KAMF: An Interchangeable-Lens Mirrorless Camera Made From A Kid's Camera

Henry Gordon Dietz; Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky

Abstract

Very low cost consumer-oriented commodity products often can be modified to support more sophisticated uses. The current work explores a variety of ways in which any of a family of small rechargeable-battery stand-alone cameras, intended for use of kids as young as three years old, can be modified to support more sophisticated use. Unfortunately, the image quality and exposure controls available are similar to what would be found in a cheap webcam; to be precise, the camera contains two separate camera modules each comparable to a webcam. However, simple modifications convert these toys into interchangeable-lens mirrorless cameras accepting lenses in a variety of standard mounts. Cameras so modified can capture full spectrum images or can employ a filter providing any desired spectral sensitivity profile. One also has limited access to the internals of the camera, easily allowing options like wiring to an external exposure trigger. The cameras record on a TF card, and typically a 32GB or 64GB card is included with the camera at a total cost virtually identical to the cost of the TF card alone.

The current work can be considered as both a study revealing the internal construction of a kids camera and a guide to adapting it for more serious uses as the KAMF mirrorless camera.

Introduction

As cell phones have pushed stand-alone cameras into a specialty niche, the low end of the commodity camera market has been decimated. Thus, ironically, as imaging sensors and systems have become cheaper, the cost of traditional low-end cameras has gone up. Using ESP32-CAM[1] and similar boardlet-level camera guts intended for IoT devices, one could make a very basic low-cost mirrorless camera, but the work involved in making the system into a stand alone camera is complex and adds significant cost. For example, adding a battery and charging interface, display, and housing to an ESP32-CAM is just too much work for many applications. However, stand-alone toy cameras targeting kids from age 3 years up have become common and extremely cheap. Figure 1 shows a few of the general type studied in the current work. The goal is to understand exactly what is in these toy cameras and to determine if simple modifications can cost-effectively turn them into stand-alone cameras suitable for more serious applications.

There are three flavors of stand-alone camera this work produced and evaluated. The first is essentially using the stock camera directly, although perhaps augmenting its control. The two other versions, both named KAMF, involve more significant modifications using 3D-printed parts to convert the camera into

Figure 1. *Tiny and obviously targeting kids, it is a real camera*

an interchangeable-lens mirrorless configuration. One of those configurations uses the camera's tiny front sensor directly, the other emulates key properties of a medium-format sensor.

Reverse engineering commodity systems for the purpose of extending their functionality is not a task in which most research groups can easily justify expending effort. It then can take considerable additional effort to create a simple set of modifications that others can easily apply. Thus, the main objectives of the current work are to perform the necessary analysis and design work to create an ultra-low-cost stand-alone camera platform that easily can be used by the research and hobbyist communities. The capabilities of the completed system are also realistically evaluated and custom postprocessing software, `postkamf`, was developed to maximize the final image quality delivered.

The Stock Camera

The camera studied appears to be very widely available. The ones tested came from 14 different vendors selling through Amazon. Aside from variations in pricing, retail packaging, bundled accessories (such as a lanyard or carrying strap or a USB TF card reader), and silicone half-case designs (e.g., the unicorn and dinosaur versions seen in Figure 1), the cameras from various vendors differ in a variety of more technically significant ways. Unfortunately, the advertisements generally do not give many specifications and those they do give are often wildly incorrect. For example, the very first specification listed by several is "Compatible Mountings: Micro Four Thirds" when in fact the only thing this camera has in common with Micro Four Thirds is that the sensor delivers still images with the same 4:3 aspect ratio. There are similar hallucinations about having a zoom lens, the camera being a DSLR, and various impressive sensor resolutions are claimed when at best the resolution quoted approximates the pixel count in the heavily upscaled JPEG images produced. It is also interesting that only one of the cameras has any brand markings at all: the FurUlicty version has the brand printed on it. In sum, it is not possible to reliably identify camera versions without directly examining them.

A few of the most technically relevant and obvious differences across the cameras from each of the 14 vendors is summarized in Table 1. Note that the table entries reflect the cameras tested, which may be significantly different from those now being supplied by the same vendors. The maximum still image pixel counts run from 11MP to 100MP. Maximum video resolution also varies, although all versions have the same image height. The aspect ratio of the video pixels is often not quite 1:1; the table resolution is marked with a T (for Tall), W (for Wide), or F (for 4:3 aspect) to indicate the video aspect ratio. Similarly, a suffix of A is added if useable Audio is recorded; some cameras either don't have a mic or simply fail to record sounds clearly. The type of USB connector used for charging the camera is noted in the table along with an M suffix if the camera can also function as a USB mass storage device to access files on a TF card. The last field gives the marked size of the included TF card in GB. Despite the large pixel dimensions of the JPEG files created by these cameras, one image typically uses between 1MB and 1.5MB of storage space, so a 32GB card might hold more than 20000 images at maximum resolution. The cards supplied generally have very slow write speeds that are not a problem in these cameras, but make them inappropriate for use in devices that write much larger files.

