NIR-sighted: A Programmable Streaming Architecture for Low-Energy Human-Centric Vision Applications

JOHN MAMISH, Human-Computer Interaction, Georgia Institute of Technology, Atlanta, United States
RAWAN ALHARBI, Northwestern University, Evanston, United States
SOUGATA SEN, BITS Pilani - KK Birla Goa Campus, Zuarinagar, India
SHASHANK HOLLA, Georgia Institute of Technology, Atlanta, United States
PANCHAMI KAMATH, Georgia Institute of Technology, Atlanta, United States
YAMAN SANGAR, Georgia Institute of Technology, Atlanta, United States
NABIL ALSHURAFA, Northwestern University, Evanston, United States
JOSIAH HESTER, Georgia Institute of Technology, Atlanta, United States

Human studies often rely on wearable lifelogging cameras that capture videos of individuals and their surroundings to aid in visual confirmation or recollection of daily activities like eating, drinking, and smoking. However, this may include private or sensitive information that may cause some users to refrain from using such monitoring devices. Also, short battery lifetime and large form factors reduce applicability for long-term capture of human activity. Solving this triad of interconnected problems is challenging due to wearable embedded systems' energy, memory, and computing constraints. Inspired by this critical use case and the unique design problem, we developed NIR-sighted, an architecture for wearable video cameras that navigates this design space via three key ideas: (i) reduce storage and enhance privacy by discarding masked pixels and frames, (ii) enable programmers to generate effective masks with low computational overhead, and (iii) enable the use of small MCUs by moving masking and compression off-chip. Combined together in an end-to-end system, NIR-sighted's masking capabilities and off-chip compression hardware shrinks systems, stores less data, and enables programmer-defined obfuscation to yield privacy enhancement. The user's privacy is enhanced significantly as nowhere in the pipeline is any part of the image stored before it is obfuscated. We design a wearable camera called NIR-sightedCam based on this architecture; it is compact and can record IR and grayscale video at 16 and 20+ fps, respectively, for 26 hours nonstop (59 hours with IR disabled) at a fraction of comparable platforms power draw. NIR-sightedCam includes a low-power Field Programmable Gate Array that implements our mJPEG compress/obfuscate hardware, Blindspot. We additionally show the potential for privacy-enhancing function and clinical utility via an in-lab eating study, validated by a nutritionist.

CCS Concepts: • **Computing methodologies** → **Machine learning;**

Authors' Contact Information: John Mamish, Human-Computer Interaction, Georgia Institute of Technology, Atlanta, Georgia, United States; e-mail: john.mamish@gatech.edu; Rawan Alharbi, Northwestern University, Evanston, Illinois, United States; e-mail: rawanalharbi2016@u.northwestern.edu; Sougata Sen, BITS Pilani - KK Birla Goa Campus, Zuarinagar, Goa, India; e-mail: sougatas@goa.bits-pilani.ac.in; Shashank Holla, Georgia Institute of Technology, Atlanta, Georgia, United States; e-mail: sholla6@gatech.edu; Panchami Kamath, Georgia Institute of Technology, Atlanta, Georgia, United States; e-mail: panchamikamath@gatech.edu; Yaman Sangar, Georgia Institute of Technology, Atlanta, Georgia, United States; e-mail: ysangar3@gatech.edu; Nabil Alshurafa, Northwestern University, Evanston, Illinois, United States; e-mail: nabil@northwestern.edu; Josiah Hester, Georgia Institute of Technology, Atlanta, Georgia, United States; e-mail: josiah@gatech.edu.

ACM 1539-9087/2024/09-ART101
https://doi.org/10.1145/3672076

Additional Key Words and Phrases: Human-Centric Vision Applications

ACM Reference Format:
John Mamish, Rawan Alharbi, Sougata Sen, Shashank Holla, Panchami Kamath, Yaman Sangar, Nabil Alshurafa, and Josiah Hester. 2024. NIR-sighted: A Programmable Streaming Architecture for Low-Energy Human-Centric Vision Applications. *ACM Trans. Embedd. Comput. Syst.* 23, 6, Article 101 (September 2024), 26 pages. https://doi.org/10.1145/3672076

1 Introduction

Continuous vision analysis with wearable cameras provides a way to unobtrusively record and infer human behaviors with a high level of information and context, including moment-by-moment details of how the environment, person, and technology are connected [1–3]. This capability is helpful for researchers in mobile computing, health, and interaction. Researchers often validate wearable devices by deploying cameras alongside them for ground-truth collection, applying machine learning to the resulting video to recognize user behaviors. These include wearables that detect eating episodes [4–11], watch usage [3], breathing [12], fluid intake [13], human activity [14–16], and life-logging [17].

Despite wearable cameras becoming smaller and more capable, serious issues still prevent the realization of these exciting applications. Specifically, we see four system requirements that are not simultaneously satisfied: *compactness*, *system lifetime*, *system performance*, and *privacy*.

Compactness may be achieved in one of two ways: by reducing the size/number of electrical components or by limiting the battery size. For instance, although application-grade **System-on-Chips (SoCs)** with accompanying external DRAM (like in References [18, 19]) guarantee system performance, they demand large motherboards and consume significant power. Mass-market wearable cameras like GoPro [19], Google Clip [18], and the Narrative Clip [20] are compact and capable of recording high-resolution colour video at high frame rates. However, a few hours of recording time constitutes a "full day of recording." Increasing the recording time requires larger batteries like in the Axon Body 2 [21] it comes at the cost of increased system bulkiness.

System lifetime refers to how long a wearable can be used without interruption. For most applications, like clinical studies, health research, or life-logging, a whole-day system lifetime is necessary. Shorter lifetimes affect wearer behavior, limit adoption, and restrict studies in populations that may have difficulty managing the device themselves (like pediatric or geriatric populations). For a wearable camera to be widely usable in clinical settings, it needs to operate continuously for day-long wear. No commercial wearable camera that we know of satisfies this lifetime requirement.

Capturing images at a high frame rate and resolution is imperative to discern activity for clinically useful applications. *System performance* can be increased through a combination of larger non volatile memory and enhanced computation capability. Current market offerings offer impressive system performance but an unsatisfactory system lifetime and are often bulky.

Wearable cameras have been deployed in several studies demonstrating *privacy* concerns as a crucial reason individuals are unwilling to wear cameras in daily life even when incentivised [22–24]. Users feel a violation of privacy when their actions or surroundings are recorded [4]. A common way to address privacy concerns is by masking out parts of the image that are irrelevant to the study at hand, a process called *obfuscation*. While this does not perfectly preserve user privacy, participants report that they feel like their privacy is significantly enhanced [22, 25, 26]. Existing studies implementing privacy-enhancing obfuscation by offloading the video to a server before obfuscating it [18, 24, 27, 28]. Although functionally effective, this approach still exposes wearers by storing sensitive pixels until offline post-processing is complete. We believe that on-device

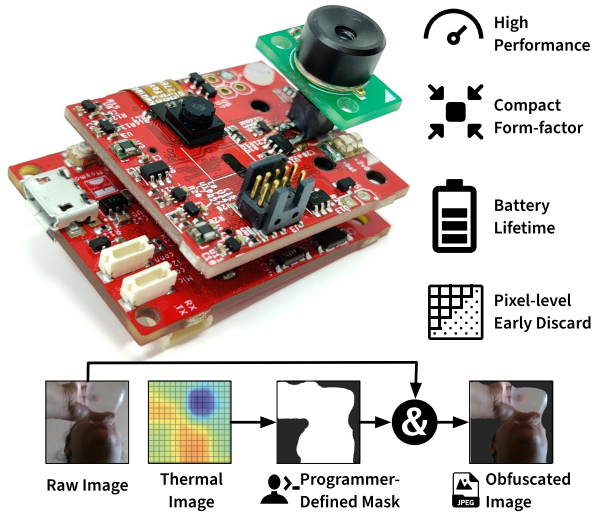

Fig. 1. The golf ball sized NIR-sightedCam supports programmable obfuscation for personal video collection in a small form factor with multi-day battery lifetime. Our approach leverages a novel architecture with a task-specific FPGA that combines selective-compression and obfuscation in a low-power envelope.

obfuscation has not yet been realized due to the computational intensity of generating masks from image sensor data.

These identified requirements are intertwined. For instance, *compactness* and *system lifetime* are both improved by reducing computational power and system memory, but this adversely impacts *system performance*. *Compactness* and *system lifetime* are at odds, because a larger battery or more non-volatile memory will extend *system lifetime* at the cost of reduced *compactness*. Finally, *privacy* affects *compactness* and *system lifetime* as well. Obfuscating images *before* they are stored to nonvolatile memory addresses this problem. However, determining which parts of the image to obfuscate requires more memory and computational performance, which is detrimental to *compactness* and *system lifetime*. No work has tackled this constellation of intertwined issues towards a long-term and high-utility wearable camera.

This article presents **NIR-sighted** (pronounced *Near*-sighted), an architecture for compact and low-power wearable video cameras. NIR-sighted enables *programmable early-discard* at a frame-level (like Reference [29]) and pixel-level granularity for continuous mobile vision. Early-discard is the notion of only storing those portions of a video stream that are relevant to the application and discarding the rest before it reaches the **microcontroller (MCU)**. With NIR-sighted, early-discard is enabled by image *masks* that are generated on the fly from sensors in a programmatic way. Masked portions are discarded as the video streams. NIR-sighted's early-discard capabilities can be used to implement on-device obfuscation, which has demonstrated utility for *privacy-enhancement of images* [22–24], and for robust attention mechanisms for human machine vision [30, 31]. They can extend *system lifetime* by recording less and giving programmers a more fine-grained ability to control data rate and image streams via sensor signals. Furthermore, NIR-sighted allows for the use of small and low-power MCUs without sacrificing resolution or frame rate. Our architectural innovations enable *system performance* and *privacy* while maintaining *compactness* and *system lifetime*.

We also present *NIR-sightedCam*, a camera that implements this architecture. As shown in Figure 1, NIR-sightedCam is a neck-worn, egocentric camera that uses a thermal sensor to enable

pixel-level obfuscation of the video stream *on-the-fly* and *fully on-device*. Enabled by NIR-sighted's architectural innovations, NIR-sightedCam has a high frame rate, compact form-factor, multi-day lifetime, and privacy-enhancing, programmer-definable video obfuscation. NIR-sighted is enabled by two key ideas as follows:

1. *Use another sensor to help with masking*. Generating masks directly from high-resolution image sensor data requires significant memory and computational power, which negatively impacts system *compactness* and *lifetime*. Instead, NIR-sighted's obfuscation masks are generated using a different sensor than the primary image sensor, like a low-resolution IR imager [32, 33] or depth camera [29]. Application-specific and *program-defined masks* can be crafted with this data as input. For example, an eating study using a neck-worn egocentric camera can mask out everything except for a wearer's face. A study focused on user surroundings can do the exact opposite, discarding all pixels belonging to the user's face before saving video to memory. Whatever the study goal, we posit that a definition of early-discard can be embedded in a binary, per-frame two-dimensional mask and that this mask can often be programmatically generated from non-visual-spectrum cameras. This programmatic mask generation capability enables NIR-sighted to provide application-specific flexibility to obfuscate any portion of the video without having to store the obfuscated portion at any time. Currently, no devices in the market offer this feature.

