CAREER: Rhythmic Pixel Region Interface Systems
for Efficient, Performant, and Precise Augmented Reality

1. Overview

Augmented reality (AR) has begun to transform the ability for users to connect with the world around them, visually transforming their spatial environments with rich virtual overlays. This has provided interactive experiences for education, remote guidance for workforce training, immersive manuals for do-it-yourself home construction, construction industry monitoring, interior design envisioning for consumer purchases, and entertaining games. As such, enabling AR use cases converges with the needs of NSF Big Ideas "Harnessing the Data Revolution" and "Future of Work at the Human-Technology Frontier."

Unfortunately, mobile and wearable AR systems are limited by their spatial precision, computational performance, and energy efficiency while positioning 3-dimensional virtual overlays over the physical environment. These systems are further constrained by their mobile and wearable form factors with limited battery sizes and heat management requirements. This proposal seeks to improve the precision, performance and energy efficiency of wearable AR systems, based on a key insight:

The precision, performance, and efficiency of AR systems are limited by the current pattern of capturing and processing entire image frames at uniformly high resolutions and high frame rates.

Modern sensors are capable of capturing at "4K" resolution and 60 frames per second (fps), which results in 4.98 x 10^8 pixels sampled per second (3840 pixel columns x 2160 pixel rows x 60 fps). Such spatiotemporal resolution provides great immersive potential for fine-grained tracking precision by observing visual details on surfaces. However, operating on nearly half of a billion pixels per second results in a data rate that overwhelms memory interfaces, creating bottlenecks to precision, performance and efficiency. Due to the inflexibility of the imaging pipeline, current application frameworks typically sacrifice image resolution to meet performance demands under energy budgets. The reduced image resolution across the entire image results in suboptimal tracking precision on current systems.

However, precise tracking does not require high resolution across the entire frame, nor is it necessary at all times; high resolution is only needed near certain visual features in the scene and on an occasional basis to maintain tracking precision. To this end, the project seeks to rethink frame-based assumptions and rearchitect the visual computing system architecture around processing and producing ***image pixel regions*** with varying ***spatial resolutions*** and ***temporal rhythms*** (intervals), selectively guided by the AR frameworks and applications. This is illustrated in Fig. 1.

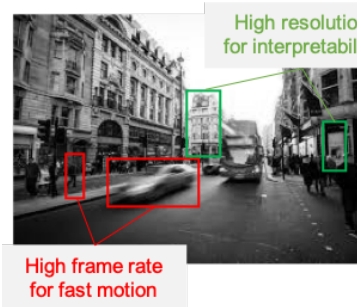

Fig. 1: Different frame areas require different spatiotemporal resolutions, some requiring high resolution and/or high capture rate.

Successful completion of the proposed project will provide order-of-magnitude improvements to precision, performance, and efficiency, striving towards 10-hour battery life with millimeter-level precision of augmented placement at fully interactive performance with less than 20 ms latency. Furthermore, the project will aim to make this possible with existing high-resolution image sensors and displays, allowing for this revolutionary upgrade on evolutionary mass-market-scale devices. The project will also aim to sustain existing augmented reality development frameworks, such as those on Unity Game Engine and Unreal Engine, allowing developers to utilize the benefits of the proposed project with no additional developer burden. Altogether, this will provide transformative benefits to the AR/VR industry, allowing for beneficial effects on a wide array of applications for education, workforce training, and entertainment.

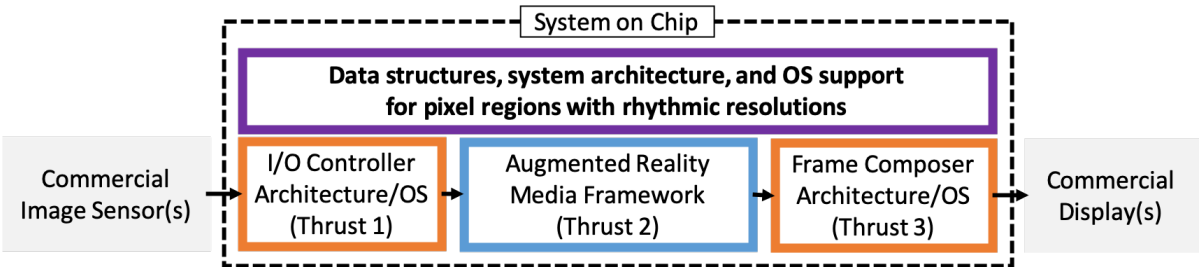

Fig. 2: Overview of the system contributions of the proposed project. The proposed project seeks to work with traditional sensors and displays, but rethinks the system architecture for visual computing around non-frame-based visual data streams with rhythmic spatiotemporal resolutions.

We will study principles to rearchitect the visual computing system at multiple layers:

(i) the I/O interface architecture between the computer system, the image sensor, and the display,
(ii) the operating system for governing memory access patterns, and
(iii) the visual algorithm implementations that power the augmented placement.

Fig, 2 shows a depiction of our envisioned system contributions and how they fit into the wearable augmented reality computer system.

2. Background on Visual Computing Systems for Augmented Reality

The ecosystem of visual computing sensors, devices, systems, and algorithms on mobile devices have rapidly evolved to provide high-performance platforms for augmented reality on smartphone, tablet, and headset form factors. Here we provide an overview of the state-of-the-art of current image sensing systems for augmented reality on mobile devices.

Image sensors collect digital readings of visual pixels in a frame. Typically, an aperture and lens focus a visual image of light onto an array of photodiodes such that each pixel receives light incoming from a different spatial angle. The photodiodes convert the light into analog charge, which is subsequently converted into digital pixel values. The analog-to-digital converters operate on the pixels in a raster-scan order, reading each row from left-to-right and iterating on rows from top-to bottom.

To communicate the pixel values the sensor sends values over a streaming **MIPI interface**, which enacts a serial transmission over multiple lanes. The MIPI receiver receives the frame information into the system-on-chip. In the sensor or on the system-on-chip, there is often an **image signal processor** inserted into the visual computing pipeline, performing image improvement operations, e.g., white balance, and format changes, e.g., YUV conversion. Regardless of the placement and operation of the image signal processor, the visual hardware pipeline eventually writes the frame into DRAM and signals to the operating system that a frame is ready for readout from the memory.

