Color Calibration for Arrays of Inexpensive Image Sensors

Master's with Distinction in Research Report

Neel S. Joshi
March 2004
Stanford University
Department of Computer Science

Abstract

The recent emergence of inexpensive image sensors has enabled the construction of large arrays of cameras for computer graphics and computer vision applications. These inexpensive image sensors have inconsistent color responses. These inconsistencies can cause significant errors in color sensitive multi-camera applications. We present an automated, robust system for calibrating large arrays of image sensors to achieve significantly improved color consistency. We acquire images of a Macbeth color checker placed in the scene and perform gain and offset calibration on each individual sensor. This process combined with a global correction step maximizes the response range by maximizing contrast and minimizing the black level and ensures linear response that is white balanced for the scene. We present results with data acquired from 45, 52, and 95-camera arrays calibrated both indoors and outdoors for a variety of color-sensitive applications including high-speed video, matted synthetic aperture photography, and multi-camera optical flow.

Acknowledgements

This work was done primarily with Bennett Wilburn, Professor Marc Levoy, and Professor Mark Horowitz. I would like to thank all of them very much for their insight and guidance. I would also like to thank the rest of the Levoy group, particularly Vaibhav Vaish, for additional advice and assistance.

I must also thank my family and friends for their endless support and for keeping me sane over the past few years. It's been fun.

Contents

Chapter 1

Introduction

Figure 1: Color variations across image sensors. These two images were taken at the same time from adjacent cameras under typical lab light levels. The cameras are made with the same CMOS image sensor. The settings affecting color balance and color gain are identical, yet there is a significant perceptible color difference between these two images.

Researchers have investigated a number of techniques that use multiple images for a variety of applications in computer graphics and vision. Techiniques such as light field rendering [Levoy and Hanrahan 1996], high-dynamic range photography [Debevec and Malik 1997], and optical-flow [Black and Anandan 1993] are traditionally implemented by acquiring multiple images from a single camera. Using a single translating camera to capture data from multiple views, limits these applications to static scenes. As image sensors have become smaller, cheaper, and more powerful, researchers have begun to use large numbers of video cameras to extend these applications to dynamic scenes. These now commodity image sensors give researchers a significant amount of flexibility that has allowed them to build on previous techniques and has allowed them to tackle new research challenges. Virtualized Reality [Rander et al. 1997] and its successor, the 3D-Room, [Kanade et al. 1998], are two large arrays that have explored the power of using multiple cameras for a variety of graphics and vision applications. Camera arrays have been used for real-time application such as MIT's distributed light field array [Yang et al. 2002]. The Stanford Light Field Camera Array [Wilburn et al. 2002] has been used for a number of applications including high-speed video [Wilburn et al. 2004], synthetic aperture photography [Vaish et al. 2004], and spatiotemporal view interpolation [Wilburn et al. 2004].

Along with the increased flexibility and power resulting from using inexpensive cameras, there are a number of hurdles. Image sensors are often designed to accurately represent relative color differences but are not designed to represent absolute color; for most single-camera applications this is acceptable. In certain multi-camera applications where images are combined from multiple sensors these inconsistencies cause large artifacts in the resulting images. As researchers begin to apply single-camera techniques to data acquired from camera arrays, consistent color response becomes critical. Figure 1 illustrates the inconsistent color response that is seen with our inexpensive video cameras.

Multi-camera applications combine images in various ways depending on the goal of the applications. For some of our applications we combine images by interleaving entire images in a sequence or by pasting together sections from multiple images. For other applications we apply computational methods on images from multiple cameras. For all of these applications it is necessary to adjust the cameras to have similar color responses. Otherwise when images are pasted together there will be perceptible "seams" between image sections and our computational methods will fail because brightness consistency assumptions are violated by inter-camera color inconsistencies. The process of ensuring inter-camera color consistency is referred to as radiometric or color calibration and it involves adjusting the on-camera controls and processing the camera's image data so that all cameras respond similarly when imaging a scene.

Color calibration is a challenge as image sensors have many sources of error that need to be accounted for. For example, image sensors often have non-linearity at the extremes of their range. In addition, many image sensor's on-board image processing introduces additional errors. A calibration process should calibrate each camera so that it gives a consistent linear response across each color channel, where the data for the imaged scene saturates or clips as little as possible. The process should be robust to non-linearity in the sensor and should be able to handle a variety of lighting conditions. For large arrays, the system should be fast and automatic and require little human intervention. The result of the calibration process should produce images with both small perceptual color differences and small absolute numerical differences.

We show a system that is completely automated that robustly, efficiently, and accurately calibrates a large number of cameras to a known desired response curve. We calibrate the brightness and contrast of our cameras using a Macbeth color checker with a novel method that provides robustness to non-linearity in the sensor response curve. We correct for non-uniform illumination on our color chart and use a simple calibration system to automatically detect corresponding points on our color target. By implementing part of our pipeline with the processor on each camera board, we compute image statistics at high-speed and in parallel for all cameras. For a final global correction step we use a floating-point gain and offset correction, a look-up table re-mapping, and a 3x3 transform to further reduce error. Figure 2 shows a block-diagram overview of our calibration process. This process will be described in detail in Chapter 3. Our calibration system enables us to produce high quality results with various graphics and vision

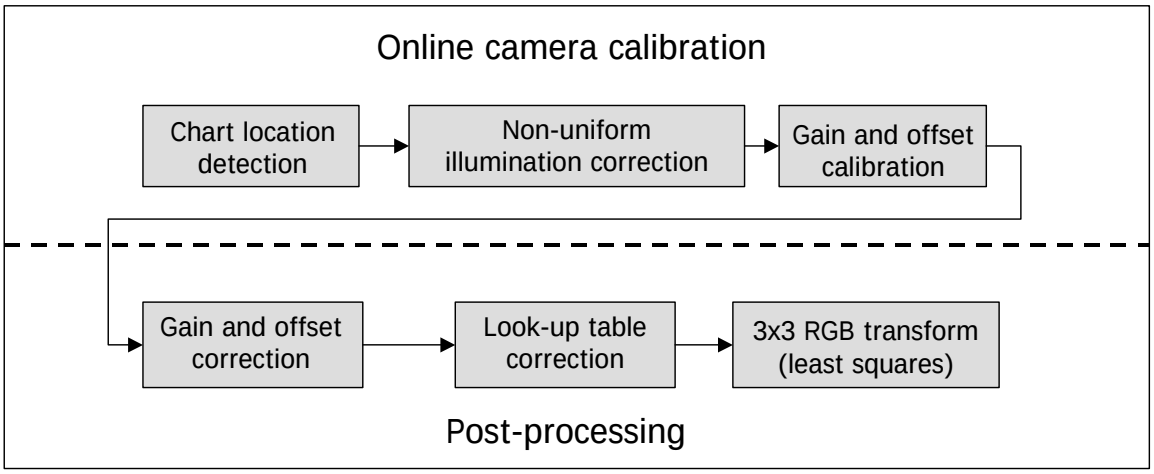


Figure 2: A diagram showing the multiple stages of our color calibration pipeline. There are "online" steps completed before acquiring data. The "online" process includes adjusting camera settings based on sampled values from a Macbeth color checker. The "post-processing" steps include corrections computed from uploaded images of the color checker that are then applied to the acquired data.

applications. We will show three color-sensitive applications that benefit from this type of color calibration: high-speed video, synthetic aperture photography with matting, and multi-camera optical flow. We have not addressed producing true-color output, as this is unnecessary for our targeted applications, although existing techniques in this area could be applied to the final stage of our processing pipeline.

1.1. Related Work

Researchers have studied the importance of color calibration for single camera systems [Barnard and Funt 1999] and [Grossberg and Nayar 2002]. This work has shown that it is possible to calibrate a single camera well using either scene statistics or images of color charts. With the increasing availability of low-cost projector systems, there has been work in color calibration for achieving uniformity across tiled projector displays [Majumder et al. 2000]; however, there has been little work in applying color calibration techniques to multi-camera systems. For certain multi-camera systems such as the 3D-room [Vedula 2001] at CMU, color calibration has been ignored. For the RingCam, an omni-directional camera used for generating panoramas, [Nanda and Cutler 2001] designed a color calibration system that uses image statistics to calibrate color response. The brightness value is calibrated by acquiring a "black" scene at zero exposure and then adjusting the brightness control on the camera so the mean intensity is some desired "black value". The contrast is adjusted by changing the gain such that the mean intensity of the image is some desired "mean brightness" value, which they default to 127; their system allows this target value to be user specified. The scene is white balanced by adjusting the red and blue gain settings on the camera to make the amount of green, blue and red in the scene equal. This can be done using the mean intensity of the images for each color channel or the user can select a "white" area to be used. The cameras are

calibrated to each other by adjusting gain and offset settings to match overlapping regions of the cameras' views.