Table 1: Basic Characteristics of Various Versions

Vendor	MP	Video	USB	GB
BEIARA	48	1440×1080A	µM	32
DIGITGIFT	48	1920×1080TA	C	32
FurUlixty	48	1440×1080A	C	64
Gavonde	40	1920×1080TA	µ	32
Geavo	48	1920×1080TA	C	32
HOAP	48	1440×1080	C	32
JAVIDE	48	1920×1080FA	C	16
jazeyeah	48	1440×1080F	C	–
JUGVOLX	48	1440×1080	C	32
kiisda	100	1440×1080FA	C	64
SplashNSpray	48	1920×1080TA	CM	32
Warmtown	48	1920×1080WA	µM	32
Yatao	11	1440×1080FA	µ	32
yunpkture	48	1440×1080	C	–

Externally, the camera housings are nearly identical, differing only in the color and shiny vs. matte texture of the plastic used. To discover more details, one must look inside multiple versions of the camera. For all versions, the back of the camera is held to the front by four screws. When opened, it immediately becomes apparent that although the outsides are identical and even the internal layout is very similar, there are minor differences. For example, the front half of the body shell mounts a lens shroud, made of white plastic on all of these versions, which serves only as a fixed enclosure for the front-facing camera. The pink and blue versions shown in Figure 2 differ significantly in the shape of the body opening that is covered by the white lens shroud, and the shroud is attached to the body by 3 screws in the pink camera and only 2 screws in the blue model. Actually, most versions examined used the 3-screw mount design seen in the pink model but secured the part using only 2 screws. Overall, the plastic body parts are extremely similar across versions, making it practical to design 3D-printed components that can substitute for or be used with any of these parts.

The front chip camera is mounted via edge clips and a drop of glue into a black plastic carrier which in turn is attached to the white plastic lens shroud using two screws. In most versions, the front of the black plastic part mounts a clear plastic lens using a small black plastic retaining ring. This plastic lens element serves no optical purpose, but does prevent dust and dirt from getting close to the actual lens. Because the front camera cable comes off a connector on the top of the circuit board mounted in the back half of the body, it has a surprisingly long cable: about 60mm. It is very difficult to be certain of the camera chip model, but markings on the cables of some suggest that the chip-camera is a GalaxyCore GC0308 using a DVP (Digital Video Port) interface. There are only 640x480 sensels on that sensor, but the chip specifications say that the sensor has an RGB Bayer filter and the FSI (Front Side Illuminated) sensels are relatively large, 3.4µm. Video output at up to 30FPS (Frames Per Second) is supported by that sensor. Another version seems to use a BYD Microelectronics BF3A03, which is similar but has a slightly smaller 3.15µm pixel size. The sensor models used for the rear-facing selfie camera appear to be very similar to those used for the front camera, also employing a DVP-interface chip camera but with a much shorter cable. This is consistent with the fact that image quality of the front and rear cameras is nearly identical.

Also attached to the front, above where the lens is, the speaker is fastened into a little rounded-rectangular cavity using a sticky gasket. Five tiny holes in the plastic body let the sound out and a red and black pair of wires provides monophonic output to the speaker. In most versions, the relatively long cable on the front chip camera is tack-glued to the back of the speaker, presumably to keep the cable from getting pinched as the body

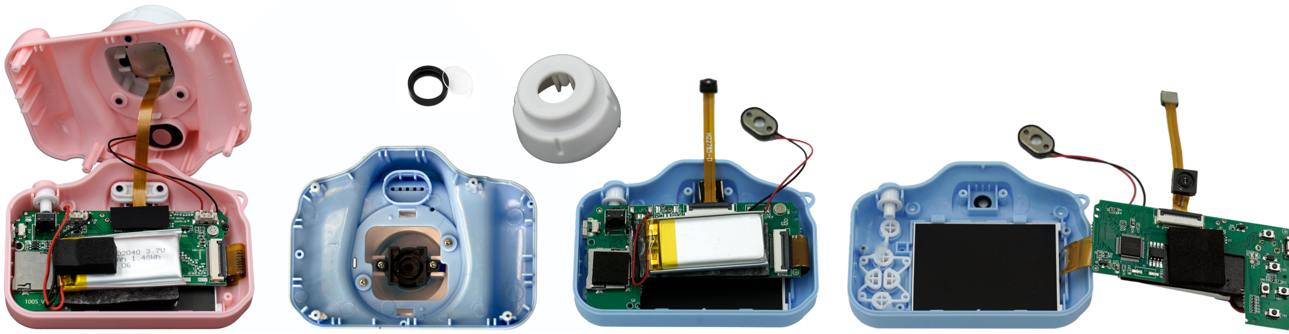

Figure 2. *Inside a couple of versions of the stock camera*

is closed, but that glue bond can be harmlessly broken. The microphone for audio recording (which is not populated on some versions) is the little metallic cylinder on the circuit board to the upper right of the lithium ion battery pouch. There is a tiny hole in the plastic back of the camera over the microphone, and that easily explains why some versions have very poor audio recording, especially if the sound source is in front of the camera.

The power source for the camera is typical of this class of rechargeable consumer device: a 3.7V Lithium Polymer Ion (Lipo) battery of approximately 40x20x5mm dimensions. The capacity varies across the different models and is not always marked, but a full charge can power most versions of the camera for an hour or more of use. This type of soft-package Lipo battery is notorious for bursting into flames if the battery package is punctured, so the cameras have some padding to protect their battery from puncture by other components within the camera. Defective Lipo batteries will often swell when charging; such batteries should not be used – dispose of them properly. Although it was not apparent from the outside, one of the cameras tested contained a Lipo battery that swells enough during charging to be a concern.

