Camera support for use of unchipped manual lenses

Henry Dietz; University of Kentucky; Lexington, Kentucky

Abstract

As interchangeable-lens cameras evolve into tightly integrated electromechanical systems, it is becoming increasingly awkward to use optics that cannot electronically communicate with the camera body. Such lenses are commonly referred to as "unchipped" because they lack integrated circuitry (aka, chips) that could interface with the camera body. Despite the awkwardness, there is a large community of photographers who prefer to use manual lenses. Not only is there an increased demand for vintage lenses, but the variety of newly-manufactured fully-manual lenses has been growing dramatically in recent years.

Although manual lenses will never provide all the features and performance of lenses created as integrated parts of a camera system, the current work explores a variety of methods by which digital cameras can significantly improve the usability of unchipped manual lenses.

Introduction

Cameras have become increasingly automatic. Current models often include automation of exposure and focus, tuning of parameters based on scene recognition, intelligent cropping to recompose your shot, and AI-based analysis to replace portions of the image with computationally-synthesized improvements (e.g., synthetic bokeh). Although these features help amateurs *take* better photos, they disconnect photographers from the level of creative control they formerly enjoyed in *making* images.

Manual lenses put photographers back in full control. Unlike lenses that contain electronics, they are not limited to working with a particular mount or camera brand; they are easily adapted to other cameras – especially mirrorless ones, thanks to the short flange distance allowing infinity focus despite adding the thickness of an adapter. The lack of electronic interfaces and drive motors in manual lenses also can make them cheaper, smaller, and mechanically more responsive and precise. Better still, old manual lenses are plentiful and often priced way below their value as optics because they were abandoned years ago as features like autofocus became available. In fact, the lenses shown in Figure 1 cost an average of less than $25 each. It's not surprising that there is now a dramatic surge in the popularity of manual lenses.

Camera makers have noticed this trend, and have taken some steps to facilitate use of manual lenses. For example, many cameras now support focus peaking and magnified live view as manual-focus aids. However, using manual lenses remains more awkward than it logically should be because some features that would be highly desirable when using a manual lens are strangely disabled. The problem is simply that some of the potentially most useful features in support of manual lens use depend on the camera knowing some properties of the lens – and they don't.

This paper suggests, and presents a preliminary evaluation of, a variety of methods that could be used to identify the relevant characteristics of a manual lens.

Figure 1. *Some of the author's collection of (mostly manual) lenses*

Recognizing A Particular Lens

By default, many cameras will not allow the shutter to fire when an unchipped lens is mounted. This behavior may have been seen as preventing faulty exposures from being made when an electronically-enabled lens has not been attached correctly or electrical contacts are dirty. In such cameras, it is common that one needs to enable an option buried in the menus that is often very misleadingly called "shoot without lens." Obviously, it would be highly preferable for the camera to know something about the unchipped lens that is attached, and there are two main reasons:

- Cameras that support in-body image stabilization (IBIS) need to know the approximate focal length in order to correctly move the sensor to compensate for camera shake. At closer focus, IBIS also needs to know the focus distance.
- It is highly desirable that the lens information be included in the metadata of each captured image – the EXIF data. For example, many lenses suffer distortion, vignetting, and lateral chromatic aberrations that image editing software will automatically correct when it recognizes the lens by the EXIF data. In-camera correction would be nice too.

Many cameras allow the user to manually enter lens data that can be used for either or both of the above purposes. Some cameras permit the user to enter a list of unchipped lenses and then run through a menu to select which of those has been mounted. For example, all Sony E-mount bodies that have IBIS provide a menu for selecting the lens focal length to apply for IBIS, but that focal length is not recorded in image EXIF data. However, Sony offers a $9.99 "Lens Compensation" camera app[1] for many (but not the latest) of their E-mount bodies. That app allows entry of a list of lens descriptions containing lens model, focal length, and aperture (f/number), along with correction parameters for periph-

Figure 2. *Three chip alternatives: NFC, chipped adapter, and LM-EA7*

Figure 3. *3D-printed QR code lenscaps.*

eral shading, chromatic aberration, and distortion. When a lens is selected from the app's list, that data is used for IBIS, saved in image EXIF, and even applied to correct both JPEG images created in-camera and the live view display. The problem is that manual entry of lens data, and even selection from a menu list, is awkward and error prone.

