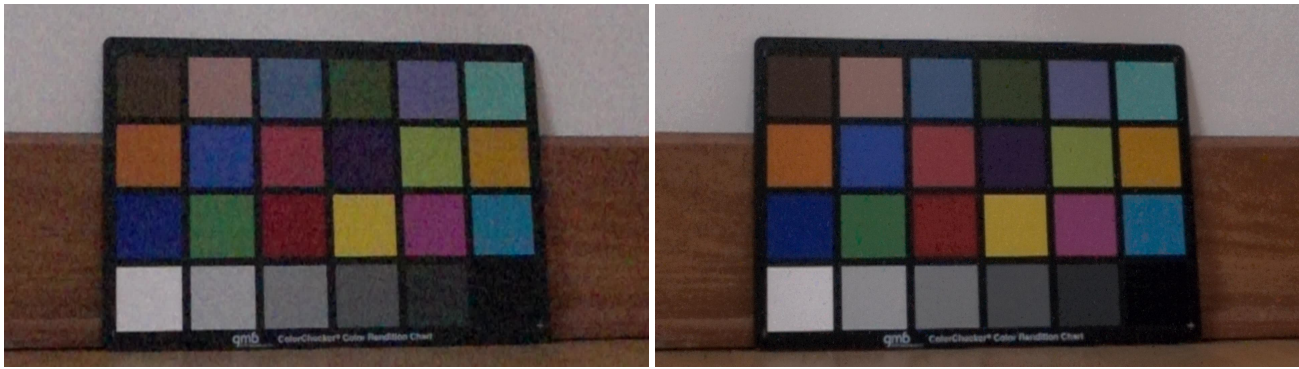
**Figure 2.** *Signal-to-noise ratio is significantly improved over 960FPS frame by 1/960s virtual exposure using* `tik` *TDCI[7]*

ventional CMOS and CCD image sensor technologies (as opposed to stacked designs, as used by Foveon[4]) do not distinguish the wavelengths of photons within the visible spectrum. There may be some variability in the quantum efficiency depending on wavelength, but the charge is accumulated without differentiating the contributions by wavelength. The sensitivity of pixels to different wavelengths is thus modulated by imposing color filters. Consumer cameras typically impose a color filter array (CFA) on the sensor so that a repeating spatial pattern of red, green, and blue sensitivities tiles the captured image, and from this data colors are interpolated. This approach works well, but is it really multispectral?

It is useful to recognize that the red, green, and blue filters used in CFAs are colors, but they are not really spectral bands. Instead of narrow-wavelength filters passing only red, green, and blue photons, each CFA filter color has wavelength-varying optical density described by a spectral sensitivity curve that undulates across the visible spectrum. In general, the relationship between colors and wavelengths is very complex[5]. For example, adding red (wavelength in 620-750nm) and green (495-570nm) light yields a color that is seen as yellow, but might not actually produce any photons with a wavelength around that of yellow (around 570-590nm). Thus, if the goal is to reliably classify how much luminance is in each relatively narrow wavelength range, it will be necessary to use more than just the three CFA filters.

Some consumer cameras have used four color CFAs, but having many colors in the CFA (an MCFA[6]) would require custom CFAs and would divide spatial resolution. There is also the issue that some filters cut light more than others, so exposing for one CFA color can compromise the dynamic range in the other colors. Thus, multispectral capture must resort to the same types of multi-sensor or multi-shot processing used in HDR capture: either using multiple sensors with different filters or a sequence of exposures with one camera taken as a sequence of filters rotate in front of the lens.

### *A Matter Of Time*

In sum, the basic problem with both HDR and multispectral capture is that requiring multiple exposures to align temporally makes capture problematic. The solution presented here is to **use conventional cameras without requiring the actual exposures to temporally align, but to computationally align timing of virtual exposures after capture**. Not only does Time Domain Continuous Imaging (TDCI) naturally support this, but it has the side benefit of significantly improving signal to noise ratio in each virtual exposure. An example of this is shown in Figure 2, where an improvement of close to three stops is achieved because the TDCI virtual exposure was able to conservatively recognize which pixels were changing only due to noise and thus use image data from before and/or after the time period represented by the virtual exposure to produce more accurate estimates of brightness within that time interval.

The following section briefly reviews `tik`[7], the temporal imaging software from Kentucky that implements TDCI using image sequences from one or more cameras. In effect, `tik` allows image sequences to be temporally aligned so that HDR and multispectral processing can done on the virtual exposures produced without concern for temporal sampling issues. The sections after that discuss methods for HDR and multispectral imaging leveraging TDCI. The last two sections describe a real-world experiment we performed and present conclusions and directions for future work.