For augmented reality, the software processes the frame through **visual computing frameworks**, extracting visual features to feed into simultaneous localization and mapping (**SLAM**) algorithms [1]. These algorithms form a spatial understanding of the scene to estimate the pose of the camera. When the virtual pose of the camera in the virtual spatial environment is updated to be where the estimated pose of the camera sits, the augmented reality illusion is created, allowing for accurate positioning of scene overlays.

For **camera pass-through-based augmented reality**, such as that found in smartphones and tablets, the camera frame serves as a background for the virtual objects to compose an entire complete display frame. For **optical see-through glass-based augmented reality,** such as that found on HoloLens/MagicLeap headsets, the real world itself serves as the background for the virtual overlay, which is projected onto the

transparent glass for the user to see. In both forms of augmented reality, the virtual scene is overlaid on the visual understanding of the physical environment to create the augmented reality perception.

3. Related Work

Multi-ROI sensors: Many commercial image sensors are capable of selecting a region-of-interest (ROI) for readout [2], [3]. There are also sensors that allow for multiple ROIs to be read out [4], [5]. This has become useful for machine vision, purposes, e.g., chip inspection. As the region selection is performed at the sensor level, it offers efficiency and speed by reducing sensor readout time. However, there are significant limitations to adopting these sensors for continuous vision purposes, e.g., augmented reality approach. The expressiveness of sensor-based region selection is limited by the footprint of introducing additional circuitry. For example, in one such sensor, the region selection is limited to 4 regions, regions cannot overlap, and only full resolution and full frame rate are available [4], [5]. Our proposed project seeks to provide *configurability* and *composability*, especially in allowing a much larger number of regions, and granting each region independent resolution and rhythm/interval control. Moreover, by implementing ROI selection at the interface level, we can ensure that any sensor can employ multi ROI benefits.

Event-driven cameras: Event-driven cameras, also known as dynamic vision sensors, only sampling pixels that change their value [6], [7]. This also allows for a significant reduction in sensor bit rate and allows microsecond time resolution. However, the circuitry is spatially expensive, reducing frame resolution, e.g., to 128 x 128 pixels, and only allowing monochromatic. More fundamentally, the logic of deciding what pixels to read out is limited at hardware design time, disallowing high-level and/or semantic knowledge from governing the pixel selection process. Thus, while our work shares similar motivations and inspirations -- reducing data rate for efficiency and performance – we uniquely allow the expressive ability to dynamically use knowledge of visual feature extraction needs to selectively sample pixels as needed.

Image/Video Compression: Decades of work in image processing has gone towards compressing images and videos to reduce the bit rate of storing and transmitting media. Many of these techniques are inspiring to this work. For example, JPEG and other image compression standards reduce information at spatial frequencies where it is perceptually less important [8]. MPEG-H Part 2 / HEVC / H.265 reduce redundant information where there is little motion from frame-to-frame [9]–[11]. However, existing compression techniques require the frame – or multiple copies of the frame – in memory before compression can be done. Our work aims to significantly reduce the memory burden to improve the efficiency and performance of the visual computing system. We can build our techniques on the similar logic and reasoning that governs compression, reducing visual redundancy as possible in spatial and temporal dimensions.

Foveated Rendering: To improve the graphical rendering capability of virtual reality systems under limited computing resources, much work has gone into focusing rendering effort where the user will notice the most. Foveated rendering uses eye tracking to estimate user gaze and renders content near the gaze at a higher spatial resolution. We apply similar motivation on the sensing stream, increasing spatiotemporal resolution where it is visually needed, but with a distinctly different goal: to capture the necessary information to support augmented reality overlay. Among other differences, our work involves multiple regions of interest, as opposed to the singular region in foveated rendering.

4. The Opportunity of Rhythmic Pixel Region Representations

To decimate the computational overhead of frame-based imaging, the proposed CAREER project aims to create system designs that replace *frame-based* pixel interfaces with *rhythmic pixel region* interfaces.

Our previous NSF-sponsored research project studied the effects of image resolution on visual computing efficiency and designed operating system mechanisms to leverage the energy efficiency gains of

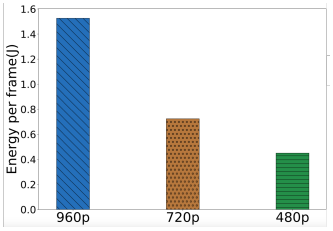

Fig. 3: The energy-per-frame of the computer system is reduced with lower frame resolutions

situationally reduced resolutions [12]. Through our investigation, we found that situationally reducing resolution, e.g., for near-camera marker-based tracking, can proportionally reduce the energy consumed per frame, as shown in Fig. 3. At the same time, we found that operating on fewer pixels allowed the system to improve its tracking frame rate. Thus, reducing pixel count can provide an effective means of improving performance *and* reducing energy consumption. Our subsequent Banner system [13] rearchitected the device driver and media framework to allow such resolution changes with reduced latency and eliminated frame drops during the reconfiguration procedure.

However, reconfiguring resolution only on a frame-by-frame basis limits the beneficial opportunities of reduced resolution; frame-based imaging limits the number of pixels that can be discarded. Consequently, rather than capturing, transmitting, and computing all image pixels throughout the computer system, we will insert an I/O interface into the pipeline that will only transfer pixels that are relevant and necessary for the ongoing visual computing tasks.

To do so, we define a *rhythmic pixel region* structure, which includes:

(i) the coordinates of the center of the region,
(ii) the width and height of the region,
(iii) the resolution of the region, i.e., the density of pixels,
(iv) the time interval (rhythm) between consecutive sampling of the region, and
(v) the bit-depth of each pixel in the region,

Fig. 4 illustrates output of rhythmic pixel regions. The table below describes opportunities for rhythmic pixel region-based visual computing to discard substantially more pixels than frame-based visual computing.

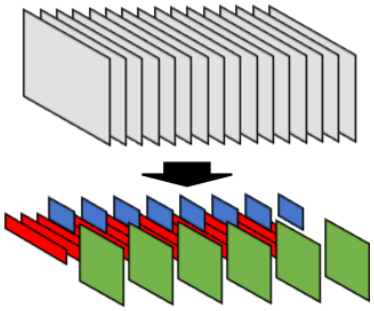

Fig. 4: Rhythmic pixel regions can be specified to capture at different resolutions and intervals, depending on visual needs.