Under the battery is the single circuit board which runs everything. Unfortunately, different versions apparently have different microcontrollers, which combined with the lack of documentation makes reprogramming the cameras impractical. Although the part numbers make it difficult to confirm, it is most likely that a Renesas R8C processor powers versions using micro USB while an Arm Cortex-M4 is used in USB C versions. The processor is mounted on the back side of the circuit board along with the rear camera DVP port, USB connector, and five tiny momentary switches. The front side of the board holds the TF card slot, a "factory reset" switch (operated by pushing a pin through a small hole in the side of the body), shutter button switch, front camera DVP port, audio output wires, microphone (not populated on some versions), and rear color LCD panel interface. All the switches use 3.3V signaling, so it would be easy to implement wired remote control or to use a second microcontroller to control the unit by virtually pressing buttons. It is likely that a second microcontroller such as a $3 ESP8266 boardlet could be powered by the camera's battery to allow WiFi or Bluetooth remote control of the camera. It might even be possible to fit an ESP8266 boardlet within the stock camera housing, but be very careful not to do anything that could puncture the Lipo battery.

The back of the camera mounts a fairly standard 2" color LCD that resolves about 320×240 pixels and is brightly backlit. There are also a few small white plastic parts: flexible button covers that allow pressing the switches on the circuit board from the outside of the camera and a little retaining bar to hold the rear chip camera in place. The retaining bar has two screw holes but in most versions is held with just one screw.

In summary, in low quantity, the parts within one of these cameras would have a retail cost significantly higher than the typical sale price of these cameras. For example, on Amazon an LCD panel alone that is similar to the one in the camera commonly retails for around $15, while these cameras can be bought for as little as $8 each. This was the realization that inspired the current work...

Use of the Stock Camera

Although this article is primarily about modifying these cameras, they are cheap, tiny, usable, stand-alone cameras as sold.

The camera menu and options vary significantly across versions, but all are controlled using the menu and buttons seen in Figure 1 and most versions use the buttons in approximately the same ways. The top button is primarily the shutter button, but is also used to select the current option. The highest-placed button on the back of the camera serves both as the power on/off button and to exit the current option. Just below that is a cluster of four buttons. The top and bottom buttons respectively point up and down, and they are used to move through options. The left and right buttons respectively have a gear icon and a picture icon, but they are also used to move through options.

Before any serious use, there are a few settings that should be changed from their defaults. Power the camera on by holding the power button down for a couple of seconds and releasing it (a long press). The main menu screen seen in Figure 1 will come on with the top leftmost option pulsing a camera icon. Press the down button to go to the gear icon option and press the shutter button to enter the configuration menu. Options you may want to set include:

- Size: Although any resolution setting just controls how much the 640×480 sensor data is interpolated up for still captures, JPEG compression artifacts become less significant at higher enlargements and lower resolutions do not

save much file space. Thus, selecting a high resolution like "48M" is a reasonable choice even though the captured image will be scaled down to just 640×480 for use or further processing.
- Capture Timer: This is the self timer and should generally be turned off.
- Resolution: This sets the video capture resolution.
- Time Stamp: If this option is turned on, the camera imposes a text timestamp in the bottom left of every capture. It should generally be turned off.
- LCD Off: This allows you to set how many minutes the rear LCD panel will remain on after a button press. Setting this to Off actually disables the timer for turning off the LCD so the display stays on until the device is powered off.
- Auto Off: This determines how long the camera can sit without a button press before turning itself off. The Off option keeps the camera from automatically shutting down.
- Volume: If you don't want to hear kid-oriented sounds for most operations, set this to 0. On most versions, the camera still plays a sound for a few operations (e.g., power on or off) even if the 0 setting has been selected.
- Date Time: This lets you set the date and time, which are recorded in the JPEG file metadata even if the timestamp option is disabled. Note that the camera forgets the date and time if the Lipo battery has no charge.

The main (power on) menu top row of icons select still image capture, video capture, and playback modes. When in still capture mode, long presses on the up or down buttons apply "digital zoom" – a fairly useless feature using the stock camera's fixed-focus lens, but very useful for checking critical focus when the camera has been modified to use a manual-focus lens. Short presses of the up or down button cycle through a variety of kid-oriented image effects that you almost certainly do not want. The right button switches between the front and rear cameras while the left button enables or disables the self timer.

Aside from use by kids (who may also appreciate the built-in games not discussed here), the stock camera is most suitable for use as a very small and lightweight (55.1g including TF card) stand-alone still camera that is cheap enough to be expendable. As can be seen in Figure 3, still images captured only contain 640×480 pixels of scene information, but are of better quality than one might expect. The video quality is poorer than some comparably-priced video-only alternatives, and thus is not as compelling. Some versions of the camera, but not most, support continuous loop video recording. Note that all these cameras have the annoying feature that they can capture to internal memory when no TF card is present, but there generally is no way to get such captures out of the camera.

It is straightforward to implement wired control of the stock camera by soldering wires to the switch contacts on the circuit board. It is even possible to route the wires out of the camera without drilling a hole by removing one of the white plastic keycaps and using that opening. This type of wired control was not experimentally tested in the current work.