## TDCI and tik

The fundamental concept behind TDCI and `tik` is the idea that scene appearance generally changes in relatively slow, continuous, ways. Conventional imaging methods represent a time-sequence of image data as a sequence of frames, each representing average scene appearance over the exposure interval in which is was captured. In contrast, TDCI assumes that individual pixels usually maintain their values over relatively long sampling periods. Only when a pixel's value has changed significantly is a pixel-value-change record created.

Clearly, if significant changes are rare, the sequence of change records can be much smaller than the pixel data for all images. However, compression is not the primary benefit. The TDCI form temporally interpolates across samples, thus allowing pixel values to be estimated for arbitrary time intervals – not just for the original frame exposure intervals. In fact, the interpolation can even be performed in ways that achieve temporal superresolution[8].

Of course, the fundamental problem is determining when a

change in a pixel's value is significant. In `tik`, this is done by constructing a detailed pixel value error model. The error in a pixel's value can come from any of a variety of fairly complex sources, including both noise sources within the camera and photon shot noise from the lighting sources. The `tik` software is able to use a user-supplied error model, but it also can automatically construct an error model. The model is empirically derived by directly measuring variations in a sequence of captures of a completely static scene, constructing an image that describes a probability density function encoding the probability that a pixel with a particular value, $P_v$, should actually have had each of the other possible values.

When a sequence of images is encoded as a TDCI stream by `tik`, only when a pixel's value deviates from expected by more than the pixel value error model predicts is a pixel value change record produced. Smaller changes are absorbed into refining the estimate of the true value of the pixel during that interval – and this pixel-level stacking-like behavior is how signal to noise ratio improves. The sequence of value change records for each pixel effectively defines a smooth curve or waveform specifying how the brightness of that pixel changed as a function of time.

Given a TDCI stream, `tik` is capable of rendering virtual exposures for any time intervals covered by the stream data. The requested virtual exposure interval need not be related to the original frame capture timing in any particular way – the curves are continuous, so estimates can be computed even for time periods entirely between original captures. This is done by, independently for each pixel, integrating the area under that pixel's brightness curve for the requested time period. The integrated values are normally then scaled to preserve the full dynamic range. In addition to the improvement in signal to noise that accompanied conversion to TDCI stream format, relatively long virtual exposures can benefit from combining longer sections of the waveform; when a particular pixel in a virtual exposure spans multiple value change records, `tik` is essentially performing a weighted averaging of values that tends to further reduce noise.

Thus, the easiest way to use `tik` for multispectral HDR imaging is:

1. Capture multiple exposures, generously covering the time interval of interest, with the parameters needed to permit HDR and/or multispectral information to be extracted; the timing of the individual exposures can be interleaved, synchronized, random, etc. – the sole requirement is that each capture is tagged with the time interval it represents
2. The captures are sorted into groups by type, and within type into earliest-to-latest order
3. The `tik` software is used on each group separately to produce a TDCI representation
4. For each time interval desired as a multispectral/HDR image, use `tik` to extract a virtual exposure from each of the relevant TDCI streams
5. Perform conventional multispectral/HDR merging of the virtual exposures

For example, suppose an HDR sequence has one camera capturing a sequence of 4 second exposures while another is capturing 1/8 second exposures at 2.5FPS. The frames don't line up temporally. However, the sequence of 4-second exposures can be transformed into a TDCI stream and the 1/8-second exposures into a second. If the goal is to produce an HDR image representing the time interval from 6.25 to 6.75 seconds from the start of capture, one would use `tik` to render virtual 1/2-second exposures starting at 6.25 seconds using each of the two TDCI streams. The final HDR image is then the result of ordinary HDR processing of the two virtual exposures.

## HDR

The basic concept of HDR capture is to acquire multiple images with different exposure parameters so that each can capture a different segment of the scene's complete brightness range. The brightness ranges captured should at least leave no gaps in covering the scene's content, but ideally should overlap by an amount sufficient to:

1. Ensure good tonal quality: recall that most camera sensors are approximately linearly sensitive to light. Thus, a camera capturing 12-bit pixel values representing a 12-stop dynamic range is using values from 2048 to 4095 solely for representing the brightest stop, while the darkest stop might be represented by a single value. Overlap should be sufficient to ensure that all portions of the dynamic range have fine enough brightness resolution to show no obvious artifacting.
2. Allow proper alignment: if the brightness ranges captured do not make any of the same image content visible in multiple images, there is no image basis for computationally correcting for any misalignment.

Largely because the dynamic range of digital cameras had been relatively small compared to that of film, various methods for capturing and processing HDR have become well developed. Despite some consumer cameras now boasting 14-stop dynamic range, which is greater than most films and even greater than instantaneous human eyesight, there is no fundamental limit on how large the brightness range in a scene can be – so HDR continues to be useful. In fact, most consumer cameras actually have built-in HDR modes.

### HDR Exposure

Fundamentally, HDR exposure consists of capturing the scene with a variety of shutter speeds while holding all other exposure parameters constant.