There are three ways that we suggest unchipped lenses could be recognized by the camera body.

Adding A Chip

Perhaps the most obvious way to make an unchipped lens be recognized is to literally add a chip to it. Such a chip may be added to the lens itself, but is more easily integrated in the design of a mount adapter.

Unfortunately, none of the lens communication protocols in common use by Canon, Sony, Nikon, etc. are public – however, all have been reverse engineered to some degree, and using the existing protocol does not require changing the camera firmware. Perhaps the best known of these chips is the "Dandelion" for Canon EF/EF-S bodies[2], but similar chips exist for other mounts, such as the M42 lens to Sony A-mount adapter in the middle of Figure 2. These simple chips are programmed with the identity of a particular lens, and some may be reprogrammed via a convoluted procedure from the camera body. The TechArt Pro LM-EA7[3], seen on the right in Figure 2, takes this concept two steps further: the chip can be programmed via Bluetooth wireless with a set of lens specifications, and the adapter not only reports that info to the camera, but also decodes focus commands from the body and incorporates a motor to move the entire lens. Thus, rather remarkably, the LM-EA7 implements a limited-range autofocus, or auto-tweaking of manual focus, for manual lenses.

A third possibility is provided by the fact that, as seen to the left in Figure 2, many cameras now provide NFC (near field communication) support – a type of RFID (radio frequency identification). Although it would require some reprogramming of the camera, it would be cheap and easy to stick a passive NFC RFID tag on each lens that would be programmed with the full lens description. In that way, simply tapping the camera's "N" symbol with the tagged lens could be used to notify the camera of the lens about to be mounted, and the same NFC RFID tag could be used with a variety of cameras independent of lens mount.

QR Code In A Lenscap

Instead of using an electronic interface to the lens data, it is possible to use an entirely optical method without requiring any modification to the camera's firmware. QR code[4] is a fault-tolerant two-dimensional matrix encoding of data. Although that data most often is a reference to a URL (universal resource locator, WWW address), it can be any data: including the identification data for a lens. If the photographer captures a photo of such a QR code each time a new lens is mounted, software scanning the images can identify and read the QR code, thus easily marking subsequent images with the appropriate lens data as EXIF.

The solution we developed in 2016, but did not publish in a scholarly venue until now, is to 3D print custom lenscaps that incorporate a QR code. The front of the cap can be labeled with human-readable text identifying the lens while the inside of the cap contains the machine-readable QR code. Figure 3 shows three different styles of QR code lenscaps 3D printed from designs created by the `OpenSCAD Customizer` program which we wrote and have made freely available[5]. The left cap is printed in clear and black layers to give higher contrast, while the center one is printed using only white filament and must be strongly backlit to photograph the QR code. The cap on the right is a plain cap with an inkjet-printed paper QR code insert; it is less durable, but can hold more data and gives good contrast with reflected light.

OOF PSF Matching

Probably the most powerful method for identifying a lens is the somewhat computationally intensive matching of what we call the out-of-focus point spread function (OOF PSF)[6]. An OOF PSF is simply the image created by a point light source photographed significantly out of focus. Measuring an OOF PSF is ideally done in a darkened area using a single point light source, but OOF PSF naturally occur in any image that has significantly defocused areas that contain light sources. The secondary issue is that in order to see identifying structures within the OOF PSF, the tones inside the OOF PSF must not be clipped by overexposure. However, for most lenses, the precise shape of the aperture changes as the lens is stopped down, so the precise shape of the OOF PSF also changes, and it should be possible not only to identify the specific lens, but at least in some cases also the current aperture setting. The OOF PSF sizes might even allow determination of approximate focus distance.

