

# Automatic Color Calibration for Large Camera Arrays

Neel Joshi\*

Bennett Wilburn<sup>†</sup>Vaibhav Vaish<sup>†</sup>Marc Levoy<sup>†</sup>Mark Horowitz<sup>†</sup>

\*University of California, San Diego

<sup>†</sup>Stanford University

**Figure 1:** Image composites using blocks from different cameras. Blocks along a diagonal are from one camera. Three such blocks are highlighted in the first image. (a) Data from cameras at default gain and offset. (b) Cameras calibrated using software auto-gain and white-balance. Color inconsistency is significant. (c) Calibration of all cameras to a single standard - sRGB. Color artifacts are less perceptible although still noticeable in the white, yellow, and light green patches. (d) Our method, which calibrates cameras to each other, rather than to a standard. There are minimal artifacts. Note: Artifacts on the edges of color patches are due to demosaicing and geometric misalignment.

## Abstract

We present a color calibration pipeline for large camera arrays. We assume static lighting conditions for each camera, such as studio lighting or a stationary array outdoors. We also assume we can place a planar calibration target so it is visible from every camera. Our goal is uniform camera color responses, not absolute color accuracy, so we match the cameras to each other instead of to a color standard. We first iteratively adjust the color channel gains and offsets for each camera to make their responses as similar as possible. This step white balances the cameras, and for studio applications, ensures that the range of intensities in the scene are mapped to the usable output range of the cameras. Residual errors are then calibrated in post-processing. We present results calibrating an array of 100 CMOS image sensors in different physical configurations, including closely or widely spaced cameras with overlapping fields of views, and tightly packed cameras with non-overlapping fields of view. The process is entirely automatic, and the camera configuration runs in less than five minutes on the 100 camera array.

**CR Categories:** I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture.Camera Calibration;

**Keywords:** color calibration, camera arrays

## 1 Introduction

As digital cameras become cheaper and more easily managed, more and more researchers are investigating the potential of large camera arrays. For example, the 3D Room at CMU captures video from 49 cameras spread around a room [Rander et al. 1997]. They use the data for 3D scene reconstruction and view interpolation -- creating virtual views corresponding to camera

positions not in their captured set of images. Their cameras are relatively high-quality, but recently several groups have begun working with large arrays of inexpensive cameras. Yang et. al constructed an array of 64 commodity webcams for live rendering of video light field [2002]. Zhang and Chen built a system of 48 Ethernet cameras equipped with horizontal pan and translation controls, also for view interpolation [2004]. We have also constructed a large array of inexpensive sensors. Our system captures video from 100 CMOS sensors and has been used for view interpolation, high-performance imaging, and synthetic aperture photography [Wilburn et al. 2005].

Users of camera arrays generally take great care to geometrically calibrate their cameras, but often neglect color calibration entirely or rely on manually adjusting their cameras. This is not because color calibration is unimportant. Wilburn et al. [2004], for example, presented a high-speed video capture method using multiple cameras with staggered trigger times. Because sections of images from many cameras are interleaved to produce the final images, minimizing color variations between cameras is critical to creating the illusion of a single, high-speed video camera. Similarly, view interpolation algorithms suffer in the absence of color calibration. Vedula, for example, observed artifacts using images from the 3D Room for view interpolation because the cameras were not color calibrated [2001]. The users of the self-reconfigurable camera array calibrated their cameras geometrically but not radiometrically, causing view-dependent color variations in their results [Zhang 2005].

Although standard geometric calibration methods exist for calibrating arrays of cameras [Zhang 1999; Bouguet; Tsai 1987], much less attention has been paid to color calibration for multi-camera systems. A common approach for configuring cameras lets each one self-calibrate using automatic gain and white balance algorithms, but this method produce varying results depending on what portion of the scene each camera views. Because we assumed fixed illumination but changing scene content, we would prefer a method that calibrates based on scene illumination. In general, single-camera color calibration methods are useful for characterizing response functions, but they are not optimal for matching multiple cameras.