The RingCam system was designed for real-time calibration to handle changing light levels and for a camera setup with partially overlapping views. For generating real-time panoramas this calibration procedure produces good results; however, this application and setup is very different from the light field acquisitions we are concerned with. In generating panoramas they were able to use blending to ease perceptual color differences when transitioning between images from two cameras. We want to composite images without blending as we have multiple image seams. Refer to Figures 8 through 10 for an example of this type of image composition. Since blending is not an option, our applications demand better color calibration. The real time distributed light field camera, [Yang et al. 2002], applied the RingCam calibration system for calibration of their light field camera array. It is unclear how color-sensitive their application was, but we have found that applying this technique to our light-field setup does not produce acceptable results with our applications.

The RingCam system has several weaknesses that cause it to be unsuitable for our needs. It is scene-based and requires the user to pick a good target value for the image mean. Although the cameras' views do overlap partially, there are large non-overlapping regions near the extremes. Calibrating using image statistics when each camera is not looking at exactly the same scene is error prone. We will show that the calibration process is very sensitive to the target image mean and differences in non-overlapping views of the scene. Another weakness of this method is its reliance on dark images for black-level calibration. A common approach for acquiring these images with a large number of cameras is to set the camera exposure to zero. For our particular image sensors we have found that even with the camera exposure at zero, some light is integrated. Placing lens caps on cameras is time consuming and tedious for 100 cameras and it is difficult to do without disturbing the focus setting of the lenses. Blocking out light with black felt has the same problems, as it is undesirable to rest anything on the camera lenses once they have been focused and aimed. In sunny outdoor settings a much more opaque covering is needed to block out all light. Our method allows calibration to proceed without dealing with these difficulties.

Several post-processing techniques produce high-quality results when applied to uncalibrated cameras. [Porikli and Divakaran 2003] successfully use a correlation modeling function to post-process images. We have opted for a set of much simpler techniques that are better suited to a large number of cameras. A correlation modeling function requires computing pair-wise modeling functions for matching color response. For a large number of cameras, computation increases on the order of n^2. Since n may be large in our application, we seek a calibration method whose computation cost does not grow quadratically with n.

Chapter 2

Framework

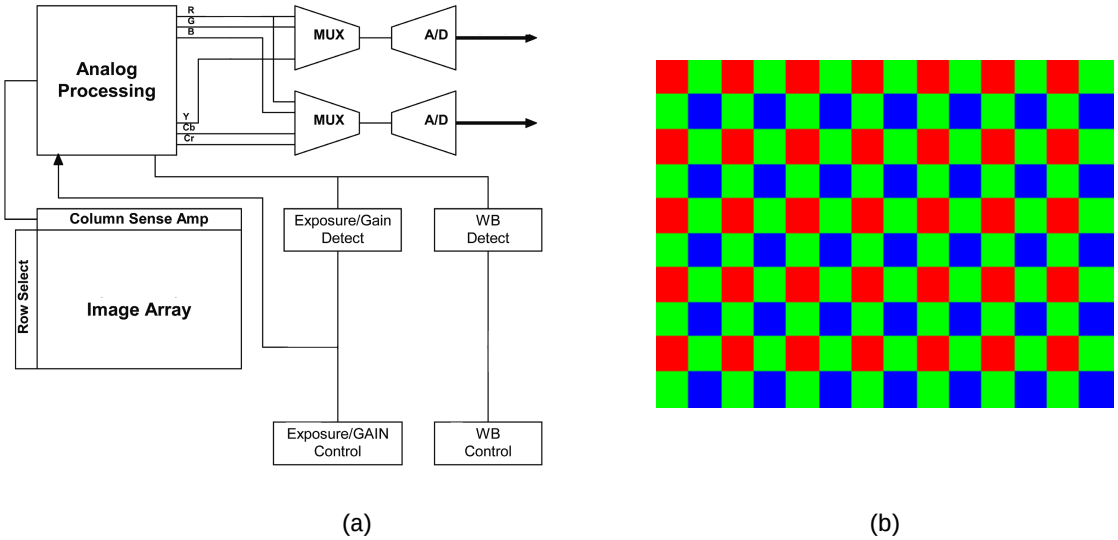


Figure 3: The image sensor. (a) A simplified diagram showing the processing pipeline in our Omnivision CMOS image sensor. All processing is done in the analog domain. Inaccuracy in the analog circuitry causes most of the color inconsistencies between image sensors. (b) The pattern of color filters covering the image array known as the Bayer mosaic.

2.1 CMOS Image Sensor Overview

We will give a brief overview of the image-processing pipeline on our Omnivision sensor to illustrate the types of errors that are introduced by the electronics and processing on the chip. Figure 3a shows the processing pipeline for our CMOS image sensor. While the analysis and calibration procedure we present was designed for our particular sensor, many image sensors have a similar structure and exhibit the same types of errors. The image sensor consists of an array of photodiodes covered by color filters arranged in what is known as a Bayer mosaic, as shown in Figure 3b. The accumulated charge is fed though a series of analog amplifiers and other analog circuitry before it is digitized. Most CMOS video cameras, including ours, have some amount of on-board image processing

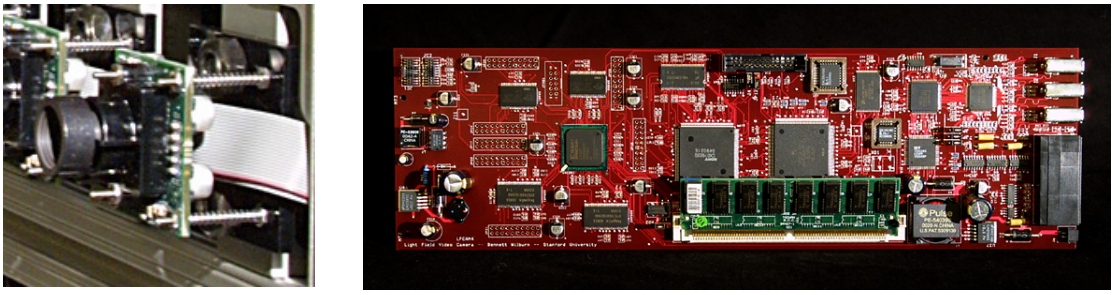


Figure 4: Our imaging hardware. Left: a close up of one of our cameras using an Omnivision CMOS image sensor. Right: a custom image processing board that includes a Motorola Coldfire processor.

to demosaic the raw sensor output and to do a RGB to YCbCr conversion, which is used for MPEG encoding the camera data. These cameras also have automatic image processing features for automatic gain, white-balance, and exposure control. Our camera also exposes an interface for manually setting color channel gains and offsets.

Our CMOS image sensor shows significant color response differences even when the parameters controlling the image processing steps are identical across different sensors. For our image sensors we have seen significant errors in both the raw sensor output and the final demosaiced YUV output. The errors in the raw sensor output break down into three categories: non-linearity in the sensor response, gain and offset setting inaccuracies isolated to each color channel, and inaccuracies due to cross-channel effects.

It's unclear what exactly causes the non-linearity within our sensor. We have seen that there is a slight non-linearity in the center of the sensor response curve in addition to significant non-linearity in the extremes of the curve. We have also found that our sensor seems to internally saturate at a level less that than its maximum output value. The camera manufacturer claims that the sensor output is normalized to a range of (16, 240) although the specifications are unclear on how or why this is done. We suspect this process is partly responsible for non-linearity at the extremes.

The errors in the gain and offset settings are due to a number of causes. On our sensor there is inaccuracy in the application of the gain and offset settings in addition to a limited amount of precision as they can only be adjusted by discreet quantities. The offset settings offer reasonable precision while gain setting has a somewhat large step size giving only coarse control. In addition our experiments have shown that the actual gain applied varies significantly from the documented values.

The cross color channel effects can occur for a number of reasons. We believe these are due to small differences in the color gels between image sensors. Differences in the color gel cause wavelength-dependent effects that cause a distortion of the color space.