2. *Never buffer the whole image, even though you've got to compress it.* Compression is a necessity for storing video data (24 hours of uncompressed 15 fps 320 × 240 grayscale video will fill *99.5 gigabytes*). Compressing in software at high frame rates is computationally intractable for small microcontrollers [29, 34]. Commercially available MCUs with hardware JPEG codecs require the full image to be buffered in memory and do not allow any type of non-MCU transformation of the image beyond compression. Even for low-resolution imagers, this immediately puts memory requirements into the 100s of kB, ruling out the most compact MCUs. Furthermore, buffering prevents the use of imagers with a resolution above 640 × 480 without using external DRAM. In fact, we found that obfuscating already-compressed JPEG images is almost as computationally expensive as compressing them in software, as demonstrated by recent work capturing images at less than 1 Hz using software JPEG compression [35].

NIR-sighted solves this issue by moving video compression off-chip to a bespoke, tunable motion JPEG (mJPEG) compressor called *Blindspot*. Blindspot is small (implemented on a 5,280-LUT iCE40UP5K **Field Programmable Gate Array (FPGA)** [36]), low-power (takes 5 mW to compress 320 × 240 images at 30 fps), and requires little memory, even for high-resolution video (it never buffers more than 16 lines of the source image). This enables systems to obfuscate and compress high-resolution video streams even with very small and low-power microcontrollers having only a few kB of RAM. Crucially different from other commercially available hardware JPEG compressors [37, 38], Blindspot's design takes as input a binary mask that is applied to the image in-situ as compression occurs.

Contributions: In this article, we expand the notion of early-discard past the removal of entire frames to include per-pixel masking *within* frames. To this end, we make three primary contributions:

(1) We introduce NIR-sighted, a wearable camera architecture that leverages the notion of *streaming mask programming* to enable *early-discard* [29] at a pixel level. We demonstrate that these masks can be generated cheaply and dramatically reduce nonvolatile memory requirements. Early-discard of pixels can enable obfuscation, enhance privacy, speed up inference, and save storage space and bandwidth.

(2) We instantiate the system into a tool for researchers: a multi-spectrum (infrared, visual), compact, wearable prototype, NIR-sightedCam. NIR-sightedCam has 320 × 240 resolution at

Table 1. Activity Monitoring or Life-logging Applications in the Literature That Used Cameras and Manual Labeling for Validation

	Video	*Lifetime*	*Wearable*	*Private*
Life-Logger or Groundtruth Collector				
FluidMeter [13]	✗	✔	✔	✗
Thomaz et al. [11]	✗	✔	✔	✗
BodyScope [39]	✗	✔	✔	✗
Ng et al. [2]	✗	✔	✔	✔
Auracle [7]	✔	✗	✔	✗
Earbit [6]	✔	✗	✔	✗
Pizza et al. [3]	✔	✔	✗	✗
SenseCam [27]	✗	✔	✔	✗
Narrative Clip [18]	✗	✔	✔	✗
Glimpse [29]	✔	✔	✔	✗
ZenCam [40]	✔	✗	✔	✗
Privacy Enhancing Video Platforms				
Zhang et al. [41]	✔	✗	✗	✔
Pinto [42]	✔	✗	✗	✔
TrustEYE.M4 [43]	✔	✗	✗	✔
CMUcam3 [44]	✔	✗	✔	✔
This Work	✔	✔	✔	✔

We denote the paper and extrapolate based on study descriptions: we claim that "wearable" means small enough to hang around the neck or easily attach to the body without extra baggage, if any privacy enhancement function exists, then we denote "private."

30 fps, on-board obfuscation, and an all-day battery lifetime, useful in a number of applications including healthcare, privacy, AR, and human activity recognition via auto-generated attention mechanism.

(3) To enable NIR-sightedCam, we introduce Blindspot, a hardware mJPEG compressor that obfuscates images before compressing them. Blindspot is low-power and has a small hardware footprint even when compressing large images.

With NIR-sighted, we aim to equip the health, behavior science, and mobile computing communities with a wearable platform that researchers can field for long-duration experiments on a diversity of human subjects across varying settings and applications. NIR-sightedCam aims to provide an alternative to commercial or scratch built systems that are not sufficiently energy-efficient or privacy preserving [6–8].

2 Background and Motivation

NIR-sighted arose from clinical health researchers' need to capture complex human behaviors in free-living situations. Since the early 2010s, researchers have relied on egocentric cameras to capture complex human activities (Table 1 shows a representative selection). However, as we will explain in this section, these cameras have deficiencies that make them less than ideal.

2.1 Satisfying Competing System Requirements

Satisfying the above-mentioned constellation of requirements (compactness, system lifetime, performance, privacy) has proven challenging. For instance, in a law-enforcement body camera as in Reference [21] compactness is sacrificed for improved system lifetime, but this trade-off is unacceptable for a consumer product that is expected to be portable. Below we identify design trade-offs between these four requirements in the context of clinical research applications.

System complexity increases with frame rate, resolution. Increasing the frame rate and resolution drastically increases storage requirements, making data storage infeasible without using prohibitively large nonvolatile memories. To this end, on-board video compression is needed for wearable cameras as system performance increases.

Systems using motion picture encoding like H.264 require sophisticated hardware and DRAM for larger memory capacity to store data structures and multiple frames of video. Even if the video is coded as a sequence of lossily compressed still images, using compression to enable increased video quality still increases system complexity. Commercially available microcontrollers integrating a JPEG encoder [37, 38] require frames to be fully double-buffered in memory for compression or streaming video to occur. They require significant on-chip SRAM adversely affecting system size and power consumption making mJPEG compression above VGA resolution impossible with commercially available microcontrollers.

Discarding pixels increases system complexity. Early-discard of individual pixels and entire frames [23, 25] can enhance privacy, reduce video storage requirements, and remove irrelevant details. However, existing methods for early-discard increase system complexity. We believe that this is due to two main reasons:

(1) Figuring out what parts of the image to obfuscate is typically done by a **Deep Neural Network (DNN)** [45, 46]. Evaluating a DNN at 20 to 30 fps takes significant memory and computing power; typically, external DRAM will be needed.
(2) If an existing architecture is used, then obfuscating the image before it is compressed requires it to be double-buffered in memory. This increased demand on memory inevitably leads to a bulkier system with low battery life.

Some systems [29] successfully use low-power sensors to discard entire frames of video, but a large amount of irrelevant and privacy-violating information remains.

2.2 Limitations in State-of-the-Art Personal Mobile Vision Systems

Current camera systems are not suitable for wearable vision. They are not simultaneously compact, long-lasting, or performant enough, and they cannot discard sensitive information. Figure 2 shows how different architectures conduct *compression*, to save storage, and *obfuscation* to enhance privacy or speedup offline inference, and their effects on performance, cost, and power draw. Typical blocks in the processing pipeline consist of an imager, compression, decompression, processing (i.e., face recognition), and storage/communication.

Current video collection methods fall into roughly two categories: high-performance multimedia processors and low-powered triggering-based video loggers as shown in Figure 2(a) and Figure 2(b), respectively.

High-performance multimedia processors shown in Figure 2(a) are in high-end smartphones [47], GoPros [19], and wearables like the Ray-Ban Meta smart glasses [48], which achieve high compression performance and may conduct extensive AI/ML operations on images. Like the older TI OMAP 4430 [49, 50] in Google Glass or the more recent Qualcomm Snapdragon QCS605 [51], these processors are expensive, dissipate significant heat, and draw >1 W when capturing and

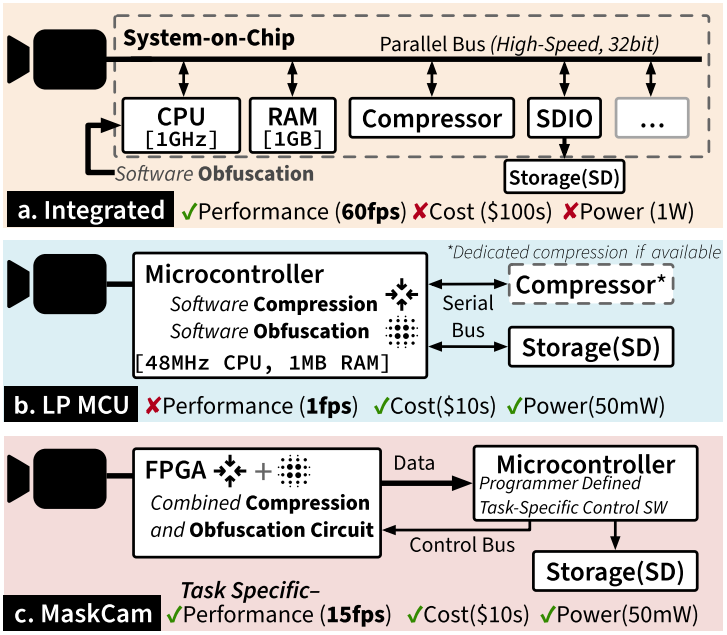

Fig. 2. (a) high-performance and highly integrated systems found in high-end phones and GoPros [19]. (b) Low-end, low-power, cheap microcontroller-based systems exemplified by Glimpse [29] and SenseCam [27]. (c) Where dedicated, low-power obfuscation+compression hardware sits *between* imager and compute.

processing video. These state-of-the-art SoCs have extensive resources, including external memory over 1 GB, clock speeds exceeding 1 GHz, and numerous on-chip accelerators but at the expense of bulk, thermal comfort, battery life, cost, and design complexity.

As shown in Figure 2(b), researchers have explored low-power MCU-based imaging devices for specific applications like life-logging [27, 52] and outdoor environmental/habitat monitoring [53, 54]. Exemplified by the Narrative Clip 1 [20] (broken down by notable maker Limor Fried in Reference [52]), these systems sacrifice performance but are compact and achieve 24+ hour battery lifetime, often operating at 1 fps or less.

Recent work has attempted to bridge this gap, recording high-quality video in short bursts to save power. Memento [55] is emotion based, using EEG signals; and ZenCam [40] triggers with an IMU-based activity classifier. However, each method discards images that may have been useful for ground truth. Worse, as the complexity of the triggering scheme grows, so does the system's bulk and power consumption. Glimpse [29] is a Mobile vision system designed to perform computational cloud offloading by doing crude image pre-processing on their hardware. It uses low-power sensors (motion, light) for triggering framewise discard of the image stream. This architecture is thus computationally heavy and at the same time the overall power saving in this system is limited to the use case. The framewise discard may result in the elimination of critical information in the frame. Thus the challenges that need to be addressed are presented in the following.

C1: Reducing Storage Requirements. To enable all-day video, memory budgets must be kept low: systems using DRAM require large board space; furthermore, data movement to and from memory contributes significantly to energy costs [29].

C2: Compression. Motion video compression requires storage of multiple raw video frames in memory, compromising bulkiness and system lifetime. The smallest commodity MCUs lack

hardware JPEG compressors and the SRAM to support mJPEG. Therefore, to realize mJPEG obfuscation and compression on systems with the smallest available MCUs, new architectures are needed.

C3: Obfuscation. Making it possible to remove parts of an image on-platform is beneficial for several reasons: Not only does it enable privacy-enhancing obfuscation [25, 30], it also allows us to record less data, extending the lifetime of systems that are bound by power or storage capacity.

3 System Design

NIR-sighted is an architecture for wearable cameras. NIR-sighted-based cameras integrate a microcontroller, an FPGA or ASIC implementing a low-memory-footprint obfuscation-aware mJPEG compressor, a CMOS imager, and a low-resolution non-visual-spectrum imager (like an infrared array). By adding a low-resource, obfuscation-aware compressor and non-visible-spectrum imager, wearable cameras following NIR-sighted's architecture require dramatically less memory and computation resources than privacy-preserving cameras integrating only a CMOS imager and a commodity SoC. This reduced memory and compute burden paves the way for smaller, less obtrusive, and easier-to-deploy wearable cameras while still preserving privacy.