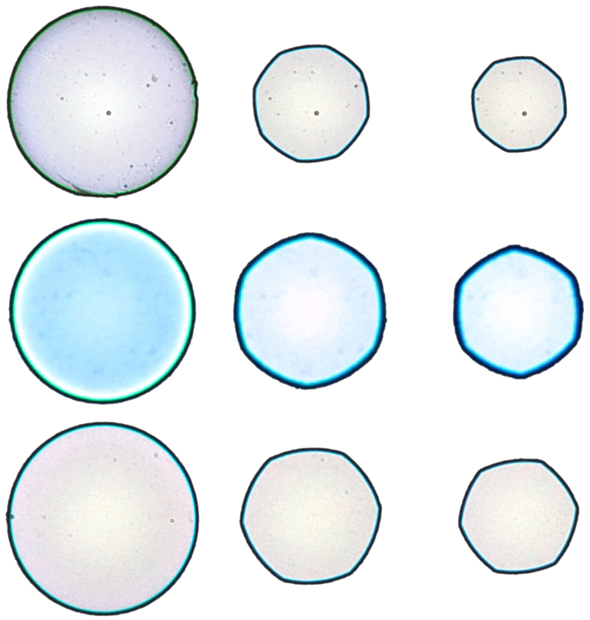

Figure 4. *OOF PSF: 58mm f/1.2, 58mm f/1.4, 50mm f/1.4.*

As was shown in earlier work[6], each individual lens has a variety of distinctive characteristics visible in the OOF PSF. For example, Figure 4 shows OOF PSF for three extremely similar Minolta Rokkor lenses. From top to bottom: a 58mm f/1.2, a 58mm f/1.4, and a 50mm f/1.4. Each row shows a sequence of OOF PSFs wide open, stopped down one click, and stopped down two clicks. The black specs are diffraction patterns from dust in the lenses, and are stable identifiers for that particular copy of a lens. In addition, as the aperture is closed, the aperture shape changes from circular to an octagon for the 58mm f/1.2, but to hexagons for the other two lenses, and with different blade angles between the 58mm f/1.4 and 50mm f/1.4. Of course, more different lenses tend to have much more different OOF PSF, but it is common that users of old manual lenses will have multiple "fast fifty" lenses, so distinguishing differences as small as these has practical value. Figure 5 proves this is possible.

To implement this matching, a camera would need to be reprogrammed not only to do this matching analysis, but also to construct and maintain a database of the user's lenses including identifying features for the OOF PSFs. The matching of OOF PSF patterns is straightforward in isolation, but in the context of a more complex image, is not much simpler than recognizing a face – and a similar neural network approach could be used. However, even a simple Hough transform[7] can be moderately effective. Easily distinguished OOF PSF will not occur in every composition of a scene, so the best tact would be to continuously scan the live view stream for a match with a known lens OOF PSF. The problem is somewhat simplified by the fact that the algorithm does not need to identify the lens, but merely to determine which of the OOF PSF registered with the camera body is the closest match. Even matching that merely reduces the length of the list of candidate lenses could be useful.

In some images, it also is possible to automatically detect vignetting, distortion, and chromatic aberrations, so incorporating these lens features with the OOF PSF matching may simplify, speed-up, and improve the robustness of the processing. Again, the primary issue is the need to construct and maintain a database of lens attributes.

Figure 5. *Photo shot by 50mm f/1.4 and crop showing OOF PSF proof.*

Estimating Focal Length

Image motion blur can be dramatically reduced by IBIS acting to counter camera shake. As the whole camera is moved during an exposure, IBIS senses the motion and dynamically repositions the sensor so that the same portion of the scene is still projected onto the same place on the sensor. Fundamentally, the camera cannot know how much to move the sensor in response to whole-camera motion without knowing the lens focal length. Thus, even if the particular lens being used cannot be identified, knowing the focal length would be very helpful.

There are a variety of methods by which a camera could estimate the focal length of an unknown lens. The key is to work the problem backward: measure the motion blur caused by camera shake in an image and derive the focal length from that. Alternatively, object movement tracked through a time sequence of multiple images can be used.