From when light metering was less reliable, it became common for cameras to incorporate "exposure bracketing" modes that can automatically capture a sequence of exposures differing by a single parameter – shutter speed. These bracketing modes typically allow up to three or five shots to be captured in a burst with exposures differing by up to about 3-5 stops. Thus, a compact camera that normally captures only about 9 stops of dynamic range might be able capture as much as a roughly 20-stop range using bracketing. Multi-shot modes intended for HDR often implement bracketing sequences containing as many as 7 exposures to collect data for a single HDR image.

### HDR Merging and Tone Mapping

The raw sensor data captured by most consumer camera sensors approximates linear sensitivity to photons through most of the dynamic range captured. In cases where the image data is nonlinear due to sensor, processing, or use of non-unit gamma (e.g., JPEG files used in place of raw sensor data), linearization can be performed using empirically-determined camera calibration. Working on linear image data, merging data from multiple exposures that differ only in shutter speed can be accomplished by multiplicatively scaling pixel values according to the difference in shutter speed. For example, if the shutter speed is half as long, the linear pixel values should be doubled.

Given the simplicity of merging capture data, it would be feasible to directly implement intelligent merging into the TDCI encoding process. Rather than merging the entire captured sequence as though it came from a single period in time, the timing of each image capture can be used to incrementally update pixel data. As the captured frame sequence is converted into TDCI pixel value change records, the following logic could be applied separately to each pixel:

- If the current frame provides a new value for this pixel which is within the linear portion of the current frame's dynamic range, that value is multiplicatively scaled and treated normally by the TDCI conversion algorithm. There are two possibilities:
    1. If the new pixel value is within the modeled value error bounds of the value in that pixel's current value change record, the value is updated by weighted averaging without creating a new value change record.
    2. If the new value differs significantly, then a new pixel value change record is produced.

- If the current frame pixel value is unreliable, there also are two possible cases:
    1. If the value predicted by that pixel's value change record would be in the same unreliable portion of the current frame's dynamic range, the value change record is extended to assume the expected value continued through the frame's time interval.
    2. If the value predicted by that pixel's value change record is not in the same unreliable portion of the current frame's dynamic range, it is clear that a new value change record should be emitted, but there is no reliable value to place in it. Thus, a value near the previous value, but in the unreliable range, would be guessed, and the new value change record marked as having that unreliable value. If a subsequent frame provides a reliable value that is in the same region indicated by this frame's unreliable data, the guessed value is then replaced by the value from that subsequent frame; otherwise, the guess is left intact in the pixel's value change record.

The latest version of the `tik` software internally uses linearized floating-point pixel values that could easily support the type of HDR merging described above. However, it does not yet support output of a TDCI stream in a format that would preserve the high dynamic range, nor does it support rendering virtual exposures in any of the HDR still image formats. Although `tik` is able to map a larger tonal range to a smaller one for output of internal HDR data in a lower dynamic range output format (e.g., JPEG), it does not do this in a sophisticated way; a method like gradient domain HDR compression[11] would produce far higher quality results. Thus, the best available option is to use `tik` to generate a low dynamic range TDCI stream for each temporal sequence of similar-range exposures, and then to perform standard HDR merges on the set of low dynamic range virtual exposures rendered from all TDCI streams for the same time interval.

## Multispectral

The key principle in multispectral imaging is that the value of a pixel is the sum of the contributions made by photons at all wavelengths. These contributions are weighted by a spectral response curve that defines the probability that a photon with a particular wavelength contributes a unit of charge to the sum.

### Gel Filters

High-quality photographic filters with a variety of precisely-specified spectral properties can be expensive and difficult to find. However, Rosco is a company producing a wide range of lighting and related equipment for stage production. Because modern theatrical lighting uses a variety of halogen, fluorescent, arc, and white LED lamps, it becomes difficult for lighting designers to predict the combined effect of a light source and filter gel without considering the spectral profile of each. To this end, Rosco produces not only inexpensive gel filter swatch books (such as the Roscolux used here), but also publishes a detailed spectral energy distribution curve for each filter[9].

Each gel profile lists the percentage transmission of light from 360-740nm wavelength in twenty 20nm steps. For example, consider Roscolux R88 Light Green, Roscolux R99 Chocolate, and Roscolux R4290 CalColor 90 Blue (equivalent to a CC90B, and informally described as "enhances blue by three stops"). The precise spectral profiles of these three filters are as shown in Figure 3. Using the gel spectral data, it is possible to select a set of filters that will allow recovery of the brightness of any scene object in each of the twenty bands.

### The Math

Let $P_v$ be the value read from the sensor for a particular pixel in linear gamma units. Let us further assume that the pixel is sensitive exclusively to photons in the spectrum for which the filters have been characterized. Using consumer cameras in natural lighting, the relevant spectrum could be as broad as 300-1200nm. Unfortunately, the filters and color reference target used in the current work have published calibration only within 360-740nm; however, that range is sufficient to cover the bulk of photons contributing to $P_v$ within our artificially-lit test scenes.

