# An Ultra-Low-Cost Large-Format Wireless IoT Camera

***Henry Dietz and Paul Eberhart; University of Kentucky; Lexington, Kentucky***

## Abstract

*This paper documents the design, construction, and experimental evaluation of an ultra-low-cost large-format digital camera. Used lenses that cover formats up to 4x5 can be surprisingly inexpensive, but large-format image sensors are not. By combining 3D printing with cheap components developed for use in IoT (Internet of Things) devices, especially the sub-$10 ESP32-CAM, a digital scanning large-format camera capable of over 2GP resolution can be constructed at very low cost. Despite the large image area, Lafodis160 is literally a wireless IoT device, fully remote controllable via Bluetooth and WiFi. This camera was originally intended to serve as a testbed for novel ways to improve capture quality for scenes that are not completely static during the scan interval, and a brief overview is given of methods employing unusual scan orderings that will be evaluated using it.*

## Introduction

This work was seeded nearly two decades ago, while exploring use of a computer-controlled telescope mount to move a still camera to stitch gigapixel images – a method now commonly implemented with purpose-built pan/tilt drive systems[1]. Even in our earliest experiments, it was immediately apparent that temporal differences between adjacent scan lines in the usual raster were causing artifacts throughout the stitched images, and so we began to use a scan order based on the Hilbert curve[2] to significantly reduce the average temporal distance between physically adjacent samples. Later work in time domain continuous imaging (TDCI)[3] provided a significantly more sophisticated set of tools for modeling how scene content changes over time, and that suggests ways to intelligently dynamically order the scanning of a non-static scene. Just last year, Senscape[4] added concepts for incorporating a confidence metric into the image fusing process. Thus, the goal of the research reported in this paper is to create a platform for experiments involving intelligent, dynamic, scan ordering.

However, the system described here does not tilt and rotate a camera and lens system to scan a scene. Instead, it uses the significantly more awkward approach of scanning the image projected by a fixed large-format lens. There are at least four key motives.

The first motivation is that large format cameras offer a different look from that obtained by stitching captures from smaller-format lenses. Not all image quality metrics are higher for the large-coverage-circle lenses used in large-format cameras, but the images produced are commonly described as being much more "organic," with shallow depth of field and a smooth and subtle increase in the prominence of optical defects further off-axis. This is prime motivation for many users of large-format cameras, and for the low resolution large-format sensor backs being produced by LargeSense[5]. Using the lens to shift or tilt the plane of focus (the Scheimpflug principle) also produces results that are extremely difficult to simulate using pan/tilt stitching.

***Figure 1.*** *Lafodis160 large-format wireless IoT camera*

Sufficiently high lens resolution at quite modest cost is the second reason to consider this approach. Despite various optical defects far off axis, large-format lenses are usually capable of comparable line-pairs-per-mm (lp/mm) resolution over most of their field to good lenses for smaller format cameras[6] – and that's even true of many enlarger lenses which sell used for under $30. In part, this performance may be due to large format lenses generally being used at higher $f$/numbers than are commonly used for 135-format ("full frame") and smaller sensors – although stopping down too far makes resolution diffraction limited, most lenses reach optimum resolution between $f/5.6$ and $f/16$. 50 lp/mm suggests a pixel size of approximately 5 microns, which would result in about 500MP for a 4x5 format camera. The best lenses peak around 85 lp/mm, yielding just shy of 1.5GP for 4x5 format. Nyquist samplings would halve pixel size in each dimension, thus multiplying the desired capture resolution by four so that even 2.5 micron pixels are easily justified. Put simply, full frame is 36x24mm while 4x5 is about 125x100mm and the camera developed here (Figure 1) captures a 160mm diameter circle; total resolution grows in proportion to image area, respectively reaching 14X or 23X full frame.

