Recent Advances in Smartphone Computational Photography

Paul Friederichsen
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
fried701@morris.umn.edu

ABSTRACT

Smartphone cameras present many challenges, most of which come from the need for them to be physically small. Their small size puts a fundamental limit on their ability to resolve detail and collect light, which makes low-light photography and zooming difficult. This paper presents two approaches to improve smartphone photography through software techniques. The first is handheld super-resolution which uses natural hand movement to improve the resolution smartphone images, especially when zoomed. The second approach is a system which improves low light photography in smartphones.

Keywords

computational photography, image processing, low-light imaging, photography, super-resolution

1. INTRODUCTION

Smartphone cameras use very small sensors with fixed apertures. This means their ability to gather light is significantly reduced compared to larger dedicated cameras. Most modern smartphones use a burst system to capture multiple images and merge them together to improve image quality.

In this paper, I first introduce some background information concerning various photography and signal processing terms. I then introduce and discuss two approaches to improving smartphone photography that augment the existing burst pipeline. The first uses hand movement to increase spatial resolution (Section 3). The second approach is a a system that uses a number of techniques to improve the low-light abilities of smartphone cameras (Section 4).

2. BACKGROUND

In this section I explain several key concepts that are important for the work in Sections 3 and 4. This includes Burst photography (Section 2.1), Bayer filters (Section 2.2), demosaicing and aliasing (Section 2.3), super-resolution (Section 2.4), and kernels (Section 2.5).

UMM CSci Senior Seminar Conference, October 2020 Morris, MN.

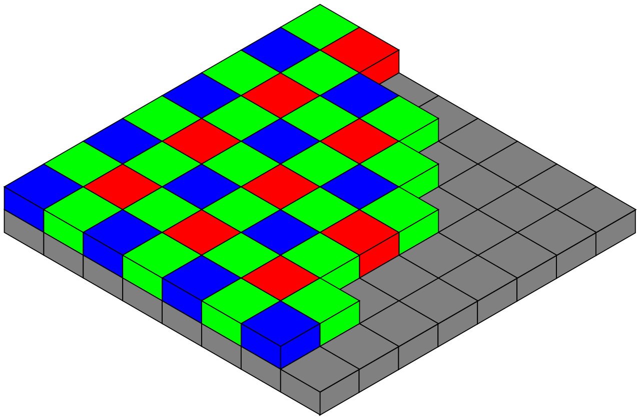


Figure 1: A Bayer pattern on a sensor in isometric perspective. [8]

2.1 Burst Photography

An image processing pipeline refers to the process the raw data from a camera sensor goes through to be turned into the final image file that can be displayed and shared. Most smartphones use a burst processing pipeline for their cameras. Generally, burst processing involves taking a series of exposures and merging them together to form the final image. Most smartphones operate in a *zero-shutter lag* mode by default. In this mode, raw frames (the full unprocessed sensor output) are continuously captured to a temporary area in memory while the camera app is open. When the user presses the shutter button, several of the most recent frames are sent to the camera processing pipeline to be merged.

Both of the approaches in this paper build on the end-to-end burst processing pipeline from Hasinoff et al. [3] which used bursts of constant low-exposure frames to increase dynamic range and signal-to-noise ratio. It specifically uses under-exposed frames to reduce motion blur.

2.2 Bayer Filters

A Bayer filter is a type of color filter array (CFA). CFAs are needed because the light sensing components in a digital image sensor can only detect the presence of light and not what specific color it is. The majority of digital image sensors in digital cameras and phones use a Bayer filter mosaic pattern to arrange RGB color filters on the sensor (Figure 1). The pattern consists of 50% green, 25% red, and 25% blue pixels. This ratio emulates the color sensitivity of the hu-