The following two sections center on significantly modifying the camera to allow use of other lenses. Although there are

Figure 3. *Unprocessed images taken with the stock camera*

two very different types of modification discussed, both are referred to by the acronym KAMF. Not by coincidence, "kamf" is a Yiddish word meaning "struggle," and a struggle it was to make these toys into devices that can be treated as serious mirrorless cameras. One version replaces the stock lens to make the front camera into one of the smallest interchangeable-lens mirrorless cameras available: Kentucky's Adorable Micro Format camera. The other version uses both the stock lens and an additional lens to allow the front camera to capture images with the depth-of-field of a medium-format camera. It is a BSI DCO (Back Side Illuminated Digital Camera Obscura) called Kentucky's Approximation to Medium Format camera.

KAMF: A Micro Format Mirrorless

The conversion of this kid-oriented camera into an interchangeable-lens mirrorless camera creates **Kentucky's Adorable Micro Format**, abbreviated KAMF. One might wonder if anyone wants such a camera, but that question has been answered with the wildly successful conclusion of a KickStarter campaign for the Yashica – I'm Back Mimi[2]. The roughly $300 Mimi claims to be the "world's smallest mirrorless" at 100g and 77×50mm without its battery grip and at this writing is switch-

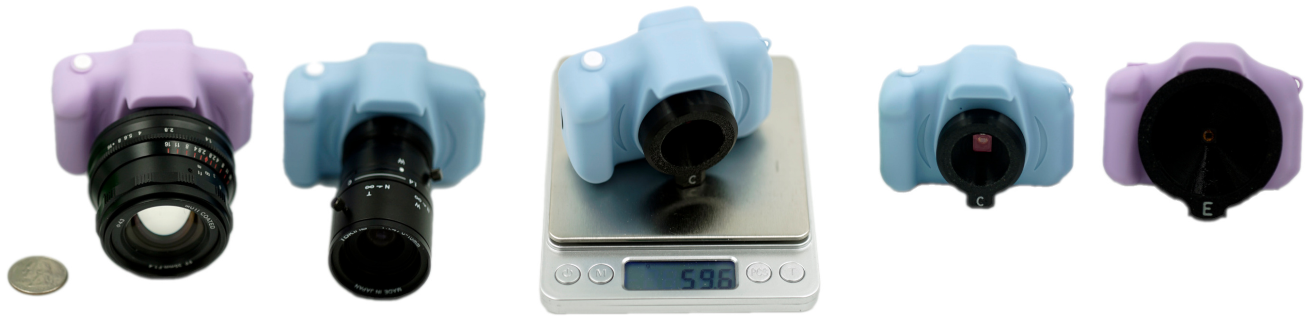

Figure 4. *A US quarter and Kentucky's Adorable Micro Format (KAMF); blue is visible light C mount, purple is full-spectrum E/FE mount*

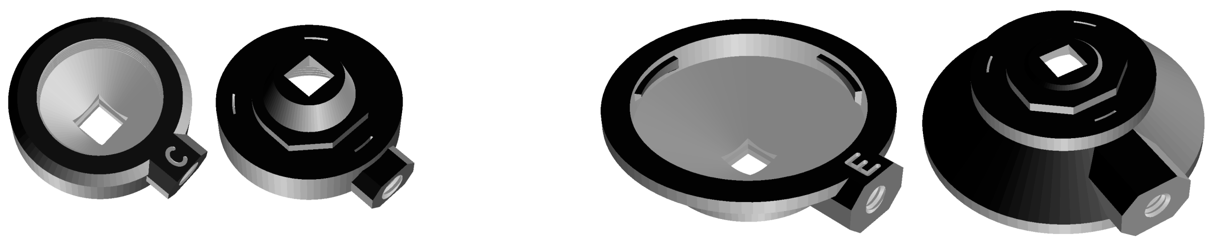

Figure 5. *3D printable KAMF micro C and Sony E/FE mounts*

ing from a proprietary bayonet mount to a C mount in front of its 12MP 5.6× crop sensor. However, the C mount KAMF is under 60g and 83.5×62mm and the E mount version of KAMF weighs under 65g and is 83.5×71.25mm. Both C and E versions of KAMF are shown in Figure 4. KAMF has a lower resolution 9.6× crop sensor and does not offer quite as many base features as Mimi, but it is a hackable open source design leveraging commodity kid camera parts to bring build cost to well under 1/10 the cost of Mimi.

The first few KAMF prototypes replaced the entire front half of the camera housing with a 3D-printed part, but it is sufficient to replace the white plastic lens shroud with a 3D-printed lens mount. The 3D-printable designs for the C and Sony E/FE lens mounts that replace the lens shroud are shown in Figure 5. They attach using the exact same screws that held the shroud, which is a bit of a trick in that those screws are too small to 3D-print a mating thread. The solution was to print narrow slots that will easily catch and hold each screw.

Unfortunately, the black plastic carrier that the chip camera clips into is problematic and also must be removed for this modification. There are three issues with that part. First, the carrier has a tube that extends significantly forward to accept the protective clear plastic element, and that extension restricts the incident angles at which light can hit the sensor: many of the lenses one might want to use have large rear elements that could be vignetted by that tube. Second, the tube would interfere with the placement of an 8×8mm filter, which is needed, for example, to prevent NIR light from being sensed because removing the original lens also removes the NIR-blocking filter integrated with it. The third issue is that the chip camera must be removed from the plastic carrier anyway in order to remove the original lens. The solution is to not use the carrier for the chip camera, but to directly attach it to the 3D-printed lens mount. The only awkward aspect of this arrangement is that it can be difficult to attach it perfectly flat.