	Traditional frame-based vision	**Rhythmic pixel region vision**
Resolution	If any part of the frame needs to be captured at a high resolution, e.g., to resolve complex texture or distant objects, the entire frame will need to be captured at a high resolution.	Frame regions with small, detailed, and/or distant visual features can be captured with the precision of high resolution, while frame regions with large, static, close visual features can be captured with the efficiency of low resolution.
Interval	If any part of the frame needs to be captured at a high frame rate, e.g., to track substantial motion, the system will need to capture a sequence of entire frames of pixels at high frame rates.	Regions can be captured at different intervals. The entire frame can be scanned to update spatial understanding at 1 fps, while regions of moving objects/surfaces can be captured at 60 fps.
Bit-depth	If any part of the image needs more contrast, e.g., due to poor lighting conditions, all pixels will need to be captured with high bit-depth. Conversely, if parts of the image are well-lit, the sensor interface must sustain a constant (high) bit-depth so as to provision for regions with poor contrast.	Poorly lit and/or with complex texture patterns can benefit from raised bit-depth. Regions that are well-lit and/or with visually untextured regions can reduce bit-depth for energy savings and performance improvement.

As we describe in Thrust 3, this leads to substantial unprecedented opportunities for smartphone-based augmented reality, as the entire frame can be captured at relatively low spatiotemporal resolution and frame rate, e.g., 1280 x 720 and 30 fps, while the visual features can be captured at higher spatiotemporal resolutions as needed for precise overlay, e.g., 3840 x 2160 (or higher) and 60 fps (or higher).

Furthermore, for head-mounted optical see-through augmented reality, many regions of pixels need not be captured at all. The system will only need to capture regions where visual features will assist the tracking of surfaces and objects, and it can do so at high resolutions and temporal rates. The system may occasionally need a full frame of capture to detect and characterize regions where rhythmic pixel regions are needed, but this can be done on a much less frequent basis.

Thus, the rhythmic pixel region representation allows computing systems to operate on many fewer pixels, eluding the data rate burden that currently comes with high resolution-based visual computing.

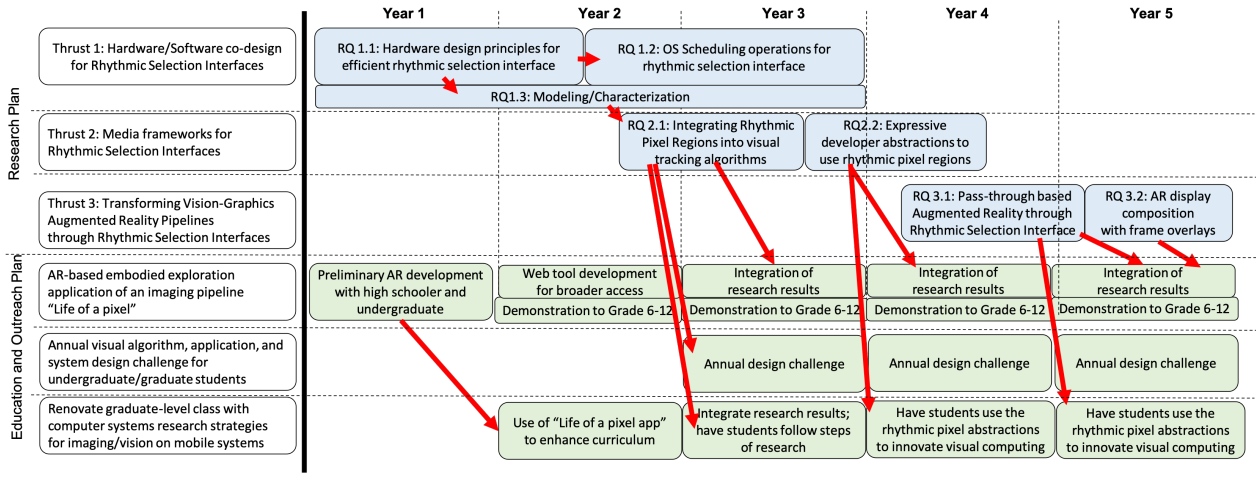

Fig. 5: An overview of the schedule of thrusts in the research plan and activities in the education and outreach plan

5. Research Plan

The chief governor of pixel selection will be a specialized hardware I/O interface, which we call the *rhythmic selection interface*, situated between the sensor and the system bus or DDR memory. The rhythmic selection interface will only emit a pixel if it satisfies one of the developer-specified rhythmic pixel regions in both time and space. Pixels that do not satisfy any rhythmic pixel regions will be "gated", that is, not read out from the rhythmic selection interface. In doing so, we will substantially reduce the pixel workload of the computer system.

Successful implementation and usage of the rhythmic pixel region and the rhythmic selection interface will require cross-layer optimization across the wearable computer system, from sensor I/O hardware interface to operating system control, to visual computing algorithm, to augmented reality developer library. This will comprise of a hardware I/O interface to select rhythmic pixel regions, software control to dynamically specify rhythmic pixel regions, software libraries to implement visual computing algorithms on the rhythmic pixel regions, and integration into a vision-graphics pipeline for wearable augmented reality computer systems.

Over the course of the proposed 5-year CAREER project, the team will embark on three thrusts, as shown in Fig. 5. Thrust 1 will study mechanisms for the rhythmic sensing hardware architecture, focusing on a software/hardware sensor interface that filters a collection of pixel regions at specified locations, spatial resolutions and temporal intervals. Thrust 2 will study design patterns and strategies for building media

frameworks around rhythmic pixel regions, including data structures and software libraries for visual computing through feature-based extraction and convolutional neural network. Thrust 3 will study an integrated vision-graphics pipeline for computer systems to efficiently compose augmented visual output to the displays, driven by the rhythmic pixel region architectures of Thrust 1 and 2.

5.1 Thrust 1: Hardware/Software co-design for Rhythmic Selection Interfaces

The crux of rhythmic visual sensing is in the ability to selectively sample pixels as governed by the visual computing algorithm. At the same time, to harmonize with the existing industry of image sensors (maximizing the potential impact of the proposed project), our proposed system will operate on pixel data read out from standard image sensors, i.e., pixels read from the sensor in raster scan order. Our key idea is to architect a sensor I/O controller that selectively discards pixels according to specified rhythmic pixel regions. We call this I/O controller the *rhythmic selection interface*.

For maximal energy savings and performance, pixels should be selected/discarded in the pipeline as close to the sensor as possible. We will architect our hardware interface to perform the selection before pixels reach the expensive read/write memory interface of the DDR memory. The same hardware interface can also be placed on the image sensor itself to reduce the overhead of the sensor-side I/O controller that pushes imaging data over a lengthy ribbon cable. Through FPGA-based prototyping, we will design and evaluate the rhythmic pixel selection interface such that it can be inserted in either position in the system.