The third motive is that when stitching samples taken from a single projected image, there is no distortion of pixels required for alignment. In contrast, stitching images captured by pan and tilt of the entire camera requires precise correction of lens distortion and will suffer minor uncorrectable parallax errors if the pivot is not

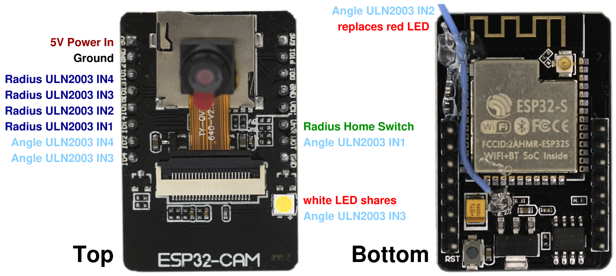

***Figure 2.*** *AI-Thinker ESP32-CAM pin assignments*

precisely centered in the sensor. This is why large format is popular for producing very high resolution images of anything from landscapes to photographs of artwork used for digital preservation. In such applications, resolution trumps capture speed; commercial scanning backs from BetterLight[7] and Rencay[8] are well known, but there also are many manual scanning backs starting around $200 that allow sliding a smaller-format camera within the image area of a large-format camera[9]. There have even been various attempts to re-purpose flatbed scanners as large-format scanning backs[10], although that generally results in very poor image quality. A common problem for digital backs is that the sensor mounts behind the normal focal plane, often resulting in vignetting or loss of infinity focus with shorter-focal-length lenses... building a complete camera avoids such issues.

Finally, the fact that the camera exterior does not move during an exposure offers a variety of benefits. Minimizing moving mass helps stability and power consumption. It also is feasible to make such a camera a weatherproofed stand-alone device, which fits nicely with the idea of literally making this an IoT (internet of things) camera controlled entirely via wireless networking. The fact that scan captures take minutes to hours makes it extremely desirable that the camera can be controlled from a remote location, which also opens the possibility of using this as a security camera that, paired with a more conventional camera, could produce high-resolution data for any areas of interest identified by the conventional camera.

The remainder of this paper focuses on how a large-format wireless IoT camera was built. More precisely, this discussion will skip discussion of the first prototype, which was an X, Y scanner: **Lafodis 4x5 – LArge FOrmat DIgital Scanning** for 4x5 format. Instead, focus will be on the second prototype, **Lafodis 160**, a more versatile and more interesting scanning camera using angle, radius positioning to capture up to a 160mm diameter image cicle. The basic specifications are:

- Scan resolution: typical 500MP @ 4x5", maximum 2.6GP
- Dynamic range: 8-10EV, HDR limit ∼20EV
- Color: RGB CFA, no integrated NIR filter
- Scan speed: currently <1MP/s, maximum ∼10MP/s
- Dimensions: ∼171mm diameter, ∼190mm deep
- Capture control: wireless host via BlueTooth (or 802.11)
- Firmware update: wireless, Arduino OTA compatible
- Power: 5V via USB connector from external source

## Lafodis160 Electronics

There are several recent technological advances that enable construction of Lafodis160, but the single most important is shown in Figure 2: the AI-Thinker ESP32-CAM. This sub-$10 boardlet is intended for use in IoT devices that would use a camera and simple artificial intelligence algorithms (e.g., face ID) to respond to what it sees.

***Figure 3.*** *Sample B&W 1600x1200 capture*

The camera generally packaged with this boardlet is the 2MP Omnivision OV2640[11], seen above the metal microSD card holder in Figure 2. This sub-$3 camera captures images in a variety of resolutions and formats, including raw encodings and some ROI windowing support, up to 1632x1220 pixels. The sensor pixels are 2.2 micron square, with a 10-bit ADC delivering surprisingly low noise if gain is kept low. The camera comes with a lens, which can be seen under the protective red tab; however, the lens is not electrically coupled, and can be removed by simply unscrewing it from its plastic mount over the sensor. Removing the lens exposes the bare sensor, which incorporates a conventional RGB Bayer CFA for color capture, but no NIR-blocking filter. An example of the image quality of a single capture is the B&W image shown in Figure 3, which was shot using the OV2640 with a 4x5 enlarger lens.