Here is the step-by-step modification sequence:

1. 3D print the KAMF lens mount. It can be printed using almost any FDM or resin printer, but the gentle slope on the E/FE mount may require adding supports. Many plastics are colored using organic dyes which are transparent in NIR, so test your material and paint the inside with something like Black 2.0 if it is not fully opaque.
2. Use a small Phillips screwdriver to remove the four screws from the back to open the camera shell.
3. The front chip camera is clipped into a black plastic carrier and typically locked in place with a drop of glue. Gently scrape the glue dot off the back of the chip using a small Flathead screwdriver. Using the same screwdriver, pry the chip out of the clips on the back of the carrier.
4. Use a small Phillips screwdriver to remove the screws that attach the white plastic lens shroud to the camera body. These same screws and holes in the body can then be used to attach the selected 3D-printed mount. However, if you will be using an 8×8mm filter, set the filter into the square filter recess in the front of the 3D-printed mount before attaching the mount to the camera.
5. The original lens is a tiny thing screwed into the front of the chip package. It does not have a focus mechanism per se, but uses turning within the screw thread to adjust focus. The lens focus is factory-set at the hyperfocal distance to give the largest possible range of in-focus distances, and a drop of glue is used to lock that focus setting. Breaking this glue bond to unscrew the lens can take a fair amount of force – and all the parts are small and made of plastic. The easiest way to break the glue bond seems to be using

Figure 6. *Monochrome 5760×7680 pixel full-spectrum from KAMF micro*

one pair of pliers to hold the sides of the chip package while using another pair of pliers to grasp and unscrew the lens. With the lens removed, the sensor is exposed as a tiny chip sitting in the middle of the threaded package – don't touch the sensor chip!

6. In this step, the chip package will be set into the back of the 3D-printed adapter. The orientation of the chip package should be identical to how it was in the original carrier, which is actually upside-down to correct for the fact that the image projected by the lens is also inverted. The square opening in the back of the printed adapter makes it easy to center the chip, but it doesn't prevent the chip from being tilted. To avoid tilting, place a narrow piece of stiff tape across the back of the package and carefully press the package into place with the back flush with the surface of the 3D-printed part. Use a few drops of glue or hot glue to tack the package in place.
7. Carefully position the cable and wiring to close the camera shell. Screw it together using the Phillips screwdriver and screws you removed in step 2.

Although 3D-printed mounts are fairly strong, it should be obvious that these tiny screws should not be supporting the weight of a very heavy lens. Always support a heavy lens directly rather than holding only the camera body, and note that the 3D-printed mounts all have tripod mounts so those tiny screws should only have to bear the weight of the camera body. Also be aware that some 3D-printed lens mounts do not lock the lens in the mounted position – and neither the C nor E mount here has a locking mechanism to prevent a mounted lens from being accidentally dismounted.

Figure 7. *A color capture from KAMF micro*

What image quality do you get from KAMF micro? Obviously, that depends significantly on what lens you use and how carefully you manually focus (use the digital zoom for critical focusing). The relatively large pixel size means that most lenses can resolve well enough, although perhaps not wide open. Figures 6 and 7 are fairly typical of the quality of the JPEG images straight out of the camera using KAMF micro wide open with a fast C-mount zoom lens. The monochrome full-spectrum image in Figure 6 was captured at the 48MP setting of the camera, but is actually 44MP with 5760×7680 pixels, and resolves only about 640×480 pixels. Note that the color cast toward the corners in Figure 7 is easily correctable using `postkamf` – as discussed in a later section of this paper. Casts and modest vignetting are common with adapted lenses on KAMF micro because the chip cameras apply corrections for the supplied lens, and the lenses adapted usually have very different ray angles to the sensor.

KAMF: A Medium Format BSI DCO

While the conversion of this kid-oriented camera into a very compact interchangeable-lens mirrorless camera is fairly intuitive, it is more difficult to see how a camera with a tiny sensor can function as a medium-format camera with a 44×33mm sensor – approximately 300× larger. The trick used in **Kentucky's Approximation to Medium Format** (KAMF) is building a digital camera obscura (DCO). Using a lens that can cover an image circle of 55mm diameter to project on a 44×33mm screen, the depth-of-field and view angle of the projected image precisely match what a sensor that large would capture. KAMF simply uses the original lens and sensor to photograph the screen image. Earlier work[3] distinguished two types of DCO: a front side illuminated (FSI) DCO has the camera on the same side of the reflective screen as the primary lens, whereas a back side illuminated (BSI) DCO has the camera behind a transmissive screen.

Figure 8. *KAMF DCO with E/FE-adapted Minolta MD 50mm f/1.7*

Figure 9. *Shot with Minolta MD 50mm f/1.7 on KAMF DCO*

Figure 10. *Shot with Spiratone 135mm f/1.8 on KAMF DCO*

This KAMF is a BSI DCO.