The only multiple camera color calibration work we are aware

Configuration Method	Characterization Method	RMS Error	Max Error
Auto-gain and white-balance	None	7.98, 7.98, 8.30	89.59, 82.99, 84.63
Cameras set to identical gains and offsets	None	9.97, 8.15, 9.06	78.88, 47.66, 58.61
Cameras set to identical gains and offsets	Matching to a standard color model - sRGB	3.40, 2.44, 4.23	36.84, 27.65, 47.59
Our configuration method	None	2.45, 1.43, 1.90	33.94, 12.67, 18.25
Our configuration method	Matching to a standard color model - sRGB	2.17, 1.00, 3.55	18.52, 9.72, 25.41
Our configuration method	Our characterization method	1.23, 0.72, 1.08	8.75, 5.16, 9.43

**Table 1:** Analyzing calibration methods with a 95-camera array. These error metrics show the root-mean-squared error and maximum error computed across all cameras and all Macbeth color patches broken down by color channel. There are large errors at initial settings. For auto-gain and white-balance, the error is worse. Matching to a standard color model – sRGB is better, but the maximum error is still high. Using our configuration method the error drops significantly. Matching to sRGB after configuration improves the error in red and green. With our full method, configuration and characterization, the error is minimal.

of is the automatic gain and white-balance method presented by Nanda and Cutler for their five-camera omni-directional RingCam [2001]. Their system addresses three needs that we do not: omni-directional viewing, a mobile array (and thus highly variable illumination conditions), and real-time operation. They color calibrate using image statistics in overlapping regions of their cameras' fields of view. This works well for their conditions, but as we will see, it is not optimal for the situations we address in this paper: stationary camera arrays under relatively static lighting conditions.

Our system is tailored for large arrays of inexpensive cameras. Thus, it is fully automatic and assumes controls over gains, offsets, and electronic shutter durations that are common in low-end CMOS sensors. In the rest of this paper, we describe the goals and operation of our multi-camera color calibration algorithm. Section 3 shows results calibrating our 100 camera array for different applications.

## 2 Automated Color Matching

We have implemented a calibration method for the camera array described by Wilburn et al. [2005]. The array consists of custom video cameras constructed from low-cost CMOS image sensors and inexpensive optics. We record raw, linear sensor data for our applications as it can be easily calibrated using the method we will now present.

### 2.1 Calibrating Overlapping Fields of View

Our method consists of two distinct stages: configuration and characterization. The configuration automatically adjusting camera gains and offsets prior to acquisition. Characterization consists of three parts: correcting for sensor non-linearity, correcting for radiometric falloff, and globally minimizing color error. After performing these steps we can acquire color calibrated data by filming with the calibrated gains and offsets, then applying our non-linearity correction and global error minimization to the acquired data. We will now walk through the calibration steps enumerated above.

**Automatic Location of Color Checker Patches.** Because the fields of view of our cameras overlap, we can calibrate using a Macbeth color chart viewable from all cameras. To avoid the necessity of searching for the color patches in each camera's view, we piggyback color calibration onto geometric calibration, by affixing the Macbeth chart atop a planar geometric calibration target. Once we have found the location of this second target, we also know the location of the color patches in the Macbeth chart. We store these locations for later use.

**Gain and Offset Configuration.** This step calculates the current gains and offsets of the sensor response and adjusts them

to match a target response function. We take images of the Macbeth chart at several different exposures in the linear middle-range of our sensors. Using the stored patch locations from the previous step, we record the RGB values for the white patch from these exposures. We then fit a line to this data to recover each channel's current gain and offset. We compute adjustments to these values such that at zero exposure the camera returns 12 in each channel and when viewing the white patch at our chosen exposure it returns 220 in each channel. We perform four iterations of this configuration step on each camera.

**Response Linearization.** After the previous step, the sensor response is linear, except at the low and high end of the range; thus we model and correct for this non-linearity. We take images of the Macbeth chart at every exposure setting, and we record the RGB values for the white patch in these images. As long as the scene is bright enough that the white patch will saturate at some exposure setting, this process allows us to map the entire sensor response function. We then compute a reverse mapping from RGB to exposure using linear interpolation on this data, we scale the output range of this mapping to match the 0 to 255 image range. This result is a look-up table that maps the original camera data to a linear 0 to 255 image range. We save these look-up tables for later use.

**Falloff Correction.** Before globally minimizing error, we must address the effect of radiometric falloff on images of the Macbeth color chart. As falloff is different for each camera, it introduces inconsistency between each camera's view. Often, we can limit falloff over a color checker image by placing the color checker at the center of each camera's view. If we place the chart so that it covers less than 25% of the field of view at the center of the falloff, there is only a 2% falloff across the chart. If we cannot place the chart as described, or if the center of falloff is unknown, we correct for it by imaging a photographic gray card at the location of the Macbeth color checker. Pixel data from the gray card is used to compute scale values to correct for falloff.

**Global Error Correction.** This final step minimizes color error globally. We image the Macbeth color checker with each camera at the initial chosen exposure, and we store RGB data for each patch. If we are performing a falloff correction, we scale the RGB data with the values from the previous step. We then average these values across all cameras. For each camera, we then compute a 3x4 transform to match its recorded color patch values to the averaged values. We use averaged values to avoid accidentally picking an outlier camera as a reference. We save these 3x4 transforms to apply on filmed data.