The AI-Thinker ESP32-CAM is not just a camera controller, but a complete System on a Chip (SoC). The Arduino-compatible 32-bit 160MHz dual-core processor is packaged with 4.5MB SRAM. If it is configured for 1.9MB APP with OTA and 190K SPIFFS, it can use the built-in WiFi for OTA (over-the-air) firmware updates. Both BlueTooth and 802.11 b/g/n WiFi are built-in with a usable antenna on the back side of the board. There are many other features too, including the microSD card

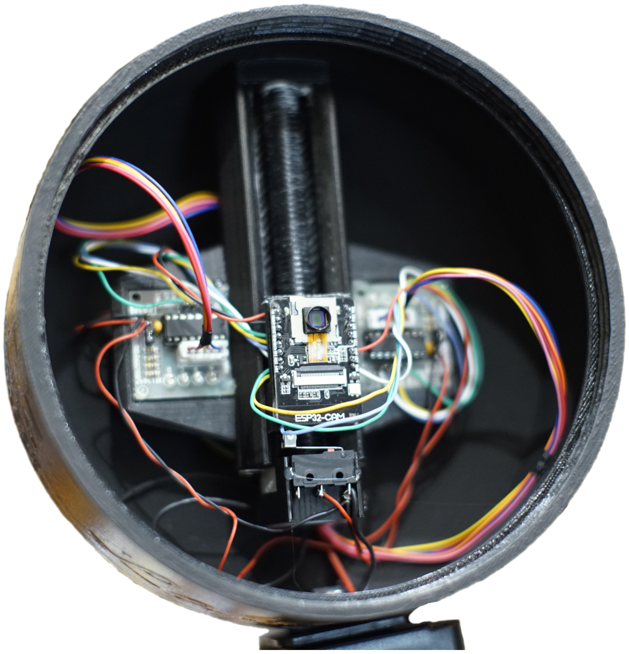

*Figure 4. Inside Lafodis160*

*Figure 5. Lafodis160 LED power light and USB power cord*

interface mentioned earlier and a white LED that is intended to serve as light source for the camera. In sum, this is a sufficiently powerful computer not only to manage the camera and wireless interactions, but also to control the steppers used for scanning.

Figure 4 shows the inside of Lafodis160. On either side of the ESP32-CAM boardlet, there are two ULN2003 driver boards used to drive the two 28BYJ-48 steppers[12]. Each of these steppers requires the appropriate sequencing of 4 control signals, so a total of 8 output pins are needed for the ESP32-CAM to implement the control. Unfortunately, the boardlet has many functions overloaded on each pin, so it takes some cleverness to make 8 output pins available. Figure 2 shows that seven of the pins (shown in blue and light blue) can be used for stepper control despite being shared with functions including the microSD card controller and the white LED. For the eighth output, it was necessary to desolder the red LED on the back side of the boardlet and solder a wire to be driven by that output. Only one more pin is needed to control the system: an input used to detect a simple home-position microswitch for the radius axis.

Because the ESP32-CAM communicates wirelessly, even for firmware updates, there is no need for an external wired connection. However, the stepper motors are a significant draw when active (which is why the ULN2003 drivers are needed). Thus, as seen in Figure 5, there is a USB connector on the back of Lafodis160 for an external power supply. The external supply can be any 5V supply that can power the camera for as long as a scan takes, which easily can be accomplished using a small USB powerbank.

## Mechanicals

With the exception of four screws for mounting the two steppers and two 1/4-20 nuts for tripod mounting, all mechanical parts of Lafodis160, the designs for which are shown in Figure 6, are 3D printed. Perhaps most surprising is the fact that no part is too large to fit on the bed of a typical low-cost extrusion-based printer – and the parts were printed on a $180 AnyCubic Linear Plus. All parts are printable without supports. Total PLA filament used was under 1kg and material cost was less than $15. The parts are:

- Main body, with angle stepper mount, USB connector, and mount for power light; 1/4-20 tripod mounts on bottom and top (for viewfinder or other attachments)
- Threaded lens extension
- Threaded lens focus tube
- Radius linear rail and mounts for ULN2003 boards
- Linear rail drive screw
- Linear rail shuttle and ESP32-CAM mount
- Slip-on microswitch mount for radius home stop
- Threaded lens mount plate

The linear rail shuttle is perhaps the most interesting of these components because precisely mounting the ESP32-CAM boardlet seems problematic. As Figure 7 shows, the top of the shuttle is essentially a traceless 3D-printed circuit board: a thin support with holes for the pins to go through. Although the boardlet was also set into a dab of hot glue, it is actually held in place by wire wrapping to make the pin connections. Flexible silicone-insulation wire-wrap wire made connections that would easily withstand the modest stresses of the angle and radius movements. Each wire-wrap connection was additionally set with a touch of solder.