Of course, the light sensitivity and image quality of a small-sensor BSI DCO is significantly lower than that of a true medium-format camera. There is also the secondary issue that medium-format lenses tend to be pricey, and it seems a waste to put an expensive lens on the KAMF DCO. The good news is that lots of lenses that were not designed for medium-format, including many old lenses designed for 135-format film SLRs, have large enough coverage circles. As part of the development of KAMF, coverage of various SLR lenses was tested[4]; there is also a table summarizing which lenses cover the 44×33mm sensor in Fuji GFX cameras[5]. For example, Figure 8 shows KAMF with a Minolta MD 50mm $f/1.7$ lens which is commonly available used for under $25; on KAMF, it behaves like a 40mm $f/1.3$ would on a 135-format camera, produicing images like Figure 9. Figure 10 was shot using a Spiratone 135mm $f/1.8$ lens which cost the author $150, which is amazingly cheap for a 135mm lens that fast covering medium format. There is slight vignetting in those images due to minor misalignment of the sensor in KAMF with the screen, and the images shown have not been improved by postprocessing, but the images have the medium-format look.

Here is the step-by-step modification sequence:

1. 3D print the KAMF projection chamber and at least one of the primary lens mounts. These parts can be printed using almost any FDM or resin printer without adding supports, but they do need to be light tight even in the NIR. Many plastics are colored using organic dyes which are transparent in NIR, so test your material and paint it with something like Black 2.0 if it is not full opaque.
2. Use a small Phillips screwdriver to remove the four screws from the back to open the camera shell. Then use that screwdriver to remove the screws that attach the black plastic chip package holder to the white plastic lens shroud and also the screws that attach the white plastic lens shroud to the body. Use the screws that held the white plastic lens shroud to attach the 3D-printed KAMF 44×33mm BSI DCO projection chamber to the body.
3. The projection chamber allows the intact black plastic chip package holder to be mounted easily and precisely, but the focus of the original lens must be adjusted. If the chip package holder has the plastic element and retaining ring, remove those. You will need to turn the original lens within its threaded mount to focus on the DCO screen, and you will most likely need to temporarily remove the chip package from the holder to make that adjustment. It should not take a lot of rotation to focus at the correct distance, but the only way to be sure is to turn the open camera on and use the digital zoom to check critical focus on the LCD while you are turning the lens. Once focus seems correct, insert the chip camera back into the holder and put the holder into the back of the projection chamber rotated 180 degrees from how it was attached to the white plastic lens shroud; this rotation compensates for the fact that the adapted lens will project the image upside-down. The cable to the camera needs to be twisted for this, so you might need to free the cable from where it is glue-tacked to the back of the speaker. Temporarily tape a thin sheet of paper with fine printed detail against the front of the projection chamber with a strong light in front: the image on the LCD should be sharp. Once

the image is satisfactory, use a drop of glue or hot glue to lock the black plastic holder into place.

4. Carefully position the cable and wiring to close the camera shell – the twisted cable makes this touchier than for the micro KAMF. Screw it together using the Phillips screwdriver and screws you removed in step 2.
5. In this step you will replace the taped test pattern on the front of the projection chamber with an appropriate screen. Choice of screen material for a BSI DCO is a more nuanced thing than one might expect[3]. The material should be thin and translucently diffuse with relatively little texture. The piece must be large enough to cover the 44×33mm screen opening; a piece cut to 47×36mm can easily be fixed to the front of the projection chamber using tiny bits of tape or dots of glue. Alternatively, the front of the projection chamber can be covered in glue and a 58mm diameter screen can be pressed onto the front. A less permanent method for fixing the screen in place makes it easier to change the screen if it becomes damaged or you find a more suitable material.
6. The final step is to screw on your choice of 3D-printed primary lens mount. Current designs include Sony E/FE, Nikon Z, and M42 screw thread mounts. This part can be changed without opening the camera, so changing the primary lens mount is possible even in the field. Figure 11 shows what KAMF looks like with the E/FE mount and the projection screen visible.

Once the camera has been modified as described above, it operates normally as a manual-focus medium-format digital camera. However, judging critical focus can be difficult on the rear screen unless you use the digital zoom as a focus aid. Highly diffuse screens also significantly reduce the light seen by the camera, which already was not very good in low light; you will need bright lighting for the best results.

Another potential issue is that some combinations of lens vignetting and insufficiently diffuse screen material will create a very bright "hot spot" in the center of the image. Ideally, `postkamf` should be able to correct for this. However, because KAMF does not provide exposure overrides, such a hot spot can easily saturate the sensor producing unusable images that cannot be repaired by postprocessing. Stopping down the lens will usually help even the exposure.

Alternatively, it is possible to gently darken toward the center of the screen material to compensate, for example by printing a pattern with an inkjet printer (preferably using dye rather than pigment ink). A filter pattern to be inkjet printed can be made by capturing a plain white scene with the camera and then inverting the pixel values. This method of correction is very similar to the concept of using a center filter[6], as is commonly done with ultra-wide large-format lenses. In this case, the darkening of the center is in the sensor plane, so it does not adversely alter the bokeh by imposing a bright edge, but too-coarse shading may add to the texturing of the screen. How coarse is problematic? KAMF covers 44mm with 640 pixels, so the virtual medium-format pixel size is 68.8μm, or about 375 pixels per inch. Inkjet-printed patterns at significantly higher dots-per-inch resolutions, e.g., 1200DPI, should not cause objectionable texturing. Use of a shaded screen is only recommended if the same lens will nearly always be used and with the same aperture setting. Slight mismatches between lens hot spotting and the compensatory shading of the screen can be corrected in postprocessing.