As we have empirically measured using ShAKY[8] (open-source hardware we created to facilitate measuring camera shake), camera motion due to shake during an exposure can be quite complex, with components in all six degrees of freedom: X, Y, Z, Roll, Pitch, and Yaw. By convention, these are oriented as shown in Figure 6. The Z axis is normally the the axis running outward

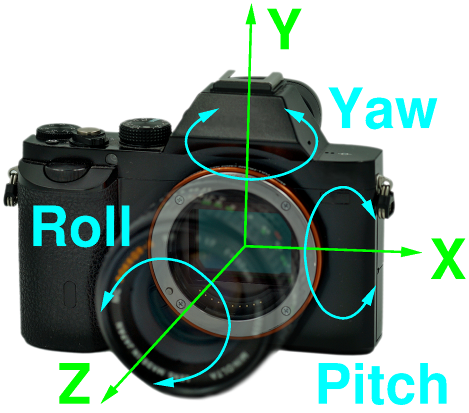

Figure 6. *Orientations of X, Y, Z, Roll, Pitch, and Yaw*

along the optical center of the camera's lens. The X and Y are thus respectively the long and short dimensions of the sensor. Roll is rotation about the Z axis, Pitch is rotation about X, and Yaw is rotation about Y. Motions in these six degrees of freedom can be directly measured or tracked by various means.

At typical, non-macro, shooting distances, the impact of small X, Y, and Z movements generally is not significant. An X or Y displacement of the whole camera system has the effect of shifting the scene region being captured by approximately the same amount; thus, the impact of a movement is essentially multiplied by the magnification factor of the optics. Shooting at a fairly typical 1:20 magnification, that would mean getting a single-pixel shift of the image would require moving the camera by about 20 times the size of a pixel. The image blur caused by motion in Z is even less significant, slightly enlarging or shrinking the size of the image of the scene.

CIPA, the Camera & Imaging Products Association, established a standard for testing the effectiveness of image stabilization systems in cameras[9]. That standard confirms the claim that X, Y, and Z motion are "practically negligible" at a subject distance of 20 or more times the lens focal length, but also claims Roll is not significant. Rotation about the optical axis of the lens is clearly important off axis, but has no impact at all on the optical center. Because image quality for many lenses falls off near the corners of the frame, it is reasonable to prefer analysis of the central image to guess focal length, and the primary motion components of interest are indeed Pitch and Yaw.

Angular Correspondence

The relationship between focal length of a rectilinear lens and the angle of view it affords is straightforward. Given the image coverage diameter (or diagonal of a rectangular sensor) and focal length of the lens, the view angle is:

$$viewangle = 2 \times atan(imagedia/(2 \times focallength)) \qquad (1)$$

This same formula can be used to determine the angular slice occupied by a pixel. For example, a full-frame 24MP camera would have a $36 \times 24mm$ sensor containing an array of 6000×4000 sensels. Thus, each pixel site is $36/6000 \times 24/4000mm$, or a square 6 microns on each side. The Yaw motion component is essentially the angle in the 36mm dimension of the sensor while the Pitch is in the 24mm dimension. The amount of Yaw or Pitch motion corresponding to one pixel is computed by using 0.006mm as the $imagedia$. This implies that $focallength$ can be computed from either of the following two formulas:

$$focallength = 1/(2 \times tan(Pitch/2)/horizontalshift) \qquad (2)$$

$$focallength = 1/(2 \times tan(Yaw/2)/verticalshift) \qquad (3)$$

in which $horizontalshift$ and $verticalshift$ are the measured position shifts of a tracked image feature as projected on the sensor (for a feature near the center). Although it is possible to obtain these shifts to accuracies that are fractions of the size of a pixel, suppose that tracking is performed at the level of nearest pixels. With a 50mm lens on the 24MP camera described above, a single pixel shift corresponds to a 0.007° view angle. That is comparable to the rotational resolution limit for a typical low-cost sensor, such as ShAKY's[8] MPU-9250[10] 9-axis accelerometer / gyroscope / magnetometer, so there is no need to track motion to sub-pixels; measurements of motion over multiple pixels suffice.

Focal Length From Images Alone?