The image processing steps for RGB to YCbCr conversion include a 3x3 RGB to RGB color-space transform applied to perform a transformation from the sensor's RGB cell response to the RGB response of a typical computer monitor. We have found that even when the raw sensor output is well matched, the RGB to YCbCr process introduces color

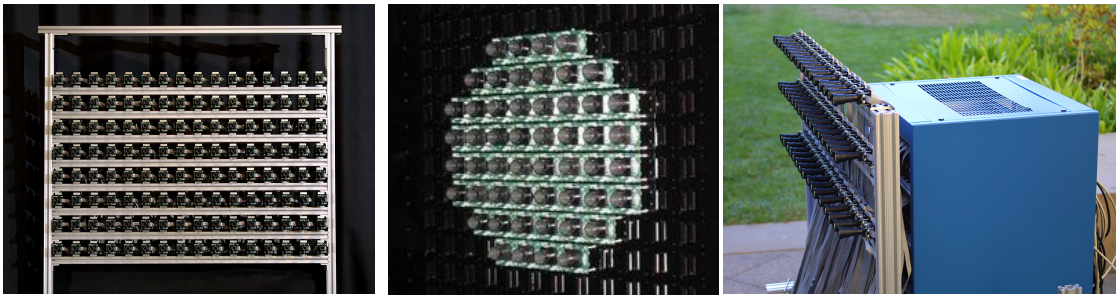


Figure 5: Three configurations of the array used for light field acquisition. Left: A 95 camera subset of this setup was used for multi-camera optical flow based interpolation and for comparative experiments on color calibration. Middle: This densely packed 52 camera setup cameras was used to acquire high-speed video. Right: This 45 camera setup with wide-angle lenses was used for outdoor acquisition.

discrepancies. The details of this process are undocumented for our sensors, but as this happens in an analog domain, the cause of the errors should be similar to those in the raw sensor processing. While our sensor provides several controls to adjust the raw sensor output, it provides an incomplete set of controls to control the RGB to YCbCr conversion. We have found that the automatic settings for white-balance, gain-control, and exposure control produce unusable results. To get reliable and consistent color data, we must use the raw Bayer data directly. With a full set of adjustments for each color channel in the raw Bayer data, we can calibrate the raw sensor output, acquire the raw sensor data, and achieve successful results with our applications. We use a publicly available demosaicing algorithm [Chang et.al.] that produces good results.

2.2 Experimental Setup

For our multi-camera experiments we used the camera array described by [Wilburn et al. 2004]. Our camera array consists of 100 custom video cameras using Omnivision OV8610 sensors to capture 640x480, Bayer mosaic color images at 30fps. Each camera has a processing board that manages the compression and IEEE1394 interface. This processing board also has a Motorola Coldfire processor and Xilinx FPGA to provide on-board image processing. Figure 4 shows a single camera and processing board. The array can take up to twenty synchronized, sequential snapshots from all of the cameras at once. The images are stored locally in memory at each camera, limiting us to only 2/3 of a second of video. Using MPEG compression at each camera, we can capture essentially indefinitely. The array can be reconfigured for a variety of setups for light-field acquisition. Figure 5 shows several configurations of the array used for the data collected for this paper.

Chapter 3

Process

Our calibration pipeline uses images of a Macbeth color checker taken by all of the cameras. We use a diffuse photographic graycard to capture the non-uniform lighting at the location where we place the color chart and use the values recorded for the gray card to adjust for the non-uniform lighting. We then image the color checker and iteratively adjust the gain and offsets on each channel so the sensor output fits a line through the six gray patches on the chart. We use a line that maps the brightest and darkest squares to RGB values of (220,220,220) and (20,20,20), respectively. By calibrating each channel to this linear response, we simultaneously white balance our images and maximize the usable data in each color channel for each camera. A post-processing step applies a floating point gain and offset correction, generates lookup tables to correct for residual non-linearity, and then determines a 3x3 color transform to best match, in the least squares sense, each camera's output to the mean values from all of the sensors. At the moment we are not correcting for $\cos^4$ falloff or vignetting as we have found that these affects cause minimal errors in our applications.

3.1 Automatic Detection of the Macbeth Color Chart

As all steps of our calibration process depend on having corresponding points on the Macbeth color checker across a large number of cameras, we developed a method to automatically detect the patches on the color chart in the view of each camera. To do this we leverage a simple geometric calibration technique typically used for image registration [Vaish et al. 2004]. We place a planar geometric calibration target of know geometry in the scene and take a single image of the target with each camera. Using a corner-detector we extract point-correspondences for each image and compute 2D homographies that warp the image-space coordinates to the coordinate system on the plane of the calibration target. We then place the Macbeth color chart at a predetermined location where we have pre-measured and recorded the locations of the centers of the patches on the color chart in the coordinate system of the geometric calibration target. By using the inverse of the 2D homography we computed from the previously acquired images, we can compute the image-space coordinates, i.e. pixel locations, of the centers of the color chart patches for each camera. Figure 6 illustrates the process.

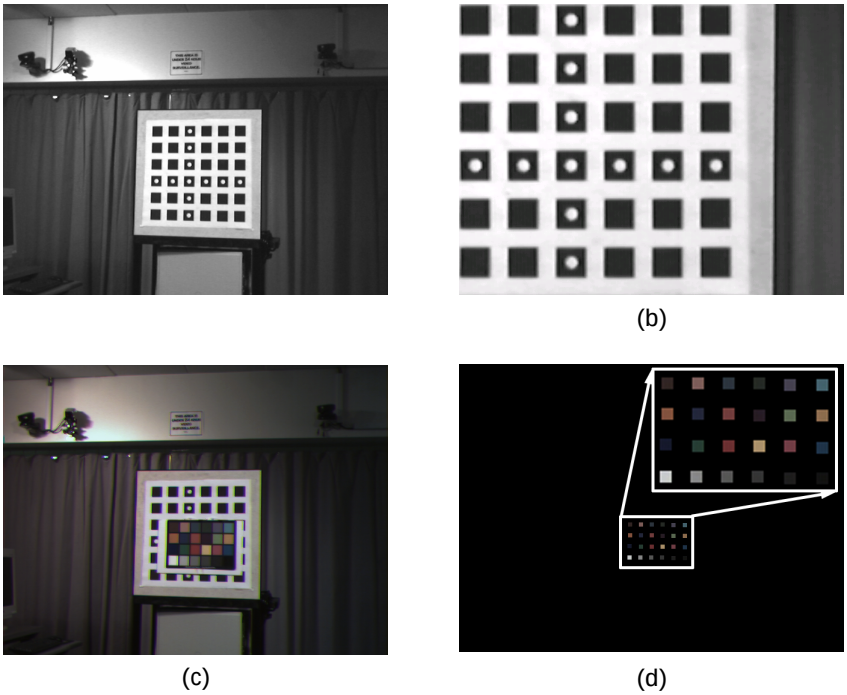


Figure 6: Automatic detection of the Macbeth color checker chart. (a) Using corner based feature detector we can find corresponding points on a planar geometric calibration target of know geometry. A 2D homography is computed to warp the image of the calibration target to a known coordinate system. The coordinate system is such that the origin is the top left corner of the top left square where one pixel corresponds to one millimeter. (b) The 2D homography applied to the original image. (c) The Macbeth color checker is placed at a known, pre-measured location on the geometric calibration target. The inverse of the 2D homography computed to warp (a) to (b) is used to warp the known locations of the color patch centers to pixel coordinates. (d) A small window around the patch center is used for spatial averaging.

To be robust to camera noise we want to average over a number of pixels lying within each patch. We average over four frames to reduce the effects of temporal noise, while we average spatially over a small window around the patch centers to reduce the effects of fixed pattern noise. If the color checker takes up too small of an area in the images we may unintentionally expand the square to include pixels outside the desired patches – this is undesirable. The problem can be exacerbated when the color chart is not fronto-parallel to the image plane. In practice this does not pose a problem as our feature detector is very conservative and will not detect enough features if the geometric calibration target is not large enough in the image (less that 300x300 pixels). When the color checker is parallel to the image plane the entire chart is on the order of 200 pixels wide making an individual patch a 30x30 pixel square. To be safe we use a much smaller window of 6x6 pixels. If the feature detector indicates that the geometric calibration target is too far away we must move the target closer. We then store the bounding coordinates for the averaging windows for every patch for each camera. The averaging windows are shown in Figure 6d. These stored coordinates are used repeatedly during the process. If the tilt of the chart became a significant problem in our use of a square patch for averaging, it is a relatively simple extension to store a pixel list that represents a

non-square region to be averaged over for each patch. This could accommodate averaging over pixels on the target from significantly off-axis views.