It is useful to note that the threaded lens plate is separate so that it may be independently replaced with other sizes or styles of mounting plate. This was done primarily because the 4x4" lens mount boards used on many 4x5 cameras are slightly too small for some lenses. A lens mount can use up to the full 160mm clear coverage diameter, so there should be no mount-internal vignetting even using a lens board providing for tilt and/or shift.

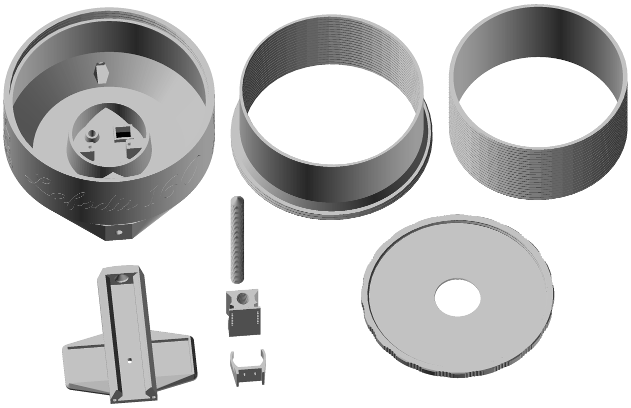

*Figure 6.* All 3D-printed parts for Lafodis160

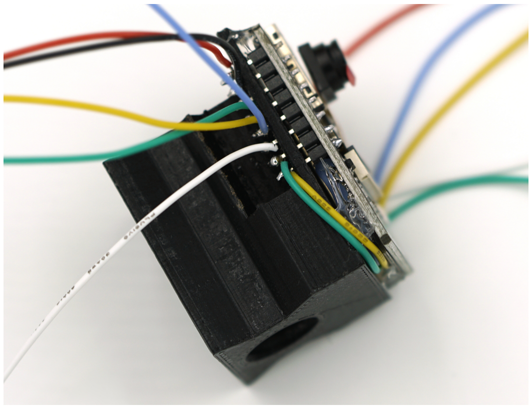

*Figure 7.* 3D-printed traceless PCB boardlet mount

## Firmware (on the ESP32-CAM)

As an IoT device, this camera is not really intended to be operated without a wireless connection to a controlling system. In fact, although a crude optical "sports finder" (wire-frame viewfinder) could be mounted in the top 1/4-20 thread, there is neither a power switch nor a shutter button on the unit. Thus, the system really requires both firmware in the camera and controlling software in a wirelessly-connected host.

The in-camera firmware is written in C++ using the Arduino environment. After an initial install using a wired connection via an CP2102 or similar USB to TTL serial converter, firmware updates are done wirelessly. At power on, Lafodis160 enables 802.11 WiFi to allow firmware updates, but if none is requested, then disables 802.11 and switches to ASCII text transmission over BlueTooth for all communication with a host control system.

The Lafodis160 firmware offers the wireless host a fairly sophisticated set of commands, briefly summarized here:

| **Command** | **Meaning** |
|---|---|
| ? | Version, currently 20201130 |
| , | Acknowledge when here with ; |
| . | Go to switch home |
| : | Set home here |
| $< n$ | Go to angle = $n$ |
| $> n$ | Go to angle = angle + $n$ |
| ! $n$ | Go to radius = $n$ |
| ˆ $n$ | Go to radius = radius + $n$ |
| $*$ | Capture and send image |
| = $r$ | Read register $r$ |
| { | Begin macro definition |
| } | End macro definition |
| @ | Apply macro |
| # | Ignore to end of line |

Each captured image is currently sent as a byte count followed by hex ASCII byte values. That is currently the capture framerate bottleneck, so the protocol may change in the future.