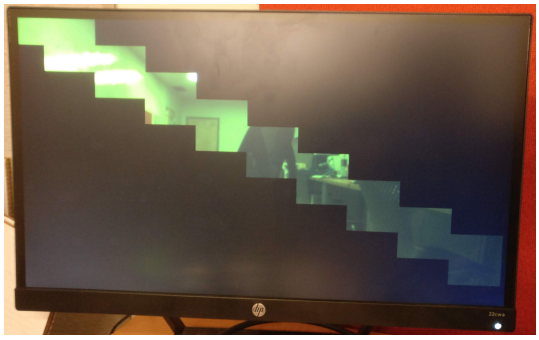

Fig. 6: The display output of our early FPGA implementation shows the capability to filter regions with different spatiotemporal resolutions through the interface on the system-on-chip.

Early implementation

As a proof-of-concept, we have constructed an early implementation of the I/O controller, capable of reading out different specified regions at various resolutions and intervals. We use Xilinx Vivado HLS to compile RTL for the ZCU102 FPGA-SoC board and insert the rhythmic sampling interface after the MIPI interface on the FPGA/System-on-Chip. Our implementation uses memory-mapped I/O to receive region requests from the software. Then, as a pixel streams into the hardware, the interface checks whether the pixel's coordinates correspond with a region's specified spatial location and timing interval. If it does, the pixel will be sent onwards. If not, the pixel will not be released. This results in imaging data composed of regions of different spatial resolutions and timing intervals, such as that of Fig. 6.

Rather than omitting pixel transmission (as is a goal of this proposed project), our current I/O controller transmits black pixels. Thus, while the current implementation demonstrates that it can filter pixels in real time, it does not actually reduce the pixel count over the interface or the memory footprint of the frame. This severely limits energy savings and performance improvements. Actual measurable improvements will require the proposed research towards a tight integration of hardware and software designs.

To this end, this first thrust of the proposed project proposes three high-level research questions: (*i*) What are principles to design an efficient rhythmic selection interface? (*ii*) What OS scheduling operations are necessary to connect the rhythmic selection interface hardware to the software abstractions? (*iii*) How can we model and characterize the interactions of the hardware/software rhythmic selection interface? We articulate our thoughts towards these research questions below.

Research Question 1.1: What are design principles toward an efficient rhythmic selection interface?

Through our proposed study of implementing the rhythmic selection interface, we plan to derive principles for efficient rhythmic selection interfaces. High performance and low energy consumption are both of paramount importance to facilitate the high-level performance and energy efficiency needs of the wearable computer system. We prioritize the servicing of high pixel rates while maintaining low power consumption. It is well known that dynamic voltage/frequency scaling can allow active power dissipation of computing to scale sub-linearly with clock frequency [14]–[16], i.e., energy efficiency improves dramatically at lower clock frequencies. Ultimately, this means we will aim to maintain a low clock speed and halt the clock (clock gating) as much as possible while servicing a high input pixel data rate and providing rhythmic pixel region output in a timely fashion. Here we list two mechanisms that can form design principles for efficient rhythmic selection interface design.

Input pixel accumulation for reduced clock frequency: Rather than process each pixel as it comes in, we will study design patterns that will allow the interface to accumulate multiple pixels into a buffer for batch processing. This will allow groups of pixels to be processed in parallel through loop unrolling or other software-hardware co-design strategies. By parallelizing operations, we can reduce the number of clock cycles required to complete the processing of a set of pixels, allowing for reduced clock frequencies. To this end, we will investigate optimal buffer sizing for the parallelism. A larger buffer size allows for a greater degree of parallelism – and therefore lower clock frequency -- but introduces latency and hardware resource utilization. Latency and hardware resource utilization will depend on the number of pixels to process, and the number and size of regions to process. Thus, it is likely that the optimal choice will be deadline-driven and workload-dependent. By studying the energy and timing implications in the tradeoff between clock frequency, pixel processing latency, and hardware resource utilization, we can determine optimal strategies for the system to adapt the pixel buffer length at run-time.

Output pixel accumulation for aggressive memory idling: Unlike our early implementation, which sends out black pixels for pixels outside of rhythmic pixel regions, we will omit the transmission of a non-regional pixel and use the lack of transmission as an opportunity to raise the efficiency of the system. As we discuss in Research Question 1.2, the operating system will be able to re-sort the pixel stream to place pixels in their correct rhythmic pixel regions. However, before the operating system can receive the pixel stream, the rhythmic selection interface will need to compose a pixel stream by packing pixels into a data stream.

We have observed that in the sample Xilinx Vivado ISP implementations for standard imaging pipelines, pixels are typically read in and out at two pixels per clock cycle. For the rhythmic selection interface, we will craft a different pattern, as: (*i*) not every pixel will be read out, (*ii*) some pixels have a higher bit-depth than others, and (*iii*) the gaps between pixel transmissions will provide opportunities for memory to enter idle states. This third implication is particularly interesting, as a greater time spent waiting to accumulate pixels will allow the interface output to remain idle for longer, giving opportunity to put the targeted memory device into a deeper idle state for substantial power savings. For example, entering the deepest idle state of LPDDR4, the "Self-Refresh Mode" (SR), would reduce power consumption of memory by an order of magnitude. However, entering/exiting SR mode requires the idle period to cover the hundreds of cycles (at least 512 clock cycles) for the memory to re-enter a DLL lock. Such opportunities could present themselves as readout is held over thousands of pixels, which may be skipped or accumulated before transmission. However, strategies will need to ensure that the system does not suffer from excessive latency penalties, especially latency that would compound and grow over time. We will investigate the use of different idle modes and strategies to reduce or mitigate the latency penalties of entering such modes, e.g., by situationally tuning the number of output pixels to accumulate.

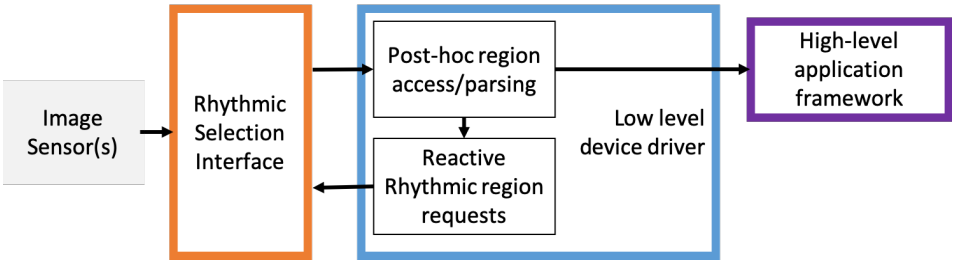

Fig. 7: As the visual computing system observes different visual needs, the logic can trigger the request of different rhythmic pixel regions. To reduce the system calls, the OS can schedule rhythmic pixel region request operations in a low-level device driver.