It is clear that focal length can be estimated by combining direct measurement of Pitch and Yaw with motion detected either as blur within a single frame or by tracking across multiple frames (e.g., while processing the live view stream), but is it possible to estimate focal length *without* use of a gyroscope? The answer depends on how consistently the level of object motion correlates with focal length – a property for which measurements do not appear to have been published.

A simple test was conducted in which a short 60FPS 4K video clip was captured hand-held using a variety of manual lenses on a Sony A7RII with in-body image stabilization disabled. This video is a good approximation to the quality of the live view stream, and uses the full 36mm width of the sensor, yielding an effective pixel size of 9.375 microns. Although a longer frame sequence would give more accurate results, just 8 frames were sampled, based on the idea that the scene feature tracking algorithms used in modern cameras (primarily intended for autofocus) can certainly be relied upon to consistently track the same object for at least that many consecutive frames. That tracking was approximated using `Hugin Panorama Creator` to align a single, central, feature.

Five manual lenses typical of their focal lengths were used: Minolta Rokkors of 16mm, 50mm, 100mm, and 200mm focal lengths and Vivitar 24mm and 2X converter (used with the 200mm to make a 400mm). All were hand-held the same way, using a two-handed grip and the electronic viewfinder. The first image sampled from the video for each lens is shown in Figure 7. The results are summarized in the following table:

Lens used	**X walk**	**Y walk**	**Angle**
16mm Fisheye	2.65	2.05	0.079°
24mm	2.05	4.25	0.070°
50mm	4.71	2.12	0.037°
100mm Macro	37.2	8.73	0.123°
200mm	22.1	57.5	0.107°
"" + 2X (400mm)	159	104	0.176°