3.2 Non-uniform Illumination Correction

One potential source of error in color calibration is the effect of illumination variation across the color checker. Non-uniform illumination of the color checker skews the gray patch values, causing gain and offset miscalibration. We endeavor to uniformly illuminate the color checker in our scene and place the checker in the center of the cameras' views, so that radiometric falloff due to the camera lens has a minimal effect. While keeping the color chart in the center of the image is simple enough it is often difficult to control the lighting in a scene without a specialized setup. Before we perform gain and offset calibration we correct for non-uniform illumination by recording the illumination by placing a photographic gray card at the same location where we will place the Macbeth color chart. We record the RGB values across the gray card at the same locations where the Macbeth chart will be sampled and compute scale values that correct for non-uniform illumination. We use these scale values in all steps of the calibration process to adjust the recorded color values from the Macbeth color checker to remove illumination effects. The data plotted in Figure 7 has been corrected for illumination effects.

3.3 Gain and Offset Calibration

We calibrate the gains and offsets for each camera to ensure that the raw data output for our scene uses the maximal range of the sensor (i.e. the data is not clipped or clustered into a small range). We turn off gamma correction and calibrate the gains and offsets so that each color channel observes the same linear response. This also serves to white-balance the scene. Instead of separately calibrating the offset (i.e. black-level) of the cameras from black images and then adjusting gain (i.e. contrast) off of a reference gray or white value, we calibrate both offset and gain in one step. We do this by acquiring an image of the Macbeth color checker and fitting a line to the recorded RGB values for the gray patches on the color chart. By using a least squares line fit to multiple gray values we are more robust than a method that computes gain and offset using black and gray images. These methods essentially fit a line to two points. By fitting to more than two points we are robust to errors in the sampled points. We have observed that the upper and lower ends of the sensor response curve tend to be non-linear. With a two-point black-level/contrast calibration, the offset is calibrated using data recorded in a known non-linear region of the sensor, while the single gray value used for gain computation could lie in a non-linear portion at the upper end of the sensor if gray-level is too bright due to scene lighting or the default gains before calibration. By using a linear fit to the four middle level gray values on the color checker, we calibrate using data in the more reliable middle range of the sensor.

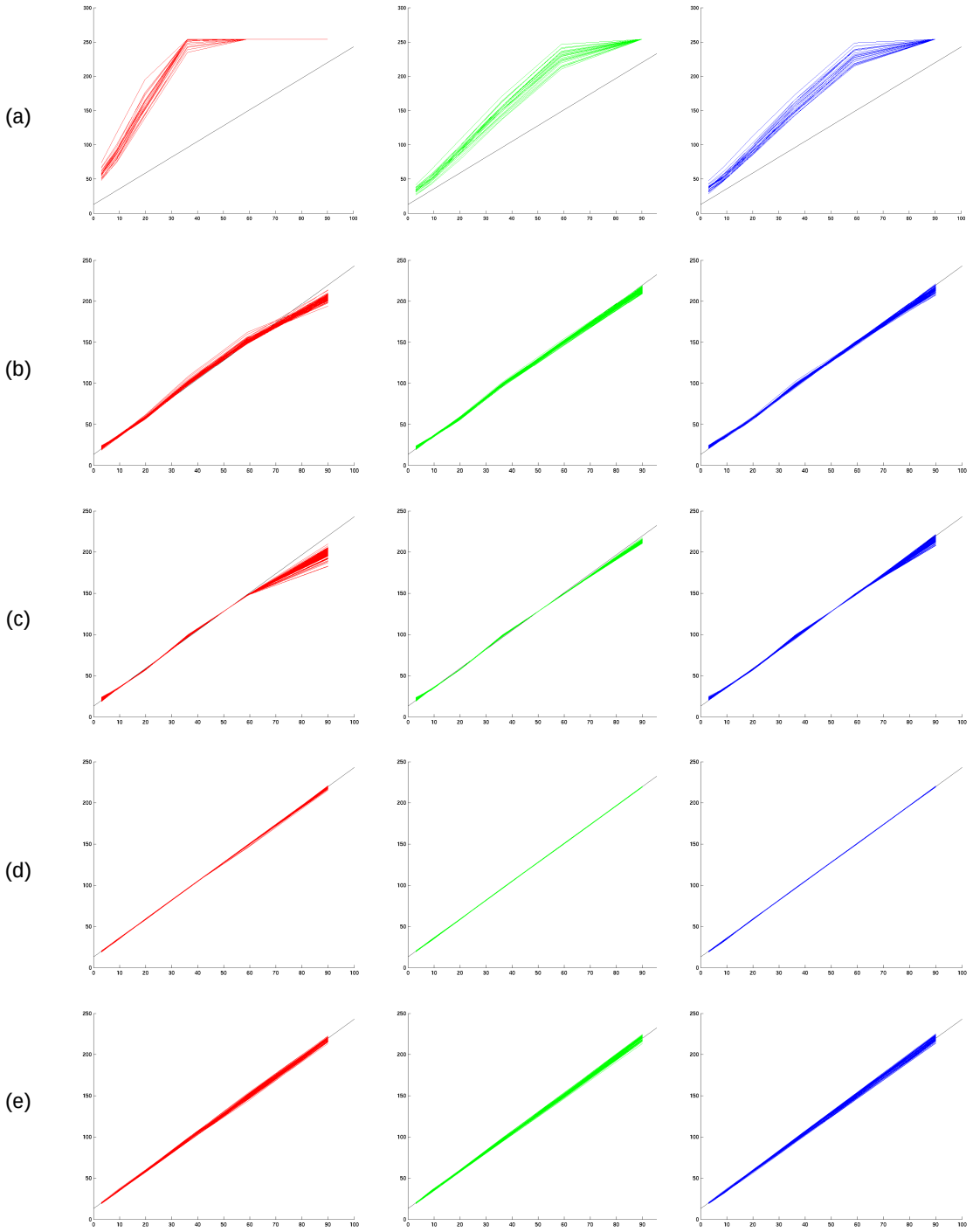


Figure 7: Sensor response curves. Red, Green, and Blue response curves plotted for 95 cameras. The X-axis is luminance while the Y-axis is the measured sensor response. (a) The response curve at default gains, with gamma off. (b) The cameras have been calibrated to the target response curve (the straight black line visible at the upper end of the response curve) using the gain and offset adjustments on the camera. There is residual non-linearity apparent for several cameras. (c) Gain and offset correction in post-processing with floating-point precision. The central part of the curve is more on target. There is residual non-linearity. (d) Look-up table remapping to correct for non-linearity. (e) The final response curves after the 3x3 transform. Note the responses are linear, but have become misaligned as the least squares optimizations makes equal trade-offs to minimize error across all colors. This introduces some error in the matching of the gray patches but reduces overall error in all the patches.

Through trial and error, we have found that a good response curve for our sensors is the line determined by the black patch (3.1% reflective) on the Macbeth color checker mapping to the value of 20 and the white patch (90.0% reflective) to value 220. The slope and y-intercept from the linear fit on the sampled gray values serves as the current gain and offset for the sensor. The ratio of the slope of the fit line to that of the target linear response curve is used as a multiplier for the current gain setting. The difference of the y-intercept of the fit line to that of the target linear response curve is used to adjust the current offset setting. Due to inaccuracy in implementation of the gain and offset settings on our cameras, we do this process iteratively. With our image sensors the gain for the green channel is global, so we first calibrate the green channel and then calibrate the red and blue channel in parallel. Figure 7a and 7b shows the sensor response curves before and after the gain and offset calibration process.

3.4 Post-processing

The gain and offset calibration does a reasonable job of calibrating the cameras to be radiometrically similar. However errors in the gain and offset settings, non-linearity in the sensors, and color distortions remain. We have designed a three-stage post-processing pipeline that explicitly addresses these issues.

The first step is to correct for residual errors in gain and offset. The settings for gain and offset have some discretization that occurs in their implementation. On our image sensors there is reasonable precision in the offset setting where the offset can be adjusted from [-64 to 64] in single increments – our experiments have show that in practice the offset adjustment adheres to this. The gain setting is far less precise and accurate. It has a somewhat large step size giving only coarse control. In addition our experiments have shown us that the actual gain applied varies significantly from the documented values. To correct for these errors we image the Macbeth color checker and perform a line fit and compute gain multipliers and offset adjustments just as in the previous step; however we apply these adjustments in a floating-point domain with more precision and accuracy. We compute these gains and offsets per channel and apply them to the images of the color checker. The gain and offset adjustments are saved for later use. Figure 7c shows the sensors responses after this correction.