In addition to those commands, Lafodis160 maintains various configuration values in a set of registers that can be read or written. For example, to read the current contrast setting for the camera (applied to JPEGs), send "`=C`" – whereas "`C-1 `" will make the contrast setting -1. The registers are:

| **Register** | **Meaning** | **Range** |
|---|---|---|
| `A` | AE Level | -2:2 |
| `B` | Brightness | -2:2 |
| `C` | Contrast | -2:2 |
| `D` | Delay ms settle time | 0:8000 |
| `E` | AGC | 0:1 |
| `F` | Image effect | 0:6 |
| `G` | Gain | 0:30 |
| `H` | Hold steppers | 0:1 |
| `I` | WPC & BPC | 0:3 |
| `J` | Gamma raw | 0:1 |
| `K` | AEC & DSP | 0:3 |
| `L` | Lens correct | 0:1 |
| `M` | AWB | 0:4 |
| `N` | No. settling frames | 0:8 |
| `O` | Orientation | 0:3 |
| `P` | Pixel format | 0:7 |
| `Q` | Quality | 10:63 |
| `R` | Resolution | 0,3:10 |
| `S` | Saturation | -2:2 |
| `T` | Exposure time | 0:1200 |
| `U` | DCW | 0:1 |
| `V` | Verbose | 0:1 |
| `W` | AWB & gain | 0:3 |
| `X` | Radial steps/s feedrate | |
| `Y` | Angle steps/s feedrate | |
| `Z` | Where am I | |

The "`Z`" register keeps track of absolute position as a single number: $(radius * 2048) + (angle\&2047)$ where the $radius$ is unsigned and $angle$ is signed and constrained to be in -1024:1023.

This odd scaling is because the 28BYJ-48 stepper motor in full step mode rotates $11.25^\circ$ for a total of 32 steps per rotation, but internal gears then apply 32/9, 22/11, 26/9, and 31/10 ratios. The resulting overall gear ratio is 567424/8910, or 63.68395:1. That ratio is often approximated as 64:1, but in our case the difference can be significant. Thus, there are 2037.886419753 steps per revolution, meaning each step is $0.1766536135235732^\circ$. For the linear (radius) axis, this gives extremely fine motion control

***Figure 8.*** *3D-printed body parts of Lafodis 4x5*

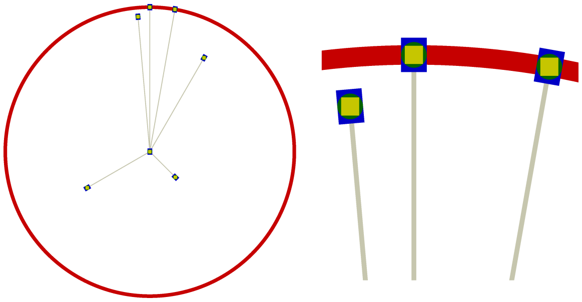

***Figure 9.*** *Scale diagram of angle, radius positioning of the sensor*

because the thread pitch of the linear rail drive screw is just 2mm, or 0.0009814mm per step; each pixel on the sensor is 2.2 microns, or 0.0022mm, so each step is less than half a pixel. The rotational axis movements are far less fine-grained. At a maximum diameter of 160mm, the linear distance covered by one step is approximately 0.246655mm; thus, one step covers 112.1159 pixels, and each image captured is a little less than 11 steps wide.

There is yet another complication involving the steppers: the ESP32-CAM's white LED and red LEDs on each of the ULN2003 drivers light as the steppers are powered. This light is unacceptable while an image is being captured. The answer is to turn power off on the steppers before capturing an image. Unfortunately, the standard Arduino stepper drivers do not support this type of power management; power stays on once a stepper has been accessed, and absolute positions are not tracked. Thus, we created a new stepper library for Lafodis160 that appropriately tracks both absolute position and power status.

## Software (on the host; control and stitching)

At this writing, we have some "first light" images from Lafodis160 obtained using simple C/C++ programs. There is also a preliminary Python script for controlling the scan. The existing code communicates with Lafodis160 using character streams through a serial device connected to Lafodis160 via BlueTooth. The C++ code uses OpenCV to assist in handling processing and display of captured image samples. Our experimenting with novel scan orders has just begun; the remainder of this section is thus a combination of motivation for building Lafodis160 and overview of the future work it is intended to support.