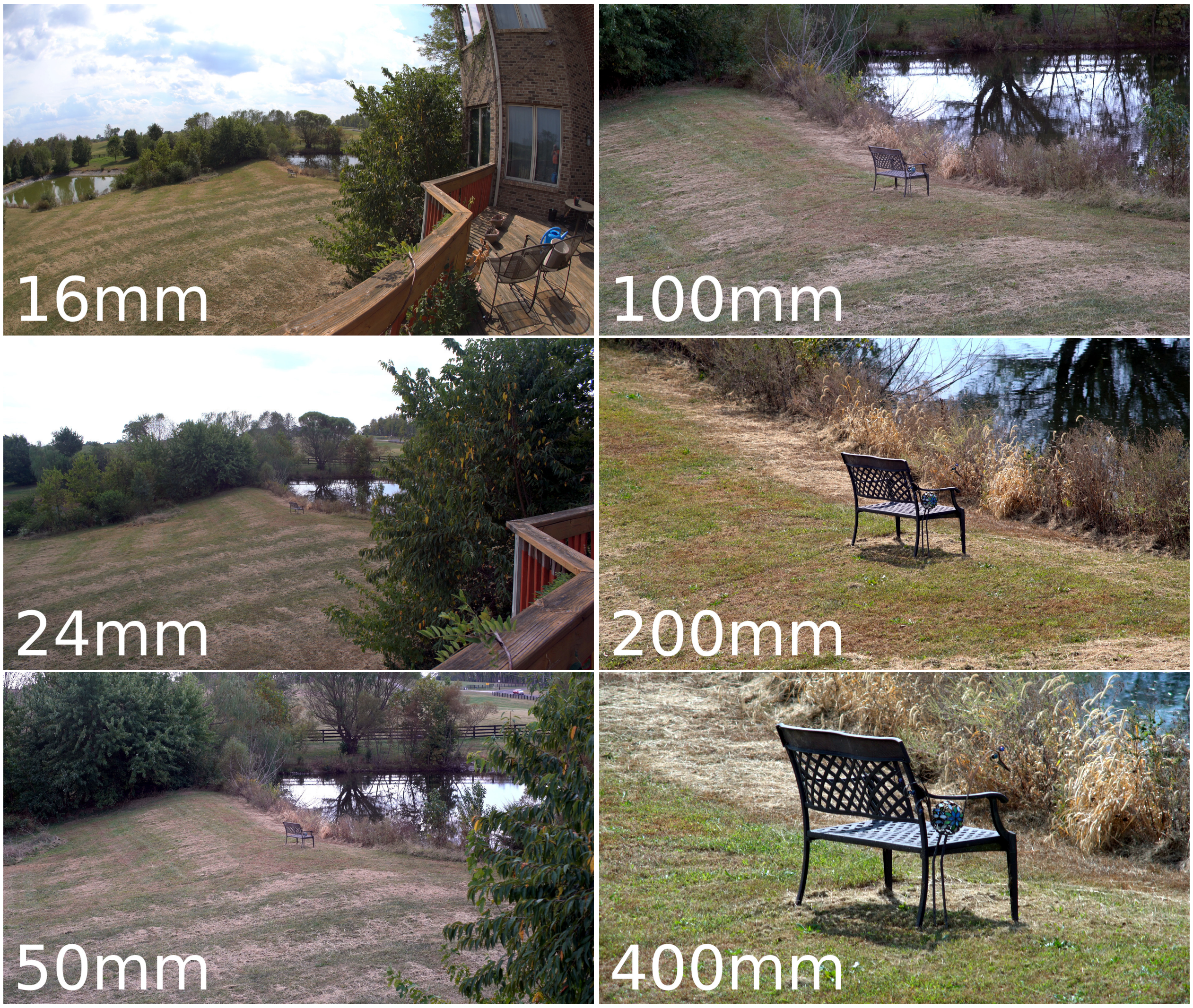

Figure 7. *Sample images at 16, 24, 50, 100, 200, and 400mm; still-image motion blur is insufficient to estimate focal length except perhaps at 400mm*

The X and Y walk entries are the bounding-box sizes, in pixels, for the movement of the selected feature (the nearest corner of the iron bench). The Angle is the average angular deflection during capture of those 8 frames. Although the data is very noisy, this does reveal several properties:

- Although the bounding box sizes for shorter focal length lenses are tightly clumped, there is a very strong correlation between walk size and telephoto lens focal length.
- The angular deflections due to shake are not very good predictors of focal length. In fact, it is difficult to see any correlation. Most likely, the angles are more a function of the ergonomics of the particular lens than of the focal length. The 50mm is the smallest lens and is arguably better balanced on the camera body than the others; the 200mm lens is quite heavy, and that mass probably explains why the shake of the light 100mm went through a larger angle.
- Although there is some motion blur in the 400mm images, the other images did not have sufficient blur to estimate focal length from a single shot. However, even the motion blur in the 400mm images did not disturb the tracking alignment.

In sum, these results suggest that it might be possible to reasonably estimate the focal length simply by measuring walk size for short sequences – without using an accelerometer nor gyroscope. The correlation of walk dimensions with focal length is particularly good for telephoto lenses, and average focal length estimation error of less than 25% should be possible. Much higher accuracy is possible by comparing walk size to gyroscope readings, but would sampling motion over more than 8 frames make comparison to gyroscope readings unnecessary?

A Live View Implementation

The ideal way to implement in-camera guessing of focal length is to implement it by intercepting the >20FPS live view stream – which requires some custom code to run in the camera. Not many cameras that are suitable for use with manual lenses are user programmable, and the ideal would be a camera body with

```
Hit any key to try a random focal length
Sampling 416mm
Sampling for 3 seconds
Recorded 37 for 416mm
Guessed 475mm for 416mm
Hit any key to try a random focal length
Guess My Focal Le<EXIT=menu>
```

Figure 8. *Canon PowerShot SX530 guessing focal length*

IBIS. Unfortunately, although Sony cameras such as the A7RII seem excellent targets thanks to supporting both IBIS and programmability via the OpenMemories hack of PlayMemories, at this writing, the programming interface provides neither access to the IBIS motion sensors nor access to the live view. Thus, the implementation in the current work instead uses CHDK[11] running on a Canon PowerShot SX530, as shown in Figure 8.