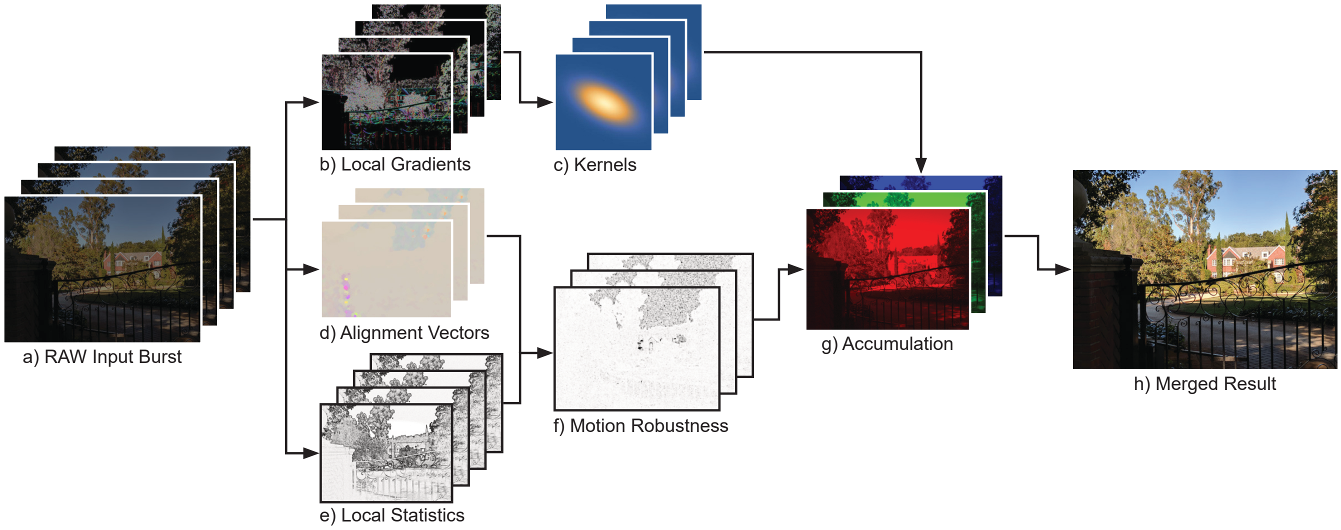


Figure 2: An overview of the approach used by Wronski et al. [12]. The initial burst of input frames (a) are aligned (d) to a base frame. The local features in the frames (b) are used to create kernels (c) (Section 3.3) which are used along with the motion robustness model (f) (Section 3.4) to combine (g) the frames separately for each color channel. The final image (h) is produced by normalizing the results of each channel.

man eye, which is more sensitive to green than it is to blue and red. Due to this pattern on the sensor, the raw output of a digital camera also has each pixel filtered to only red, green, or blue and a demosaicing algorithm must be used to interpolate the other values for each pixel. [8]

2.3 Demosaicing and Aliasing

A demosaicing algorithm reconstructs a full color image from the separate pixels that have been filtered with a CFA to just one channel: red, green, or blue [9]. There are many methods for this; the simplest ones interpolate the values for the other two color channels of a given pixel based on nearby pixels from the CFA image of those colors.

This process means two-thirds of the final image is reconstructed from the available data. The demosaicing process may introduce various artifacts in the final image due to aliasing, an effect that happens when the camera sensor is unable to correctly represent the patterns and details present in a scene due to its resolution. One type of artifact this causes are Moiré patterns, a type of interference pattern. [13]

2.4 Super-resolution

Super-resolution is any technique which increases the resolution of an image. While there are techniques that work on a single image, we will focus on using multiple frames for super-resolution. To do this each frame must have some new information to contribute to the final image. This is achieved by capturing multiple aliased input frames that are sampled at different subpixel offsets, meaning the individual pixel points in each frame capture a different sample of the area the pixel represents [12].

2.5 Kernels

A kernel (or convolution matrix) in image processing refers to a small matrix that is used to apply effects like blurring or to detect features in an image, such as edges. The kernel is applied by doing a convolution, adding each pixel of the image to its local neighbors, weighted by the kernel. [10]

For example, the following applies a kernel (the first matrix) to an image piece (the second matrix):

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

The pixel in the center of the final image (coordinates $[2, 2]$) would be this combination of the values of the image matrix weighted by the kernel matrix:

$$(i\cdot 1)+(h\cdot 2)+(g\cdot 3)+(f\cdot 4)+(e\cdot 5)+(d\cdot 6)+(c\cdot 7)+(b\cdot 8)+(a\cdot 9)$$

3. HANDHELD SUPER-RESOLUTION

Wronski et al. [12] introduced an algorithm that uses multiple shifted frames to produce higher resolution images from bursts of underexposed raw frames as part of the smartphone's imaging pipeline. The algorithm is able to directly use Bayer raw frames and removes the need for an explicit demosaicing step in the pipeline. It uses natural hand motion and is efficient enough to work in the background on smartphones. This algorithm is used as the merging algorithm in the camera pipeline of the Google Pixel 3 and newer and is what allows for the Pixel's "super-res zoom" feature. This section provides an overview and some results of the algorithm by Wronski et al. [12].