Research Question 1.2: What OS scheduling operations can connect the rhythmic selection interface hardware to the software abstractions?

Properly designed software control is imperative to facilitating efficient and correct use of the rhythmic selection interface. In its full implementation, visual computing will create a continuous stream of requests to the rhythmic selection interface. In our previous work, we characterized that frame-level synchronization throughout the operating system created latency bottlenecks to sensor resolution reconfiguration [12]. Our work Banner [13]avoided such bottlenecks by providing format-oblivious memory management and parallel reconfiguration routines. We will search for similar strategies, such as those below, to keep reconfiguration operation invisible.

Post-hoc region access: A driver in the OS will request rhythmic pixel regions to capture by communicating a metadata structure that contains the position, resolution, interval, and bit-depth of each region. Notably, the output pixels from the rhythmic selection interface will be ordered by raster-scan and *not* organized by region. Pixel regions can be sorted on-demand post-readout as long as the metadata is synchronized. The OS will be responsible for efficiently synchronizing the metadata by maintaining a log of rhythmic pixel region requests and when the requests were serviced. Any access of the regions will consult with the log to compute the memory address positions of the relevant pixels. We will study methods to predictively schedule such log-based accesses with minimal latency.

Reduced latency: In our previous works, we have found that system calls between the userspace and kernel introduce substantial latency, e.g., 10s of ms, which creates efficiency and performance penalties. Thus, in communicating rhythmic pixel region requests, we will leverage recent results in *reactive programming* to dynamically control the operation of the rhythmic selection interface based on incoming pixel data. We will provision for such logic and region request to be run in the kernel to avoid system calls. We illustrate this in Fig. 7. This can be used to define primitive operations, e.g., if substantial variation is seen in a region, increase the region's resolution. If region motion from one interval to the next contains substantial variation, decrease the interval and increase the region size.

For straightforward execution, we plan to implement the control in a low-level device driver, which will be sufficient to avoid system calls. However, we also plan to investigate the possibility of leveraging in-memory computing strategies to assist in the reactive programming operations to make rhythm decisions near the pixel region data itself.

Research Question 1.3: How can we accurately model and characterize the interactions of the hardware/software rhythmic selection interface?

Our study will require reliable realistic characterizations to experiment with various rhythmic sampling patterns, architectures, and software systems. From there, we can further experiment with modeling the use of rhythmic pixel regions for media frameworks and mobile/wearable augmented devices.

FPGA-based performance measurements: We will use FPGAs to create rhythmic selection interfaces. For our preliminary implementation, we used the Xilinx ZCU102 and Vivado HLS to compose an interface. Our setup is pictured in Fig. 8.

Measurement-driven energy models: As in our previous works, we will employ measurements from image sensor kits, mobile device development kits to estimate the potential energy savings. We will compare our models to measurements from commercial smartphones/tablets/headsets to validate the measurements in the context of real-world mobile computer systems.

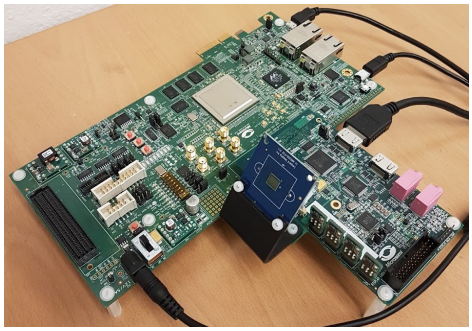

Fig. 8: FPGA setup used for early implementation, based around the Xilinx ZCU102 System-on-Chip/FPGA.

5.2 Thrust 2: Media frameworks for Rhythmic Selection Interfaces

Starting in Year 3, we will begin to integrate our rhythmic selection interface in visual tracking frameworks and augmented reality development frameworks.

Research Question 2.1: How can visual tracking algorithms use rhythmic pixel regions to improve their precision, performance, and efficiency?

As discussed in the introduction, visual algorithms observe visual features (corners/edges) in an environment to geometrically update the pose of the camera and register newly observed visual features. We will investigate systematic strategies to use rhythmic pixel regions to improve the precision, performance, and efficiency of visual tracking.

Feedback-driven rhythmic selection for feature extraction: Across intervals, the visual features in the scene are likely to be in similar locations. Thus, after sampling a frame-sized region and locating visual features, the algorithm can proceed by requesting a set of rhythmic pixel regions that encompass the visual features. Requesting one rhythmic pixel region per visual feature would create unreasonable overhead in the system, as transmitting the list and implementing the search would overwhelm the rhythmic selection interface. Instead, we will target an efficient bounding box search that captures all visual features. Such a search will aim to constrain the number of requested regions, minimize the number of transmitted bits, and satisfy pixel region needs.

Predicting motion patterns: As the camera physically moves, visual features will also move in predictable manner. We can leverage sensor fusion with accelerometer, gyroscope, magnetometer to measure device

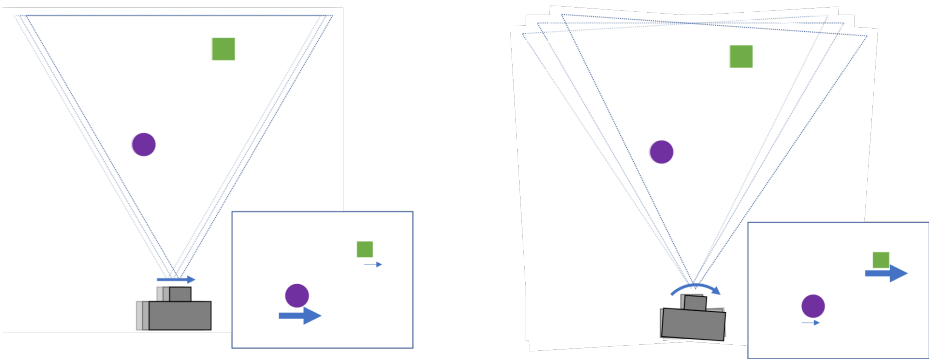

Fig. 9. Pan (left) and Tilt (right) camera motions, as inferred by motion sensors, present motion patterns in the frame, depending on the object's distance from the camera.

movement and estimate the movement of the visual features in the frame. With subtle "pan" movements, near features move more than far features. With subtle "tilt" movements, the inverse is true. (Note that most movement is "subtle" at 15-60 fps or higher). An illustration of this is shown in Fig. 9. For motion of objects within the scene, we can track the movement of extracted visual features. To accommodate for large movements, we can widen the region size around visual features when/where there's expected to be much camera motion or motion within the scene.