3.1 Privacy Enhancement through Early Discard of Frames and Pixels

In a human-centered study with day-long recording, billions of irrelevant pixels are collected that not only take up unnecessary space but also contain private information that is not relevant to the study. The NIR-sighted architecture enables systems to discard parts of frames and entire frames containing low-utility pixels.

Determining which pixels to remove is challenging. Definitions of "pixel utility" vary widely across system deployments. Consider these studies with different aims: a user study evaluating a gesture detection wearable would only need to capture the wearer and could obfuscate the scene (like in a video call); however, in a life-logging setting, blurring/masking people (including the wearer) but cataloging the environment and places visited, might be sufficient. The same pixels that have high utility in one scenario may have low utility in a different scenario.

The concept of privacy is amorphous [56, 57]; it changes based on location and context, personal values, and surrounding technology. One individual may be comfortable with a wearable that captures their surroundings as long as it obscures their face; another might find full-video recording acceptable but only when he or she is smoking a cigarette. For one platform to be useful across studies with varying definitions of pixel utility and participants with varying notions of privacy, a high level of flexibility in discarding pixels and frames is needed.

NIR-sighted allows for the discarding of specific pixels within a frame through *masking*. A mask is a low-resolution, binarized image where "false" values denote pixels that should be obfuscated (either blurring or zeroing out the pixels) and "true" values denote blocks of pixels to store. Masks are generated by code on the MCU and sent to Blindspot, where pixels are discarded before image compression occurs. The MCU gets the obfuscated and compressed images; it only ever sees the pixels it asks for. NIR-sighted accommodates discarding full frames to save battery life and storage time. Just as with pixel-level discard, the system can use the secondary imager.

NIR-sighted targets human-centered studies, so we adopt secondary non-visible-light imagers (e.g., infrared or depth), which are inherently sensitive to human wearers allowing for the generation of human-centered masks. Figure 4 explains this process of modification and movement of an image through the pipeline. Masks are generated with simple, computationally lightweight algorithms running on the MCU. Changing the mask involves writing a new mask generation algorithm (which, as discussed, is made easy by sensor choice) and flashing it to the MCU, something that's made easy by widely available open source programming tools. This enables

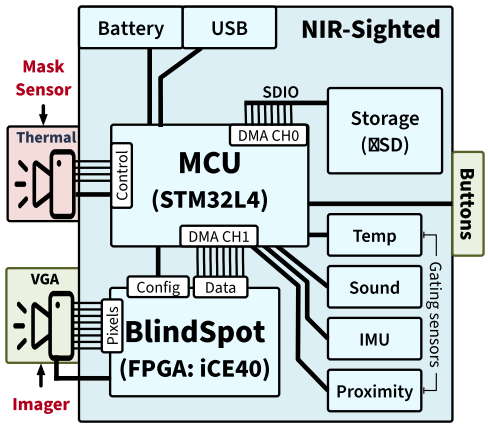

Fig. 3. The hardware architecture of the system. The MCU calculates the obfuscation mask from thermal imaging that is sent to Blindspot. Blindspot reads images from a camera and then obfuscates and compresses them.

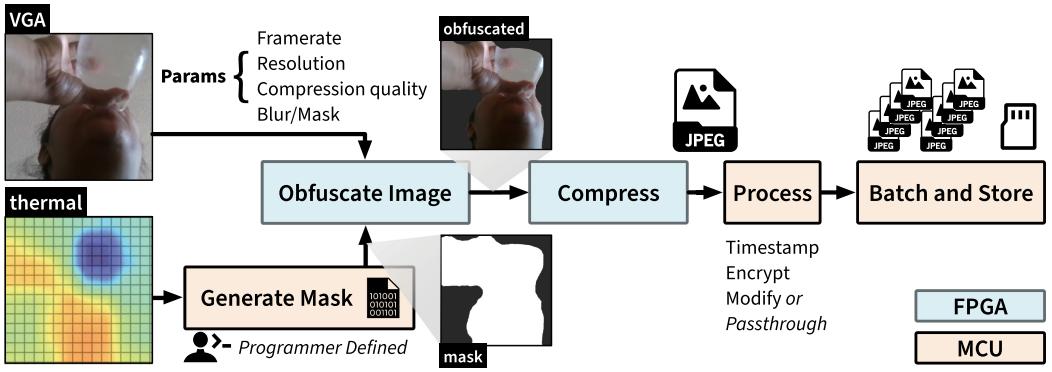

Fig. 4. NIR-sighted's on-device processing pipeline. Each RGB frame is obfuscated using a program-generated mask and then compressed. The pipeline is split across the FPGA and MCU to allow flexibility and high performance. NIR-sighted can create privacy-enhanced video on device; private data never make it to storage.

a programmable definition of pixel utility (and therefore privacy), bringing programmer-defined masking to compact, long-lifetime wearable cameras.

Masking is the most powerful tool that NIR-sighted offers for discarding low-utility pixels, but the programmer also has other tools at their disposal: (1) frame-level discard, where entire frames can be discarded, dynamically changing frame rate; (2) adjusting resolution; and (3) compression aggressiveness. By modulating these in response to sensor data, further improvements to wearer privacy and system lifetime can be realized.

3.2 NIR-sighted System Architecture

We present an overview of NIR-sighted's architecture in Figure 3. By using a non-visible-spectrum imager as an information source for generating privacy masks and by performing compression in a hardware IP block that requires minimal memory, NIR-sighted enables the design of extremely compact wearable cameras.

NIR-sighted-based systems involve cooperation between a microcontroller and our compression IP block, called Blindspot: The microcontroller reads spatial data from a non-visible-spectrum imager (like an IR camera or depth camera). The onboard firmware on the microcontroller uses this data to generate a *privacy mask*. This mask is sent to Blindspot, which reads the camera data from the system's CMOS visible-light image sensor and then obfuscates and compresses the read video. This obfuscated and compressed video is sent back to the microcontroller for optionally adding final post-processing on the data like adding timestamps, auxiliary sensor data (e.g., IMU, visible light, etc.), and/or encryption. Once the microcontroller has completed these optional tasks, the final obfuscated and compressed video stream is stored in onboard flash memory for later retrieval by clinical researchers. We elaborate upon the component-wise flow of data through this architecture below.

Secondary Non-Visible-Spectrum Imager. Enhancing wearer privacy for the body-worn applications that NIR-sighted targets involves identifying which pixels of the video stream are occupied by humans (i.e., the wearer themselves or bystanders) and creating an obfuscation mask from this information. The most straightforward way of identifying humans in the video stream is to operate directly on the video stream itself; however, known methods for doing this incur massive memory and computation costs. These costs limit how far privacy-preserving wearable cameras can be miniaturized. To overcome this issue, we turn to low-resolution non-visible-spectrum imagers. Our work focuses primarily on thermal infrared imagers, but NIR-sighted's insights apply to other similar imagers (e.g., ToF depth cameras) as well. Although these imagers consume more power per pixel than visible-spectrum CMOS imagers, much less compute and memory are required to extract human-sensitive obfuscation masks from their data streams.

Mask Generation. With the thermal scene information on the MCU, a binarized mask is generated by executing a programmer-defined function that determines which pixels to keep and which ones to remove. This way, the generated mask embeds a privacy definition specified by the programmer. Mask generation can range from speedy threshold-based setting methods, to region of interest identification, to more intensive machine learning-based approaches such as FastGRNN [58]. Because masks generated from secondary imagers using computationally efficient methods are typically low-resolution, each binary "pixel" in the mask corresponds to an 8×8 block of pixels in the video stream.

The resulting mask is sent to Blindspot via an implementation-defined on-system control interface. The programmer defines the frequency of mask updates; this will typically be limited by the frame rate of the secondary imager. Because masks are binary, only 1,200 bits are needed to transfer each mask, and therefore the control interface will only require 10s of kb/s of bandwidth.

Raw Video Stream Read, Obfuscate, and Compress. Concurrent to the mask creation and loading process, Blindspot reads image data from the camera and compresses each frame according to the jpeg specification [59]. Before an 8×8 block of pixels is coded, Blindspot checks the loaded mask if it is to be obfuscated, in which case its corresponding DCT coefficients are left at 0, rendering that part of the image as a gray box.

Video Stream Sent to MCU and Stored. After the FPGA obfuscates and compresses the video stream, it is sent to the MCU. At this point, the MCU might perform some post-processing operations on the video stream like timestamping, encryption, or association with auxiliary sensors. Once this has been done, the video stream can be sent to flash memory for storage and later retrieval by the clinical team for use in the intended application, study, or research evaluation. This step requires very little computational effort from the microcontroller; in a commodity MCU,

almost everything in this data transfer step can be handled by DMA, allowing for use of a smaller, lower-power MCU.

3.3 Combining Obfuscation and Compression in Blindspot

While systems like Glimpse [29], ZenCam [40], and others discard entire frames to save energy, NIR-sighted discards individual pixels within privacy-sensitive frames. We focus on redesigning the *compression* pipeline to enable obfuscation without requiring images to be buffered. NIR-sighted was motivated by constraints that we encountered while designing NIR-sightedCam; video had to be compressed, but commodity hardware only offered three ways to do it: software (which is far too slow [54, 60]), compression hardware integrated into a commodity MCU (which is not available on most MCUs and also requires images to be double-buffered [37, 38]), or compression hardware integrated into the image sensor itself [61] (which makes on-camera obfuscation impossible without de-compressing and re-compressing the image). For the reasons mentioned, we felt that none of these options were appropriate for the system we were trying to design.

Instead, we move compression off-chip to Blindspot, our dedicated hardware mJPEG encoder. This frees up implementations of NIR-sighted to use low-performance commodity MCUs. Although JPEG compression hardware is certainly not a new idea [62, 63], techniques described in the literature are prohibitively difficult for system designers to integrate unless they have the budget and expertise to design a custom ASIC.

By introducing Blindspot, an *open source* design that has minimal memory requirements, we significantly expand the low-power design space, even for teams unable to tape out their own chips. Blindspot has reduced memory requirements when compared to commodity hardware-JPEG-enabled MCUs, because it obfuscates the image in a streaming fashion: Obfuscation and compression are achieved without ever storing more than 16 lines of the image at once. This is important when scaling video resolution: double-buffering a raw HD image with 16-bit color depth requires over 8 MB of space, far exceeding the available SRAM of all low-cost commodity microcontrollers, and certainly exceeding the available SRAM of the most compact options. By dramatically reducing the power draw and memory requirements of compression, we can reduce battery size without giving up multi-day lifetime and reduce hardware size without giving up on high frame rates.

3.4 How NIR-sighted Shrinks Systems Designed with COTS Components

Miniaturizing a system like this would be relatively straightforward for a group with extensive expertise and a large budget: They could simply combine system functions into an ASIC whose area is optimized for this specific application. However, this avenue is not open to small research groups without the money, manpower, or extensive know-how and connections required to fabricate an ASIC. As we established before, using COTS components in a non-NIR-sighted-style design would result in a bulky and power-hungry circuit. By using a secondary non-visible-spectrum imager to generate privacy masks, we avoid the need for DRAM and high-performance processors that would be needed to generate privacy masks directly from video data. Furthermore, Blindspot frees up designers to use extremely tiny and low-performance MCUs. Setting aside the issue of mask generation (which is addressed by the non-visible-spectrum imager), they no longer need to worry about whether the MCU will have the hardware and SRAM buffer space needed to perform JPEG compression. It is through these two merits that NIR-sighted allows researchers to build smaller and lower-power cameras that still preserve privacy without turning to prohibitively difficult methods. In fact, we found that the power needed to generate a mask from a low-resolution non-visible-light imager was less than the power needed to generate a mask from a high-resolution CMOS sensor, even when the higher power consumption of the non-visible-light imager was taken into account.