Figure 11. *KAMF DCO with E/FE mount and screen visible*

Postprocessing Using postkamf

As charming as these little cameras are, their image quality is far from compelling. Thus, the `postkamf` program was created to see how much image quality could be improved by state-of-the-art postprocessing. This program was written in C++ using the OpenCV library[7] and OpenMP[8] directives to speed-up some operations by parallel processing across multiple processor cores. This software incorporates a variety of both standard and recently-developed enhancement techniques.

Perhaps the most basic transformation applied by `postkamf` is correction of vignetting and color casts. The cameras are surprisingly well corrected for the lens assembly they ship with, but that is because they internally apply corrections. These issues are largely rooted in how ray entry angle varies across the sensor, and the relatively long rear focus of the interchangeable lenses used with KAMF micro format have ray angles differing significantly from those of the original lens assembly. The KAMF BSI DCO still uses the original lens assembly so it does not have that ray angle issue and suffers relatively little position-dependent color cast, but most lenses do not cover the 44×33mm screen very evenly and most screen materials add some texture. Thus, `postkamf` allows a white reference frame, captured using the same configuration of KAMF to photograph an evenly-lit white surface, to be specified using the `-w` command line option. The white frame is processed to produce a multiplicative correction map that is applied to the 640×480 sensor pixels assumed to be the basis of each image. This correction is responsible for most of the difference seen in Figure 12, which is an image shot with KAMF micro format using a C mount lens.

The next enhancement applied by `postkamf` is a denoising and sharpening process implemented by computing a pixel value error probability density function (PDF) from the image[9] and then using it to guide a statistical texture synthesis enhancement algorithm[10]. This state-of-the-art enhancement is computationally expensive, but still fairly quick when applied to 640×480-pixel images using multi-core parallel execution via OpenMP.

Figure 12. *Unprocessed vs. basic corrections using* `postkamf`

The improvement is quite significant for noisy images and noise is plentiful when these cameras are used in poor lighting. Thus, this option is enabled by default unless disabled by `-k` on the command line. Close examination of Figure 12 will reveal that the `postkamf`-processed image has significantly improved noise and sharpness due to this processing.

Although the result of processing by `postkamf` is normally a single 640×480-pixel 24BBP color image, it also is capable of super-resolution processing to upscale the resulting image. Two different methods are implemented in `postkamf`, and both were applied to obtain the improvement shown in Figure 13.

The first super-resolution processing option is applying the algorithm used in `parsek`[11] to derive a higher-resolution image by computationally sub-pixel-aligning and intelligently merging a set of one or more captured images. Although `parsek` was built to process pixel-shift images, it was discovered that natural random movement from manually triggering exposures on a typical tripod produces offsets between images of about the same magnitude as pixel shifting, and that is how the technique is leveraged with KAMF. This algorithm begins by sub-pixel-aligning all images supplied. It then computes a pixel value error PDF from the entire sequence of images. For each pixel location in the upscaled image, the value is created by the weighted combination of values from the nearest pixels in each image. The weighting not only uses distance, but also applies the PDF to discount pixel values that are inconsistent (e.g., those resulting from portions of the scene moving between captures). It also can take the CFA (color filter array) pattern into account rather than trusting interpolated color values in the JPEG from the camera. By default, the `parsek`-style upscaling results in a 1280×960-pixel image, but any upscaling can be specified – including no upscaling at all, simply enhancing image quality at the sensor's native 640×480 resolution. Combining multiple images can remove some of the artifacts introduced by overly aggressive sharpening applied in the camera. For Figure 13, eight captures were combined to produce a very clean image at 640×480 resolution.

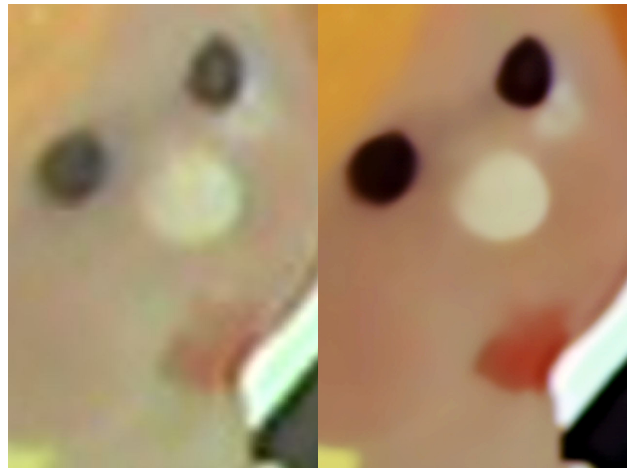

Figure 13. *640x960 crop of 7680x5760 JPEG vs. best using* `postkamf`

Of course, the image crop shown in Figure 13 is from a significantly upscaled image and upscaling from the clean 640×480 image was done using a trained-AI single-image super-resolution algorithm. The OpenCV super-resolution module provides a simple interface for using any of a variety of trained deep neural network (DNN) upscaling algorithms. Although one could train their own model for `postkamf` to use, four pre-trained models are widely available: EDSR[12], ESPCN[13], FSRCNN[14], and LapSRN[15]. There are versions of each of these models trained for specific enlargement factors. For example, the image crop in Figure 13 is the result of a 4× enlargement using EDSR, which is specified on the `postkamf` command line as `-d 4edsr`. Incidentally, the 7680×5760 resolution for the original JPEG is really 44MP despite being delivered by one of these cameras at the 48MP setting; some versions really do deliver 8000×6000-pixel JPEGs for 48MP.