3.1 Algorithm Overview

The approach by Wronski et al., as shown in Figure 2, is a process that starts with the the acquisition of a burst from the continuous ring buffer of raw frames in the phone's camera application. Next, a single frame is chosen and the rest are aligned to it using a refined version of the algorithm by Hasinoff et al. [3]. Each frame's local contributions are estimated through kernel regression (Section 3.3) and accumulated across a whole burst for each of the three color

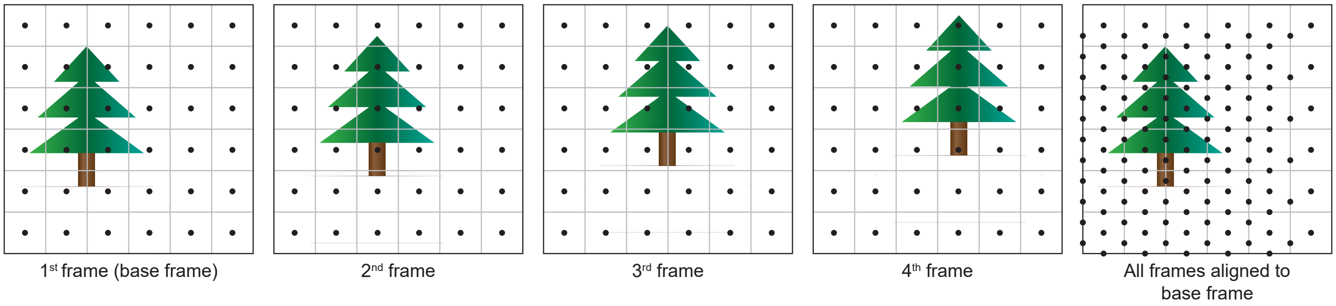


Figure 3: An illustration of subpixel displacements from a burst of four frames with linear hand motion. Each frame is offset by half a pixel on the x-axis and a quarter pixel on the y-axis from the last frame. After alignment, the pixel centers (black dots) uniformly cover the image with greater density than a single frame. [12]

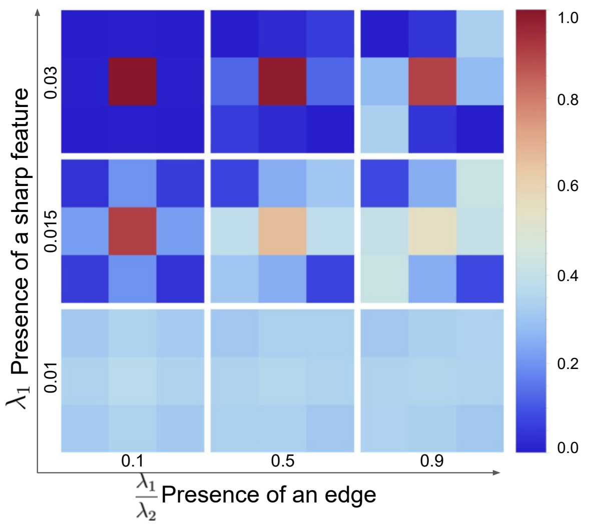


Figure 4: Plots of relative weights as a function of local features. The scale on the right indicates the weight given to each pixel in the 3×3 neighborhood. [12]

planes. This involves the kernel shapes being adjusted based on estimated local gradients and, at the same time, the sample contributions are adjusted weighted based on a statistical robustness model (Section 3.4).

The final RGB image is obtained by normalizing the accumulated contributions for each of the three color planes and merging them together. This can then be sent to the rest of the imaging pipeline.

3.2 Hand Movement Based Super-resolution

One of the important conditions for multi-frame super-resolution is that the input contains multiple images that are sampled at different subpixel offsets. When someone is holding an object there is a natural and involuntary slight hand movement present. Wronski et al. analyzed hand movement in a set of 86 bursts captured by 10 different users during regular smartphone photography using the rotational measurements from the phone's gyroscope. They determined this periodic, random movement while the camera is capturing a burst frames provides sufficient subpixel coverage to create a super-resolution image.