The angle, radius positioning used by Lafodis160 is a bit unusual and seems awkward for creating scans that are globally rectangular – but this was a well-considered choice. The first prototype, Lafodis 4x5, was designed to use conventional X, Y motion control. As Figure 8 shows, the body parts are more difficult to print than those of Lafodis160 and the steppers being positioned on the end of each linear rail meant a lot of internal space was wasted on providing a clear path for the drive system. In contrast, nearly all the internal space in Lafodis160 yields usable image area. For a 160mm coverage circle, the linear rail only needs to support 80mm of travel, as illustrated in Figure 9. Mounting the motor on the end of the radius rail nearest the center does not increase the body diameter of Lafodis160.

Despite angle, radius motion control, a conventional raster scan of any rectangle fitting within the 160mm circle is feasible, and the rectangle to capture can be freely rotated without rotating the camera body. Although the actual image dimensions on 4x5 film are somewhat smaller, the film itself is 120x100mm with an active area of 12,000mm$^2$ and a diagonal of 156.2mm – which is why a 160mm diameter was selected for Lafodis. A square format of 113mm on a side, with an active area of 12,769mm$^2$, also fits. The 135-film aspect ratio of 3:2 would become 133x88mm, with an active area of 11,704mm$^2$ – 13.5 times larger than "full frame." In the 16:9 aspect ratio used for TVs and many other displays, the largest would be a 139x78mm field with an active area of 10,842mm$^2$. Of course, the entire 20,106mm$^2$ field could be captured as a circular image inset within a 160x160mm rectangle, either to use directly or to select an aspect ratio after capture.

In Figure 9, the blue rectangles show the OV2640 sensor in various positions; no matter what angle the sensor is at, a square region (shown in yellow) fitting inside the green inscribed circle can always be extracted. The blue area is 3590x2684$\mu$m, thus the yellow region is a square just under 1.9mm on a side, and dividing any rectangular area into 1.8mm tiles of approximately 800x800 non-overlapping sensor pixels each would provide ample overlap for software refinement of alignment beyond mechanical positioning accuracy.

There is no hardware reason to prefer a raster scan ordering of samples. In a horizontal-first raster scan, vertically adjacent samples are temporally quite distant. In our unpublished work nearly two decades ago, we used a walk order corresponding to a Hilbert curve[2] (modified to allow non-power-of-2 dimensions), which dramatically reduces the statistical temporal distance between physically-adjacent samples. However, the angle, radius positioning system makes it efficient to also consider scan orderings derived from polar-coordinate walks, ranging from spirals[13] to Hilbert-like orders.

Even more interesting are walk orders that are dynamically adjusted based on the scan data collected up to that point. The normal model for scanning uses a pre-determined scan order to collect all data and then corrects, aligns, and stitches all the images together. The primary motivation behind Lafodis160 is to experiment with an incremental stitching process that dynamically modifies the scan order.

Although image samples taken with Lafodis160 may need to be slightly shifted and rotated to correct for small positioning errors in aligning for stitching, that is a far smaller and simpler search space than image stitching normally has to search. Thus, the computational overhead in aligning each sample as soon as it is received is not prohibitive, and each sample can incrementally update the system's understanding of the scene being captured. It is also possible to use the ESP32-CAM's support for different resolutions to obtain quick samples to determine if complete higher-resolution sampling is needed for an area – if there's no detail in a section of blue sky, why bother sampling it at full resolution?

Our earlier work in time domain continuous imaging (TDCI)[3] explored ways to use the fact that pixel data, which can be obtained from asynchronous sampling, can be used to construct a model of how each pixel's value continuously changes over time. This modeling reveals when scene content has changed in a statistically significant way – beyond noise. The analysis can be extended to process spatially-overlapped samples from Lafodis160, thus enabling detection of when an area's scene content has changed enough to warrant re-sampling it.