Realistically, even starting with many captures and combining both `parsek`-style and DNN super-resolution processing methods, the effective resolution obtained using this camera is no more than a few megapixels. That is way below the resolution easily delivered by modern cell phones. It is enough resolution and sufficient tonal quality for images to be marginally competitive with typical images made using 135-format film.

Conclusion

The work presented here is extremely pragmatic, attempting to discover the implementation details of a family of low-cost commodity cameras designed for kids. From disassembly and detailed study of versions from 14 different vendors, it was possible to develop a good understanding of how these cameras work and how they can be adapted for more serious uses. Three different configurations are discussed:

1. The stock camera, optionally augmented by wired control
2. Kentucky's Adorable Micro Format
3. Kentucky's Approximation to Medium Format

The last two configurations are both referred to as KAMF, but involve different modifications and 3D-printed parts. The micro version of KAMF is particularly appealing in creating a tiny, cheap, stand-alone interchangeable-lens mirrorless camera.

However, especially in the KAMF versions, image quality from the camera is disappointing. The high resolutions claimed in advertising proved to at best describe the crudely upscaled JPEG image pixel counts, not sensel count nor amount of scene detail actually resolved. Noise, vignetting, and color casts also limit image quality. However, the open-source `postkamf` post-processing software created for these cameras is capable of rendering pleasing images with up to a few megapixels resolution.

The `postkamf` software, 3D models, and other materials about these cameras are available from `https://aggregate.org/DIT/KAMF`. We have tried to make it as easy as possible for others to build and use the KAMF cameras and `postkamf` software described in this paper.

Overall, the stock cameras and KAMF versions are easily hackable stand-alone platforms for various research and other uses. They cost far less than the obvious commercially-available alternatives or ones buildable from components. They are also very cute and handle surprisingly well. Unfortunately, the sensor resolution is fundamentally limiting, as is the fact that variations in the processing system in the camera make it impractical to directly alter the programming of the camera. In sum, these fill a niche somewhere between the ESP32-CAM modules[1] and use of Canon PowerShots via CHDK[16].

References

[1] Henry Dietz, Dillon Abney, Paul Eberhart, Nick Santini, William Davis, Elisabeth Wilson, and Michael McKenzie, "ESP32-CAM as a programmable camera research platform" in Electronic Imaging, 2022, pp 232-1 - 232-6, doi: 10.2352/EI.2022.34.7.ISS-232

[2] "Mimi – Micro Mirriorless Yashica – I'm Back," KickStarter, url: https://www.kickstarter.com/projects/samellos/micro-mirrorless-yashica-im-back, accessed 2/25/2025

[3] Henry Dietz, "Digital camera obscuras" in Electronic Imaging, 2023, pp. 334–1 - 334-11, doi: 10.2352/EI.2023.35.6.ISS-334.

[4] Henry Dietz, "About FF lens coverage," DPReview Medium Format Forum, url: https://www.dpreview.com/forums/post/64723138, accessed 2/25/2025

[5] Christian Schnalzger, "Interactive table with coverage of full frame lenses on GFX online," DPReview Medium Format Forum, url: https://www.dpreview.com/forums/post/67022982, accessed 2/25/2025

[6] "Center Filters," Alpa Product Group, https://www.alpa.swiss/product-group/center-filters, accessed 2/25/2025

[7] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000

[8] OpenMP Architecture Review Board, "OpenMP Application Programming Interface," Version 6.0, November 2024, https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-6-0.pdf

[9] Henry Dietz, "Construction, Quality Assessment, and Applications of Pixel Value Error PDF Models" in Electronic Imaging, 2025

[10] Henry Dietz, "An improved raw image enhancement algorithm using a statistical model for pixel value error" in Electronic Imaging, 2022, pp 151-1 - 151-6, doi: 10.2352/EI.2022.34.14.COIMG-151

[11] Henry Dietz, "Leveraging Pixel Value Certainty in Pixel-Shift and Other Multi-Shot Super-Resolution Processing " in Electronic Imaging, 2024, pp 142-1 - 142-7, doi: 10.2352/EI.2024.36.15.COIMG-142

[12] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, pp. 1132-1140, doi: 10.1109/CVPRW.2017.151

[13] W. Shi et al., "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1874-1883, doi: 10.1109/CVPR.2016.207

[14] C. Dong, C.C. Loy, and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network," in B. Leibe, J, Matas, N. Sebe, M. Welling (eds) Computer Vision, ECCV 2016. Lecture Notes in Computer Science, vol 9906. Springer, Cham. doi: 10.1007/978-3-319-46475-6_25

[15] W-S. Lai, J-B. Huang, N. Ahuja and M-H. Yang, "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5835-5843, doi: 10.1109/CVPR.2017.618.

[16] Canon Hack Development Kit Homepage, https://chdk.fandom.com/wiki/CHDK, accessed 2/25/2025