3.3 Kernel Reconstruction

The core idea of the algorithm is considering the pixels of multiple raw frames with hand motion as randomly offset, aliased, and noisy measurements of three original continuous signals, one for each color channel. The algorithm creates the final output image pixel-by-pixel. For each output pixel, it evaluates the local contributions to each of the three color channels from different input frames since each raw image pixel is specific to a single color channel.

The process for each pixel in each color channel can be described as:

$$C(x, y) = \frac{\sum_n \sum_i c_{n,i} \cdot w_{n,i} \cdot \hat{R}_n}{\sum_n \sum_i w_{n,i} \cdot \hat{R}_n}$$

(x, y) refers to the coordinates of the pixel.

$\sum_n$ is a sum over all contributing frames.

$\sum_i$ is the sum over samples (pixels) in a local 3×3 neighborhood centered on the target pixel.

$c_{n,i}$ is the value of the individual pixel at frame n and sample i.

$w_{n,i}$ is the local sample weight for the pixel at frame n and sample i, which is described below.

$\hat{R}_n$ is the local motion robustness score at (x, y) described in Section 3.4.

The local pixel weights $(w_{n,i})$ come from a kernel which is calculated for the 3×3 neighborhood around the pixel at (x, y).[1] It produces a 3×3 matrix of values between 0 and 1 taking into account edges and sharp features in the local area. These weights determine how much each pixel in the 3×3 area will contribute to the target pixel for that frame. An example of the matrices that result from the kernel function can be seen in Figure 4. Here the presence of a sharp feature produces a matrix where more emphasis is put on just the center pixel.

[1] Specifically a 2D unnormalized anisotropic Gaussian radial basis function kernel.

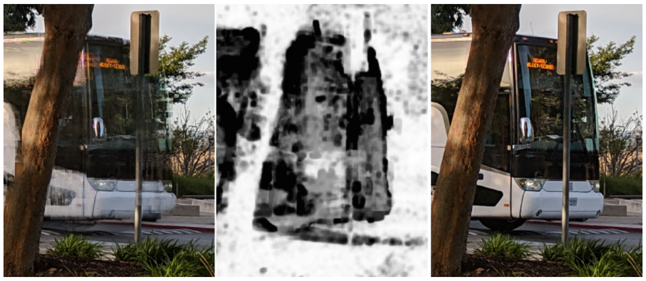


Figure 5: A photograph of a moving bus: **Left**: Without a motion robustness model there are alignment errors and occlusions that result in tiling and ghosting artifacts. **Middle**: The robustness mask produced by the robustness model. White regions are those with all frames contributing to the final merged image and darker regions are those with a lower number of contributing frames. **Right**: The result of using the robustness model when merging frames. [12]

3.4 Motion Robustness

It is difficult to reliably align the sequence of images in a burst and even assuming a perfect alignment, changes in the scene and occlusion would still result in some areas of the scene being poorly represented in many frames of the burst. This needs to be taken into account to prevent severe artifacting (Figure 5 Left). To combine frames robustly, a confidence level is assigned to the local neighborhood of each pixel; the map of these confidences is called a *robustness map*.

This confidence level is assigned by computing the standard deviation in the image and a color difference between the base frame and the aligned input frame. Regions of this frame with differences smaller than the local standard deviation and those that are close to a pre-defined fraction of the spatial standard deviation will be merged while larger differences are likely non-aligned motion and are discarded.

An example of what this can be seen in the middle image of Figure 5, where the dark areas have a lower robustness value and the lighter areas have a greater robustness value.

3.5 Results

Wronski et al. used a variety of methods, both numerical and visual, to evaluate their algorithm and compare it with other demosaicing, merging, and super-resolution techniques. They also compared their method as part of a full camera pipeline to that of Hasinoff et al. [3]. Figure 6 shows one of these comparisons.