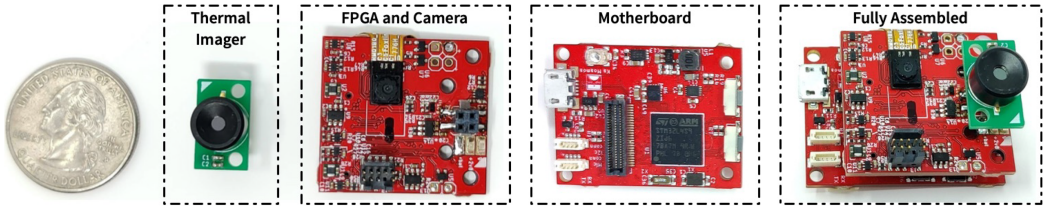

Fig. 5. To reduce costs and sourcing difficulties and enable easier debugging, and reconfigurability, NIR-sighted is composed of three separate PCBs that share an interconnect via low-profile stackable headers.

3.5 Blindspot: A Memory-constrained, Obfuscation-aware Hardware JPEG Compressor

In this section, we present Blindspot, a hardware mJPEG compressor. Blindspot is *open source* and was designed from the ground up specifically for size-constrained, privacy-preserving systems. To make sure that Blindspot could be used by teams without the ability to tape out IC's, we implemented it to fit in a compact and affordable iCE40UP5K FPGA [36]. Blindspot achieves this small size and low-power operation through three primary techniques:

— A streaming architecture: In contrast to hardware JPEG compressors integrated into off-the-shelf MCUs, Blindspot never buffers more than 16 lines of an image. This allows for the use of very little SRAM, letting us use small FPGAs. This has further benefits when scaling up camera sizes. Blindspot's memory usage scales with $O(\sqrt{N})$ in the number of pixels, meaning that larger imagers can be used without incurring massive SRAM costs.
— Parallelization of DSP operations: The central operation of JPEG compression (the DCT) is calculated by many small cores in parallel, shrinking hardware size.
— Reduction of division precision: Quantization—the critical step in JPEG where data loss actually takes place—relies on notoriously expensive division hardware. Instead of using full-precision integer division (which we found would occupy half of our FPGA), we allow for division by numbers of the form $k2^q$ for $k \in [0, 2^l]$. This lets us substitute a 16×8 bit divider for an l-bit divider and a q-bit barrel shifter.

While hardware JPEG compressors are certainly not new, we believe that Blindspot occupies a unique point in design space. Because of its low memory footprint, Blindspot stands on its own as a useful piece of hardware to design ultra-compact cameras, even without the inclusion of a mask sensor as prescribed by NIR-sighted. Furthermore, Blindspot upends the notion that transform coding is not possible in the lowest-powered systems [54].

4 NIR-sightedCam Implementation

In collaboration with human behavior researchers, we designed and fabricated NIR-sightedCam, a camera that meets the *compactness*, *system lifetime*, *performance*, and *privacy* needs demanded by clinical applications demand by implementing the NIR-sighted architecture described in Section 3. The hardware implemented is shown in Figure 5 and the FPGA firmware architecture is described in Figure 6. The hardware is designed to be modular, comprising a thermal imager, a camera board, and motherboard. In the following sections, we detail implementation decisions balancing the architecture design, application, and energy requirements.

4.1 Hardware Design

Figure 5 shows the components making up the hardware platform, which is described below. The system consists of boards vertically stacked via mezzanine connectors; the decision to stack boards

vertically was made for two primary reasons: form factor and modularity. Vertical stacking improves form-factor, because it reduces the footprint of the hardware. While laying all of the components out on a single PCB might improve overall camera volume, verbal feedback from clinicians indicated that such a design would sit uncomfortably on the body because of its shape. Second, we were interested in a modular design, where some parts of the hardware could be upgraded (for example by swapping out the non-visible-spectrum imager or replacing the camera board with a higher resolution one) without having to completely redesign the board.

Motherboard. The motherboard is the central controller, which hosts an ST Microelectronics STM32L4S9ZI microcontroller, which is an Arm Cortex-M4 running at 120 MHz, with 2 MB of Flash memory and 640 KB of SRAM onboard [64]. The motherboard includes an SD card, an IMU, and compact connectors for the addition of arbitrary i2c sensors if needed. The motherboard connects to the camera board via a stackable connector that contains an i2c control bus for the FPGA and camera, a separate i2c bus for the non-visible-spectrum imager, and an 8-bit wide parallel data bus for receiving compressed video from the FPGA. The i2c control connection is sufficient bandwidth for control signals; as mentioned in Section 3, streaming obfuscation masks to the FPGA only requires 10s of kb/s, only a few percentages of the i2c bus's bandwidth. The board also includes battery charge and management circuits, user buttons and programming ports. While implementing these mask-generation and system management functions inside the FPGA itself could improve system integration, we found that it made sense for NIR-sightedCam to implement these in an MCU instead. Implementing them in-FPGA would require a larger, more expensive FPGA and would be less efficient for the tasks in question. Furthermore, flexibility and researcher usability are important for this platform; changing a mask generation algorithm implemented in hardware would be more difficult (and implementing a soft-core on the FPGA would likely be too inefficient for the reasons mentioned above). By splitting these responsibilities between an FPGA and MCU, we can use smallest-in-class chips for both.

FPGA and Camera Board. The vision board contains a Lattice iCE40 UP5K FPGA) and a Himax HM01B0. The iCE40 is an affordable, ultra-low-power FPGA that is suitable for compact, low-power applications. The Himax HM01B0 image sensor is able to capture 30 QVGA resolution (320 $\times$ 240 pixels) frames per second while only consuming 1 mW of power.

Thermal Imager. A mid-resolution thermal imager is a good way to identify humans in a scene in a way that is robust to light/dark cycles and other environmental effects of images and depth sensors. This imager is used to create masks to hide private features of images. We use the MLX90640, which has a 110° $\times$ 75° field of view with a temperature measurement range of −40°C to 85°C and a resolution of 32 $\times$ 24 pixels.

4.2 Software

NIR-sighted is a complete platform in the sense that it also comes with supporting software and firmware for managing each stage of the image processing pipeline on the MCU. The on-device pipeline is shown in Figure 4. The implementation details of each piece of software is described below.

Operating system. On the MCU, separate threads are responsible for reading the thermal imager, extracting masks from thermal images, transferring data from the FPGA, and processing, batching, and storing privacy-enhanced images to the SD card. We use FreeRTOS to manage these multiple threads and to save power when the MCU core is asleep. We also make extensive use of the MCU's DMA features to so that <1% of CPU time is dedicated to coordinating data movement.

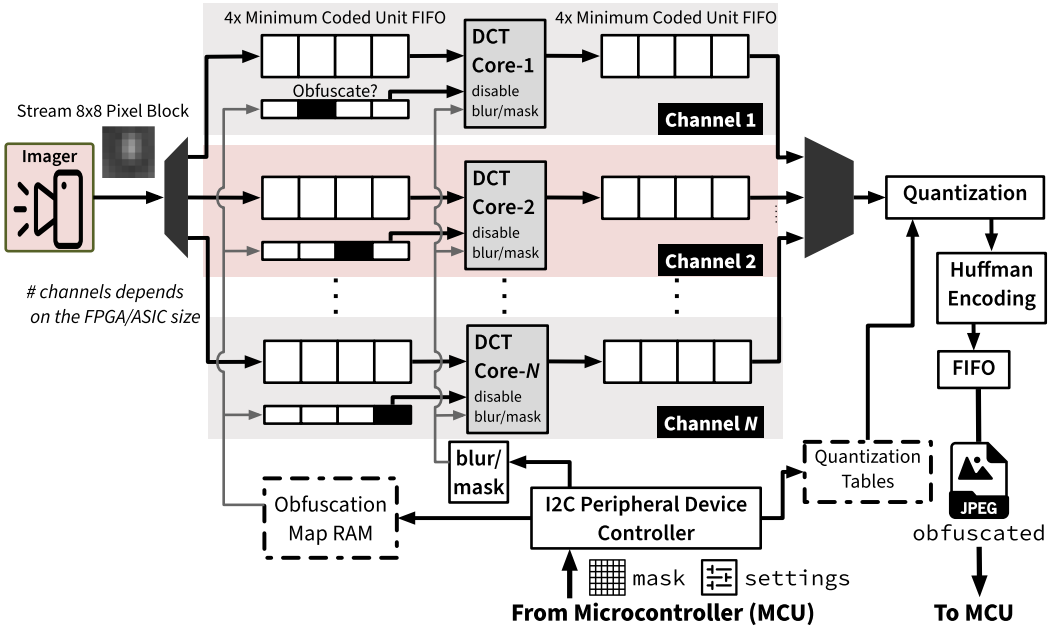

Fig. 6. Implementation of Blindspot. Streaming blocks of pixels are put into parallel channels that are obfuscated on the fly (masked completely or blurred) by the DCT core followed by JPEG specified compression routines.

CMOS and Thermal Imager Co-Calibration. In order to apply masks generated from the thermal imager's data to video data from the CMOS sensor, we must know how their fields of view overlap. To perform this alignment, a one-time, design time registration is needed. While an exact process using focal length equations and projection drawing from the exact placement of the imagers on the PCB would be best, we found that a manual alignment process was sufficient for our low-volume prototype.

4.3 Hardware JPEG Compressor Design

The FPGA comprises the selective compression and obfuscation circuit that takes in a mask provided by the MCU and outputs a privacy enhanced, obfuscated JPEG image back to the MCU. The design of the circuit embedded in the FPGA is shown in Figure 6.

The FPGA is a modified circuit level implementation of the JPEG image compression algorithm: Pixels are consumed via a parallel port CMOS image sensor interface before they are processed by parallel DCT cores (comprising a micro-coded multiplier and adder), a quantizer, and a Huffman encoder before they are buffered in an output FIFO and transmitted over a parallel port. Blindspot allows for the obfuscation of 8×8 blocks of pixels by greying out or blurring them. There is also an interface to the MCU over I2C, through which the obfuscation mask and configuration settings (i.e., quality-table updates) are captured and stored in embedded FPGA SRAM.

JPEG Pipeline. As shown in Figure 6, first the image is processed in 8×8 pixel blocks, called Minimum Coded Units; these are fed into DCT cores running in parallel, each of which is responsible for processing its own stream of Minimum Coded Units. Each DCT core has a FIFO at its input and output for buffering. To achieve obfuscation, the DCT cores can be gated. This means that when processing Minimum Coded Units that are to be obfuscated, they will output

coefficients representing a blurred or greyed-out Minimum Coded Unit. Once the DCT operation is complete, results from all of the parallel DCT cores are interleaved for the quantization step. The quantization step relies on pre-set quantization tables, and will quantize high frequency components of the images. These components are less obvious to the human eye, producing long runs of easy to encode low entropy data. This quantized stream is fed into a Huffman encoder; the fixed codebook used by the Huffman encoder prioritizes the most common symbols, giving them shorter codewords. Because of the quantization step, some symbols are much more likely to appear than others, making Huffman coding highly effective.