The next step is to correct for non-linearity in the sensor. We compute a look-up table that re-maps the response curve of each channel to the desired linear response curve. For each possible color value from 0 to 255, for each channel and camera we use a piecewise linear curve based on the 6 gray values on the Macbeth chart and their imaged RGB values. This piecewise linear curve is used to compute the luminance value for each color value. This luminance value is plugged into the equation for the desired response to compute the target color. We then have a mapping from each cameras color response to the desired response. Figure 7d shows the sensors responses after the look-up table correction.

We have now corrected each camera individually to produce images with a desired linear response across each channel independently. The final step is to correct for color distortions and to minimize error globally by computing a 3x3 RGB to RGB color transform that minimizes error in the least squares sense. Previous methods have used a similar technique to correct for color differences across cameras; however, they typically pick one camera as the reference camera and compute a per-camera transform to match to the reference camera. We have found that matching to a reference camera is often not the best way to globally minimize error. When matching to a single reference camera there is a danger that the camera is an outlier. Matching to the mean color values across all cameras is a more robust approach that is less affected by single outlier cameras. This method keeps the transformations minimal in magnitude (minimizes the per-camera colorspace scale, rotation, and shear) and provides a more attainable goal for the error minimization. We compute average values for the 24 patches on the Macbeth color checker and compute a 3x3 transform that minimizes the error between each cameras RGB values for the 24 patches and the average values. Figures 7e shows the sensor response curves after this final post-processing stage.

3.5 Scalability

Our system is scalable due to particular implementation and design decisions. The use of 2D homographies for automatic location and computation of point correspondence on the Macbeth color checker significantly enhances the scalability of our technique. Without this a user would need to manually click points to identify location on the chart. This is tedious for small numbers of cameras and impractical and error prone for large numbers of cameras.

By leveraging our camera's on-board processing power we are able to significantly cut down on image transfer and image processing time and are able to parallelize certain computations. While this is by no means necessary for the successful application of our method, we have found it to be a great asset when calibrating a large numbers of cameras. We have implemented functionality to upload the patch coordinates resulting from our automatic color chart detection to the Motorola Coldfire processors on each camera board. We then take images and have the Coldfire perform the temporal and spatial averaging in RAM for the 24 patches on the color checker. This process significantly reduces the total calibration time as the data returned from the cameras is significantly reduced and the image reading and averaging is done in parallel across all cameras. Using the Coldfire for averaging reduces the camera to host PC download to 72 bytes from the 300KB for the entire image.

Chapter 4

Results

In this section we will show images and error statistics for data acquired from a 95-camera array with no color calibration, with an implementation of the color calibration system used for the RingCam, and with our color calibration system. Further we will show results from color calibration outdoors and experimental results from high-speed video, matted synthetic aperture photography, and multi-view optical flow.

We have found that a good way to visually judge the results of color calibration is to create single composite images from multiple cameras. We create these image compositions by registering images using 2D homographies to align a geometric calibration target from all views to one reference view. We select multiple 5x5 pixel blocks from each registered image and paste together a final image. We have found this to be a good test as it simulates the type of image reconstructions often used in image-based rendering and it easily reveals perceptible color differences, as there is no blending or interpolation between adjacent blocks. See Figures 8 through 11 and Table 1 for comparison and analysis of image compositions from data acquired with a 95-camera array uncalibrated, calibrated with RingCam calibration method, and calibrated with our method with and without our post-processing steps.

We also show results for three color-sensitive applications: high-speed video, synthetic aperture photography with matting, and multi-camera optical flow. High-speed video and synthetic aperture photography composite images from multiple cameras where color inconsistency causes perceptible color artifacts. Multi-camera optical flow is a computer vision method that is sensitive to color differences because it assumes constant brightness across views for corresponding points on objects in a scene. Although it is possible to formulate optical flow and other vision methods to be more robust to color variations [Kim et al. 2003] and [Black and Anandan 1993], most formulations of popular vision methods will produce poor results with the significant color differences seen in an un-calibrated camera setup.

4.1 High-speed Video

By capturing a light-field using an array of video cameras that provides control over individual camera trigger time and exposure time, a high-speed event can be captured by staggering camera trigger times to more densely sample a scene in time. By geometrically aligning images from different cameras and properly interleaving frames from the video streams according to the staggering pattern, a single high-speed video sequence can be created. Because we interleave images from our cameras, variations in their color response will cause frame-to-frame intensity and color differences perceived as flickering in the resulting high-speed video. To correct for a particular timing artifact in our sensors we must "temporally slice" though a set of images creating an image composed of data from a large subset of the 52-camera array we used for acquisition [Wilburn et al. 2004]. With accurate color calibration, these intensity and color differences are minimized in the resulting high-speed video sequence. Figure 12 shows three frames from a high-speed sequence from the temporally sliced sequence. The reader is encouraged to view the video sequence located at *http://graphics.stanford.edu/papers/highspeedarray/balloons.mpg* to appreciate the effects of the color calibration process. In this video there are some residual color effects that are apparent after our temporal correction. See Chapter 5 for a discussion of this correction and the artifacts it introduces.

4.2 Synthetic Aperture Photography with Matting

Light fields can be used to simulate the defocus blur of a conventional lens by re-projecting some or all of the images onto a focal plane in the scene. This consists of registering the images onto a reference plane, translating the images, and averaging them. Objects on the focal plane will appear sharp, while those not on this plane will appear blurred in the resulting image [Levoy and Hanrahan 1996] and [Isaksen et al. 2000]. This synthetic focus can be thought of as resulting from a large-aperture lens. We call this synthetic aperture photography. When the aperture is wide enough, occluding objects in front of the focal plane are so blurred as to effectively disappear. In traditional synthetic aperture photography the large number of values averaged together serves to average out color differences. Synthetic aperture photography can be very successful without color calibration [Vaish et al. 2004]. One modification to synthetic aperture photography is to create per image mattes to remove the occluded pixels from individual frames before averaging them to create the synthetic aperture result. Mattes can be created using a variety of techniques. Some techniques matte out pixels of a certain color deemed the color of the occulder. Other techniques use statistical analysis of the image data to attempt to detect occluded and unoccluded pixels. The use of mattes significantly improves synthetic aperture results when attempting to see through partial occluders as occluded pixels don't contribute to the resulting image. With dense occluders only a very small subset of cameras contributes to create each individual pixel. Using a small number of cameras for averaging makes color calibration important, as color differences don't average out as well. Without accurate color calibration, there is inconsistency

between pixels averaged from different sets of cameras. Figure 13 illustrates the importance of color calibration in synthetic aperture photography when using mattes.

4.3 Multi-camera Optical Flow

Traditional optical flow techniques use the brightness consistency assumption to compute pixel flow between images from multiple viewpoints. Brightness consistency states that for a particular point on an object in a scene the brightness i.e. color value should be invariant as that point is viewed from various viewpoints. Violations of this assumption cause errors in traditional formulations of optical flow and cause incorrect flow vectors to be computed. Optical flow techniques have been modified to be more robust to brightness violations; however these techniques are intended to provide robustness to natural violations of the brightness consistency assumption due to shadows, motion boundaries, or specular reflections. As optical flow techniques are generally designed to run on multiple images from a single camera, these techniques do not handle systematic color differences well. Accurate color calibration significantly improves the results of multi-view optical flow techniques applied to multi-camera systems by calibrating the recorded data to more closely resemble data acquired from a single moving camera. Figure 14 shows a successful view-interpolation result created from running optical flow on four images with data from a fully calibrated array and it shows a failure case when run on data from a partially calibrated data set.