4. HANDHELD LOW LIGHT PHOTOGRAPHY

Liba et al. [6] introduce a system for capturing photos in very low light that produces improved color and less noise. This system is used for the "Night Sight" feature on Google Pixel phones. The system builds on the existing burst pipeline (from Hasinoff et al. [3]) and uses the burst merging algorithm from Wronski et al. [12] with modifications to improve low-light photography. It uses a *positive-shutter-lag* mode where the camera waits until the shutter button is pressed to capture the burst of frames. This is used rather than the *zero-shutter-lag* mode used for daylight photography to allow for longer exposures.

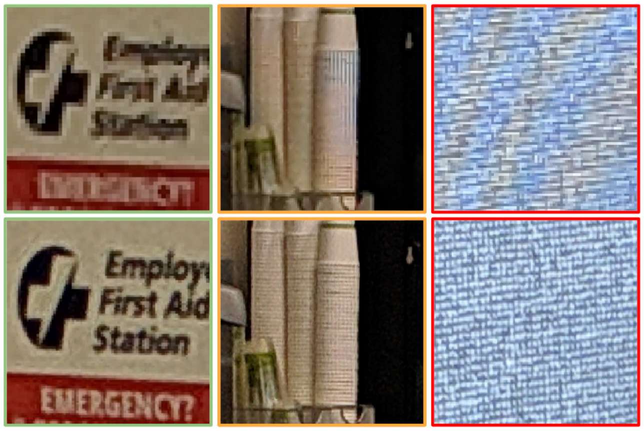


Figure 6: A comparison of the traditional frame merging approach (**top**) and the approach by Wronski et al. (**bottom**). The super resolution merge algorithm improves sharpness and detail while eliminating the Moiré aliasing artifacts (false color banding) seen in the third (**red**) frame. [12]

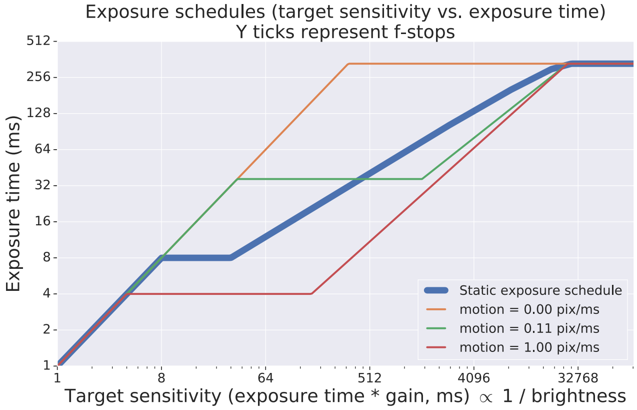


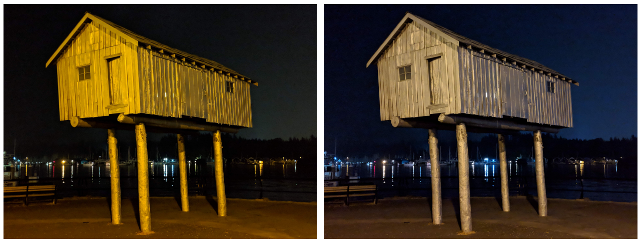
Figure 7: A traditional static exposure schedule compared to the dynamic exposure schedule of Liba et al. at various levels of motion. [6]

The main improvements to the camera pipeline from this research are the use of "motion metering" (Section 4.1) to calculate the exposure settings for each frame based on predicted motion in the scene and camera, a learning-based low-light optimized auto white balance algorithm (Section 4.2), and tone mapping (Section 4.3) to produce better colors in low light.

4.1 Motion Metering

When capturing a burst of images on a smartphone, the exposure time and sensor gain (the sensitivity of the sensor to light, also known as ISO) needs to be selected for each frame. Liba et al. use the same strategy as in [3] and capture all frames in the burst with the same exposure time and ISO. For effective low-light photography these settings need to be automatically selected within the constraints of keeping the total capture time low ($\leq$ 6 seconds) and the total number of frames within the device's memory limits.

This process consists of splitting the target sensitivity, which is based on the brightness of the scene, into expo-



(a) Pixel default AWB (b) Liba et al.

Figure 8: A comparison of the default implementation of FFCC in the Pixel's camera and the low-light optimized version by Liba et al. [5]

sure time and gain and calculating the number of frames with respect to the time and memory limits. The traditional method used in Hasinoff et al. [3] uses a fixed "exposure schedule" that simply keeps the exposure time low to limit motion blur. This method usually works well but can be improved for low-light photography.

