

# Relative Impact of Key Rendering Parameters on Perceived Quality of VR Imagery Captured by the Facebook Surround 360 Camera

Nora Pfund, Nitin Sampat, and J A Stephen Viggiano

School of Photographic Arts and Sciences

Rochester Institute of Technology (USA)

## Abstract

High quality, 360 capture for Cinematic VR is a relatively new and rapidly evolving technology. The field demands very high quality, distortion-free 360 capture which is not possible with cameras that depend on fish-eye lenses for capturing a 360 field of view. The Facebook Surround 360 Camera, one of the few “players” in this space, is an open-source license design that Facebook has released for anyone that chooses to build it from off-the-shelf components and generate 8K stereo output using open-source licensed rendering software. However, the components are expensive and the system itself is extremely demanding in terms of computer hardware and software. Because of this, there have been very few implementations of this design and virtually no real deployment in the field. We have implemented the system, based on Facebook’s design, and have been testing and deploying it in various situations; even generating short video clips.

We have discovered in our recent experience that high quality, 360 capture comes with its own set of new challenges. As an example, even the most fundamental tools of photography like “exposure” become difficult because one is always faced with ultra-high dynamic range scenes (one camera is pointing directly at the sun and the others may be pointing to a dark shadow). The conventional imaging pipeline is further complicated by the fact that the stitching software has different effects on various aspects of the calibration or pipeline optimization. Most of our focus to date has been on optimizing the imaging pipeline and improving the quality of the output for viewing in an Oculus Rift headset. We designed a controlled experiment to study 5 key parameters in the rendering pipeline – black level, neutral balance, color correction matrix (CCM), geometric calibration and vignetting. By varying all of these parameters in a combinatorial manner, we were able to assess the relative impact of these parameters on the perceived image quality of the output.

Our results thus far indicate that the output image quality is greatly influenced by the black level of the individual cameras (the Facebook camera comprised of 17 cameras whose output need to be stitched to obtain a 360 view). Neutral balance is least sensitive. We are most confused about the results we obtain from accurately calculating and applying the CCM for each individual camera. We obtained improved results by using the average of the matrices for all cameras. Future work includes evaluating the effects of geometric calibration and vignetting on quality.

## Introduction

The Facebook Surround 360 Camera System is composed of 17 individual cameras, including 3 fisheye lenses, that work together to form a 360 image and/or video content. During the render process, there is a file used in order to control correction of color, dark noise, vignetting and other factors that affect the rendering process. The values in this file are extremely important for the output quality

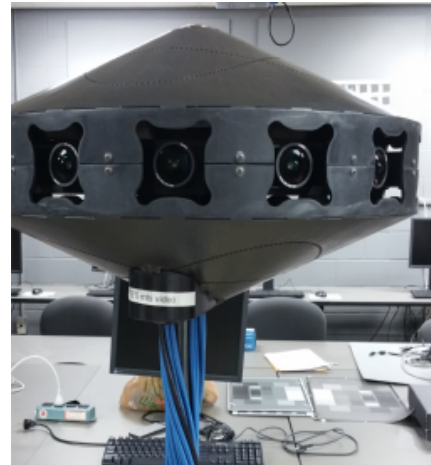


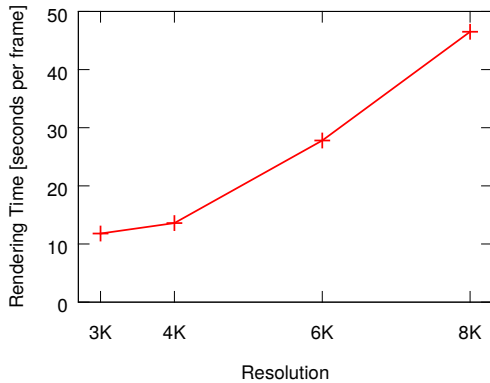
Figure 1. Surround 360 camera at RIT

of the rendered capture data. This project focused on selecting these numbers in order to produce quality result.

Facebook designed this product to have people in the field of cameras, photography and videography shoot footage in this format and share it on Facebook. There was no intention of selling the product for profit. The system is actually a “do-it-yourself” type of project. Facebook has the hardware design and software under open-source licensing, [1] so anyone can view what materials they need to buy, how to put the product together, and what software to use to get it working.