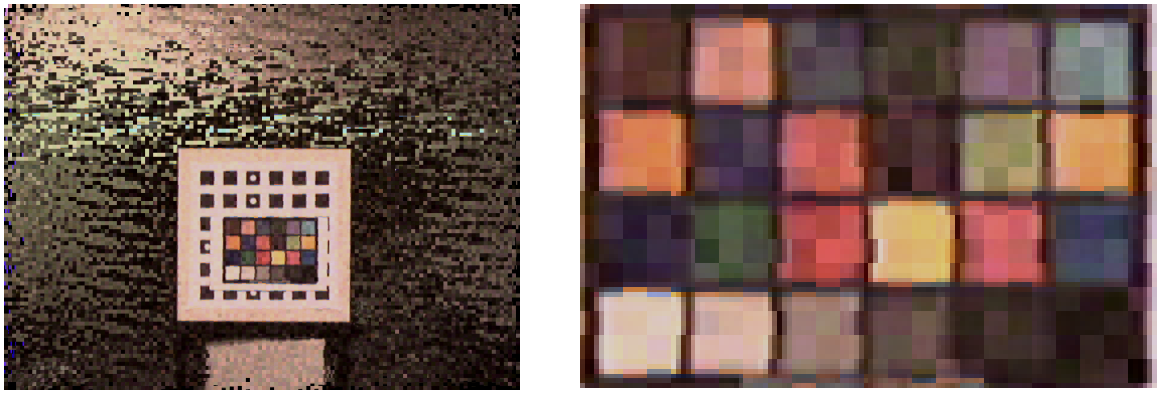
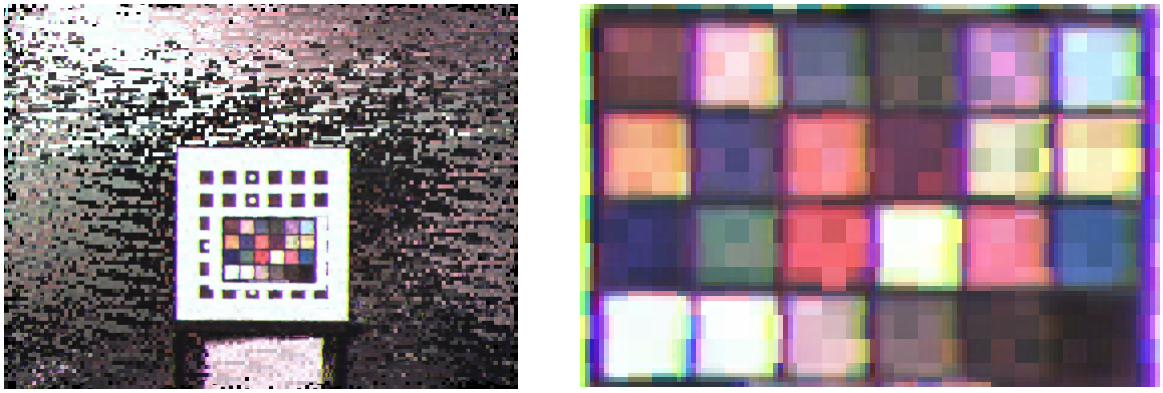
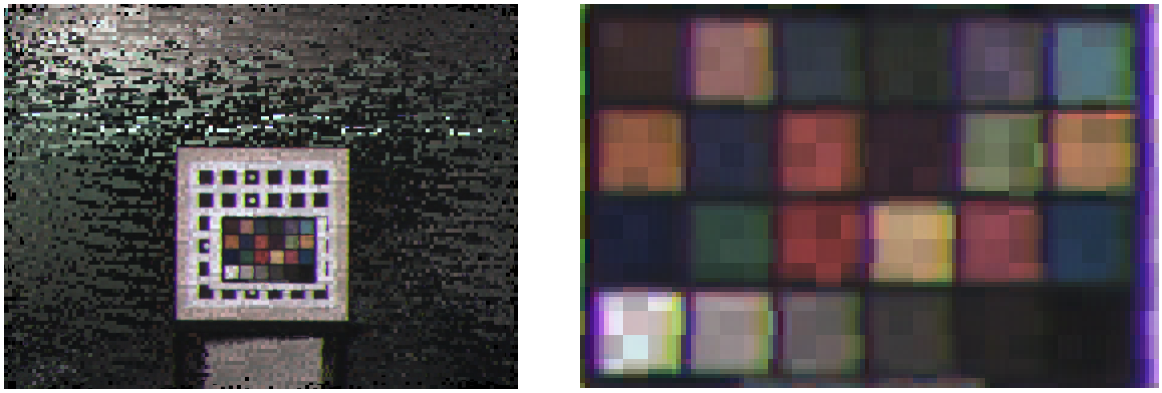


Figure 8: Uncalibrated cameras. This images shows clear color differences between cameras. There are visible patches of varying hue and brightness on the Macbeth color checker. Note: these composite images are created from source images that are registered only at the plane of the geometric calibration target – the "blocky" pattern in the background is due to pasting together pixel blocks from unaligned areas and for the sake of these comparisons can be ignored.



(a)



(b)

Figure 9: Other color calibration methods. (a) Images calibrated using scene statistics [Nanda and Culter 2001]. Color values from the Macbeth color checker were not used directly by this calibration process. The Macbeth Chart is used only to evaluate the success of the calibration process. Image (a) is saturated as we used a default value of 127 as the mean value for contrast calibration. This process is very sensitive to selecting an appropriate image mean. (b) The same process with a better-selected target image mean of 70. Color differences are still apparent.

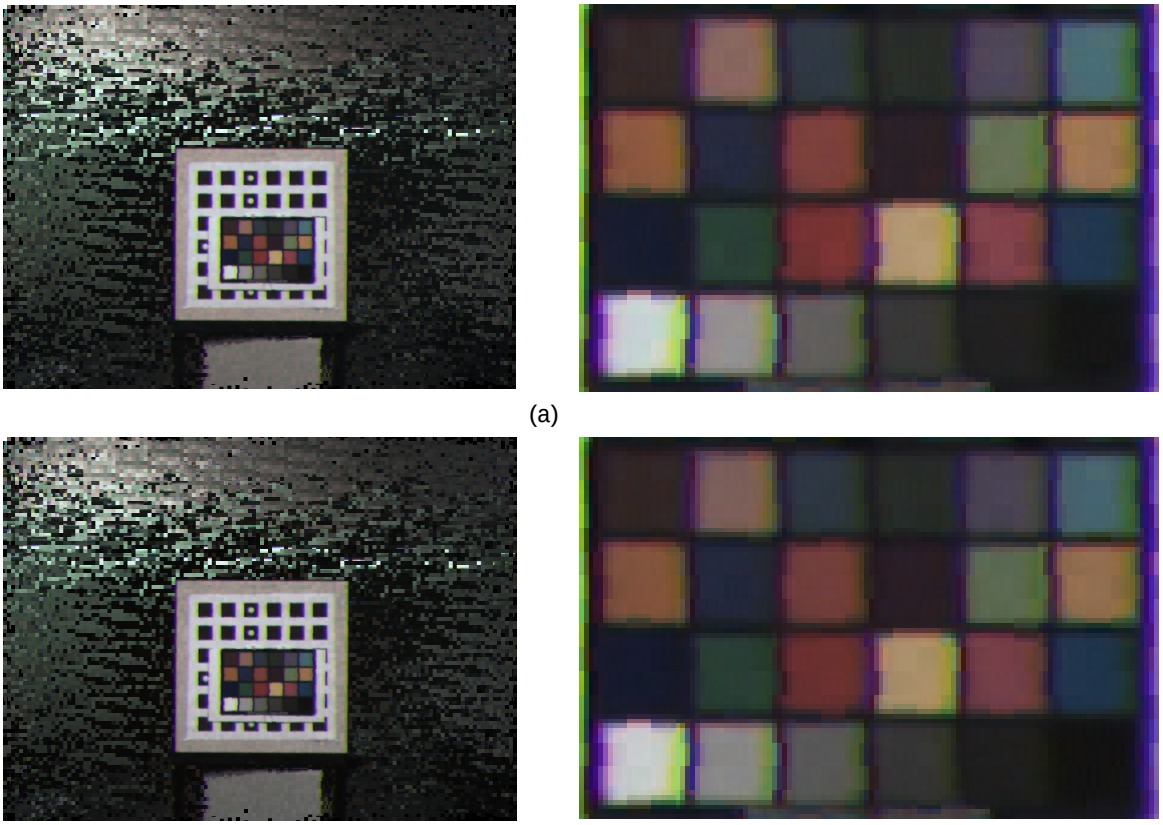


Figure 10: Our color calibration method. (a) Our gain and offset calibration alone with no post-processing. The results are significantly improved over those in Figure 9b; however, color differences are still apparent particularly in the red patches. (b) Our full color calibration pipeline. Color differences are almost imperceptible. There are artifacts on borders between color patches. These are due to geometric misalignment and demosaicing. Demosaicing artifacts appear due to color aliasing introduced by the Bayer pattern. In these images, the artifacts appear as subtle shades of red, green, and blue at the edges of some color patches. The bright yellow patch particularly shows these effects.