The "motion metering" described by Liba et al. selects the exposure time and gain by predicting future motion in the scene and motion of the camera itself. It produces a variable exposure schedule that varies based on the amount of motion detected (Figure 7). This is used to select longer exposures for scenes with no motion and shorter exposures for those with motion to reduce motion blur when needed and increase signal-to-noise ratio when possible.

Additionally, experienced photographers often brace their device against a surface or put it on a tripod in low light. Liba et al.'s motion metering system is able to use even longer exposures (up to 1 second) by detecting this using measurements from the device's gyroscope.

4.2 Auto White Balance in Low Light

Humans perceive color correctly even when objects are lit with colored light, an ability called color constancy. This perception can break down when a photograph is taken under one type of light and viewed under different light, the image can look tinted (Figure 8a). Cameras correct for this by determining the color of the majority of the illumination in the scene and correcting the colors in the image such that they appear to be lit by a neutral (white) illumination. This Automatic White Balance (AWB) step in the camera pipeline is important to produce a pleasing image. [5]

The current best-performing color constancy algorithm is the machine learning based "Fast Fourier Color Constancy" (FFCC) [1]. As described in this section, Liba et al. trained the FFCC algorithm with a new dataset and error metric to better handle challenging low-light scenes (Figure 8).

Liba et al. collected 5000 images using mobile devices of various scenes that demonstrate a range of light levels to be able to train their implementation of FFCC. To obtain better ground truth coloring for the training data they had real professional photographers manually white balance the images with the most "aesthetically preferable" white balance for the scene rather than use a color checker or grey card to empirically measure the "true" white balance.

Additionally, Liba et al. developed a new error metric for training the model that better deals with heavily tinted illuminants which are common in night scenes (like those from colorful neon lights). The issue with traditional error metrics in low light is that they are based on how well the algorithm recovers *all* of the color channels of the illuminant. This works well for brighter scenes with close to white true illumination but in dark scenes with heavily tinted illuminants the white balanced image may contain pixel values where a single color channel's values are near zero for the whole image. Such an image would look the same under all possible transformations of that channel. When a color channel is "missing" it can produce low accuracy results from the error metric. It is also unclear how to set the missing channels in the ground-truth illuminant data.

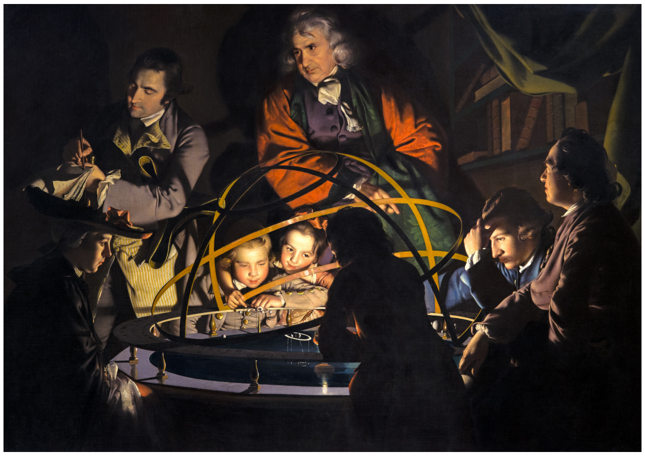
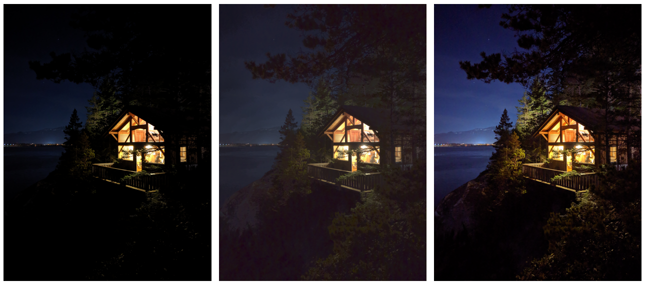