Because of the open nature of the project, it is hard to know how many people actually possess on one of these systems. Although there are a few Facebook groups revolving around this topic, there does not seem to be much content generated from the product. There are people discussing it and posting pictures of them shooting, however, it seems as though there are some bottle necks in the rendering process that hinder one’s ability to easily produce an accurate and high quality output. This is what we worked on to try to help solve that problem.

The rendering process of this system involves unpacking raw image files from a pair of merged files, demosaicing the raw frames, inferring structure from motion and optical flow, stitching the individual frames together into a large frame, generating stereo pairs, and encoding the stereo frames as a video. Because of bandwidth and speed issues (each second captured generates approximately 3 gigabytes of data; rendering takes much longer than capture), rendering is performed after capture is completed, and is accomplished using one of the open-source tools from the Facebook-provided suite.



**Figure 2.** Rendering time per frame, as a function of frame resolution

Rendering time varies quadratically with respect to final frame resolution, as is apparent from Figure 2.

### Statement of problem

We encountered problems with the quality of the rendered video when using rendering settings provided by Facebook (though for a different instance of the camera). Figure 3 shows two frames from early rendering efforts. This convinced us that a systematic investigation of rendering settings was necessary.

### Experimental

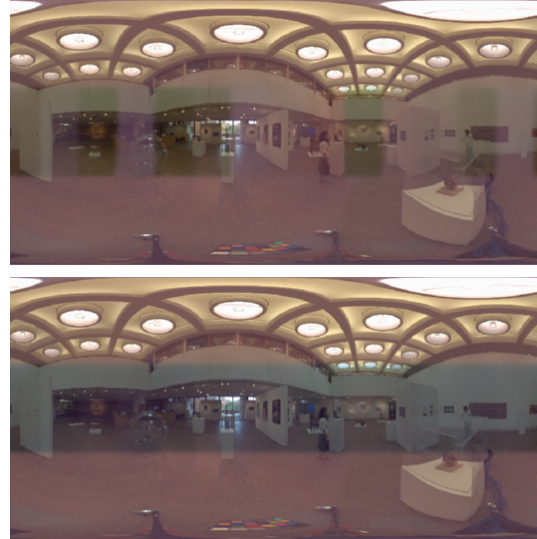
In a systematic experimental design, the experimenter must identify the variables to be manipulated (the “factors”). [2, chapters 10 and 11] The factors were selected from settings files that controlled the rendering process, one file for each camera, written in Javascript Object Notation (JSON). We referred to these files as “JSON files” for convenience. The factors we adjusted were the Black Level, the White Balance, and the Color Correction Matrix (CCM). Other factors considered were Falloff (vignetting) Correction and Geometry Correction. These were held constant during the experiment.

Not only must the factors be identified in an experimental design, but the values for each must be specified. While these may be set a priori, before beginning the experiment, they may be varied during the progress of an experiment, particularly in experiments with open-loop feedback, such as response surface exploration. [3] The specific values for a factor are referred to its “levels” in the nomenclature or experimental design. [2, *ibid*] The levels for each of the three principal factors are enumerated below:

#### Factor 1: Black Level setting

One value (as an unsigned 16-bit integer) is specified for the red black level, one for green, and one for blue.

- Level 1:** Obtain the black levels from one camera, and use them for all cameras (“1 for all;” baseline)
- Level 2:** Obtain black levels from all cameras, average them across cameras, and use the same values for all cameras (“Average”)
- Level 3:** Obtain RGB black levels for each camera, and use that camera’s own black level (“Camera-specific”)



**Figure 3.** Early renderings.

*Frames from videos rendered with generic settings. Top frame shows vertical banding, with boundaries between side cameras apparent. Bottom frame shows horizontal banding, with boundaries between top and side cameras, and side and bottom cameras, most apparent.*

#### Factor 2: White Balance

Three floating point multipliers (one for each of red, green, and blue) are specified.