To enable pixel level discard, the DCT core uses the obfuscation map stored in the RAM as a mask (more on this mask generation is described down below). As each pixel block is processed, it is blurred or masked if the corresponding bit in the mask is 1.

Blur levels To obfuscate a block of pixels, Blindspot can either fully mask or blur the corresponding block. We implement blurring pixel blocks in a JPEG-friendly way by throwing away high frequency coefficients when doing JPEG compression. This is the same as aggressively reducing the quality for that pixel block, quantizing away all DCT coefficients except the DC component.

Scaling and Resolution. The current FPGA has five channels/DCT cores and runs at 12 MHz. The DCT cores and input/output buffering comprise the majority of the chip area. The only thing that grows linearly with image width is the size of the buffers just before and after the DCT cores. The number of cores is a factor of image width, since the base pixel block size is 8×8. We observe that five cores at 12 MHz is sufficient to achieve a frame rate of 30 frames per second if all other I/O operations proceed at speed. Frame rate can be increased by increasing the number cores or clock frequency. In our implementation of Blindspot, each DCT core takes 915 clock cycles to do an 8×8 transform. With 1,200 transforms per image, the DCT cores require 18.3 ms to process 320×240 image when running at 12 MHz.

5 Evaluation

In this section we characterize the overhead associated with NIR-sighted's proposed mechanisms and evaluate NIR-sightedCam's performance. Specifically, we (1) measure Blindspot's power consumption *in situ* when implemented in an FPGA and in simulation when synthesized as an ASIC in modern flows, (2) explore the CPU and memory requirements of some effective masking algorithms, (3) describe NIR-sightedCam's performance and power consumption, and (4) use NIR-sightedCam to record video in demonstration use cases, submitting the resulting video to a dietitian for commentary.

We find that NIR-sightedCam's "obfuscate and compress" architecture, enabled by Blindspot, reduces power by an order of magnitude over A-type systems (Integrated) (Figure 2(a)), increases performance and frame rate over B-type systems (low-power MCU) (Figure 2(b)), and still enables obfuscation.

5.1 NIR-sightedCam Cost, Size, and Weight

NIR-sightedCam's design choices were made with careful consideration of system cost. Table 2 shows a system cost breakdown by component when ordering systems at qty 100. No one component dominates the cost of the system; the most expensive component—the thermal imager—only accounts for 21% of system cost, the FPGA and its associated flash memory are less than 10% of system cost.

The entire prototype, when fully assembled without batteries, weighs only 13 grams. When two 500-mAh battery packs are added, the prototype has a 26-hour battery life (49 hours with its IR imager disabled). As Figure 5 shows, it is the size of a golf ball and slightly lighter, at 35 grams

Table 2. Cost of System Components at qty 100

Component	Cost	% of sys cost
Complete System	$165.11	100%
Thermal Imager	$ 34.50	20.9%
FPGA	$ 15.65	9.5%
MCU	$ 19.64	11.9%

Table 3. SWaP and Performance Comparison for Different Systems

System	Size (XYZ, camera points at Z)	Volume	Weight	System Runtime*	Video Quality	Obfuscates?
SenseCam [27, 66]	6×8×3 cm	144 cm^3	175 g	12.0 hr	RGB×640×480 0.2 fps	✗
Narrative Clip 2 [20]	3.6×3.6×1.1 cm	14.3 cm^3	20 g	1.3 hr**	RGB1920×1080 30 fps	✗
GoPro Hero 10 [19]	7.2×5.1×3.4 cm	125 cm^3	153 g	2.0 hr	RGB×1920×1080 30 fps	✗
Axon Body 2 [21]	8.7×7.0×2.6 cm	158 cm^3	142 g	>12 hr	RGB×1920×1080 30 fps	✗
NIR-sightedCam	3.7×3.3×2.9 cm***	35.4 cm^3	35 g	23.4 hr	L×320×240 30 fps	✔
Optimized NIR-sighted-based system****	< 2.0×2.0×2.0 cm	<8 cm^3	< 35 g	> 20 hr	L×320×240 30 fps	✔

*This denotes how long the system can run for without charging or offloading data. If there are different configurations (e.g., the GoPro Hero 10 can record in different resolutions and frame rates), we give the value for the configuration with the highest runtime and specify the corresponding image quality.
**Narrative Clip 2's battery life may be longer, but it only has memory space for 1.3 hr of video so cannot be used for longer lifelogging stretches.
***Size without case.
****This represents a worst-case estimate for an ideal system designed using the NIR-sighted architecture and the smallest available and compatible COTS components.

(a golf ball is ~45 grams and a diameter of ~43 mm). This combination of size and battery life is ideal for long-term clinical studies.

As discussed in Section 3, a fundamental motivation of NIR-sighted was to make NIR-sightedCam very compact while still using COTS components. Although relying on a custom IC would have allowed for a significantly smaller platform, it would have largely defeated the purpose of this project by making it inaccessible for small research groups.

In Table 3, we compare NIR-sightedCam's SWaP (size, weight, and power) and performance with related research efforts and commercial products. Although NIR-sightedCam has lower video resolution than similar commercially available products, it can obfuscate video and has better size, weight, and power characteristics.

As discussed in Section 3.4, NIR-sighted and Blindspot allow wearable camera designers to use the smallest available COTS components to design their systems. We designed our NIR-sighted-based system, NIR-sightedCam, to be a research platform, so we sacrificed system size for flexibility. The final column of Table 3 represents estimates for a NIR-sighted-based system that is fully optimized for size. The size and weight estimates are based off of specific ultra-small COTS MCUs [65] and FPGAs [36] whose use is only made possible through the NIR-sighted architecture.

5.2 Characterizing Blindspot's Performance

To characterize Blindspot's performance, we designed a custom instrumented board for the Lattice iCE40 FPGA used in NIR-sightedCam (More details are in the appendix). The breakout board is outfitted with onboard trans-impedance amplifiers to measure the current consumption of the iCE40 FPGA. To remove variability inherent in natural images coming from the onboard camera, we supplied a static test image over the board's GPIO pins. The FPGA's core consumed 5.67 mW when compressing images at 12.5 fps. Increasing the mask's infill percentage does not cause the FPGA's

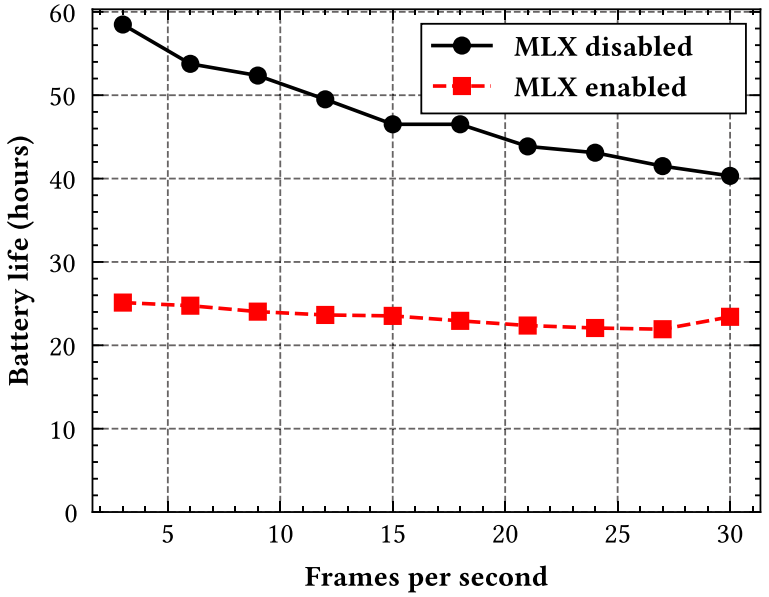

Fig. 7. NIR-sightedCam battery life as a function of frame rate assuming a 1.000-mAh battery. To produce this graph, we used an Otii Arc to measure NIR-sightedCam's current consumption at 3.7 V across a variety of frame rates.

power consumption to increase. In fact, we found that adding the obfuscation mask *decreases* the power consumption; as mask infill increases from 0% to 100%, power consumption drops by 4.1%, likely due to decreased FPGA activity factor.

With a 12-MHz system clock, Blindspot can achieve a maximum frame rate of 57 fps when compressing 320 × 240 video. Implemented in an iCE40 FPGA with the Yosys open tool chain [67], Blindspot has a max clock rate of 15.8 MHz, accommodating a maximum frame rate 78 fps at an image resolution of 320 × 240.

To compare Blindspot to software solutions, we implemented a baseline JPEG compressor in C on an ARM Cortex-M4 MCU at 96 MHz (Ambiq Apollo 3). The microcontroller consumed around 15 mW, but frame rate was only slightly higher than 2 fps. This is faster than in Reference [54], which can achieve 1 fps in software, but slower than in Reference [60] (an open source JPEG library), which can achieve 7 fps in software on the same microcontroller.

We synthesized Blindspot using an industry-standard Synopsys design flow for a Nangate 45-nm process. Simulation results estimated a static and dynamic power consumption of 260 and 35 µW, respectively. We estimate that a version of the system using an ASIC would have a current consumption of 14–15 mA and a battery life of 64–66 hours with masking disabled. Note that this is not a major gain over the non-ASIC version, because—with the MLX enabled—power consumption is dominated by the microcontroller and flash memory. Selecting a lower performance microcontroller and different flash could alleviate this issue. The integrated version is significantly lower power, and estimated footprint, even if scaled for HD video, consumes less than 1.0 mm^2 of die space, proving its suitability for compact, low-cost vision systems with high performance.

5.3 Masking Program Performance

The ability to cheaply generate effective mask programs from non-visible-spectrum imagers (mask sensors) is central to NIR-sighted. To determine whether this could be feasibly done, we

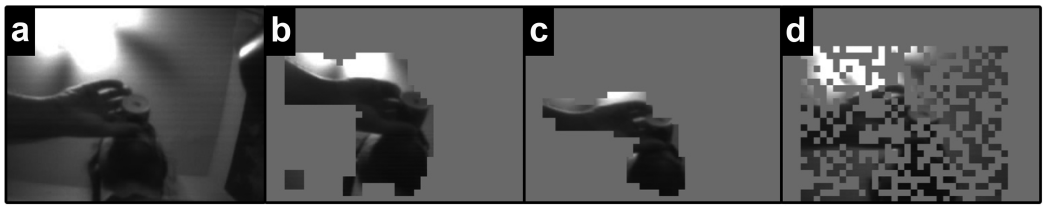

Fig. 8. Video frames with different masks; (a) none, (b) threshold (med.), (c) flood fill, and (d) random.

Table 4. Mask Algorithm Benchmarks

Mask Algorithm	CPU %	Exec. (ms)	Mem. (B)
Constant. (No MLX)	N/A	0.1	N/A
None. (MLX Calc)	1.4	0.80	3,856
Threshold (med.)	6.2	4.1	7,680
Threshold (avg.)	4.3	2.8	4,608
Wearer Flood Fill	6.7	4.6	7,680
Random	1.7	0.2	1,552

implemented three thermal-based masking algorithms, running them on NIR-sightedCam's CPU at 72 MHz and using a 32 × 24 mlx90640 IR array for sensor data. The mask algorithms we implemented are (shown in Figure 8): (i) *threshold (med.)*: pixels far from the median temperature are kept (capturing the wearer, nearby bystanders, and hot/cold objects); (ii) *threshold (avg.):* pixels far from the *average* temperature are kept; (iii) *wearer flood fill*: a heat-based flood fill is used to keep; and (iv) *random noise*: a tester mask, where points are randomly selected to be masked.