Leveraging feature resolution-scaling: The precision needs of a visual feature depends on its scale with respect to the size of the image frame. If a visual feature is far from the camera or small in size, then it will require high resolution for the system to accurately determine its position in the environment. Conversely, a large and/or close feature will suffice with lower resolutions to locate its position. We will investigate techniques to dynamically request resolution changes to accommodate the requirements of visual features.

As the regions re-locate to accommodate motion patterns or resolution changes, we will update such changes in the request log for consistency with the post-hoc region access.

Research Question 2.2: What abstractions will allow high-level AR developer frameworks to enact customizable visual computing based on rhythmic pixel regions?

We will not obligate high-level developers to think about rhythmic pixel regions. In typical use, developers should be able to directly leverage the tracking framework of Research Question 2.1. However, for direct use of the visual pipeline, our system will also provide the opportunity to operate directly with the rhythmic pixel regions.

We will provide an API that will allow developers to customize rhythmic pixel region requests by sending lists of rhythmic regions. The rhythmic region data structure will also include an event handler as a property. Inside the event handler, developers can write algorithms that operate on the region of pixels. Our runtime will gather the requests, retrieve the pixel regions, and call the appropriate event handler when ready. We will study software engineering designs.

5.3 Thrust 3: Transforming Vision-Graphics Augmented Reality Pipelines through Rhythmic Selection Interfaces

Starting in Year 4, built on the research outputs of Thrusts 1 and 2, we will investigate the integration of our rhythmic selection interface and rhythmic pixel region hardware and software into vision-graphics pipelines for augmented reality.

One of the most significant problems facing augmented reality systems today is *motion-to-photon latency*, a metric which measures the time from when a user moves her head to the time the display visually registers the virtual graphics with the physical environment. It has been a notably insurmountable problem, as camera frames in traditional frame-based systems require an entire frame to be read out from the camera before positioning can even begin. After a frame is processed, an entire graphics display frame cycle needs to synchronize with the update before display can happen. This introduces a lengthy two-frame delay into the pipeline, creating the motion-to-photon latency. Many augmented reality solutions employ sensor fusion from the inertial measurement unit (accelerometer, gyroscope, magnetometer) to rapidly estimate position and provide updates. However, visual tracking is still necessary to correct the inaccuracy from inertial drift [ref] and offers stronger precision guarantees. We aim to provide solutions to efficiently compose virtual displayed output that continuously aligns with the physical world through the use of the rhythmic pixel regions.

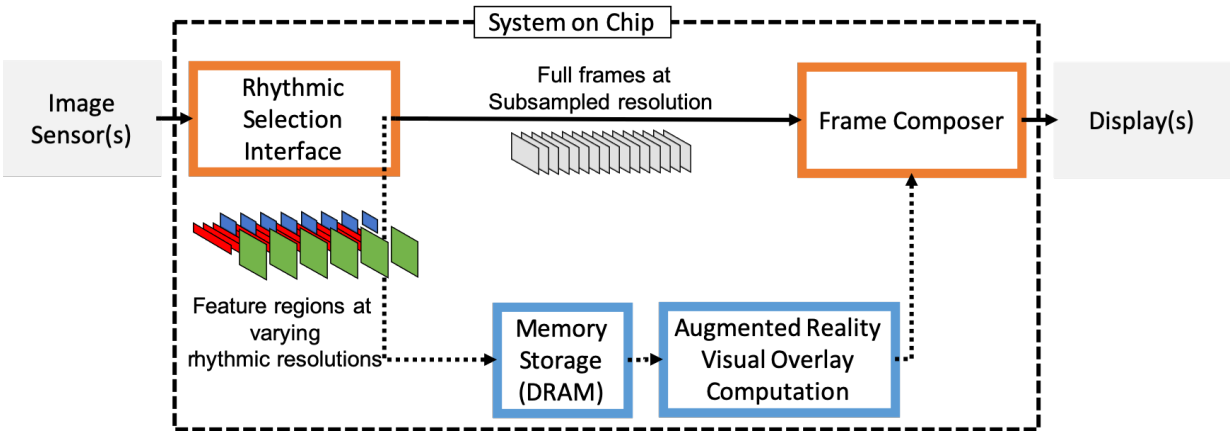

Fig. 10: System architecture prioritizes a frame pass through for efficient overlay without invoking DRAM memory. Rhythmic pixel region data can enter the DRAM with reduced data rate.

Research Question 3.1: How can rhythmic selection interfaces efficiently provision for pass-through based augmented reality with low motion-to-photon latency?

Pass-through based augmented reality – such as that on smartphones and tablets -- sends camera frames to the screen with augmented virtual objects overlaid. To minimize the latency of this process, we will create a display architecture that will allow the rhythmic selection interface to separate the stream two sets of visual data: (*i*) medium resolution camera frames directly for the display unit, and (*ii*) high resolution visual feature regions for visual tracking. Here, we discuss our envisioned plan for the direct-to-display stream, while covering the visual tracking stream in Research Question 3.2. Both are illustrated in Fig. 10.

For the display stream, we plan to investigate the use of a direct DDR-less stream to a display composition unit for pass-through with minimal latency. With line-level synchronous operation to provide near-zero latency. The display composition unit will also receive virtual graphics overlays through a separate channel, combining the two to compose augmented output. When viewed through a smartphone or tablet, the camera frame need not be at the full resolution capability of the sensor; 720p or 1080p is often standard and appears to be sufficient for user experiences. We will perform latency experiments by using mechanized robot arm apparatuses to manipulate our camera through a range of repeatable motions and high-speed cameras to measure the motion-to-photon latency of the stream.

Research Question 3.2: How can the system efficiently compose frame overlays for pass-through-based and optical see-through augmented reality?