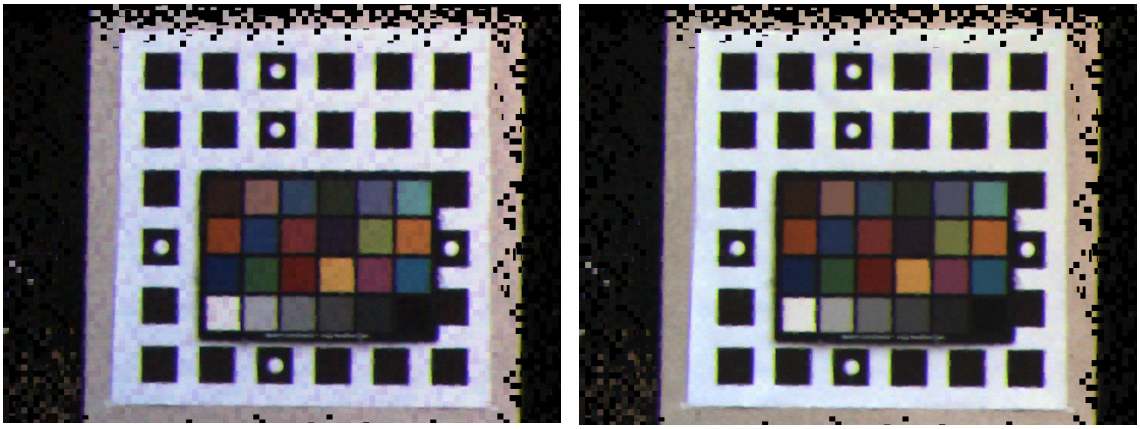


Figure 11: Color calibration outdoors. The images illustrate the application of our method outdoors with no control over lighting conditions. These images were created using 7x7 pixel blocks from registered images acquired from a 45-camera array. Left: image composition with gain calibration, but no post processing. Right: image composition after post processing stages. Without the post processing color differences are still visible in the image.

Calibration Method	RMS percent relative error in red	RMS percent relative error in green	RMS percent relative error in blue
None	16.4 %	177.2 %	70.6 %
Scene based, mean 127 (Figure 9a)	11.2 %	8.2 %	9.3 %
Scene based, mean 70 (Figure 9a)	8.9 %	7.3 %	8.1 %
Our method without post-processing (Figure 10a)	2.8 %	2.3 %	2.8%
Our method (Figure 10b)	1.9 %	1.4 %	2.1 %

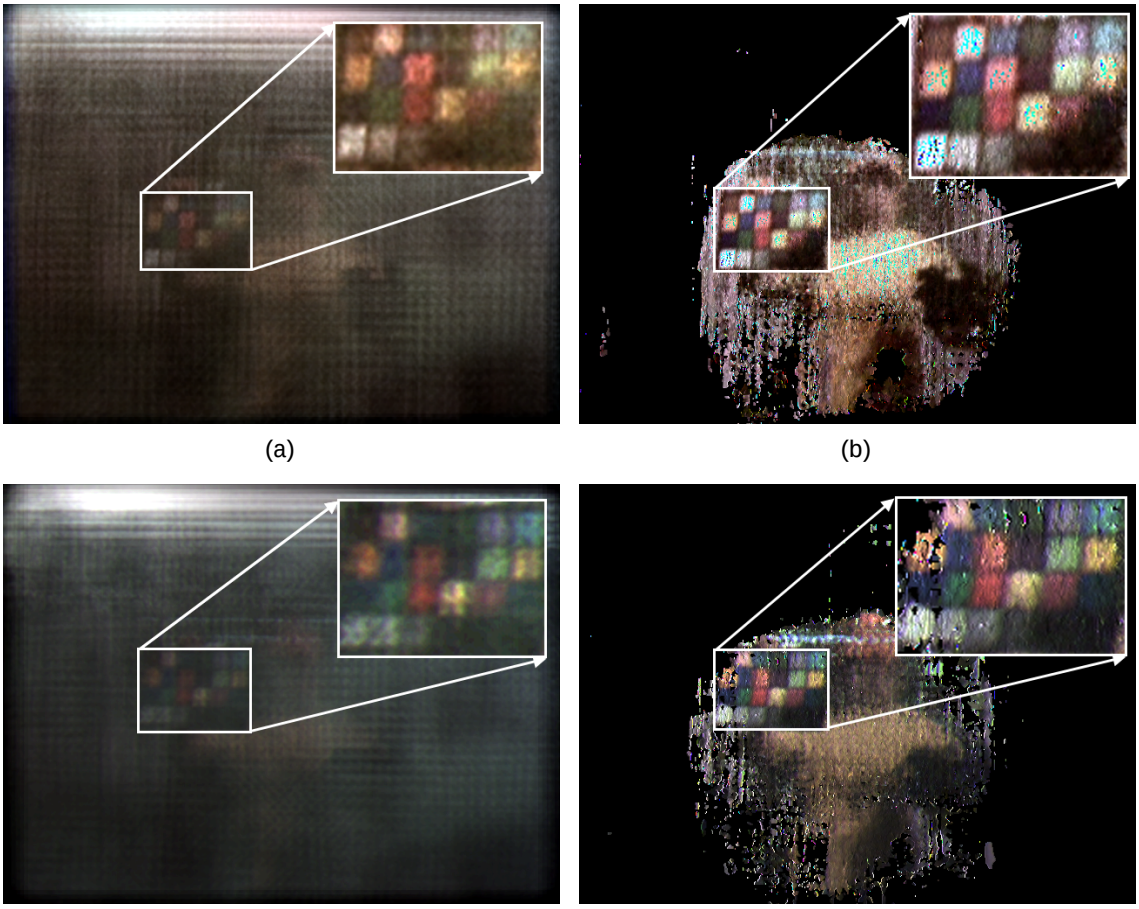
Calibration Method	RMS absolute error in red	RMS absolute error in green	RMS absolute error in blue
None	9.971	8.146	9.058
Scene based, mean 127 (Figure 9a)	15.805	12.059	14.459
Scene based, mean 70 (Figure 9a)	7.980	7.980	8.304
Our method without post-processing (Figure 10a)	2.450	1.706	2.017
Our method (Figure 10b)	1.227	0.724	1.076

Calibration Method	Maximum error in red	Maximum error in green	Maximum error in blue
None	78.879	47.656	58.610
Scene based, mean 127 (Figure 9a)	116.337	80.104	82.124
Scene based, mean 70 (Figure 9a)	89.584	82.989	84.631
Our method without post-processing (Figure 10a)	31.657	16.724	17.055
Our method (Figure 10b)	8.754	5.159	9.430

Calibration Method	RMS absolute error in x	RMS absolute error in y
None	0.018	0.019
Scene based, mean 127 (Figure 9a)	0.015	0.008
Scene based, mean 70 (Figure 9a)	0.008	0.005
Our method without post-processing (Figure 10a)	0.006	0.004
Our method (Figure 10b)	0.005	0.003

Table 1: Error statistics. Here we show several error analysis metrics to analyze the performance of various calibration methods. Our gain and calibration method without any post process significantly out performs the scene-based method. With post-processing the error is reduced further in relative and absolute errors in red, green, and blue with the most significant effect in the reduction of maximum error. With our full calibration procedure the error is on average one gray-level. RMS chromaticity (x, y) error is also reduced by our process, but not as significantly.

Figure 12: High-speed Video. Due to a particular correction that needs to be applied to correct for a timing artifact of our cameras in our high-speed video setup, each of these three sequential video frames is created from different a subset of the cameras in our array. With color calibration these images appear to come from one camera. There are residual color effects that are only apparent in video. These effects are a temporal pattern that occurs as a result of our timing correction. They become apparent to the human eye as there is a periodic nature to the residual color difference as the order in which the images are re-sampled changes over time.



(a) (b)

(c) (d)

Figure 13: Synthetic aperture photography. (a) Synthetic aperture result without matting or color calibration. (b) Result using matting without color calibration. (c) Result with color calibration without matting. (d) Result with matting and color calibration. The unmatted result (a) shows accurate color similar to the result with calibrated data (c) as color errors are averaged out over 95 cameras. With the matted results (b) and (d), on average 4 cameras are averaged for every pixel. The result with uncalibrated data (b) now shows saturated pixels and the light green and yellow patches on the color checker appear to be more similar in color than in the unmatted result. With the calibrated result (d) the color patches retain the same color balance as in the unmatted data. Note: the black patterns on the Macbeth chart in these images are due to light field aliasing where the occluder has not completely blurred out. These patterns are not due to inter-camera color variations.