These masks can further be modified with little overhead through morphological erosion/dilation and inversion. These masks can be changed dynamically via the program based on gating sensors or other program logic. We found that mask generation incurs very little CPU and memory overhead, even on a low-performance microcontroller; the results of which is recorded in Table 4. We also collected five different unmasked sequences of ego-centric video, about 10 minutes each, and applied different masks offline to test the effect on video size. We found that masking—even when video is still compressed with baseline JPEG—reduces file size by *50–70%* (Figure 9). We shared the resulting clips with a dietitian trained in performing diet recalls, who indicated that the video would be useful even with masking enabled.

5.4 NIR-sightedCam System Lifetime

We characterized the performance of our full NIR-sightedCam prototype by recording masked 15 fps video and using an Otii Arc to measure power consumption. The measured average current draw of NIR-sightedCam at 3.7 V is *36–39 mA*, indicating a *26-hour battery life*. An additional battery pack would keep the size and weight of that of a golf ball, but increase the lifetime by another 7–15 hours. Regardless, users can get all-day or even multi-day battery lifetime depending on usage in a day and sleep cycle. Removing the thermal imager and recording unobfuscated video doubles the battery life (a current consumption of *19–20 mA* gives a *49-hour battery life*), and reducing frame rate further extends battery life.

Figure 7 shows how system battery life changes with changing frame rate. With the MLX disabled, the frame rate has a significant impact on battery life; however, when enabled, the MLX dominates power consumption so reducing the frame rate does not significantly improve battery life.

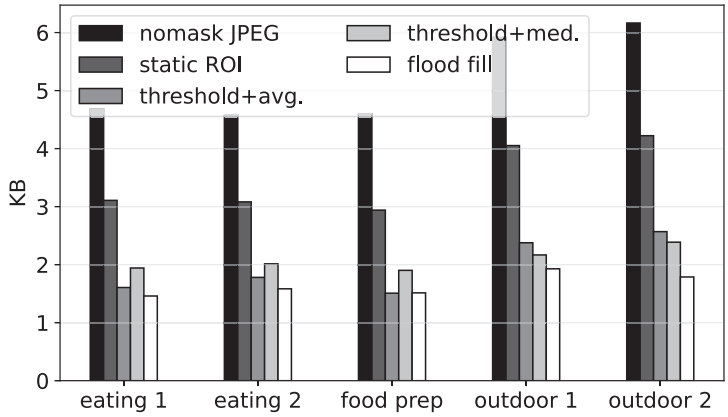

Fig. 9. Five different video scenes, all about 10 minutes long, were recorded and masks were added offline. Masking reduces video frame size by up to 70% across scenes.

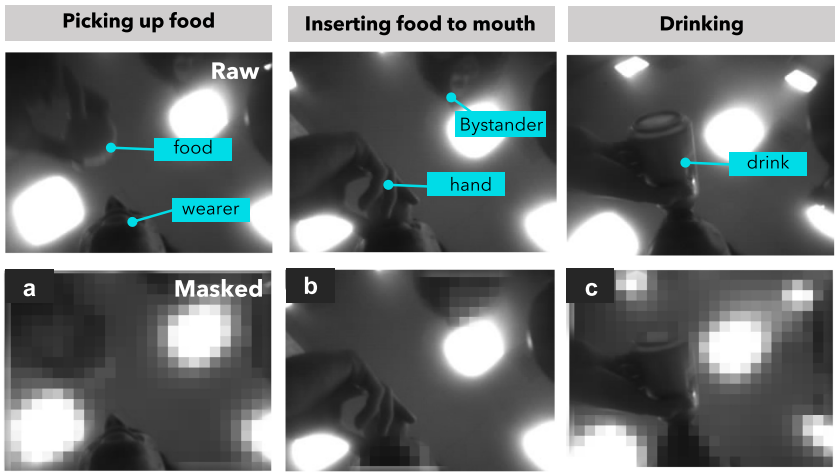

Fig. 10. Captures (using grayscale imager) and masks for eating and drinking, shown to dietitian (PhD, RD).

5.5 Demonstration Use Case: Eating Studies

In this section, we conduct a real-world case study by applying NIR-sighted for eating behavior detection. Shown in Figure 10, we have a camera roll of raw and masked images of eating and drinking. We have implemented multiple types of masks to exemplify the flexibility that NIR-sighted offers in image obfuscation, such as masking everything but the wearer (a), masking all human faces including the wearer (c), and masking everything but the wearer's hands (d). We asked a male volunteer to perform a sample of everyday meal-time activity of interest to the health research community (i.e., eating and drinking [39, 68–73]).

Tracking eating behaviors is essential in understanding and managing many health conditions such as diabetes, cancer, and obesity. However, manual tracking of eating habits suffers from recall bias and is reported to be burdensome. Although wearers of the camera might see the benefit in automated tracking, they are often concerned about the non-consented bystanders' privacy. Therefore, a masking/obfuscation method similar to what we use in Figure 10(a) and (b) has been shown

to reduce privacy concerns related to the bystander [22, 24], which can relieve discomfort caused by the wearable camera and increase wear time. A professional dietitian trained in performing diet recalls evaluated the camera roll of obfuscated eating and drinking. The dietitian reported:

> ...in conjunction with 24-hour recall [used] to identify food item[s], [this technology] will help determine meal timing—no need for reliance on self-reported fasting. Obscuring the participants face may lead to increased likelihood for the camera to remain on....

Dietitians rely on self-reported diet assessments without confirmation from objective measures. NIR-sighted can enhance the dietitians' ability to validate timing of food intake, understand the diet quality and energy intake of the patient, remove the need for deletion of recordings, reduce the burden on the wearer, and improve the design of a medical nutrition therapy plan that will account for the patient's naturally occurring behavior.

6 Related Work

NIR-sighted is the first end-to-end platform that enables programmable pixel-level early discard for mobile vision. In this section, we discuss related work beyond the architectural insights and devices mentioned in Section 2.

Mobile Vision Acceleration. Heterogeneous CPU-FPGA systems like NIR-sighted mitigate the challenges posed by the end of Moore's Law, providing task-specific acceleration at low energy and cost point [74, 75]. Everyday vision applications [18, 27, 76, 77] represent a useful target for this type of acceleration. High power draw, however, has always been a challenge [78, 79] for mobile cameras. To conserve energy, research considers various approaches. SenseCam and ZenCam used environmental cues such as change in light, temperature, or audio to automatically trigger image capture [27, 40]. Glimpse discards "uninteresting" frames to conserve power and similar to NIR-sighted provides a programmable way to decide what is not of interest [29]. Others have modified image resolution [80], frame rate [81, 82], and the processing pipeline [83]. Unlike with Glimpse, NIR-sighted discards uninteresting *parts* of a frame, determined by a programmer defined mask. NIR-sighted does not sacrifice latency for computational and image processing actions, but instead uses a streaming architecture and a low-power, low-cost, and commodity MCU to accomplish all actions.

Programmable Obfuscation: Surveillance cameras have explored types of obfuscation, with application in privacy. TrustEye and TrustCam [84–86] separate layers of the program to bar access to raw pixels. Another used thermal cameras to guide obfuscation [87]. These are all non-wearable systems.

Bystander Privacy in Video: Discomfort is expressed by the wearer [4, 88] and by bystanders [89, 90] when wearing a camera. Researchers have developed systems where bystanders can opt out from recording [91], while others have proposed computer vision techniques where specific objects in a photo are obfuscated [92]. Some techniques apply degradation in the image as a whole [24], while others have utilized activity-oriented wearable cameras to focus on capturing a specific activity of interest [4]. We enable on-device exploration of all these different privacy schemes, stemming from our obfuscation capability.

Work done in Reference [31] demonstrates that using mask obfuscation on images results in an average reduction of only 2% in human activity recognition accuracy. They highlight that despite the obfuscation, the fine-grained activities can still be accurately detected, which underscores the potential of obfuscation techniques in addressing privacy concerns. The low reduction in accuracy due to mask obfuscation indicates that the technique can provide a reasonable trade-off between

privacy and utility. NIR-sighted allows one to build upon an on-device activity classification system that can perform well when using masked images.

7 Discussion and Future Work

We built NIR-sighted to fill a hole in mobile vision systems for privacy-enhanced life-logging cameras. This article presents an extensible architecture and prototype device for capturing all-day video with high energy-efficiency and performance. The article further explores how programmers can define obfuscation for a specific application (like privacy). We discuss future work and periphery issues around NIR-sighted.

Extensions. The current platform could be extended to use alternative masking sensors (i.e., depth or time-of-flight instead of thermal, or even motion based) as well as more advanced masking algorithms. Compression schemes (MPEG-4) and tuning might bear performance improvements. Finally, higher resolution cameras could enable finer capture detail. Because of the modular design, the motherboard could be used and only the vision board redesigned. NIR-sighted in its current state provides a useful jumping off point for these explorations.

Another way to improve the battery life is by doing an activity detected recording. As discussed in the related work section, Reference [31] proves that activity detection can be done on obfuscated images too. Using a machine learning model that is trained on these obfuscated images can help in triggering continuous capture only if a certain activity is detected.

Privacy and Regulatory: Significant bodies of work explore privacy, hardware and software for enhancing privacy, and how new systems (like NIR-sighted) might fit into emerging laws like COPPA and GDPR that prohibit the collection of private data, especially of vulnerable children (COPPA). NIR-sighted provides a first step for in-hardware privacy enhancement via obfuscation masks, deeper exploration of privacy via NIR-sighted would have merit.

Community Platform. NIR-sighted seeks to empower mobile computing, health and behavioral researchers to gather rich video data in free-living settings for studies, validation, and novel applications. We view the NIR-sighted architecture as a useful approach for obfuscation, and one that can be used immediately, via NIR-sightedCam, with commodity components like low-power FPGAs, for immediate deployment, without relying on the whims of the SoC and MCU market. NIR-sighted can be used by usable privacy research to understand *in situ* considerations of bystander privacy, as opposed to existing methods using post processing or MTurk online studies [93]. Because of the sophistication of the platform and tight design requirements due to the small size, we are exploring Tindie, Macrofab, and Seeed for at-cost distribution of NIR-sighted devices to researchers. We anticipate immediate future work centered around documentation, tutorials, new masking algorithms, and engaging in community building and outreach.

8 Conclusion

We began this project as a response to a community of mobile health researchers pushing for better ways to capture strong visual ground-truth information to validate health-based wearables. While exploring this space we found an unoccupied niche, where commodity components available did not provide energy efficient ways for pixel-level early discard while giving multi-day battery life and high frame rate. We developed a new selective compression architecture, NIR-sighted, that we instantiated in an FPGA design, Blindspot, and into a prototype camera, NIR-sightedCam, the size of a golf ball, enabling enhanced image capture at 19 fps or more. Programmers create a mask program to make use of the architecture, which generates a two-dimensional mask based on a mask sensor (thermal in our case), instructing which pixels to discard frame by frame. Our evaluation showed that: masking reduced image size up to *70%*, enabling ultra long deployments with limited

storage; Blindspot enabled a *6× faster frame rate* over a state-of-the art MCU, at only a third of the power, and was able to run at 78 fps maximum; the ASIC synthesis of Blindspot has only 35 µW of dynamic power consumption; and finally our NIR-sightedCam prototype had a *49-hour battery lifetime* (26 with masking), recording at 15 fps continuously in a golf ball size form factor, and was made entirely of commercially available parts.