The goal in multispectral imaging is simply to determine the contribution in each spectral sub-band to $P_v$. Let $P_{center}$ represent what would have been the unfiltered contribution of photons in the sub-band centered at a wavelength of $center$nm to the total value of the pixel. Given filters with transmission characterized
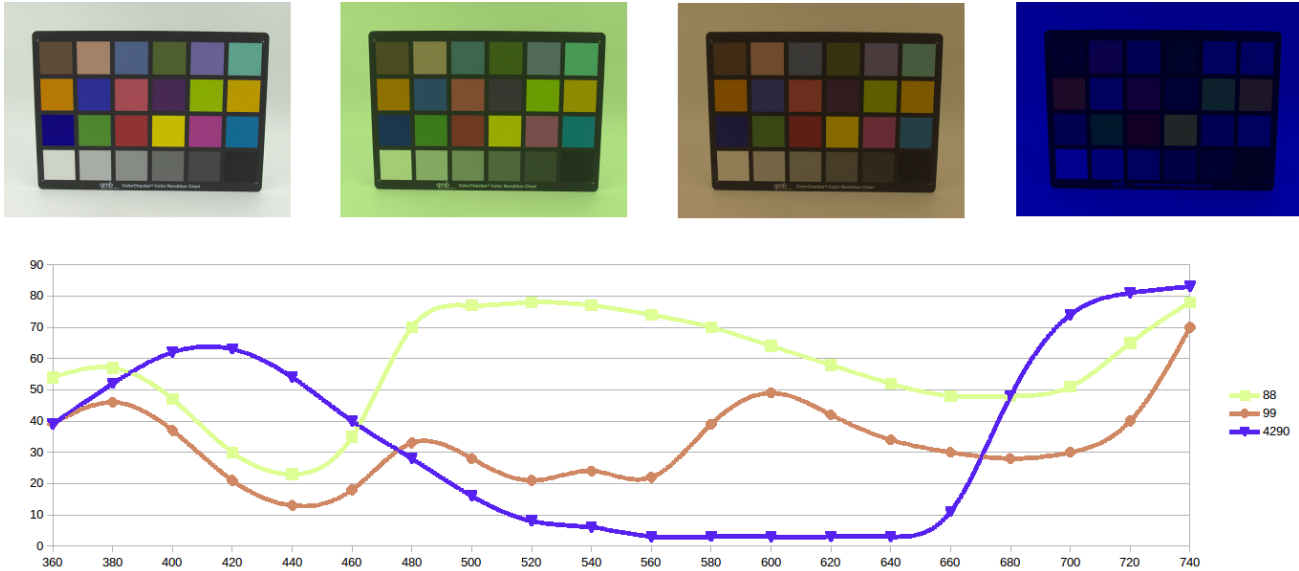
**Figure 3.** *CHDK raw captures and spectral energy plots for a few Roscolux filters using a Canon PowerShot SX530 HS*

in 20nm sub-bands, let us call the values $P_{360}$, $P_{380}$, ... $P_{740}$.

Suppose that filter $F$ is applied. In each relevant sub-band, the filter has a transmittance which is between 0% and 100%, i.e., a value between 0 and 1. Let $F_{center}$ represent the average transmittance for the filter in the sub-band centered at *center*nm. For example, the filter's spectral profile provided by Rosco directly provides the transmittance in each 20nm sub-band: $F_{360}$, $F_{380}$, ... $F_{740}$. The result is the very straightforward equation:

$$P_v = P_{360} \times F_{360} + P_{380} \times F_{380} + ... + P_{740} \times F_{740}$$

This single equation leads to a highly ambiguous set of possible solutions: there are twenty unknowns. A fully-constrained solution would require a system of twenty independent equations.

The CFAs used in most cameras provide either three or four filters in a single capture. Bayer CFAs use a repeating $2 \times 2$ pattern of {red, green, green, blue}. This provides either three or four filters – four if the two greens differ significantly in spectrum, which is not uncommon. Other four-color patterns in consumer cameras include {green, magenta, cyan, yellow} (GMCY, as used in the Canon PowerShot G1), {red, green, blue, emerald} (RGBE, as used in the Sony F828), and {red, green, blue, white} (RGBW, with variants proposed by Kodak). The white filter is actually clear, intended primarily to enhance low-light sensitivity.

By making multiple captures using additional filters, it is possible to greatly increase the number of equations. Each new capture with an external filter produces either three or four new equations. The catch is that each such set of equations contains the same CFA contributions, so these sets of equations are not fully independent. Independence is needed to resolve ambiguity, but even highly correlated samples can have the beneficial effect of reducing noise. Still, to be able to reliably recover a 20 sub-

band multispectral image will take at least 17 captures and will produce a system of 68 equations.