The visual tracking stream is relevant for both pass-through-based augmented reality on mobile devices and optical see-through augmented reality on headsets. For the visual tracking, the rhythmic selection interface will provide rhythmic pixel regions as per Thrust 2. This stream will provide updated tracking for augmented placement, assisted with sensor fusion for up-to-date predictive tracking. Notably, our rhythmic selection interface will allow the visual tracking to leverage much higher resolution camera data, while keeping memory utilization low through selectivity. We predict that this will create dramatic improvements in performance and efficiency. We will study the limits of this strategy for updating virtual placement without frame-level synchronization.

To study the perceptual effects of the composition, we will perform a user study, varying the resolution limits of the display stream, the resolution of the visual tracking stream, and other tracking parameters that we may devise. We will ask users to report on the usability of the augmented reality system, building on similar evaluative studies done for virtual reality environments [17].

6. Intellectual Merit

Successful completion of this proposal would directly advance the state-of-the-art of computer architecture (I/O controller design) and operating systems (device drivers and media frameworks) research for vision systems on mobile/wearable devices. We will submit results for publication in top-tier conferences and journals, e.g., ASPLOS, MobiSys, MobiCom, SenSys, SOSP/OSDI, IEEE Transactions on Computers, etc.

More fundamentally, thinking about imaging and vision with *rhythmic pixel regions* presents a significant departure from the traditional frame-based thinking that currently dominates imaging and vision thinking in both academia and industry. Indeed, research engineers of imaging systems on mobile devices have been racing towards larger numbers of pixels per frame and faster frame rates, provisioning for increasingly higher bandwidths. In the opposite direction, academic researchers in the computer vision research community have reduced frame resolution to provision for sufficient performance. As opposed to these two strategies, we advocate for rhythmic region-based processing, which allows for high resolution where it is needed, and minimal pixel counts everywhere else, allowing for both efficiency and performance. Through the proposed project, we will provide early hardware/software solutions for rhythmic pixel regions. Taking early steps to transform the norm of imaging/vision trends towards precise, performant, and efficient vision.

Eventually, full integration of this proposed project within image sensor hardware would be able to leverage the full potential of the device trend towards high spatiotemporal sensors, e.g., 1000 fps and 12-megapixel sensors. There, the rhythmic selection interface would have the ability to gate the ADCs on the image sensor, yielding further opportunities for precision and performance with low power consumption. Due to the difficulty of hardware integration in high-resolution sensors and the necessity of industry involvement, sensor integration is outside of the scope of this proposed project, but successful completion of this project would yield momentum towards this future direction.

7. Broader Impacts

Transformative industrial impact: We will consult with industry partners at Microsoft Research – Mobility and Networking Research, Google Daydream, and Samsung Mobile Processor Innovation lab, holding meetings on an annual basis to present findings and seek feedback, especially as it pertains to envisioned product roadmaps. To further expand our reach, we will leverage our membership in WISCA, a Center for Wireless Information Systems and Computational Architectures approved by the Arizona Board of Regents. We will leverage this partnership to present our research findings to existing WISCA commercial partners, including Defense Advanced Research Projects Agency (DARPA), Massachusetts Institute of Technology Lincoln Laboratory, Office of Naval Research (ONR), Raytheon Corporation, SAZE Technology.

While this proposed project focuses on augmented reality use cases, thinking about image streams in terms of rhythmic pixel regions would have influence on a wide array of imaging/vision application areas, including computational photography and video compression. There are a multitude of stakeholders for efficient visual sensing, including the smartphone/headset industry, automotive industry, and the Internet-of-Things. Moreover, this project can benefit assistive technology for vision-impaired and memory-impaired individuals by providing a platform for continuous visual computing on wearable devices.

Technology transfer: At the same time, we will pursue patent applications towards interest in commercial integration of these ideas. Skysong Innovations – ASU's technology transfer arm – assists university inventors with the pursuit of commercial integration. We have a track record of collaborating with Skysong Innovations to engage in commercial discussions through intellectual property, having filed three provisional patent applications with ASU on our previous technologies. Built on the intellectual property of the proposed project be successful, the team would be interested in pursuing a startup venture, designing hardware IP with software control for rhythmic pixel region-based visual sensing.

8. Education and Outreach

The high-level education and outreach goal is to educate and inspire people of all ages and diverse perspectives to explore the intersection of creative thinking and critical thinking for visual computing. Our plan includes an array of education/outreach projects, including: (*i*) building an AR-based embodied visualization of how pixel processing works in the imaging/AR pipeline, (*ii*) hosting an annual design challenge for students to prototype efficient and performant vision algorithms/applications for visual localization and detection, and, (*iii*) renovating a graduate level Mobile Systems Architecture course with research-inspired education on practices in computer systems research. A schedule of these activities is shown in Fig. 11.

Through the execution of these education/outreach activities, I will leverage my position at Arizona State University, where I am an assistant professor in two departments: The School of Electrical, Computer, and Energy Engineering (ECEE), and the School of Arts, Media and Engineering (AME). Whereas ECEE is a traditional engineering department, AME is a transdisciplinary department committed to an *experimental study of experience*, bridging media artists, philosophers, and mathematicians with engineers from a variety of disciplines. AME's undergraduate major "Digital Culture" draws a diverse array of students, including under-represented minorities – *over 40% women and 40% Hispanic/Latino students for media engineering*. This makes Digital Culture and AME a pathway to STEM-based technical practices, contextualized with socially-aware philosophies.

1) AR embodied exploration of an imaging pipeline

Goal: Encourage engagement in computer systems research from undergraduate students, middle schoolers, high schoolers, and lifelong learners.

In Years 1-5, we will construct (and iteratively improve) an augmented reality experience that will walk users through the "life of an augmented reality pixel" as it gets captured through the image sensor, stored in memory, processed upon by a vision algorithm, and overlaid upon by the graphics processor, and sent out to the display unit. We will also showcase the benefits of our rhythmic pixel region-based system, illustrating our research process of striving to innovate solutions to real problems. This embodied exploration will be viewable through a headset, e.g., Microsoft HoloLens, Meta 2, Magic Leap One, or through Android or iOS smartphones/tablets in augmented reality.