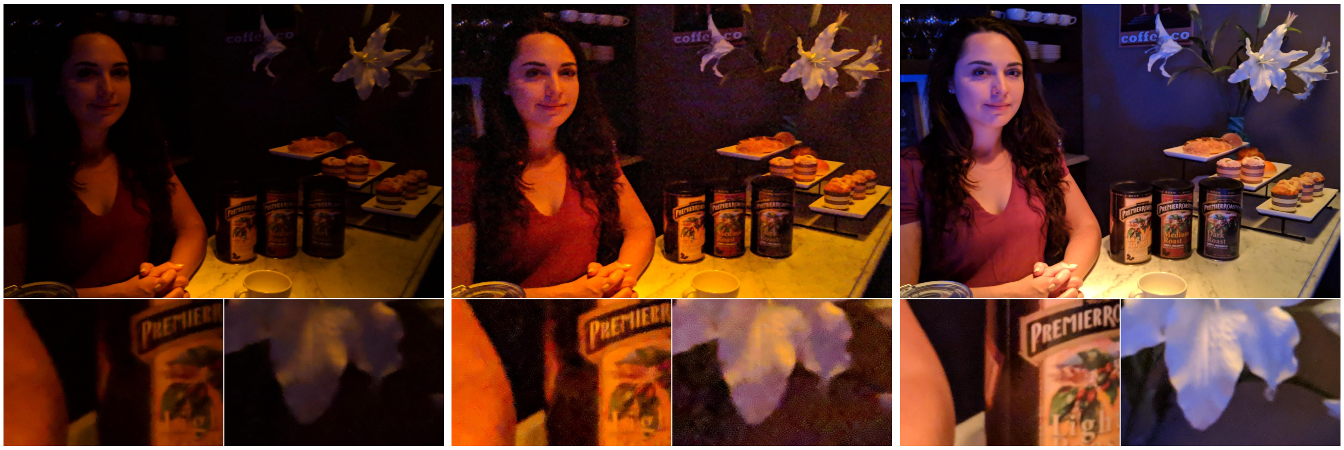
Figure 9: *Philosopher Lecturing on the Orrery*, by Joseph Wright of Derby, 1766 [7]. The artist depicts a dark scene with bright, colorful detail while still maintaining the nighttime aesthetic by increasing contrast, surrounding the scene in darkness, and keeping the shadow areas completely black. [13]



(a) Baseline (b) CLAHE (c) Liba et al.

Figure 10: An example of tone mapping a night scene. The tone mapping of Hasinoff et al. [3] (**a**) produces too dark of an image while using a different tone mapping technique (**b**) that brightens using histogram equalization (CLAHE [14]) results in more detail but lacking global contrast. The tone mapping from Liba et al. (**c**) retains detail while keeping global contrast and dark areas to look like a night scene. [6]

The existing error metric for color constancy looks at the error in appearance of a white patch in the image, but that idea doesn't work in heavily-tinted scenes with missing color channels so the improved error metric considers the appearance of an *average* portion of the image under the recovered illumination. It is able to be less sensitive to errors in channels with lower mean values in the true image.



(a) Hasinoff et al.

(b) Hasinoff et al. brightened

(c) Liba et al.

Figure 11: A comparison of images captured using the Hasinoff et al. pipeline (a and b) and the pipeline by Liba et al. showing the improvements in detail and noise from selecting a longer exposure time due to the low motion in the scene and taking more frames. It also shows the improvements in color reproduction from the improved white balance algorithm and tone mapping. [6]

4.3 Tone Mapping

Tone mapping is the process of mapping colors from a high-dynamic-range image to another in a medium with a more limited dynamic range [11]. This is usually done by applying what are called tone mapping operators (TMOs). Some TMOs attempt to create results close to human vision, while others produce a more artistic rendition. While human vision loses color sensitivity and spatial acuity as light levels are reduced, Liba et al. still wanted to render vibrant and colorful images in low-light rather than emulating human vision.

Simply brightening the whole image (Figure 10b) can result in low contrast and undesired saturated regions that make the image look flat. For centuries, artists have evoked a nighttime aesthetic through various methods such as the use of darker pigments, suppressed shadows, and increased contrast (Figure 9). Liba et al. developed a TMO, inspired by artistic techniques, that maintains vibrant color in dark scenes without looking artificial while still maintaining a nighttime aesthetic.

Their TMO uses various heuristics on top of the tone mapping of Hasinoff et al. [3]. These include allowing higher overall gains, limiting the boosting of shadows, allowing compression of higher dynamic ranges, boosting the color saturation inversely to scene brightness, and adding a vignette (darkening around the edges of the image). It produces a more detailed image while maintaining global contrast and the nighttime aesthetic (Figure 10c).