Appendix

We designed a custom PCB for the Lattice iCE40 FPGA used in NIR-sightedCam to analyze the performance of the FPGA and aid in performing experiments to characterize the performance of Blindspot's performance as shown in Figure 11. Our custom development board is instrumented with current-sensing amplifiers to measure current consumed by the FPGA.

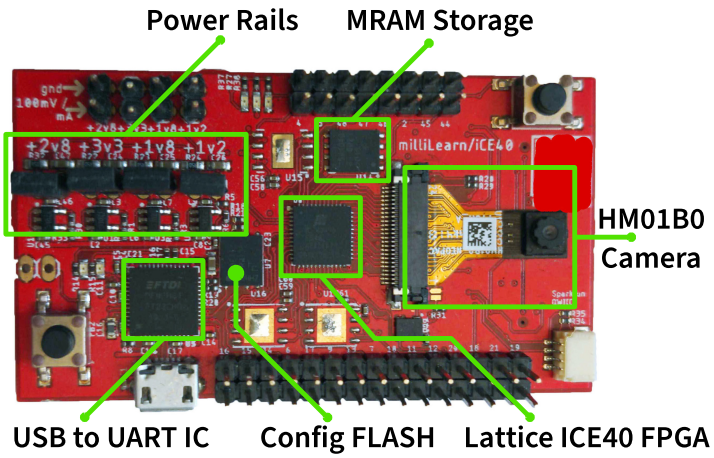

Fig. 11. Custom development and measurement board for the FPGA vision architecture bench-marking.

References

[1] Jacqueline Chen, Simon J. Marshall, Lu Wang, Suneeta Godbole, Amanda Legge, Aiden Doherty, Paul Kelly, Melody Oliver, Ruth Patterson, Charlie Foster, et al. 2013. Using the SenseCam as an objective tool for evaluating eating patterns. In *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference*. ACM, 34–41.

[2] Kher Hui Ng, Victoria Shipp, Richard Mortier, Steve Benford, Martin Flintham, and Tom Rodden. 2015. Understanding food consumption lifecycles using wearable cameras. *Pers. Ubiq. Comput.* 19, 7 (2015), 1183–1195.

[3] Stefania Pizza, Barry Brown, Donald McMillan, and Airi Lampinen. 2016. Smartwatch in vivo. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 5456–5469.

[4] Rawan Alharbi, Tammy Stump, Nilofar Vafaie, Angela Pfammatter, Bonnie Spring, and Nabil Alshurafa. 2018. I can't be myself: Effects of wearable cameras on the capture of authentic behavior in the wild. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 2, 3 (2018), 90.

[5] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *Proceedings of the 7th International Conference on Networked Sensing Systems (INSS'10)*. IEEE, 233–240.

[6] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: Using wearable sensors to detect eating episodes in unconstrained environments. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 1, 3 (2017), 37.

[7] Shengjie Bi, Tao Wang, Nicole Tobias, Josephine Nordrum, Shang Wang, George Halvorsen, Sougata Sen, Ronald Peterson, Kofi Odame, Kelly Caine, et al. 2018. Auracle: Detecting eating episodes with an ear-mounted sensor. *Proc. ACM Interact. Mob. Wear. Ubiq. Techno.* 2, 3 (2018), 92.

[8] Shibo Zhang, Rawan Alharbi, Matthew Nicholson, and Nabil Alshurafa. 2017. When generalized eating detection machine learning models fail in the field. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and the ACM International Symposium on Wearable Computers*. ACM, 613–622.

[9] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. 2014. Detecting periods of eating during free-living by tracking wrist motion. *IEEE J. Biomed. Health Inf.* 18, 4 (2014), 1253–1260.

[10] Yicheng Bai, Wenyan Jia, Zhi-Hong Mao, and Mingui Sun. 2014. Automatic eating detection using a proximity sensor. In *Proceedings of the 40th Annual Northeast Bioengineering Conference (NEBEC'14)*. IEEE, 1–2.

[11] Edison Thomaz, Irfan Essa, and Gregory D. Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1029–1040.

[12] Stephanie Balters, Elizabeth L. Murnane, James A. Landay, and Pablo E. Paredes. 2018. Breath booster!: Exploring in-car, fast-paced breathing interventions to enhance driver arousal state. In *Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare*. ACM, 128–137.

[13] Takashi Hamatani, Moustafa Elhamshary, Akira Uchiyama, and Teruo Higashino. 2018. FluidMeter: Gauging the human daily fluid intake using smartwatches. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 2, 3, Article 113 (Sep. 2018), 25 pages. DOI: http://dx.doi.org/10.1145/3264923

[14] Franklin Mingzhe Li, Di Laura Chen, Mingming Fan, and Khai N. Truong. 2019. FMT: A wearable camera-based object tracking memory aid for older adults. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 3, 3, Article 95 (Sep. 2019), 25 pages. DOI: http://dx.doi.org/10.1145/3351253

[15] David H. Nguyen, Gabriela Marcu, Gillian R. Hayes, Khai N. Truong, James Scott, Marc Langheinrich, and Christof Roduner. 2009. Encountering SenseCam: Personal recording technologies in everyday life. In *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM, 165–174.

[16] Wenxiao Zhang, Bo Han, and Pan Hui. 2018. Jaguar: Low latency mobile augmented reality with flexible tracking. In *Proceedings of the 26th ACM International Conference on Multimedia*. 355–363.

[17] Marion Koelle, Wilko Heuten, and Susanne Boll. 2017. Are you hiding it? usage habits of lifelogging camera wearers. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–8.

[18] Narrative Clip. 2019. The World's Most Wearable HD Video Camera - Narrative Clip 2. Retrieved from http://getnarrative.com

[19] 2021. Go Pro Hero 10. Retrieved from https://gopro.com/en/us/shop/cameras/hero10-black/CHDHX-101-master.html

[20] 2012. Narrative Clip 1. Retrieved from http://web.archive.org/web/20160302193636/http://getnarrative.com/narrative-clip-1/

[21] Axon. *Axon Body 2 Camera User Manual*. Axon. Retrieved from https://my.axon.com/sfc/servlet.shepherd/document/download/069f3000006Ko6BAAS

[22] Rawan Alharbi, Mariam Tolba, Lucia C. Petito, Josiah Hester, and Nabil Alshurafa. 2019. To mask or not to mask?: Balancing privacy with visual confirmation utility in activity-oriented wearable cameras. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 3, 3 (2019), 72:1–72:29. DOI: http://dx.doi.org/10.1145/3351230

[23] Yifang Li, Nishant Vishwamitra, Bart P. Knijnenburg, Hongxin Hu, and Kelly Caine. 2017. Effectiveness and users' experience of obfuscation as a privacy-enhancing technology for sharing photos. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 67 (Dec. 2017), 24 pages. DOI: http://dx.doi.org/10.1145/3134702

[24] Mariella Dimiccoli, Juan Marín, and Edison Thomaz. 2018. Mitigating bystander privacy concerns in egocentric activity recognition with deep learning and intentional image degradation. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 1, 4 (2018), 132.

[25] Yifang Li and Kelly Caine. 2022. Obfuscation remedies harms arising from content flagging of photos. In *CHI Conference on Human Factors in Computing Systems*. 1–25.

[26] Rakibul Hasan, Yifang Li, Eman Hassan, Kelly Caine, David J. Crandall, Roberto Hoyle, and Apu Kapadia. 2019. Can privacy be satisfying? on improving viewer satisfaction for privacy-enhanced photos using aesthetic transforms. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–13.

[27] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. 2006. SenseCam: A retrospective memory aid. In *Proceedings of the International Conference of Ubiquitous Computing (UbiComp'06)*.

[28] Zihao W. Wang, Vibhav Vineet, Francesco Pittaluga, Sudipta N. Sinha, Oliver Cossairt, and Sing Bing Kang. 2019. Privacy-preserving action recognition using coded aperture videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[29] Saman Naderiparizi, Pengyu Zhang, Matthai Philipose, Bodhi Priyantha, Deepak Ganesan, and Jie Liu. 2017. Glimpse: A programmable early-discard camera architecture for continuous mobile vision. In *Proceedings of the ACM*

International Conference on Mobile Systems, Applications, and Services (MobiSys'17). 292–305. DOI: http://dx.doi.org/10.1145/3081333.3081347

[30] Rawan Alharbi, Sougata Sen, Ada Ng, Nabil Alshurafa, and Josiah Hester. 2022. ActiSight: Wearer foreground extraction using a practical rgb-thermal wearable. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'22)*. IEEE, 237–246.

[31] Soroush Shahi, Rawan Alharbi, Yang Gao, Sougata Sen, Aggelos K. Katsaggelos, Josiah Hester, and Nabil Alshurafa. 2022. Impacts of image obfuscation on fine-grained activity recognition in egocentric video. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops'22)*. IEEE, 341–346.

[32] 2022. MLX90640: Far infrared thermal sensor array (32x24 RES). Retrieved from https://www.melexis.com/en/product/MLX90640/Far-Infrared-Thermal-Sensor-Array

[33] 2022. Panasonic Grid-EYE Infrared Array Sensors. Retrieved from https://na.industrial.panasonic.com/products/sensors/sensors-automotive-industrial-applications/lineup/grid-eye-infrared-array-sensor

[34] G. Suseela and Y. Asnath Victy Phamila. 2018. Energy efficient image coding techniques for low power sensor nodes: A review. *Ain Shams Eng. J.* 9, 4 (2018), 2961–2972.

[35] Harsh Desai, Matteo Nardello, Davide Brunelli, and Brandon Lucia. 2022. Camaroptera: A long-range image sensor with local inference for remote sensing applications. *ACM Trans. Embed. Comput. Syst.* (2022).

[36] Lattice Semiconductor 2021. iCE40 UltraPlus Family Data Sheet. Retrieved from https://www.latticesemi.com/-/media/LatticeSemi/Documents/DataSheets/iCE/FPGA-DS-02008-2-0-iCE40-UltraPlus-Family-Data-Sheet.ashx?document_id=51968 FPGA-DS-02008-2.0

[37] Renesas 2020. Renesas RA6M3 Group. Retrieved from https://www.renesas.com/us/en/document/mah/ra6m3-microcontroller-group-users-manual?r=1054166

[38] STMicroelectronics 2018. RM0410 Reference Manual. Retrieved from https://www.st.com/resource/en/reference_manual/dm00224583-stm32f76xxx-and-stm32f77xxx-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

[39] Koji Yatani and Khai N. Truong. 2012. BodyScope: A wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 341–350.

[40] Shiwei Fang, Ketan Mayer-Patel, and Shahriar Nirjon. 2019. ZenCam: Context-driven control of autonomous body cameras. In *Proceedings of the 15th International Conference on Distributed Computing in Sensor Systems (DCOSS'19)*. IEEE, 41–48.

[41] Y. Zhang, Y. Lu, H. Nagahara, and R. Taniguchi. 2014. Anonymous camera for privacy protection. In *Proceedings of the 22nd International Conference on Pattern Recognition*. 4170–4175. DOI: http://dx.doi.org/10.1109/ICPR.2014.715

[42] Hyunwoo Yu, Jaemin Lim, Kiyeon Kim, and Suk-Bok Lee. 2018. Pinto: Enabling video privacy for commodity IoT cameras. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*. ACM, New York, NY, 1089–1101. DOI: http://dx.doi.org/10.1145/3243734.3243830

[43] Thomas Winkler and Bernhard Rinner. 2014. TrustEYE. M4–A novel platform for secure visual sensor network applications. In *Proceedings of the International Conference on Distributed Smart Cameras*. ACM, 45.