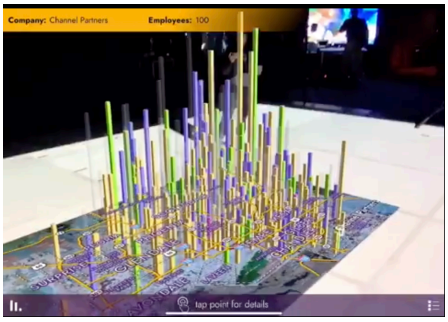

Fig. 12: Map-based data visualization in augmented reality

Through the creation of the AR experience, we will integrate undergraduate students and high school students in the development process. We have previously integrated high school sophomore Alec Gonzalez into our research, constructing a VR experience of ground viewing of spectral Mars data captured from orbit. His work earned him the first place in his category at his regional science fair and led to a publication at the Lunar and Planetary Science Conference in 2019 [cite]. We consistently integrate a diverse range of undergraduate students in our research through the Fulton Undergraduate Research Initiative, sponsored by ASU's engineering school. Over the past three years, Meteor Studio has employed thirteen undergraduate students, including five female students. In addition to assisting systems research projects, Undergraduate teams have developed contracted augmented reality visualizations for map-based visualizations of university enrollment and local employment data for the

ASU Office of the President and particle visualizations of ocean current models for the Synthesis Center at AME.

After the preliminary development -- which we expect to take one year -- we will deploy the AR experience to deliver open demonstrations at middle schools and high schools or field trips to our research lab. We will also deploy the AR experience in ASU open houses, attracting university students and post-graduate "lifelong learners" to visit the embodied educational experience. Finally, we will openly and widely disseminate materials – including source code and openness to modification -- for others to use in their high school or university-level curricula or science museums to broadly teach how pixels work in imaging/vision/AR.

Evaluation: To evaluate the success of the project, we will track the number and demographic of participants. We will also run a user study consisting of pre-experience and post-experience evaluative surveys to ascertain the engagement and educational effectiveness of the AR experience. To assist us with our development of the evaluative user study, we will consult with Dr. Mina Johnson-Glenberg in Psychology, who has created and assessed multiple NSF sponsored modules on AR/VR STEM content for grade schoolers and lifelong learners. Dr. Johnson-Glenberg also is the President and founder of Embodied Games, LLC, an NSF-funded small business that creates AR/VR STEM content for grade schoolers and lifelong learners. Beyond development experience in AR/VR settings, Johnson-Glenberg has experience evaluating the educational effectiveness of the sense of presence and embodied affordances that AR/VR provide [18]–[21].

2) Annual design challenge for vertically-integrated undergraduate/graduate student teams to prototype efficient and performant vision algorithms/apps for visual localization and detection

Goal: Facilitate collaborative participatory learning experiences through computer systems exploration. Encourage careers in engineering research.

As part of the proposed project, from years 3 – 5, we will host an annual university-wide challenge for students to competitively compose efficient and performant vision architecture/algorithms. Students will form vertical teams, mixing undergraduate and graduate students to develop solutions over a year. Our student recruitment will leverage ASU email listservs for Computer Science, Computer Engineering, Electrical Engineering, and Digital Culture programs. Through the recruitment, we will specifically encourage under-represented minorities and female students to participate in the challenge. We will provide a matchmaking service to assist students in finding team members. Teams will compete to complete a task in the categories of precision, performance, and accuracy. We will issue prizes to student teams that perform at the top of each category, as well as the best "overall" solution, and the most "innovative" solution.

For the challenge, we will issue a specific visual challenge: design a hardware/software solution that computes pose estimation from a set of signal datasets that represent the output of an image sensor on a wearable device, as well as the "ground truth" pose of the device, gathered via a VICON-like precise tracking system. We will run student solutions on the dataset. Prior to the competition, we will supply sample signal datasets, associated ground truth labels, and access to FPGA-SoC hardware for solution development.

Evaluation: We will quantitatively assess the efficacy of the challenge through the number and demographic of participants. We will also track the career path of the participants, including first job placement and/or graduate school enrollment. Furthermore, we will also use qualitative research methods – interviews, focus groups, participant observation – to understand and iteratively improve the experience of the challenge.

3) Renovate graduate-level class with computer systems research strategies for imaging/vision on mobile computing systems.

Goal: Engage and grow student competencies and interest in visual computing systems research

We plan to renovate my graduate-level course on Mobile Systems Architecture, introducing students to engage with mobile systems topics related to vision algorithms and applications. In Fall 2016, I designed the Mobile Systems Architecture as a "topics" course, studying background and active research papers from MobiSys, MobiCom, ASPLOS, and other related conferences. Since then, I have taught the course annually, lecturing about computer systems topics as they relate to mobile systems, having students give conference-style presentations on published papers, and having students work in teams on mobile application projects.

Through the CAREER plan, we will augment the student team projects to have students retake the steps of the research to the process of doing mobile systems research. Building on their growing competencies, we will have them evaluate vision applications running on the rhythmic pixel region system, as compared to traditionally processing vision algorithms through frame-based imaging. We will suggest vision applications or allow students to propose their own vision applications. We expect that this will lead to quantitative improvements in vision application metrics of performance, accuracy, and efficiency. We will encourage students to write research papers on these results. We will also encourage students to participate in the previously described annual design challenge.

Evaluation: We will assess the educational efficacy of this by assessing student understanding and growth. As an ultimate marker of success, we will count student-authored research papers submitted for peer-review. We will also track student career success, including first job placement and graduate program enrollment.

9. Results from Prior NSF Support

1: (a) CNS-1657602 R. LiKamWa PI, $174,950 2/15/2017–1/31/2019; (b) CRII: CSR: System Support for Reactive Sensor Operation for Efficiency and performance; (c) **Intellectual merit:** Currently studying the configuration latency of imaging and other sensing operations. **Broader impact:** The project produces open-source timing models for broad accessibility of data. (d) Publications: [12], [13]; (e) Open- source experimental setups, linked in publications [12], [13]; (f) N/A

2 [Recommended for funding at time of submission]: (a) SHF- S. Jayasuriya, PI, R. LiKamWa, Co-PI $332,999; (b) SHF: Small: Collaborative Research: Software-Defined Imaging for Energy-Efficient Visual Computing; (c) **Intellectual merit:** Novel CMOS sensor design for and software support for visual computing. **Broader impact:** (d) N/A (e) N/A (f) N/A

Relationship of (2) to the proposed project: While (2) will create new ***CMOS image sensor design*** for software-defined imaging (and operating system support to control the new sensor), this proposed project seeks to adapt computing systems from the ***sensor interface***, allowing the use of existing commercial sensors for high resolution visual computing pipelines. Furthermore, this proposal focuses on enabling a simultaneous variety of spatiotemporal resolution "rhythms", including differing timing intervals. This requires distinctly separate forms of architecture and runtime support to enable consistent memory access with proper representation.