### 2.2 Partially-Overlapping Fields of View

When calibrating a multi-camera setup with partially-overlapping views, we cannot place a color chart such that it can be seen by all cameras at one time. In this regime, we choose to

perform only the configuration stage with no sensor characterization. Instead of using the white patch on the Macbeth chart for gain and offset configuration, we recorded data from the center of each camera's image when viewing a large white target placed close enough to the cameras to fill their fields of view.

### 3 Results

We calibrated 95 cameras with overlapping fields of view using three different methods:

1. Using software auto-gain and auto-white balance to set camera gains and offsets [Nanda and Cutler 2001]
2. Setting identical (default) color gains and offsets for all cameras, followed by 3x4 transform from camera RGB to the XYZ color space. This is computed in a least squares sense based on the known XYZ values for the Macbeth color checkers.
3. Our method.

Table 1 shows the results for these methods. We show the RMS error across all color checkers and all cameras, as well as the maximum pixel value difference between any two cameras for any patch on the color checker. To relate matching errors in the XYZ color space to RGB, we convert to sRGB, a standardized color space. This introduces errors due to gamut clipping.

We see that using automatic gain and white balance controls performs the worst. These controls are based on each camera's image statistics, so differences in each camera's view of the scene causes variations in their color settings. Configuring all of the cameras with the same default gain and offset settings, even without matching to the XYZ reference values from the color checker, is much better. Naturally, matching with the reference values reduces the RMS error to roughly three gray levels for this dataset. Our iterative gain and offset adjustment, even without post-processing, significantly outperforms both of these methods. Our post-processing pipeline reduces the residual error by another 30 %.

To visually evaluate the consistency of color calibration results, we created single composite images of the Macbeth color checker from multiple cameras. These composites are representative of image reconstructions in image-based rendering; however, they are a harsher test as there is no interpolation or blending between camera contributions. Figure 1 shows composites from uncalibrated data and data calibrated with three methods with software auto-gain and white-balancing, individually matching each camera to the MacBeth XYZ values, and our method. There are significant artifacts in the composites from uncalibrated data and from auto-gain and white-balanced data. Matching individually to XYZ reduces the errors, and with our method they are nearly imperceptible. These visual results mirror the errors statistics in Table 1.

Figure 2a shows the same test applied to six cameras in a natural scene. There are slight color differences on the face, outstretched arm, and centered green part of the soccer jersey. The greater mismatches in the lower left are due to the highly specular jersey. In figure 2b, we show a different combination of the input images. The image is assembled by interleaving 10-pixel wide rows from the source images with 50% overlap and blending the results. This is the same resampling Wilburn et al [2004] used for overcoming artifacts in their high-speed video work. Blending the images renders the color variations invisible.

Figure 3 shows results using our calibration for cameras with non-overlapping fields of view. Here, we have created a mosaic

using images a densely packed 12x8 camera array. The cameras have 50% overlapping fields of view and telephoto lenses, so any point in the scene is viewed by at most four cameras. The panorama constructed from uncalibrated data without image blending has low contrast, poor color balance, and obvious color differences between cameras. Blending makes the transitions from camera views less noticeable, but the color differences are still visible. With color calibration and blending, the results are more pleasing. The color calibrated images have much more uniform color responses, and the variations are less noticeable in the blended image.

### 4 Conclusions

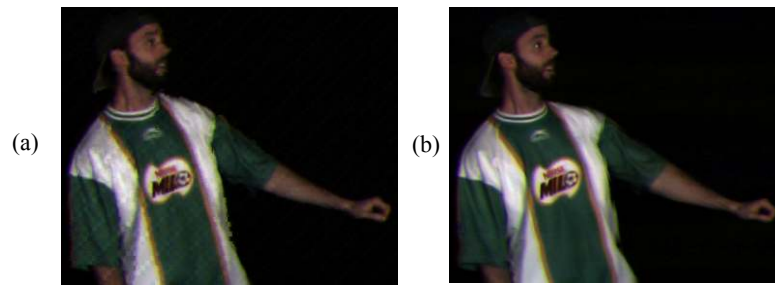
We have presented a simple, automated color calibration pipeline for large camera arrays. We describe how to handle both overlapping and non-overlapping fields of view. We take care to avoid introducing errors due to radiometric falloff, non-uniform illumination, and sensor non-linearity. Calibrating the sensors to match each other, rather than a standard, yields better color matching between cameras, and using calibration targets prevents scene reflectance from biasing our camera calibration. Our results indicate that our method is accurate enough for image-based rendering applications. The process is completely automatic and runs on an array of 100 cameras in just a few minutes.