### Gaussian Elimination Solver

Given a set of equations with low noise, it is not difficult to solve for contributions in each sub-band. For example, in 2001, our research group realized that the images produced by a Canon PowerShot G1 were subject to significant NIR contamination – wide open, the lens did not focus NIR light in exactly the same plane as visible light, so the NIR contamination appeared largely in the form of purple fringing of objects backlit by strong daylight. The goal was thus to use the camera's GMYC CFA filters to resolve {red, green, blue, near infrared} (RGBI) color channels.

The obvious method for extracting {red, green, blue} (RGB) color channels from GMCY samples is simple differencing:

$$G = G; R = Y - G; B = C - G; R = M - B; B = M - R;$$

Somewhat better results can be obtained by constructing a set of linear equations describing the spectral profile (as suggested above) and using simple Gaussian elimination to solve the system. However, there is a slight complication in that there are four equations in three unknowns when solving GMCY for RGB. In `dcraw`[10], the conversion to RGB is implemented by averaging the weightings obtained by solving for RGB using each of the possible three-color subsets of GMCY: GMC, GMY, GCY, and MCY. The solution thus obtained was:

$$R = -2.40 \times G + 3.54 \times M + -2.52 \times C + 3.42 \times Y$$

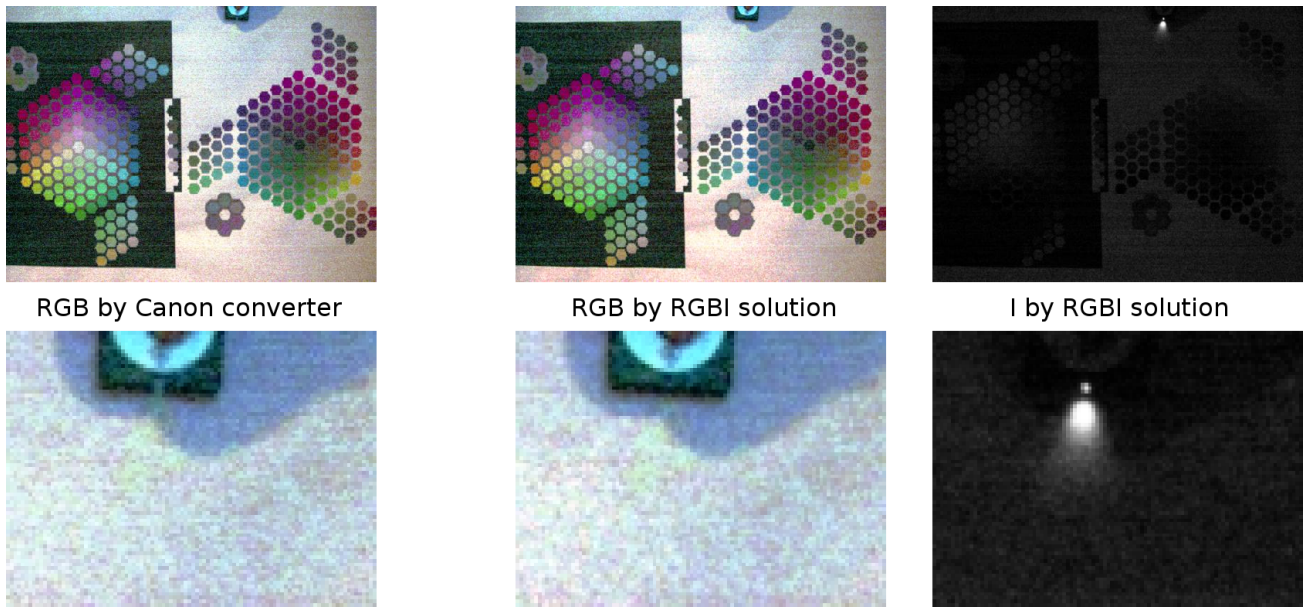$$G = 4.01 \times G + -1.71 \times M + 0.69 \times C + 0.42 \times Y$$

**Figure 4.** *Canon PowerShot G1 raw: GMYC conversion to RGB by Canon reference software, RGB and I by RGBI solution by Gaussian elimination*

$$B = -2.35 \times G + 3.39 \times M + 3.52 \times C + -2.25 \times Y$$

However, a true multispectral treatment offers significant improvement. by measuring the near infrared (I) sensitivity and solving the GMYC equations for RGBI weightings, the following were obtained:

$$R = -1.38 \times G + 0.25 \times M + -0.16 \times C + 1.29 \times Y$$

$$G = -0.27 \times G + -1.46 \times M + 0.93 \times C + 0.81 \times Y$$

$$B = -1.42 \times G + 0.17 \times M + 1.38 \times C + -0.097 \times Y$$

$$I = 15.1 \times G + 12.2 \times M + -9.04 \times C + -8.38 \times Y$$

Although these solutions are perhaps somewhat surprising, they do produce the desired spectral differentiation. Figure 4 shows the same raw capture processed by Canon's raw converter into RGB and using the above RGBI solution to obtain RGB and I renderings. The crops show the camera's near-infrared remote control, in which the reference-processed RGB image shows significant contamination from the near-infrared LED. Quality of these images is poor because the near-infrared cut-off filter in the camera required very dim visible lighting to bring both the brightness of the visible spectrum and the near-infrared LED within the camera's usable dynamic range; near infrared cut-off filters should be removed from the sensor stack to obtain better-balanced sensitivity.

### Noise-Tolerant Genetic Algorithm Solver

While simple Gaussian elimination is efficient and effective for solving small systems of independent equations, solving a noisy, partially correlated, set of 68 equations for 20 unknowns requires a more robust approach. The noise means that often no set of values that satisfy all equations will exist, and that is sufficient to cause many standard approaches for solving systems of linear equations to fail. The approach used here instead employs a genetic algorithm to search for the best possible solution in terms of minimizing error squared.

A genetic algorithm (GA) is a search procedure using simulated evolution. Rather than maintaining a single best guess at the solution, an entire population of potential solutions is maintained throughout the search. Initially, the population consists of vectors of randomly generated $P_{center}$ values.

The fitness of each population member is evaluated by inserting its $P_{center}$ values into each equation to compute an estimate of that equation's $P_v$ that we will call $P_{est}$. The $(P_v - P_{est})^2$ for all equations are then added to produce a fitness metric for which the smallest value is the most fit solution.

Many GAs are implemented such that a generation of potential solutions is evaluated a group at a time. This GA is instead implemented using the "steady state" model, in which each pass of the GA removes a single less-fit individual from the population to replace it with a new individual derived from other, more fit, individuals. The population members to be involved in the death/birth processing are randomly selected, and the least-fit individual among those selected is chosen as the victim to replace. The replacement is created either by mutation or by crossover.

Crossover models sexual reproduction, the creation of a new individual by mixing genetic material from two (or more) parent individuals. There are many ways to implement this mixing, often modeling genetic mechanisms by treating the individuals as bit strings and splicing them much as DNA is spliced. However, the goal is to tend to maintain properties of the parents so that the offspring might inherit the best features and thus exceed the fitness of its parents – treating a set of floating-point numbers as a bit string is not very effective. Instead, the crossover opera-

tor used here is essentially averaging the corresponding floating-point $P_{center}$ values from two parents. However, literally averaging the values would have the highly undesirable side-effect of decreasing diversity in the population, which could result in quickly converging on a solution that is not the global minimum. To avoid this, the averaging performed over two parents, $Y$ and $Z$, to compute a new value for population member $X$, performs the following computation in which $random(a,b)$ returns a random value in the interval $[a,b]$:

$$\delta = |Y.P_{center} - Z.P_{center}|$$

$$avg = (Y.P_{center} + Z.P_{center})/2$$

$$X.P_{center} = random(avg - \delta, avg + \delta)$$

This technique has served well in GAs our group has built for a variety of other purposes, and it appears to function well here.

The other method used to create new population members is mutation. Crossover generally is more likely to produce a superior offspring than random mutation, but the mutation operation used here is not entirely random. To begin, the new population member, $X$, is created by duplicating a better population member, $Y$. Then each equation in $X$ is evaluated to determine which is the greatest source of error. If we temporarily ignore the other equations, there are multiple trivial ways in which $X.P_{center}$ values could be adjusted to result in 0 error for this equation. The GA randomly picks between two different methods. The first method simply scales all the $X.P_{center}$ values by $P_v/P_{est}$; the second picks a single $X.P_{center}$ value and adjusts it.

If the new population member is more fit than the previously most fit, it is recorded as the best so far. The genetic search continues until the allotted run time has elapsed or the search has proceeded for a specified maximum time without recording a new most fit solution. The best found by that time is output as the final solution.

Search speed averaged 439,881 potential solutions per second to solve a system of 68 equations in 20 unknowns using the C-coded GA compiled by GCC and run on a single core of an Intel Core i7-4500U CPU @1.80GHz. Figure 5 summarizes the accuracy of the GA solutions given noisy pixel value measurements over 400 synthetic test cases. For realistic pixel noise levels (e.g., between 5 and 12 bits correct), the over-specified system quickly and consistently produces results with a higher signal to noise ratio. As pixel noise levels become very low, the maximum allowed runtime for the GA must be increased to maintain solution quality; the 100 second limit used here proved insufficient to consistently recover values accurate to more than 14 bits.