At Electronic Imaging 2020, we presented Senscape[4]: methods for sensor data fusion to produce an aligned master image that merges image data from multiple sources based on explicitly modeling value confidence. The plan is to use TDCI-based analysis to compute confidences that can incrementally merge each sample's data into a master image that tracks both pixel values and confidences. The confidences would be maintained per pixel in the assembled master image; thus, a sample taken with 2.2$\mu$m sensor pixels would have multiple sensor pixels map to the area of each 5$\mu$m "virtual pixel" in a 500MP 4x5 master image, and the confidence metrics would be merged at the virtual pixel level (as they were in Senscape).

## Conclusion

This paper has presented the detailed design of, and motives for creating, the Lafodis160 large-format wireless IoT camera. The system is fully open source and relatively easy to build. Despite being capable of gigapixel resolution, the system also is stunningly inexpensive to build, with parts cost under $100 including a suitable lens, and half that without. The authors are unaware of any other camera with similar properties.

There are disadvantages to the approach used in Lafodis160. Perhaps most obvious is that it is a scanning system moving a very small sensor, so high resolution scans thus take a very long time to capture. In fact, the system's angle, radius motion control is harder to use than more conventional scanning cameras or camera backs, but that awkwardness buys the ability to efficiently implement more intelligent, even dynamic, scan orderings. First and foremost, Lafodis160 is a proof-of-concept prototype and platform for experiments involving novel scan orderings and stitching techniques.

Everything about Lafodis160 will be available at, or linked from, `http://aggregate.org/DIT/Lafodis160`. It is our intention that this camera will not only serve our research needs, but also will be useful to others who build their own. In fact, there has been some interest in the system for purely artistic photographic purposes, and we are working on an improved version which will be even simpler for people to build while providing somewhat more robust and accurate angular motion control.

## References

[1] "GigaPan EPIC Pro User Guide," http://gigapan.com/, January 12, 2021.

[2] B. Moon, H. V. Jagadish, C. Faloutsos and J. H. Saltz. "Analysis of the clustering properties of the Hilbert space-filling curve," in IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 1, pp. 124-141, Jan.-Feb. 2001, doi: 10.1109/69.908985, 2001.

[3] Henry Dietz, Paul Eberhart, John Fike, Katie Long, Clark Demaree, and Jong Wu. "TIK: a time domain continuous imaging testbed using conventional still images and video," Society for Imaging Science and Technology, Electronic Imaging, Digital Photography and Mobile Imaging XIII, pp. 58-65(8), https://doi.org/10.2352/ISSN.2470-1173.2017.15.DPMI-081, 2017.

[4] Henry Dietz and Paul Eberhart. "Senscape: modeling and presentation of uncertainty in fused sensor data live image streams," Society for Imaging Science and Technology, Electronic Imaging, Computational Imaging XVIII, pp. 392-1-392-7(7) https://doi.org/10.2352/ISSN.2470-1173.2020.14.COIMG-392, 2020.

[5] LargeSense, http://largesense.com/, accessed February 3, 2021.

[6] Christopher Perez and Kerry Thalmann. "Test Results - Large Format photographic lenses," https://web.hevanet.com/cperez/testing.html, accessed February 3, 2021.

[7] BetterLight. https://www.betterlight.com/index.html accessed February 3, 2021.

[8] Rencay. http://www.rencay.com/, accessed February 3, 2021.

[9] "Fotodiox Pro Lens Mount Adapter - Sony Alpha E-Mount Mirrorless Camera to Large Format 4x5 View Cameras with a Graflok Rear Standard - Shift / Stitch Adapter," https://fotodioxpro.com/products/snye-4x5-pro, accessed February 3, 2021.

[10] Randy Sarafan, "DIY 4X5 Camera Scan Back," Instructables, https://www.instructables.com/DIY-4X5-Camera-Scan-Back/, November 25, 2018.

[11] OmniVision. "OV2640 Color CMOS UXGA (2.0 MegaPixel) CameraChip$^{TM}$ with OmniPixel2$^{TM}$ Technology," February 28, 2006.

[12] Last Minute Engineers. "Control 28BYJ-48 Stepper Motor with ULN2003 Driver & Arduino," https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/ accessed February 3, 2021.

[13] P. Xu, C. Nguyen and S. Tirthapura. "Onion Curve: A Space Filling Curve with Near-Optimal Clustering," 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, 2018, pp. 1236-1239, doi: 10.1109/ICDE.2018.00119, 2018.