- Level 1:** 1.0, 1.0, 1.0 (“ones;” baseline)
- Level 2:** Determine white balance for one camera, apply same values to all cameras (“1 for all”)
- Level 3:** Determine white balance for each camera and apply it to that camera (“Camera-specific”)

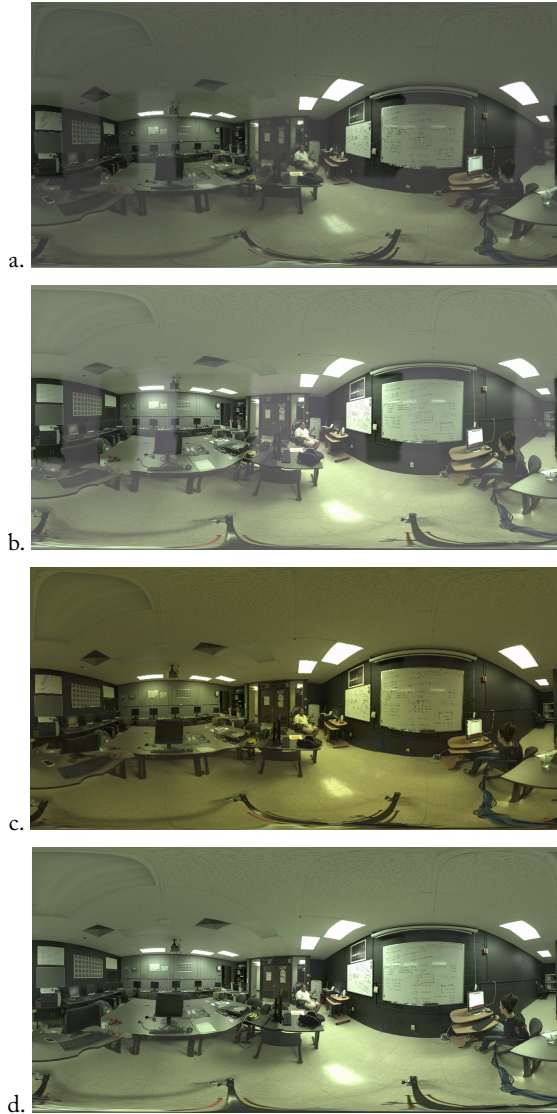
#### Factor 3: Color Correction Matrix

A 3×3 color correction matrix is specified in floating point. Facebook recommends the rows sum to one so the matrix does not materially affect neutral balance.

- Level 1:** Identity matrix (“identity matrix;” baseline)
- Level 2:** Compute matrix for each camera, apply average matrix to all cameras (“Average”)
- Level 3:** Compute matrix for each camera and apply it to that camera (“Camera-specific”)

For falloff correction we employed values provided by the Facebook team. The geometric correction parameters used for all runs were determined using the procedure recommended by Facebook.

We did not vary other settings in the JSON files. Our experiment was further controlled by using the same raw capture files for each experimental run. This precluded factors such as lighting, frame rate, temperature, or others from varying from run to run.



**Figure 4.** Selected renderings for standard scene.

Baseline rendering is shown in (a). In (b), camera-specific white balance is used, with other factors at baseline. In (c), camera-specific black level settings are used. (d) used camera-specific settings for both black levels and white balance. Together, these images are the responses for a 2x2 factorial experiment with the factors white balance and black level.

**Table 1. Image quality by black level and white balance settings.**

These runs were made with color correction matrix at baseline (identity matrix), Facebook-supplied falloff correction, and RIT-generated geometry correction. The numbers in the body of the table are subjectively assigned quality ratings, on a scale of 0 (worst) to 10 (best).

White Balance level	Black Level		
	“1 for all”	“Average”	“Camera-Specific”
Ones	3; vertical bands at transitions between side cameras objectionable	2; vertical bands objectionable	6; no banding visible; color is incorrect
One for all	(not run)	2; vertical bands objectionable	8
Camera-Specific	3; Some banding visible	2; vertical bands objectionable	9; banding not detected; good consistency between cameras; colors not completely accurate

## Results and Discussion

In all of our testing, the black level values were the rendering parameters with the greatest influence on the quality of the stitched output image. This was found across different scenarios and lighting conditions as well as a variety of testing combinations. These values must be obtained at each individual capture location in order to account for sensor temperature. This also requires inputting the values into the JSON files of each camera before going through the rendering process. This allows for the highest amount of consistency between cameras. The procedure resulting in the most consistency in the output image used the black level values specific to each individual camera in the system. The effect of black level setting may be evaluated by comparing subfigures (a) and (c) in Figure 4.