4.4 Results

This system by Liba et al. was launched in November 2018 on Pixel phones as the "Night Sight" mode in the camera app. On Pixel, Pixel 2, and Pixel 3a it uses a burst merging technique adapted from Hasinoff et al. [3] to better handle motion while on faster Pixel phones it uses the super-resolution merging algorithm described by Wronski et al. [12] (Section 3). The comparisons and results here all use the former merging technique.

Liba et al. compared their system to the pipeline it built upon from Hasinoff et al. [3]. Figure 11 shows an example of their system producing lower noise, more detail, and more pleasing colors. They also compared their pipeline with a neural network that operates on single raw images by Chen et al. [2] using frames from a similar camera to the one the neural network was trained on rather than a smartphone camera and still the system by Liba et al. produced higher-quality images and processed images much faster and using less memory.

More recently, an astrophotography mode has been added to Night Sight allowing for even longer exposures (as long as 1-4 minutes) which allows for sharp and clear images of stars and extremely dark landscapes [4].

5. CONCLUSIONS

Computational photography has allowed for massive advances in smartphone photography, allowing phone cameras to surpass what would be expected from their limited hardware. This has allowed many more people than ever before to take high quality photographs with ease. The software portion of a camera has become more influential to image quality than the physical hardware. These new computational photography techniques have allowed Google to continue using the same camera sensor hardware over multiple generations of their Pixel phones while continuing to be highly rated in photo quality. It has also enabled updates that continually improve the camera performance of previous Pixel phones to the point where they compete with the latest from other manufacturers. Computational photography techniques will continue to be ever more important in smartphones, professional photography, and other applications.

Acknowledgments

I would like to thank Nic McPhee, Elena Machkasova, and Kristin Shirley for their feedback and advice.

6. REFERENCES

[1] J. T. Barron and Y. Tsai. Fast fourier color constancy. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6950–6958, July 2017.

[2] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018.

[3] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph.*, 35(6), Nov. 2016.

[4] F. Kainz and K. Murthy. Astrophotography with Night Sight on Pixel phones, Nov 2019. `https://ai.googleblog.com/2019/11/astrophotography-with-night-sight-on.html`.

[5] M. Levoy and Y. Pritch. Night Sight: Seeing in the dark on Pixel phones, Nov 2018. `https://ai.googleblog.com/2018/11/night-sight-seeing-in-dark-on-pixel.html`.

[6] O. Liba, K. Murthy, Y.-T. Tsai, T. Brooks, T. Xue, N. Karnad, Q. He, J. T. Barron, D. Sharlet, R. Geiss, S. W. Hasinoff, Y. Pritch, and M. Levoy. Handheld mobile photography in very low light. *ACM Trans. Graph.*, 38(6), Nov. 2019.

[7] J. W. of Derby. A philosopher giving that lecture on the orrery, in which a lamp is put in place of the sun. `https://commons.wikimedia.org/w/index.php?title=File:Wright_of_Derby,_The_Orrery.jpg&oldid=480373143`, 1766. [Online; accessed 24-October-2020].

[8] Wikipedia contributors. Bayer filter — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Bayer_filter&oldid=977384299`, 2020. [Online; accessed 10-September-2020].

[9] Wikipedia contributors. Demosaicing — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Demosaicing&oldid=935211330`, 2020. [Online; accessed 10-September-2020].

[10] Wikipedia contributors. Kernel (image processing) — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Kernel_(image_processing)&oldid=984669725`, 2020. [Online; accessed 9-November-2020].

[11] Wikipedia contributors. Tone mapping — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Tone_mapping&oldid=978774859`, 2020. [Online; accessed 8-October-2020].

[12] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar. Handheld multi-frame super-resolution. *ACM Trans. Graph.*, 38(4), July 2019.

[13] B. Wronski and P. Milanfar. See better and further with Super Res Zoom on the Pixel 3, Oct 2018. `https://ai.googleblog.com/2018/10/see-better-and-further-with-super-res.html`.

[14] K. Zuiderveld. *Contrast Limited Adaptive Histogram Equalization*, page 474–485. Academic Press Professional, Inc., USA, 1994.