## A Real-World Experiment

Although multispectral and HDR imaging can be useful for many purposes, it is relatively rare that both multispectral and HDR are simultaneously needed. The August 21, 2017 solar eclipse seemed like an ideal opportunity to test both together – obtaining multispectral detail in the Sun's surface and other details moving through the dim light of totality. It was thus decided to build camera arrays to capture TDCI streams of the eclipse from each of two viewing locations. The Princeton, KY airport
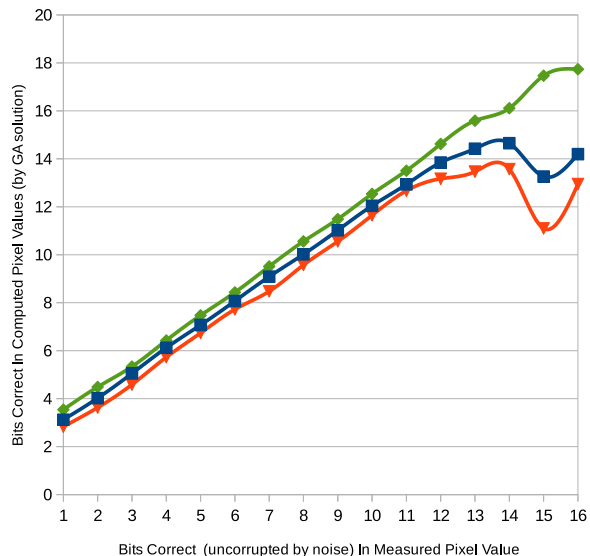
**Figure 5.** Quality bounds for GA solutions to 68 equations in 20 bands

was an ideal location near the center of the path of totality. Given the predictions of traffic and crowds in the region of totality, the 95.1% partial visible from outside our laboratory at the University of Kentucky's campus in Lexington, KY made it our secondary site.

### Equipment

The ability to use CHDK[12] to capture raw sensor data, reprogram, and synchronize Canon PowerShot cameras makes them excellent components for camera arrays. However, to obtain a relatively high resolution image of the Sun or Moon requires a lens with full-frame equivalent focal length of at least 400mm. The Canon PowerShot SX530 HS seemed the clearly best choice. It is a 16MP compact superzoom camera with a 50X zoom range from 24-1200mm, is (mostly) supported by CHDK, and we were able to purchase a refurb fleet of them from Canon for $130 each.

Rather than building two large arrays, we built a set of five more easily portable four-camera arrays for capturing multispectral HDR. Each MASK (Multicamera Array Solar from Kentucky) composed of four Canon PowerShot SX530 HS cameras mounted on a wooden rail with USB synchronization via a switched hub mounted on one end of the rail. The MASKs were named red, yellow, green, blue, and purple, and the rails were stained those colors to make them easily distinguished. Figure 6 shows the purple MASK in a field at the Princeton, KY airport.

Each MASK was given a different set of multispectral or HDR capture tasks, with correspondingly different filters used on each. Except during totality, it was necessary to use a special solar filter because of the Sun's brightness. We made custom filters using AstroSolar Safety Film, using 3D-printed holders that fit the bayonet on the front of the SX530 HS cameras (and others to fit the filter screw threads on other cameras). We detailed the

**Figure 6.** *The purple MASK at Princeton, KY*



**Figure 7.** *Roscolux and solar filters with 3D-printed holders*



**Figure 8.** *SX530 HS images of the eclipse*

complete filter creation process in an Instructable[13] so that others could easily build their own. Of course, during totality these solar filters need to be removed. Multispectral filtering was also done using filters produced in the same way, but using Roscolux gel filter material, as shown in Figure 7. A second type of filter holder was also created so that solar filters could be stacked on the gel filters yet easily removed at totality.

## Results

The peak totality was at 13:24:55 in Princeton, KY and the partial in Lexington, KY peaked at 14:30:25. All camera arrays were set-up in the early morning to give time to adjust. Weather was clear, but awkwardly warm without shade. In fact, at the airport, a few of the 3D-printed filter holders deformed slightly due to the heat before being mounted on a camera.

We were able to capture many images of the eclipse, two of which are shown in Figure 8, but only about 10% of the images captured were usable – which was not sufficient to produce the high-quality multispectral HDR TDCI we had hoped. As a result, our primary experiment did not produce sufficient data to confirm nor deny the expected benefits of our approach. Prob-

lems included:

- Alignment is absolutely critical at 1200mm and manually aiming the cameras proved untennable. Not only did we have too few people to have one dedicated to keeping each MASK tracking the eclipse, but the SX530 HS cameras do not have an electronic viewfinder. With the cameras pointed nearly straight up, the rear LCD was awkward to view and brightly reflected the light-colored and heavily patterned ground (a problem we had not experienced when we tested on campus).
- Our tripods were not sufficiently stable. Left alone, they worked fine – but they would ring with every adjustment of aim. Similarly, although the wooden bar was solid enough, each individual SX530 HS was shimmed to obtain perfect alignment and the shims slipped a tiny bit each time the array was aimed.
- Although stacking a solar filter in front of a gel filter had worked in our earlier trials photographing the Sun, during the eclipse the off-center Sun often caused reflections between the gel filter and the shiny back side of the solar filter in front of it. These flare patterns sometimes completely

overwhelmed the image. Due to a bug in the CHDK software, these reflections also sometimes caused the camera to refocus, often resulting in the camera producing seriously defocused images.