One remaining question is how much of our color calibration errors are fundamental. One source of error in our computed 3x4 color correction matrices is the non-zero specularities of some of the Macbeth color patches. We have observed that many of our worst-case errors occur for the more specular patches on the color checker. For example, the white and bright orange patches, which show some of the most noticeable artifacts in figure 1, are two of the most specular patches according to our measurements. One way to reduce these errors is to average images of the color checker taken with varying illumination. Although our color response characterization is relatively robust to radiometric falloff because we calibrate the center of our images, we will not be able to match colors in the periphery of our images without characterizing falloff.

Our color calibration works well for applications that blend images together and for more sensitive methods, such as those that use optical flow. Some applications, however, transform small color variations between cameras into coherent patterns that are more obvious. Color variations in figure 2b are barely discernable, even though the rows are resampled from different pairs of images. If we were to make a video in which the mapping of rows to cameras move in a coherent fashion (sliding down the image as in the resampling shown by Wilburn et al. [2004]), the variations would become immediately obvious as a moving pattern superimposed on the static image. Some of these errors are due to residual calibration errors, but others are unavoidable. Specular surfaces will look different from different positions. This suggests that in addition to improving our color calibration, we must also develop algorithms that prevent color variations from being presented coherently to the user.

### 5 Acknowledgements

The authors would like to thank the students of the Stanford Multi-Camera Array Project for a number of useful suggestions. Construction of the camera array used in this work was funded by Intel, Sony, and Interval Research. This work was supported by the NSF under contracts IIS-0219856-001 and DGE-0333451 and DARPA under contracts NBCH-1030009 and F29601-01-2-0085.



**Figure 2:** Image reconstruction. (a) Image composite using 5x5 pixel blocks from 9 cameras. Just as in Figure 1, blocks along a diagonal are from one camera. There are minimal errors on the face, outstretched arm, and center green part of the jersey. Errors are visible in the lower-left due to specularities and falloff. (b) An image from high-speed video using the same dataset. This image is constructed from a slice through a video cube. With our calibration and interpolation in the slicing, color variations are almost imperceptible within the image.



**Figure 3:** High-resolution image mosaics. Top Left: No color calibration and no blending. Top Right: With blending between images from each camera, the image seams disappear. Bottom Left: Using our method for gain and offset configuration, some image seams are visible, although they are not very harsh. Bottom Right: Blending between camera images removes the remaining seams.

## References

- BOUGUET, J.-Y. Camera Calibration Toolbox for Matlab. <http://www.vision.caltech.edu/bouguetj/calib.doc>.
- MATUSIK, W., PFISTER, H. 2004. 3D TV a Scalable System for Real-Time Acquisition, Transmission and Autostereoscopic Display of Dynamic Scenes. *Proceedings ACM SIGGRAPH 2004*.
- LEVOY, M. AND HANRAHAN, P. 1996. Light Field Rendering. *Proceedings of ACM SIGGRAPH 1996*.
- NANDA, H. AND CUTLER, R. 2001. Practical Calibrations for a Realtime Digital Omnidirectional Camera. *Proceedings of CVPR 2001*. Technical Sketch.
- RANDER, P., NARAYANAN, P., AND KANADE, T. 1997. Virtualized reality: Constructing time-varying virtual worlds from real events. *Proceedings of IEEE Visualization*, 277-283.
- TSAI, R. 1987. A Versatile Camera Calibration Technique for High Accuracy 3D Vision Metrology Using off-the-shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*. 3, 4, 323-344.
- VAISH, V. WILBURN, B. JOSHI, N. AND LEVOY, M. 2004. Using Plane + Parallax For Calibrating Dense Camera Arrays. *Proceedings of CVPR 2004*.
- VEDULA, S. 2001. Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events. *CMU, Tech Report*, CMU-RI-TR-01-37, Robotics Institute.
- WILBURN, B., JOSHI, N., VAISH, V., LEVOY, M., AND HOROWITZ, M. 2004. High Speed Video Using a Dense Array of Cameras. *Proceedings of CVPR 2004*.
- WILBURN, B., JOSHI, N., VAISH, V., TAVALA, E.-V., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High Performance Imaging Using Large Camera Arrays. *Submitted to ACM SIGGRAPH 2005*.
- YANG, J.C., EVERETT, M., BUEHLER, C., AND MCMILLAN, L. 2002. A Real-Time Distributed Light Field Camera. *Eurographics Symposium on Rendering*.
- ZHANG, C. 2005. *Personal Correspondence*.
- ZHANG, C. AND CHEN, T. 2004. A Self-Reconfigurable Camera Array. *Eurographics Symposium on Rendering*.
- ZHANG, Z. 1999. A Flexible New Technique for Camera Calibration. *Proc. International Conference on Computer Vision 1999*.