Figure 14: Spatiotemporal optical flow. Optical flow is an application that relies heavily on color calibration as it has a central assumption of brightness consistency. Perceptual color differences are still undesirable, but of greater concern are numerical differences. Four Images acquired from different locations and at different times are aligned with multi-dimensional optical flow. Left: Data acquired from a partially calibrated setup shows noticeable flow inaccuracies on the fine details such as the person's eye. The soccer ball is relatively sharp. Right: Data from a second acquisition after full calibration with full post-processing applied. The details of the eye are preserved and the soccer ball is sharper.

Chapter 5

Conclusions and Future Work

We have shown that it is possible to accurately and precisely calibrate a large number of lost cost video cameras with very low residual error. By using a color target of known luminance we calibrated our cameras to a desired response curve with a method that is robust to the non-linearity present in low-cost image sensors. We have shown a simple method for correcting for non-uniform illumination – an effect that can significantly affect color calibration and that can't often be controlled for in certain acquisitions. Our geometric calibration system provides a robust, automated way to detect our color checker and along with our use of on-board image processing allows the system to scale to large numbers of cameras without increased complexity. We use a multi-stage post-processing step designed specifically to address the types of color inaccuracies seen in our low-end sensors. Our error analysis shows the benefit of our method over one using image means for light field acquisition. Our error statistics show an eight to ten times reduction of RMS absolute error, RMS percent error, and maximum error in red, green, and blue relative to the image-mean based method. Using our calibration we obtain high quality computer vision and graphics results from large arrays of inexpensive image sensors.

Our results illustrate some interesting properties of color imaging. The high-speed video work shows how images can be reconstructed from a large number of cameras without objectionable color artifacts. We have noticed some residual color effects that are only apparent in video. These effects are a temporal pattern that occurs as a result of our timing correction. They become apparent to the human eye, as there is a periodic nature to the residual color difference as the order in which the images are re-sampled changes over time. Our high-speed work has also revealed some other color artifacts, which are difficult to correct. We have noticed that the color artifacts that remain in our high-speed video are most noticeable in the background. Due to our lighting setup and short exposure times, these background areas are significantly darker than the rest of the scene, but show larger color shifts. One possible explanation is that these errors are a result of color quantization. Color quantization is the loss of precision that can occur during of a number of stages of the image-processing pipeline. This loss of precision could correspond to just a few gray levels. A quantization error in bright areas in an image is imperceptible as it is represents a lower percent error for the larger color value. These

errors are also less noticeable in brighter areas due to how our eye adapts to and perceives brightness. In darker areas these errors are much more significant. Quantization errors can appear as an intensity change or as they can occur independently in each channel errors can cause a color shift. While we are able to overcome some quantization effects during our calibration procedure by using raw image frames and averaging images spatially and temporally, these effects are difficult to overcome with video as there is quantization in MPEG compression and for particular applications averaging image frames is not an option.

In our application of spatiotemporal optical flow we have found that flow errors still occur with well-calibrated cameras. Particularly we have seen color artifacts and resulting flow errors when aligning images of the soccer ball in our dataset at certain time-steps as the ball is moving through our scene. There are only a few frames where color differences are noticeable on the ball. We have found that this depends on the location of the ball in the scene and particularly on the directionality of the lighting. The soccer ball we filmed with is slightly specular, so under directional lighting, which is not completely unavoidable, the intensity of the imaged ball varies with viewing angle i.e. the color difference is real. This is a known problem with specular objects in these types of applications and it would occur even in the ideal case of perfectly color calibrated cameras or even with images acquired from a single moving camera.

There are other sources of error that we don't correct for. Many imaging devices experience changes in their response with variations in temperature. We have not attempted to model this behavior. It is unclear if errors due to these effects are even noticeable. It is further unclear how best to correct for these errors. In practice when researchers have noticed heat related affects the common approach is to let the system "warm up" and stabilize before conducting an experiment.

There are several potential directions for future work in this area. Our system is designed for light field acquisitions where all the cameras have some working volume in common. One extension to this work is to deal with a camera array setup with partially or non-overlapping views. There are several considerations here. To image a color chart one may move the chart around the scene so that all cameras can view that chart at some point in their working volume. One has to be careful when moving the chart as the lighting falling on the chart will change and each camera may see a slightly differently illuminated chart. Possible solutions to this are to create a self-illuminated target, although then very strict control is needed over the lighting conditions.

A limitation of this work is that cameras must be calibrated under filming light levels and exposure levels. While this has not yet been a limitation in our work, there are situations in which a full recalibration with a light level changes is impractical. Handling changing light levels without full calibration would be an added convenience and might be necessary for certain applications. One possible approach is to update a calibrated array by using imaging or other techniques to detect a light level change in a precise way and using this information to update camera gain and offset settings. Implementing this without re-imagine a color checker is not straight forward as each camera has to be

characterized in some way so that the errors in gain and offset adjustment are properly handled when readjusted for new light levels.

Sensor characterization leads to another potential interesting area for future work. In our calibration procedure we have made no effort to characterize individual sensors explicitly, but instead try to match all sensors equally well. Another approach would be to detect and label outlier cameras or better yet to cluster cameras by their particular types of color differences. By labeling individual sensors in this way it would be possible to bin cameras and intelligently use cameras that agree well for certain applications or pick cameras based on particular qualities. For example for a certain scene one might know that the red channel's needs to be of high accuracy, so only cameras with well behaved red response would be used. Using sensor characterization to better calibrate and better distribute cameras in an application specific way could be a fruitful way to produce even higher quality results from inexpensive sensors.

References

K. Barnard and B. Funt. " Camera characterization for color research." *Color Research and Application*, Vol. 27, No. 3, pp. 153-164, 2002.

M. Black and P. Anandan. “A framework for the robust estimation of optical flow.” *Proceedings of ICCV,* 1993.

E. Chang, S. Cheung, and D. Pan. "Color Filter Array Recovery Using a Threshold-based Variable Number of Gradients.” *Proceedings of SPIE,* January, 1999.

P. Debevec and J. Malik. “Recovering High Dynamic Range Radiance Maps from Photographs.” *Proceedings of SIGGRAPH,* 1997.

M. Grossberg and S.. Nayar. “What can be Known about the Radiometric Response Function from Images?” *Proceedings of ECCV*, 2002.

A. Isaksen, L. McMillan, and S. Gortler. “Dynamically Reparametrized Light Fields.” *Proceedings of SIGGRAPH,* 2000.

T. Kanade, H. Saito, and S. Vedula. “The 3d-room: Digitizing time-varying 3d events by synchronized multiple video streams.” Carnegie Mellon University, Tech. Rep. CMU-RI-TR-98-34, 1998.

J. Kim, V. Kolmogorov, and R. Zabih. “Visual Correspondence Using Energy Minimization and Mutual Information.” *Proceedings of ICCV,* 2003.

M. Levoy and P. Hanrahan. “Light Field Rendering.” *Proceedings SIGGRAPH,* 1996.

A. Majumder, Z. He, H. Towles, and G. Welch. “Achieving color uniformity across multi-projector displays.” *Proceedings of IEEE Visualization,* 2000.

H. Nanda and R. Cutler. “Practical calibrations for a realtime digital omnidirectional camera.” *Proceedings of CVPR, Technical Sketch,* 2001.

F. Porikli and A. Divakaran. “Multi-Camera Calibration, Object Tracking And Query Generation.” *Proceedings of ICME,* 2003.

P. Rander, P. Narayanan, and T. Kanade. “Virtualized reality: Constructing time-varying virtual worlds from real events.” *Proceedings of IEEE Visualization*, 1997.

V. Vaish, B. Wilburn, and M. Levoy. “Using plane + parallax for calibrating dense camera arrays.” *Proceedings of CVPR*, 2004 (to appear).

S. Vedula. “Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events.” Carnegie Mellon University, Tech Report, CMU-RI-TR-01-37, Robotics Institute, 2001.

B. Wilburn, N. Joshi, K. Chou, M. Levoy, and M. Horowitz. “Spatiotemporal Sampling and Interpolation for Dense Video Camera Arrays.” Stanford University, Tech Report, CSTR 2004-01, 2004.

B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. “High speed video using a dense array of cameras.” *Proceedings of CVPR*, 2004 (to appear).

B. Wilburn, M. Smulski, H. Lee, and M. Horowitz. “The light field video camera.” *Media Processors 2002*, ser. Proc. SPIE, S. Panchanathan, V. Bove, and S. Sudharsanan, Eds., vol. 4674, San Jose, USA, January 2002, pp. 29–36.

J.-C.Yang, M. Everett, C. Buehler, and L. McMillan. “A real-time distributed light field camera.” *Proceedings of Eurographics Workshop on Rendering*, 2002.