White balance did not show as much importance as the black level in terms of camera consistency. However, these values proved more important for accurate reproduction of colors. The black level adjustment allows for matching between all 17 cameras, but white balance adjustment allows for more accurate color balance of the captured scene. The setting resulting in the most accurate output image used white balance values specific to each camera. The JSON file for each camera was given the RGB values needed to adjust the white balance of the scene based on that camera’s response to a gray card in its field of view. The effect of white balance setting may be evaluated by comparing subfigures (a) and (b) in Figure 4.

Based on our results, the effect of the color correction matrix (CCM) is not as apparent as the other two parameters tested. However, again, in terms of accurate color reproduction, it allows for more color accuracy than simply adjusting the white balance. Adjusting white balance results in somewhat accurate colors, but adjusting both the white balance and CCM results in even more accurate colors. The procedure resulting in the most accurate output

image used a CCM that was an average matrix calculated from all the CCMs of the 17 cameras of the system. This combination of individual black level values, single-camera white balance values, and average CCM values produced the most consistent and accurate output image. Using camera-specific CCMs resulted in increased banding artifacts; an average CCM computed from all cameras produced the best result.

## Conclusions

Of the factors investigated, the black level appeared to have the greatest impact on image quality. Unless black levels specific to each camera were used, the boundaries between cameras in the stitched images were visibly obvious and objectionable. Even with baseline settings for the other factors, using camera-specific black level compensation produced an acceptable visual state (refer to Figure 4c).

Because the RGB black levels are dependent on temperature and shutter speed, and vary from one camera to the next, a black frame needs to be taken for each camera on location for each shoot, at the same shutter speed used for the capture.

Regarding white balance, the best results were found when using RGB white balance values that were obtained from a single camera. This is done by using a gray card and compensating for the white balance of the scene (refer to Figure 4b).

When the system was originally defined, no camera with masked pixels seemed to offer the resolution, size, framerate, and transfer speed that cinematic-quality full-sphere VR capture requires. Now that cameras with masked pixels that meet these requirements are available, they should be considered for new builds of this system so that the quality-critical compensation for each camera's specific black levels may be more conveniently performed.

## Future Work

Our plans include a closer look at the CCM settings, and investigation of the geometric and falloff (vignetting) settings. We are also considering solutions for increasing dynamic range to permit better capture of daytime outdoor scenes.

## Acknowledgements

The authors gratefully acknowledge the generous support of Brian Cabral of Facebook for funding the construction and evaluation of the Surround 360 system at RIT. We also thank RIT students Nick Franco and Andrew Palmer for their many contributions.

## References

- [1] <https://github.com/facebook/Surround360>, last updated December 2017.
- [2] Jay L Devore, *Probability & Statistics for Engineering and the Sciences*, Eighth edition. Boston, MA: Brooks/Cole, 2012.
- [3] George E P Box, William Gordon Hunter, and J Stuart Hunter, *Statistics for Experimenters*. New York: Wiley, 1978.

## Author Biography

*Nora Pfund is from Woodbridge, Connecticut. She is in her fourth year at Rochester Institute of Technology studying Imaging and Photographic Technology. Throughout her time in school, Nora has gained experience in digital photography and image quality testing. A cooperative experience in Summer 2016 allowed Pfund to gain professional experience in the field. A research position in Summer 2017 allowed her to progress in real-world applications of 360 imaging. In addition to her major, Pfund also has two minors in Imaging Systems and Psychology.*

*After graduation Nora plans to apply her knowledge in an image quality environment.*

*Nitin Sampat is a professor in the School of Photographic Arts and Sciences at RIT where he teaches and conducts research in the topics of image quality, sensor and camera characterization, color management and digital image processing. He is an active consultant to leading imaging companies.*

*JA Stephen Viggiano is Assistant Professor of Photographic Sciences at Rochester (NY) Institute of Technology, teaching classes in photographic optics, programming, and digital image processing, and guiding undergraduate researchers. Steve has a PhD in Imaging Science from RIT, as well as Master's degrees in Statistics and Printing Technology. Prior to joining Photographic Sciences, Professor Viggiano taught at RIT's School of Mathematical Sciences, and was with RIT Research Corporation until its closing. Steve holds a US Amateur Extra-class radio license, and is an ARRL-recognized instructor.*