## Conclusions And Future Work

This paper has presented a new approach to multispectral HDR imaging based on the use of TDCI to enhance the image quality and provide a mechanism for precise temporal alignment of image data captured with arbitrary timing skews. The approach is described in some detail, including a novel GA for processing multispectral data and practical configuration of low-cost capture systems.

Unfortunately, what was to have been our ultimate experimental validation – producing a multispectral HDR sequence of images of the August 21, 2017 solar eclipse – did not produce conclusive data due to unforeseen implementation issues discussed above. All the problems that crippled our eclipse experiment now have fixes or workarounds. More solid mounting combined with use of remote live view would solve the first two problems. We also have a team of undergraduates working to create an inexpensive automatic camera alignment system. Mounting the solar and gel filters together in a single holder, combined with revised CHDK software, can dramatically reduce the flare problem and eliminate the focus problem.

The next opportunity to photograph a total solar eclipse in our area is April 8, 2024, so we are looking at other ways to provide real-world validation of the approach. Less extreme experiments have produced results consistent with the expectations voiced in this paper. In particular, we have conducted some very preliminary tests involving multispectral HDR imaging of field crops and use of consumer drones. We hope to test this paper's approach with a variety of field experiments, including surveying local crops during the 2018 growing season.

## Acknowledgments

## References

[1] Henry Gordon Dietz, Frameless, time domain continuous image capture, *Proc. SPIE 9022, Image Sensors and Imaging Systems 2014, 902207 (March 4, 2014); doi:10.1117/12.2040016.* (2014)

[2] Henry Gordon Dietz and Paul Selegue Eberhart, ISO-less?, *Proc. SPIE 9404, Digital Photography XI, 9404L (February 27, 2015); doi:10.1117/12.2080168.* (2015)

[3] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*, Morgan Kaufmann. (2010)

[4] Paul M. Hubel, John Liu, and Rudolph J. Guttosch, Spatial frequency response of color image sensors: Bayer color filters and Foveon X3, *Proc. SPIE, vol. 5301, pp. 402-407.* (2004)

[5] J.J. McCann, Human color perception. Color Theory and Imaging Systems, *Society of Photographic Scientists and Engineers*, R. Eynard, ed., Washington, pp.1-23. (1973)

[6] Raju Shrestha, Jon Yngve Hardeberg, and Rahat Khan, Spatial arrangement of color filter array for multispectral image acquisition, *Proc. SPIE 7875, Sensors, Cameras, and Systems for Industrial, Scientific, and Consumer Applications XII, 787503 (17 February 2011).* (2011)

[7] Henry Dietz, Paul Eberhart, John Fike, Katie Long, Clark Demaree, and Jong Wu, TIK: a time domain continuous imaging testbed using conventional still images and video, *Electronic Imaging, Digital Photography and Mobile Imaging XIII, pp. 58-65(8).* (2017)

[8] Henry Dietz, Paul Eberhart, John Fike, Katie Long, and Clark Demaree, Temporal super-resolution for time domain continuous imaging, *Electronic Imaging, Computational Imaging XV, pp. 87-93(7).* (2017)

[9] Rosco, Mycolor Desktop, *http://us.rosco.com/en/mycolor.* (2018)

[10] Dave Coffin, Decoding raw digital photos in Linux, *http://www.cybercom.net/d̄coffin/dcraw/.* (2016)

[11] Raanan Fattal, Dani Lischinski, and Michael Werman, Gradient Domain High Dynamic Range Compression, *ACM Trans. Graph., vol. 21, no. 3, pp. 249-256; doi:10.1145/566654.566573.* (2002)

[12] Canon Hack Development Kit (CHDK), *http://chdk.wikia.com/wiki/CHDK.* (2018)

[13] Henry Dietz, Safely Shooting The Sun With The Canon Powershot SX530 HS, *http://www.instructables.com/id/Safely-Shooting-the-Sun-With-the-Canon-PowerShot-S/.* (2017)

## Author Biography

*Henry (Hank) Dietz is a Professor in the Electrical and Computer Engineering Department of the University of Kentucky. He and the student co-authors of this paper, Paul Eberhart, and Clark Demaree have been working to make Time Domain Continuous Image capture and processing practical. See `Aggregate.Org` for more information about their research on TDCI and a wide range of computer engineering topics.*