[44] Anthony G. Rowe, Adam Goode, Dhiraj Goel, and Illah Nourbakhsh. 2007. CMUcam3: An open programmable embedded vision sensor.

[45] Wuyang Zhang, Zhezhi He, Luyang Liu, Zhenhua Jia, Yunxin Liu, Marco Gruteser, Dipankar Raychaudhuri, and Yanyong Zhang. 2021. Elf: Accelerate high-resolution mobile deep vision with content-aware parallel offloading. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 201–214.

[46] Kittipat Apicharttrisorn, Xukan Ran, Jiasi Chen, Srikanth V. Krishnamurthy, and Amit K. Roy-Chowdhury. 2019. Frugal following: Power thrifty object detection and tracking for mobile augmented reality. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 96–109.

[47] 2023. Pixel Phone Hardware Tech Specs. Retrieved from https://support.google.com/pixelphone/answer/7158570

[48] 2023. Qualcomm and Meta Are Expanding Your Reality: Here's how. Retrieved from https://www.qualcomm.com/news/onq/2023/10/qualcomm-and-meta-are-expanding-your-reality-heres-how

[49] David Witt. 2009. OMAP4430 architecture and development. In *Proceedings of the IEEE Hot Chips 21 Symposium (HCS'09)*. IEEE, 1–16.

[50] Texas Instruments. 2010. OMAP4430 Multimedia Device Silicon Revision 2.x. Retrieved June 4, 2021 from https://www.ti.com/lit/ug/swpu231ap/swpu231ap.pdf

[51] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. 2018. AI benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV'18) Workshops*. 0–0.

[52] Limor Fried. 2014. Narrative Clip Teardown. Retrieved from https://www.youtube.com/watch?v=SN4YHfpH6aU

[53] Colleen Josephson, Lei Yang, Pengyu Zhang, and Sachin Katti. 2019. Wireless computer vision using commodity radios. In *Proceedings of the 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'19)*. IEEE, 229–240.

[54] Matteo Nardello, Harsh Desai, Davide Brunelli, and Brandon Lucia. 2019. Camaroptera: A batteryless long-range remote visual sensing system. In *Proceedings of the 7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. 8–14.
[55] Shiqi Jiang, Zhenjiang Li, Pengfei Zhou, and Mo Li. 2019. Memento: An emotion-driven lifelogging system with wearables. *ACM Trans. Sen. Netw.* 15, 1, Article 8 (Jan. 2019), 23 pages. DOI: http://dx.doi.org/10.1145/3281630
[56] Helen Nissenbaum. 2011. A contextual approach to privacy online. *Daedalus* 140, 4 (2011), 32–48.
[57] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.
[58] Aditya Kusupati, Manish Singh, Kush Bhatia, Ashish Kumar, Prateek Jain, and Manik Varma. 2019. Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. arXiv:1901.02358. Retrieved from https://arxiv.org/abs/1901.02358
[59] IT Union. 1992. *ITU-T81—Information Technology—Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines.*
[60] Larry Bank. 2020. JPEGDEC. Retrieved from https://github.com/bitbank2/JPEGDEC
[61] 2006. OV2640 Color CMOS UXGA (2.0 MegaPixel). Retrieved from https://www.uctronics.com/download/OV2640_DS.pdf
[62] Mohammed Elbadri, Raymond Peterkin, Voicu Groza, Dan Ionescu, and Abdulmotaleb El Saddik. 2005. Hardware support of JPEG. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*. IEEE, 812–815.
[63] Zhihui Wang, Shouyi Yin, Fengbin Tu, Leibo Liu, and Shaojun Wei. 2018. An energy efficient jpeg encoder with neural network based approximation and near-threshold computing. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'18)*. IEEE, 1–5.
[64] 2020. Ultra-low-power Arm Cortex -M4 32-bit MCU+FPU, 150DMIPS, up to 2MB Flash, 640KB SRAM, LCD-TFT & MIPI DSI, AES+HASH. Retrieved from https://www.st.com/resource/en/datasheet/stm32l4s5vi.pdf
[65] 2022. Ultra-low-power Arm Cortex-M4 32-bit MCU+FPU, 100DMIPS, 128KB flash, 40KB SRAM, analog, AES. Retrieved from https://www.st.com/resource/en/datasheet/stm32l422rb.pdf
[66] Emma Berry, Narinder Kapur, Lyndsay Williams, Steve Hodges, Peter Watson, Gavin Smyth, James Srinivasan, Reg Smith, Barbara Wilson, and Ken Wood. 2007. The use of a wearable camera, SenseCam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: A preliminary report. *Neuropsychol. Rehabil.* 17, 4–5 (2007), 582–601.
[67] Ravenslofty. 2022. yosys—Yosys Open SYnthesis Suite. Retrieved from https://github.com/YosysHQ/yosys
[68] Shengjie Bi, Kelly Caine, Ryan Halter, Jacob Sorber, David Kotz, Tao Wang, Nicole Tobias, Josephine Nordrum, Shang Wang, George Halvorsen, Sougata Sen, Ronald Peterson, and Kofi Odame. 2018. Auracle: Detecting eating episodes with an ear-mounted sensor. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 2, 3 (9 2018), 1–27. DOI: http://dx.doi.org/10.1145/3264902
[69] Edison Thomaz, Irfan Essa, and Gregory D. Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. ACM, New York, NY, 1029–1040. DOI: http://dx.doi.org/10.1145/2750858.2807545
[70] Oliver Amft, Mathias Stäger, Paul Lukowicz, and Gerhard Tröster. 2005. Analysis of chewing sounds for dietary monitoring. In *International Conference on Ubiquitous Computing*. Springer, 56–72.
[71] Abdelkareem Bedri, Diana Li, Rushil Khurana, Kunal Bhuwalka, and Mayank Goel. 2020. FitByte: Automatic diet monitoring in unconstrained situations using multimodal sensing on eyeglasses. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Vol. 20. Association for Computing Machinery (ACM), New York, NY, 1–12. DOI: http://dx.doi.org/10.1145/3313831.3376869
[72] Mark Mirtchouk, Drew Lustig, Alexandra Smith, Ivan Ching, Min Zheng, and Samantha Kleinberg. 2017. Recognizing eating from body-worn sensors. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 1, 3 (9 2017), 1–20. DOI: http://dx.doi.org/10.1145/3131894
[73] Shibo Zhang, Yuqi Zhao, Dzung Tri Nguyen, Runsheng Xu, Sougata Sen, Josiah Hester, and Nabil Alshurafa. 2020. NeckSense: A multi-sensor necklace for detecting eating activities in free-living conditions. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 4, 2 (6 2020), 1–26. DOI: http://dx.doi.org/10.1145/3397313
[74] Ang Li and David Wentzlaff. 2021. PRGA: An open-source FPGA research and prototyping framework. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 127–137.
[75] Mahadev Satyanarayanan, Nathan Beckmann, Grace A. Lewis, and Brandon Lucia. 2021. The role of edge offload for hardware-accelerated mobile devices. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications (HotMobile'21)*. 22–29.
[76] Akhil Mathur, Nicholas D. Lane, Sourav Bhattacharya, Aidan Boran, Claudio Forlivesi, and Fahim Kawsar. 2017. DeepEye: Resource efficient local execution of multiple deep vision models using wearable commodity hardware. In *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services*. Association for Computing Machinery, New York, NY, 68–81. DOI: http://dx.doi.org/10.1145/3081333.3081359

[77] Mingui Sun, Lora E. Burke, Zhi Hong Mao, Yiran Chen, Hsin Chen Chen, Yicheng Bai, Yuecheng Li, Chengliu Li, and Wenyan Jia. 2014. Ebutton: A wearable computer for health monitoring and personal assistance. In *Proceedings of the Design Automation Conference*. IEEE, Los Alamitos, CA. DOI: http://dx.doi.org/10.1145/2593069.2596678

[78] Xiang Chen, Yiran Chen, Zhan Ma, and Felix C.A. Fernandes. 2013. How is energy consumed in smartphone display applications? In *ACM Workshop on Mobile Computing Systems and Applications*. DOI: http://dx.doi.org/10.1145/2444776.2444781

[79] Robert LiKamWa and Lin Zhong. 2015. Starfish: Efficient concurrency support for computer vision applications. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'15)*. 213–226. DOI: http://dx.doi.org/10.1145/2742647.2742663

[80] Robert Likamwa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. 2013. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'13)*. 69–81. DOI: http://dx.doi.org/10.1145/2462456.2464448

[81] Jinhan Hu, Alexander Shearer, Saranya Rajagopalan, and Robert LiKamWa. 2019. Banner: An image sensor reconfiguration framework for seamless resolution-based tradeoffs. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys Adjunct'19)*. 705–706. DOI: http://dx.doi.org/10.1145/3307334.3328594

[82] Jinhan Hu, Alexander Shearer, Saranya Rajagopalan, and Robert LiKamWa. 2019. Banner: An image sensor reconfiguration framework for seamless resolution-based tradeoffs. In *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services*. Association for Computing Machinery, New York, NY, 236–248. DOI: http://dx.doi.org/10.1145/3307334.3326092

[83] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. 2018. Faster neural networks straight from JPEG. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS'18)*.

[84] Thomas Winkler and Bernhard Rinner. 2013. Sensor-level security and privacy protection by embedding video content analysis. In *Proceedings of the 18th International Conference on Digital Signal Processing (DSP'13)*. IEEE, 1–6.

[85] Thomas Winkler and Bernhard Rinner. 2011. Securing embedded smart cameras with trusted computing. *EURASIP J. Wireless Commun. Netw.* (2011), 8.

[86] Thomas Winkler and Bernhard Rinner. 2010. Trustcam: Security and privacy-protection for an embedded smart camera based on trusted computing. In *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'10)*. IEEE, 593–600.

[87] Yupeng Zhang, Yuheng Lu, Hajime Nagahara, and Rin-ichiro Taniguchi. 2014. Anonymous camera for privacy protection. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR'14)*. IEEE, 4170–4175.

[88] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. 2014. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 571–582.

[89] David H. Nguyen, Alfred Kobsa, Gillian R. Hayes, Gabriela Marcu, Gillian R. Hayes, Khai N. Truong, James Scott, Marc Langheinrich, and Christof Roduner. 2008. Encountering SenseCam: Personal recording technologies in everyday life. *Proceedings of the Annual Conference on Ubiquitous Computing (UbiComp'08)*, 182. DOI: http://dx.doi.org/10.1145/1620545.1620571

[90] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. 2014. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2377–2386.

[91] Paarijaat Aditya, Rijurekha Sen, Peter Druschel, Seong Joon Oh, Rodrigo Benenson, Mario Fritz, Bernt Schiele, Bobby Bhattacharjee, and Tong Tong Wu. 2016. I-pic: A platform for privacy-compliant image capture. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 235–248.

[92] Mohammed Korayem, Robert Templeman, and Dennis Chen. 2016. Enhancing lifelogging privacy by detecting screens. 10–15.

[93] Rakibul Hasan, Eman Hassan, Yifang Li, Kelly Caine, David J. Crandall, Roberto Hoyle, and Apu Kapadia. 2018. Viewer experience of obscuring scene elements in photos to enhance privacy. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 47.

Received 3 August 2023; revised 28 February 2024; accepted 7 May 2024