The SX530 is a fixed-lens 50× super-zoom compact camera with the equivalent of a 24-1200mm zoom range on a full-frame camera. Although the SX530 supports OIS and presumably contains a motion sensor, CHDK does not provide an interface to that – and OIS was disabled for our testing. The calibration and guessing of focal lengths was implemented entirely by a Lua script using CHDK's `md_motion_detect()` function to monitor the live view. Rather than measuring length of the random walk caused by shake, the Lua script measures the number of times a motion detection threshold is surpassed over a user-settable period of one to three seconds. Calibration normally consists of measuring motion detection rate for each of eight log-spaced focal lengths when the camera is hand-held and focused on a stationary scene, although a default calibration table can be used instead.

To compute an estimate of an unknown focal length f_u, the motion detection rate m_u is measured. For each m_i in the calibration table, a difference weight d_i is computed; using those, the focal length estimate is the weighted average:

$$d_i = 1/((m_u \times m_i)^2 + 1) \tag{4}$$

$$f_u = (\sum_i (f_i \times d_i))/\sum_i d_i \tag{5}$$

As implemented in CHDK Lua, the fractions above are scaled to avoid use of floating-point arithmetic.

The current implementation is the CHDK Lua script `gmfl.lua`: "Guess My Focal Length," freely available at `Aggregate.Org/DIT/GMFL`. It is not particularly specific to the SX530 model, and should run in most CHDK-supported cameras after re-calibration. Empirically, error when sampling for at least two seconds is generally less than 10%, but can be larger for very short or very long focal lengths, or if the default calibration is used for some individuals. Guess errors large enough to significantly impact IBIS performance happened only when the scene moved during measurement, which would be obvious to the user.

Conclusion

This paper has presented a variety of mechanisms that could be used to provide a camera with more detailed information about the unchipped manual lens mounted on it. Using this information, the most desirable modern features should should be available despite using a fully manual lens: recording of at least some lens EXIF data, setting of IBIS focal length, etc.

Although all the methods discussed are feasible, the use of shake measurements to automatically set focal length for IBIS is especially compelling. The guessing procedure could be automatically applied under various circumstances, perhaps including when the camera is first turned on with no chipped lens detected. However, the ability to program a button for automatically guessing and setting IBIS focal length would be even more useful – especially for unchipped zoom lenses. Even without access to the IBIS sensors, guessing focal length takes no more than a few seconds of live view and could be repeated if the user sees scene movement during measurement. With access to the IBIS sensors, guessing should be much faster as well as very precise and accurate.

References

[1] Sony PlayMemories Camera Apps: Lens Compensation, https://www.playmemoriescameraapps.com/ (accessed September 29, 2019)

[2] Generation 8/9 Focus Confirmation Chip Instructions, https://fotodiox.freshdesk.com/support/solutions/articles/17000003286-generation-8-9-focus-confirmation-chip-instructions (accessed September 29, 2019)

[3] Techart PRO Leica M Sony E Autofocus Adapter, http://techartpro.com/product/techart-pro-leica-m-sony-e-autofocus-adapter/ (accessed September 29, 2019)

[4] ISO/IEC 18004:2015, Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification, https://www.iso.org/standard/62021.html (February 2015)

[5] Henry Dietz, 3D-Printed Screw-In Front Lenscaps With Optional QR Code, https://www.thingiverse.com/thing:1563812 (accessed September 29, 2019)

[6] Henry Dietz, "Out-of-focus point spread functions", Proc. SPIE 9023, Digital Photography X, 90230J, https://doi.org/10.1117/12.2040490 (March 7, 2014)

[7] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Pattern Recognition, Vol.13, No.2, p.111-122 (1981)

[8] Henry Dietz, William Davis, and Paul Eberhart, "Characterization of Camera Shake," to appear in Imaging Sensors and Systems, Electronic Imaging (2020)

[9] Standard of Camera & Imaging Products Association, CIPA DC-X011-Translation-2014, Measurement and Description Method for Image Stabilization Performance of Digital Cameras (Optical System), http://www.cipa.jp/image-stabilization/documents_e/DC-X011-2014_E.pdf (2014)

[10] MPU-9250 Product Specification Revision 1.1, http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf (accessed September 24, 2019)

[11] Canon Hack Development Kit (CHDK), http://chdk.wikia.com/wiki/CHDK (accessed February 9, 2020)